



# H3C S5560X-EI 系列以太网交换机

## 二层技术-以太网交换配置指导

新华三技术有限公司  
<http://www.h3c.com>

资料版本：6W102-20171128  
产品版本：Release 1110 系列

Copyright © 2017 新华三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、**H3C**、H3CS、H3CIE、H3CNE、Aolynk、、H<sup>3</sup>Care、、IRF、NetPilot、Netflow、SecEngine、SecPath、SecCenter、SecBlade、Comware、ITCMM、HUASAN、华三均为新华三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

本配置指导主要介绍以太网交换技术的原理及具体配置方法。通过这些技术您可以实现流量控制、流量的负载分担、同一 VLAN 内用户隔离、二层环路消除、VLAN 划分、私网报文穿越公网、修改报文的 VLAN Tag 等功能。

前言部分包含如下内容：

- [读者对象](#)
- [本书约定](#)
- [资料获取方式](#)
- [技术支持](#)
- [资料意见反馈](#)

## 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
<b>粗体</b>	命令行关键字（命令中保持不变、必须照输的部分）采用 <b>加粗</b> 字体表示。
<i>斜体</i>	命令行参数（命令中必须由实际值进行替代的部分）采用 <i>斜体</i> 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. 图形界面格式约定

格 式	意 义
<>	带尖括号“<>”表示按钮名，如“单击<确定>按钮”。

格式	意义
[ ]	带方括号“[ ]”表示窗口名、菜单名和数据表，如“弹出[新建用户]窗口”。
/	多级菜单用“/”隔开。如[文件/新建/文件夹]多级菜单表示[文件]菜单下的[新建]子菜单下的[文件夹]菜单项。

### 3. 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。

### 4. 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。
	该图标及其相关描述文字代表无线控制器、无线控制器业务板和有线无线一体化交换机的无线控制引擎设备。
	该图标及其相关描述文字代表无线接入点设备。
	该图标及其相关描述文字代表无线终结单元。
	该图标及其相关描述文字代表无线终结者。
	该图标及其相关描述文字代表无线Mesh设备。
	该图标代表发散的无线射频信号。
	该图标代表点到点的无线射频信号。

	<p>该图标及其相关描述文字代表防火墙、UTM、多业务安全网关、负载均衡等安全设备。</p>
	<p>该图标及其相关描述文字代表防火墙插卡、负载均衡插卡、NetStream插卡、SSL VPN插卡、IPS插卡、ACG插卡等安全插卡。</p>

## 5. 示例约定

由于设备型号不同、配置不同、版本升级等原因，可能造成本手册中的内容与用户使用的设备显示信息不一致。实际使用中请以设备显示的内容为准。

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。

## 资料获取方式

您可以通过H3C网站（[www.h3c.com](http://www.h3c.com)）获取最新的产品资料：

- 获取安装类、配置类或维护类产品资料  
[http://www.h3c.com/cn/Technical\\_Documents](http://www.h3c.com/cn/Technical_Documents)
- 获取版本说明书等与软件版本配套的资料  
[http://www.h3c.com/cn/Software\\_Download](http://www.h3c.com/cn/Software_Download)

## 技术支持

用户支持邮箱：[service@h3c.com](mailto: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>

## 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](mailto:info@h3c.com)

感谢您的反馈，让我们做得更好！

# 目 录

1 以太网接口配置.....	1-1
1.1 管理以太网口配置.....	1-1
1.2 以太网接口编号规则.....	1-1
1.3 以太网接口通用配置.....	1-2
1.3.1 Combo接口配置（单Combo接口）.....	1-2
1.3.2 40GE接口和 10GE接口的拆分与合并.....	1-2
1.3.3 以太网接口基本配置.....	1-3
1.3.4 配置以太网接口的工作模式.....	1-4
1.3.5 配置以太网接口允许超长帧通过.....	1-5
1.3.6 配置以太网接口物理连接状态抑制功能.....	1-5
1.3.7 配置以太网接口dampening功能.....	1-6
1.3.8 开启以太网接口的环回功能.....	1-8
1.3.9 配置以太网接口的流量控制功能.....	1-8
1.3.10 配置以太网接口节能功能.....	1-9
1.3.11 配置以太网接口统计信息的时间间隔.....	1-10
1.3.12 配置以太网接口速率自动降级协商功能.....	1-10
1.3.13 配置广播/组播/未知单播风暴抑制功能.....	1-10
1.4 二层以太网接口的配置.....	1-11
1.4.1 配置以太网接口自协商速率.....	1-11
1.4.2 配置以太网接口流量阈值控制功能.....	1-12
1.4.3 强制开启光口.....	1-13
1.4.4 配置以太网接口的MDIX模式.....	1-15
1.4.5 检测以太网接口的连接电缆.....	1-16
1.4.6 配置以太网桥功能.....	1-16
1.5 三层以太网接口的配置.....	1-16
1.5.1 配置以太网接口的MTU.....	1-16
1.6 以太网接口显示和维护.....	1-17

# 1 以太网接口配置

本系列交换机支持的接口类型包括：以太网接口，管理用以太网口，Console 口，USB 口。具体机型支持的接口类型及接口数量可参见产品的安装手册。

本章节主要介绍有关管理用以太网口和以太网接口的相关配置及命令。

## 1.1 管理用以太网口配置

### 1. 管理用以太网口介绍

该端口采用 RJ-45 连接器，一般用来连接后台计算机以进行系统的程序加载、调试等工作，也可以连接远端的网管工作站等设备以实现系统的远程管理。

### 2. 管理用以太网口基本配置

表1-1 管理用以太网口基本配置

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入管理用以太网口视图	<b>interface M-GigabitEthernet</b> <i>interface-number</i>	-
(可选) 设置当前管理用以太网口的描述信息	<b>description</b> <i>text</i>	缺省情况下，管理用以太网口的描述信息为M-GigabitEthernet0/0/0 Interface
(可选) 关闭管理用以太网口	<b>shutdown</b>	缺省情况下，管理用以太网口处于打开状态

## 1.2 以太网接口编号规则

本系列交换机的以太网接口均采用 3 维编号方式：interface type A/B/C。

- **A**: IRF 中成员设备的编号，若未形成 IRF，其取值默认为 1。
- **B**: 设备上的槽位号。取值为 0，表示设备上固有接口所在的槽位。取值为 1，表示接口模块扩展卡 1 上端口所在的槽位。
- **C**: 某槽位上的端口编号。

需要注意的是：

由 40GE 接口拆分后的 10GE 接口的编号方式为：interface type A/B/C:D。其中的 A/B/C 对应该 40GE 接口的编号；D 表示拆分后的 10GE 的顺序编号，取值为 1~4。有关 40GE 接口的拆分可参见“[1.3.2 40GE 接口和 10GE 接口的拆分与合并](#)”。

## 1.3 以太网接口通用配置

该部分介绍了二层以太网接口和三层以太网接口的共有属性及其配置，各自的特有属性请参见下文中“[1.4 二层以太网接口的配置](#)”和“[1.5 三层以太网接口的配置](#)”。

### 1.3.1 Combo接口配置（单Combo接口）

#### 1. Combo接口介绍

Combo 接口是一个逻辑接口，一个 Combo 接口在物理上对应设备面板上一个电口和一个光口。电口与其对应的光口共用一个转发接口和接口视图，所以，两者不能同时工作。当激活其中的一个接口时，另一个接口就自动处于禁用状态。用户可根据组网需求选择使用电口或光口。当用户需要激活电口或光口、配置电口或光口的属性（例如速率、双工等）时，在同一接口视图下配置。

#### 2. 配置准备

- 请根据设备面板上的标识了解设备上有哪些 Combo 接口以及每个 Combo 接口的编号。
- 通过 **display interface** 命令了解当前处于激活状态的是电口还是光口。如果显示信息中包含“Media type is twisted pair, Port hardware type is 1000\_BASE\_T”，则表示电口处于激活状态，否则，则表示光口处于激活状态。也可在 Combo 端口视图下执行 **display this** 命令查看当前视图下的配置，若存在 **combo enable fiber** 命令，则表示光口处于激活状态，否则，则表示电口处于激活状态。

#### 3. 配置步骤

表1-2 配置 Combo 接口的状态

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入以太网接口视图	<b>interface interface-type interface-number</b>	-
激活Combo接口中的电口或者光口	<b>combo enable { copper   fiber }</b>	缺省情况下，电口处于激活状态

### 1.3.2 40GE接口和 10GE接口的拆分与合并

#### 1. 将一个 40GE接口拆分成四个 10GE接口

40GE 接口可以作为一个单独的接口使用，也可以拆分成四个 10GE 接口。将一个 40GE 接口拆分成四个 10GE 接口，从而能够提高端口密度，减少用户使用成本，增加组网灵活性。拆分出来的 10GE 接口除了接口编号方式外，支持的配置和特性均和普通 10GE 物理接口相同。例如，40GE 接口 FortyGigE1/0/1 可以拆分成四个 10GE 接口 Ten-GigabitEthernet1/0/1:1 ~ Ten-GigabitEthernet1/0/1:4。

40GE 接口拆分后需要使用一分四的专用线缆连接，关于线缆的具体描述请参见产品的相关手册。

表1-3 将一个 40GE 接口拆分成四个 10GE 接口

操作	命令	说明
进入系统视图	<b>system-view</b>	-

操作	命令	说明
进入40GE以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
将一个40GE接口拆分成四个10GE接口	<b>using tengige</b>	缺省情况下，40GE接口作为单个接口使用，不拆分

#### 说明

- 通过 **using tengige** 命令配置成功后，不需要重启设备，通过执行 **display interface brief** 命令就可以看到拆分成的四个 10GE 接口。
- 40GE 接口拆分后需要使用一分四的专用线缆与对端的 10GE 接口进行连接，关于线缆的具体描述请参见产品的安装手册。

## 2. 将四个 10GE 拆分接口合并成一个 40GE 接口

如果用户需要更大的带宽，可以将已拆分的 10GE 接口合并为 40GE 接口使用。

合并后，需要将一分四的专用线缆连接更换成一对一的专用线缆或者 40GE 光模块连接光纤，关于线缆的具体描述请参见产品的相关手册。

表1-4 将四个 10GE 拆分接口合并成一个 40GE 接口

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入任意一个因拆分生成的10GE接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
将四个10GE拆分接口合并成一个40GE接口	<b>using fortygige</b>	缺省情况下，40GE接口作为单个接口使用，未拆分

#### 说明

- 通过 **using fortygige** 命令配置成功后，不需要重启设备，通过执行 **display interface brief** 命令就可以看到合并后的 40GE 接口。
- 合并后，需要将一分四的专用线缆连接更换成一对一的专用线缆或者更换成 40GE 光模块连接光纤，关于线缆和光模块的具体描述请参见产品的安装手册。

### 1.3.3 以太网接口基本配置

设置以太网接口的双工模式时存在以下几种情况：

- 当希望接口在发送数据包的同时可以接收数据包，可以将接口设置为全双工（**full**）属性；
- 当希望接口同一时刻只能发送数据包或接收数据包时，可以将接口设置为半双工（**half**）属性；
- 当设置接口为自协商（**auto**）状态时，接口的双工状态由本接口和对端接口自动协商而定。

设置以太网接口的速率时，当设置接口速率为自协商（**auto**）状态时，接口的速率由本接口和对端接口双方自动协商而定。对于百兆或者千兆二层以太网接口，可以根据端口的速率自协商能力，指定自协商速率，让速率在指定范围内协商，具体配置请参见“[1.4.1 配置以太网接口自协商速率](#)”。

表1-5 以太网接口基本配置

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
设置当前接口的描述信息	<b>description</b> <i>text</i>	缺省情况下，接口的描述信息为“ <i>接口名</i> Interface”，例如：GigabitEthernet1/0/1 Interface
设置以太网接口的双工模式	<b>duplex</b> { <b>auto</b>   <b>full</b>   <b>half</b> }	光口和配置了速率为1000，10000的以太网电口不支持配置 <b>half</b> 参数 缺省情况下，以太网接口的双工模式为 <b>auto</b> （自协商）状态
设置以太网接口的速率	<b>speed</b> { <b>10</b>   <b>100</b>   <b>1000</b>   <b>10000</b>   <b>40000</b>   <b>auto</b> }	缺省情况下，以太网接口的速率为 <b>auto</b> （自协商）状态 不同类型的接口支持配置的参数不同，具体情况请在相关接口视图下执行 <b>speed ?</b> 命令查看
配置接口的期望带宽	<b>bandwidth</b> <i>bandwidth-value</i>	缺省情况下，接口的期望带宽=接口的波特率÷1000（kbps）
恢复接口的缺省配置	<b>default</b>	-
打开以太网接口	<b>undo shutdown</b>	缺省情况，以太网接口处于开启状态 <ul style="list-style-type: none"> <li>• <b>shutdown</b>、<b>port up-mode</b> 命令互斥，后配置的失败</li> <li>• 在进行环回测试时，禁止在接口上配置 <b>shutdown</b> 命令</li> </ul>

### 1.3.4 配置以太网接口的工作模式



#### 注意

工作模式切换后，除了 **shutdown** 和 **combo enable** 命令，该以太网接口下的其它所有命令都将恢复到新模式下的缺省情况。

本系列交换机上的接口比较灵活，工作模式可以通过命令行设置。

- 如果将工作模式设置为二层模式（**bridge**），则作为一个二层以太网接口使用。
- 如果将工作模式设置为三层模式（**route**），则作为一个三层以太网接口使用。

表1-6 配置以太网接口的工作模式

操作	命令	说明
进入系统视图	<b>system-view</b>	-

操作	命令	说明
进入以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
切换以太网接口工作模式	<b>port link-mode</b> { <b>bridge</b>   <b>route</b> }	缺省情况下，设备上的接口均工作在二层模式

### 1.3.5 配置以太网接口允许超长帧通过

以太网接口在进行文件传输等大吞吐量数据交换的时候，可能会收到大于标准以太网帧长的帧，这种帧称为超长帧。系统对于超长帧的处理如下：

- 如果系统配置了禁止超长帧通过，会直接丢弃该帧不再进行处理。
- 如果系统允许超长帧通过，当接口收到长度在指定范围内的超长帧时，系统会继续处理；当接口收到长度超过指定最大长度的超长帧时，系统会直接丢弃该帧不再进行处理。

表1-7 配置允许超长帧通过以太网接口

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
允许超长帧通过	<b>jumboframe enable</b> [ <i>size</i> ]	缺省情况下，设备允许最大长度为10000字节的超长帧通过 指定允许通过的超长帧的长度时，仅支持配置为偶数

### 1.3.6 配置以太网接口物理连接状态抑制功能



提示

对于开启了生成树协议、RRPP 或 Smart Link 的端口不推荐使用该功能。

以太网接口有两种物理连接状态：**up** 和 **down**。当接口状态发生改变时，接口会立即上报 CPU，CPU 会立即通知上层协议模块（例如路由、转发）以便指导报文的收发，并自动生成 Trap 和 Log 信息，来提醒用户是否需要物理链路进行相应处理。

如果短时间内接口物理状态频繁改变，上述处理方式会给系统带来额外的开销。此时，可以在接口下设置物理连接状态抑制功能，使得在抑制时间内，系统忽略接口的物理状态变化；经过抑制时间后，如果状态还没有恢复，再进行处理。

在配置本特性时，选取的参数不同，抑制效果不同：

- 不指定 **mode** 参数：表示接口状态从 **up** 变成 **down** 时，不会立即上报 CPU。而是等待 *delay-time* 时间后，再检查接口状态，如果状态仍然是 **down**，再上报。接口状态从 **down** 变成 **up** 时，立即上报 CPU。

- **mode up:** 表示接口状态从 down 变成 up 时，不会立即上报 CPU。而是等待 *delay-time* 时间后，再检查接口状态，如果状态仍然是 up，再上报。接口状态从 up 变成 down 时，立即上报 CPU。
- **mode updown:** 表示接口状态从 up 变成 down 或者 down 变成 up 时，都不会立即上报 CPU。等待 *delay-time* 时间后，再检查接口状态，如果状态仍然是 down 或者 up，再上报。

同一接口下，接口状态从 up 变成 down 的抑制时间和接口状态从 down 变成 up 的抑制时间可以不同。如果在同一端口下，多次执行本命令配置了不同的抑制时间，则两个抑制时间会分别以最新配置为准。

表1-8 设置以太网接口物理连接状态抑制功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入以太网接口视图	<b>interface interface-type interface-number</b>	-
配置以太网接口物理连接状态抑制功能	<b>link-delay [ msec ] delay-time [ mode { up   updown } ]</b>	缺省情况下，接口状态改变时，系统会将接口状态改变立即上报CPU 在同一端口上，多次配置该命令，最新配置生效

### 1.3.7 配置以太网接口dampening功能

由于线缆故障、接口连接或链路层配置错误等问题，可能会导致设备接口的状态频繁的在 down 和 up 之间切换，这种现象称为接口震荡。随着接口状态的频繁改变，设备会不停的刷新相关表项（比如路由表），消耗大量的系统资源。通过在接口上配置 dampening 功能，可以在一定条件下，屏蔽该接口的震荡对路由等上层业务的影响。此时若出现接口震荡，将不上送 CPU 处理，仅产生对应的 Trap 和 Log 信息，从而节省系统资源的消耗。

dampening 功能中各参数解释如下：

- 惩罚值 (Penalty): 配置 dampening 功能后，接口对应一个惩罚值，初始值为 0。接口状态从 up 变到 down 时，惩罚值会增加 1000；接口状态从 down 变到 up 时，惩罚值不变。同时，惩罚值随时间推移自动减少，满足半衰期衰减规律：完全衰减时（假如没有接口震荡），经过一个半衰周期，惩罚值减少为原来值的一半。
- 最大惩罚值 (Ceiling): 当惩罚值达到此值后，惩罚值将不再增加。每次接口进入抑制状态后，持续抑制的时间超过最大抑制时间时，惩罚值不再增加，此时惩罚值进入完全半衰期（此阶段接口状态变化不会增加惩罚值），直到惩罚值小于启用值，不再抑制接口（完全半衰时，接口仍然处于抑制状态，但完全半衰阶段时间不算入持续抑制时间）。
- 抑制值 (Suppress-limit): 当惩罚值大于或等于这个门限时，抑制接口，即当接口状态变化时，不上送 CPU 处理，仅产生对应的 Trap 和 Log 信息。
- 启用值 (Reuse-limit): 当惩罚值小于或等于这个门限时，不抑制接口，即当接口状态变化时，上送 CPU 处理，同时产生对应的 Trap 和 Log 信息。
- 半衰期 (Decay): 此阶段惩罚值随着时间的推移自动的减少，满足半衰期衰减规律，即经过一个半衰周期，惩罚值减半。

- 最大抑制时间 (Max-suppress-time): 如果接口一直不稳定, 网络设备不能一直抑制它, 必须要设定一个最大的抑制时间。最大抑制时间后, 惩罚值进入完全半衰期。

其中, 最大惩罚值与最大抑制时间、半衰期、启用值之间遵循公式:  $\text{最大惩罚值} = 2^{(\text{最大抑制时间}/\text{半衰期})} \times \text{启用值}$ , 其中最大惩罚值不可配。惩罚值的变化规律如下图所示。

图1-1 dampening 惩罚值变化规律图



图 1-1 中,  $t_0$  为抑制开始时间, 从  $t_0$  开始经过最大抑制时间后达到  $t_1$ ,  $t_2$  为抑制结束时间。  $t_0$  至  $t_2$  段对应接口抑制期,  $t_0$  至  $t_1$  段对应最大抑制时间,  $t_1$  至  $t_2$  段对应完全半衰期 (此阶段惩罚值不再增加)。

配置 dampening 功能时, 需要注意:

- 以太网接口上不能同时配置本功能和 **link-delay** 命令。
- 本功能对使用 **shutdown** 命令手动关闭的接口无效。
- 手工 **shutdown** 接口时, dampening 的惩罚值恢复为初始值 0。
- 对于开启了 RRPP、MSTP 或 Smart Link 的接口不建议配置该功能。

表1-9 配置以太网接口 dampening 功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入以太网接口视图	<b>interface interface-type interface-number</b>	-
开启接口的dampening功能	<b>dampening [ half-life reuse suppress max-suppress-time ]</b>	缺省情况下, 接口的dampening功能处于关闭状态

### 1.3.8 开启以太网接口的环回功能

该功能用于检测以太网转发通路能否正常工作。环回功能包括内部环回和外部环回：

- 内部环回：配置内部环回后，接口将需要从接口转发出去的报文返回给设备内部，让报文向内部线路环回。内部环回用于定位设备是否故障。
- 外部环回：配置外部环回后，接口将来自对端设备的报文返回给对端设备，让报文向外部线路环回。外部环回用于定位设备间链路是否故障。

需要注意的是：

- 开启环回功能后，接口将不能正常转发数据包，请按需配置。
- 手工关闭以太网接口（接口状态显示为 **ADM** 或者 **Administratively DOWN**）时，则不能进行内部和外部环回测试。
- 在进行环回测试时系统将禁止在接口上进行 **speed**、**duplex**、**mdix-mode** 和 **shutdown** 命令的配置。
- 配置了 **port up-mode** 的以太网接口，不能进行环回测试。
- 开启环回功能后，接口将自动切换到全双工模式，关闭环回功能后会自动恢复原有双工模式。

表1-10 开启以太网接口的环回功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
开启以太网接口的环回功能	<b>loopback</b> { <b>external</b>   <b>internal</b> }	缺省情况下，以太网接口环回测试功能处于关闭状态

### 1.3.9 配置以太网接口的流量控制功能

以太网接口流量控制功能的基本原理是：如果本端设备发生拥塞，将通知对端设备暂时停止发送报文；对端设备收到该消息后将暂时停止向本端发送报文；反之亦然。从而避免了报文丢失现象的发生。

- 配置 **flow-control** 命令后，设备具有发送和接收流量控制报文的能力：
  - 当本端发生拥塞时，设备会向对端发送流量控制报文。
  - 当本端收到对端的流量控制报文后，会停止报文发送。
- 配置 **flow-control receive enable** 命令后，设备具有接收流量控制报文的能力，但不具有发送流量控制报文的能力。
  - 当本端收到对端的流量控制报文，会停止向对端发送报文。
  - 当本端发生拥塞时，设备不能向对端发送流量控制报文。

因此，如果要应对单向网络拥塞的情况，可以在一端配置 **flow-control receive enable**，在对端配置 **flow-control**；如果要求本端和对端网络拥塞都能处理，则两端都必须配置 **flow-control**。

表1-11 开启以太网接口的流量控制功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
开启以太网接口的流量控制功能	<b>flow-control</b>	二者选其一 缺省情况下,以太网接口的流量控制功能处于关闭状态
配置以太网接口的接收流量控制功能	<b>flow-control receive enable</b>	

### 1.3.10 配置以太网接口节能功能



说明

光口不支持本特性。

#### 1. 配置down状态接口节能功能

配置 down 状态接口节能功能后,如果在连续一段时间(由芯片规格决定,不能通过命令行配置)内接口状态始终为 down,则系统会自动停止对该接口供电,接口自动进入节能模式;当接口状态变为 up 时,系统会自动恢复对该接口供电,接口自动进入正常模式,从而达到节能的效果。

表1-12 配置 down 状态接口节能功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
开启down状态接口节能功能	<b>port auto-power-down</b>	缺省情况下,down状态接口节能功能处于关闭状态

#### 2. 开启EEE节能功能(配置up状态接口节能功能)

接口开启 EEE (Energy Efficient Ethernet, 高效节能以太网) 功能后,如果在连续一段时间(由芯片规格决定,不能通过命令行配置)内接口状态始终为 up 且没有收发任何报文,则接口自动进入低功耗模式;当接口需要收发报文时,接口又自动恢复到正常工作模式,从而达到节能的效果。

表1-13 开启 EEE 节能功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
开启EEE节能功能	<b>eee enable</b>	缺省情况下,EEE节能功能处于关闭状态

### 1.3.11 配置以太网接口统计信息的时间间隔

使用本特性可以设置统计以太网接口报文信息的时间间隔。使用 **display interface** 命令可以显示端口在该间隔时间内统计的报文信息。使用 **reset counters interface** 命令可以清除端口的统计信息。

表1-14 在以太网接口视图下配置以太网接口统计信息的时间间隔

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
配置接口统计信息的时间间隔	<b>flow-interval</b> <i>interval</i>	缺省情况下，接口统计报文信息的时间间隔为300秒

### 1.3.12 配置以太网接口速率自动降级协商功能

本设备和对端设备以千兆以太网接口相连，且两接口速率均配置为自动协商，则协商后的速率为1000Mbit/s。但由于网线老化等原因，接口实际支持的工作速率仅为100Mbit/s，导致链路上存在丢包现象。此时，请开启以太网接口速率自动降级协商功能，以便接口可以将协商速率下降到100Mbit/s，保证接口正常工作。

以太网接口速率自动降级协商功能仅支持在千兆以太网接口下配置。

表1-15 配置以太网接口速率自动降级协商功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
配置以太网接口速率自动降级协商功能	<b>speed auto downgrade</b>	缺省情况下，以太网接口速率自动降级协商功能处于开启状态

### 1.3.13 配置广播/组播/未知单播风暴抑制功能

在接口上配置了广播/组播/未知单播风暴抑制功能后，当接口上的广播/组播/未知单播流量超过用户设置的抑制阈值时，系统会丢弃超出流量限制的报文，从而使接口的广播/组播/未知单播流量降低到限定范围内，保证网络业务的正常运行。

执行 **storm-constrain** 与 **broadcast-suppression**、**multicast-suppression**、**unicast-suppression** 命令都能开启端口的风暴抑制功能。**storm-constrain** 命令通过软件对报文流量进行抑制，对设备性能有一定影响；**broadcast-suppression**、**multicast-suppression**、**unicast-suppression** 通过芯片物理上对报文流量进行抑制，相对 **storm-constrain** 来说，对设备性能影响较小。对于某种类型的报文流量，请不要同时配置这两种方式，以免配置冲突，导致抑制效果不确定。**storm-constrain** 命令的详细描述请参见“[1.4.2 配置以太网接口流量阈值控制功能](#)”。

表1-16 配置以太网接口的风暴抑制比

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
开启端口广播风暴抑制功能，并设置广播风暴抑制阈值	<b>broadcast-suppression</b> { <i>ratio</i>   <b>pps</b> <i>max-pps</i>   <b>kbps</b> <i>max-kbps</i> }	缺省情况下，所有接口不对广播流量进行抑制
开启端口组播风暴抑制功能，并设置组播风暴抑制阈值	<b>multicast-suppression</b> { <i>ratio</i>   <b>pps</b> <i>max-pps</i>   <b>kbps</b> <i>max-kbps</i> }	缺省情况下，所有接口不对组播流量进行抑制
开启端口未知单播风暴抑制功能，并设置未知单播风暴抑制阈值	<b>unicast-suppression</b> { <i>ratio</i>   <b>pps</b> <i>max-pps</i>   <b>kbps</b> <i>max-kbps</i> }	缺省情况下，所有接口不对未知单播流量进行抑制

 说明

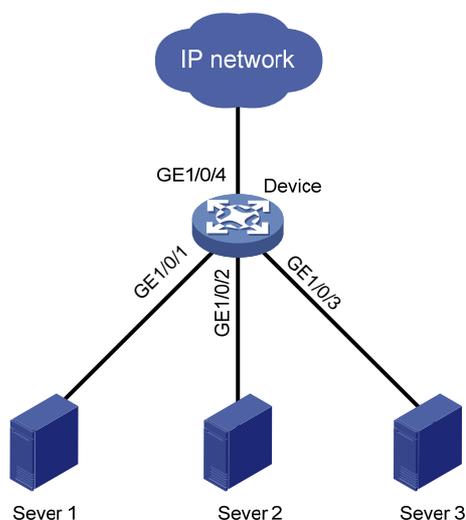
当风暴抑制阈值配置为 **kbps** 时，若配置值小于 64，则实际生效的数值为 64；若配置值大于 64 但不是 64 的整数倍，则实际生效的数值为大于且最接近于配置值的 64 的整数倍。请注意查看设备的提示信息。

## 1.4 二层以太网接口的配置

### 1.4.1 配置以太网接口自协商速率

通常情况下，设备以太网接口速率是通过和对端自协商决定的。协商得到的速率可以是接口速率能力范围内的任意一个速率。通过配置自协商速率可以让以太网接口在能力范围内只协商部分速率，从而可以控制速率的协商。

图1-2 以太网接口自协商速率应用示意图



如 图 1-2 所示，服务器群（Server 1、Server 2 和 Server 3）通过 Device 与外部网络相连，该服务器群中每台服务器的网卡速率均为 1000Mbps，Device 与外部网络相连接口 GigabitEthernet1/0/4 的速率也为 1000Mbps。如果在 Switch A 上不指定自协商速率范围，则接口 GigabitEthernet1/0/1、GigabitEthernet1/0/2 和 GigabitEthernet1/0/3 与各服务器网卡进行速率协商的结果将均为 1000Mbps，这样就可能造成出接口 GigabitEthernet1/0/4 的拥塞。在这种情况下，可通过将接口 GigabitEthernet1/0/1、GigabitEthernet1/0/2 和 GigabitEthernet1/0/3 的自协商速率范围分别设置为 100Mbps，来避免出接口的拥塞。

表 1-17 配置以太网接口自协商速率

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入以太网接口视图	<b>interface interface-type interface-number</b>	-
设置以太网接口的自协商速率范围	<b>speed auto { 10   100   1000 } *</b>	缺省情况下，未配置以太网接口的自协商速率范围 如果多次使用 <b>speed</b> 、 <b>speed auto</b> 命令设置接口的速率，则最新配置生效

## 1.4.2 配置以太网接口流量阈值控制功能

### 1. 端口流量阈值控制简介

端口流量阈值控制功能用于控制以太网上的报文风暴。启用该功能的端口会定时检测到达端口的未知单播报文流量、组播报文流量和广播报文流量。如果某类报文流量超过预先设置的上限阈值时，用户可以通过配置来决定是阻塞该端口还是关闭该端口，以及是否输出 Log 和 Trap 信息。

- 配置成 **block** 方式：当端口上未知单播、组播或广播报文中某类报文的流量大于其上限阈值时，端口将暂停转发该类报文（其它类型报文照常转发），端口处于阻塞状态，但仍会统计该类报文的流量。当该类报文的流量小于其下限阈值时，端口将自动恢复对此类报文的转发。
- 配置成 **shutdown** 方式：当端口上未知单播、组播或广播报文中某类报文的流量大于其上限阈值时，端口将被关闭，系统停止转发所有报文。当该类报文的流量小于其下限阈值时，端口状态不会自动恢复，此时可通过执行 **undo shutdown** 命令或取消端口上流量阈值的配置来恢复。

本特性实现中系统需要一个完整的周期（周期长度为 *seconds*）来收集流量数据，下一个周期分析数据、采取相应的控制措施。因此，开启端口流量阈值控制功能后，如果某类报文流量超过预先设置的上限阈值，控制动作最短将在一个周期后执行，最长不会超过两个周期。

执行 **storm-constrain** 与 **broadcast-suppression**、**multicast-suppression**、**unicast-suppression** 命令都能开启端口的风暴抑制功能。**storm-constrain** 命令通过软件对报文流量进行抑制，对设备性能有一定影响，**broadcast-suppression**、**multicast-suppression**、**unicast-suppression** 通过芯片物理上对报文流量进行抑制，相对 **storm-constrain** 来说，对设备性能影响较小。对于某种类型的报文流量，请不要同时配置这两种方式，以免配置冲突，导致抑制效果不确定。**broadcast-suppression**、**multicast-suppression**、**unicast-suppression** 命令的详细描述请参见“[1.3.13 配置广播/组播/未知单播风暴抑制功能](#)”。

## 2. 配置以太网接口流量阈值控制功能

表1-18 配置以太网接口流量阈值控制功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
(可选)配置端口流量统计时间间隔	<b>storm-constrain interval <i>interval</i></b>	缺省情况下，端口流量统计时间间隔为10秒 为了保持网络状态的稳定，建议设置的流量统计时间间隔不低于10秒
进入以太网接口视图	<b>interface <i>interface-type</i> <i>interface-number</i></b>	-
开启端口流量阈值控制功能，并设置上限阈值与下限阈值	<b>storm-constrain { broadcast   multicast   unicast } { pps   kbps   ratio } <i>upperlimit lowerlimit</i></b>	缺省情况下，端口流量阈值控制功能处于关闭状态，即端口不进行流量阈值控制
配置端口流量大于上限阈值的控制动作	<b>storm-constrain control { block   shutdown }</b>	缺省情况下，端口不进行流量阈值控制
配置端口流量大于上限阈值或者小于下限阈值时输出Log信息	<b>storm-constrain enable log</b>	缺省情况下，端口流量大于上限阈值或者小于下限阈值时输出Log信息
配置端口流量大于上限阈值或者小于下限阈值时输出Trap信息	<b>storm-constrain enable trap</b>	缺省情况下，端口流量大于上限阈值或者小于下限阈值时输出Trap信息

### 1.4.3 强制开启光口



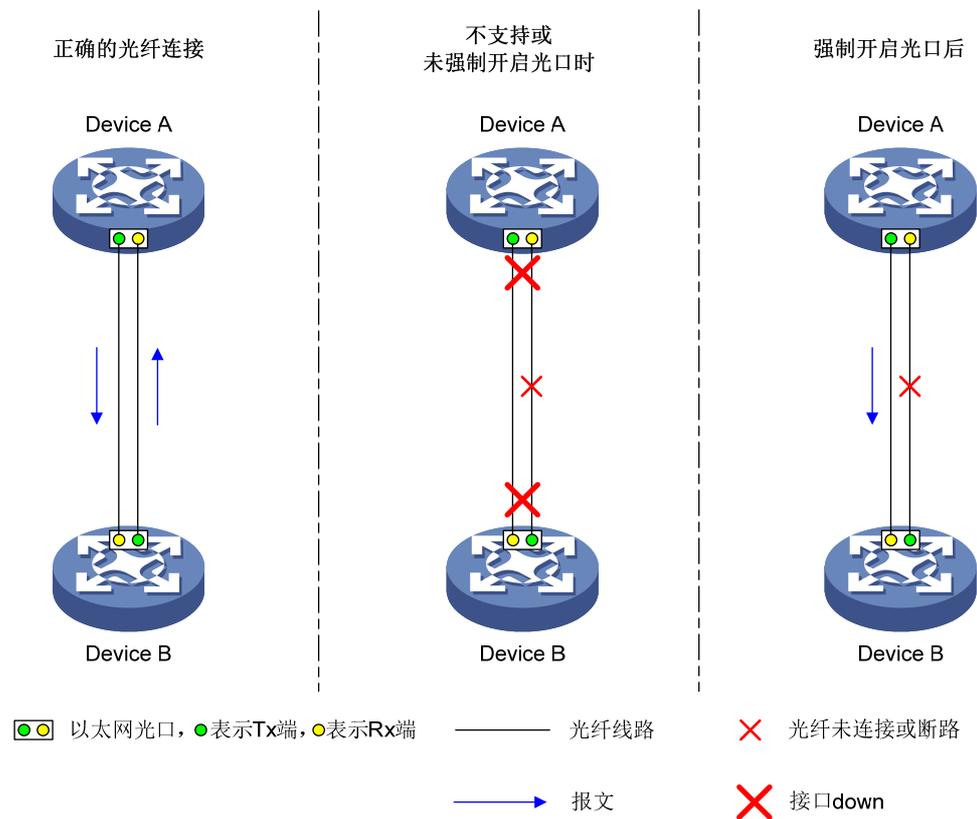
说明

电口不支持本特性。

#### 1. 简介

通常情况下，光口传输报文时要求插入两条光纤：一条用于接收报文，一条用于发送报文。只有两条光纤物理上均连通时，光口的物理状态才会变为up，才能传输报文。使用本特性强制开启光口后，不管实际的光纤链路是否连通，甚至没有插入光纤或光模块，光口的物理状态都会变为up。此时，只要光口上有一条光纤链路是连通的，就可以实现报文的单向转发，以达到节约传输链路的效果。如 [图 1-3](#) 所示。

图1-3 强制开启光口功能示意图



## 2. 配置限制和指导

- **shutdown、port up-mode** 命令互斥，不能同时配置的失败。
- 光口被强制开启后，光口的物理状态始终为 up，不受光纤/光模块拔插的影响。
- **port up-mode** 和 **speed、duplex** 命令同时配置，及光口被强制开启后拔插光纤/光模块的这两种情况都会使接口在 DOWN/UP 状态切换后再处于 UP 状态，请配置时做好相关准备。
- 光口被强制开启后，如果 GE 光口插入光电转换模块、100/1000M 光模块、100M 光模块，则流量不能正常转发。必须取消强制开启光口配置，才能正常转发。

## 3. 配置步骤

表1-19 强制开启光口

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入以太网接口视图	<b>interface interface-type interface-number</b>	-
强制开启光口	<b>port up-mode</b>	缺省情况下，没有强制开启光口。光口的物理状态由光纤的物理状态决定

## 1.4.4 配置以太网接口的MDIX模式



说明

光口不支持本特性。

物理以太网接口由 8 个引脚组成。缺省情况下，每个引脚都有专门的作用，例如，使用引脚 1 和 2 接收信号，引脚 3 和 6 发送信号。为了配合以太网接口支持使用直通线缆和交叉线缆，设备实现了三种 MDIX（Media-dependent Interface-crossover）模式：**automdix**、**mdi** 和 **mdix**。通过配置以太网接口的 MDIX 模式，可以改变引脚在通信中的作用：

- 当配置为 **mdix** 模式时，使用引脚 1 和 2 接收信号，使用引脚 3 和 6 发送信号；
- 当配置为 **mdi** 模式时，使用引脚 1 和 2 发送信号，使用引脚 3 和 6 接收信号；
- 当配置为 **automdix** 模式时，两端设备通过协商来决定引脚 1 和 2 是发送还是接收信号，引脚 3 和 6 是接收还是发送信号。



说明

物理以太网接口的引脚 4、5、7、8 不受该特性限制。

- 十兆和百兆速率接口，引脚 4、5、7、8 不收发信号。
- 千兆速率及以上接口，引脚 4、5、7、8 用来收发信号。

只有将设备的发送引脚连接到对端的接收引脚后才能正常通信，所以 MDIX 模式需要和两种线缆配合使用。

- 通常情况下，建议用户使用 **automdix** 模式。只有当设备不能获取网线类型参数时，才需要将模式手工指定为 **mdi** 或 **mdix**。
- 当使用直通线缆时，两端设备的 MDIX 模式配置不能相同。
- 当使用交叉线缆时，两端设备的 MDIX 模式配置必须相同或者至少有一端设置为 **automdix** 模式。

表1-20 配置以太网接口的 MDIX 模式

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
设置以太网接口的 MDIX 模式	<b>mdix-mode</b> { <b>automdix</b>   <b>mdi</b>   <b>mdix</b> }	缺省情况下，以太网接口的 MDIX 模式为 <b>automdix</b>

## 1.4.5 检测以太网接口的连接电缆



说明

光口不支持本特性。

通过以下配置任务，用户可以检测设备上以太网接口连接电缆的当前状况，系统将在 5 秒内返回检测结果。检测内容包括电缆的状态以及一些物理参数，同时可以检测出故障线缆的长度。

表1-21 检测以太网接口的连接电缆

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
对以太网接口连接电缆进行一次检测	<b>virtual-cable-test</b>	在以太网接口上执行该操作会使得已经 up 的链路自动 down、up 一次

## 1.4.6 配置以太网桥功能

缺省情况下，设备收到报文后会根据报文特征查找报文出接口，如果该报文出接口和入接口为同一接口，则将报文丢弃。在二层以太网接口上开启本功能后，如果该报文出接口和入接口为同一接口，则从该接口转发报文。

表1-22 配置以太网接口桥功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
配置以太网接口桥功能	<b>port bridge enable</b>	缺省情况下，以太网接口的桥功能处于关闭状态

## 1.5 三层以太网接口的配置

### 1.5.1 配置以太网接口的MTU

修改以太网接口的 MTU（Maximum Transmission Unit，最大传输单元）值，会影响 IP 报文的分片与重组。一般情况下，不需要改变 MTU 值。

表1-23 配置以太网接口的 MTU

操作	命令	说明
进入系统视图	<b>system-view</b>	-

操作	命令	说明
进入以太网接口视图	<b>interface</b> <i>interface-type interface-number</i>	-
设置MTU	<b>mtu</b> <i>size</i>	缺省情况下，以太网接口的MTU为1500字节

## 1.6 以太网接口显示和维护

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接口的运行情况，通过查看显示信息验证配置的效果。

在用户视图下执行 **reset** 命令可以清除接口统计信息。

表1-24 以太网接口显示和维护

操作	命令
显示接口的流量统计信息	<b>display counters</b> { <b>inbound</b>   <b>outbound</b> } <b>interface</b> [ <i>interface-type</i> [ <i>interface-number</i> ] ]
显示最近一个抽样间隔内处于up状态的接口的报文速率统计信息	<b>display counters rate</b> { <b>inbound</b>   <b>outbound</b> } <b>interface</b> [ <i>interface-type</i> [ <i>interface-number</i> ] ]
显示接口的运行状态和相关信息	<b>display interface</b> [ <i>interface-type</i> [ <i>interface-number</i> ] ] [ <b>brief</b> [ <b>description</b>   <b>down</b> ] ]
显示接口流量控制信息	<b>display storm-constrain</b> [ <b>broadcast</b>   <b>multicast</b>   <b>unicast</b> ] [ <b>interface</b> <i>interface-type interface-number</i> ]
显示以太网软件模块收发报文的统计信息	<b>display ethernet statistics slot</b> <i>slot-number</i>
清除接口的统计信息	<b>reset counters interface</b> [ <i>interface-type</i> [ <i>interface-number</i> ] ]
清除接口丢弃报文的统计信息	<b>reset packet-drop interface</b> [ <i>interface-type</i> [ <i>interface-number</i> ] ]
清除以太网软件模块收发报文的统计信息	<b>reset ethernet statistics</b> [ <b>slot</b> <i>slot-number</i> ]

# 目 录

1 LoopBack接口、NULL接口和InLoopBack接口 .....	1-1
1.1 LoopBack接口 .....	1-1
1.1.1 LoopBack接口简介 .....	1-1
1.1.2 配置LoopBack接口 .....	1-1
1.2 NULL接口 .....	1-2
1.2.1 NULL接口简介 .....	1-2
1.2.2 配置NULL接口 .....	1-2
1.3 InLoopBack接口 .....	1-2
1.4 LoopBack接口、NULL接口和InLoopBack接口显示和维护 .....	1-2

# 1 LoopBack接口、NULL接口和InLoopBack接口

## 1.1 LoopBack接口

### 1.1.1 LoopBack接口简介

LoopBack 接口是一种虚拟接口。LoopBack 接口创建后，除非手工关闭该接口，否则其物理层永远处于 up 状态。鉴于这个特点，LoopBack 接口的应用非常广泛，主要表现在：

- 该接口的地址常被配置为设备产生的 IP 报文的源地址。因为 LoopBack 接口地址稳定且是单播地址，所以通常将 LoopBack 接口地址视为设备的标志。在认证或安全等服务器上设置允许或禁止携带 LoopBack 接口地址的报文通过，就相当于允许或禁止某台设备产生的报文通过，这样可以简化报文过滤规则。但需要注意的是，将 LoopBack 接口地址用于 IP 报文源地址时，需借助路由配置来确保 LoopBack 接口到对端的路由可达。另外，任何送到 LoopBack 接口的 IP 报文都会被认为是送往设备本身的，设备将不再转发这些报文。
- 该接口常用于动态路由协议。比如：在一些动态路由协议中，当没有配置 Router ID 时，将选取所有 LoopBack 接口上数值最大的 IP 地址作为 Router ID；在 BGP 协议中，为了使 BGP 会话不受物理接口故障的影响，可将发送 BGP 报文的源接口配置成 LoopBack 接口。

### 1.1.2 配置LoopBack接口

表1-1 配置 LoopBack 接口

操作	命令	说明
进入系统视图	<b>system-view</b>	-
创建LoopBack接口并进入LoopBack接口视图	<b>interface loopback</b> <i>interface-number</i>	-
配置接口描述信息	<b>description</b> <i>text</i>	缺省情况下，接口描述信息为“接口名 Interface”，比如：LoopBack1 Interface
配置接口的期望带宽	<b>bandwidth</b> <i>bandwidth-value</i>	缺省情况下，LoopBack接口的期望带宽为0kbps
恢复接口的缺省配置	<b>default</b>	-
开启LoopBack接口	<b>undo shutdown</b>	缺省情况下，LoopBack接口创建后永远处于开启状态

## 1.2 NULL接口

### 1.2.1 NULL接口简介

NULL 接口是一种虚拟接口。它永远处于 up 状态，但不能转发报文，也不能配置 IP 地址和链路层协议。Null 接口为设备提供了一种过滤报文的简单方法——将不需要的网络流量发送到 NULL 接口，从而免去配置 ACL 的复杂工作。比如，在路由中指定到达某一网段的下一跳为 NULL 接口，则任何送到该网段的网络数据报文都会被丢弃。

### 1.2.2 配置NULL接口

表1-2 配置 NULL 接口

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入NULL接口视图	<b>interface null 0</b>	缺省情况下，设备上已经存在NULL0接口，用户不能创建也不能删除 设备只支持NULL0接口，因此，NULL接口的编号只能是0
配置接口描述信息	<b>description text</b>	缺省情况下，接口描述信息为NULL0 Interface
恢复接口的缺省配置	<b>default</b>	-

## 1.3 InLoopBack接口

InLoopBack 接口是一种虚拟接口。InLoopBack 接口由系统自动创建，用户不能进行配置和删除，但是可以显示，其物理层和链路层协议永远处于 up 状态。InLoopBack 接口主要用于配合实现报文的的路由和转发，任何送到 InLoopBack 接口的 IP 报文都会被看作是送往设备本身的，设备将不再转发这些报文。

## 1.4 LoopBack接口、NULL接口和InLoopBack接口显示和维护

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接口的运行情况，通过查看显示信息验证配置的效果。

在用户视图下执行 **reset** 命令可以清除接口统计信息。

表1-3 LoopBack 接口和 NULL 接口显示和维护

操作	命令
显示LoopBack接口的相关信息	<b>display interface loopback [ interface-number ] [ brief [ description   down ] ]</b>

操作	命令
显示NULL接口的状态信息	<b>display interface null [ 0 ] [ brief [ description   down ] ]</b>
显示InLoopBack接口的相关信息	<b>display interface inloopback [ 0 ] [ brief [ description   down ] ]</b>
清除LoopBack接口的统计信息	<b>reset counters interface loopback [ <i>interface-number</i> ]</b>
清除NULL接口的统计信息	<b>reset counters interface null [ 0 ]</b>

# 目 录

1 接口批量配置.....	1-1
1.1 接口批量配置.....	1-1
1.2 接口批量配置显示和维护.....	1-2

# 1 接口批量配置

当多个接口需要配置某功能（比如 **shutdown**）时，需要逐个进入接口视图，在每个接口执行一遍命令，比较繁琐。此时，可以使用接口批量配置功能，对接口进行批量配置，节省配置工作量。

## 1.1 接口批量配置

将多个接口进行绑定的时候，有如下要求：

- 设置为接口列表的第一个接口之前，需要确保可以通过 **interface interface-type interface-number** 命令进入该接口视图。
- 聚合口加入批量接口时，建议不要将该聚合口的成员接口也加入，否则在批量接口配置视图下执行某些配置命令时，可能会导致聚合分裂。
- 批量接口包含的接口数量没有上限，仅受系统资源限制。接口数量较多时，在批量接口配置视图下执行命令等待的时间将较长。
- 系统中支持的批量接口别名的个数没有上限，仅受系统资源限制。推荐用户配置 1000 个以下，配置数量过多，可能引起该特性执行效率降低。

在接口批量配置视图下配置时，有如下约定：

- 在接口批量配置视图下，只能执行接口列表中第一个接口支持的命令，不能执行第一个接口不支持但其它成员接口支持的命令。（接口列表中的第一个接口指的是执行 **interface range** 命令时指定的第一个接口）。在接口批量配置视图下，输入问号并回车，将显示该视图下支持的所有命令。
- 在接口批量配置视图下执行命令，会在绑定的所有接口下执行该命令。出现以下情况时请注意：
  - 当命令执行完成后，系统提示配置失败并保持在接口批量配置视图，如果配置失败的接口是接口列表的第一个接口，则表示列表中的所有接口都未配置该命令；如果配置失败的接口是其它接口，则表示除了提示失败的接口外，其它接口都已经配置成功。
  - 如果命令执行完成后，退回到系统视图，则表示这条命令在接口视图和系统视图下都支持，并且在列表中的某个接口上配置失败，在系统视图下配置成功，列表中位于这个接口后面的接口不再执行该命令。此时，可到列表中各接口的视图下使用 **display this** 命令验证配置效果，同时如果不需要在系统视图下配置该命令的话，请使用相应的 **undo** 命令取消该配置。
- 在接口批量配置视图下，执行 **display this** 命令，将显示接口列表中第一个接口当前生效的配置。

表1-1 接口批量配置

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入接口批量配置视图	<b>interface range</b> { <i>interface-type interface-number</i> [ <i>to interface-type interface-number</i> ] } <1-24>	二者选其一 <b>interface range name</b> 和 <b>interface range</b> 命

操作	命令	说明
	<b>interface range name name [ interface { interface-type interface-number [ to interface-type interface-number ] } &amp;&lt;1-24&gt; ]</b>	令都能提供接口批量配置功能，它们的差别在于： <b>interface range name</b> 命令在绑定接口的时候可以定义一个别名，可以进行多次绑定，给不同的绑定定义不同的别名，以示区别，方便记忆。并且，后续可以使用别名直接进入接口批量配置视图，不再需要重新输入接口列表，配置起来更简便

## 1.2 接口批量配置显示和维护

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批量接口的信息。

表1-2 接口批量配置显示和维护

操作	命令
显示通过 <b>interface range name</b> 命令创建的批量接口的信息	<b>display interface range [ name name ]</b>

# 目 录

<b>1 MAC地址表</b> .....	<b>1-1</b>
1.1 MAC地址表简介 .....	1-1
1.1.1 MAC地址表项的生成方式 .....	1-1
1.1.2 MAC地址表项的分类 .....	1-1
1.2 配置MAC地址表 .....	1-2
1.2.1 配置MAC地址表项 .....	1-2
1.2.2 关闭MAC地址学习功能 .....	1-5
1.2.3 配置动态MAC地址表项的老化时间 .....	1-6
1.2.4 配置MAC地址数学习上限 .....	1-6
1.2.5 配置当达到MAC地址数学习上限时的报文转发规则 .....	1-7
1.2.6 配置接口的MAC地址学习优先级 .....	1-7
1.2.7 开启MAC地址同步功能 .....	1-7
1.2.8 配置MAC地址迁移上报功能 .....	1-9
1.2.9 配置快速更新ARP表项功能 .....	1-10
1.2.10 开启MAC地址表告警功能 .....	1-11
1.3 MAC地址表显示和维护 .....	1-12
1.4 MAC地址表典型配置举例 .....	1-12
<b>2 MAC Information</b> .....	<b>2-1</b>
2.1 MAC Information简介 .....	2-1
2.2 配置MAC Information功能 .....	2-1
2.2.1 开启MAC Information功能 .....	2-1
2.2.2 配置发送MAC变化通知的方式 .....	2-1
2.2.3 配置发送MAC变化通知的时间间隔 .....	2-2
2.2.4 配置MAC Information缓存队列长度 .....	2-2
2.3 MAC Information典型配置举例 .....	2-2

