Hewlett Packard Enterprise

HPE FlexFabric 5944 & 5945 Switch Series

Network Management and Monitoring Configuration Guide

Part number: 5200-7961
Software version: Release 6622
Document version: 6W100-20210430

© Copyright 2021 Hewlett Packard Enterprise Development LP

The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Links to third-party websites take you outside the Hewlett Packard Enterprise website. Hewlett Packard Enterprise has no control over and is not responsible for information outside the Hewlett Packard Enterprise website.

Acknowledgments

Intel®, Itanium®, Pentium®, Intel Inside®, and the Intel Inside logo are trademarks of Intel Corporation in the United States and other countries.

Microsoft® and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Adobe® and Acrobat® are trademarks of Adobe Systems Incorporated.

Java and Oracle are registered trademarks of Oracle and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

Contents

Using ping, tracert, and system debugging 1
Ping 1
About ping 1
Using a ping command to test network connectivity 1
Example: Using the ping utility 2
Tracert 3
About tracert 3
Prerequisites 3
Using a tracert command to identify failed or all nodes in a path 4
Example: Using the tracert utility 4
System debugging 5
About system debugging 5
Debugging a feature module 6
Configuring NQA 7
About NQA 7
NQA operating mechanism 7
Collaboration with Track 7
Threshold monitoring 8
NQA templates 8
NQA tasks at a glance 8
Configuring the NQA server 9
Enabling the NQA client 9
Configuring NQA operations on the NQA client 9
NQA operations tasks at a glance 9
Configuring the ICMP echo operation 10
Configuring the ICMP jitter operation 11
Configuring the DHCP operation 12
Configuring the DNS operation 13
Configuring the FTP operation 14
Configuring the HTTP operation 15
Configuring the UDP jitter operation 16
Configuring the SNMP operation 18
Configuring the TCP operation 18
Configuring the UDP echo operation 19
Configuring the UDP tracert operation 20
Configuring the voice operation 22
Configuring the DLSw operation 24
Configuring the path jitter operation 24
Configuring optional parameters for the NQA operation 26
Configuring the collaboration feature 27
Configuring threshold monitoring 27
Configuring the NQA statistics collection feature 29
Configuring the saving of NQA history records 30
Scheduling the NQA operation on the NQA client 31
Configuring NQA templates on the NQA client 31
Restrictions and guidelines 31
NQA template tasks at a glance 31
Configuring the ICMP template 32
Configuring the DNS template 33
Configuring the TCP template 34
Configuring the TCP half open template 35
Configuring the UDP template 36
Configuring the HTTP template 38
Configuring the HTTPS template 39
Configuring the FTP template 41
Configuring the RADIUS template 42

Configuring the SSL template ····43
Configuring optional parameters for the NQA template ····44
Display and maintenance commands for NQA ····45
NQA configuration examples ····46
Example: Configuring the ICMP echo operation ····46
Example: Configuring the ICMP jitter operation ····47
Example: Configuring the DHCP operation ····50
Example: Configuring the DNS operation ····51
Example: Configuring the FTP operation ····52
Example: Configuring the HTTP operation ····53
Example: Configuring the UDP jitter operation ····54
Example: Configuring the SNMP operation ····57
Example: Configuring the TCP operation ····58
Example: Configuring the UDP echo operation ····59
Example: Configuring the UDP tracert operation ····61
Example: Configuring the voice operation ····62
Example: Configuring the DLSw operation ····65
Example: Configuring the path jitter operation ····66
Example: Configuring NQA collaboration ····68
Example: Configuring the ICMP template ····70
Example: Configuring the DNS template ····71
Example: Configuring the TCP template ····72
Example: Configuring the TCP half open template ····72
Example: Configuring the UDP template ····73
Example: Configuring the HTTP template ····74
Example: Configuring the HTTPS template ····75
Example: Configuring the FTP template ····75
Example: Configuring the RADIUS template ····76
Example: Configuring the SSL template ····77
Configuring NTP ····79
About NTP ····79
NTP application scenarios ····79
NTP working mechanism ····79
NTP architecture ····80
NTP association modes ····81
NTP security ····82
NTP for MPLS L3VPN instances ····83
Protocols and standards ····84
Restrictions and guidelines: NTP configuration ····84
NTP tasks at a glance ····84
Enabling the NTP service ····85
Configuring NTP association mode ····85
Configuring NTP in client/server mode ····85
Configuring NTP in symmetric active/passive mode ····86
Configuring NTP in broadcast mode ····86
Configuring NTP in multicast mode ····87
Configuring the local clock as the reference source ····88
Configuring access control rights ····88
Configuring NTP authentication ····89
Configuring NTP authentication in client/server mode ····89
Configuring NTP authentication in symmetric active/passive mode ····90
Configuring NTP authentication in broadcast mode ····92
Configuring NTP authentication in multicast mode ····93
Controlling NTP message sending and receiving ····95
Specifying a source address for NTP messages ····95
Disabling an interface from receiving NTP messages ····96
Configuring the maximum number of dynamic associations ····96
Setting a DSCP value for NTP packets ····97
Specifying the NTP time-offset thresholds for log and trap outputs ····97
Display and maintenance commands for NTP ····97
NTP configuration examples ····98

Example: Configuring NTP client/server association mode ····98
Example: Configuring IPv6 NTP client/server association mode ····99
Example: Configuring NTP symmetric active/passive association mode····100
Example: Configuring IPv6 NTP symmetric active/passive association mode ····102
Example: Configuring NTP broadcast association mode····103
Example: Configuring NTP multicast association mode ····105
Example: Configuring IPv6 NTP multicast association mode ····108
Example: Configuring NTP authentication in client/server association mode ····111
Example: Configuring NTP authentication in broadcast association mode····112
Example: Configuring MPLS L3VPN network time synchronization in client/server mode ····115
Example: Configuring MPLS L3VPN network time synchronization in symmetric active/passive mode116
Configuring SNTP ····119
About SNTP ····119
SNTP working mode ····119
Protocols and standards ····119
Restrictions and guidelines: SNTP configuration····119
SNTP tasks at a glance····119
Enabling the SNTP service ····119
Specifying an NTP server for the device····120
Configuring SNTP authentication····120
Specifying the SNTP time-offset thresholds for log and trap outputs····121
Display and maintenance commands for SNTP ····121
SNTP configuration examples ····122
Example: Configuring SNTP ····122
Configuring PTP ····124
About PTP····124
Basic concepts····124
Grandmaster clock selection and master-member/subordinate relationship establishment ····126
Optimal domain selection····127
Synchronization mechanism ····127
Protocols and standards ····129
Restrictions: Hardware compatibility with PTP····129
Restrictions and guidelines: PTP configuration····129
PTP tasks at a glance ····129
Configuring PTP (IEEE 1588 version 2)····129
Configuring PTP (IEEE 802.1AS) ····130
Configuring PTP (SMPTE ST 2059-2) ····131
Configuring PTP (AES67-2015)····132
Specifying PTP for obtaining the time····133
Creating a PTP instance ····133
Specifying a PTP profile····133
Configuring clock nodes····134
Specifying a clock node type····134
Configuring an OC to operate only as a member clock ····134
Specifying a PTP domain····135
Enabling PTP globally····135
Enabling PTP on a port····135
Configuring PTP ports····136
Configuring the role of a PTP port····136
Configuring the mode for carrying timestamps ····137
Specifying a delay measurement mechanism for a BC or an OC····137
Configuring one of the ports on a TC+OC clock as an OC-type port ····138
Configuring PTP message transmission and receipt ····138
Setting the interval for sending announce messages and the timeout multiplier for receiving announce messages····138
Setting the interval for sending Pdelay_Req messages····139
Setting the interval for sending Sync messages ····139
Setting the minimum interval for sending Delay_Req messages····140
Configuring parameters for PTP messages····141
Specifying the IPv4 UDP transport protocol for PTP messages ····141

Configuring a source IP address for multicast PTP message transmission over IPv4 UDP 141
Configuring a destination IP address for unicast PTP message transmission over IPv4 UDP 142
Configuring the destination MAC address for non-Pdelay messages 142
Setting a DSCP value for PTP messages transmitted over IPv4 UDP 143
Specifying a VLAN tag for PTP messages 143
Adjusting and correcting clock synchronization 144
Setting the delay correction value 144
Setting the cumulative offset between the UTC and TAI 144
Setting the correction date of the UTC 145
Configuring a priority for a clock 145
Configuring **the PTP time locking and unlocking thresholds** 145
Display and maintenance commands for PTP 146
PTP configuration examples 147
Example: Configuring PTP (IEEE 1588 version 2, IEEE 802.3/Ethernet transport, multicast transmission) 147
Example: Configuring PTP (IEEE 1588 version 2, IPv4 UDP transport, multicast transmission) 150
Example: Configuring PTP (IEEE 1588 version 2, IPv4 UDP transport, unicast transmission) 153
Example: Configuring PTP (IEEE 802.1AS, IEEE 802.3/Ethernet transport, multicast transmission) 156
Example: Configuring PTP (SMPTE ST 2059-2, IPv4 UDP transport, multicast transmission) 159
Example: Configuring PTP (SMPTE ST 2059-2, IPv4 UDP transport, unicast transmission) 163
Example: Configuring PTP (AES67-2015, IPv4 UDP transport, multicast transmission) 166
Configuring SNMP 170
About SNMP 170
SNMP framework 170
MIB and view-based MIB access control 170
SNMP operations 171
Protocol versions 171
Access control modes 171
FIPS compliance 171
SNMP tasks at a glance 172
Enabling the SNMP agent 172
Enabling SNMP versions 172
Configuring SNMP common parameters 173
Configuring an SNMPv1 or SNMPv2c community 174
About configuring an SNMPv1 or SNMPv2c community 174
Restrictions and guidelines for configuring an SNMPv1 or SNMPv2c community 174
Configuring an SNMPv1/v2c community by a community name 174
Configuring an SNMPv1/v2c community by creating an SNMPv1/v2c user 175
Configuring an SNMPv3 group and user 175
Restrictions and guidelines for configuring an SNMPv3 group and user 175
Configuring an SNMPv3 group and user in non-FIPS mode 176
Configuring an SNMPv3 group and user in FIPS mode 176
Configuring SNMP notifications 177
About SNMP notifications 177
Enabling SNMP notifications 177
Configuring parameters for sending SNMP notifications 178
Examining the system configuration for changes 180
Configuring SNMP logging 180
Display and maintenance commands for SNMP 181
SNMP configuration examples 181
Example: Configuring SNMPv1/SNMPv2c 181
Example: Configuring SNMPv3 183
Configuring RMON 186
About RMON 186
RMON working mechanism 186
RMON groups 186
Sample types for the alarm group and the private alarm group 188
Protocols and standards 188
Configuring the RMON statistics function 188
About the RMON statistics function 188

Creating an RMON Ethernet statistics entry ····· 188
Creating an RMON history control entry ····· 188
Configuring the RMON alarm function ····· 189
Display and maintenance commands for RMON ····· 190
RMON configuration examples ····· 191
Example: Configuring the Ethernet statistics function ····· 191
Example: Configuring the history statistics function ····· 191
Example: Configuring the alarm function ····· 192
Configuring the Event MIB ····· 195
About the Event MIB ····· 195
Trigger ····· 195
Monitored objects ····· 195
Trigger test ····· 195
Event actions ····· 196
Object list ····· 196
Object owner ····· 197
Restrictions and guidelines: Event MIB configuration ····· 197
Event MIB tasks at a glance ····· 197
Prerequisites for configuring the Event MIB ····· 197
Configuring the Event MIB global sampling parameters ····· 198
Configuring Event MIB object lists ····· 198
Configuring an event ····· 198
Creating an event ····· 198
Configuring a set action for an event ····· 199
Configuring a notification action for an event ····· 199
Enabling the event ····· 200
Configuring a trigger ····· 200
Creating a trigger and configuring its basic parameters ····· 200
Configuring a Boolean trigger test ····· 201
Configuring an existence trigger test ····· 201
Configuring a threshold trigger test ····· 202
Enabling trigger sampling ····· 203
Enabling SNMP notifications for the Event MIB module ····· 203
Display and maintenance commands for Event MIB ····· 204
Event MIB configuration examples ····· 204
Example: Configuring an existence trigger test ····· 204
Example: Configuring a Boolean trigger test ····· 206
Example: Configuring a threshold trigger test ····· 209
Configuring NETCONF ····· 212
About NETCONF ····· 212
NETCONF structure ····· 212
NETCONF message format ····· 212
How to use NETCONF ····· 214
Protocols and standards ····· 214
FIPS compliance ····· 214
NETCONF tasks at a glance ····· 214
Establishing a NETCONF session ····· 215
Restrictions and guidelines for NETCONF session establishment ····· 215
Setting NETCONF session attributes ····· 215
Establishing NETCONF over SOAP sessions ····· 217
Establishing NETCONF over SSH sessions ····· 218
Establishing NETCONF over Telnet or NETCONF over console sessions ····· 218
Exchanging capabilities ····· 219
Retrieving device configuration information ····· 220
Restrictions and guidelines for device configuration retrieval ····· 220
Retrieving device configuration and state information ····· 220
Retrieving non-default settings ····· 222
Retrieving NETCONF information ····· 223
Retrieving YANG file content ····· 224
Retrieving NETCONF session information ····· 224

Example: Retrieving a data entry for the interface table ····225
Example: Retrieving non-default configuration data ····226
Example: Retrieving syslog configuration data ····227
Example: Retrieving NETCONF session information····228
Filtering data ····229
About data filtering ····229
Restrictions and guidelines for data filtering ····229
Table-based filtering····230
Column-based filtering ····230
Example: Filtering data with regular expression match····233
Example: Filtering data by conditional match····234
Locking or unlocking the running configuration····235
About configuration locking and unlocking····235
Restrictions and guidelines for configuration locking and unlocking ····235
Locking the running configuration ····235
Unlocking the running configuration····236
Example: Locking the running configuration ····236
Modifying the configuration ····237
About the <edit-config> operation ····237
Procedure····237
Example: Modifying the configuration ····238
Saving the running configuration ····239
About the <save> operation ····239
Restrictions and guidelines ····239
Procedure····239
Example: Saving the running configuration····240
Loading the configuration····241
About the <load> operation····241
Restrictions and guidelines ····241
Procedure····241
Rolling back the configuration ····242
Restrictions and guidelines ····242
Rolling back the configuration based on a configuration file ····242
Rolling back the configuration based on a rollback point ····242
Enabling preprovisioning····246
Performing CLI operations through NETCONF····247
About CLI operations through NETCONF····247
Restrictions and guidelines ····247
Procedure····247
Example: Performing CLI operations ····248
Subscribing to events····249
About event subscription····249
Restrictions and guidelines ····249
Subscribing to syslog events····249
Subscribing to events monitored by NETCONF····250
Subscribing to events reported by modules ····251
Canceling an event subscription ····252
Example: Subscribing to syslog events····253
Terminating NETCONF sessions····254
About NETCONF session termination ····254
Procedure····254
Example: Terminating another NETCONF session ····254
Returning to the CLI····255
Supported NETCONF operations ····256
action····256
CLI····256
close-session ····257
edit-config: create····257
edit-config: delete····258
edit-config: merge ····258
edit-config: remove····258

edit-config: replace ... 259
edit-config: *test-option* ... 259
edit-config: default-operation ... 260
edit-config: error-option ... 261
edit-config: incremental ... 262
get ... 262
get-bulk ... 263
get-bulk-config ... 263
get-config ... 264
get-sessions ... 264
kill-session ... 264
load ... 265
lock ... 265
rollback ... 265
save ... 266
unlock ... 266
Configuring Puppet ... 267
About Puppet ... 267
Puppet network framework ... 267
Puppet resources ... 268
Restrictions and guidelines: Puppet configuration ... 268
Prerequisites for Puppet ... 268
Starting Puppet ... 269
Configuring resources ... 269
Configuring a Puppet agent ... 269
Signing a certificate for the Puppet agent ... 269
Shutting down Puppet on the device ... 269
Puppet configuration examples ... 270
Example: Configuring Puppet ... 270
Puppet resources ... 271
netdev_device ... 271
netdev_interface ... 272
netdev_l2_interface ... 273
netdev_lagg ... 274
netdev_vlan ... 276
netdev_vsi ... 276
netdev_vte ... 277
netdev_vxlan ... 278
Configuring Chef ... 280
About Chef ... 280
Chef network framework ... 280
Chef resources ... 281
Chef configuration file ... 281
Restrictions and guidelines: Chef configuration ... 282
Prerequisites for Chef ... 283
Starting Chef ... 283
Configuring the Chef server ... 283
Configuring a workstation ... 283
Configuring a Chef client ... 283
Shutting down Chef ... 284
Chef configuration examples ... 284
Example: Configuring Chef ... 284
Chef resources ... 287
netdev_device ... 287
netdev_interface ... 287
netdev_l2_interface ... 289
netdev_lagg ... 290
netdev_vlan ... 291

netdev_vsi ····· 291
netdev_vte ····· 292
netdev_vxlan ····· 293
Configuring CWMP ····· 295
About CWMP ····· 295
CWMP network framework ····· 295
Basic CWMP functions ····· 295
How CWMP works ····· 297
Restrictions and guidelines: CWMP configuration ····· 299
CWMP tasks at a glance ····· 299
Enabling CWMP from the CLI ····· 300
Configuring ACS attributes ····· 300
About ACS attributes ····· 300
Configuring the preferred ACS attributes ····· 300
Configuring the default ACS attributes from the CLI ····· 301
Configuring CPE attributes ····· 302
About CPE attributes ····· 302
Specifying an SSL client policy for HTTPS connection to ACS ····· 302
Configuring ACS authentication parameters ····· 302
Configuring the provision code ····· 303
Configuring the CWMP connection interface ····· 303
Configuring autoconnect parameters ····· 304
Setting the close-wait timer ····· 305
Display and maintenance commands for CWMP ····· 305
CWMP configuration examples ····· 305
Example: Configuring CWMP ····· 305
Configuring EAA ····· 314
About EAA ····· 314
EAA framework ····· 314
Elements in a monitor policy ····· 315
EAA environment variables ····· 316
Configuring a user-defined EAA environment variable ····· 317
Configuring a monitor policy ····· 318
Restrictions and guidelines ····· 318
Configuring a monitor policy from the CLI ····· 318
Configuring a monitor policy by using Tcl ····· 319
Suspending monitor policies ····· 320
Display and maintenance commands for EAA ····· 321
EAA configuration examples ····· 321
Example: Configuring a CLI event monitor policy by using Tcl ····· 321
Example: Configuring a CLI event monitor policy from the CLI ····· 322
Example: Configuring a track event monitor policy from the CLI ····· 323
Example: Configuring a CLI event monitor policy with EAA environment variables from the CLI ····· 325
Monitoring and maintaining processes ····· 327
About monitoring and maintaining processes ····· 327
Process monitoring and maintenance tasks at a glance ····· 327
Starting or stopping a third-party process ····· 327
About third-party processes ····· 327
Starting a third-party process ····· 328
Stopping a third-party process ····· 328
Monitoring and maintaining processes ····· 328
Monitoring and maintaining user processes ····· 329
About monitoring and maintaining user processes ····· 329
Configuring core dump ····· 329
Display and maintenance commands for user processes ····· 330
Monitoring and maintaining kernel threads ····· 330
Configuring kernel thread deadloop detection ····· 330
Configuring kernel thread starvation detection ····· 331
Display and maintenance commands for kernel threads ····· 331

Configuring samplers 333
About sampler 333
Creating a sampler 333
Display and maintenance commands for a sampler 333
Samplers and IPv4 NetStream configuration examples 333
Example: Configuring samplers and IPv4 NetStream 333
Configuring port mirroring 335
About port mirroring 335
Terminology 335
Port mirroring classification 336
Local port mirroring 336
Layer 2 remote port mirroring 336
Layer 3 remote port mirroring 338
Restrictions and guidelines: Port mirroring configuration 339
Configuring local port mirroring 339
Restrictions and guidelines for local port mirroring configuration 339
Local port mirroring tasks at a glance 339
Creating a local mirroring group 340
Configuring mirroring sources 340
Configuring the monitor port 341
Configuring Layer 2 remote port mirroring 341
Restrictions and guidelines for Layer 2 remote port mirroring configuration 341
Layer 2 remote port mirroring with reflector port configuration task list 342
Layer 2 remote port mirroring with egress port configuration task list 342
Creating a remote destination group 342
Configuring the monitor port 343
Configuring the remote probe VLAN 343
Assigning the monitor port to the remote probe VLAN 344
Creating a remote source group 344
Configuring mirroring sources 344
Configuring the reflector port 345
Configuring the egress port 346
Configuring Layer 3 remote port mirroring (in tunnel mode) 347
Restrictions and guidelines for Layer 3 remote port mirroring configuration 347
Layer 3 remote port mirroring tasks at a glance 347
Prerequisites for Layer 3 remote port mirroring 347
Configuring local mirroring groups 348
Configuring mirroring sources 348
Configuring the monitor port 349
Configuring Layer 3 remote port mirroring (in ERSPAN mode) 350
Restrictions and guidelines for Layer 3 remote port mirroring in ERSPAN mode configuration 350
Layer 3 remote port mirroring tasks at a glance 350
Creating a local mirroring group on the source device 350
Configuring mirroring sources 350
Configuring the monitor port 351
Display and maintenance commands for port mirroring 352
Port mirroring configuration examples 352
Example: Configuring local port mirroring (in source port mode) 352
Example: Configuring local port mirroring (in source CPU mode) 353
Example: Configuring Layer 2 remote port mirroring (with reflector port) 355
Example: Configuring Layer 2 remote port mirroring (with egress port) 357
Example: Configuring Layer 3 remote port mirroring in tunnel mode 359
Example: Configuring Layer 3 remote port mirroring in ERSPAN mode 361
Configuring flow mirroring 364
About flow mirroring 364
Restrictions and guidelines: Flow mirroring configuration 364
Flow mirroring tasks at a glance 365
Configuring a traffic class 365
Configuring a traffic behavior 365

Configuring a QoS policy ····366
Applying a QoS policy ····366
Applying a QoS policy to an interface ····366
Applying a QoS policy to a VLAN ····367
Applying a QoS policy globally ····367
Applying a QoS policy to the control plane ····368
Flow mirroring configuration examples ····368
Example: Configuring flow mirroring ····368
Configuring NetStream ····370
About NetStream ····370
NetStream architecture ····370
NetStream flow aging ····371
NetStream data export ····372
NetStream filtering ····374
NetStream sampling ····374
Protocols and standards ····374
Restrictions: Hardware compatibility with NetStream ····374
Restrictions and guidelines: NetStream configuration ····374
NetStream tasks at a glance ····375
Enabling NetStream ····375
Configuring NetStream filtering ····375
Configuring NetStream sampling ····376
Configuring the NetStream data export format ····376
Configuring the refresh rate for NetStream version 9 or version 10 template ····378
Configuring VXLAN-aware NetStream ····378
Configuring NetStream flow aging ····378
Configuring periodical flow aging ····378
Configuring forced flow aging ····379
Configuring the NetStream data export ····379
Configuring the NetStream traditional data export ····379
Configuring the NetStream aggregation data export ····379
Display and maintenance commands for NetStream ····380
NetStream configuration examples ····381
Example: Configuring NetStream traditional data export ····381
Example: Configuring NetStream aggregation data export ····383
Configuring IPv6 NetStream ····386
About IPv6 NetStream ····386
IPv6 NetStream architecture ····386
IPv6 NetStream flow aging ····387
IPv6 NetStream data export ····388
IPv6 NetStream filtering ····389
IPv6 NetStream sampling ····389
Protocols and standards ····389
Restrictions: Hardware compatibility with IPv6 NetStream ····389
Restrictions and guidelines: IPv6 NetStream configuration ····389
IPv6 NetStream tasks at a glance ····390
Enabling IPv6 NetStream ····390
Configuring IPv6 NetStream filtering ····390
Configuring IPv6 NetStream sampling ····391
Configuring the IPv6 NetStream data export format ····391
Configuring the refresh rate for IPv6 NetStream version 9 or version 10 template ····393
Configuring IPv6 NetStream flow aging ····393
Configuring periodical flow aging ····393
Configuring forced flow aging ····393
Configuring the IPv6 NetStream data export ····394
Configuring the IPv6 NetStream traditional data export ····394
Configuring the IPv6 NetStream aggregation data export ····394
Display and maintenance commands for IPv6 NetStream ····395
IPv6 NetStream configuration examples ····396
Example: Configuring IPv6 NetStream traditional data export ····396

Example: Configuring IPv6 NetStream aggregation data export ····· 397
Configuring sFlow ····· 400
About sFlow ····· 400
Protocols and standards ····· 400
Restrictions and guidelines: sFlow configuration ····· 400
Configuring basic sFlow information ····· 401
Configuring flow sampling ····· 401
Configuring counter sampling ····· 402
Display and maintenance commands for sFlow ····· 402
sFlow configuration examples ····· 403
Example: Configuring sFlow ····· 403
Troubleshooting sFlow ····· 404
The remote sFlow collector cannot receive sFlow packets ····· 404
Configuring the information center ····· 406
About the information center ····· 406
Log types ····· 406
Log levels ····· 406
Log destinations ····· 407
Default output rules for logs ····· 407
Default output rules for diagnostic logs ····· 407
Default output rules for security logs ····· 407
Default output rules for hidden logs ····· 408
Default output rules for trace logs ····· 408
Log formats and field descriptions ····· 408
FIPS compliance ····· 411
Information center tasks at a glance ····· 411
Managing standard system logs ····· 411
Managing hidden logs ····· 412
Managing security logs ····· 412
Managing diagnostic logs ····· 412
Managing trace logs ····· 413
Enabling the information center ····· 413
Outputting logs to various destinations ····· 413
Outputting logs to the console ····· 413
Outputting logs to the monitor terminal ····· 414
Outputting logs to log hosts ····· 415
Outputting logs to the log buffer ····· 416
Saving logs to the log file ····· 416
Setting the minimum storage period ····· 417
About setting the minimum storage period ····· 417
Procedure ····· 418
Enabling synchronous information output ····· 418
Configuring log suppression ····· 418
Enabling duplicate log suppression ····· 418
Configuring log suppression for a module ····· 419
Disabling an interface from generating link up or link down logs ····· 419
Enabling SNMP notifications for system logs ····· 420
Managing security logs ····· 420
Saving security logs to the security log file ····· 420
Managing the security log file ····· 421
Saving diagnostic logs to the diagnostic log file ····· 422
Setting the maximum size of the trace log file ····· 422
Display and maintenance commands for information center ····· 423
Information center configuration examples ····· 423
Example: Outputting logs to the console ····· 423
Example: Outputting logs to a UNIX log host ····· 424
Example: Outputting logs to a Linux log host ····· 425
Configuring GOLD ····· 427
About GOLD ····· 427

Types of GOLD diagnostics ····427
GOLD diagnostic tests ····427
GOLD tasks at a glance ····427
Configuring monitoring diagnostics ····427
Configuring on-demand diagnostics ····428
Simulating diagnostic tests ····429
Configuring the log buffer size ····429
Display and maintenance commands for GOLD ····429
GOLD configuration examples ····430
Example: Configuring GOLD ····430
Configuring the packet capture ····432
About packet capture ····432
Packet capture modes ····432
Filter rule elements ····432
Building a capture filter rule ····433
Capture filter rule keywords ····433
Capture filter rule operators ····434
Capture filter rule expressions ····435
Building a display filter rule ····436
Display filter rule keywords ····437
Display filter rule operators ····438
Display filter rule expressions ····439
Restrictions and guidelines: Packet capture ····440
Configuring local packet capture ····440
Configuring remote packet capture ····440
Configuring feature image-based packet capture ····440
Restrictions and guidelines ····440
Prerequisites ····440
Saving captured packets to a file ····441
Displaying specific captured packets ····441
Stopping packet capture ····441
Displaying the contents in a packet file ····441
Display and maintenance commands for packet capture ····442
Packet capture configuration examples ····442
Example: Configuring remote packet capture ····442
Example: Configuring feature image-based packet capture ····443
Configuring VCF fabric ····447
About VCF fabric ····447
VCF fabric topology ····447
Neutron overview ····450
Automated VCF fabric deployment ····452
Process of automated VCF fabric deployment ····453
Template file ····453
VCF fabric task at a glance ····454
Configuring automated VCF fabric deployment ····454
Enabling VCF fabric topology discovery ····456
Configuring automated underlay network deployment ····456
Specify the template file for automated underlay network deployment ····456
Specifying the role of the device in the VCF fabric ····456
Configuring the device as a master spine node ····457
Pausing automated underlay network deployment ····457
Configuring automated overlay network deployment ····457
Restrictions and guidelines for automated overlay network deployment ····457
Automated overlay network deployment tasks at a glance ····458
Prerequisites for automated overlay network deployment ····458
Configuring parameters for the device to communicate with RabbitMQ servers ····458
Specifying the network type ····459
Enabling L2 agent ····460
Enabling L3 agent ····460
Configuring the border node ····461

Enabling local proxy ARP 461
Configuring the MAC address of VSI interfaces 462
Display and maintenance commands for VCF fabric 462
Using Ansible for automated configuration management 464
About Ansible 464
Ansible network architecture 464
How Ansible works 464
Restrictions and guidelines 464
Configuring the device for management with Ansible 465
Device setup examples for management with Ansible 465
Example: Setting up the device for management with Ansible 465
Document conventions and icons 467
Conventions 467
Network topology icons 468
Support and other resources 469
Accessing Hewlett Packard Enterprise Support 469
Accessing updates 469
Websites 470
Customer self repair 470
Remote support 470
Documentation feedback 470
Index 472

Using ping, tracert, and system debugging

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

Ping

About ping

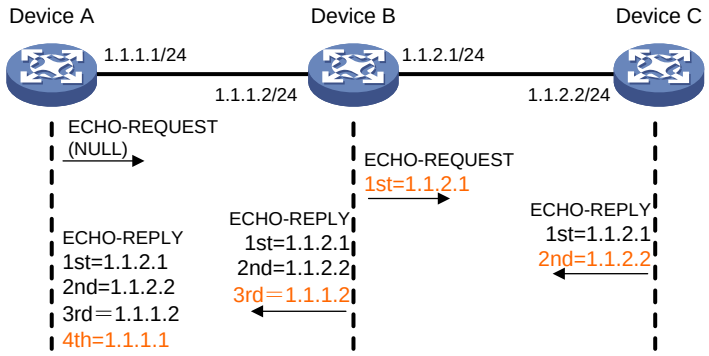
Use the ping utility to determine if an address is reachable.

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.

You can use the **`ping -r`** command to display the routers through which ICMP echo requests have passed. 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 empty.
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.

Figure 1 Ping operation



Using a ping command to test network connectivity

Perform the following tasks in any view:

- Determine if an IPv4 address is reachable.

ping [**ip**] [**-a** *source-ip* | **-c** *count* | **-f** | **-h** *ttl* | **-i** *interface-type interface-number* | **-m** *interval* | **-n** | **-p** *pad* | **-q** | **-r** | **-s** *packet-size* | **-t** *timeout* | **-tos** *tos* | **-v** | **-vpn-instance** *vpn-instance-name*] * *host*

Increase the timeout time (indicated by the **-t** keyword) on a low-speed network.

- Determine if an IPv6 address is reachable.

  ping ipv6 [**-a** *source-ipv6* | **-c** *count* | **-i** *interface-type interface-number* | **-m** *interval* | **-q** | **-s** *packet-size* | **-t** *timeout* | **-tc** *traffic-class* | **-v** | **-vpn-instance** *vpn-instance-name*] * *host*

  Increase the timeout time (indicated by the **-t** keyword) on a low-speed network.

- Determine if a node in an MPLS network is reachable.

  ping mpls ipv4

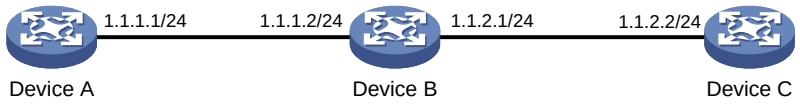
  For more information about this command, see *MPLS Command Reference*.

Example: Using the ping utility

Network configuration

As shown in Figure 2, determine if Device A and Device C can reach each other.

Figure 2 Network diagram



Procedure

Test the 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
56 bytes from 1.1.2.2: icmp_seq=0 ttl=254 time=2.137 ms
56 bytes from 1.1.2.2: icmp_seq=1 ttl=254 time=2.051 ms
56 bytes from 1.1.2.2: icmp_seq=2 ttl=254 time=1.996 ms
56 bytes from 1.1.2.2: icmp_seq=3 ttl=254 time=1.963 ms
56 bytes from 1.1.2.2: icmp_seq=4 ttl=254 time=1.991 ms

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

The output shows the following information:

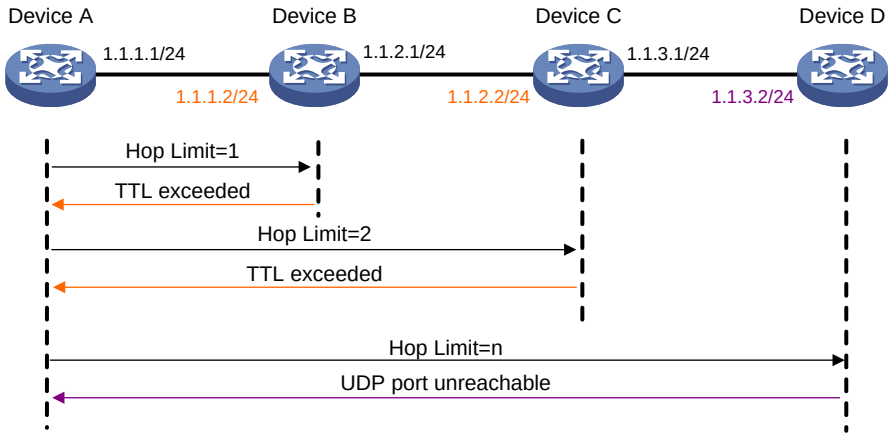
- Device A sends five ICMP packets to Device C and Device A receives five ICMP packets.
- No ICMP packet is lost.
- The route is reachable.

Tracert

About tracert

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

Figure 3 Tracert operation



Tracert uses received ICMP error messages to get the IP addresses of devices. Tracert works as shown in Figure 3:

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 determines that:
   - The packet has reached the destination device after receiving the port-unreachable ICMP message.
   - 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 HPE 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 HPE 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 HPE 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 HPE 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

Perform the following tasks in any view:

- Trace the route to an IPv4 destination.

  tracert [**-a** *source-ip* | **-f** *first-ttl* | **-m** *max-ttl* | **-p** *port* | **-q** *packet-number* | **-t** *tos* | **-vpn-instance** *vpn-instance-name* [**-resolve-as** { **global** | **none** | **vpn** }] | **-w** *timeout*] * *host*

- Trace the route to an IPv6 destination.

  tracert ipv6 [**-a** *source-ipv6* | **-f** *first-hop* | **-m** *max-hops* | **-p** *port* | **-q** *packet-number* | **-t** *traffic-class* | **-vpn-instance** *vpn-instance-name* [**-resolve-as** { **global** | **none** | **vpn** }] | **-w** *timeout*] * *host*

- Trace the route to a destination in an MPLS network.

  tracert mpls ipv4

  For more information about this command, see *MPLS Command Reference*.

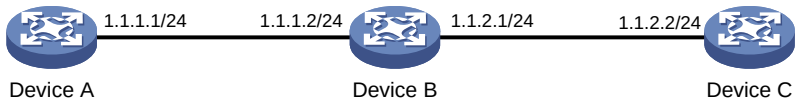
Example: Using the tracert utility

Network configuration

As shown in Figure 4, 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 4 Network diagram



Procedure

1. Configure IP addresses for the devices as shown in Figure 4.
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
```

3. 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. 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
```

   # Identify failed nodes.

```
[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 issue, 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

About system debugging

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

The following switches control the display of debugging information:

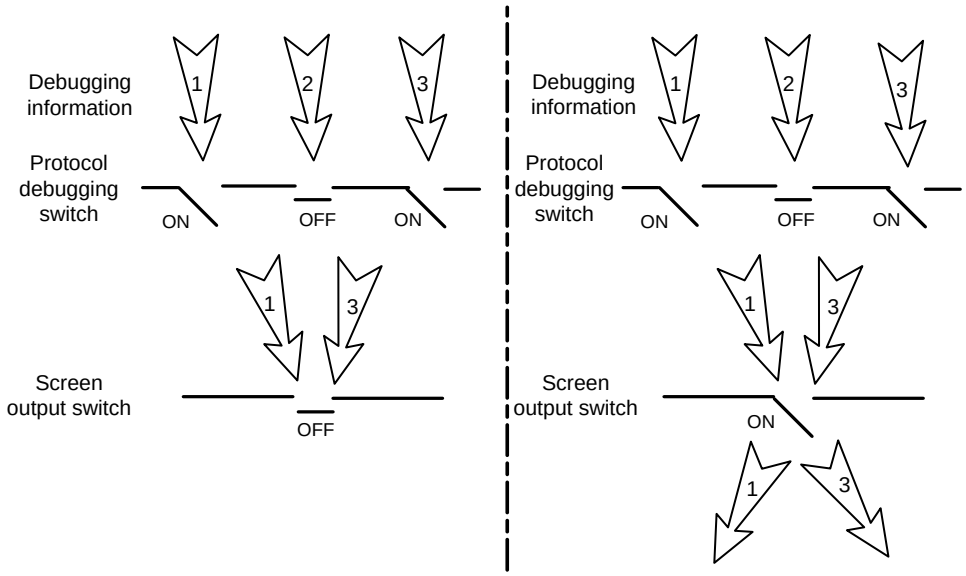
- **Module debugging switch**—Controls whether to generate the module-specific debugging information.

- **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 5, 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 5 Relationship between the module and screen output switch



Debugging a feature module

Restrictions and guidelines

Output from 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.

Procedure

1. Enable debugging for a module.

   debugging *module-name* [*option*]

   By default, debugging is disabled for all modules.

   This command is available in user view.

2. (Optional.) Display the enabled debugging features.

   display debugging [*module-name*]

   This command is available in any view.

Configuring NQA

About NQA

Network quality analyzer (NQA) allows you to measure network performance, verify the service levels for IP services and applications, and troubleshoot network problems.

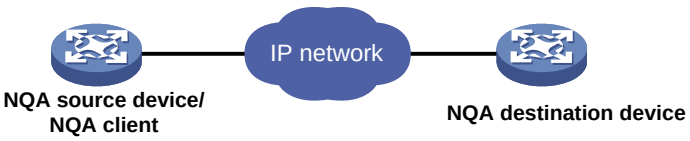
NQA operating mechanism

An NQA operation contains a set of parameters such as the operation type, destination IP address, and port number to define how the operation is performed. Each NQA operation is identified by the combination of the administrator name and the operation tag. You can configure the NQA client to run the operations at scheduled time periods.

As shown in Figure 6, the NQA source device (NQA client) sends data to the NQA destination device by simulating IP services and applications to measure network performance.

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 6 Network diagram



After starting an NQA operation, the NQA client periodically performs the operation at the interval specified by using the **frequency** command.

You can set the number of probes the NQA client performs in an operation by using the **probe count** command. For the voice and path jitter operations, the NQA client performs only one probe per operation and the **probe count** command is not available.

Collaboration with Track

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.

The NQA + Track collaboration is available for the following application modules:

- VRRP.
- Static routing.
- Policy-based routing.
- Smart Link

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 to invalid according to a predefined action.

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

Threshold monitoring

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

Table 1 describes the relationships between performance metrics and NQA operation types.

Table 1 Performance metrics and NQA operation types

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

NQA templates

An NQA template is a set of parameters (such as destination address and port number) that defines how an NQA operation is performed. Features can use the NQA template to collect statistics.

You can create multiple NQA templates on the NQA client. Each template must be identified by a unique template name.

NQA tasks at a glance

To configure NQA, perform the following tasks:

1. Configuring the NQA server

   Perform this task on the destination device before you configure the TCP, UDP echo, UDP jitter, and voice operations.
2. Enabling the NQA client
3. Configuring NQA operations or NQA templates

   Choose the following tasks as needed:
   - Configuring NQA operations on the NQA client
   - Configuring NQA templates on the NQA client

After you configure an NQA operation, you can schedule the NQA client to run the NQA operation.

An NQA template does not run immediately after it is configured. The template creates and run the NQA operation only when it is required by the feature to which the template is applied.

Configuring the NQA server

Restrictions and guidelines

To perform TCP, UDP echo, UDP jitter, and voice operations, you must configure 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.

Procedure

1. Enter system view.

   system-view

2. Enable the NQA server.

   nqa server enable

   By default, the NQA server is disabled.

3. Configure a TCP listening service.

   nqa server tcp-connect *ip-address port-number* [**vpn-instance** *vpn-instance-name*] [**tos** *tos*]

   This task is required for only TCP operations.

4. Configure a UDP listening service.

   nqa server udp-echo *ip-address port-number* [**vpn-instance** *vpn-instance-name*] [**tos** *tos*]

   This task is required for only UDP echo, UDP jitter, and voice operations.

Enabling the NQA client

1. Enter system view.

   system-view

2. Enable the NQA client.

   nqa agent enable

   By default, the NQA client is enabled.

   The NQA client configuration takes effect after you enable the NQA client.

Configuring NQA operations on the NQA client

NQA operations tasks at a glance

To configure NQA operations, perform the following tasks:

1. Configuring an NQA operation

- Configuring the ICMP echo operation
- Configuring the ICMP jitter operation
- Configuring the DHCP operation
- Configuring the DNS operation
- Configuring the FTP operation
- Configuring the HTTP operation
- Configuring the UDP jitter operation
- Configuring the SNMP operation
- Configuring the TCP operation
- Configuring the UDP echo operation
- Configuring the UDP tracert operation
- Configuring the voice operation
- Configuring the DLSw operation
- Configuring the path jitter operation

2. (Optional.) Configuring optional parameters for the NQA operation
3. (Optional.) Configuring the collaboration feature
4. (Optional.) Configuring threshold monitoring
5. (Optional.) Configuring the NQA statistics collection feature
6. (Optional.) Configuring the saving of NQA history records
7. Scheduling the NQA operation on the NQA client

Configuring the ICMP echo operation

About this task

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 sends an ICMP echo request to the destination device per probe.

Procedure

1. Enter system view.
   system-view
2. Create an NQA operation and enter NQA operation view.
   nqa entry *admin-name operation-tag*
3. Specify the ICMP echo type and enter its view.
   type icmp-echo
4. Specify the destination IP address for ICMP echo requests.
   IPv4:
   destination ip *ip-address*
   IPv6:
   destination ipv6 *ipv6-address*
   By default, no destination IP address is specified.
5. Specify the source IP address for ICMP echo requests. Choose one option as needed:
   - Use the IP address of the specified interface as the source IP address.

source interface *interface-type interface-number*

By default, the source IP address of ICMP echo requests is the primary IP address of their output interface.

The specified source interface must be up.

- Specify the source IPv4 address.

  source ip *ip-address*

  By default, the source IPv4 address of ICMP echo requests is the primary IPv4 address of their output interface.

  The specified source IPv4 address must be the IPv4 address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.

- Specify the source IPv6 address.

  source ipv6 *ipv6-address*

  By default, the source IPv6 address of ICMP echo requests is the primary IPv6 address of their output interface.

  The specified source IPv6 address must be the IPv6 address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.

6. Specify the output interface or the next hop IP address for ICMP echo requests. Choose one option as needed:
   - Specify the output interface for ICMP echo requests.

     out interface *interface-type interface-number*

     By default, the output interface for ICMP echo requests is not specified. The NQA client determines the output interface based on the routing table lookup.

   - Specify the next hop IPv4 address.

     next-hop ip *ip-address*

     By default, no next hop IPv4 address is specified.

   - Specify the next hop IPv6 address.

     next-hop ipv6 *ipv6-address*

     By default, no next hop IPv6 address is specified.
7. (Optional.) Set the payload size for each ICMP echo request.

   data-size *size*

   The default payload size is 100 bytes.
8. (Optional.) Specify the payload fill string for ICMP echo requests.

   data-fill *string*

   The default payload fill string is the hexadecimal string 00010203040506070809.

Configuring the ICMP jitter operation

About this task

The ICMP jitter operation measures unidirectional and bidirectional jitters. The operation result helps you to determine whether the network can carry jitter-sensitive services such as real-time voice and video services.

The ICMP jitter operation works as follows:

1. The NQA client sends ICMP packets to the destination device.
2. The destination device time stamps each packet 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 timestamps.

The ICMP jitter operation sends a number of ICMP packets to the destination device per probe. The number of packets to send is determined by using the **probe packet-number** command.

Restrictions and guidelines

The **display nqa history** command does not display the results or statistics of the ICMP jitter operation. To view the results or statistics of the operation, use the **display nqa result** or **display nqa statistics** command.

Before starting the operation, make sure the network devices are time synchronized by using NTP. For more information about NTP, see "Configuring NTP."

Procedure

1. Enter system view.
   system-view
2. Create an NQA operation and enter NQA operation view.
   nqa entry *admin-name operation-tag*
3. Specify the ICMP jitter type and enter its view.
   type icmp-jitter
4. Specify the destination IP address for ICMP packets.
   destination ip *ip-address*
   By default, no destination IP address is specified.
5. Set the number of ICMP packets sent per probe.
   probe packet-number *packet-number*
   The default setting is 10.
6. Set the interval for sending ICMP packets.
   probe packet-interval *interval*
   The default setting is 20 milliseconds.
7. Specify how long the NQA client waits for a response from the server before it regards the response times out.
   probe packet-timeout *timeout*
   The default setting is 3000 milliseconds.
8. Specify the source IP address for ICMP packets.
   source ip *ip-address*
   By default, the source IP address of ICMP packets is the primary IP address of their output interface.
   The source IP address must be the IP address of a local interface, and the interface must be up. Otherwise, no ICMP packets can be sent out.

Configuring the DHCP operation

About this task

The DHCP operation measures whether or not the DHCP server can respond to client requests. DHCP also measures the amount of time it takes the NQA client to obtain an IP address from a DHCP server.

The NQA client simulates the DHCP relay agent to forward DHCP requests for IP address acquisition from the DHCP server. The interface that performs the DHCP operation does not change its IP address. When the DHCP operation completes, the NQA client sends a packet to release the obtained IP address.

The DHCP operation acquires an IP address from the DHCP server per probe.

Procedure

1. Enter system view.
   system-view
2. Create an NQA operation and enter NQA operation view.
   nqa entry *admin-name operation-tag*
3. Specify the DHCP type and enter its view.
   type dhcp
4. Specify the IP address of the DHCP server as the destination IP address of DHCP packets.
   destination ip *ip-address*

   By default, no destination IP address is specified.
5. Specify the output interface for DHCP request packets.
   out interface *interface-type interface-number*

   By default, the NQA client determines the output interface based on the routing table lookup.
6. Specify the source IP address of DHCP request packets.
   source ip *ip-address*

   By default, the source IP address of DHCP request packets is the primary IP address of their output interface.

   The specified source IP address must be the IP address of a local interface, and the local interface must be up. Otherwise, no probe packets can be sent out.

Configuring the DNS operation

About this task

The DNS operation simulates domain name resolution, and it measures the time for the NQA client to resolve a domain name into an IP address through a DNS server. The obtained DNS entry is not saved.

The DNS operation resolves a domain name into an IP address per probe.

Procedure

1. Enter system view.
   system-view
2. Create an NQA operation and enter NQA operation view.
   nqa entry *admin-name operation-tag*
3. Specify the DNS type and enter its view.
   type dns
4. Specify the IP address of the DNS server as the destination IP address of DNS packets.
   destination ip *ip-address*

   By default, no destination IP address is specified.
5. Specify the domain name to be translated.
   resolve-target *domain-name*

   By default, no domain name is specified.

Configuring the FTP operation

About this task

The FTP operation measures the time for the NQA client to transfer a file to or download a file from an FTP server.

The FTP operation uploads or downloads a file from an FTP server per probe.

Restrictions and guidelines

To upload (**put**) a file to the FTP server, use the **filename** command to specify the name of the file you want to upload. The file must exist on the NQA client.

To download (**get**) a file from the FTP server, include the name of the file you want to download in the **url** command. The file must exist 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 because of the amount of network bandwidth it occupies.

Procedure

1. Enter system view.
   system-view
2. Create an NQA operation and enter NQA operation view.
   nqa entry *admin-name operation-tag*
3. Specify the FTP type and enter its view.
   type ftp
4. Specify an FTP login username.
   username *username*

   By default, no FTP login username is specified.
5. Specify an FTP login password.
   password { **cipher** | **simple** } *string*

   By default, no FTP login password is specified.
6. Specify the source IP address for FTP request packets.
   source ip *ip-address*

   By default, the source IP address of FTP request packets is the primary IP address of their output interface.

   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.
7. Set the data transmission mode.
   mode { **active** | **passive** }

   The default mode is **active**.
8. Specify the FTP operation type.
   operation { **get** | **put** }

   The default FTP operation type is **get**.
9. Specify the destination URL for the FTP operation.
   url *url*

   By default, no destination URL is specified for an FTP operation.

   Enter the URL in one of the following formats:
   - ftp://*host*/*filename*.

- ftp://*host:port*/*filename*.

  The *filename* argument is required only for the **get** operation.

10. Specify the name of the file to be uploaded.

    filename *file-name*

    By default, no file is specified.

    This task is required only for the **put** operation.

    The configuration does not take effect for the **get** operation.

Configuring the HTTP operation

About this task

The HTTP operation measures the time for the NQA client to obtain responses from an HTTP server.

The HTTP operation supports the following operation types:

- **Get**—Retrieves data such as a Web page from the HTTP server.
- **Post**—Sends data to the HTTP server for processing.
- **Raw**—Sends a user-defined HTTP request to the HTTP server. You must manually configure the content of the HTTP request to be sent.

The HTTP operation completes the operation of the specified type per probe.

Procedure

1. Enter system view.

   system-view

2. Create an NQA operation and enter NQA operation view.

   nqa entry *admin-name operation-tag*

3. Specify the HTTP type and enter its view.

   type http

4. Specify the destination URL for the HTTP operation.

   url *url*

   By default, no destination URL is specified for an HTTP operation.

   Enter the URL in one of the following formats:

   - http://*host*/*resource*
   - http://*host:port/resource*

5. Specify an HTTP login username.

   username *username*

   By default, no HTTP login username is specified.

6. Specify an HTTP login password.

   password { **cipher** | **simple** } *string*

   By default, no HTTP login password is specified.

7. Specify the HTTP version.

   version { **v1.0** | **v1.1** }

   By default, HTTP 1.0 is used.

8. Specify the HTTP operation type.

   operation { **get** | **post** | **raw** }

   The default HTTP operation type is **get**.

If you set the operation type to **raw**, the client pads the content configured in raw request view to the HTTP request to send to the HTTP server.

9. Configure the HTTP raw request.
   a. Enter raw request view.

      raw-request

      Every time you enter raw request view, the previously configured raw request content is cleared.

   b. Enter or paste the request content.

      By default, no request content is configured.

      To ensure successful operations, make sure the request content does not contain command aliases configured by using the **alias** command. For more information about the **alias** command, see CLI commands in *Fundamentals Command Reference*.

   c. Save the input and return to HTTP operation view:

      quit

   This step is required only when the operation type is set to **raw**.
10. Specify the source IP address for the HTTP packets.

    source ip *ip-address*

    By default, the source IP address of HTTP packets is the primary IP address of their output interface.

    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.

Configuring the UDP jitter operation

About this task

The UDP jitter operation measures unidirectional and bidirectional jitters. The operation result helps you determine whether the network can carry jitter-sensitive services such as real-time voice and video services.

The UDP jitter operation works as follows:

1. The NQA client sends UDP packets to the destination port.
2. The destination device time stamps each packet 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 timestamps.

The UDP jitter operation sends a number of UDP packets to the destination device per probe. The number of packets to send is determined by using the **probe packet-number** command.

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."

Restrictions and guidelines

To ensure successful UDP jitter operations and avoid affecting existing services, do not perform the operations on well-known ports from 1 to 1023.

The **display nqa history** command does not display the results or statistics of the UDP jitter operation. To view the results or statistics of the UDP jitter operation, use the **display nqa result** or **display nqa statistics** command.

Before starting the operation, make sure the network devices are time synchronized by using NTP. For more information about NTP, see "Configuring NTP."

Procedure

1. Enter system view.
   system-view
2. Create an NQA operation and enter NQA operation view.
   nqa entry *admin-name operation-tag*
3. Specify the UDP jitter type and enter its view.
   type udp-jitter
4. Specify the destination IP address for UDP packets.
   destination ip *ip-address*

   By default, no destination IP address is specified.

   The destination IP address must be the same as the IP address of the UDP listening service configured on the NQA server. To configure a UDP listening service on the server, use the **nqa server udp-echo** command.
5. Specify the destination port number for UDP packets.
   destination port *port-number*

   By default, no destination port number is specified.

   The destination port number must be the same as the port number of the UDP listening service configured on the NQA server. To configure a UDP listening service on the server, use the **nqa server udp-echo** command.
6. Specify the source IP address for UDP packets.
   source ip *ip-address*

   By default, the source IP address of UDP packets is the primary IP address of their output interface.

   The source IP address must be the IP address of a local interface, and the interface must be up. Otherwise, no UDP packets can be sent out.
7. Specify the source port number for UDP packets.
   source port *port-number*

   By default, the NQA client randomly picks an unused port as the source port when the operation starts.
8. Set the number of UDP packets sent per probe.
   probe packet-number *packet-number*

   The default setting is 10.
9. Set the interval for sending UDP packets.
   probe packet-interval *interval*

   The default setting is 20 milliseconds.
10. Specify how long the NQA client waits for a response from the server before it regards the response times out.
    probe packet-timeout *timeout*

    The default setting is 3000 milliseconds.
11. (Optional.) Set the payload size for each UDP packet.
    data-size *size*

    The default payload size is 100 bytes.
12. (Optional.) Specify the payload fill string for UDP packets.
    data-fill *string*

    The default payload fill string is the hexadecimal string 00010203040506070809.

Configuring the SNMP operation

About this task

The SNMP operation tests whether the SNMP service is available on an SNMP agent.

The SNMP operation sends one SNMPv1 packet, one SNMPv2c packet, and one SNMPv3 packet to the SNMP agent per probe.

Procedure

1. Enter system view.

   system-view
2. Create an NQA operation and enter NQA operation view.

   nqa entry *admin-name operation-tag*
3. Specify the SNMP type and enter its view.

   type snmp
4. Specify the destination address for SNMP packets.

   destination ip *ip-address*

   By default, no destination IP address is specified.
5. Specify the source IP address for SNMP packets.

   source ip *ip-address*

   By default, the source IP address of SNMP packets is the primary IP address of their output interface.

   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.
6. Specify the source port number for SNMP packets.

   source port *port-number*

   By default, the NQA client randomly picks an unused port as the source port when the operation starts.
7. Specify the community name carried in the SNMPv1 and SNMPv2c packets.

   community read { **cipher** | **simple** } *community-name*

   By default, the SNMPv1 and SNMPv2c packets carry community name **public**.

   Make sure the specified community name is the same as the community name configured on the SNMP agent.

Configuring the TCP operation

About this task

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."

The TCP operation sets up a TCP connection per probe.

Procedure

1. Enter system view.

   system-view
2. Create an NQA operation and enter NQA operation view.

nqa entry *admin-name operation-tag*

3. Specify the TCP type and enter its view.

   type tcp

4. Specify the destination address for TCP packets.

   destination ip *ip-address*

   By default, no destination IP address is specified.

   The destination address must be the same as the IP address of the TCP listening service configured on the NQA server. To configure a TCP listening service on the server, use the **nqa server tcp-connect** command.

5. Specify the destination port for TCP packets.

   destination port *port-number*

   By default, no destination port number is configured.

   The destination port number must be the same as the port number of the TCP listening service configured on the NQA server. To configure a TCP listening service on the server, use the **nqa server tcp-connect** command.

6. Specify the source IP address for TCP packets.

   source ip *ip-address*

   By default, the source IP address of TCP packets is the primary IP address of their output interface.

   The source IP address must be the IP address of a local interface, and the interface must be up. Otherwise, no TCP packets can be sent out.

Configuring the UDP echo operation

About this task

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."

The UDP echo operation sends a UDP packet to the destination device per probe.

Procedure

1. Enter system view.

   system-view

2. Create an NQA operation and enter NQA operation view.

   nqa entry *admin-name operation-tag*

3. Specify the UDP echo type and enter its view.

   type udp-echo

4. Specify the destination address for UDP packets.

   destination ip *ip-address*

   By default, no destination IP address is specified.

   The destination address must be the same as the IP address of the UDP listening service configured on the NQA server. To configure a UDP listening service on the server, use the **nqa server udp-echo** command.

5. Specify the destination port number for UDP packets.

   destination port *port-number*

By default, no destination port number is specified.

The destination port number must be the same as the port number of the UDP listening service configured on the NQA server. To configure a UDP listening service on the server, use the **nqa server udp-echo** command.

6. Specify the source IP address for UDP packets.

   source ip *ip-address*

   By default, the source IP address of UDP packets is the primary IP address of their output interface.

   The source IP address must be the IP address of a local interface, and the interface must be up. Otherwise, no UDP packets can be sent out.

7. Specify the source port number for UDP packets.

   source port *port-number*

   By default, the NQA client randomly picks an unused port as the source port when the operation starts.

8. (Optional.) Set the payload size for each UDP packet.

   data-size *size*

   The default setting is 100 bytes.

9. (Optional.) Specify the payload fill string for UDP packets.

   data-fill *string*

   The default payload fill string is the hexadecimal string 00010203040506070809.

Configuring the UDP tracert operation

About this task

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

The UDP tracert operation sends a UDP packet to a hop along the path per probe.

Restrictions and guidelines

The UDP tracert operation is not supported on 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.*

Prerequisites

Before you configure the UDP tracert operation, you must 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 HPE devices, use the **ip ttl-expires enable** command.
- Enable sending ICMP destination unreachable messages on the destination device. If the destination device is an HPE 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*.

Procedure

1. Enter system view.

   system-view

2. Create an NQA operation and enter NQA operation view.

nqa entry *admin-name operation-tag*

3. Specify the UDP tracert operation type and enter its view.

   type udp-tracert

4. Specify the destination device for the operation. Choose one option as needed:
   - Specify the destination device by its host name.

     destination host *host-name*

     By default, no destination host name is specified.
   - Specify the destination device by its IP address.

     destination ip *ip-address*

     By default, no destination IP address is specified.

5. Specify the destination port number for UDP packets.

   destination port *port-number*

   By default, the destination port number is 33434.

   This port number must be an unused number on the destination device, so that the destination device can reply with ICMP port unreachable messages.

6. Specify an output interface for UDP packets.

   out interface *interface-type interface-number*

   By default, the NQA client determines the output interface based on the routing table lookup.

7. Specify the source IP address for UDP packets.
   - Specify the IP address of the specified interface as the source IP address.

     source interface *interface-type interface-number*

     By default, the source IP address of UDP packets is the primary IP address of their output interface.
   - Specify the source IP address.

     source ip *ip-address*

     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.

8. Specify the source port number for UDP packets.

   source port *port-number*

   By default, the NQA client randomly picks an unused port as the source port when the operation starts.

9. Set the maximum number of consecutive probe failures.

   max-failure *times*

   The default setting is 5.

10. Set the initial TTL value for UDP packets.

    init-ttl *value*

    The default setting is 1.

11. (Optional.) Set the payload size for each UDP packet.

    data-size *size*

    The default setting is 100 bytes.

12. (Optional.) Enable the no-fragmentation feature.

    no-fragment enable

    By default, the no-fragmentation feature is disabled.

Configuring the voice operation

About this task

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 time stamps 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 timestamp.

The voice operation sends a number of voice packets to the destination device per probe. The number of packets to send per probe is determined by using the **probe packet-number** command.

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 on 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 set an 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."

Restrictions and guidelines

To ensure successful voice operations and avoid affecting existing services, do not perform the operations on well-known ports from 1 to 1023.

The **display nqa history** command does not display the results or statistics of the voice operation. To view the results or statistics of the voice operation, use the **display nqa result** or **display nqa statistics** command.

Before starting the operation, make sure the network devices are time synchronized by using NTP. For more information about NTP, see "Configuring NTP."

Procedure

1. Enter system view.

   system-view

2. Create an NQA operation and enter NQA operation view.

   nqa entry *admin-name operation-tag*

3. Specify the voice type and enter its view.

type voice

4. Specify the destination IP address for voice packets.

   destination ip *ip-address*

   By default, no destination IP address is configured.

   The destination IP address must be the same as the IP address of the UDP listening service configured on the NQA server. To configure a UDP listening service on the server, use the **nqa server udp-echo** command.

5. Specify the destination port number for voice packets.

   destination port *port-number*

   By default, no destination port number is configured.

   The destination port number must be the same as the port number of the UDP listening service configured on the NQA server. To configure a UDP listening service on the server, use the **nqa server udp-echo** command.

6. Specify the source IP address for voice packets.

   source ip *ip-address*

   By default, the source IP address of voice packets is the primary IP address of their output interface.

   The source IP address must be the IP address of a local interface, and the interface must be up. Otherwise, no voice packets can be sent out.

7. Specify the source port number for voice packets.

   source port *port-number*

   By default, the NQA client randomly picks an unused port as the source port when the operation starts.

8. Configure the basic voice operation parameters.
   - Specify the codec type.

     codec-type { **g711a** | **g711u** | **g729a** }

     By default, the codec type is G.711 A-law.
   - Set the advantage factor for calculating MOS and ICPIF values.

     advantage-factor *factor*

     By default, the advantage factor is 0.

9. Configure the probe parameters for the voice operation.
   - Set the number of voice packets to be sent per probe.

     probe packet-number *packet-number*

     The default setting is 1000.
   - Set the interval for sending voice packets.

     probe packet-interval *interval*

     The default setting is 20 milliseconds.
   - Specify how long the NQA client waits for a response from the server before it regards the response times out.

     probe packet-timeout *timeout*

     The default setting is 5000 milliseconds.

10. Configure the payload parameters.

    a. Set the payload size for voice packets.

    data-size *size*

    By default, the voice packet size varies by codec type. The default packet size is 172 bytes for G.711A-law and G.711 μ-law codec type, and 32 bytes for G.729 A-law codec type.

b. (Optional.) Specify the payload fill string for voice packets.

data-fill *string*

The default payload fill string is the hexadecimal string 00010203040506070809.

Configuring the DLSw operation

About this task

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

It sets up a DLSw connection to the DLSw device per probe.

Procedure

1. Enter system view.

   system-view

2. Create an NQA operation and enter NQA operation view.

   nqa entry *admin-name operation-tag*

3. Specify the DLSw type and enter its view.

   type dlsw

4. Specify the destination IP address for the probe packets.

   destination ip *ip-address*

   By default, no destination IP address is specified.

5. Specify the source IP address for the probe packets.

   source ip *ip-address*

   By default, the source IP address of the probe packets is the primary IP address of their output interface.

   The source IP address must be the IP address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.

Configuring the path jitter operation

About this task

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.

The path jitter operation performs the following steps per probe:

1. Obtains the path from the NQA client to the destination through tracert. A maximum of 64 hops can be detected.
2. Sends a number of ICMP echo requests to each hop along the path. The number of ICMP echo requests to send is set by using the **probe packet-number** command.

Prerequisites

Before you configure the path jitter operation, you must 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 HPE devices, use the **ip ttl-expires enable** command.
- Enable sending ICMP destination unreachable messages on the destination device. If the destination device is an HPE 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*.

Procedure

1. Enter system view.
   system-view
2. Create an NQA operation and enter NQA operation view.
   nqa entry *admin-name operation-tag*
3. Specify the path jitter type and enter its view.
   type path-jitter
4. Specify the destination IP address for ICMP echo requests.
   destination ip *ip-address*

   By default, no destination IP address is specified.
5. Specify the source IP address for ICMP echo requests.
   source ip *ip-address*

   By default, the source IP address of ICMP echo requests is the primary IP address of their output interface.

   The source IP address must be the IP address of a local interface, and the interface must be up. Otherwise, no ICMP echo requests can be sent out.
6. Configure the probe parameters for the path jitter operation.
   a. Set the number of ICMP echo requests to be sent per probe.
      probe packet-number *packet-number*

      The default setting is 10.
   b. Set the interval for sending ICMP echo requests.
      probe packet-interval *interval*

      The default setting is 20 milliseconds.
   c. Specify how long the NQA client waits for a response from the server before it regards the response times out.
      probe packet-timeout *timeout*

      The default setting is 3000 milliseconds.
7. (Optional.) Specify an LSR path.
   lsr-path *ip-address*&<1-8>

   By default, no LSR path is specified.

   The path jitter operation uses tracert to detect the LSR path to the destination, and sends ICMP echo requests to each hop on the LSR path.
8. Perform the path jitter operation only on the destination address.
   target-only

   By default, the path jitter operation is performed on each hop on the path to the destination.
9. (Optional.) Set the payload size for each ICMP echo request.
   data-size *size*

   The default setting is 100 bytes.
10. (Optional.) Specify the payload fill string for ICMP echo requests.
    data-fill *string*

    The default payload fill string is the hexadecimal string 00010203040506070809.

Configuring optional parameters for the NQA operation

Restrictions and guidelines

Unless otherwise specified, the following optional parameters apply to all types of NQA operations.

The parameter settings take effect only on the current operation.

Procedure

1. Enter system view.

   system-view

2. Enter the view of an existing NQA operation.

   nqa entry *admin-name operation-tag*

3. Configure a description for the operation.

   description *text*

   By default, no description is configured.

4. Set the interval at which the NQA operation repeats.

   frequency *interval*

   For a voice or path jitter operation, the default setting is 60000 milliseconds.

   For other types of operations, the default setting is 0 milliseconds, and only one operation is performed.

   When the interval expires, but the operation is not completed or is not timed out, the next operation does not start.

5. Specify the probe times.

   probe count *times*

   In an UDP tracert operation, the NQA client performs three probes to each hop to the destination by default.

   In other types of operations, the NQA client performs one probe to the destination per operation by default.

   This command is not available for the voice and path jitter operations. Each of these operations performs only one probe.

6. Set the probe timeout time.

   probe timeout *timeout*

   The default setting is 3000 milliseconds.

   This command is not available for the ICMP jitter, UDP jitter, voice, or path jitter operations.

7. Set the maximum number of hops that the probe packets can traverse.

   ttl *value*

   The default setting is 30 for probe packets of the UDP tracert operation, and is 20 for probe packets of other types of operations.

   This command is not available for the DHCP or path jitter operations.

8. Set the ToS value in the IP header of the probe packets.

   tos *value*

   The default setting is 0.

9. Enable the routing table bypass feature.

   route-option bypass-route

   By default, the routing table bypass feature is disabled.

   This command is not available for the DHCP or path jitter operations.

This command does not take effect if the destination address of the NQA operation is an IPv6 address.

10. Specify the VPN instance where the operation is performed.

    vpn-instance *vpn-instance-name*

    By default, the operation is performed on the public network.

Configuring the collaboration feature

About this task

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.

Restrictions and guidelines

The collaboration feature is not available for the following types of operations:

- ICMP jitter operation.
- UDP jitter operation.
- UDP tracert operation.
- Voice operation.
- Path jitter operation.

Procedure

1. Enter system view.

   system-view

2. Enter the view of an existing NQA operation.

   nqa entry *admin-name operation-tag*

3. Configure a reaction entry.

   reaction *item-number* **checked-element probe-fail threshold-type consecutive** *consecutive-occurrences* **action-type trigger-only**

   You cannot modify the content of an existing reaction entry.

4. Return to system view.

   quit

5. Associate Track with NQA.

   For information about the configuration, see *High Availability Configuration Guide*.

6. Associate Track with an application module.

   For information about the configuration, see *High Availability Configuration Guide*.

Configuring threshold monitoring

About this task

This feature allows you to monitor the NQA operation running status.

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.

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.

  To send traps to the NMS, the NMS address must be specified by using the **snmp-agent target-host** command. For more information about the command, see *Network Management and Monitoring Command Reference*.
- **trigger-only**—NQA displays results on the terminal screen, and meanwhile triggers other modules for collaboration.

In a reaction entry, configure a monitored element, a threshold type, and an action to be triggered to implement threshold monitoring.

The state of a reaction entry can be invalid, over-threshold, or below-threshold.

- Before an NQA operation starts, the reaction entry is in invalid state.
- If the threshold is violated, the state of the entry is set to over-threshold. Otherwise, the state of the entry is set to below-threshold.

Restrictions and guidelines

The threshold monitoring feature is not available for the path jitter operations.

Procedure

1. Enter system view.

   system-view
2. Enter the view of an existing NQA operation.

   nqa entry *admin-name operation-tag*
3. Enable sending traps to the NMS when specific conditions are met.

   reaction trap { **path-change** | **probe-failure** *consecutive-probe-failures* | **test-complete** | **test-failure** [*accumulate-probe-failures*] }

   By default, no traps are sent to the NMS.

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

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

   - The **probe-failure** *consecutive-probe-failures* option.
   - The *accumulate-probe-failures* argument.
4. Configure threshold monitoring. Choose the options to configure as needed:

   - Monitor the operation duration.

     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** }]

     This reaction entry is not supported in the ICMP jitter, UDP jitter, UDP tracert, or voice operations
   - Monitor failure times.

reaction *item-number* **checked-element probe-fail threshold-type** { **accumulate** *accumulate-occurrences* | **consecutive** *consecutive-occurrences* } [**action-type** { **none** | **trap-only** }]

This reaction entry is not supported in the ICMP jitter, UDP jitter, UDP tracert, or voice operations.

- Monitor the round-trip time.

  reaction *item-number* **checked-element rtt threshold-type** { **accumulate** *accumulate-occurrences* | **average** } **threshold-value** *upper-threshold lower-threshold* [**action-type** { **none** | **trap-only** }]

  Only the ICMP jitter, UDP jitter, and voice operations support this reaction entry.

- Monitor packet loss.

  reaction *item-number* **checked-element packet-loss threshold-type accumulate** *accumulate-occurrences* [**action-type** { **none** | **trap-only** }]

  Only the ICMP jitter, UDP jitter, and voice operations support this reaction entry.

- Monitor the one-way jitter.

  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** }]

  Only the ICMP jitter, UDP jitter, and voice operations support this reaction entry.

- Monitor the one-way delay.

  reaction *item-number* **checked-element** { **owd-ds** | **owd-sd** } **threshold-value** *upper-threshold lower-threshold*

  Only the ICMP jitter, UDP jitter, and voice operations support this reaction entry.

- Monitor the ICPIF value.

  reaction *item-number* **checked-element icpif threshold-value** *upper-threshold lower-threshold* [**action-type** { **none** | **trap-only** }]

  Only the voice operation supports this reaction entry.

- Monitor the MOS value.

  reaction *item-number* **checked-element mos threshold-value** *upper-threshold lower-threshold* [**action-type** { **none** | **trap-only** }]

  Only the voice operation supports this reaction entry.

The DNS operation does not support the action of sending trap messages. For the DNS operation, the action type can only be **none**.

Configuring the NQA statistics collection feature

About this task

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.

When the maximum number of statistics groups is reached, the NQA client deletes the oldest statistics group to save a new one.

A statistics group is automatically deleted when its hold time expires.

Restrictions and guidelines

The NQA statistics collection feature is not available for the UDP tracert operations.

If you use the **frequency** command to set the interval to 0 milliseconds for an NQA operation, NQA does not generate any statistics group for the operation.

Procedure

1. Enter system view.
   system-view
2. Enter the view of an existing NQA operation.
   nqa entry *admin-name operation-tag*
3. Set the statistics collection interval.
   statistics interval *interval*
   The default setting is 60 minutes.
4. Set the maximum number of statistics groups that can be saved.
   statistics max-group *number*
   By default, the NQA client can save a maximum of two statistics groups for an operation.
   To disable the NQA statistics collection feature, set the *number* argument to 0.
5. Set the hold time of statistics groups.
   statistics hold-time *hold-time*
   The default setting is 120 minutes.

Configuring the saving of NQA history records

About this task

This task enables the NQA client to save NQA history records. You can use the **display nqa history** command to display the NQA history records.

Restrictions and guidelines

The NQA history record saving feature is not available for the following types of operations:

- ICMP jitter operation.
- UDP jitter operation.
- Voice operation.
- Path jitter operation.

Procedure

1. Enter system view.
   system-view
2. Enter the view of an existing NQA operation.
   nqa entry *admin-name operation-tag*
3. Enable the saving of history records for the NQA operation.
   history-record enable
   By default, this feature is enabled only for the UDP tracert operation.
4. Set the lifetime of history records.
   history-record keep-time *keep-time*
   The default setting is 120 minutes.
   A record is deleted when its lifetime is reached.
5. Set the maximum number of history records that can be saved.
   history-record number *number*

The default setting is 50.

When the maximum number of history records is reached, the system will delete the oldest record to save a new one.

Scheduling the NQA operation on the NQA client

About this task

The NQA operation runs between the specified start time and 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.

Restrictions and guidelines

You cannot enter the operation type view or the operation view of a scheduled NQA operation.

A system time adjustment does not affect started or completed NQA operations. It affects only the NQA operations that have not started.

Procedure

1. Enter system view.

   system-view

2. Specify the scheduling parameters for an NQA operation.

   nqa schedule *admin-name operation-tag* **start-time** { *hh:mm:ss* [*yyyy/mm/dd* | *mm/dd/yyyy*] | **now** } **lifetime** { *lifetime* | **forever** } [**recurring**]

Configuring NQA templates on the NQA client

Restrictions and guidelines

Some operation parameters for an NQA template can be specified by the template configuration or the feature that uses the template. When both are specified, the parameters in the template configuration take effect.

NQA template tasks at a glance

To configure NQA templates, perform the following tasks:

1. Perform at least one of the following tasks:
   - Configuring the ICMP template
   - Configuring the DNS template
   - Configuring the TCP template
   - Configuring the TCP half open template
   - Configuring the UDP template
   - Configuring the HTTP template
   - Configuring the HTTPS template
   - Configuring the FTP template
   - Configuring the RADIUS template
   - Configuring the SSL template
2. (Optional.) Configuring optional parameters for the NQA template

Configuring the ICMP template

About this task

A feature that uses the ICMP template performs the ICMP operation to measure the reachability of a destination device. The ICMP template is supported on both IPv4 and IPv6 networks.

Procedure

1. Enter system view.

   system-view

2. Create an ICMP template and enter its view.

   nqa template icmp *name*

3. Specify the destination IP address for the operation.

   IPv4:

   destination ip *ip-address*

   IPv6:

   destination ipv6 *ipv6-address*

   By default, no destination IP address is configured.

4. Specify the source IP address for ICMP echo requests. Choose one option as needed:

   - Use the IP address of the specified interface as the source IP address.

     source interface *interface-type interface-number*

     By default, the primary IP address of the output interface is used as the source IP address of ICMP echo requests.

     The specified source interface must be up.

   - Specify the source IPv4 address.

     source ip *ip-address*

     By default, the primary IPv4 address of the output interface is used as the source IPv4 address of ICMP echo requests.

     The specified source IPv4 address must be the IPv4 address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.

   - Specify the source IPv6 address.

     source ipv6 *ipv6-address*

     By default, the primary IPv6 address of the output interface is used as the source IPv6 address of ICMP echo requests.

     The specified source IPv6 address must be the IPv6 address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.

5. Specify the next hop IP address for ICMP echo requests.

   IPv4:

   next-hop ip *ip-address*

   IPv6:

   next-hop ipv6 *ipv6-address*

   By default, no IP address of the next hop is configured.

6. Configure the probe result sending on a per-probe basis.

   reaction trigger per-probe

   By default, the probe result is sent to the feature that uses the template after three consecutive failed or successful probes.

If you execute the **reaction trigger per-probe** and **reaction trigger probe-pass** commands multiple times, the most recent configuration takes effect.

If you execute the **reaction trigger per-probe** and **reaction trigger probe-fail** commands multiple times, the most recent configuration takes effect.

7. (Optional.) Set the payload size for each ICMP request.

   data-size *size*

   The default setting is 100 bytes.

8. (Optional.) Specify the payload fill string for ICMP echo requests.

   data-fill *string*

   The default payload fill string is the hexadecimal string 00010203040506070809.

Configuring the DNS template

About this task

A feature that uses the DNS template performs the DNS operation to determine the status of the server. The DNS template is supported on 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.

Prerequisites

Create a mapping between the domain name and an address before you perform the DNS operation. For information about configuring the DNS server, see documents about the DNS server configuration.

Procedure

1. Enter system view.

   system-view

2. Create a DNS template and enter DNS template view.

   nqa template dns *name*

3. Specify the destination IP address for the probe packets.

   IPv4:

   destination ip *ip-address*

   IPv6:

   destination ipv6 *ipv6-address*

   By default, no destination address is specified.

4. Specify the destination port number for the probe packets.

   destination port *port-number*

   By default, the destination port number is 53.

5. Specify the source IP address for the probe packets.

   IPv4:

   source ip *ip-address*

   By default, the source IPv4 address of the probe packets is the primary IPv4 address of their output interface.

   The source IPv4 address must be the IPv4 address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.

   IPv6:

source ipv6 *ipv6-address*

By default, the source IPv6 address of the probe packets is the primary IPv6 address of their output interface.

The source IPv6 address must be the IPv6 address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.

6. Specify the source port number for the probe packets.

   source port *port-number*

   By default, no source port number is specified.

7. Specify the domain name to be translated.

   resolve-target *domain-name*

   By default, no domain name is specified.

8. Specify the domain name resolution type.

   resolve-type { **A** | **AAAA** }

   By default, the type is type A.

   A type A query resolves a domain name to a mapped IPv4 address, and a type AAAA query to a mapped IPv6 address.

9. (Optional.) Specify the IP address that is expected to be returned.

   IPv4:

   expect ip *ip-address*

   IPv6:

   expect ipv6 *ipv6-address*

   By default, no expected IP address is specified.

Configuring the TCP template

About this task

A feature that uses the TCP template performs the TCP operation to test whether the NQA client can establish a TCP connection to a specific port on the server.

In TCP template view, you can specify the expected data to be returned. If you do not specify the expected data, the TCP operation tests only whether the client can establish a TCP connection to the server.

The TCP operation requires both the NQA server and the NQA client. Before you perform a TCP operation, configure a TCP listening service on the NQA server. For more information about the TCP listening service configuration, see "Configuring the NQA server."

Procedure

1. Enter system view.

   system-view

2. Create a TCP template and enter its view.

   nqa template tcp *name*

3. Specify the destination IP address for the probe packets.

   IPv4:

   destination ip *ip-address*

   IPv6:

   destination ipv6 *ipv6-address*

   By default, no destination IP address is specified.

The destination address must be the same as the IP address of the TCP listening service configured on the NQA server. To configure a TCP listening service on the server, use the **nqa server tcp-connect** command.

4. Specify the destination port number for the operation.

   destination port *port-number*

   By default, no destination port number is specified.

   The destination port number must be the same as the port number of the TCP listening service configured on the NQA server. To configure a TCP listening service on the server, use the **nqa server tcp-connect** command.

5. Specify the source IP address for the probe packets.

   IPv4:

   source ip *ip-address*

   By default, the primary IPv4 address of the output interface is used as the source IPv4 address of the probe packets.

   The source IP address must be the IPv4 address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.

   IPv6:

   source ipv6 *ipv6-address*

   By default, the primary IPv6 address of the output interface is used as the source IPv6 address of the probe packets.

   The source IPv6 address must be the IPv6 address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.

6. (Optional.) Specify the payload fill string for the probe packets.

   data-fill *string*

   The default payload fill string is the hexadecimal string 00010203040506070809.

7. (Optional.) Configure the expected data.

   expect data *expression* [**offset** *number*]

   By default, no expected data is configured.

   The NQA client performs expect data check only when you configure both the **data-fill** and **expect-data** commands.

Configuring the TCP half open template

About this task

A feature that uses the TCP half open template performs the TCP half open operation to test whether the TCP service is available on the server. The TCP half open operation is used when the feature cannot get a response from the TCP server through an existing TCP connection.

In the TCP half open operation, the NQA client sends a TCP ACK packet to the server. If the client receives an RST packet, it considers that the TCP service is available on the server.

Procedure

1. Enter system view.

   system-view

2. Create a TCP half open template and enter its view.

   nqa template tcphalfopen *name*

3. Specify the destination IP address of the operation.

   IPv4:

destination ip *ip-address*

IPv6:

destination ipv6 *ipv6-address*

By default, no destination IP address is specified.

The destination address must be the same as the IP address of the TCP listening service configured on the NQA server. To configure a TCP listening service on the server, use the **nqa server tcp-connect** command.

4. Specify the source IP address for the probe packets.

   IPv4:

   source ip *ip-address*

   By default, the primary IPv4 address of the output interface is used as the source IPv4 address of the probe packets.

   The source IPv4 address must be the IPv4 address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.

   IPv6:

   source ipv6 *ipv6-address*

   By default, the primary IPv6 address of the output interface is used as the source IPv6 address of the probe packets.

   The source IPv6 address must be the IPv6 address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.

5. Specify the next hop IP address for the probe packets.

   IPv4:

   next-hop ip *ip-address*

   IPv6:

   next-hop ipv6 *ipv6-address*

   By default, the IP address of the next hop is configured.

6. Configure the probe result sending on a per-probe basis.

   reaction trigger per-probe

   By default, the probe result is sent to the feature that uses the template after three consecutive failed or successful probes.

   If you execute the **reaction trigger per-probe** and **reaction trigger probe-pass** commands multiple times, the most recent configuration takes effect.

   If you execute the **reaction trigger per-probe** and **reaction trigger probe-fail** commands multiple times, the most recent configuration takes effect.

Configuring the UDP template

About this task

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."

Procedure

1. Enter system view.

   system-view

2. Create a UDP template and enter its view.

   nqa template udp *name*

3. Specify the destination IP address of the operation.

   IPv4:

   destination ip *ip-address*

   IPv6:

   destination ipv6 *ipv6-address*

   By default, no destination IP address is specified.

   The destination address must be the same as the IP address of the UDP listening service configured on the NQA server. To configure a UDP listening service on the server, use the **nqa server udp-echo** command.

4. Specify the destination port number for the operation.

   destination port *port-number*

   By default, no destination port number is specified.

   The destination port number must be the same as the port number of the UDP listening service configured on the NQA server. To configure a UDP listening service on the server, use the **nqa server udp-echo** command.

5. Specify the source IP address for the probe packets.

   IPv4:

   source ip *ip-address*

   By default, the primary IPv4 address of the output interface is used as the source IPv4 address of the probe packets.

   The source IP address must be the IPv4 address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.

   IPv6:

   source ipv6 *ipv6-address*

   By default, the primary IPv6 address of the output interface is used as the source IPv6 address of the probe packets.

   The source IPv6 address must be the IPv6 address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.

6. Specify the payload fill string for the probe packets.

   data-fill *string*

   The default payload fill string is the hexadecimal string 00010203040506070809.

7. (Optional.) Set the payload size for the probe packets.

   data-size *size*

   The default setting is 100 bytes.

8. (Optional.) Configure the expected data.

   expect data *expression* [**offset** *number*]

   By default, no expected data is configured.

Expected data check is performed only when both the **data-fill** command and the **expect data** command are configured.

Configuring the HTTP template

About this task

A feature that uses the HTTP template performs the HTTP operation to measure the time it takes the NQA client to obtain data from an HTTP server.

The expected data is checked only when the data is configured and the HTTP response contains the Content-Length field in the HTTP header.

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.

Prerequisites

Before you perform the HTTP operation, you must configure the HTTP server.

Procedure

1. Enter system view.
   system-view
2. Create an HTTP template and enter its view.
   nqa template http *name*
3. Specify the destination URL for the HTTP template.
   url *url*
   By default, no destination URL is specified for an HTTP template.
   Enter the URL in one of the following formats:
   - http://*host*/*resource*
   - http://*host*:*port*/*resource*
4. Specify an HTTP login username.
   username *username*
   By default, no HTTP login username is specified.
5. Specify an HTTP login password.
   password { **cipher** | **simple** } *string*
   By default, no HTTP login password is specified.
6. Specify the HTTP version.
   version { **v1.0** | **v1.1** }
   By default, HTTP 1.0 is used.
7. Specify the HTTP operation type.
   operation { **get** | **post** | **raw** }
   By default, the HTTP operation type is **get**.
   If you set the operation type to raw, the client pads the content configured in raw request view to the HTTP request to send to the HTTP server.
8. Configure the content of the HTTP raw request.
   a. Enter raw request view.
      raw-request
      Every time you enter raw request view, the previously configured raw request content is cleared.

b. Enter or paste the request content.

By default, no request content is configured.

To ensure successful operations, make sure the request content does not contain command aliases configured by using the **alias** command. For more information about the **alias** command, see CLI commands in *Fundamentals Command Reference*.

c. Return to HTTP template view.

quit

The system automatically saves the configuration in raw request view before it returns to HTTP template view.

This step is required only when the operation type is set to **raw**.

9. Specify the source IP address for the probe packets.

IPv4:

source ip *ip-address*

By default, the primary IPv4 address of the output interface is used as the source IPv4 address of the probe packets.

The source IPv4 address must be the IPv4 address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.

IPv6:

source ipv6 *ipv6-address*

By default, the primary IPv6 address of the output interface is used as the source IPv6 address of the probe packets.

The source IPv6 address must be the IPv6 address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.

10. (Optional.) Configure the expected status codes.

expect status *status-list*

By default, no expected status code is configured.

11. (Optional.) Configure the expected data.

expect data *expression* [**offset** *number*]

By default, no expected data is configured.

Configuring the HTTPS template

About this task

A feature that uses the HTTPS template performs the HTTPS operation to measure the time it takes for the NQA client to obtain data from an HTTPS server.

The expected data is checked only when the expected data is configured and the HTTPS response contains the Content-Length field in the HTTPS header.

The status code of the HTTPS packet is a three-digit field in decimal notation, and it includes the status information for the HTTPS server. The first digit defines the class of response.

Prerequisites

Before you perform the HTTPS operation, configure the HTTPS server and the SSL client policy for the SSL client. For information about configuring SSL client policies, see *Security Configuration Guide*.

Procedure

1. Enter system view.

system-view

2. Create an HTTPS template and enter its view.

   nqa template https *name*

3. Specify the destination URL for the HTTPS template.

   url *url*

   By default, no destination URL is specified for an HTTPS template.

   Enter the URL in one of the following formats:

   - https://*host*/*resource*
   - https://*host*:*port*/*resource*

4. Specify an HTTPS login username.

   username *username*

   By default, no HTTPS login username is specified.

5. Specify an HTTPS login password.

   password { **cipher** | **simple** } *string*

   By default, no HTTPS login password is specified.

6. Specify an SSL client policy.

   ssl-client-policy *policy-name*

   By default, no SSL client policy is specified.

7. Specify the HTTPS version.

   version { **v1.0** | **v1.1** }

   By default, HTTPS 1.0 is used.

8. Specify the HTTPS operation type.

   operation { **get** | **post** | **raw** }

   By default, the HTTPS operation type is **get**.

   If you set the operation type to raw, the client pads the content configured in raw request view to the HTTPS request to send to the HTTPS server.

9. Configure the content of the HTTPS raw request.

   a. Enter raw request view.

      raw-request

      Every time you enter raw request view, the previously configured raw request content is cleared.

   b. Enter or paste the request content.

      By default, no request content is configured.

      To ensure successful operations, make sure the request content does not contain command aliases configured by using the **alias** command. For more information about the **alias** command, see CLI commands in *Fundamentals Command Reference*.

   c. Return to HTTPS template view.

      quit

      The system automatically saves the configuration in raw request view before it returns to HTTPS template view.

   This step is required only when the operation type is set to **raw**.

10. Specify the source IP address for the probe packets.

    IPv4:

    source ip *ip-address*

    By default, the primary IPv4 address of the output interface is used as the source IPv4 address of the probe packets.

The source IP address must be the IPv4 address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.

IPv6:

source ipv6 *ipv6-address*

By default, the primary IPv6 address of the output interface is used as the source IPv6 address of the probe packets.

The source IPv6 address must be the IPv6 address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.

11. (Optional.) Configure the expected data.

    expect data *expression* [**offset** *number*]

    By default, no expected data is configured.

12. (Optional.) Configure the expected status codes.

    expect status *status-list*

    By default, no expected status code is configured.

Configuring the FTP template

About this task

A feature that uses the FTP template performs the FTP operation. The operation measures the time it takes the NQA client to transfer a file to or download a file from an FTP server.

Configure the username and password for the FTP client to log in to the FTP server before you perform an FTP operation. For information about configuring the FTP server, see *Fundamentals Configuration Guide*.

Procedure

1. Enter system view.

   system-view

2. Create an FTP template and enter its view.

   nqa template ftp *name*

3. Specify an FTP login username.

   username *username*

   By default, no FTP login username is specified.

4. Specify an FTP login password.

   password { **cipher** | **simple** } *sting*

   By default, no FTP login password is specified.

5. Specify the source IP address for the probe packets.

   IPv4:

   source ip *ip-address*

   By default, the primary IPv4 address of the output interface is used as the source IPv4 address of the probe packets.

   The source IP address must be the IPv4 address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.

   IPv6:

   source ipv6 *ipv6-address*

   By default, the primary IPv6 address of the output interface is used as the source IPv6 address of the probe packets.

The source IPv6 address must be the IPv6 address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.

6. Set the data transmission mode.

   mode { **active** | **passive** }

   The default mode is **active**.

7. Specify the FTP operation type.

   operation { **get** | **put** }

   By default, the FTP operation type is **get**, which means obtaining files from the FTP server.

8. Specify the destination URL for the FTP template.

   url *url*

   By default, no destination URL is specified for an FTP template.

   Enter the URL in one of the following formats:

   - ftp://*host*/*filename*.
   - ftp://*host*:*port*/*filename*.

   When you perform the **get** operation, the file name is required.

   When you perform the **put** operation, the *filename* argument does not take effect, even if it is specified. The file name for the **put** operation is determined by using the **filename** command.

9. Specify the name of a file to be transferred.

   filename *filename*

   By default, no file is specified.

   This task is required only for the **put** operation.

   The configuration does not take effect for the **get** operation.

Configuring the RADIUS template

About this task

A feature that uses the RADIUS template performs the RADIUS authentication operation to check the availability of the authentication service on the RADIUS server.

The RADIUS authentication operation workflow is as follows:

1. The NQA client sends an authentication request (Access-Request) to the RADIUS server. The request includes the username and the password. The password is encrypted by using the MD5 algorithm and the shared key.
2. The RADIUS server authenticates the username and password.
   - If the authentication succeeds, the server sends an Access-Accept packet to the NQA client.
   - If the authentication fails, the server sends an Access-Reject packet to the NQA client.
3. The NQA client determines the availability of the authentication service on the RADIUS server based on the response packet it received:
   - If an Access-Accept packet is received, the authentication service is available on the RADIUS server.
   - If an Access-Reject packet is received, the authentication service is not available on the RADIUS server.

Prerequisites

Before you configure the RADIUS template, specify a username, password, and shared key on the RADIUS server. For more information about configuring the RADIUS server, see AAA in *Security Configuration Guide*.

Procedure

1. Enter system view.
   system-view
2. Create a RADIUS template and enter its view.
   nqa template radius *name*
3. Specify the destination IP address of the operation.
   IPv4:
   destination ip *ip-address*
   IPv6:
   destination ipv6 *ipv6-address*
   By default, no destination IP address is specified.
4. Specify the destination port number for the operation.
   destination port *port-number*
   By default, the destination port number is 1812.
5. Specify a username.
   username *username*
   By default, no username is specified.
6. Specify a password.
   password { **cipher** | **simple** } *string*
   By default, no password is specified.
7. Specify a shared key for secure RADIUS authentication.
   key { **cipher** | **simple** } *string*
   By default, no shared key is specified for RADIUS authentication.
8. Specify the source IP address for the probe packets.
   IPv4:
   source ip *ip-address*
   By default, the primary IPv4 address of the output interface is used as the source IPv4 address of the probe packets.
   The source IP address must be the IPv4 address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.
   IPv6:
   source ipv6 *ipv6-address*
   By default, the primary IPv6 address of the output interface is used as the source IPv6 address of the probe packets.
   The source IPv6 address must be the IPv6 address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.

Configuring the SSL template

About this task

A feature that uses the SSL template performs the SSL operation to measure the time required to establish an SSL connection to an SSL server.

Prerequisites

Before you configure the SSL template, you must configure the SSL client policy. For information about configuring SSL client policies, see *Security Configuration Guide*.

Procedure

1. Enter system view.
   system-view
2. Create an SSL template and enter its view.
   nqa template ssl *name*
3. Specify the destination IP address of the operation.
   IPv4:
   destination ip *ip-address*
   IPv6:
   destination ipv6 *ipv6-address*
   By default, no destination IP address is specified.
4. Specify the destination port number for the operation.
   destination port *port-number*
   By default, the destination port number is not specified.
5. Specify an SSL client policy.
   ssl-client-policy *policy-name*
   By default, no SSL client policy is specified.
6. Specify the source IP address for the probe packets.
   IPv4:
   source ip *ip-address*
   By default, the primary IPv4 address of the output interface is used as the source IPv4 address of the probe packets.
   The source IP address must be the IPv4 address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.
   IPv6:
   source ipv6 *ipv6-address*
   By default, the primary IPv6 address of the output interface is used as the source IPv6 address of the probe packets.
   The source IPv6 address must be the IPv6 address of a local interface, and the interface must be up. Otherwise, no probe packets can be sent out.

Configuring optional parameters for the NQA template

Restrictions and guidelines

Unless otherwise specified, the following optional parameters apply to all types of NQA templates.

The parameter settings take effect only on the current NQA template.

Procedure

1. Enter system view.
   system-view
2. Enter the view of an existing NQA template.
   nqa template { **dns** | **ftp** | **http** | **https** | **icmp** | **radius** | **ssl** | **tcp** | **tcphalfopen** | **udp** } *name*
3. Configure a description.
   description *text*

By default, no description is configured.

4. Set the interval at which the NQA operation repeats.
   frequency *interval*
   The default setting is 5000 milliseconds.
   If the operation is not completed when the interval expires, the next operation does not start.
5. Set the probe timeout time.
   probe timeout *timeout*
   The default setting is 3000 milliseconds.
6. Set the TTL for the probe packets.
   ttl *value*
   The default setting is 20.
   This command is not available for the ARP template.
7. Set the ToS value in the IP header of the probe packets.
   tos *value*
   The default setting is 0.
   This command is not available for the ARP template.
8. Specify the VPN instance where the operation is performed.
   vpn-instance *vpn-instance-name*
   By default, the operation is performed on the public network.
9. Set the number of consecutive successful probes to determine a successful operation event.
   reaction trigger probe-pass *count*
   The default setting is 3.
   If the number of consecutive successful probes for an NQA operation is reached, the NQA client notifies the feature that uses the template of the successful operation event.
10. Set the number of consecutive probe failures to determine an operation failure.
    reaction trigger probe-fail *count*
    The default setting is 3.
    If the number of consecutive probe failures for an NQA operation is reached, the NQA client notifies the feature that uses the NQA template of the operation failure.

Display and maintenance commands for NQA

Execute **display** commands in any view.

Task	Command
Display history records of NQA operations.	**display nqa history** [*admin-name operation-tag*]
Display the current monitoring results of reaction entries.	**display nqa reaction counters** [*admin-name operation-tag* [*item-number*]]
Display the most recent result of the NQA operation.	**display nqa result** [*admin-name operation-tag*]
Display NQA server status.	**display nqa server status**
Display NQA statistics.	**display nqa statistics** [*admin-name operation-tag*]

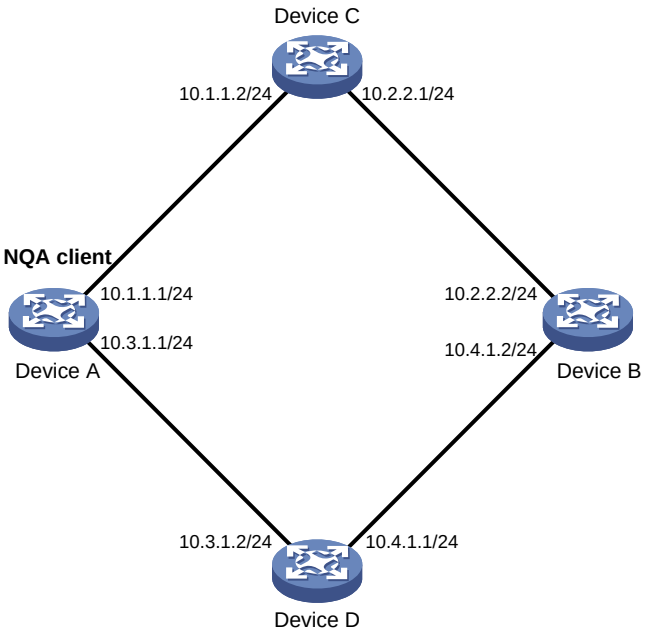
NQA configuration examples

Example: Configuring the ICMP echo operation

Network configuration

As shown in Figure 7, configure an ICMP echo operation on the NQA client (Device A) to test the round-trip time to Device B. The next hop of Device A is Device C.

Figure 7 Network diagram



Procedure

Assign IP addresses to interfaces, as shown in Figure 7. (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 10.2.2.2 as the destination IP address of ICMP echo requests.

```
[DeviceA-nqa-admin-test1-icmp-echo] destination ip 10.2.2.2
```

Specify 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 ip 10.1.1.2
```

Configure the ICMP echo operation to perform 10 probes.

```
[DeviceA-nqa-admin-test1-icmp-echo] probe count 10
```

Set the probe timeout time to 500 milliseconds for the ICMP echo operation.

```
[DeviceA-nqa-admin-test1-icmp-echo] probe timeout 500
```

Configure the ICMP echo operation to repeat every 5000 milliseconds.

```
[DeviceA-nqa-admin-test1-icmp-echo] frequency 5000
```

Enable saving history records.

```
[DeviceA-nqa-admin-test1-icmp-echo] history-record enable
```

Set the maximum number of history records to 10.

```
[DeviceA-nqa-admin-test1-icmp-echo] history-record number 10
[DeviceA-nqa-admin-test1-icmp-echo] quit
```

Start the ICMP echo operation.

```
[DeviceA] nqa schedule admin test1 start-time now lifetime forever
```

After the ICMP echo operation runs for a period of time, stop the operation.

```
[DeviceA] undo nqa schedule admin test1
```

Verifying the configuration

Display the most recent result of the ICMP echo operation.

```
[DeviceA] display nqa result admin test1
NQA entry (admin admin, tag test1) test results:
    Send operation times: 10             Receive response times: 10
    Min/Max/Average round trip time: 2/5/3
    Square-Sum of round trip time: 96
    Last succeeded probe time: 2011-08-23 15:00:01.2
  Extended results:
    Packet loss ratio: 0%
    Failures due to timeout: 0
    Failures due to internal error: 0
    Failures due to other errors: 0
```

Display the history records of the ICMP echo operation.

```
[DeviceA] display nqa history admin test1
NQA entry (admin admin, tag test) history records:
  Index      Response     Status           Time
  370        3            Succeeded        2007-08-23 15:00:01.2
  369        3            Succeeded        2007-08-23 15:00:01.2
  368        3            Succeeded        2007-08-23 15:00:01.2
  367        5            Succeeded        2007-08-23 15:00:01.2
  366        3            Succeeded        2007-08-23 15:00:01.2
  365        3            Succeeded        2007-08-23 15:00:01.2
  364        3            Succeeded        2007-08-23 15:00:01.1
  363        2            Succeeded        2007-08-23 15:00:01.1
  362        3            Succeeded        2007-08-23 15:00:01.1
  361        2            Succeeded        2007-08-23 15:00:01.1
```

The output shows that the packets sent by Device A can reach Device B through Device C. No packet loss occurs during the operation. The minimum, maximum, and average round-trip times are 2, 5, and 3 milliseconds, respectively.

Example: Configuring the ICMP jitter operation

Network configuration

As shown in Figure 8, configure an ICMP jitter operation to test the jitter between Device A and Device B.

Figure 8 Network diagram

Procedure

1. Assign IP addresses to interfaces, as shown in Figure 8. (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 A:

   # Create an ICMP jitter operation.

```
<DeviceA> system-view
[DeviceA] nqa entry admin test1
[DeviceA-nqa-admin-test1] type icmp-jitter
```

   # Specify 10.2.2.2 as the destination address for the operation.

```
[DeviceA-nqa-admin-test1-icmp-jitter] destination ip 10.2.2.2
```

   # Configure the operation to repeat every 1000 milliseconds.

```
[DeviceA-nqa-admin-test1-icmp-jitter] frequency 1000
[DeviceA-nqa-admin-test1-icmp-jitter] quit
```

   # Start the ICMP jitter operation.

```
[DeviceA] nqa schedule admin test1 start-time now lifetime forever
```

   # After the ICMP jitter operation runs for a period of time, stop the operation.

```
[DeviceA] undo nqa schedule admin test1
```

Verifying the configuration

Display the most recent result of the ICMP 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: 1/2/1
    Square-Sum of round trip time: 13
    Last packet received time: 2015-03-09 17:40:29.8
  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
  ICMP-jitter results:
   RTT number: 10
    Min positive SD: 0                    Min positive DS: 0
    Max positive SD: 0                    Max positive DS: 0
    Positive SD number: 0                 Positive DS number: 0
    Positive SD sum: 0                    Positive DS sum: 0
    Positive SD average: 0                Positive DS average: 0
    Positive SD square-sum: 0             Positive DS square-sum: 0
```

```
    Min negative SD: 1                    Min negative DS: 2
    Max negative SD: 1                    Max negative DS: 2
    Negative SD number: 1                 Negative DS number: 1
    Negative SD sum: 1                    Negative DS sum: 2
    Negative SD average: 1                Negative DS average: 2
    Negative SD square-sum: 1             Negative DS square-sum: 4
    SD average: 1                         DS average: 2
  One way results:
    Max SD delay: 1                       Max DS delay: 2
    Min SD delay: 1                       Min DS delay: 2
    Number of SD delay: 1                 Number of DS delay: 1
    Sum of SD delay: 1                    Sum of DS delay: 2
    Square-Sum of SD delay: 1             Square-Sum of DS delay: 4
    Lost packets for unknown reason: 0
```

Display the statistics of the ICMP jitter operation.

```
[DeviceA] display nqa statistics admin test1
NQA entry (admin admin, tag test1) test statistics:
  NO. : 1
    Start time: 2015-03-09 17:42:10.7
    Life time: 156 seconds
    Send operation times: 1560          Receive response times: 1560
    Min/Max/Average round trip time: 1/2/1
    Square-Sum of round trip time: 1563
  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
  ICMP-jitter results:
   RTT number: 1560
    Min positive SD: 1                    Min positive DS: 1
    Max positive SD: 1                    Max positive DS: 2
    Positive SD number: 18                Positive DS number: 46
    Positive SD sum: 18                   Positive DS sum: 49
    Positive SD average: 1                Positive DS average: 1
    Positive SD square-sum: 18            Positive DS square-sum: 55
    Min negative SD: 1                    Min negative DS: 1
    Max negative SD: 1                    Max negative DS: 2
    Negative SD number: 24                Negative DS number: 57
    Negative SD sum: 24                   Negative DS sum: 58
    Negative SD average: 1                Negative DS average: 1
    Negative SD square-sum: 24            Negative DS square-sum: 60
    SD average: 16                        DS average: 2
  One way results:
    Max SD delay: 1                       Max DS delay: 2
    Min SD delay: 1                       Min DS delay: 1
```

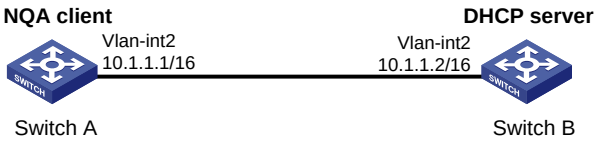
```
Number of SD delay: 4                Number of DS delay: 4
Sum of SD delay: 4                   Sum of DS delay: 5
Square-Sum of SD delay: 4            Square-Sum of DS delay: 7
Lost packets for unknown reason: 0
```

Example: Configuring the DHCP operation

Network configuration

As shown in Figure 9, configure a DHCP operation to test the time required for Switch A to obtain an IP address from the DHCP server (Switch B).

Figure 9 Network diagram



Procedure

Create a DHCP operation.

```
<SwitchA> system-view
[SwitchA] nqa entry admin test1
[SwitchA-nqa-admin-test1] type dhcp
```

Specify the DHCP server 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
```

Verifying the configuration

Display the most recent result of the DHCP operation.

```
[SwitchA] display nqa result admin test1
NQA entry (admin admin, tag test1) test results:
    Send operation times: 1              Receive response times: 1
    Min/Max/Average round trip time: 512/512/512
    Square-Sum of round trip time: 262144
    Last succeeded probe time: 2011-11-22 09:56:03.2
  Extended results:
    Packet loss ratio: 0%
    Failures due to timeout: 0
    Failures due to internal error: 0
    Failures due to other errors: 0
```

Display the history records of the DHCP operation.

```
[SwitchA] display nqa history admin test1
```

```
NQA entry (admin admin, tag test1) history records:
  Index      Response     Status           Time
  1          512          Succeeded        2011-11-22 09:56:03.2
```

The output shows that it took Switch A 512 milliseconds to obtain an IP address from the DHCP server.

Example: Configuring the DNS operation

Network configuration

As shown in Figure 10, configure a DNS operation to test whether Device A can perform address resolution through the DNS server and test the resolution time.

Figure 10 Network diagram

Procedure

Assign IP addresses to interfaces, as shown in Figure 10. (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 **host.com** as the domain name to be translated.

```
[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
```

Verifying the configuration

Display the most recent result of the DNS operation.

```
[DeviceA] display nqa result admin test1
NQA entry (admin admin, tag test1) test results:
    Send operation times: 1              Receive response times: 1
    Min/Max/Average round trip time: 62/62/62
    Square-Sum of round trip time: 3844
    Last succeeded probe time: 2011-11-10 10:49:37.3
  Extended results:
```

```
    Packet loss ratio: 0%
    Failures due to timeout: 0
    Failures due to internal error: 0
    Failures due to other errors: 0
```

Display the history records of the DNS operation.

```
[DeviceA] display nqa history admin test1
NQA entry (admin admin, tag test) history records:
  Index      Response     Status           Time
  1          62           Succeeded        2011-11-10 10:49:37.3
```

The output shows that it took Device A 62 milliseconds to translate domain name **host.com** into an IP address.

Example: Configuring the FTP operation

Network configuration

As shown in Figure 11, 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 11 Network diagram

Procedure

Assign IP addresses to interfaces, as shown in Figure 11. (Details not shown.)

Configure static routes or a routing protocol to make sure the devices can reach each other. (Details not shown.)

Create an FTP operation.

```
<DeviceA> system-view
[DeviceA] nqa entry admin test1
[DeviceA-nqa-admin-test1] type ftp
```

Specify the URL of the FTP server.

```
[DeviceA-nqa-admin-test-ftp] url ftp://10.2.2.2
```

Specify 10.1.1.1 as the source IP address.

```
[DeviceA-nqa-admin-test1-ftp] source ip 10.1.1.1
```

Configure the device to upload file **config.txt** to the FTP server.

```
[DeviceA-nqa-admin-test1-ftp] operation put
[DeviceA-nqa-admin-test1-ftp] filename config.txt
```

Set the username to **admin** for the FTP operation.

```
[DeviceA-nqa-admin-test1-ftp] username admin
```

Set the password to **systemtest** for the FTP operation.

```
[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
```

Verifying the configuration

Display the most recent result of the FTP operation.

```
[DeviceA] display nqa result admin test1
NQA entry (admin admin, tag test1) test results:
    Send operation times: 1              Receive response times: 1
    Min/Max/Average round trip time: 173/173/173
    Square-Sum of round trip time: 29929
    Last succeeded probe time: 2011-11-22 10:07:28.6
  Extended results:
    Packet loss ratio: 0%
    Failures due to timeout: 0
    Failures due to disconnect: 0
    Failures due to no connection: 0
    Failures due to internal error: 0
    Failures due to other errors: 0
```

Display the history records of the FTP operation.

```
[DeviceA] display nqa history admin test1
NQA entry (admin admin, tag test1) history records:
  Index      Response     Status           Time
  1          173          Succeeded        2011-11-22 10:07:28.6
```

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

Example: Configuring the HTTP operation

Network configuration

As shown in Figure 12, configure an HTTP operation on the NQA client to test the time required to obtain data from the HTTP server.

Figure 12 Network diagram

Procedure

Assign IP addresses to interfaces, as shown in Figure 12. (Details not shown.)

Configure static routes or a routing protocol to make sure the devices can reach each other. (Details not shown.)

Create an HTTP operation.

```
<DeviceA> system-view
[DeviceA] nqa entry admin test1
```

```
[DeviceA-nqa-admin-test1] type http
```

Specify the URL of the HTTP server.

```
[DeviceA-nqa-admin-test1-http] url http://10.2.2.2/index.htm
```

Configure the HTTP operation to get data from the HTTP server.

```
[DeviceA-nqa-admin-test1-http] operation get
```

Configure the operation to use HTTP version 1.0.

```
[DeviceA-nqa-admin-test1-http] version v1.0
```

Enable the saving of history records.

```
[DeviceA-nqa-admin-test1-http] history-record enable
[DeviceA-nqa-admin-test1-http] quit
```

Start the HTTP operation.

```
[DeviceA] nqa schedule admin test1 start-time now lifetime forever
```

After the HTTP operation runs for a period of time, stop the operation.

```
[DeviceA] undo nqa schedule admin test1
```

Verifying the configuration

Display the most recent result of the HTTP operation.

```
[DeviceA] display nqa result admin test1
NQA entry (admin admin, tag test1) test results:
    Send operation times: 1              Receive response times: 1
    Min/Max/Average round trip time: 64/64/64
    Square-Sum of round trip time: 4096
    Last succeeded probe time: 2011-11-22 10:12:47.9
  Extended results:
    Packet loss ratio: 0%
    Failures due to timeout: 0
    Failures due to disconnect: 0
    Failures due to no connection: 0
    Failures due to internal error: 0
    Failures due to other errors: 0
```

Display the history records of the HTTP operation.

```
[DeviceA] display nqa history admin test1
NQA entry (admin admin, tag test1) history records:
  Index      Response     Status           Time
  1          64           Succeeded        2011-11-22 10:12:47.9
```

The output shows that it took Device A 64 milliseconds to obtain data from the HTTP server.

Example: Configuring the UDP jitter operation

Network configuration

As shown in Figure 13, configure a UDP jitter operation to test the jitter, delay, and round-trip time between Device A and Device B.

Figure 13 Network diagram

Procedure

1. Assign IP addresses to interfaces, as shown in Figure 13. (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 UDP port 9000 on IP address 10.2.2.2.

   ```
   [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
   ```

   # Specify 10.2.2.2 as the destination address of the operation.

   ```
   [DeviceA-nqa-admin-test1-udp-jitter] destination ip 10.2.2.2
   ```

   # Set the destination port number to 9000.

   ```
   [DeviceA-nqa-admin-test1-udp-jitter] destination port 9000
   ```

   # Configure the operation to repeat every 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
   ```

Verifying the configuration

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
    SD average: 10                        DS average: 10
  One way results:
    Max SD delay: 15                      Max DS delay: 16
    Min SD delay: 7                       Min DS delay: 7
    Number of SD delay: 10                Number of DS delay: 10
    Sum of SD delay: 78                   Sum of DS delay: 85
    Square-Sum of SD delay: 666           Square-Sum of DS delay: 787
    SD lost packets: 0                  DS lost packets: 0
    Lost packets for unknown reason: 0
```

Display the statistics of the UDP jitter operation.

```
[DeviceA] display nqa statistics admin test1
NQA entry (admin admin, tag test1) test statistics:
  NO. : 1
    Start time: 2011-05-29 13:56:14.0
    Life time: 47 seconds
    Send operation times: 410           Receive response times: 410
    Min/Max/Average round trip time: 1/93/19
    Square-Sum of round trip time: 206176
  Extended results:
    Packet loss ratio: 0%
    Failures due to timeout: 0
    Failures due to internal error: 0
    Failures due to other errors: 0
    Packets out of sequence: 0
    Packets arrived late: 0
  UDP-jitter results:
   RTT number: 410
    Min positive SD: 3                    Min positive DS: 1
    Max positive SD: 30                   Max positive DS: 79
    Positive SD number: 186               Positive DS number: 158
    Positive SD sum: 2602                 Positive DS sum: 1928
    Positive SD average: 13               Positive DS average: 12
    Positive SD square-sum: 45304         Positive DS square-sum: 31682
```

```
  Min negative SD: 1                    Min negative DS: 1
  Max negative SD: 30                   Max negative DS: 78
  Negative SD number: 181               Negative DS number: 209
  Negative SD sum: 181                  Negative DS sum: 209
  Negative SD average: 13               Negative DS average: 14
  Negative SD square-sum: 46994         Negative DS square-sum: 3030
  SD average: 9                         DS average: 1
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
```

Example: Configuring the SNMP operation

Network configuration

As shown in Figure 14, configure an SNMP operation to test the time the NQA client uses to get a response from the SNMP agent.

