H3C MSR 系列路由器

网络管理和监控配置指导(V7)

杭州华三通信技术有限公司
http://www.h3c.com.cn

资料版本：6W103-20140512
产品版本：MSR-CMW710-R0105

Copyright © 2013-2014 杭州华三通信技术有限公司及其许可者版权所有，保留一切权利。

未经本公司书面许可，任何单位和个人不得擅自摘抄、复制本书内容的部分或全部，并不得以任何形式传播。

H3C、  、H3CS、H3CIE、H3CNE、Aolynk、  、H^{3}Care、  、IRF、NetPilot、Netflow、SecEngine、SecPath、SecCenter、SecBlade、Comware、ITCMM、HUASAN、华三均为杭州华三通信技术有限公司的商标。对于本手册中出现的其它公司的商标、产品标识及商品名称，由各自权利人拥有。

由于产品版本升级或其他原因，本手册内容有可能变更。H3C 保留在没有任何通知或者提示的情况下对本手册的内容进行修改的权利。本手册仅作为使用指导，H3C 尽全力在本手册中提供准确的信息，但是 H3C 并不确保手册内容完全没有错误，本手册中的所有陈述、信息和建议也不构成任何明示或暗示的担保。

前 言

H3C MSR 系列路由器 配置指导(V7)共分为十五本手册，介绍了 MSR 系列路由器各软件特性的原理及其配置方法，包含原理简介、配置任务描述和配置举例。《网络管理和监控配置指导》主要介绍了设备和网络在维护、管理的过程中所使用的技术原理及配置。

前言部分包含如下内容：

适用款型

本手册所描述的内容适用于 MSR 系列路由器中的如下款型：

款型	
MSR 2600	MSR 26-30
MSR 3600	MSR 36-10
	MSR 36-20
	MSR 36-40
	MSR 36-60
	MSR3600-28
	MSR3600-51
MSR 5600	MSR 56-60
	MSR 56-80

读者对象

本手册主要适用于如下工程师：

- 网络规划人员
- 现场技术支持与维护人员
- 负责网络配置和维护的网络管理员

本书约定

1. 命令行格式约定

格　式	意　义
粗体	命令行关键字（命令中保持不变、必须照输的部分）采用**加粗**字体表示。
斜体	命令行参数（命令中必须由实际值进行替代的部分）采用*斜体*表示。
[]	表示用“[]”括起来的部分在命令配置时是可选的。
{ x | y | ... }	表示从多个选项中仅选取一个。
[x | y | ...]	表示从多个选项中选取一个或者不选。
{ x | y | ... }*	表示从多个选项中至少选取一个。
[x | y | ...]*	表示从多个选项中选取一个、多个或者不选。
&<1-n>	表示符号&前面的参数可以重复输入1～n次。
#	由“#”号开始的行表示为注释行。

2. 各类标志

本书还采用各种醒目标志来表示在操作过程中应该特别注意的地方，这些标志的意义如下：

警告	该标志后的注释需给予格外关注，不当的操作可能会对人身造成伤害。
注意	提醒操作中应注意的事项，不当的操作可能会导致数据丢失或者设备损坏。
提示	为确保设备配置成功或者正常工作而需要特别关注的操作或信息。
说明	对操作内容的描述进行必要的补充和说明。
窍门	配置、操作、或使用设备的技巧、小窍门。

3. 图标约定

本书使用的图标及其含义如下：

	该图标及其相关描述文字代表一般网络设备，如路由器、交换机、防火墙等。
	该图标及其相关描述文字代表一般意义下的路由器，以及其他运行了路由协议的设备。
	该图标及其相关描述文字代表二、三层以太网交换机，以及运行了二层协议的设备。

4. 端口编号示例约定

本手册中出现的端口编号仅作示例，并不代表设备上实际具有此编号的端口，实际使用中请以设备上存在的端口编号为准。

产品配套资料

H3C MSR 系列路由器的配套资料包括如下部分：

大类	资料名称	内容介绍
产品知识介绍	路由器产品彩页	帮助您了解产品的主要规格参数及亮点
硬件描述与安装	路由器安装指导	帮助您详细了解设备硬件规格和安装方法，指导您对设备进行安装
	MSR 系列路由器接口模块手册	帮助您详细了解单板的硬件规格
业务配置	MSR 系列路由器配置指导(V7)	帮助您掌握设备软件功能的配置方法及配置步骤
	MSR 系列路由器命令参考(V7)	详细介绍设备的命令，相当于命令字典，方便您查阅各个命令的功能
运行维护	路由器版本说明书	帮助您了解产品版本的相关信息（包括：版本配套说明、兼容性说明、特性变更说明、技术支持信息）及软件升级方法

资料获取方式

您可以通过H3C网站（www.h3c.com.cn）获取最新的产品资料：

H3C 网站与产品资料相关的主要栏目介绍如下：

- [服务支持/文档中心]：可以获取硬件安装类、软件升级类、配置类或维护类等产品资料。
- [产品技术]：可以获取产品介绍和技术介绍的文档，包括产品相关介绍、技术介绍、技术白皮书等。
- [解决方案]：可以获取解决方案类资料。
- [服务支持/软件下载]：可以获取与软件版本配套的资料。

技术支持

用户支持邮箱：service@h3c.com

技术支持热线电话：400-810-0504（手机、固话均可拨打）

网址：http://www.h3c.com.cn

资料意见反馈

如果您在使用过程中发现产品资料的任何问题，可以通过以下方式反馈：

E-mail：info@h3c.com

感谢您的反馈，让我们做得更好！

目 录

1 系统维护与调试

1.1 Ping功能

1.1.1 Ping功能简介

通过使用 ping 功能，用户可以检查指定地址的设备是否可达，测试链路是否通畅。

Ping 功能是基于 ICMP（Internet Control Message Protocol，互联网控制消息协议）协议来实现的：源端向目的端发送 ICMP 回显请求（ECHO-REQUEST）报文后，根据是否收到目的端的 ICMP 回显应答（ECHO-REPLY）报文来判断目的端是否可达，对于可达的目的端，再根据发送报文个数、接收到响应报文个数以及 Ping 过程报文的往返时间来判断链路的质量。

1.1.2 Ping配置

表1-1 Ping 配置

操作	命令	说明
检查IP网络中的指定地址是否可达	**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** | { **-topology** *topo-name* | **-vpn-instance** *vpn-instance-name* }] * *host*	**ping**命令用于IPv4网络环境，**ping ipv6**命令用于IPv6网络环境 两条命令均可在任意视图下执行 如果网络传输速度较慢，用户在使用**ping**命令时，可以适当增大超时时间**-t**参数的值
	ping ipv6 [**-a** *source-ipv6* | **-c** *count* | **-m** *interval* | **-q** | **-s** *packet-size* | **-t** *timeout* | **-v** | **-tc** *traffic-class* | **-vpn-instance** *vpn-instance-name*] * *host* [**-i** *interface-type interface-number*]	

ping mpls 命令的详细介绍请参见“MPLS 命令参考”中的“MPLS OAM”。

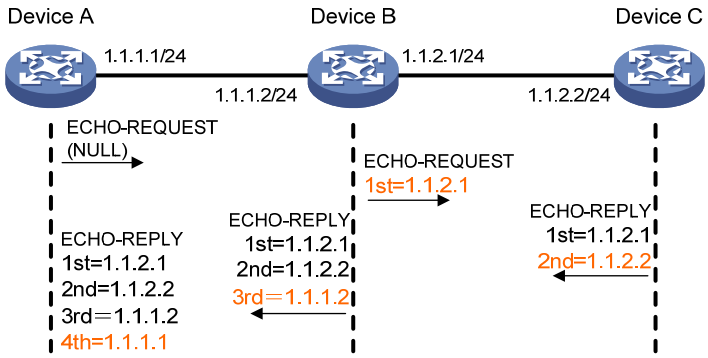
1.1.3 Ping配置举例

1. 组网需求

检查 Device A 与 Device C 之间是否路由可达，如果路由可达，需要了解 Device A 到 Device C 的路由细节。

2. 组网图

图1-1 Ping 应用组网图



3. 配置步骤

使用 **ping** 命令查看 Device A 和 Device C 之间路由是否可达。

```
<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
```

以上显示信息表明 Device A 给 Device C 发送了 5 个 ICMP 报文，收到 5 个 ICMP 报文，没有报文丢失，路由可达。

了解 Device A 到 Device C 的路由细节。

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

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

ping -r的原理如 图 1-1 所示：

(1) 源端（Device A）发送 RR 选项（ICMP 报文中的一个字段）为空的 ICMP 回显请求给目的端（Device C）。

(2) 中间设备（Device B）将自己出接口的 IP 地址（1.1.2.1）添加到 ICMP 回显请求报文的 RR 选项中，并转发该报文。

(3) 目的端收到请求报文后，发送 ICMP 回显响应报文，响应报文会拷贝请求报文的 RR 选项，并将自己出接口的 IP 地址（1.1.2.2）添加到 RR 选项中。

(4) 中间设备将自己出接口的 IP 地址（1.1.1.2）添加到 RR 选项中，并转发该报文。

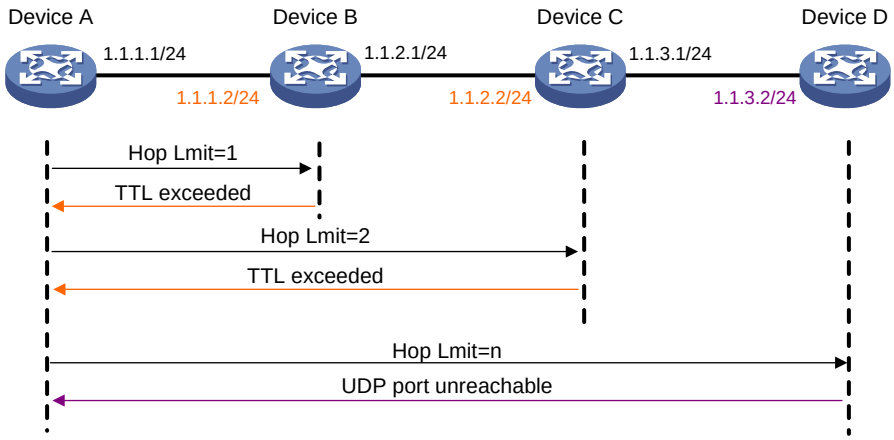
(5) 源端收到 ICMP 回显响应报文，将自己入接口的 IP 地址（1.1.1.1）添加到 RR 选项中。最后得到，Device A 到 Device C 具体路由为 1.1.1.1 <-> {1.1.1.2; 1.1.2.1} <-> 1.1.2.2。

1.2 Tracert功能

1.2.1 Tracert功能简介

通过使用 tracert 功能，用户可以查看 IP 报文从源端到达目的端所经过的三层设备，从而检查网络连接是否可用。当网络出现故障时，用户可以使用该功能分析出现故障的网络节点。

图1-2 Tracert 原理示意图



Tracert功能也是基于ICMP协议来实现的，如 图 1-2 所示，Tracert功能的原理为：

(1) 源端（Device A）向目的端（Device D）发送一个 IP 数据报文，TTL 值为 1，报文的 UDP 端口号是目的端的任何一个应用程序都不可能使用的端口号；

(2) 第一跳（即该报文所到达的第一个三层设备，Device B）回应一个 TTL 超时的 ICMP 错误消息（该报文中含有第一跳的 IP 地址 1.1.1.2），这样源端就得到了第一个三层设备的地址（1.1.1.2）；

(3) 源端重新向目的端发送一个 IP 数据报文，TTL 值为 2；

(4) 第二跳（Device C）回应一个 TTL 超时的 ICMP 错误消息，这样源端就得到了第二个三层设备的地址（1.1.2.2）；

(5) 以上过程不断进行，直到该报文到达目的端，因目的端没有应用程序使用该 UDP 端口，目的端返回一个端口不可达的 ICMP 错误消息（携带了目的端的 IP 地址 1.1.3.2）；

(6) 当源端收到这个端口不可达的 ICMP 错误消息后，就知道报文已经到达了目的端，从而得到数据报文从源端到目的端所经历的路径（1.1.1.2；1.1.2.2；1.1.3.2）。

1.2.2 Tracert配置

1. 配置准备

IPv4 网络环境：

- 需要在中间设备（源端与目的端之间的设备）上开启 ICMP 超时报文发送功能。如果中间设备是 H3C 设备，需要在设备上执行 **ip ttl-expires enable** 命令（该命令的详细介绍请参见“三层技术-IP 业务命令参考”中的“IP 性能优化”）。
- 需要在目的端开启 ICMP 目的不可达报文发送功能。如果目的端是 H3C 设备，需要在设备上执行 **ip unreachables enable** 命令（该命令的详细介绍请参见“三层技术-IP 业务命令参考”中的“IP 性能优化”）。

IPv6 网络环境：

- 需要在中间设备（源端与目的端之间的设备）上开启设备的 ICMPv6 超时报文的发送功能。如果中间设备是 H3C 设备，需要在设备上执行 **ipv6 hoplimit-expires enable** 命令（该命令的详细介绍请参见“三层技术-IP 业务命令参考”中的“IPv6 基础”）。
- 需要在目的端开启设备的 ICMPv6 目的不可达报文的发送功能。如果目的端是 H3C 设备，需要在设备上执行 **ipv6 unreachables enable** 命令（该命令的详细介绍请参见“三层技术-IP 业务命令参考”中的“IPv6 基础”）。

2. Tracert配置

表1-2 Tracert 配置

操作	命令	说明
查看源端到目的端的路由	**tracert** [**-a** *source-ip* | **-f** *first-ttl* | **-m** *max-ttl* | **-p** *port* | **-q** *packet-number* | **-t** *tos* | { **-topology** *topo-name* | **-vpn-instance** *vpn-instance-name* } | **-w** *timeout*] * *host*	**tracert**命令用于IPv4网络环境，**tracert ipv6**命令用于IPv6网络环境 两条命令均可在任意视图下执行
	tracert ipv6 [**-f** *first-hop* | **-m** *max-hops* | **-p** *port* | **-q** *packet-number* | **-t** *traffic-class* | **-vpn-instance** *vpn-instance-name* | **-w** *timeout*] * *host*	

tracert mpls ipv4 命令的详细介绍请参见“MPLS 命令参考”中的“MPLS OAM”。

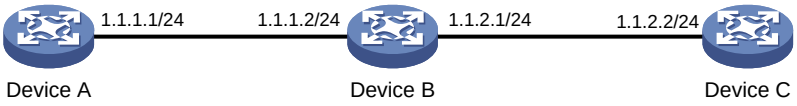
1.2.3 Tracert配置举例

1. 组网需求

Device A 使用 Telnet 登录 Device C 失败，现需要确认 Device A 与 Device C 之间是否路由可达，如果路由不可达，需要确定故障的网络节点。

2. 组网图

图1-3 Tracert 应用组网图



3. 配置步骤

(1) 在Device A、Device B和Device C上分别配置IP地址，IP地址值如 图 1-3 所示。

(2) 在 Device A 上配置一条静态路由。

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

(3) 使用 **ping** 命令查看 Device A 和 Device C 之间路由是否可达。

```
<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
```

(4) 路由不可达，使用 **tracert** 命令确定故障的网络节点。

在 Device B 上开启 ICMP 超时报文发送功能。

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

在 Device C 上开启 ICMP 目的不可达报文发送功能。

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

在 Device A 上使用 **tracert** 命令确定故障的网络节点。

```
<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>
```

从上面结果可以看出，Device A 和 Device C 之间路由不可达。Device A 发往 Device C 的报文已经到达 Device B，Device B 和 Device C 之间的连接出了问题。此时可以在 Device A 和 Device C 上使用 **debugging ip icmp** 命令打开 ICMP 报文的调试开关，查看设备有没有收发指定的 ICMP 报文。或者使用 **display ip routing-table** 查看有没有到对端的路由。

1.3 系统调试

1.3.1 系统调试简介

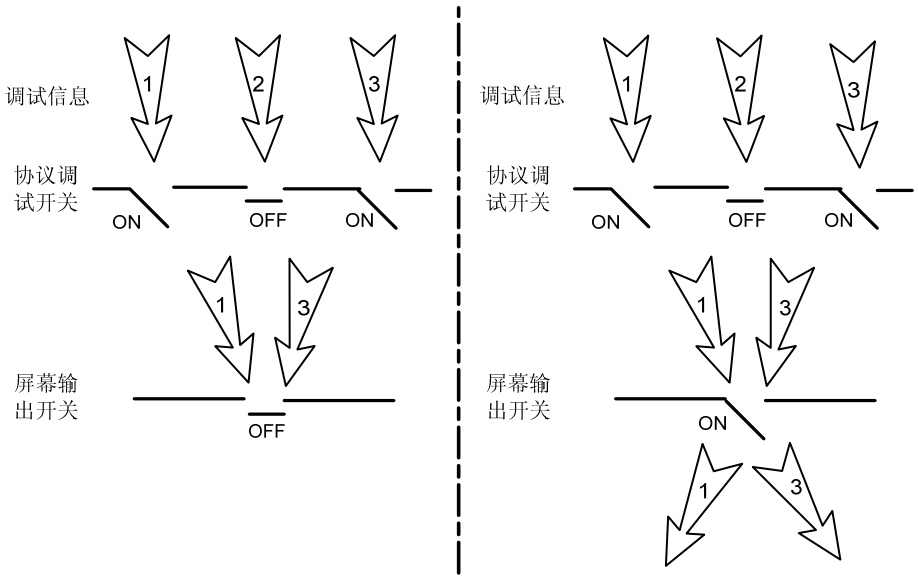
设备提供了种类丰富的调试功能。设备支持的大部分功能模块，系统都提供了相应的调试信息，帮助用户对错误进行诊断和定位。

调试信息的输出可以由两个开关控制：

- 模块调试开关，控制是否生成某模块的调试信息。
- 屏幕输出开关，控制是否在某个用户屏幕上显示调试信息。屏幕输出开关可以使用 **terminal monitor** 和 **terminal logging level** 命令打开，**terminal monitor** 和 **terminal logging level** 命令的详细介绍请参见“网络管理与监控命令参考”中的“信息中心”。

如 图 1-4 所示：假设设备可以为 1、2、3 三个模块提供调试信息，用户只有将两个开关都打开，调试信息才会在终端显示出来。

图1-4 系统调试开关关系图



在控制台上显示是最常用的调试信息输出方式，用户还可以将调试信息发送到别的输出方向，具体配置请参见“网络管理与监控配置指导”中的“信息中心”。

1.3.2 系统调试操作

debugging 命令一般在维护人员进行网络故障诊断时使用。由于调试信息的输出会影响系统的运行效率，所以建议在需要进行网络故障诊断时根据需要打开某个功能模块的调试开关，不要同时打开多个功能模块的调试开关。在调试结束后，建议使用 **undo debugging all** 命令关闭所有模块的调试开关。

表1-3 系统调试操作

操作	命令	说明
打开指定模块的调试开关	**debugging** { **all** [**timeout** *time*] | *module-name* [*option*] }	缺省情况下，所有模块的调试开关均处于关闭状态 该命令在用户视图下执行
（可选）显示已经打开的调试开关	**display debugging** [*module-name*]	该命令可在任意视图下执行

目 录

1 NQA

1.1 NQA简介

NQA（Network Quality Analyzer，网络质量分析）通过发送探测报文，对链路状态、网络性能、网络提供的服务及服务质量进行分析，并为用户提供标识当前网络性能和服务质量的参数，如时延抖动、TCP 连接建立时间、FTP 连接建立时间和文件传输速率等。

利用 NQA 的分析结果，用户可以：

- 及时了解网络的性能状况，针对不同的网络性能进行相应处理。
- 对网络故障进行诊断和定位。

1.1.1 NQA基本概念

1. 测试组

NQA 测试组是一组测试参数的集合，如测试类型、测试目的地址、测试目的端口等。NQA 测试组由一个管理员名称和一个操作标签来标识。管理员通过 NQA 测试组来实现对 NQA 测试的管理和调度。

在一台设备上可以创建多个 NQA 测试组，可以同时启动多个 NQA 测试组进行测试。

2. 测试和探测

启动 NQA 测试组后，每隔一段时间进行一次测试，测试的时间间隔由 **frequency** 命令来设定。

一次 NQA 测试由若干次连续的探测组成，探测的次数由 **probe count** 命令来设定。

说明

对于 Voice 和 Path-jitter 测试，一次测试中只能进行一次探测，不能通过配置修改测试中探测的次数。

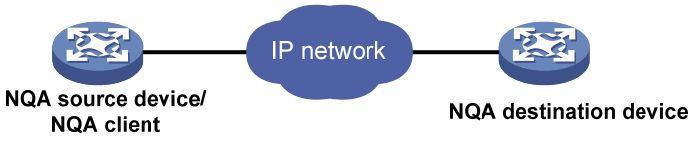
NQA 支持多种测试类型：ICMP-echo、DHCP、DNS、FTP、HTTP、UDP-jitter、SNMP、TCP、UDP-echo、Voice、Path-jitter 和 DLSw 测试。不同测试类型中，探测的含义不同：

- 对于 TCP 和 DLSw 测试，一次探测操作是指建立一次 TCP 或 DLSw 连接；
- 对于 UDP-jitter 和 Voice 测试，一次探测操作是指连续发送多个探测报文，发送探测报文的个数由 **probe packet-number** 命令来设定；
- 对于 FTP、HTTP、DHCP 和 DNS 测试，一次探测操作是指完成一次相应的功能，例如上传或下载一个文件，获取一个 Web 页面，申请一个 IP 地址，将一个域名解析为 IP 地址；
- 对于 ICMP-echo 和 UDP-echo 测试，一次探测操作是指发送一个探测报文；
- 对于 SNMP 测试，一次探测操作是指发送三个 SNMP 协议报文，分别对应 SNMPv1、SNMPv2c 和 SNMPv3 三个版本；

- 对于 Path-jitter 测试，一次探测操作分为两个步骤：首先通过 tracert 探路获取到达目的地址的路径（最大为 64 跳）；再根据 tracert 结果，分别向路径上的每一跳发送多个 ICMP-echo 探测报文，发送探测报文的个数由 **probe packet-number** 命令来设定。

1.1.2 NQA工作机制

图1-1 NQA 测试典型组网图



如 图 1-1 所示，NQA测试的典型组网中包括以下两部分：

- NQA 测试的源端设备：又称为 NQA 客户端，负责发起 NQA 测试，并统计探测结果。NQA 测试组在 NQA 客户端上创建。
- NQA 测试的目的端设备：负责接收、处理和响应 NQA 客户端发来的探测报文。
  - 在进行 TCP、UDP-echo、UDP-jitter 和 Voice 类型测试时，必须在目的端设备上配置 NQA 服务器功能，开启指定 IP 地址和端口上的监听服务。此时，目的端设备又称为 NQA 服务器。当 NQA 服务器接收到客户端发送给指定 IP 地址和端口的探测报文后，将对其进行处理，并发送响应报文。
  - 在其他类型的测试中，目的端设备只要能够处理 NQA 客户端发送的探测报文即可，不需要配置 NQA 服务器功能。例如，在 FTP 测试中，目的端设备上需要配置 FTP 服务器相关功能，以便处理客户端发送的 FTP 报文，而无需配置 NQA 服务器功能。

NQA 测试的过程为：

(1) NQA 客户端构造指定测试类型的探测报文，并发送给目的端设备；

(2) 目的端设备收到探测报文后，回复带有时间戳的应答报文；

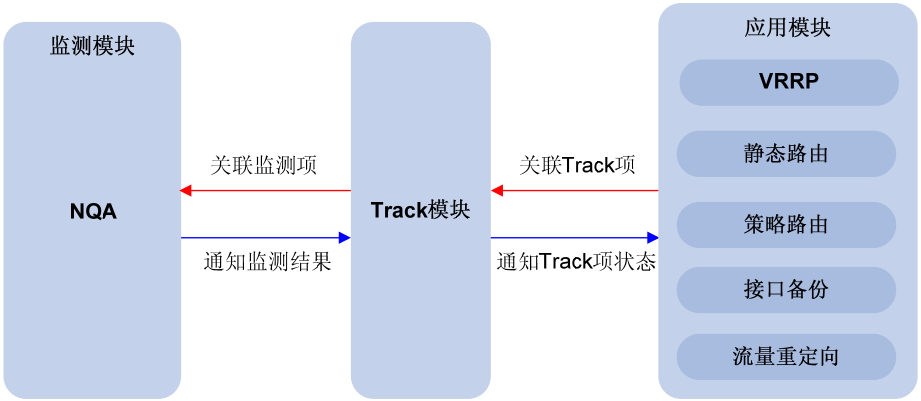
(3) NQA 客户端根据是否收到应答报文，以及应答报文中的时间戳，计算报文丢失率、往返时间等。

1.1.3 支持联动功能

联动功能是指在监测模块、Track 模块和应用模块之间建立关联，实现这些模块之间的联合动作。联动功能利用监测模块对链路状态、网络性能等进行监测，并通过 Track 模块将监测结果及时通知给应用模块，以便应用模块进行相应的处理。联动功能的详细介绍，请参见“可靠性配置指导”中的“Track”。

如 图 1-2 所示，NQA可以作为联动功能的监测模块，对NQA探测结果进行监测，当连续探测失败次数达到一定数目时，就通过Track模块触发应用模块进行相应的处理。

图1-2 联动功能实现示意图



以静态路由为例，用户配置了一条静态路由，下一跳为 192.168.0.88。通过在 NQA、Track 模块和静态路由模块之间建立联动，可以实现静态路由有效性的判断：

(1) 通过 NQA 监测地址 192.168.0.88 是否可达。

(2) 如果 192.168.0.88 可达，则认为该静态路由有效，NQA 不通知 Track 模块改变 Track 项的状态；如果 NQA 发现 192.168.0.88 不可达，则通知 Track 模块改变 Track 项的状态。

(3) Track 模块将改变后的 Track 项状态通知给静态路由模块。静态路由模块据此可以判断该静态路由项是否有效。

1.1.4 支持阈值告警功能

NQA 可以对探测结果进行监测，在本地记录监测结果，或通过 Trap 消息将监测结果通知给网络管理系统，以便网络管理员了解 NQA 测试运行结果和网络性能。

NQA 通过创建阈值告警项，并在阈值告警项中配置监测的对象、阈值类型及触发的动作，来实现阈值告警功能。

阈值告警项包括 invalid、over-threshold 和 below-threshold 三种状态：

- NQA 测试组未启动时，阈值告警项的状态为 invalid。
- NQA 测试组启动后，每次测试或探测结束时，检查监测的对象是否超出指定类型的阈值。如果超出阈值，则阈值告警项的状态变为 over-threshold；如果未超出阈值，则状态变为 below-threshold。

如果阈值告警项的触发动作为 **trap-only**，则当阈值告警项的状态改变时，向网络管理系统发送 Trap 消息。

(1) 监测对象

NQA阈值告警功能支持的监测对象及对应的测试类型，如 表 1-1 所示。

表1-1 NQA 阈值告警功能支持的监测对象及对应的测试类型

监测对象	支持的测试类型
探测持续时间	除UDP-jitter、Voice和Path-jitter之外的测试类型
探测失败次数	除UDP-jitter、Voice和Path-jitter之外的测试类型
报文往返时间	UDP-jitter和Voice测试类型

监测对象	支持的测试类型
丢弃报文数目	UDP-jitter和Voice测试类型
源到目的和目的到源的单向时延抖动	UDP-jitter和Voice测试类型
源到目的和目的到源的单向时延	UDP-jitter和Voice测试类型
ICPIF（Calculated Planning Impairment Factor，计算计划损伤元素）值 ICPIF的详细介绍请参见“1.5.11 配置Voice测试”	Voice测试类型
MOS（Mean Opinion Scores，平均意见得分）值 MOS的详细介绍请参见“1.5.11 配置Voice测试”	Voice测试类型

(2) 阈值类型

NQA 阈值告警功能支持的阈值类型包括：

- 平均值（**average**）：监测一次测试中探测结果的平均值，如果平均值不在指定的范围内，则该监测对象超出阈值。例如，监测一次测试中探测持续时间的平均值。
- 累计数目（**accumulate**）：监测一次测试中探测结果不在指定范围内的累计数目，如果累计数目达到或超过设定的值，则该监测对象超出阈值。
- 连续次数（**consecutive**）：NQA 测试组启动后，监测探测结果连续不在指定范围内的次数，如果该次数达到或超过设定的值，则该监测对象超出阈值。

(3) 触发动作

NQA 阈值告警功能可以触发如下动作：

- **none**：只在本地记录监测结果，以便通过显示命令查看，不向网络管理系统发送 Trap 消息。
- **trap-only**：不仅在本地记录监测结果，当阈值告警项的状态改变时，还向网络管理系统发送 Trap 消息。
- **trigger-only**：在显示信息中记录监测结果的同时，触发其他模块联动。


说明

DNS 测试不支持发送 Trap 消息。

1.2 NQA配置任务简介

NQA 客户端支持两种配置方式：NQA 测试组和 NQA 模板。NQA 测试组配置完毕后，通过调度测试组就可以进行测试操作；NQA 模板配置完毕后并不启动测试，需要外部特性调用 NQA 模板，为该特性创建 NQA 测试组后，并自动启动 NQA 测试。

表1-2 NQA 配置任务简介

操作	说明	详细配置
配置NQA服务器	对于TCP、UDP-echo、UDP-jitter和Voice为必选，其他测试类型为可选	1.3

操作	说明	详细配置
使能NQA客户端功能	必选	1.4
在NQA客户端上配置NQA测试组	至少选择一种NQA测试类型	1.5
在NQA客户端上调度NQA测试组	必选	1.6
在NQA客户端上配置NQA模板	可选	1.7

1.3 配置NQA服务器

在进行 TCP、UDP-echo、UDP-jitter 和 Voice 类型测试前，必须在目的端设备上进行本配置。进行其他类型测试时，不需要进行本配置。

在一个 NQA 服务器上可以配置多个 TCP（或 UDP）监听服务，每个监听服务对应一个监听的 IP 地址和一个端口号。配置的监听 IP 地址和端口号必须与 NQA 客户端上配置的目的 IP 地址和目的端口号一致，且不能与已有的 TCP（或 UDP）监听服务冲突。

表1-3 配置 NQA 服务器

操作	命令	说明
进入系统视图	**system-view**	-
开启NQA服务器功能	**nqa server enable**	缺省情况下，NQA服务器功能处于关闭状态
在NQA服务器上配置TCP监听服务	**nqa server tcp-connect** *ip-address port-number* [**tos** *tos*] [**vpn-instance** *vpn-instance-name*]	二者至少选其一 配置的IP地址和端口号必须与NQA客户端的配置一致，且不能与已有的监听服务冲突 通过本命令可以指定发送应答NQA探测报文（TCP报文或UDP报文）中携带的ToS值 缺省情况下，ToS值为0
在NQA服务器上配置UDP监听服务	**nqa server udp-echo** *ip-address port-number* [**tos** *tos*] [**vpn-instance** *vpn-instance-name*]	

1.4 使能NQA客户端功能

只有使能 NQA 客户端功能后，NQA 客户端的相关配置才会生效。

表1-4 使能 NQA 客户端功能

操作	命令	说明
进入系统视图	**system-view**	-
使能NQA客户端功能	**nqa agent enable**	缺省情况下，NQA客户端功能处于开启状态

1.5 在NQA客户端上配置NQA测试组

1.5.1 NQA测试组配置任务简介

表1-5 NQA 测试组配置任务简介

配置任务	说明	详细配置
配置ICMP-echo测试	至少选其一	1.5.2
配置DHCP测试		1.5.3
配置DNS测试		1.5.4
配置FTP测试		1.5.5
配置HTTP测试		1.5.6
配置UDP-jitter测试		1.5.7
配置SNMP测试		1.5.8
配置TCP测试		1.5.9
配置UDP-echo测试		1.5.10
配置Voice测试		1.5.11
配置DLSw测试		1.5.12
配置Path-jitter测试		1.5.13
配置NQA测试组通用可选参数	可选	1.5.14
配置联动功能	可选	1.5.15
配置阈值告警功能	可选	1.5.16
配置NQA统计功能	可选	1.5.17
配置NQA历史记录功能	可选	1.5.18

1.5.2 配置ICMP-echo测试

ICMP-echo 测试利用 ICMP 协议，根据是否接收到应答报文判断目的主机的可达性。ICMP-echo 测试的功能与 **ping** 命令类似，但 ICMP-echo 测试中可以指定测试的下一跳设备。在源端和目的端设备之间存在多条路径时，通过配置下一跳设备可以指定测试的路径。并且，与 **ping** 命令相比，ICMP-echo 测试输出的信息更为丰富。

表1-6 配置 ICMP-echo 测试

操作	命令	说明
进入系统视图	**system-view**	-
创建NQA测试组，进入NQA测试组视图	**nqa entry** *admin-name operation-tag*	缺省情况下，设备上不存在任何NQA测试组

操作	命令	说明
配置测试类型为ICMP-echo，并进入测试类型视图	**type icmp-echo**	-
配置测试操作的目的地址	**destination ip** *ip-address*	缺省情况下，未配置测试操作的目的IP地址
配置发送的探测报文中的填充内容大小	**data-size** *size*	缺省情况下，发送的探测报文中的填充内容大小为100字节
配置发送的探测报文的填充字符串	**data-fill** *string*	缺省情况下，探测报文的填充内容为十六进制数值00010203040506070809
（可选）指定该接口的IP地址作为ICMP-echo测试中探测报文的源IP地址	**source interface** *interface-type interface-number*	缺省情况下，未指定ICMP-echo测试中探测报文的源IP地址，以报文发送接口的主IP地址作为探测报文中的源IP地址
（可选）配置探测报文的源IP地址	**source ip** *ip-address*	**source ip**命令和**source interface**命令是互相覆盖的关系，新的配置会覆盖已有配置 **source interface**命令指定的接口必须为up状态；**source ip**命令指定的源IP地址必须是设备上接口的IP地址，且接口为up状态，否则测试会失败
（可选）配置探测报文的下一跳IP地址	**next-hop** *ip-address*	缺省情况下，未配置下一跳IP地址

说明

ICMP-echo 测试不支持在 IPv6 网络中使用，如果要测试 IPv6 网络中目的主机的可达性，可以使用 **ping ipv6** 命令。**ping ipv6** 命令的详细介绍，请参见“网络管理和监控命令参考”中的“系统维护与调试”。

1.5.3 配置DHCP测试

DHCP 测试主要用来测试网络上的 DHCP 服务器能否响应客户端请求，以及为客户端分配 IP 地址所需的时间。

NQA 客户端模拟 DHCP 中继转发 DHCP 请求报文向 DHCP 服务器申请 IP 地址的过程，DHCP 服务器进行 DHCP 测试的接口 IP 地址不会改变。DHCP 测试完成后，NQA 客户端会主动发送报文释放申请到的 IP 地址。

表1-7 配置 DHCP 测试

操作	命令	说明
进入系统视图	**system-view**	-
创建NQA测试组，进入NQA测试组视图	**nqa entry** *admin-name operation-tag*	缺省情况下，设备上不存在任何NQA测试组

操作	命令	说明
配置测试类型为DHCP，并进入测试类型视图	**type dhcp**	-
配置测试操作的目的地址	**destination ip** *ip-address*	缺省情况下，未配置测试操作的目的IP地址 将DHCP服务器的IP地址配置为测试操作的目的IP地址
（可选）指定该接口的IP地址作为DHCP测试中探测报文的源IP地址	**source interface** *interface-type interface-number*	缺省情况下，未指定探测报文的源IP地址，设备以发送探测报文的接口主IP地址作为探测报文的源IP地址 **source ip**命令和**source interface**命令是互相覆盖的关系，新的配置会覆盖已有配置 **source ip**命令指定的源IP地址必须是设备上接口的IP地址，且接口为up状态，否则测试将会失败 存在DHCP中继的组网中，DHCP服务器是根据请求报文中的giaddr字段来选择地址池为客户端分配IP地址的。DHCP测试中，源IP地址需要添加到DHCP请求报文中的giaddr字段。关于giaddr字段的详细介绍，请参见“三层技术-IP业务”里的“DHCP概述”
（可选）配置测试操作中探测报文的源IP地址	**source ip** *ip-address*	

1.5.4 配置DNS测试

DNS 测试主要用来测试 NQA 客户端是否可以通过指定的 DNS 服务器将域名解析为 IP 地址，以及域名解析过程需要的时间。

DNS 测试只是模拟域名解析的过程，设备上不会保存要解析的域名与 IP 地址的对应关系。

表1-8 配置 DNS 测试

操作	命令	说明
进入系统视图	**system-view**	-
创建NQA测试组，进入NQA测试组视图	**nqa entry** *admin-name operation-tag*	缺省情况下，设备上不存在任何NQA测试组
配置测试类型为DNS，并进入测试类型视图	**type dns**	-
将DNS服务器的IP地址配置为测试操作的目的地址	**destination ip** *ip-address*	缺省情况下，未配置测试操作的目的IP地址
配置要解析的域名	**resolve-target** *domain-name*	缺省情况下，没有配置要解析的域名

1.5.5 配置FTP测试

FTP 测试主要用来测试 NQA 客户端是否可以与指定的 FTP 服务器建立连接，以及与 FTP 服务器之间传送文件的时间，从而判断 FTP 服务器的连通性及性能。

在进行 FTP 测试之前，需要获取 FTP 用户的用户名和密码。

表1-9 配置 FTP 测试

操作	命令	说明
进入系统视图	**system-view**	-
创建NQA测试组，进入NQA测试组视图	**nqa entry** *admin-name operation-tag*	缺省情况下，设备上不存在任何NQA测试组
配置测试类型为FTP，并进入测试类型视图	**type ftp**	-
配置FTP测试访问的网址	**url** *url*	**url**可以包括远端主机地址和文件名。当操作类型为**get**方式时，必须在url中配置文件名 *url*可以设置为ftp://*host/filename*或ftp://*host*:*port*/*filename*
（可选）配置探测报文的源IP地址	**source ip** *ip-address*	缺省情况下，未指定源IP地址 该命令指定的源IP地址必须是设备上接口的IP地址，且接口为up状态，否则测试将会失败
配置FTP测试的操作类型	**operation** { **get** | **put** }	缺省情况下，FTP操作方式为**get**操作，即从FTP服务器获取文件
配置FTP登录用户名	**username** *username*	缺省情况下，未配置FTP登录用户名
配置FTP登录密码	**password** { **cipher** | **simple** } *password*	缺省情况下，未配置FTP登录密码
（可选）配置FTP服务器和客户端传送文件的文件名	**filename** *file-name*	缺省情况下，未配置FTP服务器和客户端之间传送文件的文件名 当操作类型为**put**方式时，此配置为必选项
配置FTP测试的数据传输方式	**mode** { **active** | **passive** }	缺省情况下，FTP测试的数据传输方式为主动方式

说明

- 进行 **put** 操作时，若配置了 **filename**，发送数据前判断 **filename** 指定的文件是否存在，如果存在则上传该文件，如果不存在则探测失败。
- 进行 **get** 操作时，如果 FTP 服务器上没有以 **url** 中所配置的文件名为名字的文件，则测试不会成功。进行 **get** 操作时，设备上不会保存从服务器获取的文件。
- 进行 **get**、**put** 操作时，请选用较小的文件进行测试，如果文件较大，可能会因为超时而导致测试失败，或由于占用较多的网络带宽而影响其他业务。

1.5.6 配置HTTP测试

HTTP 测试主要用来测试 NQA 客户端是否可以与指定的 HTTP 服务器建立连接，以及从 HTTP 服务器获取数据所需的时间，从而判断 HTTP 服务器的连通性及性能。

表1-10 配置 HTTP 测试

操作	命令	说明
进入系统视图	**system-view**	-
创建NQA测试组，进入NQA测试组视图	**nqa entry** *admin-name operation-tag*	缺省情况下，设备上不存在任何NQA测试组
配置测试类型为HTTP，并进入测试类型视图	**type http**	-
配置HTTP测试访问的网址	**url** *url*	**url**配置形式为http://*host/resource*或http://*host:port/resource*，其中*host*是远端主机的IP地址或主机名，*port*是远端主机的端口，*resource*是远端主机上的资源
配置HTTP登录用户名	**username** *username*	缺省情况下，未配置HTTP登录用户名
配置HTTP登录密码	**password** { **cipher** | **simple** } *password*	缺省情况下，未配置HTTP登录密码
（可选）配置探测报文的源IP地址	**source ip** *ip-address*	缺省情况下，未指定源IP地址 该命令指定的源IP地址必须是设备上接口的IP地址，且接口为up状态，否则测试将会失败
配置HTTP测试的操作类型	**operation** { **get** | **post** | **raw** }	缺省情况下，HTTP操作方式为**get**操作，即从HTTP服务器获取数据
配置HTTP测试所使用的协议版本	**version** { **v1.0** | **v1.1** }	缺省情况下，HTTP测试使用的版本为1.0
（可选）进入配置HTTP测试请求报文内容视图	**raw-request**	输入**raw-request**命令进入raw-request视图，每次进入视图原有报文内容清除
（可选）配置HTTP测试请求报文内容	逐个字符输入或拷贝粘贴请求报文内容	缺省情况下，未配置HTTP测试请求报文内容 当配置**operation**为**raw**时，此配置为必选项
（可选）保存输入内容并退回测试类型视图	**quit**	-

1.5.7 配置UDP-jitter测试

说明

建议不要对知名端口，即 1～1023 之间的端口，进行 UDP-jitter 测试，否则可能导致 NQA 测试失败或该知名端口对应的服务不可用。

语音、视频等实时性业务对时延抖动（Delay jitter）的要求较高。通过 UDP-jitter 测试，可以获得网络的单向和双向时延抖动，从而判断网络是否可以承载实时性业务。

UDP-jitter 测试的过程如下：

(1) 源端以一定的时间间隔向目的端发送探测报文。

(2) 目的端收到探测报文后，为它打上时间戳，并把带有时间戳的报文发送给源端。

(3) 源端收到报文后，根据报文上的时间戳，计算出时延抖动，从而清晰地反映出网络状况。时延抖动的计算方法为相邻两个报文的目的端接收时间间隔减去这两个报文的发送时间间隔。

UDP-jitter测试需要NQA服务器和客户端配合才能完成。进行UDP-jitter测试之前，必须保证NQA服务器端配置了UDP监听功能，配置方法请参见“1.3 配置NQA服务器”。

表1-11 配置 UDP-jitter 测试

操作	命令	说明
进入系统视图	**system-view**	-
创建NQA测试组，进入NQA测试组视图	**nqa entry** *admin-name operation-tag*	缺省情况下，设备上不存在任何NQA测试组
配置测试类型为UDP-jitter，并进入测试类型视图	**type udp-jitter**	-
配置测试操作的目的地址	**destination ip** *ip-address*	缺省情况下，未配置测试操作的目的IP地址 测试操作的目的地址必须与NQA服务器上所配置的监听服务的IP地址一致
配置测试操作的目的端口	**destination port** *port-number*	缺省情况下，未配置测试操作的目的端口号 测试操作的目的端口号必须与NQA服务器上所配置的监听服务的端口号一致
（可选）配置探测报文的源端口号	**source port** *port-number*	缺省情况下，未指定源端口号
配置发送的探测报文中的填充内容的大小	**data-size** *size*	缺省情况下，发送的探测报文中的填充内容大小为100字节
配置发送的探测报文的填充字符串	**data-fill** *string*	缺省情况下，探测报文的填充内容为十六进制数值00010203040506070809
配置一次UDP-jitter探测中发送探测报文的个数	**probe packet-number** *packet-number*	缺省情况下，一次UDP-jitter探测中发送10个探测报文

操作	命令	说明
配置UDP-jitter测试中发送探测报文的时间间隔	**probe packet-interval** *packet-interval*	缺省情况下，UDP-jitter测试中发送探测报文的时间间隔为20毫秒
配置UDP-jitter测试中等待响应报文的超时时间	**probe packet-timeout** *packet-timeout*	缺省情况下，UDP-jitter测试中等待响应报文的超时时间为3000毫秒
（可选）配置探测报文的源IP地址	**source ip** *ip-address*	缺省情况下，未指定源IP地址 该命令指定的源IP地址必须是设备上接口的IP地址，且接口为up状态，否则测试将会失败

说明

display nqa history 命令的显示信息无法反映 UDP-jitter 测试的结果，如果想了解 UDP-jitter 测试的结果，建议通过 **display nqa result** 命令查看最近一次 NQA 测试的结果，或通过 **display nqa statistics** 命令查看 NQA 测试的统计信息。

1.5.8 配置SNMP测试

SNMP 查询测试主要用来测试从 NQA 客户端向 SNMP agent 设备发出一个 SNMP 协议查询报文到接收响应报文的时间。

表1-12 配置 SNMP 测试

操作	命令	说明
进入系统视图	**system-view**	-
创建NQA测试组，进入NQA测试组视图	**nqa entry** *admin-name operation-tag*	缺省情况下，设备上不存在任何NQA测试组
配置测试类型为SNMP，并进入测试类型视图	**type snmp**	-
配置测试操作的目的地址	**destination ip** *ip-address*	缺省情况下，未配置测试操作的目的IP地址
（可选）配置探测报文的源端口号	**source port** *port-number*	缺省情况下，未指定源端口号
（可选）配置探测报文的源IP地址	**source ip** *ip-address*	缺省情况下，未指定源IP地址 该命令指定的源IP地址必须是设备上接口的IP地址，且接口为up状态，否则测试将会失败

1.5.9 配置TCP测试

TCP 测试用来测试客户端和服务器指定端口之间是否能够建立 TCP 连接，以及建立 TCP 连接所需的时间，从而判断服务器指定端口上提供的服务是否可用，及服务性能。

TCP测试需要NQA服务器和客户端配合才能完成。在TCP测试之前，需要在NQA服务器端配置TCP监听功能，配置方法请参见“1.3 配置NQA服务器”。

表1-13 配置 TCP 测试

操作	命令	说明
进入系统视图	**system-view**	-
创建NQA测试组，进入NQA测试组视图	**nqa entry** *admin-name operation-tag*	缺省情况下，设备上不存在任何NQA测试组
配置测试类型为TCP，并进入测试类型视图	**type tcp**	-
配置测试操作的目的地址	**destination ip** *ip-address*	缺省情况下，未配置测试操作的目的IP地址 必须与NQA服务器上配置的监听服务的IP地址一致
配置测试操作的目的端口	**destination port** *port-number*	缺省情况下，未配置测试操作的目的端口号 必须与NQA服务器上配置的监听服务的端口号一致
（可选）配置探测报文的源IP地址	**source ip** *ip-address*	缺省情况下，未指定源IP地址 该命令指定的源IP地址必须是设备上接口的IP地址，且接口为up状态，否则测试将会失败

1.5.10 配置UDP-echo测试

UDP-echo 测试可以用来测试客户端和服务器指定 UDP 端口之间的连通性以及 UDP 报文的往返时间。

UDP-echo测试需要NQA服务器和客户端配合才能完成。在进行UDP-echo测试之前，需要在NQA服务器端配置UDP监听功能，配置方法请参见“1.3 配置NQA服务器”。

表1-14 配置 UDP-echo 测试

操作	命令	说明
进入系统视图	**system-view**	-
创建NQA测试组，进入NQA测试组视图	**nqa entry** *admin-name operation-tag*	缺省情况下，设备上不存在任何NQA测试组
配置测试类型为UDP-echo，并进入测试类型视图	**type udp-echo**	-
配置测试操作的目的地址	**destination ip** *ip-address*	缺省情况下，未配置测试操作的目的IP地址 必须与NQA服务器上配置的监听服务的IP地址一致

操作	命令	说明
配置测试操作的目的端口	**destination port** *port-number*	缺省情况下，未配置测试操作的目的端口号 必须与NQA服务器上配置的监听服务的端口号一致
配置发送的探测报文中的填充内容大小	**data-size** *size*	缺省情况下，UDP-echo测试中发送的探测报文中的填充内容大小为100字节
配置发送的探测报文的填充字符串	**data-fill** *string*	缺省情况下，探测报文的填充内容为十六进制数值00010203040506070809
（可选）配置探测报文的源端口号	**source port** *port-number*	缺省情况下，未指定源端口号
（可选）配置探测报文的源IP地址	**source ip** *ip-address*	缺省情况下，未指定源IP地址 该命令指定的源IP地址必须是设备上接口的IP地址，且接口为up状态，否则测试将会失败

1.5.11 配置Voice测试

提示

建议不要对知名端口，即 1～1023 之间的端口，进行 Voice 测试，否则可能导致 NQA 测试失败或该知名端口对应的服务不可用。

Voice 测试主要用来测试 VoIP（Voice over IP，在 IP 网络上传送语音）网络情况，统计 VoIP 网络参数，以便用户根据网络情况进行相应的调整。

Voice 测试的过程如下：

(1) 源端（NQA 客户端）以一定的时间间隔向目的端（NQA 服务器）发送 G.711 A 律、G.711 μ 律或 G.729 A 律编码格式的语音数据包。

(2) 目的端收到语音数据包后，为它打上时间戳，并把带有时间戳的数据包发送给源端。

(3) 源端收到数据包后，根据数据包上的时间戳等信息，计算出时延抖动、单向延迟等网络参数，从而清晰地反映出网络状况。

除了时延抖动等参数，Voice 测试还可以计算出反映 VoIP 网络状况的语音参数值：

- ICPIF（Calculated Planning Impairment Factor，计算计划损伤元素）：用来量化网络中语音数据的衰减，由单向网络延迟和丢包率等决定。数值越大，表明语音网络质量越差。
- MOS（Mean Opinion Scores，平均意见得分）：语音网络的质量得分。MOS 值的范围为 1～5，该值越高，表明语音网络质量越好。通过计算网络中语音数据的衰减——ICPIF 值，可以估算出 MOS 值。

用户对语音质量的评价具有一定的主观性，不同用户对语音质量的容忍程度不同，因此，衡量语音质量时，需要考虑用户的主观因素。对语音质量容忍程度较强的用户，可以通过 **advantage-factor** 命令配置补偿因子，在计算 ICPIF 值时将减去该补偿因子，修正 ICPIF 和 MOS 值，以便在比较语音质量时综合考虑客观和主观因素。

Voice测试需要NQA服务器和客户端配合才能完成。进行Voice测试之前，必须保证NQA服务器端配置了UDP监听功能，配置方法请参见“1.3 配置NQA服务器”。

表1-15 配置 Voice 测试

操作	命令	说明
进入系统视图	**system-view**	-
创建NQA测试组，进入NQA测试组视图	**nqa entry** *admin-name operation-tag*	缺省情况下，设备上不存在任何NQA测试组
配置测试类型为Voice，并进入测试类型视图	**type voice**	-
配置测试操作的目的地址	**destination ip** *ip-address*	缺省情况下，未配置测试操作的目的IP地址 测试操作的目的地址必须与NQA服务器上所配置的监听服务的IP地址一致
配置测试操作的目的端口	**destination port** *port-number*	缺省情况下，未配置测试操作的目的端口号 测试操作的目的端口号必须与NQA服务器上所配置的监听服务的端口号一致
配置Voice测试的编码格式	**codec-type** { **g711a** | **g711u** | **g729a** }	缺省情况下，语音编码格式为G.711 A律
配置用于计算MOS值和ICPIF值的补偿因子	**advantage-factor** *factor*	缺省情况下，补偿因子取值为0
（可选）配置探测报文的源IP地址	**source ip** *ip-address*	缺省情况下，未指定源IP地址 该命令指定的源IP地址必须是设备上接口的IP地址，且接口为up状态，否则测试将会失败
（可选）配置探测报文的源端口号	**source port** *port-number*	缺省情况下，未指定源端口号
配置发送的探测报文中的填充内容大小	**data-size** *size*	缺省情况下，发送的探测报文中的填充内容大小与配置的编码格式有关，编码格式为**g.711a**和**g.711u**时缺省报文大小为172字节，**g.729a**时为32字节
配置发送的探测报文的填充字符串	**data-fill** *string*	缺省情况下，探测报文的填充内容为十六进制数值00010203040506070809
配置一次Voice探测中发送探测报文的个数	**probe packet-number** *packet-number*	缺省情况下，一次Voice探测中发送1000个探测报文
配置Voice探测中发送探测报文的时间间隔	**probe packet-interval** *packet-interval*	缺省情况下，Voice探测中发送探测报文的时间间隔为20毫秒
配置Voice测试中等待响应报文的超时时间	**probe packet-timeout** *packet-timeout*	缺省情况下，Voice测试中等待响应报文的超时时间为5000毫秒

说明

display nqa history 命令的显示信息无法反映 Voice 测试的结果，如果想了解 Voice 测试的结果，建议通过 **display nqa result** 命令查看最近一次 NQA 测试中当前状态的结果，或通过 **display nqa statistics** 命令查看 NQA 测试的统计信息。

1.5.12 配置DLSw测试

DLSw 测试主要用来测试 DLSw 设备的响应时间。

表1-16 配置 DLSw 测试

操作	命令	说明
进入系统视图	**system-view**	-
创建NQA测试组，进入NQA测试组视图	**nqa entry** *admin-name operation-tag*	缺省情况下，设备上不存在任何NQA测试组
配置测试类型为DLSw，并进入测试类型视图	**type dlsw**	-
配置测试操作的目的地址	**destination ip** *ip-address*	缺省情况下，未配置测试操作的目的IP地址
（可选）配置探测报文的源IP地址	**source ip** *ip-address*	缺省情况下，未指定源IP地址 该命令指定的源IP地址必须是设备上接口的IP地址，且接口为up状态，否则测试将会失败

1.5.13 配置Path-jitter测试

Path-jitter 测试可以作为 UDP-jitter 测试的一种补充，用于在抖动比较大的情况下，进一步探测中间路径的网络质量，以便查找出网络质量差的具体路段。

Path-jitter 测试的过程如下：

(1) NQA 客户端使用 tracert 机制发现到达目的地址的路径信息。

(2) NQA 客户端根据 tracert 结果，逐跳使用 ICMP 机制探测从本机至该跳设备的路径上报文是否有丢失，同时计算该跳路径的时延和时延抖动等信息。

配置 Path-jitter 测试需要在中间设备（源端与目的端之间的设备）上开启 ICMP 超时报文发送功能。如果中间设备是 H3C 设备，需要在设备上执行 **ip ttl-expires enable** 命令（该命令的详细介绍请参见“三层技术-IP 业务命令参考”中的“IP 性能优化”）。需要在目的端开启 ICMP 目的不可达报文发送功能。如果目的端是 H3C 设备，需要在设备上执行 **ip unreachables enable** 命令（该命令的详细介绍请参见“三层技术-IP 业务命令参考”中的“IP 性能优化”）。

表1-17 配置 Path-jitter 测试

操作	命令	说明
进入系统视图	**system-view**	-

操作	命令	说明
创建NQA测试组，进入NQA测试组视图	**nqa entry** *admin-name operation-tag*	缺省情况下，设备上不存在任何NQA测试组
配置测试类型为Path-jitter，并进入测试类型视图	**type path-jitter**	-
配置测试操作的目的地址	**destination ip** *ip-address*	缺省情况下，未配置测试操作的目的IP地址
配置发送的探测报文中的填充内容大小	**data-size** *size*	缺省情况下，发送的探测报文中的填充内容大小为100字节
配置发送的探测报文的填充字符串	**data-fill** *string*	缺省情况下，探测报文的填充内容为十六进制数值00010203040506070809
（可选）配置探测报文的源IP地址	**source ip** *ip-address*	缺省情况下，未指定源IP地址 该命令指定的源IP地址必须是设备上接口的IP地址，且接口为up状态，否则探测将会失败
配置一次Path-jitter探测中发送探测报文的个数	**probe packet-number** *packet-number*	缺省情况下，一次Path-jitter探测中发送10个ICMP探测报文
配置Path-jitter测试中发送探测报文的时间间隔	**probe packet-interval** *packet-interval*	缺省情况下，Path-jitter测试中发送探测报文的时间间隔为20毫秒
配置Path-jitter测试中等待响应报文的超时时间	**probe packet-timeout** *packet-timeout*	缺省情况下，Path-jitter测试中等待响应报文的超时时间为3000毫秒
（可选）配置松散路由	**lsr-path** *ip-address&<1-8>*	缺省情况下，未配置松散路由
（可选）配置仅对目的地址探测	**target-only**	缺省情况下，未配置仅对目的地址探测，Path-jitter测试中会逐跳进行探测

说明

- 通过 **lsr-path** 命令配置松散路由，在 tracert 过程使用该配置进行探路，NQA 客户端根据该松散路由计算时延和时延抖动。
- Path-jitter 测试项对每一条路径记录结果，在路径上的每一跳均记录抖动值、正向抖动值和负向抖动值。

1.5.14 配置NQA测试组通用可选参数

NQA 测试组的通用可选参数，只对当前测试组中的测试有效。

除特别说明外，所有测试类型都可以配置通用可选参数，可以根据实际情况选择配置测试组的参数。

表1-18 配置 NQA 测试组的通用可选参数

操作	命令	说明
进入系统视图	**system-view**	-
创建NQA测试组，进入NQA测试组视图	**nqa entry** *admin-name operation-tag*	缺省情况下，设备上不存在任何NQA测试组
进入测试组测试类型视图	**type** { **dhcp** | **dlsw** | **dns** | **ftp** | **http** | **icmp-echo** | **path-jitter** | **snmp** | **tcp** | **udp-echo** | **udp-jitter** | **voice** }	-
（可选）配置测试组的描述字符串	**description** *text*	缺省情况下，测试组没有描述字符串
（可选）配置测试组连续两次测试开始时间的时间间隔	**frequency** *interval*	缺省情况下，Voice、Path-jitter测试中连续两次测试开始时间的时间间隔为60000毫秒；其他类型的测试中连续两次测试开始时间的时间间隔为0毫秒，即只进行一次测试 如果到达**frequency**指定的时间间隔时，上次测试尚未完成，则不启动新一轮测试
（可选）配置一次NQA测试中进行探测的次数	**probe count** *times*	缺省情况下，一次测试中的探测次数为1次 Voice和Path-jitter测试中探测次数只能为1，不支持该命令
（可选）配置NQA探测超时时间	**probe timeout** *timeout*	缺省情况下，探测的超时时间为3000毫秒 UDP-jitter、Voice和Path-jitter测试不能配置该参数
（可选）配置探测报文在网络中可以经过的最大跳数	**ttl** *value*	缺省情况下，探测报文在网络中可以经过的最大跳数为20跳 DHCP和Path-jitter测试不能配置该参数
（可选）配置NQA探测报文IP报文头中服务类型域的值	**tos** *value*	缺省情况下，NQA探测报文IP报文头中服务类型域的值为0
（可选）启动路由表旁路功能	**route-option bypass-route**	缺省情况下，路由表旁路功能处于关闭状态 DHCP和Path-jitter测试不能配置该参数
（可选）指定测试操作所属的VPN	**vpn-instance** *vpn-instance-name*	缺省情况下，未指定测试操作所属的VPN

1.5.15 配置联动功能

联动功能是通过建立联动项，对当前所在测试组中的探测进行监测，当连续探测失败次数达到阈值时，就触发配置的动作类型。

表1-19 配置联动功能

操作	命令	说明
进入系统视图	**system-view**	-
创建NQA测试组，进入NQA测试组视图	**nqa entry** *admin-name operation-tag*	缺省情况下，设备上不存在任何NQA测试组
进入测试组测试类型视图	**type** { **dhcp** | **dlsw** | **dns** | **ftp** | **http** | **icmp-echo** | **snmp** | **tcp** | **udp-echo** }	UDP-jitter、Voice和Path-jitter测试不支持联动功能
建立联动项	**reaction** *item-number* **checked-element probe-fail threshold-type consecutive** *consecutive-occurrences* **action-type trigger-only**	缺省情况下，未配置联动项
退回系统视图	**quit**	-
配置Track与NQA联动	配置方法请参见“可靠性配置指导”中的“Track”	-
配置Track与应用模块联动	配置方法请参见“可靠性配置指导”中的“Track”	-

说明

联动项创建后，不能再通过 **reaction** 命令修改该联动项的内容。

1.5.16 配置阈值告警功能

通过 **snmp-agent target-host** 命令配置 Trap 消息的目的地址。**snmp-agent target-host** 命令的详细介绍，请参见“网络管理和监控命令参考”中的“SNMP”。

表1-20 配置阈值告警功能

操作	命令	说明
进入系统视图	**system-view**	-
创建NQA测试组，进入NQA测试组视图	**nqa entry** *admin-name operation-tag*	缺省情况下，设备上不存在任何NQA测试组
进入测试组测试类型视图	**type** { **dhcp** | **dlsw** | **dns** | **ftp** | **http** | **icmp-echo** | **snmp** | **tcp** | **udp-echo** | **udp-jitter** | **voice** }	Path-jitter测试不支持配置阈值告警功能
配置在指定条件下向网管服务器发送Trap消息	**reaction trap** { **probe-failure** *consecutive-probe-failures* | **test-complete** | **test-failure** *cumulate-probe-failures* }	缺省情况下，不向网管服务器发送Trap消息 根据实际需要，选择配置发送Trap消息的方法
创建监测探测持续时间的阈值告警组（除UDP-jitter和Voice测试外，均支持）	**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** }]	

操作	命令	说明
创建监测探测失败次数的阈值告警组（除UDP-jitter和Voice测试外，均支持）	**reaction** *item-number* **checked-element probe-fail threshold-type** { **accumulate** *accumulate-occurrences* | **consecutive** *consecutive-occurrences* } [**action-type** { **none** | **trap-only** }]	
创建监测报文往返时延的阈值告警组（仅UDP-jitter和Voice测试支持）	**reaction** *item-number* **checked-element rtt threshold-type** { **accumulate** *accumulate-occurrences* | **average** } **threshold-value** *upper-threshold lower-threshold* [**action-type** { **none** | **trap-only** }]	
创建监测每次测试中丢包数的阈值告警组（仅UDP-jitter和Voice测试支持）	**reaction** *item-number* **checked-element packet-loss threshold-type accumulate** *accumulate-occurrences* [**action-type** { **none** | **trap-only** }]	
创建监测单向时延抖动的阈值告警组（仅UDP-jitter和Voice测试支持）	**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** }]	
创建监测单向时延的阈值告警组（仅UDP-jitter和Voice测试支持）	**reaction** *item-number* **checked-element** { **owd-ds** | **owd-sd** } **threshold-value** *upper-value lower-value*	
创建监测Voice测试ICPIF值的阈值告警组（仅Voice测试支持）	**reaction** *item-number* **checked-element icpif threshold-value** *upper-threshold lower-threshold* [**action-type** { **none** | **trap-only** }]	
创建监测Voice测试MOS值的阈值告警组（仅Voice测试支持）	**reaction** *item-number* **checked-element mos threshold-value** *upper-threshold lower-threshold* [**action-type** { **none** | **trap-only** }]	

说明

- DNS 测试不支持发送 Trap 消息，即对于 DNS 测试，触发动作只能配置为 **none**。
- 在 UDP-jitter、Voice 测试类型视图下执行 **reaction trap** 命令时，只支持 **reaction trap test-complete**。

1.5.17 配置NQA统计功能

NQA 将在指定时间间隔内完成的 NQA 测试归为一组，计算该组测试结果的统计值，这些统计值构成一个统计组。通过 **display nqa statistics** 命令可以显示该统计组的信息。通过 **statistics interval** 命令可以设置统计的时间间隔。

当 NQA 设备上保留的统计组数目达到最大值时，如果形成新的统计组，保存时间最久的统计组将被删除。通过 **statistics max-group** 命令可以设置保留的最大统计组个数。

指定时间间隔内最后一次测试结束后，形成一个统计组。统计组具有老化功能，即统计组保存一定时间后，将被删除。通过 **statistics hold-time** 命令可以设置统计组的保存时间。

表1-21 配置 NQA 统计功能

操作	命令	说明
进入系统视图	**system-view**	-
创建NQA测试组，进入NQA测试组视图	**nqa entry** *admin-name operation-tag*	缺省情况下，设备上不存在任何NQA测试组
进入测试组测试类型视图	**type** { **dhcp** | **dlsw** | **dns** | **ftp** | **http** | **icmp-echo** | **path-jitter** | **snmp** | **tcp** | **udp-echo** | **udp-jitter** | **voice** }	-
（可选）配置对测试结果进行统计的时间间隔	**statistics interval** *interval*	缺省情况下，对测试结果进行统计的时间间隔为60分钟
（可选）配置能够保留的最大统计组个数	**statistics max-group** *number*	缺省情况下，能够保留的最大统计组数为2 最大统计组个数为0时，不进行统计
（可选）配置统计组的保留时间	**statistics hold-time** *hold-time*	缺省情况下，统计组的保留时间为120分钟

说明

- DHCP 测试不支持配置 NQA 统计功能。
- 如果通过 **frequency** 命令指定连续两次测试开始时间的时间间隔为 0，则不生成统计组信息。

1.5.18 配置NQA历史记录功能

开启 NQA 测试组的历史记录保存功能后，系统将记录 NQA 测试的历史信息，通过 **display nqa history** 命令可以查看该测试组的历史记录信息。

通过本配置任务还可以指定：

- 历史记录的保存时间：历史记录保存时间达到配置的值后，该历史记录将被删除。
- 一个测试组中能够保存的最大历史记录个数：如果历史记录个数超过设定的最大数目，则最早的历史记录将会被删除。

表1-22 配置 NQA 历史记录功能

操作	命令	说明
进入系统视图	**system-view**	-
创建NQA测试组，进入NQA测试组视图	**nqa entry** *admin-name operation-tag*	缺省情况下，设备上不存在任何NQA测试组
进入测试组测试类型视图	**type** { **dhcp** | **dlsw** | **dns** | **ftp** | **http** | **icmp-echo** | **snmp** | **tcp** | **udp-echo** }	UDP-jitter、Voice和Path-jitter测试不支持配置历史记录

操作	命令	说明
开启NQA测试组的历史记录保存功能	**history-record enable**	缺省情况下，NQA测试组的历史记录保存功能处于关闭状态
（可选）配置NQA测试组中历史记录的保存时间	**history-record keep-time** *keep-time*	缺省情况下，NQA测试组中历史记录的保存时间为120分钟
（可选）配置在一个测试组中能够保存的最大历史记录个数	**history-record number** *number*	缺省情况下，一个测试组中能够保存的最大历史记录个数为50

1.6 在NQA客户端上调度NQA测试组

通过本配置，可以设置测试组进行测试的启动时间和持续时间。

系统时间在<启动时间>到<启动时间＋持续时间>范围内时，测试组进行测试。执行 **nqa schedule** 命令时：

- 如果系统时间尚未到达启动时间，则到达启动时间后，启动测试；
- 如果系统时间在<启动时间>到<启动时间＋持续时间>之间，则立即启动测试；
- 如果系统时间已经超过<启动时间＋持续时间>，则不会启动测试。

通过 **display clock** 命令可以查看系统的当前时间。

若配置了 **recurring**，则指定测试组每天都被调度运行。每天启动测试的时间由<启动时间>参数指定。

表1-23 在 NQA 客户端上调度 NQA 测试组

操作	命令	说明
进入系统视图	**system-view**	-
在NQA客户端上调度NQA测试组	**nqa schedule** *admin-name operation-tag* **start-time** { *hh:mm:ss* [*yyyy/mm/dd* | *mm/dd/yyyy*] | **now** } **lifetime** { *lifetime* | **forever** } [**recurring**]	-

说明

- 测试组被调度后就不能再进入该测试组视图和测试类型视图。
- 对于已启动的测试组或已经完成测试的测试组，不受系统时间调整的影响，只有等待测试的测试组受系统时间调整的影响。

1.7 在NQA客户端上配置NQA模板

NQA 模板是一组测试参数的集合（如测试目的地址、测试目的端口、测试目标服务器的 URL 等）。NQA 模板供外部特性调用，可以为外部特性提供测试数据，以便其进行相应处理。NQA 模板通过模板名唯一标识。在一台设备上可以创建多个 NQA 模板。

目前 NQA 模板支持 ICMP、DNS、HTTP、TCP、FTP 等测试类型。

说明

对于 NQA 各类型模板，某些测试参数既可以由外部特性提供，也可以手工直接进行配置。若同时通过以上两种方式获取到测试参数，则以手工配置的测试信息为准。

1.7.1 配置ICMP类型的NQA模板

ICMP 类型的 NQA 模板为外部特性提供 ICMP 类型的测试，外部特性通过引用该模板来启动 ICMP 测试，并根据是否接收到 ICMP 应答报文判断目的主机的可达性。ICMP 类型的 NQA 模板支持 IPv4 和 IPv6 网络。