# 1 MAC地址表

## 1.1 MAC地址表简介

MAC（Media Access Control，媒体访问控制）地址表记录了 MAC 地址与接口的对应关系，以及接口所属的 VLAN 等信息。设备在转发报文时，根据报文的目的地 MAC 地址查询 MAC 地址表，如果 MAC 地址表中包含与报文目的地 MAC 地址对应的表项，则直接通过该表项中的出接口转发该报文；如果 MAC 地址表中没有包含报文目的地 MAC 地址对应的表项时，设备将采取广播方式通过对应 VLAN 内除接收接口外的所有接口转发该报文。

### 1.1.1 MAC地址表项的生成方式

MAC 地址表项的生成方式有两种：自动生成、手工配置。

#### 1. 自动生成MAC地址表项

一般情况下，MAC 地址表由设备通过源 MAC 地址学习自动生成。设备学习 MAC 地址的过程如下：

- 从某接口（假设为接口 A）收到一个数据帧，设备分析该数据帧的源 MAC 地址（假设为 MAC-SOURCE），并认为目的 MAC 地址为 MAC-SOURCE 的报文可以由接口 A 转发。
- 如果 MAC 地址表中已经包含 MAC-SOURCE，设备将对该表项进行更新。
- 如果 MAC 地址表中尚未包含 MAC-SOURCE，设备则将这个新 MAC 地址以及该 MAC 地址对应的接口 A 作为一个新的表项加入到 MAC 地址表中。

为适应网络拓扑的变化，MAC 地址表需要不断更新。MAC 地址表中自动生成的表项并非永远有效，每一条表项都有一个生存周期，到达生存周期仍得不到刷新的表项将被删除，这个生存周期被称作老化时间。如果在到达生存周期前某表项被刷新，则重新计算该表项的老化时间。

#### 2. 手工配置MAC地址表项

设备通过源 MAC 地址学习自动生成 MAC 地址表时，无法区分合法用户和非法用户的报文，带来了安全隐患。如果非法用户将攻击报文的源 MAC 地址伪装成合法用户的 MAC 地址，并从设备的其他接口进入，设备就会学习到错误的 MAC 地址表项，于是将本应转发给合法用户的报文转发给非法用户。

为了提高安全性，网络管理员可手工在 MAC 地址表中加入特定 MAC 地址表项，将用户设备与接口绑定，从而防止非法用户骗取数据。

### 1.1.2 MAC地址表项的分类

MAC 地址表项分为以下几种：

- 静态 MAC 地址表项：由用户手工配置，用于目的是某个 MAC 地址的报文从对应接口转发出去，表项不老化。静态 MAC 地址表项优先级高于自动生成的 MAC 地址表项。
- 动态 MAC 地址表项：可以由用户手工配置，也可以由设备通过源 MAC 地址学习自动生成，用于目的是某个 MAC 地址的报文从对应接口转发出去，表项有老化时间。手工配置的动态 MAC 地址表项优先级等于自动生成的 MAC 地址表项。

- 黑洞 MAC 地址表项：由用户手工配置，用于丢弃源 MAC 地址或目的 MAC 地址为指定 MAC 地址的报文（例如，出于安全考虑，可以禁止某个用户发送和接收报文），表项不老化。黑洞 MAC 地址表项优先级高于自动生成的 MAC 地址表项。
- 多端口单播 MAC 地址表项：由用户手工配置，用于目的是某个单播 MAC 地址的报文从多个接口复制转发出去，表项不老化。多端口单播 MAC 地址表项优先级高于自动生成的 MAC 地址表项。

静态 MAC 地址表项、黑洞 MAC 地址表项和多端口单播 MAC 地址表项不会被动态 MAC 地址表项覆盖，而动态 MAC 地址表项可以被静态 MAC 地址表项和黑洞 MAC 地址表项覆盖。静态 MAC 地址表项、黑洞 MAC 地址表项和多端口单播 MAC 地址表项不会彼此覆盖。

多端口单播 MAC 地址表项不影响对应 MAC 地址的动态学习，对于同一 MAC 地址，多端口单播 MAC 地址表项和动态 MAC 地址表项可以同时存在，优先根据多端口单播 MAC 地址转发报文。

## 1.2 配置MAC地址表

以下配置均为可选配置，且配置过程无先后顺序，用户可以根据实际情况选择配置。

### 1.2.1 配置MAC地址表项

配置 MAC 地址表项时，需要注意：

- 在手工配置动态 MAC 地址表项时，如果 MAC 地址表中已经存在 MAC 地址相匹配的自动生成表项，但该表项的接口与配置不符，那么该手工配置失败。
- 如果不保存配置，设备重启后所有手工配置的 MAC 地址表项都会丢失；如果保存配置，设备重启后手工配置的静态 MAC 地址表项、黑洞 MAC 地址表项和多端口单播 MAC 地址表项不会丢失，手工配置动态 MAC 地址表项会丢失。

配置 MAC 地址表项后，当设备收到的报文的源 MAC 地址与配置表项中的 MAC 地址相同时，不同类型的 MAC 地址表项处理方式不同：

表1-1 不同类型 MAC 地址表项对源 MAC 地址匹配报文的处理方式

MAC 地址表项类型	报文源 MAC 地址与配置表项中的 MAC 地址相同
静态MAC地址表项	不检查报文入接口与表项中的接口是否相同，直接根据目的MAC地址转发该报文
多端口单播MAC地址表项	进行MAC地址学习，生成动态MAC地址表项（假设源MAC为MAC A），并转发该报文。但是，当设备转发目的MAC为MAC A的报文时，由于多端口单播MAC地址表项优先级高于自动生成的MAC地址表项，报文根据多端口单播MAC地址表项转发，不根据生成的动态MAC地址表项转发
黑洞MAC地址表项	丢弃该报文
动态MAC地址表项	<ul style="list-style-type: none"> <li>• 如果报文入接口与该表项中的接口不同，则进行 MAC 地址学习，并覆盖该表项</li> <li>• 如果报文入接口与该表项中的接口相同，则转发该报文，并更新该表项老化时间</li> </ul>

### 1. 配置静态/动态MAC地址表项

#### (1) 全局配置静态/动态 MAC 地址表项

表1-2 全局配置静态/动态 MAC 地址表项

操作	命令	说明
进入系统视图	<b>system-view</b>	-
添加或者修改静态/动态MAC地址表项	<b>mac-address { dynamic   static } mac-address interface interface-type interface-number vlan vlan-id</b>	缺省情况下，未配置任何MAC地址表项 <b>interface</b> 参数指定的接口必须属于 <b>vlan-id</b> 参数指定的VLAN，而且该VLAN必须事先创建，否则将配置失败

(2) 接口配置静态/动态 MAC 地址表项

表1-3 接口配置静态/动态 MAC 地址表项

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入接口视图	二层以太网接口视图 <b>interface interface-type interface-number</b>	-
	二层聚合接口视图 <b>interface bridge-aggregation interface-number</b>	
在接口下添加或者修改静态/动态MAC地址表项	<b>mac-address { dynamic   static } mac-address vlan vlan-id</b>	缺省情况下，接口下未配置任何MAC地址表项 当前接口必须属于 <b>vlan-id</b> 参数指定的VLAN，而且该VLAN必须事先创建，否则将配置失败

2. 配置黑洞MAC地址表项

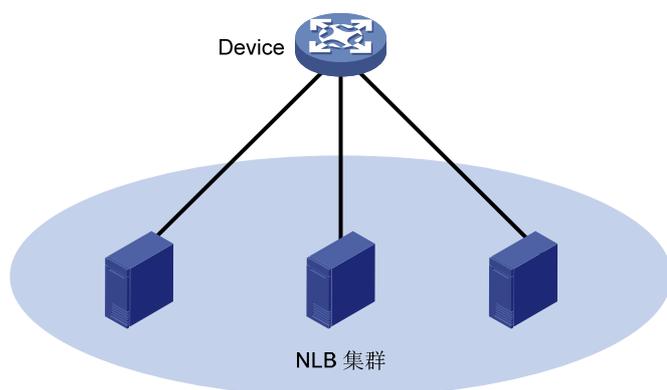
表1-4 配置黑洞 MAC 地址表项

操作	命令	说明
进入系统视图	<b>system-view</b>	-
添加或者修改黑洞MAC地址表项	<b>mac-address blackhole mac-address vlan vlan-id</b>	缺省情况下，未配置任何MAC地址表项 <b>vlan-id</b> 参数指定的VLAN必须事先创建，否则将配置失败

3. 配置多端口单播MAC地址表项

网络管理员可手工配置多端口单播MAC地址表项，将多个端口和单播MAC地址绑定，以实现目的地址匹配该MAC地址的报文通过多个端口复制转发出去。例如，如 [图 1-1](#) 在NLB（Network Load Balancing，网络负载均衡）集群的单播模式下，所有服务器使用一个共同的MAC地址（该MAC地址为集群MAC地址），发往集群MAC地址的报文要求发送到每一台服务器，这时可以在连接服务器组的设备上配置多端口单播MAC地址表项，把客户端发往服务器组的报文从所有连接服务器的端口转发出去。

图1-1 NLB 集群



有两种方式可以配置多端口单播 MAC 地址表项：

(1) 全局配置多端口单播 MAC 地址表项

表1-5 全局配置多端口单播 MAC 地址表项

操作	命令	说明
进入系统视图	<b>system-view</b>	-
配置多端口单播MAC地址表项	<b>mac-address multiport mac-address interface interface-list vlan vlan-id</b>	缺省情况下，未配置任何多端口单播MAC地址表项 <b>interface</b> 参数指定的接口必须属于 <i>vlan-id</i> 参数指定的VLAN，而且该VLAN必须事先创建，否则将配置失败

(2) 接口配置多端口单播 MAC 地址表项

表1-6 接口配置多端口单播 MAC 地址表项

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入接口视图	二层以太网接口视图 <b>interface interface-type interface-number</b>	-
	二层聚合接口视图 <b>interface bridge-aggregation interface-number</b>	
配置接口加入多端口单播MAC地址表项	<b>mac-address multiport mac-address vlan vlan-id</b>	缺省情况下，接口下未配置任何多端口单播MAC地址表项 当前接口必须属于 <i>vlan-id</i> 参数指定的VLAN，而且该VLAN必须事先创建，否则将配置失败

## 1.2.2 关闭MAC地址学习功能

缺省情况下，MAC 地址学习功能处于开启状态。有时为了保证设备的安全，需要关闭 MAC 地址学习功能。常见的危及设备安全的情况是：非法用户使用大量源 MAC 地址不同的报文攻击设备，导致设备 MAC 地址表资源耗尽，造成设备无法根据网络的变化更新 MAC 地址表。关闭 MAC 地址学习功能可以有效防止这种攻击。

关闭 MAC 地址学习功能后，设备立即删除已经存在的动态 MAC 地址表项。

### 1. 关闭全局的MAC地址学习功能

关闭全局的 MAC 地址学习功能后，接口将不再学习新的 MAC 地址。

全局 MAC 地址学习功能不能控制 VXLAN 的 VSI 中 MAC 地址的学习。有关 VXLAN 和 VSI 的介绍，请参见“VXLAN 配置指导”中的“VXLAN”。

表1-7 关闭全局 MAC 地址学习功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
关闭全局的MAC地址学习功能	<b>undo mac-address mac-learning enable</b>	缺省情况下，全局的MAC地址学习功能处于开启状态

### 2. 关闭接口的MAC地址学习功能

在开启全局的 MAC 地址学习功能的前提下，用户可以关闭设备上单个接口的 MAC 地址学习功能。

表1-8 关闭接口的 MAC 地址学习功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入接口视图	二层以太网接口视图 <b>interface interface-type interface-number</b>	-
	二层聚合接口视图 <b>interface bridge-aggregation interface-number</b>	
关闭接口的MAC地址学习功能	<b>undo mac-address mac-learning enable</b>	缺省情况下，接口的MAC地址学习功能处于开启状态

### 3. 关闭VLAN的MAC地址学习功能

在开启全局的 MAC 地址学习功能的前提下，用户可以关闭设备上指定 VLAN 的 MAC 地址学习功能。

表1-9 关闭 VLAN 的 MAC 地址学习功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入VLAN视图	<b>vlan vlan-id</b>	-
关闭VLAN的MAC地址学习功能	<b>undo mac-address mac-learning enable</b>	缺省情况下，VLAN的MAC地址学习功能处于开启状态

### 1.2.3 配置动态MAC地址表项的老化时间

当网络拓扑改变后，如果动态 MAC 地址表项不及时更新，会导致用户流量不能正常转发。配置动态 MAC 地址表项的老化时间后，超过老化时间的动态 MAC 地址表项会被自动删除，设备将重新进行 MAC 地址学习，构建新的动态 MAC 地址表项。

用户配置的老化时间过长或者过短，都可能影响设备的运行性能：

- 如果用户配置的老化时间过长，设备可能会保存许多过时的 MAC 地址表项，从而耗尽 MAC 地址表资源，导致设备无法根据网络的变化更新 MAC 地址表。
- 如果用户配置的老化时间太短，设备可能会删除有效的 MAC 地址表项，导致设备广播大量的数据报文，增加网络的负担。

用户需要根据实际情况，配置合适的老化时间。如果网络比较稳定，可以将老化时间配置得长一些或者配置为不老化；否则，可以将老化时间配置得短一些。比如在一个比较稳定的网络，如果长时间没有流量，动态 MAC 地址表项会被全部删除，可能导致设备突然广播大量的数据报文，造成安全隐患，此时可将动态 MAC 地址表项的老化时间设得长一些或不老化，以减少广播，增加网络稳定性和安全性。

动态 MAC 地址表项的老化时间作用于全部接口上。

表1-10 配置动态 MAC 地址表项的老化时间

操作	命令	说明
进入系统视图	<b>system-view</b>	-
配置动态MAC地址表项的老化时间	<b>mac-address timer { aging seconds   no-aging }</b>	缺省情况下，动态MAC地址表项的老化时间为300秒

### 1.2.4 配置MAC地址数学习上限

通过配置接口的 MAC 地址数学习上限，用户可以控制设备维护的 MAC 地址表的表项数量。如果 MAC 地址表过于庞大，可能导致设备的转发性能下降。当接口学习到的 MAC 地址数达到上限时，该接口将不再对 MAC 地址进行学习。

表1-11 配置接口的 MAC 地址数学习上限

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二层以太网接口视图	<b>interface interface-type interface-number</b>	-
配置接口的MAC地址数学习上限	<b>mac-address max-mac-count count</b>	缺省情况下，接口未配置MAC地址数学习上限

## 1.2.5 配置当达到MAC地址数学习上限时的报文转发规则

当学习到的 MAC 地址数达到上限时，用户可以选择是否允许系统转发源 MAC 不在 MAC 地址表里的报文。

表1-12 配置当达到接口的 MAC 地址数学习上限时的报文转发规则

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二层以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
配置当达到接口的MAC地址数学习上限时，允许转发源MAC地址不在MAC地址表里的报文	<b>mac-address max-mac-count</b> <b>enable-forwarding</b>	缺省情况下，当达到接口的MAC地址数学习上限时，允许转发源MAC地址不在MAC地址表里的报文

## 1.2.6 配置接口的MAC地址学习优先级

基于 MAC 地址转发报文的网络有时会因为下行接口的攻击行为或者环路，下行接口学习到网关等上层设备的 MAC 地址。为了避免这种情况，将接口的 MAC 地址学习功能分为两个优先级：高优先级和低优先级。对于高优先级的接口，可以学习任何 MAC 地址；对于低优先级的接口，在学习 MAC 地址时需要查看高优先级接口是否已经学到该 MAC 地址，如果已经学到，则不允许学习该 MAC 地址。比如，可以将上行接口的 MAC 地址学习优先级配置为高优先级，下行接口的 MAC 地址学习优先级配置为低优先级，那么，下行接口就不会学到网关等上层设备的 MAC 地址，避免了攻击。

表1-13 配置接口的 MAC 地址学习优先级

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入接口视图	二层以太网接口视图 <b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
	二层聚合接口视图 <b>interface bridge-aggregation</b> <i>interface-number</i>	
配置接口的MAC地址学习优先级	<b>mac-address mac-learning priority</b> { <b>high</b>   <b>low</b> }	缺省情况下，MAC地址学习优先级为低优先级

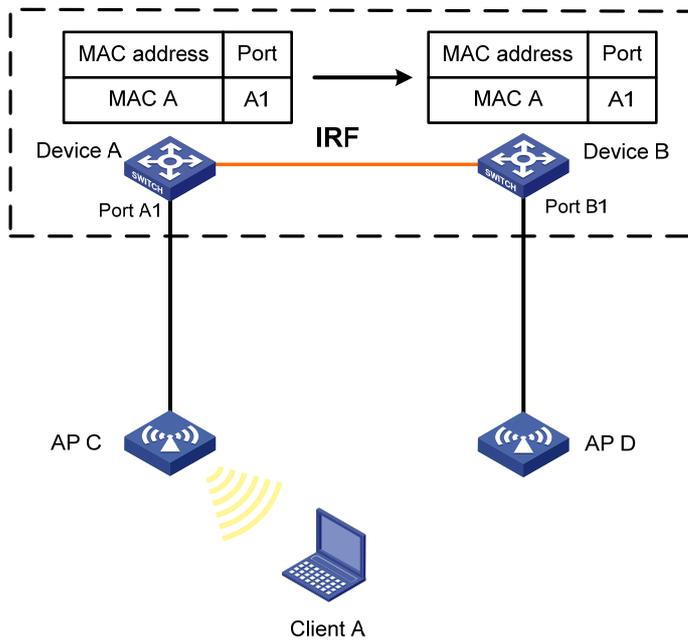
## 1.2.7 开启MAC地址同步功能

IRF 设备通常有多个成员设备，为了避免不必要的广播报文，以及提高报文转发的速度，需要所有成员设备拥有同样的 MAC 地址表。开启全局的 MAC 地址同步功能后，设备会在所有成员设备间进行 MAC 地址表的同步。

如 图 1-2 所示，是MAC地址同步功能的典型应用场景。Device A和Device B是两台配置了IRF功能的设备。无线接入点AP C和AP D分别连接到IRF成员设备Device A和Device B。

开启MAC地址同步功能后，IRF成员设备会将学习到的MAC地址同步给IRF设备内的其他成员设备。如 [图 1-2](#) 所示，当Client A通过AP C接入时，Device A会将学习到的Client A的MAC地址同步给IRF设备内的其他成员设备Device B。

图1-2 Client A 通过 AP C 接入时的 MAC 地址表



当用户的接入地点发生变化，例如从AP C的覆盖区域移动到AP D的覆盖区域时，IRF会将Client A的MAC地址重新学习到Device B上，并将更新后的MAC地址同步给IRF设备内的其他成员设备Device A（如 [图 1-3](#) 所示），使用户的通信不受任何影响。

图1-3 Client A 移动到通过 AP D 接入时的 MAC 地址表

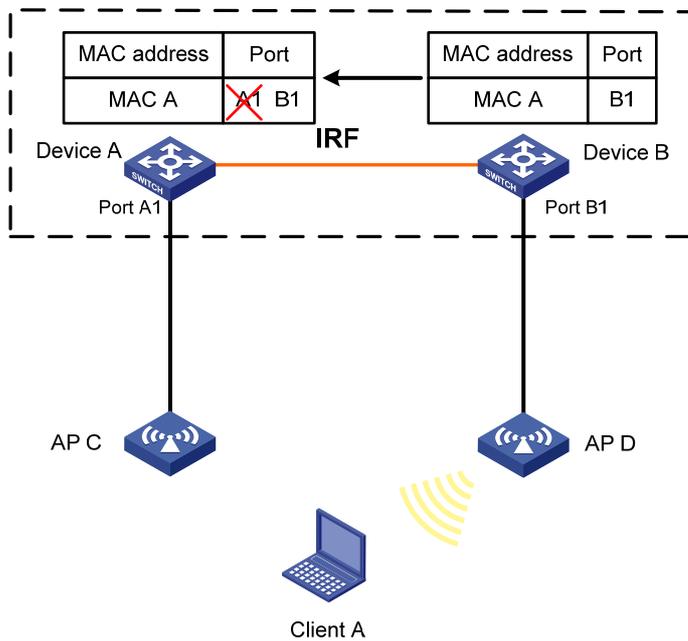


表1-14 开启 MAC 地址同步功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
开启全局的MAC地址同步功能	<b>mac-address mac-roaming enable</b>	缺省情况下，全局的MAC地址同步功能处于关闭状态

### 1.2.8 配置MAC地址迁移上报功能

MAC 地址迁移是指：设备从某接口（假设接口 A）学习到某 MAC 地址，之后从另一接口（假设接口 B）接收到了以该 MAC 地址为源 MAC 地址的报文，且接口 B 与接口 A 所属的 VLAN 相同，则该 MAC 地址表项的出接口改为接口 B，此时认为该 MAC 地址从接口 A 迁移到接口 B。

如果 MAC 地址迁移频繁出现，且同一 MAC 地址总是在特定的两个接口之间迁移，那么网络中可能存在二层环路。可以通过查看 MAC 地址迁移记录，发现和定位环路。

当监测到某端口频繁迁移时，用户可以通过配置 MAC 地址迁移抑制功能，使频繁迁移的端口 down，一定时间后该端口将自行恢复 up，或者用户通过手动方式将该端口 up。

如果需要查看设备启动后的 MAC 地址迁移记录，请使用 **display mac-address mac-move** 命令。

表1-15 配置 MAC 地址迁移上报功能

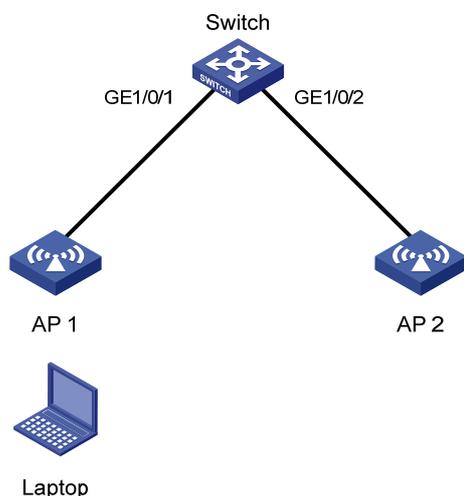
操作	命令	说明
进入系统视图	<b>system-view</b>	-

操作		命令	说明
开启MAC地址迁移上报功能		<b>mac-address notification mac-move</b> [ <i>interval interval</i> ]	缺省情况下，MAC地址迁移上报功能处于关闭状态 需要注意的是，执行本命令后，系统采用Syslog方式上报MAC地址迁移信息到信息中心模块，如果同时通过 <b>snmp-agent trap enable mac-address</b> 命令开启MAC地址表的告警功能，系统还会采用Trap信息上报MAC地址迁移信息到SNMP模块
(可选) 配置MAC地址迁移抑制功能的相关参数		<b>mac-address notification mac-move suppression</b> { <i>interval interval</i>   <i>threshold threshold</i> }	MAC地址迁移抑制功能的相关参数未配置，采用缺省抑制时间间隔30秒和缺省阈值3次 配置本命令后，当接口上开启了MAC地址迁移抑制功能时，本命令配置的参数才能生效
进入接口视图	二层以太网接口视图	<b>interface</b> <i>interface-type interface-number</i>	-
	二层聚合接口视图	<b>interface bridge-aggregation</b> <i>interface-number</i>	
(可选) 开启接口上的MAC地址迁移抑制功能		<b>mac-address notification mac-move suppression</b>	缺省情况下，MAC地址迁移抑制功能处于关闭状态
退回系统视图		<b>quit</b>	-
开启在MAC地址迁移后，快速更新ARP表项功能		<b>mac-address mac-move fast-update</b>	配置MAC地址迁移抑制功能时必须开启本功能 缺省情况下，在MAC地址迁移后，快速更新ARP表项功能处于关闭状态

## 1.2.9 配置快速更新ARP表项功能

如 [图 1-4](#) 所示，Laptop经常在无线站点AP 1 和AP 2 之间漫游，导致Switch上记录的Laptop的MAC地址与出端口的对应关系经常发生改变，但是Switch上的ARP表项不会立即更新，影响到数据业务的正常转发。

图1-4 MAC 地址迁移后 ARP 表项不能更新



配置快速更新 ARP 表项后，如果交换机上记录的 MAC 地址与出端口的对应关系发生改变，系统会立刻更新 ARP 表项，保证了数据业务的不间断转发。

表1-16 配置快速更新 ARP 表项功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
开启在MAC地址迁移后，快更新 ARP表项功能	<b>mac-address mac-move fast-update</b>	缺省情况下，在MAC地址迁移后，快速更新ARP表项功能处于关闭状态

## 1.2.10 开启MAC地址表告警功能

开启 MAC 地址表的告警功能后，MAC 地址表模块会生成告警信息，用于报告该模块的重要事件。生成的告警信息将发送到设备的 SNMP 模块，请通过设置 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。

关闭 MAC 地址表的告警功能后，设备将只发送日志信息到信息中心模块，此时请配置信息中心的输出规则和输出方向来查看 MAC 地址表模块的日志信息。

有关 SNMP 和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“SNMP”和“信息中心”。

表1-17 开启 MAC 地址表告警功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
开启MAC地址表的告警功能	<b>snmp-agent trap enable mac-address [ mac-move ]</b>	缺省情况下，MAC地址表的告警功能处于开启状态 当MAC地址表的告警功能关闭后，将采用Syslog方式上报信息

## 1.3 MAC地址表显示和维护

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 MAC 地址表的运行情况，通过查看显示信息验证配置的效果。

表1-18 MAC 地址表显示和维护

操作	命令
显示MAC地址表信息	<b>display mac-address</b> [ <i>mac-address</i> [ <b>vlan</b> <i>vlan-id</i> ] ] [ [ <b>dynamic</b>   <b>static</b> ] [ <b>interface</b> <i>interface-type interface-number</i> ]   <b>blackhole</b>   <b>multiport</b> ] [ <b>vlan</b> <i>vlan-id</i> ] [ <b>count</b> ] ]
显示MAC地址表动态表项的老化时间	<b>display mac-address aging-time</b>
显示MAC地址学习功能的开启状态	<b>display mac-address mac-learning</b> [ <b>interface</b> <i>interface-type interface-number</i> ]
显示MAC地址表的统计信息	<b>display mac-address statistics</b>
显示MAC地址迁移记录	<b>display mac-address mac-move</b> [ <b>slot</b> <i>slot-number</i> ]

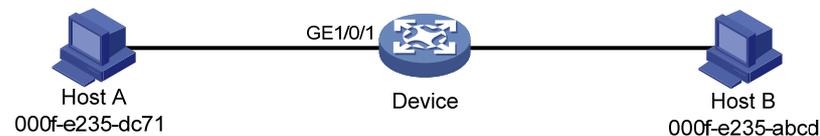
## 1.4 MAC地址表典型配置举例

### 1. 组网需求

- 现有一台用户主机，它的 MAC 地址为 000f-e235-dc71，属于 VLAN 1，连接 Device 的端口 GigabitEthernet1/0/1。为防止假冒身份的非法用户骗取数据，在设备的 MAC 地址表中为该用户主机添加一条静态表项。
- 另有一台用户主机，它的 MAC 地址为 000f-e235-abcd，属于 VLAN 1。由于该用户主机曾经接入网络进行非法操作，为了避免此种情况再次发生，在设备上添加一条黑洞 MAC 地址表项，使用该用户主机接收不到报文。
- 配置设备的动态 MAC 地址表项老化时间为 500 秒。

### 2. 组网图

图1-5 MAC 地址表典型配置组网图



### 3. 配置步骤

# 增加一个静态 MAC 地址表项，目的地址为 000f-e235-dc71，出接口为 GigabitEthernet1/0/1，且该接口属于 VLAN 1。

```
<Device> system-view
```

```
[Device] mac-address static 000f-e235-dc71 interface gigabitethernet 1/0/1 vlan 1
```

# 增加一个黑洞 MAC 地址表项，地址为 000f-e235-abcd，属于 VLAN 1。

```
[Device] mac-address blackhole 000f-e235-abcd vlan 1
```

# 配置动态 MAC 地址表项的老化时间为 500 秒。

```
[Device] mac-address timer aging 500
```

#### 4. 验证配置

# 查看端口 GigabitEthernet1/0/1 上的静态 MAC 地址表项信息。

```
[Device] display mac-address static interface gigabitethernet 1/0/1
```

MAC Address	VLAN ID	State	Port/NickName	Aging
000f-e235-dc71	1	Static	GE1/0/1	N

# 查看黑洞 MAC 地址表信息。

```
[Device] display mac-address blackhole
```

MAC Address	VLAN ID	State	Port/NickName	Aging
000f-e235-abcd	1	Blackhole	N/A	N

# 查看动态 MAC 地址表项的老化时间。

```
[Device] display mac-address aging-time
```

```
MAC address aging time: 500s.
```

# 2 MAC Information

## 2.1 MAC Information简介

由于 MAC 地址能唯一标识一个网络用户，MAC Information 功能通过监控接口学习和删除 MAC 地址表项，可以对用户加入和离开网络进行跟踪。具体机制为：当接口学习到一条新的 MAC 地址表项或删除一条已有 MAC 地址表项时，设备会将该 MAC 地址变化信息写入缓冲队列。当设定的发送 MAC 变化通知的时间间隔到期，设备立即发送记录了 MAC 地址变化信息的日志或 SNMP 告警信息。信息接收端通过对日志或 SNMP 告警信息进行分析，实现对网络中的用户进行监控，同时为分析网络的使用情况提供依据。

## 2.2 配置MAC Information功能

### 2.2.1 开启MAC Information功能

必须同时开启全局和接口的 MAC Information 功能，MAC Information 功能才会生效。

表2-1 开启 MAC Information 功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
开启全局MAC Information功能	<b>mac-address information enable</b>	缺省情况下，全局MAC Information功能处于关闭状态
进入二层以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
开启接口的MAC Information功能	<b>mac-address information enable</b> { <b>added</b>   <b>deleted</b> }	缺省情况下，接口的MAC Information功能处于关闭状态

### 2.2.2 配置发送MAC变化通知的方式

发送 MAC 变化通知的方式有两种：

- **Syslog 方式**：通过发送日志信息通知 MAC 地址的变化。采用该方式时，日志信息会被发送到设备的信息中心，由信息中心发送到监控终端。有关信息中心的详细介绍及相关配置，请参见“网络管理和监控配置指导”中的“信息中心”。
- **Trap 方式**：通过发送 SNMP 告警信息通知 MAC 地址的变化。采用该方式时，需要通过 SNMP 将 SNMP 告警信息发送到 NMS。有关 SNMP 的详细介绍及相关配置，请参见“网络管理和监控配置指导”中的“SNMP”。

表2-2 配置发送 MAC 变化通知的方式

操作	命令	说明
进入系统视图	<b>system-view</b>	-

操作	命令	说明
配置发送MAC变化通知的方式	<b>mac-address information mode</b> { <b>syslog</b>   <b>trap</b> }	缺省情况下，采用Trap方式发送MAC变化通知

### 2.2.3 配置发送MAC变化通知的时间间隔

为了防止过于频繁地发送 MAC 变化通知干扰用户，用户可以修改发送 MAC 变化通知的时间间隔。

表2-3 配置发送 MAC 变化通知的时间间隔

配置步骤	命令	说明
进入系统视图	<b>system-view</b>	-
配置发送MAC变化通知的时间间隔	<b>mac-address information interval</b> <i>interval-time</i>	缺省情况下，发送MAC变化通知的时间间隔为1秒

### 2.2.4 配置MAC Information缓存队列长度

MAC Information 缓存队列长度是否为 0 对应着不同的处理方式：

- 如果 MAC Information 缓存队列长度为 0，则当接口学习到或删除一条 MAC 地址表项时会立即发送日志或 SNMP 告警信息。
- 如果 MAC Information 缓存队列长度不为 0，则将 MAC 地址变化信息存放在缓存队列中。当未达到发送 MAC 变化通知的时间间隔，此时若缓存队列被写满，新的 MAC 地址变化信息将覆盖缓存队列中最后一条写入的信息；当达到发送 MAC 变化通知的时间间隔时，不论此时缓存队列是否已被写满，都发送日志或 SNMP 告警信息。

表2-4 配置 MAC Information 缓存队列长度

配置步骤	命令	说明
进入系统视图	<b>system-view</b>	-
配置MAC Information缓存队列长度	<b>mac-address information queue-length</b> <i>value</i>	缺省情况下，MAC Information缓存队列长度为50

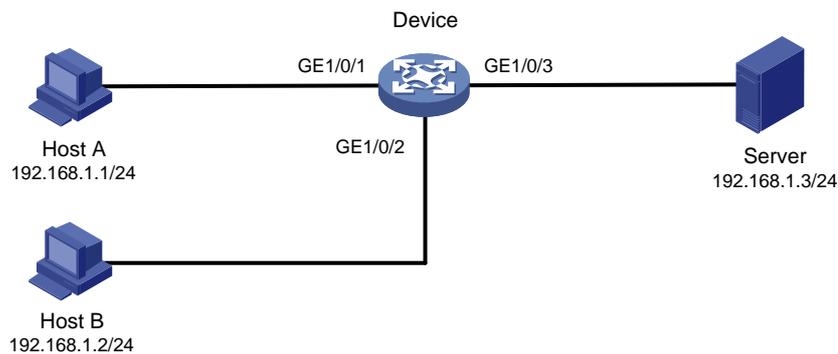
## 2.3 MAC Information典型配置举例

### 1. 组网需求

- Host A 与远端服务器 Server 通过 Device 相连。
- 在 Device 的端口 GigabitEthernet1/0/1 上开启 MAC Information 功能，Device 将端口 GigabitEthernet1/0/1 上的 MAC 地址添加或删除信息利用 Syslog 方式通过端口 GigabitEthernet1/0/2 发送给日志主机 Host B，Host B 可以对接收到的日志信息进行分析。

## 2. 组网图

图2-1 MAC Information 典型配置组网图



## 3. 配置步骤

(1) 配置 Device 可以将日志信息发送到 Host B

- 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 192.168.1.2/24 的日志主机，日志主机记录工具为 local4。

```
[Device] info-center loghost 192.168.1.2 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 MAC 地址表模块的、等级高于等于 informational 的日志信息输出到日志主机。

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

- 日志主机 Host B 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，在其他厂商的 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 Host B，Host B 会将这些日志信息存储到相应的文件中了。

### (2) 配置 MAC Information 功能

# 开启全局 MAC Information 功能。

```
[Device] mac-address information enable
```

# 配置采用 Syslog 方式发送 MAC 变化通知。

```
[Device] mac-address information mode syslog
```

# 开启端口 GigabitEthernet1/0/1 的 MAC Information 功能，使该接口在学习到和删除 MAC 地址时记录 MAC 变化信息。

```
[Device] interface gigabitethernet 1/0/1
```

```
[Device-GigabitEthernet1/0/1] mac-address information enable added
```

```
[Device-GigabitEthernet1/0/1] mac-address information enable deleted
```

```
[Device-GigabitEthernet1/0/1] quit
```

# 配置 MAC Information 缓存队列长度为 100。

```
[Device] mac-address information queue-length 100
```

# 配置发送 MAC 变化通知的时间间隔为 20 秒。