Figure 14 Network diagram

Procedure

1. Assign IP addresses to interfaces, as shown in Figure 14. (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
   ```

   # Specify 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
```

Verifying the configuration

Display the most recent result of the SNMP operation.

```
[DeviceA] display nqa result admin test1
NQA entry (admin admin, tag test1) test results:
    Send operation times: 1              Receive response times: 1
    Min/Max/Average round trip time: 50/50/50
    Square-Sum of round trip time: 2500
    Last succeeded probe time: 2011-11-22 10:24:41.1
  Extended results:
    Packet loss ratio: 0%
    Failures due to timeout: 0
    Failures due to internal error: 0
    Failures due to other errors: 0
```

Display the history records of the SNMP operation.

```
[DeviceA] display nqa history admin test1
NQA entry (admin admin, tag test1) history records:
  Index      Response     Status           Time
  1          50           Succeeded        2011-11-22 10:24:41.1
```

The output shows that it took Device A 50 milliseconds to receive a response from the SNMP agent.

Example: Configuring the TCP operation

Network configuration

As shown in Figure 15, configure a TCP operation to test the time required for Device A to establish a TCP connection with Device B.

Figure 15 Network diagram

Procedure

1. Assign IP addresses to interfaces, as shown in Figure 15. (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 TCP port 9000 on IP address 10.2.2.2.

```
[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
```

Specify 10.2.2.2 as the destination IP address.

```
[DeviceA-nqa-admin-test1-tcp] destination ip 10.2.2.2
```

Set the destination port number to 9000.

```
[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
```

Verifying the configuration

Display the most recent result of the TCP operation.

```
[DeviceA] display nqa result admin test1
NQA entry (admin admin, tag test1) test results:
    Send operation times: 1              Receive response times: 1
    Min/Max/Average round trip time: 13/13/13
    Square-Sum of round trip time: 169
    Last succeeded probe time: 2011-11-22 10:27:25.1
  Extended results:
    Packet loss ratio: 0%
    Failures due to timeout: 0
    Failures due to disconnect: 0
    Failures due to no connection: 0
    Failures due to internal error: 0
    Failures due to other errors: 0
```

Display the history records of the TCP operation.

```
[DeviceA] display nqa history admin test1
NQA entry (admin admin, tag test1) history records:
  Index      Response     Status           Time
  1          13           Succeeded        2011-11-22 10:27:25.1
```

The output shows that it took Device A 13 milliseconds to establish a TCP connection to port 9000 on the NQA server.

Example: Configuring the UDP echo operation

Network configuration

As shown in Figure 16, configure a UDP echo operation on the NQA client to test the round-trip time to Device B. The destination port number is 8000.

Figure 16 Network diagram

Procedure

1. Assign IP addresses to interfaces, as shown in Figure 16. (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 UDP port 8000 on IP address 10.2.2.2.

   ```
   [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
   ```

   # Specify 10.2.2.2 as the destination IP address.

   ```
   [DeviceA-nqa-admin-test1-udp-echo] destination ip 10.2.2.2
   ```

   # Set the destination port number to 8000.

   ```
   [DeviceA-nqa-admin-test1-udp-echo] destination port 8000
   ```

   # Enable the saving of history records.

   ```
   [DeviceA-nqa-admin-test1-udp-echo] history-record enable
   [DeviceA-nqa-admin-test1-udp-echo] quit
   ```

   # Start the UDP echo operation.

   ```
   [DeviceA] nqa schedule admin test1 start-time now lifetime forever
   ```

   # After the UDP echo operation runs for a period of time, stop the operation.

   ```
   [DeviceA] undo nqa schedule admin test1
   ```

Verifying the configuration

Display the most recent result of the UDP echo operation.

```
[DeviceA] display nqa result admin test1
NQA entry (admin admin, tag test1) test results:
    Send operation times: 1              Receive response times: 1
    Min/Max/Average round trip time: 25/25/25
    Square-Sum of round trip time: 625
    Last succeeded probe time: 2011-11-22 10:36:17.9
  Extended results:
    Packet loss ratio: 0%
    Failures due to timeout: 0
    Failures due to internal error: 0
    Failures due to other errors: 0
```

Display the history records of the UDP echo operation.

```
[DeviceA] display nqa history admin test1
NQA entry (admin admin, tag test1) history records:
  Index      Response     Status           Time
  1          25           Succeeded        2011-11-22 10:36:17.9
```

The output shows that the round-trip time between Device A and port 8000 on Device B is 25 milliseconds.

Example: Configuring the UDP tracert operation

Network configuration

As shown in Figure 17, configure a UDP tracert operation to determine the routing path from Device A to Device B.

Figure 17 Network diagram

Procedure

1. Assign IP addresses to interfaces, as shown in Figure 17. (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
   [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 Twenty-FiveGigE 1/0/1 as the output interface for UDP packets.

   ```
   [DeviceA-nqa-admin-test1-udp-tracert] out interface twenty-fivegige 1/0/1
   ```

   # 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
```

Verifying the configuration

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:
    TTL     Hop IP              Time
    1       3.1.1.1             2013-09-09 14:46:03.2
    2       10.2.2.2            2013-09-09 14:46:06.2
```

Display the history records of the UDP tracert operation.

```
[DeviceA] display nqa history admin test1
NQA entry (admin admin, tag test1) history records:
Index      TTL  Response  Hop IP          Status          Time
1          2    2         10.2.2.2        Succeeded       2013-09-09 14:46:06.2
1          2    1         10.2.2.2        Succeeded       2013-09-09 14:46:05.2
1          2    2         10.2.2.2        Succeeded       2013-09-09 14:46:04.2
1          1    1         3.1.1.1         Succeeded       2013-09-09 14:46:03.2
1          1    2         3.1.1.1         Succeeded       2013-09-09 14:46:02.2
1          1    1         3.1.1.1         Succeeded       2013-09-09 14:46:01.2
```

Example: Configuring the voice operation

Network configuration

As shown in Figure 18, configure a voice operation to test jitters, delay, MOS, and ICPIF between Device A and Device B.

Figure 18 Network diagram

Procedure

1. Assign IP addresses to interfaces, as shown in Figure 18. (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 UDP port 9000 on IP address 10.2.2.2.

```
[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
```

   # Specify 10.2.2.2 as the destination IP address.

```
[DeviceA-nqa-admin-test1-voice] destination ip 10.2.2.2
```

   # Set the destination port number to 9000.

```
[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
```

Verifying the configuration

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
    SD average: 2                          DS average: 6
  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
    SD average: 16                         DS average: 2
  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
```

Example: Configuring the DLSw operation

Network configuration

As shown in Figure 19, configure a DLSw operation to test the response time of the DLSw device.

Figure 19 Network diagram

Procedure

Assign IP addresses to interfaces, as shown in Figure 19. (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
```

Specify 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
```

Verifying the configuration

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
```

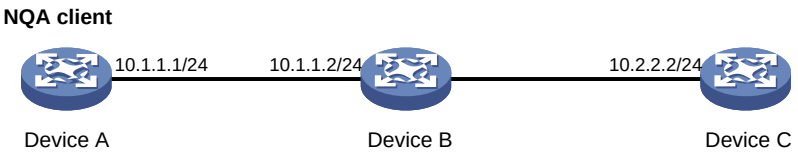
The output shows that the response time of the DLSw device is 19 milliseconds.

Example: Configuring the path jitter operation

Network configuration

As shown in Figure 20, configure a path jitter operation to test the round trip time and jitters from Device A to Device B and Device C.

Figure 20 Network diagram



Procedure

Assign IP addresses to interfaces, as shown in Figure 20. (Details not shown.)

Configure static routes or a routing protocol to make sure the devices can reach each other. (Details not shown.)

Execute the **ip ttl-expires enable** command on Device B and execute 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 every 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
```

Verifying the configuration

Display the most recent result of the path jitter operation.

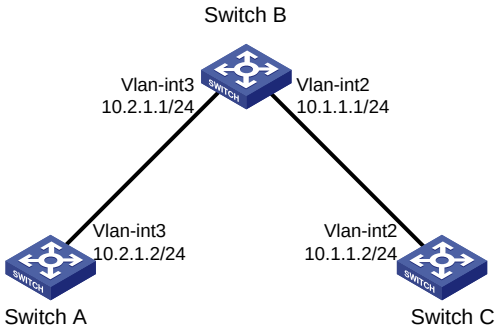
```
[DeviceA] display nqa result admin test1
NQA entry (admin admin, tag test1) test results:
  Hop IP 10.1.1.2
    Basic Results
      Send operation times: 10             Receive response times: 10
      Min/Max/Average round trip time: 9/21/14
      Square-Sum of round trip time: 2419
    Extended 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
```

Example: Configuring NQA collaboration

Network configuration

As shown in Figure 21, 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 21 Network diagram



Procedure

1. Assign IP addresses to interfaces, as shown in Figure 21. (Details not shown.)
2. On Switch A, configure a static route, and associate the static route with track entry 1.

```
<SwitchA> system-view
[SwitchA] ip route-static 10.1.1.2 24 10.2.1.1 track 1
```

3. On Switch A, configure an ICMP echo operation:

Create an NQA operation with 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
```

Specify 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 every 100 milliseconds.

```
[SwitchA-nqa-admin-test1-icmp-echo] frequency 100
```

Create reaction entry 1. If the number of consecutive probe failures reaches 5, collaboration is triggered.

```
[SwitchA-nqa-admin-test1-icmp-echo] reaction 1 checked-element probe-fail
threshold-type consecutive 5 action-type trigger-only
[SwitchA-nqa-admin-test1-icmp-echo] quit
```

Start the ICMP operation.

```
[SwitchA] nqa schedule admin test1 start-time now lifetime forever
```

4. On Switch A, create track entry 1, and associate it with reaction entry 1 of the NQA operation.

```
[SwitchA] track 1 nqa entry admin test1 reaction 1
```

Verifying the configuration

Display information about all the track entries on Switch A.

```
[SwitchA] display track all
Track ID: 1
  State: Positive
  Duration: 0 days 0 hours 0 minutes 0 seconds
```

```
  Notification delay: Positive 0, Negative 0 (in seconds)
  Tracked object:
    NQA entry: admin test1
    Reaction: 1
```

Display brief information about active routes in the routing table on Switch A.

```
[SwitchA] display ip routing-table

Destinations : 13        Routes : 13

Destination/Mask    Proto  Pre  Cost         NextHop         Interface
0.0.0.0/32          Direct 0    0            127.0.0.1       InLoop0
10.1.1.0/24         Static 60   0            10.2.1.1        Vlan3
10.2.1.0/24         Direct 0    0            10.2.1.2        Vlan3
10.2.1.0/32         Direct 0    0            10.2.1.2        Vlan3
10.2.1.2/32         Direct 0    0            127.0.0.1       InLoop0
10.2.1.255/32       Direct 0    0            10.2.1.2        Vlan3
127.0.0.0/8         Direct 0    0            127.0.0.1       InLoop0
127.0.0.0/32        Direct 0    0            127.0.0.1       InLoop0
127.0.0.1/32        Direct 0    0            127.0.0.1       InLoop0
127.255.255.255/32  Direct 0    0            127.0.0.1       InLoop0
224.0.0.0/4         Direct 0    0            0.0.0.0         NULL0
224.0.0.0/24        Direct 0    0            0.0.0.0         NULL0
255.255.255.255/32  Direct 0    0            127.0.0.1       InLoop0
```

The output shows that the static route with the next hop 10.2.1.1 is active, and the status of the track entry is positive.

Remove the IP address of VLAN-interface 3 on Switch B.

```
<SwitchB> system-view
[SwitchB] interface vlan-interface 3
[SwitchB-Vlan-interface3] undo ip address
```

Display information about all the track entries on Switch A.

```
[SwitchA] display track all
Track ID: 1
  State: Negative
  Duration: 0 days 0 hours 0 minutes 0 seconds
  Notification delay: Positive 0, Negative 0 (in seconds)
  Tracked object:
    NQA entry: admin test1
    Reaction: 1
```

Display brief information about active routes in the routing table on Switch A.

```
[SwitchA] display ip routing-table

Destinations : 12        Routes : 12

Destination/Mask    Proto  Pre  Cost         NextHop         Interface
0.0.0.0/32          Direct 0    0            127.0.0.1       InLoop0
10.2.1.0/24         Direct 0    0            10.2.1.2        Vlan3
10.2.1.0/32         Direct 0    0            10.2.1.2        Vlan3
```

```
10.2.1.2/32         Direct 0    0           127.0.0.1       InLoop0
10.2.1.255/32       Direct 0    0           10.2.1.2        Vlan3
127.0.0.0/8         Direct 0    0           127.0.0.1       InLoop0
127.0.0.0/32        Direct 0    0           127.0.0.1       InLoop0
127.0.0.1/32        Direct 0    0           127.0.0.1       InLoop0
127.255.255.255/32  Direct 0    0           127.0.0.1       InLoop0
224.0.0.0/4         Direct 0    0           0.0.0.0         NULL0
224.0.0.0/24        Direct 0    0           0.0.0.0         NULL0
255.255.255.255/32  Direct 0    0           127.0.0.1       InLoop0
```

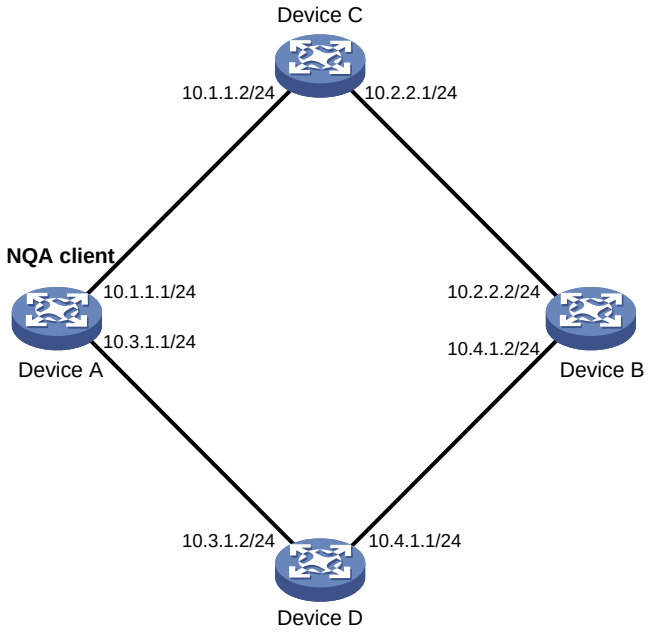
The output shows that the static route does not exist, and the status of the track entry is negative.

Example: Configuring the ICMP template

Network configuration

As shown in Figure 22, configure an ICMP template for a feature to perform the ICMP echo operation from Device A to Device B.

Figure 22 Network diagram



Procedure

Assign IP addresses to interfaces, as shown in Figure 22. (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 to 500 milliseconds for the ICMP echo operation.

```
[DeviceA-nqatplt-icmp-icmp] probe timeout 500
```

Configure the ICMP echo operation to repeat every 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
```

Example: Configuring the DNS template

Network configuration

As shown in Figure 23, 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 23 Network diagram

Procedure

Assign IP addresses to interfaces, as shown in Figure 23. (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 **host.com** as the domain name to be translated.

```
[DeviceA-nqatplt-dns-dns] resolve-target host.com
```

Set the domain name resolution type to type A.

```
[DeviceA-nqatplt-dns-dns] resolve-type A
```

Specify 3.3.3.3 as the expected IP address.

```
[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
```

Example: Configuring the TCP template

Network configuration

As shown in Figure 24, 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 24 Network diagram

Procedure

1. Assign IP addresses to interfaces, as shown in Figure 24. (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 TCP port 9000 on IP address 10.2.2.2.

   ```
   [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
   ```

   # Specify 10.2.2.2 as the destination IP address.

   ```
   [DeviceA-nqatplt-tcp-tcp] destination ip 10.2.2.2
   ```

   # Set the destination port number to 9000.

   ```
   [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
   ```

Example: Configuring the TCP half open template

Network configuration

As shown in Figure 25, configure a TCP half open template for a feature to test whether Device B can provide the TCP service for Device A.

Figure 25 Network diagram

Procedure

1. Assign IP addresses to interfaces, as shown in Figure 25. (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 A:

   # Create TCP half open template **test**.

   ```
   <DeviceA> system-view
   [DeviceA] nqa template tcphalfopen test
   ```

   # Specify 10.2.2.2 as the destination IP address.

   ```
   [DeviceA-nqatplt-tcphalfopen-test] destination ip 10.2.2.2
   ```

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

   ```
   [DeviceA-nqatplt-tcphalfopen-test] 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-tcphalfopen-test] reaction trigger probe-fail 2
   ```

Example: Configuring the UDP template

Network configuration

As shown in Figure 26, 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.

Figure 26 Network diagram

Procedure

1. Assign IP addresses to interfaces, as shown in Figure 26. (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 UDP port 9000 on IP address 10.2.2.2.

   ```
   [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
```

Example: Configuring the HTTP template

Network configuration

As shown in Figure 27, configure an HTTP template for a feature to perform the HTTP operation. The operation tests whether the NQA client can get data from the HTTP server.

Figure 27 Network diagram

Procedure

Assign IP addresses to interfaces, as shown in Figure 27. (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 **http://10.2.2.2/index.htm** as the URL of the HTTP server.

```
[DeviceA-nqatplt-http-http] url http://10.2.2.2/index.htm
```

Set the HTTP operation type to **get**.

```
[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.

```
[DeviceA-nqatplt-http-http] 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.

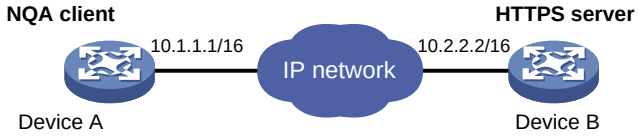
```
[DeviceA-nqatplt-http-http] reaction trigger probe-fail 2
```

Example: Configuring the HTTPS template

Network configuration

As shown in Figure 28, configure an HTTPS template for a feature to test whether the NQA client can get data from the HTTPS server (Device B).

Figure 28 Network diagram



Procedure

Assign IP addresses to interfaces, as shown in Figure 28. (Details not shown.)

Configure static routes or a routing protocol to make sure the devices can reach each other. (Details not shown.)

Configure an SSL client policy named **abc** on Device A, and make sure Device A can use the policy to connect to the HTTPS server. (Details not shown.)

Create HTTPS template **test**.

```
<DeviceA> system-view
[DeviceA] nqa template https https
```

Specify **http://10.2.2.2/index.htm** as the URL of the HTTPS server.

```
[DeviceA-nqatplt-https-https] url https://10.2.2.2/index.htm
```

Specify SSL client policy **abc** for the HTTPS template.

```
[DeviceA-nqatplt-https- https] ssl-client-policy abc
```

Set the HTTPS operation type to **get** (the default HTTPS operation type).

```
[DeviceA-nqatplt-https-https] operation get
```

Set the HTTPS version to 1.0 (the default HTTPS version).

```
 [DeviceA-nqatplt-https-https] version v1.0
```

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

```
[DeviceA-nqatplt-https-https] 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-https-https] reaction trigger probe-fail 2
```

Example: Configuring the FTP template

Network configuration

As shown in Figure 29, 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 29 Network diagram

Procedure

Assign IP addresses to interfaces, as shown in Figure 29. (Details not shown.)

Configure static routes or a routing protocol to make sure the devices can reach each other. (Details not shown.)

Create FTP template **ftp**.

```
<DeviceA> system-view
[DeviceA] nqa template ftp ftp
```

Specify the URL of the FTP server.

```
[DeviceA-nqatplt-ftp-ftp] url ftp://10.2.2.2
```

Specify 10.1.1.1 as the source IP address.

```
[DeviceA-nqatplt-ftp-ftp] source ip 10.1.1.1
```

Configure the device to upload file **config.txt** to the FTP server.

```
[DeviceA-nqatplt-ftp-ftp] operation put
[DeviceA-nqatplt-ftp-ftp] filename config.txt
```

Set the username to **admin** for the FTP server login.

```
[DeviceA-nqatplt-ftp-ftp] username admin
```

Set the password to **systemtest** for the FTP server login.

```
[DeviceA-nqatplt-ftp-ftp] password simple systemtest
```

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

```
[DeviceA-nqatplt-ftp-ftp] 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-ftp-ftp] reaction trigger probe-fail 2
```

Example: Configuring the RADIUS template

Network configuration

As shown in Figure 30, configure a RADIUS template for a feature to test whether the RADIUS server (Device B) can provide authentication service for Device A. The username and password are **admin** and **systemtest**, respectively. The shared key is **123456** for secure RADIUS authentication.

Figure 30 Network diagram

Procedure

Assign IP addresses to interfaces, as shown in Figure 30. (Details not shown.)

Configure static routes or a routing protocol to make sure the devices can reach each other. (Details not shown.)

Configure the RADIUS server. (Details not shown.)

Create RADIUS template **radius**.

```
<DeviceA> system-view
[DeviceA] nqa template radius radius
```

Specify 10.2.2.2 as the destination IP address of the operation.

```
[DeviceA-nqatplt-radius-radius] destination ip 10.2.2.2
```

Set the username to **admin**.

```
[DeviceA-nqatplt-radius-radius] username admin
```

Set the password to **systemtest**.

```
[DeviceA-nqatplt-radius-radius] password simple systemtest
```

Set the shared key to **123456** in plain text for secure RADIUS authentication.

```
[DeviceA-nqatplt-radius-radius] key simple 123456
```

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

```
[DeviceA-nqatplt-radius-radius] 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-radius-radius] reaction trigger probe-fail 2
```

Example: Configuring the SSL template

Network configuration

As shown in Figure 31, configure an SSL template for a feature to test whether Device A can establish an SSL connection to the SSL server on Device B.

Figure 31 Network diagram

Procedure

Assign IP addresses to interfaces, as shown in Figure 31. (Details not shown.)

Configure static routes or a routing protocol to make sure the devices can reach each other. (Details not shown.)

Configure an SSL client policy named **abc** on Device A, and make sure Device A can use the policy to connect to the SSL server on Device B. (Details not shown.)

Create SSL template **ssl**.

```
<DeviceA> system-view
[DeviceA] nqa template ssl ssl
```

Set the destination IP address and port number to **10.2.2.2** and **9000**, respectively.

```
[DeviceA-nqatplt-ssl-ssl] destination ip 10.2.2.2
[DeviceA-nqatplt-ssl-ssl] destination port 9000
```

Specify SSL client policy **abc** for the SSL template.

```
[DeviceA-nqatplt-ssl-ssl] ssl-client-policy abc
```

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

```
[DeviceA-nqatplt-ssl-ssl] 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-ssl-ssl] reaction trigger probe-fail 2
```

Configuring NTP

About NTP

NTP is used to synchronize system clocks among distributed time servers and clients on a network. NTP runs over UDP and uses UDP port 123.

NTP application scenarios

Various tasks, including network management, charging, auditing, and distributed computing depend on accurate and synchronized system time setting on the network devices. NTP is typically used in large networks to dynamically synchronize time among network devices.

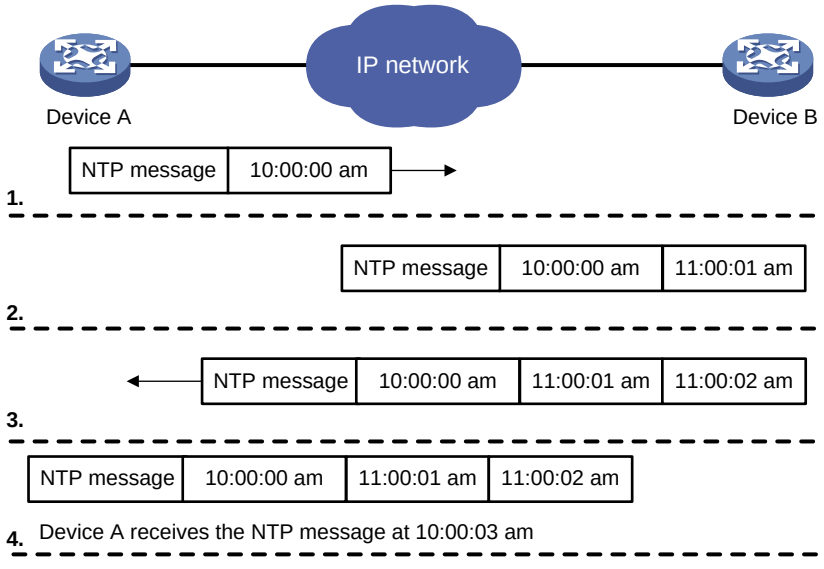
NTP guarantees higher clock accuracy than manual system clock setting. In a small network that does not require high clock accuracy, you can keep time synchronized among devices by changing their system clocks one by one.

NTP working mechanism

Figure 32 shows how NTP synchronizes the system time between two devices (Device A and Device B, in this example). Assume that:

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

Figure 32 Basic work flow



The synchronization process is as follows:

1. Device A sends Device B an NTP message, which is timestamped when it leaves Device A. The time stamp is 10:00:00 am (T1).

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

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

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

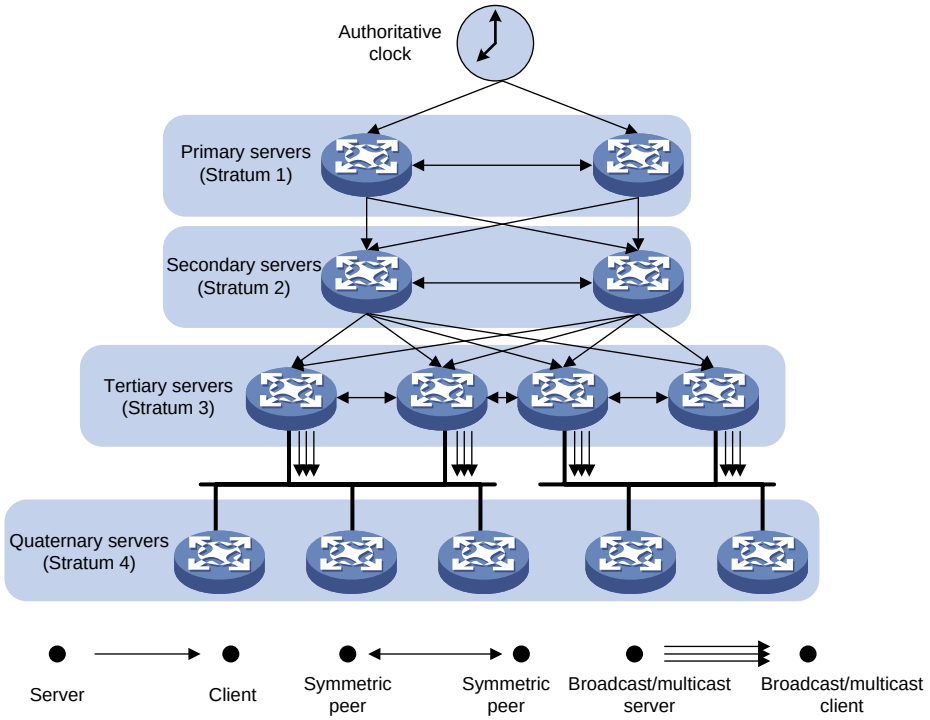
Based on these parameters, Device A can be synchronized to Device B.

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

NTP architecture

NTP uses stratums 1 to 16 to define clock accuracy, as shown in Figure 33. A lower stratum value represents higher accuracy. Clocks at stratums 1 through 15 are in synchronized state, and clocks at stratum 16 are not synchronized.

Figure 33 NTP architecture



A stratum 1 NTP server gets its time from an authoritative time source, such as an atomic clock. It provides time for other devices as the primary NTP 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 perform the following tasks:

- Select a device that has a relatively accurate clock from the network.
- Use the local clock of the device as the reference clock to synchronize other devices in the network.

NTP association modes

NTP supports the following association modes:

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

You can select one or more association modes for time synchronization. Table 2 provides detailed description for the four association modes.

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

Table 2 NTP association modes

Mode	Working process	Principle	Application scenario
Client/server	On the client, specify the IP address of the NTP server. A client sends a clock synchronization message to the NTP servers. Upon receiving the message, the servers automatically operate in server mode and send a reply. If the client can be synchronized to multiple time servers, it selects an optimal clock and synchronizes its local clock to the optimal reference source after receiving the replies from the servers.	A client can synchronize to a server, but a server cannot synchronize to a client.	As Figure 33 shows, this mode is intended for configurations where devices of a higher stratum synchronize to devices with a lower stratum.
Symmetric active/passive	On the symmetric active peer, specify the IP address of the symmetric passive peer. A symmetric active peer periodically sends clock synchronization messages to a symmetric passive peer. The symmetric passive peer automatically operates in symmetric passive mode and sends a reply. If the symmetric active peer can be synchronized to multiple time servers, it selects an optimal clock and synchronizes its local clock to the optimal reference source after receiving the replies from the servers.	A symmetric active peer and a symmetric passive peer can be synchronized to each other. If both of them are synchronized, the peer with a higher stratum is synchronized to the peer with a lower stratum.	As Figure 33 shows, this mode is most often used between 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 lower stratum, the server can still synchronize to the servers of the same stratum.

Mode	Working process	Principle	Application scenario
Broadcast	A server periodically sends clock synchronization messages to the broadcast address 255.255.255.255. Clients listen to the broadcast messages from the servers to synchronize to the server according to the broadcast messages. When a client receives the first broadcast message, the client and the server start to exchange messages to calculate the network delay between them. Then, only the broadcast server sends clock synchronization messages.	A broadcast client can synchronize to a broadcast server, but a broadcast server cannot synchronize to a broadcast client.	A broadcast server sends clock synchronization messages to synchronize clients in the same subnet. As Figure 33 shows, broadcast mode is intended for configurations involving one or a few servers and a potentially large client population. The broadcast mode has lower time accuracy than the client/server and symmetric active/passive modes because only the broadcast servers send clock synchronization messages.
Multicast	A multicast server periodically sends clock synchronization messages to the user-configured multicast address. Clients listen to the multicast messages from servers and synchronize to the server according to the received messages.	A multicast client can synchronize to a multicast server, but a multicast server cannot synchronize to a multicast client.	A multicast server can provide time synchronization for clients in the same subnet or in different subnets. The multicast mode has lower time accuracy than the client/server and symmetric active/passive modes.

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 the least restrictive to the 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.

When the device receives an NTP request, it matches the request against the access rights in order from the least restrictive to the most restrictive: **peer**, **server**, **synchronization**, and **query**.

- If no NTP access control is configured, the **peer** access right applies.
- If the IP address of the peer device matches a **permit** statement in an ACL, the 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 specified for an access right or the ACL specified for the access right is not created, the access right is not granted.

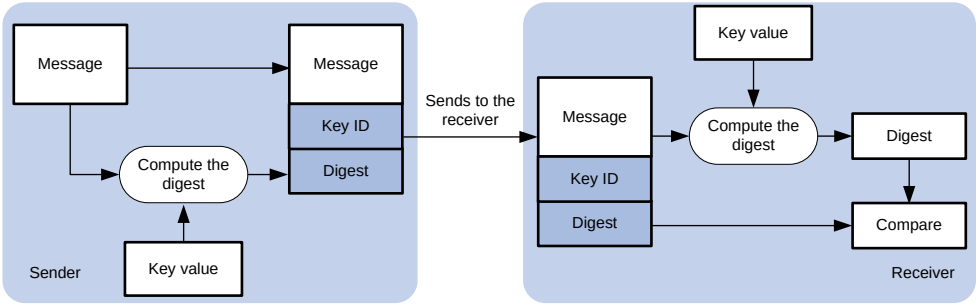
- If none of the ACLs specified for the access rights is created, the **peer** access right applies.
- If none of the ACLs specified for the access rights contains rules, no access right is granted.

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 34 NTP authentication



As shown in Figure 34, NTP authentication is performed as follows:

1. The sender uses the key identified by the key ID to calculate a digest for the NTP message through the MD5/HMAC authentication algorithm. Then it sends the calculated digest together with the NTP message and key ID to the receiver.
2. Upon receiving the message, the receiver performs the following actions:
   a. Finds the key according to the key ID in the message.
   b. Uses the key and the MD5/HMAC authentication algorithm to calculate the digest for the message.
   c. Compares the digest with the digest contained in the NTP message.
      - If they are different, the receiver discards the message.
      - If they are the same and an NTP association is not required to be established, the receiver provides a response packet. For information about NTP associations, see "Configuring the maximum number of dynamic associations."
      - If they are the same and an NTP association is required to be established or has existed, the local device determines whether the sender is allowed to use the authentication ID. If the sender is allowed to use the authentication ID, the receiver accepts the message. If the sender is not allowed to use the authentication ID, the receiver discards the message.

NTP for MPLS L3VPN instances

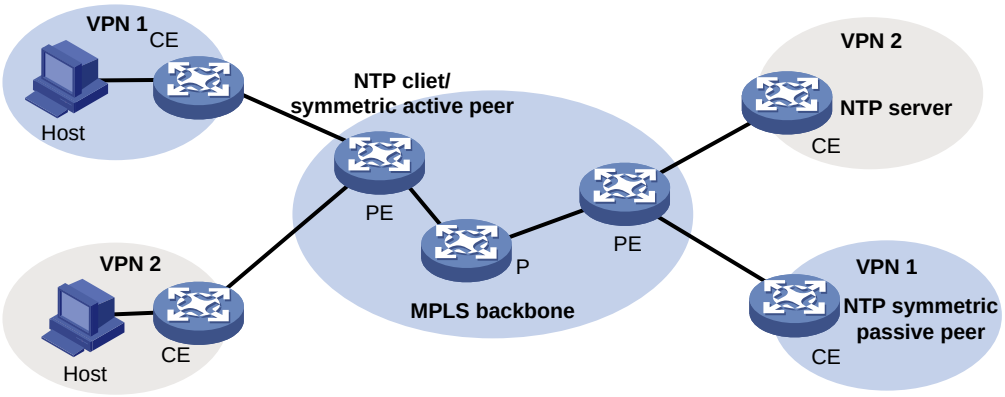
On an MPLS L3VPN network, a PE that acts as an NTP client or active peer can synchronize with the NTP server or passive peer in an MPLS L3VPN instance.

As shown in Figure 35, users in VPN 1 and VPN 2 are connected to the MPLS backbone network through provider edge (PE) devices. VPN instances vpn1 and vpn2 have been created for VPN 1 and VPN 2, respectively on the PEs. Services of the two VPN instances are isolated. Time synchronization between PEs and devices in the two VPN instances can be realized if you perform the following tasks:

- Configure the PEs to operate in NTP client or symmetric active mode.

- Specify the VPN instance to which the NTP server or NTP symmetric passive peer belongs.

Figure 35 Network diagram



For more information about MPLS L3VPN, VPN instance, and PE, see *MPLS Configuration Guide*.

Protocols and standards

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

Restrictions and guidelines: NTP configuration

- You cannot configure both NTP and SNTP on the same device.
- NTP is supported only on the following Layer 3 interfaces:
  - Layer 3 Ethernet interfaces.
  - Layer 3 Ethernet subinterfaces.
  - Layer 3 aggregate interfaces.
  - Layer 3 aggregate subinterfaces.
  - VLAN interfaces.
  - Tunnel interfaces.
- Do not configure NTP on an aggregate member port.
- The NTP service and SNTP service are mutually exclusive. You can only enable either NTP service or SNTP service at a time.
- To avoid frequent time changes or even synchronization failures, do not specify more than one reference source on a network.
- To use NTP for time synchronization, you must use the **`clock protocol`** command to specify NTP for obtaining the time. For more information about the **`clock protocol`** command, see device management commands in *Fundamentals Command Reference*.

NTP tasks at a glance

To configure NTP, perform the following tasks:

1. Enabling the NTP service
2. Configuring NTP association mode
   - Configuring NTP in client/server mode

- Configuring NTP in symmetric active/passive mode
- Configuring NTP in broadcast mode
- Configuring NTP in multicast mode

3. (Optional.) Configuring the local clock as the reference source
4. (Optional.) Configuring access control rights
5. (Optional.) 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
6. (Optional.) Controlling NTP message sending and receiving
   - Specifying a source address for NTP messages
   - Disabling an interface from receiving NTP messages
   - Configuring the maximum number of dynamic associations
   - Setting a DSCP value for NTP packets
7. (Optional.) Specifying the NTP time-offset thresholds for log and trap outputs

Enabling the NTP service

Restrictions and guidelines

NTP and SNTP are mutually exclusive. Before you enable NTP, make sure SNTP is disabled.

Procedure

1. Enter system view.
   `system-view`
2. Enable the NTP service.
   `ntp-service enable`
   By default, the NTP service is disabled.

Configuring NTP association mode

Configuring NTP in client/server mode

Restrictions and guidelines

To configure NTP in client/server mode, specify an NTP server for the client.

For a client to synchronize to an NTP server, make sure the server is synchronized by other devices or uses its local clock as the reference source.

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 specify multiple servers for a client by executing the **`ntp-service unicast-server`** or **`ntp-service ipv6 unicast-server`** command multiple times.

Procedure

1. Enter system view.
   `system-view`

2. Specify an NTP server for the device.

   IPv4:

   ntp-service unicast-server { *server-name* | *ip-address* } [**vpn-instance** *vpn-instance-name*] [**authentication-keyid** *keyid* | **maxpoll** *maxpoll-interval* | **minpoll** *minpoll-interval* | **priority** | **source** *interface-type interface-number* | **version** *number*] *

   IPv6:

   ntp-service ipv6 unicast-server { *server-name* | *ipv6-address* } [**vpn-instance** *vpn-instance-name*] [**authentication-keyid** *keyid* | **maxpoll** *maxpoll-interval* | **minpoll** *minpoll-interval* | **priority** | **source** *interface-type interface-number*] *

   By default, no NTP server is specified.

Configuring NTP in symmetric active/passive mode

Restrictions and guidelines

To configure NTP in symmetric active/passive mode, specify a symmetric passive peer for the active peer.

For a symmetric passive peer to process NTP messages from a symmetric active peer, execute the **ntp-service enable** command on the symmetric passive peer to enable NTP.

For time synchronization between the symmetric active peer and the symmetric passive peer, make sure either or both of them are in synchronized state.

You can specify multiple symmetric passive peers by executing the **ntp-service unicast-peer** or **ntp-service ipv6 unicast-peer** command multiple times.

Procedure

1. Enter system view.

   system-view

2. Specify a symmetric passive peer for the device.

   IPv4:

   ntp-service unicast-peer { *peer-name* | *ip-address* } [**vpn-instance** *vpn-instance-name*] [**authentication-keyid** *keyid* | **maxpoll** *maxpoll-interval* | **minpoll** *minpoll-interval* | **priority** | **source** *interface-type interface-number* | **version** *number*] *

   IPv6:

   ntp-service ipv6 unicast-peer { *peer-name* | *ipv6-address* } [**vpn-instance** *vpn-instance-name*] [**authentication-keyid** *keyid* | **maxpoll** *maxpoll-interval* | **minpoll** *minpoll-interval* | **priority** | **source** *interface-type interface-number*] *

   By default, no symmetric passive peer is specified.

Configuring NTP in broadcast mode

Restrictions and guidelines

To configure NTP in broadcast mode, you must configure an NTP broadcast client and an NTP broadcast server.

For a broadcast client to synchronize to a broadcast server, make sure the broadcast server is synchronized by other devices or uses its local clock as the reference source.

Configuring the broadcast client

1. Enter system view.

   system-view

2. Enter interface view.

   interface *interface-type interface-number*

3. Configure the device to operate in broadcast client mode.

   ntp-service broadcast-client

   By default, the device does not operate in any NTP association mode.

   After you execute the command, the device receives NTP broadcast messages from the specified interface.

Configuring the broadcast server

1. Enter system view.

   system-view

2. Enter interface view.

   interface *interface-type interface-number*

3. Configure the device to operate in NTP broadcast server mode.

   ntp-service broadcast-server [**authentication-keyid** *keyid* | **version** *number*] *

   By default, the device does not operate in any NTP association mode.

   After you execute the command, the device sends NTP broadcast messages from the specified interface.

Configuring NTP in multicast mode

Restrictions and guidelines

To configure NTP in multicast mode, you must configure an NTP multicast client and an NTP multicast server.

For a multicast client to synchronize to a multicast server, make sure the multicast server is synchronized by other devices or uses its local clock as the reference source.

Configuring a multicast client

1. Enter system view.

   system-view

2. Enter interface view.

   interface *interface-type interface-number*

3. Configure the device to operate in multicast client mode.

   IPv4:

   ntp-service multicast-client [*ip-address*]

   IPv6:

   ntp-service ipv6 multicast-client *ipv6-address*

   By default, the device does not operate in any NTP association mode.

   After you execute the command, the device receives NTP multicast messages from the specified interface.

Configuring the multicast server

1. Enter system view.

system-view

2. Enter interface view.

   interface *interface-type interface-number*

3. Configure the device to operate in multicast server mode.

   IPv4:

   ntp-service multicast-server [*ip-address*][**authentication-keyid** *keyid* | **ttl** *ttl-number* | **version** *number*]*

   IPv6:

   ntp-service ipv6 multicast-server *ipv6-address* [**authentication-keyid** *keyid* | **ttl** *ttl-number*] *

   By default, the device does not operate in any NTP association mode.

   After you execute the command, the device sends NTP multicast messages from the specified interface.

Configuring the local clock as the reference source

About this task

This task enables the device to use the local clock as the reference so that the device is synchronized.

Restrictions and guidelines

Make sure the local clock can provide the time accuracy required for the network. After you configure the local clock as the 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.

The system time reverts to the initial BIOS default after a cold reboot. As a best practice, do not configure the local clock as the reference source or configure the device as a time server.

Devices differ in clock precision. As a best practice to avoid network flapping and clock synchronization failure, configure only one reference clock on the same network segment and make sure the clock has high precision.

Prerequisites

Before you configure this feature, adjust the local system time to ensure that it is accurate.

Procedure

1. Enter system view.

   system-view

2. Configure the local clock as the reference source.

   ntp-service refclock-master [*ip-address*] [*stratum*]

   By default, the device does not use the local clock as the reference source.

Configuring access control rights

Prerequisites

Before you configure the right for peer devices to access the NTP services on the local device, create and configure ACLs associated with the access right. For information about configuring an ACL, see *ACL and QoS Configuration Guide*.

Procedure

1. Enter system view.
   system-view
2. Configure the right for peer devices to access the NTP services on the local device.
   IPv4:
   ntp-service access { **peer** | **query** | **server** | **synchronization** } **acl** *ipv4-acl-number*
   IPv6:
   ntp-service ipv6 { **peer** | **query** | **server** | **synchronization** } **acl** *ipv6-acl-number*
   By default, the right for peer devices to access the NTP services on the local device is **peer**.

Configuring NTP authentication

Configuring NTP authentication in client/server mode

Restrictions and guidelines

To ensure a successful NTP authentication in client/server mode, configure the same authentication key ID, algorithm, and key on the server and client. Make sure the peer device is allowed to use the key ID for authentication on the local device.

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

Table 3 NTP authentication results

Client			Server	
Enable NTP authentication	**Specify the server and key**	**Trusted key**	**Enable NTP authentication**	**Trusted key**
Successful authentication				
Yes	Yes	Yes	Yes	Yes
Failed authentication				
Yes	Yes	Yes	Yes	No
Yes	Yes	Yes	No	N/A
Yes	Yes	No	N/A	N/A
Authentication not performed				
Yes	No	N/A	N/A	N/A
No	N/A	N/A	N/A	N/A

Configuring NTP authentication for a client

1. Enter system view.
   system-view
2. Enable NTP authentication.
   ntp-service authentication enable
   By default, NTP authentication is disabled.

3. Configure an NTP authentication key.

   ntp-service authentication-keyid *keyid* **authentication-mode** { **hmac-sha-1** | **hmac-sha-256** | **hmac-sha-384** | **hmac-sha-512** | **md5** } { **cipher** | **simple** } *string* [**acl** *ipv4-acl-number* | **ipv6 acl** *ipv6-acl-number*] *

   By default, no NTP authentication key exists.

4. Configure the key as a trusted key.

   ntp-service reliable authentication-keyid *keyid*

   By default, no authentication key is configured as a trusted key.

5. Associate the specified key with an NTP server.

   IPv4:

   ntp-service unicast-server { *server-name* | *ip-address* } [**vpn-instance** *vpn-instance-name*] **authentication-keyid** *keyid*

   IPv6:

   ntp-service ipv6 unicast-server { *server-name* | *ipv6-address* } [**vpn-instance** *vpn-instance-name*] **authentication-keyid** *keyid*

Configuring NTP authentication for a server

1. Enter system view.

   system-view

2. Enable NTP authentication.

   ntp-service authentication enable

   By default, NTP authentication is disabled.

3. Configure an NTP authentication key.

   ntp-service authentication-keyid *keyid* **authentication-mode** { **hmac-sha-1** | **hmac-sha-256** | **hmac-sha-384** | **hmac-sha-512** | **md5** } { **cipher** | **simple** } *string* [**acl** *ipv4-acl-number* | **ipv6 acl** *ipv6-acl-number*] *

   By default, no NTP authentication key exists.

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.

Configuring NTP authentication in symmetric active/passive mode

Restrictions and guidelines

To ensure a successful NTP authentication in symmetric active/passive mode, configure the same authentication key ID, algorithm, and key on the active peer and passive peer. Make sure the peer device is allowed to use the key ID for authentication on the local device.

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

Table 4 NTP authentication results

Active peer				Passive peer	
Enable NTP authentication	**Specify the peer and key**	**Trusted key**	**Stratum level**	**Enable NTP authentication**	**Trusted key**
Successful authentication					
Yes	Yes	Yes	N/A	Yes	Yes
Failed authentication					
Yes	Yes	Yes	N/A	Yes	No
Yes	Yes	Yes	N/A	No	N/A
Yes	No	N/A	N/A	Yes	N/A
No	N/A	N/A	N/A	Yes	N/A
Yes	Yes	No	Larger than the passive peer	N/A	N/A
Yes	Yes	No	Smaller than the passive peer	Yes	N/A
Authentication not performed					
Yes	No	N/A	N/A	No	N/A
No	N/A	N/A	N/A	No	N/A
Yes	Yes	No	Smaller than the passive peer	No	N/A

Configuring NTP authentication for an active peer

1. Enter system view.

   system-view

2. Enable NTP authentication.

   ntp-service authentication enable

   By default, NTP authentication is disabled.

3. Configure an NTP authentication key.

   ntp-service authentication-keyid *keyid* **authentication-mode** { **hmac-sha-1** | **hmac-sha-256** | **hmac-sha-384** | **hmac-sha-512** | **md5** } { **cipher** | **simple** } *string* [**acl** *ipv4-acl-number* | **ipv6 acl** *ipv6-acl-number*] *

   By default, no NTP authentication key exists.

4. Configure the key as a trusted key.

   ntp-service reliable authentication-keyid *keyid*

   By default, no authentication key is configured as a trusted key.

5. Associate the specified key with a passive peer.

   IPv4:

   ntp-service unicast-peer { *ip-address* | *peer-name* } [**vpn-instance** *vpn-instance-name*] **authentication-keyid** *keyid*

   IPv6:

   ntp-service ipv6 unicast-peer { *ipv6-address* | *peer-name* } [**vpn-instance** *vpn-instance-name*] **authentication-keyid** *keyid*

Configuring NTP authentication for a passive peer

1. Enter system view.
   system-view
2. Enable NTP authentication.
   ntp-service authentication enable
   By default, NTP authentication is disabled.
3. Configure an NTP authentication key.
   ntp-service authentication-keyid *keyid* **authentication-mode** { **hmac-sha-1** | **hmac-sha-256** | **hmac-sha-384** | **hmac-sha-512** | **md5** } { **cipher** | **simple** } *string* [**acl** *ipv4-acl-number* | **ipv6 acl** *ipv6-acl-number*] *
   By default, no NTP authentication key exists.
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.

Configuring NTP authentication in broadcast mode

Restrictions and guidelines

To ensure a successful NTP authentication in broadcast mode, configure the same authentication key ID, algorithm, and key on the broadcast server and client. Make sure the peer device is allowed to use the key ID for authentication on the local device.

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 or not does not make any difference.)

Table 5 NTP authentication results

Broadcast server			**Broadcast client**	
Enable NTP authentication	**Specify the server and key**	**Trusted key**	**Enable NTP authentication**	**Trusted key**
Successful authentication				
Yes	Yes	Yes	Yes	Yes
Failed authentication				
Yes	Yes	Yes	Yes	No
Yes	Yes	Yes	No	N/A
Yes	Yes	No	Yes	N/A
Yes	No	N/A	Yes	N/A
No	N/A	N/A	Yes	N/A
Authentication not performed				
Yes	Yes	No	No	N/A
Yes	No	N/A	No	N/A
No	N/A	N/A	No	N/A

Configuring NTP authentication for a broadcast client

1. Enter system view.
   system-view
2. Enable NTP authentication.
   ntp-service authentication enable
   By default, NTP authentication is disabled.
3. Configure an NTP authentication key.
   ntp-service authentication-keyid *keyid* **authentication-mode** { **hmac-sha-1** | **hmac-sha-256** | **hmac-sha-384** | **hmac-sha-512** | **md5** } { **cipher** | **simple** } *string* [**acl** *ipv4-acl-number* | **ipv6 acl** *ipv6-acl-number*] *
   By default, no NTP authentication key exists.
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.

Configuring NTP authentication for a broadcast server

1. Enter system view.
   system-view
2. Enable NTP authentication.
   ntp-service authentication enable
   By default, NTP authentication is disabled.
3. Configure an NTP authentication key.
   ntp-service authentication-keyid *keyid* **authentication-mode** { **hmac-sha-1** | **hmac-sha-256** | **hmac-sha-384** | **hmac-sha-512** | **md5** } { **cipher** | **simple** } *string* [**acl** *ipv4-acl-number* | **ipv6 acl** *ipv6-acl-number*] *
   By default, no NTP authentication key exists.
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*
6. Associate the specified key with the broadcast server.
   ntp-service broadcast-server authentication-keyid *keyid*
   By default, the broadcast server is not associated with a key.

Configuring NTP authentication in multicast mode

Restrictions and guidelines

To ensure a successful NTP authentication in multicast mode, configure the same authentication key ID, algorithm, and key on the multicast server and client. Make sure the peer device is allowed to use the key ID for authentication on the local device.

NTP authentication results differ when different configurations are performed on broadcast client and server. For more information, see Table 6. (N/A in the table means that whether the configuration is performed or not does not make any difference.)

Table 6 NTP authentication results

Multicast server			Multicast client	
Enable NTP authentication	**Specify the server and key**	**Trusted key**	**Enable NTP authentication**	**Trusted key**
Successful authentication				
Yes	Yes	Yes	Yes	Yes
Failed authentication				
Yes	Yes	Yes	Yes	No
Yes	Yes	Yes	No	N/A
Yes	Yes	No	Yes	N/A
Yes	No	N/A	Yes	N/A
No	N/A	N/A	Yes	N/A
Authentication not performed				
Yes	Yes	No	No	N/A
Yes	No	N/A	No	N/A
No	N/A	N/A	No	N/A

Configuring NTP authentication for a multicast client

1. Enter system view.

   system-view
2. Enable NTP authentication.

   ntp-service authentication enable

   By default, NTP authentication is disabled.
3. Configure an NTP authentication key.

   ntp-service authentication-keyid *keyid* **authentication-mode** { **hmac-sha-1** | **hmac-sha-256** | **hmac-sha-384** | **hmac-sha-512** | **md5** } { **cipher** | **simple** } *string* [**acl** *ipv4-acl-number* | **ipv6 acl** *ipv6-acl-number*] *

   By default, no NTP authentication key exists.
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.

Configuring NTP authentication for a multicast server

1. Enter system view.

   system-view
2. Enable NTP authentication.

   ntp-service authentication enable

   By default, NTP authentication is disabled.
3. Configure an NTP authentication key.

   ntp-service authentication-keyid *keyid* **authentication-mode** { **hmac-sha-1** | **hmac-sha-256** | **hmac-sha-384** | **hmac-sha-512** | **md5** } { **cipher** | **simple** } *string* [**acl** *ipv4-acl-number* | **ipv6 acl** *ipv6-acl-number*] *

By default, no NTP authentication key exists.

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*

6. Associate the specified key with a multicast server.

   IPv4:

   ntp-service multicast-server [*ip-address*] **authentication-keyid** *keyid*

   IPv6:

   ntp-service ipv6 multicast-server *ipv6-multicast-address* **authentication-keyid** *keyid*

   By default, no multicast server is associated with the specified key.

Controlling NTP message sending and receiving

Specifying a source address for NTP messages

About this task

You can specify a source address for NTP messages directly or by specifying a source interface. If you specify a source interface for NTP messages, the device uses the IP address of the specified interface as the source address to send NTP messages.

Restrictions and guidelines

To prevent interface status changes from causing NTP communication failures, specify an interface that is always up as the source interface, a loopback interface for example.

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.

If you have specified the source interface for NTP messages in the **ntp-service unicast-server/ntp-service ipv6 unicast-server** or **ntp-service unicast-peer/ntp-service ipv6 unicast-peer** command, the IP address of the specified interface is used as the source IP address for NTP messages.

If you have configured the **ntp-service broadcast-server** or **ntp-service multicast-server/ntp-service ipv6 multicast-server** command in an interface view, the IP address of the interface is used as the source IP address for broadcast or multicast NTP messages.

Procedure

1. Enter system view.

   system-view

2. Specify the source address for NTP messages.

   IPv4:

   ntp-service source { *interface-type interface-number* | *ipv4-address* }

   IPv6:

   ntp-service ipv6 source *interface-type interface-number*

   By default, no source address is specified for NTP messages.

Disabling an interface from receiving NTP messages

About this task

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

Procedure

1. Enter system view.
   system-view
2. Enter interface view.
   interface *interface-type interface-number*
3. Disable the interface from receiving NTP packets.
   IPv4:
   undo ntp-service inbound enable
   IPv6:
   undo ntp-service ipv6 inbound enable
   By default, an interface receives NTP messages.

Configuring the maximum number of dynamic associations

About this task

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

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.

Restrictions and guidelines

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

Procedure

1. Enter system view.
   system-view
2. Configure the maximum number of dynamic sessions.
   ntp-service max-dynamic-sessions *number*
   By default, the maximum number of dynamic sessions is 100.

Setting a DSCP value for NTP packets

About this task

The DSCP value determines the sending precedence of an NTP packet.

Procedure

1. Enter system view.
   system-view
2. Set a DSCP value for NTP packets.
   IPv4:
   ntp-service dscp *dscp-value*
   IPv6:
   ntp-service ipv6 dscp *dscp-value*
   The default DSCP value is 48 for IPv4 packets and 56 for IPv6 packets.

Specifying the NTP time-offset thresholds for log and trap outputs

About this task

By default, the system synchronizes the NTP client's time to the server and outputs a log and a trap when the time offset exceeds 128 ms for multiple times.

After you set the NTP time-offset thresholds for log and trap outputs, the system synchronizes the client's time to the server when the time offset exceeds 128 ms for multiple times, but outputs logs and traps only when the time offset exceeds the specified thresholds, respectively.

Procedure

1. Enter system view.
   system-view
2. Specify the NTP time-offset thresholds for log and trap outputs.
   ntp-service time-offset-threshold { **log** *log-threshold* | **trap** *trap-threshold* } *
   By default, no NTP time-offset thresholds are set for log and trap outputs.

Display and maintenance commands for NTP

Execute **display** commands in any view.

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

NTP configuration examples

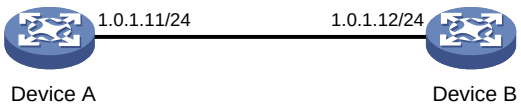
Example: Configuring NTP client/server association mode

Network configuration

As shown in Figure 36, perform the following tasks:

- Configure Device A's local clock as its reference source, with stratum level 2.
- Configure Device B to operate in client mode and specify Device A as the NTP server of Device B.

Figure 36 Network diagram



Procedure

1. Assign an IP address to each interface, and make sure Device A and Device B can reach each other, as shown in Figure 36. (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 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 NTP for obtaining the time.

   ```
   [DeviceB] clock protocol ntp
   ```

   # Specify Device A as the NTP server of Device B.

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

Verifying the configuration

Verify that Device B has synchronized its time with Device A, and the clock stratum level of Device B is 3.

```
[DeviceB] display ntp-service status
 Clock status: synchronized
 Clock stratum: 3
 System peer: 1.0.1.11
 Local mode: client
 Reference clock ID: 1.0.1.11
 Leap indicator: 00
 Clock jitter: 0.000977 s
 Stability: 0.000 pps
```

```
 Clock precision: 2^-22
 Root delay: 0.00383 ms
 Root dispersion: 16.26572 ms
 Reference time: d0c6033f.b9923965  Wed, Dec 29 2010 18:58:07.724
 System poll interval: 64 s
```

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

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

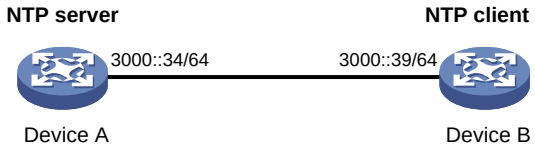
Example: Configuring IPv6 NTP client/server association mode

Network configuration

As shown in Figure 37, perform the following tasks:

- Configure Device A's local clock as its reference source, with stratum level 2.
- Configure Device B to operate in client mode and specify Device A as the IPv6 NTP server of Device B.

Figure 37 Network diagram



Procedure

1. Assign an IP address to each interface, and make sure Device A and Device B can reach each other, as shown in Figure 37. (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 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 NTP for obtaining the time.

   ```
   [DeviceB] clock protocol ntp
   ```

   # Specify Device A as the IPv6 NTP server of Device B.

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

Verifying the configuration

Verify that Device B has synchronized its time with Device A, and the clock stratum level of Device B is 3.

```
[DeviceB] display ntp-service status
 Clock status: synchronized
 Clock stratum: 3
 System peer: 3000::34
 Local mode: client
 Reference clock ID: 163.29.247.19
 Leap indicator: 00
 Clock jitter: 0.000977 s
 Stability: 0.000 pps
 Clock precision: 2^-22
 Root delay: 0.02649 ms
 Root dispersion: 12.24641 ms
 Reference time: d0c60419.9952fb3e  Wed, Dec 29 2010 19:01:45.598
 System poll interval: 64 s
```

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

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

 Source: [12345]3000::34
 Reference: 127.127.1.0          Clock stratum: 2
 Reachabilities: 15              Poll interval: 64
 Last receive time: 19           Offset: 0.0
 Roundtrip delay: 0.0            Dispersion: 0.0

 Total sessions: 1
```

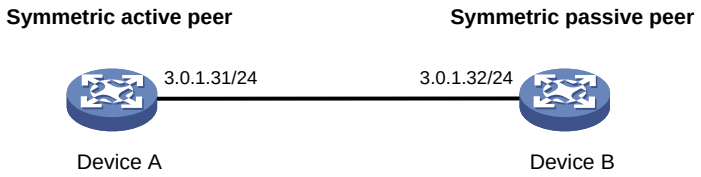
Example: Configuring NTP symmetric active/passive association mode

Network configuration

As shown in Figure 38, perform the following tasks:

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

Figure 38 Network diagram



Procedure

1. Assign an IP address to each interface, and make sure Device A and Device B can reach each other, as shown in Figure 38. (Details not shown.)
2. Configure Device B:

   # Enable the NTP service.

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

   # Specify NTP for obtaining the time.

   ```
   [DeviceB] clock protocol ntp
   ```

3. Configure Device A:

   # Enable the NTP service.

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

   # Specify NTP for obtaining the time.

   ```
   [DeviceA] clock protocol ntp
   ```

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

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

   # Configure Device B as its symmetric passive peer.

   ```
   [DeviceA] ntp-service unicast-peer 3.0.1.32
   ```

Verifying the configuration

Verify that Device B has synchronized its time with 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^-22
 Root delay: 0.00609 ms
 Root dispersion: 1.95859 ms
 Reference time: 83aec681.deb6d3e5  Wed, Jan  8 2014 14:33:11.081
 System poll interval: 64 s
```

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
```

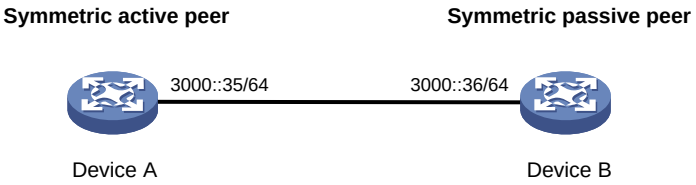
Example: Configuring IPv6 NTP symmetric active/passive association mode

Network configuration

As shown in Figure 39, perform the following tasks:

- Configure Device A's local clock as its reference source, with 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 39 Network diagram



Procedure

1. Assign an IP address to each interface, and make sure Device A and Device B can reach each other, as shown in Figure 39. (Details not shown.)
2. Configure Device B:

   # Enable the NTP service.

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

   # Specify NTP for obtaining the time.

   ```
   [DeviceB] clock protocol ntp
   ```
3. Configure Device A:

   # Enable the NTP service.

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

   # Specify NTP for obtaining the time.

   ```
   [DeviceA] clock protocol ntp
   ```

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

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

   # Configure Device B as the IPv6 symmetric passive peer.

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

Verifying the configuration

Verify that Device B has synchronized its time with 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^-22
Root delay: 0.01855 ms
Root dispersion: 9.23483 ms
Reference time: d0c6047c.97199f9f  Wed, Dec 29 2010 19:03:24.590
System poll interval: 64 s
```

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

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

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

Total sessions: 1
```

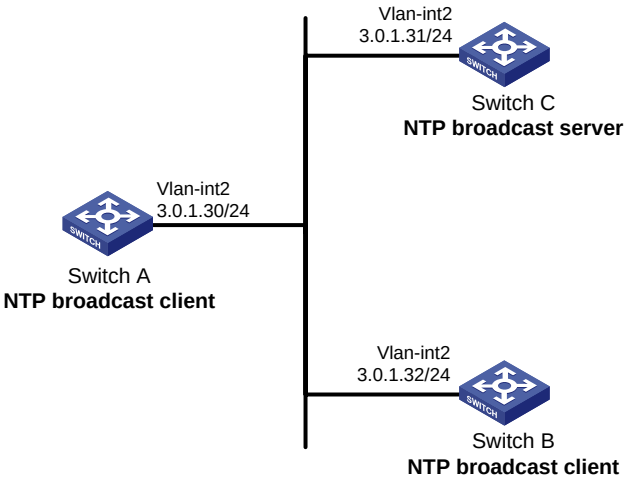
Example: Configuring NTP broadcast association mode

Network configuration

As shown in Figure 40, configure Switch C as the NTP server for multiple devices on the same network segment so that these devices synchronize the time with Switch C.

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

Figure 40 Network diagram



Procedure

1. Assign an IP address to each interface, and make sure Switch A, Switch B, and Switch C can reach each other, as shown in Figure 40. (Details not shown.)
2. Configure Switch C:

Enable the NTP service.

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

Specify NTP for obtaining the time.

```
[SwitchC] clock protocol ntp
```

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

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

Configure Switch C to operate in broadcast server mode and send broadcast messages from 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
```

Specify NTP for obtaining the time.

```
[SwitchA] clock protocol ntp
```

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
```

Specify NTP for obtaining the time.

```
[SwitchB] clock protocol ntp
```

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
```

Verifying the configuration

The following procedure uses Switch A as an example to 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^-22
 Root delay: 0.00229 ms
 Root dispersion: 4.12572 ms
```

```
 Reference time: d0d289fe.ec43c720  Sat, Jan  8 2011  7:00:14.922
 System poll interval: 64 s
```

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
```

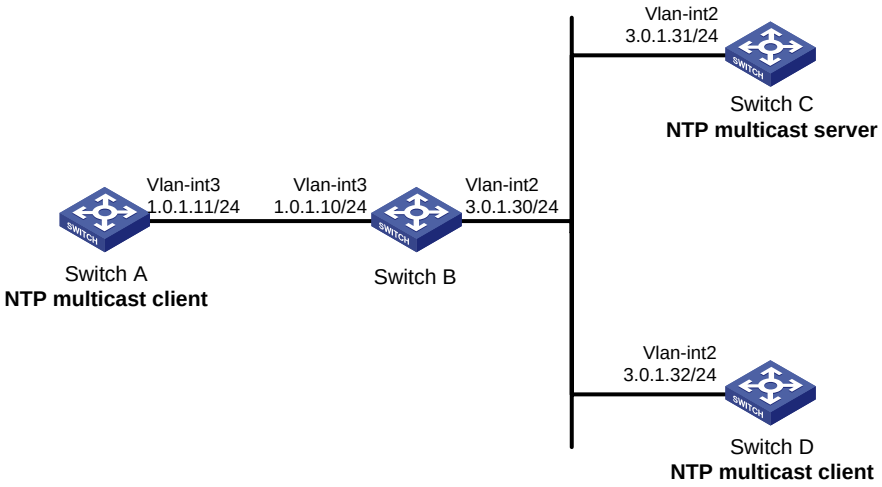
Example: Configuring NTP multicast association mode

Network configuration

As shown in Figure 41, configure Switch C as the NTP server for multiple devices on different network segments so that these devices synchronize the time with Switch C.

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

Figure 41 Network diagram



Procedure

1. Assign an IP address to each interface, and make sure the switches can reach each other, as shown in Figure 41. (Details not shown.)
2. Configure Switch C:

   # Enable the NTP service.

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

   # Specify NTP for obtaining the time.

   ```
   [SwitchC] clock protocol ntp
   ```

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

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

Configure Switch C to operate in multicast server mode and send multicast messages from 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
```

Specify NTP for obtaining the time.

```
[SwitchD] clock protocol ntp
```

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:

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

Switch D and Switch C are on the same subnet, so Switch D can receive the multicast messages from Switch C without being enabled with the multicast functions.

```
[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^-22
 Root delay: 0.00229 ms
 Root dispersion: 4.12572 ms
 Reference time: d0d289fe.ec43c720  Sat, Jan  8 2011  7:00:14.922
 System poll interval: 64 s
```

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 functions.

```
<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 twenty-fivegige 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 twenty-fivegige 1/0/1
[SwitchB-Twenty-FiveGigE1/0/1] 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
```

Specify NTP for obtaining the time.

```
[SwitchA] clock protocol ntp
```

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
```

Verifying the configuration

Verify that Switch A has synchronized its time with Switch C, and the clock stratum level of Switch A is 3.

```
[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^-22
 Root delay: 0.00534 ms
 Root dispersion: 4.51282 ms
 Reference time: d0c61289.10b1193f  Wed, Dec 29 2010 20:03:21.065
 System poll interval: 64 s
```

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
```

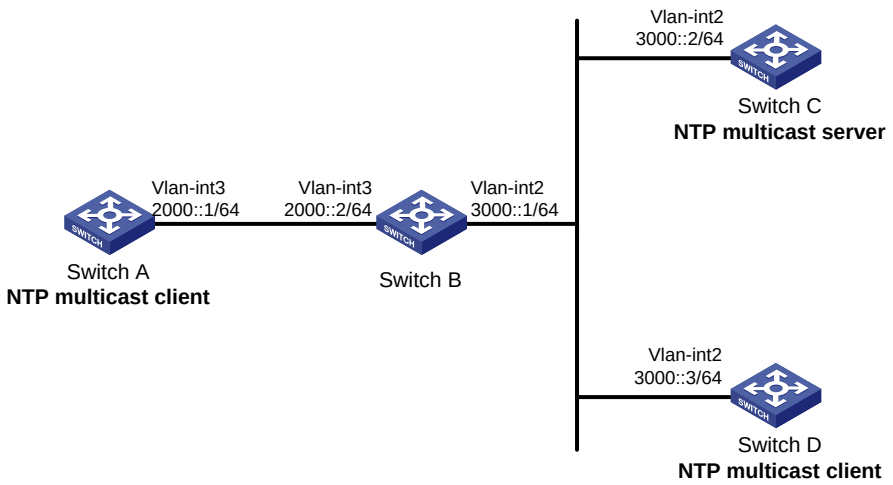
Example: Configuring IPv6 NTP multicast association mode

Network configuration

As shown in Figure 42, configure Switch C as the NTP server for multiple devices on different network segments so that these devices synchronize the time with Switch C.

- Configure Switch C's local clock as its reference source, with stratum level 2.
- Configure Switch C to operate in IPv6 multicast server mode and send 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 on VLAN-interface 3 and VLAN-interface 2, respectively.

Figure 42 Network diagram



Procedure

1. Assign an IP address to each interface, and make sure the switches can reach each other, as shown in Figure 42. (Details not shown.)
2. Configure Switch C:

   # Enable the NTP service.

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

   # Specify NTP for obtaining the time.

   ```
   [SwitchC] clock protocol ntp
   ```

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

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

   # Configure Switch C to operate in IPv6 multicast server mode and send multicast messages from 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
   ```

   # Specify NTP for obtaining the time.

   ```
   [SwitchD] clock protocol ntp
   ```

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:

Verify that Switch D has synchronized its time with Switch C, and the clock stratum level of Switch D is 3.

Switch D and Switch C are on the same subnet, so Switch D can Receive the IPv6 multicast messages from Switch C without being enabled with the IPv6 multicast functions.

```
[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^-22
 Root delay: 0.00000 ms
 Root dispersion: 8.00578 ms
 Reference time: d0c60680.9754fb17  Wed, Dec 29 2010 19:12:00.591
 System poll interval: 64 s
```

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 twenty-fivegige 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 twenty-fivegige 1/0/1
[SwitchB-Twenty-FiveGigE1/0/1] mld-snooping static-group ff24::1 vlan 3
```

6. Configure Switch A:

Enable the NTP service.

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

Specify NTP for obtaining the time.

```
[SwitchA] clock protocol ntp
```

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
```

Verifying 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^-22
 Root delay: 0.00534 ms
 Root dispersion: 4.51282 ms
 Reference time: d0c61289.10b1193f  Wed, Dec 29 2010 20:03:21.065
 System poll interval: 64 s
```

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
```

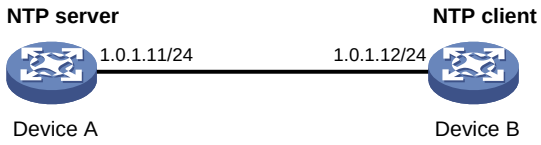
Example: Configuring NTP authentication in client/server association mode

Network configuration

As shown in Figure 43, perform the following tasks:

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

Figure 43 Network diagram



Procedure

1. Assign an IP address to each interface, and make sure Device A and Device B can reach each other, as shown in Figure 43. (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 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 NTP for obtaining the time.

   ```
   [DeviceB] clock protocol ntp
   ```

   # Enable NTP authentication on Device B.

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

   # Create a plaintext authentication key, with key ID **42** and key value **aNiceKey**.

   ```
   [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
   ```

   To enable Device B to synchronize its clock with Device A, enable NTP authentication on Device A.
4. Configure NTP authentication on Device A:

   # Enable NTP authentication.

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

   # Create a plaintext authentication key, with key ID **42** and key value **aNiceKey**.

```
[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
```

Verifying the configuration

Verify that Device B has synchronized its time with Device A, and the clock stratum level of Device B is 3.

```
[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^-22
 Root delay: 0.00655 ms
 Root dispersion: 1.15869 ms
 Reference time: d0c62687.ab1bba7d  Wed, Dec 29 2010 21:28:39.668
 System poll interval: 64 s
```

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
```

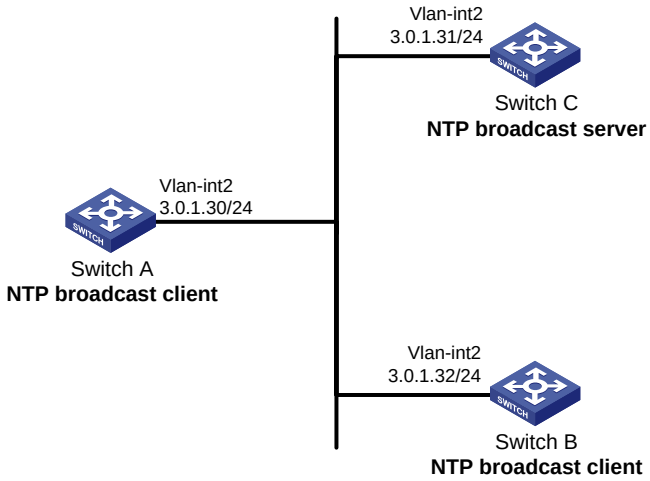
Example: Configuring NTP authentication in broadcast association mode

Network configuration

As shown in Figure 44, configure Switch C as the NTP server for multiple devices on the same network segment so that these devices synchronize the time with Switch C. Configure Switch A and Switch B to authenticate the NTP server.

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

Figure 44 Network diagram



Procedure

1. Assign an IP address to each interface, and make sure Switch A, Switch B, and Switch C can reach each other, as shown in Figure 44. (Details not shown.)
2. Configure Switch A:

   # Enable the NTP service.

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

   # Specify NTP for obtaining the time.

   ```
   [SwitchA] clock protocol ntp
   ```

   # Enable NTP authentication on Switch A. Create a plaintext NTP authentication key, with key ID of **88** and key value of **123456**. 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
   ```

   # Specify NTP for obtaining the time.

   ```
   [SwitchB] clock protocol ntp
   ```

   # Enable NTP authentication on Switch B. Create a plaintext NTP authentication key, with key ID of **88** and key value of **123456**. Specify it as a trusted key.

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

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

   ```
   [SwitchB] interface vlan-interface 2
   ```

```
[SwitchB-Vlan-interface2] ntp-service broadcast-client
```

4. Configure Switch C:

   # Enable the NTP service.

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

   # Specify NTP for obtaining the time.

```
[SwitchC] clock protocol ntp
```

   # Specify the local clock as the reference source, with 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.

```
[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. Create a plaintext NTP authentication key, with key ID of **88** and key value of **123456**. Specify it as a trusted key.

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

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

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

Verifying the configuration

Verify that Switch B has synchronized its time with Switch C, and the clock stratum level of Switch B is 4.

```
[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^-22
 Root delay: 0.00127 ms
 Root dispersion: 2.89877 ms
 Reference time: d0d287a7.3119666f  Sat, Jan  8 2011  6:50:15.191
 System poll interval: 64 s
```

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

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

Example: Configuring MPLS L3VPN network time synchronization in client/server mode

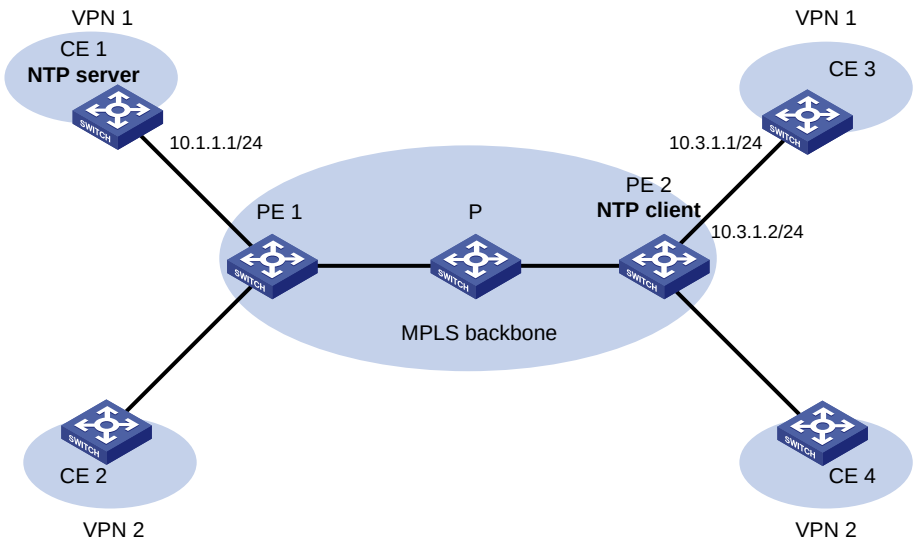
Network configuration

As shown in Figure 45, two MPLS L3VPN instances are present on PE 1 and PE 2: vpn1 and vpn2. CE 1 and CE 3 are devices in VPN 1.

To synchronize time between PE 2 and CE 1 in VPN 1, perform the following tasks:

- Configure CE 1's local clock as its reference source, with stratum level 2.
- Configure CE 1 in the VPN instance vpn1 as the NTP server of PE 2.

Figure 45 Network diagram



Procedure

Before you perform the following configuration, be sure you have completed MPLS L3VPN-related configurations. For information about configuring MPLS L3VPN, see *MPLS Configuration Guide*.

1. Assign an IP address to each interface, as shown in Figure 45. Make sure CE 1 and PE 1, PE 1 and PE 2, and PE 2 and CE 3 can reach each other. (Details not shown.)
2. Configure CE 1:

   # Enable the NTP service.

   ```
   <CE1> system-view
   [CE1] ntp-service enable
   ```

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

   ```
   [CE1] ntp-service refclock-master 2
   ```

3. Configure PE 2:

Enable the NTP service.

```
<PE2> system-view
[PE2] ntp-service enable
```

Specify NTP for obtaining the time.

```
[PE2] clock protocol ntp
```

Specify CE 1 in the VPN instance vpn1 as the NTP server of PE 2.

```
[PE2] ntp-service unicast-server 10.1.1.1 vpn-instance vpn1
```

Verifying the configuration

Verify that PE 2 has synchronized to CE 1, with stratum level 3.

```
[PE2] display ntp-service status
 Clock status: synchronized
 Clock stratum: 3
 System peer: 10.1.1.1
 Local mode: client
 Reference clock ID: 10.1.1.1
 Leap indicator: 00
 Clock jitter: 0.005096 s
 Stability: 0.000 pps
 Clock precision: 2^-22
 Root delay: 0.00655 ms
 Root dispersion: 1.15869 ms
 Reference time: d0c62687.ab1bba7d  Wed, Dec 29 2010 21:28:39.668
 System poll interval: 64 s
```

Verify that an IPv4 NTP association has been established between PE 2 and CE 1.

```
[PE2] display ntp-service sessions
       source          reference       stra reach poll  now offset  delay disper
**************************************************************************
 [1245]10.1.1.1        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
```

Verify that server 127.0.0.1 has synchronized to server 10.1.1.1, and server 10.1.1.1 has synchronized to the local clock.

```
[PE2] display ntp-service trace
Server     127.0.0.1
Stratum    3 , jitter  0.000, synch distance 796.50.
Server     10.1.1.1
Stratum    2 , jitter 939.00, synch distance 0.0000.
RefID      127.127.1.0
```

Example: Configuring MPLS L3VPN network time synchronization in symmetric active/passive mode

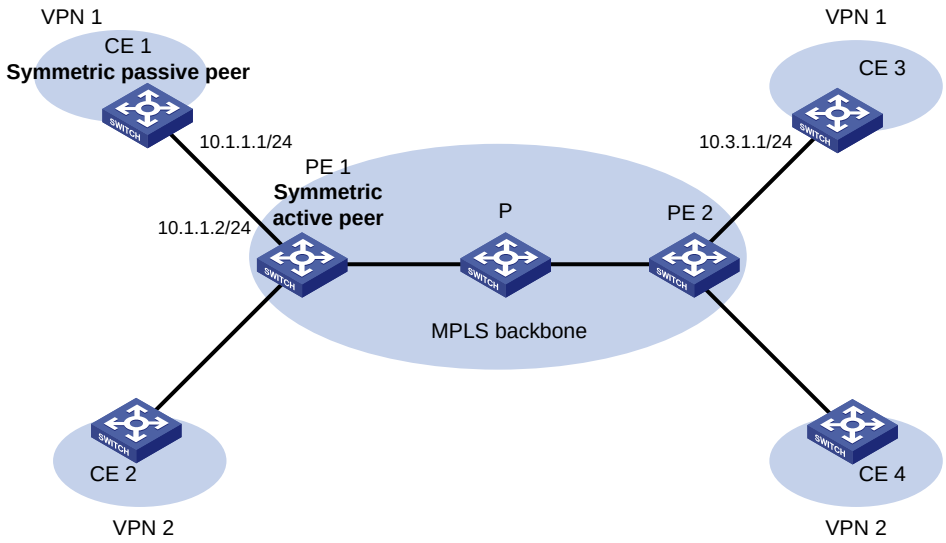
Network configuration

As shown in Figure 46, two VPN instances are present on PE 1 and PE 2: vpn1 and vpn2. CE 1 and CE 3 belong to VPN 1.

To synchronize time between PE 1 and CE 1 in VPN 1, perform the following tasks:

- Configure CE 1's local clock as its reference source, with stratum level 2.
- Configure CE 1 in the VPN instance vpn1 as the symmetric passive peer of PE 1.

Figure 46 Network diagram



Procedure

Before you perform the following configuration, be sure you have completed MPLS L3VPN-related configurations. For information about configuring MPLS L3VPN, see *MPLS Configuration Guide*.

1. Assign an IP address to each interface, as shown in Figure 46. Make sure CE 1 and PE 1, PE 1 and PE 2, and PE 2 and CE 3 can reach each other. (Details not shown.)
2. Configure CE 1:

   # Enable the NTP service.

   ```
   <CE1> system-view
   [CE1] ntp-service enable
   ```

   # Specify NTP for obtaining the time.

   ```
   [CE1] clock protocol ntp
   ```

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

   ```
   [CE1] ntp-service refclock-master 2
   ```

3. Configure PE 1:

   # Enable the NTP service.

   ```
   <PE1> system-view
   [PE1] ntp-service enable
   ```

   # Specify NTP for obtaining the time.

   ```
   [PE1] clock protocol ntp
   ```

   # Specify CE 1 in the VPN instance vpn1 as the symmetric passive peer of PE 1.

   ```
   [PE1] ntp-service unicast-peer 10.1.1.1 vpn-instance vpn1
   ```

Verifying the configuration

Verify that PE 1 has synchronized to CE 1, with stratum level 3.

```
[PE1] display ntp-service status
 Clock status: synchronized
 Clock stratum: 3
```

```
 System peer: 10.1.1.1
 Local mode: sym_active
 Reference clock ID: 10.1.1.1
 Leap indicator: 00
 Clock jitter: 0.005096 s
 Stability: 0.000 pps
 Clock precision: 2^-22
 Root delay: 0.00655 ms
 Root dispersion: 1.15869 ms
 Reference time: d0c62687.ab1bba7d  Wed, Dec 29 2010 21:28:39.668
 System poll interval: 64 s
```

Verify that an IPv4 NTP association has been established between PE 1 and CE 1.

```
[PE1] display ntp-service sessions
       source          reference       stra reach poll  now offset  delay disper
********************************************************************************
 [1245]10.1.1.1        127.127.1.0        2     1   64  519   -0.0 0.0000    0.0
Notes: 1 source(master),2 source(peer),3 selected,4 candidate,5 configured.
 Total sessions: 1
```

Verify that server 127.0.0.1 has synchronized to server 10.1.1.1, and server 10.1.1.1 has synchronized to the local clock.

```
[PE1] display ntp-service trace
Server     127.0.0.1
Stratum    3 , jitter  0.000, synch distance 796.50.
Server     10.1.1.1
Stratum    2 , jitter 939.00, synch distance 0.0000.
RefID      127.127.1.0
```

Configuring SNTP

About SNTP

SNTP is a simplified, client-only version of NTP specified in RFC 4330. It uses the same packet format and packet exchange procedure as NTP, but provides faster synchronization at the price of time accuracy.

SNTP working mode

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.

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.

Protocols and standards

RFC 4330, *Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI*

Restrictions and guidelines: SNTP configuration

When you configure SNTP, follow these restrictions and guidelines:

- You cannot configure both NTP and SNTP on the same device.
- To use NTP for time synchronization, you must use the **`clock protocol`** command to specify NTP for obtaining the time. For more information about the **`clock protocol`** command, see device management commands in *Fundamentals Configuration Guide*.

SNTP tasks at a glance

To configure SNTP, perform the following tasks:

1. Enabling the SNTP service
2. Specifying an NTP server for the device
3. (Optional.) Configuring SNTP authentication
4. (Optional.) Specifying the SNTP time-offset thresholds for log and trap outputs

Enabling the SNTP service

Restrictions and guidelines

The NTP service and SNTP service are mutually exclusive. Before you enable SNTP, make sure NTP is disabled.

Procedure

1. Enter system view.

   `system-view`

2. Enable the SNTP service.

   sntp enable

   By default, the SNTP service is disabled.

Specifying an NTP server for the device

Restrictions and guidelines

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.

Procedure

1. Enter system view.

   system-view

2. Specify an NTP server for the device.

   IPv4:

   sntp unicast-server { *server-name* | *ip-address* } [**vpn-instance** *vpn-instance-name*] [**authentication-keyid** *keyid* | **source** *interface-type interface-number* | **version** *number*] *

   IPv6:

   sntp ipv6 unicast-server { *server-name* | *ipv6-address* } [**vpn-instance** *vpn-instance-name*] [**authentication-keyid** *keyid* | **source** *interface-type interface-number*] *

   By default, no NTP server is specified for the device.

   You can specify multiple NTP servers for the client by repeating this step.

   To perform authentication, you need to specify the **authentication-keyid** *keyid* option.

Configuring SNTP authentication

About this task

SNTP authentication ensures that an SNTP client is synchronized only to an authenticated trustworthy NTP server.

Restrictions and guidelines

Enable authentication on both the NTP server and the SNTP client.

Use the same authentication key ID, algorithm, and key on the NTP server and SNTP client. 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."

On the SNTP client, associate the specified key with the NTP server. Make sure the server is allowed to use the key ID for authentication on the client.

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

Procedure

1. Enter system view.

   system-view

2. Enable SNTP authentication.

   sntp authentication enable

By default, SNTP authentication is disabled.

3. Configure an SNTP authentication key.

   sntp authentication-keyid *keyid* **authentication-mode** { **hmac-sha-1** | **hmac-sha-256** | **hmac-sha-384** | **hmac-sha-512** | **md5** } { **cipher** | **simple** } *string* [**acl** *ipv4-acl-number* | **ipv6 acl** *ipv6-acl-number*]*

   By default, no SNTP authentication key exists.

4. Specify the key as a trusted key.

   sntp reliable authentication-keyid *keyid*

   By default, no trusted key is specified.

5. Associate the SNTP authentication key with an NTP server.

   IPv4:

   sntp unicast-server { *server-name* | *ip-address* } [**vpn-instance** *vpn-instance-name*] **authentication-keyid** *keyid*

   IPv6:

   sntp ipv6 unicast-server { *server-name* | *ipv6-address* } [**vpn-instance** *vpn-instance-name*] **authentication-keyid** *keyid*

   By default, no NTP server is specified.

Specifying the SNTP time-offset thresholds for log and trap outputs

About this task

By default, the system synchronizes the SNTP client's time to the server and outputs a log and a trap when the time offset exceeds 128 ms for multiple times.

After you set the SNTP time-offset thresholds for log and trap outputs, the system synchronizes the client's time to the server when the time offset exceeds 128 ms for multiple times, but outputs logs and traps only when the time offset exceeds the specified thresholds, respectively.

Procedure

1. Enter system view.

   system-view

2. Specify the SNTP time-offset thresholds for log and trap outputs.

   sntp time-offset-threshold { **log** *log-threshold* | **trap** *trap-threshold* } *

   By default, no SNTP time-offset thresholds are set for log and trap outputs.

Display and maintenance commands for SNTP

Execute **display** commands in any view.

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

SNTP configuration examples

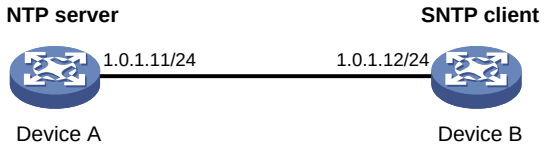
Example: Configuring SNTP

Network configuration

As shown in Figure 47, perform the following tasks:

- Configure Device A's local clock as its reference source, with 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 47 Network diagram



Procedure

1. Assign an IP address to each interface, and make sure Device A and Device B can reach each other, as shown in Figure 47. (Details not shown.)
2. Configure Device A:

   # Enable the NTP service.

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

   # Specify NTP for obtaining the time.

   ```
   [DeviceA] clock protocol ntp
   ```

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

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

   # Enable NTP authentication on Device A.

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

   # Configure a plaintext NTP authentication key, with key ID of **10** and key value of **aNiceKey**.

   ```
   [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
   ```

   # Specify NTP for obtaining the time.

   ```
   [DeviceB] clock protocol ntp
   ```

   # Enable SNTP authentication on Device B.

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

   # Configure a plaintext authentication key, with key ID of **10** and key value of **aNiceKey**.

   ```
   [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
```

Verifying the configuration

Verify that an SNTP association has been established between Device B and Device A, and Device B has synchronized its time with Device A.

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

Configuring PTP

About PTP

Precision Time Protocol (PTP) provides time synchronization among devices with submicrosecond accuracy. It provides also precise frequency synchronization.

Basic concepts

PTP profile

PTP profiles (PTP standards) include:

- **IEEE 1588 version 2**—1588v2 defines high-accuracy clock synchronization mechanisms. It can be customized, enhanced, or tailored as needed. 1588v2 is the latest version.
- **IEEE 802.1AS**—802.1AS is introduced based on IEEE 1588. It specifies a profile for use of IEEE 1588-2008 for time synchronization over a virtual bridged local area network (as defined by IEEE 802.1Q). 802.1AS supports point-to-point full-duplex Ethernet, IEEE 802.11, and IEEE 802.3 EPON links.
- **SMPTE ST 2059-2**—ST2059-2 is introduced based on IEEE 1588. It specifies a profile specifically for time synchronization of audio or video equipment in a professional broadcast environment. It includes a self-contained description of parameters, their default values, and permitted ranges.
- **AES67-2015**—AES67-2015 is introduced based on IEEE 1588. It specifies a profile specifically for time synchronization of professional equipment for broadcast, music production, and film and television post-production. It includes a self-contained description of parameters, their default values, and permitted ranges.

PTP domain

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

PTP instance

When the device belongs to multiple PTP domains, you need to configure and associate a PTP instance to each PTP domain. In PTP instance view, you can configure PTP settings such as PTP profile and clock node type. The settings take effect only on the PTP domain associated with the PTP instance. PTP instances are isolated from each other, allowing different PTP timing systems running on a network without affecting each other.

Optimal PTP domain

Each PTP instance has a reference clock and clock information. For a device that has multiple PTP instances running simultaneously, you need to select an optimal PTP instance and use the clock source traced by this PTP instance to synchronize the system time of the device. The domain to which the optimal PTP associates is the optimal domain.

Clock node and PTP port

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

- **Ordinary Clock (OC)**—A PTP clock with a single PTP port in a PTP domain for time synchronization. It synchronizes time from its upstream clock node through the port. If an OC operates as the clock source, it sends synchronization time through a single PTP port to its downstream clock nodes.

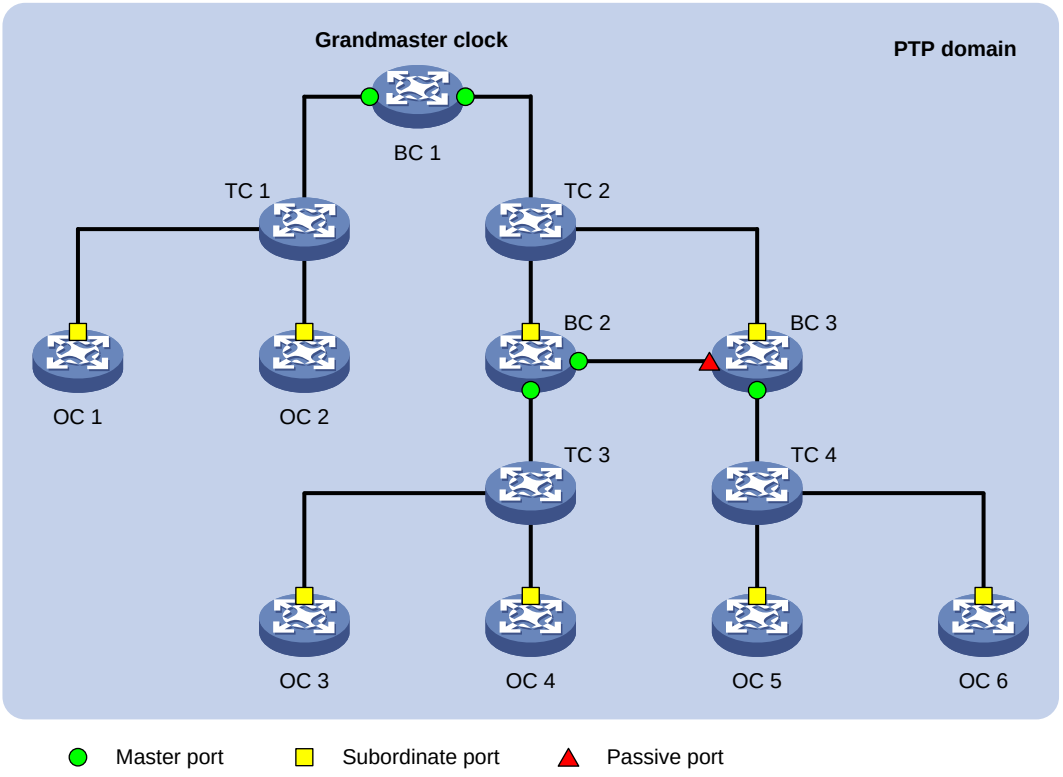
- **Boundary Clock (BC)**—A clock with more than one PTP port in a PTP domain for time synchronization. A BC uses one of the ports to synchronize time from its upstream clock node. It uses the other ports to synchronize time to the relevant upstream clock nodes. If a BC operates as the clock source, such as BC 1 in Figure 48, it synchronizes time through multiple PTP ports to its downstream clock nodes.
- **Transparent Clock (TC)**—A TC does not keep time consistency with other clock nodes. A TC has multiple PTP ports. It forwards PTP messages among these ports and performs delay corrections for the messages, instead of performing time synchronization. TCs include the following types:
  - **End-to-End Transparent Clock (E2ETC)**—Forwards all P2P PTP packets in the network and calculates the delay of the entire link.
  - **Peer-to-Peer Transparent Clock (P2PTC)**—Forwards only Sync, Follow_Up, and Announce messages, terminates other PTP messages, and calculates the delay of each link segment.

IMPORTANT:

OC and BC clock nodes do much more work than TCs. When multiple PTP domains are configured on the device, the synchronization performance might fluctuate or decrease, and synchronization failure might occur due to limitation of hardware resources. As a best practice, configure the device operating in multiple domains as an OC or BC clock node in a maximum of one PTP instance, to decrease calculations, minimize mutual influences between the domains, and optimize the synchronization performance.

Figure 48 shows the positions of these types of clock nodes in a PTP domain.

Figure 48 Clock nodes in a PTP domain



In addition to these basic types of clock nodes, PTP introduces hybrid clock nodes. For example, a TC+OC has multiple PTP ports in a PTP domain. One port is the OC type, and the others are the TC type.

A TC+OC forwards PTP messages through TC-type ports and performs delay corrections. In addition, it synchronizes time through its OC-type port. TC+OCs include these types: E2ETC+OC and P2PTC+OC.

Master-member/subordinate relationship

The master-member/subordinate relationship is automatically determined based on the Best Master Clock (BMC) algorithm. You can also manually specify a role for the clock nodes.

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

- **Master/Member node**—A master node sends a synchronization message, and a member node receives the synchronization message.
- **Master/Member clock**—The clock on a master node is a master clock (parent clock) The clock on a member node is a member clock.
- **Master/Subordinate port**—A master port sends a synchronization message, and a subordinate port receives the synchronization message. The master and subordinate ports can be on a BC or an OC.

  A port that neither receives nor sends synchronization messages is a passive port.

Clock source type

A clock node supports the following clock sources:

- **Local clock source**—38.88 MHz clock signals generated by a crystal oscillator inside the clock monitoring module.
- **External clock source**—Clock signals generated by an external clock device. The signals are received and sent by a 1PPS/ToD port on the MPU. It is also called a ToD clock source.

Grandmaster clock

As shown in Figure 48, the grandmaster clock (GM) is the ultimate source of time for clock synchronization in a PTP domain. It is elected automatically by the clock nodes in the PTP domain. The clock nodes exchange PTP messages and elect the GM by comparing the clock priority, time class, and time accuracy carried in the PTP messages.

You can also specify a GM manually.

Clock source

The clock source used by clock nodes is 38.88 MHz clock signals generated by a crystal oscillator inside the clock monitoring module of the device.

Grandmaster clock selection and master-member/subordinate relationship establishment

A GM can be manually specified. It can also be elected through the BMC algorithm as follows:

1. The clock nodes in a PTP domain exchange announce messages and elect a GM by using the following rules in descending order:
   a. Clock node with higher priority 1.
   b. Clock node with higher time class.
   c. Clock node with higher time accuracy.
   d. Clock node with higher priority 2.
   e. Clock node with a smaller port ID (containing clock number and port number).

   The master nodes, member nodes, master ports, and subordinate ports are determined during the process. Then a spanning tree with the GM as the root is generated for the PTP domain.

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

Optimal domain selection

If only one PTP instance is configured on the device, this PTP instance is the optimal instance of the device. If multiple PTP instances are configured on the device, the optimal PTP instance (domain) is selected by using the following rules in descending order:

1. Instance activated.
2. Instance on which the device is an OC or BC clock node.
3. Instance on which the clock source has a higher priority 1 value
4. Instance on which the clock source has a higher time class.
5. Instance on which the clock source has a higher time accuracy.
6. Instance on which the clock source has a lower offset from the GM.
7. Instance on which the clock source has a higher priority 2 value.
8. Instance associated with a PTP domain that has a smaller domain ID.

The PTP instance on which the **`ptp active force-state`** command is configured has the lowest priority.

Synchronization mechanism

After master-member relationships are established between the clock nodes, the master and member clock nodes exchange PTP messages and record the message transmit and receive time. Based on the timestamps, each member clock calculates the path delay and time offset between them and the master clock and adjusts their time accordingly for time synchronization with the master clock.

PTP defines two path delay measurement mechanisms: Request_Response_ and Peer Delay, both of which are based on network symmetry.

Request_Response

The Request_Response mechanism measures the average path delay between the master and member clock nodes by using the PTP messages as shown in Figure 49. A TC between master and member clock nodes does not calculate the path delay. It forwards PTP messages and carries the Sync message residence time on it to the downstream clock node.

This mechanism can be implemented in one of the following two modes:

- **Two-step mode**—t1 is carried in the Follow_Up message as shown in Figure 49.
- **Single-step mode**—t1 is carried in the Sync message, and no Follow_Up message is sent.

  This mode is not supported in the current software version.

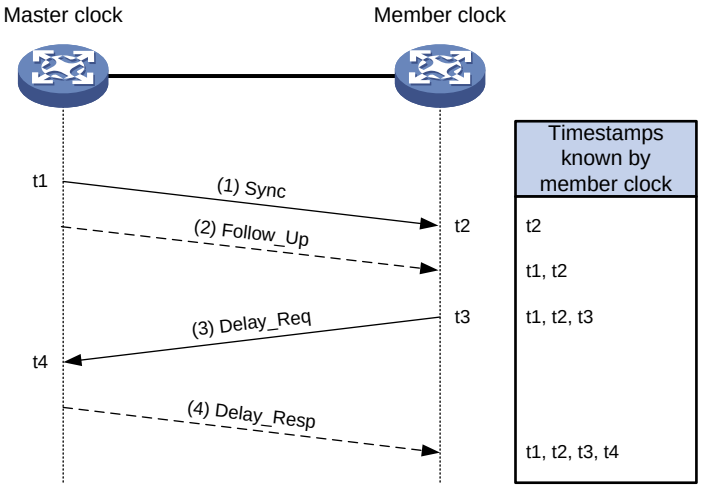
Figure 49 shows the Request_Response mechanism in two-step mode.

1. The master clock sends a Sync message to the member clock, and records the sending time t1. Upon receiving the message, the member clock records the receiving time t2.
2. After sending the Sync message, the master clock immediately sends a Follow_Up message that carries time t1.
3. The member clock sends a Delay_Req message to the master clock, and records the sending time t3. Upon receiving the message, the master clock records the receiving time t4.
4. The master clock returns a Delay_Resp message that carries time t4.

After this procedure, the member clock obtains all the four timestamps and can make the following calculations:

- Round-trip delay between the master and member clocks: (t2 – t1) + (t4 – t3)
- One-way delay between the master and member clocks: [(t2 – t1) + (t4 – t3)] / 2
- Offset between the member and master clocks: (t2 – t1) – [(t2 – t1) + (t4 – t3)] / 2 or [(t2 – t1) – (t4 – t3)] / 2

Figure 49 Request_Response mechanism (two-step node)



Peer Delay

The Peer Delay mechanism measures the average path delay between two clock nodes. The two clock nodes (BC, TC, or OC) implementing this mechanism send Pdelay messages to each other, and calculate the one-way link delay between them independently. The message interaction process and delay calculation method are identical on the two nodes. TCs that exist between master and member clock nodes divide the synchronization path into multiple links and participate in delay calculation. The link delays and Sync message residence time on the TCs are carried to downstream nodes.

This mechanism can be implemented in one of the following two modes:

- Two-step mode

  As shown in Figure 50, Pdelay messages include Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up messages. t2 is carried in the Pdelay_Resp message, and t3 is carried in the Pdelay_Resp_Follow_Up message.

- Single-step mode:
  - t1 is carried in the Sync message, and no Follow_Up message is sent.
  - The offset between t5 and t4 is carried in the Pdelay_Resp message, and no Pdelay_Resp_Follow_Up message is sent.

  This mode is not supported in the current software version.

Figure 50 uses Clock node B as an example to describe the Peer Delay mechanism.

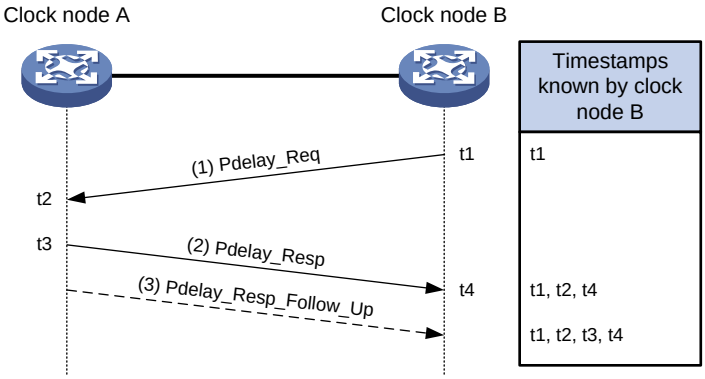
1. Clock node B sends a Pdelay_Req message to Clock node A, and records the sending time t1. Upon receiving the message, Clock node A records the receiving time t2.
2. Clock node A sends a Pdelay_Req message that carries t2 to Clock node B, and records the sending time t3. Upon receiving the message, Clock node B records the receiving time t4.
3. Clock node A immediately sends a Pdelay_Resp_Follow_Up message carrying t3 to Clock node B after sending the Pdelay_Req message.

After this procedure, Clock node B obtains all the four timestamps and can make the following calculations:

- Round-trip delay between Clock node A and Clock node B: (t2 – t1) + (t4 – t3)

- One-way delay between Clock node A and Clock node B: [(t2 – t1) + (t4 – t3)] / 2 = [(t3 – t2) + (t4 – t1)] / 2
- Time offset between the member clock and the master clock: Sync message receiving time on the member clock – Sync message sending time on the master clock – Total one-way delays on all links – Total Sync message residence time on all TCs.

Figure 50 Peer Delay mechanism (two-step mode)



Protocols and standards

- IEEE Std 1588-2008, *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, 2008*
- IEEE 802.1AS, *Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks*
- SMPTE ST 2059-2, *SMPTE Profile for Use of IEEE-1588 Precision Time Protocol in Professional Broadcast Applications*
- AES67-2015, *AES Standard for Audio Applications of Networks-High-Performance Streaming Audio-Over-IP Interoperability, 2015*

Restrictions: Hardware compatibility with PTP

The device does not provide ToD interfaces or support external clock sources.

Restrictions and guidelines: PTP configuration

For an interface to run PTP, you must enable PTP globally and on the interface.

Before configuring PTP, determine the PTP profile and define the scope of the PTP domain and the role of every clock node.

PTP tasks at a glance

Configuring PTP (IEEE 1588 version 2)

1. Specifying PTP for obtaining the time
2. Specifying a PTP profile

   Specify the IEEE 1588 version 2 PTP profile.

3. Configuring clock nodes
   - Specifying a clock node type
   - (Optional.) Configuring an OC to operate only as a member clock
4. (Optional.) Specifying a PTP domain
5. Enabling PTP

   To run PTP on an interface, enable PTP globally and on the interface.
   - Enabling PTP globally
   - Enabling PTP on a port
6. Configuring PTP ports
   - (Optional.) Configuring the role of a PTP port
   - Configuring the mode for carrying timestamps
   - Specifying a delay measurement mechanism for a BC or an OC
   - Configuring one of the ports on a TC+OC clock as an OC-type port
7. (Optional.) Configuring PTP message transmission and receipt
   - Setting the interval for sending announce messages and the timeout multiplier for receiving announce messages
   - Setting the interval for sending Pdelay_Req messages
   - Setting the interval for sending Sync messages
   - Setting the minimum interval for sending Delay_Req messages
8. (Optional.) Configuring parameters for PTP messages
   - Specifying the IPv4 UDP transport protocol for PTP messages
   - Configuring a source IP address for multicast PTP message transmission over
   - Configuring a destination IP address for unicast PTP message transmission over Configuring a destination IP address for
   - Configuring the destination MAC address for non-Pdelay messages
   - Setting a DSCP value for PTP messages transmitted over
   - Specifying a VLAN tag for PTP messages
9. (Optional.) Adjusting and correcting clock synchronization
   - Setting the delay correction value
   - Setting the cumulative offset between the UTC and TAI
   - Setting the correction date of the UTC
10. (Optional.) Configuring a priority for a clock
11. (Optional.) Configuring the PTP time locking and unlocking thresholds

Configuring PTP (IEEE 802.1AS)

1. Specifying PTP for obtaining the time
2. Specifying a PTP profile

   Specify the IEEE 802.1AS PTP profile.
3. Configuring clock nodes
   - Specifying a clock node type
   - (Optional.) Configuring an OC to operate only as a member clock
4. (Optional.) Specifying a PTP domain
5. Enabling PTP

   To run PTP on an interface, enable PTP globally and on the interface.

- Enabling PTP globally
- Enabling PTP on a port

6. Configuring PTP ports
   - (Optional.) Configuring the role of a PTP port
   - Configuring one of the ports on a TC+OC clock as an OC-type port
7. (Optional.) Configuring PTP message transmission and receipt
   - Setting the interval for sending announce messages and the timeout multiplier for receiving announce messages
   - Setting the interval for sending Pdelay_Req messages
   - Setting the interval for sending Sync messages
8. (Optional.) Specifying a VLAN tag for PTP messages
9. (Optional.) Adjusting and correcting clock synchronization
   - Setting the delay correction value
   - Setting the cumulative offset between the UTC and TAI
   - Setting the correction date of the UTC
10. (Optional.) Configuring a priority for a clock
11. (Optional.) Configuring the PTP time locking and unlocking thresholds

Configuring PTP (SMPTE ST 2059-2)

1. Specifying PTP for obtaining the time
2. Specifying a PTP profile

   Specify the SMPTE ST 2059-2 PTP profile.
3. Configuring clock nodes
   - Specifying a clock node type
   - (Optional.) Configuring an OC to operate only as a member clock
4. (Optional.) Specifying a PTP domain
5. Enabling PTP

   To run PTP on an interface, enable PTP globally and on the interface.
   - Enabling PTP globally
   - Enabling PTP on a port
6. Configuring PTP ports
   - (Optional.) Configuring the role of a PTP port
   - Configuring the mode for carrying timestamps
   - Specifying a delay measurement mechanism for a BC or an OC
7. (Optional.) Configuring PTP message transmission and receipt
   - Setting the interval for sending announce messages and the timeout multiplier for receiving announce messages
   - Setting the interval for sending Pdelay_Req messages
   - Setting the interval for sending Sync messages
   - Setting the minimum interval for sending Delay_Req messages
8. (Optional.) Configuring parameters for PTP messages
   - Configuring a source IP address for multicast PTP message transmission over
   - Configuring a destination IP address for unicast PTP message transmission over
   - Setting a DSCP value for PTP messages transmitted over

- Specifying a VLAN tag for PTP messages

9. (Optional.) Adjusting and correcting clock synchronization
   - Setting the delay correction value
   - Setting the cumulative offset between the UTC and TAI
   - Setting the correction date of the UTC
10. (Optional.) Configuring a priority for a clock
11. (Optional.) Configuring the PTP time locking and unlocking thresholds

Configuring PTP (AES67-2015)

1. Specifying PTP for obtaining the time
2. (Optional.) Creating a PTP instance
3. Specifying a PTP profile

   Specify the AES67-2015 PTP profile.
4. Configuring clock nodes
   - Specifying a clock node type
   - (Optional.) Configuring an OC to operate only as a member clock
5. (Optional.) Specifying a PTP domain
6. Enabling PTP

   To run PTP on an interface, enable PTP globally and on the interface.
   - Enabling PTP globally
   - Enabling PTP on a port
7. Configuring PTP ports
   - (Optional.) Configuring the role of a PTP port
   - Configuring the mode for carrying timestamps
   - Specifying a delay measurement mechanism for a BC or an OC
   - Configuring one of the ports on a TC+OC clock as an OC-type port
8. (Optional.) Configuring PTP message transmission and receipt
   - Setting the interval for sending announce messages and the timeout multiplier for receiving announce messages
   - Setting the interval for sending Pdelay_Req messages
   - Setting the interval for sending Sync messages
   - Setting the minimum interval for sending Delay_Req messages
9. (Optional.) Configuring parameters for PTP messages
   - Configuring a source IP address for multicast PTP message transmission over
   - Configuring a destination IP address for unicast PTP message transmission over
   - Setting a DSCP value for PTP messages transmitted over
   - Specifying a VLAN tag for PTP messages
10. (Optional.) Adjusting and correcting clock synchronization
    - Setting the delay correction value
    - Setting the cumulative offset between the UTC and TAI
    - Setting the correction date of the UTC
11. (Optional.) Configuring a priority for a clock
12. (Optional.) Configuring the PTP time locking and unlocking thresholds

Specifying PTP for obtaining the time

1. Enter system view.
   system-view
2. Specify PTP for obtaining the time.
   clock protocol ptp

   By default, the device uses NTP to obtain the system time.

   For more information about the **clock protocol** command, see device management commands in *Fundamentals Command Reference*.

Creating a PTP instance

About this task

A PTP instance is uniquely identified by its ID on a device. For easy identification and management, you can also set a name for a PTP instance.

Restrictions and guidelines

Do not set a same name for different PTP instances.

If you create PTP instances with the same ID but different names, the most recent configuration takes effect.

PTP instance 0 is the default PTP instance. You cannot create or delete PTP instance 0. PTP settings configured in system view take effect only on PTP instance 0. To configure settings for a PTP instance other than PTP instance 0, enter PTP instance view.

Procedure

1. Enter system view.
   system-view
2. Create a PTP instance.
   ptp instance *ptp-instance-id* [**name** *ptp-instance-name*]

   By default, PTP instance numbered **0** and named **default-instance** exists.

Specifying a PTP profile

Restrictions and guidelines

You must specify a PTP profile before configuring PTP settings. Changing the PTP profile clears all settings under the profile.

Procedure

1. Enter system view.
   system-view
2. (Optional.) Enter PTP instance view.
   ptp instance *ptp-instance-id*

   To configure settings for PTP instance 0, skip this step.
3. Specify a PTP profile.
   ptp profile { **1588v2** | **8021as** | **aes67-2015** | **st2059-2** }

   By default, no PTP profile is configured, and PTP is not running on the device.

Configuring clock nodes

Specifying a clock node type

Restrictions and guidelines

You can specify only one clock node type for the device. The clock node types include OC, BC, E2ETC, P2PTC, E2ETC+OC, and P2PTC+OC.

Before you specify a clock node type, specify a PTP profile.

For the IEEE 802.1AS PTP profile, you cannot specify the E2ETC or E2ETC+OC clock node type.

For the SMPTE ST 2059-2 or AES67-2015 PTP profile, you cannot specify the E2ETC+OC or P2PTC+OC clock node type.

Changing or removing the clock node type restores the default settings of the PTP profile.

Procedure

1. Enter system view.
   system-view
2. (Optional.) Enter PTP instance view.
   ptp instance *ptp-instance-id*
   To configure settings for PTP instance 0, skip this step.
3. Specify a clock node type for the device.
   - IEEE 1588v2 profile:
     ptp mode { **bc** | **e2etc** | **e2etc-oc** | **oc** | **p2ptc** | **p2ptc-oc** }
   - IEEE 802.1AS profile:
     ptp mode { **bc** | **oc** | **p2ptc** | **p2ptc-oc** }
   - AES67-2015 or SMPTE ST 2059-2 profile:
     ptp mode { **bc** | **e2etc** | **oc** | **p2ptc** }

   By default, no clock node type is specified.

Configuring an OC to operate only as a member clock

About this task

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

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

Restrictions and guidelines

This task is applicable only to OCs.

Procedure

1. Enter system view.
   system-view
2. (Optional.) Enter PTP instance view.
   ptp instance *ptp-instance-id*
   To configure settings for PTP instance 0, skip this step.

3. Configure the OC to operate only as a member clock.

   ptp slave-only

   By default, an OC operates as a master or member clock.

Specifying a PTP domain

About this task

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

Procedure

1. Enter system view.

   system-view

2. (Optional.) Enter PTP instance view.

   ptp instance *ptp-instance-id*

   To configure settings for PTP instance 0, skip this step.

