JXNET 甲信

甲信三层以太网交换机 系统管理通用配置指南

(SNMP RMON LLDP 端口镜像 电缆/光模块诊断 日志告警管理 软硬件监控 Ping Trace 设备堆叠 MAD NQA PATCH 补丁功能定时备份配置功能)

配置指南(CLI)

(Rel_01)

北京甲信技术有限公司（以下简称“甲信”）为客户提供全方位的技术支持和服务。直接向甲信购买产品的用户，如果在使用过程中有任何问题，可与甲信各地办事处或用户服务中心联系，也可直接与公司总部联系。

读者如有任何关于甲信产品的问题，或者有意进一步了解公司其他相关产品，可通过下列方式与我们联系：

公司网址：　　www.jiaxinnet.com.cn

技术支持邮箱：　jxhelp@bjjx.cc

技术支持热线：　400-179-1180

公司总部地址：　北京市海淀区丹棱 SOHO 7 层 728 室

邮政编码:　　100080

声　明

目录

1 系统管理

本章介绍系统管理与维护特性的基本原理和配置过程，并提供相关的配置案例。

- SNMP
- RMON
- LLDP
- 端口镜像
- 电缆诊断
- UDLD
- 光模块数字诊断
- 系统日志
- 告警管理
- CPU 监控
- 内存监控
- Ping
- Trace
- 硬件监控
- 风扇监控
- 设备堆叠
- MAD
- NQA
- PATCH 补丁功能
- 定时备份配置功能

1.1 SNMP

1.1.1 简介

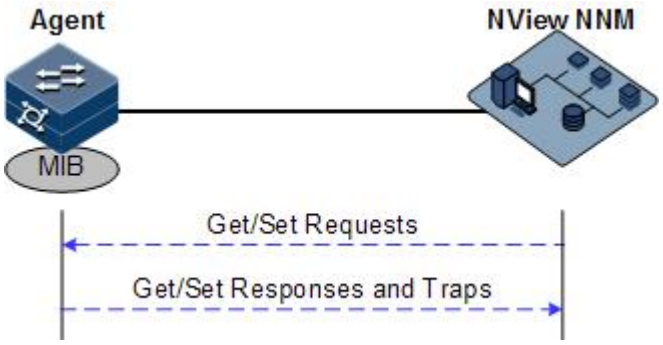
SNMP（Simple Network Management Protocol，简单网络管理协议）是由IETF（Internet Engineering Task Force，互联网工程任务组）为了解决Internet 中网络设备的管理问题而提出的一套网络管理协议。SNMP 可以

使一个网管系统远程管理所有支持这种协议的网络设备，包括监视网络状态、修改网络设备配置、接收网络事件告警等。它是目前 TCP/IP 网络中应用最广泛的网络管理协议。

工作机制

SNMP 的结构分为代理 Agent 和网管系统两部分。代理和网管系统之间是通过 UDP 传送 SNMP 报文进行通信。SNMP 工作机制如图 10-1。

图 1-1 SNMP 工作机制示意图



甲信 NView NNM 网管系统能够提供友好的人机交互界面，方便网络管理员完成网络管理工作，主要可以实现以下功能：

- 向被管设备发送请求报文。
- 接收来自被管设备的响应报文和 Trap 报文，并显示结果。

代理是驻留在被管设备上的一个进程，主要实现以下功能：

- 接收、响应来自 NView NNM 网管系统的请求报文。
- 根据报文类型，对其进行读或写操作，并生成响应报文，返回给 NView NNM 网管系统。
- 根据各协议模块对触发条件进行定义，在达到触发条件后进入系统、退出系统、设备重新启动等，响应的模块通过代理向 NView NNM 网管系统发送 Trap 报文，报告设备的当前状态。

说明

代理可以同时配置多个版本，采用不同的版本与不同的网管系统交互。但是当代理和某个网管系统通信时，代理和该网管系统上的 SNMP 版本配置必须相同，才能正确互通。

协议版本

目前，SNMP 协议共有 v1、v2c 和 v3 三个版本。

- SNMPv1 采用共同体名（Community Name）认证机制。共同体名用来定义 SNMP 网管系统和 SNMP 代理之间的关系，起到了类似于密码的作用，用来限制 SNMP 网管系统对 SNMP 代理的访问。如果

SNMP 报文携带的共同体名在设备上没有认证通过，该报文将被丢弃。

- SNMPv2c 也采用共同体名认证机制。它在兼容 SNMP v1 的同时又扩充了 SNMP v1 的功能：支持更多的操作类型、数据类型和错误代码、能够更细致地区分错误。
- SNMPv3 采用了 USM（User-Based Security Model，基于用户的安全模型）和 VACM（View-based Access Control Model，基于视图的访问控制模型）安全机制。用户可以设置认证和加密功能，通过有无认证和有无加密等功能组合，可以为 SNMP 网管系统和 SNMP 代理之间的通信提供更高的安全性。认证用于验证报文发送方的合法性，避免非法用户的访问；加密则是对网管系统和代理之间的传输报文进行加密，以免被窃听。

设备同时支持 SNMP 的 v1、v2c、v3 三个版本。

MIB

MIB（Management Information Base，管理信息库）是网管系统能够管理的所有的对象的集合。它定义了被管理对象的一些属性：

- 名字
- 访问权限
- 数据类型

通过对这些数据项目的存取访问，就可以得到与设备相关的统计内容。每个代理都有自己的 MIB。MIB 可以看成是网管系统和代理之间的一个接口，通过这个接口，网管系统可以对代理中的每一个被管对象进行读/写操作，从而达到管理和监控设备的目的。

MIB 采用树形结构进行存储，它的根在最上面，没有名字。树的节点表示被管理对象，它可以从根开始的一条路径唯一地识别（OID）。SNMP 协议报文通过遍历 MIB 树形目录中的节点来访问网络中的设备。

设备支持标准的 MIB 库和甲信自定义的 MIB 库。

1.1.2 配置准备

场景

当用户需要通过网管系统登录交换机设备时，应首先对设备配置 SNMP 基本功能。

前提

在配置 SNMP 之前，需完成以下任务：

- 配置路由协议，使设备和网管系统之间路由可达。

1.1.3 SNMP 的缺省配置

设备上 SNMP 的缺省配置如下。

功能	缺省值
SNMP 服务	缺省关闭，需要手动配置开启
SNMP 视图	缺省存在：internet 视图
SNMP 共同体	缺省存在：public、private 共同体
SNMP 访问组	缺省不存在
SNMP 用户	缺省不存在
SNMP 用户和访问组的映射关系	无
网管人员的标识和联系方法	support@JX.com
设备放置的物理位置	World China JX
Trap 状态	使能
SNMP 目标主机地址	无
SNMP 引擎 ID	800022B603000000111233

1.1.4 配置 SNMP v1/v2c 基本功能

SNMP Agent 为了保护自身及其管理的 MIB 不被非法访问，提出了共同体的概念。在某一个共同体内的管理站必须在所有对 Agent 的操作中使用该共同体的名字，否则其请求不被受理。

共同体名是用不同的字符串来标识不同的 SNMP 团体。不同的共同体可以具有只读（read-only）或读写（read-write）访问权限。具有只读权限的团体只能对设备信息进行查询，具有读写权限的团体除了可以对设备信息进行查询之外还可以对设备进行配置。

SNMP v1/v2c 采用共同体名认证方案，与设备认可的共同体名不符合的 SNMP 报文将被丢弃。

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**configure**	进入全局配置模式。
2	JX(config)#**snmp server start**	启动 SNMP 服务端功能。
3	JX(config)#**snmp mib-view** *view-name* { **included** | **excluded** } *oid-tree* [**mask** *subtree-mask*]	创建 SNMP 视图，并配置访问的 MIB 变量范围。 缺省视图是 internet，范围包括 MIB 树中“1.3.6”节点以下的所有 MIB 变量。
4	JX(config)#**snmp community** { **read** | **write** } { **cipher** | **plain** } *community-name* [**mib-view** *view-name* |**acl-ipv4** *acl-ipv4-number* |**acl-ipv6** acl-ipv6-number]*	创建共同体名并配置对应的视图和访问权限。 如果没有填写 **mib-view** *view-name* 选项，则采用缺省视图 internet。

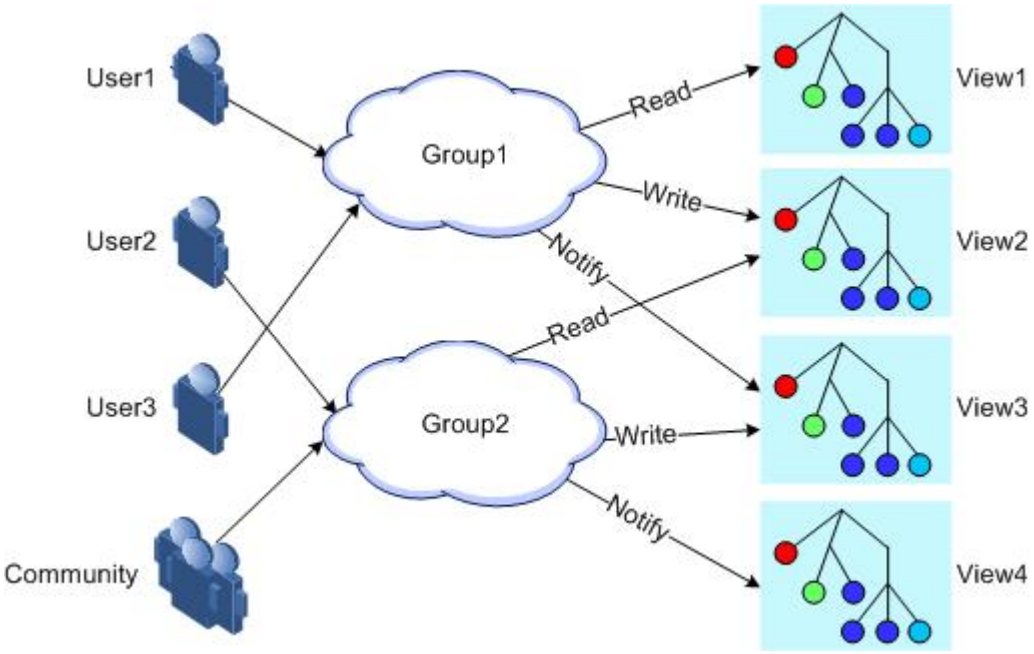
1.1.5 配置 SNMP v3 基本功能

SNMPV3 采用 USM 基于用户的认证机制。USM 提出了访问组（Group）的概念：一个或多个用户对应于一个访问组，每个访问组设定相应的读、写、通告视图，访问组中的用户拥有在该视图内的权限。发送 Get 和 Set 等请求的用户所在的访问组必须具有和其请求相应的权限，否则请求不被受理。

如图 10-2 所示，网管站采用 SNMP v3 对交换机正常的访问，需要进行的配置如下：

- 配置用户
- 确定用户属于哪个访问组
- 配置访问组拥有的视图权限
- 创建视图

图 1-2 SNMP v3 认证机制示意图



请在设备上进行以下配置。

步骤	配置	说明
1	JX#**configure**	进入全局配置模式。
2	JX(config)#**snmp mib-view** *view-name* { **included** | **excluded** } *oid-tree* [**mask** *subtree-mask*]	创建 SNMP 视图，并配置访问的 MIB 变量范围。 缺省视图是 internet，范围包括 MIB 树中“1.3.6”节点以下的所有 MIB 变量。

步骤	配置	说明
3	JX(config)#**snmp group** *group-name* { **authentication** | **privacy** | **noauthentication** } [**read-view** *read-view-name* | **write-view** *write-view-name* | **notify-view** *notify-view-name*]*	配置访问组相关。
4	JX(config)#**snmp user** *user-name* **group** *group-name* **authentication** { **md5** | **sha** } *authpassword* **privacy** { **des** | **aes** } *privkeypassword*	创建用户并绑定组，并配置认证和加密。
5	JX(config)#**snmp user** *user-name* **group** *group-name* **authentication** { **md5** | **sha** } *authpassword*	创建用户并绑定组，并配置认证方式。
6	JX(config)#**snmp user** *user-name* **group** *group-name* [**acl-ipv4** *acl-ipv4-number* | **acl-ipv6** *acl-ipv6-number*]*	创建用户并绑定访问组。

1.1.6 配置 SNMP 其他信息

SNMP v1、v2c、v3 均支持以下信息的配置。

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**configure**	进入全局配置模式。
2	JX(config)#**snmp contact** *string*	配置联系方法。
3	JX(config)#**snmp location** *location-string*	配置设备当前所在地址信息。
4	JX(config)#**snmp auth-fail-reply** { **enable** | **disable** }	配置在认证失败时是否发送 SNMP 响应报文。
5	JX(config)#**snmp auth-trap** { **enable** | **disable** }	配置是否使能认证 Trap。
6	JX(config)#**snmp auth-fail-count** *fail-count-value*	配置 SNMP 认证失败次数。
7	JX(config)#**snmp reauth-interval** *reauth-interval-value*	配置 SNMP 重认证时间。
8	JX(config)#**snmp packet max-size** { *size-value*|**default** }	配置 SNMP 收发包最大长度。
9	JX(config)#**snmp reply-source-ip** { **enable** | **disable** }	配置将所输入的 IP 地址作为应答报文的源地址
10	JX(config)#**snmp udp-port** { *port-number* | **default** }	配置 SNMP 服务端口号。

1.1.7 配置 Trap

说明

SNMP v1、v2c 和 v3 的 Trap 配置步骤除了目标主机的配置，其余都是一样的，请根据需要进行选择。

Trap 是设备主动向网管系统发送的未经请求的信息，用于报告一些紧急的重要事件。

在配置 Trap 功能之前，需完成以下任务：

- 配置 SNMP 的基本功能。如果使用 SNMP v1 和 v2c 版本需要配置共同体名；如果使用 SNMP v3 版本需要配置用户名和 SNMP 视图。
- 配置路由协议，使设备和网管系统之间路由可达。

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**configure**	进入全局配置模式。
2	JX(config)#**snmp server { start | stop }**	启动 SNMP 服务端功能。
3	JX(config)#**snmp trap-host** *ip-address* **{ v1 | v2 } [port** *port-id* **| securityname** *security-name* **| vpn-instance** *vpn-name* **]***	配置基于 SNMP v1 和 SNMP v2c 的 Trap 目标主机(IPV4)。
4	JX(config)#**snmp-ipv6 trap-host** *ipv6-address* **{ v1 | v2 } [port** *port-id* **| securityname** *security-name* **| vpn-instance** *vpn-name* **]***	配置基于 SNMP v1 和 SNMP v2c 的 Trap 目标主机(IPV6)。
5	JX(config)#**snmp trap-host** *ip-address* **v3 { authentication | privacy } securityname** *security-name* **[port** *port-id* **| vpn-instance** *vpn-name* **]**	配置基于 SNMP v3 的 Trap 目标主机(IPV4)。
6	JX(config)#**snmp-ipv6 trap-host** *ipv6-address* **v3 { authentication | privacy } securityname** *security-name* **[port** *port-id* **| vpn-instance** *vpn-name* **]***	配置基于 SNMP v3 的 Trap 目标主机IPV6)。

1.1.8 检查配置

配置完成后，请在设备上执行以下命令检查配置结果。

序号	检查项	说明
1	JX#**show snmp information**	查看 SNMP 全局的信息。 包含本地 SNMP 引擎 ID，网管人员的标识及联系方法，设备所在位置，Trap 开关状态信息，SNMP 服务开启状态，服务端口号。

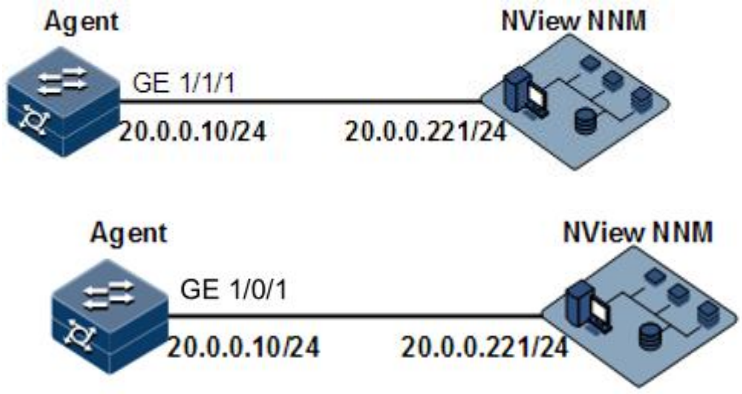
序号	检查项	说明
2	JX#show snmp group	查看 SNMP 访问组的信息。
3	JX#show snmp community	查看 SNMP 共同体的配置信息。
4	JX#show snmp config	查看 SNMP 的基本配置信息。
5	JX#show snmp trap-host	查看 Trap 目标主机信息。
6	JX#show snmp statistics	查看 SNMP 的统计信息。
7	JX#show snmp user	查看 SNMP 用户信息。
8	JX#show snmp mib-view	查看 SNMP 的视图信息。

1.1.9 配置 SNMP v1/v2c 和 Trap 示例

组网需求

如图 10-3 所示，NView NNM 网管系统与交换机设备之间路由可达，NView NNM 通过 SNMP v1/v2c 可以查看远程交换机的对应视图下的 MIB，交换机出现紧急情况时可以主动向 NView NNM 发送 Trap。