```
[Device] mac-address information interval 20
```

# 目 录

1 以太网链路聚合.....	1-1
1.1 以太网链路聚合简介.....	1-1
1.1.1 基本概念.....	1-1
1.1.2 静态聚合模式.....	1-3
1.1.3 动态聚合模式.....	1-4
1.1.4 聚合边缘接口.....	1-8
1.1.5 聚合负载分担类型.....	1-8
1.2 以太网链路聚合配置任务简介.....	1-8
1.3 配置聚合组.....	1-9
1.3.1 配置二层聚合组.....	1-9
1.3.2 配置三层聚合组.....	1-10
1.4 聚合接口相关配置.....	1-11
1.4.1 配置聚合接口的描述信息.....	1-12
1.4.2 配置二层聚合接口的忽略VLAN.....	1-12
1.4.3 配置三层聚合接口MTU.....	1-12
1.4.4 限制聚合组内选中端口的数量.....	1-13
1.4.5 配置聚合接口的期望带宽.....	1-13
1.4.6 配置聚合接口为聚合边缘接口.....	1-14
1.4.7 配置链路聚合与BFD联动.....	1-14
1.4.8 关闭聚合接口.....	1-15
1.4.9 恢复聚合接口的缺省配置.....	1-15
1.5 配置聚合负载分担.....	1-16
1.5.1 配置聚合负载分担类型.....	1-16
1.5.2 配置聚合负载分担采用本地转发优先.....	1-16
1.6 以太网链路聚合显示与维护.....	1-17
1.7 以太网链路聚合典型配置举例.....	1-18
1.7.1 二层静态聚合配置举例.....	1-18
1.7.2 二层动态聚合配置举例.....	1-19
1.7.3 三层静态聚合配置举例.....	1-21
1.7.4 三层动态聚合配置举例.....	1-23
1.7.5 二层聚合边缘接口配置举例.....	1-24

# 1 以太网链路聚合

## 1.1 以太网链路聚合简介

以太网链路聚合通过将多条以太网物理链路捆绑在一起形成一条以太网逻辑链路，实现增加链路带宽的目的，同时这些捆绑在一起的链路通过相互动态备份，可以有效地提高链路的可靠性。

如 [图 1-1](#) 所示，Device A 与 Device B 之间通过三条以太网物理链路相连，将这三条链路捆绑在一起，就成为了一条逻辑链路 Link aggregation 1。这条逻辑链路的带宽最大可等于三条以太网物理链路的带宽总和，增加了链路的带宽；同时，这三条以太网物理链路相互备份，当其中某条物理链路 down，还可以通过其他两条物理链路转发报文。

图1-1 链路聚合示意图



### 1.1.1 基本概念

#### 1. 聚合组、成员端口和聚合接口

链路捆绑是通过接口捆绑实现的，多个以太网接口捆绑在一起后形成一个聚合组，而这些被捆绑在一起的以太网接口就称为该聚合组的成员端口。每个聚合组唯一对应着一个逻辑接口，称为聚合接口。聚合组与聚合接口的编号是相同的，例如聚合组 1 对应于聚合接口 1。聚合组/聚合接口可以分为以下几种类型：

- 二层聚合组/二层聚合接口：二层聚合组的成员端口全部为二层以太网接口，其对应的聚合接口称为二层聚合接口。
- 三层聚合组/三层聚合接口：三层聚合组的成员端口全部为三层以太网接口，其对应的聚合接口称为三层聚合接口。

聚合接口的速率和双工模式取决于对应聚合组内的选中端口（请参见“[1.1.1.2. 成员端口的状态](#)”）：聚合接口的速率等于所有选中端口的速率之和，聚合接口的双工模式则与选中端口的双工模式相同。

#### 2. 成员端口的状态

聚合组内的成员端口具有以下三种状态：

- 选中（Selected）状态：此状态下的成员端口可以参与数据的转发，处于此状态的成员端口称为“选中端口”。
- 非选中（Unselected）状态：此状态下的成员端口不能参与数据的转发，处于此状态的成员端口称为“非选中端口”。
- 独立（Individual）状态：此状态下的成员端口可以作为普通物理口参与数据的转发。当聚合接口配置为聚合边缘接口，其成员端口未收到对端端口发送的 LACP（Link Aggregation Control Protocol，链路聚合控制协议）报文时，处于该状态。

### 3. 操作Key

操作 Key 是系统在进行链路聚合时用来表征成员端口聚合能力的一个数值，它是根据成员端口上的一些信息（包括该端口的速率、双工模式等）的组合自动计算生成的，这个信息组合中任何一项的变化都会引起操作 Key 的重新计算。在同一聚合组中，所有的选中端口都必须具有相同的操作 Key。

### 4. 配置分类

根据对成员端口状态的影响不同，成员端口上的配置可以分为以下两类：

- (1) 属性类配置：包含的配置内容如 [表 1-1](#) 所示。在聚合组中，只有与对应聚合接口的属性类配置完全相同的成员端口才能够成为选中端口。

表1-1 属性类配置的内容

配置项	内容
端口隔离	端口是否加入隔离组、端口所属的端口隔离组
QinQ配置	端口的QinQ功能开启/关闭状态。关于QinQ配置的详细描述请参见“二层技术-以太网交换配置指导”中的“QinQ”
VLAN配置	端口上允许通过的VLAN、端口缺省VLAN、端口的链路类型（即Trunk、Hybrid、Access类型）、基于IP子网的VLAN配置、基于协议的VLAN配置、VLAN报文是否带Tag配置。有关VLAN配置的详细描述，请参见“二层技术-以太网交换配置指导”中的“VLAN”



#### 说明

- 聚合接口上属性类配置发生变化时，会同步到成员端口上，同步失败时不会回退聚合接口上的配置。聚合接口配置同步到成员端口失败后，可能导致成员端口变为非选中状态，此时可以修改聚合接口或者成员端口上的配置，使成员端口重新选中。当聚合接口被删除后，同步成功的配置仍将保留在这些成员端口上。
- 由于成员端口上属性类配置的改变可能导致其选中/非选中状态发生变化，进而对业务产生影响，因此当在成员端口上进行此类配置时，系统将给出提示信息，由用户来决定是否继续执行该配置。

- (2) 协议类配置：是相对于属性类配置而言的，包含的配置内容有 MAC 地址学习、生成树等。在聚合组中，即使某成员端口与对应聚合接口的协议配置存在不同，也不会影响该成员端口成为选中端口。



#### 说明

- 在聚合接口上所作的协议类配置，只在当前聚合接口下生效。
- 在成员端口上所作的协议类配置，只有当该成员端口退出聚合组后才能生效。

### 5. 聚合模式

链路聚合分为静态聚合和动态聚合两种模式，它们各自的优点如下所示：

- 静态聚合模式：一旦配置好后，端口的选中/非选中状态就不会受网络环境的影响，比较稳定。

- 动态聚合模式：能够根据对端和本端的信息调整端口的选中/非选中状态，比较灵活。  
处于静态聚合模式下的聚合组称为静态聚合组，处于动态聚合模式下的聚合组称为动态聚合组。

## 1.1.2 静态聚合模式

静态聚合模式的工作机制如下所述。

### 1. 选择参考端口

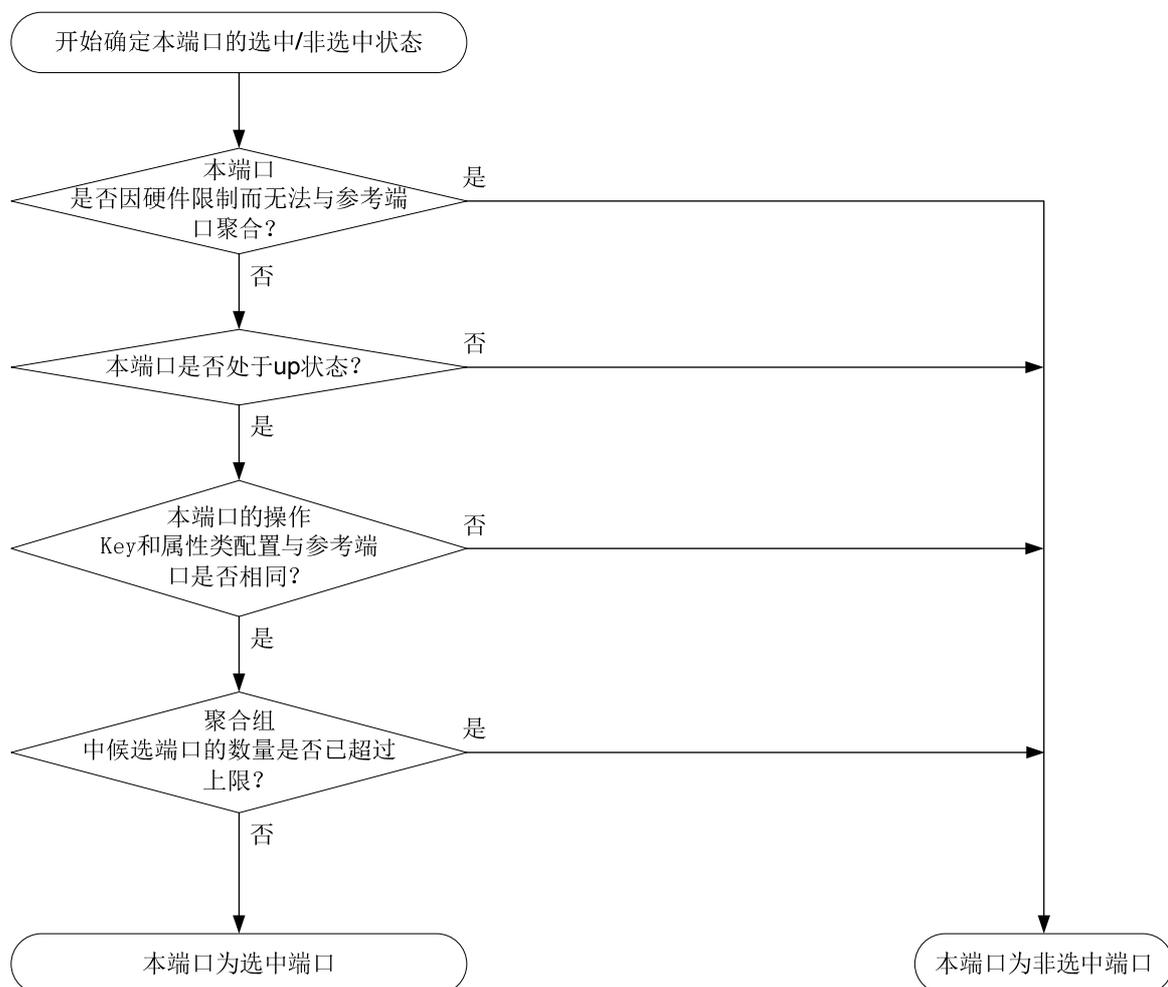
参考端口从本端的成员端口中选出，其操作 **Key** 和属性类配置将作为同一聚合组内的其他成员端口的参照，只有操作 **Key** 和属性类配置与参考端口一致的成员端口才能被选中。

对于聚合组内处于 **up** 状态的端口，按照端口的高端口优先级->全双工/高速率->全双工/低速率->半双工/高速率->半双工/低速率的优先次序，选择优先次序最高、且属性类配置与对应聚合接口相同的端口作为参考端口；如果多个端口优先次序相同，首先选择原来的选中端口作为参考端口；如果此时多个优先次序相同的端口都是原来的选中端口，则选择其中端口号最小的端口作为参考端口；如果多个端口优先次序相同，且都不是原来的选中端口，则选择其中端口号最小的端口作为参考端口。

### 2. 确定成员端口的状态

静态聚合组内成员端口状态的确定流程如 [图 1-2](#) 所示。

图1-2 静态聚合组内成员端口状态的确定流程



确定静态聚合组内成员端口状态时，需要注意：

- 当一个成员端口的操作 **Key** 或属性类配置改变时，其所在静态聚合组内各成员端口的选中/非选中状态可能会发生改变。
- 当静态聚合组内选中端口的数量已达到上限，对于后加入的成员端口和聚合组内选中端口的端口优先级：
  - 全部相同时，后加入的成员端口即使满足成为选中端口的所有条件，也不会立即成为选中端口。这样能够尽量维持当前选中端口上的流量不中断，但是由于设备重启时会重新计算选中端口，因此可能导致设备重启前后各成员端口的选中/非选中状态不一致。
  - 存在不同时，若后加入的成员端口的属性类配置与对应聚合接口相同，且端口优先级高于聚合组内选中端口的端口优先级，则端口优先级高的成员端口会立刻取代端口优先级低的选中端口成为新的选中端口。

### 1.1.3 动态聚合模式

动态聚合模式通过 LACP 协议实现，LACP 协议的内容及动态聚合模式的工作机制如下所述。

## 1. LACP协议

基于 IEEE802.3ad 标准的 LACP 协议是一种实现链路动态聚合的协议，运行该协议的设备之间通过互发 LACPDU 来交互链路聚合的相关信息。

动态聚合组内的成员端口可以收发 LACPDU（Link Aggregation Control Protocol Data Unit，链路聚合控制协议数据单元），本端通过向对端发送 LACPDU 通告本端的信息。当对端收到该 LACPDU 后，将其中的信息与所在端其他成员端口收到的信息进行比较，以选择能够处于选中状态的成员端口，使双方可以对各自接口的选中/非选中状态达成一致。

### (1) LACP 协议的功能

LACP协议的功能分为基本功能和扩展功能两大类，如 [表 1-2](#) 所示。

表1-2 LACP 协议的功能分类

类别	说明
基本功能	利用LACPDU的基本字段可以实现LACP协议的基本功能。基本字段包含以下信息：系统LACP优先级、系统MAC地址、端口优先级、端口编号和操作Key
扩展功能	通过对LACPDU的字段进行扩展，可以实现对LACP协议的扩展。通过在扩展字段中定义一个新的TLV（Type/Length/Value，类型/长度/值）数据域，可以实现IRF（Intelligent Resilient Framework，智能弹性架构）中的LACP MAD（Multi-Active Detection，多Active检测）机制。有关IRF和LACP MAD机制的详细介绍，请参见“IRF配置指导”中的“IRF” 本系列交换机可以作为成员设备或中间设备来参与LACP MAD

### (2) LACP 工作模式

LACP 工作模式分为 ACTIVE 和 PASSIVE 两种。

如果动态聚合组内成员端口的 LACP 工作模式为 PASSIVE，且对端的 LACP 工作模式也为 PASSIVE 时，两端将不能发送 LACPDU。如果两端中任何一端的 LACP 工作模式为 ACTIVE 时，两端将可以发送 LACPDU。

### (3) LACP 优先级

根据作用的不同，可以将LACP优先级分为系统LACP优先级和端口优先级两类，如 [表 1-3](#) 所示。

表1-3 LACP 优先级的分类

类别	说明	比较标准
系统LACP优先级	用于区分两端设备优先级的高低。当两端设备中的一端具有较高优先级时，另一端将根据优先级较高的一端来选择本端的选中端口，这样便使两端设备的选中端口达成了一致	优先级数值越小，优先级越高
端口优先级	用于区分各成员端口成为选中端口的优先程度	

### (4) LACP 超时时间

LACP 超时时间是指成员端口等待接收 LACPDU 的超时时间，在 LACP 超时时间之后，如果本端成员端口仍未收到来自对端的 LACPDU，则认为对端成员端口已失效。

LACP 超时时间同时也决定了对端发送 LACPDU 的速率。LACP 超时有短超时（3 秒）和长超时（90 秒）两种。若 LACP 超时时间为短超时，则对端将快速发送 LACPDU（每 1 秒发送 1 个 LACPDU）；若 LACP 超时时间为长超时，则对端将慢速发送 LACPDU（每 30 秒发送 1 个 LACPDU）。

## 2. 动态聚合模式的工作机制：

### (1) 选择参考端口

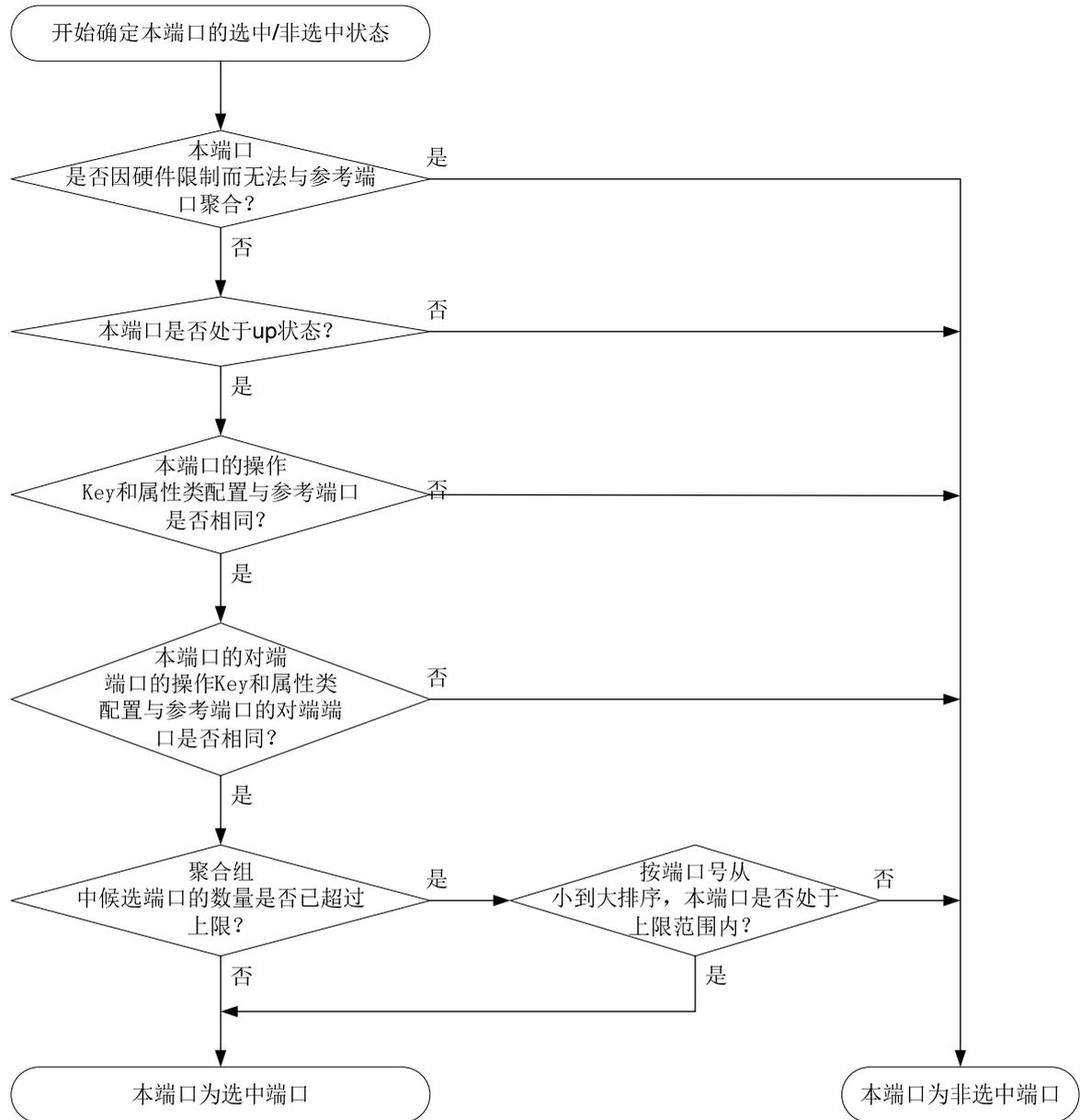
参考端口从聚合链路两端处于 up 状态的成员端口中选出，其操作 Key 和属性类配置将作为同一聚合组内的其他成员端口的参照，只有操作 Key 和属性类配置与参考端口一致的成员端口才能被选中。

- 首先，从聚合链路的两端选出设备 ID（由系统的 LACP 优先级和系统的 MAC 地址共同构成）较小的一端：先比较两端的系统 LACP 优先级，优先级数值越小其设备 ID 越小；如果优先级相同再比较其系统 MAC 地址，MAC 地址越小其设备 ID 越小。
- 其次，对于设备 ID 较小的一端，再比较其聚合组内各成员端口的端口 ID（由端口优先级和端口的编号共同构成）：先比较端口优先级，优先级数值越小其端口 ID 越小；如果优先级相同再比较其端口号，端口号越小其端口 ID 越小。端口 ID 最小、且属性类配置与对应聚合接口相同的端口作为参考端口。

### (2) 确定成员端口的状态

在设备ID较小的一端，动态聚合组内成员端口状态的确定流程如 [图 1-3](#) 所示。

图1-3 动态聚合组内成员端口状态的确定流程



与此同时，设备 ID 较大的一端也会随着对端成员端口状态的变化，随时调整本端各成员端口的状态，以确保聚合链路两端成员端口状态的一致。

确定动态聚合组内成员端口状态时，需要注意：

- 仅全双工端口可成为选中端口。
- 当一个成员端口的操作 Key 或属性类配置改变时，其所在动态聚合组内各成员端口的选中/非选中状态可能会发生改变。
- 当本端端口的选中/非选中状态发生改变时，其对端端口的选中/非选中状态也将随之改变。
- 当动态聚合组内选中端口的数量已达到上限时，后加入的成员端口一旦满足成为选中端口的所有条件，就会立刻取代已不满足条件的端口成为选中端口。

## 1.1.4 聚合边缘接口

在网络设备与服务器等终端设备相连的场景中，当网络设备配置了动态聚合模式，而终端设备未配置动态聚合模式时，聚合链路不能成功建立，网络设备与该终端设备相连多条链路中只能有一条作为普通链路正常转发报文，因而链路间也不能形成备份，当该普通链路发生故障时，可能会造成报文丢失。

若要求在终端设备未配置动态聚合模式时，该终端设备与网络设备间的链路可以形成备份，可通过配置网络设备与终端设备相连的聚合接口为聚合边缘接口，使该聚合组内的所有成员端口都作为普通物理口转发报文，从而保证终端设备与网络设备间的多条链路可以相互备份，增加可靠性。当终端设备完成动态聚合模式配置时，其聚合成员端口正常发送 LACP 报文后，网络设备上符合选中条件的聚合成员端口会自动被选中，从而使聚合链路恢复正常工作。

## 1.1.5 聚合负载分担类型

通过采用不同的聚合负载分担类型，可以实现灵活地对聚合组内流量进行负载分担。聚合负载分担的类型可以归为以下几类：

- 逐流负载分担：按照报文的源/目的 MAC 地址、源/目的服务端口、入端口、源/目的 IP 地址中的一种或某几种的组合区分流，使属于同一数据流的报文从同一条成员链路上通过。
- 按照报文类型（如二层协议报文、IPv4 报文、IPv6 报文等）自动选择所采用的聚合负载分担类型。

## 1.2 以太网链路聚合配置任务简介

表1-4 以太网链路聚合配置任务简介

配置任务		说明	详细配置
配置聚合组	配置二层聚合组	二者必选其一	<a href="#">1.3.1</a>
	配置三层聚合组		<a href="#">1.3.2</a>
聚合接口相关配置	配置聚合接口的描述信息	可选	<a href="#">1.4.1</a>
	配置二层聚合接口的忽略VLAN	可选	<a href="#">1.4.2</a>
	配置三层聚合接口MTU	可选	<a href="#">1.4.3</a>
	限制聚合组内选中端口的数量	可选	<a href="#">1.4.4</a>
	配置聚合接口的期望带宽	可选	<a href="#">1.4.5</a>
	配置聚合接口为聚合边缘接口	可选	<a href="#">1.4.6</a>
	配置链路聚合与BFD联动	可选	<a href="#">1.4.7</a>
	关闭聚合接口	可选	<a href="#">1.4.8</a>
配置聚合负载分担	恢复聚合接口的缺省配置	可选	<a href="#">1.4.9</a>
	配置聚合负载分担类型	可选	<a href="#">1.5.1</a>
	配置聚合负载分担采用本地转发优先	可选	<a href="#">1.5.2</a>

## 1.3 配置聚合组

配置聚合组时，需要注意：

- 配置了下列功能的端口将不能加入二层聚合组：**MAC** 地址认证（请参见“安全配置指导”中的“MAC 地址认证”）、端口安全（请参见“安全配置指导”中的“端口安全”）、**802.1X**（请参见“安全配置指导”中的“802.1X”）。
- 建议不要将镜像反射端口加入聚合组，有关反射端口的详细介绍请参见“网络管理和监控配置指导”中的“端口镜像”。
- 用户删除聚合接口时，系统将自动删除对应的聚合组，且该聚合组内的所有成员端口将全部离开该聚合组。
- 聚合链路的两端应配置相同的聚合模式。
- 二层聚合组和三层聚合组都分为静态聚合和动态聚合两种模式。
- 对于静态聚合模式，用户需要保证在同一链路两端端口的选中/非选中状态的一致性，否则聚合功能无法正常使用。
- 对于动态聚合模式，聚合链路两端的设备会自动协商同一链路两端的端口在各自聚合组内的选中/非选中状态，用户只需保证本端聚合在一起的端口的对端也同样聚合在一起，聚合功能即可正常使用。

### 1.3.1 配置二层聚合组

#### 1. 配置二层静态聚合组

表1-5 配置二层静态聚合组

操作	命令	说明
进入系统视图	<b>system-view</b>	-
创建二层聚合接口，并进入二层聚合接口视图	<b>interface bridge-aggregation</b> <i>interface-number</i>	创建二层聚合接口后，系统将自动生成同编号的二层聚合组，且该聚合组缺省工作在静态聚合模式下
退回系统视图	<b>quit</b>	-
进入二层以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	多次执行此步骤可将多个二层以太网接口加入聚合组
将二层以太网接口加入聚合组	<b>port link-aggregation group</b> <i>group-id</i>	
（可选）配置端口优先级	<b>link-aggregation port-priority</b> <i>priority</i>	缺省情况下，端口优先级为32768

#### 2. 配置二层动态聚合组

表1-6 配置二层动态聚合组

操作	命令	说明
进入系统视图	<b>system-view</b>	-

操作	命令	说明
配置系统的LACP优先级	<b>lacp system-priority</b> <i>priority</i>	缺省情况下,系统的LACP优先级为32768 改变系统的LACP优先级,将会影响到动态聚合组成员端口的选中/非选中状态
创建二层聚合接口,并进入二层聚合接口视图	<b>interface bridge-aggregation</b> <i>interface-number</i>	创建二层聚合接口后,系统将自动生成同编号的二层聚合组,且该聚合组缺省工作在静态聚合模式下
配置聚合组工作在动态聚合模式下	<b>link-aggregation mode dynamic</b>	缺省情况下,聚合组工作在静态聚合模式下
退回系统视图	<b>quit</b>	-
进入二层以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	多次执行此步骤可将多个二层以太网接口加入聚合组
将二层以太网接口加入聚合组	<b>port link-aggregation group</b> <i>group-id</i>	
配置端口的LACP工作模式为PASSIVE	<b>lacp mode passive</b>	二者选其一 缺省情况下,端口的LACP工作模式为ACTIVE
配置端口的LACP工作模式为ACTIVE	<b>undo lacp mode</b>	
配置端口优先级	<b>link-aggregation port-priority</b> <i>priority</i>	缺省情况下,端口优先级为32768
配置端口的LACP超时时间为短超时(3秒)	<b>lacp period short</b>	缺省情况下,端口的LACP超时时间为长超时(90秒) 请不要在ISSU升级前配置LACP超时时间为短超时,否则在ISSU升级期间会出现网络流量中断,导致流量转发不通。有关ISSU升级的详细介绍请参见“基础配置指导”中的“ISSU配置”

## 1.3.2 配置三层聚合组

### 1. 配置三层静态聚合组

表1-7 配置三层静态聚合组

操作	命令	说明
进入系统视图	<b>system-view</b>	-
创建三层聚合接口,并进入三层聚合接口视图	<b>interface route-aggregation</b> <i>interface-number</i>	创建三层聚合接口后,系统将自动生成同编号的三层聚合组,且该聚合组缺省工作在静态聚合模式下
退回系统视图	<b>quit</b>	-
进入三层以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	多次执行此步骤可将多个三层以太网接口加入聚合组
将三层以太网接口加入聚合组	<b>port link-aggregation group</b> <i>group-id</i>	

操作	命令	说明
(可选) 配置端口优先级	<b>link-aggregation port-priority <i>priority</i></b>	缺省情况下, 端口优先级为32768

## 2. 配置三层动态聚合组

表1-8 配置三层动态聚合组

操作	命令	说明
进入系统视图	<b>system-view</b>	-
配置系统的LACP优先级	<b>lacp system-priority <i>priority</i></b>	缺省情况下, 系统的LACP优先级为32768 改变系统的LACP优先级, 将会影响到动态聚合组成员的选中/非选中状态
创建三层聚合接口, 并进入三层聚合接口视图	<b>interface route-aggregation <i>interface-number</i></b>	创建三层聚合接口后, 系统将自动生成同编号的三层聚合组, 且该聚合组缺省工作在静态聚合模式下
配置聚合组工作在动态聚合模式下	<b>link-aggregation mode dynamic</b>	缺省情况下, 聚合组工作在静态聚合模式下
退回系统视图	<b>quit</b>	-
进入三层以太网接口视图	<b>interface <i>interface-type</i> <i>interface-number</i></b>	多次执行此步骤可将多个三层以太网接口加入聚合组
将三层以太网接口加入聚合组	<b>port link-aggregation group <i>group-id</i></b>	
配置端口的LACP工作模式为PASSIVE	<b>lacp mode passive</b>	二者选其一 缺省情况下, 端口的LACP工作模式为ACTIVE
配置端口的LACP工作模式为ACTIVE	<b>undo lacp mode</b>	
配置端口优先级	<b>link-aggregation port-priority <i>priority</i></b>	缺省情况下, 端口优先级为32768
配置端口的LACP超时时间为短超时(3秒)	<b>lacp period short</b>	缺省情况下, 端口的LACP超时时间为长超时(90秒) 请不要在ISSU升级前配置LACP超时时间为短超时, 否则在ISSU升级期间会出现网络流量中断, 导致流量转发不通。有关ISSU升级的详细介绍请参见“基础配置指导”中的“ISSU配置”

## 1.4 聚合接口相关配置

本节对能够在聚合接口上进行的部分配置进行介绍。除本节所介绍的配置外, 能够在二层/三层以太网接口上进行的配置大多数也能在二层/三层聚合接口上进行, 具体配置请参见相关的配置指导。

### 1.4.1 配置聚合接口的描述信息

通过在接口上配置描述信息，可以方便网络管理员根据这些信息来区分各接口的作用。

表1-9 配置聚合接口的描述信息

操作		命令	说明
进入系统视图		<b>system-view</b>	-
进入聚合接口视图	进入二层聚合接口视图	<b>interface bridge-aggregation</b> <i>interface-number</i>	-
	进入三层聚合接口视图	<b>interface route-aggregation</b> <i>interface-number</i>	-
配置当前接口的描述信息		<b>description</b> <i>text</i>	缺省情况下，接口的描述信息为“接口名 Interface”

### 1.4.2 配置二层聚合接口的忽略VLAN

未配置二层聚合接口的忽略 VLAN 时，只有当其成员端口上关于 VLAN 允许通过的配置（包括是否允许 VLAN 通过，以及通过的方式）与该二层聚合接口的配置完全相同时，该成员端口才有可能成为选中端口；配置了二层聚合接口的忽略 VLAN 后，即使其成员端口上关于这些 VLAN 允许通过的配置与该二层聚合接口上的配置不一致，也不影响该成员端口成为选中端口。

表1-10 配置二层聚合接口的忽略 VLAN

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二层聚合接口视图	<b>interface bridge-aggregation</b> <i>interface-number</i>	-
配置二层聚合接口的忽略 VLAN	<b>link-aggregation ignore vlan</b> <i>vlan-id-list</i>	缺省情况下，二层聚合接口未配置忽略VLAN

### 1.4.3 配置三层聚合接口MTU

MTU（Maximum Transmission Unit，最大传输单元）参数会影响 IP 报文的分片与重组，可以通过下面的配置来改变 MTU 值。

表1-11 配置三层聚合接口 MTU

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入三层聚合接口视图	<b>interface route-aggregation</b> <i>interface-number</i>	-
配置三层聚合接口的MTU值	<b>mtu size</b>	缺省情况下，三层聚合接口的 MTU值为1500字节

## 1.4.4 限制聚合组内选中端口的数量



提示

本端和对端配置的聚合组中的最小/最大选中端口数必须一致。

聚合链路的带宽取决于聚合组内选中端口的数量，用户通过配置聚合组中的最小选中端口数，可以避免由于选中端口太少而造成聚合链路上的流量拥塞。当聚合组内选中端口的数量达不到配置值时，对应的聚合接口将不会 up。具体实现如下：

- 如果聚合组内能够被选中的成员端口数小于配置值，这些成员端口都将变为非选中状态，对应聚合接口的链路状态也将变为 down。
- 当聚合组内能够被选中的成员端口数增加至不小于配置值时，这些成员端口都将变为选中状态，对应聚合接口的链路状态也将变为 up。

当配置了聚合组中的最大选中端口数之后，最大选中端口数将同时受配置值和设备硬件能力的限制，即取二者的较小值作为限制值。用户借此可实现两端口间的冗余备份：在一个聚合组中只添加两个成员端口，并配置该聚合组中的最大选中端口数为 1，这样这两个成员端口在同一时刻就只能有一个成为选中端口，而另一个将作为备份端口。

表1-12 限制聚合组内选中端口的数量

操作		命令	说明
进入系统视图		<b>system-view</b>	-
进入聚合接口视图	进入二层聚合接口视图	<b>interface bridge-aggregation</b> <i>interface-number</i>	-
	进入三层聚合接口	<b>interface route-aggregation</b> <i>interface-number</i>	
配置聚合组中的最小选中端口数		<b>link-aggregation selected-port minimum</b> <i>min-number</i>	缺省情况下，聚合组中的最小选中端口数不受限制
配置聚合组中的最大选中端口数		<b>link-aggregation selected-port maximum</b> <i>max-number</i>	缺省情况下，聚合组中的最大选中端口数为8

## 1.4.5 配置聚合接口的期望带宽

表1-13 配置聚合接口的期望带宽

操作		命令	说明
进入系统视图		<b>system-view</b>	-
进入聚合接口视图	进入二层聚合接口视图	<b>interface bridge-aggregation</b> <i>interface-number</i>	-
	进入三层聚合接口视图	<b>interface route-aggregation</b> <i>interface-number</i>	

操作	命令	说明
配置当前接口的期望带宽	<b>bandwidth bandwidth-value</b>	缺省情况下，接口的期望带宽=接口的波特率÷1000（kbps）

#### 1.4.6 配置聚合接口为聚合边缘接口

配置聚合接口为聚合边缘接口时，需要注意：

- 该配置仅在聚合接口对应的聚合组为动态聚合组时生效。
- 请在网络设备与服务器等终端设备相连的场景中使用边缘端口，不要在网络设备间配置边缘端口。

表1-14 配置聚合接口为聚合边缘接口

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入聚合接口视图	进入二层聚合接口视图 <b>interface bridge-aggregation interface-number</b>	-
	进入三层聚合接口视图 <b>interface route-aggregation interface-number</b>	
配置聚合接口为聚合边缘接口	<b>lACP edge-port</b>	缺省情况下，聚合接口不为聚合边缘接口

#### 1.4.7 配置链路聚合与BFD联动

链路聚合分为静态聚合和动态聚合两种模式，当链路发生故障时，静态聚合组没有检测机制来响应链路故障；动态聚合组通过 LACP 来判断链路状况，但这种方式不能快速响应链路故障。链路聚合使用 BFD（Bidirectional Forwarding Detection，双向转发检测），能够为聚合组选中端口间的链路提供快速检测功能。通过为选中端口创建 BFD 会话来实现对成员链路故障的快速检测。当链路发生故障时，该功能能够快速使双方对各自接口的选中/非选中状态达成一致。关于 BFD 的介绍和基本功能配置，请参见“可靠性配置指导”中的“BFD”。

- 静态聚合：如果 BFD 检测到链路故障，系统会通知聚合模块对端不可达，将该链路连接端口的选中状态修改为非选中状态，BFD 会话保留，并且会继续发送 BFD 报文；当故障链路恢复，能收到对端发送来的 BFD 报文时，系统会再通知聚合模块对端可达，端口又恢复为选中状态。即配置此功能后静态聚合链路不会出现一端为选中状态，另一端为非选中状态的情况。
- 动态聚合：如果 BFD 检测到链路故障，系统会通知聚合模块对端不可达，然后拆除 BFD 会话，并停止发送 BFD 报文；当故障链路恢复，通过 LACP 协议重新建立选中链路关系，并重建 BFD 会话，然后通知聚合模块对端已可达。从而使动态聚合组中成员端口选中状态快速收敛。

配置链路聚合与 BFD 联动时，需要注意：

- 两端聚合接口的 BFD 会话源地址和目的地址必须成对配置，且源地址和目的地址为不同的单播地址（0.0.0.0 除外）。例如本端聚合接口配置 **link-aggregation bfd ipv4 source 1.1.1.1**

**destination 2.2.2.2** 时，对端聚合接口要配置 **link-aggregation bfd ipv4 source 2.2.2.2 destination 1.1.1.1** 后，才能正确建立起 BFD 会话。

- 在聚合接口下配置的 BFD 会话参数，会对该聚合组内所有选中链路的 BFD 会话生效，链路聚合的 BFD 会话不支持 echo 功能和查询模式。
- 开启链路聚合的 BFD 功能后，不建议在该聚合接口上再开启其他应用与 BFD 联动。
- 开启链路聚合的 BFD 功能后，请配置聚合组中的成员端口数量不大于设备支持的 BFD 会话数量，否则可能导致聚合组内部分选中端口变为非选中状态。
- 当聚合链路两端最大选中端口数量不一致时，如果在链路两端都开启链路聚合的 BFD 功能，则可能会导致两端 BFD 会话数量不一致。

表1-15 配置链路聚合与 BFD 联动

操作		命令	说明
进入系统视图		<b>system-view</b>	-
进入聚合接口视图	进入二层聚合接口视图	<b>interface bridge-aggregation</b> <i>interface-number</i>	-
	进入三层聚合接口视图	<b>interface route-aggregation</b> <i>interface-number</i>	
开启链路聚合的BFD功能		<b>link-aggregation bfd ipv4 source</b> <i>ip-address</i> <b>destination</b> <i>ip-address</i>	缺省情况下，链路聚合的 BFD功能处于关闭状态

#### 1.4.8 关闭聚合接口

对聚合接口的开启/关闭操作，将会影响聚合接口对应的聚合组内成员端口的选中/非选中状态和链路状态：

- 关闭聚合接口时，将使对应聚合组内所有处于选中状态的成员端口都变为非选中端口，且所有成员端口的链路状态都将变为 down。
- 开启聚合接口时，系统将重新计算对应聚合组内成员端口的选中/非选中状态。

表1-16 关闭聚合接口

操作		命令	说明
进入系统视图		<b>system-view</b>	-
进入聚合接口视图	进入二层聚合接口视图	<b>interface bridge-aggregation</b> <i>interface-number</i>	-
	进入三层聚合接口视图	<b>interface route-aggregation</b> <i>interface-number</i>	
关闭当前接口		<b>shutdown</b>	-

#### 1.4.9 恢复聚合接口的缺省配置

通过执行本操作可以将聚合接口下的所有配置都恢复为缺省配置。

表1-17 恢复聚合接口的缺省配置

操作		命令	说明
进入系统视图		<b>system-view</b>	-
进入聚合接口视图	进入二层聚合接口视图	<b>interface bridge-aggregation</b> <i>interface-number</i>	-
	进入三层聚合接口视图	<b>interface route-aggregation</b> <i>interface-number</i>	
恢复当前聚合接口的缺省配置		<b>default</b>	-

## 1.5 配置聚合负载分担

### 1.5.1 配置聚合负载分担类型

聚合负载分担类型仅支持全局配置，全局的配置对所有聚合组都有效。

聚合负载分担类型会影响到等价路由的负载分担，可能导致等价路由负载分担不均匀。

表1-18 全局配置聚合负载分担类型

操作	命令	说明
进入系统视图	<b>system-view</b>	-
配置全局采用的聚合负载分担类型	<b>link-aggregation global load-sharing mode</b> { <b>destination-ip</b>   <b>destination-mac</b>   <b>destination-port</b>   <b>ingress-port</b>   <b>source-ip</b>   <b>source-mac</b>   <b>source-port</b> } *	缺省情况下，系统按照报文类型自动选择所采用的聚合负载分担类型

### 1.5.2 配置聚合负载分担采用本地转发优先

配置聚合负载分担采用本地转发优先机制可以降低数据流量对IRF物理端口之间链路的冲击，IRF中成员设备间聚合负载分担处理流程如 [图 1-4](#) 所示。有关IRF的详细介绍，请参见“IRF配置指导”中的“IRF”。

图1-4 IRF 中成员设备间聚合负载分担处理流程

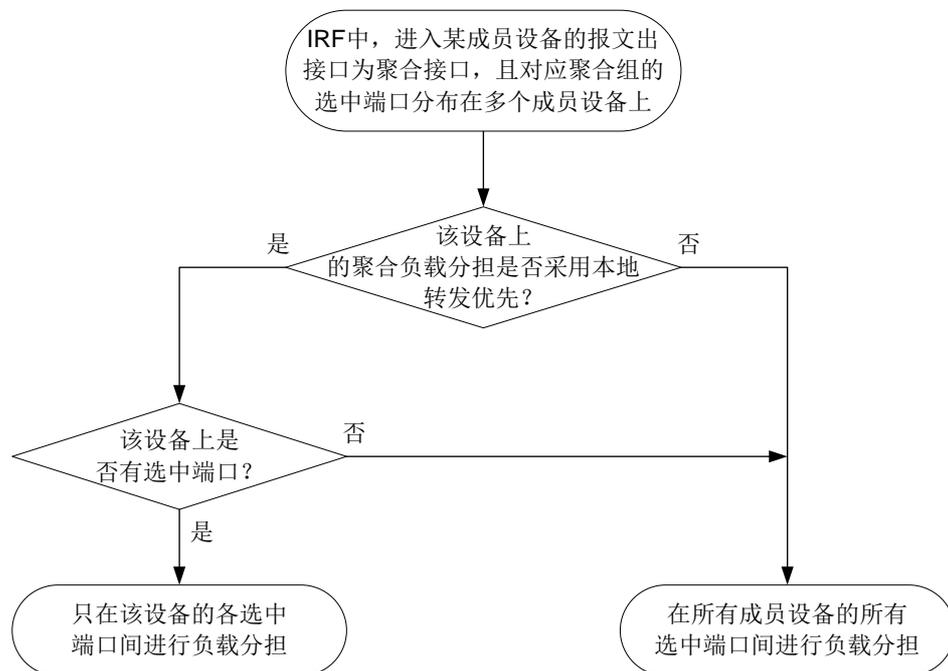


表1-19 配置聚合负载分担采用本地转发优先

操作	命令	说明
进入系统视图	<b>system-view</b>	-
配置聚合负载分担采用本地转发优先	<b>link-aggregation load-sharing mode local-first</b>	缺省情况下，聚合负载分担采用本地转发优先

## 1.6 以太网链路聚合显示与维护

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以太网链路聚合的运行情况，通过查看显示信息验证配置的效果。

在用户视图下执行 **reset** 命令可以清除端口的 LACP 和聚合接口上的统计信息。

表1-20 以太网链路聚合显示与维护

操作	命令
显示聚合接口的相关信息	<b>display interface</b> [ { <b>bridge-aggregation</b>   <b>route-aggregation</b> } [ <i>interface-number</i> ] ] [ <b>brief</b> [ <b>description</b>   <b>down</b> ] ]
显示本端系统的设备ID	<b>display lacp system-id</b>
显示全局或聚合组内采用的聚合负载分担类型	<b>display link-aggregation load-sharing mode</b> [ <b>interface</b> [ { <b>bridge-aggregation</b>   <b>route-aggregation</b> } <i>interface-number</i> ] ]
显示成员端口上链路聚合的详细信息	<b>display link-aggregation member-port</b> [ <i>interface-list</i> ]
显示所有聚合组的摘要信息	<b>display link-aggregation summary</b>

操作	命令
显示已有聚合接口所对应聚合组的详细信息	<b>display link-aggregation verbose</b> [ { <b>bridge-aggregation</b>   <b>route-aggregation</b> } [ <i>interface-number</i> ] ]
清除成员端口上的LACP统计信息	<b>reset lacp statistics</b> [ <b>interface</b> <i>interface-list</i> ]
清除聚合接口上的统计信息	<b>reset counters interface</b> [ { <b>bridge-aggregation</b>   <b>route-aggregation</b> } [ <i>interface-number</i> ] ]

## 1.7 以太网链路聚合典型配置举例

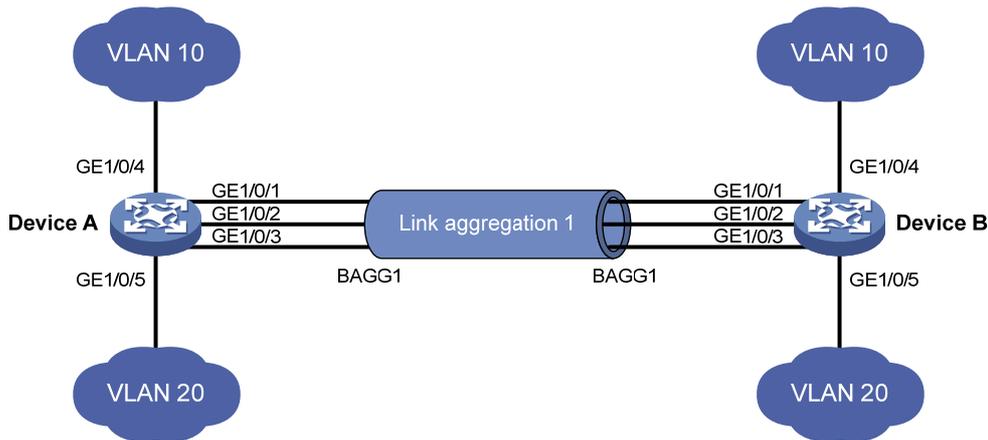
### 1.7.1 二层静态聚合配置举例

#### 1. 组网需求

- Device A 与 Device B 通过各自的二层以太网接口 GigabitEthernet1/0/1～GigabitEthernet1/0/3 相互连接。
- 在 Device A 和 Device B 上分别配置二层静态链路聚合组，并实现设备间 VLAN 10 和 VLAN 20 分别互通。

#### 2. 组网图

图1-5 二层静态聚合配置组网图



#### 3. 配置步骤

##### (1) 配置 Device A

# 创建 VLAN 10，并将端口 GigabitEthernet1/0/4 加入到该 VLAN 中。

```
<DeviceA> system-view
[DeviceA] vlan 10
[DeviceA-vlan10] port gigabitethernet 1/0/4
[DeviceA-vlan10] quit
```

# 创建 VLAN 20，并将端口 GigabitEthernet1/0/5 加入到该 VLAN 中。

```
[DeviceA] vlan 20
[DeviceA-vlan20] port gigabitethernet 1/0/5
```

```

[DeviceA-vlan20] quit
# 创建二层聚合接口 1。
[DeviceA] interface bridge-aggregation 1
[DeviceA-Bridge-Aggregation1] quit
# 分别将端口 GigabitEthernet1/0/1 至 GigabitEthernet1/0/3 加入到聚合组 1 中。
[DeviceA] interface gigabitethernet 1/0/1
[DeviceA-GigabitEthernet1/0/1] port link-aggregation group 1
[DeviceA-GigabitEthernet1/0/1] quit
[DeviceA] interface gigabitethernet 1/0/2
[DeviceA-GigabitEthernet1/0/2] port link-aggregation group 1
[DeviceA-GigabitEthernet1/0/2] quit
[DeviceA] interface gigabitethernet 1/0/3
[DeviceA-GigabitEthernet1/0/3] port link-aggregation group 1
[DeviceA-GigabitEthernet1/0/3] quit
# 配置二层聚合接口 1 为 Trunk 端口，并允许 VLAN 10 和 20 的报文通过。
[DeviceA] interface bridge-aggregation 1
[DeviceA-Bridge-Aggregation1] port link-type trunk
[DeviceA-Bridge-Aggregation1] port trunk permit vlan 10 20
[DeviceA-Bridge-Aggregation1] quit

```

## (2) 配置 Device B

Device B 的配置与 Device A 相似，配置过程略。

## 4. 验证配置

# 查看 Device A 上所有聚合组的详细信息。

```

[DeviceA] display link-aggregation verbose
Loadsharing Type: Shar -- Loadsharing, NonS -- Non-Loadsharing
Port Status: S -- Selected, U -- Unselected, I -- Individual
Port: A -- Auto port, M -- Management port, R -- Reference port
Flags: A -- LACP_Activity, B -- LACP_Timeout, C -- Aggregation,
       D -- Synchronization, E -- Collecting, F -- Distributing,
       G -- Defaulted, H -- Expired

```

```

Aggregate Interface: Bridge-Aggregation1
Aggregation Mode: Static
Loadsharing Type: NonS
Management VLANs: None

```

Port	Status	Priority	Oper-Key
GE1/0/1(R)	S	32768	1
GE1/0/2	S	32768	1
GE1/0/3	S	32768	1

以上信息表明，聚合组 1 为非负载分担类型的二层静态聚合组，包含有三个选中端口。

## 1.7.2 二层动态聚合配置举例

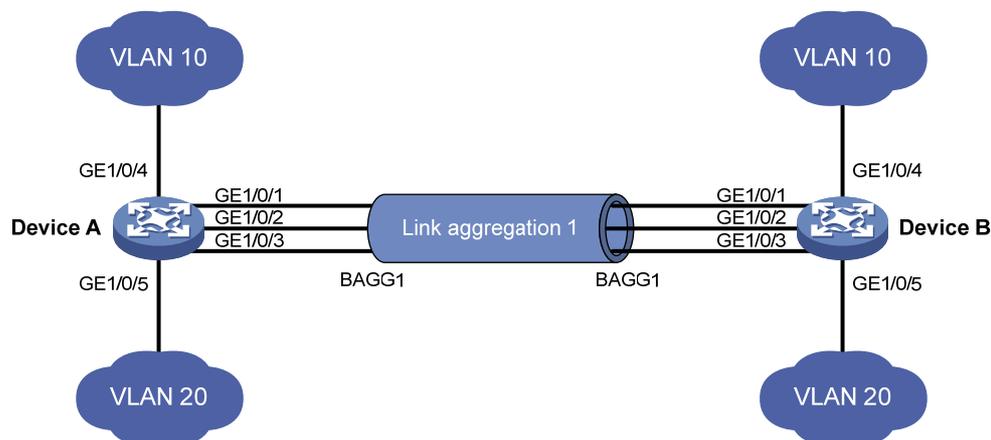
### 1. 组网需求

- Device A 与 Device B 通过各自的二层以太网接口 GigabitEthernet1/0/1~GigabitEthernet1/0/3 相互连接。

- 在 Device A 和 Device B 上分别配置二层动态链路聚合组，并实现设备间 VLAN 10 和 VLAN 20 分别互通。

## 2. 组网图

图1-6 二层动态聚合配置组网图



## 3. 配置步骤

### (1) 配置 Device A

# 创建 VLAN 10，并将端口 GigabitEthernet1/0/4 加入到该 VLAN 中。

```
<DeviceA> system-view
[DeviceA] vlan 10
[DeviceA-vlan10] port gigabitethernet 1/0/4
[DeviceA-vlan10] quit
```

# 创建 VLAN 20，并将端口 GigabitEthernet1/0/5 加入到该 VLAN 中。

```
[DeviceA] vlan 20
[DeviceA-vlan20] port gigabitethernet 1/0/5
[DeviceA-vlan20] quit
```

# 创建二层聚合接口 1，并配置该接口为动态聚合模式。

```
[DeviceA] interface bridge-aggregation 1
[DeviceA-Bridge-Aggregation1] link-aggregation mode dynamic
[DeviceA-Bridge-Aggregation1] quit
```

# 分别将端口 GigabitEthernet1/0/1 至 GigabitEthernet1/0/3 加入到聚合组 1 中。

```
[DeviceA] interface gigabitethernet 1/0/1
[DeviceA-GigabitEthernet1/0/1] port link-aggregation group 1
[DeviceA-GigabitEthernet1/0/1] quit
[DeviceA] interface gigabitethernet 1/0/2
[DeviceA-GigabitEthernet1/0/2] port link-aggregation group 1
[DeviceA-GigabitEthernet1/0/2] quit
[DeviceA] interface gigabitethernet 1/0/3
[DeviceA-GigabitEthernet1/0/3] port link-aggregation group 1
[DeviceA-GigabitEthernet1/0/3] quit
```

# 配置二层聚合接口 1 为 Trunk 端口，并允许 VLAN 10 和 20 的报文通过。

```
[DeviceA] interface bridge-aggregation 1
```

```
[DeviceA-Bridge-Aggregation1] port link-type trunk
[DeviceA-Bridge-Aggregation1] port trunk permit vlan 10 20
[DeviceA-Bridge-Aggregation1] quit
```

## (2) 配置 Device B

Device B 的配置与 Device A 相似，配置过程略。

## 4. 验证配置

# 查看 Device A 上所有聚合组的详细信息。

```
[DeviceA] display link-aggregation verbose
Loadsharing Type: Shar -- Loadsharing, NonS -- Non-Loadsharing
Port Status: S -- Selected, U -- Unselected, I -- Individual
Port: A -- Auto port, M -- Management port, R -- Reference port
Flags: A -- LACP_Activity, B -- LACP_Timeout, C -- Aggregation,
       D -- Synchronization, E -- Collecting, F -- Distributing,
       G -- Defaulted, H -- Expired
```

```
Aggregate Interface: Bridge-Aggregation1
```

```
Aggregation Mode: Dynamic
```

```
Loadsharing Type: NonS
```

```
Management VLANs: None
```

```
System ID: 0x8000, 000f-e267-6c6a
```

```
Local:
```

Port	Status	Priority	Index	Oper-Key	Flag
GE1/0/1(R)	S	32768	11	1	{ACDEF}
GE1/0/2	S	32768	12	1	{ACDEF}
GE1/0/3	S	32768	13	1	{ACDEF}

```
Remote:
```

Actor	Priority	Index	Oper-Key	SystemID	Flag
GE1/0/1	32768	81	1	0x8000, 000f-e267-57ad	{ACDEF}
GE1/0/2	32768	82	1	0x8000, 000f-e267-57ad	{ACDEF}
GE1/0/3	32768	83	1	0x8000, 000f-e267-57ad	{ACDEF}

以上信息表明，聚合组 1 为非负载分担类型的二层动态聚合组，包含有三个选中端口。

## 1.7.3 三层静态聚合配置举例

### 1. 组网需求

- Device A 与 Device B 通过各自的三层以太网接口 GigabitEthernet1/0/1~GigabitEthernet1/0/3 相互连接。
- 在 Device A 和 Device B 上分别配置三层静态链路聚合组，并为对应的三层聚合接口配置 IP 地址和子网掩码。

## 2. 组网图

图1-7 三层静态聚合配置组网图



## 3. 配置步骤

### (1) 配置 Device A

# 创建三层聚合接口 1，并为该接口配置 IP 地址和子网掩码。

```
<DeviceA> system-view
[DeviceA] interface route-aggregation 1
[DeviceA-Route-Aggregation1] ip address 192.168.1.1 24
[DeviceA-Route-Aggregation1] quit
```

# 分别将接口 GigabitEthernet1/0/1 至 GigabitEthernet1/0/3 加入到聚合组 1 中。

```
[DeviceA] interface gigabitethernet 1/0/1
[DeviceA-GigabitEthernet1/0/1] port link-aggregation group 1
[DeviceA-GigabitEthernet1/0/1] quit
[DeviceA] interface gigabitethernet 1/0/2
[DeviceA-GigabitEthernet1/0/2] port link-aggregation group 1
[DeviceA-GigabitEthernet1/0/2] quit
[DeviceA] interface gigabitethernet 1/0/3
[DeviceA-GigabitEthernet1/0/3] port link-aggregation group 1
[DeviceA-GigabitEthernet1/0/3] quit
```

### (2) 配置 Device B

Device B 的配置与 Device A 相似，配置过程略。

## 4. 验证配置

# 查看 Device A 上所有聚合组的详细信息。

```
[DeviceA] display link-aggregation verbose
Loadsharing Type: Shar -- Loadsharing, NonS -- Non-Loadsharing
Port Status: S -- Selected, U -- Unselected, I -- Individual
Port: A -- Auto port, M -- Management port, R -- Reference port
Flags: A -- LACP_Activity, B -- LACP_Timeout, C -- Aggregation,
       D -- Synchronization, E -- Collecting, F -- Distributing,
       G -- Defaulted, H -- Expired
```

```
Aggregate Interface: Route-Aggregation1
Aggregation Mode: Static
Loadsharing Type: NonS
Management VLANs: None
  Port          Status  Priority Oper-Key
  GE1/0/1(R)    S       32768   1
  GE1/0/2       S       32768   1
  GE1/0/3       S       32768   1
```

以上信息表明，聚合组 1 为非负载分担类型的三层静态聚合组，包含有三个选中端口。

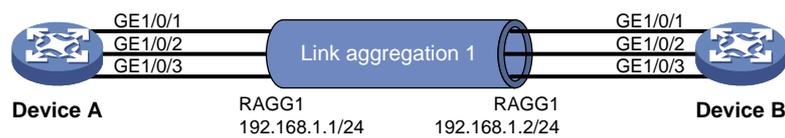
## 1.7.4 三层动态聚合配置举例

### 1. 组网需求

- Device A 与 Device B 通过各自的三层以太网接口 GigabitEthernet1/0/1～GigabitEthernet1/0/3 相互连接。
- 在 Device A 和 Device B 上分别配置三层动态链路聚合组，并为对应的三层聚合接口配置 IP 地址和子网掩码。

### 2. 组网图

图1-8 三层动态聚合配置组网图



### 3. 配置步骤

#### (1) 配置 Device A

# 创建三层聚合接口 1，配置该接口为动态聚合模式，并为其配置 IP 地址和子网掩码。

```
<DeviceA> system-view
[DeviceA] interface route-aggregation 1
[DeviceA-Route-Aggregation1] link-aggregation mode dynamic
[DeviceA-Route-Aggregation1] ip address 192.168.1.1 24
[DeviceA-Route-Aggregation1] quit
```

# 分别将接口 GigabitEthernet1/0/1 至 GigabitEthernet1/0/3 加入到聚合组 1 中。

```
[DeviceA] interface gigabitethernet 1/0/1
[DeviceA-GigabitEthernet1/0/1] port link-aggregation group 1
[DeviceA-GigabitEthernet1/0/1] quit
[DeviceA] interface gigabitethernet 1/0/2
[DeviceA-GigabitEthernet1/0/2] port link-aggregation group 1
[DeviceA-GigabitEthernet1/0/2] quit
[DeviceA] interface gigabitethernet 1/0/3
[DeviceA-GigabitEthernet1/0/3] port link-aggregation group 1
[DeviceA-GigabitEthernet1/0/3] quit
```

#### (2) 配置 Device B

Device B 的配置与 Device A 相似，配置过程略。

### 4. 验证配置

# 查看 Device A 上所有聚合组的详细信息。

```
[DeviceA] display link-aggregation verbose
Loadsharing Type: Shar -- Loadsharing, NonS -- Non-Loadsharing
Port Status: S -- Selected, U -- Unselected, I -- Individual
Port: A -- Auto port, M -- Management port, R -- Reference port
Flags: A -- LACP_Activity, B -- LACP_Timeout, C -- Aggregation,
```

D -- Synchronization, E -- Collecting, F -- Distributing,  
G -- Defaulted, H -- Expired

Aggregate Interface: Route-Aggregation1

Aggregation Mode: Dynamic

Loadsharing Type: NonS

Management VLANs: None

System ID: 0x8000, 000f-e267-6c6a

Local:

Port	Status	Priority	Index	Oper-Key	Flag
GE1/0/1(R)	S	32768	11	1	{ACDEF}
GE1/0/2	S	32768	12	1	{ACDEF}
GE1/0/3	S	32768	13	1	{ACDEF}

Remote:

Actor	Priority	Index	Oper-Key	SystemID	Flag
GE1/0/1	32768	81	1	0x8000, 000f-e267-57ad	{ACDEF}
GE1/0/2	32768	82	1	0x8000, 000f-e267-57ad	{ACDEF}
GE1/0/3	32768	83	1	0x8000, 000f-e267-57ad	{ACDEF}

以上信息表明，聚合组 1 为非负载分担类型的三层动态聚合组，包含有三个选中端口。

## 1.7.5 二层聚合边缘接口配置举例

### 1. 组网需求

- Device 与服务器 Server 通过端口 GigabitEthernet1/0/1、GigabitEthernet1/0/2 相互连接。
- 在 Device 上配置一个二层动态链路聚合组。
- 在 Device 上配置二层聚合接口为聚合边缘接口，以便当服务器上未配置动态聚合组时，Device 上聚合组成员端口都能做为普通端口正常转发报文。

### 2. 组网图

图1-9 二层聚合边缘接口配置组网图



### 3. 配置步骤

配置 Device

# 创建二层聚合接口 1，配置该接口为动态聚合模式。

```
<Device> system-view
```

```
[Device] interface bridge-aggregation 1
```

```
[Device-Bridge-Aggregation1] link-aggregation mode dynamic
```

# 配置二层聚合接口 1 为聚合边缘接口。

```
[Device-Bridge-Aggregation1] lacp edge-port
```

```
[Device-Bridge-Aggregation1] quit
```

# 分别将端口 GigabitEthernet1/0/1、GigabitEthernet1/0/2 加入到聚合组 1 中。

```
[Device] interface gigabitethernet 1/0/1
[Device-GigabitEthernet1/0/1] port link-aggregation group 1
[Device-GigabitEthernet1/0/1] quit
[Device] interface gigabitethernet 1/0/2
[Device-GigabitEthernet1/0/2] port link-aggregation group 1
[Device-GigabitEthernet1/0/2] quit
```

#### 4. 验证配置

# 当 Server 未完成动态聚合模式配置时，查看 Device 上所有聚合组的详细信息。

```
[Device] display link-aggregation verbose
Loadsharing Type: Shar -- Loadsharing, NonS -- Non-Loadsharing
Port Status: S -- Selected, U -- Unselected, I -- Individual
Port: A -- Auto port, M -- Management port, R -- Reference port
Flags: A -- LACP_Activity, B -- LACP_Timeout, C -- Aggregation,
       D -- Synchronization, E -- Collecting, F -- Distributing,
       G -- Defaulted, H -- Expired
```

```
Aggregate Interface: Bridge-Aggregation1
Aggregation Mode: Dynamic
Loadsharing Type: NonS
Management VLANs: None
System ID: 0x8000, 000f-e267-6c6a
```

Local:

Port	Status	Priority	Index	Oper-Key	Flag
GE1/0/1	I	32768	11	1	{AG}
GE1/0/2	I	32768	12	1	{AG}

Remote:

Actor	Priority	Index	Oper-Key	SystemID	Flag
GE1/0/1	32768	81	0	0x8000, 0000-0000-0000	{DEF}
GE1/0/2	32768	82	0	0x8000, 0000-0000-0000	{DEF}

以上信息表明，当 Device 未收到 Server 的 LACP 报文时，Device 的聚合成员端口都工作在 Individual 状态，该状态下所有聚合成员端口可以作为普通物理口转发报文，以保证此时 Server 与 Device 间的链路都可以正常转发报文，且相互形成备份。

# 目 录

1 端口隔离.....	1-1
1.1 端口隔离简介.....	1-1
1.2 配置隔离组.....	1-1
1.3 端口隔离显示和维护.....	1-1
1.4 端口隔离典型配置举例.....	1-2

# 1 端口隔离

## 1.1 端口隔离简介

为了实现端口间的二层隔离，可以将不同的端口加入不同的 VLAN，但 VLAN 资源有限。采用端口隔离特性，用户只需要将端口加入到隔离组中，就可以实现隔离组内端口之间二层隔离，而不关心这些端口所属 VLAN，从而节省 VLAN 资源。

隔离组内的端口与未加入隔离组的端口之间二层流量双向互通。

## 1.2 配置隔离组

设备支持多个隔离组，用户可以手工配置。隔离组内可以加入的端口数量没有限制。

表1-1 配置隔离组

操作		命令	说明
进入系统视图		<b>system-view</b>	-
创建隔离组		<b>port-isolate group</b> <i>group-id</i>	缺省情况下，不存在隔离组
进入相应视图	进入二层以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	<ul style="list-style-type: none"><li>• 二层以太网接口视图下的配置只对当前端口生效</li><li>• 二层聚合接口视图下的配置对当前接口及其成员端口生效，若某成员端口配置失败，系统会跳过该端口继续配置其他成员端口，若二层聚合接口配置失败，则不会再配置成员端口</li></ul>
	进入二层聚合接口视图	<b>interface bridge-aggregation</b> <i>interface-number</i>	
将端口加入到隔离组中		<b>port-isolate enable group</b> <i>group-id</i>	缺省情况下，当前端口不属于任何隔离组 一个端口最多只能加入一个隔离组

## 1.3 端口隔离显示和维护

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端口隔离的运行情况，通过查看显示信息验证配置的效果。

表1-2 端口隔离显示和维护

操作	命令
显示隔离组的信息	<b>display port-isolate group</b> [ <i>group-id</i> ]

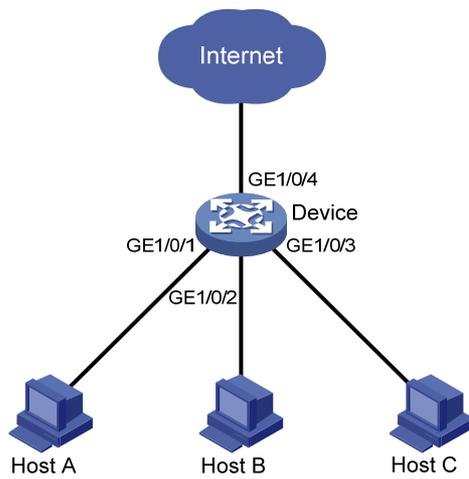
## 1.4 端口隔离典型配置举例

### 1. 组网需求

如 [图 1-1](#) 所示，小区用户Host A、Host B、Host C分别与Device的端口GigabitEthernet1/0/1、GigabitEthernet1/0/2、GigabitEthernet1/0/3 相连，Device设备通过GigabitEthernet1/0/4 端口与外部网络相连。现要实现小区用户Host A、Host B和Host C彼此之间二层报文不能互通，但可以和外部网络通信。

### 2. 组网图

图1-1 端口隔离组网图



### 3. 配置步骤

# 创建隔离组 1。

```
<Device> system-view
```

```
[Device] port-isolate group 1
```

# 将端口 GigabitEthernet1/0/1、GigabitEthernet1/0/2、GigabitEthernet1/0/3 加入隔离组 1。