3. Specify a PTP domain for the device.

   ptp domain *value*

   By default, no PTP domain exists.

   Do not configure a same domain on different PTP instances.

Enabling PTP globally

Restrictions and guidelines

For PTP to run on an interface, you must enable PTP globally and on the interface.

Procedure

1. Enter system view.

   system-view

2. Enable PTP globally.

   ptp global enable

   By default, PTP is enabled globally.

Enabling PTP on a port

About this task

A port enabled with PTP becomes a PTP port.

Restrictions and guidelines

You can enable PTP on only one port on an OC.

To enable PTP on a Layer 3 Ethernet interface that has been assigned to a VPN instance, you must specify this VPN instance in the **ptp source** *ip-address* **vpn-instance** *vpn-instance-name* command if PTP messages are to be transmitted in multicast mode over IPv4 UDP.

Procedure

1. Enter system view.

system-view

2. Enter Layer 2 Ethernet interface view or Layer 3 Ethernet interface view.
   interface *interface-type interface-number*
3. (Optional.) Assign the interface to a PTP instance and enter interface PTP instance view.
   ptp instance *ptp-instance-id*

   To configure settings for PTP instance 0, skip this step.
4. Enable PTP on the port.
   ptp enable

   By default, PTP is disabled on a port.

Configuring PTP ports

Configuring the role of a PTP port

About this task

You can configure the master, passive, or slave role for a PTP port.

For an OC that operates in **slave-only** mode, you can perform this task to change its PTP port role to master or slave.

Restrictions and guidelines

By default, the PTP port roles are automatically negotiated based on the BMC algorithm. If you use this task to change the role of one PTP port, all the other PTP ports in the PTP domain stop working. For these PTP ports to function, you must specify roles for each of them. As a best practice, enable automatic negotiation of PTP port roles based on the BMC algorithm.

Only one subordinate port is allowed to be configured for a device.

Procedure

1. Enter system view.
   system-view
2. Enter Layer 2 Ethernet interface view or Layer 3 Ethernet interface view.
   interface *interface-type interface-number*
3. (Optional.) Assign the interface to a PTP instance and enter interface PTP instance view.
   ptp instance *ptp-instance-id*

   To configure settings for PTP instance 0, skip this step.
4. Configure the role of the PTP port.
   ptp force-state { **master** | **passive** | **slave** }

   By default, the PTP port role is automatically calculated through BMC.
5. Return to system view.
   quit
6. (Optional.) Enter interface PTP instance view.
   ptp instance *ptp-instance-id*

   To configure settings for PTP instance 0, skip this step.
7. Activate the port role configuration.
   ptp active force-state

   By default, the port role configuration is not activated.

Configuring the mode for carrying timestamps

About this task

Timestamps can be carried in either of the following modes:

- **Single-step mode**—The Sync message (in the Request_Response or Peer Delay mechanism) and Pdelay_Resp message (in the Peer Delay mechanism) carry their sending timestamps by themselves.
- **Two-step mode**—The Sync message (in the Request_Response or Peer Delay mechanism) and Pdelay_Resp message (in the Peer Delay mechanism) do not carry their sending timestamps by themselves. The subsequent messages carry their sending timestamps.

Procedure

1. Enter system view.

   system-view
2. Enter Layer 2 Ethernet interface view or Layer 3 Ethernet interface view.

   interface *interface-type interface-number*
3. (Optional.) Assign the interface to a PTP instance and enter interface PTP instance view.

   ptp instance *ptp-instance-id*

   To configure settings for PTP instance 0, skip this step.
4. Configure the mode for carrying timestamps.

   ptp clock-step two-step

   By default, the two-step mode is used for carrying timestamps.

Specifying a delay measurement mechanism for a BC or an OC

About this task

PTP defines two transmission delay measurement mechanisms: Request_Response and Peer Delay. For correct communication, ports on the same link must share the same delay measurement mechanism.

Restrictions and guidelines

When the PTP profile is IEEE 1588 version 2, SMPTE ST 2059-2, or AES67-2015, the following restrictions apply:

- You can configure this task only for a BC or OC clock node.
- You cannot configure this task for an E2ETC, E2ETC+OC, P2PTC, or P2PTC+OC clock node. The E2ETC and E2ETC+OC clock nodes support both Request_Response and Peer Delay measurement mechanisms. A P2PTC clock node supports only the Peer Delay measurement mechanism.

The IEEE 802.1AS PTP profile supports only the peer delay measurement mechanism and does not support this task.

Procedure

1. Enter system view.

   system-view
2. Enter Layer 2 Ethernet interface view or Layer 3 Ethernet interface view.

   interface *interface-type interface-number*
3. (Optional.) Assign the interface to a PTP instance and enter interface PTP instance view.

ptp instance *ptp-instance-id*

To configure settings for PTP instance 0, skip this step.

4. Specify a delay measurement mechanism for a BC or an OC.

   ptp delay-mechanism { **e2e** | **p2p** }

   The default delay measurement mechanism varies by PTP profile.

   - **IEEE 1588 version 2, AES67-2015, and SMPTE ST 2059-2**—Request_Response delay measurement mechanism.
   - **IEEE 802.1AS**—Peer Delay measurement mechanism.

Configuring one of the ports on a TC+OC clock as an OC-type port

About this task

All ports on a TC+OC (E2ETC+OC or P2PTC+OC) are TC-type ports by default. This feature allows you to configure one of the ports on a TC+OC clock as an OC-type port.

Restrictions and guidelines

This task is applicable only to E2ETC+OCs and P2PTC+OCs.

This task is not available for the SMPTE ST 2059-2 or AES67-2015 PTP profile.

For time synchronization accuracy, the OC-type port on an E2ETC+OC or P2PTC+OC must be specified as the master port.

Procedure

1. Enter system view.

   system-view

2. Enter Layer 2 Ethernet interface view or Layer 3 Ethernet interface view.

   interface *interface-type interface-number*

3. (Optional.) Assign the interface to a PTP instance and enter interface PTP instance view.

   ptp instance *ptp-instance-id*

   To configure settings for PTP instance 0, skip this step.

4. Configure the port type as OC.

   ptp port-mode oc

   By default, the ports on the E2ETC+OC and P2PTC+OC clock nodes are all TC type.

Configuring PTP message transmission and receipt

Setting the interval for sending announce messages and the timeout multiplier for receiving announce messages

About this task

A master node sends announce messages to the member nodes at the specified interval. If a member node does not receive any announce messages from the master node within the specified interval, it determines that the master node is invalid.

For the IEEE 1588 version 2, AES67-2015, or SMPTE ST 2059-2 PTP profile, the timeout for receiving announce messages is the announce message sending interval for the subordinate node × *multiple-value*. For IEEE 802.1AS, the timeout for receiving announce messages is the announce message sending interval for the master node × *multiple-value*.

Procedure

1. Enter system view.
   system-view
2. Enter Layer 2 Ethernet interface view or Layer 3 Ethernet interface view.
   interface *interface-type interface-number*
3. (Optional.) Assign the interface to a PTP instance and enter interface PTP instance view.
   ptp instance *ptp-instance-id*
   To configure settings for PTP instance 0, skip this step.
4. Set the interval for sending announce messages.
   ptp announce-interval *interval*
   The default settings vary by PTP profile.
   - **IEEE 1588 version 2 or AES67-2015**—The *interval* argument value is 1 and the interval for sending announce messages is 2 (2^1) seconds.
   - **IEEE 802.1AS**—The *interval* argument value is 0 and the interval for sending announce messages is 1 (2^0)second.
   - **SMPTE ST 2059-2**—The *interval* argument value is –2 and the interval for sending announce messages is 1/4 (2^{-2}) seconds.
5. Set the number of intervals before a timeout occurs.
   ptp announce-timeout *multiple-value*
   By default, a timeout occurs when three intervals are reached.

Setting the interval for sending Pdelay_Req messages

1. Enter system view.
   system-view
2. Enter Layer 2 Ethernet interface view or Layer 3 Ethernet interface view.
   interface *interface-type interface-number*
3. (Optional.) Assign the interface to a PTP instance and enter interface PTP instance view.
   ptp instance *ptp-instance-id*
   To configure settings for PTP instance 0, skip this step.
4. Set the interval for sending Pdelay_Req messages.
   ptp pdelay-req-interval *interval*
   By default, the *interval* argument value is 0 and the interval for sending peer delay request messages is 1 (2^0) second.
   For the SMPTE ST 2059-2 or AES67-2015 PTP profile, set the *interval* argument to a value in the range of **ptp syn-interval** *interval* to **ptp syn-interval** *interval* plus 5 as a best practice.

Setting the interval for sending Sync messages

1. Enter system view.
   system-view

2. Enter Layer 2 Ethernet interface view or Layer 3 Ethernet interface view.
   interface *interface-type interface-number*
3. (Optional.) Assign the interface to a PTP instance and enter interface PTP instance view.
   ptp instance *ptp-instance-id*
   To configure settings for PTP instance 0, skip this step.
4. Set the interval for sending Sync messages.
   ptp syn-interval *interval*
   The default settings vary by PTP profile.
   - **IEEE 1588 version 2**—The *interval* argument value is 0 and the interval for sending Sync messages is 1 (2^0) second.
   - **IEEE 802.1AS, AES67-2015, or SMPTE ST 2059-2**—The *interval* argument value is –3 and the interval for sending Sync messages is 1/8 (2^{-3}) seconds.

Setting the minimum interval for sending Delay_Req messages

About this task

When receiving a Sync or Follow_Up message, an interface can send Delay_Req messages only when the minimum interval is reached. This task allows you to set the minimum interval for sending Delay_Req messages.

Restrictions and guidelines

This setting is not available for the IEEE 802.1AS PTP profile.

In PTP multicast transport mode, this setting takes effect only when configured on the master clock. The master clock sends the value to a member clock through PTP messages to control the interval for the member clock to send Delay_Req messages. To view the interval, execute the **display ptp interface** command on the member clock.

In PTP unicast transport mode, this setting takes effect when configured on member clocks. It does not take effect when configured on the master clock.

Procedure

1. Enter system view.
   system-view
2. Enter Layer 2 Ethernet interface view or Layer 3 Ethernet interface view.
   interface *interface-type interface-number*
3. (Optional.) Assign the interface to a PTP instance and enter interface PTP instance view.
   ptp instance *ptp-instance-id*
   To configure settings for PTP instance 0, skip this step.
4. Set the minimum interval for sending Delay_Req messages.
   ptp min-delayreq-interval *interval*
   The *interval* argument value is 0 and the minimum interval for sending delay request messages is 1 (2^0) second.
   For the SMPTE ST 2059-2 PTP profile, set the *interval* argument to a value in the range of **ptp syn-interval** *interval* to **ptp syn-interval** *interval* plus 5 as a best practice.

Configuring parameters for PTP messages

Specifying the IPv4 UDP transport protocol for PTP messages

About this task

PTP messages can be transported over IEEE 802.3/Ethernet or IPv4 UDP.

Restrictions and guidelines

The IEEE 802.1AS PTP profile transports PTP messages over IEEE 802.3/Ethernet and does not support this task.

The SMPTE ST 2059-2 and AES67-2015 PTP profiles transport PTP messages over IPv4 UDP and do not support this task.

Procedure

1. Enter system view.
   system-view
2. Enter Layer 2 Ethernet interface view or Layer 3 Ethernet interface view.
   interface *interface-type interface-number*
3. Specify the IPv4 UDP transport protocol for PTP messages.
   ptp transport-protocol udp

   This command is supported in both Layer 2 and Layer 3 Ethernet interface views.

   By default, PTP messages are encapsulated in IEEE 802.3/Ethernet packets.

Configuring a source IP address for multicast PTP message transmission over IPv4 UDP

About this task

To transport multicast PTP messages over IPv4 UDP, you must configure a source IP address for the messages.

Restrictions and guidelines

If both a source IP address for multicast PTP message transmission over IPv4 UDP and a destination address for unicast PTP message transmission over IPv4 UDP are configured, the system unicasts the messages.

This task is not available for the IEEE 802.1AS PTP profile.

Procedure

1. Enter system view.
   system-view
2. (Optional.) Assign the interface to a PTP instance and enter interface PTP instance view.
   ptp instance *ptp-instance-id*

   To configure settings for PTP instance 0, skip this step.
3. Configure a source IP address for multicast PTP message transmission over IPv4 UDP.
   ptp source *ip-address* [**vpn-instance** *vpn-instance-name*]

By default, no source IP address is configured for multicast PTP message transmission over IPv4 UDP.

Configuring a destination IP address for unicast PTP message transmission over IPv4 UDP

About this task

To transport unicast PTP messages over IPv4 UDP, you must configure a destination IP address for the messages.

Restrictions and guidelines

If both a source IP address for multicast PTP message transmission over IPv4 UDP and a destination address for unicast PTP message transmission over IPv4 UDP are configured, the system unicasts the messages.

This task is not available for the IEEE 802.1AS PTP profile.

Prerequisites

Configure an IP address for the current interface, and make sure the interface and the peer PTP interface can reach each other.

Procedure

1. Enter system view.

   system-view
2. Enter Layer 3 Ethernet interface view.

   interface *interface-type interface-number*
3. (Optional.) Assign the interface to a PTP instance and enter interface PTP instance view.

   ptp instance *ptp-instance-id*

   To configure settings for PTP instance 0, skip this step.
4. Configure a destination IP address for unicast PTP message transmission over IPv4 UDP.

   ptp unicast-destination *ip-address*

   By default, no destination IP address is configured for unicast PTP message transmission over IPv4 UDP.

Configuring the destination MAC address for non-Pdelay messages

About this task

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

Restrictions and guidelines

This feature takes effect only when PTP messages are encapsulated in IEEE 802.3/Ethernet packets.

This task is not available for the IEEE 802.1AS, AES67-2015, or SMPTE ST 2059-2 PTP profile.

Procedure

1. Enter system view.

system-view

2. (Optional.) Assign the interface to a PTP instance and enter interface PTP instance view.
   ptp instance *ptp-instance-id*
   To configure settings for PTP instance 0, skip this step.
3. Enter Layer 2 Ethernet interface view or Layer 3 Ethernet interface view.
   interface *interface-type interface-number*
4. Configure the destination MAC address for non-Pdelay messages.
   ptp destination-mac *mac-address*
   The default destination MAC address is 011B-1900-0000.

Setting a DSCP value for PTP messages transmitted over IPv4 UDP

About this task

The DSCP value determines the sending precedence of PTP messages transmitted over IPv4 UDP.

Restrictions and guidelines

This task is not available for the IEEE 802.1AS PTP profile.

Procedure

1. Enter system view.
   system-view
2. Enter Layer 2 Ethernet interface view or Layer 3 Ethernet interface view.
   interface *interface-type interface-number*
3. (Optional.) Assign the interface to a PTP instance and enter interface PTP instance view.
   ptp instance *ptp-instance-id*
   To configure settings for PTP instance 0, skip this step.
4. Set a DSCP value for PTP messages transmitted over IPv4 UDP.
   ptp dscp *dscp*
   This command is supported in both Layer 2 and Layer 3 Ethernet views.
   By default, the DSCP value is 56.

Specifying a VLAN tag for PTP messages

About this task

Perform this task to configure the VLAN ID and the 802.1p precedence in the VLAN tag carried by PTP messages.

Procedure

1. Enter system view.
   system-view
2. Enter Layer 2 Ethernet interface view.
   interface *interface-type interface-number*
3. (Optional.) Assign the interface to a PTP instance and enter interface PTP instance view.
   ptp instance *ptp-instance-id*
   To configure settings for PTP instance 0, skip this step.

4. Specify a VLAN tag for PTP messages.

   ptp vlan *vlan-id* [**dot1p** *dot1p-value*]

   By default, PTP messages do not have a VLAN tag.

Adjusting and correcting clock synchronization

Setting the delay correction value

About this task

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

Procedure

1. Enter system view.

   system-view

2. Enter Layer 2 Ethernet interface view or Layer 3 Ethernet interface view.

   interface *interface-type interface-number*

3. (Optional.) Assign the interface to a PTP instance and enter interface PTP instance view.

   ptp instance *ptp-instance-id*

   To configure settings for PTP instance 0, skip this step.

4. Set a delay correction value.

   ptp asymmetry-correction { **minus** | **plus** } *value*

   The default is 0 nanoseconds. Delay correction is not performed.

Setting the cumulative offset between the UTC and TAI

About this task

An offset exists between Coordinated Universal Time (UTC) and International Atomic Time (TAI). The device displays the UTC time. However, PTP uses TAI for time synchronization. For the device to synchronize correct time to other clock nodes in the PTP domain when its local clock is selected as the GM, configure this task to correct the offset between UTC and TAI.

Procedure

1. Enter system view.

   system-view

2. (Optional.) Enter PTP instance view.

   ptp instance *ptp-instance-id*

   To configure settings for PTP instance 0, skip this step.

3. Set the cumulative offset between the UTC and TAI.

   ptp utc offset *utc-offset*

   The default is 0 seconds.

Setting the correction date of the UTC

About this task

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

Restrictions and guidelines

If you configure the setting multiple times, the most recent configuration takes effect.

This setting takes effect only when it is configured on the master clock node and when the local clock of the master clock node is the GM.

Procedure

1. Enter system view.
   system-view
2. (Optional.) Enter PTP instance view.
   ptp instance *ptp-instance-id*
   To configure settings for PTP instance 0, skip this step.
3. Set the correction date of the UTC.
   ptp utc { **leap59-date** | **leap61-date** } *date*
   By default, the correction date of the UTC is not configured.

Configuring a priority for a clock

About this task

Priorities for clocks are used to elect the GM. The smaller the priority value, the higher the priority.

Procedure

1. Enter system view.
   system-view
2. (Optional.) Enter PTP instance view.
   ptp instance *ptp-instance-id*
   To configure settings for PTP instance 0, skip this step.
3. Configure the priority for the specified clock for GM election through BMC.
   ptp priority clock-source local { **priority1** *priority1* | **priority2** *priority2* }
   The default value varies by PTP profile:
   - **IEEE 1588 version 2, SMPTE ST 2059-2, or AES67-2015**—The priority 1 and priority 2 values are both 128.
   - **IEEE 802.1AS**—The priority 1 value is 246 and the priority 2 value is 248.

Configuring **the PTP time locking and unlocking thresholds**

About this task

This task enables the system to output logs that indicate the time-locked or time-unlocked state.

- When the time offset of the PTP reference clock crosses the PTP time locking threshold, the PTP time is put into unlocked state. The system outputs a time-unlocked log for notification.
- When the time offset of the PTP reference clock drops to or below the PTP time locking threshold, the PTP time is put into locked state. The system outputs a time-locked log for notification.

Procedure

1. Enter system view.
   system-view
2. **Configure the PTP time locking and unlocking thresholds.**
   ptp alarm-threshold { **time-lock** *lock-value* | **time-unlock** *unlock-value* } *
   By default, the PTP time locking threshold is 200 ns and the unlocking threshold is 300 ns.

Display and maintenance commands for PTP

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

Task	Command
Display PTP clock information.	**display ptp clock** [**all** | **instance** *ptp-instance-id*]
Display the delay correction history.	**display ptp corrections** [**all** | **instance** *ptp-instance-id*]
Display information about foreign master nodes.	**display ptp foreign-masters-record** [**interface** *interface-type interface-number*] [**all** | **instance** *ptp-instance-id*]
Display PTP information for one or all PTP interfaces.	**display ptp interface** [*interface-type interface-number* | **brief**] [**all** | **instance** *ptp-instance-id*]
Display brief PTP information for all PTP interfaces.	**display ptp interface brief**
Display parent node information for the PTP device.	**display ptp parent** [**all** | **instance** *ptp-instance-id*]
Display historical role change information for PTP ports.	**display ptp port-history** [**interface** *interface-type interface-number*]
Display PTP statistics.	**display ptp statistics** [**interface** *interface-type interface-number*] [**all** | **instance** *ptp-instance-id*]
Display PTP clock time properties.	**display ptp time-property** [**all** | **instance** *ptp-instance-id*]
Clear PTP statistics.	**reset ptp statistics** [**interface** *interface-type interface-number*] [**all** | **instance** *ptp-instance-id*]

PTP configuration examples

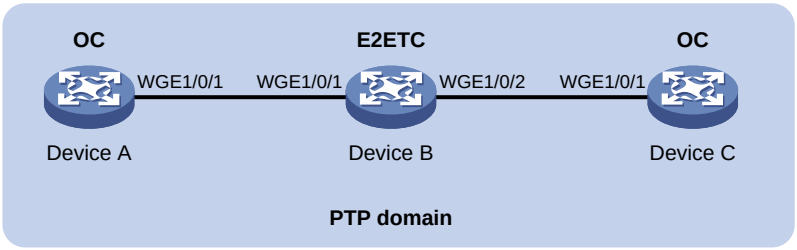
Example: Configuring PTP (IEEE 1588 version 2, IEEE 802.3/Ethernet transport, multicast transmission)

Network configuration

As shown in Figure 51, configure PTP (IEEE 1588 version 2, IEEE 802.3/Ethernet transport, multicast transmission) to enable time synchronization between Device A and Device C.

- Specify the IEEE 1588 version 2 PTP profile and the IEEE 802.3/Ethernet transport protocol for PTP messages on Device A, Device B, and Device C.
- Use the IEEE 802.3/Ethernet transport protocol for PTP messages.
- Assign Device A, Device B, and Device C to the same PTP domain. Specify the OC clock node type for Device A and Device C, and E2ETC clock node type for Device B. All clock nodes elect a GM through BMC in the PTP domain.
- Use the default Request_Response delay measurement mechanism on Device A and Device C.

Figure 51 Network diagram



Procedure

1. Configure Device A:

   # Specify the IEEE 1588 version 2 PTP profile.

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

   # Specify the OC clock node type.

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

   # Create a PTP domain.

   ```
   [DeviceA] ptp domain 0
   ```

   # Enable PTP globally. (To run PTP on an interface, enable PTP globally and on the interface.)

   ```
   [DeviceA] ptp global enable
   ```

   # Specify PTP for obtaining the time.

   ```
   [DeviceA] clock protocol ptp
   ```

   # Enable PTP on Twenty-FiveGigE 1/0/1.

   ```
   [DeviceA] interface twenty-fivegige 1/0/1
   [DeviceA-Twenty-FiveGigE1/0/1] ptp enable
   [DeviceA-Twenty-FiveGigE1/0/1] quit
   ```

2. Configure Device B:

   # Specify the IEEE 1588 version 2 PTP profile.

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

```
[DeviceB] ptp profile 1588v2
```

Specify the E2ETC clock node type.

```
[DeviceB] ptp mode e2etc
```

Create a PTP domain.

```
[DeviceB] ptp domain 0
```

Enable PTP globally.

```
[DeviceB] ptp global enable
```

Specify PTP for obtaining the time.

```
[DeviceB] clock protocol ptp
```

Enable PTP on Twenty-FiveGigE 1/0/1.

```
[DeviceB] interface twenty-fivegige 1/0/1
[DeviceB-Twenty-FiveGigE1/0/1] ptp enable
[DeviceB-Twenty-FiveGigE1/0/1] quit
```

Enable PTP on Twenty-FiveGigE 1/0/2.

```
[DeviceB] interface twenty-fivegige 1/0/2
[DeviceB-Twenty-FiveGigE1/0/2] ptp enable
[DeviceB-Twenty-FiveGigE1/0/2] quit
```

3. Configure Device C:

Specify the IEEE 1588 version 2 PTP profile.

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

Specify the OC clock node type.

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

Create a PTP domain.

```
[DeviceC] ptp domain 0
```

Enable PTP globally.

```
[DeviceB] ptp global enable
```

Specify PTP for obtaining the time.

```
[DeviceC] clock protocol ptp
```

Enable PTP on Twenty-FiveGigE 1/0/1.

```
[DeviceC] interface twenty-fivegige 1/0/1
[DeviceC-Twenty-FiveGigE1/0/1] ptp enable
[DeviceC-Twenty-FiveGigE1/0/1] quit
```

Verifying the configuration

When the network is stable, perform the following tasks to verify that Device A is elected as the GM, Twenty-FiveGigE1/0/1 on Device A is the master port, and Device B has synchronized to Device A:

- Use the **display ptp clock** command to display PTP clock information.
- Use the **display ptp interface brief** command to display brief PTP running information for all PTP interfaces.

Display PTP clock information on Device A.

```
[DeviceA] display ptp clock
PTP global state    : Enabled
PTP profile         : IEEE 1588 Version 2
PTP mode            : OC
Slave only          : No
Lock status         : Locked
```

```
Clock ID            : 000FE2-FFFE-FF0000
Clock type          : Local
Clock domain        : 0
Number of PTP ports : 1
Priority1     : 128
Priority2     : 128
Clock quality :
 Class                 : 248
 Accuracy              : 254
 Offset (log variance) : 65535
Offset from master : 0 (ns)
Mean path delay    : 0 (ns)
Steps removed      : 0
Local clock time   : Sun Jan 15 20:57:29 2019
```

Display brief PTP running information for all PTP interfaces on Device A.

```
[DeviceA] display ptp interface brief
Name         State         Delay mechanism  Clock step  Asymmetry correction
WGE1/0/1     Master        E2E              Two         0
```

Display PTP clock information on Device B.

```
[DeviceB] display ptp clock
PTP global state    : Enabled
PTP profile         : IEEE 1588 Version 2
PTP mode            : E2ETC
Slave only          : No
Lock status         : Locked
Clock ID            : 000FE2-FFFE-FF0001
Clock type          : Local
Clock domain        : 0
Number of PTP ports : 2
Priority1     : 128
Priority2     : 128
Clock quality :
 Class                 : 248
 Accuracy              : 254
 Offset (log variance) : 65535
Offset from master  : 106821530000 (ns)
Mean path delay     : 2801000 (ns)
Steps removed       : 1
Local clock time    : Sun Jan 15 20:57:29 2019
```

Display brief PTP running information for all PTP interfaces on Device B.

```
[DeviceB] display ptp interface brief
Name           State         Delay mechanism  Clock step  Asymmetry correction
WGE1/0/1       N/A           E2E              Two         0
WGE1/0/2       N/A           E2E              Two         0
```

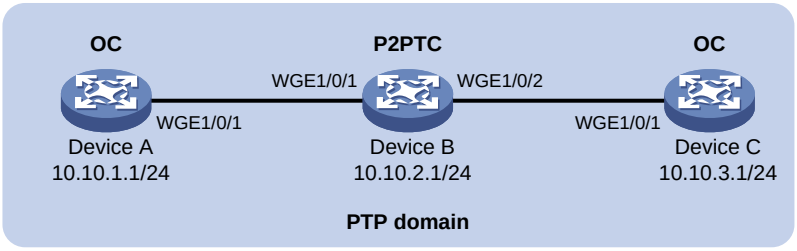
Example: Configuring PTP (IEEE 1588 version 2, IPv4 UDP transport, multicast transmission)

Network configuration

As shown in Figure 52, configure PTP (IEEE 1588 version 2, IPv4 UDP transport, multicast transmission) to enable time synchronization between the devices.

- Specify the IEEE 1588 version 2 PTP profile and IPv4 UDP transport of PTP messages on Device A, Device B, and Device C.
- Assign Device A, Device B, and Device C to the same PTP domain. Specify the OC clock node type for Device A and Device C, and the P2PTC clock node type for Device B. All clock nodes elect a GM through BMC in the PTP domain.
- Configure the peer delay measurement mechanism (p2p) for Device A and Device C.

Figure 52 Network diagram



Procedure

1. Configure Device A:

   # Specify the IEEE 1588 version 2 PTP profile.

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

   # Specify the OC clock node type.

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

   # Create a PTP domain.

   ```
   [DeviceA] ptp domain 0
   ```

   # Enable PTP globally. (To run PTP on an interface, enable PTP globally and on this interface.)

   ```
   [DeviceB] ptp global enable
   ```

   # Configure the source IP address for multicast PTP message transmission over IPv4 UDP.

   ```
   [DeviceA] ptp source 10.10.10.1
   ```

   # Specify PTP for obtaining the time.

   ```
   [DeviceA] clock protocol ptp
   ```

   # On Twenty-FiveGigE 1/0/1, specify the IPv4 UDP transport protocol for PTP messages, specify the peer delay measurement mechanism, and enable PTP.

   ```
   [DeviceA] interface twenty-fivegige 1/0/1
   [DeviceA-Twenty-FiveGigE1/0/1] ptp transport-protocol udp
   [DeviceA-Twenty-FiveGigE1/0/1] ptp delay-mechanism p2p
   [DeviceA-Twenty-FiveGigE1/0/1] ptp enable
   [DeviceA-Twenty-FiveGigE1/0/1] quit
   ```

2. Configure Device B:

   # Specify the IEEE 1588 version 2 PTP profile.

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

```
[DeviceB] ptp profile 1588v2
```

Specify the P2PTC clock node type.

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

Create a PTP domain.

```
[DeviceB] ptp domain 0
```

Enable PTP globally.

```
[DeviceB] ptp global enable
```

Configure the source IP address for multicast PTP message transmission over IPv4 UDP.

```
[DeviceB] ptp source 10.10.10.2
```

Specify PTP for obtaining the time.

```
[DeviceB] clock protocol ptp
```

On Twenty-FiveGigE 1/0/1, specify the IPv4 UDP transport protocol for PTP messages and enable PTP.

```
[DeviceB] interface twenty-fivegige 1/0/1
DeviceB-Twenty-FiveGigE1/0/1] ptp transport-protocol udp
[DeviceB-Twenty-FiveGigE1/0/1] ptp enable
[DeviceB-Twenty-FiveGigE1/0/1] quit
```

On Twenty-FiveGigE 1/0/2, specify the IPv4 UDP transport protocol for PTP messages and enable PTP.

```
[DeviceB] interface twenty-fivegige 1/0/2
[DeviceB-Twenty-FiveGigE1/0/2] ptp transport-protocol udp
[DeviceB-Twenty-FiveGigE1/0/2] ptp enable
[DeviceB-Twenty-FiveGigE1/0/2] quit
```

3. Configure Device C:

Specify the IEEE 1588 version 2 PTP profile.

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

Specify the OC clock node type.

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

Create a PTP domain.

```
[DeviceC] ptp domain 0
```

Enable PTP globally.

```
[DeviceB] ptp global enable
```

Configure the source IP address for multicast PTP message transmission over IPv4 UDP.

```
[DeviceC] ptp source 10.10.3.1
```

Specify PTP for obtaining the time.

```
[DeviceC] clock protocol ptp
```

On Twenty-FiveGigE 1/0/1, specify the IPv4 UDP transport protocol for PTP messages, specify the peer delay measurement mechanism, and enable PTP.

```
[DeviceC] interface twenty-fivegige 1/0/1
[DeviceC-Twenty-FiveGigE1/0/1] ptp transport-protocol udp
[DeviceC-Twenty-FiveGigE1/0/1] ptp delay-mechanism p2p
[DeviceC-Twenty-FiveGigE1/0/1] ptp enable
[DeviceC-Twenty-FiveGigE1/0/1] quit
```

Verifying the configuration

When the network is stable, perform the following tasks to verify that Device A is elected as the GM, Twenty-FiveGigE1/0/1 on Device A is the master port, and Device B has synchronized to Device A:

- Use the **display ptp clock** command to display PTP clock information.
- Use the **display ptp interface brief** command to display brief PTP running information for all PTP interfaces.

Display PTP clock information on Device A.

```
[DeviceA] display ptp clock
PTP global state    : Enabled
PTP profile         : IEEE 1588 Version 2
PTP mode            : OC
Slave only          : No
Lock status         : Locked
Clock ID            : 000FE2-FFFE-FF0000
Clock type          : Local
Clock domain        : 0
Number of PTP ports : 1
Priority1     : 128
Priority2     : 128
Clock quality :
 Class                 : 248
 Accuracy              : 254
 Offset (log variance) : 65535
Offset from master : 0 (ns)
Mean path delay    : 0 (ns)
Steps removed      : 0
Local clock time   : Sun Jan 15 20:57:29 2019
```

Display brief PTP running information for all PTP interfaces on Device A.

```
[DeviceA] display ptp interface brief
Name         State         Delay mechanism  Clock step  Asymmetry correction
WGE1/0/1     Master        P2P              Two         0
```

Display PTP clock information on Device B.

```
[DeviceB] display ptp clock
PTP global state    : Enabled
PTP profile         : IEEE 1588 Version 2
PTP mode            : P2PTC
Slave only          : No
Lock status         : Locked
Clock ID            : 000FE2-FFFE-FF0001
Clock type          : Local
Clock domain        : 0
Number of PTP ports : 2
Priority1     : 128
Priority2     : 128
Clock quality :
 Class                 : 248
 Accuracy              : 254
 Offset (log variance) : 65535
Offset from master  : 106368000000 (ns)
Mean path delay     : 2700000 (ns)
Steps removed       : 1
```

```
Local clock time    : Sun Jan 15 20:57:29 2019
```

Display brief PTP running information for all PTP interfaces on Device B.

```
[DeviceB] display ptp interface brief
Name          State         Delay mechanism  Clock step  Asymmetry correction
WGE1/0/1      N/A           P2P              Two         0
WGE1/0/2      N/A           P2P              Two         0
```

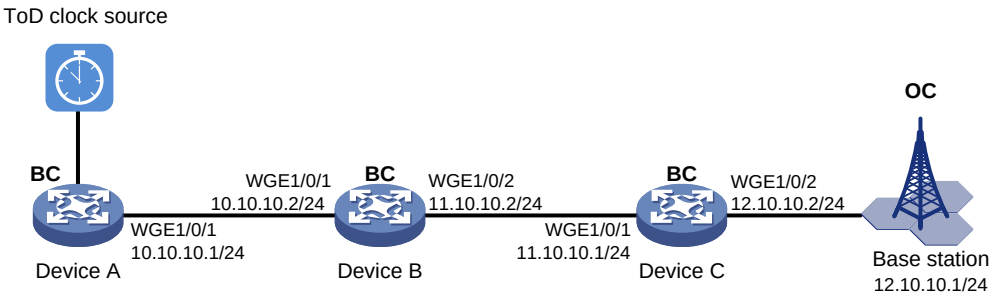
Example: Configuring PTP (IEEE 1588 version 2, IPv4 UDP transport, unicast transmission)

Network configuration

As shown in Figure 53, configure PTP (IEEE 1588 version 2, IPv4 UDP transport, unicast transmission) to enable Device A, Device B, Device C, and the base station to synchronize the time with the ToD clock source.

- Specify the IEEE 1588 version 2 PTP profile and unicast IPv4 UDP transport of PTP messages for Device A, Device B, and Device C.
- Assign Device A, Device B, Device C, and the base station to PTP domain 0. Specify the BC clock node type for Device A, Device B, and Device C.
- Connect Device A to the ToD clock source and Device C to the base station.
- Use the default Request_Response delay measurement mechanism on all clock nodes in the PTP domain.

Figure 53 Network diagram



Restrictions and guidelines

The switch does not provide ToD interfaces. It can be used as Device B or C but not Device A.

Procedure

1. Assign IP addresses to the interfaces, and make sure the devices can reach each other, as shown in Figure 53. (Details not shown.)
2. Configure Device A:

   # Specify the IEEE 1588 version 2 PTP profile.

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

   # Specify the BC clock node type.

   ```
   [DeviceA] ptp mode bc
   ```

   # Create a PTP domain.

   ```
   [DeviceA] ptp domain 0
   ```

   # Configure the delay time correction as 1000 nanoseconds for receiving ToD 0 clock signals.

```
[DeviceA] ptp tod0 input delay 1000
```

Set priority 1 to **0** for the ToD 0 clock.

```
[DeviceA] ptp priority clock-source tod0 priority1 0
```

On Twenty-FiveGigE 1/0/1, configure the destination IP address for unicast PTP message transmission over IPv4 UDP, and enable PTP.

```
[DeviceA] interface twenty-fivegige 1/0/1
[DeviceA-Twenty-FiveGigE1/0/1] ptp transport-protocol udp
[DeviceA-Twenty-FiveGigE1/0/1] ptp unicast-destination 10.10.10.2
[DeviceA-Twenty-FiveGigE1/0/1] ptp enable
[DeviceA-Twenty-FiveGigE1/0/1] quit
```

3. Configure Device B:

Specify the IEEE 1588 version 2 PTP profile.

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

Specify the BC clock node type.

```
[DeviceB] ptp mode bc
```

Create a PTP domain.

```
[DeviceB] ptp domain 0
```

Enable PTP globally. (To run PTP on an interface, enable PTP globally and on this interface.)

```
[DeviceB] ptp global enable
```

Specify PTP for obtaining the time.

```
[DeviceB] clock protocol ptp
```

On Twenty-FiveGigE 1/0/1, configure the destination IP address for unicast PTP message transmission over IPv4 UDP, and enable PTP.

```
[DeviceB] interface twenty-fivegige 1/0/1
[DeviceB-Twenty-FiveGigE1/0/1] ptp transport-protocol udp
[DeviceB-Twenty-FiveGigE1/0/1] ptp unicast-destination 10.10.10.1
[DeviceB-Twenty-FiveGigE1/0/1] ptp enable
[DeviceB-Twenty-FiveGigE1/0/1] quit
```

On Twenty-FiveGigE 1/0/2, configure the destination IP address for unicast PTP message transmission over IPv4 UDP, and enable PTP.

```
[DeviceB] interface twenty-fivegige 1/0/2
[DeviceB-Twenty-FiveGigE1/0/2] ptp transport-protocol udp
[DeviceB-Twenty-FiveGigE1/0/2] ptp unicast-destination 11.10.10.1
[DeviceB-Twenty-FiveGigE1/0/2] ptp enable
[DeviceB-Twenty-FiveGigE1/0/2] quit
```

4. Configure Device C:

Specify the IEEE 1588 version 2 PTP profile.

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

Specify the BC clock node type.

```
[DeviceC] ptp mode bc
```

Create a PTP domain.

```
[DeviceC] ptp domain 0
```

Enable PTP globally.

```
[DeviceC] ptp global enable
```

Specify PTP for obtaining the time.

```
[DeviceC] clock protocol ptp
```

On Twenty-FiveGigE 1/0/1, configure the destination IP address for unicast PTP message transmission over IPv4 UDP, and enable PTP.

```
[DeviceC] interface twenty-fivegige 1/0/1
[DeviceC-Twenty-FiveGigE1/0/1] ptp transport-protocol udp
[DeviceC-Twenty-FiveGigE1/0/1] ptp unicast-destination 11.10.10.2
[DeviceC-Twenty-FiveGigE1/0/1] ptp enable
[DeviceC-Twenty-FiveGigE1/0/1] quit
```

On Twenty-FiveGigE1/0/2, specify IPv4 UDP transport of PTP messages, configure the destination IP address for unicast PTP messages, and enable PTP.

```
[DeviceC] interface twenty-fivegige 1/0/2
[DeviceC-Twenty-FiveGigE1/0/2] ptp transport-protocol udp
[DeviceC-Twenty-FiveGigE1/0/2] ptp unicast-destination 12.10.10.1
[DeviceC-Twenty-FiveGigE1/0/2] ptp enable
[DeviceC-Twenty-FiveGigE1/0/2] quit
```

5. Configure the base station.

Specify PTP domain 0.

Specify IPv4 UDP transport of PTP messages.

Set the destination IP address of unicast PTP messages to 12.10.10.2.

Specify the Request_Response delay measurement mechanism.

For more information, see the configuration guide for the base station.

Verifying the configuration

When the network is stable, perform the following tasks:

- Use the **display ptp clock** command to display PTP clock information.
- Use the **display ptp interface brief** command to display brief PTP running information for all PTP interfaces.

Display PTP clock information on Device A.

```
[DeviceA] display ptp clock
PTP global state    : Enabled
PTP profile         : IEEE 1588 Version 2
PTP mode            : BC
Slave only          : No
Lock status         : Locked
Clock ID            : 000FE2-FFFE-FF0000
Clock type          : ToD0
 ToD direction  : In
 ToD delay time : 1000 (ns)
Clock domain        : 0
Number of PTP ports : 1
Priority1     : 0
Priority2     : 128
Clock quality :
 Class                 : 6
 Accuracy              : 32
 Offset (log variance) : 65535
Offset from master : 0 (ns)
Mean path delay    : 0 (ns)
```

```
Steps removed      : 0
Local clock time   : Sun Jan 15 20:57:29 2019
```

Display brief PTP running information for all PTP interfaces on Device A.

```
[DeviceA] display ptp interface brief
Name         State         Delay mechanism  Clock step  Asymmetry correction
WGE1/0/1     Master        E2E              Two         0
```

Display PTP clock information on Device C.

```
[DeviceC] display ptp clock
PTP global state    : Enabled
PTP profile         : IEEE 1588 Version 2
PTP mode            : BC
Slave only          : No
Lock status         : Locked
Clock ID            : 000FE2-FFFE-FF0001
Clock type          : Local
Clock domain        : 0
Number of PTP ports : 2
Priority1     : 128
Priority2     : 128
Clock quality :
 Class                 : 248
 Accuracy              : 254
 Offset (log variance) : 65535
Offset from master  : 106368539000 (ns)
Mean path delay     : 2791000 (ns)
Steps removed       : 2
Local clock time   : Sun Jan 15 20:57:29 2019
```

Display brief PTP running information for all PTP interfaces on Device C.

```
[DeviceC] display ptp interface brief
Name         State         Delay mechanism  Clock step  Asymmetry correction
WGE1/0/1     N/A           E2E              Two         0
WGE1/0/2     N/A           E2E              Two         0
```

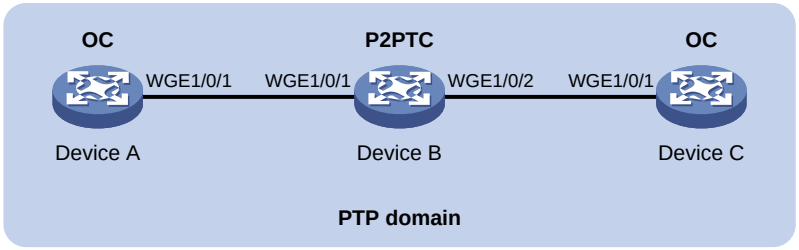
Example: Configuring PTP (IEEE 802.1AS, IEEE 802.3/Ethernet transport, multicast transmission)

Network configuration

As shown in Figure 54, configure PTP (IEEE 802.1AS, IEEE 802.3/Ethernet transport, multicast transmission) to enable time synchronization between Device A, Device B, and Device C.

- Specify the IEEE 802.1AS PTP profile for Device A, Device B, and Device C.
- Assign Device A, Device B, and Device C to the same PTP domain. Specify the OC clock node type for Device A and Device C, and P2PTC clock node type for Device B. The clock nodes elect a GM through BMC in the PTP domain.
- Use the default peer delay measurement mechanism on all clock nodes in the PTP domain.

Figure 54 Network diagram



Procedure

IMPORTANT:

The IEEE 802.1AS PTP profile transports PTP messages over IEEE 802.3/Ethernet rather than IPv4 UDP and in multicast rather than unicast mode.

1. Configure Device A:

   # Specify the IEEE 802.1AS PTP profile.

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

   # Specify the OC clock node type.

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

   # Create a PTP domain.

   ```
   [DeviceA] ptp domain 0
   ```

   # Enable PTP globally. (To run PTP on an interface, enable PTP globally and on this interface.)

   ```
   [DeviceA] ptp global enable
   ```

   # Specify PTP for obtaining the time.

   ```
   [DeviceA] clock protocol ptp
   ```

   # Enable PTP on Twenty-FiveGigE 1/0/1.

   ```
   [DeviceA] interface twenty-fivegige 1/0/1
   [DeviceA-Twenty-FiveGigE1/0/1] ptp enable
   [DeviceA-Twenty-FiveGigE1/0/1] quit
   ```

2. Configure Device B:

   # Specify the IEEE 802.1AS PTP profile.

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

   # Specify the P2PTC clock node type.

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

   # Create a PTP domain.

   ```
   [DeviceB] ptp domain 0
   ```

   # Enable PTP globally.

   ```
   [DeviceB] ptp global enable
   ```

   # Specify PTP for obtaining the time.

   ```
   [DeviceB] clock protocol ptp
   ```

   # Enable PTP on Twenty-FiveGigE 1/0/1.

   ```
   [DeviceB] interface twenty-fivegige 1/0/1
   [DeviceB-Twenty-FiveGigE1/0/1] ptp enable
   [DeviceB-Twenty-FiveGigE1/0/1] quit
   ```

Enable PTP on Twenty-FiveGigE 1/0/2.

```
[DeviceB] interface twenty-fivegige 1/0/2
[DeviceB-Twenty-FiveGigE1/0/2] ptp enable
[DeviceB-Twenty-FiveGigE1/0/2] quit
```

3. Configure Device C:

Specify the IEEE 1588 802.1AS PTP profile.

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

Specify the OC clock node type.

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

Create a PTP domain.

```
[DeviceC] ptp domain 0
```

Enable PTP globally.

```
[DeviceB] ptp global enable
```

Specify PTP for obtaining the time.

```
[DeviceC] clock protocol ptp
```

Enable PTP on Twenty-FiveGigE 1/0/1.

```
[DeviceC] interface twenty-fivegige 1/0/1
[DeviceC-Twenty-FiveGigE1/0/1] ptp enable
[DeviceC-Twenty-FiveGigE1/0/1] quit
```

Verifying the configuration

When the network is stable, perform the following tasks to verify that Device A is elected as the GM, Twenty-FiveGigE1/0/1 on Device A is the master port, and Device B has synchronized to Device A:

- Use the **display ptp clock** command to display PTP clock information.
- Use the **display ptp interface brief** command to display brief PTP running information for all PTP interfaces.

Display PTP clock information on Device A.

```
[DeviceA] display ptp clock
PTP global state    : Enabled
PTP profile         : IEEE 802.1AS
PTP mode            : OC
Slave only          : No
Lock status         : Locked
Clock ID            : 000FE2-FFFE-FF0000
Clock type          : Local
Clock domain        : 0
Number of PTP ports : 1
Priority1     : 246
Priority2     : 248
Clock quality :
 Class                 : 248
 Accuracy              : 254
 Offset (log variance) : 16640
Offset from master : 0 (ns)
Mean path delay    : 0 (ns)
Steps removed      : 0
Local clock time   : Sun Jan 15 20:57:29 2019
```

Display brief PTP running information for all PTP interfaces on Device A.

```
[DeviceA] display ptp interface brief
Name         State         Delay mechanism  Clock step  Asymmetry correction
WGE1/0/1     Master        P2P              Two         0
```

Display PTP clock information on Device B.

```
[DeviceB] display ptp clock
PTP global state    : Enabled
PTP profile         : IEEE 802.1AS
PTP mode            : P2PTC
Slave only          : No
Lock status         : Locked
Clock ID            : 000FE2-FFFE-FF0001
Clock type          : Local
Clock domain        : 0
Number of PTP ports : 2
Priority1     : 246
Priority2     : 248
Clock quality :
 Class                 : 248
 Accuracy              : 254
 Offset (log variance) : 16640
Offset from master  : 106368539000 (ns)
Mean path delay     : 2791000 (ns)
Steps removed       : 1
Local clock time    : Sun Jan 15 20:57:29 2019
```

Display brief PTP running information for all PTP interfaces on Device B.

```
[DeviceB] display ptp interface brief
Name         State         Delay mechanism  Clock step  Asymmetry correction
WGE1/0/1     N/A           P2P              Two         0
WGE1/0/2     N/A           P2P              Two         0
```

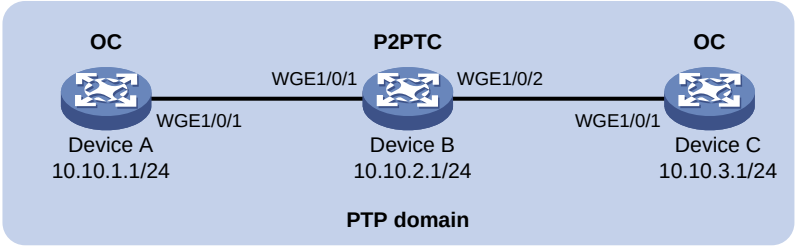
Example: Configuring PTP (SMPTE ST 2059-2, IPv4 UDP transport, multicast transmission)

Network configuration

As shown in Figure 55, configure PTP (SMPTE ST 2059-2, IPv4 UDP transport, multicast transmission) to enable time synchronization between Device A and Device C:

- Specify the SMPTE ST 2059-2 PTP profile and multicast IPv4 UDP transport of PTP messages for Device A, Device B, and Device C.
- Specify the OC clock node type for Device A and Device C, and the P2PTC clock node type for Device B. All clock nodes elect a GM through BMC.
- Use the peer delay measurement mechanism on all clock nodes in the PTP domain.

Figure 55 Network diagram



Procedure

IMPORTANT:

The SMPTE ST 2059-2 PTP profile transports PTP messages over IPv4 UDP rather than IEEE 802.3/Ethernet. The profile supports both multicast and unicast transmission modes.

1. Configure Device A:

   # Specify the SMPTE ST 2059-2 PTP profile.

   ```
   <DeviceA> system-view
   [DeviceA] ptp profile st2059-2
   ```

   # Specify the OC clock node type.

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

   # Create a PTP domain.

   ```
   [DeviceA] ptp domain 0
   ```

   # Enable PTP globally. (To run PTP on an interface, enable PTP globally and on the interface.)

   ```
   [DeviceA] ptp global enable
   ```

   # Configure the source IP address for multicast PTP message transmission over IPv4 UDP.

   ```
   [DeviceA] ptp source 10.10.1.1
   ```

   # Specify PTP for obtaining the time.

   ```
   [DeviceA] clock protocol ptp
   ```

   # On Twenty-FiveGigE 1/0/1, specify the peer delay measurement mechanism and enable PTP.

   ```
   [DeviceA] interface twenty-fivegige 1/0/1
   [DeviceA-Twenty-FiveGigE1/0/1] ptp transport-protocol udp
   [DeviceA-Twenty-FiveGigE1/0/1] ptp delay-mechanism p2p
   [DeviceA-Twenty-FiveGigE1/0/1] ptp enable
   [DeviceA-Twenty-FiveGigE1/0/1] quit
   ```

2. Configure Device B:

   # Specify the SMPTE ST 2059-2 PTP profile.

   ```
   <DeviceB> system-view
   [DeviceB] ptp profile st2059-2
   ```

   # Specify the P2PTC clock node type.

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

   # Create a PTP domain.

   ```
   [DeviceB] ptp domain 0
   ```

   # Enable PTP globally.

   ```
   [DeviceB] ptp global enable
   ```

   # Configure the source IP address for multicast PTP message transmission over IPv4 UDP.

   ```
   [DeviceB] ptp source 10.10.10.2
   ```

Specify PTP for obtaining the time.

```
[DeviceB] clock protocol ptp
```

On Twenty-FiveGigE 1/0/1, enable PTP.

```
[DeviceB] interface twenty-fivegige 1/0/1
DeviceB-Twenty-FiveGigE1/0/1] ptp transport-protocol udp
[DeviceB-Twenty-FiveGigE1/0/1] ptp enable
[DeviceB-Twenty-FiveGigE1/0/1] quit
```

On Twenty-FiveGigE 1/0/2, enable PTP.

```
[DeviceB] interface twenty-fivegige 1/0/2
[DeviceB-Twenty-FiveGigE1/0/2] ptp transport-protocol udp
[DeviceB-Twenty-FiveGigE1/0/2] ptp enable
[DeviceB-Twenty-FiveGigE1/0/2] quit
```

3. Configure Device C:

Specify the SMPTE ST 2059-2 PTP profile.

```
<DeviceC> system-view
[DeviceC] ptp profile st2059-2
```

Specify the OC clock node type.

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

Create a PTP domain.

```
[DeviceC] ptp domain 0
```

Enable PTP globally.

```
[DeviceC] ptp global enable
```

Configure the source IP address for multicast PTP message transmission over IPv4 UDP.

```
[DeviceC] ptp source 11.10.10.1
```

Specify PTP for obtaining the time.

```
[DeviceC] clock protocol ptp
```

On Twenty-FiveGigE 1/0/1, specify the delay measurement mechanism as **p2p** and enable PTP.

```
[DeviceC] interface twenty-fivegige 1/0/1
[DeviceC-Twenty-FiveGigE1/0/1] ptp transport-protocol udp
[DeviceC-Twenty-FiveGigE1/0/1] ptp delay-mechanism p2p
[DeviceC-Twenty-FiveGigE1/0/1] ptp enable
[DeviceC-Twenty-FiveGigE1/0/1] quit
```

Verifying the configuration

When the network is stable, perform the following tasks to verify the PTP configuration:

- Use the **display ptp clock** command to display PTP clock information.
- Use the **display ptp interface brief** command to display brief PTP running information for all PTP interfaces.

Display PTP clock information on Device A.

```
[DeviceA] display ptp clock
PTP global state    : Enabled
PTP profile         : SMPTE ST 2059-2
PTP mode            : OC
Slave only          : No
Lock status         : Locked
Clock ID            : 000FE2-FFFE-FF0000
Clock type          : Local
```

```
Clock domain        : 0
Number of PTP ports : 1
Priority1     : 128
Priority2     : 128
Clock quality :
 Class                : 248
 Accuracy             : 254
 Offset (log variance) : 65535
Offset from master : 0 (ns)
Mean path delay    : 0 (ns)
Steps removed      : 0
Local clock time   : Sun Jan 15 20:57:29 2019
```

Display brief PTP running information for all PTP interfaces on Device A.

```
[DeviceA] display ptp interface brief
Name       InstID     State         Delay mechanism  Clock step  Asymmetry correction
WGE1/0/1   0          Master        P2P              Two         0
```

Display PTP clock information on Device B.

```
[DeviceB] display ptp clock
PTP global state    : Enabled
PTP profile         : SMPTE ST 2059-2
PTP mode            : P2PTC
Slave only          : No
Lock status         : Locked
Clock ID            : 000FE2-FFFE-FF0001
Clock type          : Local
Clock domain        : 0
Number of PTP ports : 2
Priority1     : 128
Priority2     : 128
Clock quality :
 Class                : 248
 Accuracy             : 254
 Offset (log variance) : 65535
Offset from master  : 106368539000 (ns)
Mean path delay     : 2791000 (ns)
Steps removed       : 1
Local clock time   : Sun Jan 15 20:57:29 2019
```

Display brief PTP running information for all PTP interfaces on Device B.

```
[DeviceB] display ptp interface brief
Name       InstID     State         Delay mechanism  Clock step  Asymmetry correction
WGE1/0/1   0          N/A           P2P              Two         0
WGE1/0/2   0          N/A           P2P              Two         0
```

The output shows that Device A is elected as the GM and Twenty-FiveGigE1/0/1 on Device A is the master port.

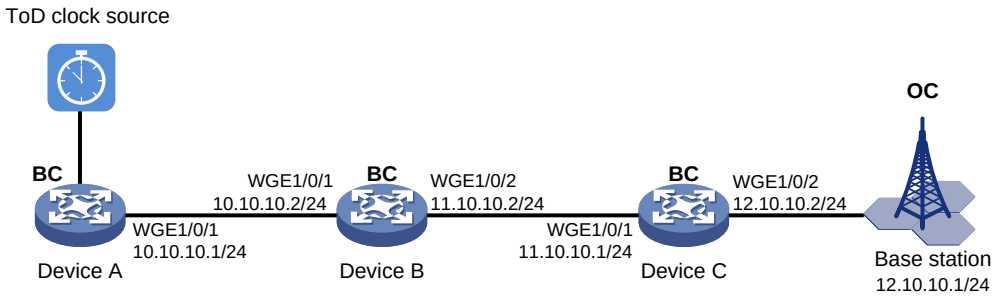
Example: Configuring PTP (SMPTE ST 2059-2, IPv4 UDP transport, unicast transmission)

Network configuration

As shown in Figure 56, configure PTP (SMPTE ST 2059-2, IPv4 UDP transport, unicast transmission) to enable Device A, Device B, Device C, and the base station to synchronize time with the ToD clock source .

- Specify the SMPTE ST 2059-2 PTP profile and unicast IPv4 UDP transport of PTP messages for Device A, Device B, and Device C.
- Assign Device A, Device B, Device C, and the base station to PTP domain 0. Specify the BC clock node type for Device A, Device B, and Device C.
- Connect Device A to the ToD clock source and Device C to the base station.
- Use the default Request_Response delay measurement mechanism on all clock nodes in the PTP domain.

Figure 56 Network diagram



Restrictions and guidelines

The switch does not provide ToD interfaces. It can be configured as Device B or C but not Device A.

Procedure

IMPORTANT:

The SMPTE ST 2059-2 PTP profile supports IPv4 UDP transport rather than IEEE 802.3/Ethernet transport of PTP messages. It supports both multicast and unicast transmission of PTP messages.

1. Assign IP addresses to the interfaces, and make sure the devices can reach each other, as shown in Figure 56. (Details not shown.)
2. Configure Device A:

   # Specify the SMPTE ST 2059-2 PTP profile.

   ```
   <DeviceA> system-view
   [DeviceA] ptp profile st2059-2
   ```

   # Specify the BC clock node type.

   ```
   [DeviceA] ptp mode bc
   ```

   # Create a PTP domain.

   ```
   [DeviceA] ptp domain 0
   ```

   # Configure the device to receive ToD 0 clock signals and set the delay correction value to 1000 nanoseconds.

   ```
   [DeviceA] ptp tod0 input delay 1000
   ```

   # Set priority 1 to **0** for the ToD 0 clock.

```
[DeviceA] ptp priority clock-source tod0 priority1 0
```

On Twenty-FiveGigE 1/0/1, configure the destination IP address for unicast PTP messages and enable PTP. (The SMPTE ST 2059-2 PTP profile uses IPv4 UDP transport of PTP messages by default.)

```
[DeviceA] interface twenty-fivegige 1/0/1
[DeviceA-Twenty-FiveGigE1/0/1] ptp transport-protocol udp
[DeviceA-Twenty-FiveGigE1/0/1] ptp unicast-destination 10.10.10.2
[DeviceA-Twenty-FiveGigE1/0/1] ptp enable
[DeviceA-Twenty-FiveGigE1/0/1] quit
```

3. Configure Device B:

Specify the SMPTE ST 2059-2 PTP profile.

```
<DeviceB> system-view
[DeviceB] ptp profile st2059-2
```

Specify the BC clock node type.

```
[DeviceB] ptp mode bc
```

Create a PTP domain.

```
[DeviceB] ptp domain 0
```

Enable PTP globally. (To run PTP on an interface, enable PTP globally and on the interface.)

```
[DeviceB] ptp global enable
```

Specify PTP for obtaining the time.

```
[DeviceB] clock protocol ptp
```

On Twenty-FiveGigE 1/0/1, configure the destination IP address for unicast PTP messages and enable PTP. (The SMPTE ST 2059-2 PTP profile uses IPv4 UDP transport of PTP messages by default.)

```
[DeviceB] interface twenty-fivegige 1/0/1
[DeviceB-Twenty-FiveGigE1/0/1] ptp unicast-destination 10.10.10.1
[DeviceB-Twenty-FiveGigE1/0/1] ptp enable
[DeviceB-Twenty-FiveGigE1/0/1] quit
```

On Twenty-FiveGigE 1/0/2, configure the destination IP address for unicast PTP messages and enable PTP. (The SMPTE ST 2059-2 PTP profile uses IPv4 UDP transport of PTP messages by default.)

```
[DeviceB] interface twenty-fivegige 1/0/2
[DeviceB-Twenty-FiveGigE1/0/2] ptp unicast-destination 11.10.10.1
[DeviceB-Twenty-FiveGigE1/0/2] ptp enable
[DeviceB-Twenty-FiveGigE1/0/2] quit
```

4. Configure Device C:

Specify the SMPTE ST 2059-2 PTP profile.

```
<DeviceC> system-view
[DeviceC] ptp profile st2059-2
```

Specify the BC clock node type.

```
[DeviceC] ptp mode bc
```

Create a PTP domain.

```
[DeviceC] ptp domain 0
```

Enable PTP globally.

```
[DeviceC] ptp global enable
```

Specify PTP for obtaining the time.

```
[DeviceC] clock protocol ptp
```

On Twenty-FiveGigE 1/0/1, configure the destination IP address for unicast PTP messages and enable PTP.

```
[DeviceC] interface twenty-fivegige 1/0/1
[DeviceC-Twenty-FiveGigE1/0/1] ptp transport-protocol udp
[DeviceC-Twenty-FiveGigE1/0/1] ptp unicast-destination 11.10.10.2
[DeviceC-Twenty-FiveGigE1/0/1] ptp enable
[DeviceC-Twenty-FiveGigE1/0/1] quit
```

On Twenty-FiveGigE1/0/2, configure the destination IP address for unicast PTP messages and enable PTP. (The SMPTE ST 2059-2 PTP profile uses IPv4 UDP transport of PTP messages by default.)

```
[DeviceC] interface twenty-fivegige 1/0/2
[DeviceC-Twenty-FiveGigE1/0/2] ptp unicast-destination 12.10.10.1
[DeviceC-Twenty-FiveGigE1/0/2] ptp enable
[DeviceC-Twenty-FiveGigE1/0/2] quit
```

5. Configure the base station.

Specify PTP domain 0.

Specify IPv4 UDP transport of PTP messages.

Set the destination IP address of unicast PTP messages to 12.10.10.2.

Specify the Request_Response delay measurement mechanism.

For more information, see the configuration guide for the base station.

Verifying the configuration

When the network is stable, perform the following tasks to verify the PTP configuration:

- Use the **`display ptp clock`** command to display PTP clock information.
- Use the **`display ptp interface brief`** command to display brief PTP running information.

Display PTP clock information on Device A.

```
[DeviceA] display ptp clock
PTP global state    : Enabled
PTP profile         : SMPTE ST 2059-2
PTP mode            : BC
Slave only          : No
Lock status         : Locked
Clock ID            : 000FE2-FFFE-FF0000
Clock type          : ToD0
 ToD direction  : In
 ToD delay time : 1000 (ns)
Clock domain        : 0
Number of PTP ports : 1
Priority1     : 0
Priority2     : 128
Clock quality :
 Class                 : 6
 Accuracy              : 32
 Offset (log variance) : 65535
Offset from master : 0 (ns)
Mean path delay    : 0 (ns)
Steps removed      : 0
```

```
Local clock time   : Sun Jan 15 20:57:29 2019
```

Display brief PTP running information on Device A.

```
[DeviceA] display ptp interface brief
Name       InstID     State         Delay mechanism  Clock step  Asymmetry correction
WGE1/0/1   0          Master        E2E              Two         0
```

Display PTP clock information on Device C.

```
[DeviceC] display ptp clock
PTP global state    : Enabled
PTP profile         : SMPTE ST 2059-2
PTP mode            : BC
Slave only          : No
Lock status         : Locked
Clock ID            : 000FE2-FFFE-FF0001
Clock type          : Local
Clock domain        : 0
Number of PTP ports : 2
Priority1     : 128
Priority2     : 128
Clock quality :
 Class                 : 248
 Accuracy              : 254
 Offset (log variance) : 65535
Offset from master  : 106361246000 (ns)
Mean path delay     : 2780000 (ns)
Steps removed       : 2
Local clock time    : Sun Jan 15 20:57:29 2019
```

Display brief PTP running information on Device B.

```
[DeviceB] display ptp interface brief
Name       InstID     State         Delay mechanism  Clock step  Asymmetry correction
WGE1/0/1   0          Slave         E2E              Two         0
WGE1/0/2   0          Master        E2E              Two         0
```

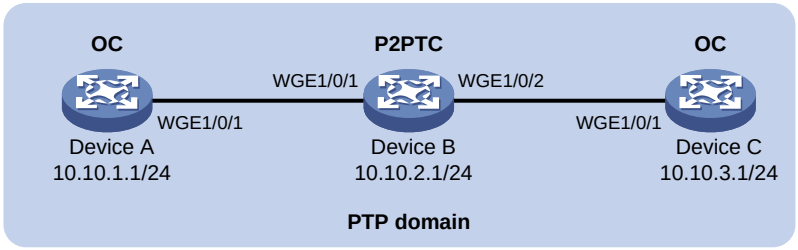
Example: Configuring PTP (AES67-2015, IPv4 UDP transport, multicast transmission)

Network configuration

As shown in Figure 55, configure PTP (AES67-2015, IPv4 UDP transport, multicast transmission) to enable time synchronization between Device A and Device C:

- Specify the AES67-2015 PTP profile and multicast IPv4 UDP transport of PTP messages for Device A, Device B, and Device C.
- Assign Device A, Device B, and Device C to the same PTP domain. Specify the OC clock node type for Device A and Device C, and the P2PTC clock node type for Device B. All clock nodes elect a GM through BMC.
- Use the peer delay measurement mechanism on all clock nodes in the PTP domain.

Figure 57 Network diagram



Procedure

IMPORTANT:

The AES67-2015 PTP profile transports PTP messages over IPv4 UDP rather than IEEE 802.3/Ethernet. The profile supports both multicast and unicast transmission modes.

1. Configure Device A:

   # Specify the AES67-2015 PTP profile.

   ```
   <DeviceA> system-view
   [DeviceA] ptp profile aes67-2015
   ```

   # Specify the OC clock node type.

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

   # Create a PTP domain.

   ```
   [DeviceA] ptp domain 0
   ```

   # Enable PTP globally.

   ```
   [DeviceB] ptp global enable
   ```

   # Configure the source IP address for multicast PTP messages transmitted over IPv4 UDP.

   ```
   [DeviceA] ptp source 10.10.1.1
   ```

   # Specify PTP for obtaining the time.

   ```
   [DeviceA] clock protocol ptp
   ```

   # On Twenty-FiveGigE 1/0/1, specify the peer delay measurement mechanism and enable PTP.

   ```
   [DeviceA] interface twenty-fivegige 1/0/1
   [DeviceA-Twenty-FiveGigE1/0/1] ptp transport-protocol udp
   [DeviceA-Twenty-FiveGigE1/0/1] ptp delay-mechanism p2p
   [DeviceA-Twenty-FiveGigE1/0/1] ptp enable
   [DeviceA-Twenty-FiveGigE1/0/1] quit
   ```

2. Configure Device B:

   # Specify the AES67-2015 PTP profile.

   ```
   <DeviceB> system-view
   [DeviceB] ptp profile aes67-2015
   ```

   # Specify the P2PTC clock node type.

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

   # Create a PTP domain.

   ```
   [DeviceB] ptp domain 0
   ```

   # Enable PTP globally.

   ```
   [DeviceB] ptp global enable
   ```

   # Configure the source IP address for multicast PTP messages transmitted over IPv4 UDP.

   ```
   [DeviceB] ptp source 10.10.2.1
   ```

Specify PTP for obtaining the time.

```
[DeviceB] clock protocol ptp
```

Enable PTP on Twenty-FiveGigE 1/0/1.

```
[DeviceB] interface twenty-fivegige 1/0/1
DeviceB-Twenty-FiveGigE1/0/1] ptp transport-protocol udp
[DeviceB-Twenty-FiveGigE1/0/1] ptp enable
[DeviceB-Twenty-FiveGigE1/0/1] quit
```

Enable PTP on Twenty-FiveGigE 1/0/2.

```
[DeviceB] interface twenty-fivegige 1/0/2
[DeviceB-Twenty-FiveGigE1/0/2] ptp transport-protocol udp
[DeviceB-Twenty-FiveGigE1/0/2] ptp enable
[DeviceB-Twenty-FiveGigE1/0/2] quit
```

3. Configure Device C:

Specify the AES67-2015 PTP profile.

```
<DeviceC> system-view
[DeviceC] ptp profile aes67-2015
```

Specify the OC clock node type.

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

Create a PTP domain.

```
[DeviceC] ptp domain 0
```

Enable PTP globally.

```
[DeviceC] ptp global enable
```

Configure the source IP address for multicast PTP messages transmitted over IPv4 UDP.

```
[DeviceC] ptp source 10.10.3.1
```

Specify PTP for obtaining the time.

```
[DeviceC] clock protocol ptp
```

On Twenty-FiveGigE 1/0/1, specify the peer delay measurement mechanism and enable PTP.

```
[DeviceC] interface twenty-fivegige 1/0/1
[DeviceC-Twenty-FiveGigE1/0/1] ptp delay-mechanism p2p
[DeviceC-Twenty-FiveGigE1/0/1] ptp enable
[DeviceC-Twenty-FiveGigE1/0/1] quit
```

Verifying the configuration

When the network is stable, perform the following tasks to verify the PTP configuration:

- Use the **display ptp clock** command to display PTP clock information.
- Use the **display ptp interface brief** command to display brief PTP running information for all PTP interfaces.

Display PTP clock information on Device A.

```
[DeviceA] display ptp clock
PTP global state    : Enabled
PTP profile         : AES67-2015
PTP mode            : OC
Slave only          : No
Lock status         : Unlocked
Clock ID            : 000FE2-FFFE-FF0000
Clock type          : Local
```

```
Clock domain        : 0
Number of PTP ports : 1
Priority1     : 128
Priority2     : 128
Clock quality :
 Class                : 248
 Accuracy             : 254
 Offset (log variance) : 65535
Offset from master  : 106368539000 (ns)
Mean path delay     : 2791000 (ns)
Steps removed       : 1
Local clock time    : Sun Jan 15 20:57:29 2019
```

Display brief PTP running information for all PTP interfaces on Device A.

```
[DeviceA] display ptp interface brief
Name       InstID     State         Delay mechanism  Clock step  Asymmetry correction
WGE1/0/1   0          Master        P2P              Two         0
```

Display PTP clock information on Device B.

```
[DeviceB] display ptp clock
PTP global state    : Enabled
PTP profile         : AES67-2015
PTP mode            : P2PTC
Slave only          : No
Lock status         : Unlocked
Clock ID            : 000FE2-FFFE-FF0001
Clock type          : Local
Clock domain        : 0
Number of PTP ports : 2
Priority1     : 128
Priority2     : 128
Clock quality :
 Class                : 248
 Accuracy             : 254
 Offset (log variance) : 65535
Offset from master : N/A
Mean path delay    : N/A
Steps removed      : N/A
Local clock time   : Sun Jan 15 20:57:29 2019
```

Display brief PTP running information for all PTP interfaces on Device B.

```
[DeviceB] display ptp interface brief
Name       InstID     State         Delay mechanism  Clock step  Asymmetry correction
WGE1/0/1   0          N/A           P2P              Two         0
WGE1/0/2   0          N/A           P2P              Two         0
```

The output shows that Device A is elected as the GM and Twenty-FiveGigE1/0/1 on Device A sends time synchronization information to its downstreams as a master port.

Configuring SNMP

About SNMP

Simple Network Management Protocol (SNMP) is used for a management station to access and operate the devices on a network, regardless of their vendors, physical characteristics, and interconnect technologies.

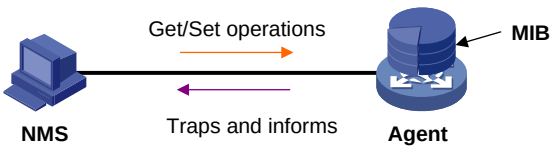
SNMP enables network administrators to read and set the variables on managed devices for state monitoring, troubleshooting, statistics collection, and other management purposes.

SNMP framework

The SNMP framework contains the following elements:

- **SNMP manager**—Works on an NMS to monitor and manage the SNMP-capable devices in the network. It can get and set values of MIB objects on an agent.
- **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.

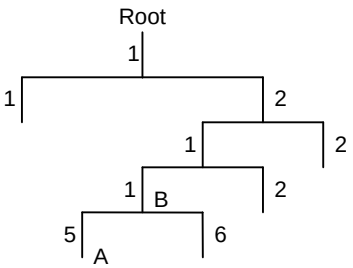
Figure 58 Relationship between NMS, agent, and MIB



MIB and view-based MIB access control

A MIB stores variables called "nodes" or "objects" in a tree hierarchy and identifies each node with a unique OID. An OID is a dotted numeric string that uniquely identifies the path from the root node to a leaf node. For example, object B in Figure 59 is uniquely identified by the OID {1.2.1.1}.

Figure 59 MIB tree



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

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

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

SNMP operations

SNMP provides the following basic operations:

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

Protocol versions

The device supports SNMPv1, SNMPv2c, and SNMPv3 in non-FIPS mode and supports only SNMPv3 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**—VACM mode controls access to MIB objects by assigning MIB views to SNMP communities or users.
- **Role based access control**—RBAC mode controls access to MIB objects by assigning user roles to SNMP communities or users.
    - SNMP communities or users with predefined user role network-admin or level-15 have read and write access to all MIB objects.
    - SNMP communities or users with predefined user role network-operator have read-only access to all MIB objects.
    - SNMP communities or users with a user-defined user role have access rights to MIB objects as specified by the **`rule`** command.

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 the RBAC mode.

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

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*.

SNMP tasks at a glance

To configure SNMP, perform the following tasks:

1. Enabling the SNMP agent
2. Enabling SNMP versions
3. Configuring SNMP basic parameters
   - (Optional.) Configuring SNMP common parameters
   - Configuring an SNMPv1 or SNMPv2c community
   - Configuring an SNMPv3 group and user
4. (Optional.) Configuring SNMP notifications
5. (Optional.) Configuring SNMP logging

Enabling the SNMP agent

Restrictions and guidelines

The SNMP agent is enabled when you use any command that begins with **`snmp-agent`** except for the **`snmp-agent calculate-password`** command.

The SNMP agent will fail to be enabled when the port that the agent will listen on is used by another service. You can use the **`snmp-agent port`** command to specify a listening port. To view the UDP port use information, execute the **`display udp verbose`** command. For more information about the **`display udp verbose`** command, see IP performance optimization commands in *Layer 3—IP Services Configuration Guide*.

If you disable the SNMP agent, the SNMP settings do not take effect. The **`display current-configuration`** command does not display the SNMP settings and the SNMP settings will not be saved in the configuration file. For the SNMP settings to take effect, enable the SNMP agent.

Procedure

1. Enter system view.

   `system-view`
2. Enable the SNMP agent.

   `snmp-agent`

   By default, the SNMP agent is disabled.

Enabling SNMP versions

Restrictions and guidelines

The device supports SNMPv1, SNMPv2c, and SNMPv3 in non-FIPS mode and supports only SNMPv3 in FIPS mode. An NMS and an SNMP agent must use the same SNMP version to communicate with each other.

The community name and data carried in SNMPv1 and SNMPv2 messages are in plaintext form. For security, use SNMPv3 as a best practice.

To use SNMP notifications in IPv6, enable SNMPv2c or SNMPv3.

Procedure

1. Enter system view.

system-view

2. Enable SNMP versions.

   In non-FIPS mode:

   snmp-agent sys-info version { **all** | { **v1** | **v2c** | **v3** } * }

   In FIPS mode:

   snmp-agent sys-info version { **all** | **v3** }

   By default, SNMPv3 is enabled.

   If you execute the command multiple times with different options, all the configurations take effect, but only one SNMP version is used by the agent and NMS for communication.

Configuring SNMP common parameters

Restrictions and guidelines

An SNMP engine ID uniquely identifies a device in an SNMP managed network. Make sure the local SNMP engine ID is unique within your SNMP managed network to avoid communication problems. By default, the device is assigned a unique SNMP engine ID.

If you have configured SNMPv3 users, change the local SNMP engine ID only when necessary. The change can void the SNMPv3 usernames and encrypted keys you have configured.

The SNMP agent will fail to be enabled when the port that the agent will listen on is used by another service. You can use the **snmp-agent port** command to change the SNMP listening port. As a best practice, execute the **display udp verbose** command to view the UDP port use information before specifying a new SNMP listening port. For more information about the **display udp verbose** command, see IP performance optimization commands in *Layer 3—IP Services Configuration Guide*.

Procedure

1. Enter system view.

   system-view

2. Specify an SNMP listening port.

   snmp-agent port *port-number*

   By default, the SNMP listening port is UDP port 161.

3. Set a local SNMP engine ID.

   snmp-agent local-engineid *engineid*

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

4. Set an engine ID for a remote SNMP entity.

   snmp-agent remote { *ipv4-address* | **ipv6** *ipv6-address* } [**vpn-instance** *vpn-instance-name*] **engineid** *engineid*

   By default, no remote entity engine IDs exist.

   This step is required for the device to send SNMPv3 notifications to a host, typically NMS.

5. Create or update a MIB view.

   snmp-agent mib-view { **excluded** | **included** } *view-name oid-tree* [**mask** *mask-value*]

   By default, the MIB view **ViewDefault** is predefined. In this view, all the MIB objects in the **iso** subtree but the **snmpUsmMIB**, **snmpVacmMIB**, and **snmpModules.18** subtrees are accessible.

   Each *view-name oid-tree* pair represents a view record. If you specify the same record with different MIB sub-tree masks multiple times, the most recent configuration takes effect.

6. Configure the system management information.
   - Configure the system contact.
     snmp-agent sys-info contact *sys-contact*
     By default, no system contact is **configured**.
   - Configure the system location.
     snmp-agent sys-info location *sys-location*
     By default, no system location is **configured**.
7. Create an SNMP context.
   snmp-agent context *context-name*
   By default, no SNMP contexts exist.
8. Configure the maximum SNMP packet size (in bytes) that the SNMP agent can handle.
   snmp-agent packet max-size *byte-count*
   By default, an SNMP agent can process SNMP packets with a maximum size of 1500 bytes.
9. Set the DSCP value for SNMP responses.
   snmp-agent packet response dscp *dscp-value*
   By default, the DSCP value for SNMP responses is 0.

Configuring an SNMPv1 or SNMPv2c community

About configuring an SNMPv1 or SNMPv2c community

You can create an SNMPv1 or SNMPv2c community by using a community name or by creating an SNMPv1 or SNMPv2c user. After you create an SNMPv1 or SNMPv2c user, the system automatically creates a community by using the username as the community name.

Restrictions and guidelines for configuring an SNMPv1 or SNMPv2c community

SNMPv1 and SNMPv2c settings are not supported in FIPS mode.

Make sure the NMS and agent use the same SNMP community name.

Only users with the network-admin or level-15 user role can create SNMPv1 or SNMPv2c communities, users, or groups. Users with other user roles cannot create SNMPv1 or SNMPv2c communities, users, or groups even if these roles are granted access to related commands or commands of the SNMPv1 or SNMPv2c feature.

Configuring an SNMPv1/v2c community by a community name

1. Enter system view.
   system-view
2. Create an SNMPv1/v2c community. Choose one option as needed.
   - In VACM mode:
     snmp-agent community { **read** | **write** } [**simple** | **cipher**] *community-name* [**mib-view** *view-name*] [**acl** { *ipv4-acl-number* | **name**

ipv4-acl-name } | **acl ipv6** { *ipv6-acl-number* | **name** *ipv6-acl-name* }] *

- In RBAC mode:

  snmp-agent community [**simple** | **cipher**] *community-name* **user-role** *role-name* [**acl** { *ipv4-acl-number* | **name** *ipv4-acl-name* } | **acl ipv6** { *ipv6-acl-number* | **name** *ipv6-acl-name* }] *

3. (Optional.) Map the SNMP community name to an SNMP context.

   snmp-agent community-map *community-name* **context** *context-name*

Configuring an SNMPv1/v2c community by creating an SNMPv1/v2c user

1. Enter system view.

   system-view

2. Create an SNMPv1/v2c group.

   snmp-agent group { **v1** | **v2c** } *group-name* [**notify-view** *view-name* | **read-view** *view-name* | **write-view** *view-name*] * [**acl** { *ipv4-acl-number* | **name** *ipv4-acl-name* } | **acl ipv6** { *ipv6-acl-number* | **name** *ipv6-acl-name* }] *

3. Add an SNMPv1/v2c user to the group.

   snmp-agent usm-user { **v1** | **v2c** } *user-name group-name* [**acl** { *ipv4-acl-number* | **name** *ipv4-acl-name* } | **acl ipv6** { *ipv6-acl-number* | **name** *ipv6-acl-name* }] *

   The system automatically creates an SNMP community by using the username as the community name.

4. (Optional.) Map the SNMP community name to an SNMP context.

   snmp-agent community-map *community-name* **context** *context-name*

Configuring an SNMPv3 group and user

Restrictions and guidelines for configuring an SNMPv3 group and user

Only users with the network-admin or level-15 user role can create SNMPv3 users or groups.Users with other user roles cannot create SNMPv3 users or groups even if these roles are granted access to related commands or commands of the SNMPv3 feature.

SNMPv3 users are managed in groups. All SNMPv3 users in a group share the same security model, but can use different authentication and encryption algorithms and keys. Table 7 describes the basic configuration requirements for different security models.

Table 7 Basic configuration requirements for different security models

Security model	Keyword for the group	Parameters for the user	Remarks
Authentication with privacy	**privacy**	Authentication and encryption algorithms and keys	For an NMS to access the agent, make sure the NMS and agent use the same authentication and encryption keys.

Security model	Keyword for the group	Parameters for the user	Remarks
Authentication without privacy	**authentication**	Authentication algorithm and key	For an NMS to access the agent, make sure the NMS and agent use the same authentication key.
No authentication, no privacy	N/A	N/A	The authentication and encryption keys, if configured, do not take effect.

Configuring an SNMPv3 group and user in non-FIPS mode

1. Enter system view.

   system-view

2. Create an SNMPv3 group.

   snmp-agent group v3 *group-name* [**authentication** | **privacy**] [**notify-view** *view-name* | **read-view** *view-name* | **write-view** *view-name*] * [**acl** { *ipv4-acl-number* | **name** *ipv4-acl-name* } | **acl ipv6** { *ipv6-acl-number* | **name** *ipv6-acl-name* }] *

3. (Optional.) Calculate the encrypted form for the key in plaintext form.

   snmp-agent calculate-password *plain-password* **mode** { **3desmd5** | **3dessha** | **aes192md5** | **aes192sha** | **aes256md5** | **aes256sha** | **md5** | **sha** } { **local-engineid** | **specified-engineid** *engineid* }

4. Create an SNMPv3 user. Choose one option as needed.

   - In VACM mode:

     snmp-agent usm-user v3 *user-name group-name* [**remote** { *ipv4-address* | **ipv6** *ipv6-address* } [**vpn-instance** *vpn-instance-name*]] [{ **cipher** | **simple** } **authentication-mode** { **md5** | **sha** } *auth-password* [**privacy-mode** { **3des** | **aes128** | **aes192** | **aes256** | **des56** } *priv-password*]] [**acl** { *ipv4-acl-number* | **name** *ipv4-acl-name* } | **acl ipv6** { *ipv6-acl-number* | **name** *ipv6-acl-name* }] *

   - In RBAC mode:

     snmp-agent usm-user v3 *user-name* **user-role** *role-name* [**remote** { *ipv4-address* | **ipv6** *ipv6-address* } [**vpn-instance** *vpn-instance-name*]] [{ **cipher** | **simple** } **authentication-mode** { **md5** | **sha** } *auth-password* [**privacy-mode** { **3des** | **aes128** | **aes192** | **aes256** | **des56** } *priv-password*]] [**acl** { *ipv4-acl-number* | **name** *ipv4-acl-name* } | **acl ipv6** { *ipv6-acl-number* | **name** *ipv6-acl-name* }] *

   To send notifications to an SNMPv3 NMS, you must specify the **remote** keyword.

5. (Optional.) Assign a user role to the SNMPv3 user created in RBAC mode.

   snmp-agent usm-user v3 *user-name* **user-role** *role-name*

   By default, an SNMPv3 user has the user role assigned to it at its creation.

Configuring an SNMPv3 group and user in FIPS mode

1. Enter system view.

   system-view

2. Create an SNMPv3 group.

   snmp-agent group v3 *group-name* { **authentication** | **privacy** } [**notify-view** *view-name* | **read-view** *view-name* | **write-view** *view-name*] * [**acl** { *ipv4-acl-number* | **name** *ipv4-acl-name* } | **acl ipv6** { *ipv6-acl-number* | **name** *ipv6-acl-name* }] *

3. (Optional.) Calculate the encrypted form for the key in plaintext form.

   snmp-agent calculate-password *plain-password* **mode** { **aes192sha** | **aes256sha** | **sha** } { **local-engineid** | **specified-engineid** *engineid* }

4. Create an SNMPv3 user. Choose one option as needed.

   - In VACM mode:

     snmp-agent usm-user v3 *user-name group-name* [**remote** { *ipv4-address* | **ipv6** *ipv6-address* } [**vpn-instance** *vpn-instance-name*]] { **cipher** | **simple** } **authentication-mode sha** *auth-password* [**privacy-mode** { **aes128** | **aes192** | **aes256** } *priv-password*] [**acl** { *ipv4-acl-number* | **name** *ipv4-acl-name* } | **acl ipv6** { *ipv6-acl-number* | **name** *ipv6-acl-name* }] *

   - In RBAC mode:

     snmp-agent usm-user v3 *user-name* **user-role** *role-name* [**remote** { *ipv4-address* | **ipv6** *ipv6-address* } [**vpn-instance** *vpn-instance-name*]] { **cipher** | **simple** } **authentication-mode sha** *auth-password* [**privacy-mode** { **aes128** | **aes192** | **aes256** } *priv-password*] [**acl** { *ipv4-acl-number* | **name** *ipv4-acl-name* } | **acl ipv6** { *ipv6-acl-number* | **name** *ipv6-acl-name* }] *

   To send notifications to an SNMPv3 NMS, you must specify the **remote** keyword.

5. (Optional.) Assign a user role to the SNMPv3 user created in RBAC mode.

   snmp-agent usm-user v3 *user-name* **user-role** *role-name*

   By default, an SNMPv3 user has the user role assigned to it at its creation.

Configuring SNMP notifications

About 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. After you enable notifications for a module, the module sends the generated notifications to the SNMP agent. The SNMP agent sends the received notifications as traps or informs based on the current configuration. Unless otherwise stated, the **trap** keyword in the command line includes both traps and informs.

Enabling SNMP notifications

Restrictions and guidelines

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 use SNMP notifications in IPv6, enable SNMPv2c or SNMPv3.

Procedure

1. Enter system view.

   system-view

2. Enable SNMP notifications.

   snmp-agent trap enable [**configuration** | *protocol* | **standard** [**authentication** | **coldstart** | **linkdown** | **linkup** | **warmstart**] * | **system**]

   By default, SNMP configuration notifications, standard notifications, and system notifications are enabled. Whether other SNMP notifications are enabled varies by modules.

   For the device to send SNMP notifications for a protocol, first enable the protocol.

3. Enter interface view.

   interface *interface-type interface-number*

4. Enable link state notifications.

   enable snmp trap updown

   By default, link state notifications are enabled.

Configuring parameters for sending SNMP notifications

About this task

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

When network congestion occurs or the destination is not reachable, the SNMP agent buffers notifications in a queue. You can set the queue size and the notification lifetime (the maximum time that a notification can stay in the queue). When the queue size is reached, the system discards the new notification it receives. If modification of the queue size causes the number of notifications in the queue to exceed the queue size, the oldest notifications are dropped for new notifications. A notification is deleted when its lifetime expires.

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

Configuring the parameters for sending SNMP traps

1. Enter system view.

   system-view

2. Configure a target host.

   In non-FIPS mode:

   snmp-agent target-host trap address udp-domain { *ipv4-target-host* | **ipv6** *ipv6-target-host* } [**udp-port** *port-number*] [**dscp** *dscp-value*] [**vpn-instance** *vpn-instance-name*] **params securityname** *security-string* [**v1** | **v2c** | **v3** [**authentication** | **privacy**]]

   In FIPS mode:

   snmp-agent target-host trap address udp-domain { *ipv4-target-host* | **ipv6** *ipv6-target-host* } [**udp-port** *port-number*] [**dscp** *dscp-value*] [**vpn-instance** *vpn-instance-name*] **params securityname** *security-string* **v3** { **authentication** | **privacy** }

By default, no target host is configured.

3. (Optional.) Configure a source address for sending traps.

   snmp-agent trap source *interface-type* { *interface-number* | *interface-number.subnumber* }

   By default, SNMP uses the IP address of the outgoing routed interface as the source IP address.

4. (Optional.) Enable SNMP alive traps and set the sending interval.

   snmp-agent trap periodical-interval *interval*

   By default, SNMP alive traps is enabled and the sending interval is 60 seconds.

Configuring the parameters for sending SNMP informs

1. Enter system view.

   system-view

2. Configure a target host.

   In non-FIPS mode:

   snmp-agent target-host inform address udp-domain { *ipv4-target-host* | **ipv6** *ipv6-target-host* } [**udp-port** *port-number*] [**vpn-instance** *vpn-instance-name*] **params** { **cipher-securityname** *cipher-security-string* **v2c** | **securityname** *security-string* { **v2c** | **v3** [**authentication** | **privacy**] } }

   In FIPS mode:

   snmp-agent target-host inform address udp-domain { *ipv4-target-host* | **ipv6** *ipv6-target-host* } [**udp-port** *port-number*] [**vpn-instance** *vpn-instance-name*] **params securityname** *security-string* **v3** { **authentication** | **privacy** }

   By default, no target host is configured.

   Only SNMPv2c and SNMPv3 support inform packets.

3. (Optional.) Configure a source address for sending informs.

   snmp-agent inform source *interface-type* { *interface-number* | *interface-number.subnumber* }

   By default, SNMP uses the IP address of the outgoing routed interface as the source IP address.

Configuring common parameters for sending notifications

1. Enter system view.

   system-view

2. (Optional.) Enable extended linkUp/linkDown notifications.

   snmp-agent trap if-mib link extended

   By default, the SNMP agent sends standard linkUp/linkDown notifications.

   If the NMS does not support extended linkUp/linkDown notifications, do not use this command.

3. (Optional.) Set the notification queue size.

   snmp-agent trap queue-size *size*

   By default, the notification queue can hold 100 notification messages.

4. (Optional.) Set the notification lifetime.

   snmp-agent trap life *seconds*

   The default notification lifetime is 120 seconds.

Examining the system configuration for changes

About this task

The SNMP module examines the system running configuration, startup configuration, and next-startup configuration file for changes periodically and generates a log if any change is found. If SNMP notifications for configuration changes has been enabled, the system generates also an SNMP notification.

Procedure

1. Enter system view.

   system-view

2. Set the interval at which the SNMP module examines the system configuration for changes.

   snmp-agent configuration-examine interval *interval*

   By default, the SNMP module examines the system configuration for changes at intervals of 600 seconds.

3. Enable SNMP notifications for system configuration changes.

   snmp-agent trap enable configuration

   By default, SNMP notifications is enabled for system configuration changes.

Configuring SNMP logging

About this task

The SNMP agent logs Get requests, Set requests, Set responses, SNMP notifications, and SNMP authentication failures, 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.
- **SNMP authentication failure**—The agent logs related information when an NMS fails to be authenticated by the agent.

The SNMP module sends these logs to the information center. You can configure the information center to output these messages to certain destinations, such as the console and the log buffer. The total output size for the node field (MIB node name) and the value field (value of the MIB node) in each log entry is 1024 bytes. If this limit is exceeded, the information center truncates the data in the fields. For more information about the information center, see "Configuring the information center."

Restrictions and guidelines

Enable SNMP logging only if necessary. SNMP logging is memory-intensive and might impact device performance.

Procedure

1. Enter system view.

   system-view

2. Enable SNMP logging.

   snmp-agent log { **all** | **authfail** | **get-operation** | **set-operation** }

   By default, SNMP logging is enabled for set operations and disabled for SNMP authentication failures and get operations.

3. Enable SNMP notification logging.

   snmp-agent trap log

   By default, SNMP notification logging is disabled.

Display and maintenance commands for SNMP

Execute **display** commands in any view.

Task	Command
Display SNMPv1 or SNMPv2c community information. (This command is not supported in FIPS mode.)	**display snmp-agent community** [**read** | **write**]
Display SNMP contexts.	**display snmp-agent context** [*context-name*]
Display SNMP group information.	**display snmp-agent group** [*group-name*]
Display the local engine ID.	**display snmp-agent local-engineid**
Display SNMP MIB node information.	**display snmp-agent mib-node** [**details** | **index-node** | **trap-node** | **verbose**]
Display MIB view information.	**display snmp-agent mib-view** [**exclude** | **include** | **viewname** *view-name*]
Display remote engine IDs.	**display snmp-agent remote** [{ *ipv4-address* | **ipv6** *ipv6-address* } [**vpn-instance** *vpn-instance-name*]]
Display SNMP agent statistics.	**display snmp-agent statistics**
Display SNMP agent system information.	**display snmp-agent sys-info** [**contact** | **location** | **version**] *
Display basic information about the notification queue.	**display snmp-agent trap queue**
Display SNMP notifications enabling status for modules.	**display snmp-agent trap-list**
Display SNMPv3 user information.	**display snmp-agent usm-user** [**engineid** *engineid* | **username** *user-name* | **group** *group-name*] *

SNMP configuration examples

Example: Configuring SNMPv1/SNMPv2c

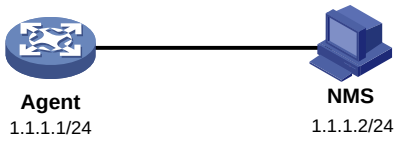
The device does not support this configuration example in FIPS mode.

The configuration procedure is the same for SNMPv1 and SNMPv2c. This example uses SNMPv1.

Network configuration

As shown in Figure 60, 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 60 Network diagram



Procedure

1. Configure the SNMP agent:

   # Assign IP address **1.1.1.1/24** to the agent and make sure the agent and the NMS can reach each other. (Details not shown.)

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

Verifying the configuration

Try to get the MTU value of the 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
```

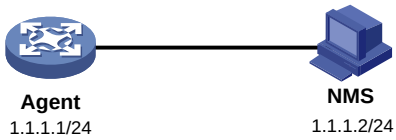
Example: Configuring SNMPv3

Network configuration

As shown in Figure 61, the NMS (1.1.1.2/24) uses SNMPv3 to monitor and manage the agent (1.1.1.1/24). The agent automatically sends notifications to report events to the NMS. The default UDP port 162 is used for SNMP notifications.

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 encryption key **123456TESTencr&!**.

Figure 61 Network diagram



Configuring SNMPv3 in RBAC mode

1. Configure the agent:

   # Assign IP address **1.1.1.1/24** to the agent and make sure the agent and the NMS can reach each other. (Details not shown.)

   #Create user role **test**, and assign **test** read-only access to the objects under the **snmpMIB** node (OID:1.3.6.1.6.3.1), including the **linkUp** and **linkDown** objects.

   ```
   <Agent> system-view
   [Agent] role name test
   [Agent-role-test] rule 1 permit read oid 1.3.6.1.6.3.1
   ```

   # Assign user role **test** read-only access to the **system** node (OID:1.3.6.1.2.1.1) and read-write access to the **interfaces** node(OID:1.3.6.1.2.1.2).

   ```
   [Agent-role-test] rule 2 permit read oid 1.3.6.1.2.1.1
   [Agent-role-test] rule 3 permit read write oid 1.3.6.1.2.1.2
   [Agent-role-test] quit
   ```

   # Create SNMPv3 user **RBACtest**. Assign user role **test** to **RBACtest.** Set the authentication algorithm to **SHA-1**, authentication key to **123456TESTauth&!**, encryption algorithm to **AES**, and encryption key to **123456TESTencr&!**.

   ```
   [Agent] snmp-agent usm-user v3 RBACtest 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 on the agent. Specify the NMS at 1.1.1.2 as the notification destination, and **RBACtest** as the username.

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

2. Configure the NMS:
   - Specify SNMPv3.
   - Create SNMPv3 user **RBACtest**.
   - Enable authentication and encryption. Set the authentication algorithm to **SHA-1**, authentication key to **123456TESTauth&!**, encryption algorithm to **AES**, and encryption key to **123456TESTencr&!**.
   - Set the timeout timer 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.

Configuring SNMPv3 in VACM mode

1. Configure the agent:

   # Assign IP address 1.1.1.1/24 to the agent, and make sure the agent and the NMS can reach each other. (Details not shown.)

   # Create SNMPv3 group **managev3group** and assign **managev3group** read-only access to the objects under the **snmpMIB** node (OID: 1.3.6.1.2.1.2.2) in the **test** view, including the **linkUp** and **linkDown** objects.

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

   #Assign SNMPv3 group **managev3group** read-write access to the objects under the **system** node (OID: 1.3.6.1.2.1.1) and **interfaces** node (OID:1.3.6.1.2.1.2) in the **test** view.

   ```
   [Agent] snmp-agent mib-view included test 1.3.6.1.2.1.1
   [Agent] snmp-agent mib-view included test 1.3.6.1.2.1.2
   [Agent] snmp-agent group v3 managev3group privacy read-view test write-view test
   ```

   # Add user **VACMtest** to SNMPv3 group **managev3group**, and set the authentication algorithm to **SHA-1**, authentication key to **123456TESTauth&!**, encryption algorithm to **AES**, and encryption key to **123456TESTencr&!**.

   ```
   [Agent] snmp-agent usm-user v3 VACMtest 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 on the agent. Specify the NMS at 1.1.1.2 as the trap destination, and **VACMtest** as the username.

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

2. Configure the SNMP NMS:
   - Specify SNMPv3.

- Create SNMPv3 user VACMtest.
- Enable authentication and encryption. Set the authentication algorithm to **SHA-1**, authentication key to **123456TESTauth&!**, encryption algorithm to **AES**, and encryption key to **123456TESTencr&!**.
- Set the timeout timer 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.

Verifying the configuration

- Use username **RBACtest** to access the agent.

  # Retrieve the value of the **sysName** node. The value **Agent** is returned.

  # Set the value for the **sysName** node to **Sysname**. The operation fails because the NMS does not have write access to the node.

  # Shut down or bring up an interface on the agent. The NMS receives linkUP (OID: 1.3.6.1.6.3.1.1.5.4) or linkDown (OID: 1.3.6.1.6.3.1.1.5.3) notifications.
- Use username **VACMtest** to access the agent.

  # Retrieve the value of the **sysName** node. The value **Agent** is returned.

  # Set the value for the **sysName** node to **Sysname**. The operation succeeds.

  # Shut down or bring up an interface on the agent. The NMS receives **linkUP** (OID: 1.3.6.1.6.3.1.1.5.4) or **linkDown** (OID: 1.3.6.1.6.3.1.1.5.3) notifications.

Configuring RMON

About RMON

Remote Network Monitoring (RMON) is an SNMP-based network management protocol. It enables proactive remote monitoring and management of network devices.

RMON working mechanism

RMON can periodically or continuously collect traffic statistics for an Ethernet port and monitor the values of MIB objects on a device. When a value reaches the threshold, the device automatically logs the event or sends a notification to the NMS. 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. SNMP reports function and interface operating status changes such as link up, link down, and module failure to the NMS.

RMON groups

Among standard RMON groups, the device implements the statistics group, history group, event group, alarm group, probe configuration group, and user history group. The Comware system 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:

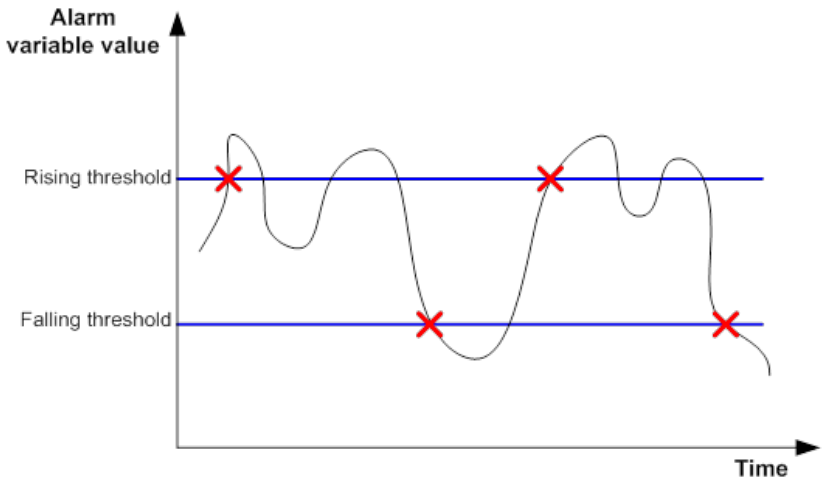
- **Log**—Logs event information (including event time and description) in the event log table so the management device can get the logs through SNMP.
- **Trap**—Sends an SNMP notification when the event occurs.
- **Log-Trap**—Logs event information in the event log table and sends an SNMP notification 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 62.

Figure 62 Rising and falling alarm events



Private alarm group

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

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

1. Samples the private alarm variables in the user-defined formula.
2. Processes the sampled values with the formula.
3. Compares the calculation result with the predefined thresholds, and then takes one of the following actions:
   - Triggers the event associated with the rising alarm event if the result is equal to or greater than the rising threshold.
   - Triggers the event associated with the falling alarm event if the result is equal to or less than the falling threshold.

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

crosses the rising threshold multiple times before it crosses the falling threshold, only the first crossing triggers a rising alarm event.

Sample types for the alarm group and the private alarm group

The RMON agent supports the following sample types:

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

Protocols and standards

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

Configuring the RMON statistics function

About 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.

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.

Creating an RMON Ethernet statistics entry

Restrictions and guidelines

The index of an RMON statistics entry must be globally unique. If the index has been used by another interface, the creation operation fails.

You can create only one RMON statistics entry for an Ethernet interface.

Procedure

1. Enter system view.
   system-view
2. Enter Ethernet interface view.
   interface *interface-type interface-number*
3. Create an RMON Ethernet statistics entry.
   rmon statistics *entry-number* [**owner** *text*]

Creating an RMON history control entry

Restrictions and guidelines

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.

Procedure

1. Enter system view.

   system-view
2. Enter Ethernet interface view.

   interface *interface-type interface-number*
3. Create an RMON history control entry.

   rmon history *entry-number* **buckets** *number* **interval** *interval* [**owner** *text*]

   By default, no RMON history control entries exist.

   You can create multiple RMON history control entries for an Ethernet interface.

Configuring the RMON alarm function

Restrictions and guidelines

When you create a new event, alarm, or private alarm entry, follow these restrictions and guidelines:

- The entry must not have the same set of parameters as an existing entry.
- The maximum number of entries is not reached.

Table 8 shows the parameters to be compared for duplication and the entry limits.

Table 8 RMON configuration restrictions

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

Prerequisites

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.

Procedure

1. Enter system view.

   system-view

2. (Optional.) Create an RMON event entry.

   rmon event *entry-number* [**description** *string*] { **log** | **log-trap** *security-string* | **none** | **trap** *security-string* } [**owner** *text*]

   By default, no RMON event entries exist.

3. Create an RMON alarm entry.

   - Create an RMON alarm entry.

     rmon alarm *entry-number alarm-variable sampling-interval* { **absolute** | **delta** } [**startup-alarm** { **falling** | **rising** | **rising-falling** }] **rising-threshold** *threshold-value1 event-entry1* **falling-threshold** *threshold-value2 event-entry2* [**owner** *text*]

   - Create an RMON private alarm entry.

     rmon prialarm *entry-number prialarm-formula prialarm-des sampling-interval* { **absolute** | **delta** } [**startup-alarm** { **falling** | **rising** | **rising-falling** }] **rising-threshold** *threshold-value1 event-entry1* **falling-threshold** *threshold-value2 event-entry2* **entrytype** { **forever** | **cycle** *cycle-period* } [**owner** *text*]

   By default, no RMON alarm entries or RMON private alarm entries exist.

   You can associate an alarm with an event that has not been created yet. The alarm will trigger the event only after the event is created.

Display and maintenance commands for RMON

Execute **display** commands in any view.

Task	Command
Display RMON alarm entries.	**display rmon alarm** [*entry-number*]
Display RMON event entries.	**display rmon event** [*entry-number*]
Display log information for event entries.	**display rmon eventlog** [*entry-number*]
Display RMON history control entries and history samples.	**display rmon history** [*interface-type interface-number*]
Display RMON private alarm entries.	**display rmon prialarm** [*entry-number*]
Display RMON statistics.	**display rmon statistics** [*interface-type interface-number*]

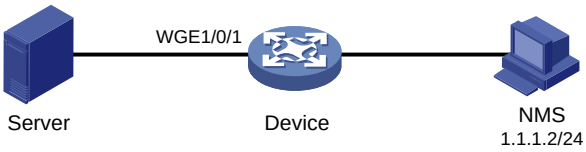
RMON configuration examples

Example: Configuring the Ethernet statistics function

Network configuration

As shown in Figure 63, create an RMON Ethernet statistics entry on the device to gather cumulative traffic statistics for Twenty-FiveGigE 1/0/1.

Figure 63 Network diagram



Procedure

Create an RMON Ethernet statistics entry for Twenty-FiveGigE 1/0/1.

```
<Sysname> system-view
[Sysname] interface twenty-fivegige 1/0/1
[Sysname-Twenty-FiveGigE1/0/1] rmon statistics 1 owner user1
```

Verifying the configuration

Display statistics collected for Twenty-FiveGigE 1/0/1.

```
<Sysname> display rmon statistics twenty-fivegige 1/0/1
EtherStatsEntry 1 owned by user1 is VALID.
  Interface : Twenty-FiveGigE1/0/1<ifIndex.3>
  etherStatsOctets         : 21657     , etherStatsPkts          : 307
  etherStatsBroadcastPkts  : 56        , etherStatsMulticastPkts : 34
  etherStatsUndersizePkts  : 0         , etherStatsOversizePkts  : 0
  etherStatsFragments      : 0         , etherStatsJabbers       : 0
  etherStatsCRCAlignErrors : 0         , etherStatsCollisions    : 0
  etherStatsDropEvents (insufficient resources): 0
  Incoming packets by size:
  64     : 235       ,  65-127  : 67        ,  128-255  : 4
  256-511: 1         ,  512-1023: 0         ,  1024-1518: 0
```

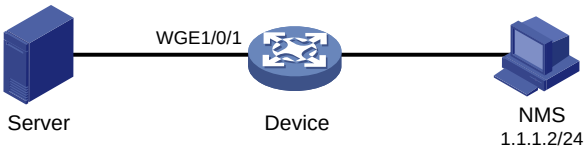
Get the traffic statistics from the NMS through SNMP. (Details not shown.)

Example: Configuring the history statistics function

Network configuration

As shown in Figure 64, create an RMON history control entry on the device to sample traffic statistics for Twenty-FiveGigE 1/0/1 every minute.

Figure 64 Network diagram



Procedure

Create an RMON history control entry to sample traffic statistics every minute for Twenty-FiveGigE 1/0/1. Retain a maximum of eight samples for the interface in the history statistics table.

```
<Sysname> system-view
[Sysname] interface twenty-fivegige 1/0/1
[Sysname-Twenty-FiveGigE1/0/1] rmon history 1 buckets 8 interval 60 owner user1
```

Verifying the configuration

Display the history statistics collected for Twenty-FiveGigE 1/0/1.

```
[Sysname-Twenty-FiveGigE1/0/1] display rmon history
HistoryControlEntry 1 owned by user1 is VALID
  Sampled interface     : Twenty-FiveGigE1/0/1<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
```

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

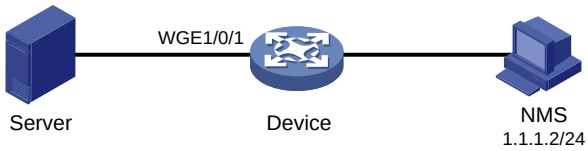
Example: Configuring the alarm function

Network configuration

As shown in Figure 65, configure the device to monitor the incoming traffic statistic on Twenty-FiveGigE 1/0/1, and send RMON alarms when either of the following conditions is met:

- 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 65 Network diagram



Procedure

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

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

Create an RMON Ethernet statistics entry for Twenty-FiveGigE 1/0/1.