图 1-3 SNMP v1/v2c 组网示意图



配置步骤

步骤 1 配置交换机设备的 IP 地址。

```
JX#config
JX(config)#interface meth 0/0/0
JX(config-meth-0/0/0)#ip address 192.168.62.100/24
JX(config-meth-0/0/0)#quit
```

步骤 2 启动 SNMP 服务端功能。

```
JX(config)#snmp server start
```

步骤 3 配置 Trap 告警。

```
JX(config)#snmp trap-host 192.168.62.1 v2
```

检查结果

通过 **show ip interface** 查看 IP 地址配置是否正确。

```
JX#show ip interface
Total number: 2
 Interface         State(a/o) Addr/Prefix       Role     Type
Vpn-instance
-------------------------------------------------------------
-----------------------------------
 loopback-0        up/up     127.0.0.1/8       primary   auto
N/A
 meth-0/0/0        up/up     192.168.62.100/24  primary
static    N/A
-------------------------------------------------------------
-----------------------------------
```

通过 **show snmp config** 查看 SNMP 配置是否正确。

```
JX#show snmp config
Software Version: SNMP_VL3.00.00.00
!
snmp server start
snmp trap-host 192.168.62.1 securityname
AJ35Grrnpx3xRv_UdbJBiGsl3Dwxp932il5pfsKvs8X-GN6R49ihEydRaxBXk
M5V-w
```

通过 **show snmp trap-host** 查看目标主机配置是否正确。

```
JX#show snmp trap-host
Trap-host : 192.168.62.1
-------------------------------------------------------------
-------------------
Status             : ACTIVE
Udp Port           : 162
MP Modle           : V2
Security Level      : None
Security Name       :
AJ35Grrnpx3xRv_UdbJBiGsl3Dwxp932il5pfsKvs8X-GN6R49ihEydRaxBXk
M5V-w
Vpn Instance Name    : public
-------------------------------------------------------------
-------------------
```

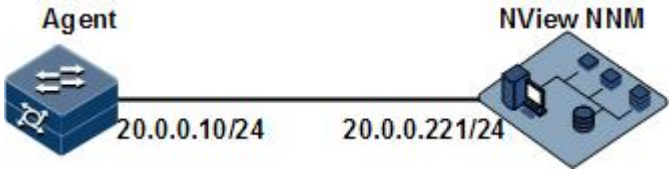
1.1.10 配置 SNMP v3 和 Trap 示例

组网需求

如图 10-4 所示，NView NNM 网管系统与 Agent 之间有可达路由，NView NNM 通过 SNMP v3 对 Agent 进行监控，Agent 出现紧急情况时可以主动向 NView NNM 发送 Trap。

缺省情况下，交换机设备上存在 VLAN 1，所有物理接口属于 VLAN 1。

图 1-4 SNMP v3 和 Trap 组网示意图



配置步骤

步骤 1 配置交换机设备的 IP 地址。

```
JX#config
JX(config)#interface meth 0/0/0
JX(config-meth-0/0/0)#ip address 192.168.62.100/24
JX(config-meth-0/0/0)#quit
```

步骤 2 配置 SNMP v3 访问。

启动 SNMP 服务端功能。

```
JX(config)#snmp server start
```

创建 g1 的访问组，安全等级为认证但不加密。

```
JX(config)#snmp group g1 authentication
```

创建用户 u1 为认证但不加密，绑定访问组 g1，采用 md5 鉴别算法，口令为 JX。

```
JX(config)#snmp user u1 group g1 authentication md5 JX
```

步骤 3 配置 Trap 告警，配置的 trap-host 类型必须与 user 的认证类型一致，不存在包含关系。

```
JX(config)#snmp trap-host 192.168.62.1 v3 authentication
securityname u1
```

检查结果

通过 **show snmp group** 查看 SNMP 访问组信息配置是否正确。

```
JX#show snmp group
Group                          Security     ReadView
WriteView                       NotifyView
------------------------------------------------------------
------------------------------------------------------------
---
g1                            authNoPriv   internet
-                             -
------------------------------------------------------------
------------------------------------------------------------
---
```

通过 **show snmp user** 查看用户和访问组的映射关系配置是否正确。

```
JX#show snmp user
```

```
User                            Group                           Auth
Priv    Filter
-------------------------------------------------------------
-------------------------------------
u1                              g1                              MD5
no-priv  N/A
-------------------------------------------------------------
-------------------------------------
```

通过 **show snmp trap-host** 查看 Trap 目标主机配置是否正确。

```
JX#show snmp trap-host
Trap-host : 192.168.62.1
-------------------------------------------------------------
------------------
Status             : ACTIVE
Udp Port           : 162
MP Modle           : V3
Security Level      : Auth
Security Name       : u1
Vpn Instance Name    : public
-------------------------------------------------------------
------------------
```

通过 **show snmp config** 查看所有配置信息是否正确。

```
JX#show snmp config
Software Version: SNMP_VL3.00.00.00
!
snmp server start
snmp trap-host 192.168.62.1 v3 authentication securityname u1
snmp group g1 authentication
snmp user u1 group g1 authentication md5
ACs87nMCx4RbzvvPS5lo3zDtbzu5xyzSZaDGmHOuqsc0
```

1.2 RMON

1.2.1 简介

RMON（Remote Network Monitoring，远端网络监控）是 IETF 制定的一种可以通过不同的网络代理 Agent 和网管中心进行网络数据监控的标准。

RMON 基于 SNMP 体系结构实现，包括网管中心和运行在各网络设备上的代理 Agent 两部分。在 SNMP 基础上增加了子网流量、统计、分析能力，可以实现对一个网段乃至整个网络的监控，而 SNMP 只能监控单个设备局部信息，对一个网段的监控非常困难。

RMON 中代理 Agent 一般称为探测程序，RMON Probe（RMON 探测器）能够统计和分析子网的通信性能指标，无论何时发现网络故障，RMON Probe 都能够上报网管中心，并描述不正常情况的捕获信息，网管中心无需对设备进行不停的轮询。RMON 可以比 SNMP 更主动、有效地监测远程设备，网络管理员可以更快地跟踪网络、网段或设备出现的故障。该方法减少了网管中心同代理 Agent 间的数据流量，使简单而有力地管理大型网络成为可能，弥补了 SNMP 在日益扩大的分布式互联中所面临的局限性。

RMON Probe 收集数据的方式：

- 分布式 RMON：利用专用的 RMON Probe 收集数据，网管中心直接从 RMON Probe 获取管理信息并控制网络资源。
- 嵌入式 RMON：将 RMON Agent 直接嵌入网络设备（如交换机）中，使它们成为带 RMON Probe 功能的网络设备。网管中心使用 SNMP 的基本操作与 RMON Agent 交换数据信息，收集网络管理信息。

设备采用嵌入式 RMON。如图 10-5 所示，在设备上实现了 RMON Agent 功能。通过该功能，管理站可以获得与被管网络设备接口相连的网段上的整体流量、错误统计和性能统计等信息，从而实现对一个网段的监控。

图 1-5 RMON 应用示意图

RMON MIB 中按照功能分成 9 个组，目前实现了 RMON 的四个功能组：即统计组、历史组、告警组和事件组。

- 统计组：负责收集在一个接口上的统计信息，包括接收到的报文计数和大小分布统计。
- 历史统计组：类似于统计组，但它是在一个指定的检测周期内收集统计信息。
- 告警组：在指定的时间间隔内，监视一个指定的管理信息库（MIB）对象，并且设定上升阈值和下降阈值，若监视对象达到阈值则触发一个事件。
- 事件组：配合告警组使用，当告警触发一个事件时，用来记录相应的事件信息，如发送 Trap 信息，写入日志等操作。

1.2.2 配置准备

场景

当用户需要对某一网段进行监控或流量统计时，可以配置 RMON。

RMON 是一种比 SNMP 更高效的监控手段。用户只需要指定告警阈值，超出该阈值时设备将会主动发送告警信息，而不用去获取变量信息。减少管理设备和被管理设备的通信量，对网络进行简单有效的管理。

前提

设备和网管系统之间链路可达。

1.2.3 RMON 的缺省配置

设备上 RMON 的缺省配置如下。

功能	缺省值
统计组	无
历史统计组	禁止
告警组	无
事件组	无

1.2.4 配置 RMON 统计功能

RMON 统计功能可以设置对接口的统计，包括接口收发报文、过小或过大包、冲突、循环冗余校验和错误数、丢弃报文，接收报文长度、碎片、广播、多播、单播消息等。

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**configure**	进入全局配置节点。
2	JX(config)#**interface** *interface-type interface-number*	进入接口节点下。
3	JX(config-ge-1/0/*)#**rmon statistics** *index-number* [**owner** *owner-name*]	添加该接口的 RMON 统计功能并配置相关参数。

说明

当在接口下使用 **no rmon statistics** *index-number* 命令关闭该接口的统计功能时，是指用户不能继续获取该接口的统计数据了，而不是接口不再进行数据统计了。

1.2.5 配置 RMON 历史统计功能

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**configure**	进入全局配置节点。
2	JX(config)#**interface** *interface-type interface-number*	进入接口节点下。
2	JX(config-ge-1/0/*)#**rmon history** *index-number* **bucket** *buckets-number* **interval** *sampling-interval* [**owner** *owner-name*]	添加该接口的 RMON 历史统计功能并配置相关参数。

说明

当在接口下使用 **no rmon history** *index-number* 命令禁止该接口的历史组统计功能时，不再进行数据收集统计，并且将以前收集的所有历史数据清除。

1.2.6 配置 RMON 告警组

可通过设置 RMON 一个告警组实例（*alarm-id*），监控一个 MIB 变量（*object-id*）。当被监控数据的值越过定义的阈值时会产生告警事件，再按照告警事件的定义进行记录日志或向网管站发送 Trap 信息。

所监控的 MIB 变量必须是真实存在的，并且数据值类型设置正确。

- 如果在设置时，变量不存在或值类型不正确，则返回错误。
- 在已经设置成功的告警中，如果后期该变量无法被采集，则该告警就被关掉，若想重新监控该变量，必须重新设置。

只要事件表中配置了上限或下限其中一个事件，符合条件便会触发相应的告警。如果告警上限和下限所对应事件（*rising-event-id*、*falling-event-id*）在事件表中均没有配置，即使达到了告警条件也不会产生告警。

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**configure**	进入全局配置节点。
2	JX(config)#**rmon alarm** *alarm-id object-id query-interval* { **absolute** | **delta** } **rising-threshold** *rising-threshold rising-event* **falling-threshold** *falling-threshold falling-event* [**startup-alarm** { **rising** | **falling** | **risingorfalling** } | **owner** *owner-name*]*	在 RMON 告警组中添加告警实例，并配置相关参数。

1.2.7 配置 RMON 事件组

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**configure**	进入全局配置节点。
2	JX(config)#**rmon event** *event-id* { **log** | **trap** | **both** } [**description** *string* | **owner** *owner-name*]*	在 RMON 事件组中添加事件，并配置事件的处理方式。

1.2.8 检查配置

配置完成后，请在设备上执行以下命令检查配置结果。

序号	检查项	说明
1	JX#**show rmon config**	查看 RMON 配置信息。
2	JX#**show rmon alarm**	查看 RMON 告警组信息。
3	JX#**show rmon event**	查看 RMON 事件组信息。
4	JX#**show rmon statistics**	查看 RMON 统计组信息。
5	JX#**show rmon history**	查看 RMON 历史统计组配置信息
6	JX#**show rmon history** *index-number*	查看指定 RMON 历史统计组信息。
7	JX#**show rmon history statistics**	查看指定 RMON 历史统计组统计信息。
8	JX#**show rmon log**	查看 RMON 事件组的日志信息。

1.2.9 维护

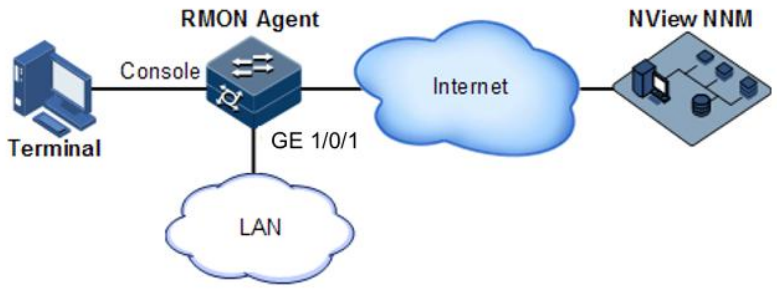
无

1.2.10 配置 RMON 告警组应用示例

组网需求

如图 10-6 所示，交换机设备作为 Agent，通过 Console 口连接配置终端，通过 Internet 连接远端 NNM 系统。使能 RMON 统计功能并对 ge 1/0/1 接口进行性能统计，当接口在一段时间内收到的报文数量超过设置的阈值后，记录日志并发送 Trap 告警。

图 1-6 RMON 典型应用组网示意图



配置步骤

步骤 1 创建索引号为 1 的事件，该事件用于记录并发送,描述字符串为 Falling-etherStatsBroadcastPkts 的日志信息，该日志信息的所有者为 **system**。

```
JX#configure
JX(config)#rmon event 1 both description
Falling-etherStatsBroadcastPkts owner system
```

步骤 2 在接口 ge 1/0/1 下创建统计表，该统计表的所有者也为 **system**

```
JX(config-ge-1/0/1)#rmon statistics 1 owner system
```

步骤 3 创建索引号为 10 的告警项，该告警项用于监控 MIB 变量 etherStatsBroadcastPkts.1 即 1.3.6.1.2.1.16.1.1.1.6.1，每 20 秒检查一次，如果该变量的取值升到 100 以上或者降到 15 以下，便触发 Trap 告警，该告警信息的所有者也为 **system**。

```
JX(config)#rmon alarm 10 1.3.6.1.2.1.16.1.1.1.6.1 20 absolute
rising-threshold 100 1 falling-threshold 15 1 owner system
```

检查结果

通过 **show rmon alarm** 命令查看设备上是否有告警组信息。

```
JX#show rmon alarm
  RMON Alarm:10
       Interval:20
       Source OID:1.3.6.1.2.1.16.1.1.1.6.1
       Sample Type:absolute value
       Alarm Value:0
       Startup Alarm:risingOrFallingAlarm
       Rising Threshold:100
       Rising Event:1
       Falling Threshold:15
       Falling Event:1
       Owner:system
       Status:valid
```

通过 **show rmon event** 命令查看设备上是否有事件组信息。

```
JX#show rmon event
  RMON Event:1
       Type:trap&log
       Status:valid
       Lastsent time:0 days 12 hours 6 minutes 1 seconds
       Description:Falling-etherStatsBroadcastPkts
       Owner:system
```

通过 **show rmon log** 命令查看设备上是否有事件记录的日志信息。

```
JX#show rmon log
  RMON Log:1/1
       Time:0 days 12 hours 6 minutes 1 seconds
       Description:alarm falling
10,1.3.6.1.2.1.16.1.1.1.6.1,1,0,15
```

当告警事件被触发时，在 NNM 系统的告警管理部分也可以查看相应的记录。

1.3 LLDP

1.3.1 简介

随着网络规模的扩大，网络设备的增多，网络拓扑日趋复杂，对网络的管理变得尤为重要。为了跟踪网络拓扑信息的变化，许多网络管理软件都采用“自动发现”功能来跟踪网络拓扑的变化，但大多数网络管理软件只能分析到网络层拓扑结构，无法确定设备通过哪些接口与其他设备相连。

LLDP（Link Layer Discovery Protocol，链路层发现协议）是由 IEEE 802.1AB 定义的一种链路层发现协议。网络管理系统可以通过该协议快速掌握二层网络的拓扑及其变化情况。

LLDP 将本地设备的信息组织成不同的 TLV（Type Length Value，类型/长度/值单元），并封装在 LLDPDU（Link Layer Discovery Protocol Data Unit，链路层发现协议数据单元）中发送给直连的邻居，同时将邻居发来的信息以标准 MIB（Management Information Base，管理信息库）的形式保存起来，以供网管系统查询及判断链路的通信状况。

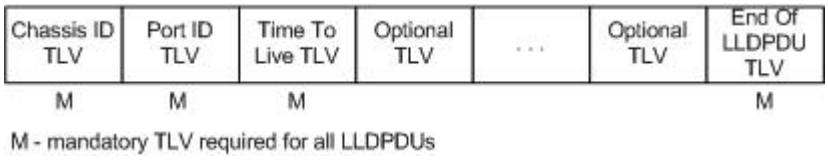
基本概念

LLDP 报文是指在数据单元封装了 LLDPDU 的以太网报文。

LLDPDU 是 LLDP 报文的数据单元。在组成 LLDPDU 之前，设备先将本地信息封装成 TLV，再由若干 TLV 组合成一个 LLDPDU，封装在以太网数据部分进行传送。

如图 10-7 所示，LLDPDU 由若干个 TLV 组合而成，其中包含四个必选的 TLV 和若干个可选的 TLV。

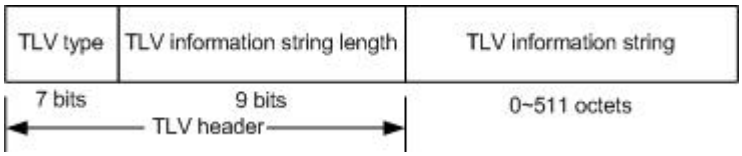
图 1-7 LLDPDU 结构图



TLV 是组成 LLDPDU 的单元，表示一个对象的类型、长度和信息的单元。

TLV 的结构如图 10-8 所示，每个 TLV 代表一个本端信息。例如设备 ID，接口 ID 等各自对应 Chassis ID TLV，Port ID TLV 固定的 TLV。