表1-24 配置 ICMP 类型的 NQA 模板

操作		命令	说明
进入系统视图		**system-view**	-
创建ICMP类型的NQA模板，并进入模板视图		**nqa template icmp** *name*	-
（可选）配置测试操作的目的地址	配置测试操作的目的IPv4地址	**destination ip** *ip-address*	两者选其一 缺省情况下，未配置测试操作的目的地址
	配置测试操作的目的IPv6地址	**destination ipv6** *ipv6-address*	
配置发送的探测报文中的填充内容大小		**data-size** *size*	缺省情况下，发送的探测报文中的填充内容大小为100字节
配置发送的探测报文的填充字符串		**data-fill** *string*	缺省情况下，探测报文的填充内容为十六进制数值00010203040506070809
（可选）指定该接口的IP地址作为ICMP-echo测试中探测报文的源IP地址		**source interface** *interface-type interface-number*	缺省情况下，未指定ICMP-echo测试中探测报文的源IP地址，以报文发送接口的主IP地址作为探测报文中的源IP地址 **source interface**命令指定的接口必须为up状态
（可选）配置探测报文的源地址	配置探测报文的源IP地址	**source ip** *ip-address*	二者选其一 缺省情况下，未指定探测报文的源地址 该命令指定的源地址必须是设备上接口的地址，且接口为up状态，否则测试将会失败
	配置探测报文的源IPv6地址	**source ipv6** *ipv6-address*	

1.7.2 配置DNS类型的NQA模板

DNS 类型的 NQA 模板为外部特性提供 DNS 类型的测试。外部特性通过引用该模板来启动 DNS 测试，NQA 客户端向指定的 DNS 服务器发送 DNS 请求报文，NQA 客户端通过是否收到应答及应答报文的合法性来确定服务器的状态。DNS 类型的 NQA 模板支持 IPv4 和 IPv6 网络。

在 DNS 类型的 NQA 模板视图下，用户可以配置期望返回的地址。如果 DNS 服务器返回的 IP 地址中包含了期望地址，则该 DNS 服务器为真实的服务器，测试成功；否则，测试失败。

在进行 DNS 测试之前，需要在 DNS 服务器上创建域名和地址的映射关系。DNS 服务器配置方法，请参见“三层技术-IP 业务配置指导”中的“域名解析”。

表1-25 配置 DNS 类型的 NQA 模板

操作		命令	说明
进入系统视图		**system-view**	-
创建DNS类型的NQA模板，并进入模板视图		**nqa template dns** *name*	-
（可选）配置测试操作的目的地址	配置测试操作的目的IPv4地址	**destination ip** *ip-address*	二者选其一 缺省情况下，未配置测试操作的目的地址
	配置测试操作的目的IPv6地址	**destination ipv6** *ipv6-address*	
配置测试操作的目的端口号		**destination port** *port-number*	缺省情况下，操作的目的端口号为53
配置要解析的域名		**resolve-target** *domain-name*	缺省情况下，没有配置要解析的域名
配置域名解析类型		**resolve-type** { **A** | **AAAA** }	缺省情况下，域名解析类型为A类型 其中A类型表示将域名解析为IPv4地址，AAAA类型表示将域名解析为IPv6地址
（可选）配置探测报文的源地址	配置探测报文的源IP地址	**source ip** *ip-address*	二者选其一 缺省情况下，未指定源地址 该命令指定的源地址必须是设备上接口的地址，且接口为up状态，否则测试将会失败
	配置探测报文的源IPv6地址	**source ipv6** *ipv6-address*	
（可选）配置测试操作中探测报文的源端口号		**source port** *port-number*	缺省情况下，未指定探测报文的源端口号
（可选）配置用户期望返回的地址	配置用户期望返回的IPv4地址	**expect ip** *ip-address*	二者选其一 缺省情况下，未设定期望返回的地址
	配置用户期望返回的IPv6地址	**expect ipv6** *ipv6-address*	

1.7.3 配置TCP类型的NQA模板

TCP 类型的 NQA 模板为外部特性提供 TCP 类型测试，外部特性通过引用该模板，测试客户端和服务器指定端口之间能否建立 TCP 连接。NQA 客户端通过处理服务器端的应答报文，判断服务器指定端口上提供的服务是否可用。

在 TCP 类型的 NQA 模板视图下，用户可以配置期望的应答内容。如果用户未配置期望的应答内容，则 NQA 客户端与服务器间只建立 TCP 连接。

TCP 测试需要 NQA 服务器和客户端配合才能完成。在 TCP 测试之前，需要在 NQA 服务器端配置 TCP 监听功能。

表1-26 配置 TCP 类型的 NQA 模板

操作		命令	说明
进入系统视图		**system-view**	-
创建TCP类型的NQA模板，并进入模板视图		**nqa template tcp** *name*	-
（可选）配置测试操作的目的地址	配置测试操作的目的IPv4地址	**destination ip** *ip-address*	两者选其一 缺省情况下，未配置测试操作的目的地址 必须与NQA服务器上配置的监听服务的IP地址一致，否则探测会失败
	配置测试操作的目的IPv6地址	**destination ipv6** *ipv6-address*	
（可选）配置测试操作的目的端口号		**destination port** *port-number*	缺省情况下，未配置测试操作的目的端口号 必须与NQA服务器上配置的监听服务的端口号一致
配置发送的探测报文的填充字符串		**data-fill** *string*	缺省情况下，探测报文的填充内容为十六进制数值00010203040506070809
（可选）配置探测报文的源地址	配置探测报文的源IP地址	**source ip** *ip-address*	二者选其一 缺省情况下，未指定源地址 该命令指定的源地址必须是设备上接口的地址，且接口为up状态，否则测试将会失败
	配置探测报文的源IPv6地址	**source ipv6** *ipv6-address*	
（可选）配置用户期望的应答内容		**expect data** *expression* [**offset** *number*]	缺省情况下，未配置期望的应答内容 仅当**data-fill**和**expect data**命令都配置时，进行期望应答内容的检查，否则不做检查

1.7.4 配置HTTP类型的NQA模板

HTTP 类型的 NQA 模板为外部特性提供 HTTP 类型测试，外部特性通过引用该模板，测试 NQA 客户端是否可以与指定的 HTTP 服务器建立连接，以及从 HTTP 服务器获取数据所需的时间，从而判断 HTTP 服务器的连通性及性能。

HTTP 类型的 NQA 模板支持用户配置期望返回的数据，对于 HTTP 测试，仅当应答报文中存在“Content-Length”字段，且配置了 **expect data** 命令时，才进行期望应答内容的检查，否则不做检查，通过该功能用户可以判断 HTTP 服务器应答报文的合法性。

HTTP 类型的 NQA 模板支持配置应答状态码。HTTP 报文的应答状态码是由 3 位十进制数组成的字段，它包含 HTTP 服务器的状态信息，用户可以根据该状态码了解 HTTP 服务器的状态。状态码的第一位规定状态码的类型，后两位编码没有规则。

在进行 HTTP 测试之前，需要完成 HTTP 服务器的配置。

表1-27 配置 HTTP 类型的 NQA 模板

操作	命令	说明
进入系统视图	**system-view**	-
创建HTTP类型的NQA模板，并进入模板视图	**nqa template http** *name*	-
配置HTTP测试访问的目的网址	**url** *url*	*url*配置形式为http://*host/resource*或http://*host:port/resource*，其中*host*为远端主机的地址或主机名，*port*是远端主机的端口，*resource*是远端主机上的资源 缺省情况下，*url*取值为http:///*resource*
配置HTTP登录用户名	**username** *username*	缺省情况下，未配置HTTP登录用户名
配置HTTP登录密码	**password** { **cipher** | **simple** } *password*	缺省情况下，未配置HTTP登录密码
配置HTTP的操作方式	**operation** { **get** | **post** | **raw** }	缺省情况下，HTTP操作方式为**get**操作，即从HTTP服务器获取数据。如果HTTP操作方式为**raw**操作，使用原始报文向服务器发送探测报文，配置为**raw**方式时，请求报文为**raw-request**子视图中内容
（可选）进入raw-request视图	**raw-request**	每次进入视图后，之前配置的报文内容将会被清除 当配置**operation**为**raw**时，必须进行本配置
（可选）配置HTTP测试请求报文内容	逐个字符输入或拷贝粘贴请求报文内容	缺省情况下，未配置HTTP测试请求报文内容 当配置**operation**为**raw**时，必须进行本配置
（可选）保存输入内容并退回模板类型视图	**quit**	-

操作		命令	说明
（可选）配置探测报文的源地址	配置探测报文的源IP地址	**source ip** *ip-address*	二者选其一 缺省情况下，未指定源地址 该命令指定的源地址必须是设备上接口的地址，且接口为up状态，否则测试将会失败
	配置探测报文的源IPv6地址	**source ipv6** *ipv6-address*	
（可选）配置期望的应答状态码		**expect status** *status-list*	缺省情况下，未配置期望的应答状态码
（可选）配置期望的应答内容		**expect data** *expression* [**offset** *number*]	缺省情况下，未配置期望的应答内容

说明

HTTP 报文中的“Content-Length”字段用来指明 HTTP 报文正文的长度，不包含报头的长度。当报文中含有该字段时，报文中的数据部分不会出现 multipart 类型，为纯数据部分，这时设备支持检查期望应答的内容。

1.7.5 配置FTP类型的NQA模板

FTP 类型的 NQA 模板为外部特性提供 FTP 类型测试，外部特性通过引用该模板，与指定的 FTP 服务器建立连接，以及与 FTP 服务器之间传送文件的时间，从而判断 FTP 服务器的连通性及性能。在进行 FTP 测试之前，需要在 FTP 服务器上进行相应的配置，包括 FTP 客户端登录 FTP 服务器的用户名、密码等。FTP 服务器的配置方法，请参见“基础配置指导”中的“FTP 和 TFTP”。

表1-28 配置 FTP 类型的 NQA 模板

操作	命令	说明
进入系统视图	**system-view**	-
创建FTP类型的NQA模板，并进入模板视图	**nqa template ftp** *name*	-
配置FTP测试访问的目的网址	**url** *url*	**url**内容包括远端主机地址和文件名。当操作类型为**get**方式时，必须在*url*中配置文件名，当操作类型为**put**方式时，不以该配置内容为准 *url*可以配置为：ftp://*host*/*filename*或ftp://*host*:*port*/*filename* 其中*host*为远端主机的地址或主机名，*port*是远端主机的端口，*resource*是远端主机上的资源
配置FTP的操作类型	**operation** { **get** | **put** }	缺省情况下，FTP操作方式为**get**操作，即从FTP服务器获取文件

操作		命令	说明
配置FTP登录用户名		**username** *username*	缺省情况下，未配置FTP登录用户名
配置FTP登录密码		**password** { **cipher** | **simple** } *password*	缺省情况下，未配置FTP登录密码
（可选）配置FTP服务器和客户端传送文件的文件名		**filename** *filename*	缺省情况下，未配置FTP服务器和客户端之间传送文件的文件名 当操作类型为**put**方式时，必须进行本配置，当配置类型为**get**方式时，不以该配置内容为准
配置FTP的数据传输方式		**mode** { **active** | **passive** }	缺省情况下，FTP数据传输方式为主动方式
（可选）配置探测报文的源地址	配置探测报文的源IP地址	**source ip** *ip-address*	二者选其一 缺省情况下，未指定源地址 该命令指定的源地址必须是设备上接口的地址，且接口为up状态，否则测试将会失败
	配置探测报文的源IPv6地址	**source ipv6** *ipv6-address*	

1.7.6 配置NQA模板通用可选参数

除特别说明外，所有类型 NQA 模板都可以配置通用可选参数，可以根据实际情况选择配置的参数。

表1-29 配置 NQA 模板通用可选参数

操作	命令	说明
进入系统视图	**system-view**	-
创建NQA模板，并进入模板视图	**nqa template** { **dns** | **ftp** | **http** | **icmp** | **tcp** } *name*	-
配置NQA模板的描述字符串	**description** *text*	缺省情况下，没有配置模板的描述字符串
配置连续两次探测开始时间的时间间隔	**frequency** *interval*	缺省情况下，连续两次探测开始时间的时间间隔为5000毫秒 如果到达**frequency**指定的时间间隔时，上次探测尚未完成，则不启动新一轮探测
配置每次探测超时时间	**probe timeout** *timeout*	缺省情况下，探测的超时时间为3000毫秒
配置探测报文在网络中可以经过的最大跳数	**ttl** *value*	缺省情况下，探测报文在网络中可以经过的最大跳数为20跳
配置NQA探测报文IP报文头中服务类型域的值	**tos** *value*	缺省情况下，NQA探测报文IP报文头中服务类型域的值为0
指定操作所属的VPN	**vpn-instance** *vpn-instance-name*	缺省情况下，未指定操作所属的VPN

操作	命令	说明
配置连续探测成功的次数，当连续探测成功次数达到命令配置的数值时，NQA客户端会把探测成功的消息发送给外部特性，使外部特性利用NQA测试的结果进行相应处理	**reaction trigger probe-pass** *count*	缺省情况下，连续探测成功3次时，NQA客户端会把探测成功的消息发送给外部特性，使外部特性利用NQA测试的结果进行相应处理
配置连续探测失败的次数，当连续探测失败次数达到命令配置的数值时，NQA客户端会把探测失败的消息发送给外部特性，使外部特性利用NQA测试的结果进行相应处理	**reaction trigger probe-fail** *count*	缺省情况下，连续探测失败3次时，NQA客户端会把探测失败的消息发送给外部特性，是外部特性利用NQA测试的结果进行相应处理

1.8 NQA显示和维护

在完成上述配置后，在任意视图下执行 **display** 命令可以显示配置后 NQA 的运行情况，通过查看显示信息验证配置的效果。

表1-30 NQA 显示和维护

操作	命令
显示NQA测试组的历史记录	**display nqa history** [*admin-name operation-tag*]
显示NQA阈值告警功能的当前监测结果	**display nqa reaction counters** [*admin-name operation-tag* [*item-number*]]
显示最近一次NQA测试中当前状态的结果	**display nqa result** [*admin-name operation-tag*]
显示NQA测试的统计信息	**display nqa statistics** [*admin-name operation-tag*]
显示服务器的状态信息	**display nqa server**

1.9 NQA典型配置举例

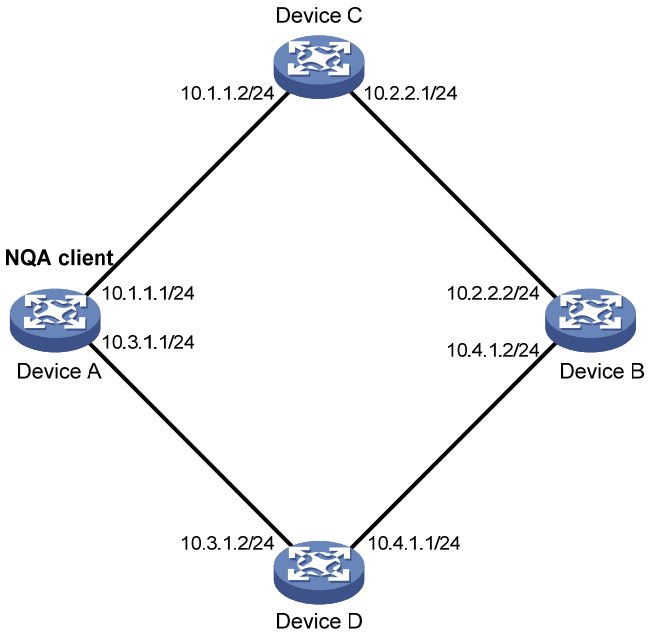
1.9.1 ICMP-echo测试配置举例

1. 组网需求

使用 NQA 的 ICMP-echo 测试功能，测试本端（Device A）发送的报文是否可以经过指定的下一跳设备（Device C）到达指定的目的端（Device B），以及报文的往返时间。

2. 组网图

图1-3 ICMP-echo 测试组网图



3. 配置步骤

配置各接口的 IP 地址。（配置过程略）

配置静态路由或动态路由协议，确保各设备之间路由可达。（配置过程略）

创建 ICMP-echo 类型的 NQA 测试组（管理员为 admin，操作标签为 test1），并配置测试操作的目的地址为 10.2.2.2。

```
<DeviceA> system-view
[DeviceA] nqa entry admin test1
[DeviceA-nqa-admin-test1] type icmp-echo
[DeviceA-nqa-admin-test1-icmp-echo] destination ip 10.2.2.2
```

配置下一跳地址为 10.1.1.2，以便测试报文经过指定的下一跳设备（Device C）到达目的端，而不是通过 Device D 到达目的端。

```
[DeviceA-nqa-admin-test1-icmp-echo] next-hop 10.1.1.2
```

配置可选参数：一次 NQA 测试中探测的次数为 10，探测的超时时间为 500 毫秒，测试组连续两次测试开始时间的时间间隔为 5000 毫秒。

```
[DeviceA-nqa-admin-test1-icmp-echo] probe count 10
[DeviceA-nqa-admin-test1-icmp-echo] probe timeout 500
[DeviceA-nqa-admin-test1-icmp-echo] frequency 5000
```

开启 NQA 历史记录保存功能，并配置一个测试组中能够保存的最大历史记录个数为 10。

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

启动 ICMP-echo 测试操作，并一直进行测试。

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

测试执行一段时间后，停止 ICMP-echo 测试操作。

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

显示 ICMP-echo 测试中最后一次测试中当前状态的结果。

```
[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
```

显示 ICMP-echo 测试的历史记录。

```
[DeviceA] display nqa history admin test1
NQA entry (admin admin, tag test1) history records:
  Index      Response     Status           Time
  370        3            Succeeded        2011-08-23 15:00:01.2
  369        3            Succeeded        2011-08-23 15:00:01.2
  368        3            Succeeded        2011-08-23 15:00:01.2
  367        5            Succeeded        2011-08-23 15:00:01.2
  366        3            Succeeded        2011-08-23 15:00:01.2
  365        3            Succeeded        2011-08-23 15:00:01.2
  364        3            Succeeded        2011-08-23 15:00:01.1
  363        2            Succeeded        2011-08-23 15:00:01.1
  362        3            Succeeded        2011-08-23 15:00:01.1
  361        2            Succeeded        2011-08-23 15:00:01.1
```

以上显示信息表示，Device A 发送的报文可以通过 Device C 到达 Device B；测试过程中未发生丢包；报文的最小、最大、平均往返时间分别为 2 毫秒、5 毫秒和 3 毫秒。

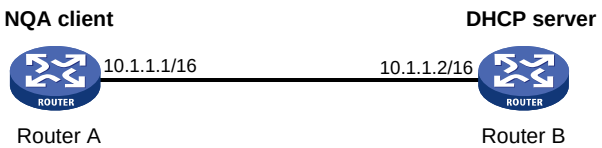
1.9.2 DHCP测试配置举例

1. 组网需求

使用 NQA 的 DHCP 测试功能，测试 Router A 从 DHCP 服务器 Router B 申请到 IP 地址所需的时间。

2. 组网图

图1-4 配置 DHCP 组网图



3. 配置步骤

创建 DHCP 类型的 NQA 测试组（管理员为 admin，操作标签为 test1），并指定进行 DHCP 测试的目的地址为 10.1.1.2。

```
<RouterA> system-view
[RouterA] nqa entry admin test1
[RouterA-nqa-admin-test1] type dhcp
[RouterA-nqa-admin-test1-dhcp] destination ip 10.1.1.2
```

开启 NQA 测试组的历史记录保存功能。

```
[RouterA-nqa-admin-test1-dhcp] history-record enable
[RouterA-nqa-admin-test1-dhcp] quit
```

启动 DHCP 测试操作，并一直进行测试。

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

测试执行一段时间后，停止 DHCP 测试操作。

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

显示 DHCP 测试中最后一次测试中当前状态的结果。

```
[RouterA] 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 09:54:03.8
  Extended results:
    Packet loss ratio: 0%
    Failures due to timeout: 0
    Failures due to internal error: 0
    Failures due to other errors: 0
```

显示 DHCP 测试的历史记录。

```
[RouterA] display nqa history admin test1
NQA entry (admin admin, tag test1) history records:
  Index      Response     Status           Time
  1          512          Succeeded        2011-11-22 09:54:03.8
```

以上显示信息表示，Router A 可以从 DHCP 服务器获取 IP 地址，获取 IP 地址所需的时间为 512 毫秒。

1.9.3 DNS测试配置举例

1. 组网需求

使用 NQA 的 DNS 测试功能，测试 Device A 是否可以通过指定的 DNS 服务器将域名 host.com 解析为 IP 地址，并测试域名解析所需的时间。

2. 组网图

图1-5 配置 DNS 组网图

3. 配置步骤

配置各接口的 IP 地址。（配置过程略）

配置静态路由或动态路由协议，确保各设备之间路由可达。（配置过程略）

创建 DNS 类型的 NQA 测试组（管理员为 admin，操作标签为 test1）。

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

配置测试操作的目的地址为 DNS 服务器的 IP 地址 10.2.2.2，要解析的域名为 host.com。

```
[DeviceA-nqa-admin-test1-dns] destination ip 10.2.2.2
[DeviceA-nqa-admin-test1-dns] resolve-target host.com
```

开启 NQA 测试组的历史记录保存功能。

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

启动 DNS 测试操作，并一直进行测试。

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

测试执行一段时间后，停止 DNS 测试操作。

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

显示 DNS 测试中最后一次测试中当前状态的结果。

```
[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
```

显示 DNS 测试的历史记录。

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

以上显示信息表示，Device A 可以通过指定的 DNS 服务器将域名 host.com 解析为 IP 地址，域名解析所需的时间为 62 毫秒。

1.9.4 FTP测试配置举例

1. 组网需求

使用 NQA 的 FTP 测试功能，测试 Device A 是否可以和指定的 FTP 服务器 Device B 建立连接，以及往 FTP 服务器上传一个文件的时间。登录 FTP 服务器的用户名为 admin，密码为 systemtest，要传送到服务器的文件名为 config.txt。

2. 组网图

图1-6 配置 FTP 组网图

3. 配置步骤

配置各接口的 IP 地址。（配置过程略）

配置静态路由或动态路由协议，确保各设备之间路由可达。（配置过程略）

创建 FTP 类型的 NQA 测试组（管理员为 admin，操作标签为 test1）。

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

配置测试操作的目的地址为 FTP 服务器的 IP 地址 10.2.2.2。

```
[DeviceA-nqa-admin-test1-ftp] url ftp://10.2.2.2
```

配置探测报文的源 IP 地址为 10.1.1.1。

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

配置测试执行的操作为向 FTP 服务器上传文件 config.txt。

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

配置 FTP 操作的登录用户名为 admin。

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

配置 FTP 操作的登录密码为 systemtest。

```
[DeviceA-nqa-admin-test1-ftp] password simple systemtest
```

开启 NQA 测试组的历史记录保存功能。

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

启动 FTP 测试操作，并一直进行测试。

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

测试执行一段时间后，停止 FTP 测试操作。

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

显示 FTP 测试中最后一次测试中当前状态的结果。

```
[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
```

显示 FTP 测试的历史记录。

```
[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
```

以上显示信息表示，Device A 可以和指定的 FTP 服务器 Device B 建立连接，向 FTP 服务器上传一个文件的时间是 173 毫秒。

1.9.5 HTTP测试配置举例

1. 组网需求

使用 NQA 的 HTTP 测试功能，测试是否可以和指定的 HTTP 服务器之间建立连接，以及从 HTTP 服务器获取数据的时间。

2. 组网图

图1-7 HTTP 测试组网图

3. 配置步骤

配置各接口的 IP 地址。（配置过程略）

配置静态路由或动态路由协议，确保各设备之间路由可达。（配置过程略）

创建 HTTP 类型的 NQA 测试组（管理员为 admin，操作标签为 test1）。

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

配置 HTTP 测试服务器的 IP 地址为 10.2.2.2，访问的网址为/index.htm。

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

配置 HTTP 测试的操作方式为 get 操作。（get 操作为缺省操作方式，因此，可以不执行本配置）

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

配置 HTTP 测试使用的版本为 1.0。（缺省情况下使用的版本为 1.0，因此，可以不执行本配置）

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

开启 NQA 测试组的历史记录保存功能。

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

启动 HTTP 测试操作，并一直进行测试。

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

测试执行一段时间后，停止 HTTP 测试操作。

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

显示 HTTP 测试中最后一次测试中当前状态的结果。

```
[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
```

显示 HTTP 测试的历史记录。

```
[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
```

以上显示信息表示，Device A 可以和指定的 HTTP 服务器 Device B 建立连接，从 HTTP 服务器获取数据的时间为 64 毫秒。

1.9.6 UDP-jitter测试配置举例

1. 组网需求

使用 NQA 的 UDP-jitter 测试功能，测试本端（Device A）和指定目的端（Device B）的端口 9000 之间传送报文的时延抖动。

2. 组网图

图1-8 UDP-jitter 测试组网图

3. 配置步骤

(1) 配置各接口的 IP 地址。（配置过程略）

(2) 配置静态路由或动态路由协议，确保各设备之间路由可达。（配置过程略）

(3) 配置 Device B

使能 NQA 服务器，配置监听的 IP 地址为 10.2.2.2，UDP 端口号为 9000。

```
<DeviceB> system-view
[DeviceB] nqa server enable
[DeviceB] nqa server udp-echo 10.2.2.2 9000
```

(4) 配置 Device A

创建 UDP-jitter 类型的 NQA 测试组（管理员为 admin，操作标签为 test1）。

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

配置测试操作的目的地址为 10.2.2.2，目的端口号为 9000。

```
[DeviceA-nqa-admin-test1-udp-jitter] destination ip 10.2.2.2
[DeviceA-nqa-admin-test1-udp-jitter] destination port 9000
```

配置可选参数：测试组连续两次测试开始时间的时间间隔为 1000 毫秒。

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

启动 UDP-jitter 测试操作，并一直进行测试。

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

测试执行一段时间后，停止 UDP-jitter 测试操作。

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

显示 UDP-jitter 测试中最后一次测试中当前状态的结果。

```
[DeviceA] display nqa result admin test1
NQA entry (admin admin, tag test1) test results:
    Send operation times: 10             Receive response times: 10
    Min/Max/Average round trip time: 15/32/17
    Square-Sum of round trip time: 3235
    Last packet received time: 2011-05-29 13:56:17.6
  Extended results:
    Packet loss ratio: 0%
    Failures due to timeout: 0
    Failures due to internal error: 0
    Failures due to other errors: 0
    Packets out of sequence: 0
    Packets arrived late: 0
  UDP-jitter results:
    RTT number: 10
    Min positive SD: 4                   Min positive DS: 1
    Max positive SD: 21                  Max positive DS: 28
    Positive SD number: 5                Positive DS number: 4
    Positive SD sum: 52                  Positive DS sum: 38
    Positive SD average: 10              Positive DS average: 10
    Positive SD square-sum: 754          Positive DS square-sum: 460
    Min negative SD: 1                   Min negative DS: 6
    Max negative SD: 13                  Max negative DS: 22
    Negative SD number: 4                Negative DS number: 5
```

```
    Negative SD sum: 38                    Negative DS sum: 52
    Negative SD average: 10                Negative DS average: 10
    Negative SD square-sum: 460            Negative DS square-sum: 754
  One way results:
    Max SD delay: 15                       Max DS delay: 16
    Min SD delay: 7                        Min DS delay: 7
    Number of SD delay: 10                 Number of DS delay: 10
    Sum of SD delay: 78                    Sum of DS delay: 85
    Square-Sum of SD delay: 666            Square-Sum of DS delay: 787
    SD lost packets: 0                     DS lost packets: 0
    Lost packets for unknown reason: 0
```

显示 UDP-jitter 测试的统计结果。

```
[DeviceA] display nqa statistics admin test1
NQA entry (admin admin, tag test1) test statistics:
  NO. : 1
    Start time: 2011-05-29 13:56:14.0
    Life time: 47 seconds
    Send operation times: 410            Receive response times: 410
    Min/Max/Average round trip time: 1/93/19
    Square-Sum of round trip time: 206176
  Extended results:
    Packet loss ratio: 0%
    Failures due to timeout: 0
    Failures due to internal error: 0
    Failures due to other errors: 0
    Packets out of sequence: 0
    Packets arrived late: 0
  UDP-jitter results:
   RTT number: 410
    Min positive SD: 3                     Min positive DS: 1
    Max positive SD: 30                    Max positive DS: 79
    Positive SD number: 186                Positive DS number: 158
    Positive SD sum: 2602                  Positive DS sum: 1928
    Positive SD average: 13                Positive DS average: 12
    Positive SD square-sum: 45304          Positive DS square-sum: 31682
    Min negative SD: 1                     Min negative DS: 1
    Max negative SD: 30                    Max negative DS: 78
    Negative SD number: 181                Negative DS number: 209
    Negative SD sum: 181                   Negative DS sum: 209
    Negative SD average: 13                Negative DS average: 14
    Negative SD square-sum: 46994          Negative DS square-sum: 3030
  One way results:
    Max SD delay: 46                       Max DS delay: 46
    Min SD delay: 7                        Min DS delay: 7
    Number of SD delay: 410                Number of DS delay: 410
    Sum of SD delay: 3705                  Sum of DS delay: 3891
    Square-Sum of SD delay: 45987          Square-Sum of DS delay: 49393
    SD lost packets: 0                     DS lost packets: 0
```

```
    Lost packets for unknown reason: 0
```

1.9.7 SNMP测试配置举例

1. 组网需求

使用 NQA 的 SNMP 测试功能，测试从 Device A 发出一个 SNMP 协议查询报文到收到 SNMP agent（Device B）响应报文所用的时间。

2. 组网图

图1-9 SNMP 配置测试组网图

3. 配置步骤

(1) 配置各接口的 IP 地址。（配置过程略）

(2) 配置静态路由或动态路由协议，确保各设备之间路由可达。（配置过程略）

(3) 配置 SNMP agent（Device B）

启动 SNMP agent 服务，设置 SNMP 版本为 all、只读团体名为 public、读写团体名为 private。

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

(4) 配置 Device A

创建 SNMP 类型的测试组（管理员为 admin，操作标签为 test1），并配置测试操作的目的地址为 SNMP agent 的 IP 地址 10.2.2.2。

```
<DeviceA> system-view
[DeviceA] nqa entry admin test1
[DeviceA-nqa-admin-test1] type snmp
[DeviceA-nqa-admin-test1-snmp] destination ip 10.2.2.2
```

开启 NQA 测试组的历史记录保存功能。

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

启动测试操作，并一直进行测试。

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

测试执行一段时间后，停止 SNMP 测试操作。

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

显示 SNMP 测试中最后一次测试中当前状态的结果。

```
[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
```

显示 SNMP 测试的历史记录。

```
[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
```

以上显示信息表示，Device A 可以和 SNMP agent（Device B）建立连接，从 Device A 发出一个 SNMP 协议查询报文到收到 SNMP agent 响应报文所用的时间为 50 毫秒。

1.9.8 TCP测试配置举例

1. 组网需求

使用 NQA 的 TCP 测试功能，测试本端（Device A）和指定目的端（Device B）的端口 9000 之间建立 TCP 连接所需的时间。

2. 组网图

图1-10 TCP 测试组网图

3. 配置步骤

(1) 配置各接口的 IP 地址。（配置过程略）

(2) 配置静态路由或动态路由协议，确保各设备之间路由可达。（配置过程略）

(3) 配置 Device B

使能 NQA 服务器，配置监听的 IP 地址为 10.2.2.2，TCP 端口号为 9000。

```
<DeviceB> system-view
[DeviceB] nqa server enable
[DeviceB] nqa server tcp-connect 10.2.2.2 9000
```

(4) 配置 Device A

创建 TCP 类型的测试组（管理员为 admin，操作标签为 test1）。

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

配置测试操作的目的地址为 10.2.2.2，目的端口号为 9000。

```
[DeviceA-nqa-admin-test1-tcp] destination ip 10.2.2.2
[DeviceA-nqa-admin-test1-tcp] destination port 9000
```

开启 NQA 测试组的历史记录保存功能。

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

启动测试操作，并一直进行测试。

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

测试执行一段时间后，停止 TCP 测试操作。

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

显示 TCP 测试中最后一次测试中当前状态的结果。

```
[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
```

显示 TCP 测试的历史记录。

```
[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
```

以上显示信息表示，Device A 可以与 Device B 的端口 9000 建立 TCP 连接，建立连接所需的时间为 13 毫秒。

1.9.9 UDP-echo测试配置举例

1. 组网需求

使用 NQA 的 UDP-echo 测试功能，测试本端（Device A）和指定目的端（Device B）的端口 8000 之间 UDP 协议报文的往返时间。

2. 组网图

图1-11 UDP-echo 测试组网图

3. 配置步骤

(1) 配置各接口的 IP 地址。（配置过程略）

(2) 配置静态路由或动态路由协议，确保各设备之间路由可达。（配置过程略）

(3) 配置 Device B

使能 NQA 服务器，配置监听的 IP 地址为 10.2.2.2，UDP 端口号为 8000。

```
<DeviceB> system-view
[DeviceB] nqa server enable
[DeviceB] nqa server udp-echo 10.2.2.2 8000
```

(4) 配置 Device A

创建 UDP-echo 类型的测试组（管理员为 admin，操作标签为 test1）。

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

配置测试操作的目的地址为 10.2.2.2，目的端口号为 8000。

```
[DeviceA-nqa-admin-test1-udp-echo] destination ip 10.2.2.2
[DeviceA-nqa-admin-test1-udp-echo] destination port 8000
```

开启 NQA 测试组的历史记录保存功能。

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

启动测试操作，并一直进行测试。

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

测试执行一段时间后，停止 UDP-echo 测试操作。

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

显示 UDP-echo 测试中最后一次测试中当前状态的结果。

```
[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
```

显示 UDP-echo 测试的历史记录。

```
[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
```

以上显示信息表示，Device A 和 Device B 的端口 8000 之间 UDP 协议报文的往返时间为 25 毫秒。

1.9.10 Voice测试配置举例

1. 组网需求

使用 NQA 的 Voice 测试功能，测试本端（Device A）和指定的目的端（Device B）之间传送语音报文的时延抖动和网络语音质量参数。

2. 组网图

图1-12 Voice 测试组网图

3. 配置步骤

(1) 配置各接口的 IP 地址。（配置过程略）

(2) 配置静态路由或动态路由协议，确保各设备之间路由可达。（配置过程略）

(3) 配置 Device B

使能 NQA 服务器，配置监听的 IP 地址为 10.2.2.2，UDP 端口号为 9000。

```
<DeviceB> system-view
[DeviceB] nqa server enable
[DeviceB] nqa server udp-echo 10.2.2.2 9000
```

(4) 配置 Device A

创建 Voice 类型的 NQA 测试组（管理员为 admin，操作标签为 test1）。

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

配置测试操作的目的地址为 10.2.2.2，目的端口号为 9000。

```
[DeviceA-nqa-admin-test1-voice] destination ip 10.2.2.2
[DeviceA-nqa-admin-test1-voice] destination port 9000
[DeviceA-nqa-admin-test1-voice] quit
```

启动 Voice 测试操作，并一直进行测试。

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

测试执行一段时间后，停止 Voice 测试操作。

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

显示 Voice 测试中最后一次测试中当前状态的结果。

```
[DeviceA] display nqa result admin test1
NQA entry (admin admin, tag test1) test results:
    Send operation times: 1000          Receive response times: 1000
    Min/Max/Average round trip time: 31/1328/33
    Square-Sum of round trip time: 2844813
    Last packet received time: 2011-06-13 09:49:31.1
  Extended results:
    Packet loss ratio: 0%
    Failures due to timeout: 0
    Failures due to internal error: 0
    Failures due to other errors: 0
Packets out of sequence: 0
    Packets arrived late: 0
  Voice results:
```

```
   RTT number: 1000
    Min positive SD: 1                    Min positive DS: 1
    Max positive SD: 204                  Max positive DS: 1297
    Positive SD number: 257               Positive DS number: 259
    Positive SD sum: 759                  Positive DS sum: 1797
    Positive SD average: 2                Positive DS average: 6
    Positive SD square-sum: 54127         Positive DS square-sum: 1691967
    Min negative SD: 1                    Min negative DS: 1
    Max negative SD: 203                  Max negative DS: 1297
    Negative SD number: 255               Negative DS number: 259
    Negative SD sum: 759                  Negative DS sum: 1796
    Negative SD average: 2                Negative DS average: 6
    Negative SD square-sum: 53655         Negative DS square-sum: 1691776
  One way results:
    Max SD delay: 343                     Max DS delay: 985
    Min SD delay: 343                     Min DS delay: 985
    Number of SD delay: 1                 Number of DS delay: 1
    Sum of SD delay: 343                  Sum of DS delay: 985
    Square-Sum of SD delay: 117649        Square-Sum of DS delay: 970225
    SD lost packets: 0                   DS lost packets: 0
    Lost packets for unknown reason: 0
  Voice scores:
    MOS value: 4.38                       ICPIF value: 0
```

显示 Voice 测试的统计结果。

```
[DeviceA] display nqa statistics admin test1
NQA entry (admin admin, tag test1) test statistics:
  NO. : 1

    Start time: 2011-06-13 09:45:37.8
    Life time: 331 seconds
    Send operation times: 4000           Receive response times: 4000
    Min/Max/Average round trip time: 15/1328/32
    Square-Sum of round trip time: 7160528
  Extended results:
    Packet loss ratio: 0%
    Failures due to timeout: 0
    Failures due to internal error: 0
    Failures due to other errors: 0
Packets out of sequence: 0
    Packets arrived late: 0
  Voice results:
   RTT number: 4000
    Min positive SD: 1                    Min positive DS: 1
    Max positive SD: 360                  Max positive DS: 1297
    Positive SD number: 1030              Positive DS number: 1024
    Positive SD sum: 4363                 Positive DS sum: 5423
    Positive SD average: 4                Positive DS average: 5
    Positive SD square-sum: 497725        Positive DS square-sum: 2254957
```

```
  Min negative SD: 1                    Min negative DS: 1
  Max negative SD: 360                  Max negative DS: 1297
  Negative SD number: 1028              Negative DS number: 1022
  Negative SD sum: 1028                 Negative DS sum: 1022
  Negative SD average: 4                Negative DS average: 5
  Negative SD square-sum: 495901        Negative DS square-sum: 5419
One way results:
  Max SD delay: 359                     Max DS delay: 985
  Min SD delay: 0                       Min DS delay: 0
  Number of SD delay: 4                 Number of DS delay: 4
  Sum of SD delay: 1390                 Sum of DS delay: 1079
  Square-Sum of SD delay: 483202        Square-Sum of DS delay: 973651
  SD lost packets: 0                  DS lost packets: 0
  Lost packets for unknown reason: 0
Voice scores:
  Max MOS value: 4.38                   Min MOS value: 4.38
  Max ICPIF value: 0                    Min ICPIF value: 0
```

1.9.11 DLSw测试配置举例

1. 组网需求

使用 NQA 的 DLSw 测试功能，测试 DLSw 设备的响应时间。

2. 组网图

图1-13 DLSw 测试组网图

3. 配置步骤

配置各接口的 IP 地址。（配置过程略）

配置静态路由或动态路由协议，确保各设备之间路由可达。（配置过程略）

创建 DLSw 类型的测试组（管理员为 admin，操作标签为 test1），并配置测试操作的目的地址为 10.2.2.2。

```
<DeviceA> system-view
[DeviceA] nqa entry admin test1
[DeviceA-nqa-admin-test1] type dlsw
[DeviceA-nqa-admin-test1-dlsw] destination ip 10.2.2.2
```

开启 NQA 测试组的历史记录保存功能。

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

启动测试操作，并一直进行测试。

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

测试执行一段时间后，停止 DLSw 测试操作。

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

显示 DLSw 测试中最后一次测试中当前状态的结果。

```
[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
```

显示 DLSw 测试的历史记录。

```
[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
```

以上显示信息表示，DLSw 设备的响应时间为 19 毫秒。

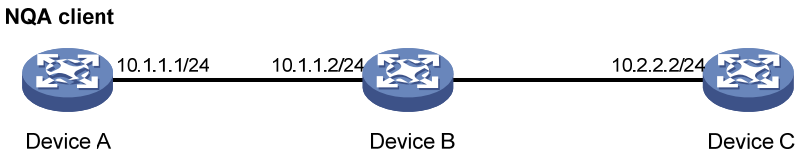
1.9.12 Path-jitter测试配置举例

1. 组网需求

使用 NQA 的 Path-jitter 测试功能，测试本端（Device A）到指定目的端（Device C）间的网络质量情况。

2. 组网图

图1-14 Path-jitter 测试组网图



3. 配置步骤

配置各接口的 IP 地址。（配置过程略）

配置静态路由或动态路由协议，确保各设备之间路由可达。（配置过程略）

在 Device B 上配置 **ip ttl-expires enable** 命令，在设备 C 上配置 **ip unreachables enable** 命令。

创建 Path-jitter 类型的 NQA 测试组（管理员为 admin，操作标签为 test1），并配置测试操作的目的地址为 10.2.2.2。

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

```
[DeviceA-nqa-admin-test1-path-jitter] destination ip 10.2.2.2
```
配置可选参数：测试组连续两次测试开始时间的时间间隔为 10000 毫秒。
```
[DeviceA-nqa-admin-test1-path-jitter] frequency 10000
[DeviceA-nqa-admin-test1-path-jitter] quit
```
启动 Path-jitter 测试操作，并一直进行测试。
```
[DeviceA] nqa schedule admin test1 start-time now lifetime forever
```
测试执行一段时间后，停止 Path-jitter 测试操作。
```
[DeviceA] undo nqa schedule admin test1
```
显示 Path-jitter 测试中最后一次测试中当前状态的结果。
```
[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
```

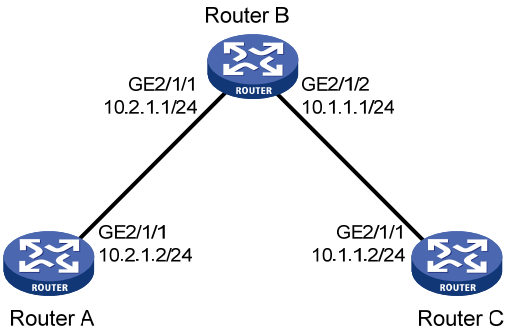
1.9.13 NQA联动配置举例

1. 组网需求

- Router A 到达 Router C 的静态路由下一跳为 Router B。
- 在 Router A 上通过静态路由、Track 与 NQA 联动，对到达 Router C 的静态路由有效性进行实时判断。

2. 组网图

图1-15 NQA 联动配置组网图



3. 配置步骤

(1) 配置各接口的 IP 地址。（配置过程略）

(2) 在 Router A 上配置静态路由，并与 Track 项关联。

配置到达 Router C 的静态路由下一跳地址为 10.2.1.1，并配置静态路由与 Track 项 1 关联。

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

(3) 在 Router A 上配置 NQA 测试组

创建管理员名为 admin、操作标签为 test1 的 NQA 测试组。

```
[RouterA] nqa entry admin test1
```

配置测试类型为 ICMP-echo。

```
[RouterA-nqa-admin-test1] type icmp-echo
```

配置目的地址为 10.2.1.1。

```
[RouterA-nqa-admin-test1-icmp-echo] destination ip 10.2.1.1
```

测试频率为 100ms。

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

配置联动项 1（连续失败 5 次触发联动）。

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

启动 ICMP-echo 探测操作，并一直进行测试。

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

(4)　在 Router A 上配置 Track 项

配置 Track 项 1，关联 NQA 测试组（管理员为 admin，操作标签为 test1）的联动项 1。

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

4. 验证配置

显示 Router A 上 Track 项的信息。

```
[RouterA] 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
```

显示 Router A 的路由表。

```
[RouterA] 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        GE2/1/1
10.2.1.0/24         Direct 0    0            10.2.1.2        GE2/1/1
10.2.1.0/32         Direct 0    0            10.2.1.2        GE2/1/1
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        GE2/1/1
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
```

以上显示信息表示，NQA 测试的结果为下一跳地址 10.2.1.1 可达（Track 项状态为 Positive），配置的静态路由生效。

在 Router B 上删除接口 GigabitEthernet2/1/1 的 IP 地址。

```
<RouterB> system-view
[RouterB] interface gigabitethernet 2/1/1
[RouterB-GigabitEthernet2/1/1] undo ip address
```

显示 Router A 上 Track 项的信息。

```
[RouterA] 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
```

显示 Router A 的路由表。

```
[RouterA] 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        GE2/1/1
10.2.1.0/32         Direct 0    0            10.2.1.2        GE2/1/1
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        GE2/1/1
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
```

以上显示信息表示，NQA 测试的结果为下一跳地址 10.2.1.1 不可达（Track 项状态为 Negative），配置的静态路由无效。

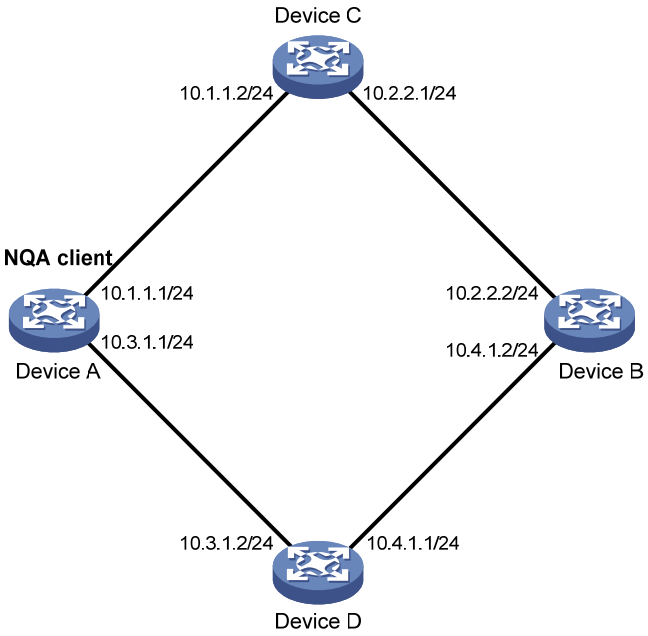
1.9.14 ICMP类型的NQA模板配置举例

1. 组网需求

外部特性通过引用 ICMP 类型的 NQA 模板，测试本端（Device A）发送的报文是否可以到达指定的目的端（Device B）。

2. 组网图

图1-16 ICMP 类型的 NQA 模板配置组网图



3. 配置步骤

配置各接口的 IP 地址。（配置过程略）

配置静态路由或动态路由协议，确保各设备之间路由可达。（配置过程略）

创建 ICMP 类型的 NQA 模板，模板名为 icmp，并配置操作的目的地址为 10.2.2.2。

```
<DeviceA> system-view
[DeviceA] nqa template icmp icmp
[DeviceA-nqatplt-icmp-icmp] destination ip 10.2.2.2
```

配置 ICMP 一次探测的超时时间为 500 毫秒，连续两次探测开始时间的时间间隔为 3000 毫秒。

```
[DeviceA-nqatplt-icmp-icmp] probe timeout 500
[DeviceA-nqatplt-icmp-icmp] frequency 3000
```

配置确定节点有效前需要连续探测成功的次数为 2。当连续探测成功次数达到 2 次时，NQA 客户端把探测成功的消息发送给外部特性，使外部特性能利用 NQA 测试的结果进行相应处理。

```
[DeviceA-nqatplt-icmp-icmp] reaction trigger probe-pass 2
```

配置确定节点失效需要连续探测失败的次数为 2。当连续探测失败次数达到 2 次时，NQA 客户端把探测失败的消息发送给外部特性，是外部特性能利用 NQA 测试的结果进行相应处理。

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

1.9.15 DNS类型的NQA模板配置举例

1. 组网需求

外部特性通过引用 DNS 类型的 NQA 模板，测试 Device A 是否可以通过指定的 DNS 服务器将域名 host.com 解析为 IP 地址。

2. 组网图

图1-17 DNS 类型的 NQA 模板配置组网图

3. 配置步骤

配置各接口的 IP 地址。（配置过程略）

配置静态路由或动态路由协议，确保各设备之间路由可达。（配置过程略）

创建 DNS 类型的 NQA 模板，模板名为 dns。

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

配置操作的目的地址为 DNS 服务器的 IP 地址 10.2.2.2，要解析的域名为 host.com，解析类型为 A，用户期望返回的 IP 地址为 3.3.3.3。

```
[DeviceA-nqatplt-dns-dns] destination ip 10.2.2.2
[DeviceA-nqatplt-dns-dns] resolve-target host.com
[DeviceA-nqatplt-dns-dns] resolve-type A
[DeviceA-nqatplt-dns-dns] expect ip 3.3.3.3
```

配置确定节点有效前需要连续探测成功的次数为 2。当连续探测成功次数达到 2 次时，NQA 客户端把探测成功的消息发送给外部特性，使外部特性能利用 NQA 测试的结果进行相应处理。

```
[DeviceA-nqatplt-dns-dns] reaction trigger probe-pass 2
```

配置确定节点失效需要连续探测失败的次数为 2。当连续探测失败次数达到 2 次时，NQA 客户端把探测失败的消息发送给外部特性，使外部特性能利用 NQA 测试的结果进行相应处理。

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

1.9.16 TCP类型的NQA模板配置举例

1. 组网需求

外部特性通过引用 TCP 类型的 NQA 模板，测试本端（Device A）和服务器（Device B）的端口之间能否建立 TCP 连接，并处理服务器端的应答数据。

2. 组网图

图1-18 TCP 类型的 NQA 模板配置组网图

3. 配置步骤

配置各接口的 IP 地址。（配置过程略）

配置静态路由或动态路由协议，确保各设备之间路由可达。（配置过程略）

(1) 配置 Device B

使能 NQA 服务器，配置监听的 IP 地址为 10.2.2.2，TCP 端口号为 9000。

```
<DeviceB> system-view
[DeviceB] nqa server enable
[DeviceB] nqa server tcp-connect 10.2.2.2 9000
```

(2) 配置 Device A

创建 TCP 类型的 NQA 模板，模板名为 tcp。

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

配置 TCP 测试操作的目的地址为 10.2.2.2，目的端口号为 9000。

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

配置确定节点有效前需要连续探测成功的次数为 2。当连续探测成功次数达到 2 次时，NQA 客户端把探测成功的消息发送给外部特性，使外部特性能利用 NQA 测试的结果进行相应处理。

```
[DeviceA-nqatplt-tcp-tcp] reaction trigger probe-pass 2
```

配置确定节点失效需要连续探测失败的次数为 2。当连续探测失败次数达到 2 次时，NQA 客户端把探测失败的消息发送给外部特性，使外部特性能利用 NQA 测试的结果进行相应处理。

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

1.9.17 HTTP类型的NQA模板配置举例

1. 组网需求

外部特性通过引用 HTTP 类型的 NQA 模板，测试是否可以和指定的 HTTP 服务器之间建立连接，以及能否从 HTTP 服务器获取数据。

2. 组网图

图1-19 HTTP 类型的 NQA 模板配置组网图

3. 配置步骤

配置各接口的 IP 地址。（配置过程略）

配置静态路由或动态路由协议，确保各设备之间路由可达。（配置过程略）

创建 HTTP 类型的 NQA 模板，模板名为 http。

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

配置 HTTP 测试的目的 IP 地址为 10.2.2.2，访问的网址为/index.htm。

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

配置 HTTP 测试的操作方式为 get 操作。（get 操作为缺省操作方式，因此，可以不执行本配置）

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

配置确定节点有效前需要连续探测成功的次数为 2。当连续探测成功次数达到 2 次时，NQA 客户端把探测成功的消息发送给外部特性，使外部特性能利用 NQA 测试的结果进行相应处理。

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

配置确定节点失效需要连续探测失败的次数为 2。当连续探测失败次数达到 2 次时，NQA 客户端把探测失败的消息发送给外部特性，使外部特性能利用 NQA 测试的结果进行相应处理。

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

1.9.18 FTP类型的NQA模板配置举例

1. 组网需求

外部特性通过引用 FTP 类型的 NQA 模板，测试 Device A 是否可以和指定的 FTP 服务器 Device B 建立连接，以及能否往 FTP 服务器上传文件。登录 FTP 服务器的用户名为 admin，密码为 systemtest，要传送到服务器的文件名为 config.txt。

2. 组网图

3. 配置步骤

配置各接口的 IP 地址。（配置过程略）

配置静态路由或动态路由协议，确保各设备之间路由可达。（配置过程略）

创建 FTP 类型的 NQA 模板，模板名为 ftp。

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

配置操作的目的地址为 FTP 服务器的 IP 地址 10.2.2.2。

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

配置探测报文的源 IP 地址为 10.1.1.1。

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

配置执行的操作为向 FTP 服务器上传文件 config.txt。

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

配置登录 FTP 服务器的用户名为 admin。

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

配置登录 FTP 服务器的密码为 systemtest。

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

配置确定节点有效前需要连续探测成功的次数为 2。当连续探测成功次数达到 2 次时，NQA 客户端把探测成功的消息发送给外部特性，使外部特性能利用 NQA 测试的结果进行相应处理。

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

配置确定节点失效需要连续探测失败的次数为 2。当连续探测失败次数达到 2 次时，NQA 客户端把探测失败的消息发送给外部特性，使外部特性能利用 NQA 测试的结果进行相应处理。

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

目 录

1 NTP

提示

设备上不能同时配置 NTP 和 SNTP 功能。

1.1 NTP简介

在大型的网络中，如果依靠管理员手工配置来修改网络中各台设备的系统时间，不但工作量巨大，而且也不能保证时间的精确性。NTP（Network Time Protocol，网络时间协议）可以用来在分布式时间服务器和客户端之间进行时间同步，使网络内所有设备的时间保持一致，并提供较高的时间同步精度。NTP 采用的传输层协议为 UDP，使用的 UDP 端口号为 123。

说明

这里的“分布式”指的是运行 NTP 的设备既可以与其他设备的时间同步，又可以作为时间服务器为其他设备提供时间同步。

NTP 主要应用于需要网络中所有设备的时间保持一致的场合，比如：

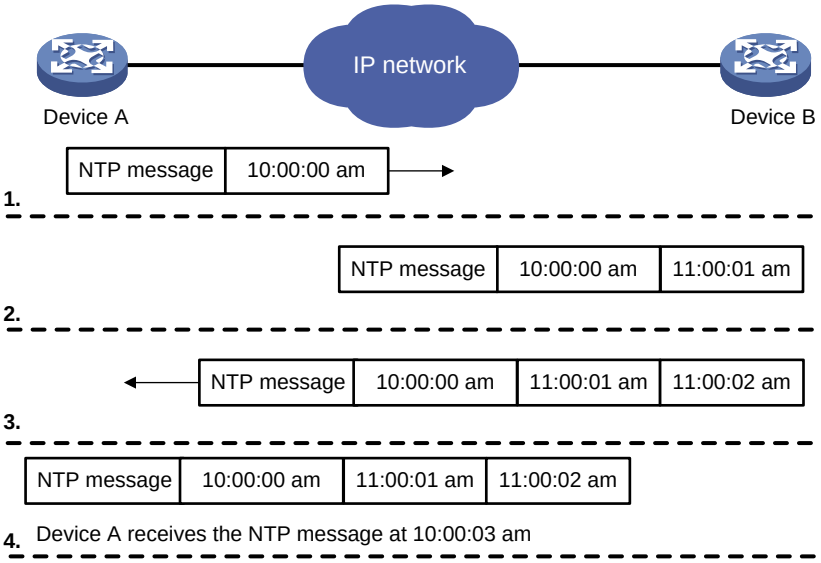
- 需要以时间作为参照依据，对从不同设备采集来的日志信息、调试信息进行分析的网络管理系统。
- 对设备时间一致性有要求的计费系统。
- 多个系统协同处理同一个比较复杂的事件的场合。此时，为保证正确的执行顺序，多个系统的时间必须保持一致。

1.1.1 NTP基本工作原理

NTP的基本工作原理如 图 1-1 所示。Device A和Device B通过网络相连，Device A和Device B的时间不同，需要通过NTP实现时间的自动同步。为便于理解，作如下假设：

- 在 Device A 和 Device B 的时间同步之前，Device A 的时间设定为 10:00:00 am，Device B 的时间设定为 11:00:00 am。
- Device B 作为 NTP 时间服务器，即 Device A 与 Device B 的时间同步。
- NTP 报文从 Device A 到 Device B、从 Device B 到 Device A 单向传输所需要的时间均为 1 秒。
- Device B 处理 NTP 报文所需的时间是 1 秒。

图1-1 NTP 基本工作原理图



Device A 和 Device B 时间同步的工作过程如下：

(1) Device A 发送一个 NTP 报文给 Device B，该报文带有它离开 Device A 时的时间戳，该时间戳为 10:00:00 am（T1）。

(2) 当此 NTP 报文到达 Device B 时，Device B 在 NTP 报文上增加该报文到达 Device B 时的时间戳，该时间戳为 11:00:01 am（T2）。

(3) 当此 NTP 报文离开 Device B 时，Device B 再在 NTP 报文上增加该报文离开 Device B 时的时间戳，该时间戳为 11:00:02 am（T3）。

(4) 当 Device A 接收到该响应报文时，Device A 的本地时间为 10:00:03 am（T4）。

至此，Device A 可以根据上述时间戳计算两个重要的参数：

- NTP 报文的往返时延 Delay = (T4 – T1) – (T3 – T2) = 2 秒。
- Device A 相对 Device B 的时间差 Offset = ((T2 – T1) + (T3 – T4)) / 2 = 1 小时。

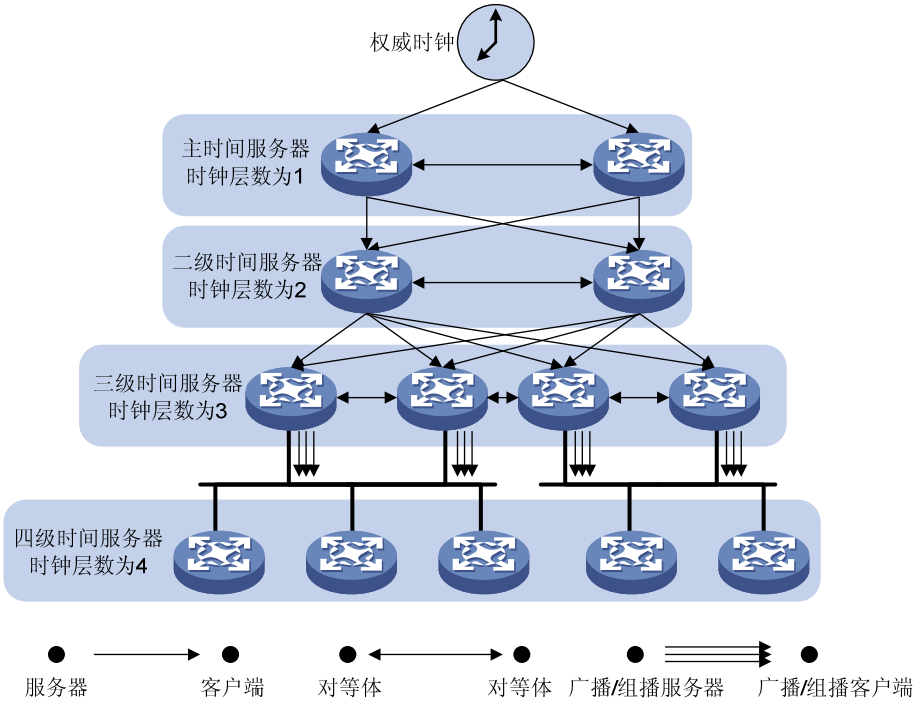
这样，Device A 就能够根据这些信息来设定自己的时间，使之与 Device B 的时间同步。

以上内容只是对 NTP 工作原理的一个粗略描述，详细内容请参阅相关的协议规范。

1.1.2 NTP网络结构及时钟层数

NTP 通过时钟层数来定义时钟的准确度。时钟层数的取值范围为 1～16，取值越小，时钟准确度越高。层数为 1～15 的时钟处于同步状态；层数为 16 的时钟处于未同步状态。

图1-2 NTP 网络结构



如 图 1-2 所示，实际网络中，通常将从权威时钟（如原子时钟）获得时间同步的NTP服务器的层数设置为 1，并将其作为主时间服务器，为网络中其他设备的时钟提供时间同步。网络中的设备与主时间服务器的NTP距离，即NTP同步链上NTP服务器的数目，决定了设备上时钟的层数。例如，从主时间服务器获得时间同步的设备的时钟层数为 2，即比主时间服务器的时钟层数大 1；从时钟层数为 2 的时间服务器获得时间同步的设备的时钟层数为 3，以此类推。

为了保证时间的准确性和可靠性，可以为一台设备指定多个时间服务器，设备根据时钟层数等参数进行时钟过滤和选择，从多个时间服务器中选择最优的时钟，与其同步。设备选中的时钟称为参考时钟。时钟优选过程的详细介绍，请参阅相关的协议规范。

在某些网络中，例如无法与外界通信的孤立网络，网络中的设备无法与权威时钟进行时间同步。此时，可以从该网络中选择一台时钟较为准确的设备，指定该设备与本地时钟进行时间同步，即采用本地时钟作为参考时钟，使得该设备的时钟处于同步状态。该设备作为时间服务器为网络中的其他设备提供时间同步，从而实现整个网络的时间同步。

1.1.3 NTP的工作模式

NTP 支持以下几种工作模式：

- 客户端/服务器模式
- 对等体模式
- 广播模式
- 组播模式

用户可以根据需要选择一种或几种工作模式进行时间同步。各种模式的详细介绍，如 表 1-1 所示。

表1-1 NTP 模式介绍

模式	工作过程	时间同步方向	应用场合
客户端/服务器模式	客户端上需要手工指定NTP服务器的地址。客户端向NTP服务器发送NTP时间同步报文。NTP服务器收到报文后会自动工作在服务器模式，并回复应答报文 如果客户端可以从多个时间服务器获取时间同步，则客户端收到应答报文后，进行时钟过滤和选择，并与优选的时钟进行时间同步	• 客户端能够与 NTP 服务器的时间同步 • NTP 服务器无法与客户端的时间同步	如图1-2所示，该模式通常用于下级的设备从上级的时间服务器获取时间同步
对等体模式	主动对等体（Symmetric active peer）上需要手工指定被动对等体（Symmetric passive peer）的地址。主动对等体向被动对等体发送NTP时间同步报文。被动对等体收到报文后会自动工作在被动对等体模式，并回复应答报文 如果主动对等体可以从多个时间服务器获取时间同步，则主动对等体收到应答报文后，进行时钟过滤和选择，并与优选的时钟进行时间同步	• 主动对等体和被动对等体的时间可以互相同步 • 如果双方的时钟都处于同步状态，则层数大的时钟与层数小的时钟的时间同步	如图1-2所示，该模式通常用于同级的设备间互相同步，以便在同级的设备间形成备份。如果某台设备与所有上级时间服务器的通信出现故障，则该设备仍然可以从同级的时间服务器获得时间同步
广播模式	广播服务器周期性地向广播地址255.255.255.255发送NTP时间同步报文。广播客户端侦听来自广播服务器的广播报文，根据接收的广播报文将设备的时间与广播服务器的时间进行同步 广播客户端接收到广播服务器发送的第一个NTP报文后，会与广播服务器进行报文交互，以获得报文的往返时延，为时间同步提供必要的参数。之后，只有广播服务器单方向发送报文	• 广播客户端能够与广播服务器的时间同步 • 广播服务器无法与广播客户端的时间同步	广播服务器广播发送时间同步报文，可以同时同步同一个子网中多个广播客户端的时间。如图1-2所示，使用同一个时间服务器为同一个子网中的大量设备提供时间同步时，可以使用广播模式，以简化网络配置 由于只有广播服务器单方向发送报文，广播模式的时间准确度不如客户端/服务器模式和对等体模式
组播模式	组播服务器周期性地向指定的组播地址发送NTP时间同步报文。客户端侦听来自服务器的组播报文，根据接收的组播报文将设备的时间与组播服务器的时间进行同步	• 组播客户端能够与组播服务器的时间同步 • 组播服务器无法与组播客户端的时间同步	组播模式对广播模式进行了扩展，组播服务器可以同时为同一子网、不同子网的多个组播客户端提供时间同步 组播模式的时间准确度不如客户端/服务器模式和对等体模式

说明

本文中 NTP 服务器或服务器指的是客户端/服务器模式中工作在服务器模式的设备；时间服务器指的是所有能够提供时间同步的设备，包括 NTP 服务器、NTP 对等体、广播服务器和组播服务器。

1.1.4 NTP安全功能

为了提高时间同步的安全性，NTP 提供了 NTP 服务的访问控制权限和 NTP 验证功能。

1. NTP服务的访问控制权限

本功能是指利用 ACL 限制对端设备对本地设备上 NTP 服务的访问控制权限。NTP 服务的访问控制权限从高到低依次为 **peer**、**server**、**synchronization**、**query**。

- **peer**：完全访问权限。该权限既允许对端设备向本地设备的时间同步，对本地设备进行控制查询（查询 NTP 的一些状态，比如告警信息、验证状态、时间服务器信息等），同时本地设备也可以向对端设备的时间同步。
- **server**：服务器访问与查询权限。该权限允许对端设备向本地设备的时间同步，对本地设备进行控制查询，但本地设备不会向对端设备的时间同步。
- **synchronization**：仅具有访问服务器的权限。该权限只允许对端设备向本地设备的时间同步，但不能进行控制查询。
- **query**：仅具有控制查询的权限。该权限只允许对端设备对本地设备的 NTP 服务进行控制查询，但是不能向本地设备的时间同步。

当设备接收到 NTP 服务请求时，会按照权限从高到低的顺序依次进行匹配。匹配原则为：

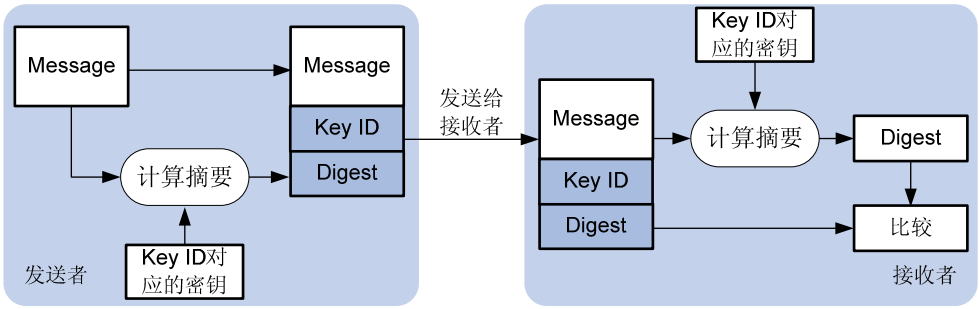
- 如果没有指定权限应用的 ACL 或权限应用的 ACL 尚未创建，则继续匹配下一个权限。
- 如果所有的权限都没有应用 ACL 或权限应用的 ACL 尚未创建，则所有对端设备对本地设备 NTP 服务的访问控制权限均为 **peer**。
- 如果存在应用了 ACL 的权限，且该 ACL 已经创建，则只有 NTP 服务请求匹配了某个权限应用的 ACL 中的 **permit** 规则，发送该 NTP 服务请求的对端设备才会具有该访问控制权限。其他情况下（NTP 服务请求匹配某个权限应用的 ACL 中的 **deny** 规则或没有匹配任何权限的任何规则），发送该 NTP 服务请求的对端设备不具有任何权限。

配置 NTP 服务的访问控制权限，仅提供了一种最小限度的安全措施，更安全的方法是使用 NTP 验证功能。

2. NTP验证功能

在一些对时间同步的安全性要求较高的网络中，运行 NTP 协议时需要使用 NTP 验证功能。NTP 验证功能可以用来验证接收到的 NTP 报文的合法性。只有报文通过验证后，设备才会接收该报文，并从中获取时间同步信息；否则，设备会丢弃该报文。从而，保证设备不会与非法的时间服务器进行时间同步，避免时间同步错误。