```
[Device] interface gigabitethernet 1/0/1
```

```
[Device-GigabitEthernet1/0/1] port-isolate enable group 1
```

```
[Device-GigabitEthernet1/0/1] quit
```

```
[Device] interface gigabitethernet 1/0/2
```

```
[Device-GigabitEthernet1/0/2] port-isolate enable group 1
```

```
[Device-GigabitEthernet1/0/2] quit
```

```
[Device] interface gigabitethernet 1/0/3
```

```
[Device-GigabitEthernet1/0/3] port-isolate enable group 1
```

```
[Device-GigabitEthernet1/0/3] quit
```

### 4. 验证配置

# 显示隔离组 1 中的信息。

```
[Device] display port-isolate group 1
```

```
Port isolation group information:
```

```
Group ID: 1
```

```
Group members:
```

GigabitEthernet1/0/1      GigabitEthernet1/0/2      GigabitEthernet1/0/3

以上信息显示 Device 上的端口 GigabitEthernet1/0/1、GigabitEthernet1/0/2、GigabitEthernet1/0/3 已经加入隔离组 1，从而实现二层隔离，Host A、Host B 和 Host C 彼此之间不能 ping 通。

# 目 录

<b>1 VLAN</b> .....	<b>1-1</b>
1.1 VLAN简介 .....	1-1
1.1.1 VLAN概述 .....	1-1
1.1.2 VLAN报文封装 .....	1-2
1.1.3 协议规范 .....	1-2
1.2 配置VLAN .....	1-3
1.3 配置VLAN接口 .....	1-3
1.3.1 配置VLAN接口基本属性 .....	1-3
1.4 配置基于端口的VLAN .....	1-4
1.4.1 基于端口的VLAN简介 .....	1-4
1.4.2 配置基于Access端口的VLAN .....	1-6
1.4.3 配置基于Trunk端口的VLAN .....	1-6
1.4.4 配置基于Hybrid端口的VLAN .....	1-7
1.5 配置基于MAC的VLAN .....	1-8
1.5.1 基于MAC的VLAN简介 .....	1-8
1.5.2 配置基于MAC的VLAN .....	1-10
1.6 配置基于IP子网的VLAN .....	1-12
1.6.1 基于IP子网的VLAN简介 .....	1-12
1.6.2 配置基于IP子网的VLAN .....	1-12
1.7 配置基于协议的VLAN .....	1-13
1.7.1 基于协议的VLAN简介 .....	1-13
1.7.2 配置基于协议的VLAN .....	1-13
1.8 VLAN显示和维护 .....	1-14
1.9 基于端口的VLAN典型配置举例 .....	1-15
1.10 基于MAC的VLAN典型配置举例 .....	1-16
1.11 基于IP子网的VLAN典型配置举例 .....	1-18
1.12 基于协议的VLAN典型配置举例 .....	1-20
<b>2 Super VLAN</b> .....	<b>2-1</b>
2.1 Super VLAN简介 .....	2-1
2.2 配置Super VLAN .....	2-1
2.3 Super VLAN显示和维护 .....	2-3
2.4 Super VLAN典型配置举例 .....	2-3

<b>3 Voice VLAN</b> .....	<b>3-1</b>
3.1 Voice VLAN简介 .....	3-1
3.2 设备识别IP电话.....	3-1
3.2.1 OUI地址 .....	3-1
3.2.2 通过LLDP自动识别IP电话.....	3-2
3.3 设备将Voice VLAN信息通告给IP电话.....	3-2
3.4 IP电话的接入方式.....	3-3
3.5 Voice VLAN的工作模式 .....	3-3
3.5.1 Voice VLAN的自动模式和手动模式 .....	3-3
3.5.2 Voice VLAN的安全模式和普通模式 .....	3-5
3.6 Voice VLAN配置任务简介.....	3-6
3.7 配置Voice VLAN .....	3-6
3.7.1 配置语音报文的QoS优先级 .....	3-6
3.7.2 配置自动模式下的Voice VLAN.....	3-7
3.7.3 配置手动模式下的Voice VLAN.....	3-8
3.7.4 配置通过LLDP自动发现IP电话功能.....	3-9
3.7.5 配置通过LLDP通告Voice VLAN信息 .....	3-9
3.7.6 配置通过CDP通告Voice VLAN信息.....	3-9
3.8 Voice VLAN显示和维护 .....	3-10
3.9 Voice VLAN典型配置举例.....	3-11
3.9.1 自动模式下Voice VLAN的配置举例 .....	3-11
3.9.2 手动模式下Voice VLAN的配置举例 .....	3-12

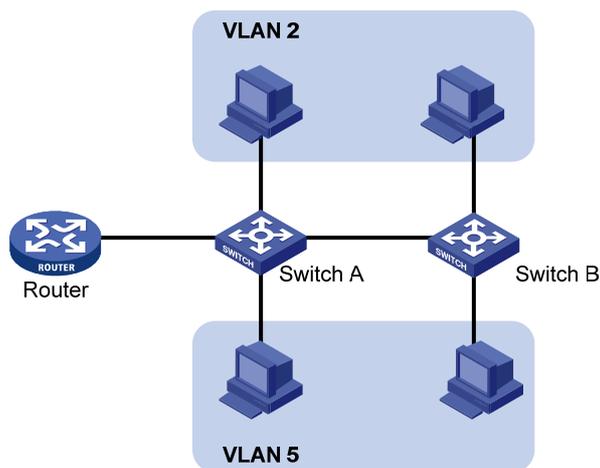
# 1 VLAN

## 1.1 VLAN简介

### 1.1.1 VLAN概述

以太网是一种基于CSMA/CD（Carrier Sense Multiple Access/Collision Detect，带冲突检测的载波侦听多路访问）技术的共享通讯介质。采用以太网技术构建的局域网，既是一个冲突域，又是一个广播域。当网络中主机数目较多时会导致冲突严重、广播泛滥、性能显著下降，甚至网络不可用等问题。通过在以太网中部署网桥或二层交换机，可以解决冲突严重的问题，但仍然不能隔离广播报文。在这种情况下出现了VLAN（Virtual Local Area Network，虚拟局域网）技术，这种技术可以把一个物理LAN划分成多个逻辑的LAN——VLAN。处于同一VLAN的主机能直接互通，而处于不同VLAN的主机则不能直接互通。这样，广播报文被限制在同一个VLAN内，即每个VLAN是一个广播域。如 [图 1-1](#) 所示，VLAN 2 内的主机可以互通，但与VLAN 5 内的主机不能互通。

图1-1 VLAN 示意图



VLAN 的划分不受物理位置的限制：物理位置不在同一范围的主机可以属于同一个 VLAN；一个 VLAN 包含的主机可以连接在同一个交换机上，也可以跨越交换机，甚至可以跨越路由器。

VLAN 根据划分方式不同可以分为不同类型。基于端口划分 VLAN 是最简单、最有效的 VLAN 划分方式。它按照设备端口来定义 VLAN 成员，将指定端口加入到指定 VLAN 中之后，端口就可以转发该 VLAN 的报文。本章将介绍基于端口的 VLAN。

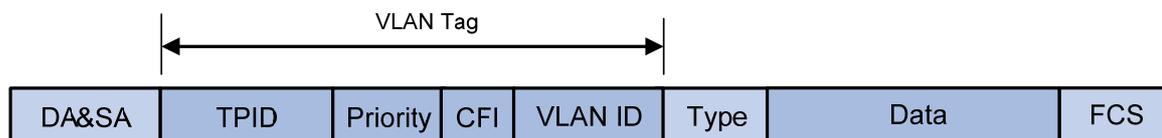
VLAN 的优点如下：

- 限制广播域。广播域被限制在一个 VLAN 内，节省了带宽，提高了网络处理能力。
- 增强局域网的安全性。VLAN 间的二层报文是相互隔离的，即一个 VLAN 内的主机不能和其他 VLAN 内的主机直接通信，如果不同 VLAN 要进行通信，则需通过路由器或三层交换机等三层设备。
- 灵活构建虚拟工作组。通过 VLAN 可以将不同的主机划分到不同的工作组，同一工作组的主机可以位于不同的物理位置，网络构建和维护更方便灵活。

## 1.1.2 VLAN报文封装

要使网络设备能够分辨不同 VLAN 的报文，需要在报文中添加标识 VLAN 的字段。IEEE 802.1Q 协议规定，在以太网报文的目的地 MAC 地址和源 MAC 地址字段之后、协议类型字段之前加入 4 个字节的 VLAN Tag，用以标识 VLAN 的相关信息。

图1-2 VLAN Tag 的组成字段



如 图 1-2 所示，VLAN Tag 包含四个字段，分别是 TPID（Tag Protocol Identifier，标签协议标识符）、Priority、CFI（Canonical Format Indicator，标准格式指示位）和 VLAN ID。

- **TPID:** 协议规定 TPID 取值为 0x8100 时表示报文带有 VLAN Tag，但各设备厂商可以自定义该字段的值。当邻居设备将 TPID 值配置为非 0x8100 时，为了能够识别这样的报文，实现互通，必须在本设备上修改 TPID 值，确保和邻居设备的 TPID 值配置一致。如果报文的 TPID 值为配置值或 0x8100，则该报文被认为带有 VLAN Tag。配置 TPID 值的相关命令请参见“二层技术-以太网交换命令参考”中的“QinQ”。
- **Priority:** 用来表示报文的 802.1p 优先级，长度为 3 比特，相关内容请参见“ACL 和 QoS 配置指导/QoS”中的“附录”。
- **CFI:** 用来表示 MAC 地址在不同的传输介质中是否以标准格式进行封装，长度为 1 比特。取值为 0 表示 MAC 地址以标准格式进行封装，为 1 表示以非标准格式封装。在以太网中，CFI 取值为 0。
- **VLAN ID:** 用来表示该报文所属 VLAN 的编号，长度为 12 比特。由于 0 和 4095 为协议保留取值，所以 VLAN ID 的取值范围为 1~4094。

网络设备根据报文是否携带 VLAN Tag 以及携带的 VLAN Tag 信息，来对报文进行处理，利用 VLAN ID 来识别报文所属的 VLAN。详细的处理方式请参见“[1.4.1 基于端口的 VLAN 简介](#)”。

### 说明

- 以太网支持 Ethernet II、802.3/802.2 LLC、802.3/802.2 SNAP 和 802.3 raw 封装格式，本文以 Ethernet II 型封装为例。802.3/802.2 LLC、802.3/802.2 SNAP 和 802.3 raw 封装格式添加 VLAN Tag 字段的方式请参见相关协议规范。
- 对于携带有多层 VLAN Tag 的报文，设备会根据其最外层 VLAN Tag 进行处理，而内层 VLAN Tag 会被视为报文的普通数据部分。

## 1.1.3 协议规范

与 VLAN 相关的协议规范有：

- IEEE 802.1Q: IEEE Standard for Local and Metropolitan Area Networks-Virtual Bridged Local Area Networks

## 1.2 配置VLAN

表1-1 配置 VLAN

配置	命令	说明
进入系统视图	<b>system-view</b>	-
(可选) 创建一个VLAN并进入VLAN视图, 或批量创建VLAN	<b>vlan { vlan-id1 [ to vlan-id2 ] / all }</b>	缺省情况下, 系统只有一个缺省VLAN (VLAN 1)
进入VLAN视图	<b>vlan vlan-id</b>	批量创建VLAN时, 为必选; 否则, 无需执行本命令
指定VLAN的名称	<b>name text</b>	缺省情况下, VLAN的名称为“VLAN <i>vlan-id</i> ”, 其中 <i>vlan-id</i> 为该VLAN的四位数编号, 如果该VLAN的编号不足四位, 则会在编号前增加0, 补齐四位。例如, VLAN 100的名称为“VLAN 0100”
配置VLAN的描述信息	<b>description text</b>	缺省情况下, VLAN的描述信息为“VLAN <i>vlan-id</i> ”, 其中 <i>vlan-id</i> 为该VLAN的四位数编号, 如果该VLAN的编号不足四位, 则会在编号前增加0, 补齐四位。例如, VLAN 100的描述信息为“VLAN 0100”

### 说明

- VLAN 1 为系统缺省 VLAN, 用户不能手工创建和删除。
- 动态学习到的 VLAN, 以及被其他应用锁定不让删除的 VLAN, 都不能使用 **undo vlan** 命令直接删除。只有将相关配置删除之后, 才能删除相应的 VLAN。

## 1.3 配置VLAN接口

不同 VLAN 间的主机不能直接通信, 通过在设备上创建并配置 VLAN 接口, 可以实现 VLAN 间的三层互通。

VLAN 接口是一种三层的虚拟接口, 它不作为物理实体存在于设备上。每个 VLAN 对应一个 VLAN 接口, 在为 VLAN 接口配置了 IP 地址后, 该 IP 地址即可作为本 VLAN 内网络设备的网关地址, 此时该 VLAN 接口能对需要跨网段的报文进行三层转发。

### 1.3.1 配置VLAN接口基本属性

配置 VLAN 接口基本属性时, 需要注意:

- 在创建 VLAN 接口之前, 对应的 VLAN 必须已经存在, 否则将不能创建指定的 VLAN 接口。
- 不能对 Sub VLAN 创建对应的 VLAN 接口。有关 Sub VLAN 的详细介绍, 请参见“二层技术-以太网交换配置指导”中的“Super VLAN”。

表1-2 配置 VLAN 接口基本属性

配置	命令	说明
进入系统视图	<b>system-view</b>	-
创建VLAN接口并进入VLAN接口视图	<b>interface vlan-interface</b> <i>interface-number</i>	如果该VLAN接口已经存在,则直接进入该VLAN接口视图 缺省情况下,不存在VLAN接口
配置VLAN接口的IP地址	<b>ip address</b> <i>ip-address</i> { <i>mask</i>   <i>mask-length</i> } [ <b>sub</b> ]	缺省情况下,未配置VLAN接口的IP地址
配置VLAN接口的描述信息	<b>description</b> <i>text</i>	缺省情况下,VLAN接口的描述信息为该VLAN接口的接口名,如“Vlan-interface1 Interface”
配置VLAN接口的MTU值	<b>mtu</b> <i>size</i>	缺省情况下VLAN接口的MTU值为1500字节
(可选)配置VLAN接口的期望带宽	<b>bandwidth</b> <i>bandwidth-value</i>	缺省情况下,接口的期望带宽=接口的波特率÷1000(kbps)
(可选)恢复VLAN接口的缺省配置	<b>default</b>	-
(可选)取消手工关闭VLAN接口	<b>undo shutdown</b>	-

## 1.4 配置基于端口的VLAN

### 1.4.1 基于端口的VLAN简介

基于端口划分 VLAN 是最简单、最有效的 VLAN 划分方法。它按照设备端口来定义 VLAN 成员,将指定端口加入到指定 VLAN 中之后,该端口就可以转发该 VLAN 的报文。

用户可以配置端口的链路类型及缺省 VLAN,其中,链路类型决定了端口能否加入多个 VLAN。

#### 1. 端口的链路类型

端口的链路类型分为三种,不同链路类型的端口在转发报文时对 VLAN Tag 的处理方式不同:

- **Access:** 端口只能发送一个 VLAN 的报文,发出去的报文不带 VLAN Tag。一般用于和不能识别 VLAN Tag 的用户终端设备相连,或者不需要区分不同 VLAN 成员时使用。
- **Trunk:** 端口能发送多个 VLAN 的报文,发出去的端口缺省 VLAN 的报文不带 VLAN Tag,其他 VLAN 的报文都必须带 VLAN Tag。通常用于网络传输设备之间的互连。
- **Hybrid:** 端口能发送多个 VLAN 的报文,端口发出去的报文可根据需要配置某些 VLAN 的报文带 VLAN Tag,某些 VLAN 的报文不带 VLAN Tag。在一些应用场景下,需要使用 Hybrid 端口的功能。比如在 1:2 VLAN 映射中,服务提供商网络的多个 VLAN 的报文在进入用户网络前,需要剥离外层 VLAN Tag,此时 Trunk 端口不能实现该功能,因为 Trunk 端口只能使该端口缺省 VLAN 的报文不带 VLAN Tag 通过。有关 1:2 VLAN 映射的详细介绍,请参见“二层技术-以太网交换配置指导”中的“VLAN 映射”。

## 2. 端口缺省VLAN

除了可以配置端口允许通过的 VLAN 外,还可以配置端口的缺省 VLAN,即端口 VLAN ID(Port VLAN ID, PVID)。当端口收到 Untagged 报文时,会认为该报文所属的 VLAN 为缺省 VLAN。

- Access 端口的缺省 VLAN 就是它所在的 VLAN。
- Trunk 端口和 Hybrid 端口可以允许多个 VLAN 通过,能够配置端口缺省 VLAN。
- 当执行 **undo vlan** 命令删除的 VLAN 是某个端口的缺省 VLAN 时,对 Access 端口,端口的缺省 VLAN 会恢复到 VLAN 1;对 Trunk 或 Hybrid 端口,端口的缺省 VLAN 配置不会改变,即它们可以使用已经不存在的 VLAN 作为端口缺省 VLAN。



### 说明

- 建议本端设备端口的缺省 VLAN ID 和相连的对端设备端口的缺省 VLAN ID 保持一致。
- 建议保证端口的缺省 VLAN 为端口允许通过的 VLAN。如果端口不允许某 VLAN 通过,但是端口的缺省 VLAN 为该 VLAN,则端口会丢弃收到的该 VLAN 的报文或者不带 VLAN Tag 的报文。

## 3. 端口对报文的处理方式

在配置了端口链路类型和端口缺省VLAN后,端口对报文的接收和发送的处理有几种不同情况,具体情况请参见 [表 1-3](#)。

表1-3 不同链路类型端口收发报文的差异

端口类型	对接收报文的处理		对发送报文的处理
	当接收到的报文不带 Tag 时	当接收到的报文带有 Tag 时	
Access端口	为报文添加端口缺省VLAN的Tag	<ul style="list-style-type: none"><li>• 当报文的 VLAN ID 与端口的缺省 VLAN ID 相同时,接收该报文</li><li>• 当报文的 VLAN ID 与端口的缺省 VLAN ID 不同时,丢弃该报文</li></ul>	去掉Tag,发送该报文
Trunk端口	<ul style="list-style-type: none"><li>• 当端口的缺省 VLAN ID 在端口允许通过的 VLAN ID 列表里时,接收该报文,给报文添加端口缺省 VLAN 的 Tag</li></ul>	<ul style="list-style-type: none"><li>• 当报文的 VLAN ID 在端口允许通过的 VLAN ID 列表里时,接收该报文</li><li>• 当报文的 VLAN ID 不在端口允许通过的 VLAN ID 列表里时,丢弃该报文</li></ul>	<ul style="list-style-type: none"><li>• 当报文的 VLAN ID 与端口的缺省 VLAN ID 相同,且是该端口允许通过的 VLAN ID 时:去掉 Tag,发送该报文</li><li>• 当报文的 VLAN ID 与端口的缺省 VLAN ID 不同,且是该端口允许通过的 VLAN ID 时:保持原有 Tag,发送该报文</li></ul>
Hybrid端口	<ul style="list-style-type: none"><li>• 当端口的缺省 VLAN ID 不在端口允许通过的 VLAN ID 列表里时,丢弃该报文</li></ul>		当报文的VLAN ID是端口允许通过的VLAN ID时,发送该报文,并可以通过 <b>port hybrid vlan</b> 命令配置端口在发送该VLAN的报文时是否携带Tag



说明

缺省情况下，对于 Untagged 报文的处理顺序为：基于 MAC 的 VLAN->基于 IP 子网的 VLAN->基于协议的 VLAN->基于端口的 VLAN。

## 1.4.2 配置基于Access端口的VLAN

配置基于 Access 端口的 VLAN 有两种方法：一种是在 VLAN 视图下进行配置，另一种是在接口视图下进行配置。

表1-4 配置基于 Access 端口的 VLAN（在 VLAN 视图下）

配置	命令	说明
进入系统视图	<b>system-view</b>	-
进入VLAN视图	<b>vlan <i>vlan-id</i></b>	-
向当前VLAN中添加一个或一组Access端口	<b>port <i>interface-list</i></b>	缺省情况下，系统将所有端口都加入到VLAN 1

表1-5 配置基于 Access 端口的 VLAN（在接口视图下）

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入相应视图	二层以太网接口视图 <b>interface <i>interface-type</i> <i>interface-number</i></b>	-
	二层聚合接口视图 <b>interface bridge-aggregation <i>interface-number</i></b>	
配置端口的链路类型为Access类型	<b>port link-type access</b>	缺省情况下，端口的链路类型为Access
将Access端口加入到指定VLAN	<b>port access vlan <i>vlan-id</i></b>	缺省情况下，所有Access端口都属于VLAN 1 在将Access端口加入到指定VLAN之前，该VLAN必须已经存在

## 1.4.3 配置基于Trunk端口的VLAN

Trunk 端口可以允许多个 VLAN 通过，只能在接口视图下进行配置。

配置基于 Trunk 端口的 VLAN 时，需要注意：

- Trunk 端口不能直接切换为 Hybrid 端口，只能先将 Trunk 端口配置为 Access 端口，再配置为 Hybrid 端口。
- 配置端口缺省 VLAN 后，必须使用 **port trunk permit vlan** 命令配置允许端口缺省 VLAN 的报文通过，接口才能转发端口缺省 VLAN 的报文。

表1-6 配置基于 Trunk 端口的 VLAN

操作		命令	说明
进入系统视图		<b>system-view</b>	-
进入相应视图	二层以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
	二层聚合接口视图	<b>interface bridge-aggregation</b> <i>interface-number</i>	
配置端口的链路类型为Trunk类型		<b>port link-type trunk</b>	缺省情况下，端口的链路类型为Access类型
允许指定的VLAN通过当前Trunk端口		<b>port trunk permit vlan</b> { <i>vlan-id-list</i>   <b>all</b> }	缺省情况下，Trunk端口只允许VLAN 1的报文通过
(可选) 配置Trunk端口的缺省VLAN		<b>port trunk pvid vlan</b> <i>vlan-id</i>	缺省情况下，Trunk端口的缺省VLAN为VLAN 1

#### 1.4.4 配置基于Hybrid端口的VLAN

Hybrid 端口可以允许多个 VLAN 通过，只能在接口视图下进行配置。

配置基于 Hybrid 端口的 VLAN 时，需要注意：

- Hybrid 端口不能直接切换为 Trunk 端口，只能先将 Hybrid 端口配置为 Access 端口，再配置为 Trunk 端口。
- 在配置允许指定的 VLAN 通过 Hybrid 端口之前，允许通过的 VLAN 必须已经存在。
- 配置端口缺省 VLAN 后，必须使用 **port hybrid vlan** 命令配置允许端口缺省 VLAN 的报文通过，出接口才能转发端口缺省 VLAN 的报文。

表1-7 配置基于 Hybrid 端口的 VLAN

操作		命令	说明
进入系统视图		<b>system-view</b>	-
进入相应视图	二层以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
	二层聚合接口视图	<b>interface bridge-aggregation</b> <i>interface-number</i>	
配置端口的链路类型为Hybrid类型		<b>port link-type hybrid</b>	缺省情况下，端口的链路类型为Access类型
允许指定的VLAN通过当前Hybrid端口		<b>port hybrid vlan</b> <i>vlan-id-list</i> { <b>tagged</b>   <b>untagged</b> }	缺省情况下，Hybrid端口只允许该端口在链路类型为Access时的所属VLAN的报文以Untagged方式通过
(可选)配置Hybrid端口的缺省VLAN		<b>port hybrid pvid vlan</b> <i>vlan-id</i>	缺省情况下，Hybrid端口的缺省VLAN为该端口在链路类型为Access时的所属VLAN

## 1.5 配置基于MAC的VLAN



说明

基于 MAC 的 VLAN 只对 Hybrid 端口配置有效。

### 1.5.1 基于MAC的VLAN简介

VLAN 最常用的划分方式是基于端口划分，该方式为从同一端口进入的 **Untagged** 报文添加相同的 **VLAN Tag**，在同一 VLAN 内进行转发处理。该方式配置简单，适用于终端设备物理位置比较固定的组网环境。随着移动办公和无线接入的普及，终端设备不再通过固定端口接入设备，它可能本次使用端口 A 接入网络，下次使用端口 B 接入网络。如果端口 A 和端口 B 的 VLAN 配置不同，则终端设备第二次接入后就会被划分到另一 VLAN，导致无法使用原 VLAN 内的资源；如果端口 A 和端口 B 的 VLAN 配置相同，当端口 B 被分配给别的终端设备时，又会引入安全问题。如何在这样灵活多变的网络环境中部署 VLAN 呢？基于 MAC 的 VLAN 应运而生。

基于 MAC 的 VLAN 是根据报文的源 MAC 地址来划分 VLAN。设备维护的 MAC VLAN 表记录了 MAC 地址和 VLAN 的对应关系。这种划分方法的最大优点就是当用户物理位置发生变化，VLAN 不用重新配置。所以这种根据 MAC 地址的划分方法也称为基于用户的 VLAN。

#### 1. 手动配置静态MAC VLAN

手动配置静态 MAC VLAN 常用于 VLAN 中用户相对较少的网络环境。在该方式下，用户需要手动配置 MAC VLAN 表项，开启基于 MAC 地址的 VLAN 功能，并将端口加入 MAC VLAN。其原理为：

- 当端口收到的报文为 **Untagged** 报文时，根据报文的源 MAC 地址匹配 MAC VLAN 表项。首先进行模糊匹配，即查询 MAC VLAN 表中掩码不是全 F 的表项，将源 MAC 地址和掩码相与运算后与 MAC VLAN 表项中的 MAC 地址匹配，如果完全相同，则模糊匹配成功，给报文添加表项中对应的 **VLAN Tag** 并转发该报文；如果模糊匹配失败，则进行精确匹配，即查询表中掩码为全 F 的表项。如果报文中的源 MAC 地址与某 MAC VLAN 表项中的 MAC 地址完全相同，则精确匹配成功，给报文添加表项中对应的 **VLAN Tag** 并转发该报文；如果没有找到匹配的 MAC VLAN 表项，则继续按照其他原则（基于 IP 子网的 VLAN、基于协议的 VLAN、基于端口的 VLAN）确定报文所属的 VLAN，给报文添加对应的 **VLAN Tag** 并转发该报文。
- 当端口收到的报文为 **Tagged** 报文时，如果报文的 VLAN ID 在该端口允许通过的 VLAN ID 列表里，则转发该报文；否则丢弃该报文。

#### 2. 动态触发端口加入静态MAC VLAN

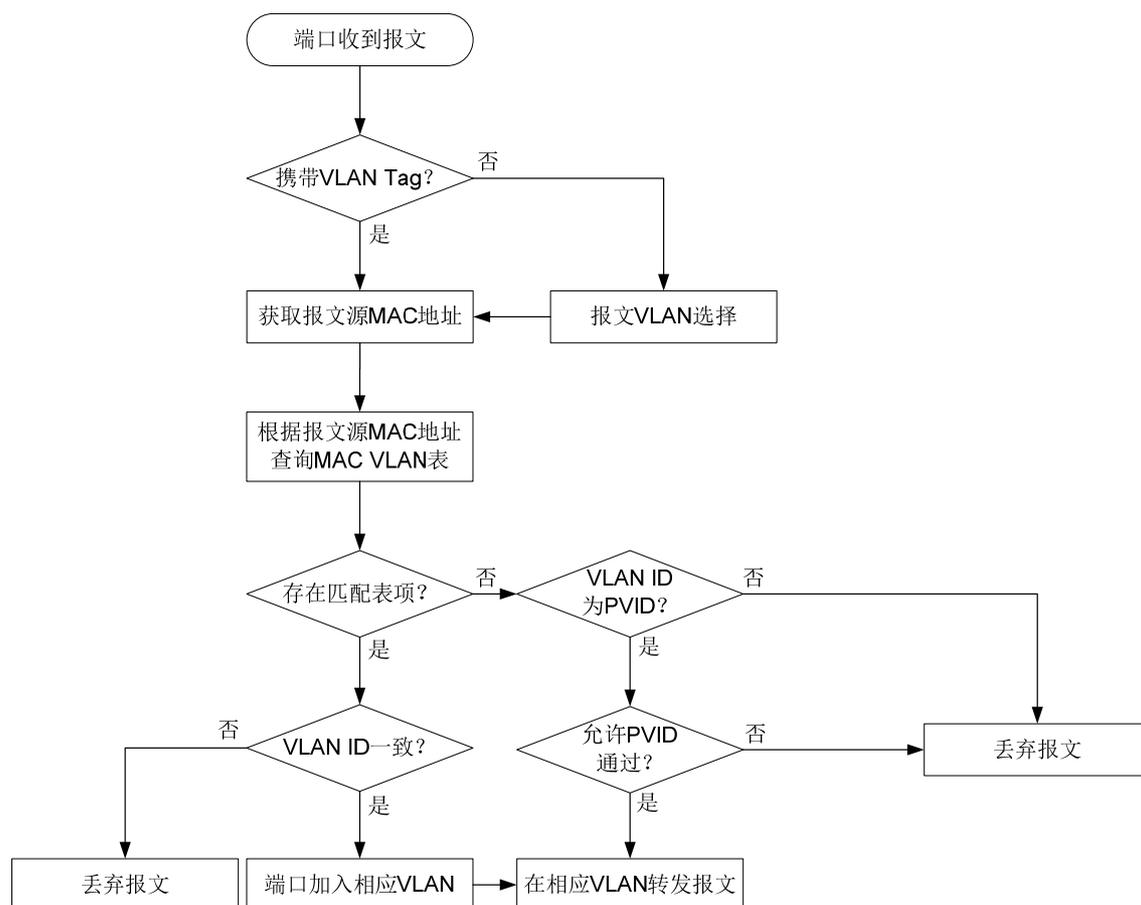
手动配置静态 MAC VLAN 时，如果不能确定从哪些端口收到指定 VLAN 的报文，就不能把相应端口加入到 MAC VLAN。此时可以采用动态触发端口加入静态 MAC VLAN 的方式。在该方式下，配置 MAC VLAN 表项后，需要在端口上开启基于 MAC 的 VLAN 功能和 MAC VLAN 的动态触发功能，不需要手动把端口加入 MAC VLAN。

配置动态触发端口加入静态 MAC VLAN 后，端口在收到报文时，首先判断报文是否携带 **VLAN Tag**，若带 **VLAN Tag**，则直接获取报文源 MAC 地址；若不带 **VLAN Tag**，则先进行报文 VLAN 选择（按照基于 MAC 的 VLAN->基于 IP 子网的 VLAN->基于协议的 VLAN->基于端口的 VLAN 的优先次序

为该 Untagged 报文添加对应的 VLAN Tag，并获取该 VLAN Tag)，再获取报文源 MAC 地址，然后根据报文的源 MAC 地址和 VLAN 查询静态 MAC VLAN 表项：

- 如果报文源 MAC 地址与 MAC VLAN 表项中的 MAC 地址精确匹配，再检查报文的 VLAN ID 是否与对应表项中的 VLAN ID 一致，若一致，通过该报文动态触发端口加入相应 VLAN，同时转发该报文；否则丢弃该报文。
- 如果报文源 MAC 地址与 MAC VLAN 表项的 MAC 地址不精确匹配，当报文 VLAN ID 为 PVID，判断端口是否允许报文在 PVID 内转发，若允许，则在 PVID 中转发该报文，否则丢弃该报文。当报文 VLAN ID 不为 PVID，则丢弃该报文。处理流程如 图 1-3 所示：

图1-3 动态触发端口加入静态 MAC VLAN 的处理



 说明

- 在端口自动加入 MAC VLAN 表项中相应的 VLAN 时，若端口此前未配置允许该 VLAN 通过，则端口自动以 Untagged 方式加入该 VLAN；若端口此前已配置允许该 VLAN 通过，则不改变原有配置。
- 如果用户在同一端口上同时开启了手动配置 MAC VLAN 和动态触发端口加入 MAC VLAN，此时该端口选择使用后者的功能。
- 当端口收到的报文与 MAC VLAN 表项匹配，对该报文进行转发时，根据 MAC VLAN 的优先级（MAC 地址对应 VLAN 的 802.1p 优先级）高低来决定报文传输的优先程度。

### 3. 动态MAC VLAN

动态 MAC VLAN 是由接入认证过程来动态决定接入用户报文所属的 VLAN。该功能需要和接入认证功能（比如端口接入控制方式为 MAC-based 的 802.1X）配合使用，以实现终端的安全、灵活接入。在设备上配置动态 MAC VLAN 功能以后，还需要在接入认证服务器上配置用户名和 VLAN 的绑定关系。

如果用户发起认证请求，接入认证服务器先对用户名和密码进行验证，如果验证通过，服务器下发 VLAN 信息。此时设备根据请求报文的源 MAC 地址和下发的 VLAN 信息生成动态 MAC VLAN 表项（要求与已有的静态 MAC VLAN 表项不能冲突），并将 MAC VLAN 添加到端口允许通过的 VLAN 列表中。用户下线后，设备自动删除 MAC VLAN 表项，并将 MAC VLAN 从端口允许通过的 VLAN 列表中删除。

有关接入认证功能的详细介绍请参见“安全配置指导”中的“802.1X”和“MAC 地址认证”。

## 1.5.2 配置基于MAC的VLAN

配置基于 MAC 的 VLAN 时，需要注意：

- Super VLAN 不能作为 MAC VLAN 表项中的 VLAN。
- 基于 MAC 的 VLAN 功能主要在用户的接入设备的下行端口上进行配置，不能和聚合功能同时使用。

### 1. 手动配置静态MAC VLAN

表1-8 手动配置静态 MAC VLAN

操作	命令	说明
进入系统视图	<b>system-view</b>	-
配置MAC VLAN表项	<b>mac-vlan mac-address mac-address [ mask mac-mask ] vlan vlan-id [ dot1q priority ]</b>	缺省情况下，不存在MAC VLAN表项
进入二层以太网接口视图	<b>interface interface-type interface-number</b>	-
配置端口的链路类型为Hybrid类型	<b>port link-type hybrid</b>	缺省情况下，所有端口的链路类型均为Access类型
允许基于MAC的VLAN通过当前Hybrid端口	<b>port hybrid vlan vlan-id-list { tagged   untagged }</b>	缺省情况下，Hybrid端口只允许该端口在链路类型为Access时的所属VLAN的报文以Untagged方式通过
开启MAC VLAN功能	<b>mac-vlan enable</b>	缺省情况下，MAC VLAN功能处于关闭状态
（可选）配置接口优先根据MAC地址来匹配VLAN	<b>vlan precedence mac-vlan</b>	缺省情况下，对于基于MAC的VLAN和基于IP子网的VLAN，优先根据MAC地址来匹配VLAN

### 2. 配置动态触发端口加入静态MAC VLAN

配置动态触发端口加入静态 MAC VLAN 时，需要注意：

- 不建议 MAC VLAN 的动态触发功能和 802.1X/MAC 地址认证功能同时使用，否则会影响 802.1X/MAC 地址认证功能的正常工作。

- 开启 MAC VLAN 的动态触发功能后，MAC 地址禁止学习功能不生效。
- MAC VLAN 的动态触发功能生效的一个前提是，源 MAC 地址所匹配的 VLAN 必须是静态 VLAN（本地手工创建的 VLAN）。
- 配置 MSTP 情况下，如果端口在要加入的 VLAN 对应的 MSTP 实例中是阻塞状态，则端口会丢弃收到的报文，造成 MAC 地址不能上送，不能完成动态触发端口加入静态 MAC VLAN，因此不建议本功能和多实例 MSTP 同时使用。
- 配置 PVST 情况下，如果端口要加入的 VLAN 不为端口允许通过的 VLAN，则端口处于阻塞状态，会丢弃收到的报文，造成 MAC 地址不能上送，不能完成动态触发端口加入静态 MAC VLAN，因此不建议本功能和 PVST 同时使用。
- 当端口配置了自动模式下的 Voice VLAN，又配置本功能时，两个功能可能会相互影响，导致其中某个功能不可用。当端口同时配置了本功能和自动模式下的 Voice VLAN，再取消其中任何一个功能的配置，会导致另一个功能不可用。因此不建议同一端口同时配置本功能和自动模式下的 Voice VLAN。

表1-9 配置动态触发端口加入静态 MAC VLAN

操作	命令	说明
进入系统视图	<b>system-view</b>	-
配置MAC VLAN表项	<b>mac-vlan mac-address mac-address vlan vlan-id [ dot1q priority ]</b>	缺省情况下，不存在MAC VLAN表项
进入二层以太网接口视图	<b>interface interface-type interface-number</b>	-
配置端口的链路类型为Hybrid类型	<b>port link-type hybrid</b>	缺省情况下，所有端口的链路类型均为Access类型
开启MAC VLAN功能	<b>mac-vlan enable</b>	缺省情况下，MAC VLAN功能处于关闭状态
开启MAC VLAN的动态触发功能	<b>mac-vlan trigger enable</b>	缺省情况下，MAC VLAN的动态触发功能处于关闭状态 开启MAC VLAN的动态触发功能后，只有端口接收的报文的源MAC地址精确匹配了MAC VLAN表项，才会动态触发该端口加入相应VLAN
（可选）配置接口优先根据MAC地址来匹配VLAN	<b>vlan precedence mac-vlan</b>	缺省情况下，对于基于MAC的VLAN和基于IP子网的VLAN，优先根据MAC地址来匹配VLAN
（可选）配置当报文源MAC地址与MAC VLAN表项的MAC地址未精确匹配时，禁止该报文在PVID内转发	<b>port pvid forbidden</b>	缺省情况下，当报文源MAC地址与MAC VLAN表项的MAC地址未精确匹配时，允许该报文在PVID内转发

### 3. 配置动态MAC VLAN

表1-10 配置动态 MAC VLAN

操作		命令	说明
进入系统视图		<b>system-view</b>	-
进入二层以太网接口视图		<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
配置端口的链路类型为Hybrid类型		<b>port link-type hybrid</b>	缺省情况下，所有端口的链路类型均为Access类型
允许基于MAC的VLAN通过当前Hybrid端口		<b>port hybrid vlan</b> <i>vlan-id-list</i> { <b>tagged</b>   <b>untagged</b> }	缺省情况下，Hybrid端口只允许该端口在链路类型为Access时的所属VLAN的报文以Untagged方式通过
开启MAC VLAN功能		<b>mac-vlan enable</b>	缺省情况下，MAC VLAN功能处于关闭状态
配置接入认证功能	配置802.1X	请参见“安全命令参考”中的“802.1X”	二者至少选其一
	配置MAC地址认证	请参见“安全命令参考”中的“MAC地址认证”	

## 1.6 配置基于IP子网的VLAN



说明

基于IP子网的VLAN只对Hybrid端口配置有效，只对Untagged报文应用。

### 1.6.1 基于IP子网的VLAN简介

基于IP子网的VLAN（简称子网VLAN）是根据报文源IP地址及子网掩码来进行划分的。设备从端口收到Untagged报文后，会根据报文的源IP地址来确定报文所属的VLAN，然后将报文自动划分到指定VLAN中传输。

此特性主要用于将指定网段或IP地址的报文划分到指定的VLAN中传送。

### 1.6.2 配置基于IP子网的VLAN

表1-11 配置基于IP子网的VLAN

操作		命令	说明
进入系统视图		<b>system-view</b>	-
进入VLAN视图		<b>vlan</b> <i>vlan-id</i>	-
配置VLAN与指定的IP子网或IP地址关联		<b>ip-subnet-vlan</b> [ <i>ip-subnet-index</i> ] <b>ip</b> <i>ip-address</i> [ <i>mask</i> ]	缺省情况下，VLAN未关联IP子网或IP地址 VLAN关联的IP网段或IP地址不允许是组播网段或组播地址

操作		命令	说明
退回系统视图		<b>quit</b>	-
进入相应视图	二层以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
	二层聚合接口视图	<b>interface bridge-aggregation</b> <i>interface-number</i>	
配置端口的链路类型为 Hybrid 类型		<b>port link-type hybrid</b>	缺省情况下，所有端口的链路类型均为 Access 类型
允许子网 VLAN 通过当前端口		<b>port hybrid vlan</b> <i>vlan-id-list</i> { <b>tagged</b>   <b>untagged</b> }	缺省情况下，Hybrid 端口只允许该端口在链路类型为 Access 时的所属 VLAN 的报文以 Untagged 方式通过
配置端口与子网 VLAN 关联		<b>port hybrid ip-subnet-vlan</b> <i>vlan</i> <i>vlan-id</i>	缺省情况下，端口未关联子网 VLAN

## 1.7 配置基于协议的 VLAN



说明

基于协议的 VLAN 只对 Hybrid 端口配置有效，只对 Untagged 报文应用。

### 1.7.1 基于协议的 VLAN 简介

基于协议的 VLAN（简称协议 VLAN）是根据端口接收到的报文所属的协议（族）类型以及封装格式来给报文分配不同的 VLAN ID。可用来划分 VLAN 的协议有 IP、IPX、AT（AppleTalk，Apple 计算机网络协议）等，封装格式有 Ethernet II、802.3 raw、802.2 LLC、802.2 SNAP 等。

此特性主要应用于将网络中提供的服务类型与 VLAN 相关联，方便管理和维护。

协议 VLAN 由协议模板定义。协议模板是用来匹配报文所属协议类型的标准，由“协议类型+封装格式”组成。对于一个协议 VLAN 来说，其绑定的多个协议模板用协议索引（*protocol-index*）来区分；对于不同的协议 VLAN 来说，其绑定的协议模板用协议 *vlan-id* 和 *protocol-index* 来唯一标识。最后通过命令行将协议 VLAN 中的协议模板与端口绑定。

当端口收到 Untagged 报文时，如果该报文携带的协议类型和封装格式与某协议模板相匹配，则为其添加该协议模板绑定的协议 *vlan-id* 的 VLAN Tag，否则为其添加 PVID 的 VLAN Tag。

### 1.7.2 配置基于协议的 VLAN

需要注意的是，协议 VLAN 特性要求 Hybrid 入端口的报文格式为 Untagged 的，而自动模式下的 Voice VLAN 只支持 Hybrid 端口对 Tagged 的语音流进行处理，因此，不能将某个 VLAN 同时配置为协议 VLAN 和 Voice VLAN。

表1-12 配置基于协议的 VLAN

操作		命令	说明
进入系统视图		<b>system-view</b>	-
进入VLAN视图		<b>vlan</b> <i>vlan-id</i>	-
配置VLAN与指定的协议模板关联		<b>protocol-vlan</b> [ <i>protocol-index</i> ] { <b>at</b>   <b>ipv4</b>   <b>ipv6</b>   <b>ipx</b> { <b>ethernetii</b> / <b>llc</b>   <b>raw</b> / <b>snap</b> }   <b>mode</b> { <b>ethernetii</b> <b>etype</b> <i>etype-id</i>   <b>llc</b> { <b>dsap</b> <i>dsap-id</i> [ <b>ssap</b> <i>ssap-id</i> ]   <b>ssap</b> <i>ssap-id</i> }   <b>snap</b> <b>etype</b> <i>etype-id</i> }	缺省情况下，当前VLAN未关联协议模板
退出VLAN视图		<b>quit</b>	-
进入相应视图	二层以太网接口视图	<b>interface</b> <i>interface-type interface-number</i>	-
	二层聚合接口视图	<b>interface bridge-aggregation</b> <i>interface-number</i>	
配置端口的链路类型为Hybrid类型		<b>port link-type hybrid</b>	缺省情况下，所有端口的链路类型均为Access类型
允许协议VLAN通过当前端口		<b>port hybrid vlan</b> <i>vlan-id-list</i> { <b>tagged</b>   <b>untagged</b> }	缺省情况下，Hybrid端口只允许该端口在链路类型为Access时的所属VLAN的报文以Untagged方式通过
配置端口与协议VLAN关联		<b>port hybrid protocol-vlan</b> <b>vlan</b> <i>vlan-id</i> { <i>protocol-index</i> [ <b>to</b> <i>protocol-end</i> ]   <b>all</b> }	缺省情况下，端口未关联协议VLAN

## 1.8 VLAN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 VLAN 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 VLAN 接口统计信息。

表1-13 VLAN 显示和维护

操作	命令
显示VLAN接口相关信息	<b>display interface vlan-interface</b> [ <i>interface-number</i> ] [ <b>brief</b> [ <b>description</b>   <b>down</b> ] ]
显示端口关联的子网VLAN的信息	<b>display ip-subnet-vlan interface</b> { <i>interface-type interface-number1</i> [ <b>to</b> <i>interface-type interface-number2</i> ]   <b>all</b> }
显示指定的或所有子网VLAN的信息	<b>display ip-subnet-vlan vlan</b> { <i>vlan-id1</i> [ <b>to</b> <i>vlan-id2</i> ]   <b>all</b> }
显示端口关联的协议VLAN的信息	<b>display protocol-vlan interface</b> { <i>interface-type interface-number1</i> [ <b>to</b> <i>interface-type interface-number2</i> ]   <b>all</b> }
显示指定的或所有协议VLAN的信息	<b>display protocol-vlan vlan</b> { <i>vlan-id1</i> [ <b>to</b> <i>vlan-id2</i> ]   <b>all</b> }
显示VLAN相关信息	<b>display vlan</b> [ <i>vlan-id1</i> [ <b>to</b> <i>vlan-id2</i> ] ]   <b>all</b>   <b>dynamic</b>   <b>reserved</b>   <b>static</b> ]

操作	命令
显示设备上所有已创建VLAN的概要信息	<b>display vlan brief</b>
显示设备上存在的Hybrid或Trunk端口	<b>display port { hybrid   trunk }</b>
清除VLAN接口的统计信息	<b>reset counters interface vlan-interface [ interface-number ]</b>

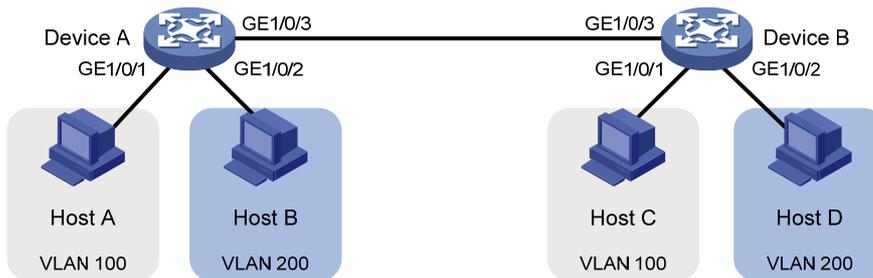
## 1.9 基于端口的VLAN典型配置举例

### 1. 组网需求

- Host A 和 Host C 属于部门 A，但是通过不同的设备接入公司网络；Host B 和 Host D 属于部门 B，也通过不同的设备接入公司网络。
- 为了通信的安全性，也为了避免广播报文泛滥，公司网络中使用 VLAN 技术来隔离部门间的二层流量。其中部门 A 使用 VLAN 100，部门 B 使用 VLAN 200。

### 2. 组网图

图1-4 基于端口的 VLAN 组网图



### 3. 配置步骤

#### (1) 配置 Device A

# 创建 VLAN 100，并将 GigabitEthernet1/0/1 加入 VLAN 100。

```
<DeviceA> system-view
[DeviceA] vlan 100
[DeviceA-vlan100] port gigabitethernet 1/0/1
[DeviceA-vlan100] quit
```

# 创建 VLAN 200，并将 GigabitEthernet1/0/2 加入 VLAN 200。

```
[DeviceA] vlan 200
[DeviceA-vlan200] port gigabitethernet 1/0/2
[DeviceA-vlan200] quit
```

# 为了使 Device A 上 VLAN 100 和 VLAN 200 的报文能发送给 Device B，将 GigabitEthernet1/0/3 的链路类型配置为 Trunk，并允许 VLAN 100 和 VLAN 200 的报文通过。

```
[DeviceA] interface gigabitethernet 1/0/3
[DeviceA-GigabitEthernet1/0/3] port link-type trunk
[DeviceA-GigabitEthernet1/0/3] port trunk permit vlan 100 200
```

- (2) Device B 上的配置与 Device A 上的配置相同，不再赘述。
- (3) 将 Host A 和 Host C 配置在一个网段，比如 192.168.100.0/24；将 Host B 和 Host D 配置在一个网段，比如 192.168.200.0/24。

#### 4. 验证配置

- (1) Host A 和 Host C 能够互相 ping 通，但是均不能 ping 通 Host B 和 Host D。Host B 和 Host D 能够互相 ping 通，但是均不能 ping 通 Host A 和 Host C。
- (2) 通过查看显示信息验证配置是否成功。

# 查看 Device A 上 VLAN 100 和 VLAN 200 的配置信息，验证以上配置是否生效。

```
[DeviceA-GigabitEthernet1/0/3] display vlan 100
```

```
VLAN ID: 100
VLAN type: Static
Route interface: Not configured
Description: VLAN 0100
Name: VLAN 0100
```

```
Tagged ports:
```

```
  GigabitEthernet1/0/3
```

```
Untagged ports:
```

```
  GigabitEthernet1/0/1
```

```
[DeviceA-GigabitEthernet1/0/3] display vlan 200
```

```
VLAN ID: 200
VLAN type: Static
Route interface: Not configured
Description: VLAN 0200
Name: VLAN 0200
```

```
Tagged ports:
```

```
  GigabitEthernet1/0/3
```

```
Untagged ports:
```

```
  GigabitEthernet1/0/2
```

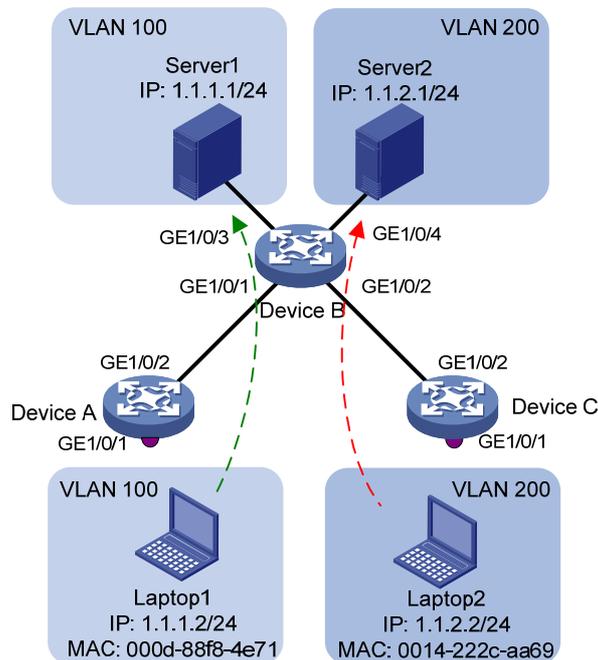
## 1.10 基于MAC的VLAN典型配置举例

### 1. 组网需求

- 如下图所示，Device A 和 Device C 的 GigabitEthernet1/0/1 端口分别连接到两个会议室，Laptop1 和 Laptop2 是会议用笔记本电脑，会在两个会议室间移动使用。
- Laptop1 和 Laptop2 分别属于两个部门，两个部门间使用 VLAN 100 和 VLAN 200 进行隔离。现要求这两台笔记本电脑无论在哪个会议室使用，均只能访问自己部门的服务器，即 Server1 和 Server2。

## 2. 组网图

图1-5 基于 MAC 的 VLAN 组网图



## 3. 配置步骤

### (1) Device A 的配置

# 创建 VLAN 100 和 VLAN 200。

```
<DeviceA> system-view
[DeviceA] vlan 100
[DeviceA-vlan100] quit
[DeviceA] vlan 200
[DeviceA-vlan200] quit
```

# 将 Laptop1 的 MAC 地址与 VLAN 100 关联，Laptop2 的 MAC 地址与 VLAN 200 关联。

```
[DeviceA] mac-vlan mac-address 000d-88f8-4e71 vlan 100
[DeviceA] mac-vlan mac-address 0014-222c-aa69 vlan 200
```

# 配置终端的接入端口：Laptop1 和 Laptop2 均可能从 GigabitEthernet1/0/1 接入，将 GigabitEthernet1/0/1 的端口类型配置为 Hybrid，并使其在发送 VLAN 100 和 VLAN 200 的报文时去掉 VLAN Tag；开启 GigabitEthernet1/0/1 端口的 MAC VLAN 功能。

```
[DeviceA] interface gigabitethernet 1/0/1
[DeviceA-GigabitEthernet1/0/1] port link-type hybrid
[DeviceA-GigabitEthernet1/0/1] port hybrid vlan 100 200 untagged
[DeviceA-GigabitEthernet1/0/1] mac-vlan enable
[DeviceA-GigabitEthernet1/0/1] quit
```

# 为了终端能够访问 Server1 和 Server2，需要将上行端口 GigabitEthernet1/0/2 的端口类型配置为 Trunk，并允许 VLAN 100 和 VLAN 200 的报文通过。

```
[DeviceA] interface gigabitethernet 1/0/2
[DeviceA-GigabitEthernet1/0/2] port link-type trunk
[DeviceA-GigabitEthernet1/0/2] port trunk permit vlan 100 200
```

```
[DeviceA-GigabitEthernet1/0/2] quit
```

## (2) Device B 的配置

# 创建 VLAN 100 和 VLAN 200，并将 GigabitEthernet1/0/3 加入 VLAN 100，GigabitEthernet1/0/4 加入 VLAN 200。

```
<DeviceB> system-view
[DeviceB] vlan 100
[DeviceB-vlan100] port gigabitethernet 1/0/3
[DeviceB-vlan100] quit
[DeviceB] vlan 200
[DeviceB-vlan200] port gigabitethernet 1/0/4
[DeviceB-vlan200] quit
```

# 配置 GigabitEthernet1/0/1 和 GigabitEthernet1/0/2 端口为 Trunk 端口，均允许 VLAN 100 和 VLAN 200 的报文通过。