图 1-8 基本 TLV 结构图



TLV 类型值对应如表 10-1 所示，目前只用到其中的 0～8 种类型。

表 1-1 TLV 类型

TLV 类型	说明	是否必选
0	End Of LLDPDU，表示 LLDP 报文结束	必选
1	Chassis Id，发送设备的 MAC 地址	必选
2	Port Id，LLDP 报文发送端的接口	必选
3	Time To Live，本设备信息在邻居设备上的老化时间	必选
4	Port Description，以太网接口的描述信息	可选
5	System Name，设备名称	可选
6	System Description，系统描述	可选
7	System Capabilities，系统的主要功能以及已使用的功能项	可选
8	Management Address，管理地址	可选

组织定义 TLV 属于可选的 TLV 集合，根据用户的实际需要在 LLDPDU 中发布。目前比较常见的组织定义 TLV 如下。

表 1-2 IEEE 802.1 组织定义的 TLV

TLV 类型	TLV 说明
Port VLAN ID TLV	端口的 VLAN 标识符
Port And Protocol VLAN ID TLV	端口的协议 VLAN 标识符
VLAN Name TLV	端口的 VLAN 名称
Protocol Identity TLV	端口支持的协议类型

表 1-3 IEEE 802.3 组织定义的 TLV

TLV 类型	TLV 说明
MAC/PHY Configuration//Status TLV	端口的速率双工状态、是否支持并使能自动协商功能
Power Via MDI TLV	端口的供电能力
Link Aggregation TLV	端口的链路聚合能力及当前的聚合状态
Maximum Frame Size TLV	端口所能传输的最大的帧的大小

LLDP 工作原理

LLDP 是一种点对点单向发布协议，通过本机向对端周期性的发送 LLDP 报文（或者本端信息有变化时发送 LLDP 报文），通知对端本机的链路状态。

其数据流如下：

- 发送时，设备从 NMS 获取所选择 TLV 需要的系统信息，以及从 LLDP MIB 中获得配置信息，生成 TLV，组成 LLDPDU，封装成 LLDP 报文发送给对端。
- 对端接收到 LLDP 报文后，对端设备会解析获得的各个 TLV 信息，如果有变更，将信息更新至 LLDP 的邻居 MIB 表中，并通知 NMS。

本端设备信息在邻居节点中老化时间 TTL（Time to live），可通过修改老化系数参数值调整，向邻居节点发送 LLDP 报文，邻居节点收到 LLDP 报文后，调整其邻居节点（即发送端）信息的老化时间。老化时间计算公式，TTL=Min{65535，(interval×hold-multiplier)}，其中：

- interval 表示设备向邻居节点发送 LLDP 报文的时间周期。
- hold-multiplier 表示设备信息在邻居节点的老化系数。

1.3.2 配置准备

场景

当用户通过 NView NNM 系统获取设备之间的连接信息，进行拓扑发现时，需要在设备之间使能 LLDP 功能，向邻居互相通告自己的信息，以及存储邻居信息，方便 NView NNM 系统查询。

前提

无

1.3.3 LLDP 的缺省配置

设备上 LLDP 的缺省配置如下。

功能	缺省值
LLDP 全局使能或禁止	禁止
接口 LLDP 功能状态	使能
延迟发送定时器	2s
周期发送定时器	30s
老化系数	4
重启定时器	2s
告警使能或禁止	去使能
告警通知定时器	5s

功能	缺省值
LLDP 报文目的 MAC 地址	0180.c200.000e

1.3.4 使能全局 LLDP 功能

注意

禁止全局 LLDP 功能后，不能立即再使能，必须等重启定时器超时后才能再次使能。

通过 NView NNM 系统获取设备之间的连接信息，进行拓扑发现时，需要在设备之间使能 LLDP 功能，向邻居互相通告自己的信息，以及存储邻居信息，方便 NView NNM 系统查询。

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**configure**	进入全局配置模式。
2	JX(config)#**lldp { start | stop }**	使能全局 LLDP 功能，使用 stop 格式禁用该功能。 start：使能 LLDP 功能 stop：禁用 LLDP 功能

1.3.5 使能接口 LLDP 功能

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**configure**	进入全局配置模式。
2	JX(config)#**interface** *interface-type interface-number*	进入物理接口配置模式。
3	JX(config-ge-1/0/1)#**lldp admin-status { tx-only | rx-only | rx-tx | disable }** Example: JX(config-ge-1/0/1)#**lldp admin-status tx-only**	使能接口 LLDP 功能，使用 disable 格式禁用该功能。 tx-only：只发 LLDP 报文 rx-only：只收 LLDP 报文 rx-tx：收发 LLDP 报文 disable：禁用 LLDP 功能

1.3.6 全局配置 LLDP 基本功能

注意

配置延时发送定时器和周期发送定时器时，延时发送定时器的取值要小于或等于周期发送定时器取值的四分之一。

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**configure**	进入全局配置模式。
2	JX(config)#**lldp tx-interval** { <5-32768> | **default** } Example: JX(config)#**lldp tx-interval 10**	（可选）配置 LLDP 报文的周期发送定时器。 second：LLDP 报文的发送周期，整数形式，取值范围是 5～32768，单位是秒 default：2 秒
3	JX(config)#**lldp tx-delay** { <1-8192> | **default** } Example: JX(config)#**lldp tx-delay 5**	可选）配置 LLDP 报文的延迟发送定时器。 second：发送延迟时间，整数形式，取值范围是 1～8192，单位是秒 default：2 秒。
4	JX(config)#**lldp reinit-delay** { <1-10> | **default** } Example: JX(config)#**lldp reinit-delay 5**	可选）配置重启定时器。即设备禁止全局 LLDP 功能后，需要等待重启定时器设定的时间后才能重新使能全局 LLDP 功能。 second：重启延迟时间值，整数形式，取值范围是 1～10，单位是秒 default：2 秒
5	JX(config)#**lldp tx-hold** { <2-10> | **default** } Example: JX(config)#**lldp tx-hold 2**	（可选）配置报文发送间隔的倍数 multiple：倍数，整数形式，取值范围是 2～10，单位是倍 default：4。

1.3.7 接口配置 LLDP 基本功能

步骤	配置	说明
1	JX#**configure**	进入全局配置模式。
2	JX(config)#**interface** *interface-type interface-number* Example: JX(config)#interface **ge** 1/0/1	进入物理接口配置模式。

步骤	配置	说明
3	JX(config-ge-1/0/1)#**lldp management-address** { A.B.C.D } { **enable** | **disable** }	（可选）在端口下配置 LLDP 的管理地址 A.B.C.D　管理地址；Enable 使能；disable 去使能

1.3.8 配置 LLDP 告警功能

当网络自身发生变化时，需要使能 LLDP 告警通知功能，及时向 NView NNM 系统发送拓扑信息更新告警。

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**configure**	进入全局配置模式。
2	JX(config)#**lldp trap-interval** { <5-5600> | **default** } Example: JX(config)#**lldp notification-interval 10**	（可选）配置 lldp 邻居变化时间通告间隔定时器 second：重启延迟时间值，整数形式，取值范围是 5～5600，单位是秒 default：5 秒
3	JX(config)#**interface** *interface-type interface-number* Example: JX(config)#interface ge-1/0/1	进入物理接口配置模式。
4	JX(config-ge-1/0/1)#**lldp trap** { **enable** | **disable** }	配置 LLDP 告警 Trap 使能/去使能

1.3.9 配置 TLV

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**configure**	进入全局配置模式。
2	JX(config)#**interface** *interface-type interface-number*	进入物理接口配置模式。
3	JX(config-ge-1/0/*)#**lldp basic-tlv-tx** { **port-description** | **system-name** | **system-description** | **system-capability** | **all** } { **enable** | **disable** }	（可选）在端口下配置 LLDP 报文的基本 TLV port-description 端口描述；system-name 系统名称；system-description 系统描述；all 所有；s enable 使能；disable 去使能

步骤	配置	说明
4	JX(config-ge-1/0/*)#**lldp dot1-tlv- tx port-vid { enable | disable }** JX(config-ge-1/0/1)#**lldp dot1-tlv-tx protocol-vid** *vlan-list* **{ enable | disable }** JX(config-ge-1/0/1)#**lldp dot1-tlv-tx vlan-name** *vlan-list* **{ enable | disable }**	可选)配置端口下设置关于 802.1 组织定义的 TLV 的相关配置。 port-vid 端口 vlan；protocol-vid 协议 vlan；vlan-namevlan 名称；enable 使能；disable 去使能
5	JX(config-ge-1/0/*)#**lldp dot3-tlv-tx { mac-phy | power | link-aggregation | max-frame-size | all } { enable | disable }**	（可选）配置端口下关于 802.3 组织定义的 TLV 的相关配置 mac-phy 端口的速率；power 端口的供电能力；link-aggregation 链路聚合；max-frame-size 最大帧长度；all 所有；enable 使能；disable 去使能
6	JX(config-ge-1/0/1)#**lldp med-tlv- tx { capabilities | network-policy | location | extended-pse | extended-pd | inventory | all } { enable | disable }**	（可选）配置端口下与 MED 相关的配置 Capabilities 能力级；network-policy 支持的应用；location 端口位置标识信息；extended-pse 供电能力；extended-pd 供电能力；inventory 详细目录； all 所有；enable 使能；disable 去使能
7	JX(config-ge-1/0/1)# **lldp voice-vlan { untagged | vlan** *vlan-id* **[cos {***cos-value* **| default} | dscp {***dscp-value* **| default}] }**	（可选）配置端口下与 network-policy tlv 封装 voice vlan 相关的配置 Untagged 配置终端设备发送语音流量时不带 VLAN ID；vlan-id 配置 voice-vlan 的 VLAN ID；cos-value 配置 CoS 优先级，默认值为 5；dscp-value 配置 DSCP 优先级，默认值为 46

1.3.10 检查配置

配置完成后，请在设备上执行以下命令检查配置结果。

序号	检查项	说明
1	JX# **show lldp config**	显示 lldp 配置
2	JX#**show lldp information**	查看 LLDP 本地系统信息。
3	JX#**show lldp remote [** *interface-type interface-number* **]**	查看 LLDP 邻居信息。
4	JX#**show lldp statistics**	查看 LLDP 统计信息。
5	JX#**show lldp interface [** *interface-type interface-number* **]**	查看 LLDP 端口信息。

1.3.11 维护

用户可以通过以下命令维护 LLDP 特性。

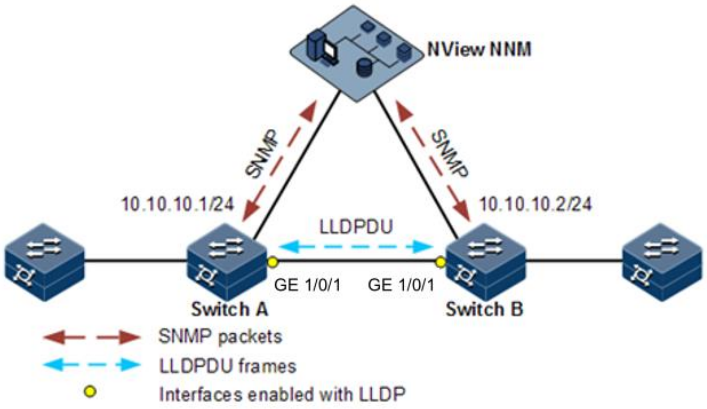
命令	说明
JX(config-ge-1/0/1)# **lldp port statistics counter**	接口下清除 LLDP 统计信息。

1.3.12 配置 LLDP 基本功能示例

组网需求

如图 10-9 所示，交换机和 NView NNM 系统相连，在 Switch A 和 Switch B 之间使能 LLDP 协议，则两设备之间二层链路的变化情况，可以通过 NView NNM 系统查询。如果邻居老化、新增邻居、邻居信息变化时会向 NView NNM 系统上报 LLDP 告警。

图 1-9 配置 LLDP 基本功能组网示意图



配置步骤

步骤 1 配置全局使能 LLDP 并使能 LLDP 告警。

配置 Switch A。

```
JX#hostname SwitchA
SwitchA#config
SwitchA(config)#lldp start
```

配置 Switch B。

```
JX#hostname SwitchB
SwitchB#config
SwitchB(config)#lldp start
```

步骤 2 配置管理 IP 地址。

配置 Switch A。

```
SwitchA(config)# vlan 10
SwitchA(config)# interface vlan 10
SwitchA(config)# interface GE 1/0/1
SwitchA(config-ge-1/0/1)# port hybrid vlan 10 tagged
SwitchA(config-ge-1/0/1)# port hybrid pvid 10
SwitchA(config-ge-1/0/1)#exit
SwitchA(config)#interface vlan 10
SwitchA(config-vlan1024)# ip address 10.0.0.1/24
SwitchA(config-vlan1024)#exit
```

配置 Switch B。

```
SwitchB(config)# vlan 10
SwitchB (config)# interface vlan 10
SwitchB (config)# interface GE 1/0/1
SwitchB (config-ge-1/0/1)# port hybrid vlan 10 tagged
SwitchB (config-ge-1/0/1)# port hybrid pvid 10
SwitchB (config-ge-1/0/1)#exit
SwitchB (config)#interface vlan 10
SwitchB (config-vlan1024)# ip address 10.0.0.2/24
SwitchB (config-vlan1024)#exit
```

步骤 3 配置 LLDP 属性。

配置 Switch A。

```
SwitchA(config)# lldp tx-interval 60
SwitchA(config)# lldp tx-delay 9
SwitchA(config)# lldp notification-interval 10
```

配置 Switch B。

```
SwitchA(config)# lldp tx-interval 60
SwitchA(config)# lldp tx-delay 9
SwitchA(config)# lldp notification-interval 10
```

检查结果

通过 **show lldp local config** 命令查看本地配置是否正确。

```
SwitchA# show lldp local
  LLDP local:
       Message tx-interval:60(s)
       Message tx-hold:4
       Reinit delay:2(s)
       Tx delay:9(s)
       Notification interval:10(s)
       Chassis type:MAC Address
       Chassis ID:f0f1:f2f3:0101
       System name:SIM-MPU
       System desc:JX-SWITCH-SIM
       System supported:Bridge/Switch,Router
       System capenabled:Bridge/Switch,Router

  Port ge-1/0/1:
       Admin status:TxRx
       Trap enable:no
```

```
       Support
tlv:port-description,system-name,system-description,system-ca
pability
       Enabled
tlv:port-description,system-name,system-description,system-ca
pability
       Port type:interface name
       Port ID:GE1/0/1
       Port description:GE1/0/1 SNMP-Index:537397249
       Number of remote system:1
       Number of MED remote system:0
……
```

通过 **show lldp remote** 命令查看邻居信息是否建立。

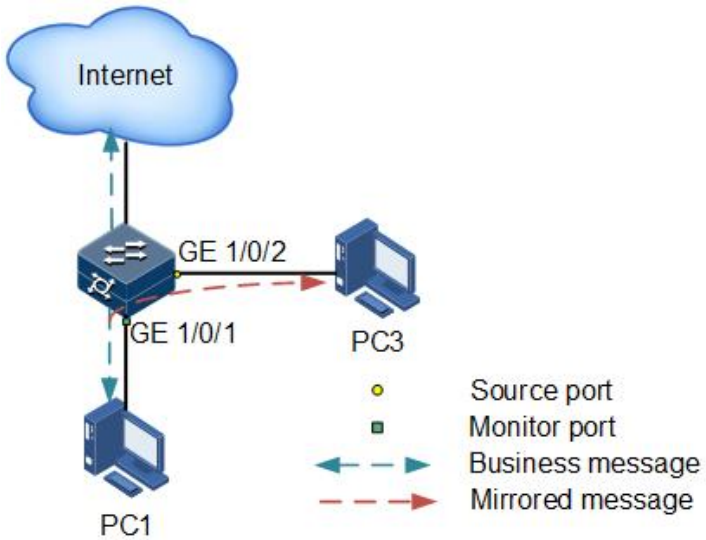
```
SwitchA#show lldp remote
Interface      Index TTL(s)  ChassId                    PortId
 SysName Vlan
ge-1/0/1       1     116     f0f1:f2f3:0201
 GE1/0/1                      SIM-MPU 10
ge-1/0/2       2     116     f0f1:f2f3:0201
 GE1/0/2                      SIM-MPU --
……
```

1.4 端口镜像

1.4.1 简介

接口镜像功能是指将指定源接口的某些报文镜像到目的接口，即监视接口，而不影响正常报文转发的功能。交换机设备用户使用该功能可以监控某个接口的报文接收和发送情况，并分析相关网络状况。

图 1-10 接口镜像功能原理示意图



接口镜像功能基本原理如上图所示。PC 1 通过交换机的 GE 1/0/1 与外网连接，PC 3 为监测 PC，通过交换机的 GE 1/0/2 与外网连接。

当要监测从 PC 1 发出的报文时，需要将 PC 1 所连接设备的 GE 1/0/1 指定为镜像源接口，并使能对入接口报文的镜像功能，而将 GE 1/0/2 指定为监视接口，即镜像目的接口。

当 GE 1/0/1 发出的业务报文进入交换机时，交换机将对入报文进行转发，并复制一份到监视接口（GE 1/0/2）。连接在监视接口的监控设备可以接收这些被镜像的报文，并进行相关的分析工作。