图1-3 NTP 验证功能示意图



如 图 1-3 所示，NTP验证功能的工作过程为：

(1) NTP 报文发送者利用密钥 ID 标识的密钥对 NTP 报文进行 MD5 运算，并将计算出来的摘要信息连同 NTP 报文和密钥 ID 一起发送给接收者。

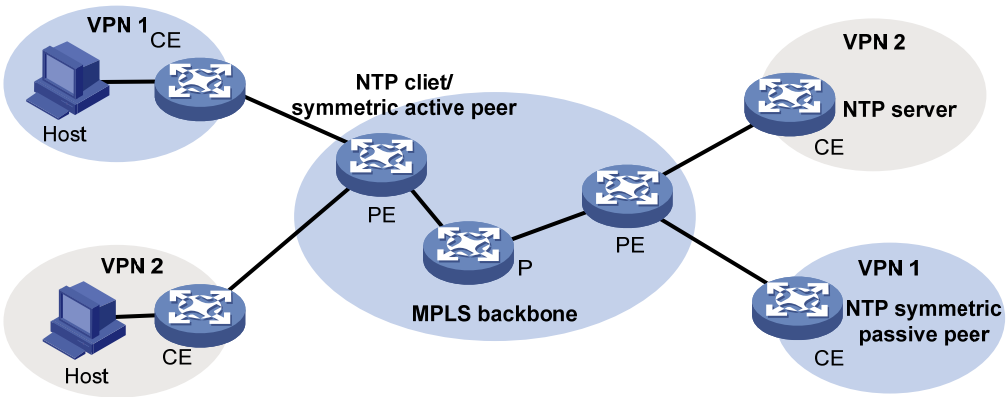
(2) 接收者接收到该 NTP 报文后，根据报文中的密钥 ID 找到对应的密钥，并利用该密钥对报文进行 MD5 运算。接收者将运算结果与报文中的摘要信息比较，如果相同，则接收该报文；否则，丢弃该报文。

1.1.5 NTP支持VPN多实例

设备作为 NTP 客户端或主动对等体时支持 VPN 多实例，实现设备与位于 MPLS L3VPN 中的 NTP 服务器或 NTP 被动对等体进行时间同步。

如下图所示，私网 VPN 1 和 VPN 2 中的用户通过 PE（Provider Edge，服务提供商网络边缘）设备接入 MPLS 骨干网，各 VPN 之间的业务相互隔离。配置 PE 设备工作在 NTP 客户端或 NTP 主动对等体模式，并指定 NTP 服务器或 NTP 被动对等体所属的 VPN 后，可以实现 PE 设备与各 VPN 中的设备进行时间同步。MPLS L3VPN、VPN 实例和 PE 的详细介绍，请参见“MPLS 配置指导”中的“MPLS L3VPN”。

图1-4 NTP 支持 VPN 多实例组网应用图



说明

目前只有单播方式（客户端/服务器模式和对等体模式）的 NTP 时间同步支持 VPN 多实例，广播模式和组播模式的时间同步暂时不支持 VPN 多实例。

1.1.6 协议规范

与 NTP 相关的协议规范有：

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

1.2 NTP配置任务简介

配置 NTP 时，需要注意：

- 为保证时间同步的准确性，不建议用户在组网中配置两个或者两个以上的时钟源，以免造成时钟震荡，甚至出现无法同步的情况。
- 建议用户不要在聚合成员口上进行 NTP 相关配置。

表1-2 NTP 配置任务简介

配置任务		说明	详细配置
开启NTP服务		必选	1.3
配置与网络中的其他设备进行时间同步	配置NTP工作模式	“配置NTP工作模式”和“配置本地时钟作为参考时钟”两个配置任务中至少选择其一，其他配置任务请根据实际需要进行选择	1.4
	配置NTP服务的访问控制权限		1.5
	配置NTP验证功能		1.6
	配置NTP可选参数		1.7
配置本地时钟作为参考时钟			1.8

1.3 开启NTP服务

NTP 服务与 SNTP 服务互斥，同一时刻只能开启其中一个服务。

表1-3 开启 NTP 客户端

操作	命令	说明
进入系统视图	**system-view**	-
开启NTP服务	**ntp-service enable**	缺省情况下，没有开启NTP服务

1.4 配置NTP工作模式

1.4.1 配置NTP客户端/服务器模式

当设备采用客户端/服务器模式时，需要在客户端上指定服务器的地址。

配置 NTP 客户端/服务器模式时，需要注意：

- 服务器需要通过与其他设备同步或配置本地时钟作为参考时钟等方式，使得自己的时钟处于同步状态，否则客户端不会将自己的时间与服务器的时间同步。
- 当服务器端的时钟层数大于或等于客户端的时钟层数时，客户端将不会与其同步。
- 可以通过多次执行 **ntp-service unicast-server** 命令和 **ntp-service ipv6 unicast-server** 命令为设备指定多个服务器。

表1-4 配置 NTP 客户端

操作	命令	说明
进入系统视图	**system-view**	-
为设备指定NTP服务器	**ntp-service unicast-server** { *server-name* | *ip-address* } [**vpn-instance** *vpn-instance-name*] [**authentication-keyid** *keyid* | **priority** | **source** *interface-type interface-number* | **version** *number*] *	缺省情况下，没有为设备指定NTP服务器
为设备指定IPv6 NTP服务器	**ntp-service ipv6 unicast-server** { *server-name* | *ipv6-address* } [**vpn-instance** *vpn-instance-name*] [**authentication-keyid** *keyid* | **priority** | **source** *interface-type interface-number*] *	

1.4.2 配置NTP对等体模式

当设备采用对等体模式时，需要在主动对等体上指定被动对等体的地址。

配置 NTP 对等体模式时，需要注意：

- 被动对等体上需要执行 **ntp-service enable** 命令来开启 NTP 服务，否则被动对等体不会处理来自主动对等体的 NTP 报文。
- 主动对等体和被动对等体的时钟至少要有一个处于同步状态，否则它们的时间都将无法同步。
- 可以通过多次执行 **ntp-service unicast-peer** 命令或 **ntp-service ipv6 unicast-peer** 命令为设备指定多个被动对等体。

表1-5 配置主动对等体

操作	命令	说明
进入系统视图	**system-view**	-
指定设备的被动对等体	**ntp-service unicast-peer** { *peer-name* | *ip-address* } [**vpn-instance** *vpn-instance-name*] [**authentication-keyid** *keyid* | **priority** | **source** *interface-type interface-number* | **version** *number*] *	缺省情况下，没有为设备指定被动对等体
指定设备的IPv6被动对等体	**ntp-service ipv6 unicast-peer** { *peer-name* | *ipv6-address* } [**vpn-instance** *vpn-instance-name*] [**authentication-keyid** *keyid* | **priority** | **source** *interface-type interface-number*] *	

1.4.3 配置NTP广播模式

广播服务器需要通过与其他设备同步或配置本地时钟作为参考时钟等方式，使得自己的时钟处于同步状态，否则广播客户端不会将自己的时间与广播服务器的时间同步。

当设备采用广播模式时，广播服务器端和广播客户端上都需要进行配置。

1. 配置广播客户端

表1-6 配置广播客户端

操作	命令	说明
进入系统视图	**system-view**	-
进入接口视图	**interface** *interface-type interface-number*	进入要接收NTP广播报文的接口
配置设备工作在NTP广播客户端模式	**ntp-service broadcast-client**	缺省情况下，设备没有工作在NTP广播客户端模式 执行本命令后，设备将通过当前接口接收NTP广播报文

2. 配置广播服务器

表1-7 配置广播服务器

操作	命令	说明
进入系统视图	**system-view**	-
进入接口视图	**interface** *interface-type interface-number*	进入要发送NTP广播报文的接口
配置设备工作在NTP广播服务器模式	**ntp-service broadcast-server** [**authentication-keyid** *keyid* | **version** *number*] *	缺省情况下，设备没有工作在NTP广播服务器模式 执行本命令后，设备将通过当前接口周期性发送NTP广播报文

1.4.4 配置NTP组播模式

组播服务器需要通过与其他设备同步或配置本地时钟作为参考时钟等方式，使得自己的时钟处于同步状态，否则组播客户端不会将自己的时间与组播服务器的时间同步。

设备采用组播模式时，在组播服务器端和组播客户端上都需要进行配置。

1. 配置组播客户端

表1-8 配置组播客户端

操作	命令	说明
进入系统视图	**system-view**	-
进入接口视图	**interface** *interface-type interface-number*	进入要接收NTP组播报文的接口
配置设备工作在NTP组播客户端模式	**ntp-service multicast-client** [*ip-address*]	缺省情况下，设备没有工作在组播客户端模式 执行本命令后，设备将通过当前接口接收NTP组播报文
配置设备工作在IPv6 NTP组播客户端模式	**ntp-service ipv6 multicast-client** *ipv6-multicast-address*	

2. 配置组播服务器

表1-9 配置组播服务器

操作	命令	说明
进入系统视图	**system-view**	-
进入接口视图	**interface** *interface-type interface-number*	进入要发送NTP组播报文的接口
配置设备工作在NTP组播服务器模式	**ntp-service multicast-server** [*ip-address*] [**authentication-keyid** *keyid* | **ttl** *ttl-number* | **version** *number*] *	缺省情况下，设备没有工作在组播服务器模式 执行本命令后，设备将通过当前接口周期性发送NTP组播报文
配置设备工作在IPv6 NTP组播服务器模式	**ntp-service ipv6 multicast-server** *ipv6-multicast-address* [**authentication-keyid** *keyid* | **ttl** *ttl-number*] *	

1.5 配置NTP服务的访问控制权限

在配置对本地设备 NTP 服务的访问控制权限之前，需要创建并配置与访问权限关联的 ACL。ACL 的配置方法请参见“ACL 和 QoS 配置指导”中的“ACL”。

表1-10 配置 NTP 服务的访问控制权限

操作	命令	说明
进入系统视图	**system-view**	-
配置对端设备对本地设备NTP服务的访问控制权限	**ntp-service** { **peer** | **query** | **server** | **synchronization** } **acl** *acl-number*	缺省情况下，对端设备对本地设备NTP服务的访问控制权限为**peer**（完全访问权限）
配置对端设备对本地设备IPv6 NTP服务的访问控制权限	**ntp-service ipv6** { **peer** | **query** | **server** | **synchronization** } **acl** *acl-number*	

1.6 配置NTP验证功能

1.6.1 配置客户端/服务器模式的NTP验证功能

配置客户端/服务器模式的 NTP 验证功能时，需要在客户端和服务器上都使能 NTP 验证功能、配置验证密钥、将验证密钥设为可信密钥，并在客户端上将可信密钥与 NTP 服务器关联。服务器端和客户端上配置的密钥 ID 和密钥值必须保持一致，否则会导致 NTP 验证失败。

表1-11 配置客户端的 NTP 验证

操作	命令	说明
进入系统视图	**system-view**	-
使能NTP身份验证功能	**ntp-service authentication enable**	缺省情况下，NTP身份验证功能处于关闭状态

操作	命令	说明
配置NTP身份验证密钥	**ntp-service authentication-keyid** *keyid* **authentication-mode md5** { **cipher** | **simple** } *value*	缺省情况下，没有配置NTP身份验证密钥
配置指定密钥为可信密钥	**ntp-service reliable authentication-keyid** *keyid*	缺省情况下，没有指定可信密钥
将指定密钥与对应的NTP服务器关联	**ntp-service unicast-server** { *server-name* | *ip-address* } [**vpn-instance** *vpn-instance-name*] **authentication-keyid** *keyid*	缺省情况下，没有指定密钥与对应的NTP服务器关联
将指定密钥与对应的IPv6 NTP服务器关联	**ntp-service ipv6 unicast-server** { *server-name* | *ipv6-address* } [**vpn-instance** *vpn-instance-name*] **authentication-keyid** *keyid*	

表1-12 配置服务器端的 NTP 验证

操作	命令	说明
进入系统视图	**system-view**	-
使能NTP身份验证功能	**ntp-service authentication enable**	缺省情况下，NTP身份验证功能处于关闭状态
配置NTP身份验证密钥	**ntp-service authentication-keyid** *keyid* **authentication-mode md5** { **cipher** | **simple** } *value*	缺省情况下，没有配置NTP身份验证密钥
配置指定密钥为可信密钥	**ntp-service reliable authentication-keyid** *keyid*	缺省情况下，没有指定可信密钥

客户端和服务器上进行不同的配置时，NTP验证结果有所不同，详细介绍请参见 表 1-13。其中，表格中的“-”表示不管此项是否配置。

表1-13 客户端和服务器上进行不同配置时的 NTP 验证结果

客户端			服务器		结果
是否使能身份验证功能	是否与服务器关联密钥	是否配置关联的验证密钥，并将其指定为可信密钥	是否使能身份验证功能	是否配置验证密钥，并将其指定为可信密钥	
是	是	是	是	是	身份验证成功，可以正常收发NTP报文
是	是	是	是	否	身份验证失败，不可以正常收发NTP报文
是	是	是	否	-	身份验证失败，不可以正常收发NTP报文
是	是	否	-	-	身份验证失败，不可以正常收发NTP报文
是	否	-	-	-	不进行身份验证，可以正常收发NTP报文

客户端			服务器		结果
是否使能身份验证功能	是否与服务器关联密钥	是否配置关联的验证密钥，并将其指定为可信密钥	是否使能身份验证功能	是否配置验证密钥，并将其指定为可信密钥	
否	-	-	-	-	不进行身份验证，可以正常收发NTP报文

1.6.2 配置对等体模式的NTP验证功能

配置对等体模式的 NTP 验证功能时，需要在主动对等体和被动对等体上都使能 NTP 验证功能、配置验证密钥、将验证密钥设为可信密钥，并在主动对等体上将可信密钥与被动对等体关联。主动对等体和被动对等体上配置的密钥 ID 和密钥值必须保持一致，否则会导致 NTP 验证失败。

表1-14 配置主动对等体的 NTP 验证

操作	命令	说明
进入系统视图	**system-view**	-
使能NTP身份验证功能	**ntp-service authentication enable**	缺省情况下，NTP身份验证功能处于关闭状态
配置NTP身份验证密钥	**ntp-service authentication-keyid** *keyid* **authentication-mode md5** { **cipher** | **simple** } *value*	缺省情况下，没有配置NTP身份验证密钥
配置指定密钥为可信密钥	**ntp-service reliable authentication-keyid** *keyid*	缺省情况下，没有指定可信密钥
将指定密钥与对应的被动对等体关联	**ntp-service unicast-peer** { *ip-address* | *peer-name* } [**vpn-instance** *vpn-instance-name*] **authentication-keyid** *keyid*	缺省情况下，没有将指定密钥与对应的被动对等体关联
将指定密钥与对应的IPv6被动对等体关联	**ntp-service ipv6 unicast-peer** { *ipv6-address* | *peer-name* } [**vpn-instance** *vpn-instance-name*] **authentication-keyid** *keyid*	

表1-15 配置被动对等体的 NTP 验证

操作	命令	说明
进入系统视图	**system-view**	-
使能NTP身份验证功能	**ntp-service authentication enable**	缺省情况下，NTP身份验证功能处于关闭状态
配置NTP身份验证密钥	**ntp-service authentication-keyid** *keyid* **authentication-mode md5** { **cipher** | **simple** } *value*	缺省情况下，没有配置NTP身份验证密钥
配置指定密钥为可信密钥	**ntp-service reliable authentication-keyid** *keyid*	缺省情况下，没有指定可信密钥

主动对等体和被动对等体上进行不同的配置时，NTP验证结果有所不同，详细介绍请参见 表 1-16。其中，表格中的“-”表示不管此项是否配置。

表1-16 主动对等体和被动对等体上进行不同配置时的 NTP 验证结果

	主动对等体			被动对等体		结果
	是否使能身份验证功能	是否与被动对等体关联密钥	是否配置关联的验证密钥，并将其指定为可信密钥	是否使能身份验证功能	是否配置验证密钥，并将其指定为可信密钥	
不考虑主动对等体和被动对等体的时钟层数	是	是	是	是	是	身份验证成功，可以正常收发NTP报文
	是	是	是	是	否	身份验证失败，不可以正常收发NTP报文
	是	是	是	否	-	身份验证失败，不可以正常收发NTP报文
	是	否	-	是	-	身份验证失败，不可以正常收发NTP报文
	是	否	-	否	-	不进行身份验证，可以正常收发NTP报文
	否	-	-	是	-	身份验证失败，不可以正常收发NTP报文
	否	-	-	否	-	不进行身份验证，可以正常收发NTP报文
主动对等体时钟层数大于被动对等体	是	是	否	-	-	身份验证失败，不可以正常收发NTP报文
被动对等体时钟层数大于主动对等体	是	是	否	是	-	身份验证失败，不可以正常收发NTP报文
	是	是	否	否	-	不进行身份验证，可以正常收发NTP报文

1.6.3 配置广播模式的NTP验证功能

配置广播模式的 NTP 验证功能时，需要在广播客户端和广播服务器上都使能 NTP 验证功能、配置验证密钥、将验证密钥设为可信密钥，并在广播服务器上指定与该服务器关联的密钥。广播服务器和广播客户端上配置的密钥 ID 和密钥值必须保持一致，否则会导致 NTP 验证失败。

表1-17 配置广播客户端的 NTP 验证

操作	命令	说明
进入系统视图	**system-view**	-
使能NTP身份验证功能	**ntp-service authentication enable**	缺省情况下，NTP身份验证功能处于关闭状态

操作	命令	说明
配置NTP身份验证密钥	**ntp-service authentication-keyid** *keyid* **authentication-mode md5** { **cipher** | **simple** } *value*	缺省情况下，没有配置NTP身份验证密钥
配置指定密钥为可信密钥	**ntp-service reliable authentication-keyid** *keyid*	缺省情况下，没有指定可信密钥

表1-18 配置广播服务器端的 NTP 验证

操作	命令	说明
进入系统视图	**system-view**	-
使能NTP身份验证功能	**ntp-service authentication enable**	缺省情况下，NTP身份验证功能处于关闭状态
配置NTP身份验证密钥	**ntp-service authentication-keyid** *keyid* **authentication-mode md5** { **cipher** | **simple** } *value*	缺省情况下，没有配置NTP身份验证密钥
配置指定密钥为可信密钥	**ntp-service reliable authentication-keyid** *keyid*	缺省情况下，没有指定可信密钥
进入接口视图	**interface** *interface-type interface-number*	-
将指定密钥与对应的广播服务器关联	**ntp-service broadcast-server authentication-keyid** *keyid*	缺省情况下，广播服务器没有与密钥关联

广播客户端和广播服务器上进行不同的配置时，NTP验证结果有所不同，详细介绍请参见 表 1-19。其中，表格中的“-”表示不管此项是否配置。

表1-19 广播客户端和广播服务器上进行不同配置时的 NTP 验证结果

广播服务器			广播客户端		结果
是否使能身份验证功能	是否与广播服务器关联密钥	是否配置关联的验证密钥，并将其指定为可信密钥	是否使能身份验证功能	是否配置验证密钥，并将其指定为可信密钥	
是	是	是	是	是	身份验证成功，可以正常收发NTP报文
是	是	是	是	否	身份验证失败，不可以正常收发NTP报文
是	是	是	否	-	身份验证失败，不可以正常收发NTP报文
是	是	否	是	-	身份验证失败，不可以正常收发NTP报文
是	是	否	否	-	不进行身份验证，可以正常收发NTP报文
是	否	-	是	-	身份验证失败，不可以正常收发NTP报文

广播服务器			广播客户端		结果
是否使能身份验证功能	是否与广播服务器关联密钥	是否配置关联的验证密钥，并将其指定为可信密钥	是否使能身份验证功能	是否配置验证密钥，并将其指定为可信密钥	
是	否	-	否	-	不进行身份验证，可以正常收发NTP报文
否	-	-	是	-	身份验证失败，不可以正常收发NTP报文
否	-	-	否	-	不进行身份验证，可以正常收发NTP报文

1.6.4 配置组播模式的NTP验证功能

配置组播模式的 NTP 验证功能时，需要在组播客户端和组播服务器上都使能 NTP 验证功能、配置验证密钥、将验证密钥设为可信密钥，并在组播服务器上指定与该服务器关联的密钥。组播服务器和组播客户端上配置的密钥 ID 和密钥值必须保持一致，否则会导致 NTP 验证失败。

表1-20 配置组播客户端的 NTP 验证

操作	命令	说明
进入系统视图	**system-view**	-
使能NTP身份验证功能	**ntp-service authentication enable**	缺省情况下，NTP身份验证功能处于关闭状态
配置NTP身份验证密钥	**ntp-service authentication-keyid** *keyid* **authentication-mode md5** { **cipher** | **simple** } *value*	缺省情况下，没有配置NTP身份验证密钥
配置指定密钥为可信密钥	**ntp-service reliable authentication-keyid** *keyid*	缺省情况下，没有指定可信密钥

表1-21 配置组播服务器端的 NTP 验证

操作	命令	说明
进入系统视图	**system-view**	-
使能NTP身份验证功能	**ntp-service authentication enable**	缺省情况下，NTP身份验证功能处于关闭状态
配置NTP身份验证密钥	**ntp-service authentication-keyid** *keyid* **authentication-mode md5** { **cipher** | **simple** } *value*	缺省情况下，没有配置NTP身份验证密钥
配置指定密钥为可信密钥	**ntp-service reliable authentication-keyid** *keyid*	缺省情况下，没有指定可信密钥
进入接口视图	**interface** *interface-type interface-number*	-
将指定密钥与对应的组播服务器关联	**ntp-service multicast-server** [*ip-address*] **authentication-keyid** *keyid*	缺省情况下，组播服务器没有与密钥关联

操作	命令	说明
将指定密钥与对应的IPv6组播服务器关联	**ntp-service ipv6 multicast-server** *ipv6-multicast-address* **authentication-keyid** *keyid*	

组播客户端和组播服务器上进行不同的配置时，NTP验证结果有所不同，详细介绍请参见 表 1-22。其中，表格中的“-”表示不管此项是否配置。

表1-22 组播客户端和组播服务器上进行不同配置时的 NTP 验证结果

组播服务器			组播客户端		结果
是否使能身份验证功能	是否与组播服务器关联密钥	是否配置关联的验证密钥，并将其指定为可信密钥	是否使能身份验证功能	是否配置验证密钥，并将其指定为可信密钥	
是	是	是	是	是	身份验证成功，可以正常收发NTP报文
是	是	是	是	否	身份验证失败，不可以正常收发NTP报文
是	是	是	否	-	身份验证失败，不可以正常收发NTP报文
是	是	否	是	-	身份验证失败，不可以正常收发NTP报文
是	是	否	否	-	不进行身份验证，可以正常收发NTP报文
是	否	-	是	-	身份验证失败，不可以正常收发NTP报文
是	否	-	否	-	不进行身份验证，可以正常收发NTP报文
否	-	-	是	-	身份验证失败，不可以正常收发NTP报文
否	-	-	否	-	不进行身份验证，可以正常收发NTP报文

1.7 配置NTP可选参数

1.7.1 配置NTP报文的源接口

如果指定了 NTP 报文的源接口，则设备在主动发送 NTP 报文时，NTP 报文的源地址为指定的源接口的地址。建议将 Loopback 接口指定为源接口，以避免设备上某个接口的状态变化而导致 NTP 报文无法接收。

设备对接收到的 NTP 请求报文进行应答时，应答报文的源地址始终为接收到的 NTP 请求报文的目的地址。

配置 NTP 报文的源接口时，需要注意：

- 如果在命令 **ntp-service** [**ipv6**] **unicast-server** 或 **ntp-service** [**ipv6**] **unicast-peer** 中指定了NTP报文的源接口，则优先使用**ntp-service** [**ipv6**] **unicast-server**或**ntp-service** [**ipv6**] **unicast-peer** 命令指定的接口作为 NTP 报文的源接口。
- 如果在接口视图下配置了 **ntp-service broadcast-server** 或 **ntp-service** [**ipv6**] **multicast-server**，则 NTP 广播或组播模式报文的源接口为配置了 **ntp-service broadcast-server** 或 **ntp-service** [**ipv6**] **multicast-server** 命令的接口。

表1-23 配置 NTP 报文的源接口

操作	命令	说明
进入系统视图	**system-view**	-
配置NTP报文的源接口	**ntp-service source** *interface-type interface-number*	缺省情况下，没有指定NTP报文的源接口
配置IPv6 NTP报文的源接口	**ntp-service ipv6 source** *interface-type interface-number*	

1.7.2 配置接口不处理收到的NTP报文

启动 NTP 服务后，缺省情况下所有接口都可以处理收到的 NTP 报文。如果出于安全性、简化网络管理等方面的考虑，不希望设备为某个接口对应网段内的对端设备提供时间同步，或不希望设备从某个接口对应网段内的对端设备获得时间同步，则可以在该接口上执行本配置，使该接口不处理收到的 NTP 报文。

表1-24 配置接口不处理收到的 NTP 报文

操作	命令	说明
进入系统视图	**system-view**	-
进入接口视图	**interface** *interface-type interface-number*	-
配置接口不处理收到的NTP报文	**ntp-service inbound disable**	缺省情况下，接口处理收到的NTP报文和IPv6 NTP报文
配置接口不处理收到的IPv6 NTP报文	**ntp-service ipv6 inbound disable**	

1.7.3 配置动态会话的最大数目

NTP 会话分为两种：

- 静态会话：用户手动配置 NTP 相关命令而建立的会话。
- 动态会话：NTP 协议运行过程中建立的临时会话，若系统长期没有报文交互就会删除该临时会话。

各种模式中，会话的建立情况如下：

- 客户端/服务器模式中，在客户端上指定了 NTP 服务器后，客户端上会建立一个静态会话，服务器端在收到报文之后只是被动的响应报文，而不会建立会话（包括静态和动态会话）。

- 对等体模式中，在主动对等体上指定了被动对等体后，主动对等体上会建立静态会话，被动对等体端会建立动态会话。
- 广播模式和组播模式中，在广播/组播服务器端上会建立静态会话，而在广播/组播客户端上会建立动态会话。

设备同一时间内最多可以建立的会话数目为 128 个，其中包括静态会话数和动态会话数。

本配置用来限制动态会话的数目，以避免设备上维护过多的动态会话，占用过多的系统资源。

表1-25 配置动态会话的最大数目

操作	命令	说明
进入系统视图	**system-view**	-
配置NTP动态会话的最大数目	**ntp-service max-dynamic-sessions** *number*	缺省情况下，NTP动态会话的最大数目为100

1.7.4 配置NTP报文的DSCP优先级

DSCP 优先级用来体现报文自身的优先等级，决定报文传输的优先程度。通过本配置可以指定 NTP/IPv6 NTP 服务器发送的 NTP 报文的 DSCP 优先级。

表1-26 配置 NTP 报文的 DSCP 优先级

操作	命令	说明
进入系统视图	**system-view**	-
配置NTP报文的DSCP优先级	**ntp-service dscp** *dscp-value*	缺省情况下，NTP报文的DSCP优先级为48，IPv6 NTP报文的DSCP优先级为56
配置IPv6 NTP报文的DSCP优先级	**ntp-service ipv6 dscp** *dscp-value*	

1.8 配置本地时钟作为参考时钟

配置本地时钟作为参考时钟时，需要注意：

- 配置本地时钟作为参考时钟后，本地设备的时钟将处于同步状态，可以作为时间服务器为网络中其他设备的时钟提供时间同步。如果本地设备的时钟不正确，则会导致网络中设备的时间错误，请谨慎使用本配置。
- 在执行本命令之前，建议先调整本地系统时间。

本配置用来指定设备与本地时钟进行时间同步，使得该设备的时钟处于同步状态。

表1-27 配置本地时钟作为参考时钟

操作	命令	说明
进入系统视图	**system-view**	-
配置本地时钟作为参考时钟	**ntp-service refclock-master** [*ip-address*] [*stratum*]	缺省情况下，设备未采用本地时钟作为参考时钟

1.9 NTP显示和维护

在完成上述配置后，在任意视图下执行 **display** 命令可以显示配置后 NTP 的运行情况，通过查看显示信息验证配置的效果。

表1-28 NTP 显示与维护

操作	命令
显示NTP服务的所有IPv6会话信息	**display ntp-service ipv6 sessions** [**verbose**]
显示NTP服务的所有IPv4会话信息	**display ntp-service sessions** [**verbose**]
显示NTP服务的状态信息	**display ntp-service status**
显示从本地设备回溯到主时间服务器的各个NTP时间服务器的简要信息	**display ntp-service trace**

1.10 NTP典型配置举例

1.10.1 配置NTP客户端/服务器模式

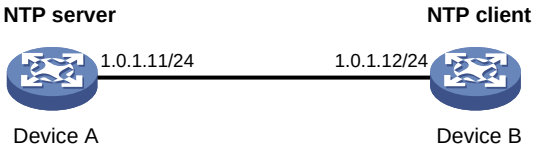
1. 组网需求

为了通过 NTP 实现 Device B 与 Device A 的时间同步，要求：

- 在 Device A 上设置本地时钟作为参考时钟，层数为 2；
- 配置 Device B 工作在客户端模式，指定 Device A 为 NTP 服务器。

2. 组网图

图1-5 配置 NTP 客户端/服务器模式组网图



3. 配置步骤

(1) 按照 图 1-5 配置各接口的IP地址，具体配置过程略。

(2) 配置 Device A

开启 NTP 服务。

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

设置本地时钟作为参考时钟，层数为 2。

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

(3) 配置 Device B

开启 NTP 服务。

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

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

设置 Device A 为 Device B 的 NTP 服务器。

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

4. 验证配置

完成上述配置后，Device B 向 Device A 进行时间同步。同步后查看 Device B 的 NTP 状态。可以看出，Device B 已经与 Device A 同步，层数比 Device A 的层数大 1，为 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^-10
 Root delay: 0.00383 ms
 Root dispersion: 16.26572 ms
 Reference time: d0c6033f.b9923965  Wed, Dec 29 2010 18:58:07.724
```

查看 Device B 的 NTP 服务的所有 IPv4 会话信息，可以看到 Device B 与 Device A 建立了会话。

```
[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
```

1.10.2 配置IPv6 NTP客户端/服务器模式

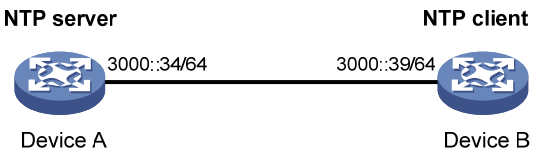
1. 组网需求

为了通过 IPv6 NTP 实现 Device B 与 Device A 的时间同步，要求：

- 在 Device A 上设置本地时钟作为参考时钟，层数为 2；
- 配置 Device B 工作在客户端模式，指定 Device A 为 IPv6 NTP 服务器。

2. 组网图

图1-6 配置 IPv6 NTP 客户端/服务器模式组网图



3. 配置步骤

(1) 按照 图 1-6 配置各接口的IP地址，具体配置过程略。

(2) 配置 Device A

开启 NTP 服务。

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

设置本地时钟作为参考时钟，层数为 2。

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

(3) 配置 Device B

开启 NTP 服务。

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

设置 Device A 为 Device B 的 IPv6 NTP 服务器。

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

4. 验证配置

完成上述配置后，Device B 向 Device A 进行时间同步。同步后查看 Device B 的 NTP 状态。可以看出，Device B 已经与 Device A 同步，层数比 Device A 的层数大 1，为 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^-10
 Root delay: 0.02649 ms
 Root dispersion: 12.24641 ms
 Reference time: d0c60419.9952fb3e  Wed, Dec 29 2010 19:01:45.598
```

查看 Device B 的 NTP 服务的所有 IPv6 会话信息，可以看到 Device B 与 Device A 建立了会话。

```
[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
```

1.10.3 配置NTP对等体模式

1. 组网需求

网络中存在时间服务器 Device A。为了通过 NTP 实现 Device B 与 Device A 进行时间同步，要求：

- 在 Device A 上设置本地时钟作为参考时钟，层数为 2；

- 配置 Device A 工作在对等体模式，指定 Device B 为被动对等体，即 Device A 为主动对等体，Device B 为被动对等体。

2. 组网图

图1-7 配置 NTP 对等体模式组网图

3. 配置步骤

(1) 按照 图 1-7 配置各接口的IP地址，并确保路由可达，具体配置过程略。

(2) 配置 Device B

开启 NTP 服务。

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

(3) 配置 Device A

开启 NTP 服务。

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

设置本地时钟作为参考时钟，层数为 2。

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

设置 Device B 为被动对等体。Device A 处于主动对等体模式。

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

4. 验证配置

完成上述配置后，Device B 选择 Device A 作为参考时钟，与 Device A 进行时间同步。同步后查看 Device B 的状态。可以看出，Device B 已经与 Device A 同步，层数比 Device A 的层数大 1，为 3。

```
[DeviceB] display ntp-service status
 Clock status: synchronized
 Clock stratum: 3
 System peer: 3.0.1.31
 Local mode: sym_passive
 Reference clock ID: 3.0.1.31
 Leap indicator: 00
 Clock jitter: 0.000916 s
 Stability: 0.000 pps
 Clock precision: 2^-17
 Root delay: 0.00609 ms
 Root dispersion: 1.95859 ms
 Reference time: 83aec681.deb6d3e5  Wed, Jan  8 2014 14:33:11.081
```

查看 Device B 的 NTP 服务的 IPv4 会话信息，可以看到 Device B 与 Device A 建立了会话。

```
[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
```

1.10.4 配置IPv6 NTP对等体模式

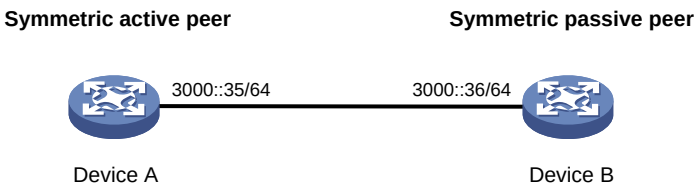
1. 组网需求

网络中存在时间服务器 Device A。为了通过 IPv6 NTP 实现 Device B 与 Device A 进行时间同步，要求：

- 在 Device A 上设置本地时钟作为参考时钟，层数为 2；
- 配置 Device A 工作在对等体模式，指定 Device B 为被动对等体，即 Device A 为主动对等体，Device B 为被动对等体。

2. 组网图

图1-8 配置 IPv6 NTP 对等体模式组网图



3. 配置步骤

(1) 按照 图 1-8 配置各接口的IP地址，并确保路由可达，具体配置过程略。

(2) 配置 Device B

开启 NTP 服务。

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

(3) 配置 Device A

开启 NTP 服务。

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

设置本地时钟作为参考时钟，层数为 2。

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

设置 Device B 为 IPv6 被动对等体。Device A 处于主动对等体模式。

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

4. 验证配置

完成上述配置后，Device B 选择 Device A 作为参考时钟，与 Device A 进行时间同步。同步后查看 Device B 的状态。可以看出，Device B 已经与 Device A 同步，层数比 Device A 的层数大 1，为 3。

```
[DeviceB] display ntp-service status
```

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

查看 Device B 的 NTP 服务的 IPv6 会话信息，可以看到 Device B 与 Device A 建立了会话。

```
[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
```

1.10.5 配置NTP广播模式

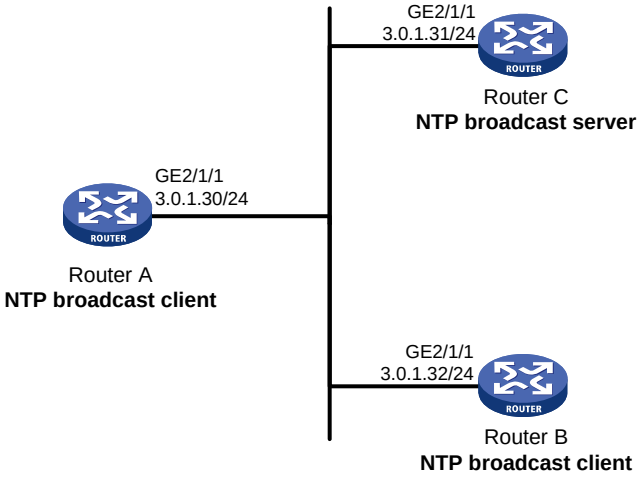
1. 组网需求

为了实现 Router C 作为同一网段中多个设备的时间服务器，同时同步多个设备的时间，要求：

- 在 Router C 上设置本地时钟作为参考时钟，层数为 2；
- Router C 工作在广播服务器模式，从接口 GigabitEthernet2/1/1 向外广播发送 NTP 报文；
- Router A 和 Router B 工作在广播客户端模式，分别从各自的接口 GigabitEthernet2/1/1 监听 NTP 广播报文。

2. 组网图

图1-9 配置 NTP 广播模式组网图



3. 配置步骤

(1) 按照 图 1-9 配置各接口的IP地址，具体配置过程略。

(2) 配置 Router C

开启 NTP 服务。

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

设置本地时钟作为参考时钟，层数为 2。

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

设置 Router C 为广播服务器，从接口 GigabitEthernet2/1/1 发送广播报文。

```
[RouterC] interface gigabitethernet 2/1/1
[RouterC-GigabitEthernet2/1/1] ntp-service broadcast-server
```

(3) 配置 Router A

开启 NTP 服务。

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

设置 Router A 为广播客户端，从接口 GigabitEthernet2/1/1 监听广播报文。

```
[RouterA] interface gigabitethernet 2/1/1
[RouterA-GigabitEthernet2/1/1] ntp-service broadcast-client
```

(4) 配置 Router B

开启 NTP 服务。

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

设置 Router B 为广播客户端，从接口 GigabitEthernet2/1/1 监听广播报文。

```
[RouterB] interface gigabitethernet 2/1/1
[RouterB-GigabitEthernet2/1/1] ntp-service broadcast-client
```

4. 验证配置

Router A 和 Router B 接收到 Router C 发出的广播报文后，与其同步。以 Router A 为例，同步后查看 Router A 的状态。可以看出，Router A 已经与 Router C 同步，层数比 Router C 的层数大 1，为 3。

```
[RouterA-GigabitEthernet2/1/1] display ntp-service status
 Clock status: synchronized
 Clock stratum: 3
 System peer: 3.0.1.31
 Local mode: bclient
 Reference clock ID: 3.0.1.31
 Leap indicator: 00
 Clock jitter: 0.044281 s
 Stability: 0.000 pps
 Clock precision: 2^-10
 Root delay: 0.00229 ms
 Root dispersion: 4.12572 ms
 Reference time: d0d289fe.ec43c720  Sat, Jan  8 2011  7:00:14.922
```

查看 Router A 的 NTP 服务的所有 IPv4 会话信息，可以看到 Router A 与 Router C 建立了会话。

```
[RouterA-GigabitEthernet2/1/1] 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
```

1.10.6 配置NTP组播模式

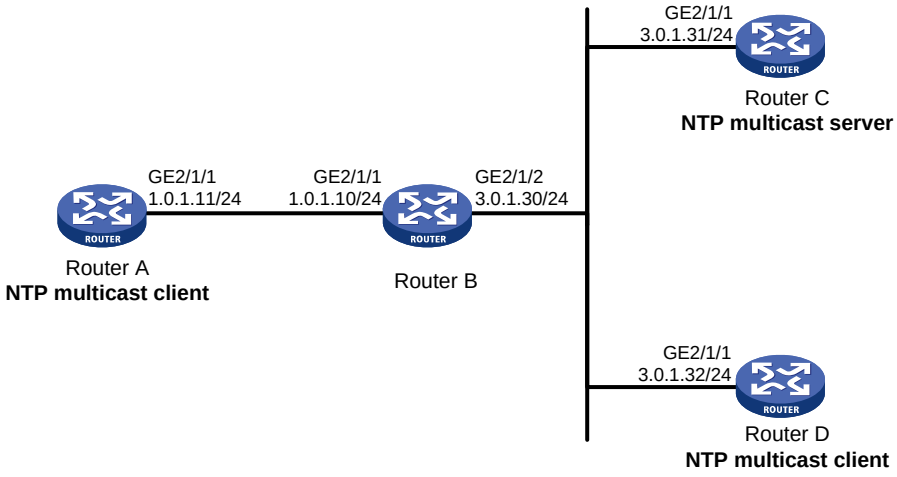
1. 组网需求

为了实现 Router C 作为不同网段中多个设备的时间服务器，同时同步多个设备的时间，要求：

- 在 Router C 上设置本地时钟作为参考时钟，层数为 2；
- Router C 工作在组播服务器模式，从接口 GigabitEthernet2/1/1 向外组播发送 NTP 报文；
- Router A 和 Router D 工作在组播客户端模式，分别从各自的接口 GigabitEthernet2/1/1 监听 NTP 组播报文。

2. 组网图

图1-10 配置 NTP 组播模式组网图



3. 配置步骤

(1) 按照 图 1-10 配置各接口的IP地址，并确保路由可达，具体配置过程略。

(2) 配置 Router C

开启 NTP 服务。

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

设置本地时钟作为参考时钟，层数为 2。

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

设置 Router C 为组播服务器，从接口 GigabitEthernet2/1/1 发送组播报文。

```
[RouterC] interface gigabitethernet 2/1/1
[RouterC-GigabitEthernet2/1/1] ntp-service multicast-server
```

(3) 配置 Router D

开启 NTP 服务。

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

设置 Router D 为组播客户端，从接口 GigabitEthernet2/1/1 监听组播报文。

```
[RouterD] interface gigabitethernet 2/1/1
[RouterD-GigabitEthernet2/1/1] ntp-service multicast-client
```

(4) 验证配置一

由于 Router D 和 Router C 在同一个网段，不需要配置组播功能，Router D 就可以收到 Router C 发出的组播报文，并与其同步。同步后查看 Router D 的状态。可以看出，Router D 已经与 Router C 同步，层数比 Router C 的层数大 1，为 3。

```
[RouterD-GigabitEthernet2/1/1] display ntp-service status
 Clock status: synchronized
 Clock stratum: 3
 System peer: 3.0.1.31
 Local mode: bclient
```

```
 Reference clock ID: 3.0.1.31
 Leap indicator: 00
 Clock jitter: 0.044281 s
 Stability: 0.000 pps
 Clock precision: 2^-10
 Root delay: 0.00229 ms
 Root dispersion: 4.12572 ms
 Reference time: d0d289fe.ec43c720  Sat, Jan  8 2011  7:00:14.922
```

查看 Router D 的 NTP 服务的所有 IPv4 会话信息，可以看到 Router D 与 Router C 建立了会话。

```
[RouterD-GigabitEthernet2/1/1] 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) 配置 Router B

由于 Router A 与 Router C 不在同一网段，所以 Router B 上需要配置组播功能，否则 Router A 收不到 Router C 发出的组播报文。

配置组播功能。

```
<RouterB> system-view
[RouterB] multicast routing
[RouterB-mrib] quit
[RouterB] interface gigabitethernet 2/1/1
[RouterB-GigabitEthernet2/1/1] igmp enable
[RouterB-GigabitEthernet2/1/1] igmp static-group 224.0.1.1
[RouterB-GigabitEthernet2/1/1] quit
[RouterB] interface gigabitethernet 2/1/2
[RouterB-GigabitEthernet2/1/2] pim dm
```

(6) 配置 Router A

开启 NTP 服务。

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

设置 Router A 为组播客户端，从接口 GigabitEthernet2/1/1 监听组播报文。

```
[RouterA] interface gigabitethernet 2/1/1
[RouterA-GigabitEthernet2/1/1] ntp-service multicast-client
```

(7) 验证配置二

同步后查看 Router A 的状态。可以看出，Router A 已经与 Router C 同步，层数比 Router C 的层数大 1，为 3。

```
[RouterA-GigabitEthernet2/1/1] display ntp-service status
 Clock status: synchronized
 Clock stratum: 3
 System peer: 3.0.1.31
 Local mode: bclient
 Reference clock ID: 3.0.1.31
 Leap indicator: 00
```

```
 Clock jitter: 0.165741 s
 Stability: 0.000 pps
 Clock precision: 2^-10
 Root delay: 0.00534 ms
 Root dispersion: 4.51282 ms
 Reference time: d0c61289.10b1193f  Wed, Dec 29 2010 20:03:21.065
```

查看 Router A 的 NTP 服务的所有 IPv4 会话信息，可以看到 Router A 与 Router C 建立了会话。

```
[RouterA-GigabitEthernet2/1/1] 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
```

1.10.7 配置IPv6 NTP组播模式

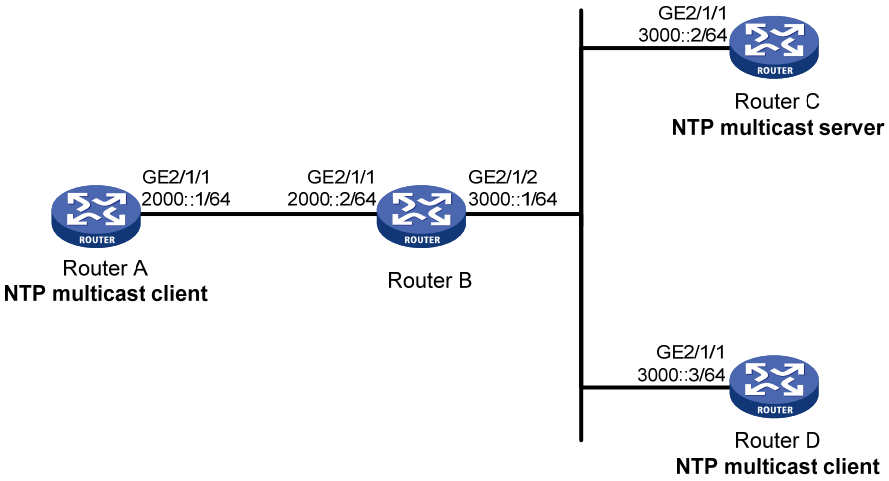
1. 组网需求

为了实现 Router C 作为不同网段中多个设备的时间服务器，同时同步多个设备的时间，要求：

- 在 Router C 上设置本地时钟作为参考时钟，层数为 2；
- Router C 工作在 IPv6 组播服务器模式，从接口 GigabitEthernet2/1/1 向外组播发送 IPv6 NTP 报文；
- Router A 和 Router D 工作在 IPv6 组播客户端模式，分别从各自的接口 GigabitEthernet2/1/1 监听 IPv6 NTP 组播报文。

2. 组网图

图1-11 配置 IPv6 NTP 组播模式组网图



3. 配置步骤

(1) 按照 图 1-11 配置各个接口的IP地址，并确保路由可达，具体配置步骤略。

(2) 配置 Router C

开启 NTP 服务。

```
<RouterC> system-view
[RouterC] ntp-service enable
```
设置本地时钟作为参考时钟，层数为 2。
```
[RouterC] ntp-service refclock-master 2
```
设置 Router C 为 IPv6 组播服务器，从接口 GigabitEthernet2/1/1 向组播地址 FF24::1 发送 NTP 报文。
```
[RouterC] interface gigabitethernet 2/1/1
[RouterC-GigabitEthernet2/1/1] ntp-service ipv6 multicast-server ff24::1
```
(3) 配置 Router D

开启 NTP 服务。
```
<RouterD> system-view
[RouterD] ntp-service enable
```
设置 Router D 为 IPv6 组播客户端，在接口 GigabitEthernet2/1/1 监听目的地址为 FF24::1 的 NTP 组播报文。
```
[RouterD] interface gigabitethernet 2/1/1
[RouterD-GigabitEthernet2/1/1] ntp-service ipv6 multicast-client ff24::1
```
(4) 验证配置一

由于 Router D 和 Router C 在同一个网段，不需要配置 IPv6 组播功能，Router D 就可以收到 Router C 发出的 IPv6 组播报文，并与其同步。同步后查看 Router D 的状态。可以看出，Router D 已经与 Router C 同步，层数比 Router C 的层数大 1，为 3。
```
[RouterD-GigabitEthernet2/1/1] display ntp-service status
 Clock status: synchronized
 Clock stratum: 3
 System peer: 3000::2
 Local mode: bclient
 Reference clock ID: 165.84.121.65
 Leap indicator: 00
 Clock jitter: 0.000977 s
 Stability: 0.000 pps
 Clock precision: 2^-10
 Root delay: 0.00000 ms
 Root dispersion: 8.00578 ms
 Reference time: d0c60680.9754fb17  Wed, Dec 29 2010 19:12:00.591
```
查看 Router D 的 NTP 服务的所有 IPv6 会话信息，可以看到 Router D 与 Router C 建立了会话。
```
[RouterD-GigabitEthernet2/1/1] 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) 配置 Router B

由于 Router A 与 Router C 不在同一网段，所以 Router B 上需要配置 IPv6 组播功能，否则 Router A 收不到 Router C 发出的 IPv6 组播报文。

配置 IPv6 组播功能。

```
<RouterB> system-view
[RouterB] ipv6 multicast routing
[RouterB-mrib6] quit
[RouterB] interface gigabitethernet 2/1/1
[RouterB-GigabitEthernet2/1/1] mld enable
[RouterB-GigabitEthernet2/1/1] mld static-group ff24::1
[RouterB-GigabitEthernet2/1/1] quit
[RouterB] interface gigabitethernet 2/1/2
[RouterB-GigabitEthernet2/1/2] ipv6 pim dm
```

(6) 配置 Router A

开启 NTP 服务。

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

设置 Router A 为 IPv6 组播客户端，在接口 GigabitEthernet2/1/1 监听目的地址为 FF24::1 的 NTP 组播报文。

```
[RouterA] interface gigabitethernet 2/1/1
[RouterA-GigabitEthernet2/1/1] ntp-service ipv6 multicast-client ff24::1
```

(7) 验证配置二

同步后查看 Router A 的状态。可以看出，Router A 已经与 Router C 同步，层数比 Router C 的层数大 1，为 3。

```
[RouterA-GigabitEthernet2/1/1] display ntp status
 Clock status: synchronized
 Clock stratum: 3
 System peer: 3000::2
 Local mode: bclient
 Reference clock ID: 165.84.121.65
 Leap indicator: 00
 Clock jitter: 0.165741 s
 Stability: 0.000 pps
 Clock precision: 2^-10
 Root delay: 0.00534 ms
 Root dispersion: 4.51282 ms
 Reference time: d0c61289.10b1193f  Wed, Dec 29 2010 20:03:21.065
```

查看 Router A 的 NTP 服务的 IPv6 会话信息，可以看到 Router A 与 Router C 建立了会话。

```
[RouterA-GigabitEthernet2/1/1] 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
```

1.10.8 配置带验证功能的NTP客户端/服务器模式

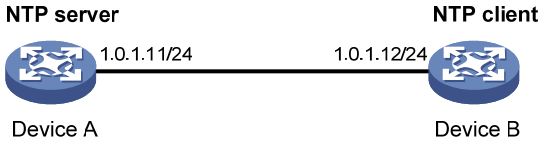
1. 组网需求

为了通过 NTP 实现 Device B 与 Device A 的时间同步，并保证时间同步的安全性，要求：

- 在 Device A 上设置本地时钟作为参考时钟，层数为 2；
- Device B 工作在客户端模式，指定 Device A 为 NTP 服务器；
- Device A 和 Device B 上同时配置 NTP 验证。

2. 组网图

图1-12 配置带身份验证的 NTP 客户端/服务器模式组网图



3. 配置步骤

(1) 按照 图 1-12 配置各接口的IP地址，具体配置过程略。

(2) 配置 Device A 的本地时钟作为参考时钟

开启 NTP 服务。

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

设置本地时钟作为参考时钟，层数为 2。

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

(3) 配置 Device B

开启 NTP 服务。

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

在 Device B 上启动 NTP 验证功能。

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

创建编号为 42 的 NTP 验证密钥，密钥值为 aNiceKey，以明文形式输入。

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

配置编号为 42 的密钥为可信密钥。

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

设置 Device A 为 Device B 的 NTP 服务器，并将该服务器与编号为 42 的密钥关联。

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

以上配置将使得 Device B 与 Device A 进行时间同步，但由于 Device A 没有使能 NTP 身份验证，所以，Device B 还是无法与 Device A 同步。

(4) 在 Device A 上配置 NTP 验证功能。

在 Device A 上启动 NTP 验证功能。

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

创建编号为 42 的 NTP 验证密钥，密钥值为 aNiceKey，以明文形式输入。

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

配置编号为 42 的密钥为可信密钥。

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

4. 验证配置

完成上述配置后，Device B 可以与 Device A 的时间同步。同步后查看 Device B 的状态。可以看出，Device B 已经与 Device A 同步，层数比 Device A 的层数大 1，为 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^-10
 Root delay: 0.00655 ms
 Root dispersion: 1.15869 ms
 Reference time: d0c62687.ab1bba7d  Wed, Dec 29 2010 21:28:39.668
```

查看 Device B 的 NTP 服务的所有 IPv4 会话信息，可以看到 Device B 与 Device A 建立了会话。

```
[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
```

1.10.9 配置带验证功能的NTP广播模式

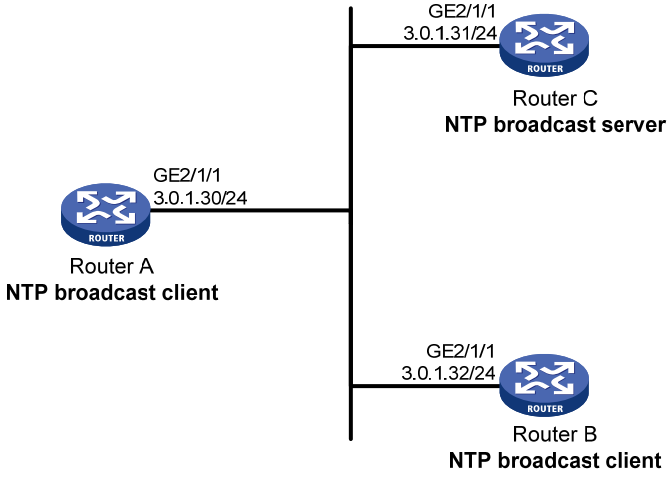
1. 组网需求

Router C 作为同一网段中多个设备的时间服务器，同时同步多个设备的时间。Router A 和 Router B 要求对时间服务器进行验证，以保证时间同步的安全性。为了实现上述需求，要求：

- 在 Router C 上设置本地时钟作为参考时钟，层数为 3；
- Router C 工作在广播服务器模式，从接口 GigabitEthernet2/1/1 向外广播发送 NTP 报文；
- Router A 和 Router B 工作在广播客户端模式，从接口 GigabitEthernet2/1/1 监听 NTP 广播报文；
- 在 Router A、Router B 和 Router C 上配置 NTP 验证功能。

2. 组网图

图1-13 配置带身份验证的 NTP 广播模式组网图



3. 配置步骤

(1) 按照 图 1-13 配置各接口的IP地址，具体配置过程略。

(2) 配置 Router A

开启 NTP 服务。

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

使能 NTP 验证功能，创建编号为 88 的 NTP 验证密钥，密钥值为 123456，以明文形式输入，并将密钥 88 指定为可信密钥。

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

设置 Router A 为 NTP 广播客户端，从接口 GigabitEthernet2/1/1 监听广播报文。

```
[RouterA] interface gigabitethernet 2/1/1
[RouterA-GigabitEthernet2/1/1] ntp-service broadcast-client
```

(3) 配置 Router B

开启 NTP 服务。

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

使能 NTP 验证功能，创建编号为 88 的 NTP 验证密钥，密钥值为 123456，以明文形式输入，并将密钥 88 指定为可信密钥。

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

设置 Router B 为 NTP 广播客户端，从接口 GigabitEthernet2/1/1 监听广播报文。

```
[RouterB] interface gigabitethernet 2/1/1
[RouterB-GigabitEthernet2/1/1] ntp-service broadcast-client
```

(4) 配置 Router C 作为 NTP 广播服务器

开启 NTP 服务。

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

设置本地时钟作为参考时钟，层数为 3。

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

设置 Router C 为 NTP 广播服务器，从接口 GigabitEthernet2/1/1 向外发送广播报文。

```
[RouterC] interface gigabitethernet 2/1/1
[RouterC-GigabitEthernet2/1/1] ntp-service broadcast-server
[RouterC-GigabitEthernet2/1/1] quit
```

(5) 验证配置一

由于 Router A 和 Router B 上使能了 NTP 验证功能，Router C 上没有使能 NTP 验证功能。因此，Router A 和 Router B 无法与 Router C 的时间同步。以 Router B 为例，查看 NTP 服务的状态。

```
[RouterB-GigabitEthernet2/1/1] display ntp-service status
 Clock status: unsynchronized
 Clock stratum: 16
 Reference clock ID: none
```

(6) 在 Router C 上配置 NTP 验证功能

在 Router C 上使能 NTP 验证功能，创建 ID 为 88 的 NTP 验证密钥，密钥值为 123456，以明文形式输入，并将密钥 88 指定为可信密钥。

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

设置 Router C 为 NTP 广播服务器并指定关联的密钥编号为 88。

```
[RouterC] interface gigabitethernet 2/1/1
[RouterC-GigabitEthernet2/1/1] ntp-service broadcast-server authentication-keyid 88
```

(7) 验证配置二

在 Router C 上使能 NTP 验证功能后，Router A 和 Router B 可以与 Router C 的时间同步。以 Router B 为例，查看 NTP 服务的状态信息，可以看到 Router B 已经与 Router C 同步，层数比 Router C 的层数大 1，为 4。

```
[RouterB-GigabitEthernet2/1/1] display ntp-service status
 Clock status: synchronized
 Clock stratum: 4
 System peer: 3.0.1.31
 Local mode: bclient
 Reference clock ID: 3.0.1.31
 Leap indicator: 00
 Clock jitter: 0.006683 s
 Stability: 0.000 pps
 Clock precision: 2^-10
 Root delay: 0.00127 ms
 Root dispersion: 2.89877 ms
 Reference time: d0d287a7.3119666f  Sat, Jan  8 2011  6:50:15.191
```

查看 Router B 的 NTP 服务的所有 IPv4 会话信息，可以看到 Router B 与 Router C 建立了会话。

```
[RouterB-GigabitEthernet2/1/1] 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
```

1.10.10 配置采用客户端/服务器模式实现MPLS VPN网络的时间同步

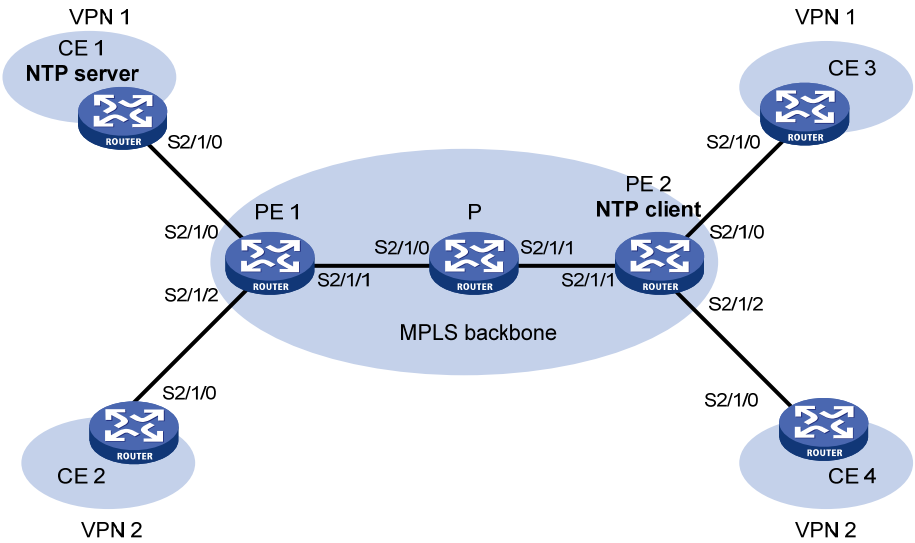
1. 组网需求

PE 1 和 PE 2 上同时存在两个 VPN：VPN 1 和 VPN 2。CE 1 和 CE 3 是 VPN 1 内的设备。为了通过 NTP 实现 PE 2 与 VPN 1 内的 CE 1 时间同步，要求：

- 在 CE 1 上设置本地时钟作为参考时钟，层数为 2；
- 采用客户端/服务器模式实现 PE 2 向 CE 1 进行时间同步，并指定 VPN 为 VPN 1。

2. 组网图

图1-14 配置 MPLS VPN 网络的时间同步组网图



设备	接口	IP地址	设备	接口	IP地址
CE 1	S2/1/0	10.1.1.1/24	PE 1	S2/1/0	10.1.1.2/24
CE 2	S2/1/0	10.2.1.1/24		S2/1/1	172.1.1.1/24
CE 3	S2/1/0	10.3.1.1/24		S2/1/2	10.2.1.2/24
CE 4	S2/1/0	10.4.1.1/24	PE 2	S2/1/0	10.3.1.2/24
P	S2/1/0	172.1.1.2/24		S2/1/1	172.2.1.2/24
	S2/1/1	172.2.1.1/24		S2/1/2	10.4.1.2/24

3. 配置步骤

说明

在下面的配置之前，MPLS VPN 的相关配置必须完成，CE 1 和 PE 1 之间、PE 1 和 PE 2 之间、PE 2 和 CE 3 之间都必须有路由可达。MPLS VPN 的配置方法请参见“MPLS 配置指导”中的“MPLS L3VPN”。

(1) 按照 图 1-14 配置各接口的IP地址，并确保路由可达，具体配置过程略。

(2) 配置 CE 1

开启 NTP 服务。

```
<CE1> system-view
[CE1] ntp-service enable
```

设置本地时钟作为参考时钟，层数为 2。

```
[CE1] ntp-service refclock-master 2
```

(3) 配置 PE 2

开启 NTP 服务。

```
<PE2> system-view
[PE2] ntp-service enable
```

设置 VPN 1 中的 CE 1 为 PE 2 的 NTP 服务器。

```
[PE2] ntp-service unicast-server 10.1.1.1 vpn-instance vpn1
```

4. 验证配置

经过一段时间之后在 PE 2 上可以查看 NTP 服务的所有 IPv4 会话信息、状态信息等，可以看出 PE 2 已经同步到 CE 1 了，层数为 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^-10
 Root delay: 0.00655 ms
 Root dispersion: 1.15869 ms
 Reference time: d0c62687.ab1bba7d  Wed, Dec 29 2010 21:28:39.668
[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
[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
```

1.10.11 配置采用对等体模式实现MPLS VPN网络的时间同步

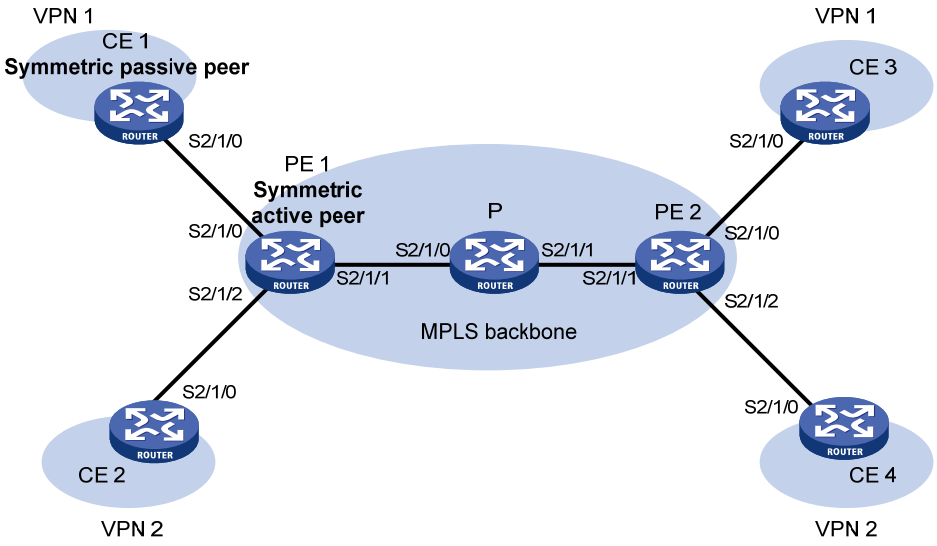
1. 组网需求

PE 1 和 PE 2 上同时存在两个 VPN：VPN 1 和 VPN 2。CE 1 和 CE 3 是 VPN 1 内的设备。为了通过 NTP 实现 PE 1 与 VPN 1 内的 CE 1 时间同步，要求：

- 在 CE 1 上设置本地时钟作为参考时钟，层数为 2；
- 采用对等体模式实现 PE 1 向 CE 1 进行时间同步，并指定 VPN 为 VPN 1。

2. 组网图

图1-15 配置采用对等体模式实现 MPLS VPN 网络的时间同步组网图



设备	接口	IP地址	设备	接口	IP地址
CE 1	S2/1/0	10.1.1.1/24	PE 1	S2/1/0	10.1.1.2/24
CE 2	S2/1/0	10.2.1.1/24		S2/1/1	172.1.1.1/24
CE 3	S2/1/0	10.3.1.1/24		S2/1/2	10.2.1.2/24
CE 4	S2/1/0	10.4.1.1/24	PE 2	S2/1/0	10.3.1.2/24
P	S2/1/0	172.1.1.2/24		S2/1/1	172.2.1.2/24
	S2/1/1	172.2.1.1/24		S2/1/2	10.4.1.2/24

3. 配置步骤

(1) 按照 图 1-15 配置各接口的IP地址，并确保路由可达，具体配置过程略。

(2) 配置 CE 1

开启 NTP 服务。

```
<CE1> system-view
[CE1] ntp-service enable
```

设置本地时钟作为参考时钟，层数为 2。

```
[CE1] ntp-service refclock-master 2
```

(3) 配置 PE 1

开启 NTP 服务。

```
<PE1> system-view
[PE1] ntp-service enable
```

设置 VPN 1 中的 CE 1 为 PE 1 的被动对等体。

```
[PE1] ntp-service unicast-peer 10.1.1.1 vpn-instance vpn1
```