```
[DeviceB] interface gigabitethernet 1/0/1
[DeviceB-GigabitEthernet1/0/1] port link-type trunk
[DeviceB-GigabitEthernet1/0/1] port trunk permit vlan 100 200
[DeviceB-GigabitEthernet1/0/1] quit
[DeviceB] interface gigabitethernet 1/0/2
[DeviceB-GigabitEthernet1/0/2] port link-type trunk
[DeviceB-GigabitEthernet1/0/2] port trunk permit vlan 100 200
[DeviceB-GigabitEthernet1/0/2] quit
```

## (3) Device C 的配置

Device C 的配置与 Device A 完全一致，这里不再赘述。

## 4. 显示与验证

(1) Laptop1 只能访问 Server1，不能访问 Server2；Laptop2 只能访问 Server2，不能访问 Server1。

(2) 在 Device A 和 Device C 上可以查看到 Laptop1 和 VLAN 100、Laptop2 和 VLAN 200 的静态 MAC VLAN 地址表项已经生成。以 Device A 为例：

```
[DeviceA] display mac-vlan all
The following MAC VLAN addresses exist:
S:Static D:Dynamic

```

MAC address	Mask	VLAN ID	Priority	State
000d-88f8-4e71	ffff-ffff-ffff	100	0	S
0014-222c-aa69	ffff-ffff-ffff	200	0	S

```
Total MAC VLAN address count: 2
```

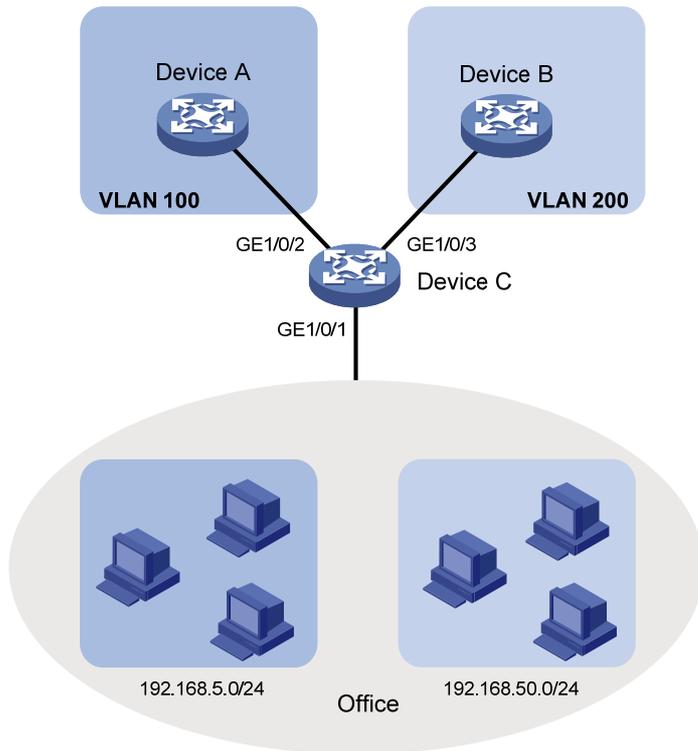
## 1.11 基于IP子网的VLAN典型配置举例

### 1. 组网需求

如下图所示，办公区的主机属于不同的网段 192.168.5.0/24 和 192.168.50.0/24，Device C 在收到来自办公区主机的报文时，根据报文的源 IP 地址，使来自不同网段主机的报文分别在指定的 VLAN 中传输，其中，来自网段 192.168.5.0/24 的报文在 VLAN 100 中传输，来自网段 192.168.50.0/24 的报文在 VLAN 200 中传输。

## 2. 组网图

图1-6 基于 IP 子网的 VLAN 组网图



## 3. 配置步骤

### (1) 配置 Device C

# 配置子网 192.168.5.0/24 与 VLAN 100 关联。

```
<DeviceC> system-view
[DeviceC] vlan 100
[DeviceC-vlan100] ip-subnet-vlan ip 192.168.5.0 255.255.255.0
[DeviceC-vlan100] quit
```

# 配置子网 192.168.50.0/24 与 VLAN 200 关联。

```
[DeviceC] vlan 200
[DeviceC-vlan200] ip-subnet-vlan ip 192.168.50.0 255.255.255.0
[DeviceC-vlan200] quit
```

# 配置端口 GigabitEthernet1/0/2 为 Hybrid 端口，允许 VLAN 100 通过，并且在发送 VLAN 100 的报文时携带 VLAN Tag。

```
[DeviceC] interface gigabitethernet 1/0/2
[DeviceC-GigabitEthernet1/0/2] port link-type hybrid
[DeviceC-GigabitEthernet1/0/2] port hybrid vlan 100 tagged
[DeviceC-GigabitEthernet1/0/2] quit
```

# 配置端口 GigabitEthernet1/0/3 为 Hybrid 端口，允许 VLAN 200 通过，并且在发送 VLAN 200 的报文时携带 VLAN Tag。

```
[DeviceC] interface gigabitethernet 1/0/3
[DeviceC-GigabitEthernet1/0/3] port link-type hybrid
[DeviceC-GigabitEthernet1/0/3] port hybrid vlan 200 tagged
```

```
[DeviceC-GigabitEthernet1/0/3] quit
```

# 配置端口 GigabitEthernet1/0/1 为 Hybrid 端口, 允许 VLAN 100、200 通过, 并且在发送 VLAN 100、200 的报文时不携带 VLAN Tag。

```
[DeviceC] interface gigabitethernet 1/0/1
```

```
[DeviceC-GigabitEthernet1/0/1] port link-type hybrid
```

```
[DeviceC-GigabitEthernet1/0/1] port hybrid vlan 100 200 untagged
```

# 配置端口 GigabitEthernet1/0/1 和基于 IP 子网的 VLAN 100、200 关联。

```
[DeviceC-GigabitEthernet1/0/1] port hybrid ip-subnet-vlan vlan 100
```

```
[DeviceC-GigabitEthernet1/0/1] port hybrid ip-subnet-vlan vlan 200
```

```
[DeviceC-GigabitEthernet1/0/1] quit
```

## (2) 配置 Device A 和 Device B

配置 Device A 和 Device B 允许对应 VLAN 通过, 配置过程略。

## 4. 显示与验证

# 查看所有子网 VLAN 的信息。

```
[DeviceC] display ip-subnet-vlan vlan all
```

```
VLAN ID: 100
```

Subnet index	IP address	Subnet mask
0	192.168.5.0	255.255.255.0

```
VLAN ID: 200
```

Subnet index	IP address	Subnet mask
0	192.168.50.0	255.255.255.0

# 查看端口 GigabitEthernet1/0/1 关联的子网 VLAN 的信息。

```
[DeviceC] display ip-subnet-vlan interface gigabitethernet 1/0/1
```

```
Interface: GigabitEthernet1/0/1
```

VLAN ID	Subnet index	IP address	Subnet mask	Status
100	0	192.168.5.0	255.255.255.0	Active
200	0	192.168.50.0	255.255.255.0	Active

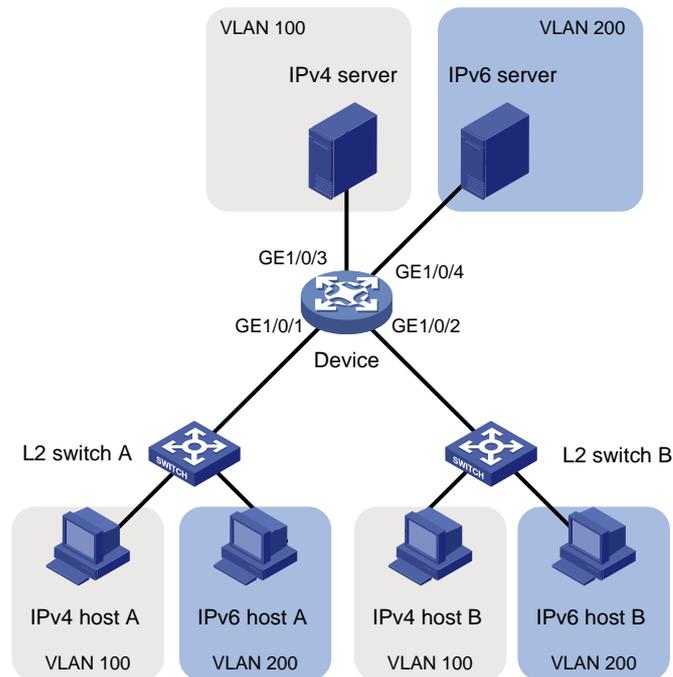
## 1.12 基于协议的VLAN典型配置举例

### 1. 组网需求

如下图所示, 实验室网络中大部分主机运行 IPv4 网络协议, 另外为了教学需要还部署了 IPv6 实验局, 因此, 有些主机运行 IPv6 网络协议。为了避免互相干扰, 将 VLAN 100 与 IPv4 协议、ARP 协议关联, VLAN 200 与 IPv6 协议关联, 通过协议 VLAN 将 IPv4 流量和 IPv6 流量二层互相隔离。

## 2. 组网图

图1-7 基于协议的 VLAN 组网图



## 3. 配置步骤

### (1) 配置 Device

# 创建 VLAN 100，将端口 GigabitEthernet1/0/3 加入 VLAN 100。

```
<Device> system-view
[Device] vlan 100
[Device-vlan100] description protocol VLAN for IPv4
[Device-vlan100] port gigabitethernet 1/0/3
[Device-vlan100] quit
```

# 创建 VLAN 200，将端口 GigabitEthernet1/0/4 加入 VLAN 200。

```
[Device] vlan 200
[Device-vlan200] description protocol VLAN for IPv6
[Device-vlan200] port gigabitethernet 1/0/4
```

# 将 IPv6 协议报文划分到 VLAN 200 中传输。

```
[Device-vlan200] protocol-vlan 1 ipv6
[Device-vlan200] quit
```

# 将 IPv4 协议报文和采用 Ethernet II 封装格式的 ARP 协议报文（ARP 报文对应的封装格式为 Ethernet II）划分到 VLAN 100 中传输。

```
[Device] vlan 100
[Device-vlan100] protocol-vlan 1 ipv4
[Device-vlan100] protocol-vlan 2 mode ethernetii etype 0806
[Device-vlan100] quit
```

# 配置端口 GigabitEthernet1/0/1 为 Hybrid 端口，允许 VLAN 100、200 通过，并且在发送 VLAN 100、200 的报文时不携带 VLAN Tag。

```
[Device] interface gigabitethernet 1/0/1
```

```
[Device-GigabitEthernet1/0/1] port link-type hybrid
```

```
[Device-GigabitEthernet1/0/1] port hybrid vlan 100 200 untagged
```

# 配置端口 GigabitEthernet1/0/1 与 VLAN 100 的协议模板 1（即 IPv4 协议模板）、协议模板 2（即 ARP 协议模板）和 VLAN 200 的协议模板 1（即 IPv6 协议模板）进行绑定。

```
[Device-GigabitEthernet1/0/1] port hybrid protocol-vlan vlan 100 1 to 2
```

```
[Device-GigabitEthernet1/0/1] port hybrid protocol-vlan vlan 200 1
```

```
[Device-GigabitEthernet1/0/1] quit
```

# 配置端口 GigabitEthernet1/0/2 为 Hybrid 端口，允许 VLAN 100、200 通过，并且在发送 VLAN 100、200 的报文时不携带 VLAN Tag。

```
[Device] interface gigabitethernet 1/0/2
```

```
[Device-GigabitEthernet1/0/2] port link-type hybrid
```

```
[Device-GigabitEthernet1/0/2] port hybrid vlan 100 200 untagged
```

# 配置端口 GigabitEthernet1/0/2 与 VLAN 100 的协议模板 1（即 IPv4 协议模板）、协议模板 2（即 ARP 协议模板）和 VLAN 200 的协议模板 1（即 IPv6 协议模板）进行绑定。

```
[Device-GigabitEthernet1/0/2] port hybrid protocol-vlan vlan 100 1 to 2
```

```
[Device-GigabitEthernet1/0/2] port hybrid protocol-vlan vlan 200 1
```

```
[Device-GigabitEthernet1/0/2] quit
```

(2) L2 switch A 和 L2 switch B 采用缺省配置

(3) 配置 Host 和 Server

将 IPv4 Host A、IPv4 Host B 和 IPv4 Server 配置在一个网段，比如 192.168.100.0/24；将 IPv6 Host A、IPv6 Host B 和 IPv6 Server 配置在一个网段，比如 2001::1/64。

#### 4. 显示与验证

(1) 通过 ping 命令查看

VLAN 100 内的主机和服务器能够互相 Ping 通；VLAN 200 内的主机和服务器能够互相 Ping 通。但 VLAN 100 内的主机/服务器与 VLAN 200 内的主机/服务器之间会 Ping 失败。

(2) 通过显示信息查看

# 查看所有协议 VLAN 的信息。

```
[Device] display protocol-vlan vlan all
```

```
VLAN ID: 100
```

```
Protocol index Protocol type
```

```
1 IPv4
```

```
2 Ethernet II Etype 0x0806
```

```
VLAN ID: 200
```

```
Protocol index Protocol type
```

```
1 IPv6
```

# 查看所有端口关联的协议 VLAN 的信息。

```
[Device] display protocol-vlan interface all
```

```
Interface: GigabitEthernet1/0/1
```

```
VLAN ID Protocol index Protocol type Status
```

```
100 1 IPv4 Active
```

```
100 2 Ethernet II Etype 0x0806 Active
```

```
200 1 IPv6 Active
```

Interface: GigabitEthernet1/0/2

VLAN ID	Protocol index	Protocol type	Status
100	1	IPv4	Active
100	2	Ethernet II Etype 0x0806	Active
200	1	IPv6	Active

# 2 Super VLAN

## 2.1 Super VLAN简介

在交换局域网中，VLAN 技术以其对广播域的灵活控制、部署方便而得到了广泛的应用。但是在一般的交换设备中，通常是采用一个 VLAN 对应一个 VLAN 接口的方式来实现广播域之间的互通，这在某些情况下导致了对 IP 地址的较大浪费。

Super VLAN 可以对 VLAN 进行聚合，从而大幅缩减实际需要的 VLAN 接口数量，解决 IP 地址紧张的问题。其原理是一个 Super VLAN 和多个 Sub VLAN 关联，关联的 Sub VLAN 公用 Super VLAN 对应的 VLAN 接口（即 Super VLAN interface）的 IP 地址作为三层通信的网关地址，此时 Sub VLAN 间的三层通信以及 Sub VLAN 与外部的三层通信均借用 Super VLAN interface 来实现，从而节省了 IP 地址资源。

- Super VLAN: 支持创建 VLAN 接口，并配置接口 IP 地址，不能加入物理接口。
- Sub VLAN: 不支持创建 VLAN 接口，可以加入物理端口，不同 Sub VLAN 之间二层相互隔离。

为了实现 Sub VLAN 之间的三层互通，在创建好 Super VLAN 及其 Super VLAN interface 之后，用户需要开启设备的本地代理功能：

- 对于 IPv4 网络环境，用户需要在 Super VLAN interface 上开启本地代理 ARP 功能，Super VLAN 利用本地代理 ARP，可以对 Sub VLAN 内用户发出的 ARP 请求和响应报文进行处理，从而实现 Sub VLAN 之间的三层互通。
- 对于 IPv6 网络环境，用户需要在 Super VLAN interface 上开启本地代理 ND 功能，Super VLAN 利用本地代理 ND，可以对 Sub VLAN 内用户发出的 NS 请求和 NA 响应报文进行处理，从而实现 Sub VLAN 之间的三层互通。

## 2.2 配置Super VLAN

Super VLAN 配置包括三个必选步骤：

- (1) 创建 Sub VLAN。
- (2) 创建 Super VLAN，并将 Super VLAN 和 Sub VLAN 关联起来。
- (3) 配置 Super VLAN interface。

### 1. 创建Sub VLAN

表2-1 创建 Sub VLAN

配置	命令	说明
进入系统视图	<b>system-view</b>	-
创建VLAN用作Sub VLAN	<b>vlan <i>vlan-id</i></b>	缺省情况下，系统只有一个缺省VLAN（VLAN 1）

### 2. 配置Super VLAN

配置 Super VLAN 功能时，需要注意：

- MAC VLAN 表项中的 VLAN 不能配为 Super VLAN。
- 如果某个 VLAN 被指定为 Super VLAN，则该 VLAN 不建议被指定为某个端口的 Guest VLAN/Auth-Fail VLAN/Critical VLAN；同样，如果某个 VLAN 被指定为某个端口的 Guest VLAN/Auth-Fail VLAN/Critical VLAN，则该 VLAN 不建议被指定为 Super VLAN。Guest VLAN/Auth-Fail VLAN/Critical VLAN 的相关内容请参见“安全配置指导”中的“802.1X”。
- 一个 VLAN 不能同时配置为 Super VLAN 和 Sub VLAN。
- 在 Super VLAN 下可以配置二层组播功能，但是由于 Super VLAN 中没有物理端口，该配置将不会生效。

表2-2 配置 Super VLAN

配置	命令	说明
进入系统视图	<b>system-view</b>	-
进入VLAN视图	<b>vlan</b> <i>vlan-id</i>	-
配置VLAN的类型为Super VLAN	<b>supervlan</b>	缺省情况下，VLAN类型不为Super VLAN
建立Super VLAN和Sub VLAN的映射关系	<b>subvlan</b> <i>vlan-id-list</i>	缺省情况下，不存在Super VLAN和Sub VLAN的映射关系 建立Super VLAN和Sub VLAN的映射关系前，指定的Sub VLAN必须已经创建

### 3. 配置Super VLAN interface

需要注意的是，在 Super VLAN interface 下配置 VRRP 功能后，会对网络性能造成影响，建议不要这样配置。VRRP 的详细描述请参见“可靠性配置指导”中的“VRRP”。

表2-3 配置 Super VLAN interface

配置	命令	说明
进入系统视图	<b>system-view</b>	-
创建VLAN接口，并进入VLAN接口视图	<b>interface</b> <i>vlan-interface</i> <i>interface-number</i>	<i>interface-number</i> 的值必须等于Super VLAN ID
配置VLAN接口的IP地址	<b>ip address</b> <i>ip-address</i> { <i>mask-length</i>   <i>mask</i> } [ <i>sub</i> ]	二者选其一
	<b>ipv6 address</b> { <i>ipv6-address</i> <i>prefix-length</i>   <i>ipv6-address/prefix-length</i> }	缺省情况下，没有配置VLAN接口的IP地址
开启本地代理ARP功能（IPv4环境）	<b>local-proxy-arp enable</b>	二者选其一 缺省情况下，本地代理ARP/ND功能处

配置	命令	说明
开启本地代理ND功能（IPv6环境）	<b>local-proxy-nd enable</b>	于关闭状态 <ul style="list-style-type: none"> <li>本地代理 ARP 功能的相关介绍请参见“三层技术-IP 业务配置指导”中的“代理 ARP”；<b>local-proxy-arp enable</b> 命令的相关描述请参见“三层技术-IP 业务命令参考”中的“代理 ARP”</li> <li>本地代理 ND 功能的相关介绍请参见“三层技术-IP 业务配置指导”中的“IPv6 基础”；<b>local-proxy-nd enable</b> 命令的相关描述请参见“三层技术-IP 业务命令参考”中的“IPv6 基础”</li> </ul>

## 2.3 Super VLAN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 Super VLAN 的运行情况，通过查看显示信息验证配置的效果。

表2-4 Super VLAN 显示和维护

操作	命令
显示Super VLAN及其关联的Sub VLAN的信息	<b>display supervlan</b> [ <i>supervlan-id</i> ]

## 2.4 Super VLAN典型配置举例

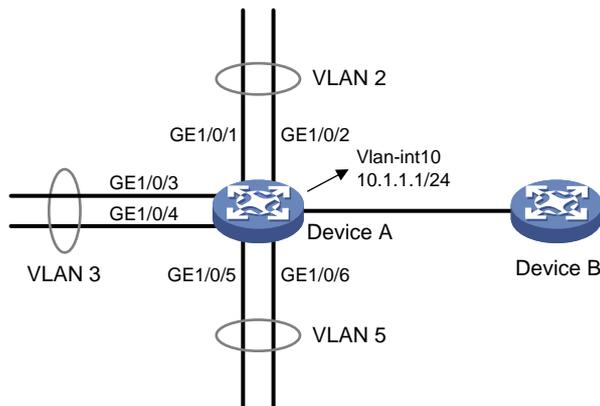
### 1. 组网需求

Device A 连接不同 VLAN 用户，其中，端口 GigabitEthernet1/0/1 和端口 GigabitEthernet1/0/2 属于 VLAN 2，端口 GigabitEthernet1/0/3 和端口 GigabitEthernet1/0/4 属于 VLAN 3，端口 GigabitEthernet1/0/5 和端口 GigabitEthernet1/0/6 属于 VLAN 5。

为实现 Device A 连接的各 VLAN 用户（均在 10.1.1.0/24 网段）之间能够满足二层隔离和三层互通的同时，节省 IP 资源，创建 Super VLAN，其关联的 Sub VLAN 公用 Super VLAN interface 的 IP 地址 10.1.1.1/24 作为三层通信的网关地址。

## 2. 组网图

图2-1 配置 Super VLAN 组网图



## 3. 配置步骤

# 创建 VLAN 10，配置 VLAN 接口的 IP 地址为 10.1.1.1/24。

```
<DeviceA> system-view
[DeviceA] vlan 10
[DeviceA-vlan10] quit
[DeviceA] interface vlan-interface 10
[DeviceA-Vlan-interface10] ip address 10.1.1.1 255.255.255.0
```

# 开启设备的本地代理 ARP 功能。

```
[DeviceA-Vlan-interface10] local-proxy-arp enable
[DeviceA-Vlan-interface10] quit
```

# 创建 VLAN 2，并向 VLAN 2 中添加端口 GigabitEthernet1/0/1 和端口 GigabitEthernet1/0/2。

```
[DeviceA] vlan 2
[DeviceA-vlan2] port gigabitethernet 1/0/1 gigabitethernet 1/0/2
[DeviceA-vlan2] quit
```

# 创建 VLAN 3，并向 VLAN 3 中添加端口 GigabitEthernet1/0/3 和端口 GigabitEthernet1/0/4。

```
[DeviceA] vlan 3
[DeviceA-vlan3] port gigabitethernet 1/0/3 gigabitethernet 1/0/4
[DeviceA-vlan3] quit
```

# 创建 VLAN 5，并向 VLAN 5 中添加端口 GigabitEthernet1/0/5 和端口 GigabitEthernet1/0/6。

```
[DeviceA] vlan 5
[DeviceA-vlan5] port gigabitethernet 1/0/5 gigabitethernet 1/0/6
[DeviceA-vlan5] quit
```

# 配置 VLAN 10 为 Super VLAN，其关联的 Sub VLAN 为 VLAN 2、VLAN 3 和 VLAN 5。

```
[DeviceA] vlan 10
[DeviceA-vlan10] supervlan
[DeviceA-vlan10] subvlan 2 3 5
[DeviceA-vlan10] quit
[DeviceA] quit
```

## 4. 显示和验证

# 查看 Super VLAN 的相关信息，验证以上配置是否生效。

```
<DeviceA> display supervlan
Super VLAN ID: 10
Sub-VLAN ID: 2-3 5

VLAN ID: 10
VLAN type: Static
It is a super VLAN.
Route interface: Configured
Ipv4 address: 10.1.1.1
Ipv4 subnet mask: 255.255.255.0
Description: VLAN 0010
Name: VLAN 0010
Tagged ports:   None
Untagged ports: None

VLAN ID: 2
VLAN type: Static
It is a sub-VLAN.
Route interface: Configured
Ipv4 address: 10.1.1.1
Ipv4 subnet mask: 255.255.255.0
Description: VLAN 0002
Name: VLAN 0002
Tagged ports:   None
Untagged ports:
    GigabitEthernet1/0/1
    GigabitEthernet1/0/2

VLAN ID: 3
VLAN type: Static
It is a sub-VLAN.
Route interface: Configured
Ipv4 address: 10.1.1.1
Ipv4 subnet mask: 255.255.255.0
Description: VLAN 0003
Name: VLAN 0003
Tagged ports:   None
Untagged ports:
    GigabitEthernet1/0/3
    GigabitEthernet1/0/4

VLAN ID: 5
VLAN type: Static
It is a sub-VLAN.
Route unterface: Configured
Ipv4 address: 10.1.1.1
Ipv4 subnet mask: 255.255.255.0
Description: VLAN 0005
```

Name: VLAN 0005

Tagged ports: None

Untagged ports:

    GigabitEthernet1/0/5

    GigabitEthernet1/0/6

# 3 Voice VLAN

## 3.1 Voice VLAN简介

Voice VLAN 是为用户的语音数据流专门划分的 VLAN。通过划分 Voice VLAN 并将连接语音设备的端口加入 Voice VLAN，系统自动为语音报文修改 QoS（Quality of Service，服务质量）参数，来提高语音数据报文优先级、保证通话质量。



说明

常见的语音设备有 IP 电话、IAD（Integrated Access Device，综合接入设备）等。本文中以 IP 电话为例进行说明。

当 IP 电话接入设备时，需要设备完成以下两个任务：

- (1) 识别 IP 电话，获取 IP 电话的 MAC 地址，从而进行安全认证及提高语音报文的优先级。
- (2) 将 Voice VLAN 信息通告给 IP 电话，IP 电话能够根据收到的 Voice VLAN 信息完成自动配置，使 IP 电话发出的语音报文在 Voice VLAN 内传输。

## 3.2 设备识别IP电话

### 3.2.1 OUI地址

设备可以根据端口接收的报文的源 MAC 地址来判断该数据流是否为语音数据流。源 MAC 地址符合系统配置的语音设备 OUI（Organizationally Unique Identifier，全球统一标识符）地址的报文被认为是语音数据流。

用户可以预先配置OUI地址，也可以使用缺省的OUI地址作为判断标准。设备缺省的OUI地址如 [表 3-1](#) 所示。

表3-1 设备缺省的 OUI 地址

序号	OUI 地址	生产厂商
1	0001-e300-0000	Siemens phone
2	0003-6b00-0000	Cisco phone
3	0004-0d00-0000	Avaya phone
4	000f-e200-0000	H3C Aolynk phone
5	0060-b900-0000	Philips/NEC phone
6	00d0-1e00-0000	Pingtel phone
7	00e0-7500-0000	Polycom phone
8	00e0-bb00-0000	3Com phone



## 说明

- 通常意义下，OUI 地址指的是 MAC 地址的前 24 位（二进制），是 IEEE 为不同设备供应商分配的一个全球唯一的标识符。本文中的 OUI 地址有别于通常意义的 OUI 地址，它是设备判断收到的报文是否为语音报文的依据，是 **voice-vlan mac-address** 命令中的 *mac-address* 和 *oui-mask* 参数相与运算后的结果。
- 设备缺省的 OUI 地址可以手工删除，删除之后也可再次手工添加。

### 3.2.2 通过LLDP自动识别IP电话

通过设备上配置的 OUI 地址识别 IP 电话的方法受限于设备上可配置的 OUI 地址的数量，并且当网络中 IP 电话数量众多时，网络管理员的配置工作量较大。如果 IP 电话支持 LLDP（Link Layer Discovery Protocol，链路层发现协议）功能，可以配置 LLDP 自动识别 IP 电话功能。

在设备上配置了通过 LLDP 自动发现 IP 电话功能后，设备将通过 LLDP 自动发现对端设备，并与对端设备通过 LLDP 的 TLV 进行信息交互。如果通过端口收到的 LLDP System Capabilities TLV 中的信息发现对端设备具有电话能力，则认为对端设备是 IP 电话并将设备上配置的 Voice VLAN 信息通过 LLDP 发送给对端设备。这种方式使接入网络的 IP 电话类型不再受限于 OUI 地址的数量。

在完成 IP 电话的发现过程后，端口将继续完成 Voice VLAN 的其他功能，即端口将自动加入 Voice VLAN，并提高从该 IP 电话发出的语音数据的优先级。为防止 IP 电话无法通过端口上配置认证功能，设备还会将 IP 电话的 MAC 地址添加到 MAC 地址表中。

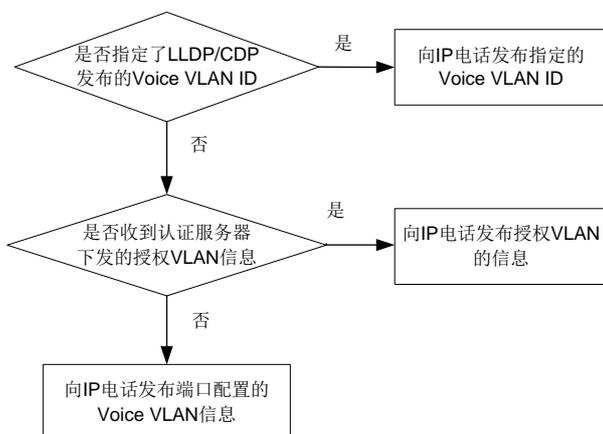
有关 LLDP 的详细信息，请参见“二层技术-以太网交换配置指导”中的“LLDP”。

### 3.3 设备将Voice VLAN信息通告给IP电话

设备可以通过三种方式将 Voice VLAN 信息通告给 IP 电话，这三种方式的优先顺序如下图所示。

- 通过命令行指定 LLDP 或 CDP (Cisco Discovery Protocol，思科发现协议) 发布的 Voice VLAN ID。
- 当 IP 电话配合接入认证功能使用时，将认证服务器下发的授权 VLAN 信息通告给 IP 电话。
- 直接将端口配置的 Voice VLAN 信息通告给 IP 电话。

图3-1 设备向 IP 电话发布 Voice VLAN 信息的过程

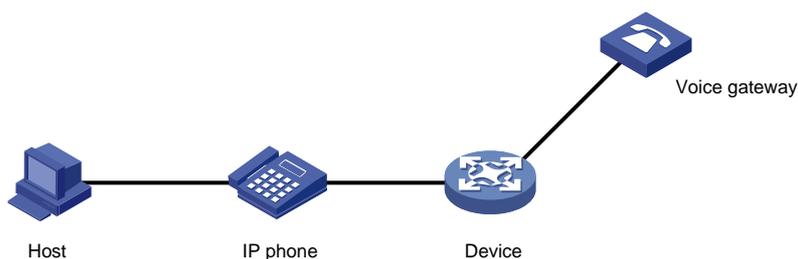


## 3.4 IP电话的接入方式

### 1. 主机和IP电话串联接入

如下图所示，主机连接到 IP 电话，IP 电话连接到接入设备。在串联接入的环境下，需要将主机和 IP 电话划分到不同的 VLAN，且需要 IP 电话能发出携带 VLAN Tag 的报文，从而区分业务数据流和语音数据流。同时，需要配置端口允许 Voice VLAN 和 PVID 通过。

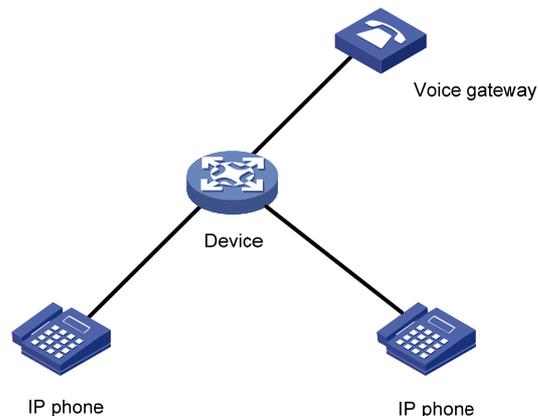
图3-2 主机与 IP 电话串联接入组网图



### 2. IP电话单独接入

如下图所示，IP 电话单独接入设备。单独接入适用于 IP 电话发出 Untagged 语音报文的情况，此时需要配置 PVID 为 Voice VLAN，并配置端口允许 PVID 通过。

图3-3 IP 电话单独接入组网图



## 3.5 Voice VLAN的工作模式

### 3.5.1 Voice VLAN的自动模式和手动模式

根据端口加入 Voice VLAN 的不同方式，可以将 Voice VLAN 的工作模式分为自动模式和手动模式。

#### 1. 自动模式

自动模式适用于主机和IP电话串联接入（端口同时传输语音数据和普通业务数据）的组网方式，如 [图 3-2](#) 所示。

自动模式下，系统利用 IP 电话上电时发出的协议报文，识别报文的源 MAC 地址，匹配 OUI 地址。匹配成功后，系统将自动把语音报文的入端口加入 Voice VLAN，并下发 ACL 规则、配置报文的优先级。用户可以在设备上配置 Voice VLAN 的老化时间，当在老化时间内，系统没有从入端口收到任何语音报文时，系统将把该端口从 Voice VLAN 中删除。端口的添加/删除到 Voice VLAN 的过程由系统自动实现。当 Voice VLAN 正常工作时，如果遇到 IP 电话重新启动，为保证已经建立的语音连接能够正常工作，系统会在 IP 电话重新启动完成后，将配置为自动模式的端口重新加入 Voice VLAN，而不需要再次通过语音流触发。

## 2. 手动模式

手动模式适用于 IP 电话单独接入（端口仅传输语音报文）的组网方式，如 [图 3-3](#) 所示。该组网方式可以使该端口专用于传输语音数据，最大限度避免业务数据对语音数据传输的影响。

手动模式下，需要手工将连接 IP 电话的端口加入 Voice VLAN 中。再通过识别报文的源 MAC 地址，匹配 OUI 地址。匹配成功后，系统将下发 ACL 规则、配置报文的优先级。端口的添加/删除到 Voice VLAN 的过程由网络管理员手动实现。

## 3. Voice VLAN工作模式和IP电话的配合

由于 IP 电话类型较多，因此需要用户保证端口的链路类型与 IP 电话匹配，不同 Voice VLAN 工作模式下的详细配合关系请见 [表 3-2](#) 和 [表 3-3](#)：

- IP 电话发送 Tagged 语音数据

表3-2 不同类型端口支持 Tagged 语音数据配置要求

Voice VLAN 工作模式	端口类型	是否支持 Tagged 语音数据	配置要求
自动模式	Access	不支持	-
	Trunk	支持	PVID不能为Voice VLAN
	Hybrid		
手工模式	Access	不支持	-
	Trunk	支持	PVID不能为Voice VLAN，需要配置端口允许Voice VLAN的报文通过
	Hybrid	支持	PVID不能为Voice VLAN，需要配置端口允许Voice VLAN的报文携带Tag通过

- IP 电话发送 Untagged 语音数据

当 IP 电话发送 Untagged 语音数据，则端口的 Voice VLAN 工作模式只能为手工模式，不能为自动模式。

表3-3 不同类型端口支持 Untagged 语音数据配置要求

Voice VLAN 工作模式	端口类型	是否支持 Untagged 语音数据	配置要求
自动模式	Access	不支持	-
	Trunk		

Voice VLAN 工作模式	端口类型	是否支持 Untagged 语音数据	配置要求
	Hybrid		
手工模式	Access	支持	端口加入Voice VLAN
	Trunk	支持	PVID必须为Voice VLAN，且接入端口允许PVID通过
	Hybrid	支持	PVID必须为Voice VLAN，且允许PVID的报文不带VLAN Tag通过

 注意

- 如果用户的 IP 电话发出的是 Tagged 语音流，且接入端口上开启了 802.1X 认证和 Guest VLAN/Auth-Fail VLAN/Critical VLAN，为保证各种功能的正常使用，请为 Voice VLAN、PVID 和 802.1X 的 Guest VLAN/Auth-Fail VLAN/Critical VLAN 分配不同的 VLAN ID。
- 如果用户的 IP 电话发出的是 Untagged 语音流，为实现 Voice VLAN 功能，只能将 PVID 配置为 Voice VLAN，此时将不能实现 802.1X 认证功能。

### 3.5.2 Voice VLAN的安全模式和普通模式

开启了 Voice VLAN 功能的端口会对接收到的报文进行过滤，根据过滤机制的不同，可以将 Voice VLAN 的工作模式分为普通模式和安全模式：

- 普通模式下，端口加入 Voice VLAN 后，设备对于接收的语音报文不再一一进行识别，凡是带有 Voice VLAN Tag 的报文，设备将不再检查其源 MAC 地址是否为语音设备的 OUI 地址，均接收并在 Voice VLAN 中转发。对于 PVID 就是 Voice VLAN 的手工模式端口，会导致任意的 Untagged 报文都可以在 Voice VLAN 中传输。这样的处理方式很容易使 Voice VLAN 收到恶意用户的流量攻击。恶意用户可以构造大量带有 Voice VLAN Tag 或 Untagged 的报文，占用 Voice VLAN 的带宽，影响正常的语音通信。
- 安全模式下，设备将对每一个要进入 Voice VLAN 传输的报文进行源 MAC 地址匹配检查，对于不能匹配 OUI 地址的报文，则将其丢弃。

对于比较安全的网络，用户可以配置 Voice VLAN 的普通模式，以减少检查报文的工作对系统资源的占用。

 提示

- 只有匹配了 OUI 地址的报文才能被修改优先级。比如在普通模式下，报文在 Voice VLAN 中转发，但如果该报文未匹配 OUI 地址，则该报文不会被修改优先级。
- 建议用户尽量不要在 Voice VLAN 中同时传输语音和业务数据。如确有此需要，请确认 Voice VLAN 的安全模式已关闭，否则业务数据会被丢弃。

表3-4 Voice VLAN 的安全/普通模式对报文的处理

Voice VLAN 工作模式	报文类型	处理方式
安全模式	Untagged报文	当报文的源MAC地址是可识别的OUI地址时，允许该报文在Voice VLAN内传输，否则将该报文丢弃
	带有Voice VLAN Tag的报文	
	带有其他VLAN Tag的报文	根据指定端口是否允许该VLAN通过来对报文进行转发和丢弃的处理，不受Voice VLAN安全/普通模式的影响
普通模式	Untagged报文	不对报文的源MAC地址进行检查，所有报文均可以在Voice VLAN内进行传输
	带有Voice VLAN Tag的报文	
	带有其他VLAN Tag的报文	根据指定端口是否允许该VLAN通过来对报文进行转发和丢弃的处理，不受Voice VLAN安全/普通模式的影响

## 3.6 Voice VLAN配置任务简介

表3-5 Voice VLAN 配置任务简介

配置任务	说明	详细配置
配置语音报文的QoS优先级	必选	<a href="#">3.7.1</a>
配置自动模式下的Voice VLAN	二者必选其一	<a href="#">3.7.2</a>
配置手动模式下的Voice VLAN		<a href="#">3.7.3</a>
配置通过LLDP自动发现IP电话功能	可选	<a href="#">3.7.4</a>
配置通过LLDP通过Voice VLAN信息	二者可选其一	<a href="#">3.7.5</a>
配置通过CDP通告Voice VLAN信息		<a href="#">3.7.6</a>

## 3.7 配置Voice VLAN

### 3.7.1 配置语音报文的QoS优先级

Voice VLAN 在实现中，通过提高语音报文的 QoS 优先级（CoS 和 DSCP 值）来保证语音通信的质量。语音报文会自带 QoS 优先级，通过配置，用户可以选择在语音报文通过设备时修改或者不修改报文的 QoS 优先级。

需要注意的是，在同一端口多次执行语音报文的 QoS 优先级配置时，最后一次执行的配置生效。

表3-6 配置语音报文的 QoS 优先级

操作	命令	说明
进入系统视图	<b>system-view</b>	-

操作	命令	说明
进入二层以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
配置端口信任Voice VLAN内语音报文的优先级	<b>voice-vlan qos trust</b>	二者选其一
配置端口将Voice VLAN内语音报文的CoS和DSCP值修改为指定值	<b>voice-vlan qos</b> <i>cos-value</i> <i>dscp-value</i>	缺省情况下，端口将Voice VLAN内语音报文的CoS值修改为6，DSCP值修改为46

### 3.7.2 配置自动模式下的Voice VLAN

配置自动模式下的 Voice VLAN 时，需要注意：

- 自动模式下的 Voice VLAN 只支持 Hybrid 端口对 Tagged 的语音流进行处理，而协议 VLAN 特性要求 Hybrid 入端口的报文格式为 Untagged 的，因此，不能将某个 VLAN 同时配置为 Voice VLAN 和协议 VLAN。
- 配置 MSTP 多实例情况下，如果端口在要加入的 Voice VLAN 对应的 MSTP 实例中是阻塞状态，则端口会丢弃收到的报文，造成 MAC 地址不能上送，不能完成动态触发功能。自动模式 Voice VLAN 的使用场景为接入侧，不建议和多实例 MSTP 同时使用。
- 配置 PVST 情况下，如果端口要加入的 Voice VLAN 不为端口允许通过的 VLAN，则端口处于阻塞状态，会丢弃收到的报文，造成 MAC 地址不能上送，不能完成动态触发功能。自动模式 Voice VLAN 的使用场景为接入侧，不建议和 PVST 同时使用。
- 当端口配置了动态触发端口加入静态 MAC VLAN，又配置本功能时，两个功能可能会相互影响，导致其中某个功能不可用。当端口同时配置了本功能和动态触发端口加入静态 MAC VLAN，再取消其中任何一个功能的配置，会导致另一个功能不可用。因此不建议同一端口同时配置本功能和动态触发端口加入静态 MAC VLAN。

表3-7 配置自动模式下的 Voice VLAN

操作	命令	说明
进入系统视图	<b>system-view</b>	-
(可选) 配置Voice VLAN的老化时间	<b>voice-vlan aging</b> <i>minutes</i>	缺省情况下，老化时间为1440分钟，老化时间只对自动模式下的端口有效
(可选) 开启Voice VLAN的安全模式	<b>voice-vlan security enable</b>	缺省情况下，Voice VLAN工作在安全模式
(可选) 配置Voice VLAN识别的OUI地址	<b>voice-vlan mac-address</b> <i>oui mask</i> <i>oui-mask</i> [ <i>description text</i> ]	Voice VLAN启动后将有缺省的OUI地址，请参见“ <a href="#">表3-1设备缺省的OUI地址</a> ”
进入二层以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
配置端口的链路类型	<b>port link-type trunk</b>	二者选其一
	<b>port link-type hybrid</b>	
配置端口的Voice VLAN工作模式为自动模式	<b>voice-vlan mode auto</b>	缺省情况下，端口的Voice VLAN工作模式为自动模式

操作	命令	说明
开启端口的Voice VLAN功能	<b>voice-vlan <i>vlan-id</i> enable</b>	缺省情况下,端口的Voice VLAN功能处于关闭状态 开启端口的Voice VLAN功能之前,须确保对应的VLAN已存在

### 3.7.3 配置手动模式下的Voice VLAN

配置手动模式下的 Voice VLAN 时, 需要注意:

- 同一设备同一时刻可以给不同的端口配置不同的 Voice VLAN, 但一个端口只能配置一个 Voice VLAN, 而且这些 VLAN 必须是已经存在的静态 VLAN。
- 不允许在聚合组的成员端口上开启 Voice VLAN 功能。有关聚合组的成员端口的详细介绍, 请参见“二层技术-以太网交换配置指导”中的“以太网链路聚合”。
- 当端口开启了 Voice VLAN 并工作在手工模式时, 必须手工将端口加入 Voice VLAN, 才能保证 Voice VLAN 功能生效。

表3-8 配置手动模式下的 Voice VLAN

操作	命令	说明
进入系统视图	<b>system-view</b>	-
(可选)开启Voice VLAN的安全模式	<b>voice-vlan security enable</b>	缺省情况下, Voice VLAN工作在安全模式
(可选)配置Voice VLAN中可识别的OUI地址	<b>voice-vlan mac-address <i>oui mask oui-mask</i> [ <i>description text</i> ]</b>	Voice VLAN启动后将有缺省的OUI地址, 请参见“ <a href="#">表3-1设备缺省的OUI地址</a> ”
进入二层以太网接口视图	<b>interface <i>interface-type interface-number</i></b>	-
配置端口的Voice VLAN工作模式为手动模式	<b>undo voice-vlan mode auto</b>	缺省情况下,端口的Voice VLAN工作模式为自动模式
将手动模式端口加入Voice VLAN	Access端口	请参见“ <a href="#">1.4.2 配置基于Access端口的VLAN</a> ”
	Trunk端口	请参见“ <a href="#">1.4.3 配置基于Trunk端口的VLAN</a> ”
	Hybrid端口	请参见“ <a href="#">1.4.4 配置基于Hybrid端口的VLAN</a> ”
(可选)配置PVID为Voice VLAN	Trunk端口	请参见“ <a href="#">1.4.3 配置基于Trunk端口的VLAN</a> ”
	Hybrid端口	请参见“ <a href="#">1.4.4 配置基于Hybrid端口的VLAN</a> ”
开启端口的Voice VLAN功能	<b>voice-vlan <i>vlan-id</i> enable</b>	缺省情况下,端口的Voice VLAN功能处于关闭状态 开启端口的Voice VLAN功能之前,须先创建对应的VLAN

### 3.7.4 配置通过LLDP自动发现IP电话功能

配置通过 LLDP 自动发现 IP 电话功能时，需要注意：

- 在配置本功能前，需要在全局和接入端口均开启 LLDP 功能。
- 通过 LLDP 自动发现 IP 电话功能只能与 Voice VLAN 自动模式配合使用，不能与手动模式配合使用。
- 通过 LLDP 自动发现 IP 电话功能与 LLDP 兼容 CDP 功能不能同时配置。
- 设备开启通过 LLDP 自动发现 IP 电话功能后，每个端口最多可以接入 5 台 IP 电话。

表3-9 配置通过 LLDP 自动发现 IP 电话功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
开启通过LLDP自动发现IP电话功能	<b>voice-vlan track lldp</b>	缺省情况下，通过LLDP自动发现IP电话功能处于关闭状态

### 3.7.5 配置通过LLDP通告Voice VLAN信息

对于支持 LLDP 的 IP 电话，可以通过 LLDP-MED 中的 Network Policy TLV 将 Voice VLAN 信息通告给 IP 电话。

需要注意的是，在配置本功能前，需要在全局和接入端口开启 LLDP 功能。

表3-10 配置通过 LLDP 通告 Voice VLAN 信息

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二层以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
配置端口上发布的Voice VLAN ID	<b>lldp tlv-enable med-tlv network-policy</b> <i>vlan-id</i>	缺省情况下，未配置端口上发布的Voice VLAN ID



说明

- LLDP-MED Network Policy TLV 中的信息可以通过 **display lldp local-information** 命令查看。
- 有关 **lldp tlv-enable** 命令和 **display lldp local-information** 命令的详细介绍，请参见“二层技术-以太网交换命令参考”中的“LLDP”。

### 3.7.6 配置通过CDP通告Voice VLAN信息

如果 IP 电话只支持 CDP，不支持 LLDP，当设备与这类 IP 电话直连时，IP 电话将会向设备发送 CDP 报文以请求在设备上所配 Voice VLAN 的 VLAN ID；如果在指定时间内没有收到设备发送的

Voice VLAN 的 VLAN ID，IP 电话将会把语音数据流以 Untagged 方式发送，从而导致语音数据流与其他类型的数据流混在一起，无法进行区分。

通过在设备上配置 LLDP 兼容 CDP 功能，可以利用 LLDP 来接收、识别从 IP 电话发送的 CDP 报文，并向 IP 电话发送 CDP 报文，该 CDP 报文携带设备配置的 Voice VLAN 信息，使 IP 电话完成 Voice VLAN 的自动配置。之后 IP 电话的语音数据流将被限制在配置的 Voice VLAN 内，与其他数据流区分开来。

需要注意的是，在配置本功能前，需要在全局和接入端口开启 LLDP 功能。

表3-11 配置通过 CDP 通告 Voice VLAN 信息

操作	命令	说明
进入系统视图	<b>system-view</b>	-
开启LLDP兼容CDP功能	<b>lldp compliance cdp</b>	缺省情况下，LLDP兼容CDP功能处于关闭状态
进入二层以太网接口视图	<b>interface interface-type interface-number</b>	-
配置LLDP兼容CDP功能的工作模式为TxRx	<b>lldp compliance admin-status cdp txrx</b>	缺省情况下，LLDP兼容CDP功能的工作模式为Disable
配置CDP报文携带的Voice VLAN ID	<b>cdp voice-vlan vlan-id</b>	缺省情况下，未配置CDP报文携带的Voice VLAN ID



说明

设备发送给 IP 电话的 CDP 报文中不包含优先级信息。

## 3.8 Voice VLAN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 Voice VLAN 的运行情况，通过查看显示信息验证配置的效果。

表3-12 Voice VLAN 显示和维护

操作	命令
显示Voice VLAN的状态	<b>display voice-vlan state</b>
显示系统当前支持的OUI地址	<b>display voice-vlan mac-address</b>

## 3.9 Voice VLAN典型配置举例

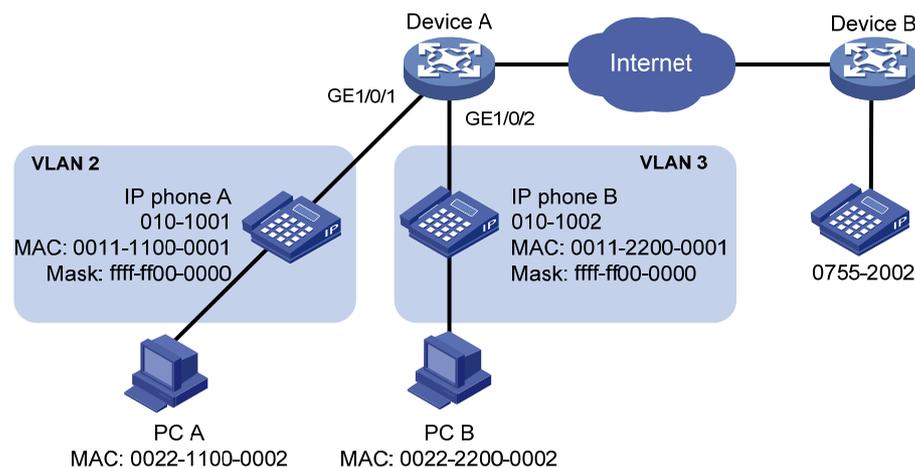
### 3.9.1 自动模式下Voice VLAN的配置举例

#### 1. 组网需求

- IP phone A 的 MAC 地址为 0011-1100-0001，下行连接 PC A(MAC 地址为 0022-1100-0002)，上行连接到 Device A 的 GigabitEthernet1/0/1 端口。
- IP phone B 的 MAC 地址为 0011-2200-0001，下行连接 PC B(MAC 地址为 0022-2200-0002)，上行连接到 Device A 的 GigabitEthernet1/0/2 端口。
- Device A 使用 Voice VLAN 2 传输 IP phone A 产生的语音报文；使用 Voice VLAN 3 传输 IP phone B 产生的语音报文。
- Device A 的端口 GigabitEthernet1/0/1 和 GigabitEthernet1/0/2 工作在自动模式，如果它们在 30 分钟内没有收到语音流，就将相应的 Voice VLAN 老化。

#### 2. 组网图

图3-4 配置自动模式下 Voice VLAN 组网图



#### 3. 配置步骤

# 创建 VLAN 2 和 VLAN 3。

```
<DeviceA> system-view
[DeviceA] vlan 2 to 3
```

# 配置 Voice VLAN 的老化时间为 30 分钟。

```
[DeviceA] voice-vlan aging 30
```

# 由于端口 GigabitEthernet1/0/1 可能会同时收到语音和数据两种流量，为了保证语音报文的质量以及带宽的高效利用，配置 Voice VLAN 工作在安全模式，即 Voice VLAN 只用于传输语音报文。（可选，缺省情况下，Voice VLAN 工作在安全模式）

```
[DeviceA] voice-vlan security enable
```

# 配置允许 OUI 地址为 0011-1100-0000 和 0011-2200-0000 的报文通过 Voice VLAN，即当报文源 MAC 地址前缀为 0011-1100-0000 或 0011-2200-0000 时，Device A 会把它当成语音报文来处理。

```
[DeviceA] voice-vlan mac-address 0011-1100-0001 mask ffff-ff00-0000 description IP phone A
```