设备支持基于入接口和出接口的数据流镜像。镜像功能生效后，出/入镜像接口的报文会被复制一份到监视接口。监视接口与镜像接口不能为同一个接口。

1.4.2 配置准备

场景

接口镜像功能主要用在网络管理人员定期监控网络内数据类型及流量。

接口镜像功能是将需要被监控接口的流量复制到一个监视接口或者 CPU，从而抓取入/出接口出现故障或异常时的数据流，用来分析、发现问题根源并及时解决。

前提

无

1.4.3 接口镜像的缺省配置

设备上接口镜像的缺省配置如下。

功能	缺省值
接口镜像功能状态	禁止
镜像源接口	无

1.4.4 配置接口镜像功能

请在设备上进行以下配置。

步骤	配置	说明
1	`JX#configure`	进入全局配置模式。

步骤	配置	说明
2	JX(config)#**mirror group** group-number *interface-type interface-number* { **rspan** *vlan-id* **tpid** { **standard** | protocol-id } }	group-number：指定镜像组号 取值范围是 1~4 interface-type：接口类型 interface-number：接口号为 unit/slot/port 形式，取值范围由接口类型决定 （可选）vlan-id：指定远程镜像 VLAN ID 取值范围是 1~4094 （可选）standard ：指定为标准值 0x8100 （可选）protocol-id：当前接口的外层 Tag 的标签协议标识 取值范围是 <0x0-0xffff>。
3	JX(config)#**interface** *interface-type interface-number*	进入物理接口配置模式
4	JX(config-ge-1/0/1)#**mirror** { **inbound** | **outbound** | **both** } **group** *group-number*	配置接口镜像规则 { **inbound** | **outbound** | **both** }：指定镜像方向，入方向、出方向、入/出同时 group-number：指定镜像组号 取值范围是 1~4。

1.4.5 检查配置

配置完成后，请在设备上执行以下命令检查配置结果。

序号	检查项	说明
1	JX#**show mirror group**	显示镜像组配置信息。
2	JX#**show mirror interface**	显示接口上的镜像配置信息

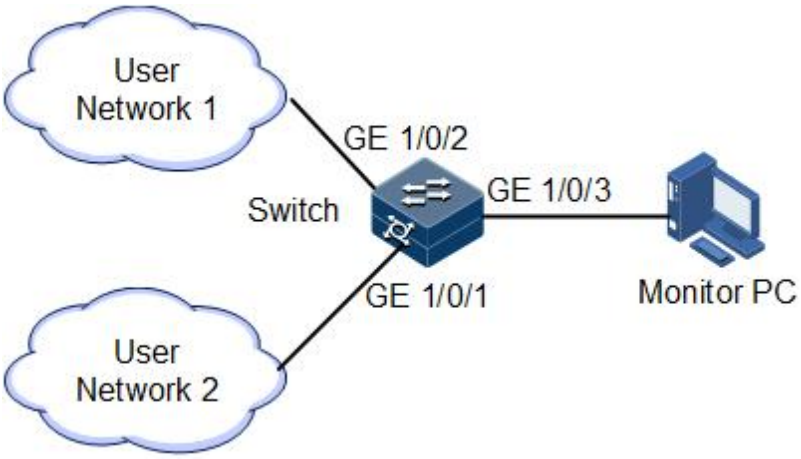
1.4.6 配置接口镜像应用示例

组网需求

如下图所示，网络管理员希望通过数据监测设备仅对用户网络 1 的报文进行监控，从而抓取出现故障或异常时的数据流，来分析、发现问题根源并及时解决。

交换机禁止所有自发包功能和风暴抑制功能。用户网络 1 通过 GE 1/0/2 接入交换机；用户网络 2 通过 GE 1/0/1 接入交换机；数据监测设备连接在交换机的 GE 1/0/3 上。

图 1-11 接口镜像应用组网示意图



配置步骤

步骤 1　在 Switch 上使能接口镜像功能。

```
JX#config
JX(config)#mirror group 1 ge 1/0/3
JX(config)#interface ge 1/0/2
JX(config-ge1/0/2)#mirror ingress group 1
```

检查结果

通过 **show mirror** 查看接口镜像信息配置是否正确。

```
JX# show mirror group
Mirror
-------------------------------------------------------------
---------
  1          ge-1/0/3          n/a
-------------------------------------------------------------
---------
```

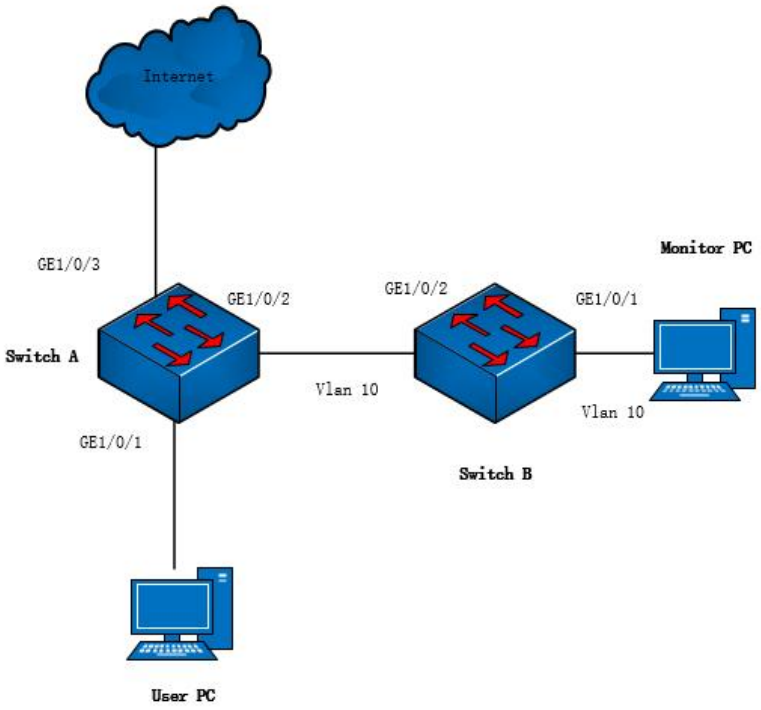
1.4.7 配置远端接口镜像应用示例

组网需求

如下图所示，网络管理员希望通过远端数据监测设备仅对用户 PC 的报文进行监控，从而抓取出现故障或异常时的数据流，来分析、发现问题根源并及时解决。

交换机禁止所有自发包功能和风暴抑制功能。用户通过 GE 1/0/1 接入 switch A；数据监测设备连接在 switch B 的 GE 1/0/1 上。

图 1-12 接口远端镜像应用组网示意图



配置步骤

步骤 1　在 Switch A 上使能接口镜像功能。

```
JX#config
JX(config)#mirror group 1 ge 1/0/2 rspan 10 tpid 0x8100
JX(config)#interface ge1/0/1
JX(config-ge1/0/1)#mirror inbound group 1
```

步骤 2　在 Switch B 上配置 VLAN 转发。

```
JX#config
JX(config)#interface ge 1/0/1
JX(config-ge1/0/1)#port trunk allow-pass vlan 10
JX(config)#interface ge 1/0/2
JX(config-ge1/0/2)#port trunk allow-pass vlan 10
```

1.5 电缆诊断

1.5.1 简介

设备支持电缆诊断功能，可以对线路进行检测。

电缆诊断可查询的结果如下：

- 发送线缆的检测结果；
- 发送线缆的错误位置；
- 接收线缆的检测结果；

- 接收线缆的错误位置。

1.5.2 配置准备

场景

使能设备的电缆诊断功能，可及时了解设备电缆线路的运行状态，及早定位并排除设备电缆故障。

前提

无

1.5.3 配置电缆诊断功能

请在需要配置电缆诊断的设备上进行以下配置。

步骤	配置	说明
1	JX(config)#**virtual-cable-test force-detect**	全局使能电缆诊断，对设备上所有支持电缆诊断的端口进行一次电缆诊断。
2	JX(config-ge-1/0/5)#**virtual-cable-test**	使能接口的电缆诊断功能。

说明

使能接口电缆诊断不重启接口的功能时，如果接口状态为 Up，接口会重启一次，获取电缆诊断数据。配置此功能后，执行电缆诊断时，如果接口状态为 Up，接口不会再次重启，直接读取缓存中上次电缆诊断的数据；如果接口状态为 Down，接口执行一次实际电缆诊断获取实际故障点的长度；新插入的接口会自动执行电缆诊断并将结果存入缓存。

1.5.4 检查配置

配置完成后，请在设备上执行以下命令检查配置结果。

序号	检查项	说明
1	JX(config)#**show virtual-cable-test config**	查看电缆诊断相关配置信息。
2	JX(config)#**show virtual-cable-test**	查看所有接口的电缆诊断的结果。
3	JX(config)#**show virtual-cable-test interface** *interface-type primary-interface-number*	查看指定接口的电缆诊断的结果。
4	JX(config)#**show virtual-cable-test link-down**	查看所有是 down 状态接口的电缆诊断的结果。

1.6 UDLD

1.6.1 简介

UDLD（UniDirectional Link Detection，单向链路检测）用于监听利用光纤或以太网线连接的物理配置，当出现单向链路（只能向一个方向传输）时，UDLD 可以检测出这一状况，关闭相应接口并发送警告信息。单向链路可能引起很多问题，尤其是生成树，可能会造成回环。

1.6.2 配置准备

场景

当出现单向链路（只能向一个方向传输）时，UDLD 可以检测出这一状况，关闭相应接口并发送警告信息。

原理

UDLD 协议通过与对方交互协议报文（DLDPDU）来识别对端设备、检测单向链路。UDLD 协议有七种状态：Init（初始化）、Linkdown（未连通）、Linkup（活动）、Advertisement（通告）、Detect（探测）和 Disable（单通）状态。

前提

UDLD 需要链路两端设备都支持才能正常运行。

1.6.3 UDLD 功能的缺省配置

设备上故障转移功能的缺省配置如下。

功能	缺省值
UDLD 功能	禁用

1.6.4 配置 UDLD

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**config**	进入全局模式。
2	JX(config)#**udld error-down recovery { disable | enable }**	配置单向链路恢复使能或去使能。
3	JX(config)#**udld error-down recovery-interval** *interval*	配置单向链路恢复时间间隔，默认值为 45 秒。
4	JX(config)#**udld work-mode { aggressive | normal }**	配置 UDLD 协议工作模式，分为普通模式和激进模式。

步骤	配置	说明
5	JX(config)#**udld advertise-interval** *interval*	配置 UDLD 协议发送 advertise 报文事件间隔，默认事件间隔为 7 秒。
6	JX(config)#**udld global up-delay** *time*	配置全局的端口单通恢复时端口延迟 up 的时间。
7	JX(config)#**udld snmp-trap { disable | enable }**	使能或去使能 UDLD 协议发送 TRAP 功能。
8	JX(config)#**udld uni-shutdown { auto | manual }**	配置当 UDLD 协议检测到端口单通时是自动还是自动将端口关闭。
9	JX(config)#**interface** *interface-type primary-interface-number* JX(config-ge-1/0/*)#**udld** **{ disable | enable }**	端口下使能或去使能 UDLD 功能。
10	JX(config-ge-1/0/*)#**udld aggressive { disable | enable }**	使能或去使能端口的 UDLD 协议工作模式为激进模式。
12	JX(config-ge-1/0/*)#**udld rx-mode { normal | rxloss }**	配置端口是否关注光模块 RXLOS 等相关消息进而进行接口 error-down 和 up 处理。
13	JX(config-ge-1/0/*)#**udld up-delay** *time*	配置端口的单通恢复时端口延迟 up 的时间。

1.6.5 检查配置

配置完成后，请在设备上执行以下命令检查配置结果。

序号	检查项	说明
1	JX#**show udld config**	查看 UDLD 配置信息。
2	JX#**show udld interface**	查看所有使能了 UDLD 协议的端口的协议状态
3	JX#**show udld local**	查看 UDLD 协议本地信息。
4	JX#**show udld peer**	查看 UDLD 协议邻居信息。

1.7 光模块数字诊断

1.7.1 简介

设备上光模块数字诊断支持对 SFP（Small Form-factor Pluggables，小型封装可插拔）光模块的诊断。

光模块数字诊断功能为系统提供一种性能监测手段，网络管理员通过分析该模块提供的监测数据，可以预测收发模块的寿命、隔离系统故障并在现场安装中验证模块的兼容性。

光模块数字诊断功能监控光模块的性能参数包括：

- 模块温度
- 内部供电电压
- 发送偏置电流
- 发送光功率
- 接收光功率

当性能参数达到了告警阈值或状态信息发生变化，会产生相应的 Trap 告警信息。

1.7.2 配置准备

场景

光模块故障诊断功能为用户提供一种对 SFP 光模块性能参数的检测手段，用户通过分析光模块的监测数据，可以预测其寿命、隔离系统故障并在现场安装中验证光模块的兼容性。

前提

设备所使用光模块要求为甲信认证光模块，如使用其他厂家光模块，可能导致业务不稳定、不支持诊断或诊断信息不准确等问题。

1.7.3 光模块数字诊断的缺省配置

设备上光模块数字诊断的缺省配置如下。

功能	缺省值
接口光模块数字诊断告警发送 Trap 功能	禁止

1.7.4 配置使能光模块数字诊断

请在设备上进行以下配置。

步骤	配置	说明
1	`JX#config`	进入全局配置模式。
2	`JX(config)#transceiver monitor interval` *`interval`*	配置光模块数字诊断轮询间隔时间。

1.7.5 配置光模块数字诊断告警发送 Trap

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**config**	进入全局配置模式。
2	JX(config)#**interface** *interface-type interface-number*	进入物理层接口配置模式。
3	JX(config 10ge 1/0/*)#**transceiver snmp-trap { enable | disable }**	使能接口光模块数字诊断告警发送 Trap。
4	JX(config 10ge 1/0/*)#**transceive rx-power low-threshold** *low-threshold-value* **high-threshold** *high-threshold-value*	设置光模块收光功率的高低门限值，在光模块收光功率低于低门限值或者高于高门限值时，光模块可以发出相应告警
5	JX(config 10ge 1/0/*)#**transceive tx-power low-threshold** *low-threshold-value* **high-threshold** *high-threshold-value*	设置光模块发光功率的高低门限值，在光模块发光功率低于低门限值或者高于高门限值时，光模块可以发出相应告警
6	JX(config 10ge 1/0/*)#**transceive temperature low-threshold** *low-threshold-value* **high-threshold** *high-threshold-value*	设置光模块温度的高低门限值，在光模块温度低于低门限值或者高于高门限值时，光模块可以发出相应告警
7	JX(config 10ge 1/0/*)#**transceive voltage low-threshold** *low-threshold-value* **high-threshold** *high-threshold-value*	设置光模块电压的高低门限值，在光模块电压低于低门限值或者高于高门限值时，光模块可以发出相应告警
8	JX(config 10ge 1/0/*)#**transceive bias-current low-threshold** *low-threshold-value* **high-threshold** *high-threshold-value*	设置光模块偏执电流的高低门限值，在光模块偏执电流低于低门限值或者高于高门限值时，光模块可以发出相应告警
9	JX(config 10ge 1/0/*)#**transceive tec-current low-threshold** *low-threshold-value* **high-threshold** *high-threshold-value*	设置光模块制冷电流的高低门限值，在光模块制冷电流低于低门限值或者高于高门限值时，光模块可以发出相应告警

1.7.6 检查配置

配置完成后，请在设备上进行以下命令检查配置结果。

序号	检查项	说明
1	JX#**show transceiver config**	查看当前配置的光模块监控信息的相关配置。
2	JX#**show transceiver interface**	查看光模块的大概数字诊断信息，展示光模块的收发光功率、温度、电压、偏执电流等实时信息。

序号	检查项	说明
3	JX#**show transceiver** *interface-type primary-interface-number*	查看指定端口的光模块的相关监控信息。
4	JX#**show transceiver** *interface-type primary-interface-number* **threshold**	查看指定端口上配置的光模块相关监控信息的门限值。

1.8 系统日志

1.8.1 简介

系统日志功能是指设备将系统信息和调试信息等内容以日志的形式记录并输出到指定的目的地，在设备发生故障时，方便用户查看和定位故障。

设备的系统消息和一些调试输出会被送至系统日志处理。系统日志根据用户的配置将信息送往不同的目的地，接收系统日志的目的地有以下几类。

- Console 控制台：将日志信息通过 Console 接口输出到本地控制台。
- Monitor 监控台：将日志信息输出到监控台，如 Telnet 终端。
- Logfile 日志文件：将日志信息以日志文件形式输出到设备的 Flash 中。
- Buffer 缓冲区：将日志信息输出到 log 缓冲区中。
- Trapbuffer 缓冲区：将日志信息输出到 trap 缓冲区中。
- SNMP 服务器：将日志信息转化为 Trap 输出到 SNMP 服务器。
- Syslog 服务器：将日志信息输出到 syslog 服务器。
- Smtp 电子邮件：将日志信息输出到 smtp 电子邮件。

系统日志信息级别，根据严重程度分为 8 个等级，如表 10-4 所示。

表 1-4 信息级别

严重等级	级别	说明
emergencies	0	系统不可以使用
alerts	1	需要立即处理
critical	2	严重状态
error	3	错误状态
warning	4	警告状态
notification	5	正常但是很重要的状态
information	6	通告事件
debugging	7	调试信息