4. 验证配置

经过一段时间之后在 PE 1 上可以查看 NTP 服务的所有 IPv4 会话信息、状态信息等，可以看出 PE 1 已经同步到 CE 1 了，层数为 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^-10
 Root delay: 0.00655 ms
 Root dispersion: 1.15869 ms
 Reference time: d0c62687.ab1bba7d  Wed, Dec 29 2010 21:28:39.668
[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
[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
```

2 SNTP

提示

设备上不能同时配置 NTP 和 SNTP 功能。

2.1 SNTP简介

NTP 时间同步过程中需要进行复杂的时钟优选运算，时间同步速度较慢，并且占用较多的系统资源。SNTP（Simple NTP，简单 NTP）是由 RFC 4330 定义的客户端版本的简单 NTP，采用与 NTP 相同的报文格式及交互过程，但简化了 NTP 的时间同步过程，以牺牲时间精度为代价实现了时间的快速同步，并减少了占用的系统资源。在时间精度要求不高的情况下，可以使用 SNTP 来实现时间同步。

SNTP 只支持客户端/服务器模式，在模式中提供客户端功能，即作为客户端，从 NTP 服务器获得时间同步，不能作为服务器为其他设备提供时间同步。

如果同时为 SNTP 客户端指定了多个 NTP 服务器，则 SNTP 客户端根据如下方法选择与哪个服务器的时间同步：

(1) 优先选择时钟层数值最小的 NTP 服务器。

(2) 如果时钟层数相同，则选择接收到的第一个 NTP 报文对应的 NTP 服务器。

与 SNTP 相关的协议规范有：

- RFC 4330：Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI

2.2 SNTP配置任务简介

表2-1 SNTP 配置任务简介

配置任务	说明	详细配置
开启SNTP服务	必选	2.3
为SNTP客户端指定NTP服务器	必选	2.4
配置SNTP验证功能	可选	2.5

2.3 开启SNTP服务

NTP 服务与 SNTP 服务互斥，同一时刻只能开启其中一个服务。

表2-2 开启 NTP 客户端

操作	命令	说明
进入系统视图	**system-view**	-
开启SNTP服务	**sntp enable**	缺省情况下，没有开启SNTP服务

2.4 为SNTP客户端指定NTP服务器

NTP 服务器的时钟只有处于同步状态时，才能作为时间服务器为 SNTP 客户端提供时间同步。当 NTP 服务器的时钟层数大于或等于客户端的时钟层数时，客户端将不会与其同步。

表2-3 为 SNTP 客户端指定 NTP 服务器

操作	命令	说明
进入系统视图	**system-view**	-
为设备指定NTP服务器	**sntp unicast-server** { *server-name* | *ip-address* } [**vpn-instance** *vpn-instance-name*] [**authentication-keyid** *keyid* | **source** *interface-type interface-number* | **version** *number*] *	缺省情况下，没有为设备指定NTP服务器 可以通过多次执行**sntp** [**ipv6**] **unicast-server**命令配置多个NTP服务器 **authentication-keyid**参数用来将指定密钥与对应的NTP服务器关联。使用验证功能时，需要指定本参数
为设备指定IPv6 NTP服务器	**sntp ipv6 unicast-server** { *server-name* | *ipv6-address* } [**vpn-instance** *vpn-instance-name*] [**authentication-keyid** *keyid* | **source** *interface-type interface-number*] *	

2.5 配置SNTP验证功能

在一些对时间同步安全性要求较高的网络中，运行 SNTP 协议时需要启用验证功能。通过客户端和服务器端的身份验证，保证客户端只与通过验证的服务器进行时间同步，提高了网络安全性。

要使 SNTP 验证功能正常工作，在配置 SNTP 验证功能时应注意以下原则：

- 在 NTP 服务器和 SNTP 客户端上都需要使能验证功能。
- NTP 服务器和 SNTP 客户端上必须配置相同的验证密钥（包括密钥 ID 和密钥值），并将密钥设为可信密钥。NTP 服务器上验证功能的配置方法，请参见“网络管理与监控配置指导”中的“NTP”。
- 在客户端需要将指定密钥与对应的 NTP 服务器关联。

如果客户端没有成功启用 SNTP 验证功能，不论服务器端是否使能验证功能，客户端均可以与服务器端同步。

表2-4 在 SNTP 客户端配置验证功能

操作	命令	说明
进入系统视图	**system-view**	-

操作	命令	说明
使能SNTP身份验证功能	**sntp authentication enable**	缺省情况下，SNTP身份验证功能处于关闭状态
配置SNTP身份验证密钥	**sntp authentication-keyid** *keyid* **authentication-mode md5** { **cipher** | **simple** } *value*	缺省情况下，没有配置SNTP身份验证密钥
配置指定密钥为可信密钥	**sntp reliable authentication-keyid** *keyid*	缺省情况下，没有指定可信密钥
将指定密钥与对应的NTP服务器关联	**sntp unicast-server** { *ip-address* | *server-name* } [**vpn-instance** *vpn-instance-name*] **authentication-keyid** *keyid*	缺省情况下，没有为设备指定NTP服务器
将指定密钥与对应的IPv6 NTP服务器关联	**sntp ipv6 unicast-server** { *ipv6-address* | *server-name* } [**vpn-instance** *vpn-instance-name*] **authentication-keyid** *keyid*	

2.6 SNTP显示和维护

在完成上述配置后，在任意视图下执行 **display** 命令可以显示配置后 SNTP 的运行情况，通过查看显示信息验证配置的效果。

表2-5 SNTP 显示与维护

操作	命令
显示SNTP服务维护的IPv6会话信息	**display sntp ipv6 sessions**
显示SNTP服务维护的IPv4会话信息	**display sntp sessions**

2.7 SNTP典型配置举例

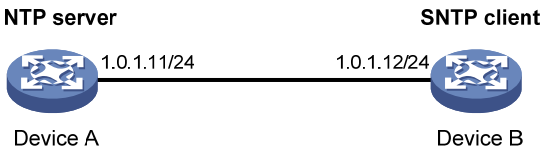
1. 组网需求

Device B 对时间精度要求不高。为了实现 Device B 与 Device A 的时间同步，要求：

- 在 Device A 上设置本地时钟作为参考时钟，层数为 2；
- Device B 工作在 SNTP 客户端模式，指定 Device A 为 NTP 服务器。
- Device B 要求对 NTP 服务器进行验证，以保证时间同步的安全性。

2. 组网图

图2-1 SNTP 配置组网图



3. 配置步骤

(1) 按照 图 2-1 配置各接口的IP地址，具体配置过程略。

(2) 配置 Device A

开启 NTP 服务。

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

设置本地时钟作为参考时钟，层数为 2。

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

在 Device A 上启动 NTP 验证功能。

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

创建编号为 10 的 NTP 验证密钥，密钥值为 aNiceKey，以明文形式输入。

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

设置编号为 10 的密钥为可信密钥。

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

(3) 配置 Device B

开启 SNTP 服务。

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

在 Device B 上启动 SNTP 验证功能。

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

创建编号为 10 的 SNTP 验证密钥，密钥值为 aNiceKey，以明文形式输入。

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

设置编号为 10 的密钥为可信密钥。

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

设置 Device A 为 Device B 的 NTP 服务器，并将该服务器与编号为 10 的密钥关联。

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

4. 验证配置

查看 Device B 的 SNTP 会话信息，可以看到 Device B 与 Device A 建立了会话，并且处于已同步状态。

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

目 录

1 PoE

说明

只有安装了 SIC-4FSWP、SIC 4GSWP、DSIC 9FSWP、HMIM 24GSW-POE 接口卡的 MSR 3600 的 PoE 款型和 MSR 5600 支持本特性。

1.1 PoE简介

1.1.1 PoE概述

PoE（Power over Ethernet，以太网供电），又称远程供电是指设备通过以太网电口，利用双绞线对外接 PD（Powered Device，受电设备）进行远程供电。

1. PoE相关概念

PoE系统如 图 1-1 所示，包括PoE电源、PSE（Power Sourcing Equipment，供电设备）、PI（Power Interface，电源接口）和PD。

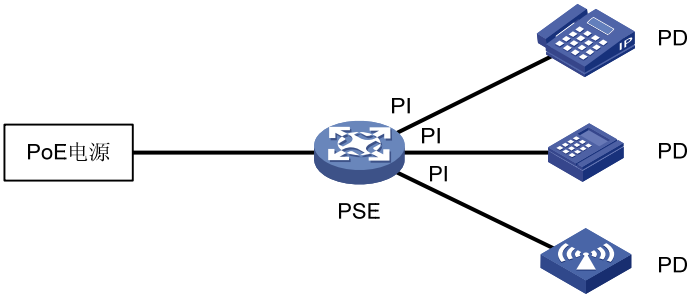
(1) PoE 电源为整个 PoE 系统提供功率。

(2) PSE：直接给 PD 供电的设备。PSE 支持的主要功能包括寻找、检测 PD，对 PD 分类，并向其供电，进行功率管理，实时监控，检测与 PD 的连接是否断开等。PSE 分为内置（Endpoint）和外置（Midspan）两种：内置指的是 PSE 集成在路由器内部，外置指的是 PSE 与路由器相互独立。H3C 的 PSE 均采用内置方式，设备中携带了多块 PSE，一块具有 PoE 供电能力的接口板就是一个 PSE，在多 PSE 设备中，系统使用 PSE ID 来识别不同 PSE，使用 **display poe device** 命令可以查看 PSE ID 和接口板槽位号的对应关系。

(3) PI：具备 PoE 供电能力的以太网接口，也称为 PoE 接口。

(4) PD：接受 PSE 供电的设备，如 IP 电话、无线 AP（Access Point，接入点）、便携设备充电器、刷卡机、网络摄像头等。PD 设备在接受 PoE 电源供电的同时，可以连接其它电源，进行电源冗余备份。

图1-1 PoE 系统示意图



2. PoE的优点

- 可靠：电源集中供电，备份方便；
- 连接简捷：网络终端不需外接电源，只需要一根网线；
- 标准：符合 IEEE 802.3af 标准，使用全球统一的电源接口；

应用前景广泛：可以用于 IP 电话、无线 AP（Access Point，接入点）、便携设备充电器、POS 机、网络摄像头等。

1.1.2 协议规范

与 PoE 相关的协议规范为：IEEE 802.3af。

1.2 PoE配置任务简介

配置 PoE 接口有两种方式：

- 通过命令行配置；
- 通过配置 PoE Profile，并将 PoE Profile 应用到指定的 PoE 接口。

两种方式的作用一样，可以根据具体的需求选择：配置单独 PoE 接口时，一般采用命令行配置；而批量配置 PoE 接口时，一般采用 PoE Profile 配置。对于同一 PoE 接口下的同一 PoE 参数，只能选择一种方式进行配置（包括修改和删除）。

配置时，请注意：

- 在配置 PoE 功能前，请确保 PoE 电源和 PSE 已经处于正常工作状态，否则，可能无法进行 PoE 配置或者配置的 PoE 功能不能生效；
- 如果在设备启动过程中不慎关闭 PoE 电源，可能会导致 PoE Profile 中的 PoE 配置无法生效。

表1-1 PoE 配置任务简介

配置任务		说明	详细配置
开启PoE功能	开启PSE的PoE功能	必选	1.3.1
	开启PoE接口的PoE功能	必选	1.3.2
检测PD	开启非标准PD检测功能	可选	1.4.1
	配置PD断开的检测方式	可选	1.4.2
配置PoE功率	配置PSE最大功率	可选	1.5.1
	配置PoE接口的最大功率	可选	1.5.2
配置PoE功率管理	配置PSE功率管理	可选	1.6.1
	配置PoE接口功率管理	可选	1.6.2
PoE监控功能	监控PSE	可选	1.7.1
	监控PD	可选 设备给PD供电时会自动监控PD，不需要通过命令行来配置	1.7.2
通过PoE Profile配置	定义PoE Profile的内容	可选	1.8.1

配置任务		说明	详细配置
PoE接口	应用PoE Profile	可选	1.8.2
在线升级固件		可选	1.9

1.3 开启PoE功能

1.3.1 开启PSE的PoE功能

如果没有开启 PSE 的 PoE 功能，系统不会给 PSE 供电，也不会给 PSE 预留功率。

开启PSE的PoE功能时，如果该PSE的加入不会导致PoE功率过载，则允许开启。否则，由该PSE是否开启PoE功率管理功能决定（PSE功率管理的详细介绍请参见 1.6.1 配置PSE功率管理）：

- 如果该 PSE 没有开启 PoE 功率管理功能，则不允许开启该 PSE 的 PoE 功能；
- 如果该 PSE 已经开启了 PoE 功率管理功能，则允许开启该 PSE 的 PoE 功能（开启后 PSE 是否能够获得供电由其它因素决定，比如 PSE 的供电优先级）。

表1-2 开启 PSE 的 PoE 功能

操作	命令	说明
进入系统视图	**system-view**	-
开启PSE远程供电功能	**poe enable pse** *pse-id*	缺省情况下，PSE远程供电功能处于关闭状态

说明

PoE 功率过载：当 PSE 要求的最大功率之和大于 PoE 最大功率时，系统将认为 PoE 功率过载（PoE 最大功率由 PoE 电源的规格和用户的配置共同决定）。

1.3.2 开启PoE接口的PoE功能

如果没有开启 PoE 接口的 PoE 功能，系统不会给 PoE 接口下挂的 PD 供电，也不会给 PD 预留功率。

开启PoE接口的PoE功能时，如果该PoE接口的加入不会导致PSE功率过载，则允许开启。否则，由该PoE接口是否开启PoE功率管理功能决定（PoE接口功率管理的详细介绍请参见 1.6.2 配置PoE接口功率管理）：

- 如果该 PoE 接口没有开启 PoE 功率管理功能，则不允许开启该 PoE 接口的 PoE 功能；
- 如果该 PoE 接口已经开启了 PoE 功率管理功能，则允许开启该 PoE 接口的 PoE 功能（开启后 PD 是否能够获得供电由其它因素决定，比如 PoE 接口的供电优先级）。

说明

PSE 功率过载：当 PSE 上已经获得供电的端口的最大功率之和大于 PSE 最大功率时，系统将认为 PSE 功率过载（PSE 最大功率由用户的配置决定）。

表1-3 开启 PoE 接口的 PoE 功能

操作	命令	说明
进入系统视图	**system-view**	-
进入PoE接口视图	**interface** *interface-type interface-number*	-
开启PoE接口远程供电功能	**poe enable**	缺省情况下，PoE接口远程供电功能处于关闭状态
（可选）配置PoE接口连接PD的描述信息	**poe pd-description** *text*	缺省情况下，PoE接口连接PD的描述信息为空，即没有描述信息

1.4 检测PD

1.4.1 开启非标准PD检测功能

PD 设备分为标准 PD 和非标准 PD，标准 PD 是指符合 IEEE 802.3af 标准的 PD 设备。通常情况下，PSE 只能检测到标准 PD，并为其供电。只有在开启 PSE 检测非标准 PD 功能后，PSE 才能检测到非标准 PD，并为其供电。

表1-4 开启 PSE 检测非标准 PD 功能

操作	命令	说明
进入系统视图	**system-view**	-
开启PSE检测非标准PD功能	**poe legacy enable pse** *pse-id*	缺省情况下，PSE只能检测到标准PD，不能检测非标准PD

1.4.2 配置PD断开的检测方式

注意

在给 PD 供电过程中，若调整 PD 断开的检测方式，可能会导致连接的 PD 断电，请慎用。

为了检测 PD 是否与 PSE 断开，PoE 提供了两种检测方式：交流方式和直流方式。交流方式较直流方式省电。

表1-5 配置 PD 断开的检测方式

操作	命令	说明
进入系统视图	**system-view**	-
配置PD断开的检测方式	**poe disconnect { ac | dc }**	缺省情况下，PD断开检测的方式为交流检测方式

1.5 配置PoE功率

1.5.1 配置PSE最大功率

PSE 最大功率指的是所有与该 PSE 相连的 PD 设备能够获取到的最大供电总功率。

配置时，请注意：

- 为了防止 PoE 功率过载而引起 PSE 断电，配置时要求设备上所有 PSE 消耗的功率之和不能超过 PoE 最大功率。
- PSE 最大供电功率必须大于或等于该 PSE 上接入的所有优先级最高的 PoE 接口的最大功率之和，以保证对这些 PoE 接口的供电。

表1-6 配置 PSE 最大功率

操作	命令	说明
进入系统视图	**system-view**	-
配置PSE最大供电功率	**poe pse** *pse-id* **max-power** *max-power*	缺省情况下，PSE的最大功率为37W

1.5.2 配置PoE接口的最大功率

PoE 接口的最大功率指的是 PoE 接口能够提供给下挂 PD 的最大功率。当 PD 要求的功率大于 PoE 接口的最大功率时，则不会给 PD 供电。

表1-7 通过命令行配置 PoE 接口

操作	命令	说明
进入系统视图	**system-view**	-
进入PoE接口视图	**interface** *interface-type interface-number*	-
（可选）配置PoE接口最大供电功率	**poe max-power** *max-power*	缺省情况下，SIC-4FSWP、SIC 4GSWP、DSIC 9FSWP接口模块的最大供电功率为15.4W，HMIM 24GSW-POE接口模块的最大供电功率为16.6W

1.6 PoE功率管理

PoE 功率管理分为两个部分：PSE 功率管理和 PoE 接口功率管理。

1.6.1 配置PSE功率管理

PSE 功率管理用于 PoE 功率过载时，决定是否允许 PSE 开启 PoE 功能、是否给特定 PSE 供电以及能够分配多少功率。当 PoE 功率能够满足所有 PSE 的功率需求时，无需启用 PSE 功率管理。

对 PSE 进行供电时：

- 在没有开启 PSE 功率管理的情况下，如果 PoE 功率过载，则不对新接入的 PSE 供电；
- 在开启 PSE 功率管理优先级策略的情况下，如果 PoE 功率过载，分为两种情况：如果新接入的 PSE 优先级是最低的（为 Low），则不对新接入的 PSE 供电；如果接入新的 PSE 优先级高（为 High 或者 Critical），将对优先级低的 PSE 断电，保证给优先级高的 PSE 供电。

PSE 供电优先级顺序从高到低为：Critical、High 和 Low。

当配置 PSE 供电优先级为 Critical 时，如果 PoE 剩余保证功率（PoE 最大功率减去该 PoE 中优先级为 Critical 的 PSE 的最大功率，与该 PSE 是否开启 PoE 功能无关）小于该 PSE 的最大功率，配置失败（如果设置为其它优先级则不受该限制）；否则，配置成功，并将抢占部分低优先级 PSE 的功率，被抢占的 PSE 断电，但是这些 PSE 的配置不变。将 PSE 的优先级从 Critical 降为其它优先级，将使其它 PSE 有得到供电的机会。

说明

PoE 保证功率用于确保设备中的关键 PSE 能够一直得到电源供应，而不受 PSE 变化影响。

表1-8 配置 PSE 功率管理

操作	命令	说明
进入系统视图	**system-view**	-
配置PSE功率管理优先级策略	**poe pse-policy priority**	缺省情况下，没有配置PSE功率管理优先级策略
（可选）配置PSE供电优先级	**poe priority** { **critical** | **high** | **low** } **pse** *pse-id*	缺省情况下，PSE供电优先级为**low** 如果配置了相同的优先级，接口编号小的PoE接口的优先级高

1.6.2 配置PoE接口功率管理

PD 供电优先级取决于 PoE 接口的优先级。PoE 接口的优先级按照高低顺序为：Critical、High 和 Low。PD 能否得到供电要受 PoE 接口功率管理策略的控制。

所有 PSE 都执行相同的 PoE 接口功率管理策略。PSE 对接入的 PD 设备进行供电时：

- 在没有开启 PoE 接口功率管理的情况下，如果 PSE 功率过载，则不对新接入的 PD 供电；
- 在开启 PoE 接口功率管理优先级策略的情况下，如果 PSE 功率过载，接入新的 PD，将对优先级低的 PD 断电，保证优先级高的 PD 供电。

说明

- 设备给每个 PoE 接口预设了 19W 的保护功率，以便适应 PD 设备的功率波动，从而防止由于 PD 的瞬间功率过大而导致 PD 断开。当接口所在 PSE 剩余功率小于 19W，而且 PoE 接口没有配置优先级时，将不能给新增的 PD 供电；当接口所在 PSE 剩余功率小于 19W，但 PoE 接口配置了优先级时，高优先级端口将抢占低优先级端口的功率，优先保证高优先级端口的正常工作。
- 如果已接入的 PD 功率突然增加，造成 PSE 功率过载时，将停止对连接在低优先级 PoE 接口上的 PD 的供电，以便保证给优先级高的 PD 供电。

当 PSE 剩余保证功率（PSE 最大功率减去该 PSE 中优先级为 Critical 的 PoE 接口的最大功率，与 PoE 接口是否开启 PoE 功能无关）小于该 PoE 接口最大功率时，设置 Critical 优先级不成功；否则，该 PoE 接口优先级成功设为 Critical，并将抢占部分低优先级 PD 的功率，被抢占的 PD 断电，但是这些 PoE 接口的配置不变。将某个 PoE 接口的优先级从 Critical 降为其它优先级，可能导致其它 PoE 接口的 PD 得到供电。

表1-9 配置 PoE 接口功率管理

操作	命令	说明
进入系统视图	**system-view**	-
开启PoE接口功率管理优先级策略	**poe pd-policy priority**	缺省情况下，没有开启PoE接口功率管理优先级策略
进入PoE接口视图	**interface** *interface-type interface-number*	-
（可选）配置PoE接口供电优先级	**poe priority** { **critical** | **high** | **low** }	缺省情况下，PoE接口供电优先级为**low**

1.7 PoE监控功能

当开启 PoE 监控功能后，系统就会实时监控一些参数的值，当这些值超出限定范围时，系统会自动采取一些措施来进行自我保护。PoE 监控功能包括对 PSE、PD 的监控以及对设备温度的监控。

1.7.1 监控PSE

当 PSE 在当前功率利用率首次超过或低于设置的功率阈值时，系统将生成告警信息，发送给设备的 SNMP 模块，通过设置 SNMP 中告警信息的发送参数，来决定告警信息输出的相关属性。

有关告警信息的详细介绍，请参见“网络管理和监控配置指导”中的“SNMP”。

表1-10 配置 PSE 的功率告警阈值

操作	命令	说明
进入系统视图	**system-view**	-
配置PSE的功率告警阈值	**poe utilization-threshold** *utilization-threshold-value* **pse** *pse-id*	缺省情况下，PSE的功率告警阈值为80%

1.7.2 监控PD

当PSE开始或终止给PD供电时，系统将生成告警信息，发送给设备的SNMP模块，通过设置SNMP中告警信息的发送参数，来决定告警信息输出的相关属性。有关告警信息的详细介绍，请参见“网络管理和监控配置指导”中的“SNMP”。

1.8 通过PoE Profile配置PoE接口

PoE Profile是一组配置的集合，一个PoE Profile中可以配置多个PoE特性。在用大型网络中，将PoE Profile应用到多个PoE接口，则这些接口就具有相同的PoE特性；如果PD的接入接口变更，则把原接口上应用的PoE Profile应用到新的接入接口即可，不再需要重新逐条配置，从而方便了网管人员对PoE特性的配置。

设备支持创建多个PoE Profile，针对不同PD定义不同的PoE配置存放在不同的PoE Profile中，并在PD的接入接口应用相应的PoE Profile即可。

1.8.1 定义PoE Profile的内容

配置时，请注意：

- 如果PoE Profile已经应用，必须先取消PoE Profile的应用后，才能删除或修改该PoE Profile。
- 对于同一PoE接口下的同一PoE配置参数，只能选择一种方式进行配置（包括修改和删除），先配置的生效。如果对命令行已经配置过的参数再通过PoE Profile配置，则PoE Profile会应用失败；如果对PoE Profile已经配置过的参数再通过命令行配置，则命令行执行失败。必须先取消现有配置，新的配置才能成功。

表1-11 定义PoE Profile的内容

操作	命令	说明
进入系统视图	**system-view**	-
创建PoE Profile，并进入PoE Profile视图	**poe-profile** *profile-name* [*index*]	-
开启PoE接口远程供电功能	**poe enable**	缺省情况下，PoE接口远程供电功能处于关闭状态
（可选）配置PoE接口供电优先级	**poe priority** { **critical** | **high** | **low** }	缺省情况下，PoE接口供电优先级为**low**

1.8.2 应用PoE Profile

应用PoE Profile有两种方式：

- 方式一是在系统视图下应用PoE Profile。
- 方式二是在接口视图下应用PoE Profile。

两种方式配置效果一样，后配置的生效。如果需要将 PoE Profile 应用到多个 PoE 接口，使用方式一更为简便。

PoE Profile 可以应用于多个 PoE 接口，但一个 PoE 接口下只能应用一个 PoE Profile。

表1-12 应用 PoE Profile（方式一）

操作	命令	说明
进入系统视图	**system-view**	-
将PoE Profile应用到指定一个或多个PoE接口	**apply poe-profile** { **index** *index* | **name** *profile-name* } **interface** *interface-range*	-

表1-13 应用 PoE Profile（方式二）

操作	命令	说明
进入系统视图	**system-view**	-
进入PoE接口视图	**interface** *interface-type interface-number*	-
将PoE Profile应用到当前PoE接口	**apply poe-profile** { **index** *index* | **name** *profile-name* }	-

1.9 在线升级PSE固件

用户可以通过以下操作在线升级 PSE 固件，在线升级指的是不用重启 PSE 就能完成升级。升级有两种模式：

- refresh 模式，该模式是在 PSE 中原有处理软件的基础上对其进行升级更新。一般情况下使用 refresh 模式来升级 PSE 固件。
- full 模式，该模式是将 PSE 中原有处理软件彻底删除，再重新安装新的 PSE 固件。PSE 固件被损坏的情况下（表现为所有的 PoE 命令执行不成功），可用 full 模式进行升级，使软件恢复。

如果 PSE 固件的升级过程因意外而中断（例如发生错误导致设备重启），重启后用 full 方式升级失败时，请将设备断电重启后再用 full 方式升级即可成功。

升级后重启设备，新的 PSE 固件就会生效。

表1-14 在线升级 PSE 固件

操作	命令	说明
进入系统视图	**system-view**	-
升级PSE固件	**poe update** { **full** | **refresh** } *filename* [**pse** *pse-id*]	-

1.10 PoE显示和维护

在完成上述配置后，在任意视图下执行 **display** 命令可以显示配置后 PoE 的运行情况，通过查看显示信息验证配置的效果。

表1-15 PoE 显示和维护

操作	命令
显示所有PSE的相关信息	**display poe device**
显示设备指定PoE接口的供电状态	**display poe interface** [*interface-type interface-number*]
显示PoE接口的功率信息	**display poe interface power** [*interface-type interface-number*]
显示PoE电源的功率和各PSE的功率信息	**display poe power-usage**
显示PSE信息	**display poe pse** [*pse-id*]
显示指定PSE上PoE接口的供电状态	**display poe pse** *pse-id* **interface**
显示PSE连接的PoE接口的功率信息	**display poe pse** *pse-id* **interface power**
显示PoE电源的信息	**display poe-power**
显示PoE Profile的配置和应用的所有信息	**display poe-profile** [**index** *index* | **name** *profile-name*]
显示指定PoE接口当前生效的PoE Profile配置项和应用的所有信息	**display poe-profile interface** *interface-type interface-number*

1.11 PoE典型配置举例

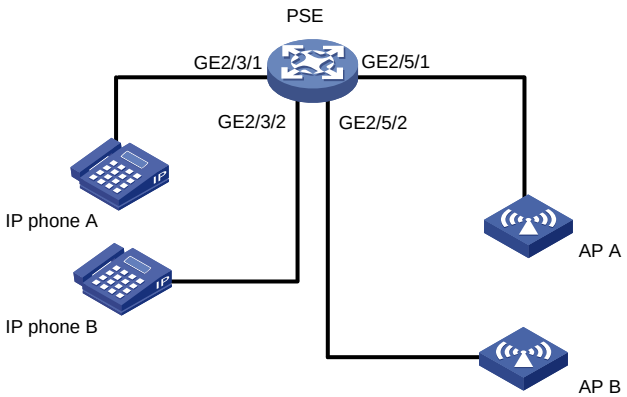
1. 组网需求

设备通过 PoE 接口为 PD 设备供电。

- 设备有两块支持远程供电的 PSE，分别插在第 3 和第 5 号槽位，PSE ID 为 10 和 16。
- 给 PSE 10 分配 400 瓦功率，假设 PSE 16 的 PSE 最大功率的缺省值可以满足需求。
- GigabitEthernet2/3/1 与 GigabitEthernet2/3/2 接入 IP 电话。
- GigabitEthernet2/5/1 与 GigabitEthernet2/5/2 接入 AP 设备。
- GigabitEthernet2/3/2 的供电优先级为 critical，要求在新接入 PD 导致 PSE 功率过载时，停止对新接 PD 入供电（即采用 PoE 接口功率管理的缺省策略）。

2. 组网图

图1-2 PoE 组网图



3. 配置步骤

开启 PSE 的远程供电功能。

```
<Sysname> system-view
[Sysname] poe enable pse 10
[Sysname] poe enable pse 16
```

配置 PSE 10 最大功率为 400 瓦。

```
[Sysname] poe max-power 400 pse 10
```

开启 PoE 接口 GigabitEthernet2/3/1 和 GigabitEthernet2/5/1 的远程供电功能。

```
[Sysname] interface gigabitethernet 2/3/1
[Sysname-GigabitEthernet2/3/1] poe enable
[Sysname-GigabitEthernet2/3/1] quit
[Sysname] interface gigabitethernet 2/5/1
[Sysname-GigabitEthernet2/5/1] poe enable
[Sysname-GigabitEthernet2/5/1] quit
```

开启 GigabitEthernet2/3/2 的远程供电功能，并配置接口对外供电的优先级为 critical。

```
[Sysname] interface gigabitethernet 2/3/2
[Sysname-GigabitEthernet2/3/2] poe enable
[Sysname-GigabitEthernet2/3/2] poe priority critical
[Sysname-GigabitEthernet2/3/2] quit
```

开启 GigabitEthernet2/5/2 的远程供电功能。

```
[Sysname] interface gigabitethernet 2/5/2
[Sysname-GigabitEthernet2/5/2] poe enable
```

4. 结果验证

配置完成后，IP 电话和 AP 设备被供电，能够正常工作。

1.12 常见配置错误举例

1.12.1 配置PoE接口优先级为critical不成功

1. 原因分析

- PoE 接口所在 PSE 剩余保证功率小于 PoE 接口的最大供电功率。
- PoE 接口的优先级已经通过其他方式进行配置。

2. 解决方法

- 对于第一种情况可以通过增大 PSE 的最大供电功率来解决，或者在 PSE 剩余保证功率不可调整时减小 PoE 接口的最大供电功率。
- 对于第二种情况需先取消其他方式的配置。

1.12.2 应用PoE Profile到PoE接口不成功

1. 原因分析

- 该 PoE Profile 的某些配置项已经通过其他方式进行配置。
- 该 PoE Profile 的某些配置项不符合 PoE 接口的配置要求。
- 已经存在 PoE Profile 在该 PoE 接口的应用。

2. 解决方法

- 对于第一种情况，可以通过取消其他方式的配置来解决。
- 对于第二种情况，需修改该 PoE Profile 的某些配置项。
- 对于第三种情况，先取消其他 PoE Profile 在该 PoE 接口的应用。

目 录

1 SNMP

说明

设备运行于FIPS模式时，本特性部分配置相对于非FIPS模式有所变化，具体差异请见本文相关描述。有关 FIPS 模式的详细介绍请参见“安全配置指导”中的“FIPS”。

1.1 SNMP简介

SNMP（Simple Network Management Protocol，简单网络管理协议）是互联网中的一种网络管理标准协议，广泛用于实现管理设备对被管理设备的访问和管理。SNMP 具有以下优势：

- 支持网络设备的智能化管理。利用基于 SNMP 的网络管理平台，网络管理员可以查询网络设备的运行状态和参数，设置参数值，发现故障，完成故障诊断，进行容量规划和制作报告。
- 支持对不同物理特性的设备进行管理。SNMP 只提供最基本的功能集，使得管理任务与被管理设备的物理特性和联网技术相对独立，从而实现对不同厂商设备的管理。

1.1.1 SNMP的网络架构

SNMP 网络架构由三部分组成：NMS、Agent 和 MIB。

- NMS（Network Management System，网络管理系统）是 SNMP 网络的管理者，能够提供友好的人机交互界面，方便网络管理员完成大多数的网络管理工作。
- Agent 是 SNMP 网络的被管理者，负责接收、处理来自 NMS 的 SNMP 报文。在某些情况下，如接口状态发生改变时，Agent 也会主动向 NMS 发送告警信息。
- MIB（Management Information Base，管理信息库）是被管理对象的集合。NMS 管理设备的时候，通常会关注设备的一些参数，比如接口状态、CPU 利用率等，这些参数就是被管理对象，在 MIB 中称为节点。每个 Agent 都有自己的 MIB。MIB 定义了节点之间的层次关系以及对象的一系列属性，比如对象的名字、访问权限和数据类型等。被管理设备都有自己的 MIB 文件，在 NMS 上编译这些 MIB 文件，就能生成该设备的 MIB。NMS 根据访问权限对 MIB 节点进行读/写操作，从而实现对 Agent 的管理。

NMS、Agent和MIB之间的关系如 图 1-1 所示。

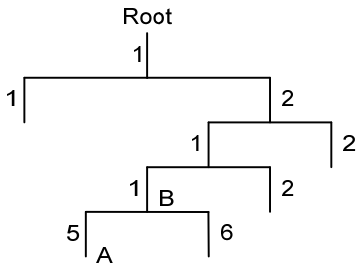
图1-1 NMS、Agent 和 MIB 关系图

1.1.2 MIB和MIB视图

MIB以树状结构进行存储。树的每个节点都是一个被管理对象，它用从根开始的一条路径唯一地识别（OID）。如 图 1-2 所示，被管理对象B可以用一串数字{1.2.1.1}唯一确定，这串数字是被管理对象的OID（Object Identifier，对象标识符）。

MIB 视图是 MIB 的子集合，将团体名/用户名与 MIB 视图绑定，可以限制 NMS 能够访问的 MIB 对象。当用户配置 MIB 视图包含（include）某个 MIB 子树时，NMS 可以访问该子树的所有节点；当用户配置 MIB 视图不包含（exclude）某个 MIB 子树时，NMS 不能访问该子树的所有节点。

图1-2 MIB 树结构



1.1.3 SNMP基本操作

SNMP 提供四种基本操作：

- Get 操作：NMS 使用该操作查询 Agent MIB 中节点的值。
- Set 操作：NMS 使用该操作设置 Agent MIB 中节点的值。
- Trap 操作：Agent 使用该操作向 NMS 发送 Trap 报文。Agent 不要求 NMS 发送回应报文，NMS 也不会对 Trap 报文进行回应。SNMPv1、SNMPv2c 和 SNMPv3 均支持 Trap 操作。
- Inform 操作：Agent 使用该操作向 NMS 发送 Inform 报文。Agent 要求 NMS 发送回应报文，因此，Inform 报文比 Trap 报文更可靠。如果 Agent 在一定时间内没有收到 NMS 的回应报文，则会启动重发机制。只有 SNMPv2c 和 SNMPv3 支持 Inform 操作。

1.1.4 SNMP版本介绍

目前，设备运行于非 FIPS 模式时，支持 SNMPv1、SNMPv2c 和 SNMPv3 三种版本；设备运行于 FIPS 模式时，只支持 SNMPv3 版本。只有 NMS 和 Agent 使用的 SNMP 版本相同，NMS 才能和 Agent 建立连接。

- SNMPv1 采用团体名（Community Name）认证机制。团体名类似于密码，用来限制 NMS 和 Agent 之间的通信。如果 NMS 设置的团体名和被管理设备上设置的团体名不同，则 NMS 和 Agent 不能建立 SNMP 连接，从而导致 NMS 无法访问 Agent，Agent 发送的告警信息也会被 NMS 丢弃。
- SNMPv2c 也采用团体名认证机制。SNMPv2c 对 SNMPv1 的功能进行了扩展：提供了更多的操作类型；支持更多的数据类型；提供了更丰富的错误代码，能够更细致地区分错误。
- SNMPv3 采用 USM（User-Based Security Model，基于用户的安全模型）认证机制。网络管理员可以设置认证和加密功能。认证用于验证报文发送方的合法性，避免非法用户的访问；

加密则是对 NMS 和 Agent 之间的传输报文进行加密，以免被窃听。采用认证和加密功能可以为 NMS 和 Agent 之间的通信提供更高的安全性。

1.2 配置SNMP基本参数

由于SNMPv3 版本的配置和SNMPv1 版本、SNMPv2c版本的配置有较大区别，所以下面分两种情况介绍SNMP基本功能的配置，详见 表 1-1 和 表 1-2。

1.2.1 配置SNMPv1/v2c版本基本参数

说明

设备运行于 FIPS 模式时，不支持 SNMPv1/v2c 版本。

表1-1 配置 SNMPv1/v2c 版本基本参数

操作	命令	说明
进入系统视图	**system-view**	-
（可选）启动SNMP Agent服务	**snmp-agent**	缺省情况下，SNMP Agent服务处于关闭状态 执行除**snmp-agent calculate-password**外任何以**snmp-agent**开头的命令，都可以启动SNMP Agent服务
（可选）配置设备的维护联系信息	**snmp-agent sys-info contact** *sys-contact*	缺省情况下，系统维护联系信息为“Hangzhou H3C Tech. Co., Ltd.”
（可选）配置设备的物理位置信息	**snmp-agent sys-info location** *sys-location*	缺省情况下，设备的物理位置信息为“Hangzhou, China”
启用SNMPv1/v2c版本	**snmp-agent sys-info version** { **all** | { **v1** | **v2c** } *}	缺省情况下，系统启用的SNMP版本号为“SNMPv3”
（可选）设置本地SNMP实体的引擎ID	**snmp-agent local-engineid** *engineid*	缺省情况下，设备引擎ID为公司的“企业号＋设备信息”
（可选）创建MIB视图或更新MIB视图内容	**snmp-agent mib-view** { **excluded** | **included** } *view-name oid-tree* [**mask** *mask-value*]	缺省情况下，设备上已创建了四个视图，视图名均为ViewDefault： • 视图一包含 MIB 子树 iso • 视图二不包含子树 snmpUsmMIB • 视图三不包含子树 snmpVacmMIB • 视图四不包含子树 snmpModules.18 MIB视图是MIB的子集，由视图名和MIB子树来唯一确定一个MIB视图。视图名相同但包含的子树不同，则认为是不同的视图。除缺省视图外，用户最多可以创建16个MIB视图

操作			命令	说明
设置访问权限	直接设置	创建SNMP团体	**snmp-agent community** { **read** | **write** } [**simple** | **cipher**] *community-name* [**mib-view** *view-name*] [**acl** *acl-number* | **acl ipv6** ipv6-*acl-number*] *	二者选其一 直接设置是以SNMPv1和SNMPv2c版本的团体名进行设置 间接设置是先创建SNMP组，再向创建的组中添加的用户，用户相当于SNMPv1和SNMPv2c版本的团体名，在NMS上配置的团体名需要跟Agent上配置的用户名一致 缺省情况下，不存在任何SNMP组和SNMP团体
	间接设置	创建SNMPv1/v2c组	**snmp-agent group** { **v1** | **v2c** } *group-name* [**read-view** *view-name*] [**write-view** *view-name*] [**notify-view** *view-name*] [**acl** *acl-number* | **acl ipv6** *ipv6-acl-number*] *	
		创建SNMPv1/v2c用户	**snmp-agent usm-user** { **v1** | **v2c** } *user-name group-name* [**acl** *acl-number* | **acl ipv6** *ipv6-acl-number*] *	
（可选）创建SNMP上下文			**snmp-agent context** *context-name*	缺省情况下，设备上没有配置SNMP上下文
（可选）创建一个团体名到SNMP上下文的映射			**snmp-agent community-map** *community-name* **context** *context-name*	缺省情况下，设备上没有团体名到SNMP上下文的映射
（可选）设置Agent能处理的SNMP报文的最大长度			**snmp-agent packet max-size** *byte-count*	缺省情况下，Agent能接收/发送的SNMP消息包长度的最大值为1500字节
（可选）设置Agent接收SNMP报文的端口号			**snmp-agent port** *port-num*	缺省情况下，使用161端口接收SNMP报文

1.2.2 配置SNMPv3 版本基本参数

1. 配置限制和指导

建立 SNMPv3 连接时，是否进行认证和加密，受 **snmp-agent group v3** 和 **snmp-agent usm-user v3** 两条命令的影响：

- 创建组时，如果不指定 **authentication** 和 **privacy** 参数，则表示不认证不加密。此时，使用和该组绑定的用户名建立 SNMP 连接时，均不认证不加密。即便用户配置了认证密码/加密密码，认证密码/加密密码也不生效。
- 创建组时，如果指定 **authentication** 参数，则表示认证不加密。此时，使用和该组绑定的用户名建立 SNMP 连接时，均认证不加密。即便用户配置了加密密码，加密密码也不生效。该组内的用户必须配置认证密码，否则，不能建立 SNMP 连接。
- 创建组时，如果指定 **privacy** 参数，则表示认证加密。此时，使用和该组绑定的用户名建立 SNMP 连接时，均认证加密。该组内的用户必须配置认证密码和加密密码，否则，不能建立 SNMP 连接。

2. 配置步骤

表1-2 配置 SNMPv3 版本基本参数

操作	命令	说明
进入系统视图	**system-view**	-
（可选）启动SNMP Agent服务	**snmp-agent**	缺省情况下，SNMP Agent服务处于关闭状态 执行除**snmp-agent calculate-password**外任何以**snmp-agent**开头的命令，都可以启动SNMP Agent服务
（可选）配置设备的维护联系信息	**snmp-agent sys-info contact** *sys-contact*	缺省情况下，系统维护联系信息为“Hangzhou H3C Tech. Co., Ltd.”
（可选）配置设备的物理位置信息	**snmp-agent sys-info location** *sys-location*	缺省情况下，设备的物理位置信息为“Hangzhou, China”
启用SNMPv3版本	**snmp-agent sys-info version v3**	缺省情况下，系统启用的SNMP版本号为“SNMPv3”
（可选）设置本地SNMP实体的引擎ID	**snmp-agent local-engineid** *engineid*	缺省情况下，设备引擎ID为公司的“企业号＋设备信息” SNMPv3版本的用户名、密文密码等都和引擎ID相关联，如果更改了引擎ID，则原引擎ID下配置的用户名、密码失效。
（可选）设置远端SNMP实体的引擎ID	**snmp-agent remote** { *ip-address* | **ipv6** *ipv6-address* } [**vpn-instance** *vpn-instance-name*] **engineid** *engineid*	缺省情况下，设备上没有配置远端SNMP实体的引擎ID 当设备需要向目的主机（能够解析Trap和Inform报文的设备，通常为NMS）发送SNMPv3 Inform报文时，该步骤必选
（可选）创建MIB视图或更新MIB视图内容	**snmp-agent mib-view** { **excluded** | **included** } *view-name oid-tree* [**mask** *mask-value*]	缺省情况下，设备上已创建了四个视图，视图名均为ViewDefault： • 视图一包含 MIB 子树 iso • 视图二不包含子树 snmpUsmMIB • 视图三不包含子树 snmpVacmMIB • 视图四不包含子树 snmpModules.18 MIB视图是MIB的子集，由视图名和MIB子树来唯一确定一个MIB视图。视图名相同但包含的子树不同，则认为是不同的视图。除缺省视图外，用户最多可以创建16个MIB视图

操作	命令	说明
创建SNMPv3组	非FIPS模式下： **snmp-agent group v3** *group-name* [**authentication** | **privacy**] [**read-view** *view-name*] [**write-view** *view-name*] [**notify-view** *view-name*] [**acl** *acl-number* | **acl ipv6** *ipv6-acl-number*] * FIPS模式下： **snmp-agent group v3** *group-name* { **authentication** | **privacy** } [**read-view** *view-name*] [**write-view** *view-name*] [**notify-view** *view-name*] [**acl** *acl-number* | **acl ipv6** *ipv6-acl-number*] *	缺省情况下，设备上没有配置SNMP组
（可选）计算用户给定明文密码通过加密算法处理后的密文密码	非FIPS模式下： **snmp-agent calculate-password** *plain-password* **mode** { **md5** | **sha** } { **local-engineid** | **specified-engineid** *engineid* } FIPS模式下： **snmp-agent calculate-password** *plain-password* **mode sha** { **local-engineid** | **specified-engineid** *engineid* }	-
创建SNMPv3用户	非FIPS模式下： **snmp-agent usm-user v3** *user-name group-name* [**remote** { *ip-address* | **ipv6** *ipv6-address* } [**vpn-instance** *vpn-instance-name*]] [{ **cipher** | **simple** } **authentication-mode** { **md5** | **sha** } *auth-password* [**privacy-mode** { **aes128** | **des56** } *priv-password*]] [**acl** *acl-number* | **acl ipv6** *ipv6-acl-number*] * FIPS模式下： **snmp-agent usm-user v3** *user-name group-name* [**remote** { *ip-address* | **ipv6** *ipv6-address* } [**vpn-instance** *vpn-instance-name*]] { **cipher** | **simple** } **authentication-mode sha** *auth-password* [**privacy-mode aes128** *priv-password*] [**acl** *acl-number* | **acl ipv6** *ipv6-acl-number*] *	当设备需要向目的主机发送SNMPv3 Inform报文时，**remote** *ip-address*参数必选 如果使用**cipher**参数，则后面的*auth-password*和*priv-password*都必须输入并被视为密文密码
（可选）创建SNMP上下文	**snmp-agent context** *context-name*	缺省情况下，设备上没有配置SNMP上下文
（可选）设置Agent能处理的SNMP报文的最大长度	**snmp-agent packet max-size** *byte-count*	缺省情况下，Agent能接收/发送的SNMP消息包长度的最大值为1500字节
（可选）设置Agent接收SNMP报文的端口号	**snmp-agent port** *port-num*	缺省情况下，使用161端口接收SNMP报文

1.3 配置SNMP日志

SNMP 日志可以记录 NMS 对 Agent 的 Get 请求、Set 请求和 Set 响应信息，不能记录 Get 响应信息。同时 SNMP 日志可以记录 Agent 对 NMS 的 Trap 和 Inform 操作信息。

- 当进行 Get 操作时，Agent 会记录 NMS 用户的 IP 地址、Get 操作的节点名和节点 OID。
- 当进行 Set 操作时，Agent 会记录 NMS 用户的 IP 地址、Set 操作的节点名、节点 OID、节点值以及 Set 操作返回的错误码和错误索引。
- 当进行 Trap 和 Inform 操作时，Agent 会向 NMS 发送告警，Agent 会记录告警相关的信息。

这些日志将被发送到设备的信息中心，级别为 informational。通过设置信息中心的参数，最终决定 SNMP 日志的输出规则(即是否允许输出以及输出方向)。SNMP 每条日志信息中记录的 node 域(信息内容对应的 MIB 节点名）和 value 域（信息内容对应的 MIB 节点值）的长度之和不能超过 1024 字节，超出的部分将不会被输出。有关信息中心的详细介绍请参见“网络管理和监控配置指导”中的“信息中心”。

表1-3 配置 SNMP 日志功能

操作	命令	说明
进入系统视图	**system-view**	-
（可选）打开SNMP日志开关	**snmp-agent log { all | get-operation | set-operation }**	大量的日志记录会占用设备的存储空间，影响设备的性能。正常情况下，建议关闭SNMP日志功能。 缺省情况下，SNMP日志开关处于关闭状态
（可选）打开SNMP告警日志开关	**snmp-agent trap log**	缺省情况下，SNMP告警日志功能处于关闭状态

1.4 配置SNMP告警

SNMP 告警信息包括 Trap 和 Inform 两种，用来告知 NMS 设备上发生了重要事件，比如，用户的登录/退出，接口状态变成 up/down 等。如无特殊说明，本文中的告警信息均指 Trap 和 Inform 两种信息。

1.4.1 开启告警功能

因为告警信息通常较多，会占用设备内存，影响设备性能，所以建议用户根据需要开启指定模块的告警功能，生成相应的告警信息。

如果要求接口在链路状态发生改变时生成相应的告警信息，需要在全局和接口下均开启接口链路状态变化的告警功能。如果要生成其它模块的告警信息，除了使用 **snmp-agent trap enable** 命令开启告警功能外，还可能需要执行各个模块的相关配置，详情请参见各模块的相关描述。

表1-4 开启告警功能

操作	命令	说明
进入系统视图	**system-view**	-

操作	命令	说明
在全局下开启告警功能	**snmp-agent trap enable** [**configuration** | *protocol* | **standard** [**authentication** | **coldstart** | **linkdown** | **linkup** | **warmstart**] * | **system**]	缺省情况下，SNMP配置告警、标准告警和系统告警功能处于开启状态，其他各模块告警功能是否开启请参见各模块手册
进入接口视图	**interface** *interface-type interface-number*	-
开启接口链路状态变化的告警功能	**enable snmp trap updown**	缺省情况下，接口状态变化的告警功能处于开启状态

1.4.2 配置告警信息发送参数

如果配置了 **snmp-agent target-host inform** 命令，则设备会向指定目的主机（能够解析 Trap 和 Inform 报文的设备，通常为 NMS）发送 Inform 报文；如果配置了 **snmp-agent target-host trap** 命令，则设备会向指定目的主机发送 Trap 报文。

设备第一次发送告警信息时，会检查设备和目的主机是否路由可达。如果可达，则直接发送。如果不可达，则先将告警信息缓存在消息队列里，等路由可达后，再发送。为防止告警信息累积占用太多内存，用户可以设置该队列的长度以及告警信息在队列里的保存时间。

- 如果在告警信息的发送队列满时系统又收到了新的告警信息，则系统会自动删除最先收到的告警信息来保存新的告警信息。
- 如果告警信息的发送队列中的某信息到达了已设定的保存时间，则系统会自动删除该告警信息。

1. 配置准备

如果要将告警信息发送给 NMS，则需要进行以下配置准备：

(1) 配置 SNMP 基本参数。

(2) 确保设备与 NMS 路由可达。

2. 配置步骤

表1-5 配置告警信息发送参数

操作	命令	说明
进入系统视图	**system-view**	-
设置Inform报文的发送参数	非FIPS模式下： **snmp-agent target-host inform address udp-domain** { *ip-address* | **ipv6** *ipv6-address* } [**udp-port** *port-number*] [**vpn-instance** *vpn-instance-name*] **params securityname** *security-string* { **v2c** | **v3** [**authentication** | **privacy**] } FIPS模式下： **snmp-agent target-host inform address udp-domain** { *ip-address* | **ipv6** *ipv6-address* } [**udp-port** *port-number*] [**vpn-instance** *vpn-instance-name*] **params securityname** *security-string* **v3** { **authentication** | **privacy** }	缺省情况下，设备上没有设置Inform报文的目的主机

操作	命令	说明
设置Trap报文的发送参数	非FIPS模式下： **snmp-agent target-host trap address udp-domain** { *ip-address* | **ipv6** *ipv6-address* } [**udp-port** *port-number*] [**vpn-instance** *vpn-instance-name*] **params securityname** *security-string* [**v1** | **v2c** | **v3** [**authentication** | **privacy**]] FIPS模式下： **snmp-agent target-host trap address udp-domain** { *ip-address* | **ipv6** *ipv6-address* } [**udp-port** *port-number*] [**vpn-instance** *vpn-instance-name*] **params securityname** *security-string* **v3** { **authentication** | **privacy** }	缺省情况下，设备上没有设置Trap报文的目的主机
（可选）设置发送告警信息的源地址	**snmp-agent** { **inform** | **trap** } **source** *interface-type* { *interface-number* | *interface-number.subnumber* }	缺省情况下，由SNMP选择路由出接口的IP地址作为告警信息的源IP地址
（可选）对标准linkUp/linkDown告警信息进行私有扩展	**snmp-agent trap if-mib link extended**	缺省情况下，系统发送的linkUp/linkDown告警信息的格式为标准格式，不对其进行私有扩展
（可选）设置告警信息发送队列的长度	**snmp-agent trap queue-size** *size*	缺省情况下，告警信息的消息队列最多可以存储100条告警信息
（可选）设置告警信息的保存时间	**snmp-agent trap life** *seconds*	缺省情况下，告警信息的保存时间为120秒

说明

对 linkUp/linkDown 告警信息进行私有扩展后，设备生成和发送的该信息由标准 linkUp/linkDown 告警信息后增加接口描述和接口类型信息构成。如果 NMS 不支持该扩展信息，请禁用私有扩展功能。

1.5 SNMP显示和维护

在完成上述配置后，在任意视图下执行 **display** 命令，均可以显示配置后 SNMP 的运行情况，通过查看显示信息，来验证配置的效果。

表1-6 SNMP 显示和维护

操作	命令
显示系统维护联络信息、系统位置信息及SNMP版本信息	**display snmp-agent sys-info** [**contact** | **location** | **version**] *
显示SNMP报文统计信息	**display snmp-agent statistics**
显示设备的SNMP实体引擎ID	**display snmp-agent local-engineid**
显示SNMP组信息	**display snmp-agent group** [*group-name*]

操作	命令
显示远端SNMP实体引擎信息	**display snmp-agent remote** [*ip-address* [**vpn-instance** *vpn-instance-name*] | **ipv6** *ipv6-address* [**vpn-instance** *vpn-instance-name*]]
显示告警信息队列的基本信息	**display snmp-agent trap queue**
显示系统当前可以发送告警信息的模块及其告警信息的使能状态	**display snmp-agent trap-list**
显示SNMPv3用户信息	**display snmp-agent usm-user** [**engineid** *engineid* | **username** *user-name* | **group** *group-name*] *
显示SNMPv1或SNMPv2c团体信息（FIPS模式下不支持该命令）	**display snmp-agent community** [**read** | **write**]
显示MIB视图的信息	**display snmp-agent mib-view** [**exclude** | **include** | **viewname** *view-name*]
显示当前SNMP支持的MIB节点信息	**display snmp-agent mib-node** [**details** | **index-node** | **trap-node** | **verbose**]
显示指定的SNMP上下文	**display snmp-agent context** [*context-name*]

1.6 SNMPv1/v2c典型配置举例

说明

设备运行于 FIPS 模式时，不支持本例。

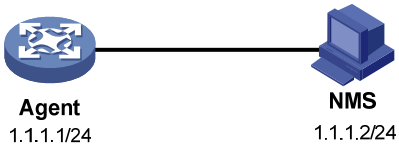
SNMPv1 和 SNMPv2c 的配置方法相同，下面以 SNMPv1 为例进行配置。

1. 组网需求

- NMS与Agent相连，设备的IP地址和掩码如 图 1-3 所示。
- NMS 通过 SNMPv1 对 Agent 进行监控管理，Agent 在故障时能够主动向 NMS 发送告警信息。

2. 组网图

图1-3 SNMPv1 配置组网图



3. 配置步骤

(1) 配置 Agent

配置 Agent 的 IP 地址为 1.1.1.1/24，并确保 Agent 与 NMS 之间路由可达。（配置步骤略）

设置 Agent 使用的 SNMP 版本为 v1、只读团体名为 public，读写团体名为 private。

```
<Agent> system-view
```

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

设置设备的联系人和位置信息，以方便维护。

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

设置允许向 NMS 发送告警信息，使用的团体名为 public。

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

snmp-agent target-host 命令中指定的版本必须和 NMS 上运行的 SNMP 版本一致，因此需要将 **snmp-agent target-host** 命令中的版本参数设置为 v1。否则，NMS 无法正确接收告警信息。

(2) 配置 NMS

设置 NMS 使用的 SNMP 版本为 SNMPv1，只读团体名为 public，读写团体名为 private。另外，还可以根据需求设置“超时”时间和“重试次数”。具体配置请参考 NMS 的相关手册。

说明

NMS 侧的配置必须和 Agent 侧保持一致，否则无法通信。

(3) 结果验证

通过查询 Agent 上相应的 MIB 节点获取 NULL0 接口的 MTU 值，结果为 1500：

```
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
```

当使用错误的团体名获取 Agent 上的 MIB 节点信息时，NMS 上将看到认证失败的 Trap 信息，即 authenticationFailure：

```
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
```

1.7 SNMPv3典型配置举例

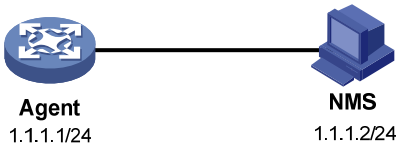
1. 组网需求

- NMS与Agent相连，设备的IP地址和掩码如 图 1-4 所示。

- NMS 通过 SNMPv3 只能对 Agent 的 SNMP 报文的相关信息进行监控管理，Agent 在出现故障时能够主动向 NMS 发送告警信息。
- NMS 与 Agent 建立 SNMP 连接时，需要认证，使用的认证算法为 SHA-1，认证密码为 123456TESTauth&!。NMS 与 Agent 之间传输的 SNMP 报文需要加密，使用的加密协议为 AES，加密密码为 123456TESTencr&!。

2. 组网图

图1-4 SNMPv3 配置组网图



3. 配置步骤

(1) 配置 Agent

配置 Agent 的 IP 地址为 1.1.1.1/24，并确保 Agent 与 NMS 之间路由可达。（配置步骤略）

设置访问权限：用户只能读写节点 snmp（OID 为 1.3.6.1.2.1.11）下的对象，不可以访问其它 MIB 对象。

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

设置 Agent 使用的用户名为 managev3user，认证算法为 SHA-1，认证密码为 123456TESTauth&!，加密算法为 AES，加密密码是 123456TESTencr&!。

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

设置设备的联系人和位置信息，以方便维护。

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

设置允许向 NMS 发送告警信息，使用的用户名为 managev3user。

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

(2) 配置 NMS

设置 NMS 使用的 SNMP 版本为 SNMPv3，用户名为 managev3user，启用认证和加密功能，认证算法为 SHA-1，认证密码为 123456TESTauth&!，加密协议为 AES，加密密码为 123456TESTencr&!。另外，还可以根据需求设置“超时”时间和“重试次数”。具体配置请参考 NMS 的相关手册。

说明

NMS 侧的配置必须和设备侧保持一致，否则无法进行相应操作。

---

(3) 结果验证

通过查询 Agent 上相应的 MIB 节点获取 NULL0 接口的 MTU 值，结果为 1500：

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

通过查询 Agent 上相应的 MIB 节点获取设备名称时，由于没有权限，结果为空（NULL）：

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

对设备上某个空闲的接口执行 **shutdown** 或 **undo shutdown** 操作，NMS 上将看到相应的 Trap 信息：

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

目 录

1 RMON

1.1 RMON简介

1.1.1 RMON概述

RMON（Remote Network Monitoring，远程网络监视）主要实现了统计和告警功能，用于网络中管理设备对被管理设备的远程监控和管理。

统计功能指的是被管理设备可以按周期或者持续跟踪统计其端口所连接的网段上的各种流量信息，比如某段时间内某网段上收到的报文总数，或收到的超长报文的总数等。

告警功能指的是被管理设备能监控指定 MIB 变量的值，当该值达到告警阈值时（比如端口速率达到指定值，或者广播报文的比例达到指定值），能自动记录日志、生成告警信息发送给 SNMP 模块，由 SNMP 模块发送给管理设备。有关告警信息的详细介绍，请参见“网络管理和监控配置指导”中的“SNMP”。

RMON 和 SNMP 都用于远程网络管理：

- SNMP 是 RMON 实现的基础。RMON 使用 SNMP 告警信息发送机制向管理设备发送告警信息告知告警变量的异常。虽然 SNMP 也定义了告警功能，但通常用于告知被管理设备上某功能是否运行正常、接口物理状态的变化等，两者监控的对象、触发条件以及报告的内容均不同。
- RMON 是 SNMP 功能的增强。RMON 能更有效、更积极主动地监测远程网络设备，为监控子网的运行提供了一种高效的手段。RMON 协议规定达到告警阈值时被管理设备能自动生成告警信息发送给设备的 SNMP 模块，所以管理设备不需要多次去获取 MIB 变量的值，进行比较，从而能够减少管理设备同被管理设备的通讯流量，达到简便而有力地管理大型互连网络的目的。

1.1.2 RMON的工作机制

RMON 允许有多个监控者，监控者可用两种方法收集数据：

- 第一种方法利用专用的 RMON probe（探测仪）收集数据，管理设备直接从 RMON probe 获取管理信息并控制网络资源。这种方式可以获取 RMON MIB 的全部信息；
- 第二种方法是将 RMON Agent 直接植入网络设备（路由器、交换机、HUB 等），使它们成为带 RMON probe 功能的网络设施。管理设备使用 SNMP 的基本操作与 RMON Agent 交换数据信息，收集网络管理信息，但这种方法受设备资源限制，不能获取 RMON MIB 的所有数据，只收集事件组、告警组、历史组和统计组四个组的信息。

H3C 采用第二种方法，在设备上实现了 RMON Agent 功能。通过该功能，管理设备可以获得与被管网络设备端口相连的网段上的整体流量、错误统计和性能统计等信息，进而实现对网络的管理。

1.1.3 RMON组

RMON 协议中定义了多个 RMON 组，设备实现了公有 MIB 中支持的统计组、历史组、事件组、告警组、代理配置组和用户历史组。此外，H3C 还自定义和实现了扩展告警组，以增强告警组的功能。

其中，代理配置组和用户历史组只支持 MIB 操作，具体内容请参考《Comware V7 Platform MIB Companion》中 RMON 章节。

1. 统计组

统计组规定系统将持续地对端口的各种流量信息进行统计（目前只支持对以太网端口的统计），并将统计结果存储在以太网统计表（etherStatsTable）中以便管理设备随时查看。在指定接口下创建统计表项成功后，统计组就对当前接口的报文数进行统计，它统计的结果是一个连续的累加值。

统计信息包括网络冲突数、CRC 校验错误报文数、过小（或超大）的数据报文数、广播、多播的报文数以及接收字节数、接收报文数等。

2. 历史组

历史组规定系统将按指定周期对端口的各种流量信息进行统计，并将统计结果存储在历史记录表（etherHistoryTable）中以便管理设备随时查看。历史组统计的是每个周期内端口接收报文的情况，统计数据包括带宽利用率、错误包数和总包数等，周期的长短可以通过命令行来配置。

3. 事件组

事件组用来定义事件索引号及事件的处理方式。事件组定义的事件用于告警组表项和扩展告警组表项中。当监控对象达到告警条件时，就会触发事件，事件有如下几种处理方式：

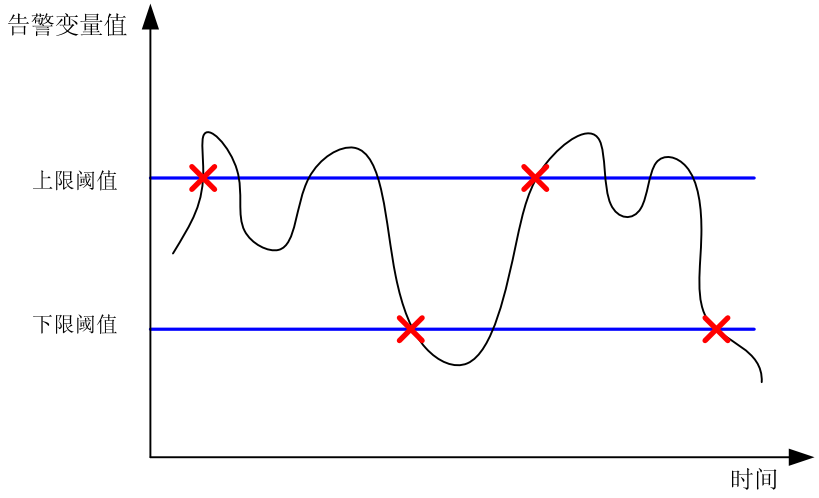
- Log：将事件相关信息（事件发生的时间、事件的内容等）记录在本设备 RMON MIB 的事件日志表中，以便管理设备通过 SNMP Get 操作进行查看。
- Trap：表示事件被触发时，会生成告警信息发送给设备的 SNMP 模块。
- Log-Trap：表示事件被触发时，既在本设备上记录日志，又会生成告警信息发送给设备的 SNMP 模块。
- None：不做任何处理。

4. 告警组

RMON 告警管理可对指定的告警变量（如端口收到的报文总数 etherStatsPkts）进行监视。用户定义了告警表项后，系统会按照定义的时间周期去获取被监视的告警变量的值，当告警变量的值大于或等于上限阈值时，触发一次上限告警事件；当告警变量的值小于或等于下限阈值，触发一次下限告警事件，告警管理将按照事件的定义进行相应的处理。

当告警变量的采样值在同一方向上连续多次超过阈值时，只会在第一次产生告警事件，后面的几次不会产生告警事件。即上限告警和下限告警是交替产生的，出现了一次上限告警，则下一次必为下限告警。如 图 1-1 所示，告警变量的值（如图中黑色曲线所示）多次超过阈值（如图中蓝色直线所示），产生了多个交叉点，但只有红叉标识的交叉点才会触发告警事件，其它交叉点不会触发告警事件。

图1-1 上下限告警示意图



5. 扩展告警组

扩展告警表项可以对告警变量进行运算，然后将运算结果和设置的阈值比较，实现更为丰富的告警功能。

用户定义了扩展告警表项后，系统对扩展告警表项的处理如下：

(1) 对定义的扩展告警公式中的告警变量按照定义的时间间隔进行采样。

(2) 将采样值按照定义的运算公式进行计算。

(3) 将计算结果和设定的阈值进行比较，大于或等于上限阈值时，触发一次上限告警事件；小于或等于下限阈值，触发一次下限告警事件。

与告警组一样，当扩展告警组的运算结果在同一方向上连续多次超过阈值时，只会在第一次产生告警事件，后面的几次不会产生告警事件，即上限告警和下限告警是交替产生的。

1.1.4 协议规范

与 RMON 相关的协议规范有：

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

1.2 配置RMON统计功能

RMON 的统计功能可以通过 RMON 统计组或者 RMON 历史组来实现，但是两者统计的对象不同，请根据实际需要配置。

- RMON统计组统计的是RMON以太网统计表里定义的变量，记录的是从RMON统计表项创建到当前阶段变量的累加值，具体配置请参见“1.2.1 配置RMON以太网统计功能”。
- RMON历史组统计的是RMON历史记录表里定义的变量，记录的是每个周期内变量的累加值，具体配置请参见“1.2.2 配置RMON历史统计功能”。

1.2.1 配置RMON以太网统计功能

表1-1 配置 RMON 以太网统计功能

操作	命令	说明
进入系统视图	**system-view**	-
进入以太网接口视图	**interface** *interface-type interface-number*	-
创建统计表项	**rmon statistics** *entry-number* [**owner** *text*]	缺省情况下，没有创建统计表项 • 每个接口下只能创建一个统计表项，整个设备允许创建的统计表项最大数目为 100 条 • 当统计表项的总数大于 100 条时，创建操作失败 • 以太网子接口下不支持配置 RMON 统计表项。

1.2.2 配置RMON历史统计功能

配置 RMON 历史统计功能时，需要注意：

- 历史控制表项的 *entry-number* 必须全局唯一，如果已经在其他接口下使用，则创建操作失败。
- 同一接口下，可以创建多条历史控制表项，但要求不同表项 *entry-number* 和 *sampling-interval* 的值必须不同，否则创建操作失败。
- 整个设备允许创建的控制历史表项最大数目为 100 条。当控制历史表项的总数大于 100 条时，创建操作失败。
- 在创建历史控制表项时，如果指定的 **buckets** *number* 参数值超出了设备实际支持的历史表容量时，该历史控制表项会被添加，但该表项对应生效的 **buckets** *number* 的值为设备实际支持的历史表容量。

表1-2 配置 RMON 历史统计功能

操作	命令	说明
进入系统视图	**system-view**	-
进入以太网接口视图	**interface** *interface-type interface-number*	-
创建历史控制表项	**rmon history** *entry-number* **buckets** *number* **interval** *sampling-interval* [**owner** *text*]	缺省情况下，没有创建历史控制表项 以太网子接口下不支持配置历史控制表项

1.3 配置RMON告警功能

1.3.1 配置准备

如果触发告警事件时，需要向管理设备（NMS）发送告警信息，则在配置 RMON 告警功能之前，必须保证 SNMP Agent 已经正确配置。SNMP Agent 的配置请参见“网络管理和监控配置指导”中的“SNMP”。

1.3.2 配置限制和指导

- 系统不允许创建两个配置完全相同的表项。如果新建表项参数的值和已存在表项对应参数的值完全相同，则创建操作失败。不同表项需要比较的参数不同，请参见 表 1-3。
- 系统对每种类型表项的总数均进行了限制，具体数目请参见 表 1-3。当某种类型表项的总数达到系统允许创建的最大数目时，创建操作失败。

表1-3 RMON 配置约束表

表项名	需要比较的参数	最多可创建的表项数
事件表项	事件描述（**description** *string*）、事件类型（**log**、**trap**、**logtrap** 或**none**）和团体名（*security-string* ）	60
告警表项	告警变量（*alarm-variable*）、采样间隔（*sampling-interval*）、采样类型（**absolute**或**delta**）、上限阈值（*threshold-value1*）和下限阈值（*threshold-value2*）	60
扩展告警表项	告警变量公式（*prialarm-formula*）、采样间隔（*sampling-interval*）、采样类型（**absolute**或**delta**）、上限阈值（*threshold-value1*）和下限阈值（*threshold-value2*）	50

1.3.3 配置步骤

表1-4 配置 RMON 告警功能

操作	命令	说明
进入系统视图	**system-view**	-
（可选）创建事件表项	**rmon event** *entry-number* [**description** *string*] { **log** | **log-trap** *security-string* | **none** | **trap** *security-string* } [**owner** *text*]	缺省情况下，没有创建事件表项
创建告警表项	**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*]	二者至少选其一 缺省情况下，没有创建告警表/扩展告警表项 配置告警/扩展告警时如果指定的事件表项 *event-entry*不存在，告警/扩展告警也允许配置，但触发告警时不会有事件动作
创建扩展告警表项	**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*]	

1.4 RMON显示和维护

在完成上述配置后，在任意视图下执行 **display** 命令可以显示配置后 RMON 的运行情况，通过查看显示信息，验证配置的效果。

表1-5 RMON 显示和维护

操作	命令
显示RMON统计信息	**display rmon statistics** [*interface-type interface-number*]
显示RMON历史控制表及历史采样信息	**display rmon history** [*interface-type interface-number*]
显示RMON告警表项的相关信息	**display rmon alarm** [*entry-number*]
显示RMON扩展告警表项的相关信息	**display rmon prialarm** [*entry-number*]
显示RMON事件表项的相关信息	**display rmon event** [*entry-number*]
显示事件日志表项的相关信息	**display rmon eventlog** [*entry-number*]

1.5 RMON典型配置举例

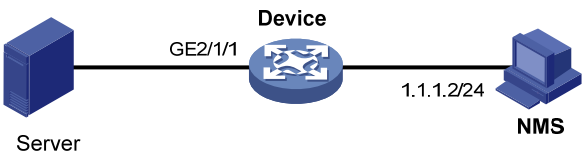
1.5.1 统计功能典型配置举例

1. 组网需求

Device 通过以太网线缆连接 Server。

现需要通过 RMON 统计表对以太网接口 GigabitEthernet2/1/1 接收的报文进行性能统计。管理员随时查看统计数据了解接口接收报文的情况。

图1-2 配置 RMON 组网图



2. 配置步骤

使用 RMON 对接口 GigabitEthernet2/1/1 进行流量统计。