```

[DeviceA] voice-vlan mac-address 0011-2200-0001 mask ffff-ff00-0000 description IP phone B
# 配置端口 GigabitEthernet1/0/1 为 Hybrid 类型端口。
[DeviceA] interface gigabitethernet 1/0/1
[DeviceA-GigabitEthernet1/0/1] port link-type hybrid
# 将端口 GigabitEthernet1/0/1 上 Voice VLAN 的工作模式配置为自动模式。（可选，缺省情况下，
端口的 Voice VLAN 工作在自动模式。）
[DeviceA-GigabitEthernet1/0/1] voice-vlan mode auto
# 开启端口 Voice VLAN 功能。
[DeviceA-GigabitEthernet1/0/1] voice-vlan 2 enable
[DeviceA-GigabitEthernet1/0/1] quit
# 在端口 GigabitEthernet1/0/2 上进行相应的配置。
[DeviceA] interface gigabitethernet 1/0/2
[DeviceA-GigabitEthernet1/0/2] port link-type hybrid
[DeviceA-GigabitEthernet1/0/2] voice-vlan mode auto
[DeviceA-GigabitEthernet1/0/2] voice-vlan 3 enable
[DeviceA-GigabitEthernet1/0/2] quit

```

#### 4. 显示和验证

# 显示当前系统支持的 OUI 地址、OUI 地址掩码和描述信息。

```

[DeviceA] display voice-vlan mac-address
OUI Address      Mask             Description
0001-e300-0000   ffff-ff00-0000   Siemens phone
0003-6b00-0000   ffff-ff00-0000   Cisco phone
0004-0d00-0000   ffff-ff00-0000   Avaya phone
000f-e200-0000   ffff-ff00-0000   H3C Aolynk phone
0011-1100-0000   ffff-ff00-0000   IP phone A
0011-2200-0000   ffff-ff00-0000   IP phone B
0060-b900-0000   ffff-ff00-0000   Philips/NEC phone
00d0-1e00-0000   ffff-ff00-0000   Pingtel phone
00e0-7500-0000   ffff-ff00-0000   Polycom phone
00e0-bb00-0000   ffff-ff00-0000   3Com phone

```

# 显示当前 Voice VLAN 的状态。

```

[DeviceA] display voice-vlan state
Current voice VLANs: 2
Voice VLAN security mode: Security
Voice VLAN aging time: 30 minutes
Voice VLAN enabled ports and their modes:

```

Port	VLAN	Mode	CoS	DSCP
GE1/0/1	2	Auto	6	46
GE1/0/2	3	Auto	6	46

### 3.9.2 手动模式下Voice VLAN的配置举例

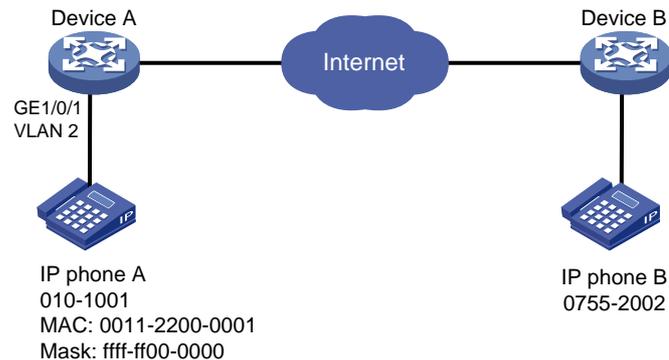
#### 1. 组网需求

- IP Phone A 接入 Device A 的 Hybrid 类型端口 GigabitEthernet1/0/1。IP Phone A 发出的报文为 Untagged 报文。

- Device A 上 VLAN 2 为 Voice VLAN。
- 手工将 Device A 的端口 GigabitEthernet1/0/1 工作加入 Voice VLAN，其 PVID 为 VLAN 2，添加 OUI 地址 0011-2200-0000，使该端口专用于传输语音报文。

## 2. 组网图

图3-5 配置手动模式下 Voice VLAN 组网图



## 3. 配置步骤

# 配置 Voice VLAN 为安全模式，使得 Voice VLAN 端口只允许合法的语音报文通过。（可选，缺省情况下，Voice VLAN 工作在安全模式）

```
<DeviceA> system-view
```

```
[DeviceA] voice-vlan security enable
```

# 配置允许 OUI 地址为 0011-2200-0000 的报文通过 Voice VLAN，即报文源 MAC 地址前缀为 0011-2200-0000 时，Device A 会把它当成语音报文来处理。

```
[DeviceA] voice-vlan mac-address 0011-2200-0001 mask ffff-ff00-0000 description test
```

# 创建 VLAN 2。

```
[DeviceA] vlan 2
```

```
[DeviceA-vlan2] quit
```

# 配置端口 GigabitEthernet1/0/1 工作在手动模式。

```
[DeviceA] interface gigabitethernet 1/0/1
```

```
[DeviceA-GigabitEthernet1/0/1] undo voice-vlan mode auto
```

# 配置端口 GigabitEthernet1/0/1 为 Hybrid 类型。

```
[DeviceA-GigabitEthernet1/0/1] port link-type hybrid
```

# 配置 Voice VLAN 是端口 GigabitEthernet1/0/1 的缺省 VLAN，且在该端口允许通过的 Untagged VLAN 列表中。

```
[DeviceA-GigabitEthernet1/0/1] port hybrid pvid vlan 2
```

```
[DeviceA-GigabitEthernet1/0/1] port hybrid vlan 2 untagged
```

# 开启端口 GigabitEthernet1/0/1 的 Voice VLAN 功能。

```
[DeviceA-GigabitEthernet1/0/1] voice-vlan 2 enable
```

```
[DeviceA-GigabitEthernet1/0/1] quit
```

## 4. 显示与验证

# 显示当前系统支持的 OUI 地址、OUI 地址掩码和描述信息。

```
[DeviceA] display voice-vlan mac-address
```

OUI Address	Mask	Description
0001-e300-0000	ffff-ff00-0000	Siemens phone
0003-6b00-0000	ffff-ff00-0000	Cisco phone
0004-0d00-0000	ffff-ff00-0000	Avaya phone
000f-e200-0000	ffff-ff00-0000	H3C Aolynk phone
0011-2200-0000	ffff-ff00-0000	test
0060-b900-0000	ffff-ff00-0000	Philips/NEC phone
00d0-1e00-0000	ffff-ff00-0000	Pingtel phone
00e0-7500-0000	ffff-ff00-0000	Polycom phone
00e0-bb00-0000	ffff-ff00-0000	3Com phone

# 显示当前 Voice VLAN 的状态。

[DeviceA] display voice-vlan state

Current voice VLANs: 1

Voice VLAN security mode: Security

Voice VLAN aging time: 1440 minutes

Voice VLAN enabled ports and their modes:

Port	VLAN	Mode	CoS	DSCP
GE1/0/1	2	Manual	6	46

# 目 录

1 MVRP .....	1-1
1.1 MVRP简介 .....	1-1
1.1.1 MRP简介 .....	1-1
1.1.2 MVRP注册模式 .....	1-4
1.1.3 协议规范 .....	1-4
1.2 MVRP配置任务简介 .....	1-4
1.3 配置MVRP功能 .....	1-5
1.3.1 配置限制和指导 .....	1-5
1.3.2 配置准备 .....	1-5
1.3.3 开启MVRP功能 .....	1-5
1.3.4 配置MVRP注册模式 .....	1-6
1.3.5 配置MRP定时器 .....	1-6
1.3.6 配置MVRP兼容GVRP .....	1-7
1.4 MVRP显示和维护 .....	1-7
1.5 MVRP典型配置举例 .....	1-8

# 1 MVRP

## 1.1 MVRP简介

MRP（Multiple Registration Protocol，多属性注册协议）作为一个属性注册协议的载体，可以用来传递属性信息。MVRP（Multiple VLAN Registration Protocol，多VLAN注册协议）是MRP的一种应用，用于在设备间发布并学习VLAN配置信息。当设备启动了MVRP之后，设备将本地的VLAN配置信息向其他设备发送，同时还能够接收来自其他设备的VLAN配置信息，并动态更新本地的VLAN配置信息，从而使所有设备的VLAN信息都达成一致，极大减少了网络管理员的VLAN配置工作。在网络拓扑发生变化后，MVRP还能根据新的拓扑重新发布及学习VLAN配置信息，做到VLAN配置信息实时与网络拓扑同步更新。

### 1.1.1 MRP简介

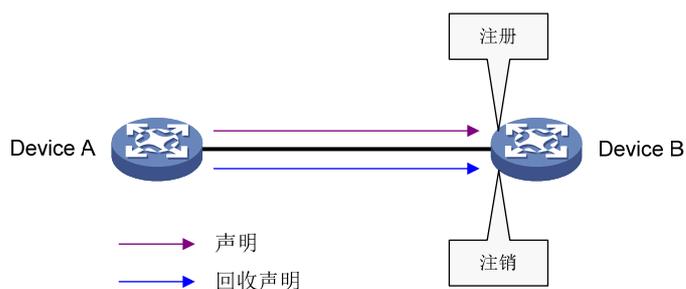
MRP支持在MSTI（Multiple Spanning Tree Instance，多生成树实例）的基础上，协助同一局域网内各成员之间传递属性信息。有关MSTI的详细介绍，请参见“二层技术-以太网交换配置指导”中的“生成树”。

#### 1. MRP实现机制

设备上每一个参与协议的端口都可以视为一个应用实体。当MRP应用（如MVRP）在端口上启动之后，该端口就可视为一个MRP应用实体（以下简称MRP实体，同样的，MVRP应用实体简称MVRP实体）。

如[图1-1](#)所示，MRP实体通过发送声明类或回收声明类消息（以下简称声明和回收声明），来通知其他MRP实体注册或注销自己的属性信息，并根据其他MRP实体发来的声明或回收声明来注册或注销对方的属性信息。通过MRP机制，一个MRP实体上的配置信息会迅速传遍整个局域网。

图1-1 MRP 实现机制示意图



以通过MVRP实现VLAN注册和注销为例，MRP的属性注册和注销过程如下：

- 当端口收到一个VLAN的声明时，该端口将注册该声明中的VLAN（该端口将加入到该VLAN中）。
- 当端口收到一个VLAN的回收声明时，该端口将注销该声明中的VLAN（该端口将退出该VLAN）。



说明

图 1-1 可以看作是 MRP 协议在某个 MSTI 上的实现机制，属于比较简单的一种情况，在实际应用的复杂组网情况下，可能存在多个 MSTI，而属性的注册和注销只会各自的 MSTI 上进行。

## 2. MRP 消息

MRP 消息主要包括 Join 消息、New 消息、Leave 消息和 LeaveAll 消息，它们通过互相配合来实现信息的注册或注销。其中，Join 消息和 New 消息属于声明，Leave 消息和 LeaveAll 消息属于回收声明。

### (1) Join 消息

- 当一个 MRP 实体配置了某些属性，需要对端实体来注册自己的属性信息时，它会向对端实体发送 Join 消息。
- 当一个 MRP 实体收到来自对端实体的 Join 消息时，它会注册该 Join 消息中的属性，并向本设备的其他实体传播该 Join 消息，其他实体收到传播的 Join 消息后，向其对端实体发送 Join 消息。

MRP 实体间发送的 Join 消息又分为 JoinEmpty 和 JoinIn 两种（对于同一设备的实体间传播的 Join 消息则不做区分），二者的区别如下：

- **JoinEmpty**: 用于声明 MRP 实体的非注册属性。比如一个 MRP 实体加入了某静态 VLAN（我们将本地手工创建的 VLAN 称为静态 VLAN，通过 MRP 消息学习并创建的 VLAN 称为动态 VLAN），此时若该实体还没有通过 MRP 消息注册该 VLAN，这时该实体向对端实体发送的 Join 消息就为 JoinEmpty 消息。
- **JoinIn**: 用于声明 MRP 实体的注册属性。比如 MRP 实体加入了某静态 VLAN 且通过 MRP 消息注册了该 VLAN，或该实体收到本设备其他实体传播的某 VLAN 的 Join 信息且通过 MRP 消息注册了该 VLAN，这时该实体向对端实体发送的 Join 消息就为 JoinIn 消息。

### (2) New 消息

New 消息的作用和 Join 消息比较类似，都是用于对属性的声明。不同的是，New 消息主要用于 MSTP（Multiple Spanning Tree Protocol，多生成树协议）拓扑变化的情况。

- 当 MSTP 拓扑变化时，MRP 实体需要向对端实体发送 New 消息声明拓扑变化。
- 当一个 MRP 实体收到来自对端实体的 New 消息时，它会注册该 New 消息中的属性，并向本设备的其他实体传播该 New 消息，其他实体收到传播的 New 消息后，向其对端实体发送该 New 消息。

### (3) Leave 消息

- 当一个 MRP 实体注销了某些属性，需要对端实体进行同步注销时，它会向对端实体发送 Leave 消息。
- 当一个 MRP 实体收到来自对端实体的 Leave 消息时，它会注销该 Leave 消息中的属性，并向本设备的其他实体传播该 Leave 消息，其他实体收到传播的 Leave 消息后，根据该 Leave 消息中的属性在本设备上的状态，决定是否向其对端实体发送该 Leave 消息（比如该 Leave 消息中的属性为某 VLAN，若该 VLAN 为动态 VLAN，且本设备上无实体注册该 VLAN，则在设备上删除该 VLAN，并向对端实体发送该 Leave 消息；若该 VLAN 为静态 VLAN，则不向对端实体发送该 Leave 消息）。

#### (4) LeaveAll 消息

- 每个 MRP 实体启动时都会启动各自的 LeaveAll 定时器，当该定时器超时后，MRP 实体就会向对端实体发送 LeaveAll 消息。
- 当一个 MRP 实体收发 LeaveAll 消息时，它会启动 Leave 定时器，同时根据自身的属性状态决定是否发送 Join 消息要求对端实体重新注册某属性。该实体在 Leave 定时器超时前，重新注册收到的来自对端实体的 Join 消息中的属性；在 Leave 定时器超时后，注销所有未重新注册的属性信息，从而周期性地清除网络中的垃圾属性。

### 3. MRP定时器

MRP 定义了四种定时器，用于控制各种 MRP 消息的发送。

#### (1) Periodic 定时器

每个 MRP 实体启动时都会启动各自的 Periodic 定时器，来控制 MRP 消息的周期发送。该定时器超时前，实体收集需要发送的 MRP 消息，在该定时器超时后，将所有待发送的 MRP 消息封装成尽可能少的报文发送出去，这样减少了报文发送数量。随后再重新启动 Periodic 定时器，开始新一轮的循环。



说明

Periodic 定时器允许用户通过命令行开启或关闭。如果关闭 Periodic 定时器，则 MRP 实体不再周期发送 MRP 消息，仅在 LeaveAll 定时器超时或收到来自对端实体的 LeaveAll 消息的情况下会发送 MRP 消息。

---

#### (2) Join 定时器

Join 定时器用来控制 Join 消息的发送。为了保证消息能够可靠地发送到对端实体，MRP 实体在发送 Join 消息时，将启动 Join 定时器。如果在该定时器超时前收到了来自对端实体的 JoinIn 消息，且该 JoinIn 消息中的属性与发出的 Join 消息中的属性一致，便不再重发该 Join 消息，否则在该定时器超时后，当 Periodic 定时器也超时，它将重发一次该 Join 消息。

#### (3) Leave 定时器

Leave 定时器用来控制属性的注销。当 MRP 实体收到来自对端实体的 Leave 消息(或收发 LeaveAll 消息)时，将启动 Leave 定时器。如果在该定时器超时前，收到来自对端实体的 Join 消息，且该 Join 消息中的属性与收到的 Leave 消息中的属性一致(或与收发的 LeaveAll 消息中的某些属性一致)，则这些属性不会在本实体被注销，其他属性则会在该定时器超时后被注销。

#### (4) LeaveAll 定时器

每个 MRP 实体启动时都会启动各自的 LeaveAll 定时器，当该定时器超时后，该实体就会向对端实体发送 LeaveAll 消息，随后再重新启动 LeaveAll 定时器，开始新一轮的循环，对端实体在收到 LeaveAll 消息后也重新启动 LeaveAll 定时器。



说明

LeaveAll 定时器具有抑制机制，即当某个 MRP 实体的 LeaveAll 定时器超时后，会向对端实体发送 LeaveAll 消息，对端实体在收到 LeaveAll 消息时，重启本实体的 LeaveAll 定时器，从而有效抑制网络中的 LeaveAll 消息数。为了防止每次都是同一实体的 LeaveAll 定时超时，每次 LeaveAll 定时器重启时，LeaveAll 定时器的值都将在一定范围内随机变动。

## 1.1.2 MVRP注册模式

MVRP 传递的 VLAN 配置信息既包括本地手工配置的静态信息，也包括来自其他设备的动态信息。MVRP 有三种注册模式，不同注册模式对动态 VLAN 的处理方式有所不同。

- Normal 模式

该模式下的 MVRP 实体允许进行动态 VLAN 的注册或注销。

- Fixed 模式

该模式下的 MVRP 实体禁止进行动态 VLAN 的注销，收到的 MVRP 报文会被丢弃。也就是说，在该模式下，实体已经注册的动态 VLAN 是不会被注销的，同时也不会注册新的动态 VLAN。

- Forbidden 模式

该模式下的 MVRP 实体禁止进行动态 VLAN 的注册，收到的 MVRP 报文会被丢弃。也就是说，在该模式下，实体不会注册新的动态 VLAN，一旦在配置该模式前注册的动态 VLAN 被注销后，不会重新进行注册。

## 1.1.3 协议规范

与 MVRP 相关的协议规范有：

- IEEE 802.1ak: IEEE Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks – Amendment 07: Multiple Registration Protocol

## 1.2 MVRP配置任务简介

表1-1 MVRP 配置任务简介

配置任务	说明	详细配置
开启MVRP功能	必选	<a href="#">1.3.3</a>
配置MVRP注册模式	可选	<a href="#">1.3.4</a>
配置MRP定时器	可选	<a href="#">1.3.5</a>
配置MVRP兼容GVRP功能	可选	<a href="#">1.3.6</a>

## 1.3 配置MVRP功能

### 1.3.1 配置限制和指导

- MVRP 功能只能与 STP、RSTP 或 MSTP 配合使用，而无法与其他二层网络拓扑协议（如 PVST、RRPP 和 Smart Link）同时配置。MVRP 报文的收发不受 STP/RSTP/MSTP 阻塞端口影响。有关 STP、RSTP、MSTP 和 PVST 的详细介绍，请参见“二层技术-以太网交换配置指导”中的“生成树”；有关 RRPP 的详细介绍，请参见“可靠性配置指导”中的“RRPP”；有关 Smart Link 的详细介绍，请参见“可靠性配置指导”中的“Smart Link”。
- 建议不要同时启用远程端口镜像功能和 MVRP 功能，否则 MVRP 可能将远程镜像 VLAN 注册到错误的端口上，导致镜像目的端口会收到很多不必要的报文。有关远程端口镜像的详细介绍，请参见“网络管理和监控配置指导”中的“镜像”。
- 在二层聚合接口上启用了 MVRP 功能后，会同时在二层聚合接口和对应的所有选中成员端口上进行动态 VLAN 的注册或注销。
- 如果二层以太网接口加入了聚合组，则加入聚合组之前和加入聚合组之后在该接口上进行的 MVRP 相关配置不会生效，该接口退出聚合组后，MVRP 的配置才会生效。

### 1.3.2 配置准备

- 由于 MVRP 需要基于 MSTI 运行，因此在配置 MVRP 时，需要保证当前网络内所有 MSTI 都生效，即网络中设备都需要至少存在一个 MSTI 对应的 VLAN 以保证 MSTI 能够生效。
- MVRP 功能只能在 Trunk 端口上生效，因此需要保证 MVRP 实体的端口链路类型为 Trunk 类型。有关 Trunk 端口的详细介绍，请参见“二层技术-以太网交换配置指导”中的“VLAN”。

### 1.3.3 开启MVRP功能

表1-2 开启 MVRP 功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
全局开启MVRP功能	<b>mvrp global enable</b>	缺省情况下，全局的MVRP功能处于关闭状态 要使端口上的MVRP功能生效，必须全局开启MVRP功能
进入二层以太网接口或二层聚合接口视图	<b>interface <i>interface-type</i> <i>interface-number</i></b>	-
配置端口的链路类型为Trunk类型	<b>port link-type trunk</b>	缺省情况下，端口的链路类型为Access类型
配置允许指定的VLAN通过当前Trunk端口	<b>port trunk permit vlan { <i>vlan-id-list</i>   all }</b>	缺省情况下，Trunk端口只允许VLAN 1通过 需要保证所有注册的VLAN都能够从该端口通过
在端口上开启MVRP功能	<b>mvrp enable</b>	缺省情况下，端口上的MVRP功能处于关闭状态



说明

有关 **port link-type trunk** 和 **port trunk permit vlan** 命令的详细介绍，请参见“二层技术-以太网交换命令参考”中的“VLAN”。

### 1.3.4 配置MVRP注册模式

表1-3 配置 MVRP 注册模式

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二层以太网接口或二层聚合接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
配置端口的MVRP注册模式	<b>mvrp registration</b> { <b>fixed</b>   <b>forbidden</b>   <b>normal</b> }	缺省情况下，当前端口的MVRP端口注册模式为Normal模式

### 1.3.5 配置MRP定时器

配置 MRP 定时器时，需要注意：

- MRP 定时器的值建议全网一致，否则会出现 VLAN 频繁注册/注销的情况。
- 设备的每个端口上都独立维护自己的 Periodic 定时器、Join 定时器和 LeaveAll 定时器，而每个端口的每个属性上分别维护着一个 Leave 定时器。

表1-4 配置 MRP 定时器

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二层以太网接口或二层聚合接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
配置LeaveAll定时器的值	<b>mrp timer leaveall</b> <i>timer-value</i>	缺省情况下，LeaveAll定时器的值为1000厘秒
配置Join定时器的值	<b>mrp timer join</b> <i>timer-value</i>	缺省情况下，Join定时器的值为20厘秒
配置Leave定时器的值	<b>mrp timer leave</b> <i>timer-value</i>	缺省情况下，Leave定时器的值为60厘秒
配置Periodic定时器的值	<b>mrp timer periodic</b> <i>timer-value</i>	缺省情况下，Periodic定时器的值为100厘秒

如 [表 1-5](#) 所示，为保证MVRP的正常运行，Join定时器、Leave定时器和LeaveAll定时器的正确取值范围间存在着相互依赖的关系，当配置某定时器时，如果配置值超出了该定时器当前正确的取值

范围，则该配置属于错误配置。用户可以通过改变相关定时器的值（必须是 20 厘秒的倍数）来重新进行配置。

表1-5 Join 定时器、Leave 定时器和 LeaveAll 定时器正确取值范围间的依赖关系

定时器	取值下限	取值上限
Join定时器	20厘秒	小于Leave定时器值的一半
Leave定时器	大于Join定时器值的两倍	小于LeaveAll定时器的值
LeaveAll定时器	大于所有端口上Leave定时器的值	32760厘秒

#### 说明

- 当用户欲恢复各定时器的值为缺省值时，建议按照 Join 定时器->Leave 定时器->LeaveAll 定时器的顺序依次恢复。
- Periodic 定时器的值可以在任何时刻恢复为缺省值。

### 1.3.6 配置MVRP兼容GVRP

MVRP 允许兼容 GVRP（GARP VLAN Registration Protocol，GARP VLAN 注册协议）。当有邻接设备支持 GVRP 功能时，可以通过配置 MVRP 兼容 GVRP，允许本设备同时收发 MVRP 和 GVRP 报文。有关 GVRP 的详细介绍，请参见相关协议规范 IEEE 802.1Q。

配置 MVRP 兼容 GVRP 时，需要注意：

- 在配置 MVRP 兼容 GVRP 后，MVRP 功能只能与 STP 或 RSTP 配合使用，而不能与 MSTP 配合使用，否则可能会造成网络工作的不正常。
- 在配置 MVRP 兼容 GVRP 后，建议关闭 Periodic 定时器，否则当系统繁忙时，容易造成 VLAN 状态的频繁改变。

表1-6 配置 MVRP 兼容 GVRP

操作	命令	说明
进入系统视图	<b>system-view</b>	-
配置MVRP兼容GVRP	<b>mvrp gvrp-compliance enable</b>	缺省情况下，MVRP不兼容GVRP

## 1.4 MVRP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 MVRP 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 MVRP 的统计信息。

表1-7 MVRP 显示和维护

操作	命令
显示MVRP运行状态信息	<b>display mvrp running-status</b> [ <b>interface</b> <i>interface-list</i> ]
显示端口在指定VLAN内的MVRP接口状态信息	<b>display mvrp state interface</b> <i>interface-type</i> <i>interface-number</i> <b>vlan</b> <i>vlan-id</i>
显示MVRP统计信息	<b>display mvrp statistics</b> [ <b>interface</b> <i>interface-list</i> ]
清除端口上的MVRP统计信息	<b>reset mvrp statistics</b> [ <b>interface</b> <i>interface-list</i> ]

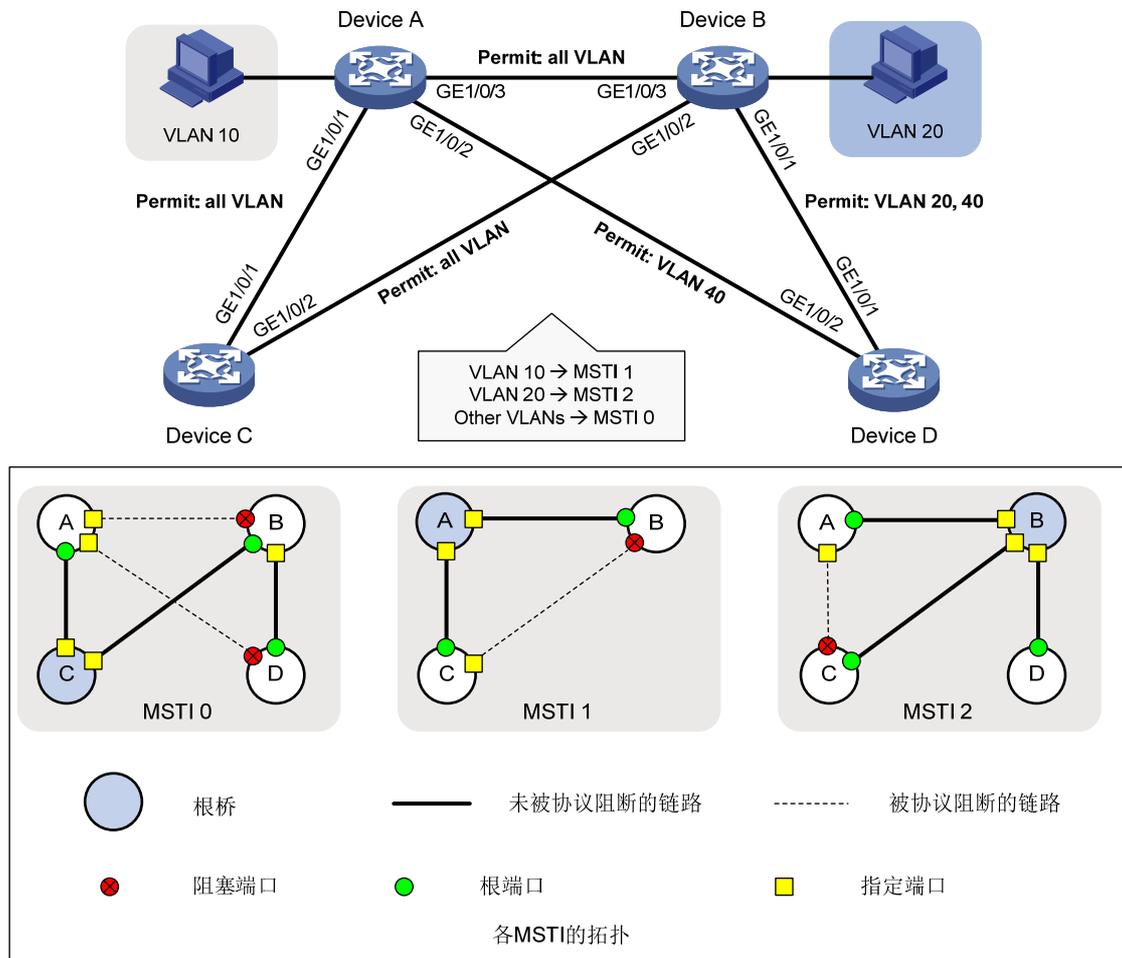
## 1.5 MVRP典型配置举例

### 1. 组网需求

- 设备A上创建了VLAN 10、设备B上创建了VLAN 20，各台设备上各个端口允许通过的VLAN如 [图 1-2](#)所示。
- 通过配置 MSTP，使不同 VLAN 的报文按照不同的 MSTI 转发：VLAN 10 的报文沿 MSTI 1 转发，VLAN 20 沿 MSTI 2 转发，其他 VLAN 沿 MSTI 0 转发。
- 通过启用 MVRP 功能，并配置 MVRP 的注册模式为 Normal 模式，来实现 Device A、Device B、Device C 和 Device D 之间的所有动态 VLAN 的注册和注销，从而保持各 MSTI 中 VLAN 配置的一致。
- 在网络稳定后，配置 Device B 上与 Device A 相连端口的 MVRP 注册模式为 Fixed 模式，使该端口注册的动态 VLAN 不被注销。

## 2. 组网图

图1-2 MVRP 配置组网图



## 3. 配置步骤

### (1) 配置 Device A

# 进入 MST 域视图。

```
<DeviceA> system-view
```

```
[DeviceA] stp region-configuration
```

# 配置 MST 域的域名、VLAN 映射关系和修订级别。

```
[DeviceA-mst-region] region-name example
```

```
[DeviceA-mst-region] instance 1 vlan 10
```

```
[DeviceA-mst-region] instance 2 vlan 20
```

```
[DeviceA-mst-region] revision-level 0
```

# 手工激活 MST 域的配置。

```
[DeviceA-mst-region] active region-configuration
```

```
[DeviceA-mst-region] quit
```

# 定义 Device A 为 MSTI 1 的根桥。

```
[DeviceA] stp instance 1 root primary
```

# 全局开启生成树协议。

```

[DeviceA] stp global enable
# 全局开启 MVRP 功能。
[DeviceA] mvrp global enable
# 将端口 GigabitEthernet1/0/1 配置为 Trunk 口，并允许所有 VLAN 通过。
[DeviceA] interface gigabitethernet 1/0/1
[DeviceA-GigabitEthernet1/0/1] port link-type trunk
[DeviceA-GigabitEthernet1/0/1] port trunk permit vlan all
# 在端口 GigabitEthernet1/0/1 上开启 MVRP。
[DeviceA-GigabitEthernet1/0/1] mvrp enable
[DeviceA-GigabitEthernet1/0/1] quit
# 将端口 GigabitEthernet1/0/2 配置为 Trunk 口，并允许 VLAN 40 通过。
[DeviceA] interface gigabitethernet 1/0/2
[DeviceA-GigabitEthernet1/0/2] port link-type trunk
[DeviceA-GigabitEthernet1/0/2] port trunk permit vlan 40
# 在端口 GigabitEthernet1/0/2 上开启 MVRP。
[DeviceA-GigabitEthernet1/0/2] mvrp enable
[DeviceA-GigabitEthernet1/0/2] quit
# 将端口 GigabitEthernet1/0/3 配置为 Trunk 口，并允许所有 VLAN 通过。
[DeviceA] interface gigabitethernet 1/0/3
[DeviceA-GigabitEthernet1/0/3] port link-type trunk
[DeviceA-GigabitEthernet1/0/3] port trunk permit vlan all
# 在端口 GigabitEthernet1/0/3 上开启 MVRP。
[DeviceA-GigabitEthernet1/0/3] mvrp enable
[DeviceA-GigabitEthernet1/0/3] quit
# 创建 VLAN 10。

```

```

[DeviceA] vlan 10
[DeviceA-vlan10] quit

```

## (2) 配置 Device B

```

# 进入 MST 域视图。
<DeviceB> system-view
[DeviceB] stp region-configuration
# 配置 MST 域的域名、VLAN 映射关系和修订级别。
[DeviceB-mst-region] region-name example
[DeviceB-mst-region] instance 1 vlan 10
[DeviceB-mst-region] instance 2 vlan 20
[DeviceB-mst-region] revision-level 0
# 手工激活 MST 域的配置。
[DeviceB-mst-region] active region-configuration
[DeviceB-mst-region] quit
# 定义 Device B 为 MSTI 2 的根桥。
[DeviceB] stp instance 2 root primary
# 全局开启生成树协议。
[DeviceB] stp global enable
# 开启全局 MVRP 功能。

```

```

[DeviceB] mvrp global enable
# 将端口 GigabitEthernet1/0/1 配置为 Trunk 口，并允许 VLAN 20、VLAN 40 通过。
[DeviceB] interface gigabitethernet 1/0/1
[DeviceB-GigabitEthernet1/0/1] port link-type trunk
[DeviceB-GigabitEthernet1/0/1] port trunk permit vlan 20 40
# 在端口 GigabitEthernet1/0/1 上开启 MVRP。
[DeviceB-GigabitEthernet1/0/1] mvrp enable
[DeviceB-GigabitEthernet1/0/1] quit
# 将端口 GigabitEthernet1/0/2 配置为 Trunk 口，并允许所有 VLAN 通过。
[DeviceB] interface gigabitethernet 1/0/2
[DeviceB-GigabitEthernet1/0/2] port link-type trunk
[DeviceB-GigabitEthernet1/0/2] port trunk permit vlan all
# 在端口 GigabitEthernet1/0/2 上开启 MVRP。
[DeviceB-GigabitEthernet1/0/2] mvrp enable
[DeviceB-GigabitEthernet1/0/2] quit
# 将端口 GigabitEthernet1/0/3 配置为 Trunk 口，并允许所有 VLAN 通过。
[DeviceB] interface gigabitethernet 1/0/3
[DeviceB-GigabitEthernet1/0/3] port link-type trunk
[DeviceB-GigabitEthernet1/0/3] port trunk permit vlan all
# 在端口 GigabitEthernet1/0/3 上开启 MVRP。
[DeviceB-GigabitEthernet1/0/3] mvrp enable
[DeviceB-GigabitEthernet1/0/3] quit
# 创建 VLAN 20。
[DeviceB] vlan 20
[DeviceB-vlan20] quit

```

### (3) 配置 Device C

```

# 进入 MST 域视图。
<DeviceC> system-view
[DeviceC] stp region-configuration
# 配置 MST 域的域名、VLAN 映射关系和修订级别。
[DeviceC-mst-region] region-name example
[DeviceC-mst-region] instance 1 vlan 10
[DeviceC-mst-region] instance 2 vlan 20
[DeviceC-mst-region] revision-level 0
# 手工激活 MST 域的配置。
[DeviceC-mst-region] active region-configuration
[DeviceC-mst-region] quit
# 定义 Device C 为 MSTI 0 的根桥。
[DeviceC] stp instance 0 root primary
# 全局开启生成树协议。
[DeviceC] stp global enable
# 全局开启 MVRP 功能。
[DeviceC] mvrp global enable
# 将端口 GigabitEthernet1/0/1 配置为 Trunk 口，并允许所有 VLAN 通过。

```

```

[DeviceC] interface gigabitethernet 1/0/1
[DeviceC-GigabitEthernet1/0/1] port link-type trunk
[DeviceC-GigabitEthernet1/0/1] port trunk permit vlan all
# 在端口 GigabitEthernet1/0/1 上开启 MVRP。
[DeviceC-GigabitEthernet1/0/1] mvrp enable
[DeviceC-GigabitEthernet1/0/1] quit
# 将端口 GigabitEthernet1/0/2 配置为 Trunk 口，并允许所有 VLAN 通过。
[DeviceC] interface gigabitethernet 1/0/2
[DeviceC-GigabitEthernet1/0/2] port link-type trunk
[DeviceC-GigabitEthernet1/0/2] port trunk permit vlan all
# 在端口 GigabitEthernet1/0/2 上开启 MVRP。
[DeviceC-GigabitEthernet1/0/2] mvrp enable
[DeviceC-GigabitEthernet1/0/2] quit
(4) 配置 Device D
# 进入 MST 域视图。
<DeviceD> system-view
[DeviceD] stp region-configuration
# 配置 MST 域的域名、VLAN 映射关系和修订级别。
[DeviceD-mst-region] region-name example
[DeviceD-mst-region] instance 1 vlan 10
[DeviceD-mst-region] instance 2 vlan 20
[DeviceD-mst-region] revision-level 0
# 手工激活 MST 域的配置。
[DeviceD-mst-region] active region-configuration
[DeviceD-mst-region] quit
# 全局开启生成树协议。
[DeviceD] stp global enable
# 全局开启 MVRP 功能。
[DeviceD] mvrp global enable
# 将端口 GigabitEthernet1/0/1 配置为 Trunk 口，并允许 VLAN 20, 40 通过。
[DeviceD] interface gigabitethernet 1/0/1
[DeviceD-GigabitEthernet1/0/1] port link-type trunk
[DeviceD-GigabitEthernet1/0/1] port trunk permit vlan 20 40
# 在端口 GigabitEthernet1/0/1 上开启 MVRP。
[DeviceD-GigabitEthernet1/0/1] mvrp enable
[DeviceD-GigabitEthernet1/0/1] quit
# 将端口 GigabitEthernet1/0/2 配置为 Trunk 口，并允许 VLAN 40 通过。
[DeviceD] interface gigabitethernet 1/0/2
[DeviceD-GigabitEthernet1/0/2] port link-type trunk
[DeviceD-GigabitEthernet1/0/2] port trunk permit vlan 40
# 在端口 GigabitEthernet1/0/2 上开启 MVRP。
[DeviceD-GigabitEthernet1/0/2] mvrp enable
[DeviceD-GigabitEthernet1/0/2] quit

```

## 4. 验证配置

### (1) 验证 Normal 注册模式配置

通过使用 **display mvrp running-status** 命令可以查看 MVRP 本地 VLAN 的信息, 验证配置是否生效。

# 查看 Device A 上的本地 VLAN 信息。

```
[DeviceA] display mvrp running-status
-----[MVRP Global Info]-----
Global Status      : Enabled
Compliance-GVRP   : False

----[GigabitEthernet1/0/1]----
Config Status      : Enabled
Running Status     : Enabled
Join Timer         : 20 (centiseconds)
Leave Timer         : 60 (centiseconds)
Periodic Timer     : 100 (centiseconds)
LeaveAll Timer      : 1000 (centiseconds)
Registration Type  : Normal
Registered VLANs  :
  1(default)
Declared VLANs    :
  1(default), 10, 20
Propagated VLANs  :
  1(default)

----[GigabitEthernet1/0/2]----
Config Status      : Enabled
Running Status     : Enabled
Join Timer         : 20 (centiseconds)
Leave Timer         : 60 (centiseconds)
Periodic Timer     : 100 (centiseconds)
LeaveAll Timer      : 1000 (centiseconds)
Registration Type  : Normal
Registered VLANs  :
  None
Declared VLANs    :
  1(default)
Propagated VLANs  :
  None

----[GigabitEthernet1/0/3]----
Config Status      : Enabled
Running Status     : Enabled
Join Timer         : 20 (centiseconds)
Leave Timer         : 60 (centiseconds)
Periodic Timer     : 100 (centiseconds)
LeaveAll Timer      : 1000 (centiseconds)
```

```
Registration Type           : Normal
Registered VLANs :
  20
Declared VLANs :
  1(default), 10
Propagated VLANs :
  20
```

由此可见，端口 **GigabitEthernet1/0/1** 注册了 VLAN 1，向外声明了 VLAN 1、VLAN 10 和 VLAN 20，传播了 VLAN 1。端口 **GigabitEthernet1/0/2** 没有注册任何 VLAN，向外声明了 VLAN 1，没有传播 VLAN。端口 **GigabitEthernet1/0/3** 注册了 VLAN 20，向外声明了 VLAN 1 和 VLAN 10，传播了 VLAN 20。

# 查看 Device B 上的本地 VLAN 信息。

```
[DeviceB] display mvrp running-status
-----[MVRP Global Info]-----
Global Status       : Enabled
Compliance-GVRP    : False

----[GigabitEthernet1/0/1]----
Config Status      : Enabled
Running Status     : Enabled
Join Timer         : 20 (centiseconds)
Leave Timer         : 60 (centiseconds)
Periodic Timer     : 100 (centiseconds)
LeaveAll Timer      : 1000 (centiseconds)
Registration Type   : Normal
Registered VLANs :
  1(default)
Declared VLANs :
  1(default), 20
Propagated VLANs :
  1(default)

----[GigabitEthernet1/0/2]----
Config Status      : Enabled
Running Status     : Enabled
Join Timer         : 20 (centiseconds)
Leave Timer         : 60 (centiseconds)
Periodic Timer     : 100 (centiseconds)
LeaveAll Timer      : 1000 (centiseconds)
Registration Type   : Normal
Registered VLANs :
  1(default), 10
Declared VLANs :
  1(default), 20
Propagated VLANs :
  1(default)
```

```

----[GigabitEthernet1/0/3]----
Config Status           : Enabled
Running Status          : Enabled
Join Timer               : 20 (centiseconds)
Leave Timer               : 60 (centiseconds)
Periodic Timer           : 100 (centiseconds)
LeaveAll Timer            : 1000 (centiseconds)
Registration Type        : Normal
Registered VLANs :
  1(default), 10
Declared VLANs :
  20
Propagated VLANs :
  10

```

由此可见，端口 GigabitEthernet1/0/1 注册了 VLAN 1，向外声明了 VLAN 1 和 VLAN 20，传播了 VLAN 1。端口 GigabitEthernet1/0/2 注册了 VLAN 1 和 VLAN 10，向外声明了 VLAN 1 和 VLAN 20，传播了 VLAN 1。端口 GigabitEthernet1/0/3 注册了 VLAN 1 和 VLAN 10，向外声明了 VLAN 20，传播了 VLAN 10。

# 查看 Device C 上的本地 VLAN 信息。

```

[DeviceC] display mvrp running-status
-----[MVRP Global Info]-----
Global Status          : Enabled
Compliance-GVRP       : False

----[GigabitEthernet1/0/1]----
Config Status           : Enabled
Running Status          : Enabled
Join Timer               : 20 (centiseconds)
Leave Timer               : 60 (centiseconds)
Periodic Timer           : 100 (centiseconds)
LeaveAll Timer            : 1000 (centiseconds)
Registration Type        : Normal
Registered VLANs :
  1(default), 10, 20
Declared VLANs :
  1(default)
Propagated VLANs :
  1(default), 10

----[GigabitEthernet1/0/2]----
Config Status           : Enabled
Running Status          : Enabled
Join Timer               : 20 (centiseconds)
Leave Timer               : 60 (centiseconds)
Periodic Timer           : 100 (centiseconds)
LeaveAll Timer            : 1000 (centiseconds)
Registration Type        : Normal

```

```
Registered VLANs :
  1(default), 20
Declared VLANs :
  1(default), 10
Propagated VLANs :
  1(default), 20
```

由此可见, 端口 **GigabitEthernet1/0/1** 注册了 VLAN 1、VLAN 10 和 VLAN 20, 向外声明了 VLAN 1, 传播了 VLAN 1 和 VLAN 10。端口 **GigabitEthernet1/0/2** 注册了 VLAN 1 和 VLAN 20, 向外声明了 VLAN 1 和 VLAN 10, 传播了 VLAN 1 和 VLAN 20。

# 查看 Device D 上的本地 VLAN 信息。

```
[DeviceD] display mvrp running-status
-----[MVRP Global Info]-----
Global Status      : Enabled
Compliance-GVRP   : False

----[GigabitEthernet1/0/1]----
Config Status      : Enabled
Running Status     : Enabled
Join Timer         : 20 (centiseconds)
Leave Timer         : 60 (centiseconds)
Periodic Timer     : 100 (centiseconds)
LeaveAll Timer      : 1000 (centiseconds)
Registration Type  : Normal
Registered VLANs  :
  1(default), 20
Declared VLANs    :
  1(default)
Propagated VLANs  :
  1(default), 20

----[GigabitEthernet1/0/2]----
Config Status      : Enabled
Running Status     : Enabled
Join Timer         : 20 (centiseconds)
Leave Timer         : 60 (centiseconds)
Periodic Timer     : 100 (centiseconds)
LeaveAll Timer      : 1000 (centiseconds)
Registration Type  : Normal
Registered VLANs  :
  1(default)
Declared VLANs    :
  None
Propagated VLANs  :
  None
```

由此可见, 端口 **GigabitEthernet1/0/1** 注册并传播了 VLAN 10 和 VLAN 20, 向外声明 VLAN 1。端口 **GigabitEthernet1/0/2** 注册了 VLAN 1, 没有向外声明和传播 VLAN。

(2) 更改注册模式并验证

配置 Device B 上端口 GigabitEthernet1/0/3 的 MVRP 注册模式为 Fixed 模式,使其注册的动态 VLAN 不被注销。

# 配置端口 GigabitEthernet1/0/3 的 MVRP 注册模式为 Fixed 模式。

```
[DeviceB] interface gigabitethernet 1/0/3
[DeviceB-GigabitEthernet1/0/3] mvrp registration fixed
[DeviceB-GigabitEthernet1/0/3] quit
```

# 查看 Device B 的端口 GigabitEthernet1/0/3 上 MVRP 本地 VLAN 的信息。

```
[DeviceB] display mvrp running-status interface gigabitethernet 1/0/3
-----[MVRP Global Info]-----
Global Status      : Enabled
Compliance-GVRP   : False

----[GigabitEthernet1/0/3]----
Config Status      : Enabled
Running Status     : Enabled
Join Timer         : 20 (centiseconds)
Leave Timer         : 60 (centiseconds)
Periodic Timer     : 100 (centiseconds)
LeaveAll Timer      : 1000 (centiseconds)
Registration Type   : Fixed
Registered VLANs   :
  1(default), 10
Declared VLANs    :
  20
Propagated VLANs  :
  10
```

由此可见,此时端口 GigabitEthernet1/0/3 上的 VLAN 信息与没有配置 Fixed 模式时的 VLAN 信息相同。

# 在 Device A 上删除 VLAN 10。

```
[DeviceA] undo vlan 10
```

# 查看 Device B 的端口 GigabitEthernet1/0/3 上 MVRP 本地 VLAN 的信息。

```
[DeviceB] display mvrp running-status interface gigabitethernet 1/0/3
-----[MVRP Global Info]-----
Global Status      : Enabled
Compliance-GVRP   : False