```
[Sysname] interface twenty-fivegige 1/0/1
[Sysname-Twenty-FiveGigE1/0/1] rmon statistics 1 owner user1
[Sysname-Twenty-FiveGigE1/0/1] quit
```

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

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

NOTE:

The string 1.3.6.1.2.1.16.1.1.1.4.1 is the object instance for Twenty-FiveGigE 1/0/1. The digits before the last digit (1.3.6.1.2.1.16.1.1.1.4) represent the object for total incoming traffic statistics. The last digit (1) is the RMON Ethernet statistics entry index for Twenty-FiveGigE 1/0/1.

Verifying the configuration

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 Twenty-FiveGigE 1/0/1.

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

```
Interface : Twenty-FiveGigE1/0/1<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
```

The NMS receives the notification when the alarm is triggered.

Configuring the Event MIB

About the Event MIB

The Event Management Information Base (Event MIB) is an SNMPv3-based network management protocol and is an enhancement to remote network monitoring (RMON). The Event MIB uses Boolean tests, existence tests, and threshold tests to monitor MIB objects on a local or remote system. It triggers the predefined notification or set action when a monitored object meets the trigger condition.

Trigger

The Event MIB uses triggers to manage and associate the three elements of the Event MIB: monitored object, trigger condition, and action.

Monitored objects

The Event MIB can monitor the following MIB objects:

- Table node.
- Conceptual row node.
- Table column node.
- Simple leaf node.
- Parent node of a leaf node.

To monitor a single MIB object, specify it by its OID or name. To monitor a set of MIB objects, specify the common OID or name of the group and enable wildcard matching. For example, specify ifDescr.2 to monitor the description for the interface with index 2. Specify ifDescr and enable wildcard matching to monitor the descriptions for all interfaces.

Trigger test

A trigger supports Boolean, existence, and threshold tests.

Boolean test

A Boolean test compares the value of the monitored object with the reference value and takes actions according to the comparison result. The comparison types include **unequal**, **equal**, **less**, **lessorequal**, **greater**, and **greaterorequal**. For example, if the comparison type is **equal**, an event is triggered when the value of the monitored object equals the reference value. The event will not be triggered again until the value becomes unequal and comes back to equal.

Existence test

An existence test monitors and manages the absence, presence, and change of a MIB object, for example, interface status. When a monitored object is specified, the system reads the value of the monitored object regularly.

- If the test type is **Absent**, the system triggers an alarm event and takes the specified action when the state of the monitored object changes to absent.
- If the test type is **Present**, the system triggers an alarm event and takes the specified action when the state of the monitored object changes to present.
- If the test type is **Changed**, the system triggers an alarm event and takes the specified action when the value of the monitored object changes.

Threshold test

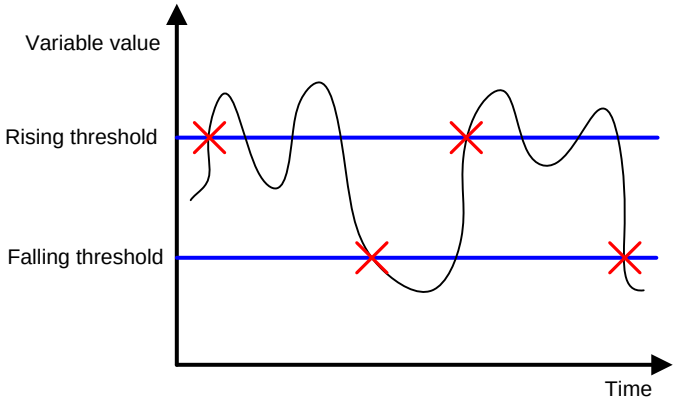
A threshold test regularly compares the value of the monitored object with the threshold values.

- A rising alarm event is triggered if the value of the monitored object is greater than or equal to the rising threshold.
- A falling alarm event is triggered if the value of the monitored object is smaller than or equal to the falling threshold.
- A rising alarm event is triggered if the difference between the current sampled value and the previous sampled value is greater than or equal to the delta rising threshold.
- A falling alarm event is triggered if the difference between the current sampled value and the previous sampled value is smaller than or equal to the delta falling threshold.
- A falling alarm event is triggered if the values of the monitored object, the rising threshold, and the falling threshold are the same.
- A falling alarm event is triggered if the delta rising threshold, the delta falling threshold, and the difference between the current sampled value and the previous sampled value is the same.

The alarm management module defines the set or notification action to take on alarm events.

If the value of the monitored object crosses a threshold multiple times in succession, the managed device triggers an alarm event only for the first crossing. For example, if the value of a sampled object 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 66.

Figure 66 Rising and falling alarm events



Event actions

The Event MIB triggers one or both of the following actions when the trigger condition is met:

- **Set action**—Uses SNMP to set the value of the monitored object.
- **Notification action**—Uses SNMP to send a notification to the NMS. If an object list is specified for the notification action, the notification will carry the specified objects in the object list.

Object list

An object list is a set of MIB objects. You can specify an object list in trigger view, trigger-test view (including trigger-Boolean view, trigger existence view, and trigger threshold view), and action-notification view. If a notification action is triggered, the device sends a notification carrying the object list to the NMS.

If you specify an object list respectively in any two of the views or all the three views, the object lists are added to the triggered notifications in this sequence: trigger view, trigger-test view, and action-notification view.

Object owner

Trigger, event, and object list use an owner and name for unique identification. The owner must be an SNMPv3 user that has been created on the device. If you specify a notification action for a trigger, you must establish an SNMPv3 connection between the device and NMS by using the SNMPv3 username. For more information about SNMPv3 user, see "SNMP configuration".

Restrictions and guidelines: Event MIB configuration

The Event MIB and RMON are independent of each other. You can configure one or both of the features for network management.

You must specify the same owner for a trigger, object lists of the trigger, and events of the trigger.

Event MIB tasks at a glance

To configure the Event MIB, perform the following tasks:

1. Configuring the Event MIB global sampling parameters
2. (Optional.) Configuring Event MIB object lists

   Perform this task so that the device sends a notification that carries the specified object list to the NMS when a notification action is triggered.
3. Configuring an event

   The device supports set and notification actions. Choose one or two of the following actions:
   - Creating an event
   - Configuring a set action for an event
   - Configuring a notification action for an event
   - Enabling the event
4. Configuring a trigger

   A trigger supports Boolean, existence, and threshold tests. Choose one or more of the following tests:
   - Creating a trigger and configuring its basic parameters
   - Configuring a Boolean trigger test
   - Configuring an existence trigger test
   - Configuring a threshold trigger test
   - Enabling trigger sampling
5. (Optional.) Enabling SNMP notifications for the Event MIB module

Prerequisites for configuring the Event MIB

Before you configure the Event MIB, perform the following tasks:

- Create an SNMPv3 user. Assign the user the rights to read and set the values of the specified MIB objects and object lists.

- Make sure the SNMP agent and NMS are configured correctly and the SNMP agent can send notifications to the NMS correctly.

Configuring the Event MIB global sampling parameters

Restrictions and guidelines

This tasks takes effect only on monitored instances to be created.

Procedure

1. Enter system view.
   system-view
2. Set the minimum sampling interval.
   snmp mib event sample minimum *min-number*
   By default, the minimum sampling interval is 1 second.
   The sampling interval of a trigger must be greater than the minimum sampling interval.
3. Configure the maximum number of object instances that can be concurrently sampled.
   snmp mib event sample instance maximum *max-number*
   By default, the value is 0. The maximum number of object instances that can be concurrently sampled is limited by the available resources.

Configuring Event MIB object lists

About this task

Perform this task so that the device sends a notification that carries the specified objects to the NMS when a notification action is triggered.

Procedure

1. Enter system view.
   system-view
2. Configure an Event MIB object list.
   snmp mib event object list owner *group-owner* **name** *group-name* *object-index* **oid** *object-identifier* [**wildcard**]
   The object can be a table node, conceptual row node, table column node, simple leaf node, or parent node of a leaf node.

Configuring an event

Creating an event

1. Enter system view.
   system-view
2. Create an event and enter its view.
   snmp mib event owner *event-owner* **name** *event-name*
3. (Optional.) Configure a description for the event.

description *text*

By default, an event does not have a description.

Configuring a set action for an event

1. Enter system view.
   system-view
2. Enter event view.
   snmp mib event owner *event-owner* **name** *event-name*
3. Enable the set action and enter set action view.
   action set
   By default, no action is specified for an event.
4. Specify an object by its OID for the set action.
   oid *object-identifier*
   By default, no object is specified for a set action.
   The object can be a table node, conceptual row node, table column node, simple leaf node, or parent node of a leaf node.
5. Enable OID wildcarding.
   wildcard oid
   By default, OID wildcarding is disabled.
6. Set the value for the object.
   value *integer-value*
   The default value for the object is 0.
7. (Optional.) Specify a context for the object.
   context *context-name*
   By default, no context is specified for an object.
8. (Optional.) Enable context wildcarding.
   wildcard context
   By default, context wildcarding is disabled.
   A wildcard context contains the specified context and the wildcarded part.

Configuring a notification action for an event

1. Enter system view.
   system-view
2. Enter event view
   snmp mib event owner *event-owner* **name** *event-name*
3. Enable the notification action and enter notification action view.
   action notification
   By default, no action is specified for an event.
4. Specify an object to execute the notification action by its OID.
   oid *object-identifier*
   By default, no object is specified for executing the notification action.
   The object must be a notification object.
5. Specify an object list to be added to the notification triggered by the event.

object list owner *group-owner* **name** *group-name*

By default, no object list is specified for the notification action.

If you do not specify an object list for the notification action or the specified object list does not contain variables, no variables will be carried in the notification.

Enabling the event

Restrictions and guidelines

The Boolean, existence, and threshold events can be triggered only after you perform this task.

To change an enabled event, first disable the event.

Procedure

1. Enter system view.
   system-view
2. Enter event view.
   snmp mib event owner *event-owner* **name** *event-name*
3. Enable the event.
   event enable
   By default, an event is disabled.

Configuring a trigger

Creating a trigger and configuring its basic parameters

1. Enter system view.
   system-view
2. Create a trigger and enter its view.
   snmp mib event trigger owner *trigger-owner* **name** *trigger-name*
   The trigger owner must be an existing SNMPv3 user.
3. (Optional.) Configure a description for the trigger.
   description *text*
   By default, a trigger does not have a description.
4. Set a sampling interval for the trigger.
   frequency *interval*
   By default, the sampling interval is 600 seconds.
   Make sure the sampling interval is greater than or equal to the Event MIB minimum sampling interval.
5. Specify a sampling method.
   sample { **absolute** | **delta** }
   The default sampling method is **absolute**.
6. Specify an object to be sampled by its OID.
   oid *object-identifier*
   By default, the OID is 0.0. No object is specified for a trigger.
   If you execute this command multiple times, the most recent configuration takes effect.
7. (Optional.) Enable OID wildcarding.

wildcard oid

By default, OID wildcarding is disabled.

8. (Optional.) Configure a context for the monitored object.

   context *context-name*

   By default, no context is configured for a monitored object.

9. (Optional.) Enable context wildcarding.

   wildcard context

   By default, context wildcarding is disabled.

10. (Optional.) Specify the object list to be added to the triggered notification.

    object list owner *group-owner* **name** *group-name*

    By default, no object list is specified for a trigger.

Configuring a Boolean trigger test

1. Enter system view.

   system-view

2. Enter trigger view.

   snmp mib event trigger owner *trigger-owner* **name** *trigger-name*

3. Specify a Boolean test for the trigger and enter trigger-Boolean view.

   test boolean

   By default, no test is configured for a trigger.

4. Specify a Boolean test comparison type.

   comparison { **equal** | **greater** | **greaterorequal** | **less** | **lessorequal** | **unequal** }

   The default Boolean test comparison type is **unequal**.

5. Set a reference value for the Boolean trigger test.

   value *integer-value*

   The default reference value for a Boolean trigger test is 0.

6. Specify an event for the Boolean trigger test.

   event owner *event-owner* **name** *event-name*

   By default, no event is specified for a Boolean trigger test.

7. (Optional.) Specify the object list to be added to the notification triggered by the test.

   object list owner *group-owner* **name** *group-name*

   By default, no object list is specified for a Boolean trigger test.

8. Enable the event to be triggered when the trigger condition is met at the first sampling.

   startup enable

   By default, the event is triggered when the trigger condition is met at the first sampling.

   Before the first sampling, you must enable this command to allow the event to be triggered.

Configuring an existence trigger test

1. Enter system view.

   system-view

2. Enter trigger view.

snmp mib event trigger owner *trigger-owner* **name** *trigger-name*

3. Specify an existence test for the trigger and enter trigger-existence view.
   test existence
   By default ,no test is configured for a trigger.
4. Specify an event for the existence trigger test.
   event owner *event-owner* **name** *event-name*
   By default, no event is specified for an existence trigger test.
5. (Optional.) Specify the object list to be added to the notification triggered by the test.
   object list owner *group-owner* **name** *group-name*
   By default, no object list is specified for an existence trigger test.
6. Specify an existence trigger test type.
   type { **absent** | **changed** | **present** }
   The default existence trigger test types are **present** and **absent**.
7. Specify an existence trigger test type for the first sampling.
   startup { **absent** | **present** }
   By default, both the **present** and **absent** existence trigger test types are allowed for the first sampling.

Configuring a threshold trigger test

1. Enter system view.
   system-view
2. Enter trigger view.
   snmp mib event trigger owner *trigger-owner* **name** *trigger-name*
3. Specify a threshold test for the trigger and enter trigger-threshold view.
   test boolean
   By default ,no test is configured for a trigger.
4. Specify the object list to be added to the notification triggered by the test.
   object list owner *group-owner* **name** *group-name*
   By default, no object list is specified for a threshold trigger test.
5. (Optional.) Specify the type of the threshold trigger test for the first sampling.
   startup { **falling** | **rising** | **rising-or-falling** }
   The default threshold trigger test type for the first sampling is **rising-or-falling**.
6. Specify the delta falling threshold and the falling alarm event triggered when the delta value (difference between the current sampled value and the previous sampled value) is smaller than or equal to the delta falling threshold.
   delta falling { **event owner** *event-owner* **name** *event-name* | **value** *integer-value* }
   By default, the delta falling threshold is 0, and no falling alarm event is specified.
7. Specify the delta rising threshold and the rising alarm event triggered when the delta value is greater than or equal to the delta rising threshold.
   delta rising { **event owner** *event-owner* **name** *event-name* | **value** *integer-value* }
   By default, the delta rising threshold is 0, and no rising alarm event is specified.

8. Specify the falling threshold and the falling alarm event triggered when the sampled value is smaller than or equal to the threshold.

   falling { **event owner** *event-owner* **name** *event-name* | **value** *integer-value* }

   By default, the falling threshold is 0, and no falling alarm event is specified.

9. Specify the rising threshold and the ring alarm event triggered when the sampled value is greater than or equal to the threshold.

   rising { **event owner** *event-owner* **name** *event-name* | **value** *integer-value* }

   By default, the rising threshold is 0, and no rising alarm event is specified.

Enabling trigger sampling

Restrictions and guidelines

Enable trigger sampling after you complete trigger parameters configuration. You cannot modify trigger parameters after trigger sampling is enabled. To modify trigger parameters, first disable trigger sampling.

Procedure

1. Enter system view.

   system-view

2. Enter trigger view.

   snmp mib event trigger owner *trigger-owner* **name** *trigger-name*

3. Enable trigger sampling.

   trigger enable

   By default, trigger sampling is disabled.

Enabling SNMP notifications for the Event MIB module

About this task

To report critical Event MIB events to an NMS, enable SNMP notifications for the Event MIB module. For Event MIB event notifications to be sent correctly, you must also configure SNMP on the device. For more information about SNMP configuration, see the network management and monitoring configuration guide for the device.

Procedure

1. Enter system view.

   system-view

2. Enable snmp notifications for the Event MIB module.

   snmp-agent trap enable event-mib

   By default, SNMP notifications are enabled for the Event MIB module.

Display and maintenance commands for Event MIB

Execute **display** commands in any view.

Task	Command
Display Event MIB configuration and statistics.	**display snmp mib event**
Display event information.	**display snmp mib event event** [**owner** *event-owner* **name** *event-name*]
Display object list information.	**display snmp mib event object list** [**owner** *group-owner* **name** *group-name*]
Display global Event MIB configuration and statistics.	**display snmp mib event summary**
Display trigger information.	**display snmp mib event trigger** [**owner** *trigger-owner* **name** *trigger-name*]

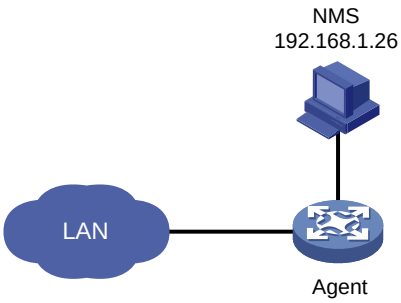
Event MIB configuration examples

Example: Configuring an existence trigger test

Network configuration

As shown in Figure 67, the device acts as the agent. Use the Event MIB to monitor the device. When interface hot-swap or virtual interface creation or deletion occurs on the device, the agent sends an mteTriggerFired notification to the NMS.

Figure 67 Network diagram



Procedure

1. Enable and configure the SNMP agent on the device:

   # Create SNMPv3 group **g3** and add SNMPv3 user **owner1** to **g3**.

   ```
   <Sysname> system-view
   [Sysname] snmp-agent usm-user v3 owner1 g3
   [Sysname] snmp-agent group v3 g3 read-view a write-view a notify-view a
   [Sysname] snmp-agent mib-view included a iso
   ```

   # Configure context **contextnameA** for the agent.

```
[Sysname] snmp-agent context contextnameA
```

Enable SNMP notifications for the Event MIB module. Specify the NMS at **192.168.1.26 as the target host** for the notifications.

```
[Sysname] snmp-agent trap enable event-mib
[Sysname] snmp-agent target-host trap address udp-domain 192.168.1.26 params securityname owner1 v3
```

2. Configure the Event MIB global sampling parameters.

Set the Event MIB minimum sampling interval to 50 seconds.

```
[Sysname] snmp mib event sample minimum 50
```

Set the maximum number to 100 for object instances that can be concurrently sampled.

```
[Sysname] snmp mib event sample instance maximum 100
```

3. Create and configure a trigger:

Create a trigger and enter its view. Specify its owner as **owner1** and its name as **triggerA**.

```
[Sysname] snmp mib event trigger owner owner1 name triggerA
```

Set the sampling interval to 60 seconds. Make sure the sampling interval is greater than or equal to the Event MIB minimum sampling interval.

```
[Sysname-trigger-owner1-triggerA] frequency 60
```

Specify object OID **1.3.6.1.2.1.2.2.1.1** as the monitored object. Enable OID wildcarding.

```
[Sysname-trigger-owner1-triggerA] oid 1.3.6.1.2.1.2.2.1.1
[Sysname-trigger-owner1-triggerA] wildcard oid
```

Configure context **contextnameA** for the monitored object and enable context wildcarding.

```
[Sysname-trigger-owner1-triggerA] context contextnameA
[Sysname-trigger-owner1-triggerA] wildcard context
```

Specify the existence trigger test for the trigger.

```
[Sysname-trigger-owner1-triggerA] test existence
[Sysname-trigger-owner1-triggerA-existence] quit
```

Enable trigger sampling.

```
[Sysname-trigger-owner1-triggerA] trigger enable
[Sysname-trigger-owner1-triggerA] quit
```

Verifying the configuration

Display Event MIB brief information.

```
[Sysname] display snmp mib event summary
TriggerFailures               : 0
EventFailures                 : 0
SampleMinimum                 : 50
SampleInstanceMaximum         : 100
SampleInstance                : 20
SampleInstancesHigh           : 20
SampleInstanceLacks           : 0
```

Display information about the trigger with owner **owner1** and name **trigger A**.

```
[Sysname] display snmp mib event trigger owner owner1 name triggerA
Trigger entry triggerA owned by owner1:
  TriggerComment              : N/A
  TriggerTest                 : existence
  TriggerSampleType           : absoluteValue
  TriggerValueID              : 1.3.6.1.2.1.2.2.1.1<ifIndex>
  TriggerValueIDWildcard      : true
```

```
  TriggerTargetTag            : N/A
  TriggerContextName          : contextnameA
  TriggerContextNameWildcard  : true
  TriggerFrequency(in seconds): 60
  TriggerObjOwner             : N/A
  TriggerObjName              : N/A
  TriggerEnabled              : true
  Existence entry:
   ExiTest                    : present | absent
   ExiStartUp                 : present | absent
   ExiObjOwner                : N/A
   ExiObjName                 : N/A
   ExiEvtOwner                : N/A
   ExiEvtName                 : N/A
```

Create VLAN-interface 2 on the device.

```
[Sysname] vlan 2
[Sysname-vlan2] quit
[Sysname] interface vlan 2
```

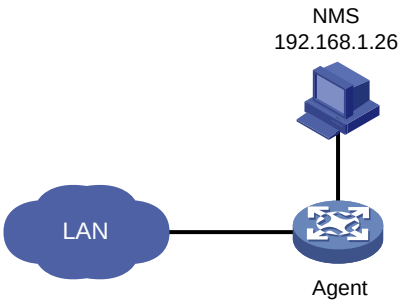
The NMS receives an mteTriggerFired notification from the device.

Example: Configuring a Boolean trigger test

Network configuration

As shown in Figure 67, the device acts as the agent. The NMS uses SNMPv3 to monitor and manage the device. Configure a trigger and configure a Boolean trigger test for the trigger. When the trigger condition is met, the agent sends an mteTriggerFired notification to the NMS.

Figure 68 Network diagram



Procedure

1. Enable and configure the SNMP agent on the device:

   # Create SNMPv3 group **g3 and** add SNNPv3 user **owner1** to **g3**.

   ```
   <Sysname> system-view
   [Sysname] snmp-agent usm-user v3 owner1 g3
   [Sysname] snmp-agent group v3 g3 read-view a write-view a notify-view a
   [Sysname] snmp-agent mib-view included a iso
   ```

   # Enable SNMP notifications for the Event MIB module. Specify the NMS at **192.168.1.26 as the target host** for the notifications.

   ```
   [Sysname] snmp-agent trap enable event-mib
   ```

```
[Sysname] snmp-agent target-host trap address udp-domain 192.168.1.26 params
securityname owner1 v3
```

2. Configure the Event MIB global sampling parameters.

Set the Event MIB minimum sampling interval to 50 seconds.

```
[Sysname] snmp mib event sample minimum 50
```

Set the maximum number to 100 for object instances that can be concurrently sampled.

```
[Sysname] snmp mib event sample instance maximum 100
```

3. Configure Event MIB object lists **objectA**, **objectB**, and **objectC**.

```
[Sysname] snmp mib event object list owner owner1 name objectA 1 oid
1.3.6.1.4.1.25506.2.6.1.1.1.1.6.11
[Sysname] snmp mib event object list owner owner1 name objectB 1 oid
1.3.6.1.4.1.25506.2.6.1.1.1.1.7.11
[Sysname] snmp mib event object list owner owner1 name objectC 1 oid
1.3.6.1.4.1.25506.2.6.1.1.1.1.8.11
```

4. Configure an event:

Create an event and enter its view. Specify its owner as **owner1** and its name as **EventA**.

```
[Sysname] snmp mib event owner owner1 name EventA
```

Specify the notification action for the event. Specify object OID **1.3.6.1.4.1.25506.2.6.2.0.5 (hh3cEntityExtMemUsageThresholdNotification)** to execute the notification.

```
[Sysname-event-owner1-EventA] action notification
[Sysname-event-owner1-EventA-notification] oid 1.3.6.1.4.1.25506.2.6.2.0.5
```

Specify the object list with owner **owner 1** and name **objectC** to be added to the notification when the notification action is triggered

```
[Sysname-event-owner1-EventA-notification] object list owner owner1 name objectC
[Sysname-event-owner1-EventA-notification] quit
```

Enable the event.

```
[Sysname-event-owner1-EventA] event enable
[Sysname-event-owner1-EventA] quit
```

5. Configure a trigger:

Create a trigger and enter its view. Specify its owner as **owner1** and its name as **triggerA**.

```
[Sysname] snmp mib event trigger owner owner1 name triggerA
```

Set the sampling interval to 60 seconds. Make sure the interval is greater than or equal to the global minimum sampling interval.

```
[Sysname-trigger-owner1-triggerA] frequency 60
```

Specify object OID **1.3.6.1.4.1.25506.2.6.1.1.1.1.9.11** as the monitored object.

```
[Sysname-trigger-owner1-triggerA] oid 1.3.6.1.4.1.25506.2.6.1.1.1.1.9.11
```

Specify the object list with owner **owner1** and name **objectA** to be added to the notification when the notification action is triggered.

```
[Sysname-trigger-owner1-triggerA] object list owner owner1 name objectA
```

Configure a Boolean trigger test. Set its comparison type to greater, reference value to 10, and specify the event with owner **owner1** and name **EventA**, object list with owner **owner1** and name **objectB** for the test.

```
[Sysname-trigger-owner1-triggerA] test boolean
[Sysname-trigger-owner1-triggerA-boolean] comparison greater
[Sysname-trigger-owner1-triggerA-boolean] value 10
[Sysname-trigger-owner1-triggerA-boolean] event owner owner1 name EventA
[Sysname-trigger-owner1-triggerA-boolean] object list owner owner1 name objectB
[Sysname-trigger-owner1-triggerA-boolean] quit
```

Enable trigger sampling.

```
[Sysname-trigger-owner1-triggerA] trigger enable
[Sysname-trigger-owner1-triggerA] quit
```

Verifying the configuration

Display Event MIB configuration and statistics.

```
[Sysname] display snmp mib event summary
TriggerFailures               : 0
EventFailures                 : 0
SampleMinimum                 : 50
SampleInstanceMaximum         : 10
SampleInstance                : 1
SampleInstancesHigh           : 1
SampleInstanceLacks           : 0
```

Display information about the Event MIB object lists.

```
[Sysname] display snmp mib event object list
Object list objectA owned by owner1:
  ObjIndex                    : 1
  ObjID                       : 1.3.6.1.4.1.25506.2.6.1.1.1.1.6.11<hh3cEntityExt
CpuUsage.11>
  ObjIDWildcard               : false
Object list objectB owned by owner1:
  ObjIndex                    : 1
  ObjID                       : 1.3.6.1.4.1.25506.2.6.1.1.1.1.7.11<hh3cEntityExt
CpuUsageThreshold.11>
  ObjIDWildcard               : false
Object list objectC owned by owner1:
  ObjIndex                    : 1
  ObjID                       : 1.3.6.1.4.1.25506.2.6.1.1.1.1.8.11<hh3cEntityExt
MemUsage.11>
  ObjIDWildcard               : false
```

Display information about the event.

```
[Sysname]display snmp mib event event owner owner1 name EventA
Event entry EventA owned by owner1:
EvtComment                   : N/A
EvtAction                    : notification
EvtEnabled                   : true
Notification entry:
NotifyOID                   : 1.3.6.1.4.1.25506.2.6.2.0.5<hh3cEntityExtMemUsag
eThresholdNotification>
NotifyObjOwner              : owner1
NotifyObjName               : objectC
```

Display information about the trigger.

```
[Sysname] display snmp mib event trigger owner owner1 name triggerA
Trigger entry triggerA owned by owner1:
  TriggerComment              : N/A
  TriggerTest                 : boolean
  TriggerSampleType           : absoluteValue
```

```
  TriggerValueID              : 1.3.6.1.4.1.25506.2.6.1.1.1.1.9.11<hh3cEntityExt
MemUsageThreshold.11>
  TriggerValueIDWildcard      : false
  TriggerTargetTag            : N/A
  TriggerContextName          : N/A
  TriggerContextNameWildcard  : false
  TriggerFrequency(in seconds): 60
  TriggerObjOwner             : owner1
  TriggerObjName              : objectA
  TriggerEnabled              : true
  Boolean entry:
   BoolCmp                    : greater
   BoolValue                  : 10
   BoolStartUp                : true
   BoolObjOwner               : owner1
   BoolObjName                : objectB
   BoolEvtOwner               : owner1
   BoolEvtName                : EventA
```

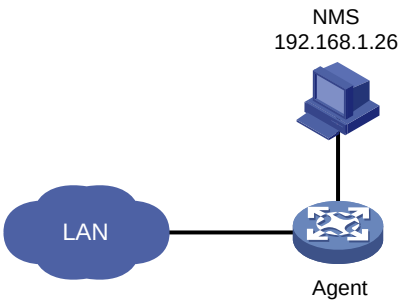
When the value of the monitored object 1.3.6.1.4.1.25506.2.6.1.1.1.1.9.11 becomes greater than 10, the NMS receives an mteTriggerFired notification.

Example: Configuring a threshold trigger test

Network configuration

As shown in Figure 67, the device acts as the agent. The NMS uses SNMPv3 to monitor and manage the device. Configure a trigger and configure a threshold trigger test for the trigger. When the trigger conditions are met, the agent sent an mteTriggerFired notification to the NMS.

Figure 69 Network diagram



Procedure

1. Enable and configure the SNMP agent on the device:

   # Create SNMPv3 group **g3 and** add SNMPv3 user **owner1** to **g3**.

   ```
   <Sysname> system-view
   [Sysname] snmp-agent usm-user v3 owner1 g3
   [Sysname] snmp-agent group v3 g3 read-view a write-view a notify-view a
   [Sysname] snmp-agent mib-view included a iso
   ```

   # Enable SNMP notifications for the Event MIB module. Specify the NMS at **192.168.1.26 as the target host** for the notifications.

   ```
   [Sysname] snmp-agent trap enable event-mib
   ```

```
[Sysname] snmp-agent target-host trap address udp-domain 192.168.1.26 params
securityname owner1 v3
[Sysname] snmp-agent trap enable
```

2. Configure the Event MIB global sampling parameters.

   # Set the Event MIB minimum sampling interval to 50 seconds.

   ```
   [Sysname] snmp mib event sample minimum 50
   ```

   # Set the maximum number to 10 for object instances that can be concurrently sampled.

   ```
   [Sysname] snmp mib event sample instance maximum 10
   ```

3. Create and configure a trigger:

   # Create a trigger and enter its view. Specify its owner as **owner1** and its name as **triggerA**.

   ```
   [Sysname] snmp mib event trigger owner owner1 name triggerA
   ```

   # Set the sampling interval to 60 seconds. Make sure the interval is greater than or equal to the Event MIB minimum sampling interval.

   ```
   [Sysname-trigger-owner1-triggerA] frequency 60
   ```

   # Specify object OID **1.3.6.1.4.1.25506.2.6.1.1.1.1.7.11** as the monitored object.

   ```
   [Sysname-trigger-owner1-triggerA] oid 1.3.6.1.4.1.25506.2.6.1.1.1.1.7.11
   ```

   # Configure a threshold trigger test. Specify the rising threshold to 80 and the falling threshold to 10 for the test.

   ```
   [Sysname-trigger-owner1-triggerA] test threshold
   [Sysname-trigger-owner1-triggerA-threshold] rising value 80
   [Sysname-trigger-owner1-triggerA-threshold] falling value 10
   [Sysname-trigger-owner1-triggerA-threshold] quit
   ```

   # Enable trigger sampling.

   ```
   [Sysname-trigger-owner1-triggerA] trigger enable
   [Sysname-trigger-owner1-triggerA] quit
   ```

Verifying the configuration

Display Event MIB configuration and statistics.

```
[Sysname] display snmp mib event summary
TriggerFailures               : 0
EventFailures                 : 0
SampleMinimum                 : 50
SampleInstanceMaximum         : 10
SampleInstance                : 1
SampleInstancesHigh           : 1
SampleInstanceLacks           : 0
```

Display information about the trigger.

```
[Sysname] display snmp mib event trigger owner owner1 name triggerA
Trigger entry triggerA owned by owner1:
  TriggerComment              : N/A
  TriggerTest                 : threshold
  TriggerSampleType           : absoluteValue
  TriggerValueID              : 1.3.6.1.4.1.25506.2.6.1.1.1.1.7.11<hh3cEntityExt
CpuUsageThreshold.11>
  TriggerValueIDWildcard      : false
  TriggerTargetTag            : N/A
  TriggerContextName          : N/A
  TriggercontextNameWildcard  : false
```

```
  TriggerFrequency(in seconds): 60
  TriggerObjOwner             : N/A
  TriggerObjName              : N/A
  TriggerEnabled              : true
Threshold entry:
   ThresStartUp               : risingOrFalling
   ThresRising                : 80
   ThresFalling               : 10
   ThresDeltaRising           : 0
   ThresDeltaFalling          : 0
   ThresObjOwner              : N/A
   ThresObjName               : N/A
   ThresRisEvtOwner           : N/A
   ThresRisEvtName            : N/A
   ThresFalEvtOwner           : N/A
   ThresFalEvtName            : N/A
   ThresDeltaRisEvtOwner      : N/A
   ThresDeltaRisEvtName       : N/A
   ThresDeltaFalEvtOwner      : N/A
   ThresDeltaFalEvtName       : N/A
```

When the rising threshold of the monitored object 1.3.6.1.4.1.25506.2.6.1.1.1.1.7.11 is greater than 80, the NMS receives an mteTriggerFired notification.

Configuring NETCONF

About NETCONF

Network Configuration Protocol (NETCONF) is an XML-based network management protocol. It provides programmable mechanisms to manage and configure network devices. Through NETCONF, you can configure device parameters, retrieve parameter values, and collect statistics. For a network that has devices from vendors, you can develop a NETCONF-based NMS system to configure and manage devices in a simple and effective way.

NETCONF structure

NETCONF has the following layers: content layer, operations layer, RPC layer, and transport protocol layer.

Table 9 NETCONF layers and XML layers

NETCONF layer	XML layer	Description
Content	Configuration data, status data, and statistics	Contains a set of managed objects, which can be configuration data, status data, and statistics. For information about the operable data, see the NETCONF XML API reference for the device.
Operations	<get>, <get-config>, <edit-config>…	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 information about operations supported on the device, see "Supported NETCONF operations."
RPC	<rpc> and <rpc-reply>	Provides a simple, transport-independent framing mechanism for encoding RPCs. The <rpc> and <rpc-reply> elements are used to enclose NETCONF requests and responses (data at the operations layer and the content layer).
Transport protocol	In non-FIPS mode: Console, Telnet, SSH, HTTP, HTTPS, and TLS In FIPS mode: Console, SSH, HTTPS, and TLS	Provides reliable, connection-oriented, serial data links. The following transport layer sessions are available in non-FIPS mode: • CLI sessions, including NETCONF over Telnet sessions, NETCONF over SSH sessions, and NETCONF over console sessions. • NETCONF over SOAP sessions, including NETCONF over SOAP over HTTP sessions and NETCONF over SOAP over HTTPS sessions. The following transport layer sessions are available in FIPS mode: • CLI sessions, including NETCONF over SSH sessions and NETCONF over console sessions. • NETCONF over SOAP over HTTPS sessions.

NETCONF message format

NETCONF

All NETCONF messages are XML-based and comply with RFC 4741. An incoming NETCONF message 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 device.

The following example shows a NETCONF message for getting all parameters of all interfaces on the device:

```
<?xml version="1.0" encoding="utf-8"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-bulk>
    <filter type="subtree">
      <top xmlns="http://www.hp.com/netconf/data:1.0">
        <Ifmgr>
          <Interfaces>
                 <Interface/>
          </Interfaces>
        </Ifmgr>
      </top>
    </filter>
  </get-bulk>
</rpc>
```

NETCONF over SOAP

All NETCONF over SOAP messages are XML-based and comply with RFC 4741. NETCONF messages are contained in the <Body> element of SOAP messages. NETCONF over SOAP messages also comply with the following rules:

- SOAP messages must use the SOAP Envelope namespaces.
- SOAP messages must use the SOAP Encoding namespaces.
- SOAP messages cannot contain the following information:
  - DTD reference.
  - XML processing instructions.

The following example shows a NETCONF over SOAP message for getting all parameters of all interfaces on the device:

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <auth:Authentication env:mustUnderstand="1"
xmlns:auth="http://www.hp.com/netconf/base:1.0">
      <auth:AuthInfo>800207F0120020C</auth:AuthInfo>
    </auth:Authentication>
  </env:Header>
  <env:Body>
    <rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <get-bulk>
        <filter type="subtree">
          <top xmlns="http://www.hp.com/netconf/data:1.0">
            <Ifmgr>
              <Interfaces>
                <Interface/>
              </Interfaces>
            </Ifmgr>
          </top>
        </filter>
```

```
      </get-bulk>
    </rpc>
  </env:Body>
</env:Envelope>
```

How to use NETCONF

You can use NETCONF to manage and configure the device by using the methods in Table 10.

Table 10 NETCONF methods for configuring the device

Configuration tool	Login method	Remarks
CLI	• Console port • SSH • Telnet	To perform NETCONF operations, copy valid NETCONF messages to the CLI in XML view.
Custom user interface	N/A	To use this method, you must enable NETCONF over SOAP. NETCONF messages will be encapsulated in SOAP for transmission.

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 tasks at a glance

To configure NETCONF, perform the following tasks:

1. Establishing a NETCONF session
   a. (Optional.) Setting NETCONF session attributes
   b. Establishing NETCONF over SOAP sessions
   c. Establishing NETCONF over SSH sessions
   d. Establishing NETCONF over Telnet or NETCONF over console sessions
   e. Exchanging capabilities
2. (Optional.) Retrieving device configuration information
   - Retrieving device configuration and state information

- Retrieving non-default settings
- Retrieving NETCONF information
- Retrieving YANG file content
- Retrieving NETCONF session information

3. (Optional.) Filtering data

- Table-based filtering
- Column-based filtering

4. (Optional.) Locking or unlocking the running configuration
   a. Locking the running configuration
   b. Unlocking the running configuration
5. (Optional.) Modifying the configuration
6. (Optional.) Managing configuration files
   - Saving the running configuration
   - Loading the configuration
   - Rolling back the configuration
7. (Optional.) Enabling preprovisioning
8. (Optional.) Performing CLI operations through NETCONF
9. (Optional.) Subscribing to events
   - Subscribing to syslog events
   - Subscribing to events monitored by NETCONF
   - Subscribing to events reported by modules
   - Canceling an event subscription
10. (Optional.) Terminating NETCONF sessions
11. (Optional.) Returning to the CLI

Establishing a NETCONF session

Restrictions and guidelines for NETCONF session establishment

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.

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.

You can use the **`aaa session-limit`** command to set the maximum number of NETCONF sessions that the device can support. If the upper limit is reached, new NETCONF users cannot access the device. For information about this command, see AAA in *Security Configuration Guide*.

Setting NETCONF session attributes

About this task

NETCONF supports the following types of namespaces:

- **Common namespace**—The common namespace is shared by all modules. In a packet that uses the common namespace, the namespace is indicated in the <top> element, and the modules are listed under the <top> element.

  Example:

```
<rpc message-id="100" xmlns="urn:ietf:Params:xml:ns:netconf:base:1.0">
   <get-bulk>
      <filter type="subtree">
         <top xmlns="http://www.hp.com/netconf/data:1.0">
            <Ifmgr>
               <Interfaces>
               </Interfaces>
            </Ifmgr>
         </top>
      </filter>
   </get-bulk>
</rpc>
```

- **Module-specific namespace**—Each module has its own namespace. A packet that uses a module-specific namespace does not have the <top> element. The namespace follows the module name.

  Example:

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <get-bulk>
      <filter type="subtree">
            <Ifmgr xmlns="http://www.hp.com/netconf/data:1.0-Ifmgr">
               <Interfaces>
               </Interfaces>
            </Ifmgr>
      </filter>
   </get-bulk>
</rpc>
```

The common namespace is incompatible with module-specific namespaces. To set up a NETCONF session, the device and the client must use the same type of namespaces. By default, the common namespace is used. If the client does not support the common namespace, use this feature to configure the device to use module-specific namespaces.

Procedure

1. Enter system view.

   system-view

2. Set the NETCONF session idle timeout time.

   netconf { **agent** | **soap** } **idle-timeout** *minute*

Keyword	Description
agent	Specifies the following sessions: • NETCONF over SSH sessions. • NETCONF over Telnet sessions. • NETCONF over console sessions. By default, the idle timeout time is 0, and the sessions never time out.

Keyword	Description
soap	Specifies the following sessions: • NETCONF over SOAP over HTTP sessions. • NETCONF over SOAP over HTTPS sessions. The default setting is 10 minutes.

3. Enable NETCONF logging.

   netconf log source { **all** | { **agent** | **soap** | **web** } * } { **protocol-operation** { **all** | { **action** | **config** | **get** | **set** | **session** | **syntax** | **others** } * } | **row-operation** | **verbose** }

   By default, the device logs only row operations for <action> and <edit-config> operations. NETCONF logging is disabled for all the other types of operations.

   The **web** keyword is not supported in the current software version.

4. Configure NETCONF to use module-specific namespaces.

   netconf capability specific-namespace

   By default, the common namespace is used.

   For the setting to take effect, you must reestablish the NETCONF session.

Establishing NETCONF over SOAP sessions

About this task

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

Restrictions and guidelines

You can add an authentication domain to the <UserName> parameter of a SOAP request. The authentication domain takes effect only on the current request.

The mandatory authentication domain configured by using the **netconf soap domain** command takes precedence over the authentication domain specified in the <UserName> parameter of a SOAP request.

Procedure

1. Enter system view.

   system-view

2. Enable NETCONF over SOAP.

   In non-FIPS mode:

   netconf soap { **http** | **https** } **enable**

   In FIPS mode:

   netconf soap https enable

   By default, the NETCONF over SOAP feature is disabled.

3. Set the DSCP value for NETCONF over SOAP packets.

   In non-FIPS mode:

   netconf soap { **http** | **https** } **dscp** *dscp-value*

   In FIPS mode:

   netconf soap https dscp *dscp-value*

   By default, the DSCP value is 0 for NETCONF over SOAP packets.

4. Use an IPv4 ACL to control NETCONF over SOAP access.

   In non-FIPS mode:

   netconf soap { **http** | **https** } **acl** { *ipv4-acl-number* | **name** *ipv4-acl-name* }

   In FIPS mode:

   netconf soap https acl { *ipv4-acl-number* | **name** *ipv4-acl-name* }

   By default, no IPv4 ACL is applied to control NETCONF over SOAP access.

   Only clients permitted by the IPv4 ACL can establish NETCONF over SOAP sessions.

5. Specify a mandatory authentication domain for NETCONF users.

   netconf soap domain *domain-name*

   By default, no mandatory authentication domain is specified for NETCONF users. For information about authentication domains, see *Security Configuration Guide*.

6. Use the custom user interface to establish a NETCONF over SOAP session with the device.

   For information about the custom user interface, see the user guide for the interface.

Establishing NETCONF over SSH sessions

Prerequisites

Before establishing a NETCONF over SSH session, make sure the custom user interface can access the device through SSH.

Procedure

1. Enter system view.

   system-view

2. Enable NETCONF over SSH.

   netconf ssh server enable

   By default, NETCONF over SSH is disabled.

3. Specify the listening port for NETCONF over SSH packets.

   netconf ssh server port *port-number*

   By default, the listening port number is 830.

4. Use the custom user interface to establish a NETCONF over SSH session with the device. For information about the custom user interface, see the user guide for the interface.

Establishing NETCONF over Telnet or NETCONF over console sessions

Restrictions and guidelines

To ensure the format correctness of a NETCONF message, do not enter the message manually. Copy and paste the message.

While the device is performing a NETCONF operation, do not perform any other operations, such as pasting a NETCONF message or pressing **Enter**.

For the device to identify NETCONF messages, you must add end mark **]]>]]>** at the end of each NETCONF message. Examples in this document do not necessarily have this end mark. Do add the end mark in actual operations.

Prerequisites

To establish a NETCONF over Telnet session or a NETCONF over console session, first log in to the device through Telnet or the console port.

Procedure

To enter XML view, execute the following command in user view:

xml

If the XML view prompt appears, the NETCONF over Telnet session or NETCONF over console session is established successfully.

Exchanging capabilities

About this task

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

Hello message from the device to the client

```
<?xml version="1.0" encoding="UTF-8"?><hello
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"><capabilities><capability>urn:ietf:pa
rams:netconf:base:1.1</capability><capability>urn:ietf:params:netconf:writable-runnin
g</capability><capability>urn:ietf:params:netconf:capability:notification:1.0</capabi
lity><capability>urn:ietf:params:netconf:capability:validate:1.1</capability><capabil
ity>urn:ietf:params:netconf:capability:interleave:1.0</capability><capability>urn:hp:
params:netconf:capability:hp-netconf-ext:1.0</capability></capabilities><session-id>1
</session-id></hello>]]>]]>
```

The <capabilities> element carries the capabilities supported by the device. The supported capabilities vary by device model.

The <session-id> element carries the unique ID assigned to the NETCONF session.

Hello message from the client to the device

After receiving the hello message from the device, copy the following hello message to notify the device of the capabilities supported by the client:

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

Item	Description
capability-set	Specifies a set of capabilities supported by the client. Use the <capability> and </capability> tags to enclose each user-defined capability set.

Retrieving device configuration information

Restrictions and guidelines for device configuration retrieval

During a <get>, <get-bulk>, <get-config>, or <get-bulk-config> operation, NETCONF replaces unidentifiable characters in the retrieved data with question marks (?) before sending the data to the client. If the process for a relevant module is not started yet, the operation returns the following message:

```
<?xml version="1.0"?>
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data/>
</rpc-reply>
```

The <get><netconf-state/></get> operation does not support data filtering.

For more information about the NETCONF operations, see the NETCONF XML API references for the device.

Retrieving device configuration and state information

You can use the following NETCONF operations to retrieve device configuration and state information:

- **<get> operation**—Retrieves all device configuration and state information that match the specified conditions.
- **<get-bulk> operation**—Retrieves 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 returned output does not include the index information.

The <get> message and <get-bulk> message share the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <getoperation>
    <filter>
      <top xmlns="http://www.hp.com/netconf/data:1.0">
          Specify the module, submodule, table name, and column name
      </top>
    </filter>
  </getoperation>
</rpc>
```

Item	Description
getoperation	Operation name, **get** or **get-bulk**.

Item	Description
filter	Specifies the filtering conditions, such as the module name, submodule name, table name, and column name. • If you specify a module name, the operation retrieves the data for the specified module. If you do not specify a module name, the operation retrieves the data for all modules. • If you specify a submodule name, the operation retrieves the data for the specified submodule. If you do not specify a submodule name, the operation retrieves the data for all submodules. • If you specify a table name, the operation retrieves the data for the specified table. If you do not specify a table name, the operation retrieves the data for all tables. • If you specify only the index column, the operation retrieves the data for all columns. If you specify the index column and any other columns, the operation retrieves the data for the index column and the specified columns.

A <get-bulk> message can carry the **count** and **index** attributes.

Item	Description
index	Specifies the index. If you do not specify this item, the index value starts with 1 by default.

count	Specifies the data entry quantity.
	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.
	• The <get-bulk> operation retrieves all the rest data entries starting from the data entry next to the one with the specified index if either of the following conditions occurs:
	  ○ You do not specify the **count** attribute.
	  ○ The number of matching data entries is less than the value of the **count** attribute.

The following <get-bulk> message example specifies the **count** and **index** attributes:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:xc="http://www.hp.com/netconf/base:1.0">
  <get-bulk>
    <filter type="subtree">
      <top xmlns="http://www.hp.com/netconf/data:1.0"
xmlns:base="http://www.hp.com/netconf/base:1.0">
        <Syslog>
          <Logs xc:count="5">
            <Log>
              <Index>10</Index>
            </Log>
              </Logs>
        </Syslog>
      </top>
    </filter>
  </get-bulk>
```

```
</rpc>
```

When retrieving interface information, the device cannot identify whether an integer value for the <IfIndex> element represents an interface name or index. When retrieving VPN instance information, the device cannot identify whether an integer value for the <vrfindex> element represents a VPN name or index. To resolve the issue, you can use the **valuetype** attribute to specify the value type.

The **valuetype** attribute has the following values:

Value	Description
name	The element is carrying a name.
index	The element is carrying an index.
auto	Default value. The device uses the value of the element as a name for information matching. If no match is found, the device uses the value as an index for interface or information matching.

The following example specifies an index-type value for the <IfIndex> element:

```
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <getoperation>
    <filter>
      <top xmlns="http://www.hp.com/netconf/config:1.0"
xmlns:base="http://www.hp.com/netconf/base:1.0">
       <VLAN>
        <TrunkInterfaces>
          <Interface>
            <IfIndex base:valuetype="index">1</IfIndex>
          </Interface>
        </TrunkInterfaces>
       </VLAN>
      </top>
    </filter >
  </getoperation>
</rpc>
```

If the <get> or < get-bulk> operation succeeds, the device returns the retrieved data in the following format:

```
<?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>
```

Retrieving non-default settings

The <get-config> and <get-bulk-config> operations are used to retrieve all non-default settings. The <get-config> and <get-bulk-config> messages can contain the <filter> element for filtering data.

The <get-config> and <get-bulk-config> messages are similar. The following is a <get-config> message example:

```
<?xml version="1.0"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
```

```
    <source>
      <running/>
    </source>
    <filter>
      <top xmlns="http://www.hp.com/netconf/config:1.0">
          Specify the module name, submodule name, table name, and column name
      </top>
    </filter>
  </get-config>
</rpc>
```

If the <get-config> or <get-bulk-config> operation succeeds, the device returns the retrieved data in the following format:

```
<?xml version="1.0"?>
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    Data matching the specified filter
  </data>
</rpc-reply>
```

Retrieving NETCONF information

Use the <get><netconf-state/></get> message to retrieve NETCONF information.

Copy the following text to the client to retrieve NETCONF information:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="m-641" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter type='subtree'>
      <netconf-state xmlns='urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring'>
        <getType/>
      </netconf-state>
    </filter>
  </get>
</rpc>
```

If you do not specify a value for *getType*, the retrieval operation retrieves all NETCONF information.

The value for *getType* can be one of the following operations:

Operation	Description
capabilities	Retrieves device capabilities.
datastores	Retrieves databases from the device.
schemas	Retrieves the list of the YANG file names from the device.
sessions	Retrieves session information from the device.
statistics	Retrieves NETCONF statistics.

If the <get><netconf-state/></get> operation succeeds, the device returns a response in the following format:

```
<?xml version="1.0"?>
```

```
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
     Retrieved NETCONF information
  </data>
</rpc-reply>
```

Retrieving YANG file content

YANG files save the NETCONF operations supported by the device. A user can know the supported operations by retrieving and analyzing the content of YANG files.

YANG files are integrated in the device software and are named in the format of *yang_identifier@yang_version.yang*. You cannot view the YANG file names by executing the **dir** command. For information about how to retrieve the YANG file names, see "Retrieving NETCONF information."

Copy the following text to the client to retrieve the YANG file named **syslog-data@2017-01-01.yang**:

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-schema xmlns='urn:ietf:params:xml:ns:yang:ietf-netconf-monitoring'>
    <identifier>syslog-data</identifier>
    <version>2017-01-01</version>
    <format>yang</format>
  </get-schema>
</rpc>
```

If the <get-schema> operation succeeds, the device returns a response in the following format:

```
<?xml version="1.0"?>
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
     Content of the specified YANG file
  </data>
</rpc-reply>
```

Retrieving NETCONF session information

Use the <get-sessions> operation to retrieve NETCONF session information of the device.

Copy the following message to the client to retrieve NETCONF session information from the device:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-sessions/>
</rpc>
```

If the <get-sessions> operation succeeds, the device returns a response in the following format:

```
<?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>
```

Example: Retrieving a data entry for the interface table

Network configuration

Retrieve a data entry for the interface table.

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.

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:web="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-bulk>
    <filter type="subtree">
      <top xmlns="http://www.hp.com/netconf/data:1.0"
xmlns:web="http://www.hp.com/netconf/base:1.0">
        <Ifmgr>
          <Interfaces web:count="1">
          </Interfaces>
        </Ifmgr>
      </top>
    </filter>
  </get-bulk>
</rpc>
```

Verifying the configuration

If the client receives the following text, the <get-bulk> operation is successful:

```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:web="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="100">
  <data>
    <top xmlns="http://www.hp.com/netconf/data:1.0">
      <Ifmgr>
        <Interfaces>
          <Interface>
            <IfIndex>3</IfIndex>
            <Name>Twenty-FiveGigE1/0/2</Name>
            <AbbreviatedName>WGE1/0/2</AbbreviatedName>
            <PortIndex>3</PortIndex>
```

```
            <ifTypeExt>22</ifTypeExt>
            <ifType>6</ifType>
            <Description>Twenty-FiveGigE1/0/2 Interface</Description>
            <AdminStatus>2</AdminStatus>
            <OperStatus>2</OperStatus>
            <ConfigSpeed>0</ConfigSpeed>
            <ActualSpeed>100000</ActualSpeed>
            <ConfigDuplex>3</ConfigDuplex>
            <ActualDuplex>1</ActualDuplex>
          </Interface>
        </Interfaces>
      </Ifmgr>
    </top>
  </data>
</rpc-reply>
```

Example: Retrieving non-default configuration data

Network configuration

Retrieve all non-default configuration data.

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 non-default configuration data.

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source>
      <running/>
    </source>
  </get-config>
</rpc>
```

Verifying the configuration

If the client receives the following text, the <get-config> operation is successful:

```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:web="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="100">
    <data>
        <top xmlns="http://www.hp.com/netconf/config:1.0">
            <Ifmgr>
                <Interfaces>
```

```
                    <Interface>
                        <IfIndex>1307</IfIndex>
                        <Shutdown>1</Shutdown>
                    </Interface>
                    <Interface>
                        <IfIndex>1308</IfIndex>
                        <Shutdown>1</Shutdown>
                    </Interface>
                    <Interface>
                        <IfIndex>1309</IfIndex>
                        <Shutdown>1</Shutdown>
                    </Interface>
                    <Interface>
                        <IfIndex>1311</IfIndex>
                            <VlanType>2</VlanType>
                    </Interface>
                    <Interface>
                        <IfIndex>1313</IfIndex>
                            <VlanType>2</VlanType>
                    </Interface>
                </Interfaces>
            </Ifmgr>
            <Syslog>
                <LogBuffer>
                    <BufferSize>120</BufferSize>
                </LogBuffer>
            </Syslog>
            <System>
                <Device>
                    <SysName>Sysname</SysName>
                    <TimeZone>
                        <Zone>+11:44</Zone>
                        <ZoneName>beijing</ZoneName>
                    </TimeZone>
                </Device>
            </System>
        </top>
    </data>
</rpc-reply>
```

Example: Retrieving syslog configuration data

Network configuration

Retrieve configuration data for the Syslog module.

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.hp.com/netconf/config:1.0">
        <Syslog/>
      </top>
    </filter>
  </get-config>
</rpc>
```

Verifying the configuration

If the client receives the following text, the <get-config> operation is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="100">
    <data>
        <top xmlns="http://www.hp.com/netconf/config:1.0">
            <Syslog>
                    <LogBuffer>
                        <BufferSize>120</BufferSize>
                    </LogBuffer>
            </Syslog>
        </top>
    </data>
</rpc-reply>
```

Example: Retrieving NETCONF session information

Network configuration

Get NETCONF session information.

Procedure

Enter XML view.

```
<Sysname> xml
```

Copy the following message to the client to exchange capabilities with the device:

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

```
    <capability>
            urn:ietf:params:netconf:base:1.0
    </capability>
  </capabilities>
</hello>
```

Copy the following message to the client to get the current NETCONF session information on the device:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-sessions/>
</rpc>
```

Verifying the configuration

If the client receives a message as follows, the operation is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="100">
    <get-sessions>
        <Session>
            <SessionID>1</SessionID>
            <Line>vty0</Line>
            <UserName></UserName>
            <Since>2017-01-07T00:24:57</Since>
            <LockHeld>false</LockHeld>
        </Session>
    </get-sessions>
</rpc-reply>
```

The output shows the following information:

- The session ID of an existing NETCONF session is 1.
- The login user type is vty0.
- The login time is 2017-01-07T00:24:57.
- The user does not hold the lock of the configuration.

Filtering data

About data filtering

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.

Restrictions and guidelines for data filtering

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

Table-based filtering

About this task

The namespace is **http://www.hp.com/netconf/base:1.0**. The attribute name is **filter**. For information about the support for table-based match, see the NETCONF XML API references.

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:

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

Restrictions and guidelines

To use table-based filtering, specify a match criterion for the **filter** row attribute.

Column-based filtering

About this task

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 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.hp.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">
  <get>
    <filter type="subtree">
      <top
xmlns="http://www.hp.com/netconf/data:1.0"xmlns:data="http://www.hp.com/netconf/data:
1.0">
        <Ifmgr>
          <Interfaces>
            <Interface data:AdminStatus="1"/>
          </Interfaces>
        </Ifmgr>
      </top>
    </filter>
  </get>
</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.

The supported data types include integer, date and time, character string, IPv4 address, IPv4 mask, IPv6 address, MAC address, OID, and time zone.

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:hp="http://www.hp.com/netconf/base:1.0">
  <get-config>
    <source>
      <running/>
    </source>
    <filter type="subtree">
      <top xmlns="http://www.hp.com/netconf/config:1.0">
        <Ifmgr>
          <Interfaces>
            <Interface>
              <Description hp:regExp="^[A-Z]*$"/>
            </Interface>
          </Interfaces>
        </Ifmgr>
      </top>
    </filter>
```

```
  </get-config>
</rpc>
```

Conditional match filtering

To implement a complex data filtering with digits and character strings, you can add a **match** attribute for a specific element. Table 11 lists the conditional match operators.

Table 11 Conditional match operators

Operation	Operator	Remarks
More than	match="more:*value*"	More than the specified value. The supported data types include date, digit, and character string.
Less than	match="less:*value*"	Less than the specified value. The supported data types include date, digit, and character string.
Not less than	match="notLess:*value*"	Not less than the specified *value*. The supported data types include date, digit, and character string.
Not more than	match="notMore:*value*"	Not more than the specified *value*. The supported data types include date, digit, and character string.
Equal	match="equal:*value*"	Equal to the specified *value*. The supported data types include date, digit, character string, OID, and BOOL.
Not equal	match="notEqual:*value*"	Not equal to the specified *value*. The supported data types include date, digit, character string, OID, and BOOL.
Include	match="include:*string*"	Includes the specified *string*. The supported data types include only character string.
Not include	match="exclude:*string*"	Excludes the specified *string*. The supported data types include only character string.
Start with	match="startWith:*string*"	Starts with the specified *string*. The supported data types include character string and OID.
End with	match="endWith:*string*"	Ends with the specified *string.* The supported data types include only character string.

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

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:hp="http://www.hp.com/netconf/base:1.0">
  <get>
    <filter type="subtree">
      <top xmlns="http://www.hp.com/netconf/data:1.0">
        <Device>
          <ExtPhysicalEntities>
            <Entity>
              <CpuUsage hp:match="more:50"></CpuUsage>
            </Entity>
          </ExtPhysicalEntities>
        </Device>
      </top>
    </filter>
  </get>
</rpc>
```

Example: Filtering data with regular expression match

Network configuration

Retrieve all data including **Gig** in the **Description** column of the Interfaces table under the Ifmgr module.

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 **Gig** in the **Description** column of the Interfaces table under the Ifmgr module.

```
<?xml version="1.0"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:hp="http://www.hp.com/netconf/base:1.0">
  <get>
    <filter type="subtree">
      <top xmlns="http://www.hp.com/netconf/data:1.0">
        <Ifmgr>
          <Interfaces>
            <Interface>
              <Description hp:regExp="(Gig)+"/>
            </Interface>
          </Interfaces>
        </Ifmgr>
      </top>
    </filter>
  </get>
</rpc>
```

Verifying the configuration

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

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:hp="http://www.hp.com/netconf/base:1.0" message-id="100">
    <data>
        <top xmlns="http://www.hp.com/netconf/data:1.0">
            <Ifmgr>
                <Interfaces>
                    <Interface>
                        <IfIndex>2681</IfIndex>
                        <Description>Twenty-FiveGigE1/0/1 Interface</Description>
```

```
                    </Interface>
                    <Interface>
                        <IfIndex>2685</IfIndex>
                        <Description>Twenty-FiveGigE1/0/2 Interface</Description>
                    </Interface>
                    <Interface>
                        <IfIndex>2689</IfIndex>
                        <Description>Twenty-FiveGigE1/0/3 Interface</Description>
                    </Interface>
                <Interface>
            </Ifmgr>
        </top>
    </data>
</rpc-reply>
```

Example: Filtering data by conditional match

Network configuration

Retrieve data in the **Name** column with the ifindex value not less than 5000 in the Interfaces table under the Ifmgr module.

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:hp="http://www.hp.com/netconf/base:1.0">
  <get>
    <filter type="subtree">
      <top xmlns="http://www.hp.com/netconf/data:1.0">
        <Ifmgr>
          <Interfaces>
            <Interface>
              <IfIndex hp:match="notLess:5000"/>
              <Name/>
            </Interface>
          </Interfaces>
        </Ifmgr>
      </top>
    </filter>
```

```
  </get>
</rpc>
```

Verifying the configuration

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

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:hp="http://www.hp.com/netconf/base:1.0" message-id="100">
    <data>
        <top xmlns="http://www.hp.com/netconf/data:1.0">
            <Ifmgr>
                <Interfaces>
                    <Interface>
                        <IfIndex>7241</IfIndex>
                        <Name>NULL0</Name>
                    </Interface>
                </Interfaces>
            </Ifmgr>
        </top>
    </data>
</rpc-reply>
```

Locking or unlocking the running configuration

About configuration locking and unlocking

Multiple methods are available for configuring the device, such as CLI, NETCONF, and SNMP. Before configuring, managing, or troubleshooting the device, you can lock the configuration to prevent other users from changing the device configuration. After you lock the configuration, only you can perform <edit-config> operations to change the configuration or unlock the configuration. Other users can only read the configuration.

If you close your NETCONF session, the system unlocks the configuration. You can also manually unlock the configuration.

Restrictions and guidelines for configuration locking and unlocking

The <lock> operation locks the running configuration of the device. You cannot use it to lock the configuration for a specific module.

Locking the running configuration

Copy the following text to the client to lock the running configuration:

```
<?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>
```

If the <lock> operation succeeds, the device returns a response in the following format:

```
<?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 running configuration

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

```
<?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>
```

If the <unlock> operation succeeds, the device returns a response in the following format:

```
<?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: Locking the running configuration

Network configuration

Lock the device configuration so other users cannot change the device 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>
```

Lock the configuration.

```
<?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>
```

Verifying the configuration

If the client receives the following response, the <lock> operation is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
    <rpc-reply message-id="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 version="1.0" encoding="UTF-8"?>
<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 because the NETCONF lock is held by another
session.</error-message>
  <error-info>
    <session-id>1</session-id>
  </error-info>
  </rpc-error>
</rpc-reply>
```

The output shows that the <lock> operation failed. The client with session ID 1 is holding the lock,

Modifying the configuration

About the <edit-config> operation

The <edit-config> operation includes the following operations: merge, create, replace, remove, delete, default-operation, error-option, test-option, and incremental. For more information about the operations, see "Supported NETCONF operations."

Procedure

Copy the following text to perform the <edit-config> operation:

```
<?xml version="1.0"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target><running></running></target>
    <error-option>
       error-option
    </error-option>
    <config>
      <top xmlns="http://www.hp.com/netconf/config:1.0">
        Specify the module name, submodule name, table name, and column name
      </top>
```

```
    </config>
  </edit-config>
</rpc>
```

The <error-option> element indicates the action to be taken in response to an error that occurs during the operation. It has the following values:

Value	Description
stop-on-error	Stops the <edit-config> operation.
continue-on-error	Continues the <edit-config> operation.
rollback-on-error	Rolls back the configuration to the configuration before the <edit-config> operation was performed. By default, an <edit-config> operation cannot be performed while the device is rolling back the configuration. If the rollback time exceeds the maximum time that the client can wait, the client determines that the <edit-config> operation has failed and performs the operation again. Because the previous rollback is not completed, the operation triggers another rollback. If this process repeats itself, CPU and memory resources will be exhausted and the device will reboot. To allow an <edit-config> operation to be performed during a configuration rollback, perform an <action> operation to change the value of the **DisableEditConfigWhenRollback** attribute to **false**.

If the <edit-config> operation succeeds, the device returns a response in the following format:

```
<?xml version="1.0">
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

You can also perform the <get> operation to verify that the current element value is the same as the value specified through the <edit-config> operation.

Example: Modifying the configuration

Network configuration

Change the log buffer size for the Syslog module to 512.

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.

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:web="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
```

```
    </target>
    <config>
      <top xmlns="http://www.hp.com/netconf/config:1.0" web:operation="merge">
        <Syslog>
          <LogBuffer>
            <BufferSize>512</BufferSize>
          </LogBuffer>
        </Syslog>
      </top>
    </config>
  </edit-config>
</rpc>
```

Verifying the configuration

If the client receives the following text, the <edit-config> operation is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <ok/>
</rpc-reply>
```

Saving the running configuration

About the <save> operation

A <save> operation saves the running configuration to a configuration file and specifies the file as the main next-startup configuration file.

Restrictions and guidelines

The <save> operation is resource intensive. Do not perform this operation when system resources are heavily occupied.

Procedure

Copy the following text to the client:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <save OverWrite="false" Binary-only="false">
    <file>Configuration file name</file>
  </save>
</rpc>
```

Item	Description
file	Specifies a .cfg configuration file by its name. The name must start with the storage medium name. If you specify the file column, a file name is required. If the **Binary-only** attribute is **false**, the device saves the running configuration to both the text and binary configuration files. • If the specified .cfg file does not exist, the device creates the binary and text configuration files to save the running configuration. • If you do not specify the file column, the device saves the running configuration to the text and binary next-startup configuration files.
OverWrite	Determines whether to overwrite the specified file if the file already exists. The following values are available: • **true**—Overwrite the file. • **false**—Do not overwrite the file. The running configuration cannot be saved, and the system displays an error message. The default value is **true**.

Binary-only	Determines whether to save the running configuration only to the binary configuration file. The following values are available:
	• **true**—Save the running configuration only to the binary configuration file.
	  ◦ If **file** specifies a nonexistent file, the <save> operation fails.
	  ◦ If you do not specify the file column, the device identifies whether the main next-startup configuration file is specified. If yes, the device saves the running configuration to the corresponding binary file. If not, the <save> operation fails.
	• **false**—Save the running configuration to both the text and binary configuration files. For more information, see the description for the file column in this table.
	Saving the running configuration to both the text and binary configuration files requires more time.
	The default value is **false**.

If the <save> operation succeeds, the device returns a response in the following format:

```
<?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: Saving the running configuration

Network configuration

Save the running configuration to the **config.cfg** file.

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>
```

Save the running configuration of the device to the **config.cfg** file.

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <save>
    <file>config.cfg</file>
  </save>
</rpc>
```

Verifying the configuration

If the client receives the following response, the <save> operation is successful:

```
<?xml version="1.0" encoding="UTF-8"?>
    <rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <ok/>
</rpc-reply>
```

Loading the configuration

About the <load> operation

The <load> operation merges the configuration from a configuration file into the running configuration as follows:

- Loads settings that do not exist in the running configuration.
- Overwrites settings that already exist in the running configuration.

Restrictions and guidelines

When you perform a <load> operation, follow these restrictions and guidelines:

- The <load> operation is resource intensive. Do not perform this operation when the system resources are heavily occupied.
- Some settings in a configuration file might conflict with the existing settings. For the settings in the file to take effect, delete the existing conflicting settings, and then load the configuration file.

Procedure

Copy the following text to the client to load a configuration file for the device:

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

The configuration file name must start with the storage media name and end with the **.cfg** extension.

If the <load> operation succeeds, the device returns a response in the following format:

```
<?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

Restrictions and guidelines

The <rollback> operation is resource intensive. Do not perform this operation when the system resources are heavily occupied.

By default, an <edit-config> operation cannot be performed while the device is rolling back the configuration. To allow an <edit-config> operation to be performed during a configuration rollback, perform an <action> operation to change the value of the **DisableEditConfigWhenRollback** attribute to **false**.

Rolling back the configuration based on a configuration file

Copy the following text to the client to roll back the running configuration to the configuration in a configuration file:

```
<?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>
```

If the <rollback> operation succeeds, the device returns a response in the following format:

```
<?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

About this task

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.

Restrictions and guidelines

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

Procedure

1. Lock the running configuration. For more information, see "Locking or unlocking the running configuration."
2. Enable configuration rollback based on a rollback point.

   # Copy the following text to the client to perform a <save-point>/<begin> operation:

```
<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>
```

Item	Description
confirm-timeout	Specifies the rollback idle timeout time in the range of 1 to 65535 seconds. The default is 600 seconds. This item is optional.

If the <save-point/begin> operation succeeds, the device returns a response in the following format:

```
<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>
```

3. Modify the running configuration. For more information, see "Modifying the configuration."
4. Mark the rollback point.

   The system supports a maximum of 50 rollback points. If the limit is reached, specify the **force** attribute for the <save-point>/<commit> operation to overwrite the earliest rollback point.

   # Copy the following text to the client to perform a <save-point>/<commit> operation:

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <save-point>
    <commit>
      <label>SUPPORT VLAN<label>
      <comment>vlan 1 to 100 and interfaces.</comment>
     </commit>
  </save-point>
</rpc>
```

   The <label> and <comment> elements are optional.

   If the <save-point>/<commit> operation succeeds, the device returns a response in the following format:

```
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <save-point>
       <commit>
          <commit-id>2</commit-id>
       </commit>
    </save-point>
  </data>
</rpc-reply>
```

5. Retrieve the rollback point configuration records.

   The following text shows the message format for a <save-point/get-commits> request:

```
<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 the <commit-id/>, <commit-index/>, or <commit-label/> element to retrieve the specified rollback point configuration records. If no element is specified, the operation retrieves records for all rollback point settings.

Copy the following text to the client to perform a <save-point>/<get-commits> operation:

```
<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>
```

If the <save-point/get-commits> operation succeeds, the device returns a response in the following format:

```
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <data>
      <save-point>
         <commit-information>
           <CommitID>2</CommitID>
           <TimeStamp>Sun Jan 1 11:30:28 2017</TimeStamp>
           <UserName>test</UserName>
           <Label>SUPPORT VLAN</Label>
         </commit-information>
    </save-point>
  </data>
</rpc-reply>
```

6. Retrieve the configuration data corresponding to a rollback point.

The following text shows the message format for a <save-point>/<get-commit-information> request:

```
<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 following elements: <commit-id/>, <commit-index/>, and <commit-label/>. The <compare-information> element is optional.

Item	Description
commit-id	Uniquely identifies a rollback point.
commit-index	Specifies 50 most recently configured rollback points. The value of 0 indicates the most recently configured one and 49 indicates the earliest configured one.
commit-label	Specifies a unique label for a rollback point.
get-commit-information	Retrieves the configuration data corresponding to the most recently configured rollback point.

Copy the following text to the client to perform a <save-point>/<get-commit-information> operation:

```
<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>
```

If the <save-point/get-commit-information> operation succeeds, the device returns a response in the following format:

```
<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>
```

7. Roll back the configuration based on a rollback point.

The configuration can also be automatically rolled back based on the most recently configured rollback point when the NETCONF session idle timer expires.

Copy the following text to the client to perform a <save-point>/<rollback> operation:

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

```
      <commit-label/>
    </rollback>
  </save-point>
</rpc>
```

Specify one of the following elements: <commit-id/>, <commit-index/>, and <commit-label/>. If no element is specified, the operation rolls back configuration based on the most recently configured rollback point.

Item	Description
commit-id	Uniquely identifies a rollback point.
commit-index	Specifies 50 most recently configured rollback points. The value of 0 indicates the most recently configured one and 49 indicates the earliest configured one.
commit-label	Specifies the unique label of a rollback point.

If the <save-point/rollback> operation succeeds, the device returns a response in the following format:

```
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok></ok>
</rpc-reply>
```

8. End the rollback configuration.

   # Copy the following text to the client to perform a <save-point>/<end> operation:

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

   If the <save-point/end> operation succeeds, the device returns a response in the following format:

```
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <ok/>
</rpc-reply>
```

9. Unlock the configuration. For more information, see "Locking or unlocking the running configuration."

Enabling preprovisioning

About this task

The <config-provisioned> operation enables preprovisioning.

- With preprovisioning disabled, the configuration for a member device or subcard is lost if the following sequence of events occur:
  a. The member device leaves the IRF fabric or the subcard goes offline.
  b. You save the running configuration and reboot the IRF fabric.

  If the member device joins the IRF fabric or the subcard comes online again, you must reconfigure the member device or subcard.
- With preprovisioning enabled, you can view and modify the configuration for a member device or subcard after the member device leaves the IRF fabric or the subcard goes offline. If you save the running configuration and reboot the IRF fabric, the configuration for the member

device or subcard is still retained. After the member device joins the IRF fabric or the subcard comes online again, the system applies the retained configuration to the member device or subcard. You do not need to reconfigure the member device or subcard.

Restrictions and guidelines

To view or modify the configuration for an offline member device or subcard, you can use only CLI commands.

Only the following commands support preprovisioning:

- Commands in the interface view of a member device or subcard.
- Commands in slot view.
- Command **qos traffic-counter**.

Only member devices and subcards in Normal state support preprovisioning.

Procedure

Copy the following text to the client to enable preprovisioning:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <config-provisioned>
  </config-provisioned>
</rpc>
```

If preprovisioning is successfully enabled, the device returns a response in the following format:

```
<?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>
```

Performing CLI operations through NETCONF

About CLI operations through NETCONF

You can enclose command lines in XML messages to configure the device.

Restrictions and guidelines

Performing CLI operations through NETCONF is resource intensive. As a best practice, do not perform the following tasks:

- Enclose multiple command lines in one XML message.
- Use NETCONF to perform a CLI operation when other users are performing NETCONF CLI operations.

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.

If the CLI operation succeeds, the device returns a response in the following format:

```
<?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>
```

Example: Performing CLI operations

Network configuration

Send the **`display vlan`** command to the device.

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>
```

Copy the following text to the client to execute the **`display vlan`** command:

```
<?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 version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <CLI>
    <Execution><![CDATA[
<Sysname>display vlan
```

```
 Total VLANs: 1
 The VLANs include:
 1(default)
   ]]>
   </Execution>
  </CLI>
</rpc-reply>
```

Subscribing to events

About event subscription

When an event takes place on the device, the device sends information about the event to NETCONF clients that have subscribed to the event.

Restrictions and guidelines

Event subscription is not supported for NETCONF over SOAP sessions.

A subscription takes effect only on the current session. It is canceled when the session is terminated.

If you do not specify the event stream to be subscribed to, the device sends syslog event notifications to the NETCONF client.

Subscribing to syslog events

Copy the following message to the client to complete the subscription:

```
<?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.hp.com/netconf/event:1.0">
          <Code>code</Code>
            <Group>group</Group>
              <Severity>severity</Severity>
        </event>
      </filter>
      <startTime>start-time</startTime>
      <stopTime>stop-time</stopTime>
   </create-subscription>
</rpc>
```

Item	Description
stream	Specifies the event stream. The name for the syslog event stream is **NETCONF**.
event	Specifies the event. For information about the events to which you can subscribe, see the system log message references for the device.
code	Specifies the mnemonic symbol of the log message.

Item	Description
group	Specifies the module name of the log message.
severity	Specifies the severity level of the log message.
start-time	Specifies the start time of the subscription.
stop-time	Specifies the end time of the subscription.

If the subscription succeeds, the device returns a response in the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
<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 version="1.0" encoding="UTF-8"?>
<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.

Subscribing to events monitored by NETCONF

After you subscribe to events as described in this section, NETCONF regularly polls the subscribed events and sends the events that match the subscription condition to the NETCONF client.

Copy the following message to the client to complete the subscription:

```
<?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_MONITOR_EXTENSION</stream>
  <filter>
    <NetconfMonitor xmlns='http://www.hp.com/netconf/monitor:1.0'>
      <XPath>XPath</XPath>
      <Interval>interval</Interval>
      <ColumnConditions>
        <ColumnCondition>
          <ColumnName>ColumnName</ColumnName>
          <ColumnValue>ColumnValue</ColumnValue>
          <ColumnCondition>ColumnCondition</ColumnCondition>
        </ColumnCondition>
      </ColumnConditions>
      <MustIncludeResultColumns>
        <ColumnName>columnName</ColumnName>
      </MustIncludeResultColumns>
```

```
    </NetconfMonitor>
  </filter>
<startTime>start-time</startTime>
<stopTime>stop-time</stopTime>
</create-subscription>
</rpc>
```

Item	Description
stream	Specifies the event stream. The name for the event stream is **NETCONF_MONITOR_EXTENSION**.
NetconfMonitor	Specifies the filtering information for the event.
XPath	Specifies the path of the event in the format of *ModuleName*[/*SubmoduleName*]/*TableName*.
interval	Specifies the interval for NETCONF to obtain events that matches the subscription condition. The value range is 1 to 4294967 seconds. The default value is 300 seconds.
ColumnName	Specifies the name of a column in the format of [*GroupName*.]*ColumnName.*
ColumnValue	Specifies the baseline value.
ColumnCondition	Specifies the operator: • **more**. • **less**. • **notLess**. • **notMore**. • **equal**. • **notEqual**. • **include**. • **exclude**. • **startWith**. • **endWith**. Choose an operator according to the type of the baseline value.
start-time	Specifies the start time of the subscription.
stop-time	Specifies the end time of the subscription.

If the subscription succeeds, the device returns a response in the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="100" xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
    <ok/>
</rpc-reply>
```

Subscribing to events reported by modules

After you subscribe to events as described in this section, the specified modules report subscribed events to NETCONF. NETCONF sends the events to the NETCONF client.

Copy the following message to the client to complete the subscription:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:xs="http://www.hp.com/netconf/base:1.0">
```

```
<create-subscription xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<stream>XXX_STREAM</stream>
   <filter type="subtree">
<event xmlns="http://www.hp.com/netconf/event:1.0/xxx-features-list-name:1.0">
       <ColumnName xs:condition="Condition">value</ColumnName>
</event>
</filter>
<startTime>start-time</startTime>
<stopTime>stop-time</stopTime>
</create-subscription>
</rpc>
```

Item	Description
stream	Specifies the event stream. Supported event streams vary by device model.
event	Specifies the event name. An event stream includes multiple events. The events use the same namespaces as the event stream.
ColumnName	Specifies the name of a column.
ColumnCondition	Specifies the operator: • **more**. • **less**. • **notLess**. • **notMore**. • **equal**. • **notEqual**. • **include**. • **exclude**. • **startWith**. • **endWith**. Choose an operator according to the type of the baseline value.
value	Specifies the baseline value for the column.
start-time	Specifies the start time of the subscription.
stop-time	Specifies the end time of the subscription.

If the subscription succeeds, the device returns a response in the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="100" xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
    <ok/>
</rpc-reply>
```

Canceling an event subscription

Copy the following message to the client to cancel a subscription:

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <cancel-subscription xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <stream>XXX_STREAM</stream>
  </cancel-subscription>
```

```
</rpc>
```

Item	Description
stream	Specifies the event stream.

If the cancelation succeeds, the device returns a response in the following format:

```
<?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 the subscription to be canceled does not exist, the device returns an error message in the following format:

```
<?xml version="1.0" encoding="UTF-8"?>
<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">The subscription stream to be canceled doesn't exist: Stream name=XXX_STREAM.</error-message>
</rpc-error>
</rpc-reply>
```

Example: Subscribing to syslog events

Network configuration

Configure a client to subscribe to syslog events with no time limitation. After the subscription, all events on the device are sent to the client before the session between the device and client is terminated.

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>
```

Subscribe to syslog 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>
```

When another client (192.168.100.130) logs in to the device, the device sends a notification to the client that has subscribed to all events:

```
<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <eventTime>2011-01-04T12:30:52</eventTime>
    <event xmlns="http://www.hp.com/netconf/event:1.0">
        <Group>SHELL</Group>
        <Code>SHELL_LOGIN</Code>
        <Slot>1</Slot>
        <Severity>Notification</Severity>
        <context>VTY logged in from 192.168.100.130.</context>
    </event>
</notification>
```

Terminating NETCONF sessions

About NETCONF session termination

NETCONF allows one client to terminate the NETCONF sessions of other clients. A client whose session is terminated returns to user view.

Procedure

Copy the following message to the client to terminate a 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>
```

If the <kill-session> operation succeeds, the device returns a response in the following format:

```
<?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: Terminating another NETCONF session

Network configuration

The user whose session's ID is 1 terminates the session with session ID 2.

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>
```

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>
```

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

Restrictions and guidelines

Before returning from XML view to the CLI, you must first complete capability exchange between the device and the client.

Procedure

Copy the following text to the client to return from XML view to the CLI:

```
<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>
```

Supported NETCONF operations

This chapter describes NETCONF operations available with Comware 7.

action

Usage guidelines

This operation issues actions for non-default settings, for example, reset action.

XML example

Clear statistics information for all interfaces.

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <action>
    <top xmlns="http://www.hp.com/netconf/action:1.0">
      <Ifmgr>
            <ClearAllIfStatistics>
                <Clear>
                </Clear>
        </ClearAllIfStatistics>
      </Ifmgr>
    </top>
  </action>
</rpc>
```

CLI

Usage guidelines

This operation executes CLI commands.

A request message encloses commands in the <CLI> element. A response message encloses the command output in the <CLI> element.

You can use the following elements to execute commands:

- **Execution**—Executes commands in user view.
- **Configuration**—Executes commands in system view. To execute commands in a lower-level view of the system view, use the <Configuration> element to enter the view first.

  To use this element, include the **exec-use-channel** attribute and specify a value for the attribute:

  - **false**—Executes commands without using a channel.
  - **true**—Executes commands by using a temporary channel. The channel is automatically closed after the execution.
  - **persist**—Executes commands by using the persistent channel for the session.

    To use the persistent channel, first perform an <Open-channel> operation to open the persistent channel. If you do not do so, the system will automatically open the persistent channel.

    After using the persistent channel, perform a <Close-channel> operation to close the channel and return to system view. If you do not perform an <Open-channel> operation, the system stays in the view and will execute subsequent commands in the view.

You can also specify the **error-when-rollback** attribute in the <Configuration> element to indicate whether CLI operations are allowed during a configuration error-triggered configuration rollback. This attribute takes effect only if the value of the <error-option> element in <edit-config> operations is set to **rollback-on-error**. It has the following values:

- **true**—Rejects CLI operation requests and returns error messages.
- **false (the default)**—Allows CLI operations.

For CLI operations to be correctly performed, set the value of the **error-when-rollback** attribute to **true**.

A NETCONF session supports only one persistent channel and but supports multiple temporary channels.

NETCONF does not support executing interactive commands.

You cannot execute the **quit** command by using a channel to exit user view.

XML example

Execute the **vlan 3** command in system view without using a channel.

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <CLI>
     <Configuration exec-use-channel="false" error-when-rollback="true">vlan
3</Configuration>
  </CLI>
</rpc>
```

close-session

Usage guidelines

This operation terminates the current NETCONF session, unlock the configuration, and release the resources (for example, memory) used by the session. After this operation, you exit the XML view.

XML example

Terminate the current NETCONF session.

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<close-session/>
</rpc>
```

edit-config: create

Usage guidelines

This operation creates target configuration items.

To use the **create** attribute in an <edit-config> operation, you must specify the target configuration item.

- If the table supports creating a target configuration item and the item does not exist, the operation creates the item and configures the item.
- If the specified item already exists, a data-exist error message is returned.

XML example

Set the buffer size to 120.

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
```

```
    <target>
      <running/>
    </target>
    <config>
      <top xmlns="http://www.hp.com/netconf/config:1.0">
        <Syslog xmlns="http://www.hp.com/netconf/config:1.0" xc:operation="create">
           <LogBuffer>
             <BufferSize>120</BufferSize>
          </LogBuffer>
        </Syslog>
      </top>
    </config>
  </edit-config>
</rpc>
```

edit-config: delete

Usage guidelines

This operation deletes the specified configuration.

- If the specified target has only the table index, the operation removes all configuration of the specified target, and the target itself.
- If the specified target has the table index and configuration data, the operation removes the specified configuration data of this target.
- If the specified target does not exist, an error message is returned, showing that the target does not exist.

XML example

The syntax is the same as the edit-config message with the **create** attribute. Change the operation attribute from **create** to **delete**.

edit-config: merge

Usage guidelines

This operation commits target configuration items to the running configuration.

To use the **merge** attribute in an <edit-config> operation, you must specify the target configuration item (on a specific level):

- If the specified item exists, the operation directly updates the setting for the item.
- If the specified item does not exist, the operation creates the item and configures the item.
- If the specified item does not exist and it cannot be created, an error message is returned.

XML example

The XML data format is the same as the edit-config message with the **create** attribute. Change the operation attribute from **create** to **merge**.

edit-config: remove

Usage guidelines

This operation removes the specified configuration.

- If the specified target has only the table index, the operation removes all configuration of the specified target, and the target itself.
- If the specified target has the table index and configuration data, the operation removes the specified configuration data of this target.
- If the specified target does not exist, or the XML message does not specify any targets, a success message is returned.

XML example

The syntax is the same as the edit-config message with the **create** attribute. Change the operation attribute from **create** to **remove**.

edit-config: replace

Usage guidelines

This operation replaces the specified configuration.

- 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, the operation creates the target and then applies the configuration.
- 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.

XML example

The syntax is the same as the edit-config message with the **create** attribute. Change the operation attribute from **create** to **replace**.

edit-config: *test-option*

Usage guidelines

This operation determines whether to commit a configuration item in an <edit-configure> operation. The <test-option> element has one of the following values:

- **test-then-set**—Performs a syntax check, and commits an item if the item passes the check. If the item fails the check, the item is not committed. This is the default test-option value.
- **set**—Commits the item without performing a syntax check.
- **test-only**—Performs only a syntax check. If the item passes the check, a success message is returned. Otherwise, an error message is returned.

XML example

Test the configuration for an interface.

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <test-option>test-only</test-option>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
      <top xmlns="http://www.hp.com/netconf/config:1.0">
        <Ifmgr xc:operation="merge">
          <Interfaces>
            <Interface>
```

```
            <IfIndex>262</IfIndex>
            <Description>222</Description>
              <ConfigSpeed>2</ConfigSpeed>
              <ConfigDuplex>1</ConfigDuplex>
             </Interface>
          </Interfaces>
        </Ifmgr>
      </top>
    </config>
  </edit-config>
</rpc>
```

edit-config: default-operation

Usage guidelines

This operation modifies the running configuration of the device by using the default operation method.

NETCONF uses one of the following operation attributes to modify configuration: **merge**, **create**, **delete**, and **replace** If you do not specify an operation attribute for an edit-config message, NETCONF uses the default operation method. Your setting of the value for the <default-operation> element takes effect only once. If you do not specify an operation attribute or the default operation method for an <edit-config> message, **merge** always applies.

The <default-operation> element has the following values:

- **merge**—Default value for the <default-operation> element.
- **replace**—Value used when the operation attribute is not specified and the default operation method is specified as **replace**.
- **none**—Value used when the operation attribute is not specified and the default operation method is specified as **none**. If this value is specified, the <edit-config> operation is used only for schema verification rather than issuing a configuration. If the schema verification is passed, a successful message is returned. Otherwise, an error message is returned.

XML example

Issue an empty operation for schema verification purposes.

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <default-operation>none</default-operation>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
      <top xmlns="http://www.hp.com/netconf/config:1.0">
        <Ifmgr>
          <Interfaces>
            <Interface>
              <IfIndex>262</IfIndex>
              <Description>222222</Description>
            </Interface>
          </Interfaces>
        </Ifmgr>
```

```
      </top>
    </config>
  </edit-config>
</rpc>
```

edit-config: error-option

Usage guidelines

This operation determines the action to take in case of a configuration error.

The <error-option> element has the following values:

- **stop-on-error**—Stops the operation and returns an error message. This is the default error-option value.
- **continue-on-error**—Continues the operation and returns an error message.
- **rollback-on-error**—Rolls back the configuration.

XML example

Issue the configuration for two interfaces with the <error-option> element value as **continue-on-error**.

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>   <error-option>continue-on-error</error-option>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
      <top xmlns="http://www.hp.com/netconf/config:1.0">
        <Ifmgr xc:operation="merge">
          <Interfaces>
            <Interface>
              <IfIndex>262</IfIndex>
              <Description>222</Description>
                <ConfigSpeed>1024</ConfigSpeed>
                <ConfigDuplex>1</ConfigDuplex>
            </Interface>
            <Interface>
              <IfIndex>263</IfIndex>
              <Description>333</Description>
                <ConfigSpeed>1024</ConfigSpeed>
                <ConfigDuplex>1</ConfigDuplex>
            </Interface>
          </Interfaces>
        </Ifmgr>
      </top>
    </config>
  </edit-config>
</rpc>
```

edit-config: incremental

Usage guidelines

This operation 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 the <replace> operation.

Support for the **incremental** attribute varies by module. For more information, see NETCONF XML API documents.

XML example

Add VLANs 1 through 10 to an untagged VLAN list that has untagged VLANs 12 through 15.

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:hp="http://www.hp.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.hp.com/netconf/config:1.0">
        <VLAN xc:operation="merge">
          <HybridInterfaces>
            <Interface>
              <IfIndex>262</IfIndex>
              <UntaggedVlanList  hp:incremental="true">1-10</UntaggedVlanList>
               </Interface>
          </HybridInterfaces>
        </VLAN>
      </top>
    </config>
  </edit-config>
</rpc>
```

get

Usage guidelines

This operation retrieves device configuration and state information.

XML example

Retrieve device configuration and state information for the Syslog module.

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:xc="http://www.hp.com/netconf/base:1.0">
  <get>
    <filter type="subtree">
      <top xmlns="http://www.hp.com/netconf/data:1.0">
        <Syslog>
        </Syslog>
      </top>
    </filter>
```

```
  </get>
</rpc>
```

get-bulk

Usage guidelines

This operation 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.

XML example

Retrieve device configuration and state information for all interfaces.

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-bulk>
    <filter type="subtree">
      <top xmlns="http://www.hp.com/netconf/data:1.0">
        <Ifmgr>
          <Interfaces xc:count="5" xmlns:xc="http://www.hp.com/netconf/base:1.0">
            <Interface/>
          </Interfaces>
        </Ifmgr>
      </top>
    </filter>
  </get-bulk>
</rpc>
```

get-bulk-config

Usage guidelines

This operation retrieves a number of non-default configuration data entries starting from the data entry next to the one with the specified index.

XML example

Retrieve non-default configuration for all interfaces.

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-bulk-config>
    <source>
      <running/>
    </source>
    <filter type="subtree">
      <top xmlns="http://www.hp.com/netconf/config:1.0">
        <Ifmgr>
        </Ifmgr>
      </top>
    </filter>
  </get-bulk-config>
</rpc>
```

get-config

Usage guidelines

This operation retrieves non-default configuration data. If no non-default configuration data exists, the device returns a response with empty data.

XML example

Retrieve non-default configuration data for the interface table.

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

get-sessions

Usage guidelines

This operation retrieves information about all NETCONF sessions in the system. You cannot specify a session ID to retrieve information about a specific NETCONF session.

XML example

Retrieve information about all NETCONF sessions in the system.

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<get-sessions/>
</rpc>
```

kill-session

Usage guidelines

This operation terminates the NETCONF session for another user. This operation cannot terminate the NETCONF session for the current user.

XML example

Terminate the NETCONF session with session ID 1.

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <kill-session>
    <session-id>1</session-id>
```

```
  </kill-session>
</rpc>
```

load

Usage guidelines

This operation loads the configuration. After the device finishes a <load> operation, the configuration in the specified file is merged into the running configuration of the device.

XML example

Merge the configuration in file **a1.cfg** to the running configuration of the device.

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <load>
    <file>a1.cfg</file>
  </load>
</rpc>
```

lock

Usage guidelines

This operation locks the configuration. After the configuration is locked, you cannot perform <edit-config> operations. Other operations are allowed.

After a user locks the configuration, other users cannot use NETCONF or any other configuration methods such as CLI and SNMP to configure the device.

XML example

Lock the configuration.

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
 <lock>
    <target>
        <running/>
    </target>
</lock>
</rpc>
```

rollback

Usage guidelines

This operation 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.

XML example

Roll back the running configuration to the configuration in file **1A.cfg**.

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <rollback>
    <file>1A.cfg</file>
  </rollback>
</rpc>
```

save

Usage guidelines

This operation saves the running configuration. You can use the <file> element to specify a file for saving the configuration. If the text does not include the file column, the running configuration is automatically saved to the main next-startup configuration file.

The **OverWrite** attribute determines whether the running configuration overwrites the original configuration file when the specified file already exists.

The **Binary-only** attribute determines whether to save the running configuration only to the binary configuration file.

XML example

Save the running configuration to file **test.cfg**.

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <save OverWrite="false" Binary-only="true">
    <file>test.cfg</file>
  </save>
</rpc>
```

unlock

Usage guidelines

This operation unlocks the configuration, so other users can configure the device.

Terminating a NETCONF session automatically unlocks the configuration.

XML example

Unlock the configuration.

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<unlock>
    <target>
        <running/>
    </target>
</unlock>
</rpc>
```

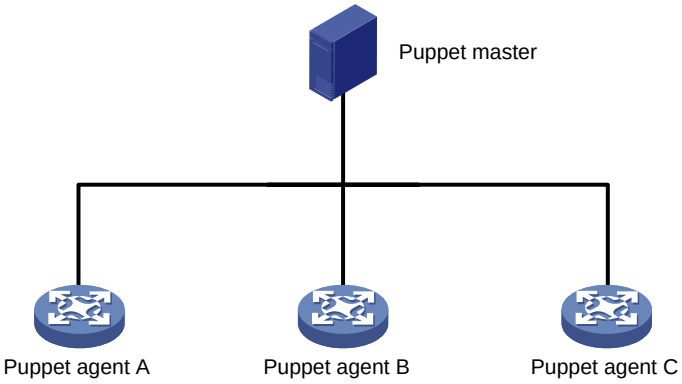
Configuring Puppet

About Puppet

Puppet is an open-source configuration management tool. It provides the Puppet language. You can use the Puppet language to create configuration manifests and save them to a server. You can then use the server for centralized configuration enforcement and management.

Puppet network framework

Figure 70 Puppet network framework



As shown in Figure 70, Puppet operates in a client/server network framework. In the framework, the Puppet master (server) stores configuration manifests for Puppet agents (clients). The Puppet agents establish SSL connections to the Puppet master to obtain their respective latest configurations.

Puppet master

The Puppet master runs the Puppet daemon process to listen to requests from Puppet agents, authenticates Puppet agents, and sends configurations to Puppet agents on demand.

For information about installing and configuring a Puppet master, see the official Puppet website at

https://puppetlabs.com/.

Puppet agent

HPE devices support Puppet 3.7.3 agent. The following is the communication process between a Puppet agent and the Puppet master:

1. The Puppet agent sends an authentication request to the Puppet master.
2. The Puppet agent checks with the Puppet master for the authentication result periodically (every two minutes by default). Once the Puppet agent passes the authentication, a connection is established to the Puppet master.
3. After the connection is established, the Puppet agent sends a request to the Puppet master periodically (every 30 minutes by default) to obtain the latest configuration.
4. After obtaining the latest configuration, the Puppet agent compares the configuration with its running configuration. If a difference exists, the Puppet agent overwrites its running configuration with the newly obtained configuration.
5. After overwriting the running configuration, the Puppet agent sends a feedback to the Puppet master.

Puppet resources

A Puppet resource is a unit of configuration. Puppet uses manifests to store resources.

Puppet manages types of resources. Each resource has a type, a title, and one or more attributes. Every attribute has a value. The value specifies the state desired for the resource. You can specify the state of a device by setting values for attributes regardless of how the device enters the state. The following resource example shows how to configure a device to create VLAN 2 and configure the description for VLAN 2.

```
netdev_vlan{'vlan2':
 ensure => undo_shutdown,
 id => 2,
 description => 'sales-private',
 require => Netdev_device['device'],
 }
```

The following are the resource type and title:

- **netdev_vlan**—Type of the resource. The **netdev_vlan** type resources are used for VLAN configuration.
- **vlan2**—Title of the resource. The title is the unique identifier of the resource.

The example contains the following attributes:

- **ensure**—Creates, modifies, or deletes a VLAN. To create a VLAN, set the attribute value to **undo_shutdown**. To delete a VLAN, set the attribute value to **shutdown**.
- **id**—Specifies a VLAN by its ID. In this example, VLAN 2 is specified.
- **description**—Configures the description for the VLAN. In this example, the description for VLAN 2 is **sales-private**.
- **require**—Indicates that the resource depends on another resource (specified by resource type and title). In this example, the resource depends on a **netdev_device** type resource titled **device**.

For information about resource types supported by Puppet, see "Puppet resources."

Restrictions and guidelines: Puppet configuration

The Puppet master cannot run a lower Puppet version than Puppet agents.

Prerequisites for Puppet

Before configuring Puppet on the device, complete the following tasks on the device:

- Enable NETCONF over SSH. The Puppet master sends configuration information to Puppet agents through NETCONF over SSH connections. For information about NETCONF over SSH, see "Configuring NETCONF."
- Configure SSH login. Puppet agents communicate with the Puppet master through SSH. For information about SSH login, see *Fundamentals Configuration Guide*.
- For successful communication, verify that the Puppet master and agents use the same system time. You can manually set the same system time for the Puppet master and agents or configure them to use a time synchronization protocol such as NTP. For more information about the time synchronization protocols, see "Configuring PTP" and "Configuring NTP."

Starting Puppet

Configuring resources

1. Install and configure the Puppet master.
2. Create manifests for Puppet agents on the Puppet master.

For more information, see the Puppet master installation and configuration guides.

Configuring a Puppet agent

1. Enter system view.

   system-view
2. Start Puppet.

   third-part-process start name puppet arg agent --certname=*certname* **--server=***server*

   By default, Puppet is shut down.

Parameter	Description
--certname=*certname*	Specifies the address of the Puppet agent.
--server=*server*	Specifies the address of the Puppet master.

After the Puppet process starts up, the Puppet agent sends an authentication request to the Puppet master. For more information about the **third-part-process start** command, see "Monitoring and maintaining processes".

Signing a certificate for the Puppet agent

To sign a certificate for the Puppet agent, execute the **puppet cert sign** *certname* command on the Puppet master.

After the signing operation succeeds, the Puppet agent establishes a connection to the Puppet master and requests configuration information from the Puppet master.

Shutting down Puppet on the device

Prerequisites

Execute the **display process all** command to identify the ID of the Puppet process. This command displays information about all processes on the device. Check the following fields:

- **THIRD**—This field displays **Y** for a third-party process.
- **PID**—Process ID.
- **COMMAND**—This field displays **puppet /opt/ruby/bin/pu** for the Puppet process.

Procedure

1. Enter system view.

   system-view
2. Shut down Puppet.

   third-part-process stop pid *pid-list*

For more information about the **third-part-process stop** command, see "Monitoring and maintaining processes".

Puppet configuration examples

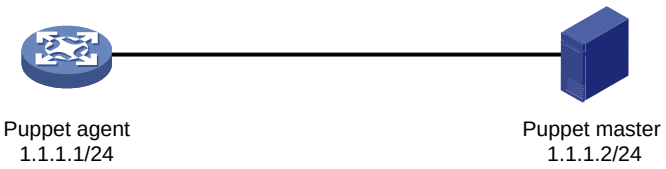
Example: Configuring Puppet

Network configuration

As shown in Figure 71, the device is connected to the Puppet master. Use Puppet to configure the device to perform the following operations:

- Set the SSH login username and password to **user** and **passwd**, respectively.
- Create VLAN 3.

Figure 71 Network diagram



Procedure

1. Configure SSH login and enable NETCONF over SSH on the device. (Details not shown.)
2. On the Puppet master, create the **modules/custom/manifests** directory in the **/etc/puppet/** directory for storing configuration manifests.
   ```
   $ mkdir -p /etc/puppet/modules/custom/manifests
   ```
3. Create configuration manifest **init.pp** in the **/etc/puppet/modules/custom/manifests** directory as follows:
   ```
   netdev_device{'device':
    ensure => undo_shutdown,
    username => 'user',
    password => 'passwd',
    ipaddr => '1.1.1.1',
    }
   netdev_vlan{'vlan3':
    ensure => undo_shutdown,
    id => 3,
    require => Netdev_device['device'],
    }
   ```
4. Start Puppet on the device.
   ```
   <PuppetAgent> system-view
   [PuppetAgent] third-part-process start name puppet arg agent --certname=1.1.1.1
   --server=1.1.1.2
   ```
5. Configure the Puppet master to authenticate the request from the Puppet agent.
   ```
   $ puppet cert sign 1.1.1.1
   ```

After passing the authentication, the Puppet agent requests the latest configuration for it from the Puppet master.

Puppet resources

netdev_device

Use this resource to specify the following items:

- Name for a Puppet agent.
- IP address, SSH username, and SSH password used by the agent to connect to a Puppet master.

Attributes

Table 12 Attributes for netdev_device

Attribute name	Description	Value type and restrictions
ensure	Establishes a NETCONF connection to the Puppet master or closes the connection.	Symbol: • **undo_shutdown**—Establishes a NETCONF connection to the Puppet master. • **shutdown**—Closes the NETCONF connection between the Puppet agent and the Puppet master. • **present**—Establishes a NETCONF connection to the Puppet master. • **absent**—Closes the NETCONF connection between the Puppet agent and the Puppet master.
hostname	Specifies the device name.	String, case sensitive. Length: 1 to 64 characters.
ipaddr	Specifies an IP address.	String, in dotted decimal notation.
username	Specifies the username for SSH login.	String, case sensitive. Length: 1 to 55 characters.
password	Specifies the password for SSH login.	String, case sensitive. Length and form requirements in non-FIPS mode: • 1 to 63 characters when in plaintext form. • 1 to 110 characters when in hashed form. • 1 to 117 characters when in encrypted form.

Resource example

Configure the device name as **PuppetAgent**. Specify the IP address, SSH username, and SSH password for the agent to connect to the Puppet master as 1.1.1.1, **user**, and **123456**, respectively.

```
netdev_device{'device':
 ensure => undo_shutdown,
 username => 'user',
 password => '123456',
 ipaddr => '1.1.1.1',
 hostname => 'PuppetAgent'
}
```

netdev_interface

Use this resource to configure attributes for an interface.

Attributes

Table 13 Attributes for netdev_interface

Attribute name	Description	Attribute type	Value type and restrictions
ifindex	Specifies an interface by its index.	Index	Unsigned integer.
ensure	Configures the attributes of the interface.	N/A	Symbol: • **undo_shutdown** • **present**.
description	Configures the description for the interface.	N/A	String, case sensitive. Length: 1 to 255 characters.
admin	Specifies the management state for the interface.	N/A	Symbol: • **up**—Brings up the interface. • **down**—Shuts down the interface.
speed	Specifies the interface rate.	N/A	Symbol: • **auto**—Autonegotiation. • **10m**—10 Mbps. • **100m**—100 Mbps. • **1g**—1 Gbps. • **10g**—10 Gbps. • **40g**—40 Gbps. • **100g**—100 Gbps.
duplex	Sets the duplex mode.	N/A	Symbol: • **full**—Full-duplex mode. • **half**—Half-duplex mode. • **auto**—Autonegotiation. This attribute applies only to Ethernet interfaces.
linktype	Sets the link type for the interface.	N/A	Symbol: • **access**—Sets the link type of the interface to **Access**. • **trunk**—Sets the link type of the interface to **Trunk**. • **hybrid**—Sets the link type of the interface to **Hybrid**. This attribute applies only to Layer 2 Ethernet interfaces.
portlayer	Sets the operation mode for the interface.	N/A	Symbol: • **bridge**—Layer 2 mode. • **route**—Layer 3 mode.

Attribute name	Description	Attribute type	Value type and restrictions
mtu	Sets the MTU permitted by the interface.	N/A	Unsigned integer in bytes. The value range depends on the interface type. This attribute applies only to Layer 3 Ethernet interface.

Resource example

Configure the following attributes for Ethernet interface 2:

- **Interface description**—puppet interface 2.
- **Management state**—Up.
- **Interface rate**—Autonegotiation.
- **Duplex mode**—Autonegotiation.
- **Link type**—Hybrid.
- **Operation mode**—Layer 2.
- **MTU**—1500 bytes.

```
netdev_interface{'ifindex2':
 ifindex => 2,
 ensure => undo_shutdown,
 description => 'puppet interface 2',
 admin => up,
 speed => auto,
 duplex => auto,
 linktype => hybrid,
 portlayer => bridge,
 mut => 1500,
 require => Netdev _device['device'],
}
```

netdev_l2_interface

Use this resource to configure the VLAN attributes for a Layer 2 Ethernet interface.

Attributes

Table 14 Attributes for netdev_l2_interface

Attribute name	Description	Attribute type	Value type and restrictions
ifindex	Specifies a Layer 2 Ethernet interface by its index.	Index	Unsigned integer.
ensure	Configures the attributes of the Layer 2 Ethernet interface.	N/A	Symbol: • **undo_shutdown** • **present**
pvid	Specifies the PVID for the interface.	N/A	Unsigned integer. Value range: 1 to 4094.

Attribute name	Description	Attribute type	Value type and restrictions
permit_vlan_list	Specifies the VLANs permitted by the interface.	N/A	String, a comma separated list of VLAN IDs or VLAN ID ranges, for example, 1,2,3,5-8,10-20. Value range for each VLAN ID: 1 to 4094. The string cannot end with a comma (,), hyphen (-), or space.
untagged_vlan_list	Specifies the VLANs from which the interface sends packets after removing VLAN tags.	N/A	String, a comma separated list of VLAN IDs or VLAN ID ranges, for example, 1,2,3,5-8,10-20. Value range for each VLAN ID: 1 to 4094. The string cannot end with a comma (,), hyphen (-), or space. A VLAN cannot be on the untagged list and the tagged list at the same time.
tagged_vlan_list	Specifies the VLANs from which the interface sends packets without removing VLAN tags.	N/A	String, a comma separated list of VLAN IDs or VLAN ID ranges, for example, 1,2,3,5-8,10-20. Value range for each VLAN ID: 1 to 4094. The string cannot end with a comma (,), hyphen (-), or space. A VLAN cannot be on the untagged list and the tagged list at the same time.

Resource example

Specify the PVID as 2 for interface 3, and configure the interface to permit packets from VLANs 1 through 6. Configure the interface to forward packets from VLANs 1 through 3 after removing VLAN tags and forward packets from VLANs 4 through 6 without removing VLAN tags.

```
netdev_l2_interface{'ifindex3':
 ifindex => 3,
 ensure => undo_shutdown,
 pvid => 2,
 permit_vlan_list => '1-6',
 untagged_vlan_list => '1-3',
 tagged_vlan_list => '4,6'
 require => Netdev _device['device'],
}
```

netdev_lagg

Use this resource to create, modify, or delete an aggregation group.

Attributes

Table 15 Attributes for netdev_lagg

Attribute name	Description	Attribute type	Value type and restrictions
group_id	Specifies an aggregation group ID.	Index	Unsigned integer. The value range for a Layer 2 aggregation group is 1 to 1024. The value range for a Layer 3 aggregation group is 16385 to 17408.
ensure	Creates, modifies, or deletes the aggregation group.	N/A	Symbol: • **present**—Creates or modifies the aggregation group. • **absent**—Deletes the aggregation group.
linkmode	Specifies the aggregation mode.	N/A	Symbol: • **static**—Static. • **dynamic**—Dynamic.
addports	Specifies the indexes of the interfaces that you want to add to the aggregation group.	N/A	String, a comma separated list of interface indexes or interface index ranges, for example, 1,2,3,5-8,10-20. The string cannot end with a comma (,), hyphen (-), or space. An interface index cannot be on the list of adding interfaces and the list of removing interfaces at the same time.
deleteports	Specifies the indexes of the interfaces that you want to remove from the aggregation group.	N/A	String, a comma separated list of interface indexes or interface index ranges, for example, 1,2,3,5-8,10-20. The string cannot end with a comma (,), hyphen (-), or space. An interface index cannot be on the list of adding interfaces and the list of removing interfaces at the same time.

Resource example

Add interfaces 1 and 2 to aggregation group 2, and remove interfaces 3 and 4 from the group.

```
netdev_lagg{ 'lagg2':
 group_id => 2,
 ensure => present,
 addports => '1,2',
 deleteports => '3,4',
 require => Netdev _device['device'],
}
```

netdev_vlan

Use this resource to create, modify, or delete a VLAN or configure the description for the VLAN.

Attributes

Table 16 Attributes for netdev_vlan

Attribute name	Description	Attribute type	Value type and restrictions
ensure	Creates, modifies, or deletes a VLAN.	N/A	Symbol: • **undo_shutdown**—Creates or modifies a VLAN. • **shutdown**—Deletes a VLAN. • **present**—Creates or modifies a VLAN. • **absent**—Deletes a VLAN.
id	Specifies the VLAN ID.	Index	Unsigned integer. Value range: 1 to 4094.
description	Configures the description for the VLAN.	N/A	String, case sensitive. Length: 1 to 255 characters.

Resource example

Create VLAN 2, and configure the description as **sales-private** for VLAN 2.

```
netdev_vlan{'vlan2':
 ensure => undo_shutdown,
 id => 2,
 description => 'sales-private',
 require => Netdev_device['device'],
}
```

netdev_vsi

Use this resource to create, modify, or delete a Virtual Switch Instance (VSI).

Attributes

Table 17 Attributes for netdev_vsi

Attribute name	Description	Attribute type	Value type and restrictions
vsiname	Specifies a VSI name.	Index	String, case sensitive. Length: 1 to 31 characters.
ensure	Creates, modifies, or deletes the VSI.	N/A	Symbol: • **present**—Creates or modifies the VSI. • **absent**—Deletes the VSI.
description	Configures the description for the VSI.	N/A	String, case sensitive. Length: 1 to 80 characters.

Resource example

Create the VSI **vsia**.

```
netdev_vsi{'vsia':
 ensure => present,
 vsiname => 'vsia',
 require => Netdev_device['device'],
}
```

netdev_vte

Use this resource to create or delete a tunnel.

Attributes

Table 18 Attributes for netdev_vte

Attribute name	Description	Attribute type	Value type and restrictions
id	Specifies a tunnel ID.	Index	Unsigned integer.
ensure	Creates or deletes the tunnel.	N/A	Symbol: • **present**—Creates the tunnel. • **absent**—Deletes the tunnel.

Attribute name	Description	Attribute type	Value type and restrictions
mode	Sets the tunnel mode.	N/A	Unsigned integer: • **1**—IPv4 GRE tunnel mode. • **2**—IPv6 GRE tunnel mode. • **3**—IPv4 over IPv4 tunnel mode. • **4**—Manual IPv6 over IPv4 tunnel mode. • **5**—Automatic IPv6 over IPv4 tunnel mode. • **6**—IPv6 over IPv4 6to4 tunnel mode. • **7**—IPv6 over IPv4 ISATAP tunnel mode. • **8**—IPv6 or IPv4 over IPv6 tunnel mode. • **14**—IPv4 multicast GRE tunnel mode. • **15**—IPv6 multicast GRE tunnel mode. • **16**—IPv4 IPsec tunnel mode. • **17**—IPv6 IPsec tunnel mode. • **24**—UDP-encapsulated IPv4 VXLAN tunnel mode. • **25**—UDP-encapsulated IPv6 VXLAN tunnel mode. You must specify the tunnel mode when creating a tunnel. After the tunnel is created, you cannot change the tunnel mode.

Resource example

Create UDP-encapsulated IPv4 VXLAN tunnel 2.

```
netdev_vte{'vte2':
 ensure => present,
 id => 2,
 mode => 24,
 require => Netdev_device['device'],
}
```

netdev_vxlan

Use this resource to create, modify, or delete a VXLAN.

Attributes

Table 19 Attributes for netdev_vxlan

Attribute name	Description	Attribute type	Value type and restrictions
vxlan_id	Specifies a VXLAN ID.	Index	Unsigned integer. Value range: 1 to 16777215.
ensure	Creates or deletes the VXLAN.	N/A	Symbol: • **present**—Creates or modifies the VXLAN. • **absent**—Deletes the VXLAN.
vsiname	Specifies the VSI name.	N/A	String, case sensitive. Length: 1 to 31 characters. You must specify the VSI name when creating a VSI. After the VSI is created, you cannot change the name.
add_tunnels	Specifies the tunnel interfaces to be associated with the VXLAN.	N/A	String, a comma separated list of tunnel interface IDs or tunnel interface ID ranges, for example, 1,2,3,5-8,10-20. The string cannot end with a comma (,), hyphen (-), or space. A tunnel interface ID cannot be on the list of adding interfaces and the list of removing interfaces at the same time.
delete_tunnels	Removes the association between the specified tunnel interfaces and the VXLAN.	N/A	String, a comma separated list of tunnel interface IDs or tunnel interface ID ranges, for example, 1,2,3,5-8,10-20. The string cannot end with a comma (,), hyphen (-), or space. A tunnel interface ID cannot be on the list of adding interfaces and the list of removing interfaces at the same time.

Resource example

Create VXLAN 10, configure the VSI name as **vsia**, and associate tunnel interfaces 7 and 8 with VXLAN 10.

```
netdev_vxlan{'vxlan10':
 ensure => present,
 vxlan_id => 10,
 vsiname => 'vsia',
 add_tunnels => '7-8',
 require=>Netdev_device['device'],
}
```

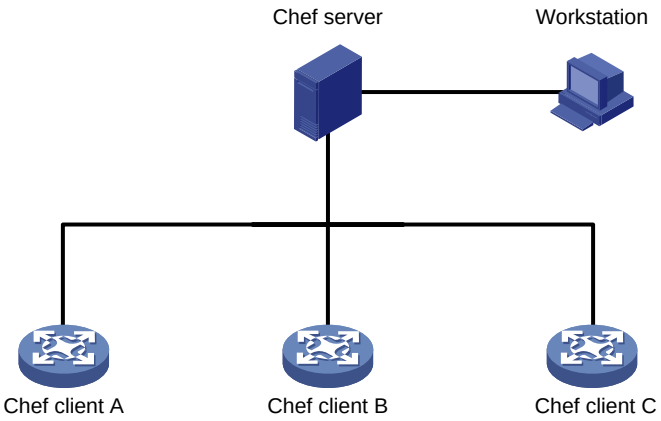
Configuring Chef

About Chef

Chef is an open-source configuration management tool. It uses the Ruby language. You can use the Ruby language to create cookbooks and save them to a server, and then use the server for centralized configuration enforcement and management.

Chef network framework

Figure 72 Chef network framework



As shown in Figure 72, Chef operates in a client/server network framework. Basic Chef network components include the Chef server, Chef clients, and workstations.

Chef server

The Chef server is used to centrally manage Chef clients. It has the following functions:

- Creates and deploys cookbooks to Chef clients on demand.
- Creates **.pem** key files for Chef clients and workstations. Key files include the following two types:
  - **User key file**—Stores user authentication information for a Chef client or a workstation. The Chef server uses this file to verify the validity of a Chef client or workstation. Before the Chef client or workstation initiates a connection to the Chef server, make sure the user key file is downloaded to the Chef client or workstation.
  - **Organization key file**—Stores authentication information for an organization. For management convenience, you can classify Chef clients or workstations that have the same type of attributes into organizations. The Chef server uses organization key files to verify the validity of organizations. Before a Chef client or workstation initiates a connection to the Chef server, make sure the organization key file is downloaded to the Chef client or workstation.

For information about installing and configuring the Chef server, see the official Chef website at

https://www.chef.io/.

Workstation

Workstations provide the interface for you to interact with the Chef server. You can create or modify cookbooks on a workstation and then upload the cookbooks to the Chef server.

A workstation can be hosted by the same host as the Chef server. For information about installing and configuring the workstation, see the official Chef website at

https://www.chef.io/.

Chef client

Chef clients are network devices managed by the Chef server. Chef clients download cookbooks from the Chef server and use the settings in the cookbooks.

The device supports Chef 12.3.0 client.

Chef resources

Chef uses Ruby to define configuration items. A configuration item is defined as a resource. A cookbook contains a set of resources for one feature.

Chef manages types of resources. Each resource has a type, a name, one or more properties, and one action. Every property has a value. The value specifies the state desired for the resource. You can specify the state of a device by setting values for properties regardless of how the device enters the state. The following resource example shows how to configure a device to create VLAN 2 and configure the description for VLAN 2.

```
netdev_vlan 'vlan2' do
 vlan_id 2
 description 'chef-vlan2'
 action :create
end
```

The following are the resource type, resource name, properties, and actions:

- **netdev_vlan**—Type of the resource.
- **vlan2**—Name of the resource. The name is the unique identifier of the resource.
- **do**/**end**—Indicates the beginning and end of a Ruby block that contains properties and actions. All Chef resources must be written by using the **do**/**end** syntax.
- **vlan_id**—Property for specifying a VLAN. In this example, VLAN 2 is specified.
- **description**—Property for configuring the description. In this example, the description for VLAN 2 is **chef-vlan2**.
- **create**—Action for creating or modifying a resource. If the resource does not exist, this action creates the resource. If the resource already exists, this action modifies the resource with the new settings. This action is the default action for Chef. If you do not specify an action for a resource, the **create** action is used.
- **delete**—Action for deleting a resource.

Chef supports only the **create** and **delete** actions.

For more information about resource types supported by Chef, see "Chef resources."

Chef configuration file

You can manually configure a Chef configuration file. A Chef configuration file contains the following items:

- Attributes for log messages generated by a Chef client.
- Directories for storing the key files on the Chef server and Chef client.
- Directory for storing the resource files on the Chef client.

After Chef starts up, the Chef client sends its key file specified in the Chef configuration file to the Chef server for authentication request. The Chef server compares its local key file for the client with

the received key file. If the two files are consistent, the Chef client passes the authentication. The Chef client then downloads the resource file to the directory specified in the Chef configuration file, loads the settings in the resource file, and outputs log messages as specified.

Table 20 Chef configuration file description

Item	Description
(Optional.) log_level	Severity level for log messages. Available values include **:auto**, **:debug**, **:info**, **:warn**, **:error**, and **:fatal**. The severity levels in ascending order are listed as follows: • **:debug** • **:info** • **:warn** (**:auto**) • **:error** • **:fatal** The default severity level is **:auto**, which is the same as **:warn**.
log_location	Log output mode: • **STDOUT**—Outputs standard Chef success log messages to a file. With this mode, you can specify the destination file for outputting standard Chef success log messages when you execute the **third-part-process start** command. The standard Chef error log messages are output to the configuration terminal. • **STDERR**—Outputs standard Chef error log messages to a file. With this mode, you can specify the destination file for outputting standard Chef error log messages when you execute the **third-part-process start** command. The standard Chef success log messages are output to the configuration terminal. • *logfilepath*—Outputs all log messages to a file, for example, **flash:/cheflog/a.log**. If you specify none of the options, all log messages are output to the configuration terminal.
node_name	Chef client name. A Chef client name is used to identify a Chef client. It is different from the device name configured by using the **sysname** command.
chef_server_url	URL of the Chef server and name of the organization created on the Chef server, in the format of **https://***localhost***:***port***/organizations/***ORG_NAME*. The *localhost* argument represents the name or IP address of the Chef server. The *port* argument represents the port number of the Chef server. The *ORG_NAME* argument represents the name of the organization.
validation_key	Path and name of the local organization key file, in the format of **flash:/***chef***/***validator***.pem**.
client_key	Path and name of the local user key file, in the format of **flash:/***chef***/***client***.pem**.
cookbook_path	Path for the resource files, in the format of **['flash:/***chef-repo***/***cookbooks***']**.