```
<Sysname> system-view
[Sysname] interface gigabitethernet 2/1/1
[Sysname-GigabitEthernet2/1/1] rmon statistics 1 owner user1
```

以上配置完成后系统就开始对接口 GigabitEthernet2/1/1 接收的报文进行分类统计，统计的结果可以通过多种方式查看：

- 查看接口的统计信息。

```
<Sysname> display rmon statistics gigabitethernet 2/1/1
EtherStatsEntry 1 owned by user1 is VALID.
  Interface : GigabitEthernet2/1/1<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
```

- 在 NMS 上通过软件执行 Get 操作，直接获取 MIB 节点的值。

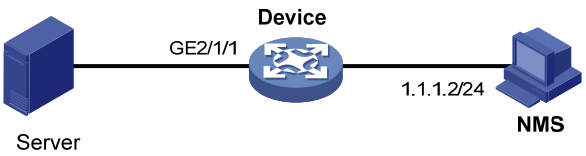
1.5.2 历史统计功能典型配置举例

1. 组网需求

Device 通过以太网线缆连接 Server。

现需要每 1 分钟通过 RMON 历史统计表对以太网接口 GigabitEthernet2/1/1 接收的报文进行周期性统计。管理员能够随时了解短时间内接口是否有数据突发。

图1-3 配置 RMON 组网图



2. 配置步骤

使用 RMON 对接口 GigabitEthernet2/1/1 进行周期性流量统计。

```
<Sysname> system-view
[Sysname] interface gigabitethernet 2/1/1
[Sysname-GigabitEthernet2/1/1] rmon history 1 buckets 8 interval 60 owner user1
```

以上配置完成后，系统就开始对接口 GigabitEthernet2/1/1 接收的报文进行周期性分类统计，每 1 分钟统计一次，历史统计列表里会保存最近的 8 次统计的结果，以便管理员查看。统计的结果可以通过以下方式查看：

- 查看接口的历史统计信息。

```
[Sysname-GigabitEthernet2/1/1] display rmon history
HistoryControlEntry 1 owned by user1 is VALID
  Sampled interface     : GigabitEthernet2/1/1<ifIndex.3>
  Sampling interval     : 60(sec) with 8 buckets max
  Sampling record 1 :
    dropevents        : 0         , octets               : 834
    packets           : 8         , broadcast packets    : 1
    multicast packets : 6         , CRC alignment errors : 0
```

```
  undersize packets : 0         , oversize packets     : 0
  fragments         : 0         , jabbers              : 0
  collisions        : 0         , utilization          : 0
Sampling record 2 :
  dropevents        : 0         , octets               : 962
  packets           : 10        , broadcast packets    : 3
  multicast packets : 6         , CRC alignment errors : 0
  undersize packets : 0         , oversize packets     : 0
  fragments         : 0         , jabbers              : 0
  collisions        : 0         , utilization          : 0
Sampling record 3 :
  dropevents        : 0         , octets               : 830
  packets           : 8         , broadcast packets    : 0
  multicast packets : 6         , CRC alignment errors : 0
  undersize packets : 0         , oversize packets     : 0
  fragments         : 0         , jabbers              : 0
  collisions        : 0         , utilization          : 0
Sampling record 4 :
  dropevents        : 0         , octets               : 933
  packets           : 8         , broadcast packets    : 0
  multicast packets : 7         , CRC alignment errors : 0
  undersize packets : 0         , oversize packets     : 0
  fragments         : 0         , jabbers              : 0
  collisions        : 0         , utilization          : 0
Sampling record 5 :
  dropevents        : 0         , octets               : 898
  packets           : 9         , broadcast packets    : 2
  multicast packets : 6         , CRC alignment errors : 0
  undersize packets : 0         , oversize packets     : 0
  fragments         : 0         , jabbers              : 0
  collisions        : 0         , utilization          : 0
Sampling record 6 :
  dropevents        : 0         , octets               : 898
  packets           : 9         , broadcast packets    : 2
  multicast packets : 6         , CRC alignment errors : 0
  undersize packets : 0         , oversize packets     : 0
  fragments         : 0         , jabbers              : 0
  collisions        : 0         , utilization          : 0
Sampling record 7 :
  dropevents        : 0         , octets               : 766
  packets           : 7         , broadcast packets    : 0
  multicast packets : 6         , CRC alignment errors : 0
  undersize packets : 0         , oversize packets     : 0
  fragments         : 0         , jabbers              : 0
  collisions        : 0         , utilization          : 0
Sampling record 8 :
  dropevents        : 0         , octets               : 1154
  packets           : 13        , broadcast packets    : 1
```

```
multicast packets : 6         , CRC alignment errors : 0
undersize packets : 0         , oversize packets     : 0
fragments         : 0         , jabbers              : 0
collisions        : 0         , utilization          : 0
```

- 在 NMS 上通过软件执行 Get 操作，直接获取 MIB 节点的值。

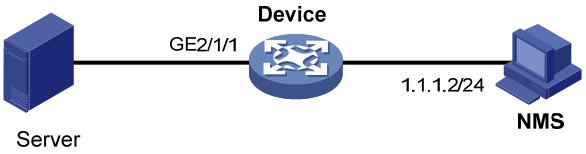
1.5.3 告警功能典型配置举例

1. 组网需求

Device 通过以太网线缆与 Server 和 NMS 相连。

GigabitEthernet2/1/1 接口连接 Server，现需要对该服务器的流量进行统计。以 5 秒的采样间隔采样，当流量过大或过低时发送相应的告警信息给设备的 SNMP 模块，由 SNMP 模块通知 NMS。

图1-4 配置 RMON 组网图



2. 配置步骤

配置 SNMP Agent。（请注意：这些参数的值应与 NMS 侧的配置一致，假设 NMS 上运行 SNMP v1，访问设备时使用的可读团体名为 public，可写团体名为 private，NMS 的 IP 地址为 1.1.1.2）

```
<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
```

使用 RMON 对以太网接口 GigabitEthernet2/1/1 进行统计。

```
[Sysname] interface gigabitethernet 2/1/1
[Sysname-GigabitEthernet2/1/1] rmon statistics 1 owner user1
[Sysname-GigabitEthernet2/1/1] quit
```

配置 RMON 告警表项。（当节点 1.3.6.1.2.1.16.1.1.1.4.1 的相对采样值超过 100 时或者低于 50 时，都触发事件 1—生成告警信息发送给设备的 SNMP 模块）。

```
[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
```

查看 RMON 告警表信息。

```
<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
```

查看以太网接口的统计信息。

```
<Sysname> display rmon statistics gigabitethernet 2/1/1
EtherStatsEntry 1 owned by user1 is VALID.
  Interface : GigabitEthernet2/1/1<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
```

完成以上配置后，当告警事件被触发时，在 NMS 的告警管理部分可以查看相应的记录。在设备侧也会有相应信息，如下所示。

```
[Sysname] % Apr  6 09:23:53:357 2013 sysname SNMP/6/SNMP_NOTIFY: Notification fallingA
larm(1.3.6.1.2.1.16.0.2) with alarmIndex(1.3.6.1.2.1.16.3.1.1.1.1)=1;alarmVariab
le(1.3.6.1.2.1.16.3.1.1.3.1)=1.3.6.1.2.1.16.1.1.1.4.1;alarmSampleType(1.3.6.1.2.
1.16.3.1.1.4.1)=2;alarmValue(1.3.6.1.2.1.16.3.1.1.5.1)=0;alarmFallingThreshold(1
.3.6.1.2.1.16.3.1.1.8.1)=50.
```

目 录

1 NETCONF

说明

设备运行于 FIPS 模式时，本特性部分配置相对于非 FIPS 模式有所变化，具体差异请见本文相关描述。有关 FIPS 模式的详细介绍请参见“安全配置指导”中的“FIPS”。

1.1 NETCONF简介

NETCONF（Network Configuration Protocol，网络配置协议）是一种基于 XML 的网络管理协议，它提供了一种可编程的、对网络设备进行配置和管理的方法。用户可以通过该协议设置参数、获取参数值、获取统计信息等。

NETCONF 报文使用 XML 格式，具有强大的过滤能力，而且每一个数据项都有一个固定的元素名称和位置，这使得同一厂商的不同设备具有相同的访问方式和结果呈现方式，不同厂商之间的设备也可以经过映射 XML 得到相同的效果，这使得它在第三方软件的开发上非常便利，很容易开发出在混合不同厂商、不同设备的环境下的特殊定制的网管软件。在这样的网管软件的协助下，使用 NETCONF 功能会使网络设备的配置管理工作，变得更简单更高效。

1.1.1 NETCONF协议结构

NETCONF 协议采用了分层结构，分成四层：内容层、操作层、RPC（Remote Procedure Call，远程调用）层和通信协议层。

表1-1 XML 分层与 NETCONF 分层模型对应关系

NETCONF 分层	XML 分层	说明
Content	配置数据、状态数据、统计信息等	内容层表示的是被管理对象的集合，可以是配置数据、状态数据、统计信息等。NETCONF协议具体可读写的数据请参见产品的XSD（XML Schema Definition，XML架构定义）标准
Operations	<get>,<get-config>,<edit-config>…	操作层定义了一系列在RPC中应用的基本的原语操作集，这些操作将组成NETCONF的基本能力。NETCONF全面地定义了对被管理设备的九种基础操作 设备支持的操作请参见“2.1 附录 A Comware V7中支持的NETCONF操作类型”
RPC	<rpc>,<rpc-reply>	RPC层为RPC模块的编码提供了一个简单的、传输协议无关的机制。通过使用<rpc>和<rpc-reply>元素分别对NETCONF请求和响应数据（即操作层和内容层的内容）进行封装

NETCONF 分层	XML 分层	说明
Transport Protocol	非FIPS模式下：Console/Telnet/SSH/ TLS FIPS模式下： Console/Telnet/SSH/ TLS	非FIPS模式下：通信协议层为NETCONF提供面向连接的、可靠的、顺序的数据链路。NETCONF支持Telnet、SSH、Console等CLI登录方式/协议，即NETCONF over SSH/Telnet/Console；NETCONF还支持封装成SOAP（Simple Object Access Protocol，简单对象访问协议）报文后通过HTTP协议传输，即NETCONF over SOAP over HTTP FIPS模式下： 通信协议层为NETCONF提供面向连接的、可靠的、顺序的数据链路。NETCONF支持Telnet、SSH、Console等CLI登录方式/协议，即NETCONF over SSH/Telnet/Console

1.1.2 NETCONF报文格式

1. NETCONF

- NETCONF 命令必须符合 XML 语言的基本格式。
- NETCONF 和 NETCONF over SOAP 报文格式请遵循 RFC 4741。
- NETCONF 操作以及可操作的数据项，请遵循 H3C 公司的 XSD（XML Schema Definition，XML 架构定义）标准，该标准在对外提供的 XSD 文档中描述。NETCONF 报文都将经过 XSD 校验才会下发，如果校验失败则会向客户端报错。

说明

在 XML 视图下进行 NETCONF 配置时，XML 报文最后需要添加“]]>]]>”，否则设备无法识别。

如下为一个 NETCONF 报文示例，用于获取设备上所有接口的所有参数：

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

2. NETCONF over SOAP

NETCONF over SOAP 之后，NETCONF 报文会放在 SOAP 报文的 BODY 元素里，这些报文除了需要遵循纯 NETCONF 报文的规则外，还需要遵循以下规则：

- SOAP 消息必须用 XML 来编码。
- SOAP 消息必须使用 SOAP Envelope 命名空间。

- SOAP 消息必须使用 SOAP Encoding 命名空间。
- SOAP 消息不能包含 DTD（Decument Type Definition，文件类型定义）引用。
- SOAP 消息不能包含 XML 处理指令。

如下为一个 NETCONF over SOAP 报文示例，用于获取设备上所有接口的所有参数：

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <auth:Authentication env:mustUnderstand="1"
xmlns:auth="http://www.h3c.com/netconf/base:1.0">
      <auth:AuthInfo>800207F0120020C</auth:AuthInfo>
    </auth:Authentication>
  </env:Header>
  <env:Body>
    <rpc message-id ="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <get-bulk>
        <filter type="subtree">
          <top xmlns="http://www.h3c.com/netconf/data:1.0">
            <Ifmgr>
              <Interfaces>
                <Interface/>
              </Interfaces>
            </Ifmgr>
          </top>
        </filter>
      </get-bulk>
    </rpc>
  </env:Body>
</env:Envelope>
```

1.1.3 如何使用NETCONF

用户可通过以下二种方式来使用 NETCONF 协议配置/管理设备：

- 用户可以通过 Telnet、SSH、Console 登录到设备的 CLI 界面，通过命令行界面编辑下发 NETCONF 指令。该方式一般用于研发和测试环境。在 XML 视图下，将合法的 NETCONF 报文直接拷贝、粘贴到命令行提示符处，即可验证设备的 NETCONF 功能是否运行正常。
- 用户使用自己开发的 Web 配置工具给设备下发 NETCONF 指令来实现对设备的访问。因此，使用该方式前必须使能 NETCONF over SOAP 功能。

1.1.4 协议规范

与 NETCONF、SOAP 协议相关的协议规范有：

- 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

1.2 NETCONF配置任务简介

表1-2 NETCONF 配置任务简介

配置任务	说明	详细配置
使能NETCONF over SOAP功能	可选	1.3
建立NETCONF会话	必选	1.4
向设备进行事件订阅	可选	1.5
给当前配置加锁/解锁	可选	1.6
<get>/<get-bulk>获取信息	可选	1.7.1
<get-config>/<get-bulk-config>获取配置信息	可选	1.7.2
<edit-config>编辑指定模块数据	可选	1.7.3
配置保存、回滚、加载	可选	1.8
通配符和过滤器功能	可选	1.9
命令行操作	可选	1.10
获取会话信息	可选	1.11
关闭另一个会话	可选	1.12
退出XML视图	可选	1.13

1.3 使能NETCONF over SOAP功能

NETCONF 支持封装成 SOAP 报文后通过 HTTP 协议传输，即 NETCONF over SOAP over HTTP，使用该功能后，用户可以通过开发 Web 配置工具给设备下发 NETCONF 指令来实现对设备的访问。

表1-3 配置 SOAP 服务

操作	命令	说明
使能HTTP的SOAP功能	**netconf soap http enable**	该命令在系统视图下执行 FIPS模式下，不支持本命令 缺省情况下，基于HTTP的SOAP功能处于关闭状态

1.4 建立NETCONF会话

提示

客户端必须给设备发送 hello 信息，完成能力集的交互后，设备才会处理客户端发送的其它请求。

设备同时最多能建立 5 个 NETCONF 连接，即最多支持 5 个用户同时使用 NETCONF 功能管理和监控设备。用户数超过上限后，新登录的用户将登录失败。

1.4.1 进入XML视图

表1-4 进入 XML 视图

操作	命令	说明
进入XML视图	**xml**	该命令在用户视图下执行

1.4.2 交换能力集

进入 XML 视图后，客户端和设备必须交换各自支持的能力集，双方收到对方的能力集后才可以进行下一步操作。

1. 设备发送给客户端的报文

客户端进入 XML 视图后，设备会发送如下报文自动告知客户端支持的 NETCONF 能力集：

```
<?xml version="1.0" encoding="UTF-8"?><hello
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"><capabilities><capability>urn:ietf:param
s:netconf:base:1.1</capability><capability>urn:ietf:params:netconf:writable-running</cap
ability><capability>urn:ietf:params:netconf:capability:notification:1.0</capability><cap
ability>urn:ietf:params:netconf:capability:validate:1.1</capability><capability>urn:ietf
:params:netconf:capability:interleave:1.0</capability><capability>urn:ietf:params:netcon
f:capability:h3c-netconf-ext:1.0</capability></capabilities><session-id>1</session-id></
hello>]]>]]>
```

其中：

- <capabilities>和</capabilities>之间的内容表示设备支持的能力集。
- <session-id>和</session-id>之间的内容表示为本次会话分配的会话 ID，用来唯一标识本次会话。

2. 客户端发送给设备的报文

客户端收到设备发送的能力集协商报文后，需要给设备发送如下格式的报文，告知设备客户端支持哪些 NETCONF 能力集。

Hello 协商报文格式如下：

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

```
 </capabilities>
</hello>
```

其中，*capability-set* 表示客户端支持的能力集，由用户定义。一个<capability>和</capability>选项对中填写一个能力集，可以使用多个选项对，发送多个能力集。

1.5 向设备进行事件订阅

用户向设备订阅事件后，设备上发生用户订阅的事件时，设备会自动向订阅的客户端发送事件的相关信息，信息内容包括事件的 code、group、severity 以及发生时间和描述信息。

需要注意的是：

- 订阅只对当前连接生效。如果连接断开，订阅会自动取消。
- 多次发送订阅报文，可以订阅多个事件。用户可以订阅哪些事件，请参见 H3C 对外提供的 XSD 文档。目前设备只支持订阅日志信息。

1.5.1 订阅事件

1. 客户端发送报文

事件订阅报文格式如下：

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <create-subscription  xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <stream>NETCONF</stream>
      <filter>
        <event xmlns="http://www.h3c.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>
```

其中：

- stream 表示支持的事件流，目前只支持 NETCONF。
- event 表示订阅的事件，用户可订阅的事件，以及该变量的表达方式，请参见产品对外提供的文档和 XSD 文档。如果不符合这些约定，设备会向客户端报错，错误信息中的错误类型、错误标签、错误级别和错误描述请参见 RFC 4741 和 RFC 5277，本文不再赘述。
- code 表示日志信息中的助记符。
- group 表示日志信息中的模块名。
- severity 表示日志信息中的安全级别。
- start-time 表示订阅的开始时间。
- stop-time 表示结束订阅的时间

2. 结果验证

设备收到订阅报文后会回应客户端，当客户端收到如下报文时，表示订阅成功：

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="101" xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
    <ok/>
</rpc-reply >
```

当订阅出错时设备会返回错误信息，例如：

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<rpc-error>
   <error-type>error-type</error-type>
   <error-tag>error-tag</error-tag>
   <error-severity>error-severity</error-severity>
   <error-message xml:lang="en">error-message</error-message>
</rpc-error>
</rpc-reply>
```

错误报文的详细定义请参见 RFC 4741。

1.5.2 订阅事件举例

1. 组网需求

客户端订阅没有时间限制的全部事件。订阅后在断开连接之前，设备发生的所有事件都会发送给客户端。

2. 配置步骤

进入 XML 视图。

```
<Sysname> xml
```

进行能力交换。

请将以下报文拷贝、粘贴到客户端：

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

订阅全部事件，不限制订阅时间。

请将以下报文拷贝、粘贴到客户端：

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <create-subscription xmlns ="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <stream>NETCONF</stream>
  </create-subscription>
</rpc>
```

3. 结果验证

如果客户端收到如下报文，则表示订阅成功：

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
    <ok/>
</rpc-reply>
```

当设备上的风扇 1 有问题时，设备会发送如下报文来通知订阅客户端：

```
<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <eventTime>2011-01-04T12:30:46</eventTime>
    <event xmlns="http://www.h3c.com/netconf/event:1.0">
        <Group>DEV</Group>
        <Code>FAN_DIRECTION_NOT_PREFERRED</Code>
        <Slot>1</Slot>
        <Severity>Alert</Severity>
        <context>Fan 1 airflow direction is not preferred on slot 1, please check
it.</context>
    </event>
</notification>
```

当用户（IP 地址为 192.168.100.130）登录设备时，设备会发送如下报文来通知订阅客户端：

```
<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <eventTime>2011-01-04T12:30:52</eventTime>
    <event xmlns="http://www.h3c.com/netconf/event:1.0">
        <Group>SHELL</Group>
        <Code>SHELL_LOGIN</Code>
        <Slot>1</Slot>
        <Severity>Notification</Severity>
        <context>VTY logged in from 192.168.100.130.</context>
    </event>
</notification>
```

1.6 给当前配置加锁/解锁

因为设备同时最多能建立 5 个 NETCONF 连接，即最多支持 5 个用户同时使用 NETCONF 功能管理和监控设备。所以，当用户管理、维护设备或者定位网络问题时，为防止其他 NETCONF 用户修改当前配置、引入干扰，可以使用本特性给当前配置加锁。给当前配置加锁后，只有持有锁的用户可以修改设备的当前配置，其他 NETCONF 用户只能读取，不能修改当前配置。

只有持有锁的用户可以解锁，解锁后其他用户才可以修改设备的当前配置或另外加锁。如果持有锁的用户的当前连接断开，系统会自动解锁。

1.6.1 给当前配置加锁

1. 客户端发送报文

目前设备只支持对当前配置加锁，不能对具体的功能模块进行加锁。请将以下报文拷贝、粘贴到客户端，用户即能完成加锁操作：

```
  <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <lock>
```

```
      <target>
        <running/>
      </target>
    </lock>
  </rpc>
```

2. 结果验证

设备收到加锁报文后会回应客户端，当客户端收到如下报文时，表示加锁成功：

```
<?xml version="1.0" encoding="UTF-8" ?>
  <rpc-reply message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
     <ok/>
</rpc-reply>
```

1.6.2 给当前配置解锁

提示

只有锁的持有用户才能解锁，其它用户不能解锁。

---

1. 客户端发送报文

请将以下报文拷贝、粘贴到客户端，用户即能完成解锁操作：

```
  <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <unlock>
      <target>
        <running/>
      </target>
    </unlock>
  </rpc>
```

2. 结果验证

设备收到解锁报文后会回应客户端，当客户端收到如下报文时，表示解锁成功：

```
<?xml version="1.0" encoding="UTF-8" ?>
    <rpc-reply message-id="101"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <ok/>
</rpc-reply>
```

1.6.3 当前配置加锁举例

1. 组网需求

给设备加锁，以免其它用户使用 XML 语言修改设备的当前配置。

2. 配置步骤

进入 XML 视图。

```
<Sysname> xml
```

进行能力交换。

请将以下报文拷贝、粘贴到客户端：

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

对当前配置加锁。

请将以下报文拷贝、粘贴到客户端：

```
  <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <lock>
      <target>
        <running/>
      </target>
    </lock>
  </rpc>
```

3. 结果验证

如果客户端收到如下报文，则表示加锁成功：

```
<?xml version="1.0" encoding="UTF-8" ?>
    <rpc-reply message-id="101"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <ok/>
</rpc-reply>
```

用户加锁成功后，另一客户端发送加锁报文，设备会返回如下报文：

```
<?xml version="1.0" encoding="UTF-8" ?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
<rpc-error>
  <error-type>protocol</error-type>
  <error-tag>lock-denied</error-tag>
  <error-severity>error</error-severity>
  <error-message xml:lang="en">Lock failed because the NETCONF lock is held by another
session.</error-message>
  <error-info>
    <session-id>1</session-id>
  </error-info>
  </rpc-error>
</rpc-reply>
```

以上报文表明：加锁失败，session-id 是 1 的用户已经持有锁。

1.7 业务处理功能

NETCONF 支持用户对设备进行业务操作，包括对指定信息的获取和修改。基本标签有<get>、<get-bulk>、<get-config>、<get-bulk-config>和<edit-config>，分别用来获取所有数据、获取配置数据和编辑指定模块的数据。详细规则参见 RFC 4741 和 XSD 文档。

1.7.1 <get>/<get-bulk>获取信息

<get>操作用来获取数据，包括运行状态数据和配置数据。

<get-bulk>操作用来从指定索引的下一条开始批量获取后续 N 条数据（索引行数据不返回），包括运行状态数据和配置数据。用户通过 index 属性指定索引，通过 count 属性指定 N。如未指定索引，则以第一条为索引；如未指定 N，或者数据表中符合条件的数据记录不足 N 条，则返回表中所有剩下的数据条目。

<get>操作会返回所有符合条件的数据，在某些情况下，会导致获取数据效率不高。<get-bulk>允许用户从固定数据项开始，向后获取指定条目的数据记录。

1. 客户端发送报文

<get>和<get-bulk>报文的通用格式如下：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <getoperation>
    <filter>
      <top xmlns=" http://www.h3c.com/netconf/data:1.0">
          指定模块，子模块，表名，列名
      </top>
    </filter>
  </getoperation>
</rpc>
```

其中，*getoperation* 可以为 get 或者 get-bulk。<filter>选项用于过滤信息，<filter>中可包括模块名、子模块名、表名和列名：

- 如果不指定模块（子模块），则表示全部模块（子模块）。一旦指定模块（子模块），则返回数据只包含指定模块（子模块）。
- 如果模块下不指定表，则表示全部表。一旦指定表，则返回数据只包含指定表。
- 如果只指定索引列，则返回的数据包括全部的列。如果同时指定了索引列之外的其他列，则意味着返回的数据仅仅包含索引列和指定的列。

<get-bulk>操作报文中还可以携带 count 和 index 参数，如下为一个携带了 count 和 index 参数的<get-bulk>操作报文示例：

```
<rpc message-id ="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:xc="http://www.h3c.com/netconf/base:1.0">
  <get-bulk>
    <filter type="subtree">
      <top xmlns="http://www.h3c.com/netconf/data:1.0"
xmlns:base="http://www.h3c.com/netconf/base:1.0">
        <Syslog>
          <Logs xc:count="5">
            <Log>
              <Index>10</Index>
            </Log>
            </Logs>
        </Syslog>
      </top>
    </filter>
  </get-bulk>
```

```
</rpc>
```

其中，<get-bulk>操作报文中的 count 属性遵循如下约定：

- count 属性的位置在可以从 top 下的节点开始，到表节点为止，即：模块节点，表节点这几个位置都能放置 count 属性，其他位置的 count 将不被解释。
- 如果 count 放在模块节点上，则报文中指定的子孙节点（表）中没有 count 的都默认 count 属性的值和模块一致。
- 如果不指定 count，则获取出从指定索引开始的所有数据。

2. 结果验证

设备收到配置获取请求报文后会将相应参数的值通过如下报文反馈给客户端：

```
<?xml version="1.0"?>
<rpc-reply message-id="100"
           xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
     全部配置数据和状态数据
  </data>
</rpc-reply>
```

1.7.2 <get-config>/<get-bulk-config>获取配置信息

<get-config>和<get-bulk-config>用来获取系统中所有可配置的变量的值，配置的方式包括 CLI、MIB、Web 等。<get-config>和<get-bulk-config>操作报文中可以包含子标签<filter>，用来对要获取的信息进行过滤。

1. 客户端发送报文：

<get-config>和<get-bulk-config>的通用报文格式如下：

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source>
      <running/>
    </source>
    <filter>
      <top xmlns="http://www.h3c.com/netconf/config:1.0">
          指定模块，子模块，表名，列名
      </top>
    </filter>
  </get-config>
</rpc>
```

2. 结果验证

设备收到配置获取请求报文后会将相应配置通过如下报文反馈给客户端：

```
<?xml version="1.0"?>
<rpc-reply message-id="100"   xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    所有指定filter内的数据
  </data>
</rpc-reply>
```

1.7.3 <edit-config>编辑指定模块数据

<edit-config>支持如下选项：merge、create、replace、remove、delete、默认操作选项、默认错误处理选项、测试处理，关于这些选项的详细描述请参见“2.1 附录 A Comware V7 中支持的NETCONF操作类型”。

1. 客户端发送报文

```
<rpc message-id="100"  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target><running></running></target>
<error-option>
   失败时默认操作
</error-option>
    <config>
      <top xmlns="http://www.h3c.com/netconf/config:1.0">
        指定模块名，子模块名，列名，表名
      </top>
    </config>
  </edit-config>
</rpc>
```

2. 结果验证

- 设备收到 edit-config 请求后会回应客户端，当客户端收到如下报文时，表示设置成功：

```
<?xml version="1.0">
<rpc-reply message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

- 另外，用户还可以通过<get>操作可以查看参数的当前值是否和 edit-config 操作设置的值一致。

1.7.4 举例——获取所有模块所有配置数据

1. 组网需求

获取所有模块所有配置数据。

2. 配置步骤

进入 XML 视图。

```
<Sysname> xml
```

进行能力交换。

请将以下报文拷贝、粘贴到客户端：

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

获取所有模块所有配置数据。

请将以下报文拷贝、粘贴到客户端：

```
<rpc message-id="100"
     xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source>
      <running/>
    </source>
  </get-config>
</rpc>
```

3. 结果验证

如果客户端收到类似如下的报文，则表示操作成功：

```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:web="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
    <data>
        <top xmlns="http://www.h3c.com/netconf/config:1.0">
            <Ifmgr>
                <Interfaces>
                    <Interface>
                        <IfIndex>1307</IfIndex>
                        <AdminStatus>2</AdminStatus>
                    </Interface>
                    <Interface>
                        <IfIndex>1308</IfIndex>
                        <AdminStatus>2</AdminStatus>
                    </Interface>
                    <Interface>
                        <Index>1309</Index>
                        <AdminStatus>2</AdminStatus>
                    </Interface>
                    <Interface>
                        <IfIndex>1311</IfIndex>
                        <VlanType>2</VlanType>
                    </Interface>
                    <Interface>
                        <Index>1313</Index>
                        <VlanType>2</VlanType>
                    </Interface>
                </Interfaces>
            </Ifmgr>
            <Syslog>
                <LogBuffer>
                    <BufferSize>120</BufferSize>
                </LogBuffer>
            </Syslog>
            <System>
                <Device>
```

```
                    <SysName>H3C</SysName>
                    <TimeZone>
                        <Zone>+11:44</Zone>
                        <ZoneName>beijing</ZoneName>
                    </TimeZone>
                </Device>
            </System>
            <Fundamentals>
                <WebUI>
                    <SessionAgingTime>98</SessionAgingTime>
                </WebUI>
            </Fundamentals>
        </top>
    </data>
</rpc-reply>
```

1.7.5 举例——获取Syslog模块的所有配置数据

1. 组网需求

获取 Syslog 模块的所有配置数据

2. 配置步骤

进入 XML 视图。

```
<Sysname> xml
```

进行能力交换。

请将以下报文拷贝、粘贴到客户端：

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

获取 Syslog 模块的所有配置数据。

请将以下报文拷贝、粘贴到客户端：

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

```
</rpc>
```

3. 结果验证

如果客户端收到类似如下的报文，则表示操作成功：

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
    <data>
        <top xmlns="http://www.h3c.com/netconf/config:1.0">
            <Syslog>
                <LogBuffer>
                    <BufferSize>120</BufferSize>
                </LogBuffer>
            </Syslog>
        </top>
    </data>
</rpc-reply>
```

1.7.6 举例——取接口表的一条数据

1. 组网需求

取接口表的一条数据

2. 配置步骤

进入 XML 视图。

```
<Sysname> xml
```

进行能力交换。

请将以下报文拷贝、粘贴到客户端：

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

取接口表的一条数据。

请将以下报文拷贝、粘贴到客户端：

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:web="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-bulk>
    <filter type="subtree">
      <top xmlns="http://www.h3c.com/netconf/data:1.0"
xmlns:web="http://www.h3c.com/netconf/base:1.0">
        <Ifmgr>
          <Interfaces web:count="1">
          </Interfaces>
        </Ifmgr>
      </top>
    </filter>
  </get-bulk>
</rpc>
```

3. 结果验证

如果客户端收到类似如下的报文，则表示操作成功：

```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:web="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <data>
    <top xmlns="http://www.h3c.com/netconf/data:1.0">
      <Ifmgr>
        <Interfaces>
          <Interface>
            <IfIndex>3</IfIndex>
            <Name>GigabitEthernet2/1/2</Name>
            <AbbreviatedName>GE2/1/2</AbbreviatedName>
            <PortIndex>3</PortIndex>
            <ifTypeExt>22</ifTypeExt>
            <ifType>6</ifType>
            <Description>GigabitEthernet 2/1/2 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>
```

1.7.7 举例——修改参数值

1. 组网需求

修改 syslog 模块中的日志缓冲区可存储的信息条数为 512。

2. 配置步骤

进入 XML 视图。

```
<Sysname> xml
```

进行能力交换。

请将以下报文拷贝、粘贴到客户端：

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

把 Syslog 模块 LogBuffer 表中的 BufferSize 列，改成 512。

请将以下报文拷贝、粘贴到客户端：

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:web="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>
      <top xmlns="http://www.h3c.com/netconf/config:1.0" web:operation="merge">
        <Syslog>
          <LogBuffer>
            <BufferSize>512</BufferSize>
          </LogBuffer>
        </Syslog>
      </top>
    </config>
  </edit-config>
</rpc>
```

3. 结果验证

如果客户端收到类似如下的报文，则表示操作成功：

```
<?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>
```

1.8 配置保存、回滚、加载

使用 NETCONF 功能，用户可以对设备配置进行保存、回滚和加载操作。

1.8.1 配置保存

1. 客户端发送报文

请将以下报文拷贝、粘贴到客户端，用户即可将设备的当前配置保存到指定名称的文件中：

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <save>
    <file>指定文件的名称</file>
  </save>
</rpc>
```

其中，“指定文件的名称”必须以存储介质的名称开头，后缀为.cfg。如果不指定文件名，则缺省保存到主用下次启动配置文件中。

2. 结果验证

设备收到配置保存请求后会回应客户端，当客户端收到如下报文时，表示保存成功：

```
<?xml version="1.0" encoding="UTF-8" ?>
    <rpc-reply message-id="101"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <ok/>
</rpc-reply>
```

1.8.2 配置回滚

1. 客户端发送报文

请将以下报文拷贝、粘贴到客户端，用户即可将设备的当前配置恢复到指定配置文件中的配置：

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <rollback>
    <file>指定文件的名称</file>
  </rollback>
</rpc>
```

2. 结果验证

设备收到配置回滚请求后会回应客户端，当客户端收到如下报文时，表示配置回滚成功：

```
<?xml version="1.0" encoding="UTF-8" ?>
    <rpc-reply message-id="101"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <ok/>
</rpc-reply>
```

1.8.3 加载文件

<load>操作执行后，指定文件中的配置会被合并到设备的当前配置中。设备会将指定文件中的配置中的配置和当前配置进行比较：对于指定文件中有，但当前配置中没有的配置，直接运行；对于指定文件中和当前配置中不一致的配置，则用指定文件中的配置替换当前配置中的对应配置。

1. 客户端发送报文

请将以下报文拷贝、粘贴到客户端，用户即可将指定文件中的配置追加到当前配置中：

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <load>
    <file>指定文件的名称</file>
  </load>
</rpc>
```

其中，“指定文件的名称”必须以存储介质的名称开头，后缀为.cfg。如果不指定文件名，则缺省保存到主用下次启动配置文件中。

2. 结果验证

设备收到配置加载请求后会回应客户端，当客户端收到如下报文时，表示配置加载成功：

```
<?xml version="1.0" encoding="UTF-8" ?>
    <rpc-reply message-id="101"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <ok/>
</rpc-reply>
```

1.8.4 配置保存举例

1. 组网需求

将设备的当前配置保存到配置文件 my_config.cfg。

2. 配置步骤

进入 XML 视图。

```
<Sysname> xml
```

进行能力交换。

请将以下报文拷贝、粘贴到客户端：

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

将设备的当前配置保存到配置文件 my_config.cfg。

请将以下报文拷贝、粘贴到客户端：

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <save>
    <file> my_config.cfg</file>
  </save>
</rpc>
```

3. 结果验证

如果客户端收到类似如下的报文，则表示操作成功。

```
<?xml version="1.0" encoding="UTF-8" ?>
    <rpc-reply message-id="101"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <ok/>
</rpc-reply>
```

1.9 数据过滤功能

1.9.1 三种数据过滤方式

当用户执行<get>、<get-bulk>、<get-config>或者<get-bulk-config>操作时，在 XML 语言中增加过滤条件，可以使用户只看到自己关心的数据。数据过滤包括严格匹配、正则表达式匹配过滤和条件匹配过滤三种。需要注意的是，同时指定多种过滤条件，则只能生效一个，其优先级从高到底依次为：严格匹配、正则表达式匹配过滤和条件匹配过滤。

1. 严格匹配

用户在 XML 语言中直接指定对应的元素值，设备将对这些值进行严格匹配。如果指定了多个元素的值，则返回同时符合这几个条件的数据。只有完全匹配条件才返回成功。

如下为一个 NETCONF 报文示例，用于获取所有状态为 UP 的接口的配置信息：

```
<rpc message-id ="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter type="subtree">
      <top xmlns="http://www.h3c.com/netconf/data:1.0">
        <Ifmgr>
```

```
          <Interfaces>
            <Interface>
              <AdminStatus>2</AdminStatus>
            </Interface>
          </Interfaces>
        </Ifmgr>
      </top>
    </filter>
  </get>
</rpc>
```

2. 正则表达式匹配

当过滤条件比较复杂时，可以在指定元素上设置 regExp 属性为一个正则表达式，以完成过滤的目的。

如下为一个 NETCONF 报文示例，用于获取接口的描述信息，并要求这些描述信息全部为大写字母，不能有其它字符。

```
<rpc message-id="1-0" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:h3c="http://www.h3c.com/netconf/base:1.0" >
  <get-config>
    <source>
      <running/>
    </source>
    <filter type="subtree">
      <top xmlns="http://www.h3c.com/netconf/config:1.0">
        <Ifmgr>
          <Interfaces>
            <Interface>
              <Description h3c:regExp="^[A-Z]*$" />
            </Interface>
          </Interfaces>
        </Ifmgr>
      </top>
    </filter>
  </get-config>
</rpc>
```

3. 条件匹配

由于正则表达式仅能够完成字符匹配，对于数值逻辑的判断过滤实现起来比较麻烦，此时，可使用条件匹配过滤功能。

条件匹配通过在元素中增加 match 属性完成，属性的值（即过滤条件）可以为数字、字符串。

表1-5 条件匹配命令

操作	命令	说明
大于	match=”more:*value*”	值大于*value*，支持的数据类型为：日期、数字、字符串
小于	match=”less:*value*”	值小于*value*，支持的数据类型为：日期、数字、字符串
不小于	match=”notLess:*value*”	值不小于*value*，支持的数据类型为：日期、数字、字符串

操作	命令	说明
不大于	match=”notMore:*value*”	值不大于*value*，支持的数据类型为：日期、数字、字符串
等于	match=”equal:*value*”	值等于*value*，支持的数据类型为：日期、数字、字符串、OID、BOOL
不等于	match=”notEqual:*value*”	值不等于*value*，支持的数据类型为：日期、数字、字符串、OID、BOOL
包含	match=”include:*string*”	包含字符串*string*，支持的数据类型为：字符串
不包含	match=”exclude:*string*”	不能包含字符串*string*，支持的数据类型为：字符串
开始于	match=”startWith:*string*”	以字符串*string*开头，支持的数据类型为：字符串、OID
结束于	match=”endWith:*string*”	以字符串*string*结束，支持的数据类型为：字符串

如下为一个 NETCONF 报文示例，用于获取实体扩展信息中 CPU 利用率大于 50%的实体。

```
<rpc message-id="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:h3c="http://www.h3c.com/netconf/base:1.0" >
  <get>
    <filter type="subtree">
      <top xmlns="http://www.h3c.com/netconf/data:1.0">
        <Device>
          <ExtPhysicalEntities>
            <Entity>
              <CpuUsage h3c:match="more:50"></CpuUsage>
            </Entity>
          </ExtPhysicalEntities>
        </Device>
      </top>
    </filter>
  </get>
</rpc>
```

1.9.2 正则表达式操作举例

1. 组网需求

取 IFmgr 模块下 Interfaces 表下 Description 列包含冒号的所有数据。

2. 配置步骤

进入 XML 视图。

```
<Sysname> xml
```

进行能力交换。

请将以下报文拷贝、粘贴到客户端：

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

```
  </capabilities>
</hello>
```

取 IFmgr 模块 Interfaces 表下 Description 列包含冒号的所有数据。

请将以下报文拷贝、粘贴到客户端：

```
<rpc message-id="100"
     xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:reg="http://www.h3c.com/netconf/base:1.0">
  <get>
    <filter type="subtree">
      <top xmlns="http://www.h3c.com/netconf/data:1.0">
        <Ifmgr>
          <Interfaces>
            <Interface>
              <Description  reg:regExp=":" />
            </Interface>
          </Interfaces>
        </Ifmgr>
      </top>
    </filter>
  </get>
</rpc>
```

3. 结果验证

如果客户端收到类似如下的报文，则表示操作成功。

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:reg="http://www.h3c.com/netconf/base:1.0" message-id="100">
    <data>
        <top xmlns="http://www.h3c.com/netconf/data:1.0">
            <Ifmgr>
                <Interfaces>
                    <Interface>
                        <IfIndex>2681</IfIndex>
                        <Description>GigabitEthernet2/1/4:1 Interface</Description>
                    </Interface>
                    <Interface>
                        <IfIndex>2682</IfIndex>
                        <Description>GigabitEthernet2/1/4:2 Interface</Description>
                    </Interface>
                    <Interface>
                        <IfIndex>2683</IfIndex>
                        <Description>GigabitEthernet2/1/4:3 Interface</Description>
                    </Interface>
                    <Interface>
                        <IfIndex>2684</IfIndex>
                        <Description>GigabitEthernet2/1/4:4 Interface</Description>
                    </Interface>
                    <Interface>
```

```
                    <IfIndex>2685</IfIndex>
                    <Description>GigabitEthernet2/1/5:1 Interface</Description>
                </Interface>
                <Interface>
                    <IfIndex>2686</IfIndex>
                    <Description>GigabitEthernet2/1/5:2 Interface</Description>
                </Interface>
                <Interface>
                    <IfIndex>2687</IfIndex>
                    <Description>GigabitEthernet2/1/5:3 Interface</Description>
                </Interface>
                <Interface>
                    <IfIndex>2688</IfIndex>
                    <Description>GigabitEthernet2/1/5:4 Interface</Description>
                </Interface>
                <Interface>
                    <IfIndex>2689</IfIndex>
                    <Description>GigabitEthernet2/1/6:1 Interface</Description>
                </Interface>
                <Interface>
                    <IfIndex>2690</IfIndex>
                    <Description>GigabitEthernet2/1/6:2 Interface</Description>
                </Interface>
                <Interface>
                    <IfIndex>2691</IfIndex>
                    <Description>GigabitEthernet2/1/6:3 Interface</Description>
                </Interface>
                <Interface>
                    <IfIndex>2692</IfIndex>
                    <Description>GigabitEthernet2/1/6:4 Interface</Description>
                </Interface>
            </Interfaces>
        </Ifmgr>
    </top>
</data>
</rpc-reply>
```

1.9.3 条件匹配操作举例

1. 组网需求

取 IFmgr 模块 Interfaces 表下 IFIndex 值大于等于 5000 的 Name 列信息。

2. 配置步骤

进入 XML 视图。

```
<Sysname> xml
```

进行能力交换。

请将以下报文拷贝、粘贴到客户端：

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

```
  <capabilities>
    <capability>
            urn:ietf:params:netconf:base:1.0
    </capability>
  </capabilities>
</hello>
```

#取 IFmgr 模块 Interfaces 表下 IFIndex 值大于等于 5000 的 Name 列信息。

请将以下报文拷贝、粘贴到客户端：

```
<rpc message-id="100"
     xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="http://www.h3c.com/netconf/base:1.0">
  <get>
    <filter type="subtree">
      <top xmlns="http://www.h3c.com/netconf/data:1.0">
        <Ifmgr>
          <Interfaces>
            <Interface>
              <IfIndex nc:match="more:5000"/>
              <Name/>
            </Interface>
          </Interfaces>
        </Ifmgr>
      </top>
    </filter>
  </get>
</rpc>
```

3. 结果验证

如果客户端收到类似如下的报文，则表示操作成功。

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:nc="http://www.h3c.com/netconf/base:1.0" message-id="100">
    <data>
        <top xmlns="http://www.h3c.com/netconf/data:1.0">
            <Ifmgr>
                <Interfaces>
                    <Interface>
                        <IfIndex>7241</IfIndex>
                        <Name>NULL0</Name>
                    </Interface>
                    <Interface>
                        <IfIndex>7243</IfIndex>
                        <Name>Register-Tunnel0</Name>
                    </Interface>
                </Interfaces>
            </Ifmgr>
        </top>
    </data>
```

```
</rpc-reply>
```

1.10 命令行操作

通过 NETCONF 功能，用户可以将命令行封装在 XML 报文中对设备进行操作。

1.10.1 命令行操作

1. 客户端发送报文

当需要给设备发送命令时，请使用格式如下的 NETCONF 报文：

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <CLI>
    <Execution>
          命令行
    </Execution>
  </CLI>
</rpc>
```

一对<Execution>子标签中可以包含多个命令行，一条命令输入完毕，换行，再输入下一条命令即可。

2. 结果验证

设备收到命令行指令后会回应客户端，当客户端收到如下报文时，表示命令行执行成功（注意命令响应被 CDATA 节点包含）：

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <CLI>
    <Execution>
      <![CDATA[对应命令行响应]]>
    </Execution>
  </CLI>
</rpc-reply>
```

1.10.2 命令行操作举例

1. 配置需求

向设备发送显示当前配置命令。

2. 配置步骤

进入 XML 视图。

```
<Sysname> xml
```

进行能力交换。

请将以下报文拷贝、粘贴到客户端：

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

```
  </capabilities>
</hello>
```

向设备发送显示当前配置命令。

请将以下报文拷贝、粘贴到客户端：

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <CLI>
    <Execution>
          display current-configuration
    </Execution>
  </CLI>
</rpc>
```

3. 结果验证

如果客户端收到类似如下的报文，则表示操作成功。

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <CLI>
    <Execution><![CDATA[<ab>display current-configuration
#
 version 5.3.1,
#
 sysname ab
#
 ftp server enable
 ftp update fast
 ftp timeout 2000
#
 domain default enable system
#
 telnet server enable
#
vlan 1
#
vlan 1000
#
radius scheme system
 primary authentication 127.0.0.1 1645
   ]]>
   </Execution>
  </CLI>
</rpc-reply>
```

1.11 获取会话信息

使用该功能用户可以获取当前设备的所有 NETCONF 会话信息。

1.11.1 获取会话信息

1. 客户端发送报文

请将以下报文拷贝、粘贴到客户端：

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-sessions/>
</rpc>
```

2. 结果验证

设备收到命令行指令后会回应客户端，当客户端收到如下报文时，表示命令行执行成功：

```
<?xml version="1.0" encoding="UTF-8" ?>
<rpc-reply message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-sessions>
    <Session>
      <SessionID>用户会话 ID 信息 </SessionID>
      <Line> line 信息</Line>
      <UserName>用户登录名称</UserName>
      <Since>用户登录时间</Since>
      <LockHeld>用户是否持有锁</LockHeld>
    </Session>
  </get-sessions>
</rpc-reply>
```

1.11.2 获取会话信息举例

1. 配置需求

获取会话信息。

2. 配置步骤

进入 XML 视图。

```
<Sysname> xml
```

进行能力交换。

请将以下报文拷贝、粘贴到客户端：

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

获取设备上当前存在的 NETCONF 会话的信息。

请将以下报文拷贝、粘贴到客户端：

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-sessions/>
</rpc>
```

3. 结果验证

如果客户端收到类似如下的报文，则表示操作成功。

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
    <get-sessions>
        <Session>
            <SessionID>1</SessionID>
            <Line>vty0</Line>
            <UserName></UserName>
            <Since>2011-01-05T00:24:57</Since>
            <LockHeld>false</LockHeld>
        </Session>
    </get-sessions>
</rpc-reply>
```

以上信息表明：目前有一个 NETCONF 连接，SessionID 是 1，用户登录的接口为 vty0，登录时间是 2011-01-05T00:24:57，此用户不持有锁。

1.12 关闭另一个会话

NETCONF 支持一个用户关闭除自己外的另一个 NETCONF 会话，被关闭会话的用户退回到用户视图。

1.12.1 Kill-session操作

1. 客户端发送报文

请将以下报文拷贝、粘贴到客户端，用户即能关闭指定的会话：

```
<rpc message-id="101"    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <kill-session>
    <session-id>
        指定 session-ID
    </session-id>
  </kill-session>
</rpc>
```

2. 结果验证

设备收到会话关闭请求后会回应客户端，当客户端收到如下报文时，表示指定会话已经被关闭：

```
<?xml version="1.0" encoding="UTF-8" ?>
    <rpc-reply message-id="101"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <ok/>
</rpc-reply>
```

1.12.2 Kill-session举例

1. 配置需求

当前有两个 NETCONF 登录用户，session ID 分别是 1 和 2，session ID 为 1 的用户要关闭另一个用户的会话。

2. 配置步骤

进入 XML 视图。

```
<Sysname> xml
```

进行能力交换。

请将以下报文拷贝、粘贴到客户端：

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

关闭 session ID 为 2 的用户会话。

请将以下报文拷贝、粘贴到客户端：

```
<rpc message-id="101"    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <kill-session>
    <session-id>2</session-id>
  </kill-session>
</rpc>
```

3. 结果验证

如果客户端收到如下报文，则表明 session ID 为 2 的 NETCONF 会话已经被关闭。建立该会话的用户会从 XML 视图退回到用户视图。

```
<?xml version="1.0" encoding="UTF-8" ?>
  <rpc-reply message-id="101"  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <ok/>
</rpc-reply>
```

1.13 退出XML视图

1. 客户端发送报文

对于处于 XML 视图的用户，要退回到用户视图，使用命令行来配置设备时，需要将以下报文拷贝、粘贴到客户端：

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <close-session/>
</rpc>
```

2. 结果验证

设备收到以上请求后会返回如下报文并退回到用户视图：

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<rpc-reply message-id="101"
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
   <ok/>
</rpc-reply>
```

2 附录

2.1 附录 A Comware V7中支持的NETCONF操作类型

Comware V7 平台对NETCONF标准协议做了一些修订，删除了不常用的操作，增加部分新的操作，如 表 2-1 所示。

表2-1 NETCONF 协议支持的操作

操作	说明	XML 格式样例
get	获取数据，包括运行状态数据和配置数据	获取Syslog模块的全部数据的XML请求如下： <rpc message-id ="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:xc="http://www.h3c.com/netconf/base:1.0"> <get> <filter type="subtree"> <top xmlns="http://www.h3c.com/netconf/data:1.0" > <Syslog> </Syslog> </top> </filter> </get> </rpc>
get-config	获取配置数据，和get不同，它只返回非缺省的配置数据。如果没有配置数据，则返回一个空的<data>	获取接口表内所有配置的XML请求如下： <rpc message-id ="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:xc="http://www.h3c.com/netconf/base:1.0"> <get-config> <source> <running/> </source> <filter type="subtree"> <top xmlns="http://www.h3c.com/netconf/config:1.0"> <Ifmgr> <Interfaces> <Interface/> </Interfaces> </Ifmgr> </top> </filter> </get-config> </rpc>

操作	说明	XML 格式样例
get-bulk	从指定索引的下一条开始批量获取后续N条数据（索引行数据不返回），包括运行状态数据和配置数据。用户通过index属性指定索引，通过count属性指定N。如未指定索引，则以第一条为索引；如未指定N，或者数据表中符合条件的数据记录不足N条，则返回表中所有剩下的数据条目 get操作会返回所有符合条件的数据，这在某些情况下会导致效率问题。get-bulk允许用户从固定数据项开始，向后获取指定条目的数据记录	取全部接口的数据的xml请求如下： <rpc message-id ="100" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <get-bulk> <filter type="subtree"> <top xmlns="http://www.h3c.com/netconf/data:1.0"> <lfmgr> <Interfaces xc:count="5" xmlns:xc="http://www.h3c.com/netconf/base:1.0”> <Interface/> </Interfaces> </lfmgr> </top> </filter> </get-bulk> </rpc>
get-bulk-config	从指定索引的下一条批量获取配置数据。和get-config类似，只返回非默认配置；其它约束类似get-bulk	获取全部接口配置信息的xml请求如下： <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.h3c.com/netconf/config:1.0"> <lfmgr> </lfmgr> </top> </filter> </get-bulk-config> </rpc>

操作	说明	XML 格式样例
edit-config: merge	在当前运行配置的基础上直接运行指定配置 merge操作必须指定具体的操作对象（行）： • 如果指定的对象存在，则直接配置该对象 • 如果指定的对象不存在，但允许创建，则先创建再配置该对象 • 如果指定的对象不存在且不允许创建，则返回 merge 失败	将BufferSize设置为120的xml请求如下： <rpc message-id ="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0"> <edit-config> <target> <running/> </target> <config> <top xmlns="http://www.h3c.com/netconf/config:1.0"> <Syslog xmlns="http://www.h3c.com/netconf/config:1.0" xc:operation="merge"> <LogBuffer> <BufferSize>120</BufferSize> </LogBuffer> </Syslog> </top> </config> </edit-config> </rpc>
edit-config: create	创建指定对象。create操作必须指定配置对象。create操作的XML数据格式和merge类似，只是operation属性需要指定为“create” • 如果当前配置表支持创建对象，且当前对象不存在，则先创建配置对象，再创建指定的配置 • 如果配置对象下对应的配置项已经存在，则返回 data-exists 错误	<rpc message-id ="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0"> <edit-config> <target> <running/> </target> <config> <top xmlns="http://www.h3c.com/netconf/config:1.0"> <ACL xmlns="http://www.h3c.com/netconf/config:1.0" xc:operation="create"> <Groups> <Group> <GroupType>1</GroupType> <GroupID>2222</GroupID> <Step>2</Step> <Name>abc</Name> <Description>ABC-la-la-la</Description> </Group> </Groups> </ACL> </top> </config> </edit-config> </rpc>

操作	说明	XML 格式样例
edit-config: replace	• 如果指定的对象存在，则替换指定对象的配置为当前配置 • 如果指定对象不存在，则不进行 replace 操作，返回 invalid-value 错误，提示用户配置对象不存在	同edit-config: merge，把merge修改为replace即可
edit-config: remove	删除指定配置 • 当指定的删除对象中只有表索引时，则删除此配置指定对象的所有配置，同时删除指定对象 • 当指定的删除对象中不仅仅有表索引还存在配置项时，则删除此对象下面的指定配置 • 如果系统中指定对象不存在，或者 XML 消息未指定对象，则直接返回成功	<rpc message-id ="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0"> <edit-config> <target> <running/> </target> <config> <top xmlns="http://www.h3c.com/netconf/config:1.0"> <ACL xmlns="http://www.h3c.com/netconf/config:1.0" xc:operation="remove"> <Groups> <Group> <GroupType>1</GroupType> <GroupID>2222</GroupID> </Group> </Groups> </ACL> </top> </config> </edit-config> </rpc>]]>]]>
edit-config: delete	删除指定配置 • 当指定的删除对象中只有表索引时，则删除此配置指定对象的所有配置，同时删除指定对象 • 当指定的删除对象中不仅仅有表索引还存在配置项时，则删除此对象下面的指定配置 • 如果系统中指定对象不存在，则直接返回不存在的错误消息	同上，把merge修改为delete即可

操作	说明	XML 格式样例
edit-config *默认操作选项*	edit-config操作用于修改当前系统配置。NETCONF定制了四种修改配置的方式：merge、create、delete和replace。当XML消息中未指定修改配置方式的时候，则使用默认操作做为当前指令的操作方式 默认操作的缺省值是merge，在XML消息中可以通过<default-operation>节点来设置默认操作，取值为： • merge：当配置方式和默认操作方式均未指定时，使用该方式 • replace：当配置方式未指定，默认操作指定为 replace 的时候，edit-config 操作会默认为 replace 操作 • none：当配置方式未指定，默认操作指定为 none 的时候，edit-config 操作会默认为 none 操作。none 操作主要用来检查，下发为 none 操作的配置仅仅做 Schema 校验，不进行真正的配置下发。语法检查通过，就返回 ok 成功，否则失败	下发一个空的操作，该操作仅仅验证格式，并不真正下发给系统，xml请求如下： <rpc message-id ="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <edit-config> <target> <running/> </target> <default-operation>none</default-operation> <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0"> <top xmlns="http://www.h3c.com/netconf/config:1.0"> <Ifmgr> <Interfaces> <Interface> <IfIndex>262</IfIndex> <Description>222222</Description> </Interface> </Interfaces> </Ifmgr> </top> </config> </edit-config> </rpc>

操作	说明	XML 格式样例
edit-config *默认错误处理选项*	edit-config将指定的配置设置到系统上，完成配置设置的操作。在执行edit-config的过程中，如果遇到一个实例配置出错，默认情况下会直接返回错误，但是为了使我们的应用更加灵活，edit-config为我们提供了错误选项，通过错误选项取值的不同，在发生错误的时候进行不同的处理操作 <error-option>节点用于设置一个实例配置出错后，后续实例配置的处理方式，缺省值为stop-on-error，全部取值为： • stop-on-error：停止处理，返回错误。此选项为默认选项 • continue-on-error：继续处理，但是报告错误 • rollback-on-error：停止并回滚配置。此选项设备暂不支持	下发两个接口的配置，当下发第一个接口的配置发生错误时，继续进行下一个接口配置的下发，XML请求如下： <rpc message-id ="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <edit-config> <target> <running/> </target> <error-option>continue-on-error</error-option> <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0"> <top xmlns="http://www.h3c.com/netconf/config:1.0"> <Ifmgr xc:operation="merge"> <Interfaces> <Interface> <IfIndex>262</IfIndex> <Description>222</Description> <ConfigSpeed>1000000</ConfigSpeed> <ConfigDuplex>1</ConfigDuplex> </Interface> <Interface> <IfIndex>263</IfIndex> <Description>333</Description> <ConfigSpeed>1000000</ConfigSpeed> <ConfigDuplex>1</ConfigDuplex> </Interface> </Interfaces> </Ifmgr> </top> </config> </edit-config> </rpc>

操作	说明	XML 格式样例
edit-config *测试处理选项*	在真正执行edit-config操作时，可指定一个测试选项，使用<test-option>节点来决定当前配置是否真正下发。该节点的缺省值为test-then-set，全部取值为： • test-then-set：如果没有错误则将配置设置到系统 • set：将配置设置到系统 • test-only：只测试，并不下发配置到系统。语法检查通过，就返回 ok 成功，否则失败	下发一个接口的配置，仅测试，XML请求如下： <rpc message-id ="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <edit-config> <target> <running/> </target> <test-option>test-only</test-option> <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0"> <top xmlns="http://www.h3c.com/netconf/config:1.0"> <Ifmgr xc:operation="merge"> <Interfaces> <Interface> <IfIndex>262</IfIndex> <Description>222</Description> <ConfigSpeed>1000000</ConfigSpeed> <ConfigDuplex>1</ConfigDuplex> </Interface> </Interfaces> </Ifmgr> </top> </config> </edit-config> </rpc>
action	下发非配置数据的设置动作，比如，reset操作	清除全部接口的统计信息，XML请求如下： <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <action> <top xmlns="http://www.h3c.com/netconf/action:1.0"> <Ifmgr> <ClearAllIfStatistics> <Clear> </Clear> </ClearAllIfStatistics> </Ifmgr> </top> </action> </rpc>

操作	说明	XML 格式样例
lock	锁保护的是配置数据，即edit-config中可以指定的那些模块的配置数据，其它操作不受锁的限制 NETCONF锁仅仅保护NETCONF会话，不保护SNMP等其它请求下发的配置	禁止其他NETCONF会话修改设备的当前配置，XML请求如下： <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <lock> <target> <running/> </target> </lock> </rpc>
unlock	取消锁保护 当会话结束时锁也会被自动释放	取消锁保护，允许其他NETCONF会话修改设备的当前配置： <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <unlock> <target> <running/> </target> </unlock> </rpc>
get-sessions	获取当前系统中所有NETCONF会话的信息（不能指定sessions-ID）	获取当前系统中所有NETCONF会话的信息： <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <get-sessions/> </rpc>
close-session	关闭当前NETCONF会话，并释放锁和这个session用到的内部资源（如内存等），退出XML视图	关闭当前NETCONF会话，并释放锁和这个session用到的内部资源（如内存等），退出XML视图 <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <close-session /> </rpc>
kill-session	关闭其他NETCONF会话，不支持关闭用户自己的NETCONF会话	关闭session-id为1的NETCONF会话： <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <kill-session> <session-id>1</session-id> </kill-session> </rpc>

操作	说明	XML 格式样例
CLI	执行命令行的命令。请求消息将命令行语句封装在<CLI>标签中，命令行输出信息被封装在<CLI>标签中返回 CLI支持Configuration和Exec两种方式： • Execution：在用户视图下执行 • Configuration：在系统视图下执行 对于其它视图下命令，则需要在Configuration下先指定进子视图的命令，再指定配置命令	在系统视图下执行**display this**命令： <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <CLI> <Configuration>display this</Configuration> </CLI> </rpc>
save	保存系统运行配置。save操作可以使用子元素<file>来指定保存的配置文件名称。如果不指定配置文件名称，则将当前运行配置保存到主用下次启动配置文件中。	将设备当前配置保存到文件test.cfg中： <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <save> <file>test.cfg</file> </save> </rpc>
load	配置加载。<load>操作执行后，指定文件中的配置会被合并到设备的当前配置中	将文件a1.cfg中的配置合并到设备的当前配置中： <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <load> <file>a1.cfg</file> </load> </rpc>
rollback	配置回滚。<rollback>操作必须使用子元素<file>来指定需要回滚的配置文件名称。<rollback>操作执行后，当前系统运行配置会被完全替换为指定文件中所描述的配置	将设备当前配置回退到文件1A.cfg中配置的状态： <rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"> <rollback> <file>1A.cfg</file> </rollback> </rpc>

目 录

1 CWMP

1.1 CWMP简介

CWMP（CPE 广域网管理协议，CPE WAN Management Protocol）是由 DSL（Digital Subscriber's Line，数字用户线路）论坛发起开发的技术规范之一，编号为 TR-069，所以又被称为 TR-069 协议。它提供了对下一代网络中家庭网络设备进行管理配置的通用框架、消息规范、管理方法和数据模型。

CWMP 主要应用于 DSL 接入网络环境。在 DSL 接入网络中，由于用户设备数量繁多、部署分散，通常位于用户侧，不易进行设备的管理和维护，CWMP 提出通过 ACS（Auto-Configuration Server，自动配置服务器）对 CPE（Customer Premises Equipment，用户侧设备）进行远程集中管理，解决 CPE 设备的管理困难，节约维护成本，提高问题解决效率。

大型数据中心的网络环境与 DSL 环境具有类似的特征，即接入设备数量众多，而且随着规模的扩张，会经常出现在网络的某个层级大量增加新设备的情况。这些新设备的业务功能基本相同，如果能够利用 CWMP 协议的特点进行远程集中管理和配置，便可以快速完成部署，并且后期的维护和管理也将变得更加便捷。

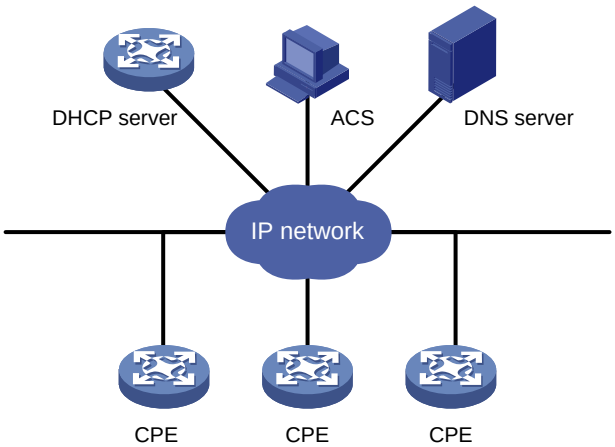
设备提供了对 CWMP 协议的支持，在空配置接入网络时可以作为 CPE 设备，自动从 ACS 下载配置文件。相对于传统的手工配置方式，利用 CWMP 进行远程统一配置具有以下优点：

- 对于业务相同的新设备统一下发配置，减少重复工作，提高部署效率
- 设备开箱后即可直接接入网络，不需要工程师现场配置，降低人力成本
- 配置文件预先生成，降低人工配置错误几率

1.1.1 CWMP网络框架

CWMP网络的基本框架如 图 1-1 所示：

图1-1 CWMP 网络基本框架示意图



CWMP 网络元素主要有：

- ACS：自动配置服务器，网络中的管理设备。用于向 CPE 设备下发配置并提供 CPE 设备的管理服务。

- CPE：用户端设备，网络中的被管理设备。可以向 ACS 上报自身信息并获取相应的配置。
- DNS server：域名服务器。CWMP 协议规定 ACS 和 CPE 使用 URL（Uniform Resource Locator，统一资源定位符）地址来互相识别和访问，DNS 用于帮助解析 URL 参数。
- DHCP server：动态主机配置协议服务器。给 CPE 分配 IP 地址，使用 DHCP 报文中的 option 字段给 CPE 配置参数。该网络元素可以根据实际网络需要选择部署，一般用于设备第一次上电接入 CWMP 环境。

1.1.2 CWMP基本功能

1. 通过ACS监控CPE的状态和性能

ACS 可以监控与其相连的 CPE 的各种参数。由于不同的 CPE 具有不同的性能，可执行的功能也各异，因此 ACS 必须能识别不同类型 CPE 的性能，并监控到 CPE 的当前配置以及配置的变更。CWMP 还允许网络管理人员自定义监控参数并通过 ACS 获取这些参数，以便了解 CPE 的状态和统计信息。

ACS 能够监控的状态和性能有：厂商名称（Manufacturer）、厂商标识 OUI（ManufacturerOUI）、序列号（SerialNumber）、硬件版本号（HardwareVersion）、软件版本号（SoftwareVersion）、设备状态（DeviceStatus）、启动时间（UpTime）、配置文件、ACS 地址、ACS 用户名、ACS 密码、Inform 报文自动发送使能标志、Inform 报文周期发送时间间隔、Inform 报文定期发送日期、CPE 地址、CPE 用户名、CPE 密码等。

2. 通过ACS向CPE自动下发配置文件

为了便于对提供相同业务功能的新设备进行快速配置，网络管理员可以在 ACS 上创建针对该类设备的配置文件。当 CPE 设备与 ACS 建立连接后，ACS 可以判断 CPE 设备所属类别，并将对应该类设备的配置文件下发给 CPE 设备，从而可以使相同类型的大量 CPE 设备获得相同的业务配置。目前 ACS 可以通过产品型号和序列号等将 CPE 划分为不同的类别。

ACS 向 CPE 下发配置文件时，可以通过以下两种方式进行：

- 部署为启动配置：ACS 向 CPE 发送配置文件，覆盖 CPE 本地的缺省配置文件，当 CPE 重启之后，便可以使用新的配置运行。
- 部署为运行配置：ACS 将配置内容直接发给 CPE，并写入到 CPE 的当前配置中，配置内容即时生效，但是需要再执行保存配置的操作，以保证重启后配置不会丢失。

3. 通过ACS管理CPE设备系统文件

网络管理员可以将 CPE 设备的应用程序文件和配置文件等重要文件保存在 ACS 上，当 ACS 发现某个文件的版本有更新，将会通知 CPE 进行下载。CPE 收到 ACS 的下载请求后，能够根据 ACS 报文中提供的下载地址和文件名，自动到指定的文件服务器下载文件。下载完成后，对下载文件的合法性做相应的检查，并将下载结果（成功或失败）反馈给 ACS。目前，设备支持应用程序文件和配置文件的下载，不支持以数字签名的方式进行文件下载。

同样，为了实现对重要数据的备份，CPE 将根据 ACS 的要求将当前的配置文件和日志文件上传到指定的服务器。

1.1.3 CWMP实现机制

1. ACS和CPE的自动连接

CPE 在第一次上电启动时，建议在网络中部署 DCHP 服务器，CPE 会通过 DHCP 自动获取 IP 地址，DHCP 服务器在分配 IP 地址的同时，将向 CPE 通告以下内容：

- ACS 的 URL 地址（通过 option 43 选项分配）
- 连接 ACS 所需要的用户名和密码（通过 option 43 选项分配）
- DNS 服务器地址（直接分配）

CPE 设备得到以上信息后，通过 DNS 服务器解析出 ACS 的 IP 地址，向 ACS 发起连接请求，如果用户名和密码验证通过，CPE 和 ACS 之间将成功建立连接。