----[GigabitEthernet1/0/3]----
Config Status      : Enabled
Running Status     : Enabled
Join Timer         : 20 (centiseconds)
Leave Timer         : 60 (centiseconds)
Periodic Timer     : 100 (centiseconds)
LeaveAll Timer      : 1000 (centiseconds)
Registration Type   : Fixed
Registered VLANs   :
  1(default), 10
Declared VLANs    :
```

20

Propagated VLANs :

10

由此可见，端口 **GigabitEthernet1/0/3** 配置 **Fixed** 模式后，该端口注册的动态 VLAN 信息不会发生变化。

# 目 录

1 QinQ.....	1-1
1.1 QinQ简介.....	1-1
1.1.1 QinQ的工作原理.....	1-1
1.1.2 QinQ的实现方式.....	1-2
1.1.3 协议规范.....	1-3
1.2 配置QinQ功能.....	1-3
1.2.1 使能QinQ功能.....	1-3
1.2.2 配置VLAN透传功能.....	1-3
1.3 配置VLAN Tag的TPID值.....	1-4
1.4 QinQ显示和维护.....	1-5
1.5 QinQ典型配置举例.....	1-5
1.5.1 QinQ配置举例.....	1-5
1.5.2 VLAN透传配置举例.....	1-8

# 1 QinQ

## 1.1 QinQ简介

IEEE 802.1Q 定义的 VLAN ID 域有 12 个比特，最多可以提供 4094 个 VLAN。但在实际应用中，尤其是在城域网中，需要大量的 VLAN 来隔离用户，4094 个 VLAN 远远不能满足需求。QinQ 使整个网络最多可以提供  $4094 \times 4094$  个 VLAN，满足了城域网对 VLAN 数量的需求。

QinQ 是 802.1Q in 802.1Q 的简称，是基于 IEEE 802.1Q 技术的一种比较简单的二层 VPN (Virtual Private Network, 虚拟专用网络) 协议。通过将一层 VLAN Tag 封装到私网报文上，使其携带两层 VLAN Tag 穿越运营商的骨干网络(又称公网),从而使运营商能够利用一个 VLAN 为包含多个 VLAN 的用户网络提供服务。

QinQ 具备以下优点:

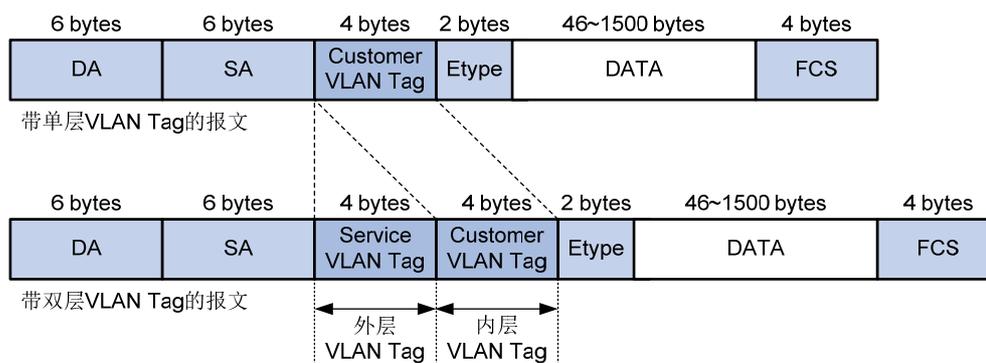
- 缓解公网 VLAN 资源日益紧缺的问题。
- 用户可以规划自己的私网 VLAN，不会导致与公网 VLAN 冲突。
- 为用户提供了一种简单、灵活的二层 VPN 解决方案。
- 当运营商进行 VLAN 规划时，用户网络不必更改原有配置，使用户网络具有了较强的独立性。

### 1.1.1 QinQ的工作原理

如 图 1-1 所示，QinQ 报文在运营商网络中传输时带有双层 VLAN Tag:

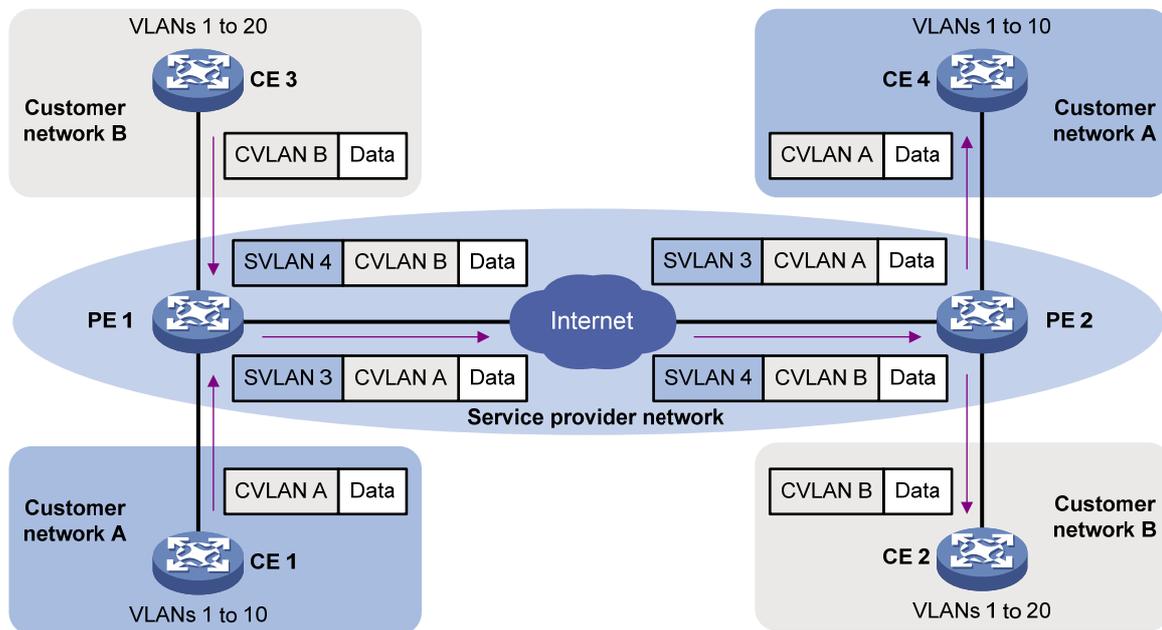
- 内层 VLAN Tag: 为用户的私网 VLAN Tag，对应图中的 Customer VLAN Tag (简称 CVLAN)。设备依靠该 Tag 在私网中传送报文。
- 外层 VLAN Tag: 为运营商分配给用户的公网 VLAN Tag，对应图中的 Service VLAN Tag (简称 SVLAN)。设备依靠该 Tag 在公网中传送 QinQ 报文。

图1-1 QinQ 的报文结构



在公网的传输过程中，设备只根据外层 VLAN Tag 转发报文，而内层 VLAN Tag 将被当作报文的数据部分进行传输。

图1-2 QinQ 典型应用组网图



如 图 1-2 所示，用户网络A和B的私网VLAN分别为VLAN 1~10 和VLAN 1~20。运营商为用户网络A和B分配的公网VLAN分别为VLAN 3 和VLAN 4。

- (1) 当用户网络 A 和 B 中带私网 VLAN Tag 的报文进入运营商网络时，报文外面就会被分别封装上 VLAN 3 和 VLAN 4 的公网 VLAN Tag。
- (2) 来自不同用户网络的报文在运营商网络中传输时被隔离，即使这些用户网络各自的 VLAN 范围存在重叠，因为分配到的公网 VLAN 不同，在运营商网络中传输时也不会产生冲突。
- (3) 当报文穿过运营商网络，到达运营商网络另一侧 PE（Provider Edge，服务提供商网络边缘）设备后，报文被剥离公网 VLAN Tag，然后再传送给用户网络的 CE（Customer Edge，用户网络边缘）设备。

### 1.1.2 QinQ的实现方式

当端口上配置了 QinQ 功能后，不论从该端口收到的报文是否带有 VLAN Tag，设备都会为该报文添加本端口缺省 VLAN 的 Tag：

- 如果收到的是带有 VLAN Tag 的报文，该报文就成为带两层 Tag 的报文；
- 如果收到的是不带 VLAN Tag 的报文，该报文就成为带有本端口缺省 VLAN Tag 的报文。

QinQ 功能是以端口来划分用户或用户网络，但当多个不同用户以不同的 VLAN 接入到同一个端口时则无法区分用户。如果需要为不同用户的 VLAN 的报文添加不同的外层 VLAN Tag，可以通过 1:2 VLAN 映射或 QoS Nest 功能实现。

- 为不同内层 VLAN Tag 的报文添加不同的外层 VLAN Tag，建议使用 1:2 VLAN 映射功能。
- 如果运营商网络需要使用 VLAN ID 之外的匹配条件更灵活的匹配用户网络报文，或在为报文添加外层 VLAN Tag 时，需要同时配置其它流行为，请使用 QoS Nest 功能。

对于带有两层 VLAN Tag 的报文，如果需要修改内层或外层 VLAN ID，或同时修改内外层 VLAN ID，可以通过 2:2 VLAN 映射功能实现。需要注意的是：QinQ 功能和 2:2 VLAN 映射功能互斥。即设备不支持在同一端口上先为报文添加外层 VLAN Tag，再修改该报文的内、外层 VLAN ID。

有关 VLAN 映射的详细介绍，请参见“二层技术-以太网交换配置指导”中的“VLAN 映射”。有关 QoS 的详细介绍，请参见“ACL 和 QoS 配置指导”中的“QoS”。

### 1.1.3 协议规范

与 QinQ 相关的协议规范有：

- IEEE 802.1Q: IEEE Standard for Local and Metropolitan Area Networks-Virtual Bridged Local Area Networks
- IEEE 802.1ad: IEEE Standard for Local and Metropolitan Area Networks-Virtual Bridged Local Area Networks-Amendment 4: Provider Bridges

## 1.2 配置QinQ功能

QinQ 功能应在 PE 设备的用户网络侧接口上进行配置。

配置 QinQ 功能时，需要注意：

- QinQ 为报文加上外层 VLAN Tag 后，内层 VLAN Tag 将被当作报文的数据部分进行传输，报文长度将增加 4 个字节。因此建议用户适当增加 QinQ 报文传输路径上各接口的 MTU（Maximum Transmission Unit，最大传输单元）值（至少为 1504 字节）。
- 若用户同时通过配置 QinQ 和 VLAN 映射或 QoS 策略来添加报文的 VLAN Tag，且配置冲突时，按如下优先顺序配置生效：QoS 策略->VLAN 映射->QinQ。

### 1.2.1 使能QinQ功能

使能了 QinQ 功能的端口将为其收到的报文添加该端口缺省 VLAN 的 Tag。

使能或关闭 QinQ 功能之前，要先清除已有的 VLAN 映射表项。

表1-1 使能 QinQ 功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二层以太网接口或二层聚合接口视图	<b>interface interface-type interface-number</b>	-
使能端口的QinQ功能	<b>qinq enable</b>	缺省情况下，端口的QinQ功能处于关闭状态

### 1.2.2 配置VLAN透传功能

端口上使能了 QinQ 功能后，从该端口收到的报文就会被打上本端口缺省 VLAN 的 Tag。而 VLAN 透传功能则可使端口在收到带有指定 VLAN Tag 的报文后，不为其添加外层 VLAN Tag 而直接在运营商网络中传输。例如，当某 VLAN 为企业专线 VLAN 或网管 VLAN 时，就可以使用 VLAN 透传功能。

配置 VLAN 透传功能时，需要注意：

- 建议在配置用户侧端口的 VLAN 透传功能前，配置该端口的链路为 Trunk/Hybrid 类型，并允许透传 VLAN 通过。
- 配置 VLAN 透传功能时，还需在报文传输路径的所有端口上都配置允许透传 VLAN 通过。
- 配置了用户侧端口对指定 VLAN 的报文进行透传后，请勿在该端口上对这些 VLAN 再进行修改报文 VLAN Tag 的相关配置。
- 同一接口上同时配置 VLAN 透传和 VLAN 映射时：
  - 透传 VLAN 不能为 1:1 VLAN 映射、1:2 VLAN 映射的原始 VLAN 和转换后 VLAN。
  - 透传 VLAN 不能为 2:2 VLAN 映射的原始外层 VLAN 和转换后外层 VLAN。

表1-2 配置 VLAN 透传功能

操作		命令	说明
进入系统视图		<b>system-view</b>	-
进入二层以太网接口或二层聚合接口视图		<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
配置端口的链路类型		<b>port link-type</b> { <b>hybrid</b>   <b>trunk</b> }	缺省情况下，端口的链路类型为 Access 类型
配置端口允许透传 VLAN 和端口缺省 VLAN 通过	当端口类型为 Hybrid 端口	<b>port hybrid vlan</b> <i>vlan-id-list</i> { <b>tagged</b>   <b>untagged</b> }	二者选其一 缺省情况下，Trunk 端口只允许 VLAN 1 的报文通过；Hybrid 端口只允许该端口在链路类型为 Access 时的所属 VLAN 的报文以 Untagged 方式通过
	当端口类型为 Trunk 端口	<b>port trunk permit vlan</b> { <i>vlan-id-list</i>   <b>all</b> }	
配置端口的 VLAN 透传功能		<b>qinq transparent-vlan</b> <i>vlan-id-list</i>	缺省情况下，未配置 VLAN 透传功能

## 1.3 配置 VLAN Tag 的 TPID 值

TPID (Tag Protocol Identifier, 标签协议标识符) 值可以用来判断报文中是否带有 VLAN Tag。例如，在设备上配置用户 VLAN Tag 和运营商 VLAN Tag 的 TPID 值分别为 0x8200 和 0x9100，如果该设备收到的报文实际携带的内、外层 VLAN Tag 的 TPID 值分别为 0x8100 和 0x9100，由于该报文外层 VLAN Tag 的 TPID 值与配置值相同，而内层 VLAN Tag 的 TPID 值与配置值不同，该设备会认为该报文只携带运营商 VLAN Tag，而没有携带用户 VLAN Tag；对于该设备收到的只带有一层 VLAN Tag 的报文，如果该 VLAN Tag 的 TPID 值不为 0x9100，则该设备会认为该报文没有携带 VLAN Tag。

第三方厂商的设备可能将 QinQ 报文外层 VLAN Tag 的 TPID 设为不同的值。为了与这些厂商的设备兼容，用户可以通过修改 TPID 值，使发送到的 QinQ 报文携带的 TPID 值与第三方厂商的相同，从而实现与这些厂商的设备互通。

### 1. 配置内层 VLAN Tag 的 TPID 值

内层 VLAN Tag 的 TPID 值应在 PE 设备上配置。

需要注意的是，配置内层 VLAN Tag 的 TPID 值仅用于判断入报文是否携带 VLAN Tag，不会对报文内层 VLAN Tag 的 TPID 值进行修改。

表1-3 配置内层 VLAN Tag 的 TPID 值

操作	命令	说明
进入系统视图	<b>system-view</b>	-
配置内层VLAN Tag的TPID值	<b>qinq ethernet-type customer-tag hex-value</b>	缺省情况下，内层VLAN Tag的TPID值为0x8100

## 2. 配置外层VLAN Tag的TPID值

外层 VLAN Tag 的 TPID 值应在 PE 设备的运营商网络侧的接口上进行配置。

需要注意的是，设备不支持在同一端口上既配置外层 VLAN Tag 的 TPID 值，又使能 QinQ 功能。

表1-4 配置外层 VLAN Tag 的 TPID 值

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入以太网接口或聚合接口视图	<b>interface interface-type interface-number</b>	-
配置外层VLAN Tag的TPID值	<b>qinq ethernet-type service-tag hex-value</b>	缺省情况下，外层VLAN Tag的TPID值为0x8100

## 1.4 QinQ显示和维护

在完成上述配置后，在任意视图下执行 **display** 命令可以显示使能了 QinQ 功能的端口，通过查看显示信息验证配置的效果。

表1-5 QinQ 显示和维护

操作	命令
显示使能了QinQ功能的端口	<b>display qinq [ interface interface-type interface-number ]</b>

## 1.5 QinQ典型配置举例

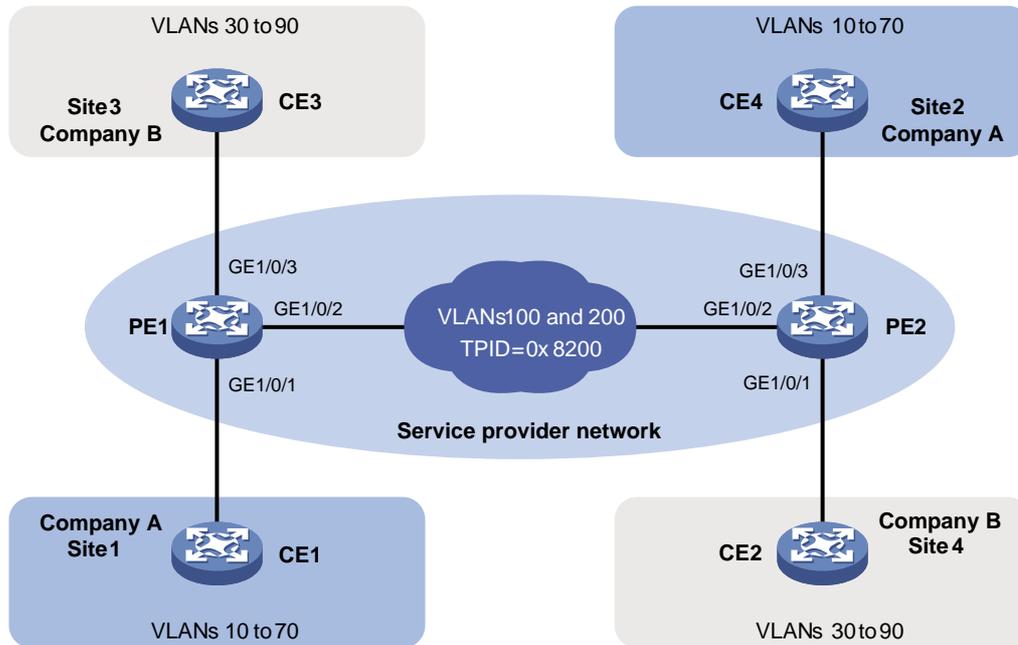
### 1.5.1 QinQ配置举例

#### 1. 组网需求

- 公司 A 的两个分支机构 Site 1 和 Site 2 通过运营商网络进行通信，该公司各业务使用的 VLAN 为 VLAN 10~70；公司 B 的两个分支机构 Site 3 和 Site 4 也通过运营商网络进行通信，该公司各业务使用的 VLAN 为 VLAN 30~90。
- PE 1 和 PE 2 为运营商网络的边缘设备，且二者通过 TPID 值为 0x8200 的运营商网络设备进行连接。
- 通过配置，利用运营商提供的 VLAN 100 使公司 A 的两个分支机构之间实现互通，利用运营商提供的 VLAN 200 使公司 B 的两个分支机构之间实现互通。

## 2. 组网图

图1-3 QinQ 配置组网图



## 3. 配置步骤

### (1) 配置 PE 1

- 配置端口 GigabitEthernet1/0/1

# 配置端口为 Trunk 端口，且允许 VLAN 100 的报文通过。

```
<PE1> system-view
[PE1] interface gigabitethernet 1/0/1
[PE1-GigabitEthernet1/0/1] port link-type trunk
[PE1-GigabitEthernet1/0/1] port trunk permit vlan 100
```

# 配置端口的缺省 VLAN 为 VLAN 100。

```
[PE1-GigabitEthernet1/0/1] port trunk pvid vlan 100
```

# 使能端口的 QinQ 功能。

```
[PE1-GigabitEthernet1/0/1] qinq enable
[PE1-GigabitEthernet1/0/1] quit
```

- 配置端口 GigabitEthernet1/0/2

# 配置端口为 Trunk 端口，且允许 VLAN 100 和 VLAN 200 的报文通过。

```
[PE1] interface gigabitethernet 1/0/2
[PE1-GigabitEthernet1/0/2] port link-type trunk
[PE1-GigabitEthernet1/0/2] port trunk permit vlan 100 200
```

# 配置外层 VLAN Tag 的 TPID 值为 0x8200。

```
[PE1-GigabitEthernet1/0/2] qinq ethernet-type service-tag 8200
[PE1-GigabitEthernet1/0/2] quit
```

- 配置端口 GigabitEthernet1/0/3

# 配置端口为 Trunk 端口，且允许 VLAN 200 的报文通过。

```
[PE1] interface gigabitethernet 1/0/3
[PE1-GigabitEthernet1/0/3] port link-type trunk
[PE1-GigabitEthernet1/0/3] port trunk permit vlan 200
# 配置端口的缺省 VLAN 为 VLAN 200。
[PE1-GigabitEthernet1/0/3] port trunk pvid vlan 200
# 使能端口的 QinQ 功能。
[PE1-GigabitEthernet1/0/3] qinq enable
[PE1-GigabitEthernet1/0/3] quit
```

## (2) 配置 PE 2

- 配置端口 GigabitEthernet1/0/1

# 配置端口为 Trunk 端口，且允许 VLAN 200 的报文通过。

```
<PE2> system-view
[PE2] interface gigabitethernet 1/0/1
[PE2-GigabitEthernet1/0/1] port link-type trunk
[PE2-GigabitEthernet1/0/1] port trunk permit vlan 200
# 配置端口的缺省 VLAN 为 VLAN 200。
[PE2-GigabitEthernet1/0/1] port trunk pvid vlan 200
# 使能端口的 QinQ 功能。
[PE2-GigabitEthernet1/0/1] qinq enable
[PE2-GigabitEthernet1/0/1] quit
```

- 配置端口 GigabitEthernet1/0/2

# 配置端口为 Trunk 端口，且允许 VLAN 100 和 VLAN 200 的报文通过。

```
[PE2] interface gigabitethernet 1/0/2
[PE2-GigabitEthernet1/0/2] port link-type trunk
[PE2-GigabitEthernet1/0/2] port trunk permit vlan 100 200
# 配置外层 VLAN Tag 的 TPID 值为 0x8200。
[PE2-GigabitEthernet1/0/2] qinq ethernet-type service-tag 8200
[PE2-GigabitEthernet1/0/2] quit
```

- 配置端口 GigabitEthernet1/0/3

# 配置端口为 Trunk 端口，且允许 VLAN 100 的报文通过。

```
[PE2] interface gigabitethernet 1/0/3
[PE2-GigabitEthernet1/0/3] port link-type trunk
[PE2-GigabitEthernet1/0/3] port trunk permit vlan 100
# 配置端口的缺省 VLAN 为 VLAN 100。
[PE2-GigabitEthernet1/0/3] port trunk pvid vlan 100
# 使能端口的 QinQ 功能。
[PE2-GigabitEthernet1/0/3] qinq enable
[PE2-GigabitEthernet1/0/3] quit
```

## (3) 配置公共网络设备

配置运营商网络中 PE 1 到 PE 2 之间路径上的设备端口都允许 VLAN 100 和 VLAN 200 的报文携带 VLAN Tag 通过，且这些端口的 MTU 值至少为 1504 字节。

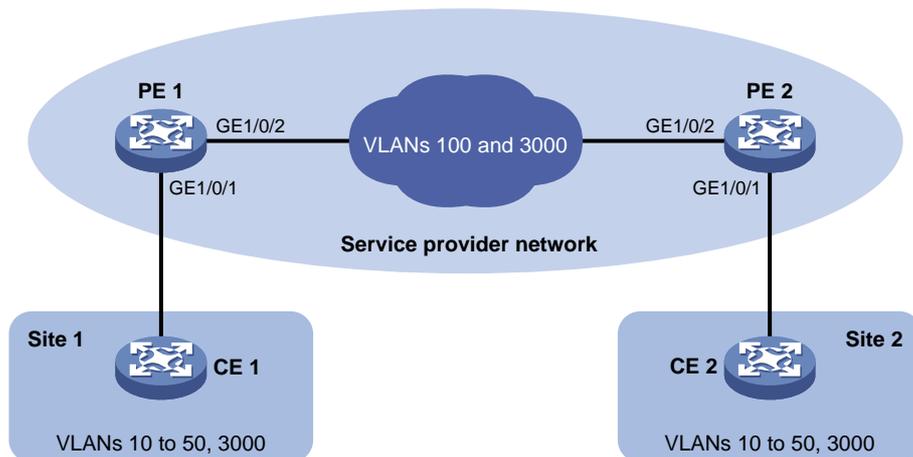
## 1.5.2 VLAN透传配置举例

### 1. 组网需求

- 某公司的两个分支机构 **Site 1** 和 **Site 2** 通过运营商网络进行通信，该公司各业务使用的 VLAN 为 VLAN 10~50 和 VLAN 3000，其中 VLAN 3000 为企业专线 VLAN。
- PE 1 和 PE 2 为运营商网络的边缘设备。
- 通过配置，使公司使用的 VLAN 10~50 利用运营商提供的 VLAN 100 实现互通，VLAN 3000 不利用运营商提供的 VLAN 就能实现互通。

### 2. 组网图

图1-4 VLAN 透传配置组网图



### 3. 配置步骤

#### (1) 配置 PE 1

- 配置端口 GigabitEthernet1/0/1

# 配置端口为 Trunk 端口，且允许 VLAN 100 和 VLAN 3000 的报文通过。

```
<PE1> system-view
[PE1] interface gigabitethernet 1/0/1
[PE1-GigabitEthernet1/0/1] port link-type trunk
[PE1-GigabitEthernet1/0/1] port trunk permit vlan 100 3000
```

# 配置端口的缺省 VLAN 为 VLAN 100。

```
[PE1-GigabitEthernet1/0/1] port trunk pvid vlan 100
```

# 使能端口的 QinQ 功能。

```
[PE1-GigabitEthernet1/0/1] qinq enable
```

# 配置端口对 VLAN 3000 的报文进行透传。

```
[PE1-GigabitEthernet1/0/1] qinq transparent-vlan 3000
[PE1-GigabitEthernet1/0/1] quit
```

- 配置端口 GigabitEthernet1/0/2

# 配置端口为 Trunk 端口，且允许 VLAN 100 和 VLAN 3000 的报文通过。

```
[PE1] interface gigabitethernet 1/0/2
[PE1-GigabitEthernet1/0/2] port link-type trunk
```

```
[PE1-GigabitEthernet1/0/2] port trunk permit vlan 100 3000
[PE1-GigabitEthernet1/0/2] quit
```

## (2) 配置 PE 2

- 配置端口 GigabitEthernet1/0/1

# 配置端口为 Trunk 端口，且允许 VLAN 100 和 VLAN 3000 的报文通过。

```
<PE2> system-view
[PE2] interface gigabitethernet 1/0/1
[PE2-GigabitEthernet1/0/1] port link-type trunk
[PE2-GigabitEthernet1/0/1] port trunk permit vlan 100 3000
```

# 配置端口的缺省 VLAN 为 VLAN 100。

```
[PE2-GigabitEthernet1/0/1] port trunk pvid vlan 100
```

# 使能端口的 QinQ 功能。

```
[PE2-GigabitEthernet1/0/1] qinq enable
```

# 配置端口对 VLAN 3000 的报文进行透传。

```
[PE2-GigabitEthernet1/0/1] qinq transparent-vlan 3000
[PE2-GigabitEthernet1/0/1] quit
```

- 配置端口 GigabitEthernet1/0/2

# 配置端口为 Trunk 端口，且允许 VLAN 100 和 VLAN 3000 的报文通过。

```
[PE2] interface gigabitethernet 1/0/2
[PE2-GigabitEthernet1/0/2] port link-type trunk
[PE2-GigabitEthernet1/0/2] port trunk permit vlan 100 3000
[PE2-GigabitEthernet1/0/2] quit
```

## (3) 配置公共网络设备

配置运营商网络中 PE 1 到 PE 2 之间路径上的设备端口都允许 VLAN 100 和 VLAN 3000 的报文携带 VLAN Tag 通过，且这些端口的 MTU 值至少为 1504 字节。

# 目 录

1 VLAN映射.....	1-1
1.1 VLAN映射简介.....	1-1
1.1.1 VLAN映射的应用.....	1-1
1.1.2 VLAN映射实现方式.....	1-3
1.2 VLAN映射配置任务简介.....	1-5
1.3 配置VLAN映射.....	1-6
1.3.1 配置 1:1 VLAN映射.....	1-6
1.3.2 配置 1:2 VLAN映射.....	1-6
1.3.3 配置 2:2 VLAN映射.....	1-7
1.4 VLAN映射显示和维护.....	1-8
1.5 VLAN映射典型配置举例.....	1-8
1.5.1 1:1 VLAN映射配置举例.....	1-8
1.5.2 1:2 和 2:2 VLAN映射配置举例.....	1-10

# 1 VLAN映射

## 1.1 VLAN映射简介

VLAN 映射(VLAN Mapping)也叫做 VLAN 转换(VLAN Translation),它可以修改报文携带的 VLAN Tag 或为报文添加 VLAN Tag,实现不同 VLAN ID 之间的相互转换。目前设备提供下面几种映射关系:

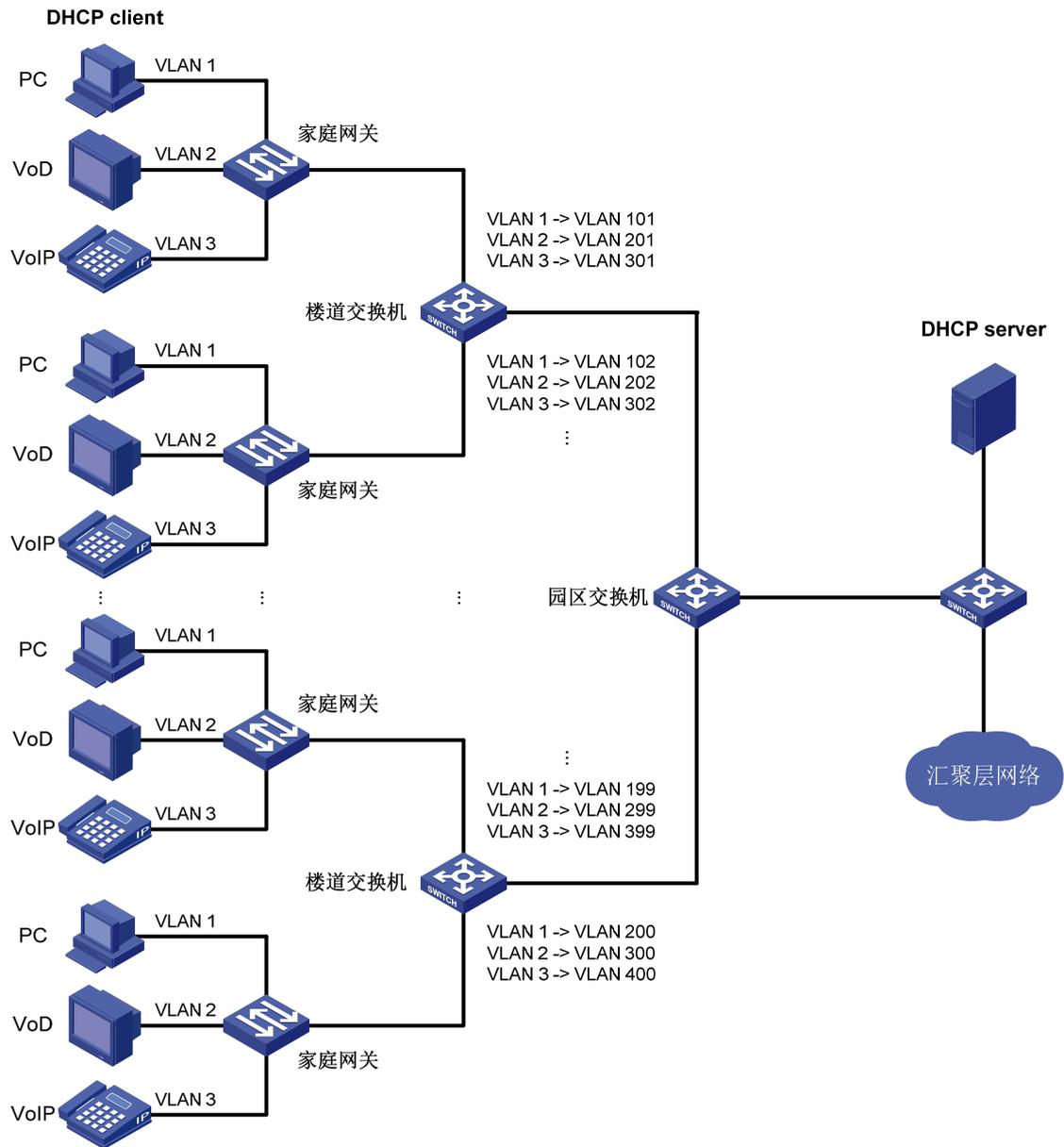
- 1:1 VLAN 映射: 将来自某一特定 VLAN 的报文所携带的 VLAN Tag 替换为新的 VLAN Tag。
- 1:2 VLAN 映射: 为携带有一层 VLAN Tag 的报文添加外层 VLAN Tag,使报文携带两层 VLAN Tag。
- 2:2 VLAN 映射: 将携带有两层 VLAN Tag 的报文的内、外层 VLAN Tag 都替换为新的 VLAN Tag。

### 1.1.1 VLAN映射的应用

#### 1. 1:1 VLAN映射的应用

1:1 VLAN映射的应用环境如 [图 1-1](#)所示,小区实现宽带上网业务。

图1-1 1:1 VLAN映射应用示意图



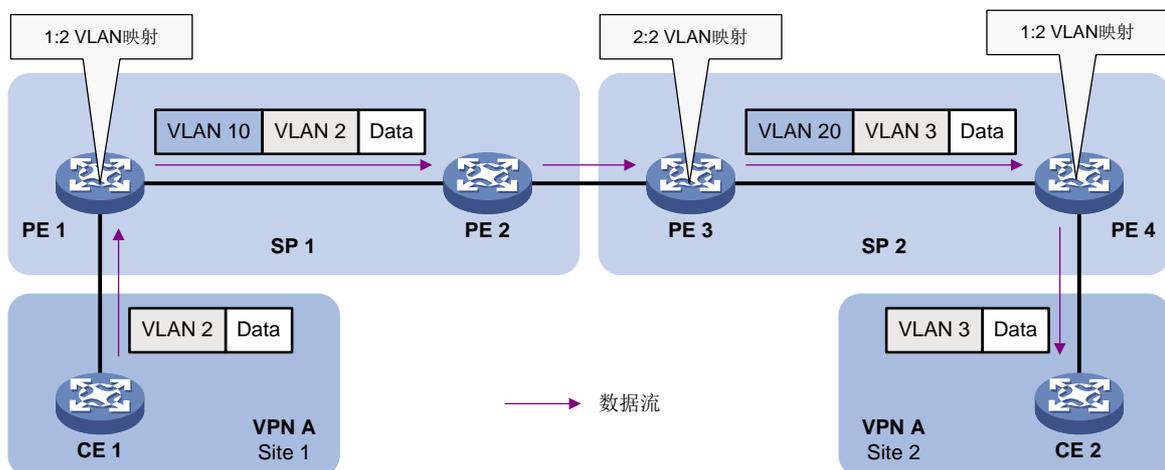
在图 1-1 中，进行了如下网络规划：

- 在家庭网关上，分别将电脑上网（PC）、视频点播（VoD）和语音电话（VoIP）业务依次划分到不同 VLAN。
- 在楼道交换机上，为了隔离不同家庭的同类业务，需要将每个家庭的每种业务都划分到不同的 VLAN，即进行 1:1 VLAN 映射，这就要用到大量的 VLAN。

## 2. 1:2 和 2:2 VLAN映射的应用

1:2 和 2:2 VLAN映射的应用环境如图 1-2 所示，VPN A中处于不同地理位置（Site 1 和Site 2）的用户跨越了两个SP（Service Provider，服务提供商）——SP 1 和SP 2 的网络进行互通。

图1-2 1:2 和 2:2 VLAN 映射应用示意图



在图 1-2 中，Site 1 和 Site 2 中的用户所在的 VLAN 分别为 VLAN 2 和 VLAN 3，SP 1 和 SP 2 分配给 VPN A 的 VLAN 分别为 VLAN 10 和 VLAN 20。当 Site 1 的报文进入 SP 1 的网络后，PE 1 为该报文添加了 VLAN 10 的 VLAN Tag，这个过程就是 1:2 VLAN 映射。这样，VPN 用户就可以自由规划自己网络中的 VLAN ID，而不用担心与 SP 的 VLAN ID 相冲突，同时也因为此时报文携带两层 VLAN Tag，网络可用的 VLAN 为  $4094 \times 4094$  个，缓解了原先 SP 网络中可用 VLAN 只有 4094 个带来的 VLAN 资源紧缺的问题。

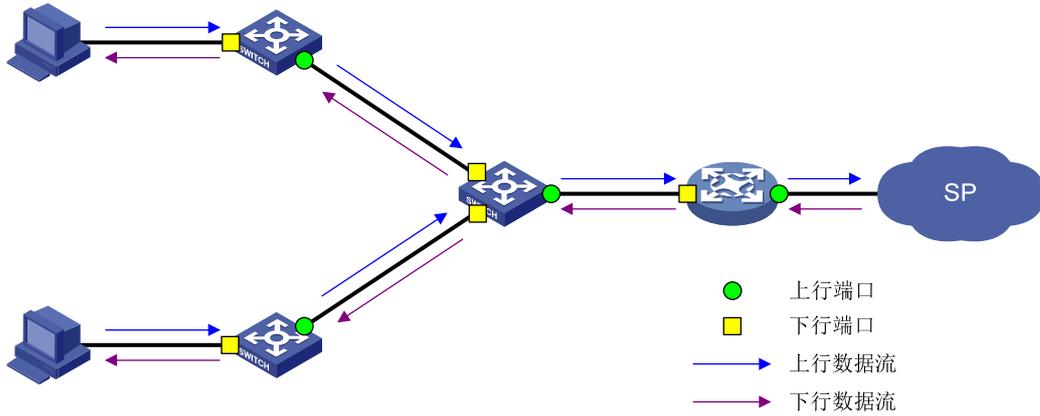
当上述报文继续由 SP 1 的网络进入 SP 2 的网络后，由于 SP 2 分配给 VPN A 的 VLAN 与 SP 1 不同，另外为了实现 Site 1 与 Site 2 中的用户互通，需要同时修改该报文的内、外两层 VLAN Tag，也就是进行 2:2 VLAN 映射。具体过程为，在 PE 3 上需将该报文的外层 VLAN Tag 替换为 VLAN 20 的 VLAN Tag，同时将其内层 VLAN Tag 替换为 VLAN 3 的 VLAN Tag。

### 1.1.2 VLAN 映射实现方式

如图 1-3 所示，VLAN 映射有如下相关概念：

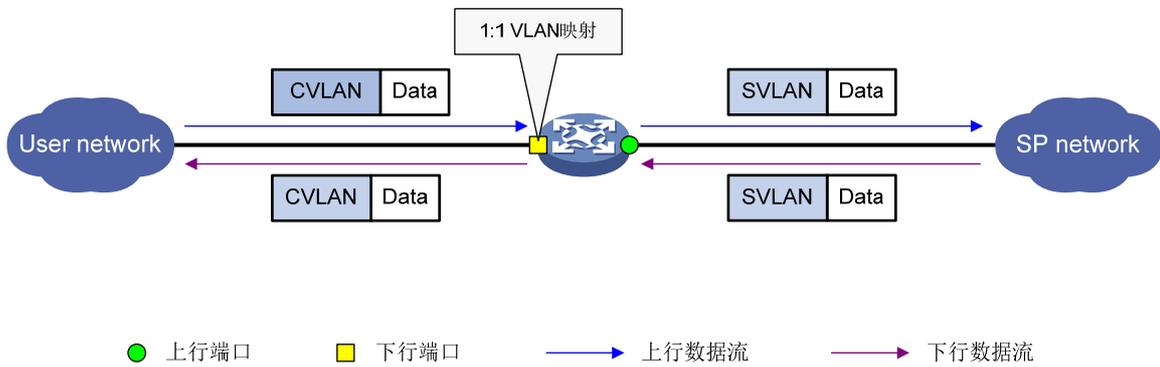
- 上行数据流：从用户网络发往汇聚层网络或 SP 网络的数据流。
- 下行数据流：从汇聚层网络或 SP 网络发往用户网络的数据流。
- 上行端口：发送上行数据流和接收下行数据流的端口。
- 下行端口：发送下行数据流和接收上行数据流的端口。

图1-3 VLAN 映射相关概念示意图



### 1. 1:1 VLAN映射实现方式

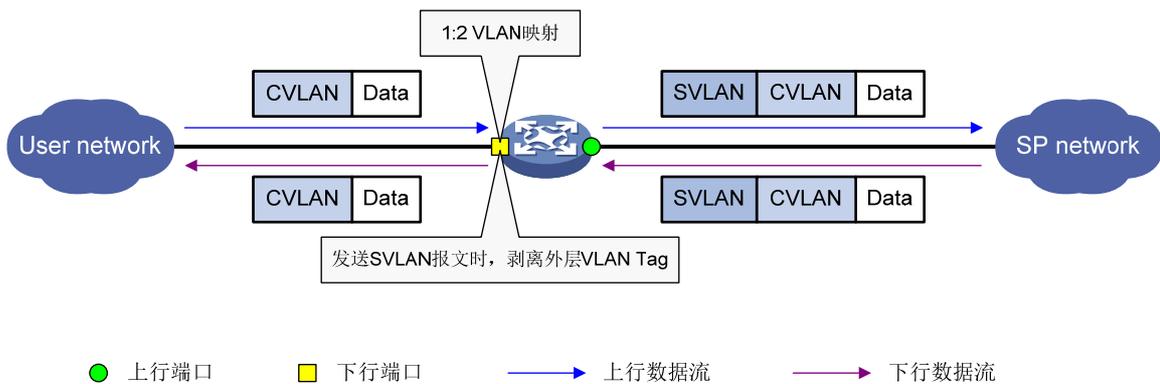
图1-4 1:1 VLAN 映射实现方式示意图



如 图 1-4 所示，通过在下行端口配置 1:1 VLAN 映射，设备将上行数据流的CVLAN替换为SVLAN，将下行数据流的SVLAN替换为CVLAN。

### 2. 1:2 VLAN映射实现方式

图1-5 1:2 VLAN 映射实现方式示意图



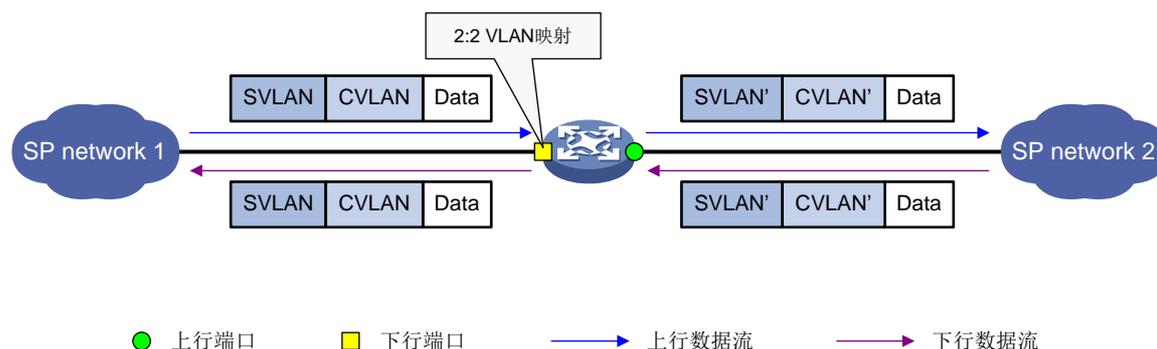
如 图 1-5 所示，通过在下行端口上配置 1:2 VLAN 映射，设备为上行数据流的 CVLAN 报文再添加一层 SVLAN 的 VLAN Tag。

配置 1:2 VLAN 映射时，为保证下行数据流可以顺利到达用户网络，在发送 SVLAN 报文时，需要剥离其外层 VLAN Tag，只保留 CVLAN Tag。可选择如下两种方式的一种来实现剥离外层 VLAN Tag：

- 配置下行端口为 Hybrid 端口，并配置当该端口发送 SVLAN 报文时不带 VLAN Tag。
- 配置下行端口成 Trunk 端口，并将 SVLAN 设为该端口的 PVID。

### 3. 2:2 VLAN 映射实现方式

图1-6 2:2 VLAN 映射实现方式示意图



如 图 1-6 所示，通过在下行端口配置 2:2 VLAN 映射，设备将上行数据流的 SVLAN、CVLAN 转换为 SVLAN'、CVLAN'，将下行数据流的 SVLAN'、CVLAN' 转换为 SVLAN、CVLAN。

## 1.2 VLAN 映射配置任务简介

用户需要根据网络规划，在不同的设备上进行不同的 VLAN 映射配置。

配置 VLAN 映射时，需要注意：

- 若用户同时通过配置 VLAN 映射和 QinQ 来添加报文的 VLAN Tag，且配置冲突时，VLAN 映射的配置生效。有关 QinQ 的详细介绍，请参见“二层技术-以太网交换配置指导”中的“QinQ”。
- 若用户同时通过配置 VLAN 映射和 QoS 策略来修改或添加报文的 VLAN Tag，且配置冲突时，QoS 策略的配置生效。有关 QoS 策略的详细介绍，请参见“ACL 和 QoS 配置指导”中的“QoS 配置方式”。
- 请不要在同一个二层以太网接口/二层聚合接口上同时配置 VLAN 映射、以太网服务实例与 VSI 关联，否则，可能导致这些功能不可用。

表1-1 VLAN 映射配置任务简介

配置任务	说明	详细配置
配置 1:1 VLAN 映射	在图 1-1 所示的组网中，需要在楼道交换机上进行此配置	<a href="#">1.3.1</a>
配置 1:2 VLAN 映射	在图 1-2 所示的组网中，需要在用户进入 SP 网络的边缘设备 PE 1 和 PE 4 上进行此配置	<a href="#">1.3.2</a>
配置 2:2 VLAN 映射	在图 1-2 所示的组网中，需要在 SP 2 网络的边缘设备 PE 3 上进行此配置	<a href="#">1.3.3</a>

## 1.3 配置VLAN映射

### 1.3.1 配置 1:1 VLAN映射

在图 1-1 所示的组网中，需要在楼道交换机上配置 1:1 VLAN 映射，以便将不同用户的不同业务用不同的 VLAN 进行隔离。

在配置 1:1 VLAN 映射前，需要先创建好原始 VLAN 和转换后 VLAN。

1:1 VLAN 映射需要在设备下行端口上进行配置。

表1-2 配置 1:1 VLAN 映射

操作		命令	说明
进入系统视图		<b>system-view</b>	-
进入相应视图	进入二层以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
	进入二层聚合接口视图	<b>interface bridge-aggregation</b> <i>interface-number</i>	-
配置端口链路类型	配置端口的链路类型为Trunk类型	<b>port link-type trunk</b>	二者选其一
	配置端口的链路类型为Hybrid类型	<b>port link-type hybrid</b>	缺省情况下，所有端口的链路类型均为Access类型
配置允许原始VLAN及转换后VLAN通过当前端口		<b>port trunk permit vlan</b> <i>vlan-id-list</i>	二者选其一
		<b>port hybrid vlan</b> <i>vlan-id-list</i> <b>tagged</b>	
配置1:1 VLAN映射		<b>vlan mapping</b> <i>vlan-id</i> <b>translated-vlan</b> <i>vlan-id</i>	缺省情况下，接口上未配置VLAN映射

### 1.3.2 配置 1:2 VLAN映射

在图 1-2 所示的组网中，需要在用户进入 SP 网络的边缘设备 PE 1 和 PE 4 上配置 1:2 VLAN 映射，以便为报文添加 SP 分配给用户的外层 VLAN Tag，使得不同用户的报文在 SP 网络中传输时被完全隔离。

在配置 1:2 VLAN 映射前，需要先创建好原始 VLAN 和转换后 VLAN。

需要注意的是，1:2 VLAN 映射为报文加上外层 VLAN Tag 后，内层 VLAN Tag 将被当作报文的数据部分进行传输，报文长度将增加 4 个字节。因此建议用户适当增加映射后报文传输路径上各接口的 MTU（Maximum Transmission Unit，最大传输单元）值（至少为 1504 字节）。

1:2 VLAN 映射需要在设备下行端口上进行配置。

表1-3 配置 1:2 VLAN 映射

操作		命令	说明
进入系统视图		<b>system-view</b>	-
进入相应视图	进入二层以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-

操作		命令	说明
	进入二层聚合接口视图	<b>interface bridge-aggregation</b> <i>interface-number</i>	
配置端口的链路类型	配置端口的链路类型为Trunk类型	<b>port link-type trunk</b>	二者选其一 缺省情况下，所有端口的链路类型均为Access类型
	配置端口的链路类型为Hybrid类型	<b>port link-type hybrid</b>	
配置允许原始VLAN通过当前端口		<b>port trunk permit vlan</b> <i>vlan-id-list</i>	二者选其一
		<b>port hybrid vlan</b> <i>vlan-id-list</i> { <i>tagged</i>   <i>untagged</i> }	
配置允许转换后外层VLAN不带Tag通过当前端口	配置Trunk端口的PVID为添加的外层VLAN并允许该VLAN通过	<b>port trunk pvid vlan</b> <i>vlan-id</i> <b>port trunk permit vlan</b> { <i>vlan-id-list</i>   <i>all</i> }	二者选其一
	配置允许添加的外层VLAN以Untagged方式通过Hybrid端口	<b>port hybrid vlan</b> <i>vlan-id-list</i> <b>untagged</b>	
配置1:2 VLAN映射		<b>vlan mapping nest</b> { <b>range</b> <i>vlan-range-list</i>   <b>single</b> <i>vlan-id-list</i> } <b>nested-vlan</b> <i>vlan-id</i>	缺省情况下，接口上未配置VLAN映射 一个原始VLAN只能映射一个转换后外层VLAN。若要为不同原始VLAN的报文添加不同的外层VLAN Tag，需将端口的链路类型配置为Hybrid类型并多次执行本命令

### 1.3.3 配置 2:2 VLAN映射

在图 1-2 所示的组网中，需要在 SP 2 网络的边缘设备 PE 3 上配置 2:2 VLAN 映射，将报文外层 VLAN Tag 替换为新 SP 网络分配给同一 VPN 用户的 VLAN Tag，同时替换内层 VLAN Tag，使得该 VPN 内原本不同 VLAN 的用户可以互通。

在配置 2:2 VLAN 映射前，需要先创建好原始 VLAN 和转换后 VLAN。

2:2 VLAN 映射需要在设备下行端口上进行配置。

表1-4 配置 2:2 VLAN 映射

操作		命令	说明
进入系统视图		<b>system-view</b>	-
进入相应视图	进入二层以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
	进入二层聚合接口视图	<b>interface bridge-aggregation</b> <i>interface-number</i>	
配置端口链路类型	配置端口的链路类型为Trunk类型	<b>port link-type trunk</b>	二者选其一 缺省情况下，所有端口的链路类

操作		命令	说明
	配置端口的链路类型为Hybrid类型	<b>port link-type hybrid</b>	型均为Access类型
配置允许原始外层VLAN及转换后外层VLAN通过当前端口		<b>port trunk permit vlan <i>vlan-id-list</i></b>	二者选其一
		<b>port hybrid vlan <i>vlan-id-list</i> tagged</b>	
配置2:2 VLAN映射		<b>vlan mapping tunnel <i>outer-vlan-id</i> <i>inner-vlan-id</i> translated-vlan <i>outer-vlan-id</i> <i>inner-vlan-id</i></b>	缺省情况下，接口上未配置VLAN映射

## 1.4 VLAN映射显示和维护

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 VLAN 映射的运行情况，通过查看显示信息验证配置的效果。

表1-5 VLAN 映射显示和维护

操作	命令
显示VLAN映射信息	<b>display vlan mapping [ interface <i>interface-type</i> <i>interface-number</i> ]</b>

## 1.5 VLAN映射典型配置举例

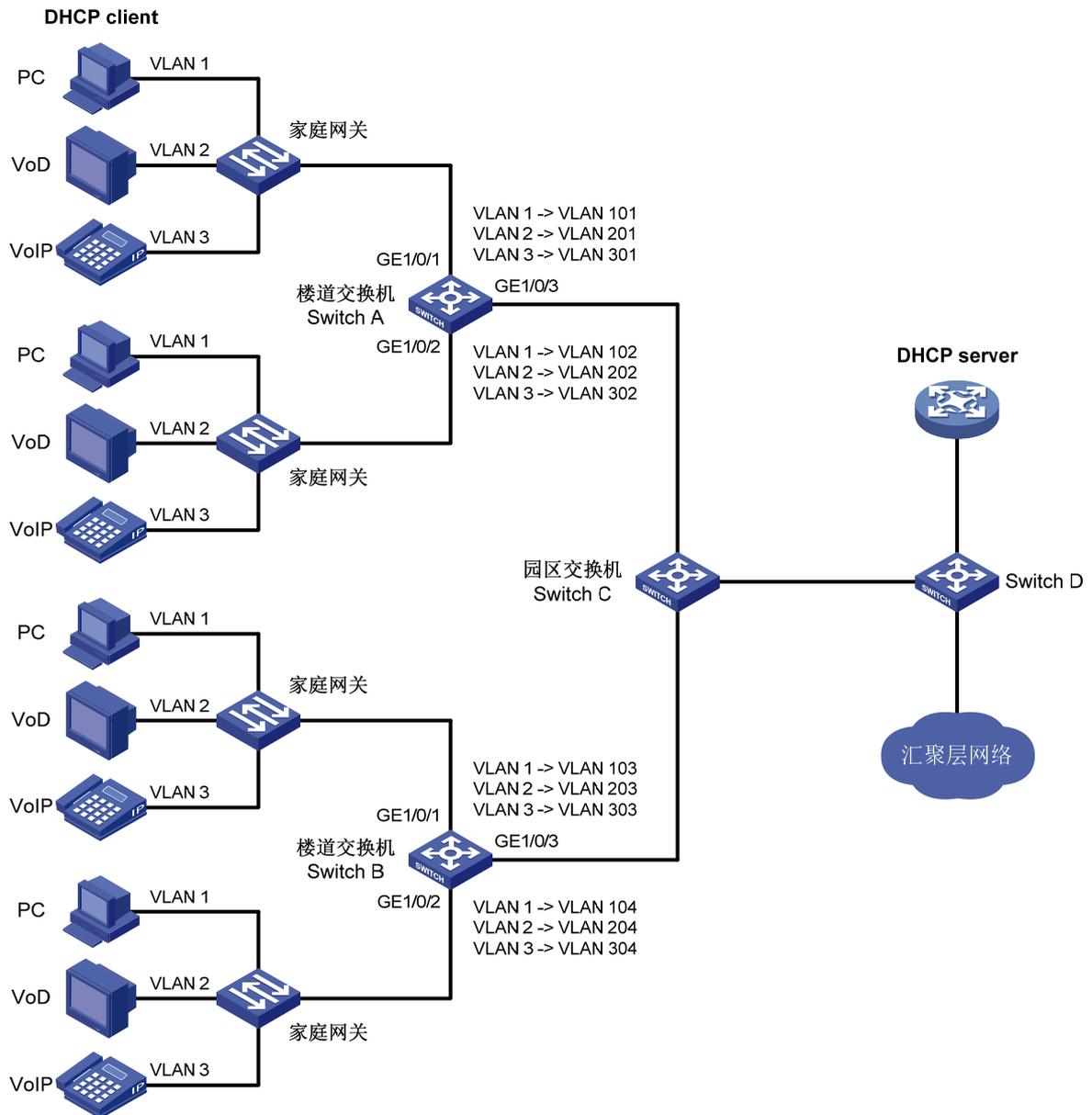
### 1.5.1 1:1 VLAN映射配置举例

#### 1. 组网需求

- 在某小区，服务提供商为每个家庭都提供了 PC、VoD 和 VoIP 这三种数据服务，每个家庭都通过各自的家庭网关接入楼道交换机，并通过 DHCP 方式自动获取 IP 地址。
- 服务提供商希望实现以下网络规划：在家庭网关上，分别将 PC、VoD 和 VoIP 业务依次划分到 VLAN 1~3；在楼道交换机上，为了隔离不同家庭的同类业务，将每个家庭的每种业务都划分到不同的 VLAN。

## 2. 组网图

图1-7 1:1 VLAN 映射配置组网图



## 3. 配置步骤

### (1) 配置 Switch A

# 配置下行端口 GigabitEthernet1/0/1 和 GigabitEthernet1/0/2 为 Trunk 端口且允许原始 VLAN 及转换后 VLAN 通过，同时在端口上配置 1:1 VLAN 映射。

```
<SwitchA> system-view
[SwitchA] vlan 2 to 3
[SwitchA] vlan 101 to 102
[SwitchA] vlan 201 to 202
[SwitchA] vlan 301 to 302
[SwitchA] interface gigabitethernet 1/0/1
```