Restrictions and guidelines: Chef configuration

The Chef server cannot run a lower version than Chef clients.

Prerequisites for Chef

Before configuring Chef on the device, complete the following tasks on the device:

- Enable NETCONF over SSH. The Chef server sends configuration information to Chef clients through NETCONF over SSH. For information about NETCONF over SSH, see "Configuring NETCONF."
- Configure SSH login. Chef clients communicate with the Chef server through SSH. For information about SSH login, see *Fundamentals Configuration Guide*.

Starting Chef

Configuring the Chef server

1. Create key files for the workstation and the Chef client.
2. Create a Chef configuration file for the Chef client.

For more information about configuring the Chef server, see the Chef server installation and configuration guides.

Configuring a workstation

1. Create the working path for the workstation.
2. Create the directory for storing the Chef configuration file for the workstation.
3. Create a Chef configuration file for the workstation.
4. Download the key file for the workstation from the Chef server to the directory specified in the workstation configuration file.
5. Create a Chef resource file.
6. Upload the resource file to the Chef server.

For more information about configuring a workstation, see the workstation installation and configuration guides.

Configuring a Chef client

1. Download the key file from the Chef server to a directory on the Chef client.

   The directory must be the same as the directory specified in the Chef client configuration file.
2. Download the Chef configuration file from the Chef server to a directory on the Chef client.

   The directory must be the same as the directory that will be specified by using the **--config=***filepath* option in the **third-part-process start** command.
3. Start Chef on the device:
   a. Enter system view.

      system-view

   b. Start Chef.

      third-part-process start name chef-client arg --config=*filepath* **--runlist recipe[***Directory***]**

      By default, Chef is shut down.

Parameter	Description
--config=*filepath*	Specifies the path and name of the Chef configuration file.
--runlist recipe[*Directory***]**	Specifies the name of the directory that contains files and subdirectories associated with the resource.

For more information about the **third-part-process start** command, see "Monitoring and maintaining processes."

Shutting down Chef

Prerequisites

Before you shut down Chef, execute the **display process all** command to identify the ID of the Chef process. This command displays information about all processes on the device. Check the following fields:

- **THIRD**—This field displays **Y** for a third-party process.
- **COMMAND**—This field displays **chef-client /opt/ruby/b** for the Chef process.
- **PID**—Process ID.

Procedure

1. Enter system view.

   system-view
2. Shut down Chef.

   third-part-process stop pid *pid-list*

   For more information about the **third-part-process stop** command, see "Monitoring and maintaining processes."

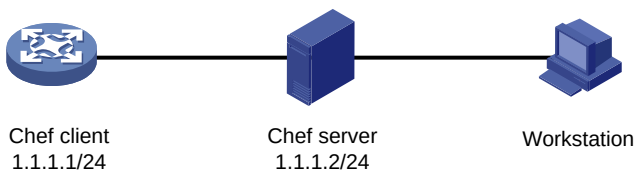
Chef configuration examples

Example: Configuring Chef

Network configuration

As shown in Figure 73, the device is connected to the Chef server. Use Chef to configure the device to create VLAN 3.

Figure 73 Network diagram



Procedure

1. Configure the Chef server:

   # Create user key file **admin.pem** for the workstation. Specify the workstation username as **Herbert George Wells**, the Email address as **abc@xyz.com**, and the password as **123456**.

```
$ chef-server-ctl user-create Herbert George Wells abc@xyz.com 123456
–filename=/etc/chef/admin.pem
```

Create organization key file **admin_org.pem** for the workstation. Specify the abbreviated organization name as **ABC** and the organization name as **ABC Technologies Co., Limited**. Associate the organization with the user **Herbert**.

```
$ chef-server-ctl org-create ABC_org "ABC Technologies Co., Limited"
–association_user Herbert –filename =/etc/chef/admin_org.pem
```

Create user key file **client.pem** for the Chef client. Specify the Chef client username as **Herbert George Wells**, the Email address as **abc@xyz.com**, and the password as **123456**.

```
$ chef-server-ctl user-create Herbert George Wells abc@xyz.com 123456
–filename=/etc/chef/client.pem
```

Create organization key file **validator.pem** for the Chef client. Specify the abbreviated organization name as **ABC** and the organization name as **ABC Technologies Co., Limited**. Associate the organization with the user **Herbert**.

```
$ chef-server-ctl org-create ABC "ABC Technologies Co., Limited" –association_user
Herbert –filename =/etc/chef/validator.pem
```

Create Chef configuration file **chefclient.rb** for the Chef client.

```
log_level :info
log_location STDOUT
node_name 'Herbert'
chef_server_url 'https://1.1.1.2:443/organizations/abc'
validation_key 'flash:/chef/validator.pem'
client_key 'flash:/chef/client.pem'
cookbook_path [ 'flash:/chef-repo/cookbooks' ]
```

2. Configure the workstation:

Create the **chef-repo** directory on the workstation. This directory will be used as the working path.

```
$ mkdir /chef-repo
```

Create the **.chef** directory. This directory will be used to store the Chef configuration file for the workstation.

```
$ mkdir –p /chef-repo/.chef
```

Create Chef configuration file **knife.rb** in the **/chef-repo/.chef0** directory.

```
log_level               :info
log_location            STDOUT
node_name               'admin'
client_key              '/root/chef-repo/.chef/admin.pem'
validation_key          '/root/chef-repo/.chef/admin_org.pem'
chef_server_url         'https://chef-server:443/organizations/abc'
```

Use TFTP or FTP to download the key files for the workstation from the Chef server to the **/chef-repo/.chef** directory on the workstation. (Details not shown.)

Create resource directory **netdev**.

```
$ knife cookbook create netdev
```

After the command is executed, the **netdev** directory is created in the current directory. The directory contains files and subdirectories for the resource. The **recipes** directory stores the resource file.

Create resource file **default.rb** in the **recipes** directory.

```
netdev_vlan 'vlan3' do
 vlan_id 3
 action :create
end
```

Upload the resource file to the Chef server.

```
$ knife cookbook upload –all
```

3. Configure the Chef client:

Configure SSH login and enable NETCONF over SSH on the device. (Details not shown.)

Use TFTP or FTP to download Chef configuration file **chefclient.rb** from the Chef server to the root directory of the Flash memory on the Chef client. Make sure this directory is the same as the directory specified by using the **`--config=`***`filepath`* option in the **`third-part-process start`** command.

Use TFTP or FTP to download key files **validator.pem** and **client.pem** from the Chef server to the **flash:/chef/** directory.

Start Chef. Specify the Chef configuration file name and path as **flash:/chefclient.rb** and the resource file name as **netdev**.

```
<ChefClient> system-view
[ChefClient] third-part-process start name chef-client arg
--config=flash:/chefclient.rb --runlist recipe[netdev]
```

After the command is executed, the Chef client downloads the resource file from the Chef server and loads the settings in the resource file.

Chef resources

netdev_device

Use this resource to specify a device name for a Chef client, and specify the SSH username and password used by the client to connect to the Chef server.

Properties and action

Table 21 Properties and action for netdev_device

Property/Action name	Description	Value type and restrictions
hostname	Specifies the device name.	String, case insensitive. Length: 1 to 64 characters.
user	Specifies the username for SSH login.	String, case sensitive. Length: 1 to 55 characters.
password	Specifies the password for SSH login.	String, case sensitive. Length and form requirements in non-FIPS mode: • 1 to 63 characters when in plaintext form. • 1 to 110 characters when in hashed form. • 1 to 117 characters when in encrypted form.
action	Specifies the action for the resource.	Symbol: • **create**—Establishes a NETCONF connection to the Chef server. • **delete**—Closes the NETCONF connection to the Chef server. The default action is **create**.

Resource example

Configure the device name as **ChefClient**, and set the SSH username and password to **user** and **123456** for the Chef client.

```
netdev_device 'device' do
 hostname "ChefClient"
 user "user"
 passwd "123456"
end
```

netdev_interface

Use this resource to configure attributes for an interface.

Properties

Table 22 Properties for netdev_interface

Property name	Description	Property type	Value type and restrictions
ifindex	Specifies an interface by its index.	Index	Unsigned integer.

Property name	Description	Property type	Value type and restrictions
description	Configures the description for the interface.	N/A	String, case sensitive. Length: 1 to 255 characters.
admin	Specifies the management state for the interface.	N/A	Symbol: • **up**—Brings up the interface. • **down**—Shuts down the interface.
speed	Specifies the interface rate.	N/A	Symbol: • **auto**—Autonegotiation. • **10m**—10 Mbps. • **100m**—100 Mbps. • **1g**—1 Gbps. • **10g**—10 Gbps. • **40g**—40 Gbps. • **100g**—100 Gbps.
duplex	Sets the duplex mode.	N/A	Symbol: • **full**—Full-duplex mode. • **half**—Half-duplex mode. • **auto**—Autonegotiation. This attribute applies only to Ethernet interfaces.
linktype	Sets the link type for the interface.	N/A	Symbol: • **access**—Sets the link type of the interface to **Access**. • **trunk**—Sets the link type of the interface to **Trunk**. • **hybrid**—Sets the link type of the interface to **Hybrid**. This attribute applies only to Layer 2 Ethernet interfaces.
portlayer	Sets the operation mode for the interface.	N/A	Symbol: • **bridge**—Layer 2 mode. • **route**—Layer 3 mode.
mtu	Sets the MTU permitted by the interface.	N/A	Unsigned integer in bytes. The value range depends on the interface type. This attribute applies only to Layer 3 Ethernet interface.

Resource example

Configure the following attributes for Ethernet interface 2:

- **Interface description**—ifindex2.
- **Management state**—Up.
- **Interface rate**—Autonegotiation.
- **Duplex mode**—Autonegotiation.
- **Link type**—Hybrid.
- **Operation mode**—Layer 2.
- **MTU**—1500 bytes.

```
netdev_interface 'ifindex2' do
```

```
 ifindex 2
 description 'ifindex2'
 admin 'up'
 speed 'auto'
 duplex 'auto'
 linktype 'hybrid'
 portlayer 'bridge'
 mtu 1500
end
```

netdev_l2_interface

Use this resource to configure VLAN attributes for a Layer 2 Ethernet interface.

Properties

Table 23 Properties for netdev_l2_interface

Property name	Description	Property type	Value type and restrictions
ifindex	Specifies a Layer 2 Ethernet interface by its index.	Index	Unsigned integer.
pvid	Specifies the PVID for the interface.	N/A	Unsigned integer. Value range: 1 to 4094.
permit_vlan_list	Specifies the VLANs permitted by the interface.	N/A	String, a comma separated list of VLAN IDs or VLAN ID ranges, for example, 1,2,3,5-8,10-20. Value range for each VLAN ID: 1 to 4094. The string cannot end with a comma (,), hyphen (-), or space.
untagged_vlan_list	Specifies the VLANs from which the interface sends packets after removing VLAN tags.	N/A	String, a comma separated list of VLAN IDs or VLAN ID ranges, for example, 1,2,3,5-8,10-20. Value range for each VLAN ID: 1 to 4094. The string cannot end with a comma (,), hyphen (-), or space. A VLAN cannot be on the untagged list and the tagged list at the same time.
tagged_vlan_list	Specifies the VLANs from which the interface sends packets without removing VLAN tags.	N/A	String, a comma separated list of VLAN IDs or VLAN ID ranges, for example, 1,2,3,5-8,10-20. Value range for each VLAN ID: 1 to 4094. The string cannot end with a comma (,), hyphen (-), or space. A VLAN cannot be on the untagged list and the tagged list at the same time.

Resource example

Specify the PVID as 2 for interface 5, and configure the interface to permit packets from VLANs 2 through 6. Configure the interface to forward packets from VLAN 3 after removing VLAN tags and forward packets from VLANs 2, 4, 5, and 6 without removing VLAN tags.

```
netdev_l2_interface 'ifindex5' do
 ifindex 5
 pvid 2
 permit_vlan_list '2-6'
 tagged_vlan_list '2,4-6'
 untagged_vlan_list '3'
end
```

netdev_lagg

Use this resource to create, modify, or delete an aggregation group.

Properties and action

Table 24 Properties and action for netdev_lagg

Property/Action name	Description	Property type	Value type and restrictions
group_id	Specifies an aggregation group ID.	Index	Unsigned integer. The value range for a Layer 2 aggregation group is 1 to 1024. The value range for a Layer 3 aggregation group is 16385 to 17408.
linkmode	Specifies the aggregation mode.	N/A	Symbol: • **static**—Static. • **dynamic**—Dynamic.
addports	Specifies the indexes of the interfaces that you want to add to the aggregation group.	N/A	String, a comma separated list of interface indexes or interface index ranges, for example, 1,2,3,5-8,10-20. The string cannot end with a comma (,), hyphen (-), or space. An interface index cannot be on the list of adding interfaces and the list of removing interfaces at the same time.
deleteports	Specifies the indexes of the interfaces that you want to remove from the aggregation group.	N/A	String, a comma separated list of interface indexes or interface index ranges, for example, 1,2,3,5-8,10-20. The string cannot end with a comma (,), hyphen (-), or space. An interface index cannot be on the list of adding interfaces and the list of removing interfaces at the same time.
action	Specifies the action for the resource.	N/A	Symbol: • **create**—Creates or modifies an aggregation group. • **delete**—Deletes an aggregation group. The default action is **create**.

Resource example

Create aggregation group 16386 and set the aggregation mode to **static**. Add interfaces 1 through 3 to the group, and remove interface 8 from the group.

```
netdev_lag 'lagg16386' do
 group_id 16386
 linkmode 'static'
 addports '1-3'
 deleteports '8'
end
```

netdev_vlan

Use this resource to create, modify, or delete a VLAN, or configure the name and description for the VLAN.

Properties and action

Table 25 Properties and action for netdev_vlan

Property/Action name	Description	Property type	Value type and restrictions
vlan_id	Specifies a VLAN ID.	Index	Unsigned integer. Value range: 1 to 4094.
description	Configures the description for the VLAN.	N/A	String, case sensitive. Length: 1 to 255 characters.
vlan_name	Configures the VLAN name.	N/A	String, case sensitive. Length: 1 to 32 characters.
action	Specifies the action for the resource.	N/A	Symbol: • **create**—Creates or modifies a VLAN. • **delete**—Deletes a VLAN. The default action is **create**.

Resource example

Create VLAN 2, configure the description as **vlan2**, and configure the VLAN name as **vlan2**.

```
netdev_vlan 'vlan2' do
 vlan_id 2
 description 'vlan2'
 vlan_name ‘vlan2’
end
```

netdev_vsi

Use this resource to create, modify, or delete a Virtual Switch Instance (VSI).

Properties and action

Table 26 Properties and action for netdev_vsi

Property/Action name	Description	Property type	Value type and restrictions
vsiname	Specifies a VSI name.	Index	String, case sensitive. Length: 1 to 31 characters.
admin	Enable or disable the VSI.	N/A	Symbol: • **up**—Enables the VSI. • **down**—Disables the VSI. The default value is **up**.
action	Specifies the action for the resource.	N/A	Symbol: • **create**—Creates or modifies a VSI. • **delete**—Deletes a VSI. The default action is **create**.

Resource example

Create the VSI **vsia** and enable the VSI.

```
netdev_vsi 'vsia' do
 vsiname 'vsia'
 admin 'up'
end
```

netdev_vte

Use this resource to create or delete a tunnel.

Properties and action

Table 27 Properties and action for netdev_vte

Property/Action name	Description	Property type	Value type and restrictions
vte_id	Specifies a tunnel ID.	Index	Unsigned integer.

Property/Action name	Description	Property type	Value type and restrictions
mode	Sets the tunnel mode.	N/A	Unsigned integer: • **1**—IPv4 GRE tunnel mode. • **2**—IPv6 GRE tunnel mode. • **3**—IPv4 over IPv4 tunnel mode. • **4**—Manual IPv6 over IPv4 tunnel mode. • **5**—Automatic IPv6 over IPv4 tunnel mode. • **6**—IPv6 over IPv4 6to4 tunnel mode. • **7**—IPv6 over IPv4 ISATAP tunnel mode. • **8**—IPv6 over IPv6 or IPv4 tunnel mode. • **14**—IPv4 multicast GRE tunnel mode. • **15**—IPv6 multicast GRE tunnel mode. • **16**—IPv4 IPsec tunnel mode. • **17**—IPv6 IPsec tunnel mode. • **24**—UDP-encapsulated IPv4 VXLAN tunnel mode. • **25**—UDP-encapsulated IPv6 VXLAN tunnel mode. You must specify the tunnel mode when creating a tunnel. After the tunnel is created, you cannot change the tunnel mode.
action	Specifies the action for the resource.	N/A	Symbol: • **create**—Creates a tunnel. • **delete**—Deletes a tunnel. The default action is **create**.

Resource example

Create UDP-encapsulated IPv4 VXLAN tunnel 2.

```
netdev_vte 'vte2' do
 vte_id 2
 mode 24
end
```

netdev_vxlan

Use this resource to create, modify, or delete a VXLAN.

Properties and action

Table 28 Properties and action for netdev_vxlan

Property/Action name	Description	Property type	Value type and restrictions
vxlan_id	Specifies a VXLAN ID.	Index	Unsigned integer. Value range: 1 to 16777215.

Property/Action name	Description	Property type	Value type and restrictions
vsiname	Specifies the VSI name.	N/A	String, case sensitive. Length: 1 to 31 characters. You must specify the VSI name when creating a VSI. After the VSI is created, you cannot change its name.
add_tunnels	Specifies the tunnel interfaces to be associated with the VXLAN.	N/A	String, a comma separated list of tunnel interface IDs or tunnel interface ID ranges, for example, 1,2,3,5-8,10-20. The string cannot end with a comma (,), hyphen (-), or space. A tunnel interface ID cannot be on the list of adding interfaces and the list of removing interfaces at the same time.
delete_tunnels	Removes the association between the specified tunnel interfaces and the VXLAN.	N/A	String, a comma separated list of tunnel interface IDs or tunnel interface ID ranges, for example, 1,2,3,5-8,10-20. The string cannot end with a comma (,), hyphen (-), or space. A tunnel interface ID cannot be on the list of adding interfaces and the list of removing interfaces at the same time.
action	Specifies the action for the resource.	N/A	Symbol: • **create**—Creates or modifies a VXLAN. • **delete**—Deletes a VXLAN. The default action is **create**.

Resource example

Create VXLAN 10, configure the VSI name as **vsia**, add tunnel interfaces 2 and 4 to the VXLAN, and remove tunnel interfaces 1 and 3 from the VXLAN.

```
netdev_vxlan 'vxlan10' do
 vxlan_id 10
 visname 'vsia'
 add_tunnels '2,4'
 delete_tunnels '1,3'
end
```

Configuring CWMP

About CWMP

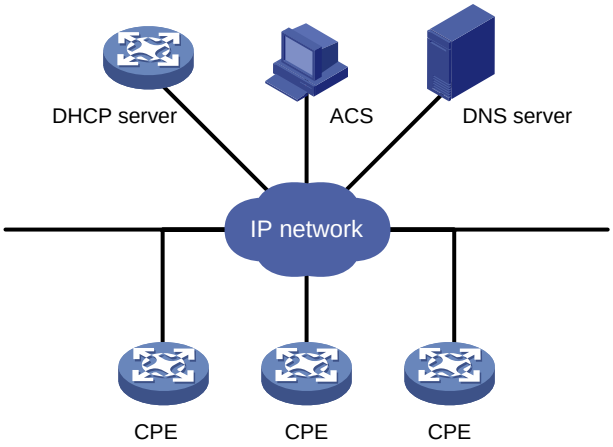
CPE WAN Management Protocol (CWMP), also called "TR-069," is a DSL Forum technical specification for remote management of network devices.

The protocol was initially designed to provide remote autoconfiguration through a server for large numbers of dispersed end-user devices in a network. CWMP can be used on different types of networks, including Ethernet.

CWMP network framework

Figure 74 shows a basic CWMP network framework.

Figure 74 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 can 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

You can autoconfigure and upgrade CPEs in bulk from the ACS.

Autoconfiguration

You can create configuration files for different categories of CPEs on the ACS. Based on the device models and serial numbers of the CPEs, the ACS verifies the categories of the CPEs and issues the associated configuration to them.

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.

CPE software 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.

If the CPE successfully downloads the software image file and the file is validated, the CPE notifies the ACS of a successful download. If the CPE fails to download the software image file or the file is invalidated, the CPE notifies the ACS of an unsuccessful download.

Data backup

The ACS can require the CPE to upload a configuration file or log file to a specific location. The destination location can be the ACS or a file server.

CPE status and performance monitoring

The ACS can monitor the status and performance of CPEs. Table 29 shows the available CPE status and performance objects for the ACS to monitor.

Table 29 CPE status and performance objects available for the ACS to monitor

Category	Objects	Remarks
Device information	Manufacturer ManufacturerOUI SerialNumber HardwareVersion SoftwareVersion	N/A
Operating status and information	DeviceStatus UpTime	N/A
Configuration file	ConfigFile	Local configuration file stored on CPE for upgrade. The ACS can issue configuration to the CPE by transferring a configuration file to the CPE or running the configuration in CPE's RAM.
CWMP settings	ACS URL	URL address of the ACS to which the CPE initiates a CWMP connection. This object is also used for main/backup ACS switchover.
	ACS username ACS password	When the username and password of the ACS are changed, the ACS changes the ACS username and password on the CPE to the new username and password. When a main/backup ACS switchover occurs, the main ACS also changes the ACS username and password to the backup ACS username and password.
	PeriodicInformEnable	Whether to enable or disable the periodic Inform feature.

Category	Objects	Remarks
	PeriodicInformInterval	Interval for periodic connection from the CPE to the ACS for configuration and software update.
	PeriodicInformTime	Scheduled time for connection from the CPE to the ACS for configuration and software update.
	ConnectionRequestURL (CPE URL)	N/A
	ConnectionRequestUsername (CPE username) ConnectionRequestPassword (CPE password)	CPE username and password for authentication from the ACS to the CPE.

How CWMP works

RPC methods

CWMP uses remote procedure call (RPC) methods for bidirectional communication between CPE and ACS. The RPC methods are encapsulated in HTTP or HTTPS.

Table 30 shows the primary RPC methods used in CWMP.

Table 30 RPC methods

RPC method	Description
Get	The ACS obtains the values of parameters on the CPE.
Set	The ACS modifies the values of parameters on the CPE.
Inform	The CPE sends an Inform message to the ACS for the following purposes: • Initiates a connection to the ACS. • Reports configuration changes to the ACS. • Periodically updates CPE settings to the ACS.
Download	The ACS requires the CPE to download a configuration or software image file from a specific URL for software or configuration update.
Upload	The ACS requires the CPE to upload a file to a specific URL.
Reboot	The ACS reboots the CPE remotely for the CPE to complete an upgrade or recover from an error condition.

Autoconnect between ACS and CPE

The CPE automatically initiates a connection to the ACS when one of the following events occurs:

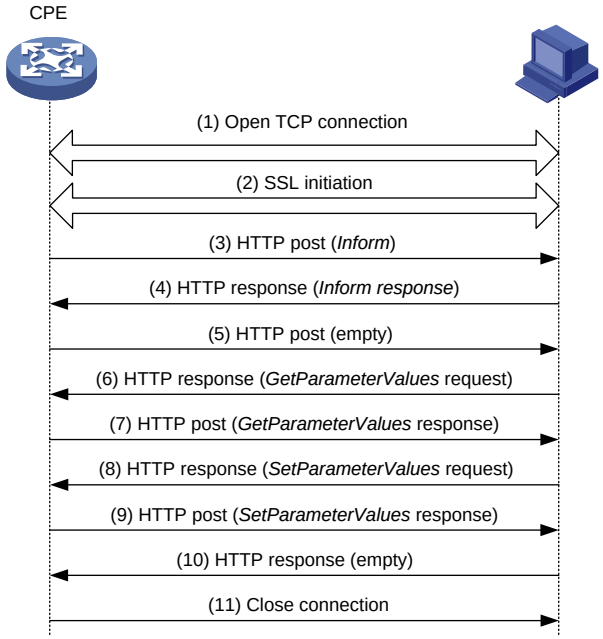
- ACS URL change. The CPE initiates a connection request to the new ACS URL.
- CPE startup. The CPE initiates a connection to the ACS after the startup.
- Timeout of the periodic Inform interval. The CPE re-initiates a connection to the ACS at the Inform interval.
- Expiration of the scheduled connection initiation time. The CPE initiates a connection to the ACS at the scheduled time.

CWMP connection establishment

Step 1 through step 5 in Figure 75 show the procedure of establishing a connection between the CPE and the ACS.

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.

Figure 75 CWMP connection establishment

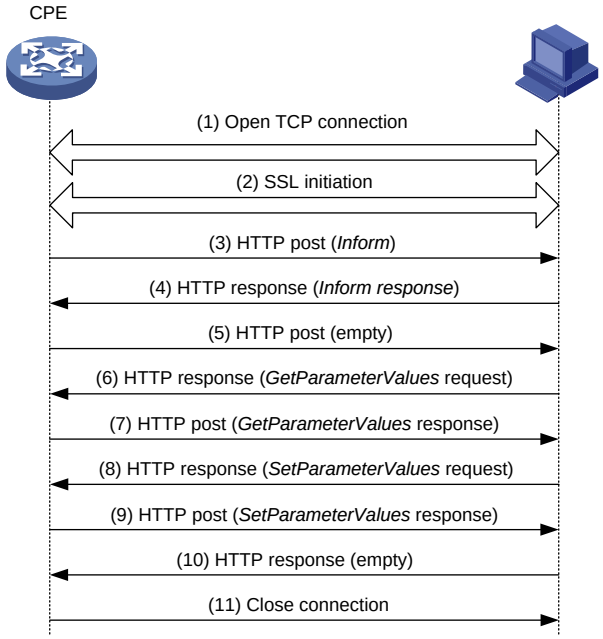


Main/backup ACS switchover

Typically, two ACSs are used in a CWMP network for consecutive monitoring on CPEs. When the main ACS needs to reboot, it points the CPE to the backup ACS. Step 6 through step 11 in Figure 76 show the procedure of a main/backup ACS switchover.

1. Before the main ACS reboots, it queries the ACS URL set on the CPE.
2. The CPE replies with its ACS URL setting.
3. The main ACS sends a Set request to change the ACS URL on the CPE to the backup ACS URL.
4. After the ACS URL is modified, the CPE sends a response.
5. The main ACS sends an empty HTTP message to notify the CPE that it has no other requests.
6. The CPE closes the connection, and then initiates a new connection to the backup ACS URL.

Figure 76 Main and backup ACS switchover



Restrictions and guidelines: CWMP configuration

You can 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.

CWMP tasks at a glance

To configure CWMP, perform the following tasks:

1. Enabling CWMP from the CLI

   You can also enable CWMP from a DHCP server.
2. Configuring ACS attributes
   a. Configuring the preferred ACS attributes
   b. (Optional.) Configuring the default ACS attributes from the CLI
3. Configuring CPE attributes
   a. Specifying an SSL client policy for HTTPS connection to ACS

      This task is required when the ACS uses HTTPS for secure access.
   b. (Optional.) Configuring ACS authentication parameters
   c. (Optional.) Configuring the provision code
   d. (Optional.) Configuring the CWMP connection interface
   e. (Optional.) Configuring autoconnect parameters
   f. (Optional.) Setting the close-wait timer

Enabling CWMP from the CLI

1. Enter system view.
   system-view
2. Enter CWMP view.
   cwmp
3. Enable CWMP.
   cwmp enable
   By default, CWMP is disabled.

Configuring ACS attributes

About 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.
- The default ACS attributes are configurable only from the CLI.

If the preferred ACS attributes are not configured, the CPE uses the default ACS attributes for connection establishment.

Configuring the preferred ACS attributes

Assigning ACS attributes from the DHCP server

The DHCP server in a CWMP network assigns the following information to CPEs:

- IP addresses for the CPEs.
- DNS server address.
- ACS URL and ACS login authentication information.

This section introduces how to use DHCP option 43 to assign the ACS URL and ACS login authentication username and password. For more information about DHCP and DNS, see *Layer 3—IP Services Configuration Guide*.

If the DHCP server is an HPE device, you can configure DHCP option 43 by using the **option 43 hex 01***length URL username password* command.

- *length*—A hexadecimal number that indicates the total length of the *length*, *URL*, *username*, and *password* arguments, including the spaces between these arguments. No space is allowed between the **01** keyword and the length value.
- *URL*—ACS URL.
- *username*—Username for the CPE to authenticate to the ACS.
- *password*—Password for the CPE to authenticate to the ACS.

NOTE:

The ACS URL, username and password must use the hexadecimal format and be space separated.

The following example configures the ACS address as **http://169.254.76.31:7547/acs**, username as **1234**, and password as **5678**:

```
<Sysname> system-view
```

```
[Sysname] dhcp server ip-pool 0
[Sysname-dhcp-pool-0] option 43 hex
0127687474703A2F2F3136392E3235342E37362E33313A373534372F6163732031323334203536373 8
```

Table 31 Hexadecimal forms of the ACS attributes

Attribute	Attribute value	Hexadecimal form
Length	39 characters	27
ACS URL	http://169.254.76.31:7547/acs	687474703A2F2F3136392E3235342E37362E33313A373534372F61637320 **NOTE:** The two ending digits (20) represent the space.
ACS connect username	1234	3132333420 **NOTE:** The two ending digits (20) represent the space.
ACS connect password	5678	35363738

Configuring the preferred ACS attributes from the CLI

1. Enter system view.
   system-view
2. Enter CWMP view.
   cwmp
3. Configure the preferred ACS URL.
   cwmp acs url *url*
   By default, no preferred ACS URL has been configured.
4. Configure the username for authentication to the preferred ACS URL.
   cwmp acs username *username*
   By default, no username has been configured for authentication to the preferred ACS URL.
5. (Optional.) Configure the password for authentication to the preferred ACS URL.
   cwmp acs password { **cipher** | **simple** } *string*
   By default, no password has been configured for authentication to the preferred ACS URL.

Configuring the default ACS attributes from the CLI

1. Enter system view.
   system-view
2. Enter CWMP view.
   cwmp
3. Configure the default ACS URL.
   cwmp acs default url *url*
   By default, no default ACS URL has been configured.
4. Configure the username for authentication to the default ACS URL.
   cwmp acs default username *username*
   By default, no username has been configured for authentication to the default ACS URL.
5. (Optional.) Configure the password for authentication to the default ACS URL.

cwmp acs default password { **cipher** | **simple** } *string*

By default, no password has been configured for authentication to the default ACS URL.

Configuring CPE attributes

About CPE attributes

You can configure the following CPE attributes only from the CPE's CLI.

- CWMP connection interface.
- Maximum number of connection retries.
- SSL client policy for HTTPS connection to ACS.

For other CPE attribute values, you can assign them to the CPE from the CPE's CLI or the ACS. The CLI- and ACS-assigned values overwrite each other, whichever is assigned later.

Specifying an SSL client policy for HTTPS connection to ACS

About this task

This task is required when the ACS uses HTTPS for secure access. CWMP uses HTTP or HTTPS for data transmission. When HTTPS is used, the ACS URL begins with **https://**. You must specify an SSL client policy for the CPE to authenticate the ACS for HTTPS connection establishment.

Prerequisites

Before you perform this task, first create an SSL client policy. For more information about configuring SSL client policies, see *Security Configuration Guide*.

Procedure

1. Enter system view.

   system-view
2. Enter CWMP view.

   cwmp
3. Specify an SSL client policy.

   ssl client-policy *policy-name*

   By default, no SSL client policy is specified.

Configuring ACS authentication parameters

About this task

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.

Procedure

1. Enter system view.

   system-view
2. Enter CWMP view.

   cwmp
3. Configure the username for authentication to the CPE.

cwmp cpe username *username*

By default, no username has been configured for authentication to the CPE.

4. (Optional.) Configure the password for authentication to the CPE.

   cwmp cpe password { **cipher** | **simple** } *string*

   By default, no password has been configured for authentication to the CPE.

   The password setting is optional. You can specify only a username for authentication.

Configuring the provision code

About this task

The ACS can use 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. For information about the support of your ACS for provision codes, see the ACS documentation.

Procedure

1. Enter system view.

   system-view

2. Enter CWMP view.

   cwmp

3. Configure the provision code.

   cwmp cpe provision-code *provision-code*

   The default provision code is **PROVISIONINGCODE**.

Configuring the CWMP connection interface

About this task

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 CWMP connection interface is not the interface that connects the CPE to the ACS, the CPE fails to establish a CWMP connection with the ACS. In this case, you need to manually set the CWMP connection interface.

Procedure

1. Enter system view.

   system-view

2. Enter CWMP view.

   cwmp

3. Specify the interface that connects to the ACS as the CWMP connection interface.

   cwmp cpe connect interface *interface-type interface-number*

   By default, no CWMP connection interface is specified.

Configuring autoconnect parameters

About this task

You can configure the CPE to connect to the ACS periodically, or at a scheduled time for configuration or software update.

The CPE retries a connection automatically when one of the following events occurs:

- The CPE fails to connect to the ACS. 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.
- The connection is disconnected before the session on the connection is completed.

To protect system resources, limit the number of retries that the CPE can make to connect to the ACS.

Configuring the periodic Inform feature

1. Enter system view.
   `system-view`
2. Enter CWMP view.
   `cwmp`
3. Enable the periodic Inform feature.
   `cwmp cpe inform interval enable`
   By default, this function is disabled.
4. Set the Inform interval.
   `cwmp cpe inform interval` *`interval`*
   By default, the CPE sends an Inform message to start a session every 600 seconds.

Scheduling a connection initiation

1. Enter system view.
   `system-view`
2. Enter CWMP view.
   `cwmp`
3. Schedule a connection initiation.
   `cwmp cpe inform time` *`time`*
   By default, no connection initiation has been scheduled.

Setting the maximum number of connection retries

1. Enter system view.
   `system-view`
2. Enter CWMP view.
   `cwmp`
3. Set the maximum number of connection retries.
   `cwmp cpe connect retry` *`retries`*
   By default, the CPE retries a failed connection until the connection is established.

Setting the close-wait timer

About this task

The close-wait timer specifies the following:

- 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.
- 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 sent or received before the timer expires.

Procedure

1. Enter system view.
   `system-view`
2. Enter CWMP view.
   `cwmp`
3. Set the close-wait timer.
   `cwmp cpe wait timeout` *seconds*
   By default, the close-wait timer is 30 seconds.

Display and maintenance commands for CWMP

Execute **`display`** commands in any view.

Task	Command
Display CWMP configuration.	**`display cwmp configuration`**
Display the current status of CWMP.	**`display cwmp status`**

CWMP configuration examples

Example: Configuring CWMP

Network configuration

As shown in Figure 77, use HPE 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 CPEs in equipment rooms A and B are **configure1.cfg** and **configure2.cfg**, respectively.

Figure 77 Network diagram

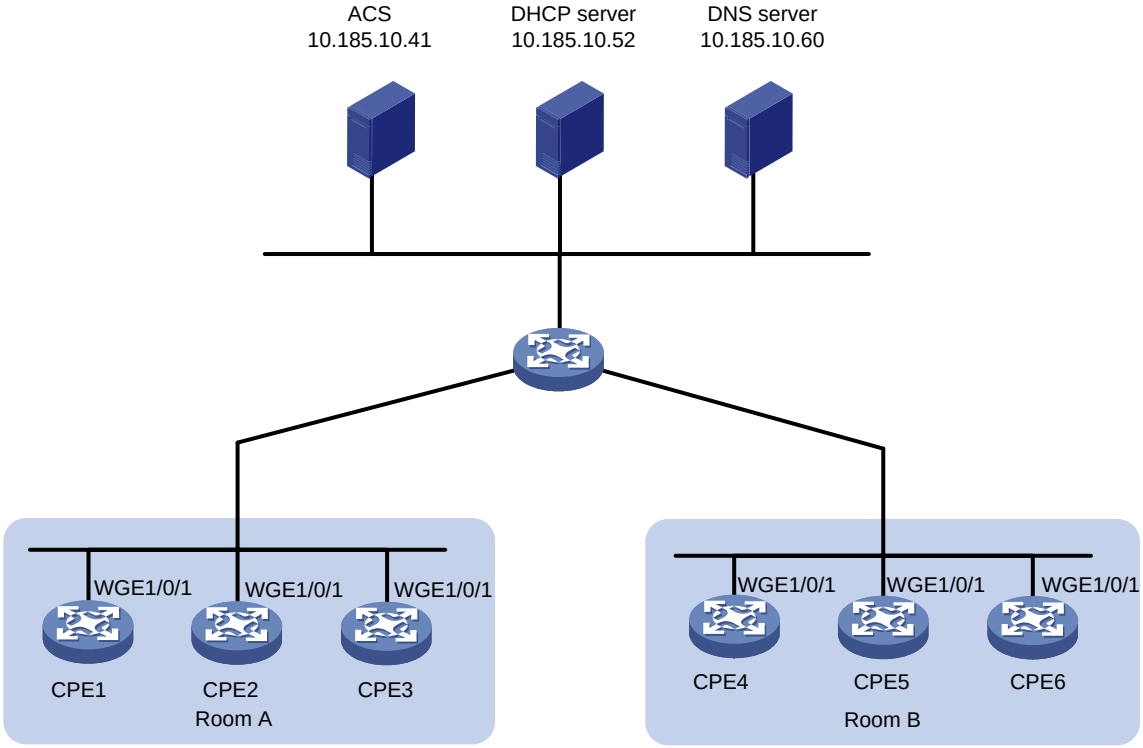


Table 32 shows the ACS attributes for the CPEs to connect to the ACS.

Table 32 ACS attributes

Item	Setting
Preferred ACS URL	http://10.185.10.41:9090
ACS username	admin
ACS password	12345

Table 33 lists serial numbers of the CPEs.

Table 33 CPE list

Room	Device	Serial number
A	CPE 1	210231A95YH10C000045
	CPE 2	210235AOLNH12000010
	CPE 3	210235AOLNH12000015
B	CPE 4	210235AOLNH12000017
	CPE 5	210235AOLNH12000020
	CPE 6	210235AOLNH12000022

Configuring the ACS

Figures in this section are for illustration only.

To configure the ACS:

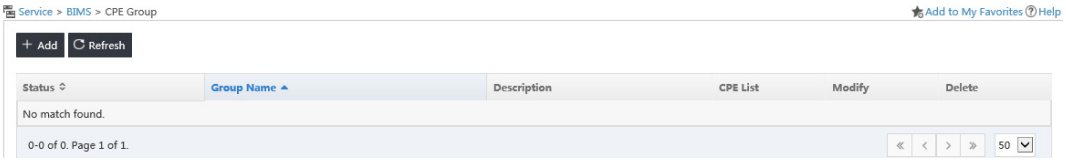
1. Log in to the ACS:

a. Launch a Web browser on the ACS configuration terminal.
b. In the address bar of the Web browser, enter the ACS URL and port number. This example uses **http://10.185.10.41:8080/imc**.
c. On the login page, enter the ACS login username and password, and then click **Login**.

2. Create a CPE group for each equipment room:
   a. Select **Service** > **BIMS** > **CPE Group** from the top navigation bar.

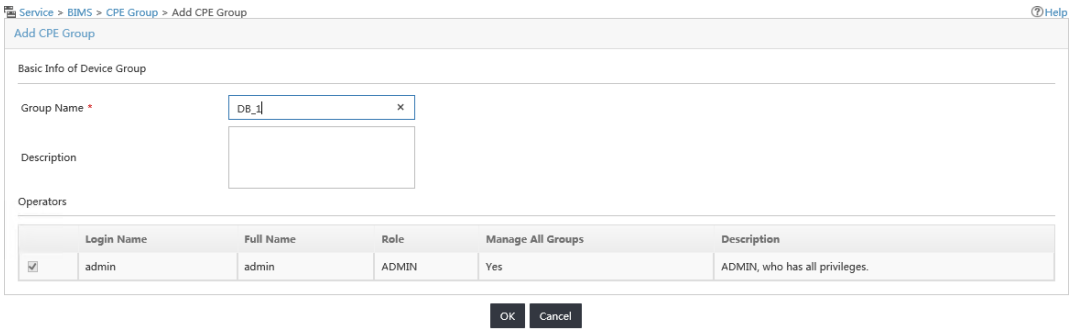
      The **CPE Group** page appears.

      Figure 78 CPE Group page

   b. Click **Add**.
   c. Enter a username, and then click **OK**.

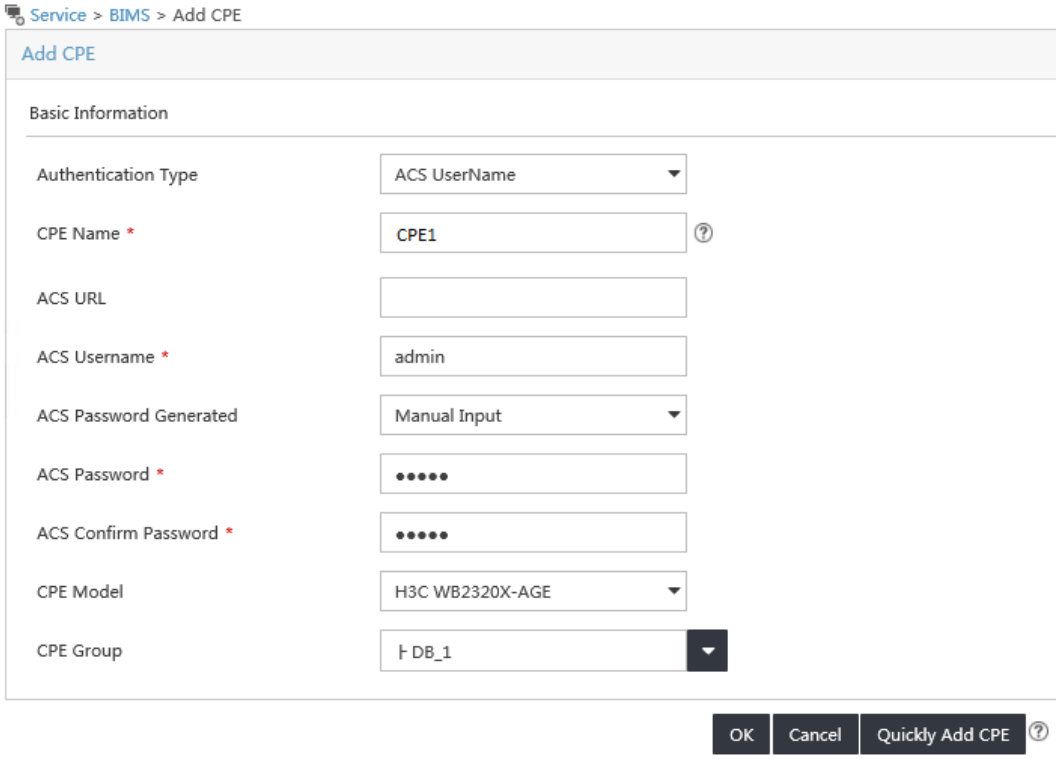
      Figure 79 Adding a CPE group

   d. Repeat the previous two steps to create a CPE group for CPEs in Room B.

3. Add CPEs to the CPE group for each equipment room:
   a. Select **Service** > **BIMS** > **Resource Management** > **Add CPE** from the top navigation bar.
   b. On the **Add CPE** page, configure the following parameters:
      - **Authentication Type**—Select **ACS UserName**.
      - **CPE Name**—Enter a CPE name.
      - **ACS Username**—Enter **admin**.
      - **ACS Password Generated**—Select **Manual Input**.
      - **ACS Password**—Enter a password for ACS authentication.
      - **ACS Confirm Password**—Re-enter the password.
      - **CPE Model**—Select the CPE model.
      - **CPE Group**—Select the CPE group.

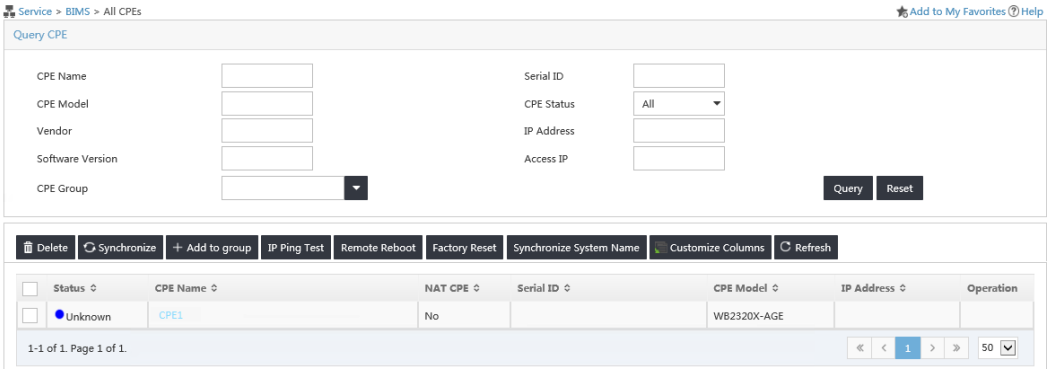
Figure 80 Adding a CPE



c. Click **OK**.

d. Verify that the CPE has been added successfully from the **All CPEs** page.

Figure 81 Viewing CPEs

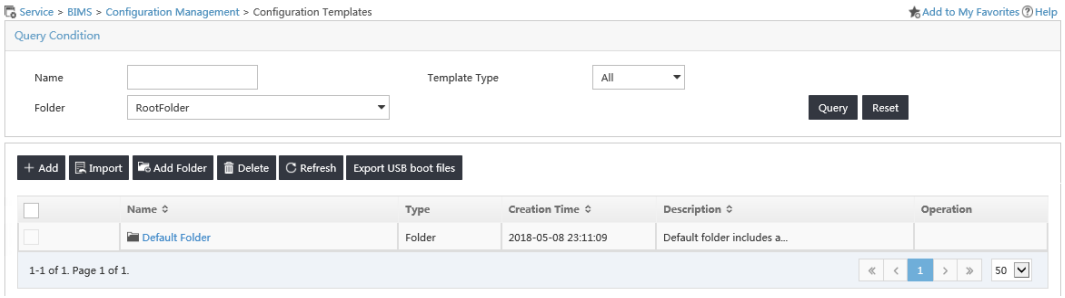


e. Repeat the previous steps to add CPE 2 and CPE 3 to the CPE group for Room A, and add CPEs in Room B to the CPE group for Room B.

4. Configure a configuration template for each equipment room:

a. Select **Service** > **BIMS** > **Configuration Management** > **Configuration Templates** from the top navigation bar.

Figure 82 Configuration Templates page



b. Click **Import**.

c. Select a source configuration file, select **Configuration Segment** as the template type, and then click **OK**.

The created configuration template will be displayed in the **Configuration Template** list after a successful file import.

IMPORTANT:

If the first command in the configuration template file is **`system-view`**, make sure no characters exist in front of the command.

Figure 83 Importing a configuration template

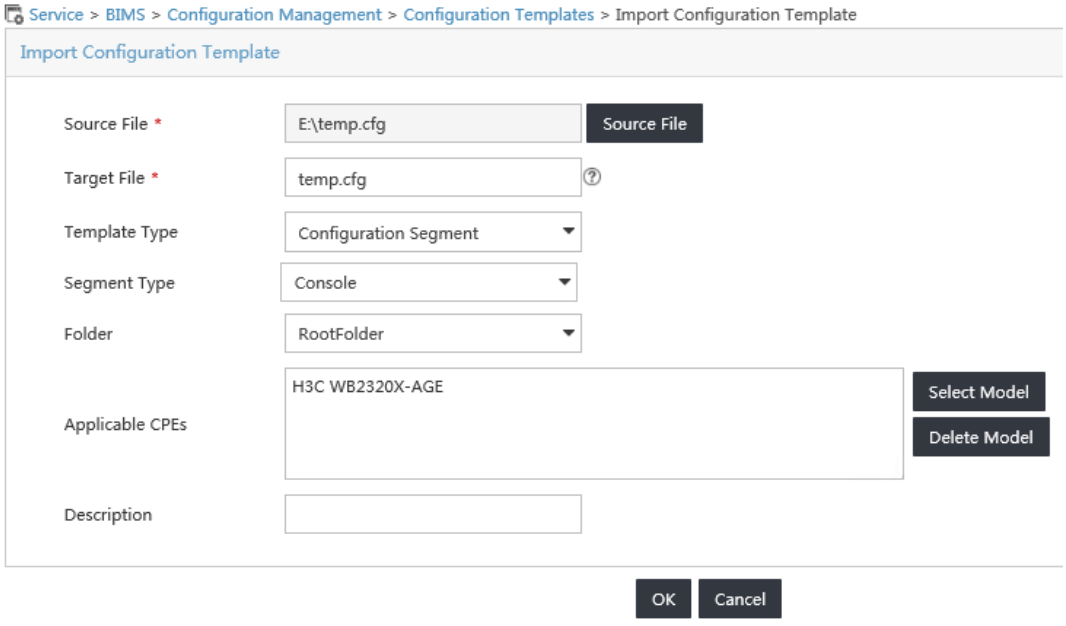
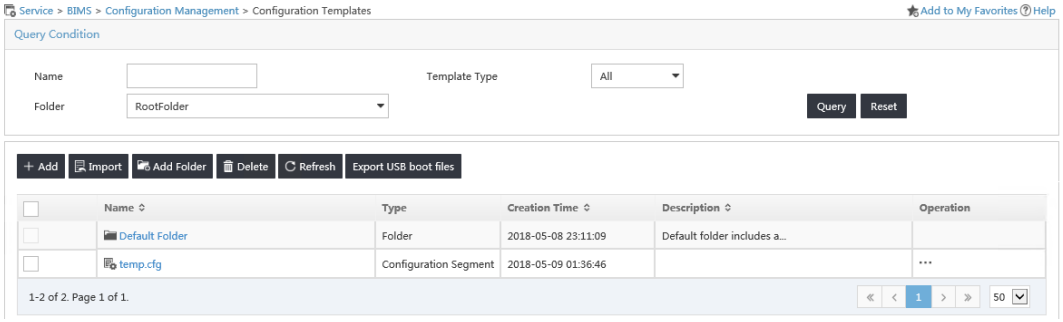


Figure 84 Configuration Template list

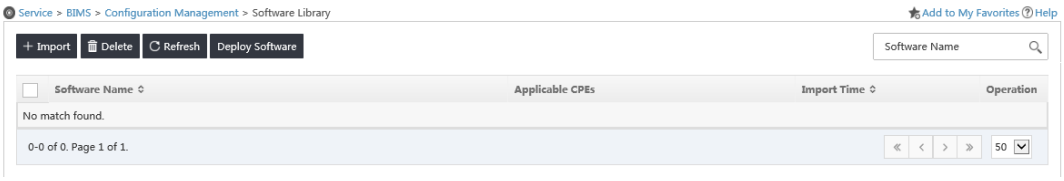


d. Repeat the previous steps to configure a configuration template for Room B.

5. Add software library entries:

a. Select **Service** > **BIMS** > **Configuration Management** > **Software Library** from the top navigation bar.

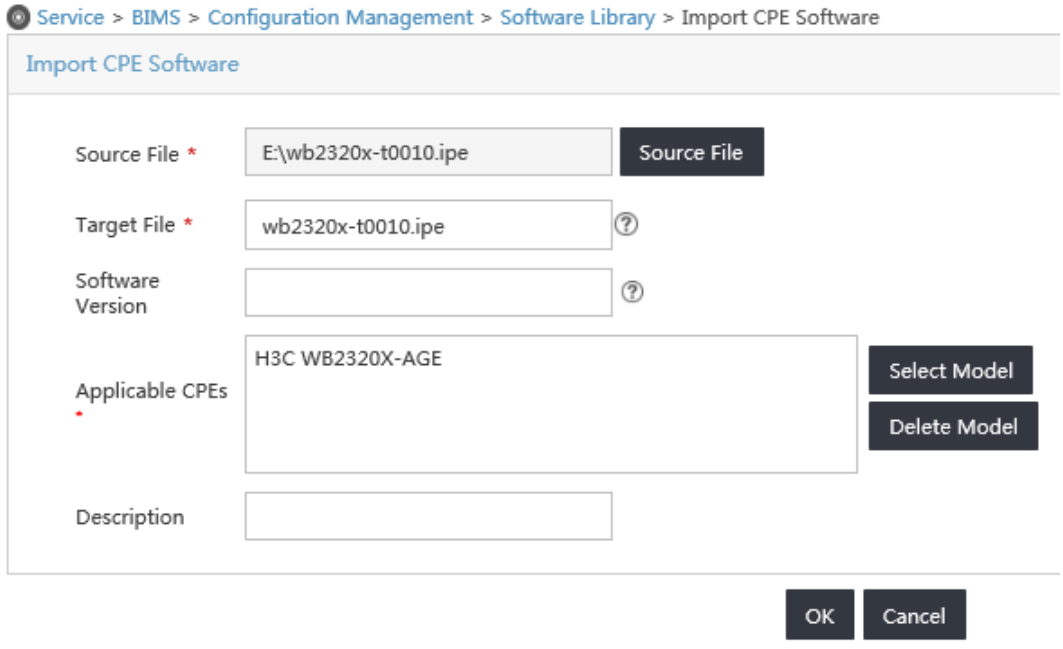
Figure 85 Software Library page



b. Click **Import**.

c. Select a source file, and then click **OK**.

Figure 86 Importing CPE software

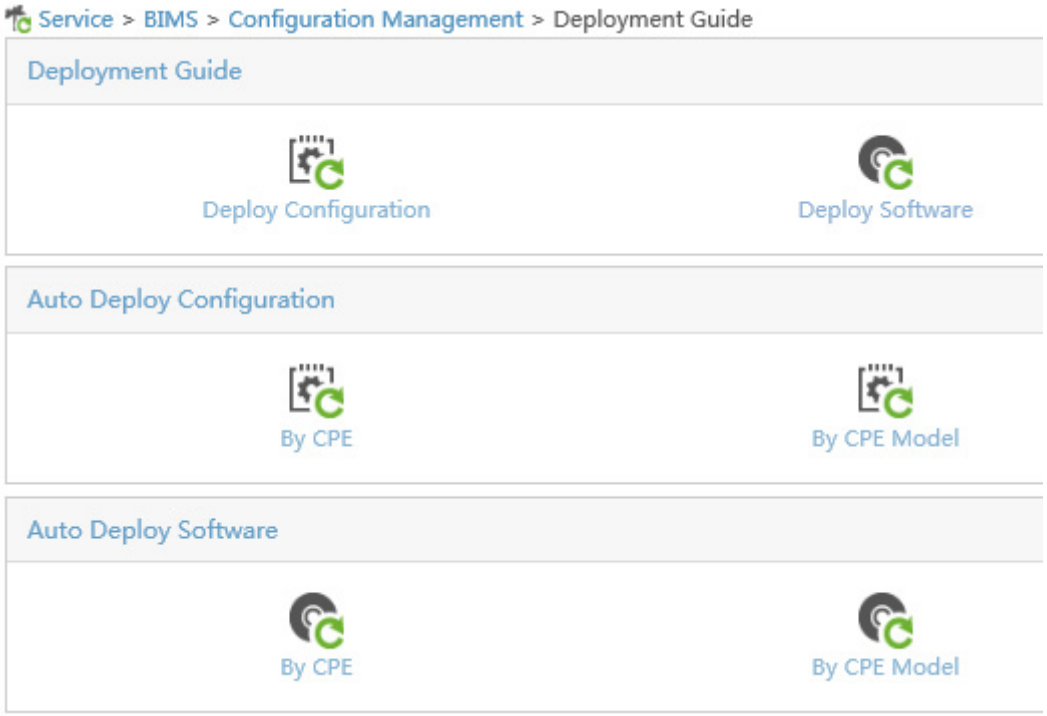


d. Repeat the previous steps to add software library entries for CPEs of different models.

6. Create an auto-deployment task for each equipment room:

a. Select **Service** > **BIMS** > **Configuration Management** > **Deployment Guide** from the top navigation bar.

Figure 87 Deployment Guide

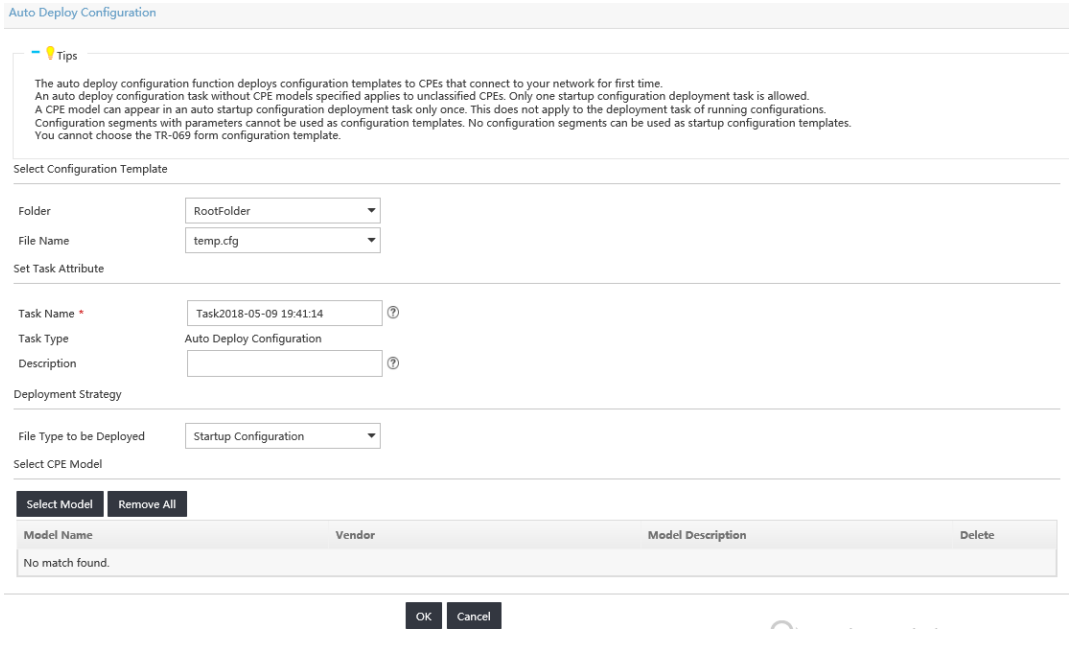


b. Click **By CPE Model** from the **Auto Deployment Configuration** field.

c. Select a configuration template, select **Startup Configuration** from the **File Type to be Deployed** list, and click **Select Model** to select CPEs in Room A. Then, click **OK**.

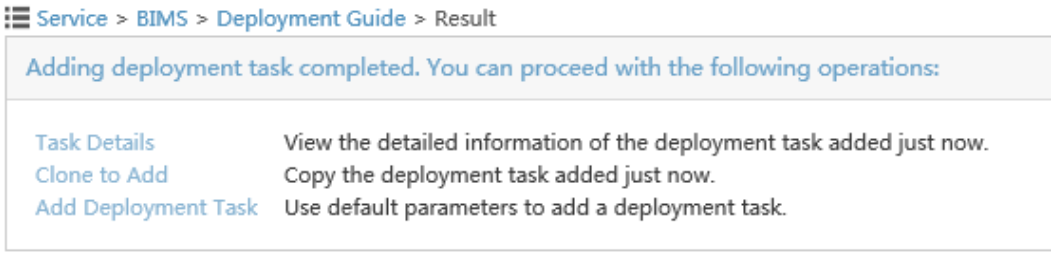
You can search for CPEs by CPE group.

Figure 88 Auto deployment configuration



d. Click **OK** on the **Auto Deploy Configuration** page.

Figure 89 Operation result



e. Repeat the previous steps to add a deployment task for CPEs in Room B.

Configuring the DHCP server

In this example, an HPE device is operating as the DHCP server.

1. Configure an IP address pool to assign IP addresses and DNS server address to the CPEs. This example uses subnet 10.185.10.0/24 for IP address assignment.

Enable DHCP.

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

Enable DHCP server on VLAN-interface 1.

```
[DHCP_server] interface vlan-interface 1
[DHCP_server-Vlan-interface1] dhcp select server
[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
013B687474703A2F2F6163732E6461746162617365 3A393039302F616373207669636B79203132333
435
```

Configuring the DNS server

Map http://acs.database:9090 to http://10.185.1.41:9090 on the DNS server. For more information about DNS configuration, see DNS server documentation.

Connecting the CPEs to the network

Connect CPE 1 to the network, and then power on the CPE. (Details not shown.)

Log in to CPE 1 and configure its interface GigabitEthernet 1/0/1 to use DHCP for IP address acquisition. At startup, the CPE obtains the IP address and ACS information from the DHCP server to initiate a connection to the ACS. After the connection is established, the CPE interacts with the ACS to complete autoconfiguration.

```
<CPE1> system-view
[CPE1] interface interface-type twenty-fivegige 1/0/1
```

```
[CPE1] ip address dhcp-alloc
```

Repeat the previous steps to configure the other CPEs.

Verifying the configuration

Execute the **display current-configuration** command to verify that the running configurations on CPEs are the same as the configurations issued by the ACS.

Configuring EAA

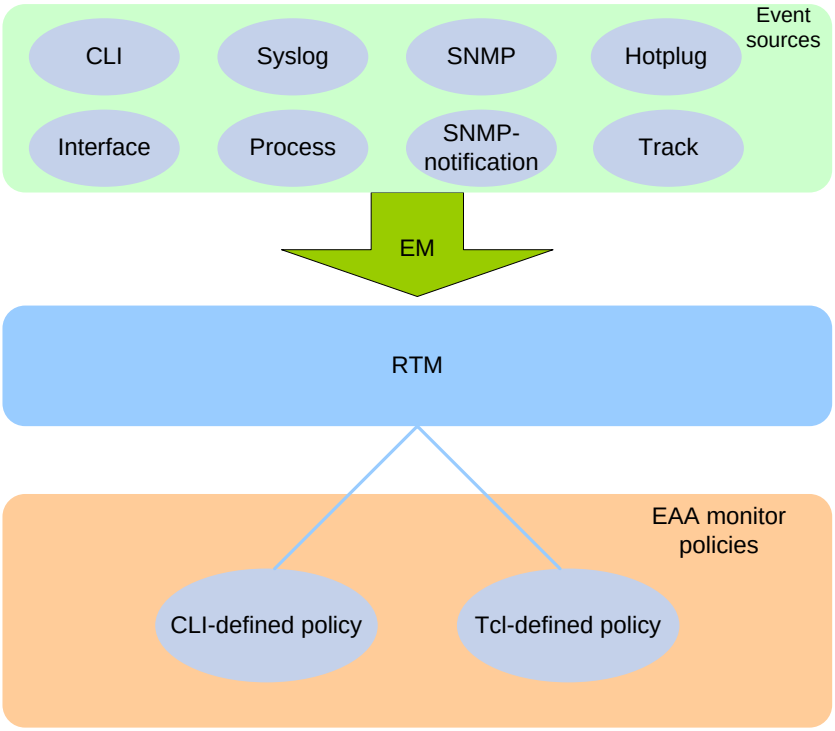
About EAA

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 90.

Figure 90 EAA framework



Event sources

Event sources are software or hardware modules that trigger events (see Figure 90).

For example, the CLI module triggers an event when you enter a command. 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.
- A minimum of one action.
- A minimum of one user role.
- One running time setting.

For more information about these elements, see "Elements in a monitor policy."

Elements in a monitor policy

Elements in an EAA monitor policy include event, action, user role, and runtime.

Event

Table 34 shows types of events that EAA can monitor.

Table 34 Monitored events

Event type	Description
CLI	CLI event occurs in response to monitored operations performed at the CLI. For example, a command is entered, a question mark (?) is entered, or the **Tab** key is pressed to complete a command.
Syslog	Syslog event occurs when the information center receives the monitored log within a specific period. NOTE: The log that is generated by the EAA RTM does not trigger the monitor policy to run.
Process	Process event occurs in response to a state change of the monitored process (such as an exception, shutdown, start, or restart). Both manual and automatic state changes can cause the event to occur.
Hotplug	Hot-swapping event occurs when the monitored member device joins or leaves the IRF fabric or a card is inserted in or removed from the monitored slot.
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 reaches or exceeds 30%.

Track	Track event occurs when the state of the track entry changes from Positive to Negative or from Negative to Positive. If you specify multiple track entries for a policy, EAA triggers the policy only when the state of all the track entries changes from Positive (Negative) to Negative (Positive). If you set a suppress time for a policy, the timer starts when the policy is triggered. The system does not process the messages that report the track entry state change from Positive (Negative) to Negative (Positive) until the timer times out.

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. However, 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

The runtime limits the amount of time that the monitor policy runs its actions from the time it is triggered. This setting prevents a policy from running its actions permanently to occupy resources.

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*). 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. The variable values are set automatically depending on the event setting in the policy that uses 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. The hotplug event-specific variables are _slot and _subslot. When a member device in slot 1 joins or leaves the IRF fabric,

the value of _slot is 1. When a member device in slot 2 joins or leaves the IRF fabric, the value of _slot is 2.

Table 35 shows all system-defined variables.

Table 35 System-defined EAA environment variables by event type

Event	Variable name and description
Any event	_event_id: Event ID _event_type: Event type _event_type_string: Event type description _event_time: Time when the event occurs _event_severity: Severity level of an event
CLI	_cmd: Commands that are matched
Syslog	_syslog_pattern: Log message content
Hotplug	_slot: ID of the member device that joins or leaves the IRF fabric
Hotplug	_subslot: ID of the subslot where subcard hot-swapping occurs. Only HPE FlexFabric 5945 2-slot Switch (JQ075A) and HPE FlexFabric 5945 48SFP28 8QSFP28 Switch (JQ076A) support subcards.
Interface	_ifname: Interface name
SNMP	_oid: OID of the MIB variable where an SNMP operation is performed _oid_value: Value of the MIB variable
SNMP-Notification	_oid: OID that is included in the SNMP notification.
Process	_process_name: Process name

User-defined variables

You can use user-defined variables for all types of events.

User-defined variable names can contain digits, characters, and the underscore sign (_), except that the underscore sign cannot be the leading character.

Configuring a user-defined EAA environment variable

About this task

Configure user-defined EAA environment variables so that you can use them when creating EAA monitor policies.

Procedure

1. Enter system view.
   `system-view`
2. Configure a user-defined EAA environment variable.
   `rtm environment` *`var-name var-value`*
   For the system-defined variables, see Table 35.

Configuring a monitor policy

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. However, you cannot assign the same name to policies that are the same type.

A monitor policy supports only one event and runtime. If you configure multiple events for a policy, the most recent one takes effect.

A monitor policy supports a maximum of 64 valid user roles. User roles added after this limit is reached do not take effect.

Configuring a monitor policy from the CLI

Restrictions and guidelines

You can configure a series of actions to be executed in response to the event specified in a monitor policy. EAA executes the actions in ascending order of action IDs. When you add actions to a policy, you must make sure the execution order is correct. If two actions have the same ID, the most recent one takes effect.

Procedure

1. Enter system view.

   system-view

2. (Optional.) Set the size for the EAA-monitored log buffer.

   rtm event syslog buffer-size *buffer-size*

   By default, the EAA-monitored log buffer stores a maximum of 50000 logs

3. Create a CLI-defined policy and enter its view.

   rtm cli-policy *policy-name*

4. Configure an event for the policy.
   - Configure a CLI event.

     event cli { **async** [**skip**] | **sync** } **mode** { **execute** | **help** | **tab** } **pattern** *regular-exp*
   - Configure a hotplug event.

     event hotplug [**insert** | **remove**] **slot** *slot-number* [**subslot** *subslot-number*]
   - Configure an interface event.

     event interface *interface-list* **monitor-obj** *monitor-obj* **start-op** *start-op* **start-val** *start-val* **restart-op** *restart-op* **restart-val** *restart-val* [**interval** *interval*]
   - Configure a process event.

     event process { **exception** | **restart** | **shutdown** | **start** } [**name** *process-name* [**instance** *instance-id*]] [**slot** *slot-number*]
   - Configure an SNMP event.

     event snmp oid *oid* **monitor-obj** { **get** | **next** } **start-op** *start-op* **start-val** *start-val* **restart-op** *restart-op* **restart-val** *restart-val* [**interval** *interval*]

- Configure an SNMP-Notification event.

  event snmp-notification oid *oid* **oid-val** *oid-val* **op** *op* [**drop**]
- Configure a Syslog event.

  event syslog priority *priority* **msg** *msg* **occurs** *times* **period** *period*
- Configure a track event.

  event track *track-list* **state { negative** | **positive** } [**suppress-time** *suppress-time*]

By default, a monitor policy does not contain an event.

If you configure multiple events for a policy, the most recent one takes effect.

5. Configure the actions to take when the event occurs.

   Choose the following tasks as needed:
   - Configure a CLI action.

     action *number* **cli** *command-line*
   - Configure a reboot action.

     action *number* **reboot** [**slot** *slot-number*]
   - Configure an active/standby switchover action.

     action *number* **switchover**
   - Configure a logging action.

     action *number* **syslog priority** *priority* **facility** *local-number* **msg** *msg-body*

   By default, a monitor policy does not contain any actions.
6. (Optional.) Assign a user role to the policy.

   user-role *role-name*

   By default, a monitor policy contains user roles that its creator had at the time of policy creation.

   An EAA policy cannot have both the **security-audit** user role and any other user roles. Any previously assigned user roles are automatically removed when you assign the **security-audit** user role to the policy. The previously assigned **security-audit** user role is automatically removed when you assign any other user roles to the policy.
7. (Optional.) Configure the policy action runtime.

   running-time *time*

   The default policy action runtime is 20 seconds.

   If you configure multiple action runtimes for a policy, the most recent one takes effect.
8. Enable the policy.

   commit

   By default, CLI-defined policies are not enabled.

   A CLI-defined policy can take effect only after you perform this step.

Configuring a monitor policy by using Tcl

About this task

A Tcl script contains two parts: Line 1 and the other lines.

- Line 1

  Line 1 defines the event, user roles, and policy action runtime. After you create and enable a Tcl monitor policy, the device immediately parses, delivers, and executes Line 1.

  Line 1 must use the following format:

::platformtools::rtm::event_register *event-type arg1 arg2 arg3* ... **user-role** *role-name1* | [**user-role** *role-name2* | [...]] [**running-time** *running-time*]

- The *arg1 arg2 arg3* ... arguments represent event matching rules. If an argument value contains spaces, use double quotation marks ("") to enclose the value. For example, "a b c."
- The configuration requirements for the *event-type*, *user-role*, and *running-time* arguments are the same as those for a CLI-defined monitor policy.

- The other lines

  From the second line, the Tcl script defines the actions to be executed when the monitor policy is triggered. You can use multiple lines to define multiple actions. The system executes these actions in sequence. The following actions are available:

  - Standard Tcl commands.
  - EAA-specific Tcl actions:
    - switchover (**::platformtools::rtm::action switchover**)
    - syslog (**::platformtools::rtm::action syslog priority** *priority* **facility** *local-number* **msg** *msg-body*). For more information about these arguments, see EAA commands in *Network Management and Monitoring Command Reference*.
  - Commands supported by the device.

Restrictions and guidelines

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.

Procedure

1. Download the Tcl script file to the device by using FTP or TFTP.

   For more information about using FTP and TFTP, see *Fundamentals Configuration Guide*.
2. Create and enable a Tcl monitor policy.
   a. Enter system view.

      system-view

   b. Create a Tcl-defined policy and bind it to the Tcl script file.

      rtm tcl-policy *policy-name tcl-filename*

   By default, no Tcl policies exist.

   Make sure the script file is saved on all IRF member devices. This practice ensures that the policy can run correctly after a master/subordinate switchover occurs or the member device where the script file resides leaves the IRF.

Suspending monitor policies

About this task

This task suspends all CLI-defined and Tcl-defined monitor policies. If a policy is running when you perform this task, the system suspends the policy after it executes all the actions.

Restrictions and guidelines

To restore the operation of the suspended policies, execute the **undo rtm scheduler suspend** command.

Procedure

1. Enter system view.

system-view

2. Suspend monitor policies.
   rtm scheduler suspend

Display and maintenance commands for EAA

Execute **display** commands except for the **display this** command in any view.

Task	Command
Display the running configuration of all CLI-defined monitor policies.	**display current-configuration**
Display user-defined EAA environment variables.	**display rtm environment** [*var-name*]
Display EAA monitor policies.	**display rtm policy** { **active** | **registered** [**verbose**] } [*policy-name*]
Display the running configuration of a CLI-defined monitor policy in CLI-defined monitor policy view.	**display this**

EAA configuration examples

Example: Configuring a CLI event monitor policy by using Tcl

Network configuration

As shown in Figure 91, 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.

Figure 91 Network diagram

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.

```
::platformtools::rtm::event_register cli sync mode execute pattern display this user-role network-admin
::platformtools::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 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

Execute the **display rtm policy registered** command to verify that a Tcl-defined policy named **test** is displayed in the command output.

```
<Sysname> display rtm policy registered
Total number: 1
Type  Event             TimeRegistered       PolicyName
TCL   CLI               Jan 01 09:47:12 2019 test
```

Enable the information center to output log messages to the current monitoring terminal.

```
<Sysname> terminal monitor
The current terminal is enabled to display logs.
<Sysname> system-view
[Sysname] info-center enable
Information center is enabled.
[Sysname] quit
```

Execute the **display this** command. Verify that the system displays an "rtm_tcl_test is running" message and a message that the policy is being executed successfully.

```
[Sysname] display this
%Jan  1 09:50:04:634 2019 Sysname RTM/1/RTM_ACTION: rtm_tcl_test is running
%Jan  1 09:50:04:636 2019 Sysname RTM/6/RTM_POLICY: TCL policy test is running
successfully.
#
return
```

Example: Configuring a CLI event monitor policy from the CLI

Network configuration

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.

Procedure

Create 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 a priority of 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
```

Add an action that creates VLAN 2 when the event occurs.

```
[Sysname-rtm-test] action 3 cli vlan 2
```

Set the policy action runtime to 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

Execute the **`display rtm policy registered`** command to verify that a CLI-defined policy named **test** is displayed in the command output.

```
[Sysname-rtm-test] display rtm policy registered
Total number: 1
Type  Event      TimeRegistered       PolicyName
CLI   CLI        Jan  1 14:56:50 2019 test
```

Enable the information center to output log messages to the current monitoring terminal.

```
[Sysname-rtm-test] return
<Sysname> terminal monitor
The current terminal is enabled to display logs.
<Sysname> system-view
[Sysname] info-center enable
Information center is enabled.
[Sysname] quit
```

Enter a question mark (?) at a command line that contains a letter **d**. Verify that the system displays a "hello world" message and a message that the policy is being executed successfully on the terminal screen.

```
<Sysname> d?
  debugging
  delete
  diagnostic-logfile
  dir
  display

<Sysname>d%Jan  1 14:57:20:218 2019 Sysname RTM/4/RTM_ACTION: "hello world"
%Jan  1 14:58:11:170 2019 Sysname RTM/6/RTM_POLICY: CLI policy test is running
successfully.
```

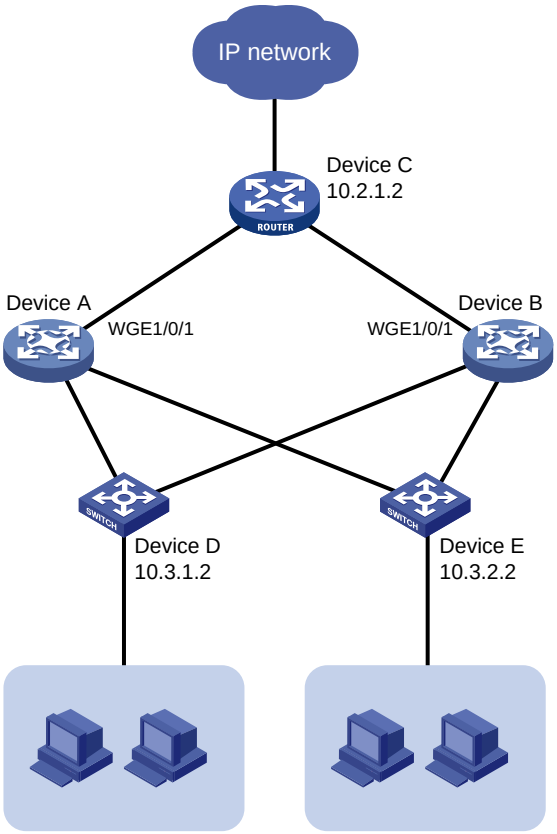
Example: Configuring a track event monitor policy from the CLI

Network configuration

As shown in Figure 92, Device A has established BGP sessions with Device D and Device E. Traffic from Device D and Device E to the Internet is forwarded through Device A.

Configure a CLI-defined EAA monitor policy on Device A to disconnect the sessions with Device D and Device E when Twenty-FiveGigE 1/0/1 connected to Device C is down. In this way, traffic from Device D and Device E to the Internet can be forwarded through Device B.

Figure 92 Network diagram



Procedure

Display BGP peer information for Device A.

```
<DeviceA> display bgp peer ipv4

 BGP local router ID: 1.1.1.1
 Local AS number: 100
 Total number of peers: 3                 Peers in established state: 3

  * - Dynamically created peer
  Peer                    AS  MsgRcvd  MsgSent OutQ PrefRcv Up/Down  State

  10.2.1.2                200       13       16    0       0 00:16:12 Established
  10.3.1.2                300       13       16    0       0 00:10:34 Established
  10.3.2.2                300       13       16    0       0 00:10:38 Established
```

Create track entry **1** and associate it with the link state of Twenty-FiveGigE 1/0/1.

```
<Device A> system-view
[Device A] track 1 interface twenty-fivegige 1/0/1
```

Configure a CLI-defined EAA monitor policy so that the system automatically disables session establishment with Device D and Device E when Twenty-FiveGigE 1/0/1 is down.

```
[Device A] rtm cli-policy test
[Device A-rtm-test] event track 1 state negative
[Device A-rtm-test] action 0 cli system-view
```

```
[Device A-rtm-test] action 1 cli bgp 100
[Device A-rtm-test] action 2 cli peer 10.3.1.2 ignore
[Device A-rtm-test] action 3 cli peer 10.3.2.2 ignore
[Device A-rtm-test] user-role network-admin
[Device A-rtm-test] commit
[Device A-rtm-test] quit
```

Verifying the configuration

Shut down Twenty-FiveGigE 1/0/1.

```
[Device A] interface twenty-fivegige 1/0/1
[Device A-Twenty-FiveGigE1/0/1] shutdown
```

Execute the **display bgp peer ipv4** command on Device A to display BGP peer information. If no BGP peer information is displayed, Device A does not have any BGP peers.

Example: Configuring a CLI event monitor policy with EAA environment variables from the CLI

Network configuration

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.

Procedure

Configure an EAA environment variable for IP address assignment. The variable name is **loopback0IP**, and the variable value is **1.1.1.1**.

```
<Sysname> system-view
[Sysname] rtm environment loopback0IP 1.1.1.1
```

Create the CLI-defined policy **test** and enter its view.

```
[Sysname] rtm cli-policy test
```

Add a CLI event that occurs when a command line that contains **loopback0** is executed.

```
[Sysname-rtm-test] event cli async mode execute pattern loopback0
```

Add an action that enters system view when the event occurs.

```
[Sysname-rtm-test] action 0 cli system-view
```

Add an action that creates the interface Loopback 0 and enters loopback interface view.

```
[Sysname-rtm-test] action 1 cli interface loopback 0
```

Add an action that assigns the IP address 1.1.1.1 to Loopback 0. The **loopback0IP** variable is used in the action for IP address assignment.

```
[Sysname-rtm-test] action 2 cli ip address $loopback0IP 24
```

Add an action that sends the matching **loopback0** command with a priority of 0 from the logging facility **local7** when the event occurs.

```
[Sysname-rtm-test] action 3 syslog priority 0 facility local7 msg $_cmd
```

Specify the **network-admin** user role for executing the policy.

```
[Sysname-rtm-test] user-role network-admin
```

Enable the policy.

```
[Sysname-rtm-test] commit
[Sysname-rtm-test] return
<Sysname>
```

Verifying the configuration

Enable the information center to output log messages to the current monitoring terminal.

```
<Sysname> terminal monitor
The current terminal is enabled to display logs.
<Sysname> terminal log level debugging
<Sysname> system-view
[Sysname] info-center enable
Information center is enabled.
```

Execute the **loopback0** command. Verify that the system displays a "loopback0" message and a message that the policy is being executed successfully on the terminal screen.

```
[Sysname] interface loopback0
[Sysname-LoopBack0]%Jan  1 09:46:10:592 2019 Sysname RTM/7/RTM_ACTION: interface
loopback0
%Jan  1 09:46:10:613 2019 Sysname RTM/6/RTM_POLICY: CLI policy test is running
successfully.
```

Verify that Loopback 0 has been created and its IP address is 1.1.1.1.

```
<Sysname-LoopBack0> display interface loopback brief
Brief information on interfaces in route mode:
Link: ADM - administratively down; Stby - standby
Protocol: (s) - spoofing
Interface            Link Protocol Primary IP         Description
Loop0                UP   UP(s)    1.1.1.1

<Sysname-LoopBack0>
```

Monitoring and maintaining processes

About monitoring and maintaining processes

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

- **User process**—Runs in user space. Most system 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. The system 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.

Process monitoring and maintenance tasks at a glance

To monitor and maintain processes, perform the following tasks:

- (Optional.) Starting or stopping a third-party process
  - Starting a third-party process
  - Stopping a third-party process
- Monitoring and maintaining user processes
  - Monitoring and maintaining processes

    The commands in this section apply to both user processes and kernel threads.
  - Monitoring and maintaining user processes

    The commands in this section apply only to user processes.
- Monitoring and maintaining kernel threads
  - Monitoring and maintaining processes

    The commands in this section apply to both user processes and kernel threads.
  - Monitoring and maintaining kernel threads

    The commands in this section apply only to kernel threads.

Starting or stopping a third-party process

About third-party processes

Third-party processes do not start up automatically. Use this feature to start or stop a third-party process, such as Puppet or Chef.

Starting a third-party process

Restrictions and guidelines

If you execute the **third-part-process start** command multiple times but specify the same process name, whether the command can be executed successfully depends on the process. You can use the **display current-configuration | include third-part-process** command to view the result.

Procedure

1. Enter system view.
   system-view
2. Start a third-party process.
   third-part-process start name *process-name* [**arg** *args*]

Stopping a third-party process

1. Display the IDs of third party processes.
   display process all
   This command is available in any view. "Y" in the THIRD field from the output indicates a third-party process, and the PID field indicates the ID of the process.
2. Enter system view.
   system-view
3. Stop a third-party process.
   third-part-process stop pid *pid*&<1-10>
   This command can be used to stop only processes started by the **third-part-process start** command.

Monitoring and maintaining processes

About this task

The commands in this section apply to both user processes and kernel threads. You can use the commands for the following purposes:

- Display the overall memory usage.
- Display the running processes and their memory and CPU usage.
- Locate abnormal processes.

If a process consumes excessive memory or CPU resources, the system identifies the process as an abnormal process.

- If an abnormal process is a user process, troubleshoot the process as described in "Monitoring and maintaining user processes."
- If an abnormal process is a kernel thread, troubleshoot the process as described in "Monitoring and maintaining kernel threads."

Procedure

Execute the following commands in any view.

Task	Command
Display memory usage.	**display memory** [**summary**] [**slot** *slot-number* [**cpu** *cpu-number*]]
Display process state information.	**display process** [**all** | **job** *job-id* | **name** *process-name*] [**slot** *slot-number* [**cpu** *cpu-number*]]
Display CPU usage for all processes.	**display process cpu** [**slot** *slot-number* [**cpu** *cpu-number*]]
Monitor process running state.	**monitor process** [**dumbtty**] [**iteration** *number*] [**slot** *slot-number* [**cpu** *cpu-number*]]
Monitor thread running state.	**monitor thread** [**dumbtty**] [**iteration** *number*] [**slot** *slot-number* [**cpu** *cpu-number*]]

For more information about the **display memory** command, see *Fundamentals Command Reference*.

Monitoring and maintaining user processes

About monitoring and maintaining user processes

Use this feature to monitor abnormal user processes and locate problems.

Configuring core dump

About this task

The core dump feature enables the system to generate a core dump file each time a process crashes until the maximum number of core dump files is reached. A core dump file stores information about the process. You can send the core dump files to Hewlett Packard Enterprise Support to troubleshoot the problems.

Restrictions and guidelines

Core dump files consume storage resources. Enable core dump only for processes that might have problems.

Procedure

Execute the following commands in user view:

1. (Optional.) Specify the directory for saving core dump files.

   exception filepath *directory*

   By default, the directory for saving core dump files is the root directory of the default file system. For more information about the default file system, see file system management in *Fundamentals Configuration Guide*.

2. Enable core dump for a process and specify the maximum number of core dump files, or disable core dump for a process.

   process core { **maxcore** *value* | **off** } { **job** *job-id* | **name** *process-name* }

   By default, a process generates a core dump file for the first exception and does not generate any core dump files for subsequent exceptions.

Display and maintenance commands for user processes

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

Task	Command
Display context information for process exceptions.	**display exception context** [**count** *value*] [**slot** *slot-number* [**cpu** *cpu-number*]]
Display the core dump file directory.	**display exception filepath** [**slot** *slot-number* [**cpu** *cpu-number*]]
Display log information for all user processes.	**display process log** [**slot** *slot-number* [**cpu** *cpu-number*]]
Display memory usage for all user processes.	**display process memory** [**slot** *slot-number* [**cpu** *cpu-number*]]
Display heap memory usage for a user process.	**display process memory heap job** *job-id* [**verbose**] [**slot** *slot-number* [**cpu** *cpu-number*]]
Display memory content starting from a specified memory block for a user process.	**display process memory heap job** *job-id* **address** *starting-address* **length** *memory-length* [**slot** *slot-number* [**cpu** *cpu-number*]]
Display the addresses of memory blocks with a specified size used by a user process.	**display process memory heap job** *job-id* **size** *memory-size* [**offset** *offset-size*] [**slot** *slot-number* [**cpu** *cpu-number*]]
Clear context information for process exceptions.	**reset exception context** [**slot** *slot-number* [**cpu** *cpu-number*]]

Monitoring and maintaining kernel threads

Configuring kernel thread deadloop detection

About this task

Kernel threads share resources. If a kernel thread monopolizes the CPU, other threads cannot run, resulting in a deadloop.

This feature enables the device to detect deadloops. If a thread occupies the CPU for a specific interval, the device determines that a deadloop has occurred and generates a deadloop message.

Restrictions and guidelines

Change kernel thread deadloop detection settings only under the guidance of Hewlett Packard Enterprise Support. Inappropriate configuration can cause system breakdown.

Procedure

1. Enter system view.

   system-view

2. Enable kernel thread deadloop detection.

monitor kernel deadloop enable [**slot** *slot-number* [**cpu** *cpu-number* [**core** *core-number*&<1-64>]]]

By default, kernel thread deadloop detection is enabled.

3. (Optional.) Set the interval for identifying a kernel thread deadloop.

   monitor kernel deadloop time *time* [**slot** *slot-number* [**cpu** *cpu-number*]]

   By default, the threshold for identifying a kernel thread deadloop is 22 seconds.

4. (Optional.) Exclude a kernel thread from kernel thread deadloop detection.

   monitor kernel deadloop exclude-thread *tid* [**slot** *slot-number* [**cpu** *cpu-number*]]

   When enabled, kernel thread deadloop detection monitors all kernel threads by default.

5. (Optional.) Specify the action to be taken in response to a kernel thread deadloop.

   monitor kernel deadloop action { **reboot** | **record-only** } [**slot** *slot-number* [**cpu** *cpu-number*]]

   The default action is **reboot**.

Configuring kernel thread starvation detection

About this task

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 determines 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.

Restrictions and guidelines

Configure kernel thread starvation detection only under the guidance of Hewlett Packard Enterprise Support. Inappropriate configuration can cause system breakdown.

Procedure

1. Enter system view.

   system-view

2. Enable kernel thread starvation detection.

   monitor kernel starvation enable [**slot** *slot-number* [**cpu** *cpu-number*]]

   By default, kernel thread starvation detection is disabled.

3. (Optional.) Set the interval for identifying a kernel thread starvation.

   monitor kernel starvation time *time* [**slot** *slot-number* [**cpu** *cpu-number*]]

   By default, the threshold for identifying a kernel thread starvation is 120 seconds.

4. (Optional.) Exclude a kernel thread from kernel thread starvation detection.

   monitor kernel starvation exclude-thread *tid* [**slot** *slot-number* [**cpu** *cpu-number*]]

   When enabled, kernel thread starvation detection monitors all kernel threads by default.

Display and maintenance commands for kernel threads

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

Task	Command
Display kernel thread deadloop detection configuration.	**display kernel deadloop configuration** [**slot** *slot-number* [**cpu** *cpu-number*]]
Display kernel thread deadloop information.	**display kernel deadloop** *show-number* [*offset*] [**verbose**] [**slot** *slot-number* [**cpu** *cpu-number*]]
Display kernel thread exception information.	**display kernel exception** *show-number* [*offset*] [**verbose**] [**slot** *slot-number* [**cpu** *cpu-number*]]
Display kernel thread reboot information.	**display kernel reboot** *show-number* [*offset*] [**verbose**] [**slot** *slot-number* [**cpu** *cpu-number*]]
Display kernel thread starvation detection configuration.	**display kernel starvation configuration** [**slot** *slot-number* [**cpu** *cpu-number*]]
Display kernel thread starvation information.	**display kernel starvation** *show-number* [*offset*] [**verbose**] [**slot** *slot-number* [**cpu** *cpu-number*]]
Clear kernel thread deadloop information.	**reset kernel deadloop** [**slot** *slot-number* [**cpu** *cpu-number*]]
Clear kernel thread exception information.	**reset kernel exception** [**slot** *slot-number* [**cpu** *cpu-number*]]
Clear kernel thread reboot information.	**reset kernel reboot** [**slot** *slot-number* [**cpu** *cpu-number*]]
Clear kernel thread starvation information.	**reset kernel starvation** [**slot** *slot-number* [**cpu** *cpu-number*]]

Configuring samplers

About sampler

A sampler selects a packet from sequential packets and sends the packet to other service modules for processing. Sampling is useful when you want to limit the volume of traffic to be analyzed. The sampled data is statistically accurate and sampling decreases the impact on the forwarding capacity of the device.

The device supports random sampling mode.

Creating a sampler

1. Enter system view.
   system-view
2. Create a sampler.
   sampler *sampler-name* **mode random packet-interval n-power** *rate*
   By default, no samplers exist.

Display and maintenance commands for a sampler

Execute **display** commands in any view.

Task	Command
Display configuration information about the sampler.	**display sampler** [*sampler-name*] [**slot** *slot-number*]

Samplers and IPv4 NetStream configuration examples

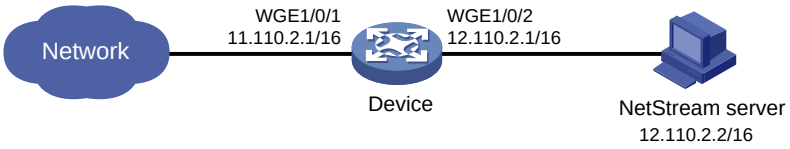
Example: Configuring samplers and IPv4 NetStream

Network configuration

As shown in Figure 93, configure samplers and NetStream as follows:

- Configure IPv4 NetStream on the device to collect statistics on outgoing traffic.
- Send the NetStream data to port 5000 on the NetStream server.
- Configure random sampling in the outbound direction to select one packet randomly from 256 packets on Twenty-FiveGigE 1/0/2.

Figure 93 Network diagram



Configuration procedure

Create sampler **256** in random sampling mode, and set the sampling rate to 8. One packet from 256 packets is selected.

```
[Router] sampler 256 mode random packet-interval n-power 8
```

Enable IPv4 NetStream to use sampler **256** to collect statistics about outgoing traffic on Twenty-FiveGigE 1/0/2.

```
[Device] interface twenty-fivegige 1/0/2
[Device-Twenty-FiveGigE1/0/2] ip netstream outbound
[Device-Twenty-FiveGigE1/0/2] ip netstream outbound sampler 256
[Device-Twenty-FiveGigE1/0/2] quit
```

Configure the address and port number of the NetStream server as the destination for the NetStream data export. Use the default source interface for the NetStream data export.

```
[Device] ip netstream export host 12.110.2.2 5000
```

Verifying the configuration

Display configuration information for sampler **256**.

```
[Router] display sampler 256
 Sampler name: 256
  Mode: Random;  Packet-interval: 8; IsNpower: Y
```

Configuring port mirroring

About port mirroring

Port mirroring copies the packets passing through a port or CPU to a port that connects to a data monitoring device for packet analysis.

Terminology

The following terms are used in port mirroring configuration.

Mirroring source

The mirroring sources can be one or more monitored ports (called source ports) or CPUs (called source CPUs).

Packets passing through mirroring sources are copied to a port connecting to a data monitoring device for packet analysis. The copies are called mirrored packets.

Source device

The device where the mirroring sources reside is called a source device.

Mirroring destination

The mirroring destination connects to a data monitoring device and is the destination port (also known as the monitor port) of mirrored packets. Mirrored packets are sent out of the monitor port to the data monitoring device.

A monitor port might receive multiple copies of a packet when it monitors multiple mirroring sources. For example, two copies of a packet are received on Port A when the following conditions exist:

- Port A is monitoring bidirectional traffic of Port B and Port C on the same device.
- The packet travels from Port B to Port C.

Destination device

The device where the monitor port resides is called the destination device.

Mirroring direction

The mirroring direction specifies the direction of the traffic that is copied on a mirroring source.

- **Inbound**—Copies packets received.
- **Outbound**—Copies packets sent.
- **Bidirectional**—Copies packets received and sent.

Mirroring group

Port mirroring is implemented through mirroring groups. Mirroring groups can be classified into local mirroring groups, remote source groups, and remote destination groups.

Reflector port, egress port, and remote probe VLAN

Reflector ports, remote probe VLANs, and egress ports are used for Layer 2 remote port mirroring. The remote probe VLAN is a dedicated VLAN for transmitting 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.

On port mirroring devices, all ports except source, destination, reflector, and egress ports are called common ports.

Port mirroring classification

Port mirroring can be classified into local port mirroring and remote port mirroring.

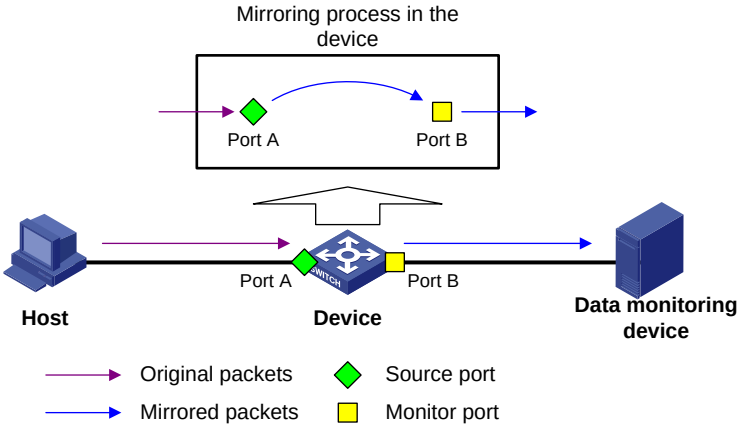
- **Local port mirroring**—The source device is directly connected to a data monitoring device. The source device also acts as the destination device and forwards mirrored packets directly to the data monitoring device.
- **Remote port mirroring**—The source device is not directly connected to a data monitoring device. The source device sends mirrored packets to the destination device, which forwards the packets to the data monitoring device.

  Remote port mirroring can be further classified into Layer 2 and Layer 3 remote port mirroring:

  - **Layer 2 remote port mirroring**—The source device and destination device are on the same Layer 2 network.
  - **Layer 3 remote port mirroring**—The source device and destination device are separated by IP networks.

Local port mirroring

Figure 94 Local port mirroring implementation



As shown in Figure 94, the source port (Port A) and the monitor port (Port B) reside on the same device. Packets received on Port A are copied to Port B. Port B then forwards the packets to the data monitoring device for analysis.

Layer 2 remote port mirroring

In Layer 2 remote port mirroring, the mirroring sources and destination reside on different devices and are in different mirroring groups.

A remote source group is a mirroring group that contains the mirroring sources. A remote destination group is a mirroring group that contains the mirroring destination. Intermediate devices are the devices between the source device and the destination device.

Layer 2 remote port mirroring can be implemented through the reflector port method or the egress port method.

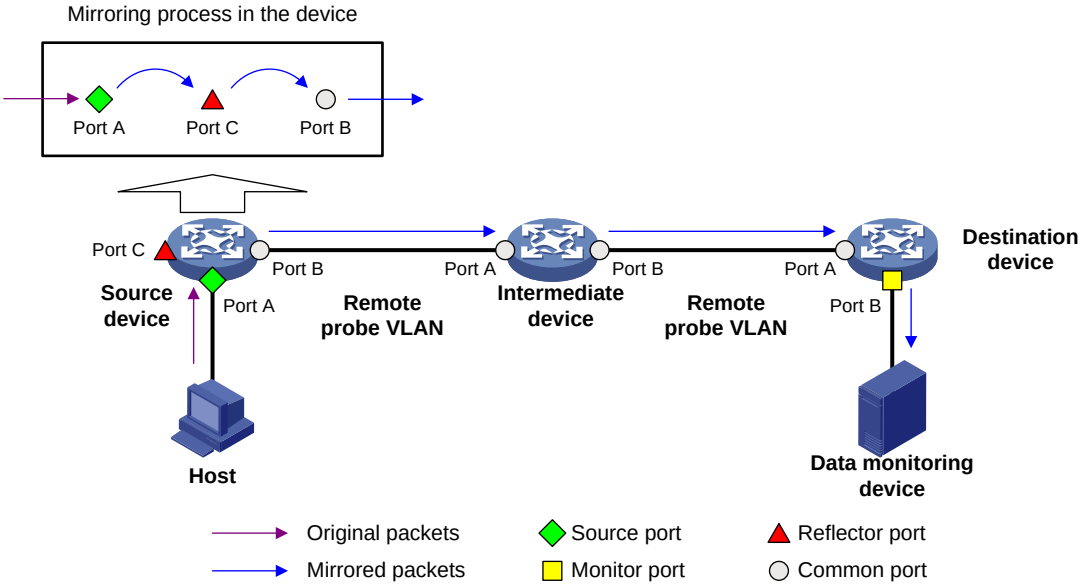
Reflector port method

In Layer 2 remote port mirroring that uses the reflector port method, packets are mirrored as follows:

1. The source device copies packets received on the mirroring sources to the reflector port.
2. The reflector port broadcasts the mirrored packets in the remote probe VLAN.

3. The intermediate devices transmit the mirrored packets to the destination device through the remote probe VLAN.
4. Upon receiving the mirrored packets, the destination device determines whether the ID of the mirrored packets is the same as the remote probe VLAN ID. If the two VLAN IDs match, the destination device forwards the mirrored packets to the data monitoring device through the monitor port.

Figure 95 Layer 2 remote port mirroring implementation through the reflector port method

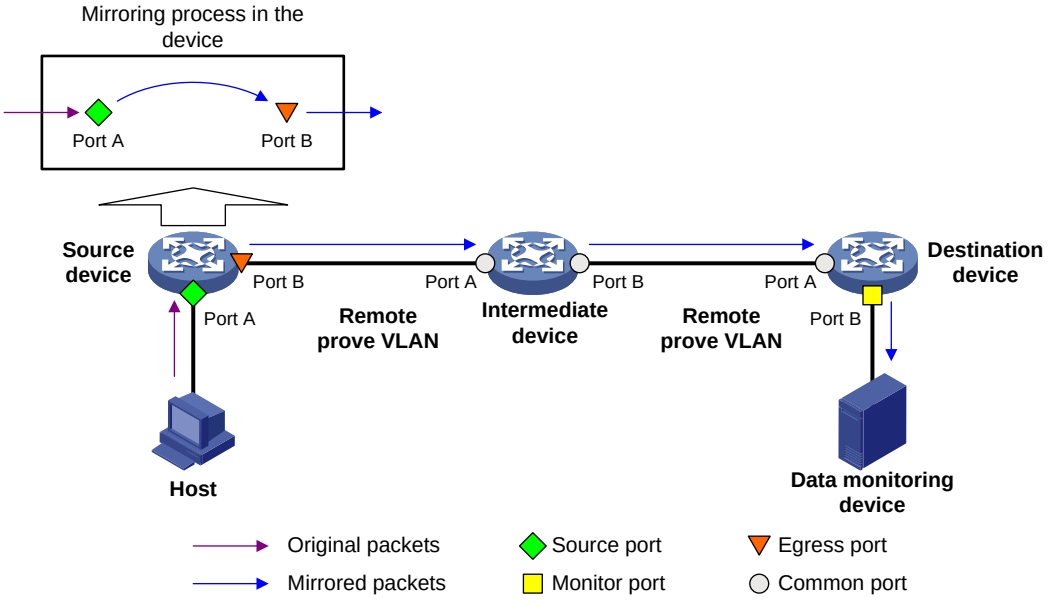


Egress port method

In Layer 2 remote port mirroring that uses the egress port method, packets are mirrored as follows:

1. The source device copies packets received on the mirroring sources to the egress port.
2. The egress port forwards the mirrored packets to the intermediate devices.
3. The intermediate devices flood the mirrored packets in the remote probe VLAN and transmit the mirrored packets to the destination device.
4. Upon receiving the mirrored packets, the destination device determines whether the ID of the mirrored packets is the same as the remote probe VLAN ID. If the two VLAN IDs match, the destination device forwards the mirrored packets to the data monitoring device through the monitor port.

Figure 96 Layer 2 remote port mirroring implementation through the egress port method



Layer 3 remote port mirroring

Layer 3 remote port mirroring is implemented through configuring a local mirroring group on both the source device and the destination device.

Configure the mirroring sources and destination for the local mirroring groups on the source device and destination device as follows:

- On the source device:
  - Configure the ports to be monitored as source ports.
  - Configure the CPUs to be monitored as source CPUs.
  - Configure the tunnel interface through which mirrored packets are forwarded to the destination device as the monitor port.
- On the destination device:
  - Configure the physical port corresponding to the tunnel interface as the source port.
  - Configure the port that connects to the data monitoring device as the monitor port.

For example, in a network as shown in Figure 97, Layer 3 remote port mirroring works as follows:

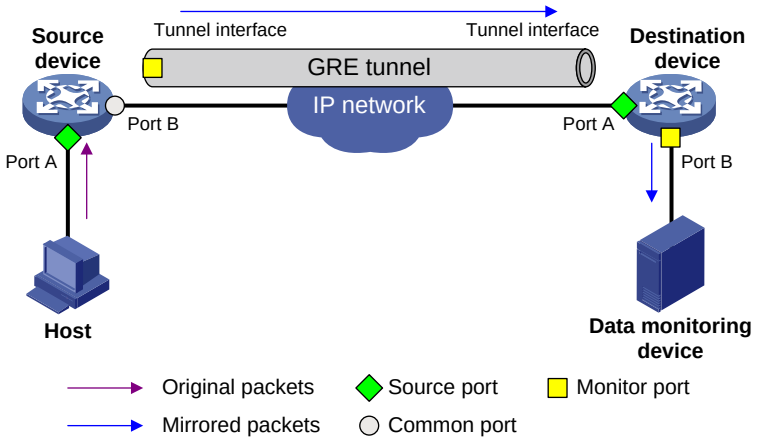
1. The source device sends one copy of a packet received on the source port (Port A) to the tunnel interface.

   The tunnel interface acts as the monitor port in the local mirroring group created on the source device.
2. The tunnel interface on the source device forwards the mirrored packet to the tunnel interface on the destination device through the GRE tunnel.
3. The destination device receives the mirrored packet from the physical interface of the tunnel interface.

   The tunnel interface acts as the source port in the local mirroring group created on the destination device.
4. The physical interface of the tunnel interface sends one copy of the packet to the monitor port (Port B).
5. The monitor port (Port B) forwards the packet to the data monitoring device.

For more information about GRE tunnels and tunnel interfaces, see *Layer 3—IP Services Configuration Guide*.

Figure 97 Layer 3 remote port mirroring implementation



Restrictions and guidelines: Port mirroring configuration

The reflector port method for Layer 2 remote port mirroring can be used to implement local port mirroring with multiple data monitoring devices.

In the reflector port method, the reflector port broadcasts mirrored packets in the remote probe VLAN. By assigning the ports that connect to data monitoring devices to the remote probe VLAN, you can implement local port mirroring to mirror packets to multiple data monitoring devices. The egress port method cannot implement local port mirroring in this way.

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.

Configuring local port mirroring

Restrictions and guidelines for local port mirroring configuration

A local mirroring group takes effect only after it is configured with the monitor port and mirroring sources.

A Layer 3 aggregate interface cannot be configured as the monitor port for a local mirroring group.

Local port mirroring tasks at a glance

To configure local port mirroring, perform the following tasks:

1. Configuring mirroring sources

   Choose one of the following tasks:

   - Configuring source ports
   - Configuring source CPUs

2. Configuring the monitor port

Creating a local mirroring group

1. Enter system view.
   system-view
2. Create a local mirroring group.
   mirroring-group *group-id* **local**

Configuring mirroring sources

Restrictions and guidelines for mirroring source configuration

When you configure source ports for a local mirroring group, follow these restrictions and guidelines:

- A mirroring group can contain multiple source ports.
- A port can be assigned to different mirroring groups as follows:
  - When acting as a source port for unidirectional mirroring, the port can be assigned to up to four mirroring groups.
  - When acting as a source port for bidirectional mirroring, the port can be assigned to up to two mirroring groups.
  - When acting as a source port for unidirectional and bidirectional mirroring, the port can be assigned to up to three mirroring groups. One mirroring group is used for bidirectional mirroring and the other two for unidirectional mirroring.
- A source port cannot be configured as a reflector port, egress port, or monitor port.

A local mirroring group can contain multiple source CPUs.

Configuring source ports

- Configure source ports in system view:
  a. Enter system view.
     system-view
  b. Configure source ports for a local mirroring group.
     mirroring-group *group-id* **mirroring-port** *interface-list* { **both** | **inbound** | **outbound** }
     By default, no source port is configured for a local mirroring group.
- Configure source ports in interface view:
  c. Enter system view.
     system-view
  d. Enter interface view.
     interface *interface-type interface-number*
  e. Configure the port as a source port for a local mirroring group.
     mirroring-group *group-id* **mirroring-port** { **both** | **inbound** | **outbound** }
     By default, a port does not act as a source port for any local mirroring groups.

Configuring source CPUs

1. Enter system view.
   system-view
2. Configure source CPUs for a local mirroring group.

mirroring-group *group-id* **mirroring-cpu slot** *slot-number-list* **inbound**

By default, no source CPU is configured for a local mirroring group.

The device supports mirroring only inbound traffic of a source CPU.

Configuring the monitor port

Restrictions and guidelines

Do not enable the spanning tree feature on the monitor port.

Only one monitor port can be configured for a local mirroring group.

For a Layer 2 aggregate interface configured as the monitor port of a mirroring group, do not configure its member ports as source ports of the mirroring group.

Use a monitor port only for port mirroring, so the data monitoring device receives only the mirrored traffic.

Procedure

- Configure the monitor port in system view:
  a. Enter system view.
  system-view
  b. Configure the monitor port for a local mirroring group.
  mirroring-group *group-id* **monitor-port** *interface-list*
  By default, no monitor port is configured for a local mirroring group.
- Configure the monitor port in interface view:
  c. Enter system view.
  system-view
  d. Enter interface view.
  interface *interface-type interface-number*
  e. Configure the port as the monitor port for a mirroring group.
  mirroring-group *group-id* **monitor-port**
  By default, a port does not act as the monitor port for any local mirroring groups.

Configuring Layer 2 remote port mirroring

Restrictions and guidelines for Layer 2 remote port mirroring configuration

To ensure successful traffic mirroring, configure devices in the order of the destination device, the intermediate devices, and the source device.

If intermediate devices exist, configure the intermediate devices to allow the remote probe VLAN to pass through.

For a mirrored packet to successfully arrive at the remote destination device, make sure its VLAN ID is not removed or changed.

Do not configure both MVRP and Layer 2 remote port mirroring. Otherwise, MVRP might register the remote probe VLAN with incorrect ports, which would cause the monitor port to receive undesired copies. For more information about MVRP, see *Layer 2—LAN Switching Configuration Guide*.

To monitor the bidirectional traffic of a source port, 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*.

Layer 2 remote port mirroring with reflector port configuration task list

Configuring the destination device

1. Creating a remote destination group
2. Configuring the monitor port
3. Configuring the remote probe VLAN
4. Assigning the monitor port to the remote probe VLAN

Configuring the source device

1. Creating a remote source group
2. Configuring mirroring sources

   Choose one of the following tasks:
   - Configuring source ports
   - Configuring source CPUs
3. Configuring the reflector port
4. Configuring the remote probe VLAN

Layer 2 remote port mirroring with egress port configuration task list

Configuring the destination device

1. Creating a remote destination group
2. Configuring the monitor port
3. Configuring the remote probe VLAN
4. Assigning the monitor port to the remote probe VLAN

Configuring the source device

1. Creating a remote source group
2. Configuring mirroring sources

   Choose one of the following tasks:
   - Configuring source ports
   - Configuring source CPUs
3. Configuring the egress port
4. Configuring the remote probe VLAN

Creating a remote destination group

Restrictions and guidelines

Perform this task on the destination device only.

Procedure

1. Enter system view.
   system-view
2. Create a remote destination group.
   mirroring-group *group-id* **remote-destination**

Configuring the monitor port

Restrictions and guidelines for monitor port configuration

Perform this task on the destination device only.

Do not enable the spanning tree feature on the monitor port.

Only one monitor port can be configured for a remote destination group.

For a Layer 2 aggregate interface configured as the monitor port of a mirroring group, do not configure its member ports as source ports of the mirroring group.

Use a monitor port only for port mirroring, so the data monitoring device receives only the mirrored traffic.

A monitor port can belong to only one mirroring group.

Configuring the monitor port in system view

1. Enter system view.
   system-view
2. Configure the monitor port for a remote destination group.
   mirroring-group *group-id* **monitor-port** *interface-list*

   By default, no monitor port is configured for a remote destination group.

Configuring the monitor port in interface view

1. Enter system view.
   system-view
2. Enter interface view.
   interface *interface-type interface-number*
3. Configure the port as the monitor port for a remote destination group.
   mirroring-group *group-id* **monitor-port**

   By default, a port does not act as the monitor port for any remote destination groups.

Configuring the remote probe VLAN

Restrictions and guidelines

This task is required on the both the source and destination devices.

Only an existing static VLAN can be configured as a remote probe VLAN.

When a VLAN is configured as a remote probe VLAN, use the remote probe VLAN for port mirroring exclusively.

Configure the same remote probe VLAN for the remote source group and the remote destination group.

Procedure

1. Enter system view.

system-view

2. Configure the remote probe VLAN for the remote source or destination group.
   mirroring-group *group-id* **remote-probe vlan** *vlan-id*
   By default, no remote probe VLAN is configured for a remote source or destination group.

Assigning the monitor port to the remote probe VLAN

Restrictions and guidelines

Perform this task on the destination device only.

Procedure

1. Enter system view.
   system-view
2. Enter the interface view of the monitor port.
   interface *interface-type interface-number*
3. Assign the port to the remote probe VLAN.
   - Assign an access port to the remote probe VLAN.
     port access vlan *vlan-id*
   - Assign a trunk port to the remote probe VLAN.
     port trunk permit vlan *vlan-id*
   - Assign a hybrid port to the remote probe VLAN.
     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*.

Creating a remote source group

Restrictions and guidelines

Perform this task on the source device only.

Procedure

1. Enter system view.
   system-view
2. Create a remote source group.
   mirroring-group *group-id* **remote-source**

Configuring mirroring sources

Restrictions and guidelines for mirroring source configuration

Perform this task on the source device only.

When you configure source ports for a remote source group, follow these restrictions and guidelines:

- Do not assign a source port of a mirroring group to the remote probe VLAN of the mirroring group.
- A mirroring group can contain multiple source ports.
- A port can be assigned to different mirroring groups as follows:

    - When acting as a source port for unidirectional mirroring, the port can be assigned to up to four mirroring groups.
    - When acting as a source port for bidirectional mirroring, the port can be assigned to up to two mirroring groups.
    - When acting as a source port for unidirectional and bidirectional mirroring, the port can be assigned to up to three mirroring groups. One mirroring group is used for bidirectional mirroring and the other two for unidirectional mirroring.
- A source port cannot be configured as a reflector port, monitor port, or egress port.

A mirroring group can contain multiple source CPUs.

Configuring source ports

- Configure source ports in system view:
  a. Enter system view.
  system-view
  b. Configure source ports for a remote source group.
  mirroring-group *group-id* **mirroring-port** *interface-list* { **both** | **inbound** | **outbound** }
  By default, no source port is configured for a remote source group.
- Configure source ports in interface view:
  c. Enter system view.
  system-view
  d. Enter interface view.
  interface *interface-type interface-number*
  e. Configure the port as a source port for a remote source group.
  mirroring-group *group-id* **mirroring-port** { **both** | **inbound** | **outbound** }
  By default, a port does not act as a source port for any remote source groups.