如果当前会话没有结束，但是连接异常中断，CPE 将自动尝试重新连接，直至重新连接的次数达到上限。

在运行过程中，CPE 设备还可以根据配置周期性或者定时向 ACS 服务器发起连接。

2. ACS向CPE下发配置项参数

CPE与ACS建立连接之后，ACS可以自动下发一些配置给CPE，完成对CPE的自动配置。CPE可以从ACS获取的自动配置项参数如 表 1-1 所示：

表1-1 CPE 可以从 ACS 获取的配置项

配置项	用途
配置文件（ConfigFile）	用于更新CPE的本地配置文件，ACS可以使用文件形式和当前配置形式向CPE下发配置文件
ACS地址（URL）	更新CPE记录的ACS地址，可用于主备ACS服务器之间的切换
ACS用户名（Username）	当ACS上连接用户名和密码发生变更时，可以自动同步到CPE设备，也可用于主备ACS服务器切换时向CPE通告备用ACS服务器的验证信息
ACS密码（Password）	
Inform报文自动发送使能标志（PeriodicInformEnable）	开启CPE设备发送Inform报文的功能
Inform报文周期发送时间间隔（PeriodicInformInterval）	配置CPE周期性向ACS发送Inform报文建立连接，用于定期查询更新和信息备份
Inform报文定期发送日期（PeriodicInformTime）	配置CPE在指定时间点向ACS发送Inform报文建立连接，用于在指定时间查询更新和信息备份
CPE用户名（ConnectionRequestUsername）	配置CPE在接受ACS发起的连接时所需要的验证信息
CPE密码（ConnectionRequestPassword）	

3. ACS对CPE的管理方式

ACS 对 CPE 的管理和监控是通过一系列的操作来实现的，这些操作在 CWMP 协议里称为 RPC（Remote Procedure Call，远程调用）方法。主要方法的描述如下：

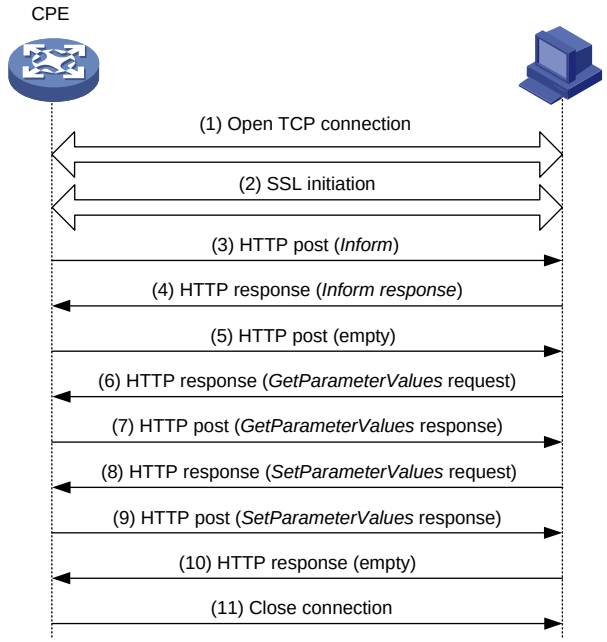
- Get：ACS 使用该方法可以获取 CPE 上参数的值。
- Set：ACS 使用该方法可以设置 CPE 上参数的值。

- Inform：当 CPE 与 ACS 建立连接时、各应用模块具有主动通知属性的配置发生改变时和 CPE 周期性发送本地信息到 ACS 时，CPE 都要通过该方法向 ACS 发起通告信息。ACS 通过跟 CPE 报文的交互可以设置哪些具有主动通知属性。
- Download：为了保证 CPE 端硬件的升级以及厂商配置文件的自动下载，ACS 使用该方法可以要求 CPE 到指定的 URL 下载指定的文件来更新 CPE 的本地文件。
- Upload：为了方便 ACS 对 CPE 端的管理，ACS 使用该方法可以要求 CPE 将指定的文件上传到 ACS 指定的位置。
- Reboot：当 CPE 故障或者需要软件升级的时候，ACS 使用该方法可以对 CPE 进行远程重启。

4. 主备ACS切换时的操作

下面以一个具体的例子，结合 CWMP 方法来描述 CWMP 具体实现。场景如下：区域内有主、备两台 ACS，主 ACS 系统升级，需要重启。为了连续监控，主 ACS 需要将区域内的 CPE 都连接到备用 ACS 上，处理流程如下：

图1-2 CWMP 消息交互举例



(1) 建立 TCP 连接。

(2) SSL 初始化，建立安全机制。

(3) CPE 发送 Inform 报文，开始建立 CWMP 连接。Inform 报文使用 Eventcode 字段描述发送 Inform 报文的原因，该举例为“6 CONNECTION REQUEST”，表示 ACS 要求建立连接。

(4) 如果 CPE 通过 ACS 的认证，ACS 将返回 Inform 响应报文，连接建立。

(5) 如果 CPE 没有别的请求，就会发送一个空报文，以满足 HTTP/HTTPS 报文请求/响应报文交互规则（CWMP 是基于 HTTP/HTTPS 协议的，CWMP 报文作为 HTTP/HTTPS 报文的数据部分封装在 HTTP/HTTPS 报文中）。

(6) ACS 查询 CPE 上设置的 ACS URL 的值。

(7) CPE 把获取到的 ACS URL 的值回复给 ACS。

(8) ACS 发现 CPE 的 ACS URL 是本机 URL 的值，于是发起 Set 请求，要求将 CPE 的 ACS URL 设置为备用 ACS 的 URL 的值。

(9) 设置成功，CPE 发送响应报文。

(10) ACS 发送空报文通知 CPE 没有别的请求。

(11) CPE 关闭连接。

之后，CPE 将向备用 ACS 发起连接。

1.2 CWMP配置任务简介

ACS 属性可以通过 ACS、DHCP 和命令行三种方式来配置，CPE 的部分属性可以通过 ACS 和命令行方式配置。当某参数支持多种配置方式时，DHCP 配置的优先级最低，ACS 配置和命令行配置的优先级相同。高优先级配置方式可以修改低优先级配置方式配置的参数，配置方式优先级相同时，以最新的配置为准。

1.2.1 通过DHCP配置

在 CWMP 网络中，DHCP 服务器主要用于向 CPE 通告 ACS 的位置和验证信息，因此 DHCP 服务器上的配置主要包含以下内容：

- 配置地址池，为 CPE 设备分配 IP 地址
- 配置 DNS 服务器
- 配置 option 43 选项，向 CPE 通告 ACS 信息

ACS 属性可以通过在 DHCP server 上配置 option 43 参数来实现。当 CPE 访问 DHCP server 时，DHCP server 会将 ACS 参数发送给 CPE。这里主要介绍 option 43 选项的配置方法，关于地址池和 DNS 服务器的配置请参见“三层技术-IP 业务配置指导”中的“域名解析”。

当使用 H3C 设备作为 DHCP server 时，可以使用命令行配置 ACS 参数，命令格式为：**option 43 hex 01***length URL username password*。

- *length*：表示关键字 **option 43 hex 01** 后面参数的总长度，用十六进制数表示。
- *URL*：ACS 的地址。
- *username*：ACS 的用户名。
- *password*：ACS 的密码。

ACS 的 URL、用户名和密码参数的格式必须为字符对应的 ASCII 码的十六进制值格式。比如，要将 ACS 地址配置为 http://169.254.76.31:7547/acs（http://对应的 ASCII 码的十六进制值为 68 74 74 70 3A 2F 2F，169.254.76.31 对应的 ASCII 码的十六进制值为 31 36 39 2E 32 35 34 2E 37 36 2E 33 31，:7547 对应的 ASCII 码的十六进制值为 3A 37 35 34 37，/acs 对应的 ASCII 码的十六进制值为 2F 61 63 73）、用户名配置为 1234（对应的 ASCII 码的十六进制值为 31 32 33 34）、密码配置为 5678（对应的 ASCII 码的十六进制值为 35 36 37 38），空格对应的 ASCII 码的十六进制值为 20，（*URL*+1 个空格+*username*+1 个空格+*password*）一共为 39 个字符（39 对应的十六进制为 27），所以 *length* 值为 0x27，可使用以下配置步骤：

```
<Sysname> system-view
[Sysname] dhcp server ip-pool 0
[Sysname-dhcp-pool-0] option 43 hex
0127687474703A2F2F3136392E3235342E37362E33313A373534372F6163732031323334203536373 8
```

有关 DHCP、option 43 参数以及 **option** 命令的详细介绍，请参见“三层技术-IP 业务配置指导”中的“DHCP”。

1.2.2 通过ACS配置

由ACS远程管理，对CPE进行自动配置。可配置的主要参数请参见 1.1.3 2.。ACS服务器上的配置请参见您所选ACS服务器的软件使用说明。

1.2.3 通过命令行配置

即通过命令行手工指定CWMP参数，可配置的内容如 表 1-2 所示。

表1-2 CWMP 配置任务简介

配置任务		说明	详细配置
使能CWMP功能		必选	1.3
配置ACS属性	配置CPE连接到ACS的URL值	必选	1.4.1
	配置CPE连接到ACS的用户名和密码	可选	1.4.2
配置CPE设备属性	配置CPE的用户名和密码	可选	1.5.1
	配置CWMP接口	可选	1.5.2
	配置发送Inform报文属性	可选	1.5.3
	配置CPE自动重新连接的次数	可选	1.5.4
	配置CPE的NAT穿越功能	可选	1.5.5
	配置CPE的业务代码信息	可选	1.5.6
	配置CPE无数据传输的超时时间	可选	1.5.7
	配置CWMP引用的SSL客户端策略	可选	1.5.8

1.3 使能CWMP功能

使能 CWMP 后，CWMP 的其它配置才能生效。

表1-3 使能 CWMP 功能

操作	命令	说明
进入系统视图	**system-view**	-
进入CWMP视图	**cwmp**	-
使能CWMP功能	**cwmp enable**	缺省情况下，CWMP功能处于关闭状态

1.4 配置ACS属性

ACS 属性包括 ACS 的 URL、用户名和密码。当 CPE 发起连接请求时，连接请求报文里会携带 CPE 连接到 ACS 的 URL、用户名和密码。当 ACS 收到该报文后，如果这些参数的值和本地配置的值一致，则验证成功，允许建立连接；如果不一致，则验证失败，禁止建立连接。

1.4.1 配置CPE连接到ACS的URL值

表1-4 配置 CPE 连接到 ACS 的 URL 值

操作		命令	说明
进入系统视图		**system-view**	-
进入CWMP视图		**cwmp**	-
配置ACS的URL值	配置CPE连接到ACS的URL值	**cwmp acs url** *url*	二者选其一 一个CPE只能配置一个连接到ACS的URL和缺省URL，多次配置不同URL时，最新的配置生效 当用户没有为ACS配置URL地址，也没有通过DHCP服务器获取到ACS的URL地址时，设备会尝试和ACS的缺省URL建立CWMP连接 缺省情况下，没有配置CPE连接到ACS的URL和缺省URL
	配置CPE连接到ACS的缺省URL值	**cwmp acs default url** *url*	

1.4.2 配置CPE连接到ACS的用户名和密码

表1-5 配置 CPE 连接到 ACS 的用户名和密码

操作		命令	说明
进入系统视图		**system-view**	-
进入CWMP视图		**cwmp**	-
配置连接到ACS的用户名	配置CPE连接到ACS的用户名	**cwmp acs username** *username*	二者选其一 缺省情况下，没有配置CPE连接到ACS的用户名和缺省用户名
	配置CPE连接到ACS的缺省用户名	**cwmp acs default username** *username*	
（可选）配置连接到ACS的密码	配置CPE连接到ACS的密码	**cwmp acs password** { **cipher** | **simple** } *password*	二者选其一 可以只使用用户名进行认证，不使用密码认证 缺省情况下，没有配置CPE连接到ACS的密码和缺省密码
	配置CPE连接到ACS的缺省密码	**cwmp acs default password** { **cipher** | **simple** } *password*	

1.5 配置CPE属性

CPE 的用户名和密码，用于 CPE 对 ACS 的合法性进行验证。当连接由 ACS 发起时，会话请求报文里会携带 CPE 用户名和密码。设备收到该报文后，会与本地设置的 CPE 用户名和密码比较，如果相同则通过认证，进入连接建立的下一阶段，否则，认证失败，退出连接建立过程。

1.5.1 配置CPE的用户名和密码

表1-6 配置 CPE 用户名和密码

操作	命令	说明
进入系统视图	**system-view**	-
进入CWMP视图	**cwmp**	-
配置ACS连接到CPE的认证用户名	**cwmp cpe username** *username*	缺省情况下，没有配置ACS连接到CPE的认证用户名
（可选）配置ACS连接到CPE的认证密码	**cwmp cpe password { cipher | simple }** *password*	可以只使用用户名进行认证，不使用密码认证 缺省情况下，没有配置ACS连接到CPE的认证密码

1.5.2 配置CWMP连接接口

CWMP 连接接口指的是 CPE 上用于连接 ACS 的接口。CPE 会在 Inform 报文中携带 CWMP 连接接口的 IP 地址，要求 ACS 通过此 IP 地址和自己建立连接；相应的，ACS 会向该 IP 地址回复 Inform 响应报文。

缺省情况下，系统会会采用一定的机制去获取一个 CWMP 连接接口，但如果获取的 CWMP 连接接口不是 CPE 和 ACS 相连的接口时，就会导致 CWMP 连接建立失败。因此，在这种情况下需要手工指定 CWMP 连接接口。

表1-7 配置 CWMP 连接接口

操作	命令	说明
进入系统视图	**system-view**	-
进入CWMP视图	**cwmp**	-
设置CPE上用于连接ACS的接口	**cwmp cpe connect interface** *interface-type interface-number*	缺省情况下，系统自动选择第一个存在通路的接口

1.5.3 配置发送Inform报文

CPE 与 ACS 之间连接的建立过程需要发送 Inform 报文。通过设置 Inform 报文发送参数，可以触发 CPE 向 ACS 自动发起连接。

1. 配置周期性发送Inform报文

表1-8 配置周期性发送 Inform 报文

操作	命令	说明
进入系统视图	**system-view**	-
进入CWMP视图	**cwmp**	-
使能CPE周期发送Inform报文功能	**cwmp cpe inform interval enable**	缺省情况下，CPE周期发送Inform报文功能处于关闭状态
配置CPE发送Inform报文的周期	**cwmp cpe inform interval** *seconds*	缺省情况下，CPE每隔600秒发送一次Inform报文

2. 配置定时发送Inform报文

表1-9 配置定时发送 Inform 报文

操作	命令	说明
进入系统视图	**system-view**	-
进入CWMP视图	**cwmp**	-
配置CPE在指定时刻发送一次Inform报文	**cwmp cpe inform time** *time*	缺省情况下，没有配置CPE定时发送Inform报文的时间

1.5.4 配置CPE自动重新连接的次数

当 CPE 向 ACS 请求建立连接失败，或者在会话过程中连接异常中止（CPE 没有收到表示会话正常结束的报文）时，设备可以自动重新发起连接。

表1-10 配置 CPE 自动重新连接的次数

操作	命令	说明
进入系统视图	**system-view**	-
进入CWMP视图	**cwmp**	-
配置当创建连接失败时自动重新连接的次数	**cwmp cpe connect retry** *times*	缺省情况下，自动重新连接的次数为无限次，即设备会一直按照一定周期给ACS发送连接请求

1.5.5 配置CPE的NAT穿越功能

无论 CPE 与 ACS 之间是否存在 NAT 网关，CPE 的主动连接请求都能到达 ACS。而当 CPE 与 ACS 之间存在 NAT 网关时，ACS 主动发起的连接请求不能到达 CPE。此时，可以在设备上开启 NAT 穿越功能，使 ACS 的请求可以穿越网关。本特性的实现遵循 RFC 3489 定义的 STUN（Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)，NAT 的 UDP 简单穿越）。有关 NAT 的详细描述，请参见“三层技术-IP 业务配置指导”中的“NAT”。

表1-11 配置 CPE 的 NAT 穿越功能

操作	命令	说明
进入系统视图	**system-view**	-
进入CWMP视图	**cwmp**	-
使能CPE的NAT穿越功能	**cwmp cpe stun enable**	缺省情况下，CPE的NAT穿越功能处于关闭状态

1.5.6 配置CPE的业务代码信息

当 CPE 与 ACS 之间建立连接时，CPE 需要在 Inform 报文中携带 provision-code 信息，ACS 根据此信息可以识别设备定制的业务以及相应的参数，以便更好地管理 CPE 设备。

表1-12 配置 CPE 业务代码信息

操作	命令	说明
进入系统视图	**system-view**	-
进入CWMP视图	**cwmp**	-
配置CPE的业务代码	**cwmp cpe provision-code** *provision-code*	缺省情况下，CPE的业务代码为PROVISIONINGCODE

1.5.7 配置CPE无数据传输超时的时间

无数据传输超时时间主要用于以下两种情况：

- 在连接建立过程中，CPE 向 ACS 发送连接请求，但是经过无数据传输超时时间还没有收到响应报文，CPE 将认为连接失败。
- 连接建立后，如果 CPE 与 ACS 在无数据传输超时时间内没有报文交互，CPE 将认为连接失效，并断开连接。

表1-13 配置 CPE 无数据传输超时的时间

操作	命令	说明
进入系统视图	**system-view**	-
进入CWMP视图	**cwmp**	-
配置CPE无数据传输超时的时间	**cwmp cpe wait timeout** *seconds*	缺省情况下，无数据传输超时的时间为30秒

1.5.8 配置CWMP引用的SSL客户端策略

CWMP 是基于 HTTP/HTTPS 协议的，CWMP 报文作为 HTTP/HTTPS 报文的数据部分封装在 HTTP/HTTPS 报文中。如果 ACS 的 URL 以“http://”开头，则使用 HTTP 协议；如果 ACS 的 URL

以“https://”开头，则使用 HTTPS 协议。使用 HTTPS 协议时，ACS 作为 HTTPS 服务器端，CPE 作为 HTTPS 客户端。

在 CPE 上需要配置 CWMP 引用的 SSL 客户端策略，以便从该策略中获取 SSL 客户端运行时需要的参数（如加密算法、SSL 协议版本等）。关于 SSL 客户端策略的详细介绍和配置请参见“安全配置指导”中的“SSL”。

表1-14 配置 CWMP 引用的 SSL 客户端策略

操作	命令	说明
进入系统视图	**system-view**	-
进入CWMP视图	**cwmp**	-
配置CWMP引用的SSL客户端策略	**ssl client-policy** *policy-name*	缺省情况下，CWMP没有引用SSL客户端策略

1.6 CWMP显示和维护

在完成上述配置后，在任意视图下执行 **display** 命令可以显示配置后 CWMP 的运行情况，通过查看显示信息验证配置的效果。

表1-15 cwmp 显示和维护

操作	命令
显示CWMP的当前配置信息	**display cwmp configuration**
显示CWMP的当前状态信息	**display cwmp status**

1.7 CWMP典型配置举例

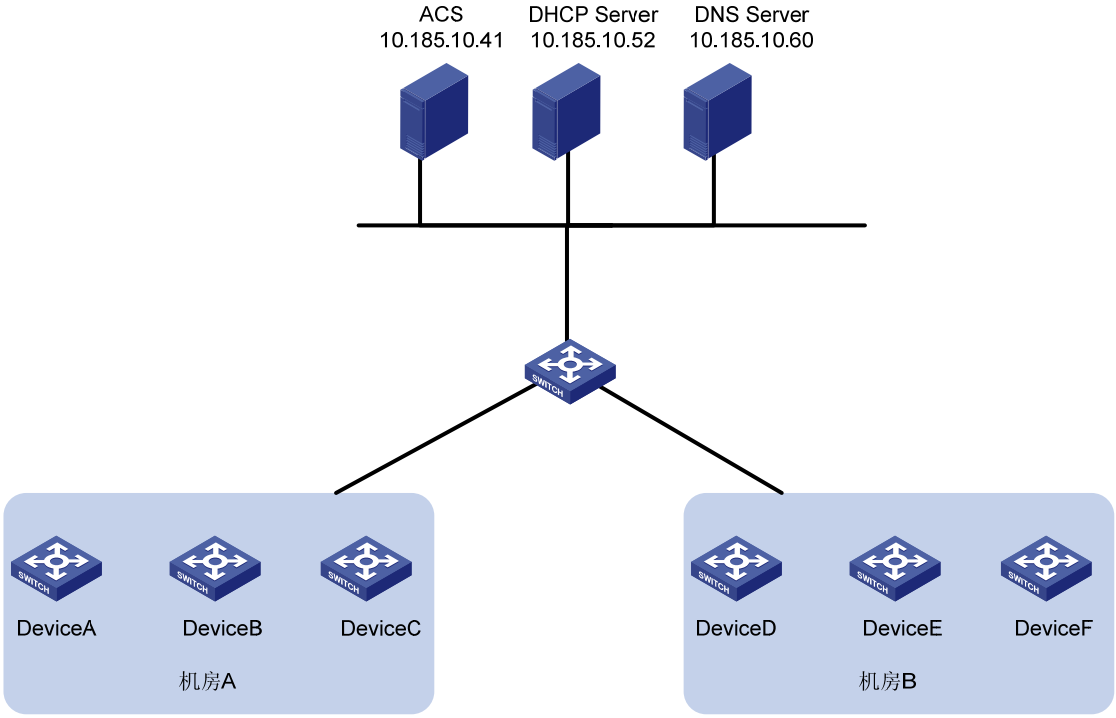
说明

本举例使用安装了 H3C iMC BIMS 软件的服务器作为 ACS。随着软件版本的更新，BIMS 的功能和界面可能会有变化，如您所使用的软件界面与本例中不同，请参阅对应您使用版本的软件用户手册进行配置。

1.7.1 组网需求

某数据中心有两个机房（A 和 B）需要部署大量的设备，目前网络中已存在 ACS 服务器/DHCP 服务器/DNS 服务器，为提高部署效率，要求利用 CWMP 功能为两个机房中的 CPE 设备分别自动下发不同的配置文件。下面以每个机房内的三台设备为例介绍 CWMP 功能的配置方法。

图1-3 CWMP 典型配置案例组网图



其中部署到两个机房的设备及序列号如 表 1-16 所示。

表1-16 部署到机房的设备列表

机房	设备	序列号
A	DeviceA	210231A95YH10C0000045
	DeviceB	210235AOLNH12000010
	DeviceC	210235AOLNH12000015
B	DeviceD	210235AOLNH12000017
	DeviceE	210235AOLNH12000020
	DeviceF	210235AOLNH12000022

网络管理员已经为机房 A 和机房 B 的设备分别创建了配置文件 configure1.cfg 和 configure2.cfg，ACS 服务器的访问用户名和密码分别为“admin”和“12345”，URL 地址为 http://10.185.10.41:8080/acs。

1.7.2 配置步骤

1. ACS服务器上的配置

(1) 登录 ACS 服务器（iMC）。

在 ACS 服务器上直接运行 Web 浏览器，在地址栏中输入 http://10.185.10.41:8080/imc（即 ACS 服务器的 IP 地址和端口号），并输入用户名和密码成功登录 iMC 界面。

(2) 在 ACS 服务器上配置 CPE 认证用户。

在导航栏中选择“业务>分支网点管理>CPE认证用户”，进入 图 1-4 所示页面。

图1-4 CPE 认证用户管理页面

单击<增加>按钮，进入 图 1-5 所示页面。

图1-5 增加 CPE 认证用户页面

将用户名配置为 admin，将密码和确认密码都配置为 12345，单击<确定>按钮完成增加操作。

(3) 在 ACS 上增加 CPE 设备分组和类型，这里以增加 DeviceA 设备到“DB_1”组的“Device_A”类型为例。

在导航栏中选择“业务>分支网点管理>CPE分组”，单击<增加>按钮，进入 图 1-6 所示页面。

图1-6 增加 CPE 分组页面

设置分组名称后，单击<确定>按钮完成增加操作。

在导航栏中选择“业务>分支网点管理>CPE类型”，单击<增加>按钮，进入 图 1-7 所示页面。

图1-7 增加 CPE 类型页面

设置 CPE 类型名称后，单击<确定>按钮完成增加操作。

在导航栏中选择“业务>分支网点管理>增加CPE”，进入 图 1-8 所示页面。

图1-8 增加 CPE 页面

设置设备名称及相关信息，并选择设备类型和设备分组后，单击<确定>按钮完成增加操作。

图1-9 增加设备成功页面

重复以上步骤，将 DeviceB 和 DeviceC 设备的信息输入，完成机房 A 中的设备添加任务。

(4) 为 ACS 配置系统参数

在导航栏中选择“业务>分支网点管理>系统参数”，进入 图 1-10 所示页面。

图1-10 系统参数页面

设置完系统参数后，单击<确定>按钮完成设置操作。

(5) 在 ACS 上配置模板库和 CPE 软件库类型。

在导航栏中选择“业务>分支网点管理>配置管理>配置模板库”，进入 图 1-11 所示页面。

图1-11 配置模板库页面

单击<导入>按钮，进入 图 1-12 所示页面。

图1-12 导入配置模板页面

选择文件和模板类型后，显示导入配置模板成功页面。

图1-13 导入配置模板库成功页面

在导航栏中选择“业务>分支网点管理>配置管理>CPE软件库”，进入 图 1-14 所示页面。

图1-14 CPE 软件库配置页面

单击<导入>按钮，进入 图 1-15 所示页面。

图1-15 导入 CPE 软件库页面

(6) 选择 CPE 对应的软件版本，在 ACS 上将配置文件与不同组别的 CPE 设备绑定，实现自动部署。

在导航栏中选择“业务>分支网点管理>配置管理>部署向导”，进入 图 1-16 所示页面。

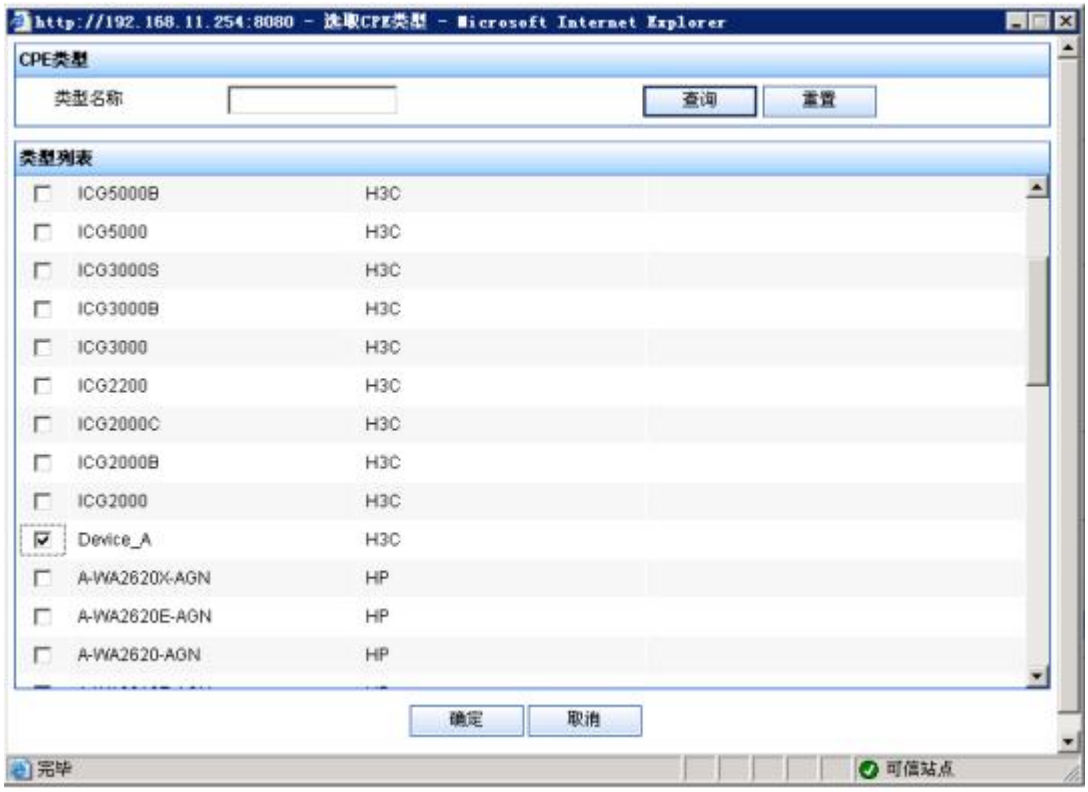
图1-16 部署向导页面

在导航栏中选择“业务>分支网点管理>配置管理>部署向导>自动部署CPE配置”，并将其选择部署为“启动配置”，进入 图 1-17 所示页面。

图1-17 自动部署设备配置页面

单击[选取类型]按钮，进入设备类型选择页面，选取之前创建的 Device_A 类型，单击<确定>按钮完成选择操作。

图1-18 选取设备类型页面



完成选择页面后将返回自动部署设备配置页面，单击<确定>按钮完成部署任务的创建。

图1-19 部署任务创建成功页面

对于机房 B 中的三台设备，配置步骤基本类似，主要区别如下：

- 需要为机房 B 的设备创建新的设备类型，例如“Device_B”。
- 在增加设备时，将机房 B 中的三台设备加入到 Device_B 类型中。
- 创建部署配置的任务时，将对应机房 B 的配置文件与 Device_B 进行关联。

2. DHCP服务器上的配置

(1) 配置地址池，为 CPE 设备分配 IP 地址和 DNS 服务器，此处以分配 10.185.10.0/24 网段的地址为例。

使能 DHCP 服务。

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

配置 VLAN 接口 1 工作在 DHCP 服务器模式。

```
[DHCP_server] interface vlan-interface 1
[DHCP_server-Vlan-interface1] dhcp select server
[DHCP_server-Vlan-interface1] quit
```

配置不参与自动分配的 IP 地址（此处包括 DNS 服务器、ACS 服务器）。

```
[DHCP_server] dhcp server forbidden-ip 10.185.10.41
[DHCP_server] dhcp server forbidden-ip 10.185.10.60
```

配置 DHCP 地址池 0 的共有属性（网段、DNS 服务器地址）。

```
[DHCP_server] dhcp server ip-pool 0
[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) 配置 option 43 选项。选项内容包括 ACS 的地址、用户名和密码。

将 ACS 的地址、用户名和密码转换成 ASCII 码。其中 URL 地址对应的 ASCII 码为 68 74 74 70 3A 2F 2F 61 63 73 2E 64 61 74 61 62 61 73 65 3A 39 30 39 30 2F 61 63 73。用户名 admin 对应的 ASCII 码为 76 69 63 6B 79，密码 12345 对应的 ASCII 码为 31 32 33 34 35。

```
[DHCP_server-dhcp-pool-0] option 43 hex
0140687474703A2F2F6163732E6461746162617365 3A393039302F616373207669636B79203132333435
```

3. DNS服务器上的配置

在 DNS 服务器上需要配置域名和地址的映射，将 http://acs.database:9090/acs 地址映射为 http://10.185.1.41:9090/acs。具体配置方法请参见您使用的 DNS 服务器软件手册。

4. 将CPE接入网络

将 CPE 上电并连接网线后，CPE 会先通过 DHCP server 获取 IP 地址和连接 ACS 所需的信息，再按照 CWMP 协议的流程自动从 ACS 处获取配置文件。

1.7.3 验证配置

在导航栏中选择“业务>分支网点管理>CPE 交互记录”，进入设备交互记录查询页面，查看与序列号对应的设备是否已经完成部署配置的操作。

图1-20 设备交互记录界面

目 录

1 EAA

1.1 EAA简介

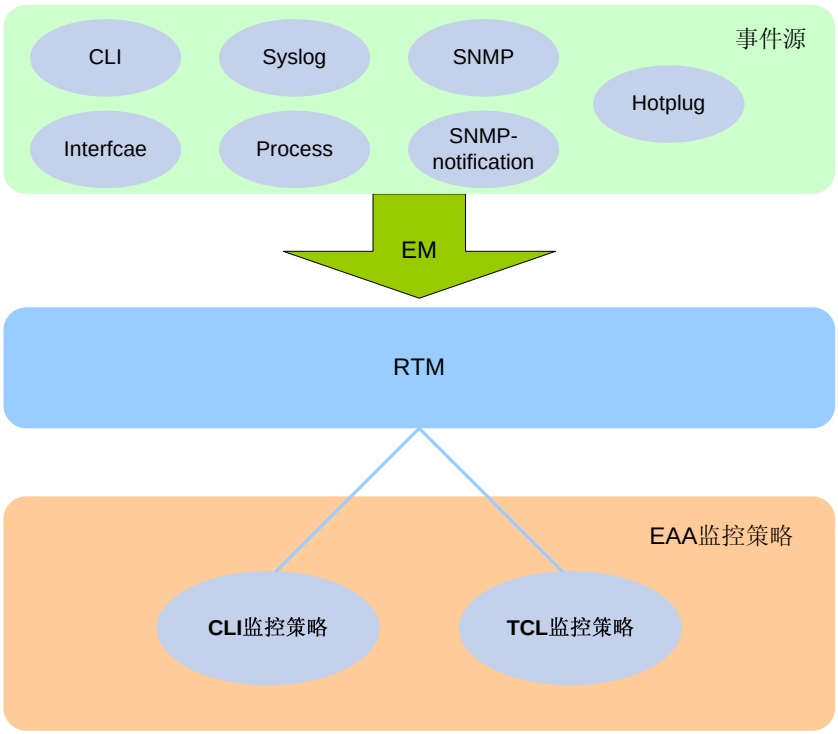
EAA（Embedded Automation Architecture，嵌入式自动化架构）是集成在 Comware V7 平台上的一系列相关软件模块的总称。

使用 EAA 功能：

- 用户可以定制一系列监控策略，在策略中定义自己感兴趣的事件以及事件发生时的处理动作。监控策略被启用后，系统会实时监控设备的运行，当用户定制的事件发生时，就触发相应的监控策略并自动执行监控策略中的动作。
- 设备能够智能地监控多种事件，并做出灵活多变的响应，从而大大地提升系统的可维护性。

1.1.1 EAA框架

图1-1 EAA 框架示意图



EAA框架如 图 1-1 所示，它包括事件源、EM（Event Monitor，事件监控）模块、RTM（Real-Time Event Manager，实时事件管理）模块和EAA监控策略。

1. 事件源

事件源是系统中的软件或硬件模块，它们会触发事件。例如，CLI 事件源能触发命令行事件，Syslog 事件源能触发日志事件。

2. EM

EM 根据用户配置对事件源中发生的事件进行过滤匹配，匹配成功则通知 RTM 执行相应监控策略。当用户配置了多个监控策略，系统会创建多个 EM 模块，每个 EM 组件监控一个事件。

3. RTM

RTM 是 EAA 的核心部件，负责管理监控策略，包括监控策略的创建、状态变化和执行。

4. EAA监控策略

通过监控策略，用户可以定义自己感兴趣的事件以及事件发生时的处理动作。

监控策略有两种配置方式：一种是通过命令行来配置，一种是通过 Tcl 脚本来定义。通过命令行配置的监控策略称为基于 CLI 的监控策略，以下简称 CLI 监控策略；通过 Tcl 脚本定义的监控策略称为基于 Tcl 的监控策略，以下简称 Tcl 监控策略。

1.1.2 CLI监控策略简介

CLI 监控策略通过如下命令行来定义：一条 **event** 命令、一条或多条（不超过 232 条）**action** 命令、一条或多条（不超过 64 条）**user-role** 命令、一条 **running-time** 命令以及一条 **commit** 命令。其中：

- **event**命令用来定义监控策略的触发事件。目前，EAA能够监控的事件类型如 表 1-1 所示。

表1-1 监控事件类型描述表

事件的名称	描述
cli	监控命令行事件 配置该事件后，当用户输入指定的命令并对其进行特定操作（执行、帮助或者补全）就会触发策略执行
syslog	监控日志事件 配置该事件后，当系统在指定时间段内生成指定规格的日志信息时触发监控策略执行；RTM 模块产生的日志不会触发策略执行
process	监控进程事件 配置该事件后，当指定进程（可以为用户命令行触发的或者系统自动触发的）发生指定状态变化（异常、关闭、启动或重启时），触发监控策略执行
hotplug	监控板卡热插拔事件 配置该事件后，当插入和拔出指定板卡，均会触发监控策略执行
interface	监控接口事件 接口事件中存在一个触发开关： (1) 配置该事件后，触发开关立即打开 (2) 当指定接口上的指定报文的数目达到 **start-op** *start-op* **start-val** *start-val* 参数指定的条件时，触发监控策略执行一次（第一次执行），并关闭触发开关，但系统会继续监控接口事件 (3) 当满足 **restart-op** *restart-op* **restart-val** *restart-val* 参数指定的条件时，才重新开启触发开关 (4) 如果指定接口上的指定报文的数目再次达到 **start-op** *start-op* **start-val** *start-val* 参数指定的条件时，则再次触发监控策略执行一次（第二次执行），并关闭触发开关，系统继续监控接口事件 (5) 如此循环

事件的名称	描述
snmp	监控 SNMP 节点值变化事件 SNMP 节点值变化事件中存在一个触发开关： (1) 配置该事件后，触发开关立即打开 (2) 系统根据用户配置，定时轮询设备上某个节点的值，当该值达到 **start-op** *start-op* **start-val** *start-val* 指定的条件时，触发监控策略执行一次（第一次执行），并关闭触发开关，但系统会继续监控 SNMP 节点值变化事件 (3) 当节点值满足 **restart-op** *restart-op* **restart-val** *restart-val* 指定的条件时，才重新开启触发开关 (4) 当节点值再次达到 **start-op** *start-op* **start-val** *start-val* 指定的条件时，则再次触发监控策略执行一次（第二次执行），并关闭触发开关，系统继续监控 SNMP 节点值变化事件 (5) 如此循环
snmp_notification	监控 SNMP Trap 事件 配置该事件后，当系统生成一条 Trap，Trap 中携带的 MIB 对象（由 *oid* 参数指定）的值到达 **oid-val** *oid-val* **op** *op* 指定的条件时，触发监控策略执行

- **action** 命令用来定义事件发生时，监控策略将执行的动作。目前支持的动作有执行指定的命令行、生成一条指定内容的日志、主备倒换和重启。
- **user-role** 命令用来指定执行监控策略的用户角色。用户角色中定义了允许用户操作哪些系统功能以及资源对象，设备支持的每条命令都有缺省用户角色，如果监控策略中指定的用户角色权限比命令行的缺省用户角色的权限小，则不能执行该命令以及该命令后面的所有动作。如果指定的用户角色不存在，则监控策略不能执行。如果给某个监控策略配置了多个用户角色，则使用这些用户角色权限的并集去执行该策略。例如，给某策略配置了用户角色 A 和 B，如果策略中的动作是角色 A 或者 B 允许执行的，则策略可以执行；如果策略中存在角色 A 和 B 都不能执行的命令，则该命令以及该命令后面的所有动作都不能执行。关于用户角色的详细描述请参见“基础配置指导”中的“RBAC”。
- **running-time** 命令用来指定监控策略的运行时间，运行时间达到时即使策略没有执行完毕，也会立即停止执行策略。该命令用于限制策略的运行时间，以免策略长时间运行占用系统资源。而策略是否会触发以及停止后是否会被再次触发则由 **event** 配置决定。
- **commit** 命令用来启用 CLI 监控策略。CLI 监控策略创建和定义后，必须启用才能生效。

1.1.3 Tcl监控策略简介

Tcl 监控策略通过 Tcl 脚本来定义。按内容 Tcl 脚本可以分为两大部分：首行和其它部分。首行用于定义事件、用户角色和运行时间；从第二行开始，定义监控事件发生时执行的动作脚本。

用户创建 Tcl 监控策略并绑定 Tcl 脚本后，设备会首先解析 Tcl 脚本首行，获取监控事件、用户角色、运行时间，并立即下发以上配置且生效。当事件发生后从第二行开始执行动作脚本。

Tcl 脚本首行的格式为**::comware::rtm::event_register** *eventname arg1 arg2 arg3 …***user-role** *rolename1* | [**user-role** *rolename2* | [ ]] [**running-time** *running-time*]。其中：

- *eventname*用来指定事件的类型，Tcl监控策略中支持的事件类型同CLI监控策略，详情请参见表 1-1。需要特别说明的是，注册snmp trap事件时，使用的事件类型为“snmp_notification”，与命令行中不同。

- *arg* 用来指定事件的匹配规则，具体定义与相应 **event** 命令中的参数规格一致，请参考 EAA 命令手册。
- **user-role** 用来指定执行脚本的用户角色，参数含义和配置要求同 CLI 监控策略。
- **running-time** 用来指定脚本运行的最大时间，参数含义和配置要求同 CLI 监控策略。

Tcl 监控策略支持如下三类动作：

- Tcl 语言标准命令。
- EAA 模块的专属 Tcl 命令。
- 设备支持的其它命令行。

1.1.4 EAA环境变量

EAA 环境变量指的是专用于监控策略的环境变量。环境变量由<环境变量名、环境变量值>字对组成。在配置监控策略的动作时，我们可以在应该输入参数的地方输入“$环境变量名”，表示此处需要引用环境变量值。系统在运行监控策略的时候，会自动用环境变量值去替代“$环境变量名”。如果要修改监控参数的值，只需在系统视图下，修改环境变量值即可，而无需进入监控策略视图，修改监控策略下的具体配置。因此，定义和使用环境变量可以简化监控策略的配置，提高监控策略的灵活性和易用性。

目前，EAA 支持的环境变量包括内部环境变量和用户自定义环境变量。

(1) 内部环境变量

设备缺省支持的环境变量，用户不能创建、删除和修改。内部环境变量名均以“_”开头，内部环境变量的值由系统决定。内部环境变量又包括两类：一类是公共环境变量，另一类是非公共环境变量。

- 公共环境变量可用于所有类型的事件：其中_event_id、_event_type、_event_type_string 在系统启动时生成，关联的事件类型不同，其值不同，其值一旦确定不能更改；_event_time、_event_severity 在运行时产生。
- 非公共环境变量只能用于对应的事件，其值在事件触发时获得，它标示当前事件的部分信息。例如，HOTPLUG 事件对应的内部环境变量为“_slot”和“_subslot”，当 1 号单板被插拔的时候，环境变量“_slot”的值为 1；当 2 号单板被插拔的时候，环境变量“_slot”的值为 2。

目前EAA支持内部环境变量如 表 1-2 所示。

(2) 用户自定义环境变量

用户定义环境变量名可包含数字、字符或者“_”，但不能以“_”开头。用户自定义环境变量可用于所有类型的事件，其值由用户配置决定。用户定义环境变量可修改、删除。

表1-2 内部环境变量描述表

事件	内部环境变量的名称	描述
CLI	_cmd	匹配上的命令
SYSLOG	_syslog_pattern	匹配的日志信息的内容
HOTPLUG	_slot	发生热插拔的单板所在的槽位号
	_subslot	发生热插拔的子卡所在的子槽位号
INTERFACE	_ifname	接口的名称

事件	内部环境变量的名称	描述
SNMP	_oid	SNMP操作中携带的OID
	_oid_value	OID对应节点的值
SNMP TRAP	_oid	SNMP Trap信息中携带的OID
PROCESS	_process_name	进程的名称
公共环境变量	_event_id	事件的ID
	_event_type	事件的类型
	_event_type_string	事件类型的描述
	_event_time	事件发生的时间
	_event_severity	事件的严重级别

1.2 配置EAA环境变量

使用本特性，用户可以自定义 EAA 环境变量的名称和值，以便定义监控策略时可以引用。引用环境变量时，环境变量名前面必须加“$”符号，表示引用，如定义了名称为“hostname”的环境变量，引用时需写成“$hostname”。

表1-3 配置 EAA 环境变量

配置步骤	命令	说明
进入系统视图	**system-view**	-
配置环境变量	**rtm environment** *envname env-value*	缺省情况下，无用户自定义的环境变量，系统中支持一系列内部环境变量，如表1-2所示

1.3 配置CLI监控策略

1.3.1 配置指导和注意事项

- 如果监控策略中要用到系统内部环境变量，请先了解相应环境变量的值；如果监控策略中要用到的环境变量不是系统缺省支持的，请先定义，具体配置请参见“1.2 配置EAA环境变量”。
- 同一设备上可创建多个 CLI 监控策略且其数量没有限制。请确保同时启用的策略间动作不要冲突，因为当系统同时执行多个策略，且不同策略间动作有冲突时，执行结果是随机的。
- 给 CLI 监控策略配置事件、动作、用户角色和运行时间后，必须执行 **commit** 命令，该策略才会启用，该策略下的配置才会生效。
- 同一个策略下，只能配置一个触发事件和运行时间。当多次执行 **event** 或者 **running-time** 命令时，则最近配置并且 **commit** 的生效。
- 同一个策略下，最多可配置 232 个动作。当条件满足，策略被触发时，系统会按照动作的编号由小到大顺序执行该策略下的所有动作。
- 如果新配置的动作的编号和已有动作的编号相同，则最近配置并且 **commit** 的生效。

- 同一监控策略下可配置多个用户角色，最多可以配置 64 个有效用户角色，超过该上限后，新配置的用户角色即便 **commit** 也不会生效。

1.3.2 配置步骤

表1-4 配置 CLI 监控策略

配置步骤	命令	说明
进入系统视图	**system-view**	-
创建CLI监控策略并进入CLI监控策略视图	**rtm cli-policy** *policy-name*	如果CLI监控策略已创建，则直接进入CLI监控策略视图
配置命令行事件	**event cli** { **async** [**skip**] | **sync** } **mode** { **execute** | **help** | **tab** } **pattern** *regular-exp*	七种事件选其一 缺省情况下，未配置任何命令行事件
配置板卡热插拔事件（MSR 2600/MSR 3600）	**event hotplug slot** *slot-number*	
配置板卡热插拔事件（MSR 5600）	**event hotplug slot** *slot-number* [**subslot** *subslot-number*]	
配置接口事件	**event interface** *interface-type interface-number* **monitor-obj** *monitor-obj* **start-op** *start-op* **start-val** *start-val* **restart-op** *restart-op* **restart-val** *restart-val* [**interval** *interval*]	
配置进程事件	**event process** { **exception** | **restart** | **shutdown** | **start** } [**name** *process-name* [**instance** *instance-id*]] [**slot** *slot-number*]	
配置SNMP操作事件	**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*]	
配置SNMP Trap事件	**event snmp-notification oid** *oid* **oid-val** *oid-val* **op** *op* [**drop**]	
配置日志事件	**event syslog priority** *level* **msg** *msg* **occurs** *times* **period** *period*	
配置事件发生时执行指定的命令行	**action** *number* **cli** *command-line*	缺省情况下，监控策略下未配置任何动作
配置事件发生时执行重启操作（MSR 2600/MSR 3600）	**action** *number* **reboot** [**slot** *slot-number*]	
配置事件发生时执行重启操作（MSR 5600）	**action** *number* **reboot** [**slot** *slot-number* [**subslot** *subslot-number*]]	
配置事件发生时发送指定的LOG	**action syslog priority** *level* **facility** *local-number* **msg** *msg-body*	
配置事件发生时进行主备倒换	**action** *number* **switchover**	

配置步骤	命令	说明
（可选）配置执行CLI监控策略时使用的用户角色	**user-role** *role-name*	缺省情况下，执行CLI监控策略时使用的用户角色为创建该策略的用户的角色
（可选）配置事件发生时CLI监控策略的运行时间	**running-time** *time*	缺省情况下，CLI监控策略的运行时间为20秒
启用CLI监控策略	**commit**	缺省情况下，CLI监控策略未被启用

1.4 配置Tcl监控策略

提示

Tcl 监控策略启用后，不允许修改 Tcl 脚本。如需修改，请先停用 Tcl 监控策略，修改后，再启用 Tcl 监控策略。否则，Tcl 监控策略将不能运行。

请参照以下步骤来配置 Tcl 监控策略：

(1) 如果监控策略中要用到系统内部环境变量，请先了解相应环境变量的值；如果监控策略中要用到的环境变量不是系统缺省支持的，请先定义，具体配置请参见“1.2 配置EAA环境变量”。

(2) 使用 Tcl 脚本编辑工具或者普通文档编辑工具，编辑格式符合要求的 Tcl 脚本。目前支持的 Tcl 版本为 8.5.8。

(3) 使用 FTP 或者 TFTP 功能将 Tcl 脚本下载到设备上，FTP 及 TFTP 具体配置请参见“基础配置指导”中的“FTP 和 TFTP”。

(4) 创建 Tcl 监控策略。

表1-5 配置 Tcl 监控策略

配置步骤	命令	说明
进入系统视图	**system-view**	-
创建并启用Tcl监控策略，并将它和Tcl脚本绑定	**rtm tcl-policy** *policy-name tcl-filename*	缺省情况下，未创建Tcl监控策略

1.5 暂停运行监控策略

使用本特性可以暂停运行所有的监控策略。如果要将暂停运行的所有策略恢复运行，请使用 **undo rtm scheduler suspend** 命令。

表1-6 配置 Tcl 监控策略

配置步骤	命令	说明
进入系统视图	**system-view**	-
暂停运行所有的监控策略	**rtm scheduler suspend**	-

1.6 EAA显示和维护

在完成上述配置后，在任意视图下执行 **display** 命令可以显示配置后 EAA 的运行情况，通过查看显示信息验证配置的效果。

表1-7 EAA 显示和维护

操作	命令
显示用户自定义的EAA环境变量配置	**display rtm environment** [*var-name*]
显示监控策略的相关信息	**display rtm policy** { **active** | **registered** [**verbose**] } [*policy-name*]

1.7 EAA典型配置举例

1.7.1 配置Tcl监控策略

1. 组网需求

配置一个 Tcl 监控策略，当设备执行包含字符串 **display this** 的命令时，让系统自动发送日志信息 rtm_tcl_test is running。

2. 组网图

图1-2 配置 Tcl 监控策略

3. 配置步骤

在 PC 上使用写字板或者 word 编辑 Tcl 策略脚本 rtm_tcl_test.tcl，内容如下：

```
::comware::rtm::event_register cli sync mode execute pattern display this user-role
network-admin
::comware::rtm::action syslog priority 1 facility local4 msg rtm_tcl_test is running
```

以上脚本的含义为：当设备执行包含字符串 **display this** 的命令时，让系统自动发送日志信息 rtm_tcl_test is running。

通过 TFTP 将 rtm_tcl_test.tcl 下载到设备上。

```
<Sysname> tftp 1.2.1.1 get rtm_tcl_test.tcl
```

进入系统视图。

```
<Sysname> system-view
```
创建并启用 Tcl 监控策略，并将它和 Tcl 脚本 rtm_tcl_test.tcl 绑定。
```
[Sysname] rtm tcl-policy test rtm_tcl_test.tcl
[Sysname] quit
```
4. 验证配置结果

- 通过 **display rtm policy registered** 命令可以看到存在策略名为 test，策略类型为 Tcl 的策略。

```
<Sysname> display rtm policy registered
Total number: 1
Type  Event      TimeRegistered       PolicyName
TLI   CLI        Aug 29 14:54:50 2013 test
```

- 打开允许日志输出的开关，执行 **display this** 命令，有 rtm_tcl_test is running 日志输出，同时有策略运行成功的日志输出。

```
<Sysname> terminal monitor
<Sysname> display this
#
return
<Sysname>%Jun  4 15:02:30:354 2013 Sysname RTM/1/RTM_ACTION:; rtm_tcl_test is running
%Jun  4 15:02:30:382 2013 Sysname RTM/6/RTM_POLICY:; TCL policy test is running successfully.
```

1.7.2 配置CLI监控策略

1. 组网需求

配置一个 CLI 监控策略，当设备执行包含数字、字母（大小写均可）的命令行的帮助操作时，让系统自动发送日志信息 hello world，并创建一个 VLAN。

2. 配置步骤

进入系统视图。
```
<Sysname> system-view
```
创建 CLI 策略。
```
[Sysname] rtm cli-policy test
```
配置监控事件：监控包含数字、字母（大小写均可）的命令行的帮助。
```
[Sysname-rtm-test] event cli async mode help pattern [a-zA-Z0-9]
```
事件发生时，发送优先级为 4，日志记录工具为 local3，信息为 hello world 的日志。
```
[Sysname-rtm-test] action 0 syslog priority 4 facility local3 msg “hello world”
```
事件发生时，进入系统视图。
```
[Sysname-rtm-test] action 2 cli system-view
```
事件发生时，创建 VLAN 2。
```
[Sysname-rtm-test] action 3 cli vlan 2
```
配置策略运行时间。当策略运行时间超过此值时，策略将终止执行，并输出策略执行失败信息。
```
[Sysname-rtm-test] running-time 2000
```
配置用户角色 network-admin 具有执行该策略的权限。
```
[Sysname-rtm-test] user-role network-admin
```
确认执行该策略。
```
[Sysname-rtm-test] commit
```

3. 验证配置结果

- 通过 **display rtm policy registered** 查看，可以看到策略名为 test，策略类型为 CLI 的策略。

```
[Sysname-rtm-test] display rtm policy registered
Total number: 1
Type  Event      TimeRegistered       PolicyName
CLI   CLI        Aug 29 14:56:50 2013 test
```

- 打开允许日志输出的开关，并对包含字母 d 的命令进行帮助，可以看到有信息为“hello world”的日志输出，同时有策略运行成功的日志输出。

```
[Sysname-rtm-test] return
<Sysname> terminal monitor
<Sysname> d?
  debugging
  delete
  diagnostic-logfile
  dir
  display

<Sysname>d%May  7 02:10:03:218 2013 Sysname RTM/4/RTM_ACTION: "hello world"
%May  7 02:10:04:176 2013 Sysname RTM/6/RTM_POLICY: CLI policy test is running su
ccessfully.
```

目 录

1 进程监控和维护

Comvare V7 是 H3C 公司的核心软件平台，它基于 Linux 内核，各个网络服务功能分别运行各自的进程，实现模块化。运行在用户空间的进程称为用户态进程，与用户态进程相对的是内核线程，内核线程运行在内核态空间。

- Comware V7 系统的绝大部分程序是用户态进程。每个用户态进程拥有独立的进程空间，单个进程的异常不会影响系统其他进程，从而提高了系统的可靠性。通常情况下，系统会自动监控用户态进程，不需要用户干预。当单个用户态进程中包含多个独立或半独立的活动，可以将这些活动拆分成多个线程。Comware V7 支持多线程并发和抢占，多个线程分工合作共同实现某个功能。一个进程是否包含多个线程，由软件实现需要决定。
- 内核线程用来执行 Comware V7 内核代码和系统调用。它拥有比用户态进程更高的安全级别，当内核线程发生异常，通常系统会完全崩溃。用户可以使用命令行来监控内核线程的运行状态。

1.1 进程显示和维护

本小节涉及的命令对用户态进程和内核线程均适用，可在任意视图下执行。使用这些命令，可以进行如下操作：

- 显示内存的整体使用情况。
- 显示系统当前运行了哪些进程，每个进程占用了多少内存和多少 CPU。
- 如果某个进程占用内存或者CPU过多，则确认该进程为异常源。如果异常源是用户态进程，可参考“1.2 用户态进程显示和维护”来进一步定位解决问题；如果异常源是内核线程，可参考“1.3 监控内核线程”来进一步定位解决问题。

表1-1 进程显示和维护（MSR 2600/MSR 3600）

操作	命令
显示系统内存使用情况（本命令的详细描述请参见“基础配置命令参考中的“设备管理”）	**display memory**
显示进程的状态信息	**display process** [**all** | **job** *job-id* | **name** *process-name*]
显示所有进程的CPU占有率信息	**display process cpu**
监控进程运行状态	**monitor proces** [**dumbtty**] [**iteration** *number*]
监控线程运行状态	**monitor thread** [**dumbtty**] [**iteration** *number*]

表1-2 进程显示和维护（MSR 5600）

操作	命令
显示系统内存使用情况（本命令的详细描述请参见“基础配置命令参考”中的“设备管理”）	**display memory** [**slot** *slot-number* [**cpu** *cpu-number*]]

操作	命令
显示进程的状态信息	**display process** [**all** | **job** *job-id* | **name** *process-name*] [**slot** *slot-number* [**cpu** *cpu-number*]]
显示所有进程的CPU占有率信息	**display process cpu** [**slot** *slot-number* [**cpu** *cpu-number*]]
监控进程运行状态	**monitor proces** [**dumbtty**] [**iteration** *number*] [**slot** *slot-number* [**cpu** *cpu-number*]]
监控线程运行状态	**monitor thread** [**dumbtty**] [**iteration** *number*] [**slot** *slot-number* [**cpu** *cpu-number*]]

1.2 用户态进程显示和维护

当用户态进程运行异常，可使用如下命令来定位故障。其中，**display** 命令可在任意视图下执行，其它命令在用户视图下执行。

表1-3 用户态进程显示和维护（MSR 2600/MSR 3600）

操作	命令	
显示所有用户态进程的日志信息	**display process log**	-
显示所有用户态进程的代码段、数据段以及堆栈等的内存使用信息	**display process memory**	-
显示用户态进程堆内存的使用情况	**display process memory heap job** *job-id* [**verbose**]	-
显示指定大小已使用内存块的地址	**display process memory heap job** *job-id* **size** *memory-size* [**offset** *offset-size*]	-
显示从指定地址开始的内存空间的内容	**display process memory heap job** *job-id* **address** *starting-address* **length** *memory-length*	-
显示用户态进程异常时的上下文信息	**display exception context** [**count** *value*]	-
显示core文件的保存路径	**display exception filepath**	-
开启/关闭用户态进程异常时的生成core文件的功能，以及配置能生成的core文件的最大个数	**process core** { **maxcore** *value* | **off** } { **job** *job-id* | **name** *process-name* }	缺省情况下，用户态进程在首次异常时会生成core文件，后续异常不再生成core文件。即**maxcore**的最大数值为1
设置core文件的保存路径	**exception filepath** *directory*	缺省情况下，core文件的保存在存储介质的根目录下
清除用户态进程异常时记录的上下文信息	**reset exception context**	-

表1-4 用户态进程显示和维护（MSR 5600）

操作	命令	
显示所有用户态进程的日志信息	**display process log** [**slot** *slot-number* [**cpu** *cpu-number*]]	-
显示所有用户态进程的代码段、数据段以及堆栈等的内存使用信息	**display process memory** [**slot** *slot-number* [**cpu** *cpu-number*]]	-
显示用户态进程堆内存的使用情况	**display process memory heap job** job-id [**verbose**] [**slot** *slot-number* [**cpu** *cpu-number*]]	-
显示指定大小已使用内存块的地址	**display process memory heap job** *job-id* **size** *memory-size* [**offset** *offset-size*] [**slot** *slot-number* [**cpu** *cpu-number*]]	-
显示从指定地址开始的内存空间的内容	**display process memory heap job** *job-id* **address** *starting-address* **length** *memory-length* [**slot** *slot-number* [**cpu** *cpu-number*]]	-
显示用户态进程异常时的上下文信息	**display exception context** [**count** *value*] [**slot** *slot-number* [**cpu** *cpu-number*]]	-
显示core文件的保存路径	**display exception filepath** [**slot** *slot-number* [**cpu** *cpu-number*]]	-
开启/关闭用户态进程异常时的生成core文件的功能，以及配置能生成的core文件的最大个数	**process core** { **maxcore** *value* | **off** } { **job** *job-id* | **name** *process-name* } [**slot** *slot-number* [**cpu** *cpu-number*]]	缺省情况下，用户态进程在首次异常时会生成core文件，后续异常不再生成core文件。即**maxcore**的最大数值为1
设置core文件的保存路径	**exception filepath** *directory*	缺省情况下，core文件的保存在存储介质的根目录下
清除用户态进程异常时记录的上下文信息	**reset exception context** [**slot** *slot-number* [**cpu** *cpu-number*]]	-

1.3 监控内核线程

如果某个内核线程运行异常，可以显示该内核线程是否发生了死循环或者处于饿死状态，以及异常、重启的原因。

1.3.1 配置内核线程死循环检测功能

提示

对于本特性，建议用户使用缺省配置即可。如果确实需要修改缺省配置，请在 H3C 工程师的指导下进行，以免引起系统异常。

在内核态空间中，所有资源都是共享的，多个内核线程之间通过任务调度协调工作。如果某个内核线程长时间一直占用 CPU，就会导致其它内核线程获取不到运行机会，整个系统挂死，我们称这种现象为死循环。

开启内核线程死循环检测功能后，如果系统发现某内核线程在指定时间内一直占用 CPU，则判定该内核线程为死循环。系统会记录一条死循环信息供管理员查询，并自动重启整个系统来解除死循环。

表1-5 配置内核线程死循环检测功能（MSR 2600/MSR 3600）

操作	命令	说明
进入系统视图	**system-view**	-
开启内核线程死循环检测功能	**monitor kernel deadloop enable**	缺省情况下，内核线程死循环检测功能处于关闭状态
（可选）配置判定内核线程是否死循环的时长	**monitor kernel deadloop time** *interval*	缺省情况下，当某内核线程连续运行超过8秒钟，则判定为死循环
（可选）配置不检测指定内核线程是否发生了死循环	**monitor kernel deadloop exclude-thread** *tid*	缺省情况下，开启内核线程死循环检测功能后，会监控所有内核线程是否发生了死循环

表1-6 配置内核线程死循环检测功能（MSR 5600）

操作	命令	说明
进入系统视图	**system-view**	-
开启内核线程死循环检测功能	**monitor kernel deadloop enable** [**slot** *slot-number* [**cpu** *cpu-number*]]	缺省情况下，内核线程死循环检测功能处于关闭状态
（可选）配置判定内核线程是否死循环的时长	**monitor kernel deadloop time** *interval* [**slot** *slot-number* [**cpu** *cpu-number*]]	缺省情况下，当某内核线程连续运行超过8秒钟，则判定为死循环
（可选）配置不检测指定内核线程是否发生了死循环	**monitor kernel deadloop exclude-thread** *tid* [**slot** *slot-number* [**cpu** *cpu-number*]]	缺省情况下，开启内核线程死循环检测功能后，会监控所有内核线程是否发生了死循环

1.3.2 配置内核线程饿死检测功能

提示

开机后，系统会自动检测内核线程是否发生了饿死，建议用户不要随意配置内核线程饿死检测功能。如果确实需要配置，请在 H3C 工程师的指导下进行，以免引起系统异常。

如果内核线程本身的触发条件没有达到，会导致该内核线程在一段时间内一直得不到调度，我们称这种现象为饿死。

开启内核线程饿死检测功能后，当系统检测到某内核线程饿死时，会记录一条饿死信息供管理员查询。

内核线程饿死并不会影响整个系统的运行，当触发条件达到，处于饿死状态的内核线程会自动执行。

表1-7 配置内核线程饿死检测功能（MSR 2600/MSR 3600）

操作	命令	说明
进入系统视图	**system-view**	-
开启内核线程饿死检测功能	**monitor kernel starvation enable**	缺省情况下，内核线程饿死检测功能处于关闭状态
（可选）配置判定内核线程是否饿死的时长	**monitor kernel starvation time** *interval*	缺省情况下，当某内核线程在120秒内一直没有运行，则认为该内核线程被饿死
（可选）配置不检测指定内核线程是否发生了饿死	**monitor kernel starvation exclude-thread** *tid*	缺省情况下，开启内核线程饿死检测功能后，会监控所有内核线程是否发生了饿死

表1-8 配置内核线程饿死检测功能（MSR 5600）

操作	命令	说明
进入系统视图	**system-view**	-
开启内核线程饿死检测功能	**monitor kernel starvation enable** [**slot** *slot-number* [**cpu** *cpu-number*]]	缺省情况下，内核线程饿死检测功能处于关闭状态
（可选）配置判定内核线程是否饿死的时长	**monitor kernel starvation time** *interval* [**slot** *slot-number* [**cpu** *cpu-number*]]	缺省情况下，当某内核线程连续运行超过8秒钟，则判定为饿死
（可选）配置不检测指定内核线程是否发生了饿死	**monitor kernel starvation exclude-thread** *tid* [**slot** *slot-number* [**cpu** *cpu-number*]]	缺省情况下，开启内核线程饿死检测功能后，会监控所有内核线程是否发生了饿死

1.3.3 内核线程显示和维护

在任意视图下，通过显示内核线程的显示信息，用户可以更好的了解内核线程的实时运行状态；同时，当出现系统异常繁忙或者资源消耗异常等故障时，显示信息可以帮助用户确认出现故障的功能点，以便尽快进行功能的恢复。

在用户视图下，执行 **reset** 命令，可以清除内核线程的统计信息。

表1-9 内核线程显示和维护（MSR 2600/MSR 3600）

操作	命令
显示内核线程死循环信息	**display kernel deadloop** *show-number* [*offset*] [**verbose**]
显示内核线程死循环监控参数配置	**display kernel deadloop configuration**
显示内核线程的异常信息	**display kernel exception** *show-number* [*offset*] [**verbose**]
显示内核线程的重启信息	**display kernel reboot** *show-number* [*offset*] [**verbose**]
显示内核线程饿死信息	**display kernel starvation** *show-number* [*offset*] [**verbose**]

操作	命令
显示内核线程饿死监控参数配置	**display kernel starvation configuration**
清除内核线程死循环信息	**reset kernel deadloop**
清除内核线程的异常信息	**reset kernel exception**
清除内核线程重启信息	**reset kernel reboot**
清除内核线程饿死信息	**reset kernel starvation**

表1-10 内核线程显示和维护（MSR 5600）

操作	命令
显示内核线程死循环信息	**display kernel deadloop** *show-number* [*offset*] [**verbose**] [**slot** *slot-number* [**cpu** *cpu-number*]]
显示内核线程死循环监控参数配置	**display kernel deadloop configuration** [**slot** *slot-number* [**cpu** *cpu-number*]]
显示内核线程的异常信息	**display kernel exception** *show-number* [*offset*] [**verbose**] [**slot** *slot-number* [**cpu** *cpu-number*]]
显示内核线程的重启信息	**display kernel reboot** *show-number* [*offset*] [**verbose**] [**slot** *slot-number* [**cpu** *cpu-number*]]
显示内核线程饿死信息	**display kernel starvation** *show-number* [*offset*] [**verbose**] [**slot** *slot-number* [**cpu** *cpu-number*]]
显示内核线程饿死监控参数配置	**display kernel starvation configuration** [**slot** *slot-number* [**cpu** *cpu-number*]]
清除内核线程死循环信息	**reset kernel deadloop** [**slot** *slot-number* [**cpu** *cpu-number*]]
清除内核线程的异常信息	**reset kernel exception** [**slot** *slot-number* [**cpu** *cpu-number*]]
清除内核线程重启信息	**reset kernel reboot** [**slot** *slot-number* [**cpu** *cpu-number*]]
清除内核线程饿死信息	**reset kernel starvation** [**slot** *slot-number* [**cpu** *cpu-number*]]

目 录

1 Sampler

1.1 Sampler简介

Sampler 即报文采样功能，用来从一组固定数量的报文中抽取一个报文，送交其他业务模块处理。Sampler 支持固定采样和随机采样两种方式：

- 固定采样：每组报文中的第一个报文被抽取。
- 随机采样：每组报文中，任意一个报文都有可能被抽取。

目前 Sampler 可以和流量监控特性 NetStream 配合使用，达到以下目的：

- 对网络流量先进行采样，然后将采样后的流量发送给 NetStream 系统进行流量统计。通过设定适当的采样间隔，在减少统计的报文数量的前提下，使 NetStream 系统收集到的统计信息能基本反映整个网络的状况。
- 通过采样可以减少网络的流量，避免网络中的大流量对设备转发性能造成影响。

有关 NetStream 的详细介绍，请参见“网络管理和监控配置指导”中的“NetStream”。

1.2 创建采样器

表1-1 创建采样器

操作	命令	说明
进入系统视图	**system-view**	-
创建采样器	**sampler** *sampler-name* **mode** { **fixed** | **random** } **packet-interval** *rate*	缺省情况下，未创建任何采样器

1.3 Sampler显示和维护

在完成上述配置后，在任意视图下执行 **display** 命令可以显示配置后 Sampler 的配置情况，通过查看显示信息验证配置的效果。

表1-2 Sampler 显示和维护

操作	命令
查看采样器的配置信息（MSR 2600/MSR 3600）	**display sampler** [*sampler-name*]
查看采样器的配置信息（MSR 5600）	**display sampler** [*sampler-name*] [**slot** *slot-number*]

1.4 Sampler典型配置举例

1.4.1 Sampler与IPv4 NetStream配合使用

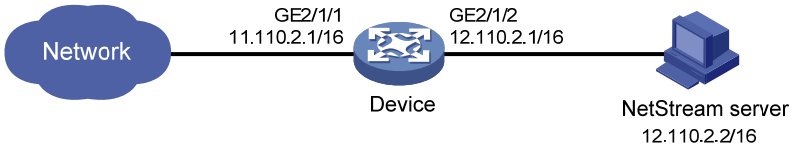
1. 组网需求

如 图 1-1 所示，在Device上配置IPv4 NetStream入统计和出统计，并将统计结果发送到NetStream服务器。NetStream的IP地址为 12.110.2.2/16，UDP端口号为 5000。具体要求如下：

- 在接口 GigabitEthernet2/1/2 的入方向上配置固定采样，每 100 个报文中抽取一个进行 NetStream 统计。
- 在接口 GigabitEthernet2/1/2 的出方向上配置随机采样，每 200 个报文中抽取一个进行 NetStream 统计。

2. 组网图

图1-1 Sampler 与 IPv4 NetStream 配合使用组网图



3. 配置步骤

创建一个名为 100 的采样器，采用固定采样方式，设置采样率为 100，即 100 个报文中采样 1 个报文。