说明

输出信息的严重等级是可手动设置的。根据配置的严重等级输出信息时，仅输出级别小于或等于所配置的严重等级的信息。比如，配置输出级别指定为 3（也可直接指定严重等级 error）的信息输出，则级别为 0～3 的信息，即严重等级为 emergencies～error 的信息均可以输出。

1.8.2 配置准备

场景

设备会将系统的登录成功失败、关键信息、调试信息、错误信息等生成系统日志，输出为日志文件或传送到日志主机、Console 接口或监控台，以便用户查看并定位故障。

前提

无

1.8.3 系统日志的缺省配置

设备上系统日志的缺省配置如下。

功能	缺省值
系统日志功能状态	使能
日志消息输出到 console 控制台功能	使能，缺省级别为 warning（4）
日志信息输出到 file 文件功能	使能，缺省级别为 debugging(7)
日志输出到 monitor 监控台功能	使能，缺省级别为 warning（4）
日志输出到 buffer 缓冲区功能	使能，缺省级别为 debugging（7）
日志输出到 SNMP 服务器功能	禁止，缺省级别为 debugging（7）
日志输出到 Trapbuffer 缓冲区功能	禁止，缺省级别为 information（6）
日志输出到 syslog 服务器功能	禁止，缺省级别为 information（6）
日志输出到 smtp 电子邮件功能	禁止，缺省级别为 warning（4）

1.8.4 配置系统日志基本信息

请在设备上进行以下配置。

步骤	配置	说明
1	`JX#configure`	进入全局配置模式。

步骤	配置	说明
2	JX(config)#**logging { start | stop }**	配置使能系统日志功能。
3	JX(config)#**logging file max-number** { *max-num* | **default** }	配置系统日志的日志文件最大个数。
4	JX(config)#**logging file size kbytes** { *size-value* | **default** }	配置系统日志的日志文件大小。

1.8.5 配置系统日志输出

步骤	配置	说明
1	JX#**configure**	进入全局配置模式。
2	JX(config)#**logging module** *module-name* **action { console | monitor | logfile | buffer | trap | trapbuffer | syslog | smtp } { log | debug | trap } [state { enable | disable | default } | level { emergencies | alert | critical | error | warning | notification | information | debugging | default }]**	配置模块日志输出方向，输出使能以及输出级别

1.8.6 配置系统日志输出 TELNET/SSH 终端

步骤	配置	说明
1	JX#**terminal monitor** JX#**no terminal monitor**	打开关闭信息输出到 TELNET/SSH 终端。
2	JX#**terminal log** JX#**no terminal log**	打开关闭日志信息输出到 TELNET/SSH 终端。缺省为开启状态。
3	JX#**terminal trap** JX#**no terminal trap**	打开关闭 Trap 信息输出到 TELNET/SSH 终端。缺省为开启状态。
4	JX#**terminal debug** JX#**no terminal debug**	打开关闭 Trap 信息输出到 TELNET/SSH 终端。缺省为关闭状态。

1.8.7 检查配置

在设备上执行以下命令可以查看日志相关信息与配置。

序号	检查项	说明
1	JX#**show logging information**	查看日志配置的全局信息。
2	JX#**show logging { buffer | trap }**	查看日志或者 trap 缓冲区信息。
3	JX#**show logging { buffer | trap } { include | exclude | begin } string** *string*	查看日志或者 trap 缓冲区信息，指定字符串过滤条件。

序号	检查项	说明
4	JX#**show logging { buffer | trap } size** *size-number*	查看日志或者 trap 缓冲区信息，指定条目数。
5	JX#**show logging { buffer | trap } module** *module-name*	查看指定模块的日志或者 trap 缓冲区信息。
6	JX#**show logging { buffer | trap } size** *size-number* **{ include | exclude | begin } string** *string*	查看日志或者 trap 缓冲区信息，指定条目数和字符串过滤条件。
7	JX#**show logging { buffer | trap } start-time** *start-time* **[end-time** *end-time* **]**	查看日志或者 trap 缓冲区信息，指定日志产生的时间范围。
8	JX#**show logging action**	查看日志所有模块的动作信息。
9	JX#**show logging action { console | monitor|logfile | buffer | trap | trapbuffer | syslog | smtp }**	查看日志所有模块指定动作信息。
10	JX#**show logging error**	查看所有模块日志的错误码信息。
11	JX#**show logging event**	查看所有模块日志的事件信息。
12	JX#**show logging file** *file-name*	查看指定日志文件的内容
13	JX#**show logging file** *file-name* **{ include | exclude | begin } string** *string*	查看指定日志文件的指定内容，指定字符串过滤条件。
14	JX#**show logging module**	查看所有模块日志的信息，包括模块名、模块 ID、日志文件名、事件数目和错误码条目数。
15	JX#**show logging statistics**	查看日志模块的统计信息。

1.8.8 维护

用户可以通过以下命令，维护系统日志特性的运行情况和配置情况。

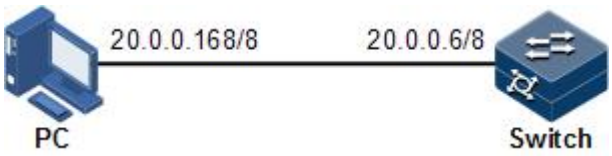
命令	描述
JX(config)#**clear logging** { **buffer** | **trap** }	清除 log 或者 trap 缓冲区中的日志信息。
JX(config)#**clear logging statistics**	清除日志的统计信息。
JX(config)#**clear logging file all**	清除所有日志文件。
JX(config)#**clear logging module** *module-name*	清除指定模块日志的统计信息。

1.8.9 配置系统日志输出到当前终端示例

组网需求

如图 10-13 所示，配置系统日志功能，将设备上的日志信息输出到日志主机，以便用户随时查看。

图 1-13 系统日志输出到日志主机组网示意图



配置步骤

步骤 1 配置设备的 IP 地址。

```
JX#config
JX(config)#interface vlan 1
JX(config-vlanif-1)#ip address 20.0.0.6 255.0.0.0
JX(config-vlanif-1)#quit
```

步骤 2 配置系统日志输出到终端上。

```
JX(config)#terminal monitor
```

步骤 3 配置调试信息输出到终端上。

```
JX(config)#terminal debug
```

检查结果

通过 **show logging information** 命令查看系统日志全局配置信息是否正确。

```
JX#show logging information
------------------------------------------------------------
  Logging               : on
  Module number          : 125
  Logfile path           : "/ram/log"
  Logfile max size       : 3072 Kb
  Logfile max number     : 3
  Logbuffer Max number    : 2000
  Logbuffer Current number : 201
  Logbuffer history number : 216
------------------------------------------------------------
```

1.9 告警管理

1.9.1 简介

告警是指当设备出现故障或某些工作状态发生变化时，系统能够根据不同故障类型、不同告警来源产生的信息。

告警信息用于报告一些紧急且重要的事件，及时通知网络管理人员，为监控设备运行和进行故障诊断提供有力支持。

告警信息保存在设备的告警缓存区中，同时将生成日志信息。若配置了网管系统，则该告警信息会通过 SNMP 向网管系统发送，发送给网管系统的信息称为 Trap 信息。

1.9.2 配置准备

场景

设备发生故障时，由告警管理模块来收集设备故障信息，以日志等形式输出告警的发生时间、告警的名称和描述信息等，帮助用户快速进行问题定位。

如果设备上配置网管系统，告警信息可以直接上报网管系统，给出告警产生的可能原因和处理建议，帮助用户及时处理故障。

如果设备上配置了硬件监控功能。当设备运行环境出现异常时，会记录硬件监控告警表、产生 Syslog 系统日志或发送 Trap 等告警信息，通知用户进行相应处理，防止故障发生。

告警管理方便用户直接在设备上进行告警抑制，告警自动上报，告警监控，告警反转，告警延迟，告警存储模式，清除告警，查看告警等配置。

前提

如果设备上需要配置硬件监控功能时：

- 以 Syslog 方式输出时，告警信息会生成系统日志。当需要把告警信息发送到系统日志主机时，设备上需要配置系统日志主机的 IP 地址等信息。
- 需要把告警信息以 Trap 方式发送到网管中心时，设备上需要配置网管中心的 IP 地址等信息。

1.9.3 配置告警基本功能

无。

1.9.4 检查配置

配置完成后，请在设备上执行以下命令检查配置结果。

序号	检查项	说明
1	JX#show alarm information	查看告警管理模块的表项数量信息。
2	JX#show alarm description	查看告警管理模块的所有告警信息。 包括告警 id、告警名称、告警级别、对应 Trap-OID 和描述信息。
3	JX#show alarm { active | history | all } time-order [object *object-name*]	查看指定告警信息，并且按时序排列（活动告警、历史告警、全部告警）。
4	JX#show alarm { active | history | all } [object *object-name*]	查看指定告警信息（活动告警、历史告警、全部告警）。
6	JX#show alarm object	查看告警对象信息。
7	JX#show event description	查看告警管理模块的所有事件信息。 包括事件 id、事件名称、事件级别、对应 Trap-OID 和描述信息。
8	JX#show event all [object *object-name*]	查看告警管理模块的所有当前事件。

1.10 CPU 监控

1.10.1 简介

设备支持 CPU 监控功能，可以实时监控系统中各任务的状态、CPU 利用率和堆栈使用情况，帮助网管人员快速定位故障。

CPU 监控可以提供以下功能：

- 查看 CPU 利用率

提供查看各周期（5 秒，1 分钟，10 分钟，2 小时）内各任务的 CPU 占用时间和利用率。可以静态显示，也可以动态显示各周期内 CPU 总的利用率。

提供查看所有任务的运行状态信息和指定任务的详细运行状态信息。

提供查看各周期内 CPU 历史利用率。

提供查看死亡任务信息。

- CPU 利用率门限告警

在指定的采样周期内，系统的 CPU 利用率从低于下限阈值上升到高于上限阈值或者从高于上限阈值下降到低于下限阈值时，会产生告警并发送 Trap，Trap 信息会提供最近某个周期（5 秒，1 分钟，10 分钟）内 CPU 利用率最高的 5 个任务序号及其 CPU 利用率。

1.10.2 配置准备

场景

CPU 监控功能可以实时监控系统中各任务的状态、CPU 利用率和堆栈使用情况，提供 CPU 利用率门限值告警，方便及时发现并消除隐患，或帮助网管人员进行故障定位。

前提

在配置 CPU 监控之前，需完成以下任务：

- 当需要把 CPU 监控告警信息以 Trap 方式输出时，需在设备上配置 Trap 输出目标主机地址，即网管中心的 IP 地址等信息。

1.10.3 CPU 监控的缺省配置

设备上 CPU 监控的缺省配置如下。

功能	缺省值
CPU 利用率告警 Trap 输出	使能
CPU 利用率告警的上限阈值	80%
CPU 利用率采样周期	10s

1.10.4 配置 CPU 监控告警

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**config**	进入全局配置模式。
2	JX(config)#**cpu** { **all** | *cpu-index* } **high-threshold** *high-threshold-value*	设置所有 CPU 表项或指定 CPU 表项的告警的上限阈值。
3	JX(config)#**cpu** { **all** | *cpu-index* } **description** *description-string*	设置所有 CPU 表项或指定 CPU 表项的描述信息。
4	JX(config)#**cpu** { **all** | *cpu-index* } **snmp-trap** { **disable** | **enable** }	设置所有 CPU 表项或指定 CPU 表项的发送 TRAP 使能或去使能。
5	JX(config)#**cpu monitor interval** *interval-value*	配置 CPU 利用率采样时间间隔。
6	JX(config)#**cpu monitor** { **disable** | **enable** }	使能或去使能 CPU 利用率监控。

1.10.5 检查配置

配置完成后，请在设备上执行以下命令检查配置结果。

序号	检查项	说明
1	JX#**show cpu**	查看 CPU 利用率。
2	JX#**show cpu verbose**	查看 CPU 利用率的相关详细信息。
3	JX#**show cpu task**	查看各任务的 CPU 利用率。

1.11 内存监控

1.11.1 配置准备

场景

内存利用率监控功能可以实时监控系统的内存利用率，提供内存利用率阈值告警，方便及时发现并消除隐患，或帮助网管人员进行故障定位。

前提

当用户需要把内存利用率监控告警信息以 Trap 方式输出时，应首先在设备上配置 Trap 输出目标主机地址，即网管中心的 IP 地址等信息。

1.11.2 配置内存监控

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**config**	进入全局配置模式。
2	JX(config)#**memory** { **all** | *memory-index* } **high-threshold** *high-threshold-value*	配置所有内存表项或者指定内存表项的监控告警门限阈值。
3	JX(config)#**memory** { **all** | *memory-index* } **description** *description-string*	配置所有内存表项或者指定内存表项的描述信息。
4	JX(config)#**memory** { **all** | *memory-index* } **snmp-trap** { **disable** | **enable** }	配置所有内存表项或者指定内存表项的 TRAP 使能或去使能。
5	JX(config)#**memory monitor interval** *interval-value*	配置内存利用率的监控周期。
6	JX(config)#**memory monitor** { **disable** | **enable** }	配置内存利用率监控使能或去使能。

1.11.3 检查配置

配置完成后，请在设备上执行以下命令检查配置结果。

序号	检查项	说明
1	JX#show memory	查看系统内存利用率。
2	JX#show memory verbose	查看系统内存利用率的详细信息。
4	JX#show memory task	查看系统内每个进程的内存占用信息。

1.12 Ping

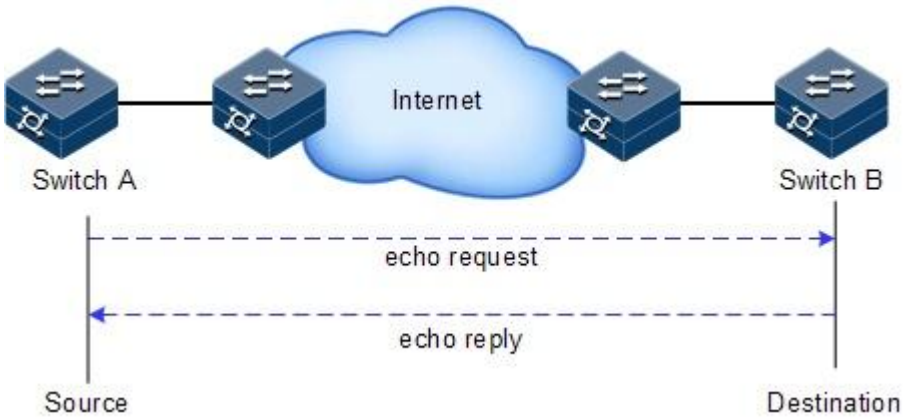
1.12.1 简介

Ping 的名字源于声纳定位操作，用于检测网络连接是否正常。

Ping 功能一般借助 ICMP echo 报文来实现。首先发送 echo request 报文到某个地址，然后等待该地址对应的设备响应 echo reply 报文，当 echo request 到达目标地址以后，在一个有效的时间内返回 echo reply 报文给源地址，则说明目的地可达。如在有效时间内没有收到回应，则在发送端显示超时，并表明目的地不可达。

Ping 功能实现原理如图 10-14 所示。

图 1-14 Ping 功能实现原理组网示意图



1.12.2 配置 Ping 功能

请在设备上进行以下操作。

步骤	配置	说明
1	JX#**ping** { *ipv4-address* | *hostname* } [**-n** *nvalue*|**-system-time**|**-t**|**-q**|**-f**|**-h** *hvalue*|**-l** *lengthvalue*|**-w** *waitvalue*|**-tos** *tosvalue*|**-m** *mvalue*|**-dscp** *dscpvalue*|**-8021p** *8021pvalue*|**-range** *rangevalue*|**-s** *source-ipv4-address*|**-nexthop** *next-ipv4-address*|**-vpn-instance** *vpn-name*|**-eth-trunk** *trunk-number*|**-bridge-domain** *bridge-domain-number*|**-loopback** *loopback-number*|**-vlan** *vlan-id*|**-ge** *interface-number1*|**-10ge** *interface-number2*|**-25ge** *interface-number3*|**-40ge** *interface-number4*|**-100ge** *interface-number5*|**-400ge** *interface-number6*|**-meth** *meth-number*]	通过 **ping** 命令测试 IPv4 网络的连通性。
2	JX#**ping-ipv6** {*ipv6-address* | *hostname* } [**-n** *nvalue* |**-system-time**|**-t**|**-q**|**-f**|**-h** *hvalue*|**-l** *lengthvalue*|**-w** *waitvalue*|**-m** *mvalue*|**-tc** *tcvalue*|**-8021p** *8021pvalue*|**-range** *rangevalue*|**-s** *source-ipv6-adress*|**-nexthop** *next-ipv6-address*|**-vpn-instance** *vpn-name*|**-eth-trunk** *trunk-number*|**-bridge-domain** *bridge-domain-number*|**-loopback** *loopback-number*|**-vlan** *vlan-id*|**-ge** *interface-number1*|**-10ge** *interface-number2*|**-25ge** *interface-number3*|**-40ge** *interface-number4*|**-100ge** *interface-number5*|**-400ge** *interface-number6*|**-meth** *meth-number*]	通过 **ping** 命令测试 IPv6 网络的连通性。