Configuring source CPUs

1. Enter system view.
   system-view
2. Configure source CPUs for a remote source group.
   mirroring-group *group-id* **mirroring-cpu slot** *slot-number-list* **inbound**
   By default, no source CPU is configured for a remote source group.
   The device supports mirroring only inbound traffic of a source CPU.

Configuring the reflector port

Restrictions and guidelines for reflector port configuration

Perform this task on the source device only.

The port to be configured as a reflector port must be a port not in use. Do not connect a network cable to a reflector port.

When a port is configured as a reflector port, the default settings of the port are automatically restored. You cannot configure other features on the reflector port.

If an IRF port is bound to only one physical interface, do not configure the physical interface as a reflector port. Otherwise, the IRF might split.

A remote source group supports only one reflector port.

Configuring the reflector port in system view

1. Enter system view.
   system-view
2. Configure the reflector port for a remote source group.
   mirroring-group *group-id* **reflector-port** *interface-type interface-number*
   By default, no reflector port is configured for a remote source group.

Configuring the reflector port in interface view

1. Enter system view.
   system-view
2. Enter interface view.
   interface *interface-type interface-number*
3. Configure the port as the reflector port for a remote source group.
   mirroring-group *group-id* **reflector-port**
   By default, a port does not act as the reflector port for any remote source groups.

Configuring the egress port

Restrictions and guidelines for egress port configuration

Perform this task on the source device only.

Disable the following features on the egress port:

- Spanning tree.
- 802.1X.
- IGMP snooping.
- Static ARP.
- MAC address learning.

A port of an existing mirroring group cannot be configured as an egress port.

A mirroring group supports only one egress port.

Configuring the egress port in system view

1. Enter system view.
   system-view
2. Configure the egress port for a remote source group.
   mirroring-group *group-id* **monitor-egress** *interface-type interface-number*
   By default, no egress port is configured for a remote source group.
3. Enter the egress port view.
   interface *interface-type interface-number*
4. Assign the egress port to the remote probe VLAN.
   - Assign a trunk port to the remote probe VLAN.
     port trunk permit vlan *vlan-id*
   - Assign a hybrid port to the remote probe VLAN.
     port hybrid vlan *vlan-id* { **tagged** | **untagged** }

For more information about the **port trunk permit vlan** and **port hybrid vlan** commands, see *Layer 2—LAN Switching Command Reference*.

Configuring the egress port in interface view

1. Enter system view.
   system-view
2. Enter interface view.
   interface *interface-type interface-number*
3. Configure the port as the egress port for a remote source group.
   mirroring-group *group-id* **monitor-egress**
   By default, a port does not act as the egress port for any remote source groups.

Configuring Layer 3 remote port mirroring (in tunnel mode)

Restrictions and guidelines for Layer 3 remote port mirroring configuration

To implement Layer 3 remote port mirroring, you must configure a unicast routing protocol on the intermediate devices to ensure Layer 3 reachability between the source and destination devices.

Layer 3 remote port mirroring tasks at a glance

Configuring the source device

1. Configuring local mirroring groups
2. Configuring mirroring sources
   Choose one of the following tasks:
   - Configuring source ports
   - Configuring source CPUs
3. Configuring the monitor port

Configuring the destination device

1. Configuring local mirroring groups
2. Configuring mirroring sources
3. Configuring the monitor port

Prerequisites for Layer 3 remote port mirroring

Before configuring Layer 3 remote mirroring, complete the following tasks:

- Create a tunnel interface and a GRE tunnel.
- Configure the source and destination addresses of the tunnel interface as the IP addresses of the physical interfaces on the source and destination devices, respectively.

For more information about tunnel interfaces, see *Layer 3—IP Services Configuration Guide*.

Configuring local mirroring groups

Restrictions and guidelines

Configure a local mirroring group on both the source device and the destination device.

Procedure

1. Enter system view.
   system-view
2. Create a local mirroring group.
   mirroring-group *group-id* **local**

Configuring mirroring sources

Restrictions and guidelines for mirroring source configuration

When you configure source ports for a local mirroring group, follow these restrictions and guidelines:

- On the source device, configure the ports you want to monitor as the source ports. On the destination device, configure the physical interface corresponding to the tunnel interface as the source port.
- A port can be assigned to different mirroring groups as follows:
  - When acting as a source port for unidirectional mirroring, the port can be assigned to up to four mirroring groups.
  - When acting as a source port for bidirectional mirroring, the port can be assigned to up to two mirroring groups.
  - When acting as a source port for unidirectional and bidirectional mirroring, the port can be assigned to up to three mirroring groups. One mirroring group is used for bidirectional mirroring and the other two for unidirectional mirroring
- A source port cannot be configured as a reflector port, egress port, or monitor port.

When you configure source CPUs for a local mirroring group, follow these restrictions and guidelines:

- Perform this task on the source device only.
- A mirroring group can contain multiple source CPUs.

Configuring source ports

- Configure source ports in system view:
  a. Enter system view.
  system-view
  b. Configure source ports for a local mirroring group.
  mirroring-group *group-id* **mirroring-port** *interface-list* { **both** | **inbound** | **outbound** }
  By default, no source port is configured for a local mirroring group.
- Configure source ports in interface view:
  c. Enter system view.
  system-view
  d. Enter interface view.
  interface *interface-type interface-number*
  e. Configure the port as a source port for a local mirroring group.

mirroring-group *group-id* **mirroring-port** { **both** | **inbound** | **outbound** }

By default, a port does not act as a source port for any local mirroring groups.

Configuring source CPUs

1. Enter system view.

   system-view

2. Configure source CPUs for a local mirroring group.

   mirroring-group *group-id* **mirroring-cpu slot** *slot-number-list* **inbound**

   By default, no source CPU is configured for a local mirroring group.

   The device supports mirroring only the inbound traffic of a source CPU.

Configuring the monitor port

Restrictions and guidelines for monitor port configuration

On the source device, configure a tunnel interface as a monitor port. On the destination device, configure the port that connects to a data monitoring device as a monitor port.

On the source device, only one tunnel interface can be configured as the monitor port for a local mirroring group.

On the destination device, do not enable the spanning tree feature on a monitor port.

On the destination device, only one monitor port can be configured for a local mirroring group.

Use a monitor port only for port mirroring, so the data monitoring device receives only the mirrored traffic.

If the monitor port of a local mirroring group is an aggregate interface, make sure the member ports in the service loopback group and the source ports in the local mirroring group belong to the same interface group. Execute the **display drv system 9** command in probe view. In the command output, interfaces in the same pipe belong to the same interface group.

Procedure

- Configure the monitor port in system view:
  a. Enter system view.

     system-view

  b. Configure the monitor port for a local mirroring group.

     mirroring-group *group-id* **monitor-port** *interface-list*

     By default, no monitor port is configured for a local mirroring group.
- Configure the monitor port in interface view:
  c. Enter system view.

     system-view

  d. Enter interface view.

     interface *interface-type interface-number*

  e. Configure the port as the monitor port for a local mirroring group.

     mirroring-group *group-id* **monitor-port**

     By default, a port does not act as the monitor port for any local mirroring groups.

Configuring Layer 3 remote port mirroring (in ERSPAN mode)

Restrictions and guidelines for Layer 3 remote port mirroring in ERSPAN mode configuration

To implement Layer 3 remote port mirroring in Encapsulated Remote Switch Port Analyzer (ERSPAN) mode, perform the following tasks:

1. On the source device, create a local mirroring group and configure the mirroring sources, the monitor port, and the encapsulation parameters for mirrored packets.

   The mirrored packet sent to the monitor port is first encapsulated in a GRE packet with a protocol number of 0x88BE. The GRE packet is then encapsulated in a delivery protocol by using the encapsulation parameters and routed to the destination data monitoring device.
2. On all devices from source to destination, configure a unicast routing protocol to ensure Layer 3 reachability between the devices.

In Layer 3 remote port mirroring in ERSPAN mode, the data monitoring device must be able to remove the outer headers to obtain the original mirrored packets for analysis.

Layer 3 remote port mirroring tasks at a glance

To configure Layer 3 remote port mirroring in ERSPAN mode, perform the following tasks:

1. Creating a local mirroring group on the source device
2. Configuring mirroring sources

   Choose one of the following tasks:
   - Configuring source ports
   - Configuring source CPUs
3. Configuring the monitor port

Creating a local mirroring group on the source device

1. Enter system view.

   system-view
2. Create a local mirroring group.

   mirroring-group *group-id* **local**

Configuring mirroring sources

Restrictions and guidelines for mirroring source configuration

When you configure source ports for the local mirroring group, follow these restrictions and guidelines:

- An interface can be assigned to a maximum of four mirroring groups as a unidirectional source port, to a maximum of two mirroring groups as a bidirectional source port, or to one mirroring group as a bidirectional source port and to two mirroring groups as a unidirectional source port.
- A source port cannot be configured as a reflector port, egress port, or monitor port.

A local mirroring group can contain multiple source CPUs.

Configuring source ports

- Configure source ports in system view:
  a. Enter system view.
     system-view
  b. Configure source ports for a local mirroring group.
     mirroring-group *group-id* **mirroring-port** *interface-list* { **both** | **inbound** | **outbound** }
     By default, no source port is configured for a local mirroring group.
- Configure source ports in interface view:
  c. Enter system view.
     system-view
  d. Enter interface view.
     interface *interface-type interface-number*
  e. Configure the port as a source port for a local mirroring group.
     mirroring-group *group-id* **mirroring-port** { **both** | **inbound** | **outbound** }

Configuring source CPUs

1. Enter system view.
   system-view
2. Configure source CPUs for a local mirroring group.
   mirroring-group *group-id* **mirroring-cpu slot** *slot-number-list* **inbound**
   By default, no source CPU is configured for a local mirroring group.

Configuring the monitor port

Restrictions and guidelines

Do not enable the spanning tree feature on the monitor port.

Only one monitor port can be configured for a local mirroring group.

Use a monitor port only for port mirroring, so the data monitoring device receives only the mirrored traffic.

If the monitor port of a local mirroring group is an aggregate interface, make sure the member ports in the aggregate interface and the source ports in the local mirroring group belong to the same interface group. Execute the **display drv system 9** command in probe view. In the command output, interfaces in the same pipe belong to the same interface group.

Procedure

- Configure the monitor port in system view:
  a. Enter system view.
     system-view
  b. Configure the monitor port in a local mirroring group and specify the encapsulation parameters.
     mirroring-group *group-id* **monitor-port** *interface-type interface-number* **destination-ip** *destination-ip-address* **source-ip** *source-ip-address* [**dscp** *dscp-value* | **vlan** *vlan-id* | **vrf-instance** *vrf-name*] *
     By default, no monitor port is configured for a local mirroring group.

- Configure the monitor port in interface view:
  c. Enter system view.
     system-view
  d. Enter interface view.
     interface *interface-type interface-number*
  e. Specify the port as the monitor port in a local mirroring group and configure the encapsulation parameters in a local mirroring group.
     mirroring-group *group-id* **monitor-port destination-ip** *destination-ip-address* **source-ip** *source-ip-address* [**dscp** *dscp-value* | **vlan** *vlan-id* | **vrf-instance** *vrf-name*] *
     By default, a port does not act as the monitor port for any local mirroring groups.

Display and maintenance commands for port mirroring

Execute **display** commands in any view.

Task	Command
Display mirroring group information.	**display mirroring-group** { *group-id* | **all** | **local** | **remote-destination** | **remote-source** }

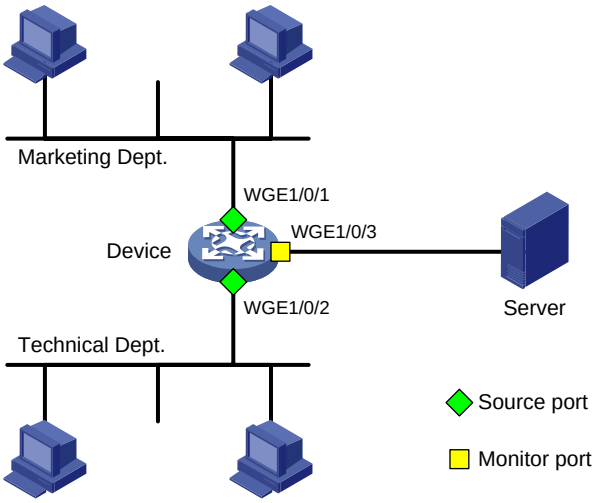
Port mirroring configuration examples

Example: Configuring local port mirroring (in source port mode)

Network configuration

As shown in Figure 98, configure local port mirroring in source port mode to enable the server to monitor the bidirectional traffic of the two departments.

Figure 98 Network diagram



Procedure

Create local mirroring group 1.

```
<Device> system-view
[Device] mirroring-group 1 local
```

Configure Twenty-FiveGigE 1/0/1 and Twenty-FiveGigE 1/0/2 as source ports for local mirroring group 1.

```
[Device] mirroring-group 1 mirroring-port twenty-fivegige 1/0/1 twenty-fivegige 1/0/2
both
```

Configure Twenty-FiveGigE 1/0/3 as the monitor port for local mirroring group 1.

```
[Device] mirroring-group 1 monitor-port twenty-fivegige 1/0/3
```

Disable the spanning tree feature on the monitor port (Twenty-FiveGigE 1/0/3).

```
[Device] interface twenty-fivegige 1/0/3
[Device-Twenty-FiveGigE1/0/3] undo stp enable
[Device-Twenty-FiveGigE1/0/3] quit
```

Verifying the configuration

Verify the mirroring group configuration.

```
[Device] display mirroring-group all
Mirroring group 1:
    Type: Local
    Status: Active
    Mirroring port:
        Twenty-FiveGigE1/0/1  Both
        Twenty-FiveGigE1/0/2  Both
    Monitor port: Twenty-FiveGigE1/0/3
```

Example: Configuring local port mirroring (in source CPU mode)

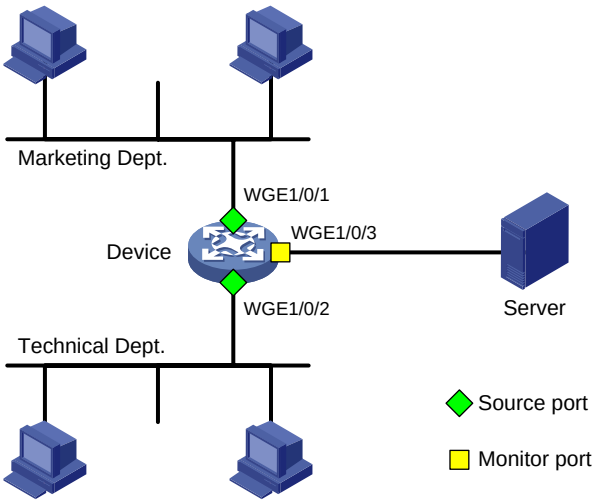
Network configuration

As shown in Figure 99, Twenty-FiveGigE 1/0/1 and Twenty-FiveGigE 1/0/2 are located on the card in slot 1.

Configure local port mirroring in source CPU mode to enable the server to monitor all packets matching the following criteria:

- Received by the Marketing Department and the Technical Department.
- Processed by the CPU in slot 1 of the device.

Figure 99 Network diagram



Procedure

Create local mirroring group 1.

```
<Device> system-view
[Device] mirroring-group 1 local
```

Configure the CPU in slot 1 of the device as a source CPU for local mirroring group 1.

```
[Device] mirroring-group 1 mirroring-cpu slot 1 inbound
```

Configure Twenty-FiveGigE 1/0/3 as the monitor port for local mirroring group 1.

```
[Device] mirroring-group 1 monitor-port twenty-fivegige 1/0/3
```

Disable the spanning tree feature on the monitor port (Twenty-FiveGigE 1/0/3).

```
[Device] interface twenty-fivegige 1/0/3
[Device-Twenty-FiveGigE1/0/3] undo stp enable
[Device-Twenty-FiveGigE1/0/3] quit
```

Verifying the configuration

Verify the mirroring group configuration.

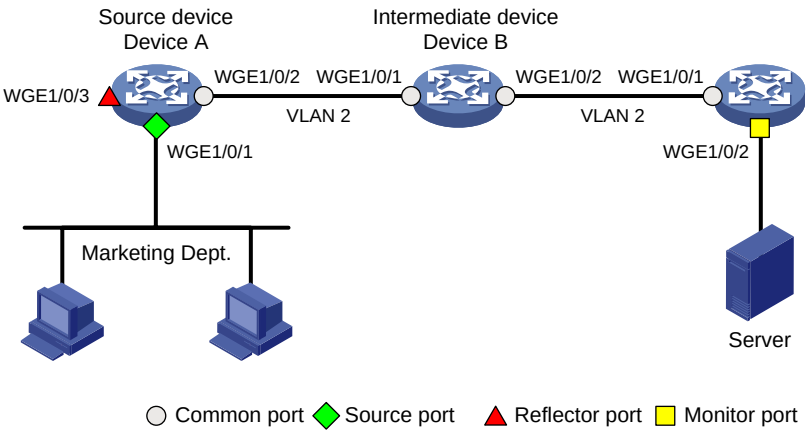
```
[Device] display mirroring-group all
Mirroring group 1:
    Type: Local
    Status: Active
    Mirroring CPU:
        Slot 1  Inbound
    Monitor port: Twenty-FiveGigE1/0/3
```

Example: Configuring Layer 2 remote port mirroring (with reflector port)

Network configuration

As shown in Figure 100, configure Layer 2 remote port mirroring to enable the server to monitor the bidirectional traffic of the Marketing Department.

Figure 100 Network diagram



Procedure

1. Configure Device C (the destination device):

   # Configure Twenty-FiveGigE 1/0/1 as a trunk port, and assign the port to VLAN 2.

   ```
   <DeviceC> system-view
   [DeviceC] interface twenty-fivegige 1/0/1
   [DeviceC-Twenty-FiveGigE1/0/1] port link-type trunk
   [DeviceC-Twenty-FiveGigE1/0/1] port trunk permit vlan 2
   [DeviceC-Twenty-FiveGigE1/0/1] 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 for the mirroring group.

   ```
   [DeviceC] mirroring-group 2 remote-probe vlan 2
   ```

   # Configure Twenty-FiveGigE 1/0/2 as the monitor port for the mirroring group.

   ```
   [DeviceC] interface twenty-fivegige 1/0/2
   [DeviceC-Twenty-FiveGigE1/0/2] mirroring-group 2 monitor-port
   ```

   # Disable the spanning tree feature on Twenty-FiveGigE 1/0/2.

   ```
   [DeviceC-Twenty-FiveGigE1/0/2] undo stp enable
   ```

   # Assign Twenty-FiveGigE 1/0/2 to VLAN 2.

   ```
   [DeviceC-Twenty-FiveGigE1/0/2] port access vlan 2
   [DeviceC-Twenty-FiveGigE1/0/2] 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 Twenty-FiveGigE 1/0/1 as a trunk port, and assign the port to VLAN 2.

```
[DeviceB] interface twenty-fivegige 1/0/1
[DeviceB-Twenty-FiveGigE1/0/1] port link-type trunk
[DeviceB-Twenty-FiveGigE1/0/1] port trunk permit vlan 2
[DeviceB-Twenty-FiveGigE1/0/1] quit
```

Configure Twenty-FiveGigE 1/0/2 as a trunk port, and assign the port to VLAN 2.

```
[DeviceB] interface twenty-fivegige 1/0/2
[DeviceB-Twenty-FiveGigE1/0/2] port link-type trunk
[DeviceB-Twenty-FiveGigE1/0/2] port trunk permit vlan 2
[DeviceB-Twenty-FiveGigE1/0/2] 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 for the mirroring group.

```
[DeviceA] mirroring-group 1 remote-probe vlan 2
```

Configure Twenty-FiveGigE 1/0/1 as a source port for the mirroring group.

```
[DeviceA] mirroring-group 1 mirroring-port twenty-fivegige 1/0/1 both
```

Configure Twenty-FiveGigE 1/0/3 as the reflector port for the mirroring group.

```
[DeviceA] mirroring-group 1 reflector-port twenty-fivegige 1/0/3
This operation may delete all settings made on the interface. Continue? [Y/N]: y
```

Configure Twenty-FiveGigE 1/0/2 as a trunk port, and assign the port to VLAN 2.

```
[DeviceA] interface twenty-fivegige 1/0/2
[DeviceA-Twenty-FiveGigE1/0/2] port link-type trunk
[DeviceA-Twenty-FiveGigE1/0/2] port trunk permit vlan 2
[DeviceA-Twenty-FiveGigE1/0/2] quit
```

Verifying the configuration

Verify the mirroring group configuration on Device C.

```
[DeviceC] display mirroring-group all
Mirroring group 2:
    Type: Remote destination
    Status: Active
    Monitor port: Twenty-FiveGigE1/0/2
    Remote probe VLAN: 2
```

Verify the mirroring group configuration on Device A.

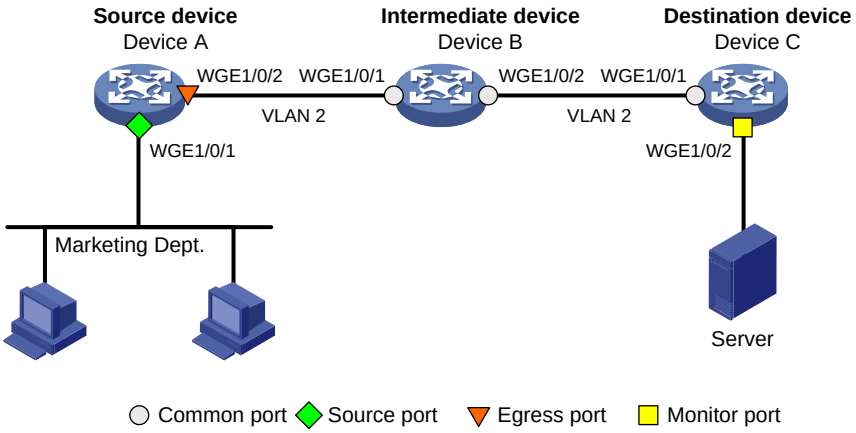
```
[DeviceA] display mirroring-group all
Mirroring group 1:
    Type: Remote source
    Status: Active
    Mirroring port:
        Twenty-FiveGigE1/0/1  Both
    Reflector port: Twenty-FiveGigE1/0/3
    Remote probe VLAN: 2
```

Example: Configuring Layer 2 remote port mirroring (with egress port)

Network configuration

On the Layer 2 network shown in Figure 101, configure Layer 2 remote port mirroring to enable the server to monitor the bidirectional traffic of the Marketing Department.

Figure 101 Network diagram



Procedure

1. Configure Device C (the destination device):

   # Configure Twenty-FiveGigE 1/0/1 as a trunk port, and assign the port to VLAN 2.

   ```
   <DeviceC> system-view
   [DeviceC] interface twenty-fivegige 1/0/1
   [DeviceC-Twenty-FiveGigE1/0/1] port link-type trunk
   [DeviceC-Twenty-FiveGigE1/0/1] port trunk permit vlan 2
   [DeviceC-Twenty-FiveGigE1/0/1] 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 for the mirroring group.

   ```
   [DeviceC] mirroring-group 2 remote-probe vlan 2
   ```

Configure Twenty-FiveGigE 1/0/2 as the monitor port for the mirroring group.

```
[DeviceC] interface twenty-fivegige 1/0/2
[DeviceC-Twenty-FiveGigE1/0/2] mirroring-group 2 monitor-port
```

Disable the spanning tree feature on Twenty-FiveGigE 1/0/2.

```
[DeviceC-Twenty-FiveGigE1/0/2] undo stp enable
```

Assign Twenty-FiveGigE 1/0/2 to VLAN 2 as an access port.

```
[DeviceC-Twenty-FiveGigE1/0/2] port access vlan 2
[DeviceC-Twenty-FiveGigE1/0/2] 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 Twenty-FiveGigE 1/0/1 as a trunk port, and assign the port to VLAN 2.

```
[DeviceB] interface twenty-fivegige 1/0/1
[DeviceB-Twenty-FiveGigE1/0/1] port link-type trunk
[DeviceB-Twenty-FiveGigE1/0/1] port trunk permit vlan 2
[DeviceB-Twenty-FiveGigE1/0/1] quit
```

Configure Twenty-FiveGigE 1/0/2 as a trunk port, and assign the port to VLAN 2.

```
[DeviceB] interface twenty-fivegige 1/0/2
[DeviceB-Twenty-FiveGigE1/0/2] port link-type trunk
[DeviceB-Twenty-FiveGigE1/0/2] port trunk permit vlan 2
[DeviceB-Twenty-FiveGigE1/0/2] 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 Twenty-FiveGigE 1/0/1 as a source port for the mirroring group.

```
[DeviceA] mirroring-group 1 mirroring-port twenty-fivegige 1/0/1 both
```

Configure Twenty-FiveGigE 1/0/2 as the egress port for the mirroring group.

```
[DeviceA] mirroring-group 1 monitor-egress twenty-fivegige 1/0/2
```

Configure Twenty-FiveGigE 1/0/2 as a trunk port, and assign the port to VLAN 2.

```
[DeviceA] interface twenty-fivegige 1/0/2
[DeviceA-Twenty-FiveGigE1/0/2] port link-type trunk
[DeviceA-Twenty-FiveGigE1/0/2] port trunk permit vlan 2
```

Disable the spanning tree feature on the port.

```
[DeviceA-Twenty-FiveGigE1/0/2] undo stp enable
[DeviceA-Twenty-FiveGigE1/0/2] quit
```

Verifying the configuration

Verify the mirroring group configuration on Device C.

```
[DeviceC] display mirroring-group all
Mirroring group 2:
    Type: Remote destination
    Status: Active
    Monitor port: Twenty-FiveGigE1/0/2
    Remote probe VLAN: 2
```

Verify the mirroring group configuration on Device A.

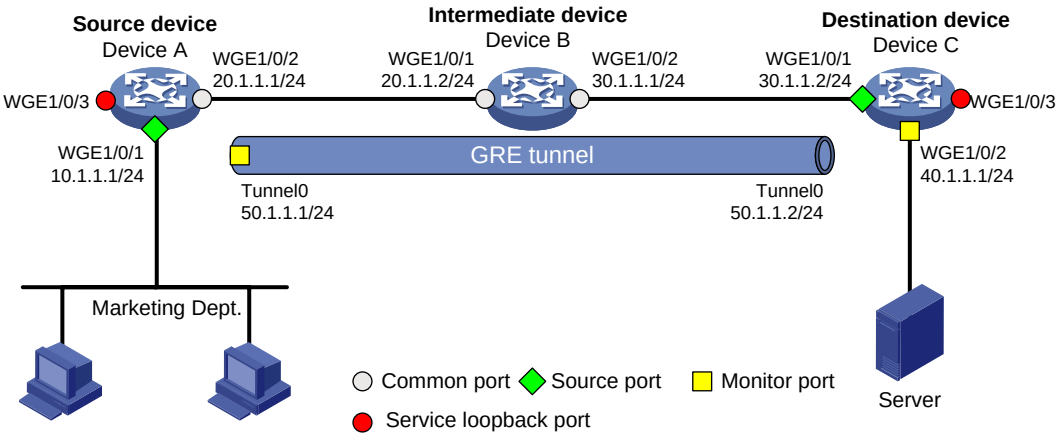
```
[DeviceA] display mirroring-group all
Mirroring group 1:
    Type: Remote source
    Status: Active
    Mirroring port:
        Twenty-FiveGigE1/0/1  Both
    Monitor egress port: Twenty-FiveGigE1/0/2
    Remote probe VLAN: 2
```

Example: Configuring Layer 3 remote port mirroring in tunnel mode

Network configuration

On a Layer 3 network shown in Figure 102, configure Layer 3 remote port mirroring to enable the server to monitor the bidirectional traffic of the Marketing Department.

Figure 102 Network diagram



Procedure

1. Configure IP addresses for the tunnel interfaces and related ports on the devices. (Details not shown.)
2. Configure Device A (the source device):

   # Create service loopback group 1 and specify the unicast tunnel service for the group.

   ```
   <DeviceA> system-view
   [DeviceA] service-loopback group 1 type tunnel
   ```

   # Assign Twenty-FiveGigE 1/0/3 to service loopback group 1.

```
[DeviceA] interface twenty-fivegige 1/0/3
[DeviceA-Twenty-FiveGigE1/0/3] port service-loopback group 1
All configurations on the interface will be lost. Continue?[Y/N]:y
[DeviceA-Twenty-FiveGigE1/0/3] quit
```

Create tunnel interface Tunnel 0 that operates in GRE mode, and configure an IP address and subnet mask for the interface.

```
[DeviceA] interface tunnel 0 mode gre
[DeviceA-Tunnel0] ip address 50.1.1.1 24
```

Configure source and destination IP addresses for Tunnel 0.

```
[DeviceA-Tunnel0] source 20.1.1.1
[DeviceA-Tunnel0] destination 30.1.1.2
[DeviceA-Tunnel0] quit
```

Enable the OSPF protocol.

```
[DeviceA] ospf 1
[DeviceA-ospf-1] area 0
[DeviceA-ospf-1-area-0.0.0.0] network 10.1.1.0 0.0.0.255
[DeviceA-ospf-1-area-0.0.0.0] network 20.1.1.0 0.0.0.255
[DeviceA-ospf-1-area-0.0.0.0] quit
[DeviceA-ospf-1] quit
```

Create local mirroring group 1.

```
[DeviceA] mirroring-group 1 local
```

Configure Twenty-FiveGigE 1/0/1 as a source port and Tunnel 0 as the monitor port of local mirroring group 1.

```
[DeviceA] mirroring-group 1 mirroring-port twenty-fivegige 1/0/1 both
[DeviceA] mirroring-group 1 monitor-port tunnel 0
```

3. Enable the OSPF protocol on Device B (the intermediate device).

```
<DeviceB> system-view
[DeviceB] ospf 1
[DeviceB-ospf-1] area 0
[DeviceB-ospf-1-area-0.0.0.0] network 20.1.1.0 0.0.0.255
[DeviceB-ospf-1-area-0.0.0.0] network 30.1.1.0 0.0.0.255
[DeviceB-ospf-1-area-0.0.0.0] quit
[DeviceB-ospf-1] quit
```

4. Configure Device C (the destination device):

Create service loopback group 1 and specify the unicast tunnel service for the group.

```
<DeviceC> system-view
[DeviceC] service-loopback group 1 type tunnel
```

Assign Twenty-FiveGigE 1/0/3 to service loopback group 1.

```
[DeviceC] interface twenty-fivegige 1/0/3
[DeviceC-Twenty-FiveGigE1/0/3] port service-loopback group 1
All configurations on the interface will be lost. Continue?[Y/N]:y
[DeviceC-Twenty-FiveGigE1/0/3] quit
```

Create tunnel interface Tunnel 0 that operates in GRE mode, and configure an IP address and subnet mask for the interface.

```
[DeviceC] interface tunnel 0 mode gre
[DeviceC-Tunnel0] ip address 50.1.1.2 24
```

Configure source and destination IP addresses for Tunnel 0.

```
[DeviceC-Tunnel0] source 30.1.1.2
```

```
[DeviceC-Tunnel0] destination 20.1.1.1
[DeviceC-Tunnel0] quit
```

Enable the OSPF protocol.

```
[DeviceC] ospf 1
[DeviceC-ospf-1] area 0
[DeviceC-ospf-1-area-0.0.0.0] network 30.1.1.0 0.0.0.255
[DeviceC-ospf-1-area-0.0.0.0] network 40.1.1.0 0.0.0.255
[DeviceC-ospf-1-area-0.0.0.0] quit
[DeviceC-ospf-1] quit
```

Create local mirroring group 1.

```
[DeviceC] mirroring-group 1 local
```

Configure Twenty-FiveGigE 1/0/1 as a source port for local mirroring group 1.

```
[DeviceC] mirroring-group 1 mirroring-port twenty-fivegige 1/0/1 inbound
```

Configure Twenty-FiveGigE 1/0/2 as the monitor port for local mirroring group 1.

```
[DeviceC] mirroring-group 1 monitor-port twenty-fivegige 1/0/2
```

Verifying the configuration

Verify the mirroring group configuration on Device A.

```
[DeviceA] display mirroring-group all
Mirroring group 1:
    Type: Local
    Status: Active
    Mirroring port:
        Twenty-FiveGigE1/0/1  Both
    Monitor port: Tunnel0
```

Display information about all mirroring groups on Device C.

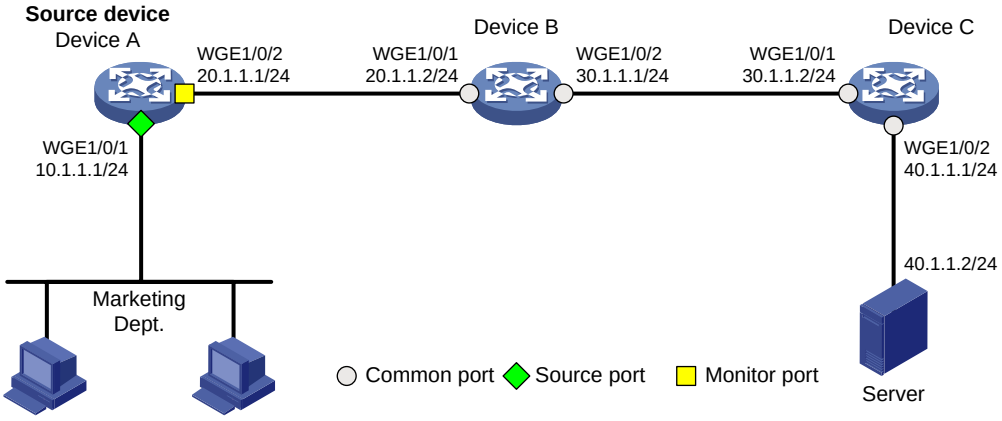
```
[DeviceC] display mirroring-group all
Mirroring group 1:
    Type: Local
    Status: Active
    Mirroring port:
        Twenty-FiveGigE1/0/1  Inbound
        Monitor port: Twenty-FiveGigE1/0/2
```

Example: Configuring Layer 3 remote port mirroring in ERSPAN mode

Network configuration

On a Layer 3 network shown in Figure 103, configure Layer 3 remote port mirroring in ERSPAN mode to enable the server to monitor the bidirectional traffic of the Marketing Department.

Figure 103 Network diagram



Procedure

1. Configure IP addresses for the interfaces as shown in Figure 103. (Details not shown.)
2. Configure Device A (the source device):

   # Enable the OSPF protocol.

   ```
   [DeviceA] ospf 1
   [DeviceA-ospf-1] area 0
   [DeviceA-ospf-1-area-0.0.0.0] network 10.1.1.0 0.0.0.255
   [DeviceA-ospf-1-area-0.0.0.0] network 20.1.1.0 0.0.0.255
   [DeviceA-ospf-1-area-0.0.0.0] quit
   [DeviceA-ospf-1] quit
   ```

   # Create local mirroring group 1.

   ```
   [DeviceA] mirroring-group 1 local
   ```

   # Configure Twenty-FiveGigE 1/0/1 as a source port.

   ```
   [DeviceA] mirroring-group 1 mirroring-port twenty-fivegige 1/0/1 both
   ```

   # Configure Twenty-FiveGigE 1/0/2 as the monitor port. Specify the destination and source IP addresses for mirrored packets as 40.1.1.2 and 20.1.1.1, respectively.

   ```
   [DeviceA] mirroring-group 1 monitor-port twenty-fivegige 1/0/2 destination-ip
   40.1.1.2 source-ip 20.1.1.1
   ```

3. Enable the OSPF protocol on Device B.

   ```
   <DeviceB> system-view
   [DeviceB] ospf 1
   [DeviceB-ospf-1] area 0
   [DeviceB-ospf-1-area-0.0.0.0] network 20.1.1.0 0.0.0.255
   [DeviceB-ospf-1-area-0.0.0.0] network 30.1.1.0 0.0.0.255
   [DeviceB-ospf-1-area-0.0.0.0] quit
   [DeviceB-ospf-1] quit
   ```

4. Enable the OSPF protocol on Device C.

   ```
   [DeviceC] ospf 1
   [DeviceC-ospf-1] area 0
   [DeviceC-ospf-1-area-0.0.0.0] network 30.1.1.0 0.0.0.255
   [DeviceC-ospf-1-area-0.0.0.0] network 40.1.1.0 0.0.0.255
   [DeviceC-ospf-1-area-0.0.0.0] quit
   [DeviceC-ospf-1] quit
   ```

Verifying the configuration

Verify the mirroring group configuration on Device A.

```
[DeviceA] display mirroring-group all
Mirroring group 1:
    Type: Local
    Status: Active
    Mirroring port:
        Twenty-FiveGigE1/0/1  Both
    Monitor port: Twenty-FiveGigE1/0/2
                  Encapsulation: Destination IP address 40.1.1.2
                                 Source IP address 20.1.1.1
                                 Destination MAC address 000f-e241-5e5b
```

Configuring flow mirroring

About flow mirroring

Flow mirroring copies packets matching a class to a destination for packet analyzing and monitoring. It is implemented through QoS.

To implement flow mirroring through QoS, perform the following tasks:

- Define traffic classes and configure match criteria to classify packets to be mirrored. Flow mirroring allows you to flexibly classify packets to be analyzed by defining match criteria.
- Configure traffic behaviors to mirror the matching packets to the specified destination.

You can configure an action to mirror the matching packets to one of the following destinations:

- **Interface**—The matching packets are copied to an interface and then forwarded to a data monitoring device for analysis.
- **CPU**—The matching packets are copied to the CPU of an IRF member device. The CPU analyzes the packets or delivers them to upper layers.
- **gRPC**—The matching packets are copied to a directly-connected Google Remote Procedure Call (gRPC) network management server for further analysis.
- **In-band network telemetry (INT) processor**—The matching packets are copied to the INT processor.

For more information about QoS policies, traffic classes, and traffic behaviors, see *ACL and QoS Configuration Guide*.

Restrictions and guidelines: Flow mirroring configuration

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

To apply a QoS policy to a Layer 3 Ethernet interface or subinterface for outbound flow mirroring, do not configure VLAN-based match criteria in the traffic class of the policy.

You can enable sampling for only one of the following features on the device:

- Mirroring.
- NetStream.
- IPv6 NetStream.
- sFlow.
- INT.
- Telemetry stream.
- MOD.

For more information about NetStream, IPv6 NetStream, and sFlow, see *Network Management and Monitoring Configuration Guide.* For more information about INT, telemetry stream, and MOD, see *Telemetry Configuration Guide.*

Flow mirroring tasks at a glance

To configure flow mirroring, perform the following tasks:

1. Configuring a traffic class

   A traffic class defines the criteria that filters the traffic to be mirrored.
2. Configuring a traffic behavior

   A traffic behavior specifies mirroring destinations.
3. Configuring a QoS policy
4. Applying a QoS policy

   Choose one of the following tasks:
   - Applying a QoS policy to an interface
   - Applying a QoS policy to a VLAN
   - Applying a QoS policy globally
   - Applying a QoS policy to the control plane

Configuring a traffic class

1. Enter system view.

   system-view
2. Create a class and enter class view.

   traffic classifier *classifier-name* [**operator** { **and** | **or** }]
3. Configure match criteria.

   if-match *match-criteria*

   By default, no match criterion is configured in a traffic class.
4. (Optional.) Display traffic class information.

   display traffic classifier

   This command is available in any view.

Configuring a traffic behavior

Procedure

1. Enter system view.

   system-view
2. Create a traffic behavior and enter traffic behavior view.

   traffic behavior *behavior-name*
3. Configure mirroring destinations for the traffic behavior. Choose one option as needed:
   - Mirror traffic to interfaces.

     Mirror traffic to the specified interface:

     mirror-to interface *interface-type interface-number* [**sampler** *sampler-name*] [**truncation**] [**loopback** | [**destination-ip** *destination-ip-address* **source-ip** *source-ip-address* [**dscp** *dscp-value* | **vlan** *vlan-id* | **vrf-instance** *vrf-name*] *]

     Mirror traffic to interfaces based on routes matching the specified destination IP address:

mirror-to interface destination-ip *destination-ip-address* **source-ip** *source-ip-address* [**sampler** *sampler-name*] [**truncation**] [**dscp** *dscp-value* | **vlan** *vlan-id* | **vrf-instance** *vrf-name*] *

By default, no mirroring actions exist to mirror traffic to interfaces.

If traffic is mirrored to an aggregate interface, make sure the member ports in the aggregate interface and the incoming interface of the original traffic belong to the same interface group. Execute the **display drv system 9** command in probe view. In the command output, interfaces in the same pipe belong to the same interface group.

You can mirror traffic to a maximum of four Ethernet interfaces or Layer 2 aggregate interfaces in a traffic behavior.

- Mirror traffic to the CPU.

  mirror-to cpu

  By default, no mirroring actions exist to mirror traffic to the CPU.
- Mirror traffic to the directly-connected gRPC network management server.

  mirror-to grpc

  By default, no mirroring actions exist to mirror traffic to the directly-connected gRPC network management server.
- Mirror traffic to the INT processor.

  mirror-to ifa-processor [**sampler** *sampler-name*]

  By default, no mirroring actions exist to mirror traffic to the INT processor.

  For more information about the INT processor, see INT configuration in *Telemetry Configuration Guide*.

4. (Optional.) Display traffic behavior configuration.

   display traffic behavior

   This command is available in any view.

Configuring a QoS policy

1. Enter system view.

   system-view
2. Create a QoS policy and enter QoS policy view.

   qos policy *policy-name*
3. Associate a class with a traffic behavior in the QoS policy.

   classifier *classifier-name* **behavior** *behavior-name*

   By default, no traffic behavior is associated with a class.
4. (Optional.) Display QoS policy configuration.

   display qos policy

   This command is available in any view.

Applying a QoS policy

Applying a QoS policy to an interface

Restrictions and guidelines

You can apply a QoS policy to an interface to mirror the traffic of the interface.

A policy can be applied to multiple interfaces.

In one traffic direction of an interface, only one QoS policy can be applied.

To apply a QoS policy to the outbound traffic of an interface, make sure mirroring actions do not coexist with non-mirroring actions in the same traffic behavior to avoid conflicts.

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

Procedure

1. Enter system view.
   system-view
2. Enter interface view.
   interface *interface-type interface-number*
3. Apply a policy to the interface.
   qos apply policy *policy-name* { **inbound** | **outbound** }
4. (Optional.) Display the QoS policy applied to the interface.
   display qos policy interface
   This command is available in any view.

Applying a QoS policy to a VLAN

Restrictions and guidelines

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

Procedure

1. Enter system view.
   system-view
2. Apply a QoS policy to a VLAN.
   qos vlan-policy *policy-name* **vlan** *vlan-id-list* { **inbound** | **outbound** }
3. (Optional.) Display the QoS policy applied to the VLAN.
   display qos vlan-policy
   This command is available in any view.

Applying a QoS policy globally

Restrictions and guidelines

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

Procedure

1. Enter system view.
   system-view
2. Apply a QoS policy globally.
   qos apply policy *policy-name* **global** { **inbound** | **outbound** }
3. (Optional.) Display global QoS policies.
   display qos policy global
   This command is available in any view.

Applying a QoS policy to the control plane

Restrictions and guidelines

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

Procedure

1. Enter system view.
   system-view
2. Enter control plane view.
   control-plane slot *slot-number*
3. Apply a QoS policy to the control plane.
   qos apply policy *policy-name* **inbound**
4. (Optional.) Display QoS policies applied to the control plane
   display qos policy control-plane
   This command is available in any view.

Flow mirroring configuration examples

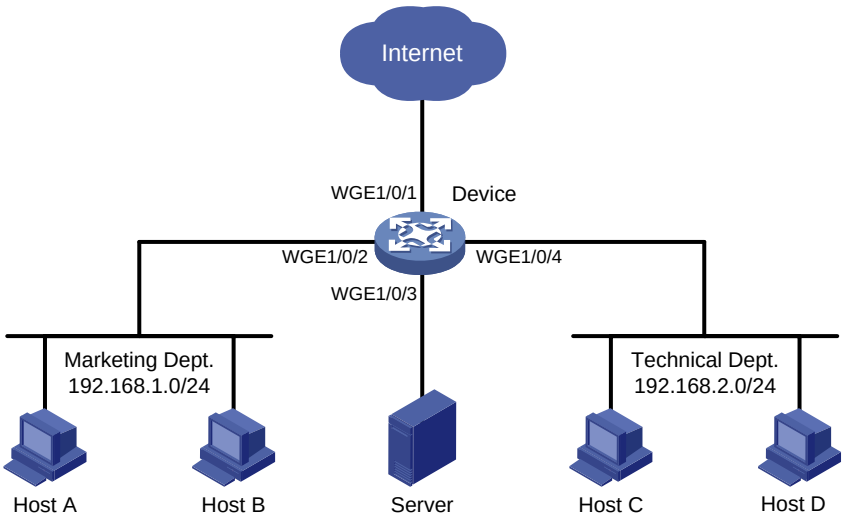
Example: Configuring flow mirroring

Network configuration

As shown in Figure 104, configure flow mirroring so that the server can monitor the following traffic:

- All traffic that the Technical Department sends to access the Internet.
- IP traffic that the Technical Department sends to the Marketing Department during working hours (8:00 to 18:00) on weekdays.

Figure 104 Network diagram



Procedure

Create working hour range **work**, in which working hours are from 8:00 to 18:00 on weekdays.

```
<Device> system-view
[Device] time-range work 8:00 to 18:00 working-day
```

Create IPv4 advanced ACL 3000 to allow packets from the Technical Department to access the Internet and the Marketing Department during working hours.

```
[Device] acl advanced 3000
[Device-acl-ipv4-adv-3000] rule permit tcp source 192.168.2.0 0.0.0.255 destination-port eq www
[Device-acl-ipv4-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
[Device-acl-ipv4-adv-3000] quit
```

Create traffic class **tech_c**, and configure the match criterion as ACL 3000.

```
[Device] traffic classifier tech_c
[Device-classifier-tech_c] if-match acl 3000
[Device-classifier-tech_c] quit
```

Create traffic behavior **tech_b**, configure the action of mirroring traffic to Twenty-FiveGigE 1/0/3.

```
[Device] traffic behavior tech_b
[Device-behavior-tech_b] mirror-to interface twenty-fivegige 1/0/3
[Device-behavior-tech_b] quit
```

Create QoS policy **tech_p**, and associate traffic class **tech_c** with traffic behavior **tech_b** in the QoS policy.

```
[Device] qos policy tech_p
[Device-qospolicy-tech_p] classifier tech_c behavior tech_b
[Device-qospolicy-tech_p] quit
```

Apply QoS policy **tech_p** to the incoming packets of Twenty-FiveGigE 1/0/4.

```
[Device] interface twenty-fivegige 1/0/4
[Device-Twenty-FiveGigE1/0/4] qos apply policy tech_p inbound
[Device-Twenty-FiveGigE1/0/4] quit
```

Verifying the configuration

Verify that the server can monitor the following traffic:

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

(Details not shown.)

Configuring NetStream

About NetStream

NetStream is an accounting technology that provides statistics on a per-flow basis. An IPv4 flow is defined by the following 7-tuple elements:

- Destination IP address.
- Source IP address.
- Destination port number.
- Source port number.
- Protocol number.
- ToS.
- Inbound or outbound interface.

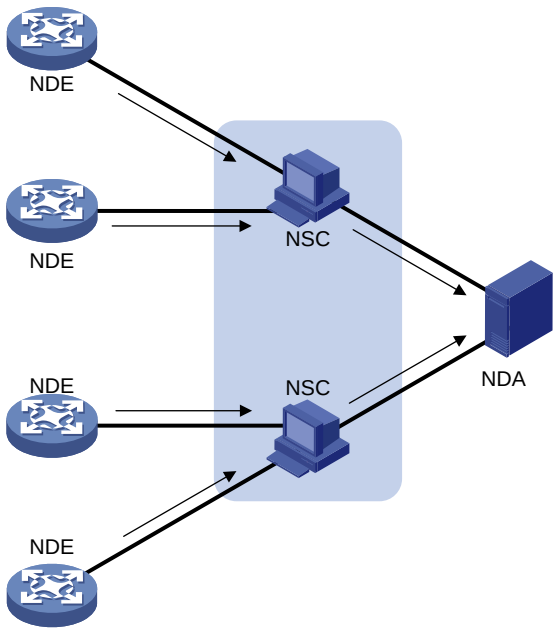
NetStream architecture

A typical NetStream system includes the following elements:

- **NetStream data exporter**—A device configured with NetStream. The NDE provides the following functions:
  - Classifies traffic flows by using the 7-tuple elements.
  - Collects data from the classified flows.
  - Aggregates and exports the data to the NSC.
- **NetStream collector**—A program running on an operating system. The NSC parses the packets received from the NDEs, and saves the data to its database.
- **NetStream data analyzer**—A network traffic analyzing tool. Based on the data in NSC, the NDA generates reports for traffic billing, network planning, and attack detection and monitoring. The NDA can collect data from multiple NSCs. Typically, the NDA features a Web-based system for easy operation.

NSC and NDA are typically integrated into a NetStream server.

Figure 105 NetStream system



NetStream flow aging

NetStream uses flow aging to enable the NDE to export NetStream data to NetStream servers. NetStream creates a NetStream entry for each flow for storing the flow statistics in the cache.

When a flow is aged out, the NDE performs the following operations:

- Exports the summarized data to NetStream servers in a specific format.
- Clears NetStream entry information in the cache.

NetStream supports the following flow aging methods:

- Periodical aging.
- Forced aging.

Periodical aging

Periodical aging uses the following methods:

- **Inactive flow aging**—A flow is inactive if no packet arrives for the NetStream entry within the inactive flow aging timer. When the timer expires, the following events occur:
  - The inactive flow entry is aged out.
  - The statistics of the flow are sent to NetStream servers and are cleared in the cache. The statistics can no longer be displayed by using the **`display ip netstream cache`** command.

  This method ensures that inactive flow entries are cleared from the cache in a timely manner so new entries can be cached.
- **Active flow aging**—A flow is active if packets arrive for the NetStream entry within the active flow aging timer. When the timer expires, the statistics of the active flow are exported to NetStream servers. The device continues to collect active flow statistics.

  This method periodically exports the statistics of active flows to NetStream servers.

Forced aging

Execute the **`reset ip netstream statistics`** command to clear the NetStream cache immediately and export the cached entries to NetStream servers.

NetStream data export

Traditional data export

Traditional NetStream collects the statistics of each flow and exports the statistics to NetStream servers.

This method consumes more bandwidth and CPU than the aggregation method, and it requires a large cache size.

Aggregation data export

NetStream aggregation merges the flow statistics according to the aggregation criteria of an aggregation mode, and it sends the summarized data to NetStream servers. The NetStream aggregation data export uses less bandwidth than the traditional data export.

Table 36 lists the available aggregation modes. In each mode, the system merges statistics for multiple flows into statistics for one aggregate flow if each aggregation criterion is of the same value. The system records the statistics for the aggregate flow. These aggregation modes work independently and can take effect concurrently.

For example, when the aggregation mode configured on the NDE is protocol-port, NetStream aggregates the statistics of flow entries by protocol number, source port, and destination port. Four NetStream entries record four TCP flows with the same destination address, source port, and destination port, but with different source addresses. In the aggregation mode, only one NetStream aggregation entry is created and sent to NetStream servers.

Table 36 NetStream aggregation modes

Aggregation mode	Aggregation criteria
Protocol-port aggregation	• Protocol number • Source port • Destination port
Source-prefix aggregation	• Source AS number • Source address mask length • Source prefix (source network address) • Inbound interface index
Destination-prefix aggregation	• Destination AS number • Destination address mask length • Destination prefix (destination network address) • Outbound interface index
Prefix aggregation	• Source AS number • Destination AS number • Source address mask length • Destination address mask length • Source prefix • Destination prefix • Inbound interface index • Outbound interface index

Aggregation mode	Aggregation criteria
Prefix-port aggregation	• Source prefix • Destination prefix • Source address mask length • Destination address mask length • ToS • Protocol number • Source port • Destination port • Inbound interface index • Outbound interface index
ToS-source-prefix aggregation	• ToS • Source AS number • Source prefix • Source address mask length • Inbound interface index
ToS-destination-prefix aggregation	• ToS • Destination AS number • Destination address mask length • Destination prefix • Outbound interface index
ToS-prefix aggregation	• ToS • Source AS number • Source prefix • Source address mask length • Destination AS number • Destination address mask length • Destination prefix • Inbound interface index • Outbound interface index
ToS-protocol-port aggregation	• ToS • Protocol type • Source port • Destination port • Inbound interface index • Outbound interface index

NetStream export formats

NetStream exports data in UDP datagrams in one of the following formats:

- **Version 5**—Exports original statistics collected based on the 7-tuple elements and does not support the NetStream aggregation data export. The packet format is fixed and cannot be extended.
- **Version 8**—Supports the NetStream aggregation data export. The packet format is fixed and cannot be extended.
- **Version 9**—Based on a template that can be configured according to the template formats defined in RFCs. Version 9 supports exporting the NetStream aggregation data and collecting statistics about BGP next hop and MPLS packets.
- **Version 10**—Similar to version 9. The difference between version 9 and version 10 is that version 10 export format is compliant with the IPFIX standard.

NetStream filtering

NetStream filtering uses an ACL to identify packets. Whether NetStream collects data for identified packets depends on the action in the matching rule.

- NetStream collects data for packets that match permit rules in the ACL.
- NetStream does not collect data for packets that match deny rules in the ACL.

For more information about ACL, see *ACL and QoS Configuration Guide*.

NetStream sampling

NetStream sampling collects statistics on fewer packets and is useful when the network has a large amount of traffic. NetStream on sampled traffic lessens the impact on the device's performance. For more information about sampling, see "Configuring samplers."

Enabling NetStream sampling takes effect for both IPv4 and IPv6 NetStream.

Protocols and standards

RFC 5101, *Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information*

Restrictions: Hardware compatibility with NetStream

NetStream is supported only on the HPE FlexFabric 5945 2-slot switch(JQ075A), HPE FlexFabric 5945 4-slot switch (JQ076A), and HPE FlexFabric 5945 32QSFP28 switch (JQ077A).

The service interfaces near power module side on the rear panel of the switch are used for internal loopback of NetStream traffic. After NetStream is enabled on an interface on the front panel, you cannot use service interfaces on the rear panel and the settings on these service interfaces become invalid. Before enabling NetStream on an interface, clear the configurations on the service interfaces.

Restrictions and guidelines: NetStream configuration

You can enable sampling for only one of the following features on the device:

- NetStream.
- Mirroring.
- sFlow.
- INT.
- Telemetry stream.
- MOD.

For more information about mirroring and sFlow, see *Network Management and Monitoring Configuration Guide.* For more information about INT, telemetry stream, and MOD, see *Telemetry Configuration Guide.*

NetStream tasks at a glance

To configure NetStream, perform the following tasks:

1. Enabling NetStream
2. (Optional.) Configuring NetStream filtering
3. (Optional.) Configuring NetStream sampling
4. (Optional.) Configuring the NetStream data export format
5. (Optional.) Configuring the refresh rate for NetStream version 9 or version 10 template
6. (Optional.) Configuring VXLAN-aware NetStream
7. (Optional.) Configuring NetStream flow aging
   - Configuring periodical flow aging
   - Configuring forced flow aging
8. Configuring the NetStream data export
   a. Configuring the NetStream traditional data export
   b. (Optional.) Configuring the NetStream aggregation data export

Enabling NetStream

Restrictions and guidelines

This feature is available on only Layer 2 Ethernet interfaces and Layer 3 Ethernet interfaces. It is not available on Layer 3 Ethernet subinterfaces or other types of interfaces.

Procedure

1. Enter system view.
   system-view
2. Enter interface view.
   interface *interface-type interface-number*
3. Enable NetStream on the interface.
   ip netstream { **inbound** | **outbound** }
   By default, NetStream is disabled on an interface.

Configuring NetStream filtering

About this task

NetStream filtering uses an ACL to identify packets.

- To enable NetStream to collect statistics for specific flows, use the ACL permit statements to identify these flows
- To disable NetStream from collecting statistics for specific flows, use the ACL deny statements to identify these flows.

Restrictions and guidelines

When NetStream filtering and sampling are both configured, packets are filtered first, and then the permitted packets are sampled.

The NetStream filtering feature does not take effect on MPLS packets.

If you use NetStream filtering on the interface where IPv4 and IPv6 Netstream are enabled in the same direction, make sure NetStream filtering is enabled for both IPv4 and IPv6 in this direction. For more information about IPv6 NetStream, see *Network Management and Monitoring Configuration Guide*.

Procedure

1. Enter system view.
   system-view
2. Enter interface view.
   interface *interface-type interface-number*
3. Enable NetStream filtering on the interface.
   ip netstream inbound filter acl *acl-number*
   By default, NetStream filtering is disabled. NetStream collects statistics of all IPv4 packets passing through the interface.

Configuring NetStream sampling

Restrictions and guidelines

By default, NetStream collects all data of target flows. If the flow traffic is heavy, NetStream is resource-consuming and can cause high CPU usage, which impacts the device forwarding performance. NetStream sampling is helpful to decrease the NetStream traffic volume. If the collected statistics can basically reflect the network status, you can enable this feature and set a proper sampling rate. The higher the sampling rate, the less impact on device performance.

If NetStream sampling and filtering are both configured, packets are filtered first, and then the permitted packets are sampled.

Procedure

1. Enter system view.
   system-view
2. Create a sampler.
   sampler *sampler-name* **mode random packet-interval n-power** *rate*
   For more information about a sampler, see "Configuring samplers."
3. Enter interface view.
   interface *interface-type interface-number*
4. Enable NetStream sampling.
   ip netstream { inbound | **outbound } sampler** *sampler-name*
   This command enables both IPv4 NetStream sampling and IPv6 NetStream sampling.
   By default, NetStream sampling is disabled.

Configuring the NetStream data export format

About this task

When you configure the NetStream data export format, you can also specify the following settings:

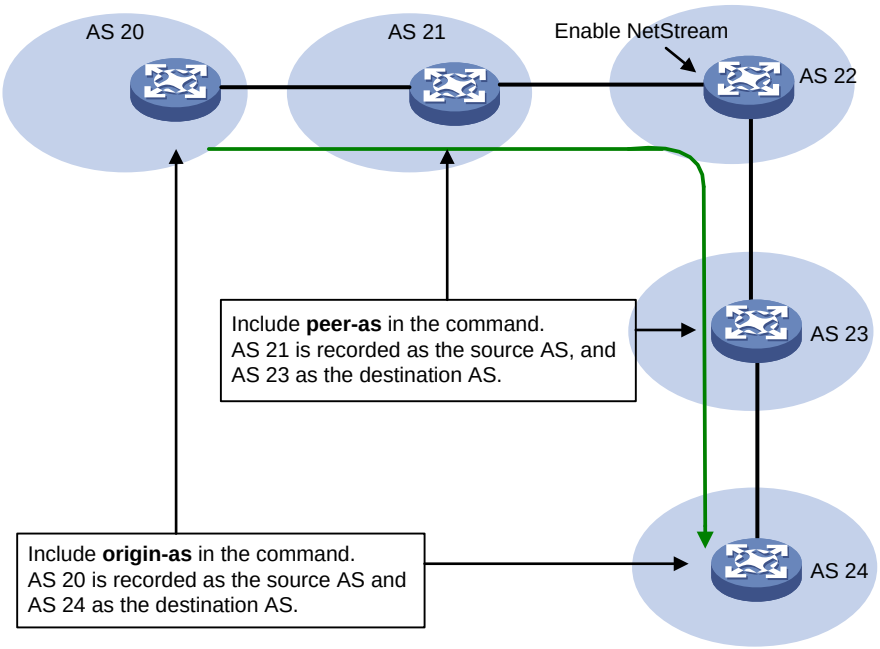
- Whether or not to export the BGP next hop information.
  Only version 9 and version 10 formats support exporting the BGP next hop information.
- How to export the autonomous system (AS) information: **origin-as** or **peer-as**.
  - **origin-as**—Records the original AS numbers for the flow source and destination.

- **peer-as**—Records the peer AS numbers for the flow source and destination.

For example, as shown in Figure 106, a flow starts at AS 20, passes AS 21 through AS 23, and then reaches AS 24. NetStream is enabled on the device in AS 22.

- Specify the **origin-as** keyword to export AS 20 as the source AS and AS 24 as the destination AS.
- Specify the **peer-as** keyword to export AS 21 as the source AS and AS 23 as the destination AS.

Figure 106 Recorded AS information varies by different keyword configurations



Procedure

1. Enter system view.

   system-view

2. Configure the NetStream data export format, and configure the AS and BGP next hop export attributes. Choose one option as needed:

   - Set NetStream data export format to version 5 and configure the AS export attribute.

     ip netstream export version 5 { **origin-as** | **peer-as** }

   - Set NetStream data export format to version 9 or version 10 and configure the AS and BGP export attributes.

     ip netstream export version { **9** | **10** } { **origin-as** | **peer-as** } [**bgp-nexthop**]

   By default:

   - NetStream data export uses the version 9 format.
   - The peer AS numbers for the flow source and destination are exported.
   - The BGP next hop information is not exported.

Configuring the refresh rate for NetStream version 9 or version 10 template

About this task

Version 9 and version 10 are template-based and support user-defined formats. A NetStream device must send the template to NetStream servers regularly to update the template on the servers.

For a NetStream server to use the correct version 9 or version 10 template, configure the time-based or packet count-based refresh rate. If both settings are configured, the template is sent when either of the conditions is met.

Procedure

1. Enter system view.

   system-view

2. Configure the refresh rate for the NetStream version 9 or version 10 template.

   ip netstream export template refresh-rate { **packet** *packets* | **time** *minutes* }

   By default, the packet count-based refresh rate is 20 packets, and the time-based refresh interval is 30 minutes.

Configuring VXLAN-aware NetStream

About this task

A VXLAN flow is identified by the same destination UDP port number. VXALN-aware NetStream collects statistics on the VNI information in the VXLAN packets.

Restrictions and guidelines

NetStream supports collecting statistics about only inbound VXLAN packets on VXLAN tunnel interfaces.

Procedure

1. Enter system view.

   system-view

2. Collect statistics on VXLAN packets.

   ip netstream vxlan udp-port *port-number*

   By default, statistics about VXLAN packets are not collected.

Configuring NetStream flow aging

Configuring periodical flow aging

1. Enter system view.

   system-view

2. Set the aging timer for active flows.

   ip netstream timeout active *minutes*

   By default, the aging timer for active flows is 5 minutes.

3. Set the aging timer for inactive flows.

ip netstream timeout inactive *seconds*

By default, the aging timer for inactive flows is 300 seconds.

Configuring forced flow aging

1. Enter system view.
   system-view
2. Set the upper limit for cached entries.
   ip netstream max-entry *max-entries*
   By default, a maximum of 1048576 NetStream entries can be cached.
3. Return to user view.
   quit
4. Clear the cache, including the cached NetStream entries and the related statistics.
   reset ip netstream statistics

Configuring the NetStream data export

Configuring the NetStream traditional data export

1. Enter system view.
   system-view
2. Specify a destination host for NetStream traditional data export.
   ip netstream export host *ip-address udp-port* [**vpn-instance** *vpn-instance-name*]
   By default, no destination host is specified.
3. (Optional.) Specify the source interface for NetStream data packets sent to NetStream servers.
   ip netstream export source interface *interface-type interface-number*
   By default, NetStream data packets take the IP address of their output interface (interface that is connected to the NetStream device) as the source IP address.
   As a best practice, connect the management Ethernet interface to a NetStream server, and configure the interface as the source interface.
4. (Optional.) Limit the data export rate.
   ip netstream export rate *rate*
   By default, the data export rate is not limited.

Configuring the NetStream aggregation data export

About this task

NetStream aggregation can be implemented by software or hardware. Unless otherwise noted, NetStream aggregation refers to software NetStream aggregation.

NetStream hardware aggregation uses hardware to directly merge the flow statistics according to the aggregation mode criteria, and stores the data in the cache. The aging of NetStream hardware aggregation entries is the same as the aging of NetStream traditional data entries. When a hardware aggregation entry is aged out, the data is exported.

NetStream hardware aggregation reduces the resource consumption by NetStream aggregation.

Restrictions and guidelines

NetStream hardware aggregation does not take effect in the following situations:

- The destination host is configured for NetStream traditional data export.
- The configured aggregation mode is not supported by NetStream hardware aggregation.

Configurations in NetStream aggregation mode view apply only to the NetStream aggregation data export, and those in system view apply to the NetStream traditional data export. If configurations in NetStream aggregation mode view are not provided, the configurations in system view apply to the NetStream aggregation data export.

If the version 5 format is configured to export NetStream data, NetStream aggregation data export uses the version 8 format.

Procedure

1. Enter system view.

   system-view

2. Enable NetStream hardware aggregation.

   ip netstream aggregation advanced

   By default, NetStream hardware aggregation is disabled.

3. Specify a NetStream aggregation mode and enter its view.

   ip netstream aggregation { **destination-prefix** | **prefix** | **prefix-port** | **protocol-port** | **source-prefix** | **tos-destination-prefix** | **tos-prefix** | **tos-protocol-port** | **tos-source-prefix** }

   By default, no NetStream aggregation mode is configured.

4. Enable the NetStream aggregation mode.

   enable

   By default, all NetStream aggregation modes are disabled.

5. Specify a destination host for NetStream aggregation data export.

   ip netstream export host *ip-address udp-port* [**vpn-instance** *vpn-instance-name*]

   By default, no destination host is specified.

   If you expect only NetStream aggregation data, specify the destination host only in the related NetStream aggregation mode view.

6. (Optional.) Specify the source interface for NetStream data packets sent to NetStream servers.

   ip netstream export source interface *interface-type interface-number*

   By default, no source interface is specified for NetStream data packets. The packets take the IP address of the output interface as the source IP address.

   Source interfaces in different NetStream aggregation mode views can be different.

   If no source interface is configured in NetStream aggregation mode view, the source interface configured in system view applies.

Display and maintenance commands for NetStream

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

Task	Command
Display NetStream entry information.	**display ip netstream cache** [**verbose**] [**type** { **ip** | **ipl2** | **l2** }] [**destination** *destination-ip* | **interface** *interface-type interface-number* | **source** *source-ip*] * [**slot** *slot-number*]
Display information about the NetStream data export.	**display ip netstream export**
Display NetStream template information.	**display ip netstream template** [**slot** *slot-number*]
Age out and export all NetStream data, and clear the cache.	**reset ip netstream statistics**

NetStream configuration examples

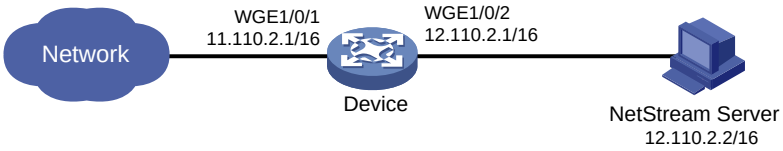
Example: Configuring NetStream traditional data export

Network configuration

As shown in Figure 107, configure NetStream on the device to collect statistics on packets passing through the device.

- Enable NetStream for incoming and outgoing traffic on Twenty-FiveGigE 1/0/1.
- Configure the device to export NetStream traditional data to UDP port 5000 of the NetStream server.

Figure 107 Network diagram



Procedure

Assign an IP address to each interface, as shown in Figure 107. (Details not shown.)

Enable NetStream for incoming and outgoing traffic on Twenty-FiveGigE 1/0/1.

```
<Device> system-view
[Device] interface twenty-fivegige 1/0/1
[Device-Twenty-FiveGigE1/0/1] ip netstream inbound
[Device-Twenty-FiveGigE1/0/1] ip netstream outbound
[Device-Twenty-FiveGigE1/0/1] quit
```

Specify 12.110.2.2 as the IP address of the destination host and UDP port 5000 as the export destination port number.

```
[Device] ip netstream export host 12.110.2.2 5000
```

Verifying the configuration

Display NetStream entry information.

```
[Device] display ip netstream cache
IP NetStream cache information:
  Active flow timeout             : 5 min
```

```
  Inactive flow timeout           : 300 sec
  Inactive flow timeout           : 30 sec
  Max number of entries           : 1024
  IP active flow entries          : 2
  MPLS active flow entries        : 0
  L2 active flow entries          : 0
  IPL2 active flow entries        : 0
  IP flow entries counted         : 0
  MPLS flow entries counted       : 0
  L2 flow entries counted         : 0
  IPL2 flow entries counted       : 0
  Last statistics resetting time  : Never

IP packet size distribution (11 packets in total):

 1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
 .000 .000 .909 .000 .000 .090 .000 .000 .000 .000 .000 .000 .000 .000 .000

  512  544  576 1024 1536 2048 2560 3072 3584 4096 4608 >4608
 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

 Protocol          Total  Packets    Flows  Packets Active(sec) Idle(sec)
                   Flows  /sec       /sec   /flow   /flow       /flow
-------------------------------------------------------------------------------

Type DstIP(Port)            SrcIP(Port)            Pro ToS If(Direct)  Pkts
     DstMAC(VLAN)           SrcMAC(VLAN)
     TopLblType(IP/MASK)    Lbl-Exp-S-List
-------------------------------------------------------------------------------
IP   10.1.1.1 (21)          100.1.1.2(1024)        1   0   GE1/0/1(I)  5
IP   100.1.1.2 (1024)       10.1.1.1 (21)          1   0   GE1/0/1(O)  5
```

Display information about the NetStream data export.

```
[Device] display ip netstream cache
  IP active flow entries          : 2
  MPLS active flow entries        : 0
  L2 active flow entries          : 0
  IPL2 active flow entries        : 0
  IP flow entries counted         : 0
  MPLS flow entries counted       : 0
  L2 flow entries counted         : 0
  IPL2 flow entries counted       : 0
  Last statistics resetting time  : Never

IP packet size distribution (11 packets in total):

 1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
 .000 .000 .909 .000 .000 .090 .000 .000 .000 .000 .000 .000 .000 .000 .000
```

```
  512  544  576 1024 1536 2048 2560 3072 3584 4096 4608 >4608
 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

 Protocol          Total  Packets    Flows  Packets Active(sec) Idle(sec)
                   Flows  /sec       /sec   /flow   /flow       /flow
-------------------------------------------------------------------------------

Type DstIP(Port)            SrcIP(Port)           Pro ToS If(Direct)  Pkts
     DstMAC(VLAN)           SrcMAC(VLAN)
     TopLblType(IP/MASK)    Lbl-Exp-S-List
-------------------------------------------------------------------------------
IP   10.1.1.1 (21)          100.1.1.2(1024)        1   0   WGE1/0/1(I) 5
IP   100.1.1.2 (1024)       10.1.1.1 (21)          1   0   WGE1/0/1(O) 5
```

Display information about the NetStream data export.

```
[Device] display ip netstream export
IP export information:
  Flow source interface                          : Not specified
  Flow destination VPN instance                  : Not specified
  Flow destination IP address (UDP)              : 12.110.2.2 (5000)
  Version 5 exported flow number                 : 0
  Version 5 exported UDP datagram number (failed) : 0 (0)
  Version 9 exported flow number                 : 10
  Version 9 exported UDP datagram number (failed) : 10 (0)
```

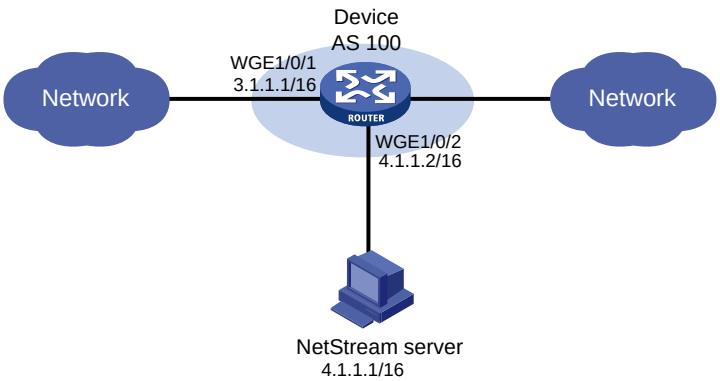
Example: Configuring NetStream aggregation data export

Network configuration

As shown in Figure 108, all routers in the network are running EBGP. Configure NetStream on the device to meet the following requirements:

- Use version 5 format to export NetStream traditional data to port 5000 of the NetStream server.
- Perform NetStream aggregation in the modes of protocol-port, source-prefix, destination-prefix, and prefix.
- Export the aggregation data of different modes to 4.1.1.1, with UDP ports 3000, 4000, 6000, and 7000.

Figure 108 Network diagram



Procedure

Assign an IP address to each interface, as shown in Figure 108. (Details not shown.)

Specify version 5 format to export NetStream traditional data and record the original AS numbers for the flow source and destination.

```
<Device> system-view
[Device] ip netstream export version 5 origin-as
```

Enable NetStream for incoming and outgoing traffic on Twenty-FiveGigE 1/0/1.

```
[Device] interface twenty-fivegige 1/0/1
[Device-Twenty-FiveGigE1/0/1] ip netstream inbound
[Device-Twenty-FiveGigE1/0/1] ip netstream outbound
[Device-Twenty-FiveGigE1/0/1] quit
```

Specify 4.1.1.1 as the IP address of the destination host and UDP port 5000 as the export destination port number.

```
[Device] ip netstream export host 4.1.1.1 5000
```

Set the aggregation mode to protocol-port, and specify the destination host for the aggregation data export.

```
[Device] ip netstream aggregation protocol-port
[Device-ns-aggregation-protport] enable
[Device-ns-aggregation-protport] ip netstream export host 4.1.1.1 3000
[Device-ns-aggregation-protport] quit
```

Set the aggregation mode to source-prefix, and specify the destination host for the aggregation data export.

```
[Device] ip netstream aggregation source-prefix
[Device-ns-aggregation-srcpre] enable
[Device-ns-aggregation-srcpre] ip netstream export host 4.1.1.1 4000
[Device-ns-aggregation-srcpre] quit
```

Set the aggregation mode to destination-prefix, and specify the destination host for the aggregation data export.

```
[Device] ip netstream aggregation destination-prefix
[Device-ns-aggregation-dstpre] enable
[Device-ns-aggregation-dstpre] ip netstream export host 4.1.1.1 6000
[Device-ns-aggregation-dstpre] quit
```

Set the aggregation mode to prefix, and specify the destination host for the aggregation data export.

```
[Device] ip netstream aggregation prefix
[Device-ns-aggregation-prefix] enable
[Device-ns-aggregation-prefix] ip netstream export host 4.1.1.1 7000
[Device-ns-aggregation-prefix] quit
```

Verifying the configuration

Display information about the NetStream data export.

```
[Device] display ip netstream export
protocol-port aggregation export information:
  Flow source interface                          : Not specified
  Flow destination VPN instance                  : Not specified
  Flow destination IP address (UDP)              : 4.1.1.1 (3000)
  Version 8 exported flow number                 : 2
  Version 8 exported UDP datagram number (failed) : 2 (0)
```

```
  Version 9 exported flow number                  : 0
  Version 9 exported UDP datagram number (failed) : 0 (0)

source-prefix aggregation export information:
  Flow source interface                           : Not specified
  Flow destination VPN instance                   : Not specified
  Flow destination IP address (UDP)               : 4.1.1.1 (4000)
  Version 8 exported flow number                  : 2
  Version 8 exported UDP datagram number (failed) : 2 (0)
  Version 9 exported flow number                  : 0
  Version 9 exported UDP datagram number (failed) : 0 (0)

destination-prefix aggregation export information:
  Flow source interface                           : Not specified
  Flow destination VPN instance                   : Not specified
  Flow destination IP address (UDP)               : 4.1.1.1 (6000)
  Version 8 exported flow number                  : 2
  Version 8 exported UDP datagram number (failed) : 2 (0)
  Version 9 exported flow number                  : 0
  Version 9 exported UDP datagram number (failed) : 0 (0)

prefix aggregation export information:
  Flow source interface                           : Not specified
  Flow destination VPN instance                   : Not specified
  Flow destination IP address (UDP)               : 4.1.1.1 (7000)
  Version 8 exported flow number                  : 2
  Version 8 exported UDP datagram number (failed) : 2 (0)
  Version 9 exported flow number                  : 0
  Version 9 exported UDP datagram number (failed) : 0 (0)

IP export information:
  Flow source interface                           : Not specified
  Flow destination VPN instance                   : Not specified
  Flow destination IP address (UDP)               : 4.1.1.1 (5000)
  Version 5 exported flow number                  : 10
  Version 5 exported UDP datagram number (failed) : 10 (0)
  Version 9 exported flow number                  : 0
  Version 9 exported UDP datagram number (failed) : 0 (0)
```

Configuring IPv6 NetStream

About IPv6 NetStream

IPv6 NetStream is an accounting technology that provides statistics on a per-flow basis. An IPv6 flow is defined by the following 8-tuple elements:

- Destination IPv6 address.
- Source IPv6 address.
- Destination port number.
- Source port number.
- Protocol number.
- Traffic class.
- Flow label.
- Input or output interface.

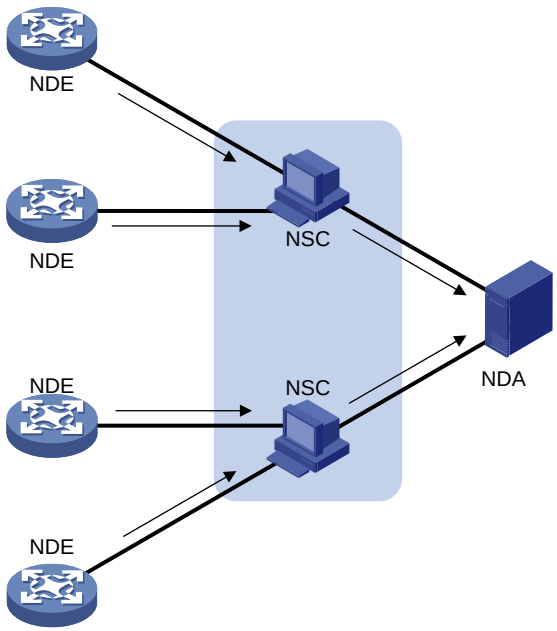
IPv6 NetStream architecture

A typical IPv6 NetStream system includes the following elements:

- **NetStream data exporter**—A device configured with IPv6 NetStream. The NDE provides the following functions:
  - Classifies traffic flows by using the 8-tuple elements.
  - Collects data from the classified flows.
  - Aggregates and exports the data to the NSC.
- **NetStream collector**—A program running in a Unix or Windows operating system. The NSC parses the packets received from the NDEs, and saves the data to its database.
- **NetStream data analyzer**—A network traffic analyzing tool. Based on the data in NSC, the NDA generates reports for traffic billing, network planning, and attack detection and monitoring. The NDA can collect data from multiple NSCs. Typically, the NDA features a Web-based system for easy operation.

NSC and NDA are typically integrated into a NetStream server.

Figure 109 IPv6 NetStream system



IPv6 NetStream flow aging

IPv6 NetStream uses flow aging to enable the NDE to export IPv6 NetStream data to NetStream servers. IPv6 NetStream creates an IPv6 NetStream entry for each flow for storing the flow statistics in the cache.

When a flow is aged out, the NDE performs the following operations:

- Exports the summarized data to NetStream servers in a specific format.
- Clears IPv6 NetStream entry information in the cache.

IPv6 NetStream supports the following flow aging methods:

- Periodical aging.
- Forced aging.

Periodical aging

Periodical aging uses the following methods:

- **Inactive flow aging**—A flow is inactive if no packet arrives for the IPv6 NetStream entry within the inactive flow aging timer. When the timer expires, the following events occur:
  - The inactive flow entry is aged out.
  - The statistics of the flow are sent to NetStream servers and are cleared in the cache. The statistics can no longer be displayed by using the **`display ipv6 netstream cache`** command.

  This method ensures that inactive flow entries are cleared from the cache in a timely manner so new entries can be cached.
- **Active flow aging**—A flow is active if packets arrive for the IPv6 NetStream entry within the active flow aging timer. When the timer expires, the statistics of the active flow are exported to NetStream servers. The device continues to collect its statistics, which can be displayed by using the **`display ipv6 netstream cache`** command.

  The active flow aging method periodically exports the statistics of active flows to NetStream servers.

Forced aging

To implement forced aging, use one of the following methods:

- Clear the IPv6 NetStream cache immediately. All entries in the cache are aged out and exported to NetStream servers.
- Specify the upper limit for cached entries. When the limit is reached, new entries will overwrite the oldest entries in the cache.

IPv6 NetStream data export

Traditional data export

IPv6 NetStream collects the statistics of each flow and exports the statistics to NetStream servers.

This method consumes a lot of bandwidth and CPU usage, and requires a large cache size. In addition, you do not need all of the data in most cases.

Aggregation data export

An IPv6 NetStream aggregation mode merges the flow statistics according to the aggregation criteria of the aggregation mode, and it sends the summarized data to NetStream servers. The IPv6 NetStream aggregation data export uses less bandwidth than the traditional data export.

Table 37 lists the available IPv6 NetStream aggregation modes. In each mode, the system merges multiple flows with the same values for all aggregation criteria into one aggregate flow. The system records the statistics for the aggregate flow. These aggregation modes work independently and can take effect concurrently.

Table 37 IPv6 NetStream aggregation modes

Aggregation mode	Aggregation criteria
Protocol-port aggregation	• Protocol number • Source port • Destination port
Source-prefix aggregation	• Source AS number • Source mask • Source prefix (source network address) • Input interface index
Destination-prefix aggregation	• Destination AS number • Destination mask • Destination prefix (destination network address) • Output interface index
Source-prefix and destination-prefix aggregation	• Source AS number • Source mask • Source prefix (source network address) • Input interface index • Destination AS number • Destination mask • Destination prefix (destination network address) • Output interface index

IPv6 NetStream data export format

IPv6 NetStream exports data in the version 9 or version 10 format.

Both formats are template-based and support exporting the IPv6 NetStream aggregation data and collecting statistics about BGP next hop and MPLS packets.

The version 10 export format is compliant with the IPFIX standard.

IPv6 NetStream filtering

IPv6 NetStream filtering uses an ACL to identify packets. Whether IPv6 NetStream collects data for identified packets depends on the action in the matching rule.

- IPv6 NetStream collects data for packets that match permit rules in the ACL.
- IPv6 NetStream does not collect data for packets that match deny rules in the ACL.

For more information about ACLs, see *ACL and QoS Configuration Guide*.

IPv6 NetStream sampling

IPv6 NetStream sampling collects statistics on fewer packets and is useful when the network has a large amount of traffic. IPv6 NetStream on sampled traffic lessens the impact on the device's performance. For more information about sampling, see "Configuring samplers."

Protocols and standards

RFC 5101, *Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information*

Restrictions: Hardware compatibility with IPv6 NetStream

IPv6 NetStream is supported only on the HPE FlexFabric 5945 2-slot switch (JQ075A), HPE FlexFabric 5945 4-slot switch (JQ076A), and HPE FlexFabric 5945 32QSFP28 switch (JQ077A).

The service interfaces near power module side on the rear panel of the switch are used for internal loopback of IPv6 NetStream traffic. After IPv6 NetStream is enabled on an interface on the front panel, you cannot use service interfaces on the rear panel and the settings on these service interfaces become invalid. Before enabling IPv6 NetStream on an interface, clear the configurations on the service interfaces.

Restrictions and guidelines: IPv6 NetStream configuration

You can enable sampling for only one of the following features on the device:

- IPv6 NetStream.
- Mirroring.
- sFlow.
- INT.
- Telemetry stream.
- MOD.

For more information about mirroring and sFlow, see *Network Management and Monitoring Configuration Guide.* For more information about INT, telemetry stream, and MOD, see *Telemetry Configuration Guide.*

IPv6 NetStream tasks at a glance

To configure IPv6 NetStream, perform the following tasks:

1. Enabling IPv6 NetStream
2. (Optional.) Configuring IPv6 NetStream filtering
3. (Optional.) Configuring IPv6 NetStream sampling
4. (Optional.) Configuring the IPv6 NetStream data export format
5. (Optional.) Configuring the refresh rate for IPv6 NetStream version 9 or version 10 template
6. (Optional.) Configuring IPv6 NetStream flow aging
   - Configuring periodical flow aging
   - Configuring forced flow aging
7. Configuring the IPv6 NetStream data export
   a. Configuring the IPv6 NetStream traditional data export
   b. (Optional.) Configuring the IPv6 NetStream aggregation data export

Enabling IPv6 NetStream

Restrictions and guidelines

This feature is available on only Layer 2 Ethernet interfaces and Layer 3 Ethernet interfaces. It is not available on Layer 3 Ethernet subinterfaces or other types of interfaces.

Procedure

1. Enter system view.
   `system-view`
2. Enter interface view.
   `interface` *`interface-type interface-number`*
3. Enable IPv6 NetStream on the interface.
   `ipv6 netstream { inbound | outbound }`
   By default, IPv6 NetStream is disabled on an interface.

Configuring IPv6 NetStream filtering

About this task

IPv6 NetStream filtering uses an ACL to identify packets.

- To enable IPv6 NetStream to collect statistics for specific flows, use the ACL permit statements to identify these flows
- To disable IPv6 NetStream from collecting statistics for specific flows, use the ACL deny statements to identify these flows.

Restrictions and guidelines

If IPv6 NetStream filtering and sampling are both configured, IPv6 packets are filtered first, and then the permitted packets are sampled.

The IPv6 NetStream filtering feature does not take effect on MPLS packets.

If you use NetStream filtering on the interface where IPv4 and IPv6 NetStream are enabled in the same direction, make sure NetStream filtering is enabled for both IPv4 and IPv6 in this direction. For

more information about IPv4 NetStream, see *Network Management and Monitoring Configuration Guide*.

Procedure

1. Enter system view.
   system-view
2. Enter interface view.
   interface *interface-type interface-number*
3. Configure IPv6 NetStream filtering on the interface.
   ipv6 netstream inbound filter acl *acl-number*
   By default, IPv6 NetStream filtering is disabled. IPv6 NetStream collects statistics of all IPv6 packets passing through the interface.

Configuring IPv6 NetStream sampling

Restrictions and guidelines

By default, IPv6 NetStream collects all data of target flows. If the flow traffic is heavy, IPv6 NetStream is resource-consuming and can cause high CPU usage, which impacts the device forwarding performance. IPv6 NetStream sampling is helpful to decrease the IPv6 NetStream traffic volume. If the collected statistics can basically reflect the network status, you can enable this feature and set a proper sampling rate. The higher the sampling rate, the less impact on device performance.

If IPv6 NetStream sampling and filtering are both configured, IPv6 packets are filtered first, and then the permitted packets are sampled.

Procedure

1. Enter system view.
   system-view
2. Create a sampler.
   sampler *sampler-name* **mode random packet-interval n-power** *rate*
   For more information about samplers, see "Configuring samplers."
3. Enter interface view.
   interface *interface-type interface-number*
4. Configure IPv6 NetStream sampling.
   ip netstream { **inbound** | **outbound** } **sampler** *sampler-name*
   This command enables both IPv4 NetStream sampling and IPv6 NetStream sampling.
   By default, IPv6 NetStream sampling is disabled.
   For more information about the **ip netstream sampler** command, see "Configuring NetStream."

Configuring the IPv6 NetStream data export format

About this task

When you configure the IPv6 NetStream data export format, you can also specify the following settings:

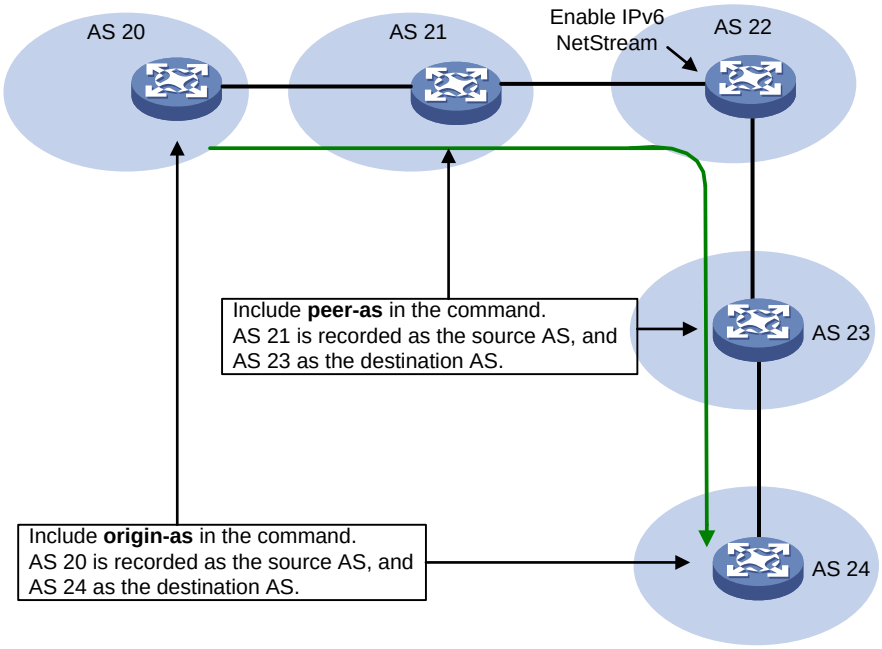
- Whether or not to export the BGP next hop information.

- How to export the autonomous system (AS) information: **origin-as** or **peer-as**.
  - **origin-as**—Records the original AS numbers for the flow source and destination.
  - **peer-as**—Records the peer AS numbers for the flow source and destination.

For example, as shown in Figure 110, a flow starts at AS 20, passes AS 21 through AS 23, and then reaches AS 24. IPv6 NetStream is enabled on the device in AS 22.

- Specify the **origin-as** keyword to export AS 20 as the source AS and AS 24 as the destination AS.
- Specify the **peer-as** keyword to export AS 21 as the source AS and AS 23 as the destination AS.

Figure 110 Recorded AS information varies by different keyword configurations



Procedure

1. Enter system view.

   system-view

2. Configure the IPv6 NetStream data export format, and configure the AS and BGP next hop export attributes.
   - Configure the version 9 format.

     ipv6 netstream export version 9 { **origin-as** | **peer-as** } [**bgp-nexthop**]
   - Configure the version 10 format.

     ipv6 netstream export version 10 [**origin-as** | **peer-as**] [**bgp-nexthop**]

   By default:
   - The version 9 format is used to export IPv6 NetStream data.
   - The peer AS numbers for the flow source and destination are exported.
   - The BGP next hop information is not exported.

Configuring the refresh rate for IPv6 NetStream version 9 or version 10 template

About this task

Version 9 and version 10 are template-based and support user-defined formats. An IPv6 NetStream device must send the updated template to NetStream servers regularly, because the servers do not permanently save templates.

For a NetStream server to use the correct version 9 or version 10 template, configure the time-based or packet count-based refresh rate. If both settings are configured, the template is sent when either of the conditions is met.

Procedure

1. Enter system view.

   system-view
2. Configure the refresh rate for the IPv6 NetStream version 9 or version 10 template.

   ipv6 netstream export template refresh-rate { **packet** *packets* | **time** *minutes* }

   By default, the packet count-based refresh rate is 20 packets, and the time-based refresh interval is 30 minutes.

Configuring IPv6 NetStream flow aging

Configuring periodical flow aging

1. Enter system view.

   system-view
2. Set the aging timer for active flows.

   ipv6 netstream timeout active *minutes*

   By default, the aging timer for active flows is 5 minutes.
3. Set the aging timer for inactive flows.

   ipv6 netstream timeout inactive *seconds*

   By default, the aging timer for inactive flows is 300 seconds.

Configuring forced flow aging

1. Enter system view.

   system-view
2. Set the upper limit for cached entries.

   ipv6 netstream max-entry *max-entries*

   By default, a maximum of 1048576 IPv6 NetStream entries can be cached.
3. Return to user view.

   quit
4. Clear the cache, including the cached IPv6 NetStream entries and the related statistics.

   reset ipv6 netstream statistics

Configuring the IPv6 NetStream data export

Configuring the IPv6 NetStream traditional data export

1. Enter system view.

   system-view

2. Specify a destination host for IPv6 NetStream traditional data export.

   ipv6 netstream export host { *ipv4-address* | *ipv6-address* } *udp-port* [**vpn-instance** *vpn-instance-name*]

   By default, no destination host is specified.

3. (Optional.) Specify the source interface for IPv6 NetStream data packets sent to the NetStream servers.

   ipv6 netstream export source interface *interface-type interface-number*

   By default, no source interface is specified for IPv6 NetStream data packets. The packets take the IPv6 address of the output interface (interface that is connected to the NetStream server) as the source IPv6 address.

   As a best practice, connect the management Ethernet interface to a NetStream server, and configure the interface as the source interface.

4. (Optional.) Limit the IPv6 NetStream data export rate.

   ipv6 netstream export rate *rate*

   By default, the data export rate is not limited.

Configuring the IPv6 NetStream aggregation data export

About this task

The IPv6 NetStream aggregation can be implemented by software or hardware. Unless otherwise noted, NetStream aggregation refers to software NetStream aggregation.

IPv6 NetStream hardware aggregation uses hardware to directly merge the flow statistics according to the aggregation mode criteria, and stores the data in the cache. The aging of IPv6 NetStream hardware aggregation entries is the same as the aging of IPv6 NetStream traditional data entries. When a hardware aggregation entry is aged out, the data is exported.

IPv6 NetStream hardware aggregation reduces resource consumption.

Restrictions and guidelines

The IPv6 NetStream hardware aggregation does not take effect in the following situations:

- The destination host is configured for NetStream traditional data export.
- The configured aggregation mode is not supported by IPv6 NetStream hardware aggregation.

Configurations in IPv6 NetStream aggregation mode view apply only to the IPv6 NetStream aggregation data export. Configurations in system view apply to the IPv6 NetStream traditional data export. When no configuration in IPv6 NetStream aggregation mode view is provided, the configurations in system view apply to the IPv6 NetStream aggregation data export.

Procedure

1. Enter system view.

   system-view

2. Enable IPv6 NetStream hardware aggregation.

   ipv6 netstream aggregation advanced

By default, IPv6 NetStream hardware aggregation is disabled.

3. Specify an IPv6 NetStream aggregation mode and enter its view.

   ipv6 netstream aggregation { **destination-prefix** | **prefix** | **protocol-port** | **source-prefix** }

   By default, no IPv6 NetStream aggregation mode is specified.

4. Enable the IPv6 NetStream aggregation mode.

   enable

   By default, the IPv6 NetStream aggregation is disabled.

5. Specify a destination host for IPv6 NetStream aggregation data export.

   ipv6 netstream export host { *ipv4-address* | *ipv6-address* } *udp-port* [**vpn-instance** *vpn-instance-name*]

   By default, no destination host is specified.

   If you expect only IPv6 NetStream aggregation data, specify the destination host only in the related IPv6 NetStream aggregation mode view.

6. (Optional.) Specify the source interface for IPv6 NetStream data packets sent to the NetStream servers.

   ipv6 netstream export source interface *interface-type interface-number*

   By default, no source interface is specified for IPv6 NetStream data packets. The packets take the IPv6 address of the output interface as the source IPv6 address.

   You can configure different source interfaces in different IPv6 NetStream aggregation mode views.

   If no source interface is configured in IPv6 NetStream aggregation mode view, the source interface configured in system view applies.

Display and maintenance commands for IPv6 NetStream

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

Task	Command
Display IPv6 NetStream entry information.	**display ipv6 netstream cache** [**verbose**] [**type** { **ip** | **ipl2** | **l2** }] [**destination** *destination-ipv6* | **interface** *interface-type interface-number* | **source** *source-ipv6*] * [**slot** *slot-number*]
Display information about the IPv6 NetStream data export.	**display ipv6 netstream export**
Display IPv6 NetStream template information.	**display ipv6 netstream template** [**slot** *slot-number*] **display ipv6 netstream template**
Age out, export all IPv6 NetStream data, and clear the cache.	**reset ipv6 netstream statistics**

IPv6 NetStream configuration examples

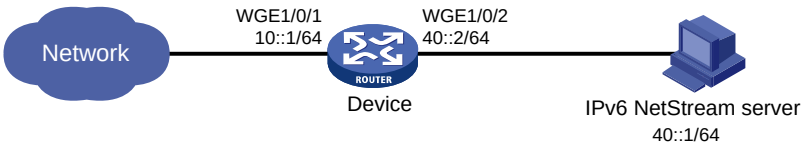
Example: Configuring IPv6 NetStream traditional data export

Network configuration

As shown in Figure 111, configure IPv6 NetStream on the device to collect statistics on packets passing through the device.

- Enable IPv6 NetStream for incoming and outgoing traffic on Twenty-FiveGigE 1/0/1.
- Configure the device to export the IPv6 NetStream traditional data to UDP port 5000 of the NetStream server.

Figure 111 Network diagram



Procedure

Assign an IP address to each interface, as shown in Figure 111. (Details not shown.)

Enable IPv6 NetStream for incoming and outgoing traffic on Twenty-FiveGigE 1/0/1.

```
<Device> system-view
[Device] interface twenty-fivegige 1/0/1
[Device-Twenty-FiveGigE1/0/1] ipv6 netstream inbound
[Device-Twenty-FiveGigE1/0/1] ipv6 netstream outbound
[Device-Twenty-FiveGigE1/0/1] quit
```

Specify 40::1 as the IP address of the destination host and UDP port 5000 as the export destination port number.

```
[Device] ipv6 netstream export host 40::1 5000
```

Verifying the configuration

Display information about IPv6 NetStream entries.

```
<Device> display ipv6 netstream cache
IPv6 NetStream cache information:
  Active flow timeout             : 5 min
  Inactive flow timeout           : 300 sec
  Max number of entries           : 1000
  IPv6 active flow entries        : 2
  MPLS active flow entries        : 0
  IPL2 active flow entries        : 0
  IPv6 flow entries counted       : 10
  MPLS flow entries counted       : 0
  IPL2 flow entries counted       : 0
  Last statistics resetting time  : 01/01/2000 at 00:01:02

IPv6 packet size distribution (1103746 packets in total):
1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
.249 .694 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000
```

```
512  544  576 1024 1536 2048 2560 3072 3584 4096 4608 >4608
.000 .000 .027 .000 .027 .000 .000 .000 .000 .000 .000 .000

Protocol          Total Packets    Flows  Packets Active(sec) Idle(sec)
                  Flows /sec       /sec   /flow   /flow       /flow
------------------------------------------------------------------------------
TCP-Telnet      2656855     372        4       86        49         27
TCP-FTP         5900082      86        9        9        11         33
TCP-FTPD        3200453    1006        5      193        45         33
TCP-WWW       546778274   11170      887       12         8         32
TCP-other      49148540    3752       79       47        30         32
UDP-DNS       117240379     570      190        3         7         34
UDP-other      45502422    2272       73       30         8         37
ICMP           14837957     125       24        5        12         34
IP-other          77406       5        0       47        52         27

 Type DstIP(Port)        SrcIP(Port)      Pro TC  FlowLbl If(Direct)   Pkts
      DstMAC(VLAN)       SrcMAC(VLAN)
      TopLblType(IP/MASK)Lbl-Exp-S-List
------------------------------------------------------------------------------
IP   2001::1(1024)      2002::1(21)        6   0   0x0     WGE1/0/1(I) 42996
IP   2002::1(21)        2001::1(1024)      6   0   0x0     WGE1/0/1(O) 42996
```

Display information about the IPv6 NetStream data export.

```
[Device] display ipv6 netstream export
IPv6 export information:
  Flow source interface                          : Not specified
  Flow destination VPN instance                  : Not specified
  Flow destination IP address (UDP)              : 40::1 (5000)
  Version 9 exported flow number                 : 10
  Version 9 exported UDP datagram number (failed) : 10 (0)
```

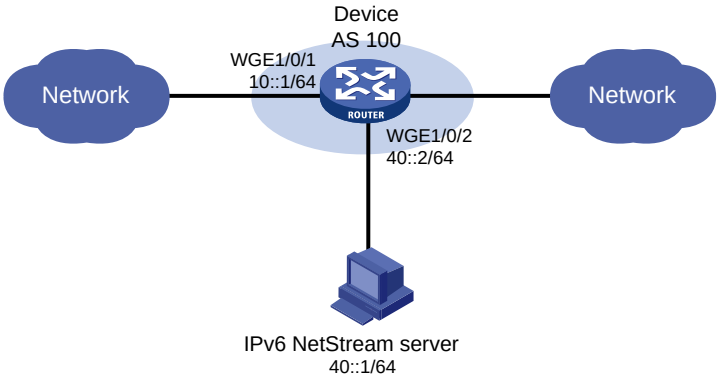
Example: Configuring IPv6 NetStream aggregation data export

Network configuration

As shown in Figure 112, all routers in the network are running IPv6 EBGP. Configure IPv6 NetStream on the device to meet the following requirements:

- Export the IPv6 NetStream traditional data to port 5000 of the NetStream server.
- Perform the IPv6 NetStream aggregation in the modes of protocol-port, source-prefix, destination-prefix, and prefix.
- Export the aggregation data of different modes to the UDP ports 3000, 4000, 6000, and 7000.

Figure 112 Network diagram



Procedure

Assign an IP address to each interface, as shown in Figure 112. (Details not shown.)

Enable IPv6 NetStream for incoming and outgoing traffic on Twenty-FiveGigE 1/0/1.

```
<Device> system-view
[Device] interface twenty-fivegige 1/0/1
[Device-Twenty-FiveGigE1/0/1] ipv6 netstream inbound
[Device-Twenty-FiveGigE1/0/1] ipv6 netstream outbound
[Device-Twenty-FiveGigE1/0/1] quit
```

Specify 40::1 as the IP address of the destination host and UDP port 5000 as the export destination port number.

```
[Device] ipv6 netstream export host 40::1 5000
```

Set the aggregation mode to protocol-port, and specify the destination host for the aggregation data export.

```
[Device] ipv6 netstream aggregation protocol-port
[Device-ns6-aggregation-protport] enable
[Device-ns6-aggregation-protport] ipv6 netstream export host 40::1 3000
[Device-ns6-aggregation-protport] quit
```

Set the aggregation mode to source-prefix, and specify the destination host for the aggregation data export.

```
[Device] ipv6 netstream aggregation source-prefix
[Device-ns6-aggregation-srcpre] enable
[Device-ns6-aggregation-srcpre] ipv6 netstream export host 40::1 4000
[Device-ns6-aggregation-srcpre] quit
```

Set the aggregation mode to destination-prefix, and specify the destination host for the aggregation data export.

```
[Device] ipv6 netstream aggregation destination-prefix
[Device-ns6-aggregation-dstpre] enable
[Device-ns6-aggregation-dstpre] ipv6 netstream export host 40::1 6000
[Device-ns6-aggregation-dstpre] quit
```

Set the aggregation mode to prefix, and specify the destination host for the aggregation data export.

```
[Device] ipv6 netstream aggregation prefix
[Device-ns6-aggregation-prefix] enable
[Device-ns6-aggregation-prefix] ipv6 netstream export host 40::1 7000
[Device-ns6-aggregation-prefix] quit
```

Verifying the configuration

Display information about the IPv6 NetStream data export.

```
[Device] display ipv6 netstream export
as aggregation export information:
  Flow source interface                          : Not specified
  Flow destination VPN instance                  : Not specified
  Flow destination IP address (UDP)              : 40::1 (2000)
  Version 9 exported flow number                 : 0
  Version 9 exported UDP datagram number (failed) : 0(0)

protocol-port aggregation export information:
  Flow source interface                          : Not specified
  Flow destination VPN instance                  : Not specified
  Flow destination IP address (UDP)              : 40::1 (3000)
  Version 9 exported flow number                 : 0
  Version 9 exported UDP datagram number (failed) : 0 (0)

source-prefix aggregation export information:
  Flow source interface                          : Not specified
  Flow destination VPN instance                  : Not specified
  Flow destination IP address (UDP)              : 40::1 (4000)
  Version 9 exported flow number                 : 0
  Version 9 exported UDP datagram number (failed) : 0 (0)

destination-prefix aggregation export information:
  Flow source interface                          : Not specified
  Flow destination VPN instance                  : Not specified
  Flow destination IP address (UDP)              : 40::1 (6000)
  Version 9 exported flow number                 : 0
  Version 9 exported UDP datagram number (failed) : 0 (0)

prefix aggregation export information:
  Flow source interface                          : Not specified
  Flow destination VPN instance                  : Not specified
  Flow destination IP address (UDP)              : 40::1 (7000)
  Version 9 exported flow number                 : 0
  Version 9 exported UDP datagram number (failed) : 0 (0)

IPv6 export information:
  Flow source interface                          : Not specified
  Flow destination VPN instance                  : Not specified
  Flow destination IP address (UDP)              : 40::1 (5000)
  Version 9 exported flow number                 : 0
  Version 9 exported UDP datagram number (failed) : 0 (0)
```

Configuring sFlow

About sFlow

sFlow is a traffic monitoring technology.

As shown in Figure 113, 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 actions:

- Encapsulates the sFlow packets in the UDP datagrams.
- Sends the UDP datagrams to the specified sFlow collector.

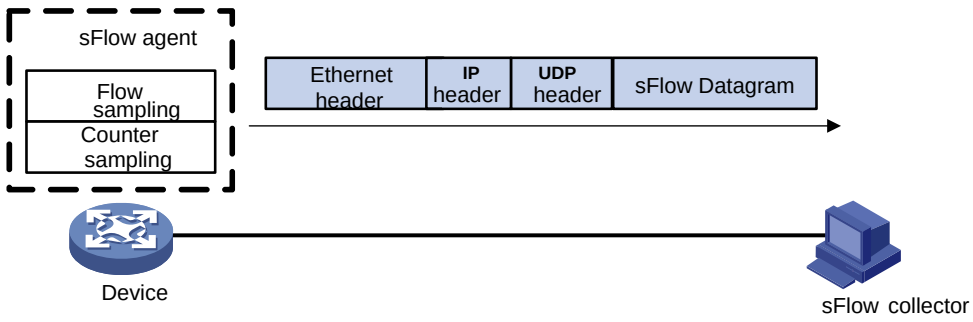
The sFlow collector analyzes the information and displays the results. One sFlow collector can monitor multiple sFlow agents.

sFlow provides the following sampling mechanisms:

- **Flow sampling**—Obtains packet information.
- **Counter sampling**—Obtains interface counter information.

sFlow can use flow sampling and counter sampling at the same time.

Figure 113 sFlow system



Protocols and standards

- RFC 3176, *InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks*
- sFlow.org, *sFlow Version 5*

Restrictions and guidelines: sFlow configuration

You can enable sampling for only one of the following features on the device:

- sFlow.
- Mirroring.
- NetStream.
- IPv6 NetStream.
- INT.

- Telemetry stream.
- MOD.

For more information about mirroring, NetStream, IPv6 NetStream, and sFlow, see *Network Management and Monitoring Configuration Guide.* For more information about INT, telemetry stream, and MOD, see *Telemetry Configuration Guide.*

Configuring basic sFlow information

Restrictions and guidelines

As a best practice, manually configure an IP address for the sFlow agent. The device periodically checks whether the sFlow agent has an IP address. If the sFlow agent does not have an IP address, the device automatically selects an IPv4 address for the sFlow agent but does not save the IPv4 address in the configuration file.

Only one IP address can be configured for the sFlow agent on the device, and a newly configured IP address overwrites the existing one.

Procedure

1. Enter system view.

   system-view
2. Configure an IP address for the sFlow agent.

   sflow agent { **ip** *ipv4-address* | **ipv6** *ipv6-address* }

   By default, no IP address is configured for the sFlow agent.
3. Configure the sFlow collector information.

   sflow collector *collector-id* [**vpn-instance** *vpn-instance-name*] { **ip** *ipv4-address* | **ipv6** *ipv6-address* } [**port** *port-number* | **datagram-size** *size* | **time-out** *seconds* | **description** *string*] *

   By default, no sFlow collector information is configured.
4. Specify the source IP address of sFlow packets.

   sflow source { **ip** *ipv4-address* | **ipv6** *ipv6-address* } *

   By default, the source IP address is determined by routing.

Configuring flow sampling

About this task

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.

Procedure

1. Enter system view.

   system-view
2. Enter Layer 2 Ethernet interface view or Layer 3 Ethernet interface view.

   interface *interface-type interface-number*
3. (Optional.) Set the flow sampling mode.

sflow sampling-mode random

By default, random sampling is used.

4. Enable flow sampling and specify the number of packets out of which flow sampling samples a packet on the interface.

   sflow sampling-rate *rate*

   By default, flow sampling is disabled.

   As a best practice, set the sampling interval to 2^n that is greater than or equal to 8192, for example, 32768.

5. (Optional.) Set the maximum number of bytes (starting from the packet header) that flow sampling can copy per packet.

   sflow flow max-header *length*

   The default setting is 128 bytes.

   As a best practice, use the default setting.

6. Specify the sFlow instance and sFlow collector for flow sampling.

   sflow flow [**instance** *instance-id*] **collector** *collector-id*

   By default, no sFlow instance or sFlow collector is specified for flow sampling.

Configuring counter sampling

About this task

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.

Procedure

1. Enter system view.

   system-view

2. Enter Layer 2 Ethernet interface view or Layer 3 Ethernet interface view.

   interface *interface-type interface-number*

3. Enable counter sampling and set the counter sampling interval.

   sflow counter interval *interval*

   By default, counter sampling is disabled.

4. Specify the sFlow instance and sFlow collector for counter sampling.

   sflow counter [**instance** *instance-id*] **collector** *collector-id*

   By default, no sFlow instance or sFlow collector is specified for counter sampling.

Display and maintenance commands for sFlow

Execute **display** commands in any view.

Task	Command
Display sFlow configuration.	**display sflow**

sFlow configuration examples

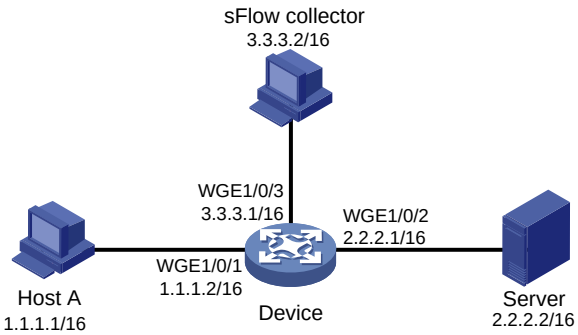
Example: Configuring sFlow

Network configuration

As shown in Figure 114, perform the following tasks:

- Configure flow sampling in random mode and counter sampling on Twenty-FiveGigE 1/0/1 of the device to monitor traffic on the port.
- Configure the device to send sampled information in sFlow packets through Twenty-FiveGigE 1/0/3 to the sFlow collector.

Figure 114 Network diagram



Procedure

1. Configure the IP addresses and subnet masks for interfaces, as shown in Figure 114. (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 Twenty-FiveGigE 1/0/1.

   ```
   [Device] interface twenty-fivegige 1/0/1
   [Device-Twenty-FiveGigE1/0/1] sflow counter interval 120
   ```

   # Specify sFlow collector 1 for counter sampling.

   ```
   [Device-Twenty-FiveGigE1/0/1] sflow counter collector 1
   ```

4. Configure flow sampling:

   # Enable flow sampling and set the flow sampling mode to random and sampling interval to 32768.

   ```
   [Device-Twenty-FiveGigE1/0/1] sflow sampling-mode random
   [Device-Twenty-FiveGigE1/0/1] sflow sampling-rate 32768
   ```

   # Specify sFlow collector 1 for flow sampling.

   ```
   [Device-Twenty-FiveGigE1/0/1] sflow flow collector 1
   ```

Verifying the configuration

Verify the following items:

- Twenty-FiveGigE 1/0/1 enabled with sFlow is active.
- The counter sampling interval is 120 seconds.
- The flow sampling interval is 4000 (one packet is sampled from every 4000 packets).

```
[Device-Twenty-FiveGigE1/0/1] display sflow
sFlow datagram version: 5
Global information:
Agent IP: 3.3.3.1(CLI)
Source address:
Collector information:
ID    IP              Port  Aging      Size VPN-instance Description
1     3.3.3.2         6343  N/A        1400              netserver
Port counter sampling information:
Interface   Instance   CID   Interval(s)
WGE1/0/1    1          1     120
Port flow sampling information:
Interface   Instance   FID   MaxHLen Rate       Mode      Status
WGE1/0/1    1          1     128     32768      Random    Active
```

Troubleshooting sFlow

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 sFlow collector is bound to a non-existent VPN.
- 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. Use the **`display sflow`** command to verify that sFlow is correctly configured.

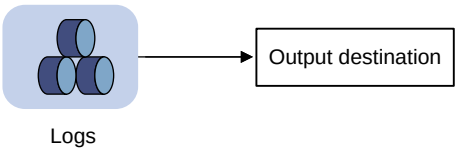
2. Verify that a correct IP address is configured for the device to communicate with the sFlow collector.
3. Verify that the physical link between the device and the sFlow collector is up.
4. Verify that the VPN bound to the sFlow collector already exists.
5. Verify that the length of an sFlow packet 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 setting) that flow sampling can copy per packet.

Configuring the information center

About the information center

The information center on the device receives logs generated by source modules and outputs logs to different destinations according to log output rules. Based on the logs, you can monitor device performance and troubleshoot network problems.

Figure 115 Information center diagram



Log types

Logs are classified into the following types:

- **Standard system logs**—Record common system information. Unless otherwise specified, the term "logs" in this document refers to standard system logs.
- **Diagnostic logs**—Record debugging 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 debugging 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 information center outputs logs with a severity level that is higher than or equal to the specified level. For example, if you specify a severity level of 6 (informational), logs that have a severity level from 0 to 6 are output.