```
<Device> system-view
[Device] sampler 100 mode fixed packet-interval 100
```

创建一个名为 200 的采样器，采用随机采样方式，设置采样率为 200，即 200 个报文中采样 1 个报文。

```
[Device] sampler 200 mode random packet-interval 200
```

配置 GigabitEthernet2/1/2，在此接口上启动 IPv4 NetStream 入统计并使用采样器 100。

```
[Device] interface gigabitethernet 2/1/2
[Device-GigabitEthernet2/1/2] ip address 11.110.2.1 255.255.0.0
[Device-GigabitEthernet2/1/2] ip netstream inbound
[Device-GigabitEthernet2/1/2] ip netstream sampler 100 inbound
```

配置 GigabitEthernet2/1/2，在此接口上启动 IPv4 NetStream 出统计并使用采样器 200。

```
[Device-GigabitEthernet2/1/2] ip netstream outbound
[Device-GigabitEthernet2/1/2] ip netstream sampler 200 outbound
[Device-GigabitEthernet2/1/2] quit
```

配置 IPv4 NetStream 统计输出报文的目的 IP 地址和目的端口（即 NSC 的地址以及端口），源接口采用缺省配置。

```
[Device] ip netstream export host 12.110.2.2 5000
```

4. 验证配置

通过 **display sampler** 命令查看采样器 100 和采样器 200 的配置信息。

```
[Device] display sampler 100
 Sampler name: 100
```

```
  Mode: Fixed;  Packet-interval: 100
[Device] display sampler 200
 Sampler name: 200
  Mode: Random;  Packet-interval: 200
```

目 录

1 端口镜像

1.1 端口镜像简介

端口镜像通过将指定端口的报文复制到与数据监测设备相连的端口，使用户可以利用数据监测设备分析这些复制过来的报文，以进行网络监控和故障排除。

1.1.1 基本概念

1. 镜像源

镜像源是指被监控的对象，该对象可以是端口，我们将之依次称为源端口。经由被监控的对象收发的报文会被复制一份到与数据监测设备相连的端口，用户就可以对这些报文（称为镜像报文）进行监控和分析了。镜像源所在的设备就称为源设备。

2. 镜像目的

镜像目的是指镜像报文所要到达的目的地，即与数据监测设备相连的那个端口，我们称之为目的端口，目的端口所在的设备就称为目的设备。目的端口会将镜像报文转发给与之相连的数据监测设备。

由于一个目的端口可以同时监控多个镜像源，因此在某些组网环境下，目的端口可能收到对同一报文的多份拷贝。例如，目的端口 Port 1 同时监控同一台设备上的源端口 Port 2 和 Port 3 收发的报文，如果某报文从 Port 2 进入该设备后又从 Port 3 发送出去，那么该报文将被复制两次给 Port 1。

3. 镜像方向

镜像方向是指在镜像源上可复制哪些方向的报文：

- 入方向：是指仅复制镜像源收到的报文。
- 出方向：是指仅复制镜像源发出的报文。
- 双向：是指对镜像源收到和发出的报文都进行复制。

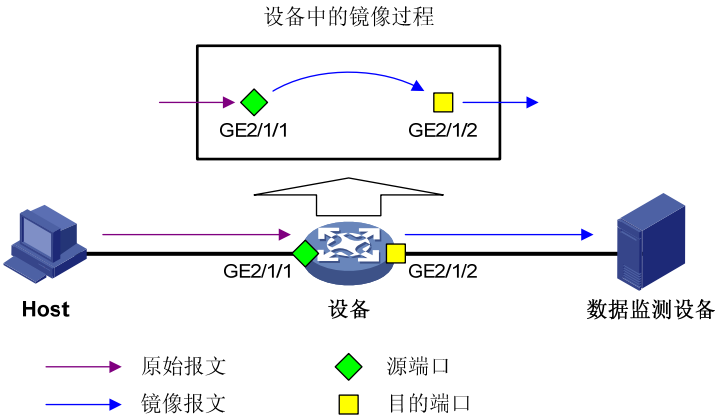
4. 镜像组

镜像组是一个逻辑上的概念，镜像源和镜像目的都要属于某一个镜像组。

1.1.2 端口镜像的实现方式

当源设备与数据监测设备直接相连时，源设备可以同时作为目的设备，即由本设备将镜像报文转发至数据检测设备，这种方式实现的端口镜像称为本地端口镜像。对于本地端口镜像，镜像源和镜像目的属于同一台设备上的同一个镜像组，该镜像组称为本地镜像组。

图1-1 本地端口镜像示意图



如 图 1-1 所示，现在需要设备将进入端口GigabitEthernet2/1/1 的报文复制一份，从端口GigabitEthernet2/1/2 将报文转发给数据监测设备。为满足该需求，可以配置本地镜像组，其中源端口为GigabitEthernet2/1/1，镜像方向为入方向，目的端口为GigabitEthernet2/1/2。

1.2 配置本地端口镜像

1.2.1 配置任务简介

在完成源端口和目的端口的配置之后，本地镜像组才能生效。

表1-1 本地端口镜像配置任务简介

配置任务	说明	详细配置
创建本地镜像组	必选	1.2.2
配置源端口	必选	1.2.3
配置目的端口	必选	1.2.4

1.2.2 创建本地镜像组

表1-2 创建本地镜像组

操作	命令	说明
进入系统视图	**system-view**	-
创建本地镜像组	**mirroring-group** *group-id* **local**	缺省情况下，不存在任何本地镜像组

1.2.3 配置源端口

可以在系统视图下为指定镜像组配置一个或多个源端口，也可以在接口视图下将当前接口配置为指定镜像组的源端口，二者的配置效果相同。

配置源端口时，需要注意：

- 一个镜像组内可以配置多个源端口。
- 通常，一个端口只能被一个镜像组使用。

1. 在系统视图下配置源端口

表1-3 在系统视图下配置源端口

操作	命令	说明
进入系统视图	**system-view**	-
为本地镜像组配置源端口	**mirroring-group** *group-id* **mirroring-port** *interface-list* { **both** | **inbound** | **outbound** }	缺省情况下，本地镜像组没有源端口

2. 在接口视图下配置源端口

表1-4 在接口视图下配置源端口

操作	命令	说明
进入系统视图	**system-view**	-
进入接口视图	**interface** *interface-type interface-number*	必选
配置本端口为本地镜像组的源端口	**mirroring-group** *group-id* **mirroring-port** { **both** | **inbound** | **outbound** }	缺省情况下，端口不是任何本地镜像组的源端口

1.2.4 配置目的端口

可以在系统视图下为指定镜像组配置目的端口，也可以在接口视图下将当前接口配置为指定镜像组的目的端口，二者的配置效果相同。

配置目的端口时，需要注意：

- 从目的端口发出的报文包括镜像报文和其他端口正常转发来的报文。为了保证数据监测设备只对镜像报文进行分析，请将目的端口只用于端口镜像，不作其他用途。
- 一个镜像组内只能配置一个目的端口。

1. 在系统视图下配置目的端口

表1-5 在系统视图下配置目的端口

操作	命令	说明
进入系统视图	**system-view**	-
为本地镜像组配置目的端口	**mirroring-group** *group-id* **monitor-port** *interface-type interface-number*	缺省情况下，本地镜像组没有目的端口

2. 在接口视图下配置目的端口

表1-6 在接口视图下配置目的端口

操作	命令	说明
进入系统视图	**system-view**	-

操作	命令	说明
进入接口视图	**interface** *interface-type interface-number*	-
配置本端口为本地镜像组的目的端口	**mirroring-group** *group-id* **monitor-port**	缺省情况下，端口不是任何本地镜像组的目的端口

1.3 端口镜像显示和维护

在完成上述配置后，在任意视图下执行 **display** 命令可以显示配置后镜像组的运行情况，通过查看显示信息验证配置的效果。

表1-7 端口镜像显示和维护

操作	命令
显示镜像组的配置信息	**display mirroring-group** { *group-id* | **all** | **local** }

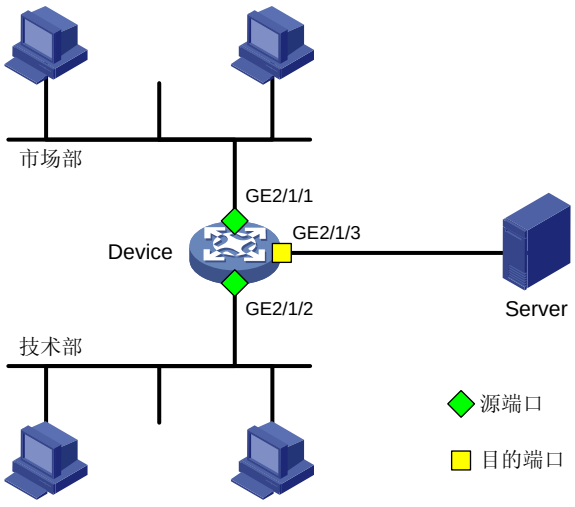
1.4 端口镜像典型配置举例

1. 组网需求

- Device 通过端口 GigabitEthernet2/1/1 和 GigabitEthernet2/1/2 分别连接市场部和技术部，并通过端口 GigabitEthernet2/1/3 连接 Server。
- 通过配置源端口方式的本地端口镜像，使 Server 可以监控所有进、出市场部和技术部的报文。

2. 组网图

图1-2 本地端口镜像配置组网图



3. 配置步骤

创建本地镜像组 1。

```
<Device> system-view
```

```
[Device] mirroring-group 1 local
```

配置本地镜像组 1 的源端口为 GigabitEthernet2/1/1 和 GigabitEthernet2/1/2，目的端口为 GigabitEthernet2/1/3。

```
[Device] mirroring-group 1 mirroring-port gigabitethernet 2/1/1 gigabitethernet 2/1/2 both
[Device] mirroring-group 1 monitor-port gigabitethernet 2/1/3
```

在目的端口 GigabitEthernet2/1/3 上关闭生成树协议。

```
[Device] interface gigabitethernet 2/1/3
[Device-GigabitEthernet2/1/3] undo stp enable
[Device-GigabitEthernet2/1/3] quit
```

4. 验证配置

显示所有镜像组的配置信息。

```
[Device] display mirroring-group all
Mirroring group 1:
    Type: Local
    Status: Active
    Mirroring port:
        GigabitEthernet2/1/1  Both
        GigabitEthernet2/1/2  Both
    Monitor port: GigabitEthernet2/1/3
```

配置完成后，用户可以通过 Server 监控所有进、出市场部和技术部的报文。

2 流镜像

说明

本特性仅在安装了二层接口模块的款型上和 MSR3600-28/MSR3600-51 的固定二层接口上支持。

2.1 流镜像简介

流镜像是指将指定报文复制到指定目的地，以便于对报文进行分析和监控。流镜像通过 QoS 策略来实现，即使用流分类技术为待镜像报文定义匹配条件，再通过配置流行为将符合条件的报文镜像至指定目的地。其优势在于用户通过流分类技术可以灵活地配置匹配条件，从而对报文进行精细区分，并将区分后的报文复制到目的地进行分析。有关 QoS 策略、流分类和流行为的详细介绍，请参见“ACL 和 QoS 配置指导”中的“QoS 配置方式”。

流镜像到接口是将符合条件的报文复制一份到指定接口（与数据检测设备相连的接口），利用数据检测设备分析接口收到的报文。

2.2 流镜像配置任务简介

表2-1 流镜像配置任务简介

配置任务		说明	详细配置
配置报文匹配规则		必选 用来匹配待镜像的报文	2.3.1
配置流行为		必选 用来指定将报文镜像到哪里（即报文的目的地址）	2.3.2
配置QoS策略		必选 为流分类指定采用的流行为，即指定哪些报文需要镜像到哪里	2.3.3
应用QoS策略	基于接口应用	二者至少选其一 指定对来自哪个端口的流量进行镜像	2.3.4 1.
	基于控制平面应用		2.3.4 2.

2.3 配置流镜像

除 **mirror-to** 命令外的其他配置命令及相关显示命令的详细介绍，请参见“ACL 和 QoS 命令参考”中的“QoS 策略”。

2.3.1 配置报文匹配规则

表2-2 配置报文匹配规则

操作	命令	说明
进入系统视图	**system-view**	-
定义流分类，并进入流分类视图	**traffic classifier** *tcl-name* [**operator** { **and** | **or** }]	缺省情况下，不存在任何流分类
配置报文匹配规则	**if-match** [**not**] *match-criteria*	缺省情况下，流分类中不存在任何报文匹配规则

2.3.2 配置流行为

表2-3 配置流行为

操作		命令	说明
进入系统视图		**system-view**	-
定义流行为，并进入流行为视图		**traffic behavior** *behavior-name*	缺省情况下，不存在任何流行为
在流行为中配置流量的目的地	配置流镜像到接口	**mirror-to interface** *interface-type interface-number*	必选 缺省情况下，流行为中未指定流量的目的地

说明

在完成上述配置后，在任意视图下执行 **display traffic behavior** 命令可以显示用户定义流行为的配置信息，通过查看显示信息验证配置的效果。

2.3.3 配置QoS策略

表2-4 配置 QoS 策略

操作	命令	说明
进入系统视图	**system-view**	-
定义QoS策略，并进入QoS策略视图	**qos policy** *policy-name*	缺省情况下，不存在任何策略
为流分类指定采用的流行为	**classifier** *tcl-name* **behavior** *behavior-name*	缺省情况下，没有为流分类指定采用的流行为

说明

在完成上述配置后，在任意视图下执行**display qos policy**命令可以显示用户定义策略的配置信息，通过查看显示信息验证配置的效果。

2.3.4 应用QoS策略

1. 基于接口应用

将 QoS 策略应用到某接口，可以对该接口指定方向上的流量进行镜像。一个 QoS 策略可以应用于多个接口，而接口在每个方向上只能应用一个 QoS 策略。

表2-5 基于接口应用

操作	命令	说明
进入系统视图	**system-view**	-
进入接口视图	**interface** *interface-type interface-number*	-
应用QoS策略到接口	**qos apply policy** *policy-name* { **inbound** | **outbound** }	-

2. 基于控制平面应用

将 QoS 策略应用到控制平面，可以对控制平面各端口指定方向上的流量进行镜像。

表2-6 基于控制平面应用

操作	命令	说明
进入系统视图	**system-view**	-
进入控制平面视图（MSR 2600/MSR 3600）	**control-plane**	二者选其一
进入控制平面视图（MSR 5600）	**control-plane slot** *slot-number*	
应用QoS策略到控制平面	**qos apply policy** *policy-name* **inbound**	-

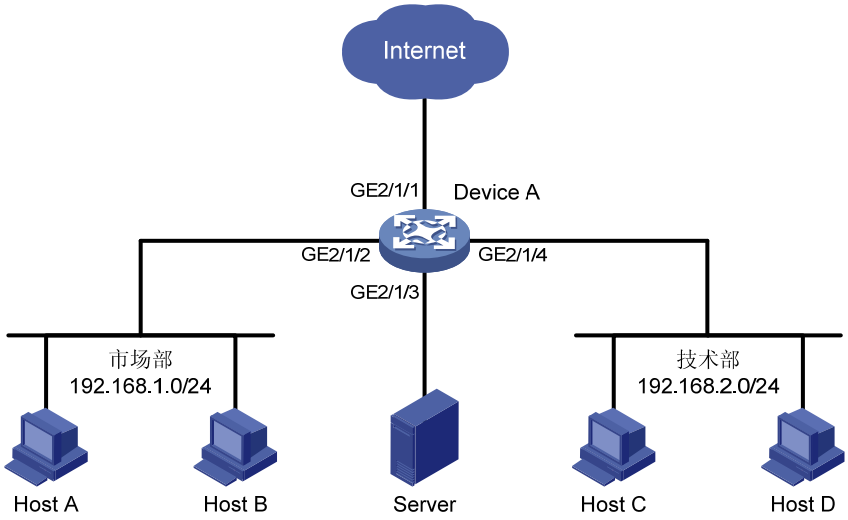
2.4 流镜像典型配置举例

1. 组网需求

- 某公司内的各部门之间使用不同网段的IP地址，其中市场部和技术部分别使用192.168.1.0/24和192.168.2.0/24网段，该公司的工作时间为每周工作日的8点到18点。
- 通过配置流镜像，使Server可以监控技术部访问互联网的WWW流量，以及技术部在工作时间发往市场部的IP流量。

2. 组网图

图2-1 流镜像典型配置组网图



3. 配置步骤

定义工作时间：创建名为 work 的时间段，其时间范围为每周工作日的 8 点到 18 点。

```
<DeviceA> system-view
[DeviceA] time-range work 8:00 to 18:00 working-day
```

创建 ACL 3000，并定义如下规则：匹配技术部访问 WWW 的报文，以及在工作时间由技术部发往市场部的 IP 报文。

```
[DeviceA] acl number 3000
[DeviceA-acl-adv-3000] rule permit tcp source 192.168.2.0 0.0.0.255 destination-port eq www
[DeviceA-acl-adv-3000] rule permit ip source 192.168.2.0 0.0.0.255 destination 192.168.1.0
0.0.0.255 time-range work
[DeviceA-acl-adv-3000] quit
```

创建流分类 tech_c，并配置报文匹配规则为 ACL 3000。

```
[DeviceA] traffic classifier tech_c
[DeviceA-classifier-tech_c] if-match acl 3000
[DeviceA-classifier-tech_c] quit
```

创建流行为 tech_b，并配置流镜像到接口 GigabitEthernet2/1/3。

```
[DeviceA] traffic behavior tech_b
[DeviceA-behavior-tech_b] mirror-to interface gigabitethernet 2/1/3
[DeviceA-behavior-tech_b] quit
```

创建 QoS 策略 tech_p，在策略中为流分类 tech_c 指定采用流行为 tech_b。

```
[DeviceA] qos policy tech_p
[DeviceA-qospolicy-tech_p] classifier tech_c behavior tech_b
[DeviceA-qospolicy-tech_p] quit
```

将 QoS 策略 tech_p 应用到接口 GigabitEthernet2/1/4 的入方向上。

```
[DeviceA] interface gigabitethernet 2/1/4
[DeviceA-GigabitEthernet2/1/4] qos apply policy tech_p inbound
[DeviceA-GigabitEthernet2/1/4] quit
```

4. 验证配置

配置完成后，用户可以通过 Server 监控技术部访问互联网的 WWW 流量，以及技术部在工作时间发往市场部的 IP 流量。

目 录

1 NetStream

1.1 NetStream简介

1.1.1 NetStream技术应用背景

随着 Internet 的高速发展，Internet 为用户提供了更高的带宽，Internet 上支持的业务和应用种类也日渐增多。传统的流量统计技术（如 SNMP、端口镜像等），由于统计方式不灵活、需要投资专用服务器成本高等原因，无法对网络进行更细致的管理。因此，需要一种新的技术来更好地支持网络流量统计。

NetStream 技术是一种基于网络流信息的统计技术，可以对网络中的业务流量情况进行统计和分析。在网络的接入层、汇聚层、核心层上，都可以部署 NetStream。

NetStream 技术的应用有以下几种。

- 计费：NetStream 为基于资源（如线路、带宽、时段等）占用情况的计费提供了精细的数据。ISP（Internet Service Provider，互联网服务提供商）可以利用这些信息来实行灵活的计费策略，如基于时间、带宽、应用、服务质量等。企业客户可以使用这些信息计算部门费用或分配成本，以便有效利用资源。
- 网络规划：NetStream 可以为网络管理工具提供关键信息，比如各个 AS（Autonomous System，自治系统）域之间的网络流量情况，以便优化网络设计和规划，实现以最小的网络运营成本达到最佳的网络性能和可靠性。有关自治系统的相关介绍，请参见“三层技术-IP 路由配置指导”中的“BGP”。
- 网络监控：通过在出口部署 NetStream，对连接 Internet 网络的接口进行实时的流量监控，可以分析各种业务占用出口带宽的情况。网管人员可以根据这些信息判断网络的运行情况，尽早发现不合理的网络结构或网络中的性能瓶颈，方便网管人员规划和分配网络资源。
- 用户监控和分析：通过 NetStream 技术可以使网管人员轻松获取用户使用网络和应用资源的详细情况，进而高效地规划以及分配网络资源，并保障网络的安全运行。

1.1.2 基本概念

1. NetStream流

NetStream 是一项基于“流”来提供报文统计信息的技术。它根据 IPv4 报文的目的 IP 地址、源 IP 地址、目的端口号、源端口号、协议号、ToS（Type of Service，服务类型）、输入接口或输出接口来定义流，七元组相同的报文属于同一条流。

2. NetStream系统组成

一个典型的 NetStream 系统由 NDE（NetStream Data Exporter，网络流数据输出者）、NSC（NetStream Collector，网络流数据收集者）和 NDA（NetStream Data Analyzer，网络流数据分析者）三部分组成。

- NDE

NDE 根据七元组对网络流进行分类，提取符合条件的流进行统计，并将统计信息输出给 NSC 设备。输出前也可对数据进行一些处理，比如聚合。配置了 NetStream 功能的设备在 NetStream 系统中担当 NDE 角色。

- NSC

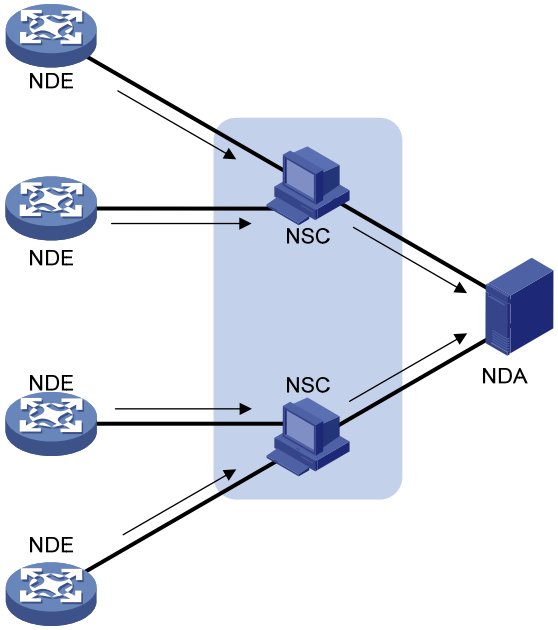
NSC 通常为运行于某种操作系统上的一个应用程序，负责解析来自 NDE 的报文，把统计数据收集到数据库中，可供 NDA 进行解析。NSC 可以采集多个 NDE 设备输出的数据。

- NDA

NDA 是一个网络流量分析工具，它从 NSC 中提取统计数据，进行进一步的加工处理，生成报表，为各种业务提供依据（比如流量计费、网络规划，攻击监测）。NDA 可以提取多个 NSC 中的数据。通常，NDA 具有图形化用户界面，可以使用户方便地获取、显示和分析收集到的数据。

NSC 和 NDA 可以集成在一台 NetStream 服务器上。

图1-1 NetStream 系统中的设备角色



1.1.3 NetStream工作机制

如 图 1-1 所示，NetStream的工作过程如下：

(1) 配置了 NetStream 功能的设备（即 NDE）把采集到的关于流的详细信息定期发送给 NSC；

(2) 信息由 NSC 初步处理后发送给 NDA；

(3) NDA 对数据进行分析，以用于计费、网络规划等应用。

由于设备在 NetStream 系统中担任 NDE 角色，所以本文重点介绍 NDE 的实现以及配置。

NetStream 工作机制中有如下几项关键技术：

1. 流老化

NetStream流老化是设备向NetStream服务器输出流统计信息的一种手段。当设备启用NetStream功能后，流统计信息首先会被存储在设备的NetStream缓冲区中。当存储在设备上的NetStream流信息老化后，设备会把缓冲区中的流统计信息通过指定版本的NetStream输出报文发送给NetStream

服务器，同时清除缓冲区中的对应信息。流老化的三种方式及其配置请参见“1.3.5 配置NetStream的流老化”。

2. 流输出

(1) 普通流输出

普通流输出是指所有流的统计信息都要被统计。在流老化后，每条流的统计信息都要输出到NetStream 服务器。

普通流输出的优点是，NetStream 服务器可以得到每条流的详细统计信息。但其缺点也是很明显的，这种方式增加了网络带宽和设备的 CPU 占有率，而且为了存储这些信息，需要大量的存储介质空间。

(2) 聚合流输出

聚合流输出是指设备对与聚合关键项完全相同的流的统计信息进行汇总，从而得到对应的聚合流统计信息，并将该聚合统计信息发送到相应的 NetStream 服务器。

目前，聚合流输出支持的聚合方式如 表 1-1 所示。系统根据选择的聚合方式的聚合关键项，将聚合关键项相同的多条流的统计信息合并为一条聚合流的统计信息，记录该聚合流的统计信息。这些聚合方式相互独立，可以同时配置。

例如，设备采集到四条TCP流，其目的地址相同、源地址不同、源端口和目的端口均为 10。选择 表 1-1 中的“协议-端口聚合”方式，该聚合方式的依据为“协议号、源端口、目的端口”。因为这四条TCP流的源端口、目的端口和协议号相同，所以在聚合流统计表项中只会记录一条聚合流统计信息。设备只将聚合统计信息发送给相应的NetStream服务器。由此可见，聚合的最大好处是可以减少对网络带宽的占用。

表1-1 聚合流输出支持的聚合方式

聚合方式	聚合关键项
自治系统聚合（**as**）	源AS号、目的AS号、输入接口索引、输出接口索引
协议-端口聚合（**protocol-port**）	协议号、源端口、目的端口
源前缀聚合（**source-prefix**）	源AS号、源掩码长度（源IP的掩码长度）、源前缀（源IP的网络地址）、输入接口索引
目的前缀聚合（**destination-prefix**）	目的AS号、目的掩码长度（目的IP的掩码长度）、目的前缀（目的IP的网络地址）、输出接口索引
源和目的前缀聚合（**prefix**）	源AS号、目的AS号、源掩码长度、目的掩码长度、源前缀、目的前缀、输入接口索引、输出接口索引
前缀-端口聚合（**prefix-port**）	源前缀、目的前缀、源掩码长度、目的掩码长度、ToS、协议号、源端口、目的端口、输入接口索引、输出接口索引
服务类型-自治系统聚合（**tos-as**）	ToS、源AS号、目的AS号、输入接口索引、输出接口索引
服务类型-源前缀聚合（**tos-source-prefix**）	ToS、源AS号、源前缀、源掩码长度、输入接口索引
服务类型-目的前缀聚合（**tos-destination-prefix**）	ToS、目的AS号、目的掩码长度、目的前缀、输出接口索引

聚合方式	聚合关键项
服务类型-前缀聚合（**tos-prefix**）	ToS、源AS号、源前缀、源掩码长度、目的AS号、目的掩码长度、目的前缀、输入接口索引、输出接口索引
服务类型-协议-端口聚合（**tos-protocol-port**）	ToS、协议类型、源端口、目的端口、输入接口索引、输出接口索引
服务类型-BGP下一跳聚合（**tos-bgp-nexthop**）	ToS、BGP下一跳地址、输出接口索引

说明

- 在统计 AS 号时，如果流量没有按照 BGP 的路由表进行转发，则系统无法统计出 AS 号。
- 在统计 BGP 下一跳地址时，如果流量没有按照 BGP 的路由表进行转发，则系统无法统计出 BGP 下一跳地址。

3. 输出报文的版本

目前 NetStream 输出的报文主要有 5、8、9 三个版本。

- 版本 5：根据七元组产生原始的数据流，不支持聚合流输出，报文格式固定，不易扩展。
- 版本 8：支持聚合流输出，报文格式固定，不易扩展。
- 版本 9：基于模板方式，模板可在遵循 RFC 定义的模板格式的前提下自定义。版本 9 支持聚合流输出，及对 BGP 下一跳信息、MPLS 报文的统计输出。

1.1.4 NetStream过滤采样功能

1. NetStream过滤

NetStream 可以与 ACL（Access Control List，访问控制列表）配合使用，NetStream 只统计 ACL 筛选出的报文。通过这种方式可以使 NetStream 只对用户关注的数据进行统计，更能满足用户多样的统计要求。有关 ACL 的详细介绍，请参见“ACL 和 QoS 配置指导”中的“ACL”。

2. NetStream采样

NetStream 可以与 Sampler（采样器）配合使用。通过设定适当的采样间隔，不但减少了统计的报文数量，也可以保证收集到的统计信息基本正确地反映整个网络流的状况。另外，采样还可以减小网络的流量，避免网络中的大流量对设备转发性能造成影响。有关 Sampler 的详细介绍，请参见“网络管理和监控配置指导”中的“Sampler”。

1.2 NetStream配置任务简介

在配置 NetStream 过程中，请根据实际需求选择相应的配置步骤：

- 明确需要在网络环境中的哪台设备上开启 NetStream 功能。
- 如果网络上有各种业务流，可以考虑使用 ACL 筛选出需要统计的特定数据。
- 如果网络上的流量很大，可以考虑对数据流进行采样。
- 确定采用的输出报文版本，对 NetStream 统计输出报文的属性进行设置。

- 根据实际网络情况和需求，配置 NetStream 流老化功能。
- 如果统计输出的报文过多，可以配置聚合输出统计信息，避免重复的流输出信息占用网络带宽。

具体的配置步骤可以参考 图 1-2。

图1-2 NetStream 配置步骤流程图

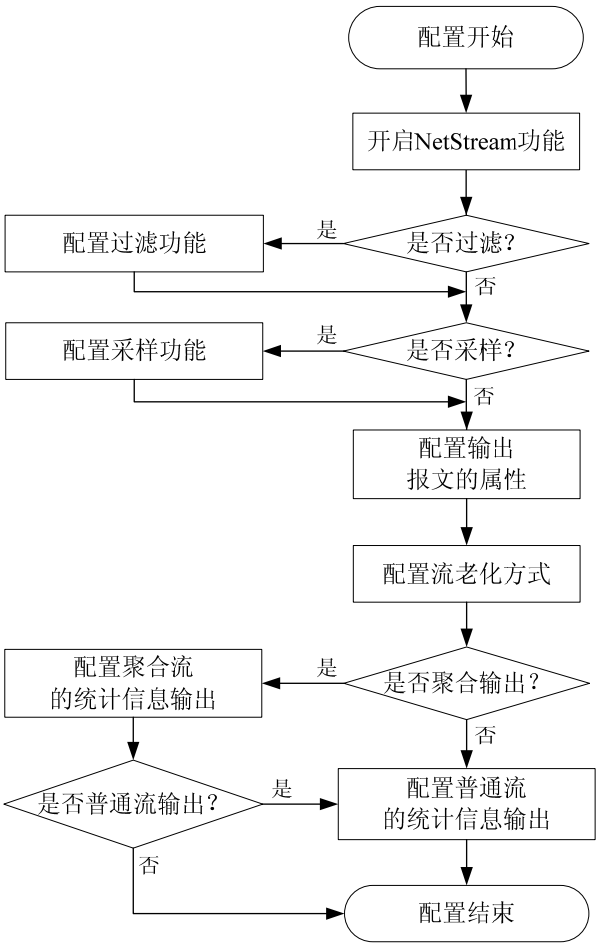


表1-2 NetStream 配置任务简介

配置任务		说明	详细配置
开启NetStream功能		必选	1.3.1
配置NetStream过滤功能		可选	1.3.2
配置NetStream采样功能		可选	1.3.3
配置NetStream输出报文的属性		可选	1.3.4
配置NetStream的流老化		可选	1.3.5
配置NetStream统计信息的输出	配置NetStream普通流的统计信息输出	二者至少选其一	1.3.6 1.
	配置NetStream聚合流的统计信息输出		1.3.6 2.

1.3 NetStream配置

1.3.1 开启NetStream功能

表1-3 开启接口的 NetStream 功能

操作	命令	说明
进入系统视图	**system-view**	-
进入接口视图	**interface** *interface-type interface-number*	-
开启接口的NetStream功能	**ip netstream** { **inbound** | **outbound** }	缺省情况下，接口的NetStream功能处于关闭状态

1.3.2 配置NetStream过滤功能

配置 NetStream 过滤功能时，需要注意：

- 如果在设备上同时配置过滤和采样，设备会先过滤后采样报文。
- 过滤功能对 MPLS 报文无效。

表1-4 配置 NetStream 过滤功能

操作	命令	说明
进入系统视图	**system-view**	-
进入接口视图	**interface** *interface-type interface-number*	-
配置NetStream过滤功能，根据指定ACL规则对报文进行过滤	**ip netstream filter acl** *acl-number* { **inbound** | **outbound** }	缺省情况下，未配置NetStream过滤功能，此时统计所有IPv4报文

1.3.3 配置NetStream采样功能

表1-5 配置 NetStream 的采样功能

操作	命令	说明
进入系统视图	**system-view**	-
创建采样器	**sampler** *sampler-name* **mode** { **fixed** | **random** } **packet-interval** *rate*	关于采样器的详细介绍请参见“网络管理和监控配置指导”中的“Sampler”
进入接口视图	**interface** *interface-type interface-number*	-

操作	命令	说明
启用NetStream采样功能	**ip netstream sampler** *sampler-name* { **inbound** | **outbound** }	缺省情况下，未启用NetStream采样功能

1.3.4 配置NetStream输出报文的属性

1. 配置NetStream输出报文的格式

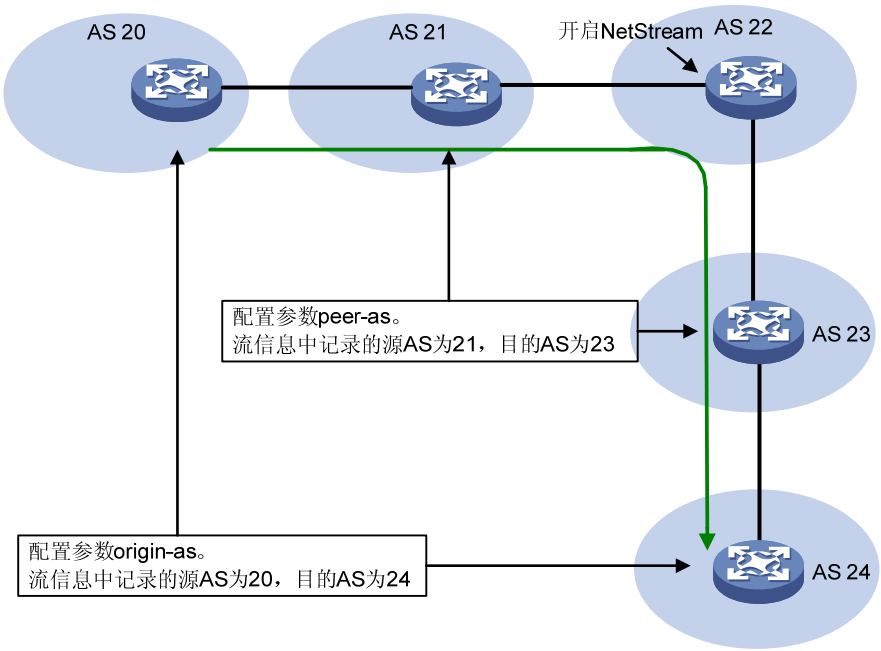
用户可以通过配置 NetStream 输出报文的格式，进一步明确需要统计的选项（如确定记录的自治系统号及是否记录 BGP 下一跳地址）。

NetStream 流统计信息中会记录流的源 IP 地址及其对应的自治系统号、目的 IP 地址及其对应的自治系统号。设备会根据用户配置的自治系统参数来确定记录的自治系统号。自治系统参数包括 **origin-as**（起始自治系统）和 **peer-as**（邻接自治系统）：

- **origin-as** 表示流统计信息中记录的自治系统号为起始自治系统号。
- **peer-as** 表示流统计信息中记录的自治系统号为邻接自治系统号。

如 图 1-3 所示，有一条数据流从AS 20 开始，依次经过AS 21、AS 22、AS 23，到达AS 24。如果配置参数**origin-as**，那么流统计信息中记录该流的源AS为 20，目的AS为 24。如果配置参数**peer-as**，那么流统计信息中记录该流的源AS为 21，目的AS为 23。

图1-3 自治系统参数示意图



当用户选择版本 9 的 NetStream 输出报文格式时，用户可以选择是否在流统计信息中记录 BGP 下一跳地址。

表1-6 配置 NetStream 输出报文的格式

操作	命令	说明
进入系统视图	**system-view**	-
（可选）配置NetStream统计输出报文版本以及其自治系统选项、BGP下一跳选项	**ip netstream export version 5** [**origin-as** | **peer-as**]	二者选其一 缺省情况下，IPv4流统计信息通过版本9的NetStream报文发送，MPLS流信息不输出。自治系统选项使用邻接自治系统号（**peer-as**），流统计信息中不记录BGP下一跳地址
	ip netstream export version 9 [**origin-as** | **peer-as**] [**bgp-nexthop**]	

2. 配置NetStream输出报文版本 9 模板的刷新率

V9 版本是基于模板方式的、支持自定义格式的输出报文版本（即可以决定输出报文的内容）。由于 NetStream 服务器不会永久保存模板，所以设备需要定期通知 NetStream 服务器最新的 V9 模板格式。用户可以根据实际情况，配置版本 9 模板的刷新率（包括包刷新率和时间刷新率），及时更新模板。当同时配置包刷新率和时间刷新率时，只要满足任意一个刷新条件，设备就会将激活的模板发送给 NetStream 服务器。

表1-7 配置 NetStream 输出报文版本 9 模板的刷新率

操作	命令	说明
进入系统视图	**system-view**	-
（可选）配置NetStream统计输出报文版本9模板的包刷新率	**ip netstream export v9-template refresh-rate packet** *packets*	缺省情况下，每隔20个包发送一次版本9模板
（可选）配置NetStream统计输出报文版本9模板的时间刷新率	**ip netstream export v9-template refresh-rate time** *minutes*	缺省情况下，每隔30分钟发送一次版本9模板

3. 配置NetStream的MPLS报文统计功能

开启 NetStream 的 MPLS 报文统计功能后，NetStream 将七元组以及 MPLS 标签相同的报文视为同一条流，统计流信息中的 MPLS 报文数据，记录栈顶标签的类型（即分配该标签的协议）、栈顶标签对应的 IP 地址及掩码长度（即申请该标签的 FEC 信息），并可以配置是否统计 IP 数据内容。

表1-8 配置 NetStream MPLS 报文统计功能

操作	命令	说明
进入系统视图	**system-view**	-
开启MPLS报文统计功能	**ip netstream mpls** [**label-positions** *label-position1* [*label-position2* [*label-position3*]]] [**no-ip-fields**]	缺省情况下，未开启MPLS报文统计功能

1.3.5 配置NetStream的流老化

NetStream 流老化有以下三种机制：

- 按时老化
- 强制老化
- TCP 的 FIN 和 RST 报文触发老化（该形式的老化在 TCP 连接拆除时自动进行）

(1) 按时老化

按时老化分为以下两种方式：

- 流的不活跃老化：从采集到的最后一个报文开始，该流在 **ip netstream timeout inactive** 命令指定的时间内没有被采集到（即在设定的 **inactive** 时长内统计到的该流统计信息没有增加），那么设备会向 NetStream 服务器输出该流的统计信息，这种老化称为流的不活跃老化。通过这种老化，可以清除设备上 NetStream 缓冲区中的无用表项（此时使用 **display ip netstream cache** 命令无法看到这条老化的流），充分利用统计表项资源。
- 流的活跃老化：从采集到的第一个报文开始，该流在 **ip netstream timeout active** 命令指定的时间内能被采集到。活跃时间超过设定的 **active** 时长后，需要输出该流的统计信息，这种老化称为流的活跃老化。设备向 NetStream 服务器输出流的统计信息后，因为该流还存在，所以设备会继续统计该流（此时使用 **display ip netstream cache** 命令可以看到这条流的统计表项）。这种老化方式是设备定期向 NetStream 服务器输出流统计信息的一种机制。

(2) 强制老化

用户可以执行 **reset ip netstream statistics** 命令强制将 NetStream 缓冲区中所有流老化、输出，并清空 NetStream 缓冲区信息；或者根据实际需要，使用 **ip netstream max-entry** 命令配置 NetStream 流缓存区中流表项的最大数目，以及当达到 NetStream 流缓存区中流表项的最大数目时的处理方式。

(3) TCP 的 FIN 和 RST 报文触发老化

对于 TCP 连接，当收到标志为 FIN 或 RST 的报文时，表示一次会话结束。因此当一条已经存在的 TCP 协议 NetStream 流中流过一个标志为 FIN 或 RST 的报文时，可以立即老化、输出相应的 NetStream 流，并清除该 NetStream 流。但是假如一条流的第一个报文就是 TCP 的 FIN 或 RST 报文，则会按正常的流程创建一条新流，不进行老化。这种方式在设备上始终开启，不能通过执行命令开启或关闭该老化方式。

表1-9 配置 NetStream 的流老化

操作		命令	说明
进入系统视图		**system-view**	-
配置按时老化	（可选）配置流的活跃老化时间	**ip netstream timeout active** *minutes*	缺省情况下，流的活跃老化时间为30分钟
	（可选）配置流的不活跃老化时间	**ip netstream timeout inactive** *seconds*	缺省情况下，流的不活跃老化时间为30秒
配置强制老化	（可选）配置NetStream流缓存区中流表项的最大数目	**ip netstream max-entry** *max-entries*	缺省情况下，NetStream流缓存区中流表项的最大数目为10000
	退回用户视图	**quit**	-

操作		命令	说明
	（可选）将流缓存区中所有流强制老化，并清除NetStream缓冲区的状态信息和输出报文信息	**reset ip netstream statistics**	-

1.3.6 配置NetStream统计信息的输出

1. 配置NetStream普通流的统计信息输出

表1-10 配置 NetStream 普通流的统计信息输出

操作	命令	说明
进入系统视图	**system-view**	-
配置NetStream普通流统计信息输出的目的地址和目的UDP端口号	**ip netstream export host** *ip-address udp-port* [**vpn-instance** *vpn-instance-name*]	缺省情况下，系统视图下没有配置目的地址和目的UDP端口号
（可选）配置NetStream统计输出报文的源接口	**ip netstream export source interface** *interface-type interface-number*	缺省情况下，采用统计输出报文的出接口（即与服务器相连的接口）作为源接口
（可选）配置输出速率限制	**ip netstream export rate** *rate*	缺省情况下，NetStream统计输出报文的输出速率不受限制

2. 配置NetStream聚合流的统计信息输出

配置 NetStream 聚合流的统计信息输出时，需要注意：

- 在聚合视图下，用户配置的 NetStream 统计输出报文的输出属性，仅对聚合报文生效；而系统视图下的配置对普通报文生效，并且当聚合视图下没有配置以上属性时，也会对聚合报文生效。
- 如果设备上配置了聚合，并配置输出报文版本为 V5，则聚合流统计信息采用 V8 版本输出。

表1-11 配置 NetStream 聚合流的统计信息输出

操作	命令	说明
进入系统视图	**system-view**	-
进入NetStream聚合视图	**ip netstream aggregation** { **as** | **destination-prefix** | **prefix** | **prefix-port** | **protocol-port** | **source-prefix** | **tos-as** | **tos-bgp-nexthop** | **tos-destination-prefix** | **tos-prefix** | **tos-protocol-port** | **tos-source-prefix** }	-
配置NetStream聚合流统计信息输出的目的地址和目的UDP端口号	**ip netstream export host** *ip-addrress udp-port* [**vpn-instance** *vpn-instance-name*]	缺省情况下，聚合视图下没有配置目的地址和目的UDP端口号。 为了减少对网络带宽的占用，可以只在聚合视图下配置本命令，此时设备只会输出聚合流信息

操作	命令	说明
（可选）配置NetStream聚合统计信息输出的源接口	**ip netstream export source interface** *interface-type interface-number*	缺省情况下，采用统计输出报文的出接口（即与服务器相连的接口）作为源接口 需要注意的是： • 不同聚合视图下可以配置不同的源接口 • 聚合视图下若没有配置源接口，则使用系统视图下的配置
开启当前聚合视图对应的聚合功能	**enable**	缺省情况下，未开启任何NetStream聚合功能

1.4 NetStream显示和维护

在完成上述配置后，在任意视图下执行 **display** 命令可以显示配置后 NetStream 的运行情况，通过查看显示信息验证配置的效果。

在用户视图下执行 **reset** 命令可以清除 NetStream 的统计信息。

表1-12 NetStream 显示和维护

操作	命令
查看NetStream流缓存区的配置和状态信息（MSR 2600/MSR 3600）	**display ip netstream cache** [**verbose**]
查看NetStream流缓存区的配置和状态信息（MSR 5600）	**display ip netstream cache** [**slot** *slot-number*] [**verbose**]
查看NetStream统计输出报文信息	**display ip netstream export**
查看NetStream模板的配置和状态信息（MSR 2600/MSR 3600）	**display ip netstream template**
查看NetStream模板的配置和状态信息（MSR 5600）	**display ip netstream template** [**slot** *slot-number*]
将流缓存区中所有流强制老化，输出报文信息，并清空NetStream缓冲区的状态信息	**reset ip netstream statistics**

1.5 NetStream典型配置举例

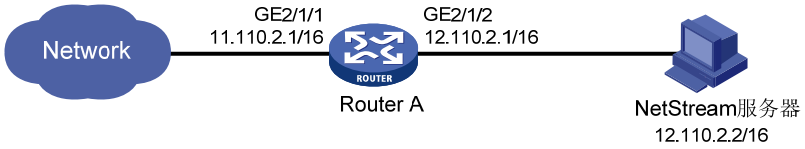
1.5.1 NetStream普通流的统计信息输出配置举例

1. 组网需求

如 图 1-4 所示，在Router A上启动NetStream功能。要求在GigabitEthernet2/1/1 上配置NetStream入统计，GigabitEthernet2/1/2 上配置NetStream出统计，并将NetStream普通流的统计信息输出到NetStream服务器。NetStream服务器的IP地址为 12.110.2.2/16，UDP端口号为 5000。

2. 组网图

图1-4 NetStream 普通流的统计信息输出配置组网图



3. 配置步骤

在 GigabitEthernet2/1/1 上启动 NetStream 入统计。

```
<RouterA> system-view
[RouterA] interface gigabitethernet 2/1/1
[RouterA-GigabitEthernet2/1/1] ip address 11.110.2.1 255.255.0.0
[RouterA-GigabitEthernet2/1/1] ip netstream inbound
[RouterA-GigabitEthernet2/1/1] quit
```

在 GigabitEthernet2/1/2 上启动 NetStream 出统计。

```
[RouterA] interface gigabitethernet 2/1/2
[RouterA-GigabitEthernet2/1/2] ip address 12.110.2.1 255.255.0.0
[RouterA-GigabitEthernet2/1/2] ip netstream outbound
[RouterA-GigabitEthernet2/1/2] quit
```

配置 NetStream 普通流统计信息输出的目的地址为 12.110.2.2 和目的 UDP 端口号为 5000。

```
[RouterA] ip netstream export host 12.110.2.2 5000
```

4. 配置验证

设备运行一段时间后，查看 NetStream 普通流的统计信息。

查看 NetStream 流缓冲区信息。

```
[RouterA] display ip netstream cache
IP NetStream cache information:
  Active flow timeout             : 30 min
  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   12.110.2.2 (0)         11.111.2.2 (0)         1   0   GE2/1/1(I)  5
IP   12.110.2.2 (0)         11.111.2.2 (0)         1   0   GE2/1/2(O)  5
```

查看 NetStream 统计输出报文的各种信息。

```
[RouterA] 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 flows number                : 0
  Version 5 exported UDP datagrams number (failed): 0 (0)
  Version 9 exported flows number                : 10
  Version 9 exported UDP datagrams number (failed): 10 (0)
```

1.5.2 NetStream聚合流的统计信息输出配置举例

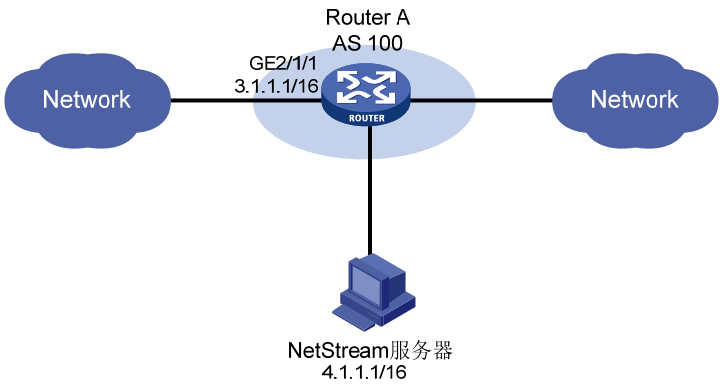
1. 组网需求

在 Router A 上配置 NetStream，具体要求为：

- 普通流的统计信息使用版本 5 格式输出到 NetStream 服务器。NetStream 服务器的 IP 地址为 4.1.1.1/16，UDP 端口号为 5000；
- 使用版本 8 格式对 5 种聚合流（**as**、**protocol-port**、**source-prefix**、**destination-prefix** 和 **prefix**）进行统计，并将各聚合流分别输出到该 NetStream 服务器的 2000、3000、4000、6000 和 7000 端口。

2. 组网图

图1-5 NetStream 聚合流的统计信息输出配置组网图



3. 配置步骤

网络之间均运行 EBGP 路由协议。相关配置请参见“三层技术-IP 路由配置指导”中的“BGP”。

在 GigabitEthernet2/1/1 上开启 NetStream 功能。

```
<RouterA> system-view
[RouterA] interface gigabitethernet 2/1/1
[RouterA-GigabitEthernet2/1/1] ip address 3.1.1.1 255.255.0.0
[RouterA-GigabitEthernet2/1/1] ip netstream inbound
[RouterA-GigabitEthernet2/1/1] ip netstream outbound
[RouterA-GigabitEthernet2/1/1] quit
```

配置普通流统计信息输出的目的地址为 4.1.1.1 和目的 UDP 端口号为 5000。

```
[RouterA] ip netstream export host 4.1.1.1 5000
```

配置自治系统聚合模式，以及该聚合流统计信息输出的目的地址为 4.1.1.1 和目的 UDP 端口号为 2000。

```
[RouterA] ip netstream aggregation as
[RouterA-ns-aggregation-as] enable
[RouterA-ns-aggregation-as] ip netstream export host 4.1.1.1 2000
[RouterA-ns-aggregation-as] quit
```

配置协议－端口聚合模式，以及该聚合流统计信息输出的目的地址为 4.1.1.1 和目的 UDP 端口号为 3000。

```
[RouterA] ip netstream aggregation protocol-port
[RouterA-ns-aggregation-protport] enable
[RouterA-ns-aggregation-protport] ip netstream export host 4.1.1.1 3000
[RouterA-ns-aggregation-protport] quit
```

配置源前缀聚合模式，以及该聚合流统计信息输出的目的地址为 4.1.1.1 和目的 UDP 端口号为 4000。

```
[RouterA] ip netstream aggregation source-prefix
[RouterA-ns-aggregation-srcpre] enable
[RouterA-ns-aggregation-srcpre] ip netstream export host 4.1.1.1 4000
[RouterA-ns-aggregation-srcpre] quit
```

配置目的前缀聚合模式，以及该聚合流统计信息输出的目的地址为 4.1.1.1 和目的 UDP 端口号为 6000。

```
[RouterA] ip netstream aggregation destination-prefix
[RouterA-ns-aggregation-dstpre] enable
[RouterA-ns-aggregation-dstpre] ip netstream export host 4.1.1.1 6000
[RouterA-ns-aggregation-dstpre] quit
```

配置前缀聚合模式，以及该聚合流统计信息输出的目的地址为 4.1.1.1 和目的 UDP 端口号为 7000。

```
[RouterA] ip netstream aggregation prefix
[RouterA-ns-aggregation-prefix] enable
[RouterA-ns-aggregation-prefix] ip netstream export host 4.1.1.1 7000
[RouterA-ns-aggregation-prefix] quit
```

4. 配置验证

设备运行一段时间后，查看 NetStream 聚合流的统计信息。

查看 NetStream 流缓冲区信息。

```
[RouterA] display ip netstream cache
IP NetStream cache information:
  Active flow timeout             : 30 min
  Inactive flow timeout           : 10 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   3.1.1.1(0)             3.1.1.2 (0)            1   0   GE2/1/1(I)  5
IP   3.1.1.2 (0)            3.1.1.1 (0)            1   0   GE2/1/1(O)  5
```

查看 NetStream 统计输出报文的各种信息。

```
[RouterA] display ip netstream export
AS aggregation export information:
  Flow source interface                          : Not specified
  Flow destination VPN instance                  : Not specified
  Flow destination IP address (UDP)              : 4.1.1.1 (2000)
  Version 8 exported flows number                : 2
  Version 8 exported UDP datagrams number (failed): 2 (0)
  Version 9 exported flows number                : 0
  Version 9 exported UDP datagrams 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)              : 4.1.1.1 (3000)
  Version 8 exported flows number                : 2
  Version 8 exported UDP datagrams number (failed): 2 (0)
  Version 9 exported flows number                : 0
  Version 9 exported UDP datagrams 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 flows number                : 2
  Version 8 exported UDP datagrams number (failed): 2 (0)
  Version 9 exported flows number                : 0
  Version 9 exported UDP datagrams 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 flows number                : 2
  Version 8 exported UDP datagrams number (failed): 2 (0)
  Version 9 exported flows number                : 0
  Version 9 exported UDP datagrams 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 flows number                : 2
  Version 8 exported UDP datagrams number (failed): 2 (0)
  Version 9 exported flows number                : 0
  Version 9 exported UDP datagrams 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 flows number                 : 10
Version 5 exported UDP datagrams number (failed): 10 (0)
Version 9 exported flows number                 : 0
Version 9 exported UDP datagrams number (failed): 0 (0)
```

目 录

1 IPv6 NetStream

1.1 IPv6 NetStream 简介

1.1.1 IPv6 NetStream技术应用背景

Internet 的高速发展为用户提供了更高的带宽，支持的业务和应用日渐增多，传统流量统计如 SNMP、端口镜像等，由于统计流量方式不灵活或是需要投资专用服务器成本高等原因，无法满足对网络进行更细致的管理，需要一种新技术来更好的支持网络流量统计。

NetStream 技术是一种基于网络流信息的统计技术，可以对网络中的业务流量情况进行统计和分析。在网络的接入层、汇聚层、核心层上，都可以部署 NetStream。

NetStream 技术的应用有以下几种：

- 计费：NetStream 为基于资源（如线路、带宽、时段等）占用情况的计费提供了精细的数据。ISP（Internet Service Provider，互联网服务提供商）可以利用这些信息来实行灵活的计费策略，如基于时间、带宽、应用、服务质量等。企业客户可以使用这些信息计算部门费用或分配成本，以便有效利用资源。
- 网络规划：NetStream 可以为网络管理工具提供关键信息，比如各个 AS 域之间的网络流量情况，以便优化网络设计和规划，实现以最小的网络运营成本达到最佳的网络性能和可靠性。
- 网络监控：通过在出口部署 NetStream，对连接 Internet 网络的接口进行实时的流量监控，可以分析各种业务占用出口带宽的情况。网管人员可以根据这些信息判断网络的运行情况，尽早发现不合理的网络结构或是网络中的性能瓶颈，方便网管人员规划和分配网络资源。
- 用户监控和分析：通过 NetStream 技术可以使网络管理者轻松获取用户使用网络和应用资源的详细情况，进而高效地规划以及分配网络资源，并保障网络的安全运行。

1.1.2 IPv6 NetStream的基本概念

1. IPv6 NetStream流

IPv6 NetStream 是一项基于“流”来提供报文统计信息的技术。它根据 IPv6 报文的目的 IP 地址、源 IP 地址、目的端口号、源端口号、协议号、流量分类、流标签、输入接口或输出接口来定义流，具有相同的八元组的报文标识为同一条流。

2. IPv6 NetStream系统组成

一个典型的 IPv6 NetStream 系统由 NDE（NetStream Data Exporter，网络流数据输出者）、NSC（NetStream Collector，网络流数据收集者）和 NDA（NetStream Data Analyzer，网络流数据分析者）三部分组成。

- NDE

NDE 根据八元组对网络流进行分类，提取符合条件的流进行统计，并将统计信息输出给 NSC 设备。输出前也可对数据进行一些处理，比如聚合。配置了 IPv6 NetStream 功能的设备在 IPv6 NetStream 系统中担当 NDE 角色。

- NSC

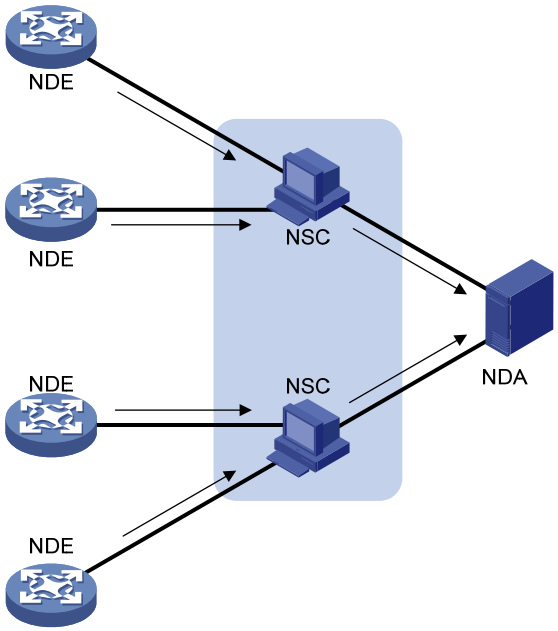
NSC 通常为运行于 Unix 或者 Windows 上的一个应用程序，负责解析来自 NDE 的报文，把统计数据收集到数据库中，可供 NDA 进行解析。NSC 可以采集多个 NDE 设备输出的数据。

- NDA

NDA 是一个网络流量分析工具，它从 NSC 中提取统计数据，进行进一步的加工处理，生成报表，为各种业务提供依据（比如流量计费、网络规划，攻击监测）。NDA 可以提取多个 NSC 中的数据。通常，NDA 具有图像化用户界面，可以使用户方便地获取、显示和分析收集到的数据。

NSC 和 NDA 可以集成在一台 NetStream 服务器上。

图1-1 IPv6 NetStream 系统中的设备角色



1.1.3 IPv6 NetStream工作机制

如 图 1-1 所示，IPv6 NetStream进行数据采集和分析的过程如下：

(1) 配置了 IPv6 NetStream 功能的设备（即 NDE）把采集到的关于流的详细信息定期发送给 NSC；

(2) 信息由 NSC 初步处理后发送给 NDA；

(3) NDA 对数据进行分析，以用于计费、网络规划等应用。

由于设备在 NetStream 系统中担任 NDE 角色，所以本文重点介绍 NDE 的实现以及配置。

NetStream 工作机制中有如下几项关键技术：

1. 流老化

IPv6 NetStream流老化是设备向NetStream服务器输出流统计信息的一种手段。当设备启用IPv6 NetStream功能后，流统计信息首先会被存储在设备的IPv6 NetStream缓冲区中。当存储在设备上的IPv6 NetStream流统计信息老化后，设备会把缓冲区中的流统计信息通过指定版本的IPv6 NetStream输出报文发送给NetStream服务器，同时清除缓冲区中的对应信息。流老化的三种方式及其配置请参见“1.3.5 配置IPv6 NetStream的流老化”。

2. 流输出

(1) 普通流输出

普通流输出是指所有流的统计信息都要被统计。在流老化后，每条流的统计信息都要输出到NetStream服务器。

普通流输出的优点是NetStream服务器可以得到每条流的详细统计信息。但是缺点也是很明显的，这种方式增加了网络带宽和设备的CPU占有率，而且为了存储这些信息，需要大量的存储介质空间，而很多情况下，用户并不需要获取所有流的统计信息。

(2) 聚合流输出

聚合流输出是指设备对与聚合关键项完全相同的流统计信息进行汇总，从而得到对应的聚合流统计信息，并且将该聚合统计信息发送到相应的接收聚合统计信息的NetStream服务器。聚合的最大好处是可以减少对网络带宽的占用。

目前，聚合流输出支持的聚合方式如表 1-1所示。系统根据聚合方式的聚合关键项，将聚合关键项相同的多条流统计信息合并为一条聚合流的统计信息，对应一条聚合记录。每种聚合方式相互独立，可以同时配置。

表1-1 IPv6 NetStream聚合流输出支持的6种方式

聚合方式	聚合关键项
自治系统聚合（as）	源AS号、目的AS号、输入接口索引、输出接口索引
协议-端口聚合（protocol-port）	协议号、源端口、目的端口
源前缀聚合（prefix）	源AS号、源掩码长度（源IP的掩码长度）、源前缀（源IP的网络地址）、输入接口索引
目的前缀聚合（destination-prefix）	目的AS号、目的掩码长度、目的前缀、输出接口索引
源和目的前缀聚合（source-prefix）	源AS号、目的AS号、源掩码长度（源IP的掩码长度）、目的掩码长度、源前缀（源IP的网络地址）、目的前缀、输入接口索引、输出接口索引
BGP下一跳聚合（bgp-nexthop）	BGP下一跳地址、输出接口索引

说明

- 在统计AS域号时，如果流量没有按照BGP的路由表进行转发，则系统无法统计出AS域号。
- 在统计BGP下一跳地址时，如果流量没有按照BGP的路由表进行转发，则系统无法统计出BGP下一跳地址。

3. 输出报文的版本格式

目前IPv6 NetStream输出的统计信息只能使用版本9格式。版本9是一种基于模板方式的版本格式，使统计信息的输出更为灵活，可以用来灵活输出各种组合格式的数据，支持对BGP下一跳信息、MPLS报文的统计输出。

1.1.4 IPv6 NetStream过滤采样功能

1. IPv6 NetStream过滤

IPv6 NetStream 可以与 ACL（Access Control List，访问控制列表）配合使用，IPv6 NetStream 只统计符合ACL筛选出的报文。通过这种方式可以使IPv6 NetStream只对用户关注的数据进行统计，更能满足用户多样的统计要求。有关 ACL 的详细介绍，请参见“ACL 和 QoS 配置指导”中的“ACL”。

2. IPv6 NetStream采样

IPv6 NetStream 可以与 Sampler（采样器）配合使用。通过设定适当的采样间隔，不但减少了统计的报文数量，也可以保证收集到的统计信息基本正确地反映整个网络流的状况。另外，采样还可以减小对设备转发性能造成的影响。

1.2 IPv6 NetStream配置任务简介

在配置 IPv6 NetStream 过程中，请根据实际需求选择相应的配置步骤：

- 明确需要在网络环境中的哪台设备上开启 IPv6 NetStream 功能。
- 如果网络上有各种业务流，可以考虑使用 ACL 筛选出需要统计的特定数据。
- 如果网络上的流量很大，可以考虑对数据流进行采样。
- 对 IPv6 NetStream 统计输出报文的属性进行设置。
- 根据实际网络情况和需求，配置 IPv6 NetStream 流老化功能。
- 如果统计输出的报文过多，可以配置聚合输出统计信息，避免重复的流输出信息占用网络带宽。

具体配置步骤可以参考 图 1-2。

图1-2 IPv6 NetStream 配置步骤流程图

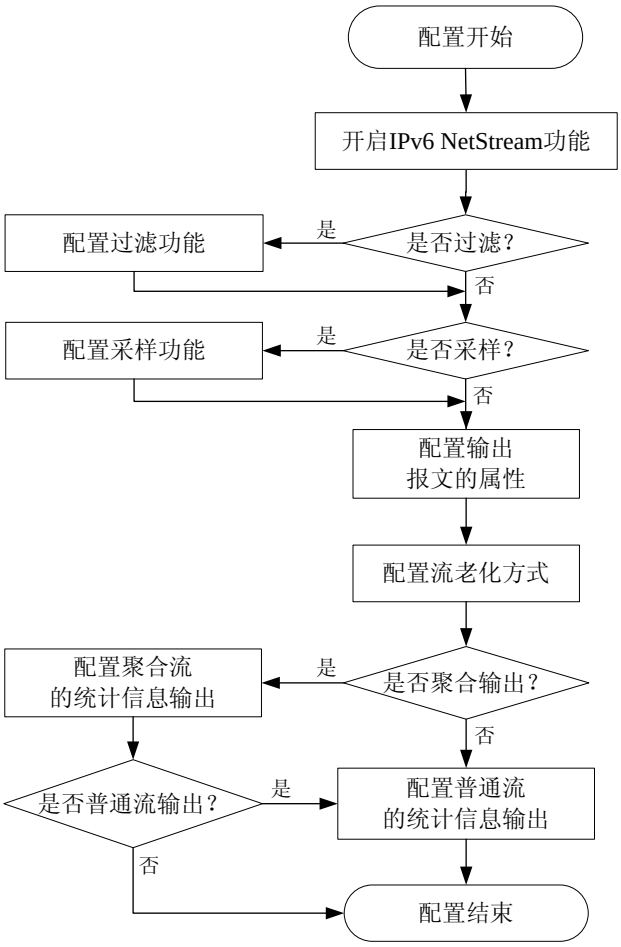


表1-2 IPv6 NetStream 配置任务简介

配置任务		说明	详细配置
开启IPv6 NetStream功能		必选	1.3.1
配置IPv6 NetStream过滤功能		可选	1.3.2
配置IPv6 NetStream采样功能		可选	1.3.3
配置IPv6 NetStream输出报文的属性		可选	1.3.4
配置IPv6 NetStream的流老化		可选	1.3.5
配置IPv6 NetStream统计信息的输出	配置IPv6 NetStream普通流的统计信息输出	二者至少选其一	1.3.6 1.
	配置IPv6 NetStream聚合流的统计信息输出		1.3.6 2.

1.3 IPv6 NetStream配置

1.3.1 开启IPv6 NetStream功能

表1-3 开启接口下 IPv6 NetStream 功能

操作	命令	说明
进入系统视图	**system-view**	-
进入接口视图	**interface** *interface-type interface-number*	-
开启接口的IPv6 NetStream功能	**ipv6 netstream** { **inbound** | **outbound** }	缺省情况下，IPv6 NetStream功能处于关闭状态

1.3.2 配置IPv6 NetStream过滤功能

配置 IPv6 NetStream 过滤功能时，需要注意：

- 过滤功能对 MPLS 报文无效。
- 如果在设备上同时配置过滤和采样，设备会先过滤后采样报文。

表1-4 配置 IPv6 NetStream 过滤功能

操作	命令	说明
进入系统视图	**system-view**	–
进入接口视图	**interface** *interface-type interface-number*	–
配置根据指定ACL规则对过滤后的流进行IPv6 NetStream统计	**ipv6 netstream filter acl** *acl-number* { **inbound** | **outbound** }	缺省情况下，未配置过滤功能，此时统计所有IPv6报文

1.3.3 配置IPv6 NetStream采样功能

表1-5 配置 NetStream 的采样功能

操作	命令	说明
进入系统视图	**system-view**	-
创建采样器	**sampler** *sampler-name* **mode** { **fixed** | **random** } **packet-interval** *rate*	关于采样器的详细介绍请参见“网络管理和监控配置指导”中的“Sampler”
进入接口视图	**interface** *interface-type interface-number*	-
启用IPv6 NetStream采样功能	**ipv6 netstream sampler** *sampler-name* { **inbound** | **outbound** }	缺省情况下，IPv6 NetStream统计所有IPv6报文，未用采样功能

1.3.4 配置IPv6 NetStream输出报文的属性

1. 配置输出报文格式

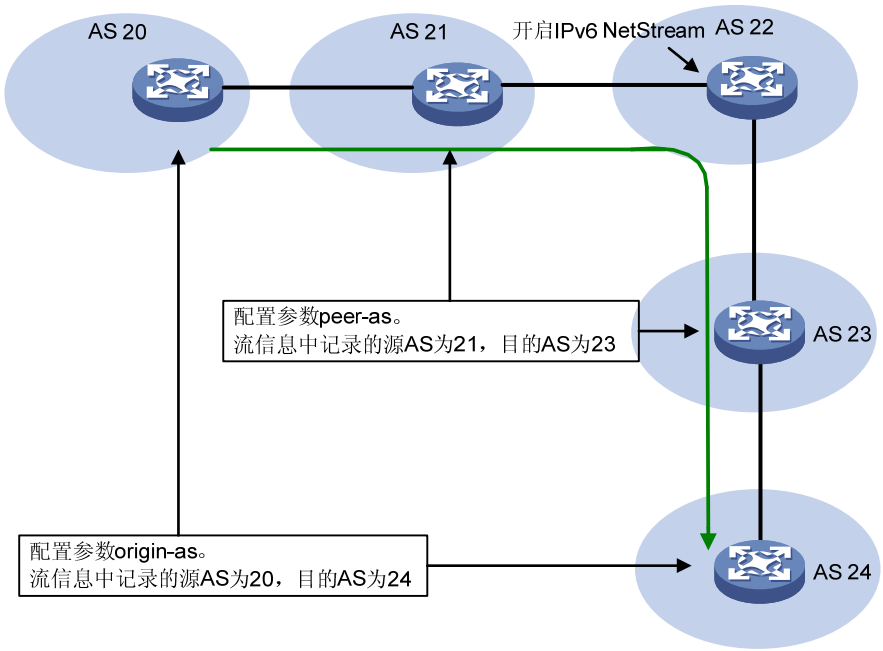
用户可以通过配置 IPv6 NetStream 输出报文的格式，进一步明确需要统计的选项（如确定记录的自治系统号及是否记录 BGP 下一跳地址）。

IPv6 NetStream 流统计信息中会记录流的源 IP 地址及其对应的自治系统号、目的 IP 地址及其对应的自治系统号。设备会根据用户配置的自治系统参数来确定记录的自治系统号。自治系统参数包括 **origin-as**（起始自治系统）和 **peer-as**（邻接自治系统）：

- **origin-as** 表示流统计信息中记录的自治系统号为起始自治系统号。
- **peer-as** 表示流统计信息中记录的自治系统号为邻接自治系统号。

如 图 1-3 所示，有一条数据流从AS 20 开始，依次经过AS 21、AS 22、AS 23，到达AS 24。如果配置参数**origin-as**，那么流统计信息中记录该流的源AS为 20，目的AS为 24。如果配置参数**peer-as**，那么流统计信息中记录该流的源AS为 21，目的AS为 23。

图1-3 自治系统参数示意图



IPv6 NetStream 统计输出报文还支持 BGP 下一跳选项，用户可以选择是否在流信息中记录 BGP 下一跳地址。

表1-6 配置 IPv6 NetStream 输出报文的属性

操作	命令	说明
进入系统视图	**system-view**	-

操作	命令	说明
（可选）配置IPv6 NetStream统计输出报文版本以及其自治系统选项、BGP下一跳选项	**ipv6 netstream export version 9** [**origin-as** | **peer-as**] [**bgp-nexthop**]	缺省情况下，IPv6普通流信息、IPv6聚合统计流信息和带IPv6选项信息的MPLS流信息都通过版本9的NetStream统计输出报文发送。流统计信息中记录邻接自治系统（**peer-as**），流信息中不记录BGP下一跳地址

2. 配置IPv6 NetStream输出报文版本 9 模板的刷新率

版本 9 是基于模板方式的、支持自定义格式的输出报文版本（即可以决定输出报文的内容）。由于 NetStream 服务器不会永久保存模板，所以设备需要定期通知 NetStream 服务器最新的版本 9 模板格式。用户可以根据实际情况，配置版本 9 模板的刷新率（包括包刷新率和时间刷新率），及时更新模板。当同时配置包刷新率和时间刷新率时，只要满足任意一个刷新条件，设备就会将激活的模板发送给 NetStream 服务器。

表1-7 配置 IPv6 NetStream 输出报文版本 9 模板的刷新率

操作	命令	说明
进入系统视图	**system-view**	-
（可选）配置IPv6 NetStream统计输出报文版本9模板的包刷新率	**ipv6 netstream export v9-template refresh-rate packet** *packets*	缺省情况下，每隔20个包发送一次版本9模板
（可选）配置IPv6 NetStream统计输出报文版本9模板的时间刷新率	**ipv6 netstream export v9-template refresh-rate time** *minutes*	缺省情况下，每隔30分钟发送一次版本9模板

3. 配置NetStream的MPLS报文统计功能

IPv6 NetStream 采集流（IPv6 NetStream 八元组以及 MPLS 标签相同的报文被视为同一条流）信息中 MPLS 报文数据，按照标签进行统计并记录标签栈顶标签的类型、标签栈顶标签对应的 IP 地址及掩码长度。在缺省情况下还会统计 IP 数据内容。

表1-8 配置 IPv6 NetStream MPLS 报文统计功能

操作	命令	说明
进入系统视图	**system-view**	-
开启MPLS报文统计功能	**ip netstream mpls** [**label-positions** *label-position1* [*label-position2* [*label-position3*]]] [**no-ip-fields**]	缺省情况下，未开启MPLS报文统计功能

1.3.5 配置IPv6 NetStream的流老化

1. 流老化的三种机制

IPv6 NetStream 流老化有以下三种机制：

- 按时老化

- 强制老化
- TCP 的 FIN 和 RST 报文触发老化（该形式的老化在 TCP 连接拆除时自动进行）

(1) 按时老化

按时老化分为以下两种方式：

- 流的不活跃老化：从采集到的最后一个报文开始，该流在 **ipv6 netstream timeout inactive** 指定的时间内没有被采集到报文（即在设定的 **inactive** 时长内统计到的该流的统计信息没有变化），那么设备会向 IPv6 NetStream 服务器输出该流的统计信息，这种老化称为流的不活跃老化。通过这种老化，可以清除设备上 IPv6 NetStream 缓冲区中的无用表项（此时使用 **display ipv6 netstream cache** 命令无法看到这条老化的流），充分利用统计表项资源。
- 流的活跃老化：从采集到的第一个报文开始，该流在 **ipv6 netstream timeout active** 指定的时间内能采集到新的报文。活跃时间超过设定的 **active** 时长后，需要输出该流的统计信息，这种老化称为流的活跃老化。设备向 IPv6 NetStream 服务器输出活跃流的统计信息后，因为该流实际上还存在，所以在设备会继续统计该流（此时使用 **display ipv6 netstream cache** 可以看到这条流的统计表项）。这种老化方式是设备定期向 NetStream 服务器输出活跃流统计信息的一种机制。

(2) 强制老化

用户可以执行 **reset ipv6 netstream statistics** 命令强制将 IPv6 NetStream 缓冲区中所有流老化，并清空 IPv6 NetStream 缓冲区信息；或者根据实际需要，使用 **ipv6 netstream max-entry** 命令配置 NetStream 流缓存区中流表项的最大数目，以及当达到 NetStream 流缓存区中流表项的最大数目时的处理方式。

(3) TCP 的 FIN 和 RST 报文触发老化

对于 TCP 连接，当有标志为 FIN 或 RST 的报文发送时，表示一次会话结束。因此当一条已经存在的 TCP 协议 IPv6 NetStream 流中流过一条标志为 FIN 或 RST 的报文时，可以立即老化相应的 IPv6 NetStream 流。但是假如一条流的第一个报文就是 TCP 的 FIN 或 RST 报文，则会按正常的流程创建一条新流，不进行老化。这种方式在设备上缺省开启，不能通过执行命令开启或关闭该老化方式。

2. 配置IPv6 NetStream的流老化

表1-9 配置 IPv6 NetStream 的流老化

操作		命令	说明
进入系统视图		**system-view**	-
配置按时老化	（可选）配置流的活跃老化时间	**ipv6 netstream timeout active** *minutes*	缺省情况下，流的活跃老化时间为30分钟
	（可选）配置流的不活跃老化时间	**ipv6 netstream timeout inactive** *seconds*	缺省情况下，流的不活跃老化时间为30秒
配置强制老化	（可选）配置IPv6 NetStream流缓存区中流表项的最大数目	**ipv6 netstream max-entry** *max-entries*	缺省情况下，IPv6 NetStream流缓存区中流表项的最大数目为10000
	退回用户视图	**quit**	-

操作		命令	说明
	（可选）将流缓存区中所有流强制老化，并清除IPv6 NetStream缓冲区的状态信息和输出报文信息	**reset ipv6 netstream statistics**	-

1.3.6 配置IPv6 NetStream统计信息的输出

在配置 IPv6 NetStream 统计信息的输出时，需要对 IPv6 NetStream 统计信息输出的源接口和目的地址进行设置，后者为必选配置。

1. 配置IPv6 NetStream普通流的统计信息输出

表1-10 配置 IPv6 NetStream 普通流的统计信息输出

操作	命令	说明
进入系统视图	**system-view**	-
配置IPv6 NetStream普通流统计信息输出的目的地址和目的UDP端口号	**ipv6 netstream export host** { [*ip-address*] | [*ipv6-address*] } *udp-port* [**vpn-instance** *vpn-instance-name*]	缺省情况下，系统视图下没有配置目的地址和目的UDP端口号
（可选）配置IPv6 NetStream统计输出报文的源接口	**ipv6 netstream export source interface** *interface-type interface-number*	缺省情况下，采用统计输出报文的出接口（即与服务器相连的接口）作为源接口 建议使用网管口作为源接口与服务器相连，并向服务器输出统计信息
（可选）配置输出速率限制	**ipv6 netstream export rate** *rate*	缺省情况下，IPv6 NetStream统计输出报文的输出速率不受限制

2. 配置IPv6 NetStream聚合流的统计信息输出

配置 NetStream 聚合流的统计信息输出时，需要注意在聚合视图下，用户配置的 IPv6 NetStream 统计输出报文的输出属性，仅对聚合报文生效；而系统视图下的配置对普通报文生效，并且当聚合视图下没有配置以上属性时，也会对聚合报文生效。

表1-11 配置 IPv6 NetStream 聚合流的统计信息输出

操作	命令	说明
进入系统视图	**system-view**	-
进入IPv6 NetStream聚合视图	**ipv6 netstream aggregation** { **as** | **bgp-nexthop** | **destination-prefix** | **prefix** | **protocol-port** | **source-prefix** }	-

操作	命令	说明
配置IPv6 NetStream聚合统计信息输出的目的地址和目的UDP端口号	**ipv6 netstream export host** { [*ip-address*] | [*ipv6-address*] } *udp-port* [**vpn-instance** *vpn-instance-name*]	缺省情况下，聚合视图下没有配置目的地址和目的UDP端口号 为了减少对网络带宽的占用，可以只在聚合视图下配置本命令，此时设备只会输出聚合流信息
（可选）配置IPv6 NetStream聚合统计信息输出的源接口	**ipv6 netstream export source interface** *interface-type interface-number*	缺省情况下，采用统计输出报文的出接口（即与服务器相连的接口）作为源接口 需要注意的是： • 不同聚合视图下可以配置不同的源接口 • 聚合视图下若没有配置源接口，则使用系统视图下的配置
开启当前聚合视图对应的聚合功能	**enable**	缺省情况下，未开启任何IPv6 NetStream聚合功能

1.4 IPv6 NetStream显示和维护

在完成上述配置后，在任意视图下执行 **display** 命令可以显示配置后 IPv6 NetStream 的运行情况，通过查看显示信息验证配置的效果。

在用户视图下执行 **reset** 命令可以清除 IPv6 NetStream 的统计信息。

表1-12 IPv6 NetStream 显示和维护

操作	命令
查看IPv6 NetStream流缓存区的配置和状态信息（MSR 2600/MSR 3600）	**display ipv6 netstream cache** [**verbose**]
查看IPv6 NetStream流缓存区的配置和状态信息（MSR 5600）	**display ipv6 netstream cache** [**slot** *slot-number* [**cpu** *cpu-number*]] [**verbose**]
查看IPv6 NetStream统计输出报文信息	**display ipv6 netstream export**
查看IPv6 NetStream模板的配置和状态信息（MSR 2600/MSR 3600）	**display ipv6 netstream template**
查看IPv6 NetStream模板的配置和状态信息（MSR 5600）	**display ipv6 netstream template** [**slot** *slot-number* [**cpu** *cpu-number*]]
将流缓存区中所有流强制老化，并清除IPv6 NetStream缓冲区的状态信息和输出报文信息	**reset ipv6 netstream statistics**

1.5 IPv6 NetStream典型配置举例

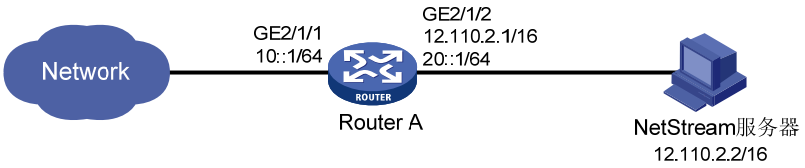
1.5.1 IPv6 NetStream普通流的统计信息输出配置举例

1. 组网需求

如 图 1-4 所示，在Router A上启动IPv6 NetStream功能。要求在Gigabitethernet2/1/1 上配置IPv6 NetStream入统计，并将IPv6 NetStream普通流的统计信息输出到NetStream服务器。NetStream服务器的IP地址为 12.110.2.2/16，UDP端口号为 5000。

2. 组网图

图1-4 IPv6 NetStream 普通流的统计信息输出配置组网图



3. 配置步骤

在 GigabitEthernet2/1/1 上启动 IPv6 NetStream 入统计。