说明

在 **ping** 命令执行的过程中，无法对设备进行其他操作，只有等待执行过程结束或者通过“Ctrl＋C”键强制中断执行过程后才能进行其他操作。

1.13 Trace

1.13.1 简介

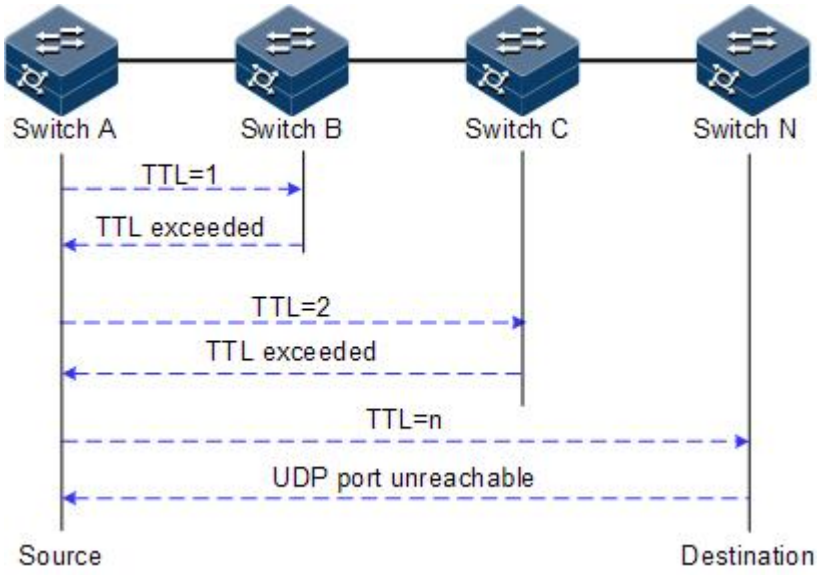
Trace 和 Ping 一样，是网络管理中常用的维护手段。Trace 功能常用于测试报文从发送端到目的端所经过的网络节点，检测网络连接是否可达，并分析网络中的故障点。

Trac 的执行过程如下：

- 首先发送一份 TTL 为 1 的嗅探报文（其中报文的 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）。
- 到达第 1 跳时 TTL 减 1，由于 TTL 的值为 0，第 1 跳设备发回一个 ICMP 超时报文，指明此报文不能被发送。
- 发送主机将 TTL 加 1 后重新发送此报文。
- 到达第 2 跳时由于 TTL 的值被减为 0，第 2 跳设备发回一个 ICMP 超时报文，指明此报文不能被发送。

以上步骤循环进行，直到到达目的主机，目的主机并不会送回 ICMP 超时报文，由于目的主机的端口号没有被使用，目的主机会发送端口不可达报文，测试结束。这样，发送主机就能够记录每一个 ICMP TTL 超时报文的源地址，根据得到的回应报文分析出到达目的地所经历的路径。Trace 功能实现原理如图 10-15 所示。

图 1-15 Trace 功能实现原理组网示意图



1.13.2 配置 IPv4 Trace 功能

请在设备上进行以下操作。

步骤	配置	说明
1	JX#**trace** *ip-address* [**-n** *number*|**-r**|**-q**|**-f**|**-fh** *first-ttl-value*|**-h** *max-ttl-value*|**-l** *length-value*|**-w** *wait-for-each-value*|**-tos** *tosvalue*|**-m** *wait-for-send-value*|**-dscp** *dscpvalue*|**-8021p** *8021p-value*|**-range** *range-value*|**-s** *source-ip-address*|**-nexthop** *next-ip-address*|**-vpn-instance** *vpn-name*|**-eth-trunk** *trunk-number*|**-bridge-domain** *bridge-domain-id*|**-loopback** *loopback-number*|**-vlan** *vlan-id*|**-ge** *interface-number1*|**-10ge** *interface-number2*|**-25ge** *interface-number3*|**-40ge** *interface-number4*|**-100ge** *interface-number5*|**-400ge** *interface-number6*|**-meth** *meth-number*]	通过 **trace** 命令测试 IPv4 网络的连通性并查看报文经过的网络节点。

1.13.3 配置 IPv6 Trace 功能

请在设备上进行以下操作。

步骤	配置	说明
1	JX#**trace-ipv6** *ipv6-address* [**-n** *number*|**-r**|**-q**|**-f**|**-fh** *first-ttl-value*|**-h** *max-ttl-value*|**-l** *length-value*|**-w** *wait-for-each-value*|**-m** *wait-for-send-value*|**-tc** *traffic-class-value*|**-8021p** *8021p-value*|**-range** *pkt-length-value*|**-s** *source-ipv6-address*|**-nexthop** *next-ipv6-address*|**-vpn-instance** *vpn-name*|**-eth-trunk** *trunk-number*|**-bridge-domain** *bridge-domain-id*|**-loopback** *loopback-number*|**-vlan** *vlan-id*|**-ge** *interface-number1*|**-10ge** *interface-number2*|**-25ge** *interface-number3*|**-40ge** *interface-number4*|**-100ge** *interface-number5*|**-400ge** *interface-number6*|**-meth** *meth-number*]	通过 **trace** 命令测试 IPv6 网络的连通性并查看报文经过的网络节点。

1.14 硬件监控

说明

不是所有设备都支持温度告警。请以具体设备支持情况为准。

1.14.1 简介

硬件环境监控主要是对设备的运行环境进行监控，监控的对象主要包括设备的温度和电源。

1.14.2 温度监控

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**config**	进入全局配置模式。
2	JX(config)#**temperature monitor** { **disable** | **enable** }	使能或去使能温度监控。
3	JX(config)#**temperature** { **all** | *temperature-index* } **description** *description-string*	设置所有或指定温度监控表项的描述信息。
4	JX(config)#**temperature** { **all** | *temperature-index* } **low-threshold** *low-threshold-value* **high-threshold** *high-threshold-value*	设置所有或指定温度监控表项的高低门限值，当温度高于高门限值或低于低门限值时都会产生告警。
5	JX(config)#**temperature** { **all** | *temperature-index* } **snmp-trap** { **disable** | **enable** }	使能或去使能所有温度监控表项或指定温度监控表项发送 TRAP 信息的功能。。

1.14.3 电源监控

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**config**	进入全局配置模式。
2	JX(config)#**power monitor** { **disable** | **enable** }	使能或去使能电源信息监控。
3	JX(config)#**power monitor interval** *interval-value*	设置电源信息监控的周期，默认值是 25 秒。
4	JX(config)#**power** { **all** | *power-index* } **description** *description-string*	设置所有电源表项或指定电源表项的描述信息。

步骤	配置	说明
5	JX(config)#**power** { **all** | *power-index* } **snmp-trap** { **disable** | **enable** }	使能或去使能所有电源监控表项或指定电源监控表项发送 TRAP 信息的功能。

1.14.4 检查配置

配置完成后，请在设备上执行以下命令检查配置结果。

步骤	配置	说明
1	JX#**show temperature config**	显示对温度监控所作的相关配置信息。
2	JX#**show power config**	显示对电源监控所作的相关配置信息。

1.15 风扇监控

说明

本节仅适用于带风扇款型。

1.15.1 简介

设备支持风扇监控功能，可以对风扇的转速和温度进行监控，当设备监控到风扇的转速和温度出现异常时，会产生告警并发送 Trap 信息。

设备对风扇的监控模式有两种：

- 固定转速等级模式，即强制设定风扇的转速等级；
- 温控模式，即风扇根据温度的变化自动调节转速等级。

在温控模式下，当检测到风扇的温度超出高门限值时，就会将风扇的转速等级提升一档，如果检测到风扇的温度低于低门限值时，就会将风扇的转速等级下调一档。

1.15.2 配置准备

场景

当设备安放于比较炎热的环境时，过高的温度会影响设备的散热性能，此时需要配置风扇监控功能，使设备的风扇能够依据周围的环境温度自动调节，以维护设备的正常运转。

前提

无

1.15.3 配置风扇监控功能

请在需要配置风扇监控功能的设备上进行以下配置。

步骤	配置	说明
1	JX#**config**	进入全局配置模式。
2	JX(config)#**fan monitor { disable | enable }**	使能或去使能风扇监控。
3	JX(config)#**fan monitor interval** *interval-value*	配置风扇监控周期。
4	JX(config)#**fan { all |** *fan-index* **} mode { speed-level | temperature-control }**	配置所有风扇或指定风扇的模式，speed-level 为固定转速等级模式，temperature-control 为温控模式。
5	JX(config)#**fan { all |** *fan-index* **} description** *description-string*	配置所有风扇或指定风扇的描述信息。
6	JX(config)#**fan { all |** *fan-index* **} level** *level-value*	配置所有风扇或指定风扇的转速等级。
7	JX(config)#**fan { all |** *fan-index* **} snmp-trap { disable | enable }**	使能或去使能所有风扇或指定风扇的 TRAP 发送功能。

1.15.4 检查配置

配置完成后，请在设备上执行以下命令检查配置结果。

序号	检查项	说明
1	JX#**show fan**	查看风扇监控的相关信息，当前转速、转速等级等信息。
2	JX#**show fan verbose**	查看风扇的基本信息。
3	JX#**show fan config**	查看风扇监控配置的相关信息。

1.16 设备堆叠

1.16.1 简介

堆叠，是指将两台及以上支持堆叠特性的交换机设备组合在一起，从逻辑上形成一台交换设备。使用这种虚拟化技术可以集合多台设备的硬件资源能力和软件处理能力，实现多台设备的协同工作、统一管理和不间断维护。

堆叠基本概念

- 运行模式

设备支持两种运行模式：

普通模式：该模式下，设备不能与其他设备形成堆叠。

堆叠模式：该模式下，进行堆叠配置后可与其他堆叠设备形成堆叠。

当两种堆叠模式切换时，将会引起设备重启。

- 成员设备角色

堆叠中的每台设备都称为成员设备，每个成员设备根据协商的结果，可能处于不同的角色：

Master：主用设备，负责管理和控制整个堆叠系统。

Slave：处理业务、转发报文的同时作为主用设备的备份设备运行。当主用设备故障时，堆叠系统会自动从备设备中选举一个新的主设备接替原主用设备的工作。

- 成员设备编号

堆叠系统使用成员设备编号来标识和管理成员设备。在进行堆叠配置时，必须保证设备的成员设备编号是唯一的，如果两台设备的成员编号是一样的，则在协商时将会出现问题，导致堆叠系统建立出现问题。

- 成员优先级

成员优先级是成员设备的一个属性，默认情况下，设备的成员优先级是 1。在堆叠协商的过程中，成员优先级的大小将影响成员设备是否能被选举成员堆叠主用设备。如果想让某台设备称为堆叠系统的主用设备，可以将设备的成员优先级设置为一个较大的值。

- 堆叠端口

堆叠端口是一个逻辑端口，专门用于堆叠成员设备之间的连接。每台成员设备可以配置两个堆叠端口，堆叠端口需要和物理端口绑定之后才能生效。

- 堆叠域

堆叠域是一个逻辑概念，一个堆叠系统对应一个堆叠域。只有堆叠域 ID 一致的成员设备之间才能建立堆叠系统，不同堆叠域之间的成员设备即使能够收到对端设备的堆叠报文，也无法建立堆叠系统。

1.16.2 堆叠拓扑

本系列交换机可以支持 3 台设备组成堆叠，堆叠设备之间组成链型或环型拓扑的堆叠系统。

图 1-16 3 台成员设备组成链型拓扑

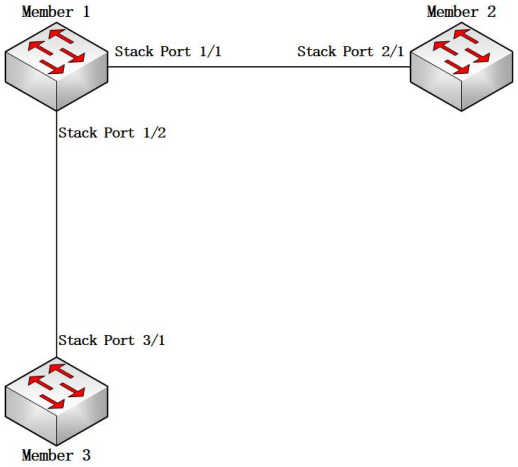
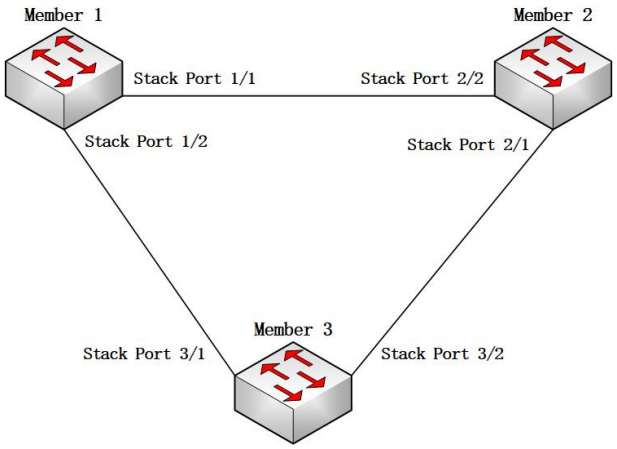


图 1-17 3 台成员设备组成环形型拓扑



注意

- 成员设备组成环型堆叠拓扑时，相邻两台堆叠成员设备之间的堆叠端口号是不一样的。如图 10-16 所示，堆叠成员设备 1 的堆叠端口 1/1 和堆叠成员设备 2 的堆叠端口的 2/2 相连。如果出现两台相连设备之间相同堆叠端口号一样，在主备切换时将会导致堆叠系统出现同步失败。
- 堆叠配置时，需要在普通模式下先规划好堆叠系统拓扑，然后按照拓扑设计配置好对应的堆叠端口，最后配置堆叠模式。否则设备没有保存堆叠端口的配置，设备重启后堆叠端口未创建，不能进行堆叠协商。

堆叠角色选举

当堆叠系统出现如下情况时都将进行角色选举：

- 堆叠系统建立。
- 堆叠系统分裂，即堆叠链路断开。

- 两个独立运行的堆叠系统合并。

角色选举时将按照如下优先级选举主备设备：

- 当前的主设备优先，如果是两个独立堆叠系统合并，此时将在两个对立堆叠系统的主用设备中选举出新的主用设备。
- 成员设备配置了强制为主的设备。
- 成员设备优先级大的设备。
- 系统运行时间长的设备，如果设备启动时间间隔超过 1 分钟，则运行时间长的设备优先。
- 成员设备的 mac 地址，mac 地址小的设备优先。

通过如上规则选举出来的最优设备即为主用设备，其他设备为从设备。

1.16.3 堆叠缺省值

堆叠功能相关缺省值如下：

功能	缺省值
堆叠功能	禁用
堆叠成员优先级	1
堆叠成员编号	1
堆叠成员所属域 ID	1
堆叠成员发生堆叠协议报文时间间隔	3（单位秒）

1.16.4 配置堆叠

请在设备上进行以下配置：

步骤	配置	说明
1	JX(config)#**interface stack-port** *number*	创建堆叠端口。
2	JX(config-stack-port-*number*)#**add interface** ge 1/0/1	将物理端口添加值堆叠端口。
3	JX(config)#**hvs hello-interval** *time*	配置堆叠协议发生报文的时间间隔，单位秒。
4	JX(config)#**hvs master**	配置堆叠成员在堆叠协商时强制为主用设备。
5	JX(config)#**hvs member-id** *id*	配置堆叠设备的成员编号。
6	JX(config)#**hvs priority** *value*	配置堆叠成员设备的优先级。
7	JX(config)#**hvs switch-master member** { **myself** | *id* }	配置堆叠系统进行主备切换。
8	JX(config)#**hvs mode** { **normal** | **stack** }	配置堆叠成员设备进行模式切换，模式切换后设备将自动重启。

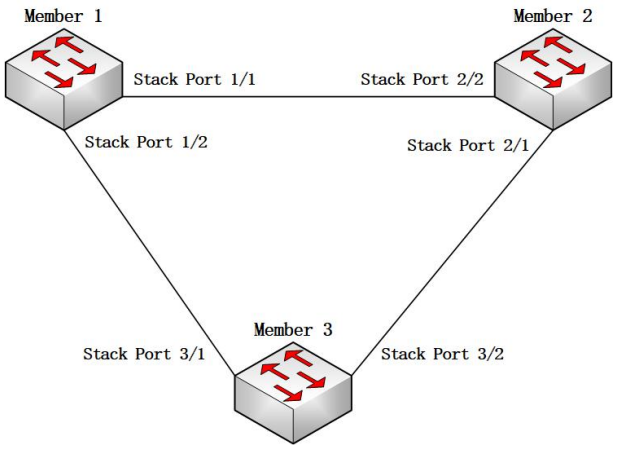
步骤	配置	说明
9	JX(config)#show hvs member	查看当前堆叠系统中成员以及主备角色
10	JX(config)#show hvs topo	查看整个堆叠系统中的拓扑信息
11	JX(config)#show hvs interface	查看本设备堆叠端口的信息
12	JX(config)#show hvs config	查看本设备堆叠相关配置

1.16.5 配置举例

如网需求

如下图的拓扑结构，如果 3 台设备需要组建链型拓扑结构的堆叠系统，需要进行如下配置。

图 1-18 3 台堆叠拓扑图



配置步骤

设备 1 的配置：

```
JX(config)#interface stack-port 1
JX(config-stack-port-1)#add interface ge 1/0/1
JX(config-stack-port-1)#add interface ge 1/0/2
JX(config-stack-port-1)#quit
JX(config)#interface stack-port 2
JX(config-stack-port-2)#add interface ge 1/0/3
JX(config-stack-port-2)#quit
JX(config)#hvs member-id 1
JX(config)#hvs mode stack
```