Table 38 Log levels

Severity value	Level	Description
0	Emergency	The system is unusable. For example, the system authorization has expired.
1	Alert	Action must be taken immediately. For example, traffic on an interface exceeds the upper limit.
2	Critical	Critical condition. For example, the device temperature exceeds the upper limit, the power module fails, or the fan tray fails.
3	Error	Error condition. For example, the link state changes.
4	Warning	Warning condition. For example, an interface is disconnected, or the memory resources are used up.
5	Notification	Normal but significant condition. For example, a terminal logs in to the device, or the device reboots.

Severity value	Level	Description
6	Informational	Informational message. For example, a command or a ping operation is executed.
7	Debugging	Debugging message.

Log destinations

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

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 39 shows the default log output rules.

Table 39 Default output rules

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

Default output rules for diagnostic logs

Diagnostic logs can only be output to the diagnostic log file, and cannot be filtered by source modules and severity levels. Table 40 shows the default output rule for diagnostic logs.

Table 40 Default output rule for diagnostic logs

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

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 41 shows the default output rule for security logs.

Table 41 Default output rule for security logs

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

Default output rules for hidden logs

Hidden logs can be output to the log host, the log buffer, and the log file. Table 42 shows the default output rules for hidden logs.

Table 42 Default output rules for hidden logs

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

Default output rules for trace logs

Trace logs can only be output to the trace log file, and cannot be filtered by source modules and severity levels. Table 43 shows the default output rules for trace logs.

Table 43 Default output rules for trace logs

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

Log formats and field descriptions

Log formats

The format of logs varies by output destinations. Table 44 shows the original format of log information, which might be different from what you see. The actual format varies by the log resolution tool used.

Table 44 Log formats

Output destination	Format
Console, monitor terminal, log buffer, or log file	Prefix Timestamp Sysname Module/Level/Mnemonic: Content **Example**: `%Nov 24 14:21:43:502 2016 Sysname SHELL/5/SHELL_LOGIN: VTY logged in from 192.168.1.26`

Output destination	Format
Log host	**Standard format**: <PRI>Timestamp Sysname %%vvModule/Level/Mnemonic: Location; Content **Example**: <190>Nov 24 16:22:21 2016 Sysname %%10SHELL/5/SHELL_LOGIN: -DevIP=1.1.1.1; VTY logged in from 192.168.1.26<190>Nov 24 16:22:21 2016 Sysname %%10 SHELL/5/SHELL_LOGIN: -DevIP=1.1.1.1; VTY logged in from 192.168.1.26 **Unicom format**: <PRI>Timestamp Hostip vvModule/Level/Serial_number: Content **Example**: <189>Oct 13 16:48:08 2016 10.1.1.1 10SHELL/5/210231a64jx073000020: VTY logged in from 192.168.1.21 **CMCC format**: <PRI>Timestamp Sysname %vvModule/Level/Mnemonic: Location; Content **Example**: <189>Oct 9 14:59:04 2016 Sysname %10SHELL/5/SHELL_LOGIN: -DevIP=1.1.1.1; VTY logged in from 192.168.1.21

Log field description

Table 45 Log field description

Field	Description
Prefix (information type)	A log to a destination other than the log host has an identifier in front of the timestamp: • An identifier of percent sign (%) indicates a log with a level equal to or higher than informational. • An identifier of asterisk (*) indicates a debugging log or a trace log. • An identifier of caret (^) indicates a diagnostic log.
PRI (priority)	A log destined for the log host has a priority identifier in front of the timestamp. The priority is calculated by using this formula: facility*8+level, where: • **facility** is the facility name. Facility names local0 through local7 correspond to values 16 through 23. The facility name can be configured using the **info-center loghost** command. It is used to identify log sources on the log host, and to query and filter the logs from specific log sources. • **level** is in the range of 0 to 7. See Table 38 for more information about severity levels.
Timestamp	Records the time when the log was generated. Logs sent to the log host and those sent to the other destinations have different timestamp precisions, and their timestamp formats are configured with different commands. For more information, see Table 46 and Table 47.
Hostip	Source IP address of the log. If the **info-center loghost source** command is configured, this field displays the IP address of the specified source interface. Otherwise, this field displays the sysname. This field exists only in logs that are sent to the log host in unicom format.
Serial number	Serial number of the device that generated the log. This field exists only in logs that are sent to the log host in unicom format.
Sysname (host name or host IP address)	The sysname is the host name or IP address of the device that generated the log. You can use the **sysname** command to modify the name of the device.

Field	Description
%% (vendor ID)	Indicates that the information was generated by an HPE device. This field exists only in logs sent to the log host.
vv (version information)	Identifies the version of the log, and has a value of 10. This field exists only in logs that are sent to the log host.
Module	Specifies the name of the module that generated the log. You can enter the **info-center source ?** command in system view to view the module list.
Level	Identifies the level of the log. See Table 38 for more information about severity levels.
Mnemonic	Describes the content of the log. It contains a string of up to 32 characters.
Location	Optional field that identifies the log sender. This field exists only in logs that are sent to the log host in standard or CMCC format. The field contains the following information: • **DevIp**—IP address of the log sender. • **Slot**—Member ID of the IRF member device that sent the log.
Content	Provides the content of the log.

Table 46 Timestamp precisions and configuration commands

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

Table 47 Description of the timestamp parameters

Timestamp parameters	Description
boot	Time that has elapsed since system startup, in the format of xxx.yyy. xxx represents the higher 32 bits, and yyy represents the lower 32 bits, of milliseconds elapsed. Logs that are sent to all destinations other than a log host support this parameter. **Example**: `%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.

Timestamp parameters	Description
date	Current date and time. • For logs output to a log host, the timestamp can be in the format of MMM DD hh:mm:ss YYYY (accurate to seconds) or MMM DD hh:mm:ss.ms YYYY (accurate to milliseconds). • For logs output to other destinations, the timestamp is in the format of MMM DD hh:mm:ss:ms YYYY. All logs support this parameter. **Example**: `%May 30 05:36:29:579 2018 Sysname FTPD/5/FTPD_LOGIN: User ftp (192.168.1.23) has logged in successfully.` May 30 05:36:29:579 2018 is a timestamp in the **date** format in logs sent to the console.
iso	Timestamp format stipulated in ISO 8601, accurate to seconds (default) or milliseconds. Only logs that are sent to a log host support this parameter. **Example**: `<189>2018-05-30T06:42:44 Sysname %%10FTPD/5/FTPD_LOGIN: User ftp (192.168.1.23) has logged in successfully.` 2018-05-30T06:42:44 is a timestamp in the **iso** format accurate to seconds. A timestamp accurate to milliseconds is like 2018-05-30T06:42:44.708.
none	No timestamp is included. All logs support this parameter. **Example**: `% Sysname FTPD/5/FTPD_LOGIN: User ftp (192.168.1.23) has logged in successfully.` No timestamp is included.
no-year-date	Current date and time without year or millisecond information, in the format of MMM DD hh:mm:ss. Only logs that are sent to a log host support this parameter. **Example**: `<189>May 30 06:44:22 Sysname %%10FTPD/5/FTPD_LOGIN: User ftp (192.168.1.23) has logged in successfully.` May 30 06:44:22 is a timestamp in the **no-year-date** format.

FIPS compliance

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

Information center tasks at a glance

Managing standard system logs

1. Enabling the information center
2. Outputting logs to various destinations

   Choose the following tasks as needed:

- Outputting logs to the console
- Outputting logs to the monitor terminal
- Outputting logs to log hosts
- Outputting logs to the log buffer
- Saving logs to the log file

3. (Optional.) Setting the minimum storage period
4. (Optional.) Enabling synchronous information output
5. (Optional.) Configuring log suppression

   Choose the following tasks as needed:
   - Enabling duplicate log suppression
   - Configuring log suppression for a module
   - Disabling an interface from generating link up or link down logs
6. (Optional.) Enabling SNMP notifications for system logs

Managing hidden logs

1. Enabling the information center
2. Outputting logs to various destinations

   Choose the following tasks as needed:
   - Outputting logs to log hosts
   - Outputting logs to the log buffer
   - Saving logs to the log file
3. (Optional.) Setting the minimum storage period
4. (Optional.) Configuring log suppression

   Choose the following tasks as needed:
   - Enabling duplicate log suppression
   - Configuring log suppression for a module

Managing security logs

1. Enabling the information center
2. (Optional.) Configuring log suppression

   Choose the following tasks as needed:
   - Enabling duplicate log suppression
   - Configuring log suppression for a module
3. Managing security logs
   - Saving security logs to the security log file
   - Managing the security log file

Managing diagnostic logs

1. Enabling the information center
2. (Optional.) Configuring log suppression

   Choose the following tasks as needed:
   - Enabling duplicate log suppression

- Configuring log suppression for a module

3. Saving diagnostic logs to the diagnostic log file

Managing trace logs

1. Enabling the information center
2. (Optional.) Configuring log suppression

   Choose the following tasks as needed:
   - Enabling duplicate log suppression
   - Configuring log suppression for a module
3. Setting the maximum size of the trace log file

Enabling the information center

About this task

The information center can output logs only after it is enabled.

Procedure

1. Enter system view.

   system-view
2. Enable the information center.

   info-center enable

   The information center is enabled by default.

Outputting logs to various destinations

Outputting logs to the console

Restrictions and guidelines

The **terminal monitor**, **terminal debugging**, and **terminal logging** commands take effect only for the current connection between the terminal and the device. If a new connection is established, the default is restored.

Procedure

1. Enter system view.

   system-view
2. (Optional.) Configure an output rule for sending logs to the console.

   info-center source { *module-name* | **default** } **console** { **deny** | **level** *severity* }

   For information about the default output rules, see "Default output rules for logs."
3. (Optional.) Configure the timestamp format.

   info-center timestamp { **boot** | **date** | **none** }

   The default timestamp format is **date**.
4. Return to user view.

   quit
5. Enable log output to the console.

terminal monitor

By default, log output to the console is enabled.

6. Enable output of debugging messages to the console.

   terminal debugging

   By default, output of debugging messages to the console is disabled.

   This command enables output of debugging-level log messages to the console.

7. Set the lowest severity level of logs that can be output to the console.

   terminal logging level *severity*

   The default setting is 6 (informational).

Outputting logs to the monitor terminal

About this task

Monitor terminals refer to terminals that log in to the device through the AUX, VTY, or TTY line.

Restrictions and guidelines

The **terminal monitor**, **terminal debugging**, and **terminal logging** commands take effect only for the current connection between the terminal and the device. If a new connection is established, the default is restored.

Procedure

1. Enter system view.

   system-view

2. (Optional.) Configure an output rule for sending logs to the monitor terminal.

   info-center source { *module-name* | **default** } **monitor** { **deny** | **level** *severity* }

   For information about the default output rules, see "Default output rules for logs."

3. (Optional.) Configure the timestamp format.

   info-center timestamp { **boot** | **date** | **none** }

   The default timestamp format is **date**.

4. Return to user view.

   quit

5. Enable log output to the monitor terminal.

   terminal monitor

   By default, log output to the monitor terminal is disabled.

6. Enable output of debugging messages to the monitor terminal.

   terminal debugging

   By default, output of debugging messages to the monitor terminal is disabled.

   This command enables output of debugging-level log messages to the monitor terminal.

7. Set the lowest level of logs that can be output to the monitor terminal.

   terminal logging level *severity*

   The default setting is 6 (informational).

Outputting logs to log hosts

Restrictions and guidelines

The device supports the following methods (in descending order of priority) for outputting logs of a module to designated log hosts:

- Fast log output.

  For information about the modules that support fast log output and how to configure fast log output, see "Configuring fast log output."
- Flow log.

  For information about the modules that support flow log output and how to configure flow log output, see "Configuring flow log."
- Information center.

If you configure multiple log output methods for a module, only the method with the highest priority takes effect.

Procedure

1. Enter system view.

   system-view
2. (Optional.) Configure a log output filter or a log output rule. Choose one option as needed:
   - Configure a log output filter.

     info-center filter *filter-name* { *module-name* | **default** } { **deny** | **level** *severity* }

     You can create multiple log output filters. When specifying a log host, you can apply a log output filter to the log host to control log output.
   - Configure a log output rule for the log host output destination.

     info-center source { *module-name* | **default** } **loghost** { **deny** | **level** *severity* }

     For information about the default log output rules for the log host output destination, see "Default output rules for logs."

   The system chooses the settings to control log output to a log host in the following order:

   a. Log output filter applied to the log host.

   b. Log output rules configured for the log host output destination by using the **info-center source** command.

   c. Default log output rules (see "Default output rules for logs").
3. (Optional.) Specify a source IP address for logs sent to log hosts.

   info-center loghost source *interface-type interface-number*

   By default, the source IP address of logs sent to log hosts is the primary IP address of their outgoing interfaces.
4. (Optional.) Specify the format in which logs are output to log hosts.

   info-center format { **unicom** | **cmcc** }

   By default, logs are output to log hosts in standard format.
5. (Optional.) Configure the timestamp format.

   info-center timestamp loghost { **date** [**with-milliseconds**] | **iso** [**with-milliseconds** | **with-timezone**] * | **no-year-date** | **none** }

   The default timestamp format is **date**.
6. Specify a log host and configure related parameters.

info-center loghost [**vpn-instance** *vpn-instance-name*] { *hostname* | *ipv4-address* | **ipv6** *ipv6-address* } [**port** *port-number*] [**dscp** *dscp-value*] [**facility** *local-number*] [**filter** *filter-name*]

By default, no log hosts or related parameters are specified.

The value for the *port-number* argument must be the same as the value configured on the log host. Otherwise, the log host cannot receive logs.

Outputting logs to the log buffer

1. Enter system view.

   system-view
2. (Optional.) Configure an output rule for sending logs to the log buffer.

   info-center source { *module-name* | **default** } **logbuffer** { **deny** | **level** *severity* }

   For information about the default output rules, see "Default output rules for logs."
3. (Optional.) Configure the timestamp format.

   info-center timestamp { **boot** | **date** | **none** }

   The default timestamp format is **date**.
4. Enable log output to the log buffer.

   info-center logbuffer

   By default, log output to the log buffer is enabled.
5. (Optional.) Set the maximum log buffer size.

   info-center logbuffer size *buffersize*

   By default, a maximum of 512 logs can be buffered.

Saving logs to the log file

About this task

By default, the log file feature saves logs from the log file buffer to the log file at the specified saving interval. You can also manually trigger an immediate saving of buffered logs to the log file. After saving logs to the log file, the system clears the log file buffer.

The device automatically creates log files as needed. Each log file has a maximum capacity.

The device supports multiple general log files. The log files are named as **logfile1.log**, **logfile2.log**, and so on.

When **logfile1.log** is full, the system compresses **logfile1.log** as **logfile1.log.gz** and creates a new log file named **logfile2.log**. The process repeats until the last log file is full.

After the last log file is full, the device repeats the following process:

1. The device locates the oldest compressed log file **logfileX.log.gz** and creates a new file using the same name (**logfileX.log**).
2. When **logfileX.log** is full, the device compresses the log file as **logfileX.log.gz** to replace the existing file **logfileX.log.gz**.

As a best practice, back up the log files regularly to avoid loss of important logs.

You can enable log file overwrite-protection to stop the device from saving new logs when no log file space or storage device space is available.

TIP:

Clean up the storage space of the device regularly to ensure sufficient storage space for the log file feature.

Procedure

1. Enter system view.

   system-view
2. (Optional.) Configure an output rule for sending logs to the log file.

   info-center source { *module-name* | **default** } **logfile** { **deny** | **level** *severity* }

   For information about the default output rules, see "Default output rules for logs."
3. Enable the log file feature.

   info-center logfile enable

   By default, the log file feature is enabled.
4. (Optional.) Enable log file overwrite-protection.

   info-center logfile overwrite-protection [**all-port-powerdown**]

   By default, log file overwrite-protection is disabled.

   Log file overwrite-protection is supported only in FIPS mode.
5. (Optional.) Set the maximum log file size.

   info-center logfile size-quota *size*

   The default maximum log file size is 20 MB.
6. (Optional.) Specify the log file directory.

   info-center logfile directory *dir-name*

   The default log file directory is **flash:/logfile**.

   The device uses the default log file directory after an IRF reboot or a master/subordinate switchover.
7. Save logs in the log file buffer to the log file. Choose one option as needed:
   - Configure the automatic log file saving interval.

     info-center logfile frequency *freq-sec*

     The default saving interval is 86400 seconds.
   - Manually save logs in the log file buffer to the log file.

     logfile save

     This command is available in any view.

Setting the minimum storage period

About setting the minimum storage period

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

For logs

By default, when the number of buffered logs reaches the maximum, 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.

For general log files

By default, when the last general log file is full, the device locates the oldest compressed general log file **logfileX.log.gz** and creates a new file using the same name (**logfileX.log**).

After the minimum storage period is set, the system identifies the storage period of the compressed log file before creating a new log file with the same name. The system current time minus the log file's last modification time is the log file's storage period.

- If the storage period of the compressed log file is shorter than or equal to the minimum storage period, the system stops saving new logs.
- If the storage period of the compressed log file is longer than the minimum storage period, the system creates a new file to save new logs.

For more information about log saving, see "Saving logs to the log file."

Procedure

1. Enter system view.
   ```
   system-view
   ```
2. Set the minimum storage period.
   ```
   info-center syslog min-age min-age
   ```
   By default, the minimum storage period is not set.

Enabling synchronous information output

About this task

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.

Procedure

1. Enter system view.
   ```
   system-view
   ```
2. Enable synchronous information output.
   ```
   info-center synchronous
   ```
   By default, synchronous information output is disabled.

Configuring log suppression

Enabling duplicate log suppression

About this task

Output of consecutive duplicate logs (logs that have the same module name, level, mnemonic, location, and text) wastes system and network resources.

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

- If only duplicate logs are received during the suppression period, the information center does not output the duplicate logs. When the suppression period expires, the information center outputs the suppressed log and the number of times the log is suppressed.
- If a different log is received during the suppression period, the information center performs the following operations:
  - Outputs the suppressed log and the number of times the log is suppressed.
  - Outputs the different log and starts a suppression period for that log.
- If no log is received within the suppression period, the information center does not output any message when the suppression period expires.

Procedure

1. Enter system view.

   system-view
2. Enable duplicate log suppression.

   info-center logging suppress duplicates

   By default, duplicate log suppression is disabled.

Configuring log suppression for a module

About this task

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.

Procedure

1. Enter system view.

   system-view
2. Configure a log suppression rule for a module.

   info-center logging suppress module *module-name* **mnemonic** { **all** | *mnemonic-value* }

   By default, the device does not suppress output of any logs from any modules.

Disabling an interface from generating link up or link down logs

About this task

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

- You are concerned about the states of only 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.

Use the default setting in normal cases to avoid affecting interface status monitoring.

Procedure

1. Enter system view.
   system-view
2. Enter interface view.
   interface *interface-type interface-number*
3. Disable the interface from generating link up or link down logs.
   undo enable log updown
   By default, an interface generates link up and link down logs when the interface state changes.

Enabling SNMP notifications for system logs

About this task

This feature enables the device to send an SNMP notification for each log message it outputs. The device encapsulates the logs in SNMP notifications and then sends them to the SNMP module and the log trap buffer.

You can configure the SNMP module to send received SNMP notifications in SNMP traps or informs to remote hosts. For more information, see "Configuring SNMP."

To view the traps in the log trap buffer, access the MIB corresponding to the log trap buffer.

Procedure

1. Enter system view.
   system-view
2. Enable SNMP notifications for system logs.
   snmp-agent trap enable syslog
   By default, the device does not send SNMP notifications for system logs.
3. Set the maximum number of traps that can be stored in the log trap buffer.
   info-center syslog trap buffersize *buffersize*
   By default, the log trap buffer can store a maximum of 1024 traps.

Managing security logs

Saving security logs to the security log file

About this task

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.

After you enable the security log file feature, the system processes security logs as follows:

1. Outputs security logs to the security log file buffer.
2. Saves logs from the security log file buffer to the security log file at the specified interval.
   If you have the security-audit role, you can also manually save security logs to the security log file.
3. Clears the security log file buffer immediately after the security logs are saved to the security log file.

Restrictions and guidelines

The device supports only one security log file. The system will overwrite old logs with new logs when the security log file is full. To avoid security log loss, you can set an alarm threshold for the security log file usage ratio. When the alarm threshold is reached, the system outputs a message to inform you of the alarm. You can log in to the device with the security-audit user role and back up the security log file to prevent the loss of important data.

Procedure

1. Enter system view.

   system-view

2. Enable the security log file feature.

   info-center security-logfile enable

   By default, the security log file feature is disabled.

3. Set the interval at which the system saves security logs.

   info-center security-logfile frequency *freq-sec*

   The default security log file saving interval is 86400 seconds.

4. (Optional.) Set the maximum size for the security log file.

   info-center security-logfile size-quota *size*

   The default maximum security log file size is 10 MB.

5. (Optional.) Set the alarm threshold of the security log file usage.

   info-center security-logfile alarm-threshold *usage*

   By default, the alarm threshold of the security log file usage ratio is 80. When the usage of the security log file reaches 80%, the system will send a message.

Managing the security log file

Restrictions and guidelines

To use the security log file management commands, you must have the security-audit user role. For information about configuring the security-audit user role, see AAA in *Security Configuration Guide*.

Procedure

1. Enter system view.

   system-view

2. Change the directory of the security log file.

   info-center security-logfile directory *dir-name*

   By default, the security log file is saved in the **seclog** directory in the root directory of the storage device.

   The device uses the default security log file directory after an IRF reboot or a master/subordinate switchover.

3. Manually save all logs in the security log file buffer to the security log file.

   security-logfile save

   This command is available in any view.

4. (Optional.) Display the summary of the security log file.

   display security-logfile summary

   This command is available in any view.

Saving diagnostic logs to the diagnostic log file

About this task

By default, the diagnostic log file feature saves diagnostic logs from the diagnostic log file buffer to the diagnostic log file at the specified saving interval. You can also manually trigger an immediate saving of 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 device supports only one diagnostic log file. The diagnostic log file has a maximum capacity. When the capacity is reached, the system replaces the oldest diagnostic logs with new logs.

Procedure

1. Enter system view.
   `system-view`
2. Enable the diagnostic log file feature.
   `info-center diagnostic-logfile enable`
   By default, the diagnostic log file feature is enabled.
3. (Optional.) Set the maximum diagnostic log file size.
   `info-center diagnostic-logfile quota` *`size`*
   The default maximum diagnostic log file size is 10 MB.
4. (Optional.) Specify the diagnostic log file directory.
   `info-center diagnostic-logfile directory` *`dir-name`*
   The default diagnostic log file directory is **flash:/diagfile**.
   The device uses the default diagnostic log file directory after an IRF reboot or a master/subordinate switchover.
5. Save diagnostic logs in the diagnostic log file buffer to the diagnostic log file. Choose one option as needed:
   - Configure the automatic diagnostic log file saving interval.
     `info-center diagnostic-logfile frequency` *`freq-sec`*
     The default diagnostic log file saving interval is 86400 seconds.
   - Manually save diagnostic logs to the diagnostic log file.
     `diagnostic-logfile save`
     This command is available in any view.

Setting the maximum size of the trace log file

About this task

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.

Procedure

1. Enter system view.
   `system-view`
2. Set the maximum size for the trace log file.
   `info-center trace-logfile quota` *`size`*
   The default maximum size of the trace log file is 10 MB.

Display and maintenance commands for information center

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

Task	Command
Display the diagnostic log file configuration.	**display diagnostic-logfile summary**
Display the information center configuration.	**display info-center**
Display information about log output filters.	**display info-center filter** [*filter-name*]
Display log buffer information and buffered logs.	**display logbuffer** [**reverse**] [**level** *severity* | **size** *buffersize* | **slot** *slot-number*] * [**last-mins** *mins*]
Display the log buffer summary.	**display logbuffer summary** [**level** *severity* | **slot** *slot-number*] *
Display the content of the log file buffer.	**display logfile buffer** [**module** *module-name*]
Display the log file configuration.	**display logfile summary**
Display the content of the security log file buffer. (To execute this command, you must have the security-audit user role.)	**display security-logfile buffer**
Display summary information of the security log file .	**display security-logfile summary**
Clear the log buffer.	**reset logbuffer**

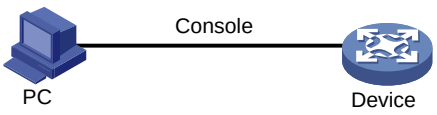
Information center configuration examples

Example: Outputting logs to the console

Network configuration

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

Figure 116 Network diagram



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 minimum severity level of **warning**.

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

Enable log output to the console.

```
<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.

Example: Outputting logs to a UNIX log host

Network configuration

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

Figure 117 Network diagram

Procedure

1. Make sure the device and the log host can reach each other. (Details not shown.)
2. Configure the device:

   # Enable the information center.

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

   # Specify log host 1.2.0.1/16 with **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 minimum severity level of **informational**.

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

3. Configure the log host:

   The log host configuration procedure varies by the vendor of the UNIX operating system. The following shows an example:

   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 the device.

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

c. Edit 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. The value **info** indicates the **informational** severity level. The UNIX system records the log information that has a minimum severity level of **informational** to file **/var/log/Device/info.log**.

NOTE:

Follow these guidelines while editing 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 to the log host correctly.

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

```
# 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.

Example: Outputting logs to a Linux log host

Network configuration

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

Figure 118 Network diagram

Procedure

1. Make sure the device and the log host can reach each other. (Details not shown.)
2. Configure the device:

Enable the information center.

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

Specify log host 1.2.0.1/16 with **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 minimum severity level of **informational**.

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

3. Configure the log host:

The log host configuration procedure varies by the vendor of the Linux operating system. The following shows an example:

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

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

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

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

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

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

NOTE:

Follow these guidelines while editing 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 to the log host correctly.

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

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

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

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

Configuring GOLD

About GOLD

Generic Online Diagnostics (GOLD) performs the following operations:

- Runs diagnostic tests on a device to inspect device ports, RAM, chip, connectivity, forwarding paths, and control paths for hardware faults.
- Reports the problems to the system.

Types of GOLD diagnostics

GOLD diagnostics are divided into the following types:

- **Monitoring diagnostics**—Run diagnostic tests periodically when the system is in operation and record test results. Monitoring diagnostics execute only non-disruptive tests.
- **On-demand diagnostics**—Enable you to manually start or stop diagnostic tests during system operation.

GOLD diagnostic tests

Each kind of diagnostics runs its diagnostic tests. The parameters of a diagnostic test include test name, type, description, attribute (disruptive or non-disruptive), default status, and execution interval.

Support for the diagnostic tests and default values for a test's parameters depend on the device model. You can modify part of the parameters by using the commands provided by this document.

The diagnostic tests are released with the system software image of the device. All enabled diagnostic tests run in the background. You can use the **`display`** commands to view test results and logs to verify hardware faults.

GOLD tasks at a glance

To configure GOLD, perform the following tasks:

1. Configuring diagnostics

   Choose the following tasks as needed:

   - Configuring monitoring diagnostics
   - Configuring on-demand diagnostics
2. (Optional.) Simulating diagnostic tests
3. (Optional.) Configuring the log buffer size

Configuring monitoring diagnostics

About this task

The system automatically executes monitoring diagnostic tests that are enabled by default after the device starts. Use the **`diagnostic monitor enable`** command to enable monitoring diagnostic tests that are disabled by default.

Procedure

1. Enter system view.

   system-view

2. Enable monitoring diagnostics.

   diagnostic monitor enable slot *slot-number-list* [**test** *test-name*]

   By default, monitoring diagnostics are enabled.

3. Set an execution interval for monitoring diagnostic tests.

   diagnostic monitor interval slot *slot-number-list* [**test** *test-name*] **time** *interval*

   By default, the execution interval varies by monitoring diagnostic test. To display the execution interval of a monitoring diagnostic test, execute the **display diagnostic content** command.

   The configured interval cannot be smaller than the minimum execution interval of the tests. Use the **display diagnostic content verbose** command to view the minimum execution interval of the tests.

Configuring on-demand diagnostics

About this task

You can stop an on-demand diagnostic test by using any of the following commands:

- Use the **diagnostic ondemand stop** command to immediately stop the test.
- Use the **diagnostic ondemand repeating** command to configure the number of executions for the test.
- Use the **diagnostic ondemand failure** command to configure the maximum number of failed tests before the system stops the test.

Restrictions and guidelines

The **diagnostic ondemand** commands are effective only during the current system operation. These commands are restored to the default after you restart the device.

Procedure

To configure on-demand diagnostics, perform the following steps in user view:

1. Configure the number of executions.

   diagnostic ondemand repeating *repeating-number*

   The default value for the *repeating-number* argument is 1.

   This command applies only to diagnostic tests to be enabled.

2. Configure the number of failed tests.

   diagnostic ondemand failure *failure-number*

   By default, the maximum number of failed tests is not specified.

   Configure a number no larger than the configured *repeating-number* argument.

   This command applies only to diagnostic tests to be enabled.

3. Enable on-demand diagnostics.

   diagnostic ondemand start slot *slot-number-list* **test** { *test-name* | **non-disruptive** } [**para** *parameters*]

   The system runs the tests according to the default configuration if you do not perform the first two configurations.

4. (Optional.) Stop on-demand diagnostics.

diagnostic ondemand stop slot *slot-number-list* **test** { *test-name* | **non-disruptive** }

You can manually stop all on-demand diagnostic tests.

Simulating diagnostic tests

About this task

Test simulation verifies GOLD frame functionality. When you use the **diagnostic simulation** commands to simulate a diagnostic test, only part of the test code is executed to generate a test result. Test simulation does not trigger hardware correcting actions such as device restart and active/standby switchover.

Restrictions and guidelines

Only monitoring diagnostics and on-demand diagnostics support test simulation.

Procedure

To simulate a test, execute the following command in user view:

diagnostic simulation slot *slot-number-list* **test** *test-name* { **failure** | **random-failure** | **success** }

By default, the system runs a test instead of simulating it.

Configuring the log buffer size

About this task

GOLD saves test results in the form of logs. You can use the **display diagnostic event-log** command to view the logs.

Procedure

1. Enter system view.

   system-view

2. Configure the maximum number of GOLD logs that can be saved.

   diagnostic event-log size *number*

   By default, GOLD saves 512 log entries at most.

   When the number of logs exceeds the configured log buffer size, the system deletes the oldest entries.

Display and maintenance commands for GOLD

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

Task	Command
Display test content.	**display diagnostic content** [**slot** *slot-number*] [**verbose**]
Display GOLD logs.	**display diagnostic event-log** [**error**	**info**]
Display configurations of on-demand diagnostics.	**display diagnostic ondemand configuration**

Task	Command
Display test results.	**display diagnostic result** [**slot** *slot-number* [**test** *test-name*]] [**verbose**]
Display statistics for packet-related tests.	**display diagnostic result** [**slot** *slot-number* [**test** *test-name*]] **statistics**
Display configurations for simulated tests.	**display diagnostic simulation** [**slot** *slot-number*]
Clear GOLD logs.	**reset diagnostic event-log**
Clear test results.	**reset diagnostic result** [**slot** *slot-number* [**test** *test-name*]]

GOLD configuration examples

Example: Configuring GOLD

Network configuration

Enable monitoring diagnostic test **ComponentMonitor** on slot 1, and set its execution interval to 1 minute.

Procedure

View the default status and execution interval of the test on slot 1.

```
<Sysname> display diagnostic content slot 1 verbose
Diagnostic test suite attributes:
#B/*: Bootup test/NA
#O/*: Ondemand test/NA
#M/*: Monitoring test/NA
#D/*: Disruptive test/Non-disruptive test
#P/*: Per port test/NA
#A/I/*: Monitoring test is active/Monitoring test is inactive/NA

Slot 1 cpu 0:
Test name        : ComponentMonitor
Test attributes  : **M*PI
Test interval    : 00:00:10
Min interval     : 00:00:10
Correct-action   : -NA-
Description      : A Real-time test, disabled by default that checks link status between
ports.
Exec             : -NA-
```

Enable test **ComponentMonitor** on slot 1.

```
<Sysname> system-view
[Sysname] diagnostic monitor enable slot 1 test ComponentMonitor
```

Set the execution interval to **1** minute.

```
[Sysname] diagnostic monitor interval slot 1 test ComponentMonitor time 0:1:0
```

Verifying the configuration

View the test configuration.

```
[Sysname] display diagnostic content slot 1 verbose
Diagnostic test suite attributes:
#B/*: Bootup test/NA
#O/*: Ondemand test/NA
#M/*: Monitoring test/NA
#D/*: Disruptive test/Non-disruptive test
#P/*: Per port test/NA
#A/I/*: Monitoring test is active/Monitoring test is inactive/NA

Slot 1 cpu 0:
Test name        : ComponentMonitor
Test attributes  : **M*PA
Test interval    : 00:01:00
Min interval     : 00:00:10
Correct-action   : -NA-
Description      : A Real-time test, disabled by default that checks link status between
ports.
Exec             : -NA-
```

View the test result.

```
[Sysname] display diagnostic result slot 1 verbose
Slot 1 cpu 0:
  Test name               : ComponentMonitor
  Total run count         : 1247
  Total failure count     : 0
  Consecutive failure count: 0
  Last execution time     : Mon Feb 25 18:09:21 2019
  First failure time      : -NA-
  Last failure time       : -NA-
  Last pass time          : Tue Dec 25 18:09:21 2012
  Last execution result   : Success
  Last failure reason     : -NA-
  Next execution time     : Mon Feb 25 18:10:21 2019
  Port link status : Normal
```

Configuring the packet capture

About packet capture

The packet capture feature captures incoming packets. It can display the captured packets in real time, or save the captured packets to a .pcap file for future analysis.

Packet capture modes

The device supports the following packet capture modes: local packet capture, remote packet capture, and feature image-based packet capture.

Local packet capture

Local packet capture saves captured packets to a remote file on an FTP server or to a local file, or displays captured packets on the terminal.

Remote packet capture

Remote packet capture sends captured packets to the Wireshark packet analyzer installed on a PC.

Before using remote packet capture, you must install the Wireshark software on a PC and connect the PC to the device.

Feature image-based packet capture

Feature image-based packet capture saves the captured packets to a local file or displays the captured packets on the terminal. This mode can also display contents of .pcap and .pcapng files.

Feature image-based packet capture requires you to install a specific image called the packet feature image.

Only feature image-based packet capture requires the packet feature image.

Filter rule elements

Packet capture supports using a capture filter rule to filter packets to be captured or using a display filter rule to filter packets to be displayed.

A filter rule is represented by a filter expression. A filter expression contains a keyword string or multiple keyword strings that are connected by operators.

Keywords include the following types:

- **Qualifiers**—Fixed keyword strings. To use a qualifier, you must enter the qualifier literally as shown.
- **Variables**—Values assigned in the required format.

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.

For more information about capture and display filters, go to the following websites:

- http://wiki.wireshark.org/CaptureFilters
- http://wiki.wireshark.org/DisplayFilters

Building a capture filter rule

Capture filter rule keywords

Qualifiers

Table 48 Qualifiers for capture filter rules

Category	Description	Examples
Protocol	Matches a protocol. If you do not specify a protocol qualifier, the filter matches any supported protocols.	• **arp**—Matches ARP. • **icmp**—Matches ICMP. • **ip**—Matches IPv4. • **ip6**—Matches IPv6. • **tcp**—Matches TCP. • **udp**—Matches UDP.
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. For example, **port 23** is equivalent to **src or dst port 23**.	• **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.
Type	Specifies the direction type. 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**.	• **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.
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.

Variables

A capture filter variable must be modified by one or more qualifiers.

The **broadcast**, **multicast**, and all protocol qualifiers cannot modify variables. The other qualifiers must be followed by variables.

Table 49 Variable types for capture filter rules

Variable type	Description	Examples
Integer	Represented in binary, octal, decimal, or hexadecimal notation.	The **port 23** expression matches traffic sent to or from port number 23.
Integer range	Represented by hyphenated integers.	The **portrange 100-200** expression matches traffic sent to or from any ports in the range of 100 to 200.

Variable type	Description	Examples
IPv4 address	Represented in dotted decimal notation.	The **src 1.1.1.1** expression matches traffic sent from the IPv4 host at 1.1.1.1.
IPv6 address	Represented in colon hexadecimal notation.	The **dst host 1::1** expression matches traffic sent to the IPv6 host at 1::1.
IPv4 subnet	Represented by an IPv4 network ID or an IPv4 address with a mask.	Both of the following expressions match traffic sent to or from the IPv4 subnet 1.1.1.0/24: • **src 1.1.1**. • **src net 1.1.1.0/24**.
IPv6 network segment	Represented by an IPv6 address with a prefix length.	The **dst net 1::/64** expression matches traffic sent to the IPv6 network 1::/64.

Capture filter rule operators

Logical operators

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 50 Logical operators for capture filter rules

Nonalphanumeric symbol	Alphanumeric symbol	Description
!	not	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**.
&&	and	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**.
||	or	Joins two conditions. Use this operator to capture traffic that matches either of the conditions. For example, to capture traffic that is sent to or from 1.1.1.1 or 2.2.2.2, use **host 1.1.1.1 or host 2.2.2.2**.

Arithmetic operators

Table 51 Arithmetic operators for capture filter rules

Nonalphanumeric symbol	Description
+	Adds two values.
-	Subtracts one value from another.
*	Multiplies one value by another.
/	Divides one value by another.

Nonalphanumeric symbol	Description
&	Returns the result of the bitwise AND operation on two integral values in binary form.
|	Returns the result of the bitwise OR operation on two integral values in binary form.
<<	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.
>>	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.
[ ]	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).

Relational operators

Table 52 Relational operators for capture filter rules

Nonalphanumeric symbol	Description
=	Equal to. For example, **ip[6]=0x1c** matches an IPv4 packet if its seventh byte of payload is equal to 0x1c.
!=	Not equal to. For example, **len!=60** matches a packet if its length is not equal to 60 bytes.
>	Greater than. For example, **len>100** matches a packet if its length is greater than 100 bytes.
<	Less than. For example, **len<100** matches a packet if its length is less than 100 bytes.
>=	Greater than or equal to. For example, **len>=100** matches a packet if its length is greater than or equal to 100 bytes.
<=	Less than or equal to. For example, **len<=100** matches a packet if its length is less than or equal to 100 bytes.

Capture filter rule expressions

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 ip** captures IPv4 packets in VLAN 1.

To capture packets of a VLAN, set a capture filter as follows:

- To capture tagged packets that are permitted on the interface, you must use the **vlan** *vlan_id* expression prior to any other expressions. For example, use the **vlan 3 and src 192.168.1.10 and dst 192.168.1.1** expression to capture packets of VLAN 3 that are sent from 192.168.1.10 to 192.168.1.1.
- After receiving an untagged packet, the device adds a VLAN tag to the packet header. To capture the packet, add "vlan xx" to the capture filter expression. For Layer 3 packets, the xx represents the default VLAN ID of the outgoing interface. For Layer 2 packets, the xx represents the default VLAN ID of the incoming interface.

Building a display filter rule

A display filter rule only identifies the packets to display. It does not affect which packets to save in a file.

Display filter rule keywords

Qualifiers

Table 53 Qualifiers for display filter rules

Category	Description	Examples
Protocol	Matches a protocol. If you do not specify a protocol qualifier, the filter matches any supported protocols.	• **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.
Packet field	Matches a field in packets by using a dotted string in the *protocol.field*[*.level1-subfield*]…[*.leveln-subfield*] format.	• **tcp.flags.syn**—Matches the SYN bit in the flags field of TCP. • **tcp.port**—Matches the source or destination port field of TCP.

Variables

A packet field qualifier requires a variable.

Table 54 Variable types for display filter rules

Variable type	Description
Integer	Represented in binary, octal, decimal, or hexadecimal notation. For example, to display IP packets that are less than or equal to 1500 bytes, use one of the following expressions: • **ip.len le 1500**. • **ip.len le 02734**. • **ip.len le 0x436**.
Boolean	This variable type has two values: true or false. This variable type applies if you use a packet field string alone to identify the presence of a field in a packet. • If the field is present, the match result is true. The filter displays the packet. • If the field is not present, the match result is false. The filter does not display the packet. For example, to display TCP packets that contain the SYN field, use **tcp.flags.syn**.
MAC address (6 bytes)	Uses colons (:), dots (.), or hyphens (-) to break up the MAC address into two or four segments. For example, to display packets that contain a destination MAC address of ffff.ffff.ffff, use one of the following expressions: • **eth.dst==ff:ff:ff:ff:ff:ff**. • **eth.dst==ff-ff-ff-ff-ff-ff**. • **eth.dst ==ffff.ffff.ffff**.

Variable type	Description
IPv4 address	Represented in dotted decimal notation. For example: • To display IPv4 packets that are sent to or from 192.168.0.1, use **ip.addr==192.168.0.1**. • To display IPv4 packets that are sent to or from 129.111.0.0/16, use **ip.addr==129.111.0.0/16**.
IPv6 address	Represented in colon hexadecimal notation. For example: • To display IPv6 packets that are sent to or from 1::1, use **ipv6.addr==1::1**. • To display IPv6 packets that are sent to or from 1::/64, use **ipv6.addr==1::/64**.
String	Character string. For example, to display HTTP packets that contain the string **HTTP/1.1** for the request version field, use **http.request version=="HTTP/1.1"**.

Display filter rule operators

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

Logical operators

Table 55 Logical operators for display filter rules

Nonalphanumeric symbol	Alphanumeric symbol	Description
[ ]	No alphanumeric symbol is available.	Used with protocol qualifiers. For more information, see "The proto[…] expression."
!	not	Displays packets that do not match the condition connected to this operator.
&&	and	Joins two conditions. Use this operator to display traffic that matches both conditions.
||	or	Joins two conditions. Use this operator to display traffic that matches either of the conditions.

Relational operators

Table 56 Relational operators for display filter rules

Nonalphanumeric symbol	Alphanumeric symbol	Description
==	eq	Equal to. For example, **ip.src==10.0.0.5** displays packets with the source IP address as 10.0.0.5.
!=	ne	Not equal to. For example, **ip.src!=10.0.0.5** displays packets whose source IP address is not 10.0.0.5.

Nonalphanumeric symbol	Alphanumeric symbol	Description
>	gt	Greater than. For example, **frame.len>100** displays frames with a length greater than 100 bytes.
<	lt	Less than. For example, **frame.len<100** displays frames with a length less than 100 bytes.
>=	ge	Greater than or equal to. For example, **frame.len ge 0x100** displays frames with a length greater than or equal to 256 bytes.
<=	le	Less than or equal to. For example, **frame.len le 0x100** displays frames with a length less than or equal to 256 bytes.

Display filter rule expressions

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.

Restrictions and guidelines: Packet capture

To capture packets forwarded through chips, first configure a traffic behavior to mirror the traffic to the CPU.

To capture packets forwarded by the CPU, enable packet capture directly.

The packet capture feature can capture only frames 9196 bytes long or shorter.

Configuring local packet capture

To configure local packet capture, execute the following command in user view:

packet-capture local interface *interface-type interface-number* [**capture-filter** *capt-expression* | **limit-frame-size** *bytes* | **autostop filesize** *kilobytes* | **autostop duration** *seconds*] * **write** { *filepath* | **url** *url* [**username** *username* [**password** { **cipher** | **simple** } *string*]] }

The packet capture is executed in the background. After issuing this command, you can continue to configure other commands.

Configuring remote packet capture

Prerequisites

Before performing this task, prepare a PC installed with the Wireshark packet analyzer and connect the PC to the device. For more information about Wireshark, see Wireshark user guides.

Procedure

To configure remote packet capture, execute the following command in user view:

packet-capture remote interface *interface-type interface-number* [**port** *port*]

Configuring feature image-based packet capture

Restrictions and guidelines

After configuring feature image-based packet capture, you cannot configure any other commands at the CLI until the capture finishes or is stopped.

There might be a delay for the capture to stop because of heavy traffic.

Prerequisites

1. Use the **display boot-loader** command to check whether the packet capture feature image is installed.
2. If the image is not installed, install the image by using the **boot-loader**, **install**, or **issu** command series.
3. Log out of the device and then log in again.

For more information about the commands, see *Fundamentals Command Reference*.

Saving captured packets to a file

To configure feature image-based packet capture and save the captured packets to a file, execute the following command in user view:

packet-capture interface *interface-type interface-number* [**capture-filter** *capt-expression* | **limit-captured-frames** *limit* | **limit-frame-size** *bytes* | **autostop filesize** *kilobytes* | **autostop duration** *seconds* | **autostop files** *numbers* | **capture-ring-buffer filesize** *kilobytes* | **capture-ring-buffer duration** *seconds* | **capture-ring-buffer files** *numbers*] * **write** *filepath* [**raw** | { **brief** | **verbose** }] *

Displaying specific captured packets

To configure feature image-based packet capture and display specific packet data, execute the following command in user view:

packet-capture interface *interface-type interface-number* [**capture-filter** *capt-expression* | **display-filter** *disp-expression* | **limit-captured-frames** *limit* | **limit-frame-size** *bytes* | **autostop duration** *seconds*] * [**raw** | { **brief** | **verbose** }] *

Stopping packet capture

About this task

Use this task to manually stop packet capture.

Procedure

Choose one option as needed:

- Stop local or remote packet capture.

  packet-capture stop

  Execute this command in user view.
- Stop feature image-based packet capture.

  Press **Ctrl+C**.

Displaying the contents in a packet file

About this task

Use this task to display the contents of a .pcap or .pcapng file on the device. Alternatively, you can transfer the file to a PC and use Wireshark to display the file content.

Prerequisites

1. Use the **display boot-loader** command to check whether the packet capture feature image is installed.
2. If the image is not installed, install the image by using **boot-loader**, **install**, or **issu** commands.
3. Log out of the device and then log in again.

For more information about the commands, see *Fundamentals Command Reference*.

Restrictions and guidelines

To stop displaying the contents, press **Ctrl+C**.

Procedure

To display the contents in a local packet file, execute the following command in user view:

packet-capture read *filepath* [**display-filter** *disp-expression*] [**raw** | { **brief** | **verbose** }] *

Display and maintenance commands for packet capture

Execute **display** commands in any view.

Task	Command
Display status information about local or remote packet capture.	**display packet-capture status**

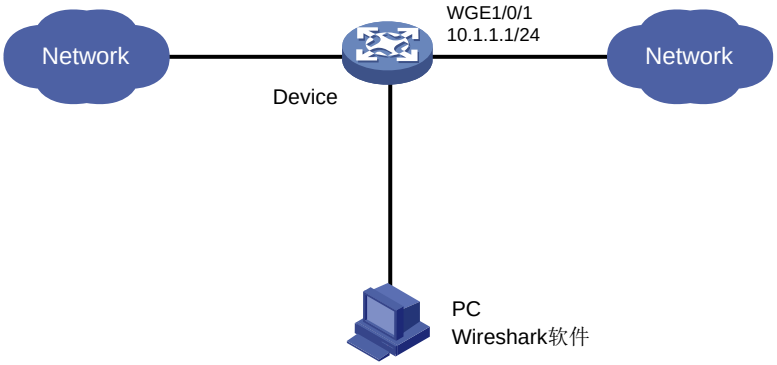
Packet capture configuration examples

Example: Configuring remote packet capture

Network configuration

As shown in Figure 119, capture packets forwarded through the CPU or chips on Layer 2 interface Twenty-FiveGigE 1/0/1. Use Wireshark to display the captured packets.

Figure 119 Network diagram



Procedure

1. Configure the device:

   # Apply a QoS policy to the incoming direction of Twenty-FiveGigE 1/0/1 to capture packets destined for the 20.1.1.0/16 network that are forwarded through chips.

   a. Create an IPv4 advanced ACL to match packets that are sent to the 20.1.1.0/16 network.

```
<Device> system-view
[Device] acl advanced 3000
[Device-acl-ipv4-adv-3000] rule permit ip destination 20.1.1.0 255.255.0.0
[Device-acl-ipv4-adv-3000] quit
```

b. Configure a traffic behavior to mirror traffic to the CPU.

```
[Device] traffic behavior behavior1
[Device-behavior-behavior1] mirror-to cpu
[Device-behavior-behavior1] quit
```

c. Configure a traffic class to use the ACL to match traffic.

```
[Device] traffic classifier classifier1
[Device-classifier-class1] if-match acl 3000
[Device-classifier-class1] quit
```

d. Configure a QoS policy. Associate the traffic class with the traffic behavior.

```
[Device] qos policy user1
[Device-qospolicy-user1] classifier classifier1 behavior behavior1
[Device-qospolicy-user1] quit
```

e. Apply the QoS policy to the incoming direction of Twenty-FiveGigE 1/0/1.

```
[Device] interface twenty-fivegige 1/0/1
[Device-Twenty-FiveGigE1/0/1] qos apply policy user1 inbound
[Device-Twenty-FiveGigE1/0/1] quit
[Device] quit
```

Configure remote packet capture on Twenty-FiveGigE 1/0/1. Set the RPCAP service port number to 2014.

```
<Device> packet-capture remote interface twenty-fivegige 1/0/1 port 2014
```

2. Configure Wireshark:

a. Start Wireshark on the PC and select **Capture** > **Options**.

b. Select **Remote** from the **Interface** list.

c. Enter the IP address of the device **10.1.1.1** and the RPCAP service port number **2014**. Make sure there are routes available between the IP address and the PC.

d. Click **OK** and then click **Start**.

The captured packets are displayed on the Wireshark.

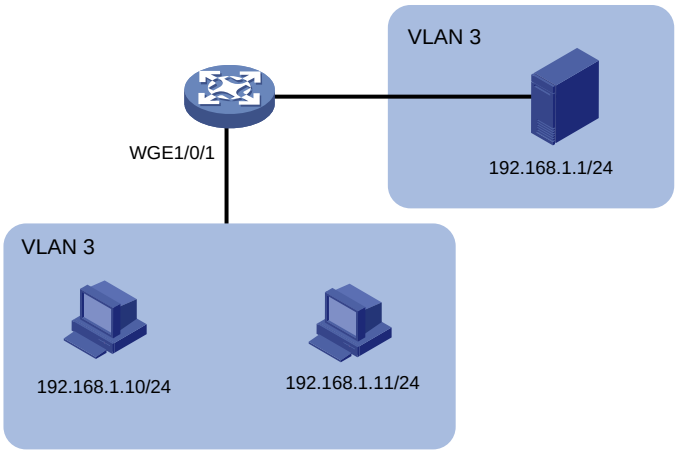
Example: Configuring feature image-based packet capture

Network configuration

As shown in Figure 120, capture incoming IP packets of VLAN 3 on Layer 2 interface Twenty-FiveGigE 1/0/1 that meet the following conditions:

- Sent from 192.168.1.10 or 192.168.1.11 to 192.168.1.1.
- Forwarded through the CPU or chips.

Figure 120 Network diagram



Procedure

1. Install the packet capture feature.

   # Display the device version information.

```
<Device> display version
HPE Comware Software, Version 7.1.070, Demo 01
Copyright (c) 2004-2017 Hewlett-Packard Development Company, L.P All rights reserved.
HPE XXX uptime is 0 weeks, 0 days, 5 hours, 33 minutes
Last reboot reason : Cold reboot
Boot image: flash:/boot-01.bin
Boot image version: 7.1.070, Demo 01
  Compiled Oct 20 2016 16:00:00
System image: flash:/system-01.bin
System image version: 7.1.070, Demo 01
  Compiled Oct 20 2016 16:00:00
...
```

   # Prepare a packet capture feature image that is compatible with the current boot and system images.

   # Download the packet capture feature image to the device. In this example, the image is stored on the TFTP server at 192.168.1.1.

```
<Device> tftp 192.168.1.1 get packet-capture-01.bin
Press CTRL+C to abort.
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 11.3M    0 11.3M    0     0   155k      0 --:--:--  0:01:14 --:--:--  194k
Writing file...Done.
```

   # Install the packet capture feature image on all IRF member devices and commit the software change. In this example, there are two IRF member devices.

```
<Device> install activate feature flash:/packet-capture-01.bin slot 1
Verifying the file flash:/packet-capture-01.bin on slot 1....Done.
Identifying the upgrade methods....Done.
Upgrade summary according to following table:

flash:/packet-capture-01.bin
```

```
  Running Version             New Version
  None                        Demo 01

  Slot                        Upgrade Way
  1                           Service Upgrade
Upgrading software images to compatible versions. Continue? [Y/N]:y
This operation might take several minutes, please wait...................Done.
<Device> install activate feature flash:/packet-capture-01.bin slot 2
Verifying the file flash:/packet-capture-01.bin on slot 2....Done.
Identifying the upgrade methods....Done.
Upgrade summary according to following table:

flash:/packet-capture-01.bin
  Running Version             New Version
  None                        Demo 01

  Slot                        Upgrade Way
  2                           Service Upgrade
Upgrading software images to compatible versions. Continue? [Y/N]:y
This operation might take several minutes, please wait...................Done.
<Device> install commit
This operation will take several minutes, please wait.....................Done.
```

Log out and then log in to the device again so you can execute the **packet-capture interface** and **packet-capture read** commands.

2. Apply a QoS policy to the incoming direction of Twenty-FiveGigE 1/0/1 to capture packets from 192.168.1.10 or 192.168.1.11 to 192.168.1.1 that are forwarded through chips.

Create an IPv4 advanced ACL to match packets that are sent from 192.168.1.10 or 192.168.1.11 to 192.168.1.1.

```
<Device> system-view
[Device] acl advanced 3000
[Device-acl-ipv4-adv-3000] rule permit ip source 192.168.1.10 0 destination
192.168.1.1 0
[Device-acl-ipv4-adv-3000] rule permit ip source 192.168.1.11 0 destination
192.168.1.1 0
[Device-acl-ipv4-adv-3000] quit
```

Configure a traffic behavior to mirror traffic to the CPU.

```
[Device] traffic behavior behavior1
[Device-behavior-behavior1] mirror-to cpu
[Device-behavior-behavior1] quit
```

Configure a traffic class to use the ACL to match traffic.

```
[Device] traffic classifier classifier1
[Device-classifier-class1] if-match acl 3000
[Device-classifier-class1] quit
```

Configure a QoS policy. Associate the traffic class with the traffic behavior.

```
[Device] qos policy user1
[Device-qospolicy-user1] classifier classifier1 behavior behavior1
[Device-qospolicy-user1] quit
```

Apply the QoS policy to the incoming direction of Twenty-FiveGigE 1/0/1.

```
[Device] interface twenty-fivegige 1/0/1
[Device-Twenty-FiveGigE1/0/1] qos apply policy user1 inbound
[Device-Twenty-FiveGigE1/0/1] quit
[Device] quit
```

3. Enable packet capture.

Capture incoming traffic on Twenty-FiveGigE 1/0/1. Set the maximum number of captured packets to 10. Save the captured packets to the **flash:/a.pcap** file.

```
<Device> packet-capture interface twenty-fivegige 1/0/1 capture-filter "vlan 3 and src 192.168.1.10 or 192.168.1.11 and dst 192.168.1.1" limit-captured-frames 10 write flash:/a.pcap
Capturing on 'Twenty-FiveGigE1/0/1'
10
```

Verifying the configuration

Telnet to 192.168.1.1 from 192.168.1.10. (Details not shown.)

Display the contents in the packet file on the device.

```
<Device> packet-capture read flash:/a.pcap
  1  0.000000 192.168.1.10 -> 192.168.1.1 TCP 62 6325 > telnet [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
  2  0.000061 192.168.1.10 -> 192.168.1.1 TCP 60 6325 > telnet [ACK] Seq=1 Ack=1 Win=65535 Len=0
  3   0.024370 192.168.1.10 -> 192.168.1.1 TELNET 60 Telnet Data ...
  4   0.024449 192.168.1.10 -> 192.168.1.1 TELNET 78 Telnet Data ...
  5   0.025766 192.168.1.10 -> 192.168.1.1 TELNET 65 Telnet Data ...
  6   0.035096 192.168.1.10 -> 192.168.1.1 TELNET 60 Telnet Data ...
  7   0.047317 192.168.1.10 -> 192.168.1.1 TCP 60 6325 > telnet [ACK] Seq=42 Ack=434 Win=65102 Len=0
  8   0.050994 192.168.1.10 -> 192.168.1.1 TCP 60 6325 > telnet [ACK] Seq=42 Ack=436 Win=65100 Len=0
  9   0.052401 192.168.1.10 -> 192.168.1.1 TCP 60 6325 > telnet [ACK] Seq=42 Ack=438 Win=65098 Len=0
 10   0.057736 192.168.1.10 -> 192.168.1.1 TCP 60 6325 > telnet [ACK] Seq=42 Ack=440 Win=65096 Len=0
```

Configuring VCF fabric

About VCF fabric

Based on OpenStack Networking (Neutron), the Virtual Converged Framework (VCF) solution provides virtual network services from Layer 2 to Layer 7 for cloud tenants. This solution breaks the boundaries between the network, cloud management, and terminal platforms and transforms the IT infrastructure to a converged framework to accommodate all applications. It also implements automated topology discovery and automated deployment of underlay networks and overlay networks to reduce the administrators' workload and speed up network deployment and upgrade.

VCF fabric topology

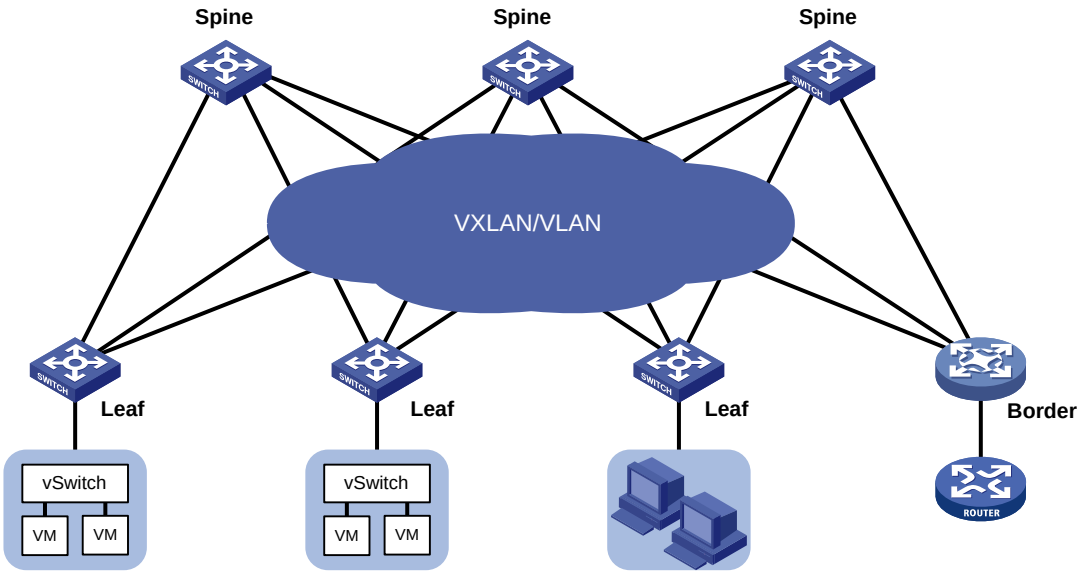
Topology for a Layer 2 data center VCF fabric

In a Layer 2 data center VCF fabric, a device has one of the following roles:

- **Spine node**—Connects to leaf nodes.
- **Leaf node**—Connects to servers.
- **Border node**—Located at the border of a VCF fabric to provide access to the external network.

Spine nodes and leaf nodes form a large Layer 2 network, which can be a VLAN, a VXLAN with a centralized IP gateway, or a VXLAN with distributed IP gateways. For more information about centralized IP gateways and distributed IP gateways, see *VXLAN Configuration Guide.*

Figure 121 Topology for a Layer 2 data center VCF fabric



Topology for a Layer 3 data center VCF fabric

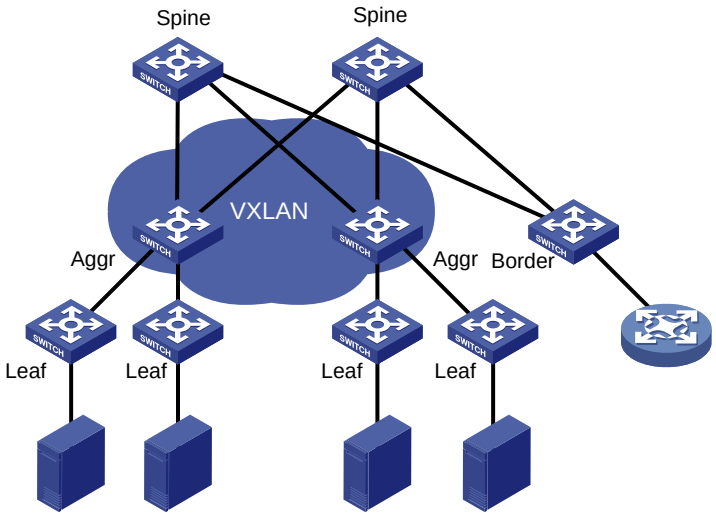
In a Layer 3 data center VCF fabric, a device has one of the following roles:

- **Spine node**—Connects to aggregate nodes.
- **Aggregate node**—Resides on the distribution layer and is located between leaf nodes and spine nodes.
- **Leaf node**—Connects to servers.

- **Border node**—Located at the border of a VCF fabric to provide access to the external network.

OSPF runs on the Layer 3 networks between the spine and aggregate nodes and between the aggregate and leaf nodes. VXLAN is used to set up the Layer 2 overlay network.

Figure 122 Topology for a Layer 3 data center VCF fabric



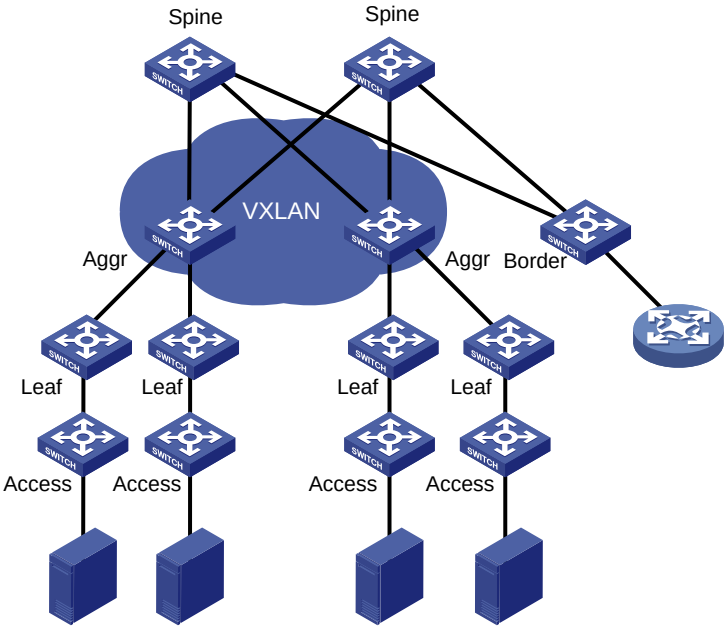
Topology for a 4-layer data center VCF fabric

In a 4-layer data center VCF fabric, a device has one of the following roles:

- **Spine node**—Connects to aggregate nodes.
- **Aggregate node**—Located between leaf nodes and spine nodes.
- **Leaf node**—Located between aggregate nodes and access nodes.
- **Access node**—An access device, which connects to an upstream leaf node and a downstream server.
- **Border node**—Located at the border of a VCF fabric to provide access to the external network.

OSPF runs on the Layer 3 networks between the spine and aggregate nodes and between the aggregate and leaf nodes. VXLAN is used to set up the Layer 2 overlay network.

Figure 123 Topology for a 4-layer data center VCF fabric



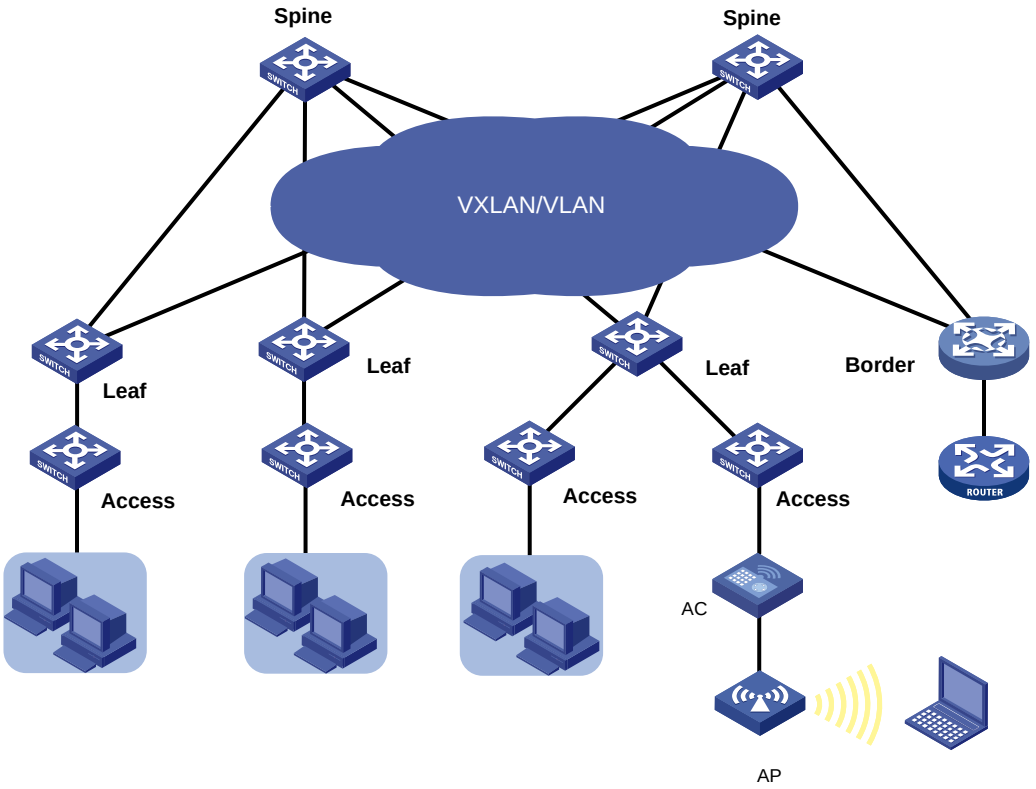
VCF fabric topology for a campus network

In a campus VCF fabric, a device has one of the following roles:

- **Spine node**—Connects to leaf nodes.
- **Leaf node**—Connects to access nodes.
- **Access node**—Connects to an upstream leaf node and downstream terminal devices. Cascading of access nodes is supported.
- **Border node**—Located at the border of a VCF fabric to provide access to the external network.

Spine nodes and leaf nodes form a large Layer 2 network, which can be a VLAN, a VXLAN with a centralized IP gateway, or a VXLAN with distributed IP gateways. For more information about centralized IP gateways and distributed IP gateways, see *VXLAN Configuration Guide.*

Figure 124 VCF fabric topology for a campus network



Neutron overview

Neutron concepts and components

Neutron is a component in OpenStack architecture. It provides networking services for VMs, manages virtual network resources (including networks, subnets, DHCP, virtual routers), and creates an isolated virtual network for each tenant. Neutron provides a unified network resource model, based on which VCF fabric is implemented.

The following are basic concepts in Neutron:

- **Network**—A virtual object that can be created. It provides an independent network for each tenant in a multitenant environment. A network is equivalent to a switch with virtual ports which can be dynamically created and deleted.
- **Subnet**—An address pool that contains a group of IP addresses. Two different subnets communicate with each other through a router.
- **Port**—A connection port. A router or a VM connects to a network through a port.
- **Router**—A virtual router that can be created and deleted. It performs routing selection and data forwarding.

Neutron has the following components:

- **Neutron server**—Includes the daemon process neutron-server and multiple plug-ins (neutron-*-plugin). The Neutron server provides an API and forwards the API calls to the configured plugin. The plug-in maintains configuration data and relationships between routers, networks, subnets, and ports in the Neutron database.
- **Plugin agent (neutron-*-agent)**—Processes data packets on virtual networks. The choice of plug-in agents depends on Neutron plug-ins. A plug-in agent interacts with the Neutron server and the configured Neutron plug-in through a message queue.

- **DHCP agent (neutron-dhcp-agent)**—Provides DHCP services for tenant networks.
- **L3 agent (neutron-l3-agent)**—Provides Layer 3 forwarding services to enable inter-tenant communication and external network access.

Neutron deployment

Neutron needs to be deployed on servers and network devices.

Table 57 shows Neutron deployment on a server.

Table 57 Neutron deployment on a server

Node	Neutron components
Controller node	• Neutron server • Neutron DB • Message server (such as RabbitMQ server) • ML2 Driver
Network node	• neutron-openvswitch-agent • neutron-dhcp-agent
Compute node	• neutron-openvswitch-agent • LLDP

Table 58 shows Neutron deployments on a network device.

Table 58 Neutron deployments on a network device

Network type	Network device	Neutron components
Centralized VXLAN IP gateway deployment	Spine	• neutron-l2-agent • neutron-l3-agent
	Leaf	neutron-l2-agent
Distributed VXLAN IP gateway deployment	Spine	N/A
	Leaf	• neutron-l2-agent • neutron-l3-agent

Figure 125 Example of Neutron deployment for centralized gateway deployment

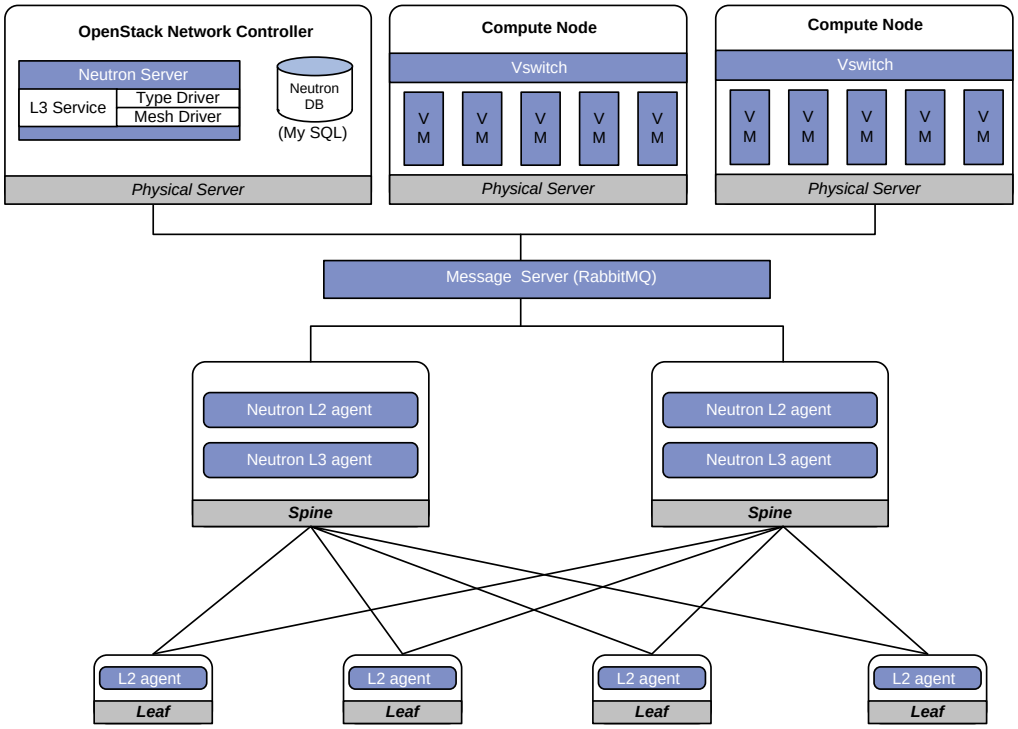
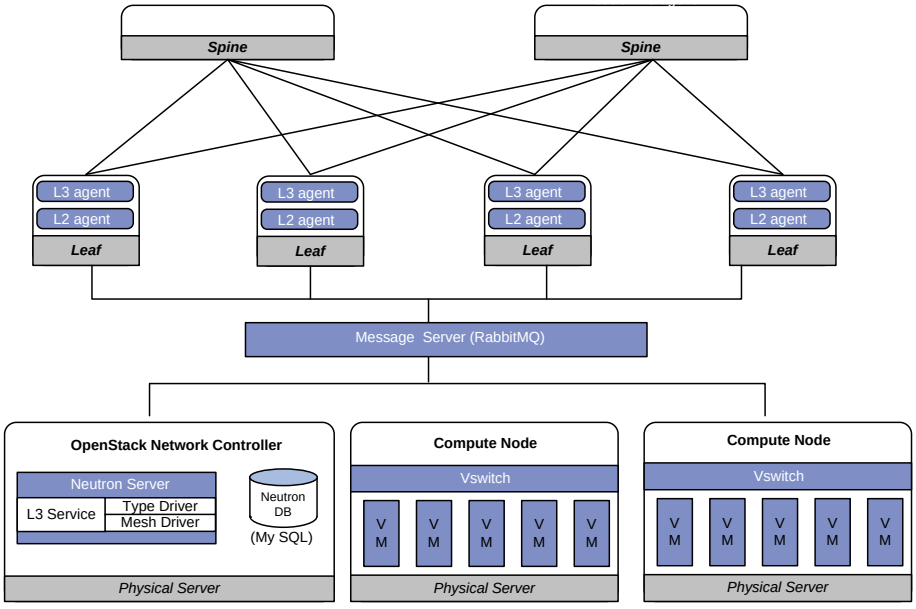


Figure 126 Example of Neutron deployment for distributed gateway deployment



Automated VCF fabric deployment

VCF provides the following features to ease deployment:

- Automated topology discovery.

  In a VCF fabric, each device uses LLDP to collect local topology information from directly-connected peer devices. The local topology information includes connection interfaces, roles, MAC addresses, and management interface addresses of the peer devices. If multiple

spine nodes exist in a VCF fabric, the master spine node collects the topology for the entire network.

- Automated underlay network deployment.

  Automated underlay network deployment sets up a Layer 3 underlay network (a physical Layer 3 network) for users. It is implemented by automatically executing configurations (such as IRF configuration and Layer 3 reachability configurations) in user-defined template files.

- Automated overlay network deployment.

  Automated overlay network deployment sets up an on-demand and application-oriented overlay network (a virtual network built on top of the underlay network). It is implemented by automatically obtaining the overlay network configuration (including VXLAN and EVPN configuration) from the Neutron server.

Process of automated VCF fabric deployment

The device finishes automated VCF fabric deployment as follows:

1. Starts up without loading configuration and then obtains an IP address, the IP address of the TFTP server, and a template file name from the DHCP server.
2. Determines the name of the template file to be downloaded based on the device role and the template file name obtained from the DHCP server. For example, 1_leaf**.template** represents a template file for leaf nodes.
3. Downloads the template file from the TFTP server.
4. Parses the template file and performs the following operations:
   - Deploys static configurations that are independent from the VCF fabric topology.
   - Deploys dynamic configurations according to the VCF fabric topology.

     The topology process notifies the automation process of creation, deletion, and status change of neighbors. Based on the topology information, the automation process completes role discovery, automatic aggregation, and IRF fabric setup.

Template file

A template file contains the following contents:

- **System-predefined variables**—The variable names cannot be edited, and the variable values are set by the VCF topology discovery feature.
- **User-defined variables**—The variable names and values are defined by the user. These variables include the username and password used to establish a connection with the RabbitMQ server, network type, and so on. The following are examples of user-defined variables:

```
#USERDEF
_underlayIPRange = 10.100.0.0/16
_master_spine_mac = 1122-3344-5566
_backup_spine_mac = aabb-ccdd-eeff
_username = aaa
_password = aaa
_rbacUserRole = network-admin
_neutron_username = openstack
_neutron_password = 12345678
_neutron_ip = 172.16.1.136
_loghost_ip = 172.16.1.136
_network_type = centralized-vxlan
...
```

- **Static configurations**—Static configurations are independent from the VCF fabric topology and can be directly executed. The following are examples of static configurations:

```
#STATICCFG
#
 clock timezone beijing add 08:00:00
#
 lldp global enable
#
 stp global enable
#
```

- **Dynamic configurations**—Dynamic configurations are dependent on the VCF fabric topology. The device first obtains the topology information through LLDP and then executes dynamic configurations. The following are examples of dynamic configurations:

```
#
interface $$_underlayIntfDown
 port link-mode route
 ip address unnumbered interface LoopBack0
 ospf 1 area 0.0.0.0
 ospf network-type p2p
 lldp management-address arp-learning
 lldp tlv-enable basic-tlv management-address-tlv interface LoopBack0
#
```

VCF fabric task at a glance

To configure a VCF fabric, perform the following tasks:

- Configuring automated VCF fabric deployment

  No tasks are required to be made on the device for automated VCF fabric deployment. However, you must make related configuration on the DHCP server and the TFTP server so the device can download and parse a template file to complete automated VCF fabric deployment.

- (Optional.) Adjust VCF fabric deployment

  If the device cannot obtain or parse the template file to complete automated VCF fabric deployment, choose the following tasks as needed:

  - Enabling VCF fabric topology discovery
  - Configuring automated underlay network deployment
  - Configuring automated overlay network deployment

Configuring automated VCF fabric deployment

Restrictions and guidelines

On a data center network, if the template file contains software version information, the device first compares the software version with the current software version. If the two versions are inconsistent, the device downloads the new software version to perform software upgrade. After restarting up, the device executes the configurations in the template file.

On a data center network, only links between leaf nodes and servers are automatically aggregated.

On a campus network, links between two access nodes cascaded through GigabitEthernet interfaces and links between leaf nodes and access nodes are automatically aggregated. For links

between spine nodes and leaf nodes, the **trunk permit vlan** command is automatically executed.

On a campus network where multiple leaf nodes that have an access node attached form an IRF fabric, make sure only one link exists between each leaf node and its connected access node.

Do not perform link migration when devices in the VCF fabric are in the process of coming online or powering down after the automated VCF fabric deployment finishes. A violation might cause link-related configuration fails to update.

The version format of a template file for automated VCF fabric deployment is *x*.*y*. Only the *x* part is examined during a version compatibility check. For successful automated deployment, make sure *x* in the version of the template file to be used is not greater than *x* in the supported version. To display the supported version of the template file for automated VCF fabric deployment, use the **display vcf-fabric underlay template-version** command.

If the template file does not include IRF configurations, the device does not save the configurations after executing all configurations in the template file. To save the configurations, use the **save** command.

Two devices with the same role can automatically set up an IRF fabric only when the IRF physical interfaces on the devices are connected.

Two IRF member devices in an IRF fabric use the following rules to elect the IRF master during automated VCF fabric deployment:

- If the uptime of both devices is shorter than two hours, the device with the higher bridge MAC address becomes the IRF master.
- If the uptime of one device is equal to or longer than two hours, that device becomes the IRF master.
- If the uptime of both devices are equal to or longer than two hours, the IRF fabric cannot be set up. You must manually reboot one of the member devices. The rebooted device will become the IRF subordinate.

If the IRF member ID of a device is not 1, the IRF master might reboot during automatic IRF fabric setup.

Procedure

1. Finish the underlay network planning (such as IP address assignment, reliability design, and routing deployment) based on user requirements.
2. Configure the DHCP server.

   Configure the IP address of the device, the IP address of the TFTP server, and names of template files saved on the TFTP server. For more information, see the user manual of the DHCP server.
3. Configure the TFTP server.

   Create template files and save the template files to the TFTP server. For more information about template files, see "Template file."
4. (Optional.) Configure the NTP server.
5. Connect the device to the VCF fabric and start the device.

   After startup, the device uses a management Ethernet interface or VLAN-interface 1 to connect to the fabric management network. Then, it downloads the template file corresponding to its device role and parses the template file to complete automated VCF fabric deployment.
6. (Optional.) Save the deployed configuration.

   If the template file does not include IRF configurations, the device will not save the configurations after executing all configurations in the template file. To save the configurations, use the **save** command. For more information about this command, see configuration file management commands in *Fundamentals Command Reference*.

Enabling VCF fabric topology discovery

1. Enter system view.
   system-view
2. Enable LLDP globally.
   lldp global enable
   By default, LLDP is disabled globally.
   You must enable LLDP globally before you enable VCF fabric topology discovery, because the device needs LLDP to collect topology data of directly-connected devices.
3. Enable VCF fabric topology discovery.
   vcf-fabric topology enable
   By default, VCF fabric topology discovery is disabled.

Configuring automated underlay network deployment

Specify the template file for automated underlay network deployment

1. Enter system view.
   system-view
2. Specify the template file for automated underlay network deployment.
   vcf-fabric underlay autoconfigure *template*
   By default, no template file is specified for automated underlay network deployment.

Specifying the role of the device in the VCF fabric

About this task

Perform this task to change the role of the device in the VCF fabric.

Restrictions and guidelines

If the device completes automated underlay network deployment by automatically downloading and parsing a template file, reboot the device after you change the device role. In this way, the device can obtain the template file corresponding to the new role and complete the automated underlay network deployment.

To use devices that have come online after automated deployment to form an IRF fabric, make sure all member devices in the IRF fabric have the same VCF fabric role.

Procedure

1. Enter system view.
   system-view
2. Specify the role of the device in the VCF fabric.
   vcf-fabric role { **access** | **aggr** | **leaf** | **spine** }
   By default, the device is a leaf node.
3. Return to system view.

```
quit
```

4. Reboot the device.

   ```
   reboot
   ```

   For the new role to take effect, you must reboot the device.

Configuring the device as a master spine node

About this task

If multiple spine nodes exist on a VCF fabric, you must configure a device as the master spine node to collect the topology for the entire VCF fabric network.

Procedure

1. Enter system view.

   ```
   system-view
   ```

2. Configure the device as a master spine node.

   ```
   vcf-fabric spine-role master
   ```

   By default, the device is not a master spine node.

Pausing automated underlay network deployment

About this task

If you pause automated underlay network deployment, the VCF fabric will save the current status of the device. It will not respond to new LLDP events, set up the IRF fabric, aggregate links, or discover uplink or downlink interfaces.

Perform this task if all devices in the VCF fabric complete automated deployment and new devices are to be added to the VCF fabric.

Procedure

1. Enter system view.

   ```
   system-view
   ```

2. Pause automated underlay network deployment.

   ```
   vcf-fabric underlay pause
   ```

   By default, automated underlay network deployment is not paused.

Configuring automated overlay network deployment

Restrictions and guidelines for automated overlay network deployment

If the network type is VLAN or VXLAN with a centralized IP gateway, perform this task on both the spine node and the leaf nodes.

If the network type is VXLAN with distributed IP gateways, perform this task on leaf nodes.

As a best practice, do not perform any of the following tasks while the device is communicating with a RabbitMQ server:

- Change the source IPv4 address for the device to communicate with RabbitMQ servers.
- Bring up or shut down a port connected to the RabbitMQ server.

If you do so, it will take the CLI a long time to respond to the **`l2agent enable`**, **`undo l2agent enable`**, **`l3agent enable`**, or **`undo l3agent enable`** command.

Automated overlay network deployment is not supported on aggregate or access nodes.

Automated overlay network deployment tasks at a glance

To configure automated overlay network deployment, perform the following tasks:

1. Configuring parameters for the device to communicate with RabbitMQ servers
2. Specifying the network type
3. Enabling L2 agent

   On a VLAN network or a VXLAN network with a centralized IP gateway, perform this task on both spine nodes and leaf nodes.

   On a VXLAN network with distribute IP gateways, perform this task only on leaf nodes.
4. Enabling L3 agent

   On a VLAN network or a VXLAN network with a centralized IP gateway, perform this task only on spine nodes.

   On a VXLAN network with distribute IP gateways, perform this task only on leaf nodes.
5. Configuring the border node

   Perform this task only when the device is the border node.
6. (Optional.) Enabling local proxy ARP
7. (Optional.) Configuring the MAC address of VSI interfaces

Prerequisites for automated overlay network deployment

Before you configure automated overlay network deployment, you must complete the following tasks:

1. Install OpenStack Neutron components and plugins on the controller node in the VCF fabric.
2. Install OpenStack Nova components, openvswitch, and neutron-ovs-agent on compute nodes in the VCF fabric.
3. Make sure LLDP and automated VCF fabric topology discovery are enabled.

Configuring parameters for the device to communicate with RabbitMQ servers

About this task

In the VCF fabric, the device communicates with the Neutron server through RabbitMQ servers. You must specify the IP address, login username, login password, and listening port for the device to communicate with RabbitMQ servers.

Restrictions and guidelines

Make sure the RabbitMQ server settings on the device are the same as those on the controller node. If the durable attribute of RabbitMQ queues is set on the Neutron server, you must enable creation of RabbitMQ durable queues on the device so that RabbitMQ queues can be correctly created.

When you set the RabbitMQ server parameters or remove the settings, make sure the routes between the device and the RabbitMQ server is reachable. Otherwise, the CLI does not respond until the TCP connection between the device and the RabbitMQ server is terminated.

Multiple virtual hosts might exist on the RabbitMQ server. Each virtual host can independently provide RabbitMQ services for the device. For the device to correctly communicate with the Neutron server, specify the same virtual host on the device and the Neutron server.

Procedure

1. Enter system view.

   system-view

2. Enable Neutron and enter Neutron view.

   neutron

   By default, Neutron is disabled.

3. Specify the IPv4 address, port number, and MPLS L3VPN instance of a RabbitMQ server.

   rabbit host ip *ipv4-address* [**port** *port-number*] [**vpn-instance** *vpn-instance-name*]

   By default, no IPv4 address or MPLS L3VPN instance of a RabbitMQ server is specified, and the port number of a RabbitMQ server is 5672.

4. Specify the source IPv4 address for the device to communicate with RabbitMQ servers.

   rabbit source-ip *ipv4-address* [**vpn-instance** *vpn-instance-name*]

   By default, no source IPv4 address is specified for the device to communicate with RabbitMQ servers. The device automatically selects a source IPv4 address through the routing protocol to communicate with RabbitMQ servers.

5. (Optional.) Enable creation of RabbitMQ durable queues.

   rabbit durable-queue enable

   By default, RabbitMQ non-durable queues are created.

6. Configure the username for the device to establish a connection with a RabbitMQ server.

   rabbit user *username*

   By default, the device uses username **guest** to establish a connection with a RabbitMQ server.

7. Configure the password for the device to establish a connection with a RabbitMQ server.

   rabbit password { **cipher** | **plain** } *string*

   By default, the device uses plaintext password **guest** to establish a connection with a RabbitMQ server.

8. Specify a virtual host to provide RabbitMQ services.

   rabbit virtual-host *hostname*

   By default, the virtual host **/** provides RabbitMQ services for the device.

9. Specify the username and password for the device to deploy configurations through RESTful.

   restful user *username* **password** { **cipher** | **plain** } *password*

   By default, no username or password is configured for the device to deploy configurations through RESTful.

Specifying the network type

About this task

After you change the network type of the VCF fabric where the device resides, Neutron deploys new configuration to all devices according to the new network type.

Procedure

1. Enter system view.
   system-view
2. Enter Neutron view.
   neutron
3. Specify the network type.
   network-type { **centralized-vxlan**| **distributed-vxlan**|**vlan** }
   By default, the network type is VLAN.

Enabling L2 agent

About this task

Layer 2 agent (L2 agent) responds to OpenStack events such as network creation, subnet creation, and port creation. It deploys Layer 2 networking to provide Layer 2 connectivity within a virtual network and Layer 2 isolation between different virtual networks

Restrictions and guidelines

On a VLAN network or a VXLAN network with a centralized IP gateway, perform this task on both spine nodes and leaf nodes.

On a VXLAN network with distribute IP gateways, perform this task only on leaf nodes.

Procedure

1. Enter system view.
   system-view
2. Enter Neutron view.
   neutron
3. Enable the L2 agent.
   l2agent enable
   By default, the L2 agent is disabled.

Enabling L3 agent

About this task

Layer 3 agent (L3 agent) responds to OpenStack events such as virtual router creation, interface creation, and gateway configuration. It deploys the IP gateways to provide Layer 3 forwarding services for VMs.

Restrictions and guidelines

On a VLAN network or a VXLAN network with a centralized IP gateway, perform this task only on spine nodes.

On a VXLAN network with distribute IP gateways, perform this task only on leaf nodes.

Procedure

1. Enter system view.
   system-view
2. Enter Neutron view.
   neutron
3. Enable the L3 agent.

L3agent enable

By default, the L3 agent is disabled.

Configuring the border node

About this task

On a VXLAN network with a centralized IP gateway or on a VLAN network, configure a spine node as the border node. On a VXLAN network with distributed IP gateways, configure a leaf node as the border.

You can use the following methods to configure the IP address of the border gateway:

- Manually specify the IP address of the border gateway.
- Enable the border node service on the border gateway and create the external network and routers on the OpenStack Dashboard. Then, VCF fabric automatically deploys the routing configuration to the device to implement connectivity between tenant networks and the external network.

If the manually specified IP address is different from the IP address assigned by VCF fabric, the IP address assigned by VCF fabric takes effect.

The border node connects to the external network through an interface which belongs to the global VPN instance. For the traffic from the external network to reach a tenant network, the border node needs to add the routes of the tenant VPN instance into the routing table of the global VPN instance. You must configure export route targets of the tenant VPN instance as import route targets of the global VPN instance. This setting enables the global VPN instance to import routes of the tenant VPN instance.

Procedure

1. Enter system view.

   system-view
2. Enter Neutron view.

   neutron
3. Enable the border node service.

   border enable

   By default, the device is not a border node.
4. (Optional.) Specify the IPv4 address of the border gateway.

   gateway ip *ipv4-address*

   By default, the IPv4 address of the border gateway is not specified.
5. Configure export route targets for a tenant VPN instance.

   vpn-target *target* **export-extcommunity**

   By default, no export route targets are configured for a tenant VPN instance.
6. (Optional.) Configure import route targets for a tenant VPN instance.

   vpn-target *target* **import-extcommunity**

   By default, no import route targets are configured for a tenant VPN instance.

Enabling local proxy ARP

About this task

This feature enables the device to use the MAC address of VSI interfaces to answer ARP requests for MAC addresses of VMs on a different site from the requesting VMs.

Restrictions and guidelines

Perform this task only on leaf nodes on a VXLAN network with distributed IP gateways.

This configuration takes effect on VSI interfaces that are created after the **proxy-arp enable** command is executed. It does not take effect on existing VSI interfaces.

Procedure

1. Enter system view.
   system-view
2. Enter Neutron view.
   neutron
3. Enable local proxy ARP.
   proxy-arp enable
   By default, local proxy ARP is disabled.

Configuring the MAC address of VSI interfaces

About this task

After you perform this task, VCF fabric assigns the MAC address to all VSI interfaces newly created by automated overlay network deployment on the device.

Restrictions and guidelines

Perform this task only on leaf nodes on a VXLAN network with distributed IP gateways.

This configuration takes effect only on VSI interfaces newly created after this command is executed.

Procedure

1. Enter system view.
   system-view
2. Enter Neutron view.
   neutron
3. Configure the MAC address of VSI interfaces.
   vsi-mac *mac-address*
   By default, no MAC address is configured for VSI interfaces.

Display and maintenance commands for VCF fabric

Execute **display** commands in any view.

Task	Command
Display the role of the device in the VCF fabric.	**display vcf-fabric role**
Display VCF fabric topology information.	**display vcf-fabric topology**
Display information about automated underlay network deployment.	**display vcf-fabric underlay autoconfigure**

Task	Command
Display the supported version and the current version of the template file for automated VCF fabric provisioning.	**display vcf-fabric underlay template-version**

Using Ansible for automated configuration management

About Ansible

Ansible is a configuration tool programmed in Python. It uses SSH to connect to devices.

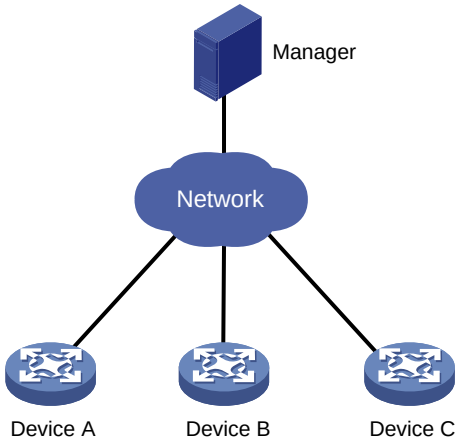
Ansible network architecture

As shown in Figure 127, an Ansible system consists of the following elements:

- **Manager**—A host installed with the Ansible environment. For more information about the Ansible environment, see Ansible documentation.
- **Managed devices**—Devices to be managed. These devices do not need to install any agent software. They only need to be able to act as an SSH server. The manager communicates with managed devices through SSH to deploy configuration files.

HPE devices can act as managed devices.

Figure 127 Ansible network architecture



How Ansible works

The following the steps describe how Ansible works:

1. On the manager, create a configuration file and specify the destination device.
2. The manager (SSH client) initiates an SSH connection to the device (SSH server).
3. The manager deploys the configuration file to the device.
4. After receiving a configuration file from the manager, the device loads the configuration file.

Restrictions and guidelines

Not all services modules are configurable through Ansible. To identify the service modules that you can configure by using Ansible, access the Comware 7 Python library.

Configuring the device for management with Ansible

Before you use Ansible to configure the device, complete the following tasks:

- Configure a time protocol (NTP or PTP) or manually configure the system time on the Ansible server and the device to synchronize their system time. For more information about NTP and PTP configuration, see *Network Management and Monitoring Configuration Guide*.
- Configure the device as an SSH server. For more information about SSH configuration, see *Security Configuration Guide*.

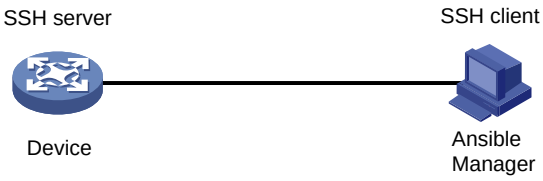
Device setup examples for management with Ansible

Example: Setting up the device for management with Ansible

Network configuration

As shown in Figure 128, enable SSH server on the device and use the Ansible manager to manage the device over SSH.

Figure 128 Network diagram



Prerequisites

Assign IP addresses to the device and manager so you can access the device from the manager. (Details not shown.)

Procedure

1. Configure a time protocol (NTP or PTP) or manually configure the system time on both the device and manager so they use the same system time. (Details not shown.)
2. Configure the device as an SSH server:

   # Create local key pairs. (Details not shown.)

   # Create a local user named **abc** and set the password to **123456** in plain text.

   ```
   <Device> system-view
   [Device]local-user abc
   [Device-luser-manage-abc] password simple 123456
   ```

   # Assign the network-admin user role to the user and authorize the user to use SSH, HTTP, and HTTPS services.

   ```
   [Device-luser-manage-abc] authorization-attribute user-role network-admin
   [Device-luser-manage-abc] service-type ssh http https
   [Device-luser-manage-abc] quit
   ```

   # Enable NETCONF over SSH.

   ```
   [Device] netconf ssh server enable
   ```

Enable scheme authentication for SSH login and assign the network-admin user role to the login users.

```
[Device] line vty 0 63
[Device-line-vty0-63] authentication-mode scheme
[Device-line-vty0-63] user-role network-admin
[Device-line-vty0-63] quit
```

Enable the SSH server.

```
[Device] ssh server enable
```

Authorize SSH user abc to use all service types, including SCP, SFTP, Stelnet, and NETCONF. Set the authentication method to password.

```
[Device] ssh user abc service-type all authentication-type password
```

Enable the SFTP server or SCP server.

- If the device supports SFTP, enable the SFTP server.

  ```
  [Device] sftp server enable
  ```

- If the device does not support SFTP, enable the SCP server.

  ```
  [Device] scp server enable
  ```

Procedure

Install Ansible on the manager. Create a configuration script and deploy the script. For more information, see the relevant documents.

Document conventions and icons

Conventions

This section describes the conventions used in the documentation.

Command conventions

Convention	Description
Boldface	**Bold** text represents commands and keywords that you enter literally as shown.
Italic	*Italic* text represents arguments that you replace with actual values.
[ ]	Square brackets enclose syntax choices (keywords or arguments) that are optional.
{ x | y | ... }	Braces enclose a set of required syntax choices separated by vertical bars, from which you select one.
[x | y | ...]	Square brackets enclose a set of optional syntax choices separated by vertical bars, from which you select one or none.
{ x | y | ... } *	Asterisk marked braces enclose a set of required syntax choices separated by vertical bars, from which you select at least one.
[x | y | ...] *	Asterisk marked square brackets enclose optional syntax choices separated by vertical bars, from which you select one choice, multiple choices, or none.
&<1-n>	The argument or keyword and argument combination before the ampersand (&) sign can be entered 1 to n times.
#	A line that starts with a pound (#) sign is comments.

GUI conventions

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

Symbols

Convention	Description
⚠ **WARNING!**	An alert that calls attention to important information that if not understood or followed can result in personal injury.
△ **CAUTION:**	An alert that calls attention to important information that if not understood or followed can result in data loss, data corruption, or damage to hardware or software.
ⓘ **IMPORTANT:**	An alert that calls attention to essential information.
NOTE:	An alert that contains additional or supplementary information.
TIP:	An alert that provides helpful information.

Network topology icons

Convention	Description
	Represents a generic network device, such as a router, switch, or firewall.
	Represents a routing-capable device, such as a router or Layer 3 switch.
	Represents a generic switch, such as a Layer 2 or Layer 3 switch, or a router that supports Layer 2 forwarding and other Layer 2 features.
	Represents an access controller, a unified wired-WLAN module, or the access controller engine on a unified wired-WLAN switch.
	Represents an access point.
	Represents a wireless terminator unit.
	Represents a wireless terminator.
	Represents a mesh access point.
	Represents omnidirectional signals.
	Represents directional signals.
	Represents a security product, such as a firewall, UTM, multiservice security gateway, or load balancing device.
	Represents a security module, such as a firewall, load balancing, NetStream, SSL VPN, IPS, or ACG module.

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.

Support and other resources

Accessing Hewlett Packard Enterprise Support

- For live assistance, go to the Contact Hewlett Packard Enterprise Worldwide website:

  www.hpe.com/assistance
- To access documentation and support services, go to the Hewlett Packard Enterprise Support Center website:

  www.hpe.com/support/hpesc

Information to collect

- Technical support registration number (if applicable)
- Product name, model or version, and serial number
- Operating system name and version
- Firmware version
- Error messages
- Product-specific reports and logs
- Add-on products or components
- Third-party products or components

Accessing updates

- Some software products provide a mechanism for accessing software updates through the product interface. Review your product documentation to identify the recommended software update method.
- To download product updates, go to either of the following:
  - Hewlett Packard Enterprise Support Center **Get connected with updates** page:

    www.hpe.com/support/e-updates
  - Software Depot website:

    www.hpe.com/support/softwaredepot
- To view and update your entitlements, and to link your contracts, Care Packs, and warranties with your profile, go to the Hewlett Packard Enterprise Support Center **More Information on Access to Support Materials** page:

  www.hpe.com/support/AccessToSupportMaterials

IMPORTANT:

Access to some updates might require product entitlement when accessed through the Hewlett Packard Enterprise Support Center. You must have an HP Passport set up with relevant entitlements.

Websites

Website	Link
Networking websites	
Hewlett Packard Enterprise Information Library for Networking	www.hpe.com/networking/resourcefinder
Hewlett Packard Enterprise Networking website	www.hpe.com/info/networking
Hewlett Packard Enterprise My Networking website	www.hpe.com/networking/support
Hewlett Packard Enterprise My Networking Portal	www.hpe.com/networking/mynetworking
Hewlett Packard Enterprise Networking Warranty	www.hpe.com/networking/warranty
General websites	
Hewlett Packard Enterprise Information Library	www.hpe.com/info/enterprise/docs
Hewlett Packard Enterprise Support Center	www.hpe.com/support/hpesc
Hewlett Packard Enterprise Support Services Central	ssc.hpe.com/portal/site/ssc/
Contact Hewlett Packard Enterprise Worldwide	www.hpe.com/assistance
Subscription Service/Support Alerts	www.hpe.com/support/e-updates
Software Depot	www.hpe.com/support/softwaredepot
Customer Self Repair (not applicable to all devices)	www.hpe.com/support/selfrepair
Insight Remote Support (not applicable to all devices)	www.hpe.com/info/insightremotesupport/docs

Customer self repair

Hewlett Packard Enterprise customer self repair (CSR) programs allow you to repair your product. If a CSR part needs to be replaced, it will be shipped directly to you so that you can install it at your convenience. Some parts do not qualify for CSR. Your Hewlett Packard Enterprise authorized service provider will determine whether a repair can be accomplished by CSR.

For more information about CSR, contact your local service provider or go to the CSR website:

www.hpe.com/support/selfrepair

Remote support

Remote support is available with supported devices as part of your warranty, Care Pack Service, or contractual support agreement. It provides intelligent event diagnosis, and automatic, secure submission of hardware event notifications to Hewlett Packard Enterprise, which will initiate a fast and accurate resolution based on your product's service level. Hewlett Packard Enterprise strongly recommends that you register your device for remote support.

For more information and device support details, go to the following website:

www.hpe.com/info/insightremotesupport/docs

Documentation feedback

Hewlett Packard Enterprise is committed to providing documentation that meets your needs. To help us improve the documentation, send any errors, suggestions, or comments to Documentation Feedback (docsfeedback@hpe.com). When submitting your feedback, include the document title,

part number, edition, and publication date located on the front cover of the document. For online help content, include the product name, product version, help edition, and publication date located on the legal notices page.