```
<RouterA> system-view
[RouterA] interface gigabitethernet 2/1/1
[RouterA-GigabitEthernet2/1/1] ipv6 address 10::1/64
[RouterA-GigabitEthernet2/1/1] ipv6 netstream inbound
[RouterA-GigabitEthernet2/1/1] ipv6 netstream outbound
[RouterA-GigabitEthernet2/1/1] quit
```

配置 IPv6 NetStream 普通流统计信息输出的目的地址为 12.110.2.2 和目的 UDP 端口号为 5000。

```
[RouterA] ipv6 netstream export host 12.110.2.2 5000
```

4. 验证配置

设备运行一段时间后，查看 IPv6 NetStream 普通流的统计信息。

查看 IPv6 NetStream 流缓冲区信息。

```
<Sysname> display ipv6 netstream cache verbose slot 1
IPv6 NetStream cache information:
  Active flow timeout             : 60 min
  Inactive flow timeout           : 10 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  : 02/11/2000 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     GE2/1/1(I)  42996
IP   2001::1(1024)      2002::1(21)        6   0   0x0     GE2/1/1(O)  42996
```

查看 IPv6 NetStream 统计输出报文的各种信息。

```
[RouterA] display ipv6 netstream export
IPv6 export information:
  Flow source interface                          : Not specified
  Flow destination VPN instance                  : Not specified
  Flow destination IP address (UDP)              : 12.110.2.2 (5000)
  Version 9 exported flows number                : 10
  Version 9 exported UDP datagrams number (failed): 10 (0)
```

1.5.2 IPv6 NetStream聚合流的统计信息输出配置举例

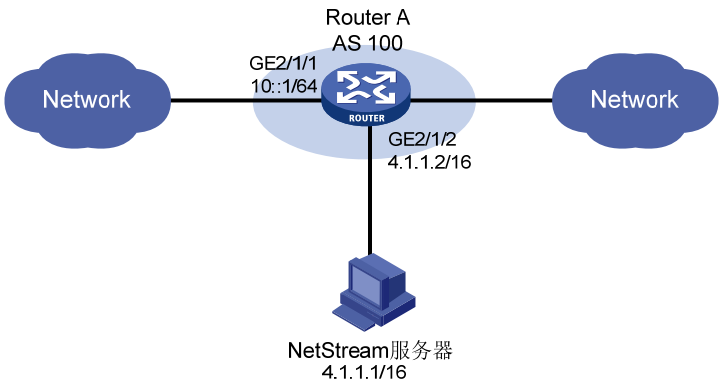
1. 组网需求

在 Router A 上配置 IPv6 NetStream，具体要求为：

- 普通流的统计信息使用输出到 IPv6 NetStream 服务器。IPv6 NetStream 服务器的 IP 地址为 4.1.1.1/16，UDP 端口号为 5000；
- 5 种聚合流（**as**、**protocol-port**、**source-prefix**、**destination-prefix** 和 **prefix**）的统计信息输出到该 NetStream 服务器的 2000、3000、4000、6000 和 7000 端口。

2. 组网图

图1-5 IPv6 NetStream 聚合流的统计信息输出配置组网图



3. 配置步骤

网络之间运行 EBGP 路由协议。相关配置请参见“三层技术-IP 路由配置指导”中的“BGP”。

在 GigabitEthernet2/1/1 上启动 IPv6 NetStream 统计。

```
<RouterA> system-view
[RouterA] interface gigabitEthernet 2/1/1
[RouterA-GigabitEthernet2/1/1] ipv6 address 10::1/64
[RouterA-GigabitEthernet2/1/1] ipv6 netstream inbound
[RouterA-GigabitEthernet2/1/1] ipv6 netstream outbound
[RouterA-GigabitEthernet2/1/1] quit
```

配置普通流统计信息输出的目的地址为 4.1.1.1 和目的 UDP 端口号为 5000。

```
[RouterA] ipv6 netstream export host 4.1.1.1 5000
```

配置自治系统聚合模式，以及该聚合流统计信息输出的目的地址为 4.1.1.1 和目的 UDP 端口号为 2000。

```
[RouterA] ipv6 netstream aggregation as
[RouterA-ns6-aggregation-as] enable
[RouterA-ns6-aggregation-as] ipv6 netstream export host 4.1.1.1 2000
[RouterA-ns6-aggregation-as] quit
```

配置协议－端口聚合模式，以及该聚合流统计信息输出的目的地址为 4.1.1.1 和目的 UDP 端口号为 3000。

```
[RouterA] ipv6 netstream aggregation protocol-port
[RouterA-ns6-aggregation-protport] enable
[RouterA-ns6-aggregation-protport] ipv6 netstream export host 4.1.1.1 3000
[RouterA-ns6-aggregation-protport] quit
```

配置源前缀聚合模式，以及该聚合流统计信息输出的目的地址为 4.1.1.1 和目的 UDP 端口号为 4000。

```
[RouterA] ipv6 netstream aggregation source-prefix
[RouterA-ns6-aggregation-srcpre] enable
[RouterA-ns6-aggregation-srcpre] ipv6 netstream export host 4.1.1.1 4000
[RouterA-ns6-aggregation-srcpre] quit
```

配置目的前缀聚合模式，以及该聚合流统计信息输出的目的地址为 4.1.1.1 和目的 UDP 端口号为 6000。

```
[RouterA] ipv6 netstream aggregation destination-prefix
[RouterA-ns6-aggregation-dstpre] enable
[RouterA-ns6-aggregation-dstpre] ipv6 netstream export host 4.1.1.1 6000
[RouterA-ns6-aggregation-dstpre] quit
```

配置前缀聚合模式，以及该聚合流统计信息输出的目的地址为 4.1.1.1 和目的 UDP 端口号为 7000。

```
[RouterA] ipv6 netstream aggregation prefix
[RouterA-ns6-aggregation-prefix] enable
[RouterA-ns6-aggregation-prefix] ipv6 netstream export host 4.1.1.1 7000
[RouterA-ns6-aggregation-prefix] quit
```

4. 验证配置

查看 NetStream 统计输出报文的各种信息。

```
[RouterA] display ipv6 netstream export
as aggreagtion export information:
  Flow source interface                          : Not specified
  Flow destination VPN instance                  : Not specified
  Flow destination IP address (UDP)              : 4.1.1.1 (2000)
  Version 9 exported flows number                : 0
  Version 9 exported UDP datagrams number (failed): 0(0)

protocol-port aggreagtion export information:
  Flow source interface                          : Not specified
  Flow destination VPN instance                  : Not specified
  Flow destination IP address (UDP)              : 4.1.1.1 (3000)
  Version 9 exported flows number                : 0
  Version 9 exported UDP datagrams number (failed): 0 (0)

source-prefix aggreagtion export information:
  Flow source interface                          : Not specified
  Flow destination VPN instance                  : Not specified
  Flow destination IP address (UDP)              : 4.1.1.1 (4000)
  Version 9 exported flows number                : 0
  Version 9 exported UDP datagrams number (failed): 0 (0)

destination-prefix aggreagtion export information:
  Flow source interface                          : Not specified
  Flow destination VPN instance                  : Not specified
  Flow destination IP address (UDP)              : 4.1.1.1 (6000)
  Version 9 exported flows number                : 0
  Version 9 exported UDP datagrams number (failed): 0 (0)

prefix aggreagtion export information:
  Flow source interface                          : Not specified
  Flow destination VPN instance                  : Not specified
  Flow destination IP address (UDP)              : 4.1.1.1 (7000)
  Version 9 exported flows number                : 0
```

```
  Version 9 exported UDP datagrams number (failed): 0 (0)

IPv6 export information:
  Flow source interface                          : Not specified
  Flow destination VPN instance                  : Not specified
  Flow destination IP address (UDP)              : 4.1.1.1 (5000)
  Version 9 exported flows number                : 0
  Version 9 exported UDP datagrams number (failed): 0 (0)
```

目 录

1 sFlow

1.1 sFlow简介

sFlow（Sampled Flow，采样流）是一种基于报文采样的网络流量监控技术，主要用于对网络流量进行统计分析。

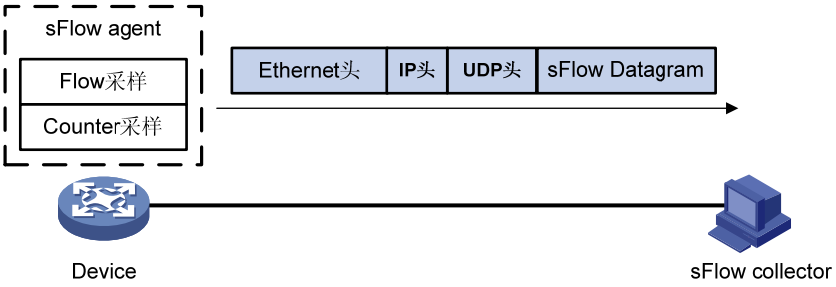
1.1.1 sFlow的工作机制

如 图 1-1 所示，sFlow系统包含嵌入在设备中的sFlow Agent和远端的sFlow Collector。其中，sFlow Agent通过采样机制获取接口的统计信息和数据包信息，将信息封装成sFlow报文，当存放sFlow报文的缓冲区满或是在sFlow报文发送定时器（定时器时间间隔固定为 1 秒）超时后，会将sFlow报文封装在UDP报文里发送到指定的sFlow Collector。sFlow Collector会对sFlow报文进行分析，并显示分析结果。

sFlow 使用以下两种采样机制：

- Flow 采样：基于数据包的流采样，用于获取数据包内容的相关信息。
- Counter 采样：基于时间的接口统计信息采样，用于获取接口的统计信息。

图1-1 sFlow 工作机制



作为一种网络流量监控技术，sFlow 具有如下优点：

- 支持在千兆或更高速的网络上精确地监控网络流量。
- 一个 sFlow Collector 能够监控多个 sFlow Agent，具有良好的扩展性。
- sFlow Agent 可以内嵌在设备中，不需要专门的 sFlow Agent 设备，节省了成本。

1.1.2 协议规范

与 sFlow 相关的协议规范有：

- RFC3176：InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks
- sFlow.org：sFlow Version 5

1.2 sFlow配置任务简介

表1-1 sFlow 配置任务简介

配置任务	说明	详细配置
配置sFlow Agent和sFlow Collector信息	必选 配置设备的sFlow特性，使远端的sFlow Collector能够监控网络	1.3.1
配置sFlow Flow采样	二者至少选其一	1.3.2
配置sFlow Counter采样		1.3.3

1.3 sFlow配置

1.3.1 配置sFlow Agent和sFlow Collector信息

表1-2 配置 sFlow Agent 和 sFlow Collector 信息

操作	命令	说明
进入系统视图	**system-view**	-
（可选）配置sFlow Agent的IP地址	**sflow agent** { **ip** *ip-address* | **ipv6** *ipv6-address* }	缺省情况下，未配置sFlow Agent的IP地址。设备会定期检查是否存在sFlow Agent的IP地址，如果不存在，设备会自动查找一个IPv4地址作为sFlow Agent的IP地址。自动查找的IP地址信息不会保存在配置文件中 需要注意的是： • 建议用户手工配置 sFlow Agent 的 IP 地址 • 在设备上只能配置一个 sFlow Agent 的 IP 地址，新配置的 IP 地址会覆盖已有的配置
配置sFlow Collector的参数	**sflow collector** *collector-id* [**vpn-instance** *vpn-instance-name*] { **ip** *ip-address* | **ipv6** *ipv6-address* } [**port** *port-number* | **datagram-size** *size* | **time-out** *seconds* | **description** *text*] *	缺省情况下，没有sFlow Collector的相关信息存在
（可选）配置sFlow报文的源IP地址	**sflow source** { **ip** *ip-address* | **ipv6** *ipv6-address* } *	缺省情况下，设备使用路由决定的源IP地址作为sFlow报文的源IP地址

1.3.2 配置Flow采样

在接口上配置 Flow 采样后，sFlow Agent 会根据配置的参数对该接口上的报文进行采样，然后将采样报文封装为 sFlow 报文。

表1-3 配置 Flow 采样

操作	命令	说明
进入系统视图	**system-view**	-
进入二层/三层以太网接口视图	**interface** *interface-type interface-number*	-
（可选）配置Flow采样的采样模式	**sflow sampling-mode** { **determine** | **random** }	缺省情况下，采样模式为固定数目的报文采样
配置Flow采样的报文采样率，即在*rate*个报文中抽取一个报文进行采样，同时开启Flow采样功能	**sflow sampling-rate** *rate*	缺省情况下，未开启Flow采样功能
（可选）配置在进行报文内容拷贝时，从原始报文的头部开始，允许拷贝的最大字节数	**sflow flow max-header** *length*	缺省情况下，从原始报文的头部开始，允许拷贝的最大字节数为128字节 建议用户使用缺省配置
配置经过Flow采样后，sFlow Agent输出sFlow报文的目的sFlow Collector编号	**sflow flow collector** *collector-id*	缺省情况下，Flow采样和sFlow Collector没有绑定关系，即没有指定目的sFlow Collector编号

1.3.3 配置Counter采样

在接口上配置 Counter 采样后，sFlow Agent 会周期性提取该接口上的统计信息，然后将统计信息封装为 sFlow 报文。

表1-4 配置 Counter 采样

操作	命令	说明
进入系统视图	**system-view**	-
进入二层/三层以太网接口视图	**interface** *interface-type interface-number*	-
配置Counter采样的时间间隔，同时开启Counter采样功能	**sflow counter interval** *interval-time*	缺省情况下，不进行Counter采样
配置经过Counter采样后，sFlow Agent输出sFlow报文的目的sFlow Collector编号	**sflow counter collector** *collector-id*	缺省情况下，Counter采样和sFlow Collector没有绑定关系，即没有指定目的sFlow Collector编号

1.4 sFlow显示和维护

在完成上述配置后，在任意视图下执行 **display** 命令可以显示配置后 sFlow 的运行情况，通过查看显示信息验证配置的效果。

表1-5 sFlow 显示和维护

操作	命令
显示sFlow配置信息	**display sflow**

1.5 sFlow典型配置举例

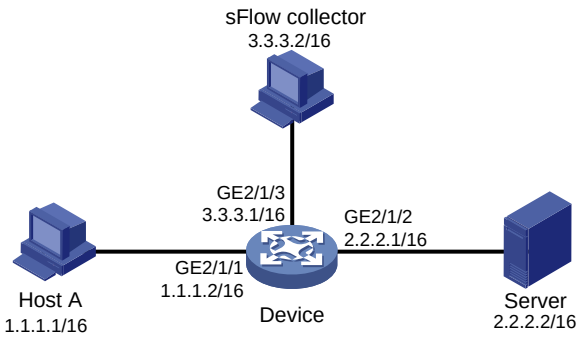
1.5.1 sFlow配置举例

1. 组网需求

如 图 1-2 所示，在Device上运行sFlow Agent，并在接口GigabitEthernet2/1/1 上开启sFlow功能，包括Flow采样（选择随机模式）和Counter采样，从而对该接口的网络流量进行监控。最后Device将采样结果封装为sFlow报文，通过接口GigabitEthernet2/1/3 发送给sFlow Collector，sFlow Collector对sFlow报文进行分析并显示分析结果。

2. 组网图

图1-2 配置 sFlow 组网图



3. 配置步骤

(1) 配置 IP 地址

请按照 图 1-2 配置各接口的IP地址和子网掩码，具体配置过程略。

(2) 配置 sFlow Agent 和 sFlow Collector 信息

配置 sFlow Agent 的 IP 地址。

```
<Sysname> system-view
[Sysname] sflow agent ip 3.3.3.1
```

配置 sFlow Collector 信息：sFlow Collector 编号为 1，IP 地址为 3.3.3.2，端口号保持缺省值 6343，描述信息为 netserver。

```
[Sysname] sflow collector 1 ip 3.3.3.2 description netserver
```

(3) 配置 Counter 采样

在 GigabitEthernet2/1/1 上配置 Counter 采样的时间间隔为 120 秒，同时开启 Counter 采样功能。

```
[Sysname] interface gigabitethernet 2/1/1
[Sysname-GigabitEthernet2/1/1] sflow counter interval 120
```

配置 sFlow Agent 输出 sFlow 报文的目的 sFlow Collector 编号为 1。

```
[Sysname-GigabitEthernet2/1/1] sflow counter collector 1
```

(4) 配置 Flow 采样

在 GigabitEthernet2/1/1 上配置 Flow 采样的采样模式为随机采样，并配置 Flow 采样的报文采样率为 4000，同时开启 Flow 采样功能。

```
[Sysname-GigabitEthernet2/1/1] sflow sampling-mode random
[Sysname-GigabitEthernet2/1/1] sflow sampling-rate 4000
```

配置 sFlow Agent 输出 sFlow 报文的目的 sFlow Collector 编号为 1。

```
[Sysname-GigabitEthernet2/1/1] sflow flow collector 1
```

4. 验证配置

显示 sFlow 的配置和运行信息。

```
[Sysname-GigabitEthernet2/1/1] display sflow
sFlow datagram version: 5
Global information:
Agent IP: 3.3.3.1(CLI)
Source address:
Collector information:
ID    IP              Port  Aging      Size VPN-instance Description
1     3.3.3.2         6343  N/A        1400              netserver
Port information:
Interface      CID   Interval(s) FID   MaxHLen Rate     Mode      Status
GE2/1/1         1     120         1     128     4000     Random    Active
```

从上面的显示信息中可以看到开启 sFlow 功能的 GigabitEthernet2/1/1 接口处于“Active”状态，Counter 采样的时间间隔为 120 秒，Flow 采样的报文采样率为 4000，即在 4000 个报文中抽取一个报文进行采样，表示 sFlow 功能正在正常运行。

1.6 常见配置错误举例

1.6.1 远端的sFlow Collector无法收到sFlow报文

1. 故障现象

远端的 sFlow Collector 无法收到 sFlow 报文。

2. 故障分析

可能有以下几个原因：

- 没有配置 sFlow Collector 的 IP 地址；
- 接口没有配置 sFlow 采样，导致没有接口提供采样数据；
- 配置的 sFlow Collector 的 IP 地址和远端的 sFlow Collector 的 IP 地址不同，导致远端的 sFlow Collector 无法收到 sFlow 报文；
- 设备没有配置发送 sFlow 报文的三层接口的 IP 地址，或配置了 IP 地址但是以此 IP 为源 IP 地址的 UDP 报文无法到达 sFlow Collector；
- 设备和 sFlow Collector 之间的物理连接中断；
- 为 sFlow Collector 绑定了 VPN 但是未配置 VPN 实例；

- 配置的 sFlow 报文的长度小于 sFlow 报文头长度与配置的采样样本长度之和。

3. 处理过程

(1) 执行 **display sflow** 命令，查看当前的 sFlow 配置信息，检查是否为 sFlow 特性配置错误；

(2) 检查设备是否已经配置可以和 sFlow Collector 通信的 IP 地址；

(3) 检查设备和 sFlow Collector 之间的物理连接是否正常；

(4) 如果为 sFlow Collector 绑定了 VPN，查看该 VPN 实例是否已经配置；

(5) 检查配置的 sFlow 报文的长度是否大于 sFlow 报文头长度与配置的采样样本长度（建议使用缺省配置）之和。

目 录

1 信息中心

说明

设备运行于 FIPS 模式时，本特性部分配置相对于非 FIPS 模式有所变化，具体差异请见本文相关描述。有关 FIPS 模式的详细介绍请参见“安全配置指导”中的“FIPS”。

1.1 信息中心简介

1.1.1 信息中心概述

信息中心是设备的信息枢纽，它能够对系统内所有模块的日志信息进行分类、管理，为网络管理员监控网络运行情况和诊断网络故障提供了有力的支持。系统由众多模块构成，日志信息可按来源模块进行划分并过滤输出。系统支持的来源模块可以通过在系统视图下输入 **info-center source ?**进行查看。

信息中心的工作过程如下：

(1) 接收各模块生成的日志信息。

(2) 根据用户设置的输出规则，将收到的日志信息输出到不同方向。

图1-1 信息中心功能示意图

缺省情况下，信息中心处于开启状态，当需要处理的信息较多时，会对系统性能有一定的影响。在系统资源不足时可以关闭信息中心来节约系统资源。

1.1.2 日志信息的分类

系统产生的日志信息共分为：

- 普通日志：用于记录日常信息。除特殊说明外，下文中的日志均指普通日志。
- 诊断日志：用于记录调试信息。
- 安全日志：用于记录与认证、授权等安全相关的信息。
- 隐藏日志：用于记录需要以日志的方式记录下来但不需要在终端上显示的信息（如用户通过命令行输入命令的记录信息等）。
- 调试跟踪日志：用于记录系统跟踪调试信息，调试跟踪日志信息，必须加载 devkit 包后才可以查看，普通用户无需关注，主要提供给服务工程师定位问题。

- 用户定制日志：用于记录用户厂商定制的特定操作产生的固定格式的日志信息，一般针对特定的目标客户。此类型的日志，日志产生的频率比较高。

1.1.3 日志信息的等级

日志信息按严重性可划分为如 表 1-1 所示的八个等级，各等级的严重性依照数值从 0～7 依次降低。在系统输出信息时，所有信息等级高于或等于设置等级的信息都会被输出。例如，输出规则中指定允许等级为 6（informational）的信息输出，则等级 0～6 的信息均会被输出。

表1-1 日志信息等级列表

数值	信息等级	描述
0	emergency	表示设备不可用的信息，如系统授权已到期
1	alert	表示设备出现重大故障，需要立刻做出反应的信息，如流量超出接口上限
2	critical	表示严重信息，如设备温度已经超过预警值，设备电源、风扇出现故障等
3	error	表示错误信息，如接口链路状态变化，存储卡拔出等
4	warning	表示警告信息，如接口连接断开，内存耗尽告警等
5	notification	表示正常出现但是重要的信息，如通过终端登录设备，设备重启等
6	informational	表示需要记录的通知信息，如通过命令行输入命令的记录信息，执行**ping**命令的日志信息等
7	debugging	表示调试过程产生的信息

1.1.4 日志信息的输出方向

系统可以向以下方向发送日志信息：控制台（console）、监视终端（monitor）、日志缓冲区（logbuffer）、日志主机（loghost）和日志文件（logfile）。日志信息的各个输出方向相互独立，可在开启信息中心后分别进行设置。

1.1.5 日志信息的缺省输出规则

日志信息的输出规则规定了各个输出方向可以输出的日志信息模块和输出的日志信息等级，日志信息的输出方向包括控制台、监控终端、日志主机、日志缓冲区和日志文件。各个输出方向的缺省情况如 表 1-2 所示：

表1-2 输出方向的缺省输出规则

输出方向	日志信息来源	开关	等级
控制台	所有支持的模块	开	debugging
监视终端	所有支持的模块	关	debugging
日志主机	所有支持的模块	开	informational
日志缓冲区	所有支持的模块	开	informational
日志文件	所有支持的模块	开	informational

1.1.6 诊断日志信息的缺省输出规则

诊断日志信息的输出方向只有诊断日志文件。诊断日志文件不能进行输出模块和输出级别的过滤配置，输出方向的缺省情况如 表 1-3 所示：

表1-3 输出方向的缺省输出规则

输出方向	日志信息来源	开关	等级
诊断日志文件	所有支持的模块	开	debugging

1.1.7 安全日志信息的缺省输出规则

安全日志信息的输出方向只有安全日志文件。安全日志文件不能进行输出模块和输出级别的过滤配置，输出方向的缺省情况如 表 1-4 所示：

表1-4 输出方向的缺省输出规则

输出方向	日志信息来源	开关	等级
安全日志文件	所有支持的模块	关	debugging

1.1.8 隐藏日志信息的缺省输出规则

隐藏日志信息的输出方向包括日志主机、日志缓冲区和日志文件。各个输出方向的缺省情况如 表 1-5 所示：

表1-5 输出方向的缺省输出规则

输出方向	日志信息来源	开关	等级
日志主机	所有支持的模块	开	informational
日志缓冲区	所有支持的模块	开	informational
日志文件	所有支持的模块	开	informational

1.1.9 调试跟踪日志信息的缺省输出规则

调试跟踪日志信息的输出方向只有调试跟踪日志文件。调试跟踪日志文件不能进行输出模块和输出级别的过滤配置，输出方向的缺省情况如 表 1-6 所示：

表1-6 输出方向的缺省输出规则

输出方向	日志信息来源	开关	等级
调试跟踪日志文件	所有支持的模块	开	debugging

1.1.10 用户定制日志信息的缺省输出规则

用户定制日志信息的输出方向只有用户定制日志主机。用户定制日志不能进行输出模块和输出级别的过滤配置，输出方向的缺省情况如 表 1-7 所示：

表1-7 输出方向的缺省输出规则

输出方向	日志信息来源	开关	等级
用户定制日志主机	特定的系统操作	开	debugging

1.1.11 日志信息的格式

1. 格式

根据输出方向不同，日志信息的输出格式如下：

表1-8 日志信息格式表

<table>
<tr><th>输出方向</th><th colspan="2">格式</th><th>举例</th></tr>
<tr><td>控制台、监视终端、日志缓冲区或日志文件</td><td colspan="2">Prefix Timestamp Sysname Module/Level/Mnemonic: Content</td><td>%Nov 24 14:21:43:502 2010 H3C SYSLOG/6/SYSLOG_RESTART: System restarted –-
H3C Comware Software.</td></tr>
<tr><td rowspan="3">日志主机</td><td>H3C格式</td><td><PRI>Timestamp Sysname %%vvModule/Level/Mnemonic: Source; Content</td><td><190>Nov 24 16:22:21 2010 H3C %%10SYSLOG/6/SYSLOG_RESTART: -DevIP=1.1.1.1; System restarted –-
H3C Comware Software.</td></tr>
<tr><td>unicom格式</td><td><PRI>Timestamp Hostip vvModule/Level/Serial_number: Content</td><td><189>Oct 13 16:48:08 2000 10.1.1.1 10IFNET/2/210231a64jx073000020: VTY logged in from 192.168.1.21</td></tr>
<tr><td>cmcc格式</td><td><PRI>Timestamp Sysname %vv Module/Level/Mnemonic: Source Content</td><td><189>Oct 9 14:59:04 2009 Sysname %10SHELL/5/SHELL_LOGIN: VTY logged in from 192.168.1.21</td></tr>
</table>

2. 字段说明

(1) Prefix（信息类型）

对于输出方向为控制台、监视终端、日志缓冲区或日志文件的日志信息，时间戳前面会有一个信息类型标识符：

- 百分号（%）：表示该日志信息为 informational 级别及以上级别的 log 日志。
- 星号（*）：表示该日志信息为 debugging 级别的 log 日志。
- 指数符号（^）：表示该日志信息为诊断日志（不区分级别）。

(2) PRI（优先级）

对于输出方向为日志主机的日志信息，时间戳前面会有一个优先级标识符。优先级的计算公式为：facility*8+level。

- facility 表示工具名称，由 **info-center loghost** 命令设置，主要用于在日志主机端标志不同的日志来源，查找、过滤对应日志源的日志。其中，local0～local7 分别对应取值 16～23。
- level表示日志信息的等级，具体含义请参见 表 1-1。

(3) Timestamp（时间戳）

时间戳记录了日志信息产生的时间，方便用户查看和定位系统事件。发送到日志主机和发送到其它方向的日志信息的时间戳精度不同：

- 发送到日志主机的日志信息的时间戳精确到秒。
- 发送到其它方向的日志信息的时间戳精确到毫秒。

发送到日志主机和发送到其它方向的日志信息的时间戳的设置命令也不同：

- 发送到日志主机的日志信息的时间戳格式由 **info-center timestamp loghost** 命令设置。
- 发送到其它方向的日志信息的时间戳格式由 **info-center timestamp** 命令设置。

各时间戳格式的详细描述如 表 1-9 所示：

表1-9 时间戳参数描述表

时间戳参数	说明	举例
boot	系统启动后经历的时间（即设备本次运行的持续时间），格式为：xxx.yyy，其中xxx是系统启动后经历时间的毫秒数高32位，yyy是低32位 除日志主机方向外，发往其它方向的日志信息均支持该参数	%0.109391473 Sysname FTPD/5/FTPD_LOGIN: User ftp (192.168.1.23) has logged in successfully. 其中0.109391473即为**boot**格式的时间戳
date	系统当前的日期和时间，格式为： • 日志主机：“mmm dd hh:mm:ss yyyy” • 其他方向：“MMM DD hh:mm:ss:xxx YYYY” 发往所有方向的日志信息均支持该参数	%May 30 05:36:29:579 2013 Sysname FTPD/5/FTPD_LOGIN: User ftp (192.168.1.23) has logged in successfully. 其中May 30 05:36:29:579 2013即为**date**格式的时间戳
iso	ISO 8601中规定的时间戳格式 只有发往日志主机方向的日志信息支持该参数	<189>2013-05-30T06:42:44 Sysname %%10FTPD/5/FTPD_LOGIN(l): User ftp (192.168.1.23) has logged in successfully. 其中2013-05-30T06:42:44即为**iso**格式的时间戳
none	不带时间信息 发往所有方向的日志信息均支持该参数	% Sysname FTPD/5/FTPD_LOGIN: User ftp (192.168.1.23) has logged in successfully. 其中没有包含时间戳
no-year-date	系统当前日期和时间，但不包含年份信息，格式为“MMM DD hh:mm:ss:xxx” 只有发往日志主机方向的日志信息支持该参数	<189>May 30 06:44:22 Sysname %%10FTPD/5/FTPD_LOGIN(l): User ftp (192.168.1.23) has logged in successfully. 其中May 30 06:44:22即为**no-year-date**格式的时间戳

(4) Hostip（出接口 IP 地址）

本字段表示发送的日志信息的源 IP 地址。只有配置 **info-center loghost source** 后，此字段才显示为出接口的 IP 地址，未配置时，使用 Sysname 显示。本字段只有在使用 unicom 格式发往日志主机时才存在。

(5) Serial_number(设备序列号)

本字段为当前系统的设备实体序列号，只有在使用 unicom 格式发往日志主机时才存在。

(6) Sysname（主机名或主机 IP 地址）

本字段为生成该日志信息的设备的名称或 IP 地址。用户可使用 **sysname** 命令修改设备的名称。

(7) %%（厂家标志）

本字段表示本日志信息由 H3C 设备生成。

本字段只有在日志信息发往日志主机时才会存在。

(8) vv（版本信息）

本字段为日志信息的版本标识，取值为 10。

本字段只有在日志信息发往日志主机时才会存在。

(9) Module（模块名）

本字段为生成该日志信息的功能模块的名称。模块列表可以通过在系统视图下输入命令 **info-center source ?**进行查看。

(10) Level（信息等级）

本字段为日志信息的等级，具体说明请参见 表 1-1。

(11) Mnemonic（助记符）

本字段为该日志信息的概述，是一个不超过 32 个字符的字符串。

(12) Source（定位信息）

本字段为该日志信息的产生者，是可选字段。本字段的具体内容可能为：

- 单板槽位号（MSR 5600）
- 日志发送者的源 IP

(13) Content（信息文本）

本字段为该日志信息的具体内容。

1.2 配置信息中心

1.2.1 信息中心配置任务简介

表1-10 信息中心配置任务简介

配置任务	说明	详细配置
配置日志信息发送到控制台	五者至少选其一	1.2.2
配置日志信息发送到监视终端		1.2.3
配置日志信息发送到日志主机		1.2.4
配置日志信息发送到日志缓冲区		1.2.5

配置任务	说明	详细配置
配置日志信息保存到日志文件		1.2.6
配置安全日志同步保存和管理功能	可选	1.2.7
配置诊断日志信息保存到诊断日志文件	可选	1.2.8
配置调试跟踪日志文件	可选	1.2.9
配置命令行输入回显功能	可选	1.2.10
配置重复日志抑制功能	可选	1.2.11
禁止接口生成Link up/Link down日志信息	可选	1.2.12

1.2.2 配置日志信息发送到控制台

表1-11 配置日志信息发送到控制台

操作	命令	说明
进入系统视图	**system-view**	-
开启信息中心	**info-center enable**	缺省情况下，信息中心处于开启状态
配置日志信息的输出规则	**info-center source** { *module-name* | **default** } { **console** | **monitor** | **logbuffer** | **logfile** | **loghost** } { **deny** | **level** *severity* }	缺省情况下，日志信息的输出规则请参见1.1.5
（可选）设置时间戳输出格式	**info-center timestamp** { **boot** | **date** | **none** }	缺省情况下，信息的时间戳输出格式为date格式
退回到用户视图	**quit**	-
开启控制台对日志信息的监视功能	**terminal monitor**	缺省情况下，允许日志信息输出到控制台
开启当前终端对调试信息的显示功能	**terminal debugging**	缺省情况下，当前终端对调试信息的显示功能处于关闭状态
（可选）设置控制台显示日志信息的等级	**terminal logging level** *severity*	缺省情况下，控制台显示的日志信息最低等级为6（informational）

1.2.3 配置日志信息发送到监视终端

监视终端是指以 AUX、VTY、TTY 类型用户线登录的用户终端。

表1-12 配置日志信息发送到监视终端

操作	命令	说明
进入系统视图	**system-view**	-

操作	命令	说明
开启信息中心	**info-center enable**	缺省情况下，信息中心处于开启状态
配置日志信息的输出规则	**info-center source** { *module-name* | **default** } { **console** | **monitor** | **logbuffer** | **logfile** | **loghost** } { **deny** | **level** *severity* }	缺省情况下，日志信息的输出规则请参见1.1.5
（可选）设置时间戳输出格式	**info-center timestamp** { **boot** | **date** | **none** }	缺省情况下，信息的时间戳输出格式为date格式
退回到用户视图	**quit**	-
开启监视终端对日志信息的监视功能	**terminal monitor**	缺省情况下，不允许日志信息输出到监视终端
（可选）开启当前终端对调试信息的显示功能	**terminal debugging**	缺省情况下，当前终端对调试信息的显示功能处于关闭状态
（可选）设置监控终端显示日志信息的等级	**terminal logging level** *severity*	缺省情况下，监控终端显示的日志信息最低等级为6（informational）

1.2.4 配置日志信息发送到日志主机

表1-13 配置日志信息发送到日志主机

操作	命令	说明
进入系统视图	**system-view**	-
开启信息中心	**info-center enable**	缺省情况下，信息中心处于开启状态
配置日志信息的输出规则	**info-center source** { *module-name* | **default** } { **console** | **monitor** | **logbuffer** | **logfile** | **loghost** } { **deny** | **level** *severity* }	缺省情况下，日志信息的输出规则请参见1.1.5
（可选）配置发送日志信息时使用的源IP地址	**info-center loghost source** *interface-type interface-number*	缺省情况下，系统根据路由来确定发送日志信息的出接口，使用该接口的主IP地址作为发送的日志信息的源IP地址
（可选）设置发送的日志信息的时间戳格式	**info-center timestamp loghost** { **date** | **iso** | **no-year-date** | **none** }	缺省情况下，发往日志主机的日志信息的时间戳输出格式为**date**格式
指定日志主机并设置相关参数	**info-center loghost** [**vpn-instance** *vpn-instance-name*] { *ipv4-address* | **ipv6** *ipv6-address* } [**port** *port-number*] [**facility** *local-number*]	缺省情况下，系统中没有指定日志主机和相关参数 *port-number*参数的值需要和日志主机侧的设置一致，否则，日志主机接收不到日志信息

1.2.5 配置日志信息发送到日志缓冲区

表1-14 配置日志信息发送到日志缓冲区

操作	命令	说明
进入系统视图	**system-view**	-
开启信息中心	**info-center enable**	缺省情况下，信息中心处于开启状态
允许日志信息输出到日志缓冲区	**info-center logbuffer**	缺省情况下，允许日志输出到日志缓冲区
（可选）设置系统向日志缓冲区输出信息以及日志缓冲区的容量	**info-center logbuffer size** *buffersize*	缺省情况下，日志缓冲区可存储512条信息
配置日志信息的输出规则	**info-center source** { *module-name* | **default** } { **console** | **monitor** | **logbuffer** | **logfile** | **loghost** } { **deny** | **level** *severity* }	缺省情况下，日志信息的输出规则请参见1.1.5
（可选）设置时间戳输出格式	**info-center timestamp** { **boot** | **date** | **none** }	缺省情况下，日志信息的时间戳输出格式为date格式

1.2.6 配置日志信息保存到日志文件

通过使用本特性，用户可以将系统产生的日志信息保存到设备的日志文件中以便随时查看。

日志在保存到日志文件前，先保存在日志文件缓冲区。系统会按照指定的频率将日志文件缓冲区的内容写入日志文件，频率一般配置为 24 小时一次，在设备比较空闲的时候（比如清晨）进行保存，用户也可以手工触发保存。成功保存后，保存前的日志文件缓冲区里的内容会被清空。

日志文件有容量限制，当日志文件的大小达到最大值，且日志文件写满保护功能处于关闭状态，系统会自动创建新的日志文件来保存新信息，日志文件的名称为 logfile1.log、logfile2.log、……。当日志文件的个数达到设备支持的最大值或者设备可用存储介质的空间不足时，系统会删除最旧的日志文件再创建新的日志文件。在产生新的日志文件时，设备支持自动压缩日志文件，当 logfile1.log 写满，需要产生 logfile2.log 时，logfile1.log 会被压缩成 logfile1.log.gz，用户可以将压缩文件下载到本地后进行解压查看。

日志文件有容量限制，当日志文件的大小达到最大值，且日志文件写满保护功能处于开启状态，不再覆盖旧日志或删除最旧的日志文件，而是停止记录日志文件。

表1-15 配置日志信息保存到日志文件

操作	命令	说明
进入系统视图	**system-view**	-
开启信息中心	**info-center enable**	缺省情况下，信息中心处于开启状态
开启保存日志文件功能	**info-center logfile enable**	缺省情况下，允许日志信息输出到日志文件
（可选）开启日志文件写满保护功能	**info-center logfile overwrite-protection** [**all-port-powerdown**]	缺省情况下，日志文件写满保护功能处于关闭状态 本命令仅FIPS模式下支持

操作		命令	说明
（可选）设置单个日志文件最大能占用的存储空间的大小		**info-center logfile size-quota** *size*	缺省情况下，单个日志文件最大能占用的存储空间的大小为5MB 为了保证设备的正常运行，**info-center logfile size-quota**设置的日志文件的大小最小不能低于1MB，最大不能超过10MB
（可选）设置存储日志文件的目录		**info-center logfile directory** *dir-name*	缺省情况下，存储日志文件目录为存储设备根目录下的logfile目录 配置时，请注意： • 配置会在设备重启后失效（MSR 2600/MSR 3600） • 配置会在设备重启或主备倒换后失效（MSR 5600）
将日志文件缓冲区中的内容保存到日志文件	设置自动保存的频率	**info-center logfile frequency** *freq-sec*	二者选其一 缺省情况下，系统自动保存日志文件的频率为86400秒 **logfile save**命令在任意视图均可执行
	手动保存	**logfile save**	

1.2.7 配置安全日志同步保存和管理功能

1. 安全日志同步保存功能简介

查看系统日志是了解设备状态、定位和排除网络问题的一个重要方法，而在系统日志中与设备安全相关的安全日志显得尤为重要。但通常情况下，安全日志与其它日志一同输出，经常被淹没在大量的系统日志中，很难识别、不便于查看。针对这个问题，系统提供了安全日志同步保存功能。安全日志同步保存功能的配置和安全日志文件的管理相互分离，安全日志文件实行专人专管。

开启安全日志同步保存功能后，系统将安全日志进行集中处理：当生成的日志信息中有安全日志，在不影响日志信息现有输出规则的前提下，系统会将安全日志信息同步保存到专用的安全日志文件。这样既实现了安全日志的集中管理，又有利于用户随时快捷地查看安全日志，了解设备状态。

2. 配置安全日志同步保存功能

安全日志会先被输出到安全日志文件缓冲区（security-logfile buffer），系统会按照配置中指定的频率将安全日志文件缓冲区的内容写入安全日志文件（安全日志管理员也可以手工触发保存）。当安全日志文件缓冲区里的内容成功保存到安全日志文件后，安全日志文件缓冲区会被立即清空。系统只支持单个安全日志文件。

为了防止安全日志的丢失，用户可以设置安全日志文件使用率告警上限。当达到上限时，系统会输出日志信息提醒管理员，此时，管理员可以使用安全日志管理员身份登录设备，将安全日志文件进行备份，以防止重要历史数据丢失。

表1-16 配置安全日志同步保存功能

操作	命令	说明
进入系统视图	**system-view**	-
开启信息中心	**info-center enable**	缺省情况下，信息中心处于开启状态

操作	命令	说明
开启安全日志同步保存功能	**info-center security-logfile enable**	缺省情况下，安全日志同步保存功能处于关闭状态
设置设备自动保存安全日志文件的频率	**info-center security-logfile frequency** *freq-sec*	缺省情况下，设备自动保存安全日志文件的频率为86400秒
（可选）设置单个安全日志文件最大能占用的存储空间的大小	**info-center security-logfile size-quota** *size*	缺省情况下，单个安全日志文件最大能占用的存储空间的大小为10MB
（可选）设置安全日志文件使用率的告警上限	**info-center security-logfile alarm-threshold** *usage*	缺省情况下，安全日志文件使用率的告警门限是80（即当安全日志文件使用率达到80%时，系统会发出日志提醒用户）

3. 管理安全日志文件

安全日志管理员通过 AAA 本地认证登录设备后能对安全日志文件进行维护。安全日志管理员的相关配置请参见“安全配置指导”中的“AAA”。

表1-17 管理安全日志文件

操作	命令	说明
（可选）显示安全日志文件的概要信息	**display security-logfile summary**	本命令在用户视图下执行
修改存储安全日志文件的路径	**info-center security-logfile directory** *dir-name*	缺省情况下，存储安全日志文件路径为存储设备根目录下的seclog文件夹 配置时，请注意： • 配置会在设备重启后失效（MSR 2600/MSR 3600） • 配置会在设备重启或主备倒换后失效（MSR 5600）
手动将安全日志文件缓冲区中的内容全部保存到安全日志文件	**security-logfile save**	任意视图均可执行

1.2.8 配置诊断日志信息保存到诊断日志文件

通过使用本特性，用户可以将系统产生的诊断日志信息保存到设备的诊断日志文件中以便随时查看。

诊断日志在保存到诊断日志文件前，先保存在诊断日志文件缓冲区。系统会按照指定的频率将诊断日志文件缓冲区的内容写入诊断日志文件，频率一般配置为 24 小时一次，在设备比较空闲的时候（比如清晨）进行保存，用户也可以手工触发保存。成功保存后，保存前的诊断日志文件缓冲区里的内容会被清空。

诊断日志文件有容量限制，当诊断日志文件的大小达到最大值时，系统会自动创建新的诊断日志文件来保存新信息，日志文件的名称为 diagfile1.log、diagfile2.log、……。当日志文件的个数达到设备支持的最大值或者设备可用存储介质的空间不足时，系统会删除最旧的日志文件再创建新的日志文件。在产生新的诊断日志文件时，设备支持自动压缩诊断日志文件，当 diagfile1.log 写满，需要

产生 diagfile2.log 时，diagfile1.log 会被压缩成 diagfile1.log.gz，用户可以将压缩文件下载到本地后进行解压查看。

表1-18 配置诊断日志信息保存到诊断日志文件

操作		命令	说明
进入系统视图		**system-view**	-
开启信息中心		**info-center enable**	缺省情况下，信息中心处于开启状态
开启保存诊断日志文件功能		**info-center diagnostic-logfile enable**	缺省情况下，允许诊断日志信息输出到诊断日志文件
（可选）设置单个诊断日志文件最大能占用的存储空间的大小		**info-center diagnostic-logfile quota** *size*	缺省情况下，单个诊断日志文件最大能占用的存储空间的大小为5MB 为了保证设备的正常运行，**info-center diagnostic-logfile quota**设置的日志文件的大小最小不能低于1MB，最大不能超过10MB
（可选）设置存储诊断日志文件的目录		**info-center diagnostic-logfile directory** *dir-name*	缺省情况下，存储诊断日志文件路径为存储设备根目录下的diagfile文件夹 配置时，请注意： • 配置会在设备重启后失效（MSR 2600/MSR 3600） • 配置会在设备重启或主备倒换后失效（MSR 5600）
将诊断日志文件缓冲区中的内容保存到诊断日志文件	设置自动保存的频率	**info-center diagnostic-logfile frequency** *freq-sec*	二者选其一 缺省情况下，诊断日志自动保存到文件的频率为86400秒 **diagnostic-logfile save**命令在任意视图均可执行
	手动保存	**diagnostic-logfile save**	

1.2.9 配置调试跟踪日志文件

设备在调试过程中，会产生大量的调试跟踪日志。如果调试跟踪日志文件存储空间太小，可能会导致日志被很快覆盖，不利于定位问题。使用本特性，可以用来设置调试跟踪日志文件最大能占用的存储空间的大小。系统只支持单个跟踪调试日志文件。

表1-19 配置调试跟踪日志文件

操作	命令	说明
进入系统视图	**system-view**	-
设置调试跟踪日志文件最大能占用的存储空间的大小	**info-center trace-logfile quota**	缺省情况下，调试跟踪日志文件最大能占用的存储空间的大小为1MB

1.2.10 配置命令行输入回显功能

当用户进行命令行、参数或者 Y/N 确认信息输入时，如果被大量的日志信息打断，用户可能记不清已经输入了哪些字符串，还需要输入哪些字符串。使用命令行输入回显功能，能够协助用户配置。系统会在日志信息输出完毕后回显用户已有的输入或者 Y/N 确认信息，以便用户继续执行配置。

表1-20 配置命令行输入回显功能

操作	命令	说明
进入系统视图	**system-view**	-
开启命令行输入回显功能	**info-center synchronous**	缺省情况下，命令行输入回显功能处于关闭状态

1.2.11 配置重复日志抑制功能

当设备持续向某个方向发送同一条日志信息时（发送间隔小于 30 秒），大量重复的信息会浪费设备资源和网络资源，并导致有用的信息被淹没，不利于设备的维护。为了避免此问题，可开启重复日志抑制功能。

开启重复日志抑制功能后，设备每产生一条新日志信息，在输出该日志信息的同时会启动该日志的抑制周期：

- 该日志抑制周期内：如果设备后续连续生成的日志信息均与该日志信息相同（要求日志信息的如下字段均完全相同：模块名、信息等级、日志助记符、定位信息和信息文本），则系统会认为后续生成的日志是该日志的相同日志，后续生成的日志信息不再输出。
- 该日志抑制周期结束后：如果设备后续仍连续生成该日志，系统输出被抑制的日志信息以及被抑制的数量，并启动下一个日志抑制周期。日志信息的第一个抑制周期为 30 秒，第二个抑制周期为 2 分钟，以后的抑制周期都是 10 分钟。
- 如果在日志抑制周期内有其它新日志信息产生：系统会先输出被抑制的日志信息以及被抑制的数量，再输出新的日志信息，并开始新日志的抑制周期。

表1-21 配置重复日志抑制功能

操作	命令	说明
进入系统视图	**system-view**	-
开启重复日志抑制功能	**info-center logging suppress duplicates**	缺省情况下，重复日志抑制功能处于关闭状态

1.2.12 禁止接口生成Link up/Link down日志信息

缺省情况下，设备的所有接口在接口状态改变时都会生成 Link up/Link down 的日志信息。为了方便管理，用户可以根据实际情况禁止某些接口生成接口 Link up/Link down 的日志信息：

- 用户只关心某个或某些接口的状态时，可以使用该功能禁止其它接口生成 Link up/Link down 日志信息。
- 某个接口的状态因不稳定而频繁地改变，生成大量的 Link up/Link down 日志信息时，可以使用该功能禁止该接口生成 Link up/Link down 日志信息。

使用本特性后，如果接口状态改变，将不再生成接口 Link up/Link down 的日志信息。这样可能会影响用户监控接口状态，所以在一般情况下建议采用缺省配置。

表1-22 禁止接口生成 Link up/Link down 日志信息

操作	命令	说明
进入系统视图	**system-view**	-
进入接口视图	**interface** *interface-type interface-number*	-
禁止接口生成Link up/Link down日志信息	**undo enable log updown**	缺省情况下，允许所有接口在状态发生改变时生成接口Link up和Link down的日志信息

1.3 信息中心显示和维护

在完成上述配置后，在任意视图下执行 **display** 命令可以显示配置后信息中心的运行情况，通过查看显示信息验证配置的效果。

在用户视图下执行 **reset logbuffer** 命令可以将日志缓冲区的统计信息清除。

表1-23 信息中心显示和维护

操作	命令
显示各个输出方向的信息	**display info-center**
显示日志缓冲区的状态和日志缓冲区记录的日志信息（MSR 2600/MSR 3600）	**display logbuffer** [**reverse**] [**level** *severity* | **size** *buffersize*] *
显示日志缓冲区的状态和日志缓冲区记录的日志信息（MSR 5600）	**display logbuffer** [**reverse**] [**level** *severity* | **size** *buffersize* | **slot** *slot-number*] *
显示日志缓冲区的概要信息（MSR 2600/MSR 3600）	**display logbuffer summary** [**level** *severity*]
显示日志缓冲区的概要信息（MSR 5600备）	**display logbuffer summary** [**level** *severity* | **slot** *slot-number*] *
显示日志文件的配置	**display logfile summary**
清除日志缓冲区内的信息	**reset logbuffer**

1.4 信息中心典型配置举例

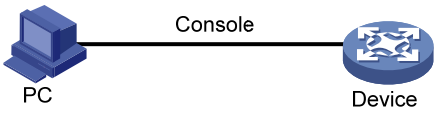
1.4.1 日志发送到控制台的配置举例

1. 组网需求

- 将信息等级高于等于 warning 的日志信息发送到控制台上；
- 允许输出日志信息的模块为 FTP。

2. 组网图

图1-2 日志信息发送到控制台配置组网图



3. 配置步骤

开启信息中心。

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

关闭控制台方向所有模块日志信息的输出开关。

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

说明

由于系统对各方向允许输出的日志信息的缺省情况不一样，所以配置前必须将所有模块指定方向（本例为 console）上日志信息的输出开关关闭，再根据当前的需求配置输出规则，以免输出太多不需要的信息。

配置输出规则：允许 FTP 模块的、等级高于等于 warning 的日志信息输出。

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

开启终端显示功能（该功能缺省情况下为开启状态）。

```
<Sysname> terminal logging level 6
<Sysname> terminal monitor
 Current terminal monitor is on.
```

以上命令配置成功后，如果指定的模块产生了日志信息，信息中心会自动把这些日志发送到控制台，在控制台的屏幕上显示。

1.4.2 日志发送到Unix日志主机的配置举例

1. 组网需求

- 将系统的日志信息发送到 Unix 日志主机；
- 将信息等级高于等于 informational 的日志信息将会发送到日志主机上；
- 仅允许输出日志信息的模块为 FTP。

2. 组网图

图1-3 日志信息发送到 Unix 日志主机配置组网图

3. 配置步骤

配置前请确保 Device 和 PC 之间路由可达。（具体配置步骤略）

(1) Device 上的配置

开启信息中心。

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

配置发送日志信息到 IP 地址为 1.2.0.1/16 的日志主机，日志主机记录工具为 local4。

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

关闭 loghost 方向所有模块日志信息的输出开关。

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

说明

由于系统对各方向允许输出的日志信息的缺省情况不一样，所以配置前必须将所有模块指定方向（本例为 loghost）上日志信息的输出开关关闭，再根据当前的需求配置输出规则，以免输出太多不需要的信息。

配置输出规则：允许 FTP 模块的、等级高于等于 informational 的日志信息输出到日志主机（注意：允许输出信息的模块由产品决定）。

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

(2) 日志主机上的配置

下面以 Solaris 操作系统上的配置为例介绍日志主机上的配置，在其它厂商的 Unix 操作系统上的配置操作基本类似。

第一步：以超级用户的身份登录日志主机。

第二步：在/var/log/路径下为 Device 创建同名日志文件夹 Device，在该文件夹创建文件 info.log，用来存储来自 Device 的日志。

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

第三步：编辑/etc/路径下的文件 syslog.conf，添加以下内容。

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

以上配置中，local4 表示日志主机接收日志的工具名称，info 表示信息等级。Unix 系统会把等级高于等于 informational 的日志记录到/var/log/Device/info.log 文件中。

说明

在编辑/etc/syslog.conf 时应注意以下问题：

- 注释必须独立成行，并以字符#开头。
- 在文件名之后不得有多余的空格。
- /etc/syslog.conf 中指定的工具名称及信息等级与 Device 上 **info-center loghost** 和 **info-center source** 命令的相应参数的指定值要保持一致，否则日志信息可能无法正确输出到日志主机上。

第四步：查看系统守护进程 syslogd 的进程号，中止 syslogd 进程，并重新用-r 选项在后台启动 syslogd，使修改后配置生效。

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

进行以上操作之后，Device 的日志信息会输出到 PC，PC 会将这些日志信息存储到相应的文件中了。

1.4.3 日志发送到Linux日志主机的配置举例

1. 组网需求

- 系统的日志信息发送到 Linux 日志主机上；
- 将信息等级高于等于 informational 的日志信息发送到日志主机上；
- 仅允许输出日志信息的模块为 FTP。

2. 组网图

图1-4 日志信息发送到 Linux 日志主机配置组网图

3. 配置步骤

配置前请确保 Device 和 PC 之间路由可达，具体配置步骤略。

(1) Device 上的配置

开启信息中心。

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

#配置发送日志信息到 IP 地址为 1.2.0.1/16 的日志主机，日志主机记录工具为 local5。

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

关闭 loghost 方向所有模块日志信息的输出开关。

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

说明

由于系统对各方向允许输出的日志信息的缺省情况不一样，所以配置前必须将所有模块的需求方向（本例为 loghost）上日志信息的输出开关关闭，再根据当前的需求配置输出规则，以免输出太多不需要的信息。

配置输出规则：允许 FTP 模块、等级高于等于 informational 的日志信息输出到日志主机。

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

(2) 日志主机上的配置

下面以 Solaris 操作系统上的配置为例介绍日志主机上的配置，在其它厂商的 Unix 操作系统上的配置操作基本类似。

第一步：以超级用户的身份登录日志主机。

第二步：在/var/log/路径下为 Device 创建同名日志文件夹 Device，在该文件夹创建文件 info.log，用来存储来自 Device 的日志。

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

第三步：编辑/etc/路径下的文件 syslog.conf，添加以下内容。

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

以上配置中，local5 表示日志主机接收日志的工具名称，info 表示信息等级。Linux 系统会把等级高于等于 informational 的日志记录到/var/log/Device/info.log 文件中。

 说明

在编辑/etc/syslog.conf 时应注意以下问题:

- 注释必须独立成行，并以字符#开头。
- 在文件名之后不得有多余的空格。
- /etc/syslog.conf 中指定的工具名称及信息等级与 Device 上 **info-center loghost** 和 **info-center source** 命令的相应参数的指定值要保持一致，否则日志信息可能无法正确输出到日志主机上。

第四步：查看系统守护进程 syslogd 的进程号，中止 syslogd 进程，并重新用-r 选项在后台启动 syslogd，使修改后配置生效。对 Linux 日志主机，必须保证 syslogd 进程是以-r 选项启动。

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

进行以上操作之后，系统就可以在相应的文件中记录日志信息了。