设备 2 的配置：

```
JX(config)#interface stack-port 1
JX(config-stack-port-1)#add interface ge 1/0/3
JX(config-stack-port-1)#quit
```

```
JX(config)#interface stack-port 2
JX(config-stack-port-2)#add interface ge 1/0/1
JX(config-stack-port-2)#add interface ge 1/0/2
JX(config-stack-port-2)#quit
JX(config)#hvs member-id 2
JX(config)#hvs mode stack
```

设备 3 的配置：

```
JX(config)#interface stack-port 1
JX(config-stack-port-1)#add interface ge 1/0/1
JX(config-stack-port-1)#quit
JX(config)#interface stack-port 2
JX(config-stack-port-2)#add interface ge 1/0/2
JX(config-stack-port-2)#quit
JX(config)#hvs member-id 3
JX(config)#hvs mode stack
```

待到 3 台设备重启完成之后，它们之间就会开始协商，然后可以通过命令查看堆叠是否建立完成。

此时在设备 3 上通过 **show hvs member** 的命令的结果如下：

```
JX(config)#show hvs member
SysId Role/ConfRole  State/Pri     Mac             Port0
Port0Mac         Port1   Port1Mac         Uptime
  1    master/no     done/1        f0f1:f2f3:0101    up
f0f1:f2f3:0101    up     f0f1:f2f3:0101    05h:37m:34s
  2    slave/no      done/1        f0f1:f2f3:0201    up
f0f1:f2f3:0201    up     f0f1:f2f3:0201    05h:37m:34s
 * 3    slave/no      done/1        f0f1:f2f3:0301    up
f0f1:f2f3:0301    up     f0f1:f2f3:0301    05h:37m:34s
```

通过 **show hvs topo** 的命令的结果如下：

```
JX(config)#show hvs topo
 Interface: stack-port-3/1,  State: up,  MAC: f0:f1:f2:f3:03:01
 Hop   SysId    Port0  State   MAC             Port1  State
MAC
 0    3        1     up     f0:f1:f2:f3:03:01  2     up
f0:f1:f2:f3:03:01
 1    1        1     up     f0:f1:f2:f3:01:01  2     up
f0:f1:f2:f3:01:01
 2    2        1     up     f0:f1:f2:f3:02:01  2     up
f0:f1:f2:f3:02:01
 Interface: stack-port-3/2,  State: up,  MAC: f0:f1:f2:f3:03:01
 Hop   SysId    Port0  State   MAC             Port1  State
MAC

 0    3        1     up     f0:f1:f2:f3:03:01  2     up
f0:f1:f2:f3:03:01
 1    2        1     up     f0:f1:f2:f3:02:01  2     up
f0:f1:f2:f3:02:01
```

```
 2    1       1     up     f0:f1:f2:f3:01:01  2     up
f0:f1:f2:f3:01:01
```

1.17 MAD

1.17.1 简介

当堆叠系统中成员设备之间链路端口，可能会导致堆叠系统分裂称为多个新的堆叠系统，这些新的堆叠系统有相同的 IP 地址等三层配置，会引起地址冲突，导致网络故障。MAD(Multi-Active Detection，多主冲突检测)协议用来进行堆叠分裂检测、冲突处理和故障恢复，以提高系统的可用性。

MAD 协议检测冲突主要有几种工作模式。

- 直连模式

直连模式下，需要在要进行多主冲突检测的堆叠成员设备之间分配一个额外的物理端口，用于收发 MAD 协议报文，可如图 10-18 建立拓扑。

- 代理模式

代理模式下，需要另外一个设备启用 MAD 协议，然后该设备与堆叠的各个成员设备之间建立跨设备聚合链路，然后在该聚合口上使能 MAD 协议，可如图 10-19 建立拓扑。

带外口检测模式

该模式下，只需要在带外口使能 MAD 协议，并且将所有堆叠成员设备的带外口连接到同一交换机，且保证所有带外口之间能正常通信即可。

图 1-19 MAD 直连模式

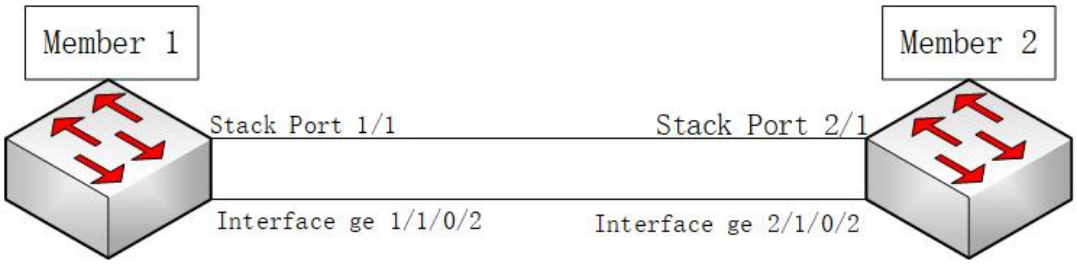
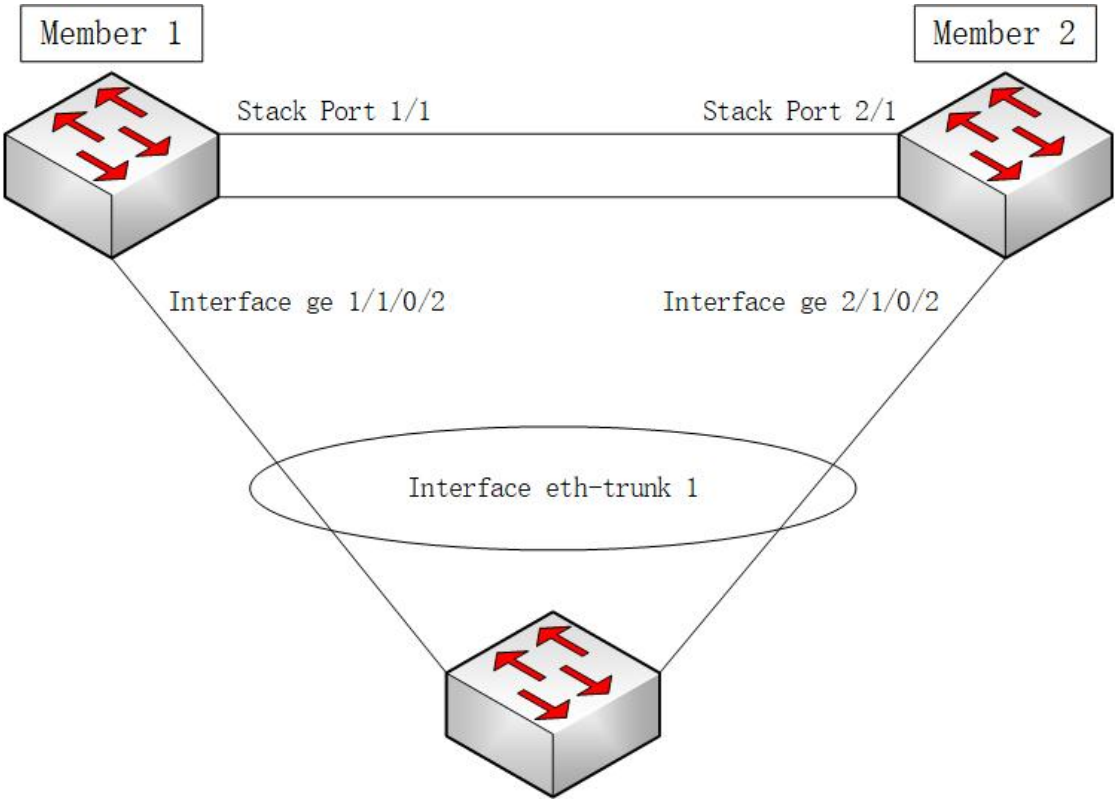


图 1-20 MAD 代理模式



1.17.2 配置准备

场景

- 备用管理 IP 地址

MAD 协议除了可以配置端口的工作模式外，可以为每个堆叠成员配置一个备份 IP 地址，该 IP 在 MAD 协议检测到多主时，自动将设备的带外口设置为该 IP 地址。

- 保留端口

MAD 协议在检测到多主时，进行比较之后，如果发现自己不是优选设备，则会将设备自己的端口进行 shutdown 操作。如果用户不想某些端口在 MAD 检测到多主时被 shutdown，可以进行保留端口的配置。

前提

无

1.17.3 配置 MAD

请在设备上进行如下配置：

步骤	配置	说明
1	JX(config)#**multi-active-detect backup ip address** *ip-address* **member** { **all** | *member-id*}	为指定的堆叠成员设备设置备份管理 IPv4 地址。
2	JX(config)#**multi-active-detect backup ipv6 address** *ipv6-address* **member** { **all** | *member-id*}	为指定的堆叠成员设备设置备份管理 IPv6 地址。
3	JX(config)#**interface GE 1/0/0/*** JX(config-ge-1/1/0/*)#**multi-active-detect mode direct**	配置该端口下 MAD 协议的工作模式。
4	JX(config)#**interface eth-trunk 1** JX(config-eth-trunk-1)#**multi-active-detect mode relay**	配置该聚合组端口下 MAD 协议的工作模式。
5	JX(config)#**interface meth 0/0/0** JX(config-meth-0/0/0)#**multi-active-detect** { **disable** | **enable** } JX(config)#**interface GE 1/0/0/*** JX(config-ge-1/1/0/*)#**multi-active-detect** { **disable** | **enable** }	在端口下使能或去使能 MAD 协议。
6	JX(config-meth-0/0/0)#**multi-active-exclude** { **disable** | **enable** }	在端口下使能 MAD 保留端口功能。
7	JX(config)#**show multi-active-detect config**	查看当前设备上的 MAD 相关配置。
8	JX(config)#**show multi-active-detect information**	查看你当前设备上 MAD 协议相关状态。

1.17.4 配置举例

组网需求

如图 10-19，首先需要进行堆叠向配置，让两台设备建立堆叠系统，然后在配置 MAD。

配置步骤

步骤 1 设备 1 的配置：

```
JX(config)#interface stack-port 1
JX(config-stack-port-1)#add interface ge 1/0/1
JX(config-stack-port-1)#quit
```

```
JX(config)#hvs member-id 1
JX(config)#hvs mode stack
```

步骤 2 设备 2 的配置：

```
JX(config)#interface stack-port 1
JX(config-stack-port-1)#add interface ge 1/0/1
JX(config-stack-port-1)#quit
JX(config)#hvs member-id 2
JX(config)#hvs mode stack
```

步骤 3 待到两台设备建立堆叠系统后，开始进行 MAD 协议相关配置。

```
JX(config)#interface GE 1/0/0/2
JX(config-ge-1/1/0/2))#stp disable
JX(config-ge-1/1/0/2))#multi-active-detect mode direct
JX(config-ge-1/1/0/2)#quit
JX(config)#interface ge 2/1/0/2
JX(config-ge-1/1/0/2))#stp disable
JX(config-ge-2/1/0/2))#multi-active-detect mode direct
```

检查配置

此时将堆叠端口端口，然后通过命令 **show multi-active-detect information** 查看 MAD 协议状态。

设备 1 状态：

```
JX(config)#show multi-active-detect information
  Current status : normal
  Direct detect information:
    ge-1/1/0/2      up
  Excluded ports(configurable):
    ge-1/1/0/5
  Excluded ports(can not be configured):
    ge-1/1/0/1
```

设备 2 状态：

```
JX(config)#show multi-active-detect information
  Current status : conflict
  Direct detect information:
    ge-2/1/0/2      up
  Excluded ports(configurable):
    ge-2/1/0/5
  Excluded ports(can not be configured):
    ge-2/1/0/1
```

1.18 NQA

1.18.1 简介

网络质量分析 NQA（Network Quality Analysis）是一种实时的网络性能探测和统计技术，可以对响应时间、网络抖动、丢包率等网络信息进行统计。NQA 能够实时监视网络 QoS，在网络发生故障时进行有效的故障诊断和定位。

1.18.2 配置准备

场景

为了使网络服务质量可见，使用户能够自行检查网络服务质量是否达到要求。需要在网络中部署探针设备对网络服务质量进行监控。

当设备提供 NQA 时，就不用部署专门的探针设备，可以有效的节约成本。NQA 可以实现对网络运行状况的准确测试，输出统计信息。

NQA 监测网络上运行的多种协议的性能，使用户能够实时采集到各种网络运行指标，例如：HTTP 的总时延、TCP 连接时延、DNS 解析时延、文件传输速率、FTP 连接时延、DNS 解析错误率等。

前提

无

1.18.3 NQA 的缺省配置

设备上 NQA 功能的缺省配置如下。

功能	缺省值
NQA 功能	禁用

1.18.4 配置 ICMP-echo 测试

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**config**	进入全局配置模式。
2	JX(config)#**nqa test-instance** *admin-name operate-tag* **type icmp-echo**	配置测试实例，类型为 icmp-echo
3	JX(nqa-admin-test-icmp-echo)#**destination ip** *ip-address*	配置目的 ip 地址

4	JX(nqa-admin-test-icmp-echo)#**frequency** *freq-ms*	配置测试频率，单位：毫秒
5	JX(nqa-admin-test-icmp-echo)#**probe count** *count*	配置每次测试探测的次数
6	JX(nqa-admin-test-icmp-echo)#**probe timeout** *timeout-time-ms*	配置每次探测等待的时间，单位：毫秒
7	JX(config)#**nqa schedule** *admin-name operate-tag* **start-time now life-time forever**	配置检测调度策略为立即开始，一直循环 注：该配置与绑定 timerange 策略二选一即可
8	JX(config)#**nqa schedule** *admin-name operate-tag* **time-range** *timer-ange-list*	配置检测调度策略绑定 timerange 注：该策略与立即开始策略二选一即可

1.18.5 配置 UDP-echo 测试

请在服务端设备上进行以下配置。

步骤	配置	说明
1	JX#**config**	进入全局配置模式。
2	JX(config)#**nqa server enable**	配置 nqa 服务端使能
3	JX(config)#**nqa server udp-connect** *ip-address udp-port*	配置 udp 服务端

请在客户端设备上进行以下配置。

步骤	配置	说明
1	JX#**config**	进入全局配置模式。
2	JX(config)#**nqa test-instance** *admin-name operate-tag* **type udp-echo**	配置测试实例,类型为 udp-echo
3	JX(nqa-admin-test-udp-echo)#**destination ip** *ip-address*	配置目的 ip 地址
4	JX(nqa-admin-test-udp-echo)#**destination port** *udp-port*	配置目的 udp 端口号
5	JX(nqa-admin-test-udp-echo)#**frequency** *freq-ms*	配置测试频率，单位：毫秒
6	JX(nqa-admin-test-udp-echo)#**probe count** *count*	配置每次测试探测的次数
7	JX(nqa-admin-test-udp-echo)#**probe timeout** *timeout-time-ms*	配置每次探测等待的时间，单位：毫秒

8	JX(config)#**nqa schedule** *admin-name operate-tag* **start-time now life-time forever**	配置检测调度策略为立即开始，一直循环 注：该配置与绑定 timerange 策略二选一即可
9	JX(config)#**nqa schedule** *admin-name operate-tag* **time-range** *timer-ange-list*	配置检测调度策略绑定 timerange 注：该策略与立即开始策略二选一即可

1.18.6 配置 TCP 测试

请在服务端设备上进行以下配置。

步骤	配置	说明
1	JX#**config**	进入全局配置模式。
2	JX(config)#**nqa server enable**	配置 nqa 服务端使能
3	JX(config)#**nqa server tcp-connect** *ip-address tcp-port*	配置 tcp 服务端

请在客户端设备上进行以下配置。

步骤	配置	说明
1	JX#**config**	进入全局配置模式。
2	JX(config)#**nqa test-instance** *admin-name operate-tag* **type tcp**	配置测试实例,类型为 tcp
3	JX(nqa-admin-test-tcp)#**destination ip** *ip-address*	配置目的 ip 地址
4	JX(nqa-admin-test-tcp)#**destination port** *tcp-port*	配置目的 tcp 端口号
5	JX(nqa-admin-test-tcp)#**frequency** *freq-ms*	配置测试频率，单位：毫秒
6	JX(nqa-admin-test-tcp)#**probe count** *count*	配置每次测试探测的次数
7	JX(nqa-admin-test-tcp)#**probe timeout** *timeout-time-ms*	配置每次探测等待的时间，单位：毫秒
8	JX(config)#**nqa schedule** *admin-name operate-tag* **start-time now life-time forever**	配置检测调度策略为立即开始，一直循环 注：该配置与绑定 timerange 策略二选一即可
9	JX(config)#**nqa schedule** *admin-name operate-tag* **time-range** *timer-ange-list*	配置检测调度策略绑定 timerange 注：该策略与立即开始策略二选一即可

1.18.7 配置 DNS 测试

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**config**	进入全局配置模式。
2	JX(config)#**nqa test-instance** *admin-name operate-tag* **type dns**	配置测试实例,类型为 dns
3	JX(nqa-admin-test-dns)#**destination ip** *ip-address*	配置 DNS 服务器地址
4	JX(nqa-admin-test-dns)#**resolve-target** *domain-name*	配置需要解析的域名
5	JX(nqa-admin-test-tcp)#**frequency** *freq-ms*	配置测试频率，单位：毫秒
6	JX(nqa-admin-test-tcp)#**probe count** *count*	配置每次测试探测的次数
7	JX(nqa-admin-test-tcp)#**probe timeout** *timeout-time-ms*	配置每次探测等待的时间，单位：毫秒
8	JX(config)#**nqa schedule** *admin-name operate-tag* **start-time now life-time forever**	配置检测调度策略为立即开始，一直循环 注：该配置与绑定 timerange 策略二选一即可
9	JX(config)#**nqa schedule** *admin-name operate-tag* **time-range** *timer-ange-list*	配置检测调度策略绑定 timerange 注：该策略与立即开始策略二选一即可