Index

A

access control
- SNMP MIB, 170
- SNMP view-based MIB, 170

accessing
- NTP access control, 82
- SNMP access control mode, 171

accounting
- IPv6 NetStream configuration, 386, 396

ACS
- CWMP ACS-CPE autoconnect, 297

action
- Event MIB notification, 196
- Event MIB set, 196

address
- ping address reachability determination, 2

agent
- sFlow agent+collector information configuration, 401

aggregating
- IPv6 NetStream data export, 394
- IPv6 NetStream data export (aggregation), 388, 397
- NetStream aggregation data export, 372, 379
- NetStream data export configuration (aggregation), 383

aggregation group
- Chef resources (netdev_lagg), 290
- Puppet resources (netdev_lagg), 274

aging
- IPv6 NetStream flow, 387
- IPv6 NetStream flow aging, 393
- NetStream flow aging, 371, 378
- NetStream flow aging configuration (forced), 379
- NetStream flow aging configuration (periodic), 378

alarm
- RMON alarm configuration, 189, 192
- RMON alarm group sample types, 188
- RMON configuration, 186, 191
- RMON group, 187
- RMON private group, 187

announcing
- PTP announce message interval+timeout, 138

applying
- flow mirroring QoS policy, 366
- flow mirroring QoS policy (control plane), 368
- flow mirroring QoS policy (global), 367
- flow mirroring QoS policy (interface), 366
- flow mirroring QoS policy (VLAN), 367

architecture
- IPv6 NetStream, 386
- NetStream, 370
- NTP, 80

arithmetic
- packet capture filter configuration (expr relop expr expression), 436
- packet capture filter configuration (proto [exprsize] expression), 436
- packet capture filter operator, 434
- packet capture operator, 432

assigning
- CWMP ACS attribute (preferred)(CLI), 301
- CWMP ACS attribute (preferred)(DHCP server), 300
- port mirroring monitor port to remote probe VLAN, 344

associating
- IPv6 NTP client/server association mode, 99
- IPv6 NTP multicast association mode, 108
- IPv6 NTP symmetric active/passive association mode, 102
- NTP association mode, 85
- NTP broadcast association mode, 81, 86, 103
- NTP broadcast association mode+authentication, 112
- NTP client/server association mode, 81, 85, 98
- NTP client/server association mode+authentication, 111
- NTP client/server mode+MPLS L3VPN network time synchronization, 115
- NTP multicast association mode, 81, 87, 105
- NTP symmetric active/passive association mode, 81, 86, 100
- NTP symmetric active/passive mode+MPLS L3VPN network time synchronization, 116

attribute
- NETCONF session attribute, 215
- NetStream data export format, 376

authenticating
- CWMP CPE ACS authentication, 302
- NTP, 83
- NTP broadcast authentication, 92
- NTP broadcast mode+authentication, 112
- NTP client/server mode authentication, 89

NTP client/server mode+authentication, 111
NTP configuration, 89
NTP multicast authentication, 93
NTP security, 82
NTP symmetric active/passive mode authentication, 90
SNTP authentication, 120
auto
CWMP ACS-CPE autoconnect, 297
VCF fabric automated deployment, 452
VCF fabric automated deployment process, 453
VCF fabric automated underlay network deployment configuration, 456, 457
autoconfiguration server (ACS)
CWMP, 295
CWMP ACS authentication parameters, 302
CWMP attribute configuration, 300
CWMP attribute type (default)(CLI), 301
CWMP attributes (preferred), 300
CWMP autoconnect parameters, 304
CWMP CPE ACS provision code, 303
CWMP CPE connection interface, 303
HTTPS SSL client policy, 302
automated overlay network deployment
border node configuration, 461
L2 agent, 460
L3 agent, 460
local proxy ARP, 461
MAC address of VSI interfaces, 462
network type specifying, 459
RabbiMQ server communication parameters, 458
automated underlay network deployment
pausing deployment, 457
automtaed underlay network deploying
template file, 453

B

bidirectional
port mirroring, 335
Boolean
Event MIB trigger test, 195
Event MIB trigger test configuration, 206
booting
GOLD configuration, 427, 430
GOLD configuration (centralized IRF devices), 430
boundary
PTP clock node (BC), 124
broadcast
NTP association mode, 103
NTP broadcast association mode, 81, 86, 92
NTP broadcast association mode+authentication, 112
NTP broadcast mode dynamic associations max, 96
buffer
GOLD log buffer size, 429
buffering
information center log storage period (log buffer), 417
building
packet capture display filter, 436, 439
packet capture filter, 433, 435

C

capturing
packet capture configuration, 432, 442
packet capture configuration (feature image-based), 443
remote packet capture configuration, 442
Chef
client configuration, 283
configuration, 280, 284, 284
configuration file, 281
network framework, 280
resources, 281, 287
resources (netdev_device), 287
resources (netdev_interface), 287
resources (netdev_l2_interface), 289
resources (netdev_lagg), 290
resources (netdev_vlan), 291
resources (netdev_vsi), 291
resources (netdev_vte), 292
resources (netdev_vxlan), 293
server configuration, 283
shutdown, 284
start, 283
workstation configuration, 283
classifying
port mirroring classification, 336
CLI
EAA configuration, 314, 321
EAA event monitor policy configuration, 322
EAA monitor policy configuration (CLI-defined+environment variables), 325
NETCONF CLI operations, 247, 248
NETCONF return to CLI, 255
client
Chef client configuration, 283
NQA client history record save, 30

NQA client operation (DHCP), 12
NQA client operation (DLSw), 24
NQA client operation (DNS), 13
NQA client operation (FTP), 14
NQA client operation (HTTP), 15
NQA client operation (ICMP echo), 10
NQA client operation (ICMP jitter), 11
NQA client operation (path jitter), 24
NQA client operation (SNMP), 18
NQA client operation (TCP), 18
NQA client operation (UDP echo), 19
NQA client operation (UDP jitter), 16
NQA client operation (UDP tracert), 20
NQA client operation (voice), 22
NQA client operation scheduling, 31
NQA client statistics collection, 29
NQA client template, 31
NQA client template (DNS), 33
NQA client template (FTP), 41
NQA client template (HTTP), 38
NQA client template (HTTPS), 39
NQA client template (ICMP), 32
NQA client template (RADIUS), 42
NQA client template (SSL), 43
NQA client template (TCP half open), 35
NQA client template (TCP), 34
NQA client template (UDP), 36
NQA client template optional parameters, 44
NQA client threshold monitoring, 8, 27
NQA client+Track collaboration, 27
NQA collaboration configuration, 68
NQA enable, 9
NQA operation, 9
NQA operation configuration (DHCP), 50
NQA operation configuration (DLSw), 65
NQA operation configuration (DNS), 51
NQA operation configuration (FTP), 52
NQA operation configuration (HTTP), 53
NQA operation configuration (ICMP echo), 46
NQA operation configuration (ICMP jitter), 47
NQA operation configuration (path jitter), 66
NQA operation configuration (SNMP), 57
NQA operation configuration (TCP), 58
NQA operation configuration (UDP echo), 59
NQA operation configuration (UDP jitter), 54
NQA operation configuration (UDP tracert), 61
NQA operation configuration (voice), 62
SNTP configuration, 84, 119, 122, 122
client/server
IPv6 NTP client/server association mode, 99
NTP association mode, 81, 85
NTP client/server association mode, 89, 98
NTP client/server association mode+authentication, 111
NTP client/server mode dynamic associations max, 96
NTP client/server mode+MPLS L3VPN network time synchronization, 115
clock
NTP local clock as reference source, 88
PTP clock node (BC), 124
PTP clock node (hybrid), 124
PTP clock node (OC), 124
PTP clock node (TC), 124
PTP clock node type, 134
PTP clock priority, 145
PTP clock type, 126
PTP grandmaster clock, 126
PTP OC configuration as member clock, 134
PTP system time source, 133
close-wait timer (CWMP ACS), 305
collaborating
NQA client+Track function, 27
NQA+Track collaboration, 7
collecting
IPv6 NetStream collector (NSC), 386, 386
sFlow agent+collector information configuration, 401
troubleshooting sFlow remote collector cannot receive packets, 404
common
information center standard system logs, 406
community
SNMPv1 community direct configuration, 174
SNMPv1 community indirect configuration, 175
SNMPv1 configuration, 174, 174
SNMPv2c community direct configuration by community name, 174
SNMPv2c community indirect configuration by creating SNMPv2c user, 175
SNMPv2c configuration, 174, 174
comparing
packet capture display filter operator, 438
packet capture filter operator, 434
conditional match
NETCONF data filtering, 234
NETCONF data filtering (column-based), 232
configuration
NETCONF configuration modification, 238
configuration file
Chef configuration file, 281

configuration management
Chef configuration, 280, 284, 284
Puppet configuration, 267, 270, 270
configure
RabbiMQ server communication parameters, 458
VCF fabric overlay network border node, 461
configuring
Chef, 280, 284, 284
Chef client, 283
Chef server, 283
Chef workstation, 283
CWMP, 295, 299, 305
CWMP ACS attribute, 300
CWMP ACS attribute (default)(CLI), 301
CWMP ACS attribute (preferred), 300
CWMP ACS autoconnect parameters, 304
CWMP ACS close-wait timer, 305
CWMP ACS connection retry max number, 304
CWMP ACS periodic Inform feature, 304
CWMP CPE ACS authentication parameters, 302
CWMP CPE ACS connection interface, 303
CWMP CPE ACS provision code, 303
CWMP CPE attribute, 302
EAA, 314, 321
EAA environment variable (user-defined), 317
EAA event monitor policy (CLI), 322
EAA event monitor policy (Track), 323
EAA monitor policy, 318
EAA monitor policy (CLI-defined+environment variables), 325
EAA monitor policy (Tcl-defined), 321
Event MIB, 195, 197, 204
Event MIB event, 198
Event MIB trigger test, 200
Event MIB trigger test (Boolean), 206
Event MIB trigger test (existence), 204
Event MIB trigger test (threshold), 202, 209
feature image-based packet capture, 440
flow mirroring, 364, 368
flow mirroring traffic behavior, 365
flow mirroring traffic class, 365
GOLD, 427, 430
GOLD (centralized IRF devices), 430
GOLD diagnostic test simulation, 429
GOLD diagnostics (monitoring), 427
GOLD diagnostics (on-demand), 428
GOLD log buffer size, 429
information center, 406, 411, 423
information center log output (console), 423
information center log output (Linux log host), 425
information center log output (UNIX log host), 424
information center log suppression, 418
information center log suppression for module, 419
information center trace log file max size, 422
IPv6 NetStream, 386, 390, 396
IPv6 NetStream data export, 394
IPv6 NetStream data export (aggregation), 394, 397
IPv6 NetStream data export (traditional), 394, 396
IPv6 NetStream data export format, 391
IPv6 NetStream filtering, 390
IPv6 NetStream flow aging, 393
IPv6 NetStream flow aging (periodic), 393
IPv6 NetStream sampling, 391
IPv6 NetStream v9/v10 template refresh rate, 393
IPv6 NTP client/server association mode, 99
IPv6 NTP multicast association mode, 108
IPv6 NTP symmetric active/passive association mode, 102
Layer 2 remote port mirroring, 341
Layer 2 remote port mirroring (egress port), 357
Layer 2 remote port mirroring (reflector port configurable), 355
Layer 3 remote port mirroring, 359
Layer 3 remote port mirroring (in ERSPAN mode), 350, 361
Layer 3 remote port mirroring (in tunnel mode), 347
Layer 3 remote port mirroring local group, 348
Layer 3 remote port mirroring local group monitor port, 349, 351
Layer 3 remote port mirroring local group source CPU, 349, 351
Layer 3 remote port mirroring local group source ports, 351
local packet capture (wired device), 440
local port mirroring, 339
local port mirroring (source CPU mode), 353
local port mirroring (source port mode), 352
local port mirroring group monitor port, 341
local port mirroring group source CPU, 340
local port mirroring group source ports, 340
mirroring sources, 340, 348, 350
NETCONF, 212, 214
NetStream, 370, 375, 381
NetStream data export, 379
NetStream data export (aggregation), 379, 383
NetStream data export (traditional), 379, 381
NetStream data export format, 376

NetStream filtering, 375
NetStream flow aging, 378
NetStream flow aging (forced), 379, 393
NetStream flow aging (periodic), 378
NetStream sampling, 376
NetStream v9/v10 template refresh rate, 378
NQA, 7, 8, 46
NQA client history record save, 30
NQA client operation, 9
NQA client operation (DHCP), 12
NQA client operation (DLSw), 24
NQA client operation (DNS), 13
NQA client operation (FTP), 14
NQA client operation (HTTP), 15
NQA client operation (ICMP echo), 10
NQA client operation (ICMP jitter), 11
NQA client operation (path jitter), 24
NQA client operation (SNMP), 18
NQA client operation (TCP), 18
NQA client operation (UDP echo), 19
NQA client operation (UDP jitter), 16
NQA client operation (UDP tracert), 20
NQA client operation (voice), 22
NQA client operation optional parameters, 26
NQA client statistics collection, 29
NQA client template, 31
NQA client template (DNS), 33
NQA client template (FTP), 41
NQA client template (HTTP), 38
NQA client template (HTTPS), 39
NQA client template (ICMP), 32
NQA client template (RADIUS), 42
NQA client template (SSL), 43
NQA client template (TCP half open), 35
NQA client template (TCP), 34
NQA client template (UDP), 36
NQA client template optional parameters, 44
NQA client threshold monitoring, 27
NQA client+Track collaboration, 27
NQA collaboration, 68
NQA operation (DHCP), 50
NQA operation (DLSw), 65
NQA operation (DNS), 51
NQA operation (FTP), 52
NQA operation (HTTP), 53
NQA operation (ICMP echo), 46
NQA operation (ICMP jitter), 47
NQA operation (path jitter), 66
NQA operation (SNMP), 57
NQA operation (TCP), 58
NQA operation (UDP echo), 59
NQA operation (UDP jitter), 54
NQA operation (UDP tracert), 61
NQA operation (voice), 62
NQA server, 9
NQA template (DNS), 71
NQA template (FTP), 75
NQA template (HTTP), 74
NQA template (HTTPS), 75
NQA template (ICMP), 70
NQA template (RADIUS), 76
NQA template (SSL), 77
NQA template (TCP half open), 72
NQA template (TCP), 72
NQA template (UDP), 73
NTP, 79, 84, 98
NTP association mode, 85
NTP broadcast association mode, 86, 103
NTP broadcast mode authentication, 92
NTP broadcast mode+authentication, 112
NTP client/server association mode, 85, 98
NTP client/server mode authentication, 89
NTP client/server mode+authentication, 111
NTP client/server mode+MPLS L3VPN network time synchronization, 115
NTP dynamic associations max, 96
NTP local clock as reference source, 88
NTP multicast association mode, 87, 105
NTP multicast mode authentication, 93
NTP optional parameters, 95
NTP symmetric active/passive association mode, 86, 100
NTP symmetric active/passive mode authentication, 90
NTP symmetric active/passive mode+MPLS L3VPN network time synchronization, 116
packet capture, 432, 442
packet capture (feature image-based), 443
PMM kernel thread deadloop detection, 330
PMM kernel thread starvation detection, 331
port mirroring, 352
port mirroring remote destination group monitor port, 343
port mirroring remote probe VLAN, 343
PTP, 124, 147
PTP (AES67-2015, IPv4 UDP transport, multicast transmission), 166
PTP (IEEE 1588 v2, IEEE 802.3/Ethernet transport, multicast transmission), 147
PTP (IEEE 1588 v2, IPv4 UDP transport, multicast transmission), 150

PTP (IEEE 1588 v2, IPv4 UDP transport, unicast transmission), 153
PTP (IEEE 802.1AS, IEEE 802.3/Ethernet transport, multicast transmission), 156
PTP (SMPTE ST 2059-2, IPv4 UDP transport, multicast transmission), 159
PTP (SMPTE ST 2059-2, IPv4 UDP transport, unicast transmission), 163
PTP clock priority, 145
PTP multicast message source IP address (UDP), 141
PTP non-Pdelay message MAC address, 142
PTP OC as member clock, 134
PTP OC-type port on a TC+OC clock, 138
PTP port role, 136
PTP system time source, 133
PTP timestamp carry mode, 137
PTP unicast message destination IP address (IPv4 UDP), 142
PTP UTC correction date, 145
Puppet, 267, 270, 270
remote packet capture, 442
remote packet capture (wired device), 440
remote port mirroring source group egress port, 346
remote port mirroring source group reflector port, 345
remote port mirroring source group source CPU, 345
remote port mirroring source group source ports, 344
RMON, 186, 191
RMON alarm, 189, 192
RMON Ethernet statistics group, 191
RMON history group, 191
RMON statistics, 188
sampler, 333
sampler (IPv4 NetStream), 333
sFlow, 400, 403, 403
sFlow agent+collector information, 401
sFlow counter sampling, 402
sFlow flow sampling, 401
SNMP, 170, 181
SNMP common parameters, 173
SNMP logging, 180
SNMP notification, 177
SNMPv1, 181
SNMPv1 community, 174, 174
SNMPv1 community by community name, 174
SNMPv1 community by creating SNMPv1 user, 175
SNMPv1 host notification send, 178
SNMPv2c, 181
SNMPv2c community, 174, 174
SNMPv2c community by community name, 174
SNMPv2c community by creating SNMPv2c user, 175
SNMPv2c host notification send, 178
SNMPv3, 183
SNMPv3 group and user, 175
SNMPv3 group and user in FIPS mode, 176
SNMPv3 group and user in non-FIPS mode, 176
SNMPv3 host notification send, 178
SNTP, 84, 119, 122, 122
SNTP authentication, 120
VCF fabric, 447, 454
VCF fabric automated underlay network deployment, 456, 457
VCF fabric MAC address of VSI interfaces, 462
VXLAN-aware NetStream, 378
connecting
CWMP ACS connection initiation, 304
CWMP ACS connection retry max number, 304
CWMP CPE ACS connection interface, 303
console
information center log output, 413
information center log output configuration, 423
NETCONF over console session establishment, 218
content
packet file content display, 441
control plane
flow mirroring QoS policy application, 368
controlling
RMON history control entry, 188
converging
VCF fabric configuration, 447, 454
cookbook
Chef resources, 281
correcting
PTP delay correction value, 144
CPE
CWMP ACS-CPE autoconnect, 297
CPU
flow mirroring configuration, 364, 368
Layer 3 remote port mirroring local group source CPU, 349, 351
local port mirroring (source CPU mode), 353
creating
Layer 3 remote port mirroring local group, 350
local port mirroring group, 340
remote port mirroring destination group, 342
remote port mirroring source group, 344

RMON Ethernet statistics entry, 188
RMON history control entry, 188
sampler, 333
cumulative offset (UTC\:TAI), 144
customer premise equipment (CPE)
CPE WAN Management Protocol. *Use* CWMP
CWMP
ACS attribute (default)(CLI), 301
ACS attribute (preferred), 300
ACS attribute configuration, 300
ACS autoconnect parameters, 304
ACS HTTPS SSL client policy, 302
ACS-CPE autoconnect, 297
autoconfiguration server (ACS), 295
basic functions, 295
configuration, 295, 299, 305
connection establishment, 298
CPE ACS authentication parameters, 302
CPE ACS connection interface, 303
CPE ACS provision code, 303
CPE attribute configuration, 302
customer premise equpment (CPE), 295
DHCP server, 295
DNS server, 295
enable, 300
how it works, 297
main/backup ACS switchover, 298
network framework, 295
RPC methods, 297
settings display, 305

D

data
feature image-based packet capture data display filter, 441, 441
IPv6 NetStream analyzer (NDA), 386
IPv6 NetStream data export, 394
IPv6 NetStream data export (aggregation), 388, 394, 397
IPv6 NetStream data export (traditional), 388, 394, 396
IPv6 NetStream export format, 388
IPv6 NetStream exporter (NDE), 386
NETCONF configuration data retrieval (all modules), 226
NETCONF configuration data retrieval (Syslog module), 227
NETCONF data entry retrieval (interface table), 225
NETCONF filtering (column-based), 230
NETCONF filtering (column-based) (conditional match), 232
NETCONF filtering (column-based) (full match), 230
NETCONF filtering (column-based) (regex match), 231
NETCONF filtering (conditional match), 234
NETCONF filtering (regex match), 233
NETCONF filtering (table-based), 230
NetStream data export, 372, 379
NetStream data export (aggregation), 372, 379
NetStream data export (traditional), 372, 379
NetStream data export configuration (aggregation), 383
NetStream data export configuration (traditional), 381
NetStream data export format, 376
deadloop detection (Linux kernel PMM), 330
debugging
feature module, 6
system, 5
system maintenance, 1
default
information center log default output rules, 407
NETCONF non-default settings retrieval, 222
system information default output rules (diagnostic log), 407
system information default output rules (hidden log), 408
system information default output rules (security log), 407
system information default output rules (trace log), 408
delaying
PTP BC delay measurement, 137
PTP delay correction value, 144
PTP OC delay measurement, 137
deploying
VCF fabric automated deployment, 452
VCF fabric automated underlay network deployment configuration, 457
deployment
VCF fabric automated underlay network deployment configuration, 456
destination
information center system logs, 407
port mirroring, 335
port mirroring destination device, 335
detecting
PMM kernel thread deadloop detection, 330
PMM kernel thread starvation detection, 331
determining
ping address reachability, 2
device

Chef configuration, 280, 284, 284
Chef resources (netdev_device), 287
configuration information retrieval, 220
CWMP configuration, 295, 299, 305
feature image-based packet capture configuration, 440
feature image-based packet capture file save, 441
GOLD configuration, 427, 430
GOLD configuration (centralized IRF devices), 430
GOLD diagnostics (monitoring), 427
GOLD diagnostics (on-demand), 428
information center configuration, 406, 411, 423
information center log output configuration (console), 423, 423
information center log output configuration (Linux log host), 425
information center log output configuration (UNIX log host), 424
information center system log types, 406
IPv6 NTP multicast association mode, 108
Layer 2 remote port mirroring (egress port), 357
Layer 2 remote port mirroring (reflector port configurable), 355
Layer 2 remote port mirroring configuration, 341
Layer 3 remote port mirroring configuration, 359
Layer 3 remote port mirroring configuration (in ERSPAN mode), 350, 361
Layer 3 remote port mirroring configuration (in tunnel mode), 347
Layer 3 remote port mirroring local group, 348, 350
Layer 3 remote port mirroring local group monitor port, 349, 351
Layer 3 remote port mirroring local group source CPU, 349, 351
Layer 3 remote port mirroring local group source port, 351
local packet capture configuration (wired device), 440
local port mirroring (source CPU mode), 353
local port mirroring (source port mode), 352
local port mirroring configuration, 339
local port mirroring group monitor port, 341
local port mirroring group source CPU, 340
NETCONF capability exchange, 219
NETCONF CLI operations, 247, 248
NETCONF configuration, 212, 214, 214
NETCONF configuration modification, 237
NETCONF device configuration+state information retrieval, 220
NETCONF information retrieval, 223
NETCONF management, 214
NETCONF non-default settings retrieval, 222
NETCONF running configuration lock/unlock, 235, 236
NETCONF session information retrieval, 224, 228
NETCONF session termination, 254
NETCONF YANG file content retrieval, 224
NQA client operation, 9
NQA collaboration configuration, 68
NQA operation configuration (DHCP), 50
NQA operation configuration (DNS), 51
NQA server, 9
NTP architecture, 80
NTP broadcast association mode, 103
NTP broadcast mode+authentication, 112
NTP client/server mode+MPLS L3VPN network time synchronization, 115
NTP MPLS L3VPN instance support, 83
NTP multicast association mode, 105
NTP symmetric active/passive mode+MPLS L3VPN network time synchronization, 116
packet capture configuration (feature image-based), 443
port mirroring configuration, 335, 352
port mirroring remote destination group, 342
port mirroring remote source group, 344
port mirroring remote source group egress port, 346
port mirroring remote source group reflector port, 345
port mirroring remote source group source CPU, 345
port mirroring remote source group source ports, 344
port mirroring source device, 335
Puppet configuration, 267, 270, 270
Puppet resources (netdev_device), 271
Puppet shutdown, 269
remote packet capture configuration, 442
remote packet capture configuration (wired device), 440
SNMP common parameter configuration, 173
SNMP configuration, 170, 181
SNMP MIB, 170
SNMP notification, 177
SNMP view-based MIB access control, 170
SNMPv1 community configuration, 174, 174

SNMPv1 community configuration by community name, 174
SNMPv1 community configuration by creating SNMPv1 user, 175
SNMPv1 configuration, 181
SNMPv2c community configuration, 174, 174
SNMPv2c community configuration by community name, 174
SNMPv2c community configuration by creating SNMPv2c user, 175
SNMPv2c configuration, 181
SNMPv3 configuration, 183
SNMPv3 group and user configuration, 175
SNMPv3 group and user configuration in FIPS mode, 176
SNMPv3 group and user configuration in non-FIPS mode, 176
device role
master spine node configuration, 457
VCF fabric automated underlay network device role configuration, 456
DHCP
CWMP DHCP server, 295
NQA client operation, 12
NQA operation configuration, 50
diagnosing
GOLD configuration, 427, 430
GOLD configuration (centralized IRF devices), 430
GOLD diagnostics (on-demand), 428
GOLD type, 427
information center diagnostic log, 406
information center diagnostic log save (log file), 422
direction
port mirroring (bidirectional), 335
port mirroring (inbound), 335
port mirroring (outbound), 335
disabling
information center interface link up/link down log generation, 419
NTP message receiving, 96
displaying
CWMP settings, 305
EAA settings, 321
Event MIB, 204
feature image-based packet capture data display filter, 441, 441
GOLD, 429
information center, 423
IPv6 NetStream, 395
NetStream, 380
NQA, 45
NTP, 97
packet capture, 442
packet capture display filter configuration, 436, 439
packet file content, 441
PMM, 328
PMM kernel threads, 331
PMM user processes, 330
port mirroring, 352
RMON settings, 190
sampler, 333
sFlow, 402
SNMP settings, 181
SNTP, 121
user PMM, 329
VCF fabric, 462
DLSw
NQA client operation, 24
NQA operation configuration, 65
DNS
CWMP DNS server, 295
NQA client operation, 13
NQA client template, 33
NQA operation configuration, 51
NQA template configuration, 71
domain
name system. *Use* DNS
PTP domain, 124, 135
DSCP
NTP packet value setting, 97
DSCP value
PTP packet DSCP value (IPv4 UDP), 143
DSL network
CWMP configuration, 295, 299
duplicate log suppression, 418
dynamic
Dynamic Host Configuration Protocol. *Use* DHCP
NTP dynamic associations max, 96

E

EAA
configuration, 314, 321
environment variable configuration (user-defined), 317
event monitor, 314
event monitor policy action, 316
event monitor policy configuration (CLI), 322
event monitor policy configuration (Track), 323
event monitor policy element, 315
event monitor policy environment variable, 316

event monitor policy runtime, 316
event monitor policy user role, 316
event source, 314
how it works, 314
monitor policy, 315
monitor policy configuration, 318
monitor policy configuration (CLI-defined+environment variables), 325
monitor policy configuration (Tcl-defined), 321
monitor policy configuration restrictions, 318
monitor policy configuration restrictions (Tcl), 320
monitor policy suspension, 320
RTM, 314
settings display, 321
echo
NQA client operation (ICMP echo), 10
NQA operation configuration (ICMP echo), 46
NQA operation configuration (UDP echo), 59
egress port
Layer 2 remote port mirroring, 335
Layer 2 remote port mirroring (egress port), 357
port mirroring remote source group egress port, 346
Embedded Automation Architecture. *Use* EAA
enable
VCF fabric local proxy ARP, 461
VCF fabric overlay network L2 agent, 460
VCF fabric overlay network L3 agent, 460
enabling
CWMP, 300
Event MIB SNMP notification, 203
information center, 413
information center duplicate log suppression, 418
information center synchronous output, 418
information center system log SNMP notification, 420
NETCONF preprovisioning, 246
NQA client, 9
PTP on port, 135
SNMP agent, 172
SNMP notification, 177
SNMP version, 172
SNTP, 119
VCF fabric topology discovery, 456
encapsulating
IPv4 UDP transport protocol for PTP messages, 141
environment
EAA environment variable configuration (user-defined), 317
EAA event monitor policy environment variable, 316
establishing
NETCONF over console sessions, 218
NETCONF over SOAP sessions, 217
NETCONF over SSH sessions, 218
NETCONF over Telnet sessions, 218
NETCONF session, 215
Ethernet
CWMP configuration, 295, 299, 305
Layer 2 remote port mirroring configuration, 341
Layer 3 remote port mirroring configuration (in ERSPAN mode), 350
Layer 3 remote port mirroring configuration (in tunnel mode), 347
port mirroring configuration, 335, 352
RMON Ethernet statistics group configuration, 191
RMON statistics configuration, 188
RMON statistics entry, 188
RMON statistics group, 186
sampler configuration, 333
sampler configuration (IPv4 NetStream), 333
sFlow configuration, 400, 403, 403
Ethernet interface
Chef resources (netdev_l2_interface), 289
Puppet resources (netdev_l2_interface), 273
event
EAA configuration, 314, 321
EAA environment variable configuration (user-defined), 317
EAA event monitor, 314
EAA event monitor policy element, 315
EAA event monitor policy environment variable, 316
EAA event source, 314
EAA monitor policy, 315
NETCONF event subscription, 249, 253
NETCONF module report event subscription, 251
NETCONF module report event subscription cancel, 252
NETCONF monitoring event subscription, 250
NETCONF syslog event subscription, 249
RMON event group, 186
Event Management Information Base. *See* Event MIB
Event MIB
configuration, 195, 197, 204
display, 204
event actions, 196

event configuration, 198
monitored object, 195
object owner, 197
SNMP notification enable, 203
trigger test configuration, 200
trigger test configuration (Boolean), 206
trigger test configuration (existence), 204
trigger test configuration (threshold), 202, 209
exchanging
NETCONF capabilities, 219
existence
Event MIB trigger test, 195
Event MIB trigger test configuration, 204
exporting
IPv6 NetStream data export, 394
IPv6 NetStream data export (aggregation), 388, 394, 397
IPv6 NetStream data export (traditional), 388, 394, 396
IPv6 NetStream data export format, 391
NetStream data export, 372, 379
NetStream data export (aggregation), 372, 379
NetStream data export (traditional), 372, 379
NetStream data export configuration (aggregation), 383
NetStream data export configuration (traditional), 381
NetStream data export format, 376
NetStream format, 373

F

field
packet capture display filter keyword, 437
file
Chef configuration file, 281
information center diagnostic log output destination, 422
information center log save (log file), 416
information center log storage period (log buffer), 417
information center security log file management, 421
information center security log save (log file), 420
NETCONF YANG file content retrieval, 224
packet file content display, 441
filtering
feature image-based packet capture data display, 441, 441
IPv6 NetStream, 389
IPv6 NetStream configuration, 390
IPv6 NetStream filtering, 389
IPv6 NetStream filtering configuration, 390
NETCONF column-based filtering, 229
NETCONF data (conditional match), 234
NETCONF data (regex match), 233
NETCONF data filtering (column-based), 230
NETCONF data filtering (table-based), 230
NETCONF table-based filtering, 229
NetStream configuration, 370, 375, 381
NetStream filtering, 374
NetStream filtering configuration, 375
packet capture display filter configuration, 436, 439
packet capture filter configuration, 433, 435
FIPS compliance
information center, 411
NETCONF, 214
SNMP, 171
FIPS mode
SNMPv3 group and user configuration, 176
fixed mode (NMM sampler), 333
flow
IPv6 NetStream configuration, 386, 396
IPv6 NetStream flow aging, 387, 393
mirroring. *See* flow mirroring
NetStream flow aging, 371, 378
Sampled Flow. *Use* sFlow
flow mirroring
configuration, 364, 368
QoS policy application, 366
QoS policy application (control plane), 368
QoS policy application (global), 367
QoS policy application (interface), 366
QoS policy application (VLAN), 367
traffic behavior configuration, 365
traffic class configuration, 365
forced
IPv6 NetStream flow forced aging, 388
NetStream flow aging, 393
NetStream flow aging configuration, 379
format
information center system logs, 408
IPv6 NetStream data export, 388
IPv6 NetStream data export format, 391
IPv6 NetStream v9/v10 template refresh rate, 393
NETCONF message, 212
NetStream data export format, 376
NetStream export, 373
NetStream v9/v10 template refresh rate, 378
FTP

NQA client operation, 14
NQA client template, 41
NQA operation configuration, 52
NQA template configuration, 75
full match
NETCONF data filtering (column-based), 230

G

generating
information center interface link up/link down log generation, 419
Generic Online Diagnostics. *Use* GOLD
get operation
SNMP, 171
SNMP logging, 180
GOLD
configuration, 427, 430
configuration (centralized IRF devices), 430
diagnostic test simulation, 429
diagnostics configuration (monitoring), 427
diagnostics configuration (on-demand), 428
display, 429
log buffer size configuration, 429
maintain, 429
type, 427
grandmaster clock (PTP), 126
group
Chef resources (netdev_lagg), 290
Layer 3 remote port mirroring local group, 348, 350
Layer 3 remote port mirroring local group monitor port, 349, 351
Layer 3 remote port mirroring local group source port, 351
local port mirroring group monitor port, 341
local port mirroring group source CPU, 340
local port mirroring group source port, 340
port mirroring group, 335
Puppet resources (netdev_lagg), 274
RMON, 186
RMON alarm, 187
RMON Ethernet statistics, 186
RMON event, 186
RMON history, 186
RMON private alarm, 187
SNMPv3 configuration in non-FIPS mode, 176
group and user
SNMPv3 configuration, 175

H

hardware
GOLD configuration, 427, 430
GOLD configuration (centralized IRF devices), 430
GOLD diagnostic test simulation, 429
GOLD diagnostics (monitoring), 427
GOLD diagnostics (on-demand), 428
hidden log (information center), 406
history
NQA client history record save, 30
RMON group, 186
RMON history control entry, 188
RMON history group configuration, 191
host
information center log output (log host), 415
HTTP
NQA client operation, 15
NQA client template, 38
NQA operation configuration, 53
NQA template configuration, 74
HTTPS
CWMP ACS HTTPS SSL client policy, 302
NQA client template, 39
NQA template configuration, 75
hybrid
PTP clock node (hybrid), 124

I

ICMP
NQA client operation (ICMP echo), 10
NQA client operation (ICMP jitter), 11
NQA client template, 32
NQA collaboration configuration, 68
NQA operation configuration (ICMP echo), 46
NQA operation configuration (ICMP jitter), 47
NQA template configuration, 70
ping command, 1
identifying
tracert node failure, 4, 4
image
packet capture configuration (feature image-based), 443
packet capture feature image-based configuration, 440
packet capture feature image-based mode, 432
inbound
port mirroring, 335
information
device configuration information retrieval, 220
information center
configuration, 406, 411, 423
default output rules (diagnostic log), 407
default output rules (hidden log), 408

default output rules (security log), 407
default output rules (trace log), 408
diagnostic log save (log file), 422
display, 423
duplicate log suppression, 418
enable, 413
FIPS compliance, 411
interface link up/link down log generation, 419
log default output rules, 407
log output (console), 413
log output (log host), 415
log output (monitor terminal), 414
log output configuration (console), 423
log output configuration (Linux log host), 425
log output configuration (UNIX log host), 424
log output destinations, 413
log save (log file), 416
log storage period (log buffer), 417
log suppression configuration, 418
log suppression for module, 419
maintain, 423
security log file management, 421
security log management, 420
security log save (log file), 420
synchronous log output, 418
system information log types, 406
system log destinations, 407
system log formats and field descriptions, 408
system log levels, 406
system log SNMP notification, 420
trace log file max size, 422
initiating
CWMP ACS connection initiation, 304
interface
Chef resources (netdev_interface), 287
Puppet resources (netdev_interface), 272
Puppet resources (netdev_l2_interface), 273
Internet
NQA configuration, 7, 8, 46
SNMP common parameter configuration, 173
SNMP configuration, 170, 181
SNMP MIB, 170
SNMP2c community configuration by community name, 174
SNMP2c community configuration by creating SNMPv2c user, 175
SNMPv1 community configuration, 174, 174
SNMPv1 community configuration by community name, 174
SNMPv1 community configuration by creating SNMPv1 user, 175
SNMPv2c community configuration, 174, 174
SNMPv3 group and user configuration, 175
SNMPv3 group and user configuration in FIPS mode, 176
SNMPv3 group and user configuration in non-FIPS mode, 176
interval
CWMP ACS periodic Inform feature, 304
PTP announce message interval+timeout, 138
sampler creation, 333
IP addressing
PTP multicast message source IP address (UDP), 141
PTP unicast message destination IP address (IPv4 UDP), 142
tracert, 3
tracert node failure identification, 4, 4
IP services
NQA client history record save, 30
NQA client operation (DHCP), 12
NQA client operation (DLSw), 24
NQA client operation (DNS), 13
NQA client operation (FTP), 14
NQA client operation (HTTP), 15
NQA client operation (ICMP echo), 10
NQA client operation (ICMP jitter), 11
NQA client operation (path jitter), 24
NQA client operation (SNMP), 18
NQA client operation (TCP), 18
NQA client operation (UDP echo), 19
NQA client operation (UDP jitter), 16
NQA client operation (UDP tracert), 20
NQA client operation (voice), 22
NQA client operation optional parameters, 26
NQA client operation scheduling, 31
NQA client statistics collection, 29
NQA client template (DNS), 33
NQA client template (FTP), 41
NQA client template (HTTP), 38
NQA client template (HTTPS), 39
NQA client template (ICMP), 32
NQA client template (RADIUS), 42
NQA client template (SSL), 43
NQA client template (TCP half open), 35
NQA client template (TCP), 34
NQA client template (UDP), 36
NQA client template optional parameters, 44
NQA client threshold monitoring, 27
NQA client+Track collaboration, 27
NQA collaboration configuration, 68
NQA configuration, 7, 8, 46

NQA operation configuration (DHCP), 50
NQA operation configuration (DLSw), 65
NQA operation configuration (DNS), 51
NQA operation configuration (FTP), 52
NQA operation configuration (HTTP), 53
NQA operation configuration (ICMP echo), 46
NQA operation configuration (ICMP jitter), 47
NQA operation configuration (path jitter), 66
NQA operation configuration (SNMP), 57
NQA operation configuration (TCP), 58
NQA operation configuration (UDP echo), 59
NQA operation configuration (UDP jitter), 54
NQA operation configuration (UDP tracert), 61
NQA operation configuration (voice), 62
NQA template configuration (DNS), 71
NQA template configuration (FTP), 75
NQA template configuration (HTTP), 74
NQA template configuration (HTTPS), 75
NQA template configuration (ICMP), 70
NQA template configuration (RADIUS), 76
NQA template configuration (SSL), 77
NQA template configuration (TCP half open), 72
NQA template configuration (TCP), 72
NQA template configuration (UDP), 73
IPv4
PTP multicast message source IP address (UDP), 141
PTP unicast message destination IP address (IPv4 UDP), 142
UDP transport protocol for PTP messages, 141
IPv4 UDP
PTP packet DSCP value (IPv4 UDP), 143
PTP unicast message destination IP address, 142
transport protocol for PTP messages, 141
IPv6
NTP client/server association mode, 99
NTP multicast association mode, 108
NTP symmetric active/passive association mode, 102
IPv6 NetStream
architecture, 386
configuration, 386, 390, 396
data export (aggregation), 388
data export (traditional), 388
data export configuration, 394
data export configuration (aggregation), 394, 397
data export configuration (traditional), 394, 396
data export configuration restrictions, 394
data export format, 391
display, 395
enable, 390
export format, 388
filtering, 389
filtering configuration, 390
filtering configuration restrictions, 390
flow aging, 387
flow aging configuration, 393
maintain, 395
protocols and standards, 389
sampling, 389
sampling configuration, 391
v9/v10 template refresh rate, 393

K

kernel thread
display, 331
Linux process, 327
maintain, 331
PMM, 330
PMM deadloop detection, 330
PMM starvation detection, 331
keyword
packet capture, 432
packet capture filter, 433

L

label
VXLAN-aware NetStream, 378
language
Puppet configuration, 267, 270, 270
Layer 2
port mirroring configuration, 335, 352
remote port mirroring, 336
remote port mirroring (egress port), 357
remote port mirroring (reflector port configurable), 355
remote port mirroring configuration, 341
Layer 3
port mirroring configuration, 335, 352
remote port mirroring, 338
remote port mirroring configuration, 359
remote port mirroring configuration (in ERSPAN mode), 350, 361
remote port mirroring configuration (in tunnel mode), 347
tracert, 3
tracert node failure identification, 4, 4
level

information center system logs, 406
link
information center interface link up/link down log generation, 419
Linux
information center log host output configuration, 425
kernel thread, 327
PMM, 327
PMM kernel thread, 330
PMM kernel thread deadloop detection, 330
PMM kernel thread display, 331
PMM kernel thread maintain, 331
PMM kernel thread starvation detection, 331
PMM user process display, 330
PMM user process maintain, 330
Puppet configuration, 267, 270, 270
loading
NETCONF configuration, 241
local
NTP local clock as reference source, 88
packet capture configuration (wired device), 440
packet capture mode, 432
port mirroring, 336
port mirroring configuration, 339
port mirroring group creation, 340
port mirroring group monitor port, 341
port mirroring group source CPU, 340
port mirroring group source port, 340
PTP clock type, 126
locking
NETCONF running configuration, 235, 236
log field description
information center system logs, 408
logging
GOLD log buffer size, 429
information center configuration, 406, 411, 423
information center diagnostic log save (log file), 422
information center diagnostic logs, 406
information center duplicate log suppression, 418
information center hidden logs, 406
information center interface link up/link down log generation, 419
information center log default output rules, 407
information center log output (console), 413
information center log output (log host), 415
information center log output (monitor terminal), 414
information center log output configuration (console), 423
information center log output configuration (Linux log host), 425
information center log output configuration (UNIX log host), 424
information center log save (log file), 416
information center log storage period (log buffer), 417
information center security log file management, 421
information center security log management, 420
information center security log save (log file), 420
information center security logs, 406
information center standard system logs, 406
information center synchronous log output, 418
information center system log destinations, 407
information center system log formats and field descriptions, 408
information center system log levels, 406
information center system log SNMP notification, 420
information center trace log file max size, 422
SNMP configuration, 180
system information default output rules (diagnostic log), 407
system information default output rules (hidden log), 408
system information default output rules (security log), 407
system information default output rules (trace log), 408
logical
packet capture display filter configuration (logical expression), 439
packet capture display filter operator, 438
packet capture filter configuration (logical expression), 435
packet capture filter operator, 434
packet capture operator, 432

M

MAC addressing
PTP non-Pdelay message MAC address, 142
maintaining
GOLD, 429
information center, 423
IPv6 NetStream, 395
NetStream, 380
PMM kernel thread, 330
PMM kernel threads, 331

PMM Linux, 327
PMM user processes, 330
process monitoring and maintenance. *See* PMM
user PMM, 329
Management Information Base. *Use* MIB
managing
information center security log file, 421
information center security logs, 420
manifest
Puppet resources, 268, 271
master
PTP master-member/subordinate relationship, 126
matching
NETCONF data filtering (column-based), 230
NETCONF data filtering (column-based) (conditional match), 232
NETCONF data filtering (column-based) (full match), 230
NETCONF data filtering (column-based) (regex match), 231
NETCONF data filtering (conditional match), 234
NETCONF data filtering (regex match), 233
NETCONF data filtering (table-based), 230
packet capture display filter configuration (proto[…] expression), 439
member
PTP OC configuration as member clock, 134
message
IPv4 UDP transport protocol for PTP messages, 141
NETCONF format, 212
NTP message receiving disable, 96
NTP message source address, 95
PTP announce message interval+timeout, 138
MIB
Event MIB configuration, 195, 197, 204
Event MIB event actions, 196
Event MIB event configuration, 198
Event MIB monitored object, 195
Event MIB object owner, 197
Event MIB trigger test configuration, 200
Event MIB trigger test configuration (Boolean), 206
Event MIB trigger test configuration (existence), 204
Event MIB trigger test configuration (threshold), 202, 209
SNMP, 170, 170
SNMP Get operation, 171
SNMP Set operation, 171
SNMP view-based access control, 170
mirroring
flow. *See* flow mirroring
port. *See* port mirroring
mode
NTP association, 85
NTP broadcast association, 81, 86
NTP client/server association, 81, 85
NTP multicast association, 81, 87
NTP symmetric active/passive association, 81, 86
packet capture feature image-based, 432
packet capture local, 432
packet capture remote, 432
PTP timestamp single-step, 137
PTP timestamp two-step, 137
sampler fixed, 333
sampler random, 333
SNMP access control (rule-based), 171
SNMP access control (view-based), 171
modifying
NETCONF configuration, 237, 238
module
feature module debug, 6
information center configuration, 406, 411, 423
information center log suppression for module, 419
NETCONF configuration data retrieval (all modules), 226
NETCONF configuration data retrieval (Syslog module), 227
NETCONF module report event subscription, 251
NETCONF module report event subscription cancel, 252
monitor terminal
information center log output, 414
monitoring
EAA configuration, 314
EAA environment variable configuration (user-defined), 317
Event MIB configuration, 195, 197, 204
Event MIB trigger test configuration (Boolean), 206
Event MIB trigger test configuration (existence), 204
Event MIB trigger test configuration (threshold), 209
GOLD configuration, 430
GOLD configuration (centralized IRF devices), 430
GOLD diagnostics (monitoring), 427

NETCONF monitoring event subscription, 250
network, 370, *See also* NMM
NQA client threshold monitoring, 27
NQA threshold monitoring, 8
PMM, 328
PMM kernel thread, 330
PMM Linux, 327
process monitoring and maintenance. *See* PMM
user PMM, 329
MPLS L3VPN
NTP support for MPLS L3VPN instance, 83
multicast
IPv6 NTP multicast association mode, 108
NTP multicast association mode, 81, 87, 105
NTP multicast mode authentication, 93
NTP multicast mode dynamic associations max, 96
PTP multicast message source IP address (UDP), 141

N

NDA
IPv6 NetStream data analyzer, 386
NetStream architecture, 370
NDE
IPv6 NetStream data exporter, 386
NetStream architecture, 370
NETCONF
capability exchange, 219
Chef configuration, 280, 284, 284
CLI operations, 247, 248
CLI return, 255
configuration, 212, 214
configuration data retrieval (all modules), 226
configuration data retrieval (Syslog module), 227
configuration load, 241
configuration modification, 237, 238
configuration rollback, 242
configuration rollback (configuration file-based), 242
configuration rollback (rollback point-based), 242
configuration save, 239
data entry retrieval (interface table), 225
data filtering, 229
data filtering (conditional match), 234
data filtering (regex match), 233
device configuration, 214
device configuration information retrieval, 220
device configuration+state information retrieval, 220
device management, 214
event subscription, 249, 253
FIPS compliance, 214
information retrieval, 223
message format, 212
module report event subscription, 251
module report event subscription cancel, 252
monitoring event subscription, 250
NETCONF over console session establishment, 218
NETCONF over SOAP session establishment, 217
NETCONF over SSH session establishment, 218
NETCONF over Telnet session establishment, 218
non-default settings retrieval, 222
over SOAP, 212
preprovisioning enable, 246
protocols and standards, 214
Puppet configuration, 267, 270, 270
running configuration lock/unlock, 235, 236
running configuration save, 240
session attribute set, 215
session establishment, 215
session establishment restrictions, 215
session information retrieval, 224, 228
session termination, 254
structure, 212
supported operations, 256
syslog event subscription, 249
YANG file content retrieval, 224
NetStream
architecture, 370
configuration, 370, 375, 381
data export, 372
data export (aggregation), 372
data export (traditional), 372
data export configuration, 379
data export configuration (aggregation), 379, 383
data export configuration (traditional), 379, 381
data export format configuration, 376
data export restrictions (aggregation), 380
display, 380
enable, 375
export format, 373
filtering, 374
filtering configuration, 375
flow aging, 371
flow aging configuration, 378

flow aging configuration (forced), 379
flow aging configuration (periodic), 378
IPv6. *See* IPv6 NetStream
maintain, 380
NDA, 370
NDE, 370
NSC, 370
protocols and standards, 374
sampler configuration, 333
sampler configuration (IPv4 NetStream), 333
sampler creation, 333
sampling configuration, 376
sampling configuration restrictions, 376
v9/v10 template refresh rate, 378
VXLAN-aware configuration, 378
network
Chef network framework, 280
Chef resources, 281, 287
Event MIB SNMP notification enable, 203
Event MIB trigger test configuration (Boolean), 206
Event MIB trigger test configuration (existence), 204
Event MIB trigger test configuration (threshold), 209
feature module debug, 6
flow mirroring configuration, 364, 368
flow mirroring traffic behavior, 365
GOLD log buffer size, 429
information center diagnostic log save (log file), 422
information center duplicate log suppression, 418
information center interface link up/link down log generation, 419
information center log output configuration (console), 423
information center log output configuration (Linux log host), 425
information center log output configuration (UNIX log host), 424
information center log storage period (log buffer), 417
information center security log file management, 421
information center security log save (log file), 420
information center synchronous log output, 418
information center system log SNMP notification, 420
information center system log types, 406
information center trace log file max size, 422
IPv6 NetStream filtering, 389
IPv6 NetStream filtering configuration, 390
IPv6 NetStream sampling, 389
IPv6 NetStream sampling configuration, 391
Layer 2 remote port mirroring (egress port), 357
Layer 2 remote port mirroring (reflector port configurable), 355
Layer 2 remote port mirroring configuration, 341
Layer 3 remote port mirroring configuration, 359
Layer 3 remote port mirroring configuration (in ERSPAN mode), 350, 361
Layer 3 remote port mirroring configuration (in tunnel mode), 347
Layer 3 remote port mirroring local group, 348, 350
Layer 3 remote port mirroring local group monitor port, 349, 351
Layer 3 remote port mirroring local group source CPU, 349, 351
Layer 3 remote port mirroring local group source port, 351
local port mirroring (source CPU mode), 353
local port mirroring (source port mode), 352
local port mirroring configuration, 339
local port mirroring group monitor port, 341
local port mirroring group source CPU, 340
local port mirroring group source port, 340
monitoring, 370, *See also* NMM
NETCONF preprovisioning enable, 246
NetStream data export configuration (traditional), 381
NetStream filtering, 374
NetStream filtering configuration, 375
NetStream sampling, 374
NetStream sampling configuration, 376
Network Configuration Protocol. *Use* NETCONF
Network Time Protocol. *Use* NTP
NQA client history record save, 30
NQA client operation, 9
NQA client operation (DHCP), 12
NQA client operation (DLSw), 24
NQA client operation (DNS), 13
NQA client operation (FTP), 14
NQA client operation (HTTP), 15
NQA client operation (ICMP echo), 10
NQA client operation (ICMP jitter), 11
NQA client operation (path jitter), 24
NQA client operation (SNMP), 18
NQA client operation (TCP), 18
NQA client operation (UDP echo), 19

NQA client operation (UDP jitter), 16
NQA client operation (UDP tracert), 20
NQA client operation (voice), 22
NQA client operation optional parameters, 26
NQA client operation scheduling, 31
NQA client statistics collection, 29
NQA client template, 31
NQA client threshold monitoring, 27
NQA client+Track collaboration, 27
NQA collaboration configuration, 68
NQA operation configuration (DHCP), 50
NQA operation configuration (DLSw), 65
NQA operation configuration (DNS), 51
NQA operation configuration (FTP), 52
NQA operation configuration (HTTP), 53
NQA operation configuration (ICMP echo), 46
NQA operation configuration (ICMP jitter), 47
NQA operation configuration (path jitter), 66
NQA operation configuration (SNMP), 57
NQA operation configuration (TCP), 58
NQA operation configuration (UDP echo), 59
NQA operation configuration (UDP jitter), 54
NQA operation configuration (UDP tracert), 61
NQA operation configuration (voice), 62
NQA server, 9
NQA template configuration (DNS), 71
NQA template configuration (FTP), 75
NQA template configuration (HTTP), 74
NQA template configuration (HTTPS), 75
NQA template configuration (ICMP), 70
NQA template configuration (RADIUS), 76
NQA template configuration (SSL), 77
NQA template configuration (TCP half open), 72
NQA template configuration (TCP), 72
NQA template configuration (UDP), 73
NTP association mode, 85
NTP client/server mode+MPLS L3VPN network time synchronization, 115
NTP message receiving disable, 96
NTP MPLS L3VPN instance support, 83
NTP symmetric active/passive mode+MPLS L3VPN network time synchronization, 116
ping network connectivity test, 1
PMM 3rd party process start, 328
PMM 3rd party process stop, 328
port mirroring remote destination group, 342
port mirroring remote source group, 344
port mirroring remote source group egress port, 346
port mirroring remote source group reflector port, 345
port mirroring remote source group source CPU, 345
port mirroring remote source group source ports, 344
PTP configuration (AES67-2015, IPv4 UDP transport, multicast transmission), 166
PTP configuration (IEEE 1588 v2, IPv4 UDP transport, multicast transmission), 150
PTP configuration (IEEE 1588 v2, IPv4 UDP transport, unicast transmission), 153
PTP configuration (IEEE 802.1AS, IEEE 802.3/Ethernet transport, multicast transmission), 156
PTP configuration (SMPTE ST 2059-2, IPv4 UDP transport, multicast transmission), 159
PTP configuration (SMPTE ST 2059-2, IPv4 UDP transport, unicast transmission), 163
Puppet network framework, 267
Puppet resources, 268, 271
quality analyzer. *See* NQA
RMON alarm configuration, 189, 192
RMON alarm group sample types, 188
RMON Ethernet statistics group configuration, 191
RMON history group configuration, 191
RMON statistics configuration, 188
RMON statistics function, 188
sFlow counter sampling configuration, 402
sFlow flow sampling configuration, 401
SNMP common parameter configuration, 173
SNMPv1 community configuration, 174, 174
SNMPv1 community configuration by community name, 174
SNMPv1 community configuration by creating SNMPv1 user, 175
SNMPv2c community configuration, 174, 174
SNMPv2c community configuration by community name, 174
SNMPv2c community configuration by creating SNMPv2c user, 175
SNMPv3 group and user configuration, 175
SNMPv3 group and user configuration in FIPS mode, 176
SNMPv3 group and user configuration in non-FIPS mode, 176
tracert node failure identification, 4, 4
VCF fabric automated deployment, 452
VCF fabric automated underlay network deployment configuration, 456, 457
VCF fabric Neutron deployment, 451
VCF fabric topology, 447

VXLAN-aware NetStream, 378
network management
Chef configuration, 280, 284, 284
CWMP basic functions, 295
CWMP configuration, 295, 299, 305
EAA configuration, 314, 321
Event MIB configuration, 195, 197, 204
GOLD configuration, 427, 430
GOLD configuration (centralized IRF devices), 430
information center configuration, 406, 411, 423
IPv6 NetStream configuration, 386, 390, 396
NETCONF configuration, 212
NetStream configuration, 370, 375, 381
NQA configuration, 7, 8, 46
NTP configuration, 79, 84, 98
packet capture configuration, 432, 442
PMM Linux network, 327
port mirroring configuration, 335, 352
PTP configuration, 124
Puppet configuration, 267, 270, 270
RMON configuration, 186, 191
sampler configuration, 333
sampler configuration (IPv4 NetStream), 333
sampler creation, 333
sFlow configuration, 400, 403, 403
SNMP configuration, 170, 181
SNMPv1 configuration, 181
SNMPv2c configuration, 181
SNMPv3 configuration, 183
VCF fabric configuration, 447, 454
Neutron
VCF fabric, 450
VCF fabric Neutron deployment, 451
NMM
CWMP ACS attributes, 300
CWMP ACS attributes (default)(CLI), 301
CWMP ACS attributes (preferred), 300
CWMP ACS autoconnect parameters, 304
CWMP ACS HTTPS SSL client policy, 302
CWMP basic functions, 295
CWMP configuration, 295, 299, 305
CWMP CPE ACS authentication parameters, 302
CWMP CPE ACS connection interface, 303
CWMP CPE ACS provision code, 303
CWMP CPE attributes, 302
CWMP framework, 295
CWMP settings display, 305
device configuration information retrieval, 220
EAA configuration, 314, 321
EAA environment variable configuration (user-defined), 317
EAA event monitor, 314
EAA event monitor policy configuration (CLI), 322
EAA event monitor policy configuration (Track), 323
EAA event monitor policy element, 315
EAA event monitor policy environment variable, 316
EAA event source, 314
EAA monitor policy, 315
EAA monitor policy configuration, 318
EAA monitor policy configuration (CLI-defined+environment variables), 325
EAA monitor policy configuration (Tcl-defined), 321
EAA monitor policy suspension, 320
EAA RTM, 314
EAA settings display, 321
feature image-based packet capture configuration, 440
feature module debug, 6
flow mirroring configuration, 364, 368
flow mirroring QoS policy application, 366
flow mirroring traffic behavior, 365
GOLD configuration, 427
GOLD diagnostic test simulation, 429
GOLD diagnostics (monitoring), 427
GOLD diagnostics (on-demand), 428
GOLD display, 429
GOLD maintain, 429
GOLD type, 427
information center configuration, 406, 411, 423
information center diagnostic log save (log file), 422
information center display, 423
information center duplicate log suppression, 418
information center interface link up/link down log generation, 419
information center log default output rules, 407
information center log destinations, 407
information center log formats and field descriptions, 408
information center log levels, 406
information center log output (console), 413
information center log output (log host), 415
information center log output (monitor terminal), 414
information center log output configuration (console), 423

information center log output configuration (Linux log host), 425
information center log output configuration (UNIX log host), 424
information center log output destinations, 413
information center log save (log file), 416
information center log storage period (log buffer), 417
information center log suppression for module, 419
information center maintain, 423
information center security log file management, 421
information center security log management, 420
information center security log save (log file), 420
information center synchronous log output, 418
information center system log SNMP notification, 420
information center system log types, 406
information center trace log file max size, 422
IPv4 UDP transport protocol for PTP messages, 141
IPv6 NetStream architecture, 386
IPv6 NetStream configuration, 386, 390
IPv6 NetStream data export, 388
IPv6 NetStream data export configuration, 394
IPv6 NetStream data export configuration restrictions, 394
IPv6 NetStream data export format, 391
IPv6 NetStream display, 395
IPv6 NetStream enable, 390
IPv6 NetStream filtering, 389
IPv6 NetStream filtering configuration, 390
IPv6 NetStream filtering configuration restrictions, 390
IPv6 NetStream flow aging, 393
IPv6 NetStream maintain, 395
IPv6 NetStream protocols and standards, 389
IPv6 NetStream sampling, 389
IPv6 NetStream sampling configuration, 391
IPv6 NetStream v9/v10 template refresh rate, 393
IPv6 NTP client/server association mode configuration, 99
IPv6 NTP multicast association mode configuration, 108
IPv6 NTP symmetric active/passive association mode configuration, 102
Layer 2 remote port mirroring (egress port), 357
Layer 2 remote port mirroring (reflector port configurable), 355
Layer 2 remote port mirroring configuration, 341
Layer 3 remote port mirroring configuration, 359
Layer 3 remote port mirroring configuration (in ERSPAN mode), 350, 361
Layer 3 remote port mirroring configuration (in tunnel mode), 347
Layer 3 remote port mirroring local group, 348, 350
Layer 3 remote port mirroring local group monitor port, 349, 351
Layer 3 remote port mirroring local group source CPU, 349, 351
Layer 3 remote port mirroring local group source port, 351
local packet capture configuration (wired device), 440
local port mirroring (source CPU mode), 353
local port mirroring (source port mode), 352
local port mirroring configuration, 339
local port mirroring group, 340
local port mirroring group monitor port, 341
local port mirroring group source CPU, 340
local port mirroring group source port, 340
NETCONF capability exchange, 219
NETCONF CLI operations, 247, 248
NETCONF CLI return, 255
NETCONF configuration, 212, 214
NETCONF configuration data retrieval (all modules), 226
NETCONF configuration data retrieval (Syslog module), 227
NETCONF configuration modification, 237, 238
NETCONF data entry retrieval (interface table), 225
NETCONF data filtering, 229
NETCONF device configuration+state information retrieval, 220
NETCONF event subscription, 249, 253
NETCONF information retrieval, 223
NETCONF module report event subscription, 251
NETCONF module report event subscription cancel, 252
NETCONF monitoring event subscription, 250
NETCONF non-default settings retrieval, 222
NETCONF over console session establishment, 218
NETCONF over SOAP session establishment, 217
NETCONF over SSH session establishment, 218

NETCONF over Telnet session establishment, 218
NETCONF protocols and standards, 214
NETCONF running configuration lock/unlock, 235, 236
NETCONF session establishment, 215
NETCONF session information retrieval, 224, 228
NETCONF session termination, 254
NETCONF structure, 212
NETCONF supported operations, 256
NETCONF syslog event subscription, 249
NETCONF YANG file content retrieval, 224
NetStream architecture, 370
NetStream configuration, 370, 375, 381, 381
NetStream data export, 372, 379
NetStream data export format, 376
NetStream data export restrictions (aggregation), 380
NetStream display, 380
NetStream enable, 375
NetStream filtering, 374
NetStream filtering configuration, 375
NetStream flow aging, 371, 378
NetStream format, 373
NetStream maintain, 380
NetStream protocols and standards, 374
NetStream sampling, 374
NetStream sampling configuration, 376
NetStream sampling configuration restrictions, 376
NetStream v9/v10 template refresh rate, 378
NQA client history record save, 30
NQA client history record save restrictions, 30
NQA client operation, 9
NQA client operation (DHCP), 12
NQA client operation (DLSw), 24
NQA client operation (DNS), 13
NQA client operation (FTP), 14
NQA client operation (HTTP), 15
NQA client operation (ICMP echo), 10
NQA client operation (ICMP jitter), 11
NQA client operation (path jitter), 24
NQA client operation (SNMP), 18
NQA client operation (TCP), 18
NQA client operation (UDP echo), 19
NQA client operation (UDP jitter), 16
NQA client operation (UDP tracert), 20
NQA client operation (voice), 22
NQA client operation optional parameter configuration restrictions, 26
NQA client operation optional parameters, 26
NQA client operation restrictions (FTP), 14
NQA client operation restrictions (ICMP jitter), 12
NQA client operation restrictions (UDP tracert), 20
NQA client operation scheduling, 31
NQA client statistics collection, 29
NQA client statistics collection restrictions, 29
NQA client template, 31
NQA client template (DNS), 33
NQA client template (FTP), 41
NQA client template (HTTP), 38
NQA client template (HTTPS), 39
NQA client template (ICMP), 32
NQA client template (RADIUS), 42
NQA client template (SSL), 43
NQA client template (TCP half open), 35
NQA client template (TCP), 34
NQA client template (UDP), 36
NQA client template configuration restrictions, 31
NQA client template optional parameter configuration restrictions, 44
NQA client template optional parameters, 44
NQA client threshold monitoring, 27
NQA client threshold monitoring configuration restrictions, 28
NQA client+Track collaboration, 27
NQA client+Track collaboration restrictions, 27
NQA collaboration configuration, 68
NQA configuration, 7, 8, 46
NQA display, 45
NQA operation configuration (DHCP), 50
NQA operation configuration (DLSw), 65
NQA operation configuration (DNS), 51
NQA operation configuration (FTP), 52
NQA operation configuration (HTTP), 53
NQA operation configuration (ICMP echo), 46
NQA operation configuration (ICMP jitter), 47
NQA operation configuration (path jitter), 66
NQA operation configuration (SNMP), 57
NQA operation configuration (TCP), 58
NQA operation configuration (UDP echo), 59
NQA operation configuration (UDP jitter), 54
NQA operation configuration (UDP tracert), 61
NQA operation configuration (voice), 62
NQA server, 9
NQA server configuration restrictions, 9
NQA template, 8
NQA template configuration (DNS), 71
NQA template configuration (FTP), 75
NQA template configuration (HTTP), 74
NQA template configuration (HTTPS), 75

NQA template configuration (ICMP), 70
NQA template configuration (RADIUS), 76
NQA template configuration (SSL), 77
NQA template configuration (TCP half open), 72
NQA template configuration (TCP), 72
NQA template configuration (UDP), 73
NQA threshold monitoring, 8
NQA+Track collaboration, 7
NTP architecture, 80
NTP association mode, 85
NTP authentication configuration, 89
NTP broadcast association mode configuration, 86, 103
NTP broadcast mode authentication configuration, 92
NTP broadcast mode+authentication, 112
NTP client/server association mode configuration, 98
NTP client/server mode authentication configuration, 89
NTP client/server mode+authentication, 111
NTP client/server mode+MPLS L3VPN network time synchronization, 115
NTP configuration, 79, 84, 98
NTP display, 97
NTP dynamic associations max, 96
NTP local clock as reference source, 88
NTP message receiving disable, 96
NTP message source address specification, 95
NTP multicast association mode, 87
NTP multicast association mode configuration, 105
NTP multicast mode authentication configuration, 93
NTP optional parameter configuration, 95
NTP packet DSCP value setting, 97
NTP protocols and standards, 84, 119
NTP security, 82
NTP symmetric active/passive association mode configuration, 100
NTP symmetric active/passive mode authentication configuration, 90
NTP symmetric active/passive mode+MPLS L3VPN network time synchronization, 116
packet capture configuration, 432, 442
packet capture configuration (feature image-based), 443
packet capture display, 442
packet capture display filter configuration, 436, 439
packet capture filter configuration, 433, 435
packet file content display, 441
ping address reachability determination, 2
ping command, 1
ping network connectivity test, 1
port mirroring classification, 336
port mirroring configuration, 335, 352
port mirroring display, 352
port mirroring remote destination group, 342
port mirroring remote source group, 344
PTP announce message interval+timeout, 138
PTP basic concepts, 124
PTP BC delay measurement, 137
PTP clock node, 124
PTP clock node type, 134
PTP clock priority, 145
PTP clock type, 126
PTP configuration, 124, 147
PTP configuration (IEEE 1588 v2, IEEE 802.3/Ethernet transport, multicast transmission), 147
PTP configuration (IEEE 1588 v2, IPv4 UDP transport, unicast transmission), 153
PTP cumulative offset (UTC:TAI), 144
PTP delay correction value, 144
PTP domain, 124, 135
PTP grandmaster clock, 126
PTP master-member/subordinate relationship, 126
PTP multicast message source IP address (UDP), 141
PTP non-Pdelay message MAC address, 142
PTP OC configuration as member clock, 134
PTP OC delay measurement, 137
PTP OC-type port configuration on a TC+OC clock, 138
PTP packet DSCP value (IPv4 UDP), 143
PTP port role, 136
PTP profile, 124, 133, 133
PTP protocols and standards, 129
PTP synchronization, 127
PTP system time source, 133
PTP timestamp, 137
PTP unicast message destination IP address (IPv4 UDP), 142
PTP UTC correction date, 145
PTPconfiguration (AES67-2015, IPv4 UDP transport, multicast transmission), 166
PTPconfiguration (IEEE 1588 v2, IPv4 UDP transport, multicast transmission), 150

PTPconfiguration (IEEE 802.1AS, IEEE 802.3/Ethernet transport, multicast transmission), 156
PTPconfiguration (SMPTE ST 2059-2, IPv4 UDP transport, multicast transmission), 159
PTPconfiguration (SMPTE ST 2059-2, IPv4 UDP transport, unicast transmission), 163
remote packet capture configuration, 442
remote packet capture configuration (wired device), 440
RMON alarm configuration, 192
RMON configuration, 186, 191
RMON Ethernet statistics group configuration, 191
RMON group, 186
RMON history group configuration, 191
RMON protocols and standards, 188
RMON settings display, 190
sampler configuration, 333
sampler configuration (IPv4 NetStream), 333
sampler creation, 333
sFlow agent+collector information configuration, 401
sFlow configuration, 400, 403, 403
sFlow counter sampling configuration, 402
sFlow display, 402
sFlow flow sampling configuration, 401
sFlow protocols and standards, 400
SNMP access control mode, 171
SNMP configuration, 170, 181
SNMP framework, 170
SNMP Get operation, 171
SNMP host notification send, 178
SNMP logging configuration, 180
SNMP MIB, 170
SNMP notification, 177
SNMP protocol versions, 171
SNMP settings display, 181
SNMP view-based MIB access control, 170
SNMPv1 configuration, 181
SNMPv2c configuration, 181
SNMPv3 configuration, 183
SNTP authentication, 120
SNTP configuration, 84, 119, 122, 122
SNTP display, 121
SNTP enable, 119
system debugging, 1, 5
system information default output rules (diagnostic log), 407
system information default output rules (hidden log), 408
system information default output rules (security log), 407
system information default output rules (trace log), 408
system maintenance, 1
tracert, 3
tracert node failure identification, 4, 4
troubleshooting sFlow, 404
troubleshooting sFlow remote collector cannot receive packets, 404
VCF fabric configuration, 454
VCF fabric topology discovery, 456
VXLAN-aware NetStream, 378
NMS
Event MIB SNMP notification enable, 203
RMON configuration, 186, 191
SNMP Notification operation, 171
SNMP protocol versions, 171
SNMP Set operation, 171, 171
node
Event MIB monitored object, 195
PTP clock node type, 134
non-default
NETCONF non-default settings retrieval, 222
non-FIPS mode
SNMPv3 group and user configuration, 176
non-Pdelay message, 142
notifying
Event MIB SNMP notification enable, 203
information center system log SNMP notification, 420
NETCONF syslog event subscription, 249
SNMP configuration, 170, 181
SNMP host notification send, 178
SNMP notification, 177
SNMP Notification operation, 171
NQA
client enable, 9
client history record save, 30
client history record save restrictions, 30
client operation, 9
client operation (DHCP), 12
client operation (DLSw), 24
client operation (DNS), 13
client operation (FTP), 14
client operation (HTTP), 15
client operation (ICMP echo), 10
client operation (ICMP jitter), 11
client operation (path jitter), 24
client operation (SNMP), 18
client operation (TCP), 18

client operation (UDP echo), 19
client operation (UDP jitter), 16
client operation (UDP tracert), 20
client operation (voice), 22
client operation optional parameter configuration restrictions, 26
client operation optional parameters, 26
client operation restrictions (FTP), 14
client operation restrictions (ICMP jitter), 12
client operation restrictions (UDP tracert), 20
client operation scheduling, 31
client operation scheduling restrictions, 31
client statistics collection, 29
client statistics collection restrictions, 29
client template (DNS), 33
client template (FTP), 41
client template (HTTP), 38
client template (HTTPS), 39
client template (ICMP), 32
client template (RADIUS), 42
client template (SSL), 43
client template (TCP half open), 35
client template (TCP), 34
client template (UDP), 36
client template configuration, 31
client template configuration restrictions, 31
client template optional parameter configuration restrictions, 44
client template optional parameters, 44
client threshold monitoring, 27
client threshold monitoring configuration restrictions, 28
client+Track collaboration, 27
client+Track collaboration restrictions, 27
collaboration configuration, 68
configuration, 7, 8, 46
display, 45
how it works, 7
operation configuration (DHCP), 50
operation configuration (DLSw), 65
operation configuration (DNS), 51
operation configuration (FTP), 52
operation configuration (HTTP), 53
operation configuration (ICMP echo), 46
operation configuration (ICMP jitter), 47
operation configuration (path jitter), 66
operation configuration (SNMP), 57
operation configuration (TCP), 58
operation configuration (UDP echo), 59
operation configuration (UDP jitter), 54
operation configuration (UDP tracert), 61
operation configuration (voice), 62
server configuration, 9
server configuration restrictions, 9
template, 8
template configuration (DNS), 71
template configuration (FTP), 75
template configuration (HTTP), 74
template configuration (HTTPS), 75
template configuration (ICMP), 70
template configuration (RADIUS), 76
template configuration (SSL), 77
template configuration (TCP half open), 72
template configuration (TCP), 72
template configuration (UDP), 73
threshold monitoring, 8
Track collaboration function, 7

NSC
NetStream architecture, 370

NTP
access control, 82
architecture, 80
association mode configuration, 85
authentication, 83
authentication configuration, 89
broadcast association mode, 81
broadcast association mode configuration, 86, 103
broadcast mode authentication configuration, 92
broadcast mode dynamic associations max, 96
broadcast mode+authentication, 112
client/server association mode, 81
client/server association mode configuration, 85, 98
client/server mode authentication configuration, 89
client/server mode dynamic associations max, 96
client/server mode+authentication, 111
client/server mode+MPLS L3VPN network time synchronization, 115
configuration, 79, 84, 98
configuration restrictions, 84
display, 97
IPv6 client/server association mode configuration, 99
IPv6 multicast association mode configuration, 108
IPv6 symmetric active/passive association mode configuration, 102
local clock as reference source, 88
message receiving disable, 96
message source address specification, 95

MPLS L3VPN instance support, 83
multicast association mode, 81
multicast association mode configuration, 87, 105
multicast mode authentication configuration, 93
multicast mode dynamic associations max, 96
optional parameter configuration, 95
packet DSCP value setting, 97
protocols and standards, 84, 119
security, 82
SNTP authentication, 120
SNTP configuration, 84, 119, 122, 122
SNTP configuration restrictions, 119
symmetric active/passive association mode, 81
symmetric active/passive association mode configuration, 86, 100
symmetric active/passive mode authentication configuration, 90
symmetric active/passive mode dynamic associations max, 96
symmetric active/passive mode+MPLS L3VPN network time synchronization, 116

O

object
Event MIB monitored, 195
Event MIB object owner, 197
OC
PTP OC-type port configuration on a TC+OC clock, 138
operator
packet capture arithmetic, 432
packet capture logical, 432
packet capture relational, 432
ordinary
PTP clock node (OC), 124
outbound
port mirroring, 335
outputting
information center log configuration (console), 423
information center log configuration (Linux log host), 425
information center log default output rules, 407
information center logs configuration (UNIX log host), 424
information center synchronous log output, 418
information logs (console), 413
information logs (log host), 415
information logs (monitor terminal), 414
information logs to various destinations, 413

P

packet
flow mirroring configuration, 364, 368
flow mirroring QoS policy application, 366
flow mirroring traffic behavior, 365
Layer 3 remote port mirroring configuration (in ERSPAN mode), 350
Layer 3 remote port mirroring configuration (in tunnel mode), 347
NTP DSCP value setting, 97
packet capture display filter configuration (packet field expression), 439
port mirroring configuration, 335, 352
PTP packet DSCP value (IPv4 UDP), 143
sampler configuration, 333
sampler configuration (IPv4 NetStream), 333
sampler creation, 333
SNTP configuration, 84, 119, 122, 122
packet capture
capture filter keywords, 433
capture filter operator, 434
configuration, 432, 442
display, 442
display filter configuration, 436, 439
display filter configuration (logical expression), 439
display filter configuration (packet field expression), 439
display filter configuration (proto[…] expression), 439
display filter configuration (relational expression), 439
display filter keyword, 437
display filter operator, 438
feature image-based configuration, 440, 443
feature image-based file save, 441
feature image-based packet data display filter, 441, 441
file content display, 441
filter configuration, 433, 435
filter configuration (expr relop expr expression), 436
filter configuration (logical expression), 435
filter configuration (proto [exprsize] expression), 436
filter configuration (vlan vlan_id expression), 436
filter elements, 432
local configuration (wired device), 440
mode, 432

remote configuration, 442
remote configuration (wired device), 440
parameter
CWMP CPE ACS authentication, 302
NQA client history record save, 30
NQA client operation optional parameters, 26
NQA client template optional parameters, 44
NTP dynamic associations max, 96
NTP local clock as reference source, 88
NTP message receiving disable, 96
NTP message source address, 95
NTP optional parameter configuration, 95
SNMP common parameter configuration, 173
SNMPv3 group and user configuration in FIPS mode, 176
path
NQA client operation (path jitter), 24
NQA operation configuration, 66
pause
automated underlay network deployment, 457
peer
PTP Peer Delay, 128
performing
NETCONF CLI operations, 247, 248
periodic
IPv6 NetStream flow aging, 393
IPv6 NetStream flow aging (periodic), 387
NetStream flow aging configuration, 378
ping
address reachability determination, 1, 2
network connectivity test, 1
system maintenance, 1
PMM
3rd party process start, 328
3rd party process stop, 328
display, 328
kernel thread deadloop detection, 330
kernel thread maintain, 330
kernel thread monitoring, 330
kernel thread starvation detection, 331
Linux kernel thread, 327
Linux network, 327
Linux user, 327
monitor, 328
user PMM display, 329
user PMM maintain, 329
user PMM monitor, 329
policy
CWMP ACS HTTPS SSL client policy, 302
EAA configuration, 314, 321
EAA environment variable configuration (user-defined), 317
EAA event monitor policy configuration (CLI), 322
EAA event monitor policy configuration (Track), 323
EAA event monitor policy element, 315
EAA event monitor policy environment variable, 316
EAA monitor policy, 315
EAA monitor policy configuration, 318
EAA monitor policy configuration (CLI-defined+environment variables), 325
EAA monitor policy configuration (Tcl-defined), 321
EAA monitor policy suspension, 320
flow mirroring QoS policy application, 366
port
IPv6 NTP client/server association mode, 99
IPv6 NTP multicast association mode, 108
IPv6 NTP symmetric active/passive association mode, 102
mirroring. *See* port mirroring
NTP association mode, 85
NTP broadcast association mode, 103
NTP broadcast mode+authentication, 112
NTP client/server association mode, 98
NTP client/server mode+authentication, 111
NTP client/server mode+MPLS L3VPN network time synchronization, 115
NTP configuration, 79, 84, 98
NTP multicast association mode, 105
NTP symmetric active/passive association mode, 100
NTP symmetric active/passive mode+MPLS L3VPN network time synchronization, 116
PTP configuration, 124, 147
PTP configuration (AES67-2015, IPv4 UDP transport, multicast transmission), 166
PTP configuration (IEEE 1588 v2, IEEE 802.3/Ethernet transport, multicast transmission), 147
PTP configuration (IEEE 1588 v2, IPv4 UDP transport, multicast transmission), 150
PTP configuration (IEEE 1588 v2, IPv4 UDP transport, unicast transmission), 153
PTP configuration (IEEE 802.1AS, IEEE 802.3/Ethernet transport, multicast transmission), 156
PTP configuration (SMPTE ST 2059-2, IPv4 UDP transport, multicast transmission), 159
PTP configuration (SMPTE ST 2059-2, IPv4 UDP transport, unicast transmission), 163

PTP OC-type port configuration on a TC+OC clock, 138
PTP port enable, 135
PTP port role, 136
SNTP configuration, 84, 119, 122, 122
port mirroring
classification, 336
configuration, 335, 352
configuration restrictions, 339
display, 352
Layer 2 remote configuration, 341
Layer 2 remote port mirroring, 336
Layer 2 remote port mirroring configuration (egress port), 357
Layer 2 remote port mirroring configuration (reflector port configurable), 355
Layer 2 remote port mirroring configuration restrictions, 341
Layer 2 remote port mirroring egress port configuration restrictions, 346
Layer 2 remote port mirroring reflector port configuration restrictions, 345
Layer 2 remote port mirroring remote destination group configuration restrictions, 343
Layer 2 remote port mirroring remote probe VLAN configuration restrictions, 343, 344, 344
Layer 2 remote port mirroring source port configuration restrictions, 344
Layer 3 remote configuration (in ERSPAN mode), 350
Layer 3 remote configuration (in tunnel mode), 347
Layer 3 remote port mirroring, 338
Layer 3 remote port mirroring configuration, 359
Layer 3 remote port mirroring configuration (in ERSPAN mode), 361
Layer 3 remote port mirroring in tunnel mode configuration restrictions, 347
Layer 3 remote port mirroring local mirroring group monitor port configuration restrictions, 349, 351
local configuration, 339
local group creation, 340
local group monitor port, 341
local group monitor port configuration restrictions, 341
local group source CPU, 340
local group source port, 340
local mirroring configuration (source CPU mode), 353
local mirroring configuration (source port mode), 352
local port mirroring, 336
mirroring source configuration, 340, 348, 350
monitor port to remote probe VLAN assignment, 344
remote probe VLAN, 343
remote destination group creation, 342
remote destination group monitor port, 343
remote source group creation, 344
terminology, 335
Precision Time Protocol. *Use* PTP
preprovisioning
NETCONF enable, 246
private
RMON private alarm group, 187
procedure
applying flow mirroring QoS policy, 366
applying flow mirroring QoS policy (control plane), 368
applying flow mirroring QoS policy (global), 367
applying flow mirroring QoS policy (interface), 366
applying flow mirroring QoS policy (VLAN), 367
assigning CWMP ACS attribute (preferred)(CLI), 301
assigning CWMP ACS attribute (preferred)(DHCP server), 300
canceling subscription to NETCONF module report event, 252
configuring a Puppet agent, 269
configuring border node, 461
configuring Chef, 284, 284
configuring Chef client, 283
configuring Chef server, 283
configuring Chef workstation, 283
configuring CWMP, 299, 305
configuring CWMP ACS attribute, 300
configuring CWMP ACS attribute (default)(CLI), 301
configuring CWMP ACS attribute (preferred), 300
configuring CWMP ACS autoconnect parameters, 304
configuring CWMP ACS close-wait timer, 305
configuring CWMP ACS connection retry max number, 304
configuring CWMP ACS periodic Inform feature, 304
configuring CWMP CPE ACS authentication parameters, 302
configuring CWMP CPE ACS connection interface, 303
configuring CWMP CPE ACS provision code, 303

configuring CWMP CPE attribute, 302
configuring EAA environment variable (user-defined), 317
configuring EAA event monitor policy (CLI), 322
configuring EAA event monitor policy (Track), 323
configuring EAA monitor policy, 318
configuring EAA monitor policy (CLI-defined+environment variables), 325
configuring EAA monitor policy (Tcl-defined), 321
configuring Event MIB, 197
configuring Event MIB event, 198
configuring Event MIB trigger test, 200
configuring Event MIB trigger test (Boolean), 206
configuring Event MIB trigger test (existence), 204
configuring Event MIB trigger test (threshold), 202, 209
configuring feature image-based packet capture, 440
configuring flow mirroring, 368
configuring flow mirroring traffic behavior, 365
configuring flow mirroring traffic class, 365
configuring GOLD, 430
configuring GOLD (centralized IRF devices), 430
configuring GOLD diagnostics (monitoring), 427
configuring GOLD diagnostics (on-demand), 428
configuring GOLD log buffer size, 429
configuring information center, 411
configuring information center log output (console), 423
configuring information center log output (Linux log host), 425
configuring information center log output (UNIX log host), 424
configuring information center log suppression, 418
configuring information center log suppression for module, 419
configuring information center trace log file max size, 422
configuring IPv6 NetStream, 390
configuring IPv6 NetStream data export, 394
configuring IPv6 NetStream data export (aggregation), 394, 397
configuring IPv6 NetStream data export (traditional), 394, 396
configuring IPv6 NetStream data export format, 391
configuring IPv6 NetStream filtering, 390
configuring IPv6 NetStream flow aging, 393
configuring IPv6 NetStream flow aging (periodic), 393
configuring IPv6 NetStream sampling, 391
configuring IPv6 NetStream v9/v10 template refresh rate, 393
configuring IPv6 NTP client/server association mode, 99
configuring IPv6 NTP multicast association mode, 108
configuring IPv6 NTP symmetric active/passive association mode, 102
configuring Layer 2 remote port mirroring, 341
configuring Layer 2 remote port mirroring (egress port), 357
configuring Layer 2 remote port mirroring (reflector port configurable), 355
configuring Layer 3 remote port mirroring, 359
configuring Layer 3 remote port mirroring (in ERSPAN mode), 350, 361
configuring Layer 3 remote port mirroring (in tunnel mode), 347
configuring Layer 3 remote port mirroring local group, 348
configuring Layer 3 remote port mirroring local group source port, 351
configuring Layer 3 remote port mirroring local mirroring group monitor port, 349, 351
configuring Layer 3 remote port mirroring local mirroring group source CPU, 349, 351
configuring local packet capture (wired device), 440
configuring local port mirroring, 339
configuring local port mirroring (source CPU mode), 353
configuring local port mirroring (source port mode), 352
configuring local port mirroring group monitor port, 341
configuring local port mirroring group source CPUs, 340
configuring local port mirroring group source ports, 340
configuring MAC address of VSI interfaces, 462
configuring master spinde node, 457
configuring mirroring sources, 340, 348, 350
configuring NETCONF, 214
configuring NetStream, 375
configuring NetStream data export, 379

configuring NetStream data export (aggregation), 379, 383
configuring NetStream data export (traditional), 379, 381
configuring NetStream data export format, 376
configuring NetStream filtering, 375
configuring NetStream flow aging, 378
configuring NetStream flow aging (forced), 379, 393
configuring NetStream flow aging (periodic), 378
configuring NetStream sampling, 376
configuring NetStream v9/v10 template refresh rate, 378
configuring NQA, 8
configuring NQA client history record save, 30
configuring NQA client operation, 9
configuring NQA client operation (DHCP), 12
configuring NQA client operation (DLSw), 24
configuring NQA client operation (DNS), 13
configuring NQA client operation (FTP), 14
configuring NQA client operation (HTTP), 15
configuring NQA client operation (ICMP echo), 10
configuring NQA client operation (ICMP jitter), 11
configuring NQA client operation (path jitter), 24
configuring NQA client operation (SNMP), 18
configuring NQA client operation (TCP), 18
configuring NQA client operation (UDP echo), 19
configuring NQA client operation (UDP jitter), 16
configuring NQA client operation (UDP tracert), 20
configuring NQA client operation (voice), 22
configuring NQA client operation optional parameters, 26
configuring NQA client statistics collection, 29
configuring NQA client template, 31
configuring NQA client template (DNS), 33
configuring NQA client template (FTP), 41
configuring NQA client template (HTTP), 38
configuring NQA client template (HTTPS), 39
configuring NQA client template (ICMP), 32
configuring NQA client template (RADIUS), 42
configuring NQA client template (SSL), 43
configuring NQA client template (TCP half open), 35
configuring NQA client template (TCP), 34
configuring NQA client template (UDP), 36
configuring NQA client template optional parameters, 44
configuring NQA client threshold monitoring, 27
configuring NQA client+Track collaboration, 27
configuring NQA collaboration, 68
configuring NQA operation (DHCP), 50
configuring NQA operation (DLSw), 65
configuring NQA operation (DNS), 51
configuring NQA operation (FTP), 52
configuring NQA operation (HTTP), 53
configuring NQA operation (ICMP echo), 46
configuring NQA operation (ICMP jitter), 47
configuring NQA operation (path jitter), 66
configuring NQA operation (SNMP), 57
configuring NQA operation (TCP), 58
configuring NQA operation (UDP echo), 59
configuring NQA operation (UDP jitter), 54
configuring NQA operation (UDP tracert), 61
configuring NQA operation (voice), 62
configuring NQA server, 9
configuring NQA template (DNS), 71
configuring NQA template (FTP), 75
configuring NQA template (HTTP), 74
configuring NQA template (HTTPS), 75
configuring NQA template (ICMP), 70
configuring NQA template (RADIUS), 76
configuring NQA template (SSL), 77
configuring NQA template (TCP half open), 72
configuring NQA template (TCP), 72
configuring NQA template (UDP), 73
configuring NTP, 84
configuring NTP association mode, 85
configuring NTP broadcast association mode, 86, 103
configuring NTP broadcast mode authentication, 92
configuring NTP broadcast mode+authentication, 112
configuring NTP client/server association mode, 85, 98
configuring NTP client/server mode authentication, 89
configuring NTP client/server mode+authentication, 111
configuring NTP client/server mode+MPLS L3VPN network time synchronization, 115
configuring NTP dynamic associations max, 96
configuring NTP local clock as reference source, 88
configuring NTP multicast association mode, 87, 105

configuring NTP multicast mode authentication, 93
configuring NTP optional parameters, 95
configuring NTP symmetric active/passive association mode, 86, 100
configuring NTP symmetric active/passive mode authentication, 90
configuring NTP symmetric active/passive mode+MPLS L3VPN network time synchronization, 116
configuring OC-type port on a TC+OC clock, 138
configuring packet capture (feature image-based), 443
configuring PMM kernel thread deadloop detection, 330
configuring PMM kernel thread starvation detection, 331
configuring port mirroring monitor port to remote probe VLAN assignment, 344
configuring port mirroring remote destination group monitor port, 343
configuring port mirroring remote probe VLAN, 343
configuring port mirroring remote source group egress port, 346
configuring port mirroring remote source group reflector port, 345
configuring port mirroring remote source group source CPU, 345
configuring port mirroring remote source group source ports, 344
configuring PTP (AES67-2015, IPv4 UDP transport, multicast transmission), 166
configuring PTP (IEEE 1588 v2, IEEE 802.3/Ethernet transport, multicast transmission), 147
configuring PTP (IEEE 1588 v2, IPv4 UDP transport, multicast transmission), 150
configuring PTP (IEEE 1588 v2, IPv4 UDP transport, unicast transmission), 153
configuring PTP (IEEE 802.1AS, IEEE 802.3/Ethernet transport, multicast transmission), 156
configuring PTP (SMPTE ST 2059-2, IPv4 UDP transport, multicast transmission), 159
configuring PTP (SMPTE ST 2059-2, IPv4 UDP transport, unicast transmission), 163
configuring PTP clock priority, 145
configuring PTP delay measurement mechanism, 137
configuring PTP multicast message source IP address (UDP), 141
configuring PTP non-Pdelay message MAC address, 142
configuring PTP OC as member clock, 134
configuring PTP port role, 136
configuring PTP system time source, 133
configuring PTP timestamp carry mode, 137
configuring PTP unicast message destination IP address (IPv4 UDP), 142
configuring PTP UTC correction date, 145
configuring Puppet, 270, 270
configuring RabbiMQ server communication parameters, 458
configuring remote packet capture, 442
configuring remote packet capture (wired device), 440
configuring resources, 269
configuring RMON alarm, 189, 192
configuring RMON Ethernet statistics group, 191
configuring RMON history group, 191
configuring RMON statistics, 188
configuring sampler (IPv4 NetStream), 333
configuring sFlow, 403, 403
configuring sFlow agent+collector information, 401
configuring sFlow counter sampling, 402
configuring sFlow flow sampling, 401
configuring SNMP common parameters, 173
configuring SNMP logging, 180
configuring SNMP notification, 177
configuring SNMPv1, 181
configuring SNMPv1 community, 174, 174
configuring SNMPv1 community by community name, 174
configuring SNMPv1 host notification send, 178
configuring SNMPv2c, 181
configuring SNMPv2c community, 174, 174
configuring SNMPv2c community by community name, 174
configuring SNMPv2c host notification send, 178
configuring SNMPv3, 183
configuring SNMPv3 group and user, 175
configuring SNMPv3 group and user in FIPS mode, 176
configuring SNMPv3 group and user in non-FIPS mode, 176
configuring SNMPv3 host notification send, 178
configuring SNTP, 84, 122, 122
configuring SNTP authentication, 120
configuring VCF fabric, 454
configuring VCF fabric automated underlay network deployment, 456, 457
configuring VXLAN-aware NetStream, 378

creating Layer 3 remote port mirroring local group, 350
creating local port mirroring group, 340
creating port mirroring remote destination group on the destination device, 342
creating port mirroring remote source group on the source device, 344
creating RMON Ethernet statistics entry, 188
creating RMON history control entry, 188
creating sampler, 333
debugging feature module, 6
determining ping address reachability, 2
disabling information center interface link up/link down log generation, 419
disabling NTP message interface receiving, 96
displaying CWMP settings, 305
displaying EAA settings, 321
displaying Event MIB, 204
displaying GOLD, 429
displaying information center, 423
displaying IPv6 NetStream, 395
displaying NetStream, 380
displaying NMM sFlow, 402
displaying NQA, 45
displaying NTP, 97
displaying packet capture, 442
displaying packet file content, 441
displaying PMM, 328
displaying PMM kernel threads, 331
displaying PMM user processes, 330
displaying port mirroring, 352
displaying RMON settings, 190
displaying sampler, 333
displaying SNMP settings, 181
displaying SNTP, 121
displaying user PMM, 329
displaying VCF fabric, 462
enabling CWMP, 300
enabling Event MIB SNMP notification, 203
enabling information center, 413
enabling information center duplicate log suppression, 418
enabling information center synchronous log output, 418
enabling information center system log SNMP notification, 420
enabling L2 agent, 460
enabling L3 agent, 460
enabling local proxy ARP, 461
enabling NETCONF preprovisioning, 246
enabling NQA client, 9
enabling PTP on port, 135
enabling SNMP agent, 172
enabling SNMP notification, 177
enabling SNMP version, 172
enabling SNTP, 119
enabling VCF fabric topology discovery, 456
establishing NETCONF over console sessions, 218
establishing NETCONF over SOAP sessions, 217
establishing NETCONF over SSH sessions, 218
establishing NETCONF over Telnet sessions, 218
establishing NETCONF session, 215
exchanging NETCONF capabilities, 219
filtering feature image-based packet capture data display, 441, 441
filtering NETCONF data, 229
filtering NETCONF data (conditional match), 234
filtering NETCONF data (regex match), 233
identifying tracert node failure, 4, 4
loading NETCONF configuration, 241
locking NETCONF running configuration, 235, 236
maintaining GOLD, 429
maintaining information center, 423
maintaining IPv6 NetStream, 395
maintaining NetStream, 380
maintaining PMM kernel thread, 330
maintaining PMM kernel threads, 331
maintaining PMM user processes, 330
maintaining user PMM, 329
managing information center security log, 420
managing information center security log file, 421
modifying NETCONF configuration, 237, 238
monitoring PMM, 328
monitoring PMM kernel thread, 330
monitoring user PMM, 329
outputting information center logs (console), 413
outputting information center logs (log host), 415
outputting information center logs (monitor terminal), 414
outputting information center logs to various destinations, 413
pausing underlay network deployment, 457
performing NETCONF CLI operations, 247, 248
retrieving device configuration information, 220
retrieving NETCONF configuration data (all modules), 226
retrieving NETCONF configuration data (Syslog module), 227

retrieving NETCONF data entry (interface table), 225
retrieving NETCONF information, 223
retrieving NETCONF non-default settings, 222
retrieving NETCONF session information, 224, 228
retrieving NETCONF YANG file content information, 224
returning to NETCONF CLI, 255
rolling back NETCONF configuration, 242
rolling back NETCONF configuration (configuration file-based), 242
rolling back NETCONF configuration (rollback point-based), 242
saving feature image-based packet capture to file, 441
saving information center diagnostic logs (log file), 422
saving information center log (log file), 416
saving information center security logs (log file), 420
saving NETCONF configuration, 239
saving NETCONF running configuration, 240
scheduling CWMP ACS connection initiation, 304
scheduling NQA client operation, 31
setting information center log storage period (log buffer), 417
setting NETCONF session attribute, 215
setting NTP packet DSCP value, 97
setting PTP announce message interval+timeout, 138
setting PTP cumulative offset (UTC:TAI), 144
setting PTP delay correction value, 144
setting PTP packet DSCP value (IPv4 UDP), 143
shutting down Chef, 284
shutting down Puppet (on device), 269
signing a certificate for Puppet agent, 269
simulating GOLD diagnostic tests, 429
specifying automated underlay network deployment template file, 456
specifying CWMP ACS HTTPS SSL client policy, 302
specifying IPv4 UDP transport protocol for PTP message, 141
specifying NTP message source address, 95
specifying overlay network type, 459
specifying PTP clock node type, 134
specifying PTP domain, 135
specifying PTP profile, 133, 133
specifying VCF fabric automated underlay network device role, 456
starting Chef, 283
starting PMM 3rd party process, 328
starting Puppet, 269
stopping PMM 3rd party process, 328
subscribing to NETCONF events, 249, 253
subscribing to NETCONF module report event, 251
subscribing to NETCONF monitoring event, 250
subscribing to NETCONF syslog event, 249
suspending EAA monitor policy, 320
terminating NETCONF session, 254
testing network connectivity with ping, 1
troubleshooting sFlow remote collector cannot receive packets, 404
unlocking NETCONF running configuration, 235, 236
process
monitoring and maintenance. *See* PMM
profile
PTP, 133, 133
PTP profile, 124
protocols and standards
IPv4 UDP transport protocol for PTP messages, 141
IPv6 NetStream, 389
NETCONF, 212, 214
NetStream, 374
NTP, 84, 119
packet capture display filter keyword, 437
PTP, 129
RMON, 188
sFlow, 400
SNMP configuration, 170, 181
SNMP versions, 171
provision code (ACS), 303
provisioning
NETCONF preprovisioning enable, 246
PTP
announce message interval+timeout, 138
basic concepts, 124
BC delay measurement, 137
clock node, 124
clock node type, 134
clock priority configuration, 145
clock type, 126
configuration, 124, 147
configuration (AES67-2015, IPv4 UDP transport, multicast transmission), 166
configuration (IEEE 1588 v2, IEEE 802.3/Ethernet transport, multicast transmission), 147

configuration (IEEE 1588 v2, IPv4 UDP transport, multicast transmission), 150
configuration (IEEE 1588 v2, IPv4 UDP transport, unicast transmission), 153
configuration (IEEE 802.1AS, IEEE 802.3/Ethernet transport, multicast transmission), 156
configuration (SMPTE ST 2059-2, IPv4 UDP transport, multicast transmission), 159
configuration (SMPTE ST 2059-2, IPv4 UDP transport, unicast transmission), 163
cumulative offset (UTC:TAI), 144
delay correction value, 144
domain, 124
domain specification, 135
grandmaster clock, 126
IEEE 1588 v2 profile, 124
IEEE 802.1AS profile, 124
IPv4 UDP transport protocol configuration, 141
master-member/subordinate relationship, 126
multicast message source IP address configuration (UDP), 141
non-Pdelay message MAC address, 142
OC configuration, 134
OC delay measurement, 137
OC-type port configuration on a TC+OC clock, 138
packet DSCP value configuration (IPv4 UDP), 143
Peer Delay, 128
port enable, 135
port role configuration, 136
profile specification, 133, 133
protocols and standards, 129
Request_Response, 127
synchronization, 127
system time source, 133
timestamp mode configuration, 137
unicast message destination IP address configuration (IPv4 UDP), 142
UTC correction date, 145
Puppet
configuration, 267, 270, 270
configuring a Puppet agent, 269
configuring resources, 269
network framework, 267
Puppet agent certificate signing, 269
resources, 268, 271
resources (netdev_device), 271
resources (netdev_interface), 272
resources (netdev_l2_interface), 273
resources (netdev_lagg), 274
resources (netdev_vlan), 276
resources (netdev_vsi), 276
resources (netdev_vte), 277
resources (netdev_vxlan), 278
shutting down (on device), 269
start, 269

Q

QoS
flow mirroring configuration, 364, 368
flow mirroring QoS policy application, 366

R

RADIUS
NQA client template, 42
NQA template configuration, 76
random mode (NMM sampler), 333
real-time
event manager. *See* RTM
reflector port
Layer 2 remote port mirroring, 335
port mirroring remote source group reflector port, 345
refreshing
IPv6 NetStream v9/v10 template refresh rate, 393
NetStream v9/v10 template refresh rate, 378
regex match
NETCONF data filtering, 233
NETCONF data filtering (column-based), 231
regular expression. *Use* regex
relational
packet capture display filter configuration (relational expression), 439
packet capture operator, 432
remote
Layer 2 remote port mirroring, 341
Layer 3 port mirroring local group, 348, 350
Layer 3 port mirroring local group monitor port, 349, 351
Layer 3 port mirroring local group source CPU, 349, 351
Layer 3 port mirroring local group source port, 351
Layer 3 remote port mirroring configuration (in ERSPAN mode), 350
Layer 3 remote port mirroring configuration (in tunnel mode), 347
packet capture configuration, 442
packet capture configuration (wired device), 440
packet capture mode, 432
port mirroring destination group, 342
port mirroring destination group monitor port, 343

port mirroring destination group remote probe VLAN, 343
port mirroring monitor port to remote probe VLAN assignment, 344
port mirroring source group, 344
port mirroring source group egress port, 346
port mirroring source group reflector port, 345
port mirroring source group remote probe VLAN, 343
port mirroring source group source CPU, 345
port mirroring source group source ports, 344
Remote Network Monitoring. *Use* RMON
remote probe VLAN
Layer 2 remote port mirroring, 335
port mirroring monitor port to remote probe VLAN assignment, 344
port mirroring remote destination group, 343
port mirroring remote source group, 343
reporting
NETCONF module report event subscription, 251
NETCONF module report event subscription cancel, 252
Request_Response mechanism (PTP), 127
resource
Chef, 281, 287
Chef netdev_device, 287
Chef netdev_interface, 287
Chef netdev_l2_interface, 289
Chef netdev_lagg, 290
Chef netdev_vlan, 291
Chef netdev_vsi, 291
Chef netdev_vte, 292
Chef netdev_vxlan, 293
Puppet, 268, 271
Puppet netdev_device, 271
Puppet netdev_interface, 272
Puppet netdev_l2_interface, 273
Puppet netdev_lagg, 274
Puppet netdev_vlan, 276
Puppet netdev_vsi, 276
Puppet netdev_vte, 277
Puppet netdev_vxlan, 278
restrictions
EAA monitor policy configuration, 318
EAA monitor policy configuration (Tcl), 320
IPv6 NetStream data export configuration, 394
IPv6 NetStream filtering configuration, 390
Layer 2 remote port configuration, 341
Layer 2 remote port mirroring egress port configuration, 346
Layer 2 remote port mirroring reflector port configuration, 345
Layer 2 remote port mirroring remote destination group configuration, 343
Layer 2 remote port mirroring remote probe VLAN configuration, 343, 344, 344
Layer 2 remote port mirroring source port configuration, 344
Layer 3 remote port mirroring in tunnel mode configuration, 347
Layer 3 remote port mirroring local group monitor port configuration, 349, 351
local port mirroring group monitor port configuration, 341
NETCONF session establishment, 215
NetStream data export (aggregation), 380
NetStream sampling configuration, 376
NQA client history record save, 30
NQA client operation (FTP), 14
NQA client operation (ICMP jitter), 12
NQA client operation (UDP tracert), 20
NQA client operation optional parameter configuration, 26
NQA client operation scheduling, 31
NQA client statistics collection, 29
NQA client template configuration, 31
NQA client template optional parameter configuration, 44
NQA client threshold monitoring configuration, 28
NQA client+Track collaboration, 27
NQA server configuration, 9
NTP configuration, 84
port mirroring configuration, 339
RMON alarm configuration, 189
RMON history control entry creation, 188
SNMPv1 community configuration, 174
SNMPv2 community configuration, 174
SNMPv3 group and user configuration, 175
SNTP configuration, 84
SNTP configuration restrictions, 119
retrieivng
device configuration information, 220
retrieving
NETCONF configuration data (all modules), 226
NETCONF configuration data (Syslog module), 227
NETCONF data entry (interface table), 225
NETCONF device configuration+state information, 220
NETCONF information, 223

NETCONF non-default settings, 222
NETCONF session information, 224, 228
NETCONF YANG file content, 224
returning
NETCONF CLI return, 255
RMON
alarm configuration, 189, 192
alarm configuration restrictions, 189
alarm group, 187
alarm group sample types, 188
configuration, 186, 191
Ethernet statistics entry creation, 188
Ethernet statistics group, 186
Ethernet statistics group configuration, 191
event group, 186
Event MIB configuration, 195, 197, 204
Event MIB event configuration, 198
Event MIB trigger test configuration (Boolean), 206
Event MIB trigger test configuration (existence), 204
Event MIB trigger test configuration (threshold), 209
group, 186
history control entry creation, 188
history control entry creation restrictions, 188
history group, 186
history group configuration, 191
how it works, 186
private alarm group, 187
protocols and standards, 188
settings display, 190
statistics configuration, 188
statistics function, 188
role
PTP port, 136
rolling back
NETCONF configuration, 242
NETCONF configuration (configuration file-based), 242
NETCONF configuration (rollback point-based), 242
routing
IPv6 NTP client/server association mode, 99
IPv6 NTP multicast association mode, 108
IPv6 NTP symmetric active/passive association mode, 102
NTP association mode, 85
NTP broadcast association mode, 103
NTP broadcast mode+authentication, 112
NTP client/server association mode, 98
NTP client/server mode+authentication, 111
NTP client/server mode+MPLS L3VPN network time synchronization, 115
NTP configuration, 79, 84, 98
NTP multicast association mode, 105
NTP symmetric active/passive association mode, 100
NTP symmetric active/passive mode+MPLS L3VPN network time synchronization, 116
PTP configuration, 124, 147
PTP configuration (AES67-2015, IPv4 UDP transport, multicast transmission), 166
PTP configuration (IEEE 1588 v2, IEEE 802.3/Ethernet transport, multicast transmission), 147
PTP configuration (IEEE 1588 v2, IPv4 UDP transport, multicast transmission), 150
PTP configuration (IEEE 1588 v2, IPv4 UDP transport, unicast transmission), 153
PTP configuration (IEEE 802.1AS, IEEE 802.3/Ethernet transport, multicast transmission), 156
PTP configuration (SMPTE ST 2059-2, IPv4 UDP transport, multicast transmission), 159
PTP configuration (SMPTE ST 2059-2, IPv4 UDP transport, unicast transmission), 163
SNTP configuration, 84, 119, 122, 122
RPC
CWMP RPC methods, 297
RTM
EAA, 314
EAA configuration, 314, 321
Ruby
Chef configuration, 280, 284, 284
Chef resources, 281
rule
information center log default output rules, 407
SNMP access control (rule-based), 171
system information default output rules (diagnostic log), 407
system information default output rules (hidden log), 408
system information default output rules (security log), 407
system information default output rules (trace log), 408
runtime
EAA event monitor policy runtime, 316

S

sampler
configuration, 333
configuration (IPv4 NetStream), 333

creation, 333
display, 333
sampling
IPv6 NetStream, 389
IPv6 NetStream configuration, 390
IPv6 NetStream sampling configuration, 391
NetStream configuration, 370, 375, 381
NetStream sampling, 374
NetStream sampling configuration, 376
Sampled Flow. *Use* sFlow
sFlow counter sampling, 402
sFlow flow sampling configuration, 401
saving
feature image-based packet capture to file, 441
information center diagnostic logs (log file), 422
information center log (log file), 416
information center security logs (log file), 420
NETCONF configuration, 239
NETCONF running configuration, 240
NQA client history records, 30
scheduling
CWMP ACS connection initiation, 304
NQA client operation, 31
security
information center security log file management, 421
information center security log management, 420
information center security log save (log file), 420
information center security logs, 406
NTP, 82
NTP authentication, 83, 89
NTP broadcast mode authentication, 92
NTP client/server mode authentication, 89
NTP multicast mode authentication, 93
NTP symmetric active/passive mode authentication, 90
SNTP authentication, 120
server
Chef server configuration, 283
NQA configuration, 9
SNTP configuration, 84, 119, 122, 122
service
NETCONF configuration data retrieval (all modules), 226
NETCONF configuration data retrieval (Syslog module), 227
NETCONF configuration modification, 238
session
NETCONF session attribute, 215
NETCONF session establishment, 215
NETCONF session information retrieval, 224, 228
NETCONF session termination, 254
sessions
NETCONF over console session establishment, 218
NETCONF over SOAP session establishment, 217
NETCONF over SSH session establishment, 218
NETCONF over Telnet session establishment, 218
set operation
SNMP, 171
SNMP logging, 180
setting
information center log storage period (log buffer), 417
NETCONF session attribute, 215
NTP packet DSCP value, 97
PTP announce message interval+timeout, 138
PTP cumulative offset (UTC:TAI), 144
PTP delay correction value, 144
PTP packet DSCP value (UDP), 143
severity level (system information), 406
sFlow
agent+collector information configuration, 401
configuration, 400, 403, 403
counter sampling configuration, 402
display, 402
flow sampling configuration, 401
protocols and standards, 400
troubleshoot, 404
troubleshoot remote collector cannot receive packets, 404
shutting down
Chef, 284
Puppet (on device), 269
Simple Network Management Protocol. *Use* SNMP
Simplified NTP. *See* SNTP
simulating
GOLD diagnostic test simulation, 429
SNMP
access control mode, 171
agent, 170
agent enable, 172
agent notification, 177
common parameter configuration, 173
configuration, 170, 181
Event MIB configuration, 195, 197, 204

Event MIB display, 204
Event MIB event configuration, 198
Event MIB SNMP notification enable, 203
Event MIB trigger test configuration, 200
Event MIB trigger test configuration (Boolean), 206
Event MIB trigger test configuration (existence), 204
Event MIB trigger test configuration (threshold), 202, 209
FIPS compliance, 171
framework, 170
get operation, 180
Get operation, 171
host notification send, 178
information center system log SNMP notification, 420
logging configuration, 180
manager, 170
MIB, 170, 170
MIB view-based access control, 170
notification configuration, 177
notification enable, 177
Notification operation, 171
NQA client operation, 18
NQA operation configuration, 57
protocol versions, 171
RMON configuration, 186, 191
set operation, 180
Set operation, 171
settings display, 181
SNMPv1 community configuration, 174, 174
SNMPv1 community configuration by community name, 174
SNMPv1 community configuration by creating SNMPv1 user, 175
SNMPv1 configuration, 181
SNMPv2c community configuration, 174, 174
SNMPv2c community configuration by community name, 174
SNMPv2c community configuration by creating SNMPv2c user, 175
SNMPv2c configuration, 181
SNMPv3 configuration, 183
SNMPv3 group and user configuration, 175
SNMPv3 group and user configuration in FIPS mode, 176
SNMPv3 group and user configuration in non-FIPS mode, 176
version enable, 172
SNMPv1
community configuration, 174, 174
community configuration restrictions, 174
configuration, 181
host notification send, 178
Notification operation, 171
protocol version, 171
SNMPv2
community configuration restrictions, 174
SNMPv2c
community configuration, 174, 174
configuration, 181
host notification send, 178
Notification operation, 171
protocol version, 171
SNMPv3
configuration, 183
Event MIB object owner, 197
group and user configuration, 175
group and user configuration in FIPS mode, 176
group and user configuration in non-FIPS mode, 176
group and user configuration restrictions, 175
Notification operation, 171
notification send, 178
protocol version, 171
SNTP
authentication, 120
configuration, 84, 119, 122, 122
configuration restrictions, 84, 119
display, 121
enable, 119
SOAP
NETCONF message format, 212
NETCONF over SOAP session establishment, 217
source
port mirroring source, 335
port mirroring source device, 335
specify
master spine node, 457
VCF fabric automated underlay network deployment device role, 456
VCF fabric automated underlay network deployment template file, 456
VCF fabric overlay network type, 459
specifying
CWMP ACS HTTPS SSL client policy, 302
IPv4 UDP transport protocol for PTP messages, 141
NTP message source address, 95
PTP BC delay measurement, 137
PTP clock node type, 134

PTP domain, 135
PTP OC delay measurement, 137
PTP profile, 133, 133
SSH
Chef configuration, 280, 284, 284
NETCONF over SSH session establishment, 218
Puppet configuration, 267, 270, 270
SSL
CWMP ACS HTTPS SSL client policy, 302
NQA client template (SSL), 43
NQA template configuration, 77
starting
Chef, 283
PMM 3rd party process, 328
Puppet, 269
starvation detection (Linux kernel thread PMM), 331
statistics
IPv6 NetStream configuration, 386, 390, 396
IPv6 NetStream data export format, 388
IPv6 NetStream filtering, 389
IPv6 NetStream filtering configuration, 390
IPv6 NetStream sampling, 389
IPv6 NetStream sampling configuration, 391
NetStream configuration, 370, 375, 381
NetStream filtering, 374
NetStream filtering configuration, 375
NetStream sampling, 374
NetStream sampling configuration, 376
NQA client statistics collection, 29
RMON configuration, 186, 191
RMON Ethernet statistics entry, 188
RMON Ethernet statistics group, 186
RMON Ethernet statistics group configuration, 191
RMON history control entry, 188
RMON statistics configuration, 188
RMON statistics function, 188
sampler configuration, 333
sampler configuration (IPv4 NetStream), 333
sampler creation, 333
sFlow agent+collector information configuration, 401
sFlow configuration, 400, 403, 403
sFlow counter sampling configuration, 402
sFlow flow sampling configuration, 401
VXLAN-aware NetStream, 378
stopping
PMM 3rd party process, 328
storage
information center log storage period (log buffer), 417
subordinate
PTP master-member/subordinate relationship, 126
subscribing
NETCONF event subscription, 249, 253
NETCONF module report event subscription, 251
NETCONF monitoring event subscription, 250
NETCONF syslog event subscription, 249
suppressing
information center duplicate log suppression, 418
information center log suppression for module, 419
suspending
EAA monitor policy, 320
switch
module debug, 5
screen output, 5
symmetric
IPv6 NTP symmetric active/passive association mode, 102
NTP symmetric active/passive association mode, 81, 86, 90, 100
NTP symmetric active/passive mode dynamic associations max, 96
NTP symmetric active/passive mode+MPLS L3VPN network time synchronization, 116
synchronizing
information center synchronous log output, 418
NTP client/server mode+MPLS L3VPN network time synchronization, 115
NTP configuration, 79, 84, 98
NTP symmetric active/passive mode+MPLS L3VPN network time synchronization, 116
PTP, 127
PTP configuration, 124, 147
PTP configuration (AES67-2015, IPv4 UDP transport, multicast transmission), 166
PTP configuration (IEEE 1588 v2, IEEE 802.3/Ethernet transport, multicast transmission), 147
PTP configuration (IEEE 1588 v2, IPv4 UDP transport, multicast transmission), 150
PTP configuration (IEEE 1588 v2, IPv4 UDP transport, unicast transmission), 153
PTP configuration (IEEE 802.1AS, IEEE 802.3/Ethernet transport, multicast transmission), 156
PTP configuration (SMPTE ST 2059-2, IPv4 UDP transport, multicast transmission), 159
PTP configuration (SMPTE ST 2059-2, IPv4 UDP transport, unicast transmission), 163

PTP domain, 124
SNTP configuration, 84, 119, 122, 122
syslog
NETCONF configuration data retrieval (Syslog module), 227
NETCONF syslog event subscription, 249
system
default output rules (diagnostic log), 407
default output rules (hidden log), 408
default output rules (security log), 407
default output rules (trace log), 408
information center duplicate log suppression, 418
information center interface link up/link down log generation, 419
information center log destinations, 407
information center log levels, 406
information center log output (console), 413
information center log output (log host), 415
information center log output (monitor terminal), 414
information center log output configuration (console), 423
information center log output configuration (Linux log host), 425
information center log output configuration (UNIX log host), 424
information center log save (log file), 416
information center log types, 406
information center security log file management, 421
information center security log management, 420
information center security log save (log file), 420
information center synchronous log output, 418
information center system log SNMP notification, 420
information log formats and field descriptions, 408
log default output rules, 407
PTP system time source, 133
system administration
Chef configuration, 280, 284, 284
debugging, 1
feature module debug, 6
ping, 1
ping address reachability, 2
ping command, 1
ping network connectivity test, 1
Puppet configuration, 267, 270, 270
system debugging, 5
tracert, 1, 3
tracert node failure identification, 4, 4
system debugging
module debugging switch, 5
screen output switch, 5
system information
information center configuration, 406, 411, 423

T

table
NETCONF data entry retrieval (interface table), 225
TAI
PTP cumulative offset (UTC:TAI), 144
TC
PTP OC-type port configuration on a TC+OC clock, 138
Tcl
EAA configuration, 314, 321
EAA monitor policy configuration, 321
TCP
NQA client operation, 18
NQA client template, 34
NQA client template (TCP half open), 35
NQA operation configuration, 58
NQA template configuration, 72
NQA template configuration (half open), 72
Telnet
NETCONF over Telnet session establishment, 218
template
NetStream v9/v10 template refresh rate, 378
NQA, 8
NQA client template (DNS), 33
NQA client template (FTP), 41
NQA client template (HTTP), 38
NQA client template (HTTPS), 39
NQA client template (ICMP), 32
NQA client template (RADIUS), 42
NQA client template (SSL), 43
NQA client template (TCP half open), 35
NQA client template (TCP), 34
NQA client template (UDP), 36
NQA client template configuration, 31
NQA client template optional parameters, 44
NQA template configuration (DNS), 71
NQA template configuration (FTP), 75
NQA template configuration (HTTP), 74
NQA template configuration (HTTPS), 75
NQA template configuration (ICMP), 70

NQA template configuration (RADIUS), 76
NQA template configuration (SSL), 77
NQA template configuration (TCP half open), 72
NQA template configuration (TCP), 72
NQA template configuration (UDP), 73
template file
automated underlay network deployment, 453
VCF fabric automated underlay network deployment configuration, 456
terminating
NETCONF session, 254
testing
Event MIB trigger test configuration, 200
Event MIB trigger test configuration (Boolean), 206
Event MIB trigger test configuration (existence), 204
Event MIB trigger test configuration (threshold), 202, 209
GOLD diagnostic test simulation, 429
ping network connectivity test, 1
threshold
Event MIB trigger test, 196
Event MIB trigger test configuration, 202, 209
NQA client threshold monitoring, 8, 27
time
NTP configuration, 79, 84, 98
NTP local clock as reference source, 88
PTP clock priority, 145
PTP configuration, 124, 147
PTP configuration (AES67-2015, IPv4 UDP transport, multicast transmission), 166
PTP configuration (IEEE 1588 v2, IEEE 802.3/Ethernet transport, multicast transmission), 147
PTP configuration (IEEE 1588 v2, IPv4 UDP transport, multicast transmission), 150
PTP configuration (IEEE 1588 v2, IPv4 UDP transport, unicast transmission), 153
PTP configuration (IEEE 802.1AS, IEEE 802.3/Ethernet transport, multicast transmission), 156
PTP configuration (SMPTE ST 2059-2, IPv4 UDP transport, multicast transmission), 159
PTP configuration (SMPTE ST 2059-2, IPv4 UDP transport, unicast transmission), 163
PTP cumulative offset (UTC:TAI), 144
PTP system time source, 133
PTP UTC correction date, 145
SNTP configuration, 84, 119, 122, 122
timeout
PTP announce message interval+timeout, 138
timer
CWMP ACS close-wait timer, 305
ToD
PTP clock priority, 145
PTP clock type, 126
topology
VCF fabric, 447
VCF fabric topology discovery, 456
traceroute. *See* tracert
tracert
IP address retrieval, 3
node failure detection, 3, 4, 4
NQA client operation (UDP tracert), 20
NQA operation configuration (UDP tracert), 61
system maintenance, 1
tracing
information center trace log file max size, 422
Track
EAA event monitor policy configuration, 323
NQA client+Track collaboration, 27
NQA collaboration, 7
NQA collaboration configuration, 68
traditional
IPv6 NetStream data export, 388, 394, 396
traditional NetStream
data export configuration, 381
traditional NetStream data export, 372
traffic
IPv6 NetStream configuration, 386, 390, 396
IPv6 NetStream enable, 390
IPv6 NetStream filtering, 389
IPv6 NetStream filtering configuration, 390
IPv6 NetStream sampling, 389
IPv6 NetStream sampling configuration, 391
NetStream configuration, 370, 375, 381
NetStream enable, 375
NetStream filtering, 374
NetStream filtering configuration, 375
NetStream flow aging, 378
NetStream flow aging configuration (forced), 379
NetStream flow aging configuration (periodic), 378
NetStream sampling, 374
NetStream sampling configuration, 376
NQA client operation (voice), 22
RMON configuration, 186, 191
sampler configuration, 333
sampler configuration (IPv4 NetStream), 333
sampler creation, 333

sFlow agent+collector information configuration, 401
sFlow configuration, 400, 403, 403
sFlow counter sampling configuration, 402
sFlow flow sampling configuration, 401
transparency
PTP clock node (TC), 124
trapping
Event MIB SNMP notification enable, 203
information center system log SNMP notification, 420
SNMP notification, 177
triggering
Event MIB trigger test configuration, 200
Event MIB trigger test configuration (Boolean), 206
Event MIB trigger test configuration (existence), 204
Event MIB trigger test configuration (threshold), 202, 209
troubleshooting
sFlow, 404
sFlow remote collector cannot receive packets, 404
tunneling
Chef resources (netdev_vte), 292
Puppet resources (netdev_vte), 277

U

UDP
IPv6 NetStream v10 data export format, 388
IPv6 NetStream v9 data export format, 388
IPv6 NTP client/server association mode, 99
IPv6 NTP multicast association mode, 108
IPv6 NTP symmetric active/passive association mode, 102
NQA client operation (UDP echo), 19
NQA client operation (UDP jitter), 16
NQA client operation (UDP tracert), 20
NQA client template, 36
NQA operation configuration (UDP echo), 59
NQA operation configuration (UDP jitter), 54
NQA operation configuration (UDP tracert), 61
NQA template configuration, 73
NTP association mode, 85
NTP broadcast association mode, 103
NTP broadcast mode+authentication, 112
NTP client/server association mode, 98
NTP client/server mode+authentication, 111
NTP client/server mode+MPLS L3VPN network time synchronization, 115
NTP configuration, 79, 84, 98
NTP multicast association mode, 105
NTP symmetric active/passive association mode, 100
NTP symmetric active/passive mode+MPLS L3VPN network time synchronization, 116
PTP configuration, 124, 147
PTP configuration (AES67-2015, IPv4 UDP transport, multicast transmission), 166
PTP configuration (IEEE 1588 v2, IEEE 802.3/Ethernet transport, multicast transmission), 147
PTP configuration (IEEE 1588 v2, IPv4 UDP transport, multicast transmission), 150
PTP configuration (IEEE 1588 v2, IPv4 UDP transport, unicast transmission), 153
PTP configuration (IEEE 802.1AS, IEEE 802.3/Ethernet transport, multicast transmission), 156
PTP configuration (SMPTE ST 2059-2, IPv4 UDP transport, multicast transmission), 159
PTP configuration (SMPTE ST 2059-2, IPv4 UDP transport, unicast transmission), 163
PTP multicast message source IP address, 141
sFlow configuration, 400, 403, 403
unicast
PTP unicast message destination IP address (IPv4 UDP), 142
UNIX
information center log host output configuration, 424
unlocking
NETCONF running configuration, 235
user
PMM Linux user, 327
user process
display, 330
maintain, 330
UTC
PTP correction date, 145
PTP cumulative offset (UTC:TAI), 144

V

value
PTP delay correction value, 144
variable
EAA environment variable configuration (user-defined), 317
EAA event monitor policy environment (user-defined), 317
EAA event monitor policy environment system-defined (event-specific), 316
EAA event monitor policy environment system-defined (public), 316

EAA event monitor policy environment variable, 316
EAA monitor policy configuration (CLI-defined+environment variables), 325
packet capture, 432
VCF fabric
automated deployment, 452
automated deployment process, 453
automated underlay network delpoyment template file configuration, 456
automated underlay network deployment configuration, 456, 457
automated underlay network deployment device role configuration, 456
configuration, 447, 454
display, 462
local proxy ARP, 461
MAC address of VSI interfaces, 462
master spine node configuration, 457
Neutron components, 450
Neutron deployment, 451
overlay network border node configuration, 461
overlay network L2 agent, 460
overlay network L3 agent, 460
overlay network tyep specifying, 459
pausing automated underlay network deployment, 457
RabbitMQ server communication parameters configuration, 458
topology, 447
topology discovery enable, 456
version
IPv6 NetStream v10 data export format, 388
IPv6 NetStream v9 data export format, 388
IPv6 NetStream v9/v10 template refresh rate, 393
NetStream v10 export format, 373
NetStream v5 export format, 373
NetStream v8 export format, 373
NetStream v9 export format, 373
NetStream v9/v10 template refresh rate, 378
view
SNMP access control (view-based), 171
virtual
Virtual Converged Framework. *Use* VCF
VLAN
Chef resources, 287
Chef resources (netdev_l2_interface), 289
Chef resources (netdev_vlan), 291
flow mirroring configuration, 364, 368
flow mirroring QoS policy application, 367
Layer 2 remote port mirroring configuration, 341
Layer 3 remote port mirroring configuration (in ERSPAN mode), 350
Layer 3 remote port mirroring configuration (in tunnel mode), 347
local port mirroring configuration, 339
local port mirroring group monitor port, 341
local port mirroring group source port, 340
packet capture filter configuration (vlan vlan_id expression), 436
port mirroring configuration, 335, 352
port mirroring remote probe VLAN, 335
Puppet resources, 271
Puppet resources (netdev_l2_interface), 273
Puppet resources (netdev_vlan), 276
VCF fabric configuration, 447, 454
voice
NQA client operation, 22
NQA operation configuration, 62
VPN
NTP MPLS L3VPN instance support, 83
VSI
Chef resources (netdev_vsi), 291
Puppet resources (netdev_vsi), 276
VTE
Chef resources (netdev_vte), 292
Puppet resources (netdev_vte), 277
VXLAN
Chef resources (netdev_vxlan), 293
Puppet resources (netdev_vxlan), 278
VCF fabric configuration, 447, 454
VXLAN-aware NetStream, 378

W

workstation
Chef workstation configuration, 283

X

XML
NETCONF capability exchange, 219
NETCONF configuration, 212, 214
NETCONF data filtering, 229
NETCONF data filtering (conditional match), 234
NETCONF data filtering (regex match), 233
NETCONF message format, 212
NETCONF structure, 212
XSD
NETCONF message format, 212

Y

YANG
NETCONF YANG file content retrieval, 224