1.18.8 配置 HTTP 测试

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**config**	进入全局配置模式。
2	JX(config)#**nqa test-instance** *admin-name operate-tag* **type http**	配置测试实例,类型为 http
3	JX(nqa-admin-test-http)#**destination ip** *ip-address*	配置 HTTP 服务器地址
4	JX(nqa-admin-test-http)#**url** *url-path*	配置探测的 URL
5	JX(nqa-admin-test-http)#**frequency** *freq-ms*	配置测试频率，单位：毫秒
6	JX(nqa-admin-test-http)#**probe count** *count*	配置每次测试探测的次数
7	JX(nqa-admin-test-http)#**probe timeout** *timeout-time-ms*	配置每次探测等待的时间，单位：毫秒

8	JX(config)#**nqa schedule** *admin-name operate-tag* **start-time now life-time forever**	配置检测调度策略为立即开始，一直循环 注：该配置与绑定 timerange 策略二选一即可
9	JX(config)#**nqa schedule** *admin-name operate-tag* **time-range** *timer-ange-list*	配置检测调度策略绑定 timerange 注：该策略与立即开始策略二选一即可

1.18.9 配置 FTP 测试

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**config**	进入全局配置模式。
2	JX(config)#**nqa test-instance** *admin-name operate-tag* **type ftp**	配置测试实例,类型为 ftp
3	JX(nqa-admin-test-ftp)#**destination ip** *ip-address*	配置 FTP 服务器地址
4	JX(nqa-admin-test-ftp)#**username** *ftp-username*	配置 FTP 用户名
5	JX(nqa-admin-test-ftp)#**password** *ftp-password*	配置 FTP 用户密码
6	JX(nqa-admin-test-ftp)#**filename** *ftp-filename*	配置用于探测的文件名
7	JX(nqa-admin-test-ftp)#**operation** （**get** | **put**）	（可选）配置 ftp 操作，默认：get
5	JX(nqa-admin-test-ftp)#**frequency** *freq-ms*	配置测试频率，单位：毫秒
6	JX(nqa-admin-test-ftp)#**probe count** *count*	配置每次测试探测的次数
7	JX(nqa-admin-test-ftp)#**probe timeout** *timeout-time-ms*	配置每次探测等待的时间，单位：毫秒
8	JX(config)#**nqa schedule** *admin-name operate-tag* **start-time now life-time forever**	配置检测调度策略为立即开始，一直循环 注：该配置与绑定 timerange 策略二选一即可
9	JX(config)#**nqa schedule** *admin-name operate-tag* **time-range** *timer-ange-list*	配置检测调度策略绑定 timerange 注：该策略与立即开始策略二选一即可

1.18.10 配置 SNMP 测试

请在设备上进行以下配置。

步骤	配置	说明

1	JX#**config**	进入全局配置模式。
2	JX(config)#**nqa test-instance** *admin-name operate-tag* **type snmp**	配置测试实例,类型为 snmp
3	JX(nqa-admin-test-snmp)#**destination ip** *ip-address*	配置 SNMP 服务器地址
4	JX(nqa-admin-test-snmp)#**frequency** *freq-ms*	配置测试频率，单位：毫秒
5	JX(nqa-admin-test-snmp)#**probe count** *count*	配置每次测试探测的次数
6	JX(nqa-admin-test-snmp)#**probe timeout** *timeout-time-ms*	配置每次探测等待的时间，单位：毫秒
7	JX(config)#**nqa schedule** *admin-name operate-tag* **start-time now life-time forever**	配置检测调度策略为立即开始，一直循环 注：该配置与绑定 timerange 策略二选一即可
8	JX(config)#**nqa schedule** *admin-name operate-tag* **time-range** *timer-ange-list*	配置检测调度策略绑定 timerange 注：该策略与立即开始策略二选一即可

1.18.11 配置测试历史记录功能

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**config**	进入全局模式。
2	JX(config)#**nqa test-instance** *admin-name operate-tag* { **type** { **icmp-echo** | **udp-echo** | **tcp** | **dns** | **http** | **ftp** | **snmp** } }	进入任意 nqa 测试实例
3	JX(nqa-admin-test-icmp-echo)#**history-record enable**	使能历史记录功能
	JX(nqa-admin-test-icmp-echo)#**history-record keep-time** *time-minutes*	（可选）配置历史记录保存时间，单位：分钟

1.18.12 配置测试统计功能

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**config**	进入全局模式。
2	JX(config)#**nqa test-instance** *admin-name operate-tag* { **type** { **icmp-echo** | **udp-echo** | **tcp** | **dns** | **http** | **ftp** | **snmp** } }	进入任意 nqa 测试实例

3	JX(nqa-admin-test-icmp-echo)#**statistics enable**	使能测试统计功能
4	JX(nqa-admin-test-icmp-echo)#**statistics interval** *interval-minutes*	（可选）配置统计周期

1.18.13 配置测试告警功能

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**config**	进入全局模式。
2	JX(config)#**nqa test-instance** *admin-name operate-tag* { **type** { **icmp-echo** | **udp-echo** | **tcp** | **dns** | **http** | **ftp** | **snmp** } }	进入任意 nqa 测试实例
3	SwitchA(nqa-admin-test-icmp-echo)#**reaction trap probe-failure** *fail-count*	每次探测连续发生 n 次失败时，发送 trap
4	SwitchA(nqa-admin-test-icmp-echo)#**reaction trap test-failure** *fail-count*	每次探测累积发生 n 次失败时，发送 trap
5	SwitchA(nqa-admin-test-icmp-echo)#**reaction trap test-complete**	每次探测完成，发送 trap

1.18.14 检查配置

配置完成后，请在设备上执行以下命令检查配置结果。

序号	检查项	说明
1	JX#**show nqa config**	查看 nqa 配置信息。

1.18.15 维护

用户可以通过以下命令维护 NQA 特性。

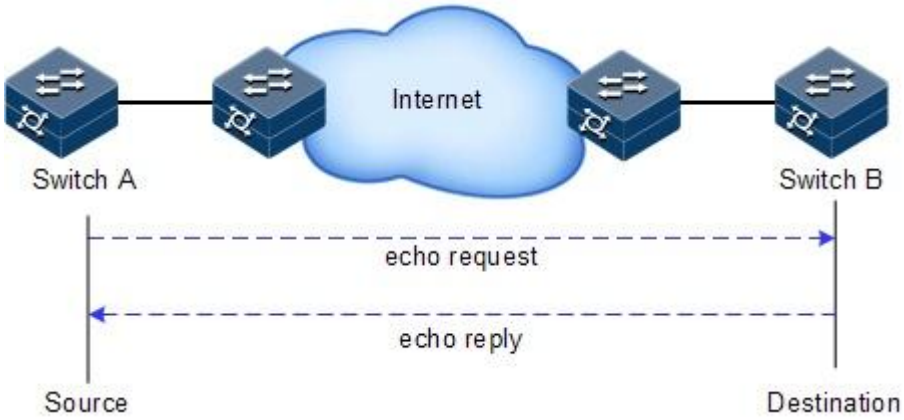
命令	描述
JX#**show nqa agent**	查看 nqa 客户端信息
JX#**show nqa server**	查看 nqa 服务端信息
JX#**show nqa result** *admin-name operate-tag*	查看 nqa 客户端测试结果
JX#**show nqa history** *admin-name operate-tag*	查看 nqa 客户端测试结果历史
JX#**show nqa statistics** *admin-name operate-tag*	查看 nqa 客户端测试结果统计

1.18.16 配置 ICMP-echo 测试功能示例

组网需求

如图 10-9 所示，在 Switch A 使能 NQA icmp-echo 探测，探测 SwitchA 与 SwitchB 之间的 IP 层连通性。

图 1-21 配置 ICMP-echo 测试功能组网示意图



配置步骤

步骤 1 配置 IP 层互通。

配置 Switch A。

```
JX#hostname SwitchA
SwitchA#config
SwitchA(config)#vlan 10
SwitchA(config-vlan-10)#quit
SwitchA(config)#interface ge 1/0/1
SwitchA(config-ge-1/0/1)#port hybrid vlan 10 tagged
SwitchA(config-ge-1/0/1)#quit
SwitchA(config)#interface vlan 10
SwitchA(config-vlanif-10)#ip address 10.1.1.1/24
```

配置 Switch B。

```
JX#hostname SwitchB
SwitchB#config
SwitchB(config)#vlan 10
SwitchB(config-vlan-10)#quit
SwitchB(config)#interface ge 1/0/1
SwitchB(config-ge-1/0/1)#port hybrid vlan 10 tagged
SwitchB(config-ge-1/0/1)#quit
SwitchB(config)#interface vlan 10
SwitchB(config-vlanif-10)#ip address 10.1.1.2/24
```

步骤 2 配置 NQA 的 icmp-echo 探测实例。

配置 Switch A。

```
SwitchA(config)#nqa test-instance admin test type icmp-echo
SwitchA(nqa-admin-test-icmp-echo)#destination ip 10.1.1.2
SwitchA(nqa-admin-test-icmp-echo)#frequency 10000
SwitchA(nqa-admin-test-icmp-echo)#probe count 3
SwitchA(nqa-admin-test-icmp-echo)#history-record enable
SwitchA(nqa-admin-test-icmp-echo)#reaction trap probe-failure 3
SwitchA(config)#nqa schedule admin test start-time now life-time
forever
```

检查结果

通过 **show nqa config** 命令查看本地配置是否正确。

```
SwitchA(config)#show nqa config
!
nqa test-instance admin test type icmp-echo
 frequency 10000
 history-record enable
 statistics enable
 reaction trap probe-failure 3
 probe count 3
 destination ip 10.1.1.2
nqa schedule admin test start-time now life-time forever
……
```

通过 **show nqa history admin test** 命令查看测试结果历史。

```
SwitchA(config)#show nqa history admin test
 Index    Response          Status           Time
-----------------------------------------------------------------
 ------------
 9        11              successed         2023-11-10
 17:05:26
 8        7               successed         2023-11-10
 17:05:26
 7        10              successed         2023-11-10
 17:05:26
 6        12              successed         2023-11-10
 17:05:16
 5        7               successed         2023-11-10
 17:05:16
 4        9               successed         2023-11-10
 17:05:16
 3        11              successed         2023-11-10
 17:05:05
 2        8               successed         2023-11-10
 17:05:05
 1        9               successed         2023-11-10
 17:05:05
-----------------------------------------------------------------
 ------------
```

1.19 PATCH 补丁功能

1.19.1 简介

patch 功能允许在不更换系统软件的情况下，通过补丁方式在线修复系统漏洞，解决软件问题。系统最多支持 32 个补丁。

1.19.2 配置准备

场景

系统发现的问题，可以通过补丁方式进行修复，无需重启设备。

前提

已准备好补丁文件。

1.19.3 加载补丁

设备上加载一个补丁的配置如下。

步骤	配置	说明
1	JX#**config**	进入全局配置模式。
2	JX(config)#**patch 1 load** *FILENAME*	加载一个 patch 补丁

1.19.4 激活补丁

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**config**	进入全局配置模式。
2	JX(config)#**patch 1 active**	激活 patch 1 补丁

1.19.5 去激活补丁

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**config**	进入全局配置模式。
2	JX(config)#**patch 1 deactive**	去激活 patch 1 补丁，使其失效

1.19.6 删除补丁

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**config**	进入全局配置模式。
2	JX(config)#**patch 1 delete**	删除 patch 1 补丁，使其失效

1.19.7 维护

用户可以通过以下命令维护 PATCH 特性。

命令	描述
JX#**show patch information**	查看设备的补丁情况

1.19.8 PATCH 示例

组网需求

通过打补丁的方式修复系统功能。

设备和 PC 通过 SNMP 接口互联，PC 上开启 TFTP 软件，以便从设备下载补丁文件。

配置步骤

步骤 1 通过 tftp 协议下载补丁文件到设备

```
JX(config)#tftp get 22.1.1.1 libmspaaa_patch.pat localfile
libmspaaa_patch.pat
```

步骤 2 加载补丁。

```
JX(config)#patch 1 load libmspaaa_patch.pat
```

步骤 3 激活补丁

```
JX(config)#patch 1 active
```

检查结果

通过 **show** patch information 命令查看 patch 状态：

```
JX(config)#show patch information
 Patch max number      : 32
 Patch current number  : 1
 Seq  ProcName   State  Type  Valid  EffetiveTime
PatchName
```

```
------------------------------------------------------------
-----------
 1    mspaaa   ACTIVE  REP   YES    2024-03-20 17:08:52
libmspaaa_patch.pat
```

1.20 定时备份配置功能

1.20.1 简介

定时备份配置功能目前是基于自动上传配置绑定 timerange 实现。

1.20.2 配置准备

场景

定时备份设备的当前配置到指定服务器。

前提

设备和 pc 可以 ping 通，且 pc 端 tftp 或者 ftp 服务器已打开。

1.20.3 配置 timerange

设备上配置一个 timerange list，并配置相应的规则配置如下。

步骤	配置	说明
1	`JX#config`	进入全局配置模式。
2	`JX(config)#time-range list` *id*	创建一个 timerange 列表，进入节点

3	JX(config-timerange-*)#**time-range** *timerange-id* **absolute from** *hour:minute:second year/month/monthday* **to** *hour:minute:second year/month/monthday* JX(config-timerange-*)#**time-range** *timerange-id* **everyhour** *minute:second* **to** *minute:second* JX(config-timerange-*)#**time-range** *timerange-id* **everyday** *hour:minute:second* **to** *hour:minute:second* JX(config-timerange-*)#**time-range** *timerange-id* **everyweek** *hour:minute:second* { **mon** | **tue** | **wed** | **thu** | **fri** | **sat** | **sun** } **to** { **mon** | **tue** | **wed** | **thu** | **fri** | **sat** | **sun** } JX(config-timerange-*)#**time-range** *timerange-id* **everymonth** *hour:minute:second monthday* **to** *hour:minute:second monthday* JX(config-timerange-*)#**time-range** *timerange-id* **everyweekend** *hour:minute:second* **to** *hour:minute:second* JX(config-timerange-*)#**time-range** *timerange-id* **everyyear** *hour:minute:second month/monthday* **to** *hour:minute:second month/monthday*	创建时间段，可应用于自动上传。 相对时间段（周期时间段）：是指由 everyxxx 描述的以一周为间隔的周期性时间，同时可以依靠绝对时间段指明时间范围有效日期。 绝对时间段：是指由 absolute 指定的特定的时间范围。可以通过绝对时间段指定的时间范围来限制相对时间段（周期时间段）在什么时间范围生效。 如果该时间段配置了多个生效时间，生效原则为：周期时间段之间取“或”，周期时间段和绝对时间段之间取“与”

1.20.4 配置自动上传

请在设备上进行以下配置。

步骤	配置	说明
1	JX#**config**	进入全局配置模式。
2	JX(config)#**auto-upload start**	启动自动上传功能
3	JX(config)#**auto-upload tftp server** {*ipv4-address*|*ipv6-address*} *remotefile* {**config**|**running-config**|**log**} [**interval** *interval-value*| **port** *port-number*|**time-range** *list-number*]*	配置自动上传 tftp 服务器属性并指定上传的文件，绑定的 timerange 列表
4	JX(config)#**auto-upload ftp server** {*ipv4-address*|*ipv6-address*} *username password remotefile* {**config**|**running-config**|**log**} [**interval** *interval-value*| **port** *port-number*|**time-range** *list-number*]*	配置自动上传 ftp 服务器属性并指定上传的文件，绑定的 timerange 列表

1.20.5 检查配置

配置完成后，请在设备上执行以下命令检查配置结果。

命令	描述
JX#show auto-upload server	查看配置的自动上传表项
JX#show auto-upload config	查看自动上传的配置

1.20.6 定时备份配置示例

组网需求

通过 tftp 方式自动备份设备当前配置到 PC。

设备和 PC 通过 SNMP 接口互联，PC 上开启 TFTP 软件。

配置步骤

步骤 1 配置 timerange list 1 并配置规则 1（以每小时定时备份举例）

```
JX(config)#time-range list 1。
JX(config-timerange-1)#time-range 1 everyhour 3:10 to 13:10
```

步骤 2 配置自动上传属性,启动自动上传功能，创建表项（tftp 服务器地址 192.168.62.1,本地文件为运行配置文件，上传到服务器的文件为 config.txt），并绑定 timerange list 1

```
JX(config)#auto-upload start
JX(config)#auto-upload tftp server 192.168.62.1 config.txt
running-config time-range 1
```

步骤 3 配置完成后，每小时 3 分 10 秒设备会定时上传运行配置文件到 PC 端 tftp 服务器一次

检查结果

通过 **show auto-upload server** 命令查看所配置的自动上传属性：

```
JX(config)#show auto-upload server
 Type  Server               Port    Interval(min) Localfile
Vpn            TimerangeId
------------------------------------------------------------
--------------------------------------
 tftp  192.168.62.1           69      10
running-config     n/a             1
------------------------------------------------------------
--------------------------------------
JX(config)#
```