```

[SwitchA-GigabitEthernet1/0/1] port link-type trunk
[SwitchA-GigabitEthernet1/0/1] port trunk permit vlan 1 2 3 101 201 301
[SwitchA-GigabitEthernet1/0/1] vlan mapping 1 translated-vlan 101
[SwitchA-GigabitEthernet1/0/1] vlan mapping 2 translated-vlan 201
[SwitchA-GigabitEthernet1/0/1] vlan mapping 3 translated-vlan 301
[SwitchA-GigabitEthernet1/0/1] quit
[SwitchA] interface gigabitethernet 1/0/2
[SwitchA-GigabitEthernet1/0/2] port link-type trunk
[SwitchA-GigabitEthernet1/0/2] port trunk permit vlan 1 2 3 102 202 302
[SwitchA-GigabitEthernet1/0/2] vlan mapping 1 translated-vlan 102
[SwitchA-GigabitEthernet1/0/2] vlan mapping 2 translated-vlan 202
[SwitchA-GigabitEthernet1/0/2] vlan mapping 3 translated-vlan 302
[SwitchA-GigabitEthernet1/0/2] quit
# 配置上行端口 GigabitEthernet1/0/3 为 Trunk 端口，且允许转换后 VLAN 通过。
[SwitchA] interface gigabitethernet 1/0/3
[SwitchA-GigabitEthernet1/0/3] port link-type trunk
[SwitchA-GigabitEthernet1/0/3] port trunk permit vlan 101 201 301 102 202 302
[SwitchA-GigabitEthernet1/0/3] quit

```

## (2) 配置 Switch B

Switch B 的配置与 Switch A 相似，配置过程略。

## 4. 验证配置

### (1) 查看 Switch A 上的 VLAN 映射配置信息

```

[SwitchA] display vlan mapping
Interface GigabitEthernet1/0/1:

```

Outer VLAN	Inner VLAN	Translated Outer VLAN	Translated Inner VLAN
1	N/A	101	N/A
2	N/A	201	N/A
3	N/A	301	N/A

```

Interface GigabitEthernet1/0/2:

```

Outer VLAN	Inner VLAN	Translated Outer VLAN	Translated Inner VLAN
1	N/A	102	N/A
2	N/A	202	N/A
3	N/A	302	N/A

### (2) 查看 Switch B 上的 VLAN 映射配置信息

Switch B 上的 VLAN 映射配置信息与 Switch A 相似，显示信息略。

以上信息表明，Switch A 和 Switch B 上的 1:1 VLAN 映射配置成功。

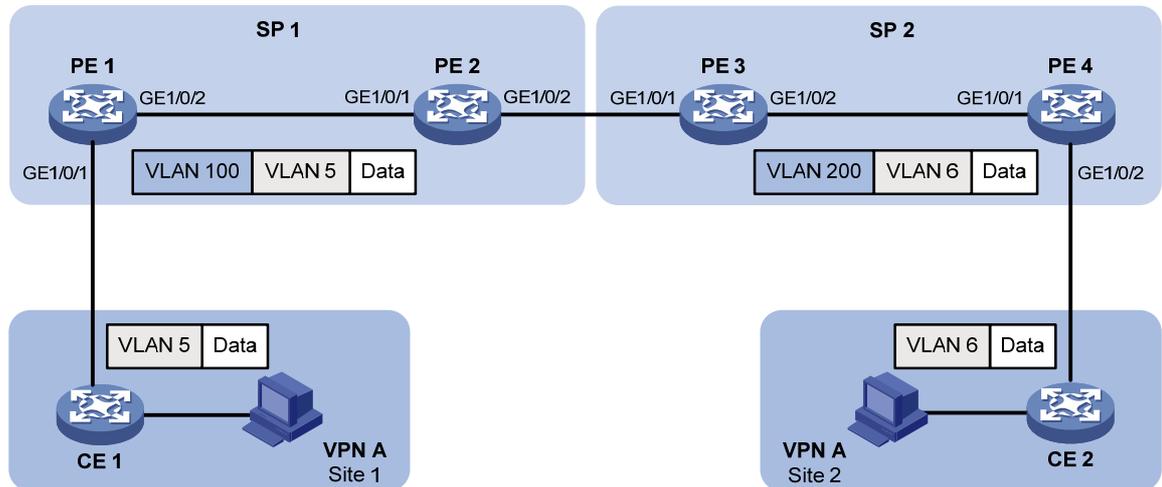
## 1.5.2 1:2 和 2:2 VLAN 映射配置举例

### 1. 组网需求

- VPN A 中的 Site 1 和 Site 2 是某公司的两个分支机构，且分别利用 VLAN 5 和 VLAN 6 承载业务。由于分处不同地域，这两个分支机构采用了不同的 SP 所提供的 VPN 接入服务，SP 1 和 SP 2 分别将 VLAN 100 和 VLAN 200 分配给这两个分支机构使用。
- 该公司希望其下属的这两个分支机构可以跨越 SP 1 和 SP 2 的网络实现互通。

## 2. 组网图

图1-8 1:2 和 2:2 VLAN 映射配置组网图



## 3. 配置步骤

### (1) 配置 PE 1

# 创建 VLAN 5 和 VLAN 100。

```
<PE1> system-view
[PE1] vlan 5
[PE1-vlan5] quit
[PE1] vlan 100
[PE1-vlan100] quit
```

# 在下行端口 GigabitEthernet1/0/1 上配置 1:2 VLAN 映射，为 VLAN 5 的报文添加 VLAN 100 的外层 VLAN Tag。

```
[PE1] interface gigabitEthernet 1/0/1
[PE1-GigabitEthernet1/0/1] vlan mapping nest single 5 nested-vlan 100
```

# 配置 GigabitEthernet1/0/1 为 Hybrid 端口且允许 VLAN 5 的报文携带 VLAN Tag 通过、VLAN 100 的报文不携带 VLAN Tag 通过。

```
[PE1-GigabitEthernet1/0/1] port link-type hybrid
[PE1-GigabitEthernet1/0/1] port hybrid vlan 5 tagged
[PE1-GigabitEthernet1/0/1] port hybrid vlan 100 untagged
[PE1-GigabitEthernet1/0/1] quit
```

# 配置上行端口 GigabitEthernet1/0/2 为 Trunk 端口且允许 VLAN 100 通过。

```
[PE1] interface gigabitEthernet 1/0/2
[PE1-GigabitEthernet1/0/2] port link-type trunk
[PE1-GigabitEthernet1/0/2] port trunk permit vlan 100
[PE1-GigabitEthernet1/0/2] quit
```

### (2) 配置 PE 2

# 创建 VLAN 100。

```
<PE2> system-view
[PE2] vlan 100
[PE2-vlan100] quit
```

# 配置端口 GigabitEthernet1/0/1 为 Trunk 端口且允许 VLAN 100 通过。

```
[PE2] interface gigabitethernet 1/0/1
[PE2-GigabitEthernet1/0/1] port link-type trunk
[PE2-GigabitEthernet1/0/1] port trunk permit vlan 100
[PE2-GigabitEthernet1/0/1] quit
```

# 配置端口 GigabitEthernet1/0/2 为 Trunk 端口且允许 VLAN 100 通过。

```
[PE2] interface gigabitethernet 1/0/2
[PE2-GigabitEthernet1/0/2] port link-type trunk
[PE2-GigabitEthernet1/0/2] port trunk permit vlan 100
[PE2-GigabitEthernet1/0/2] quit
```

### (3) 配置 PE 3

# 创建 VLAN 5、VLAN 6、VLAN 100 和 VLAN 200。

```
<PE3> system-view
[PE3] vlan 5 to 6
[PE3] vlan 100
[PE3-vlan100] quit
[PE3] vlan 200
[PE3-vlan200] quit
```

# 配置端口 GigabitEthernet1/0/1 为 Trunk 端口且允许 VLAN 100 和 200 通过。

```
[PE3] interface gigabitethernet 1/0/1
[PE3-GigabitEthernet1/0/1] port link-type trunk
[PE3-GigabitEthernet1/0/1] port trunk permit vlan 100 200
```

# 在端口 GigabitEthernet1/0/1 上配置 2:2 VLAN 映射，将外层 VLAN 为 100、内层 VLAN 为 5 的报文的 VLAN ID 转换为外层 VLAN 为 200、内层 VLAN 为 6。

```
[PE3-GigabitEthernet1/0/1] vlan mapping tunnel 100 5 translated-vlan 200 6
[PE3-GigabitEthernet1/0/1] quit
```

# 配置端口 GigabitEthernet1/0/2 为 Trunk 端口且允许 VLAN 200 通过。

```
[PE3] interface gigabitethernet 1/0/2
[PE3-GigabitEthernet1/0/2] port link-type trunk
[PE3-GigabitEthernet1/0/2] port trunk permit vlan 200
[PE3-GigabitEthernet1/0/2] quit
```

### (4) 配置 PE 4

# 创建 VLAN 6 和 VLAN 200。

```
<PE4> system-view
[PE4] vlan 6
[PE4-vlan6] quit
[PE4] vlan 200
[PE4-vlan200] quit
```

# 配置上行端口 GigabitEthernet1/0/1 为 Trunk 端口且允许 VLAN 200 通过。

```
[PE4] interface gigabitethernet 1/0/1
[PE4-GigabitEthernet1/0/1] port link-type trunk
[PE4-GigabitEthernet1/0/1] port trunk permit vlan 200
[PE4-GigabitEthernet1/0/1] quit
```

# 配置下行端口 GigabitEthernet1/0/2 为 Hybrid 端口且允许 VLAN 6 的报文携带 VLAN Tag 通过，允许 VLAN 200 的报文不携带 VLAN Tag 通过。

```
[PE4] interface gigabitethernet 1/0/2
[PE4-GigabitEthernet1/0/2] port link-type hybrid
[PE4-GigabitEthernet1/0/2] port hybrid vlan 6 tagged
[PE4-GigabitEthernet1/0/2] port hybrid vlan 200 untagged
```

# 在端口 GigabitEthernet1/0/2 上配置 1:2 VLAN 映射，为 VLAN 6 的报文添加 VLAN 200 的外层 VLAN Tag。

```
[PE4-GigabitEthernet1/0/2] vlan mapping nest single 6 nested-vlan 200
[PE4-GigabitEthernet1/0/2] quit
```

#### 4. 验证配置

##### (1) 查看 PE 1 上的 VLAN 映射配置信息

```
[PE1] display vlan mapping
```

```
Interface GigabitEthernet1/0/1:
```

Outer VLAN	Inner VLAN	Translated Outer VLAN	Translated Inner VLAN
5	N/A	100	5

##### (2) 查看 PE 3 上的 VLAN 映射配置信息

```
[PE3] display vlan mapping
```

```
Interface GigabitEthernet1/0/1:
```

Outer VLAN	Inner VLAN	Translated Outer VLAN	Translated Inner VLAN
100	5	200	6

##### (3) 查看 PE 4 上的 VLAN 映射配置信息

```
[PE4] display vlan mapping
```

```
Interface GigabitEthernet1/0/2:
```

Outer VLAN	Inner VLAN	Translated Outer VLAN	Translated Inner VLAN
6	N/A	200	6

以上信息表明，PE 1 和 PE 4 上的 1:2 VLAN 映射，以及 PE 3 上的 2:2 VLAN 映射配置成功。

# 目 录

1 环路检测.....	1-1
1.1 环路检测简介.....	1-1
1.1.1 环路检测产生背景.....	1-1
1.1.2 环路检测报文.....	1-1
1.1.3 环路检测运行机制.....	1-2
1.2 环路检测配置任务简介.....	1-3
1.3 配置环路检测.....	1-3
1.3.1 开启环路检测功能.....	1-3
1.3.2 配置环路检测处理模式.....	1-4
1.3.3 配置环路检测时间间隔.....	1-5
1.4 环路检测显示和维护.....	1-5
1.5 环路检测典型配置举例.....	1-5

# 1 环路检测

## 1.1 环路检测简介

### 1.1.1 环路检测产生背景

网络连接错误或配置错误都容易导致二层网络中出现转发环路，使设备对广播、组播以及未知单播报文进行重复发送，造成网络资源的浪费甚至导致网络瘫痪。为了能够及时发现二层网络中的环路，以避免对整个网络造成严重影响，需要提供一种检测机制，使网络中出现环路时能及时通知用户检查网络连接和配置情况，这种机制就是环路检测机制。当网络中出现环路时，环路检测机制通过生成日志信息（请参见“网络管理和监控配置指导”中的“信息中心”）来通知用户，并可根据用户事先的配置来选择是否关闭出现环路的端口。

### 1.1.2 环路检测报文

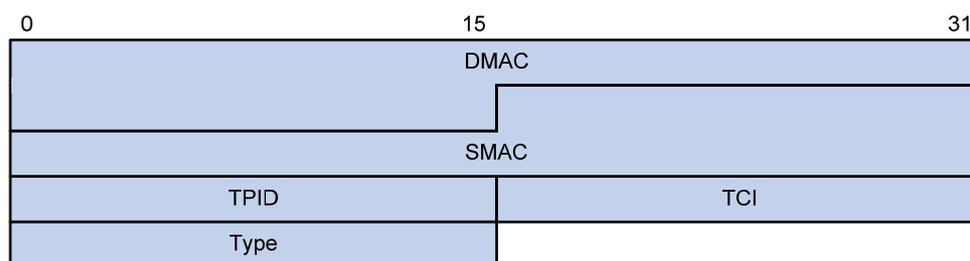
设备通过发送环路检测报文并检测其是否返回本设备（不要求收、发端口为同一端口）以确认是否存在环路，若某端口收到了由本设备发出的环路检测报文，就认定该端口所在链路存在环路。



提示

环路检测通常工作在特定的 VLAN 内，但也可能因 QinQ 或 VLAN 映射等特性的配置错误而导致 VLAN 间的环路（即尽管发出和收到的报文所携带的 VLAN 信息不同，但仍认为存在环路）。有关 QinQ 和 VLAN 映射的详细介绍，请分别参见“二层技术-以太网交换配置指导”中的“QinQ”和“VLAN 映射”。

图1-1 环路检测报文以太网头的封装格式



如 [图 1-1](#) 所示，为环路检测报文以太网头的封装格式，其中各字段的解释如下：

- **DMAC**：报文的目的 MAC 地址，使用组播 MAC 地址 010f-e200-0007。当设备开启了环路检测功能时，会将该目的地址的报文上送 CPU 处理，并在收到该报文的 VLAN 内将原始报文广播一份。
- **SMAC**：报文的源 MAC 地址，采用发送该报文的设备的桥 MAC。
- **TPID**：VLAN 标签的类型，取值为 0x8100。

- TCI: VLAN 标签的具体值，具体内容为优先级、VLAN ID 等。
- Type: 协议类型，取值为 0x8918。

图1-2 环路检测报文内部头的封装格式

0	15	31
Code	Version	
Length	Reserved	

如 图 1-2 所示，为环路检测报文的内部头的封装格式，其中各字段的解释如下：

- Code: 协议子类型，取值为 0x0001，表示环路检测协议。
- Version: 版本，取值为 0x0000，目前保留。
- Length: 报文长度（包括环路检测报文的头部，但不包括以太网头部）。
- Reserved: 保留字段。

环路检测报文的内容以TLV（Type/Length/Value，类型/长度/值）格式进行封装，环路检测支持的 TLV 类型如 表 1-1 所示。

表1-1 环路检测支持的 TLV 类型

TLV 名称	说明	携带要求
End of PDU	结束TLV，用来标志PDU结束	可选
Device ID	设备标识TLV，表示发送设备的桥MAC地址	必须
Port ID	端口标识TLV，用来标识PDU发送端的端口索引	可选
Port Name	端口名称TLV，用来标识PDU发送端的端口名称	可选
System Name	系统名称TLV，表示设备的名称	可选
Chassis ID	框号TLV，表示发送端口所在的框号	可选
Slot ID	槽位号TLV，表示发送端口所在的槽位号	可选
Sub Slot ID	子槽位号TLV，表示发送端口所在的子槽位号	可选

### 1.1.3 环路检测运行机制

#### 1. 环路检测时间间隔

由于网络时刻处于变化中，因此环路检测是一个持续的过程，它以一定的时间间隔发送环路检测报文来确定各端口是否出现环路、以及存在环路的端口上是否已消除环路等，这个时间间隔就称为环路检测的时间间隔。

#### 2. 环路检测处理模式

环路检测的处理模式是指当系统检测到端口出现环路时的处理方式，包括以下几种：

- Block 模式：当系统检测到端口出现环路时，除了生成日志信息外，还会禁止端口学习 MAC 地址并将端口阻塞。

- **No-learning** 模式：当系统检测到端口出现环路时，除了生成日志信息外，还会禁止端口学习 MAC 地址。
- **Shutdown** 模式：当系统检测到端口出现环路时，除了生成日志信息外，还会自动关闭该端口，使其不能收发任何报文。被关闭的端口将在 **shutdown-interval** 命令（请参考“基础配置命令参考”中的“设备管理”）所配置的时间之后自动恢复。

缺省情况下，系统不采用上述任何一种模式，当系统检测到端口出现环路时，除了生成日志信息外不对该端口进行任何处理。

### 3. 端口状态自动恢复

在 **Block** 模式和 **No-learning** 模式下，当设备检测到某端口出现环路后，若在三倍的环路检测时间间隔内仍未收到环路检测报文，就认为该端口上的环路已消除，自动将该端口恢复为正常转发状态，并通知给用户。这个过程就是端口状态的自动恢复过程。

在 **Shutdown** 模式下，出现环路的端口先被自动关闭，然后在 **shutdown-interval** 命令所配置的时间之后自动恢复。如果此时环路尚未消除，该端口将被再次关闭，然后恢复……如此往复直至环路消除。



提示

当网络中存在环路时，为防止大量报文的冲击，设备会丢弃部分报文。而如果环路检测报文也被丢弃，设备在端口状态自动恢复功能的作用下会误判定环路已消除。在这种情况下，建议将环路检测的处理模式配置为 **Shutdown** 模式，或当设备提示出现环路时通过手工排查来消除环路。

## 1.2 环路检测配置任务简介

表1-2 环路检测配置任务简介

配置任务	说明	详细配置
开启环路检测功能	必选	<a href="#">1.3.1</a>
配置环路检测处理模式	可选	<a href="#">1.3.2</a>
配置环路检测时间间隔	可选	<a href="#">1.3.3</a>

## 1.3 配置环路检测

### 1.3.1 开启环路检测功能

设备全局或者端口开启环路检测功能，当设备上任一端口收到设备发送的任一 VLAN 的环路检测报文时，会触发该端口的环路保护动作。

## 1. 全局开启环路检测功能

表1-3 全局开启环路检测功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
全局开启环路检测功能	<b>loopback-detection global enable vlan { vlan-id-list   all }</b>	缺省情况下，环路检测功能处于全局关闭状态

## 2. 在端口上开启环路检测功能

表1-4 在端口上开启环路检测功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二层以太网/二层聚合接口视图	<b>interface interface-type interface-number</b>	-
在端口上开启环路检测功能	<b>loopback-detection enable vlan { vlan-id-list   all }</b>	缺省情况下，端口上的环路检测功能处于关闭状态

### 1.3.2 配置环路检测处理模式

用户可以在系统视图下全局配置环路检测的处理模式，也可以在接口视图下配置当前端口的环路检测处理模式。系统视图下的配置对所有端口都有效，接口视图下的配置则只对当前端口有效，且接口视图下的配置优先级较高。

#### 1. 全局配置环路检测处理模式

表1-5 全局配置环路检测处理模式

操作	命令	说明
进入系统视图	<b>system-view</b>	-
全局配置环路检测的处理模式	<b>loopback-detection global action shutdown</b>	缺省情况下，当系统检测到端口出现环路时不对该端口进行任何处理，仅生成日志信息

#### 2. 在二层以太网接口上配置环路检测处理模式

表1-6 在二层以太网接口上配置环路检测处理模式

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二层以太网接口视图	<b>interface interface-type interface-number</b>	-
在端口上配置环路检测的处理模式	<b>loopback-detection action { block   no-learning   shutdown }</b>	缺省情况下，当系统检测到端口出现环路时不对该端口进行任何处理，仅生成日志信息

### 3. 在二层聚合接口上配置环路检测处理模式

表1-7 在二层聚合接口上配置环路检测处理模式

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二层聚合接口视图	<b>interface interface-type interface-number</b>	-
在端口上配置环路检测的处理模式	<b>loopback-detection action shutdown</b>	缺省情况下，当系统检测到端口出现环路时不对该端口进行任何处理，仅生成日志信息

#### 1.3.3 配置环路检测时间间隔

当开启了环路检测功能后，系统开始以一定的时间间隔发送环路检测报文，该间隔越长耗费的系统性能越少，该间隔越短环路检测的灵敏度越高。用户可以通过本配置调整发送环路检测报文的时间间隔，以在系统性能和环路检测的灵敏度之间进行平衡。

表1-8 配置环路检测时间间隔

操作	命令	说明
进入系统视图	<b>system-view</b>	-
配置环路检测的时间间隔	<b>loopback-detection interval-time interval</b>	缺省情况下，环路检测的时间间隔为30秒

#### 1.4 环路检测显示和维护

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环路检测的运行情况，通过查看显示信息验证配置的效果。

表1-9 环路检测显示和维护

操作	命令
显示环路检测的配置和运行情况	<b>display loopback-detection</b>

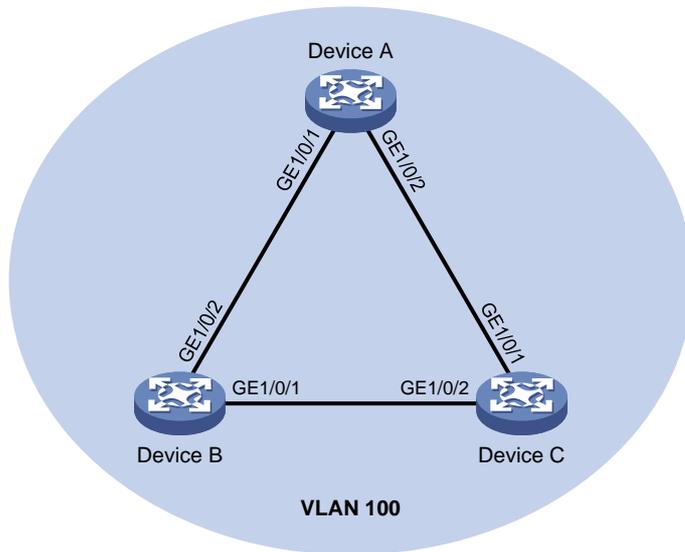
#### 1.5 环路检测典型配置举例

##### 1. 组网需求

- 三台设备 Device A、Device B 和 Device C 组成一个物理上的环形网络。
- 通过在 Device A 上配置环路检测功能，使系统能够自动关闭 Device A 上出现环路的端口，并通过打印日志信息来通知用户检查网络。

## 2. 组网图

图1-3 环路检测典型组网图



## 3. 配置步骤

### (1) 配置 Device A

# 创建 VLAN 100，并全局开启该 VLAN 内的环路检测功能。

```
<DeviceA> system-view
[DeviceA] vlan 100
[DeviceA-vlan100] quit
[DeviceA] loopback-detection global enable vlan 100
```

# 配置端口 GigabitEthernet1/0/1 和 GigabitEthernet1/0/2 为 Trunk 类型，并允许 VLAN 100 通过。

```
[DeviceA] interface gigabitethernet 1/0/1
[DeviceA-GigabitEthernet1/0/1] port link-type trunk
[DeviceA-GigabitEthernet1/0/1] port trunk permit vlan 100
[DeviceA-GigabitEthernet1/0/1] quit
[DeviceA] interface gigabitethernet 1/0/2
[DeviceA-GigabitEthernet1/0/2] port link-type trunk
[DeviceA-GigabitEthernet1/0/2] port trunk permit vlan 100
[DeviceA-GigabitEthernet1/0/2] quit
```

# 全局配置环路检测的处理模式为 Shutdown 模式。

```
[DeviceA] loopback-detection global action shutdown
```

# 配置环路检测的时间间隔为 35 秒。

```
[DeviceA] loopback-detection interval-time 35
```

### (2) 配置 Device B

# 创建 VLAN 100。

```
<DeviceB> system-view
[DeviceB] vlan 100
[DeviceB-vlan100] quit
```

# 配置端口 GigabitEthernet1/0/1 和 GigabitEthernet1/0/2 为 Trunk 类型，并允许 VLAN 100 通过。

```
[DeviceB] interface gigabitethernet 1/0/1
[DeviceB-GigabitEthernet1/0/1] port link-type trunk
[DeviceB-GigabitEthernet1/0/1] port trunk permit vlan 100
[DeviceB-GigabitEthernet1/0/1] quit
[DeviceB] interface gigabitethernet 1/0/2
[DeviceB-GigabitEthernet1/0/2] port link-type trunk
[DeviceB-GigabitEthernet1/0/2] port trunk permit vlan 100
[DeviceB-GigabitEthernet1/0/2] quit
```

### (3) 配置 Device C

# 创建 VLAN 100。

```
<DeviceC> system-view
[DeviceC] vlan 100
[DeviceC-vlan100] quit
```

# 配置端口 GigabitEthernet1/0/1 和 GigabitEthernet1/0/2 为 Trunk 类型，并允许 VLAN 100 通过。

```
[DeviceC] interface gigabitethernet 1/0/1
[DeviceC-GigabitEthernet1/0/1] port link-type trunk
[DeviceC-GigabitEthernet1/0/1] port trunk permit vlan 100
[DeviceC-GigabitEthernet1/0/1] quit
[DeviceC] interface gigabitethernet 1/0/2
[DeviceC-GigabitEthernet1/0/2] port link-type trunk
[DeviceC-GigabitEthernet1/0/2] port trunk permit vlan 100
[DeviceC-GigabitEthernet1/0/2] quit
```

## 4. 验证配置

当配置完成后，系统在一个环路检测时间间隔内在 Device A 的端口 GigabitEthernet1/0/1 和 GigabitEthernet1/0/2 上都检测到了环路，于是将这两个端口自动关闭，并打印了如下日志信息：

```
[DeviceA]
%Feb 24 15:04:29:663 2013 DeviceA LPDT/4/LPDT_LOOPED: Loopback exists on
GigabitEthernet1/0/1.
%Feb 24 15:04:29:667 2013 DeviceA LPDT/4/LPDT_LOOPED: Loopback exists on
GigabitEthernet1/0/2.
%Feb 24 15:04:44:243 2013 DeviceA LPDT/5/LPDT_RECOVERED: Loopback on GigabitEthernet1/0/1
recovered.
%Feb 24 15:04:44:248 2013 DeviceA LPDT/5/LPDT_RECOVERED: Loopback on GigabitEthernet1/0/2
recovered.
```

使用 **display loopback-detection** 命令可以查看 Device A 上环路检测的配置和运行情况：

# 显示 Device A 上环路检测的配置和运行情况。

```
[DeviceA] display loopback-detection
Loopback detection is enabled.
Loopback detection interval is 35 second(s).
No loopback is detected.
```

由此可见，Device A 上并未显示在端口 GigabitEthernet1/0/1 和 GigabitEthernet1/0/2 上检测到环路，这是由于环路检测功能运行在 Shutdown 模式下，端口 GigabitEthernet1/0/1 和 GigabitEthernet1/0/2 上出现环路后均已被自动关闭，因此这两个端口上的环路已消除。此时，使用 **display interface** 命令分别查看 Device A 上端口 GigabitEthernet1/0/1 和 GigabitEthernet1/0/2 的状态信息：

# 显示 Device A 上端口 GigabitEthernet1/0/1 的状态信息。

```
[DeviceA] display interface gigabitethernet 1/0/1  
GigabitEthernet1/0/1 current state: DOWN (Loopback detection down)  
...
```

# 显示 Device A 上端口 GigabitEthernet1/0/2 的状态信息。

```
[DeviceA] display interface gigabitethernet 1/0/2  
GigabitEthernet1/0/2 current state: DOWN (Loopback detection down)  
...
```

由此可见，端口 GigabitEthernet1/0/1 和 GigabitEthernet1/0/2 均已被环路检测模块自动关闭。

# 目 录

<b>1 生成树</b> .....	<b>1-1</b>
1.1 生成树简介.....	1-1
1.1.1 STP简介.....	1-1
1.1.2 RSTP简介.....	1-9
1.1.3 PVST简介.....	1-11
1.1.4 MSTP简介.....	1-12
1.1.5 快速收敛机制.....	1-18
1.1.6 协议规范.....	1-21
1.2 生成树配置任务简介.....	1-21
1.2.1 STP配置任务简介.....	1-22
1.2.2 RSTP配置任务简介.....	1-23
1.2.3 PVST配置任务简介.....	1-24
1.2.4 MSTP配置任务简介.....	1-25
1.3 配置生成树.....	1-26
1.3.1 配置生成树的工作模式.....	1-26
1.3.2 配置MST域.....	1-27
1.3.3 配置根桥和备份根桥.....	1-27
1.3.4 配置设备的优先级.....	1-28
1.3.5 配置MST域的最大跳数.....	1-29
1.3.6 配置交换网络的网络直径.....	1-29
1.3.7 配置生成树的时间参数.....	1-30
1.3.8 配置超时时间因子.....	1-31
1.3.9 配置端口发送BPDU的速率.....	1-31
1.3.10 配置端口为边缘端口.....	1-32
1.3.11 配置端口的路径开销.....	1-32
1.3.12 配置端口的优先级.....	1-35
1.3.13 配置端口的链路类型.....	1-36
1.3.14 配置端口收发的MSTP报文格式.....	1-36
1.3.15 打开端口状态变化信息显示开关.....	1-37
1.3.16 开启生成树协议.....	1-37
1.3.17 执行mCheck操作.....	1-38
1.3.18 关闭PVST的PVID不一致保护功能.....	1-39
1.3.19 配置摘要侦听功能.....	1-39

1.3.20 配置No Agreement Check功能	1-41
1.3.21 配置TC Snooping功能	1-43
1.3.22 配置生成树保护功能	1-44
1.3.23 配置在PVST模式下设备检测或接收到TC报文时打印日志信息	1-49
1.3.24 配置被BPDU保护功能关闭的端口不再自动恢复	1-49
1.3.25 配置生成树的网管功能	1-50
1.4 生成树显示和维护	1-50
1.5 生成树典型配置举例	1-51
1.5.1 MSTP典型配置举例	1-51
1.5.2 PVST典型配置举例	1-54
1.5.3 生成树支持分布式聚合典型配置举例	1-57

# 1 生成树

## 1.1 生成树简介

生成树协议是一种二层管理协议，它通过选择性地阻塞网络中的冗余链路来消除二层环路，同时还具备链路备份的功能。

与众多协议的发展过程一样，生成树协议也是随着网络的发展而不断更新的，从最初的 STP（Spanning Tree Protocol，生成树协议）到 RSTP（Rapid Spanning Tree Protocol，快速生成树协议）和 PVST（Per-VLAN Spanning Tree，每 VLAN 生成树），再到最新的 MSTP（Multiple Spanning Tree Protocol，多生成树协议）。本文将对 STP、RSTP、PVST 和 MSTP 各自的特点及其相互间的关系进行介绍。

### 1.1.1 STP简介

STP 由 IEEE 制定的 802.1D 标准定义，用于在局域网中消除数据链路层物理环路的协议。运行该协议的设备通过彼此交互信息发现网络中的环路，并有选择的对某些端口进行阻塞，最终将环路网络结构修剪成无环路的树型网络结构，从而防止报文在环路网络中不断增生和无限循环，避免设备由于重复接收相同的报文造成的报文处理能力下降的问题发生。

STP 包含了两个含义，狭义的 STP 是指 IEEE 802.1D 中定义的 STP 协议，广义的 STP 是指包括 IEEE 802.1D 定义的 STP 协议以及各种在它的基础上经过改进的生成树协议。

#### 1. STP的协议报文

STP 采用的协议报文是 BPDU（Bridge Protocol Data Unit，网桥协议数据单元），也称为配置消息。本文中把生成树协议的协议报文均简称为 BPDU。

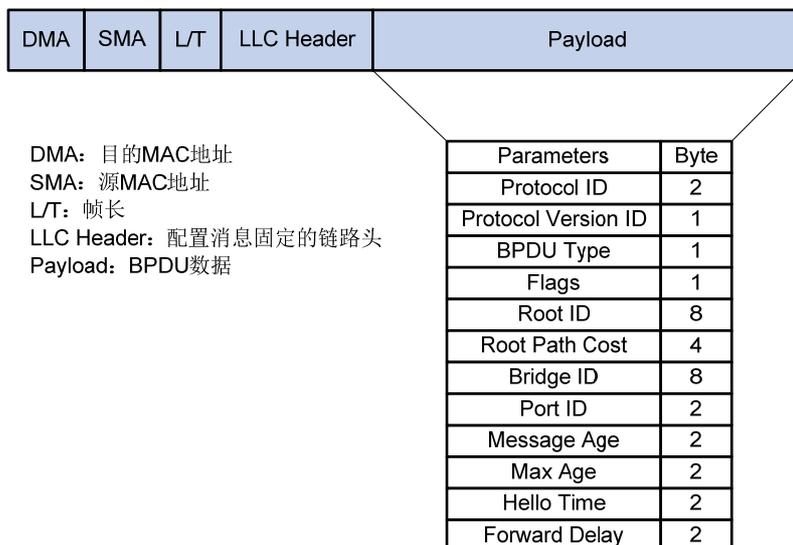
STP 通过在设备之间传递 BPDU 来确定网络的拓扑结构。BPDU 中包含了足够的信息来保证设备完成生成树的计算过程。STP 协议的 BPDU 分为以下两类：

- 配置 BPDU（Configuration BPDU）：用来进行生成树计算和维护生成树拓扑的报文。
- TCN BPDU（Topology Change Notification BPDU，拓扑变化通知 BPDU）：当拓扑结构发生变化时，用来通知相关设备网络拓扑结构发生变化的报文。

#### (1) 配置 BPDU

网桥之间通过交互配置BPDU来进行根桥的选举以及端口角色的确定。配置BPDU的格式如 [图 1-1](#) 所示。

图1-1 配置 BPDU 格式



配置 BPDU 中 BPDU 数据的信息包括:

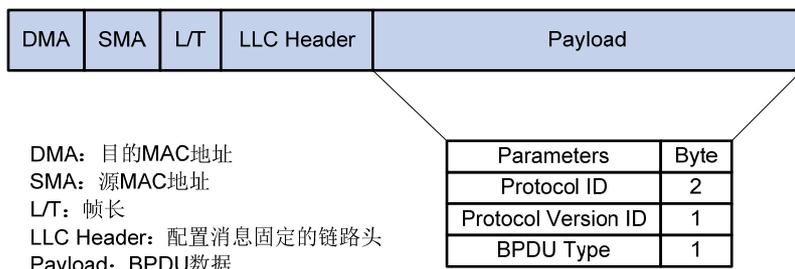
- 协议类型 (Protocol ID): 固定为 0x0000, 表示生成树协议。
- 协议版本号 (Protocol Version ID): 目前生成树有三个版本, STP 的协议版本号为 0x00。
- BPDU 类型: 配置 BPUD 类型为 0x00。
- BPDU Flags 位: BPDU 标志位, 表示是哪种 BPDU。由 8 位组成, 最低位 (0 位) 为 TC (Topology Change, 拓扑改变) 标志位; 最高位 (7 位) 为 TCA (Topology Change Acknowledge, 拓扑改变确认) 标志位; 其他 6 位保留。
- 根桥 (Root Bridge) ID: 由根桥的优先级和 MAC 地址组成。
- 根路径开销: 到根桥的路径开销。
- 指定桥 ID: 由指定桥的优先级和 MAC 地址组成。
- 指定端口 ID: 由指定端口的优先级和该端口的全局编号组成。
- Message Age: BPDU 在网络中传播的生存期。
- Max Age: BPDU 在设备中的最大生存期。
- Hello Time: BPDU 的发送周期。
- Forward Delay: 端口状态迁移的延迟时间。

其中通过根桥 ID、路径开销、指定桥 ID、指定端口 ID、Message Age、Max Age、Hello Time 和 Forward Delay 信息来保证设备完成生成树的计算过程。

## (2) TCN BPDU

如 图 1-2 所示, TCN BPDU和配置BPDU在结构上基本相同, 也是由源/目的MAC地址、L/T位、逻辑链路头和BPDU数据组成。但是TCN BPDU的BPDU数据组成非常简单, 只包含三部分信息: 协议类型、协议版本号和BPDU类型。协议类型和协议版本号字段和配置BPDU相同, BPDU类型字段的值为 0x80, 表示该BPDU为TCN BPDU。

图1-2 TCN BPDU 格式



TCN BPDU 有两个产生条件:

- 网桥上有端口转变为 **Forwarding** 状态，且该网桥至少包含一个指定端口。
- 网桥上有端口从 **Forwarding** 状态或 **Learning** 状态转变为 **Blocking** 状态。

当上述两个条件之一满足时，说明网络拓扑发生了变化，网桥需要使用 TCN BPDU 通知根桥。根桥可以通过将配置 BPDU 中对应标志位置位来通知所有网桥网络拓扑发生了变化，需要使用较短的 MAC 地址老化时间，保证拓扑的快速收敛。

## 2. STP的基本概念

### (1) 根桥

树形的网络结构必须有树根，于是 STP 引入了根桥的概念。根桥在全网中有且只有一个，其他设备则称为叶子节点。根桥会根据网络拓扑的变化而改变，因此根桥并不是固定的。

在网络初始化过程中，所有设备都视自己为根桥，生成各自的配置 BPDU 并周期性地向外发送；但当网络拓扑稳定以后，只有根桥设备才会向外发送配置 BPDU，其他设备则对其进行转发。

### (2) 根端口

所谓根端口，是指非根桥设备上离根桥最近的端口。根端口负责与根桥进行通信。非根桥设备上有且只有一个根端口，根桥上没有根端口。

### (3) 指定桥与指定端口

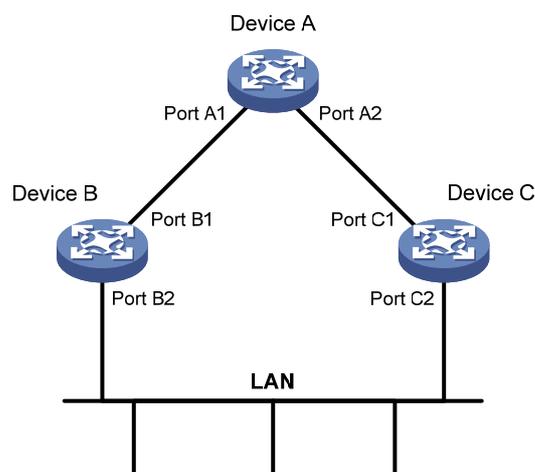
有关指定桥与指定端口的含义，请参见 [表 1-1](#) 的说明。

表1-1 指定桥与指定端口的含义

分类	指定桥	指定端口
对于一台设备而言	与本机直接相连并且负责向本机转发BPDU的设备	指定桥向本机转发BPDU的端口
对于一个局域网而言	负责向本网段转发BPDU的设备	指定桥向本网段转发BPDU的端口

如 [图 1-3](#) 所示，Device B和Device C与LAN直接相连。如果Device A通过Port A1 向Device B转发BPDU，则Device B的指定桥就是Device A，指定端口就是Device A上的Port A1；如果Device B负责向LAN转发BPDU，则LAN的指定桥就是Device B，指定端口就是Device B上的Port B2。

图1-3 指定桥与指定端口示意图



#### (4) 端口状态

STP的端口有 5 种工作状态。如 [表 1-2](#) 所示。

表1-2 STP 的端口状态

状态	描述
Disabled	该状态下的端口没有激活，不参与STP的任何动作，不转发用户流量
Listening	该状态下的端口可以接收和发送BPDU，但不转发用户流量
Learning	该状态下建立无环的转发表，不转发用户流量
Forwarding	该状态下的端口可以接收和发送BPDU，也转发用户流量
Blocking	该状态下的端口可以接收BPDU，但不转发用户流量

#### (5) 路径开销

路径开销是 STP 协议用于选择链路的参考值。STP 协议通过计算路径开销，选择较为“强壮”的链路，阻塞多余的链路，将网络修剪成无环路的树型网络结构。

### 3. STP的基本原理

STP 算法实现的基本过程如下：

#### (1) 初始状态

各设备的各端口在初始时会生成以本设备为根桥的 BPDU，根路径开销为 0，指定桥 ID 为自身设备 ID，指定端口为本端口。

#### (2) 选择根桥

网络初始化时，网络中所有的 STP 设备都认为自己是“根桥”，根桥 ID 为自身的设备 ID。通过交换 BPDU，设备之间比较根桥 ID，网络中根桥 ID 最小的设备被选为根桥。

#### (3) 选择根端口和指定端口

根端口和指定端口的选择过程如 [表 1-3](#) 所示。

表1-3 根端口和指定端口的选择过程

步骤	内容
1	非根桥设备将接收最优BPDU（最优BPDU的选择过程如表1-4所示）的那个端口定为根端口
2	设备根据根端口的BPDU和根端口的路径开销，为每个端口计算一个指定端口BPDU： <ul style="list-style-type: none"> <li>根桥 ID 替换为根端口的 BPDU 的根桥 ID；</li> <li>根路径开销替换为根端口 BPDU 的根路径开销加上根端口对应的路径开销；</li> <li>指定桥 ID 替换为自身设备的 ID；</li> <li>指定端口 ID 替换为自身端口 ID。</li> </ul>
3	设备将计算出的BPDU与角色待定端口自己的BPDU进行比较： <ul style="list-style-type: none"> <li>如果计算出的 BPDU 更优，则该端口被确定为指定端口，其 BPDU 也被计算出的 BPDU 替换，并周期性地向外发送；</li> <li>如果该端口自己的 BPDU 更优，则不更新该端口的 BPDU 并将该端口阻塞。该端口将不再转发数据，且只接收不发送 BPDU。</li> </ul>

 说明

当拓扑处于稳定状态时，只有根端口和指定端口在转发用户流量。其他端口都处于阻塞状态，只接收 STP 协议报文而不转发用户流量。

表1-4 最优 BPDU 的选择过程

步骤	内容
1	每个端口将收到的BPDU与自己的BPDU进行比较： <ul style="list-style-type: none"> <li>如果收到的 BPDU 优先级较低，则将其直接丢弃，对自己的 BPDU 不进行任何处理；</li> <li>如果收到的 BPDU 优先级较高，则用该 BPDU 的内容将自己 BPDU 的内容替换掉。</li> </ul>
2	设备将所有端口的BPDU进行比较，选出最优的BPDU

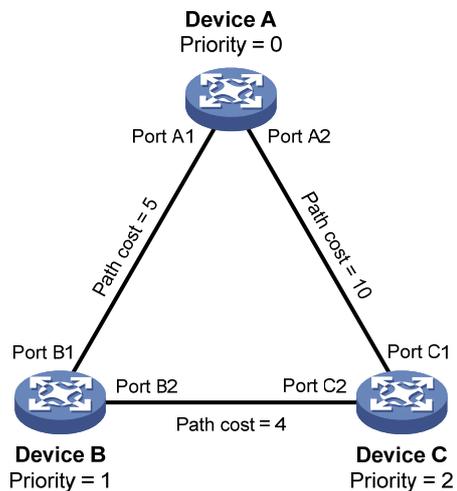
 说明

BPDU 优先级的比较规则如下：

- 根桥 ID 较小的 BPDU 优先级较高；
- 若根桥 ID 相同，则比较根路径开销：将 BPDU 中的根路径开销与本端口对应的路径开销相加，二者之和较小的 BPDU 优先级较高；
- 若根路径开销也相同，则依次比较指定桥 ID、指定端口 ID、接收该 BPDU 的端口 ID 等，上述值较小的 BPDU 优先级较高。

一旦根桥、根端口和指定端口选举成功，整个树形拓扑就建立完毕了。下面结合例子说明 STP 算法实现的具体过程。

图1-4 STP 算法实现过程组网图



如 图 1-4 所示， Device A、 Device B和Device C的优先级分别为 0、 1 和 2， Device A与Device B 之间、 Device A与Device C之间以及Device B与Device C之间链路的路径开销分别为 5、 10 和 4。

(4) 各设备的初始状态

各设备的初始状态如 表 1-5 所示。

表1-5 各设备的初始状态

设备	端口名称	端口的 BPDU
Device A	Port A1	{0, 0, 0, Port A1}
	Port A2	{0, 0, 0, Port A2}
Device B	Port B1	{1, 0, 1, Port B1}
	Port B2	{1, 0, 1, Port B2}
Device C	Port C1	{2, 0, 2, Port C1}
	Port C2	{2, 0, 2, Port C2}



说明

表 1-5 中BPDU各项的具体含义为： {根桥ID， 根路径开销， 指定桥ID， 指定端口ID}。

(5) 各设备的比较过程及结果

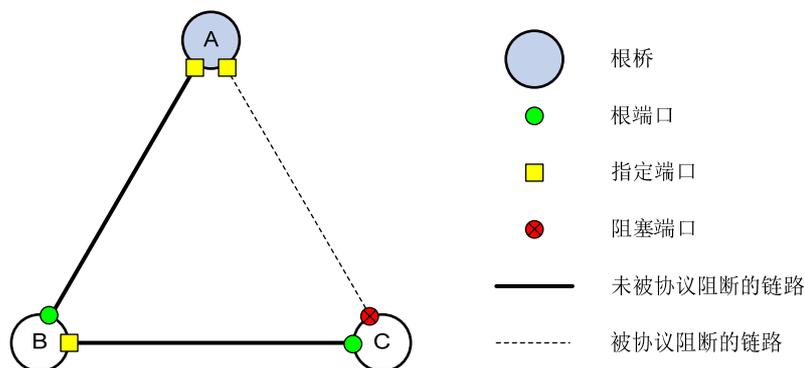
各设备的比较过程及结果如 表 1-6 所示。

表1-6 各设备的比较过程及结果

设备	比较过程	比较后端口的 BPDU
Device A	<ul style="list-style-type: none"> <li>Port A1 收到 Port B1 的 BPDU {1, 0, 1, Port B1}, 发现自己的 BPDU {0, 0, 0, Port A1}更优, 于是将其丢弃。</li> <li>Port A2 收到 Port C1 的 BPDU {2, 0, 2, Port C1}, 发现自己的 BPDU {0, 0, 0, Port A2}更优, 于是将其丢弃。</li> <li>Device A 发现自己各端口的 BPDU 中的根桥和指定桥都是自己, 于是认为自己就是根桥, 各端口的 BPDU 都不作任何修改, 此后便周期性地向外发送 BPDU。</li> </ul>	<ul style="list-style-type: none"> <li>Port A1: {0, 0, 0, Port A1}</li> <li>Port A2: {0, 0, 0, Port A2}</li> </ul>
Device B	<ul style="list-style-type: none"> <li>Port B1 收到 Port A1 的 BPDU {0, 0, 0, Port A1}, 发现其比自己的 BPDU {1, 0, 1, Port B1}更优, 于是更新自己的 BPDU。</li> <li>Port B2 收到 Port C2 的 BPDU {2, 0, 2, Port C2}, 发现自己的 BPDU {1, 0, 1, Port B2}更优, 于是将其丢弃。</li> </ul>	<ul style="list-style-type: none"> <li>Port B1: {0, 0, 0, Port A1}</li> <li>Port B2: {1, 0, 1, Port B2}</li> </ul>
	<ul style="list-style-type: none"> <li>Device B 比较自己各端口的 BPDU, 发现 Port B1 的 BPDU 最优, 于是该端口被确定为根端口, 其 BPDU 不变。</li> <li>Device B 根据根端口的 BPDU 和路径开销, 为 Port B2 计算出指定端口的 BPDU {0, 5, 1, Port B2}, 然后与 Port B2 本身的 BPDU {1, 0, 1, Port B2}进行比较, 发现计算出的 BPDU 更优, 于是 Port B2 被确定为指定端口, 其 BPDU 也被替换为计算出的 BPDU, 并周期性地向外发送。</li> </ul>	<ul style="list-style-type: none"> <li>根端口 Port B1: {0, 0, 0, Port A1}</li> <li>指定端口 Port B2: {0, 5, 1, Port B2}</li> </ul>
Device C	<ul style="list-style-type: none"> <li>Port C1 收到 Port A2 的 BPDU {0, 0, 0, Port A2}, 发现其比自己的 BPDU {2, 0, 2, Port C1}更优, 于是更新自己的 BPDU。</li> <li>Port C2 收到 Port B2 更新前的 BPDU {1, 0, 1, Port B2}, 发现其比自己的 BPDU {2, 0, 2, Port C2}更优, 于是更新自己的 BPDU。</li> </ul>	<ul style="list-style-type: none"> <li>Port C1: {0, 0, 0, Port A2}</li> <li>Port C2: {1, 0, 1, Port B2}</li> </ul>
	<ul style="list-style-type: none"> <li>Device C 比较自己各端口的 BPDU, 发现 Port C1 的 BPDU 最优, 于是该端口被确定为根端口, 其 BPDU 不变。</li> <li>Device C 根据根端口的 BPDU 和路径开销, 为 Port C2 计算出指定端口的 BPDU {0, 10, 2, Port C2}, 然后与 Port C2 本身的 BPDU {1, 0, 1, Port B2}进行比较, 发现计算出的 BPDU 更优, 于是 Port C2 被确定为指定端口, 其 BPDU 也被替换为计算出的 BPDU。</li> </ul>	<ul style="list-style-type: none"> <li>根端口 Port C1: {0, 0, 0, Port A2}</li> <li>指定端口 Port C2: {0, 10, 2, Port C2}</li> </ul>
	<ul style="list-style-type: none"> <li>Port C2 收到 Port B2 更新后的 BPDU {0, 5, 1, Port B2}, 发现其比自己的 BPDU {0, 10, 2, Port C2}更优, 于是更新自己的 BPDU。</li> <li>Port C1 收到 Port A2 周期性发来的 BPDU {0, 0, 0, Port A2}, 发现其与自己的 BPDU 一样, 于是将其丢弃。</li> </ul>	<ul style="list-style-type: none"> <li>Port C1: {0, 0, 0, Port A2}</li> <li>Port C2: {0, 5, 1, Port B2}</li> </ul>
	<ul style="list-style-type: none"> <li>Device C 比较 Port C1 的根路径开销 10 (收到的 BPDU 中的根路径开销 0+本端口所在链路的路径开销 10) 与 Port C2 的根路径开销 9 (收到的 BPDU 中的根路径开销 5+本端口所在链路的路径开销 4), 发现后者更小, 因此 Port C2 的 BPDU 更优, 于是 Port C2 被确定为根端口, 其 BPDU 不变。</li> <li>Device C 根据根端口的 BPDU 和路径开销, 为 Port C1 计算出指定端口的 BPDU {0, 9, 2, Port C1}, 然后与 Port C1 本身的 BPDU {0, 0, 0, Port A2}进行比较, 发现本身的 BPDU 更优, 于是 Port C1 被阻塞, 其 BPDU 不变。从此, Port C1 不再转发数据, 直至有触发生成树计算的新情况出现, 譬如 Device B 与 Device C 之间的链路 down 掉。</li> </ul>	<ul style="list-style-type: none"> <li>阻塞端口 Port C1: {0, 0, 0, Port A2}</li> <li>根端口 Port C2: {0, 5, 1, Port B2}</li> </ul>

经过上述比较过程之后，以Device A为根桥的生成树就确定下来了，其拓扑如 [图 1-5](#) 所示。

图1-5 计算后得到的拓扑



#### 说明

为了便于描述，本例简化了生成树的计算过程，实际的过程要更加复杂。

## 4. STP的BPDU传递机制

STP的BPDU传递机制如下：

- 当网络初始化时，所有的设备都将自己作为根桥，生成以自己为根的BPDU，并以Hello Time为周期定时向外发送。
- 接收到BPDU的端口如果是根端口，且接收的BPDU比该端口的BPDU优，则设备将BPDU中携带的Message Age按照一定的原则递增，并启动定时器为这条BPDU计时，同时将此BPDU从设备的指定端口转发出去。
- 如果指定端口收到的BPDU比本端口的BPDU优先级低时，会立刻发出自己的更好的BPDU进行回应。
- 如果某条路径发生故障，则这条路径上的根端口不会再收到新的BPDU，旧的BPDU将会因为超时而被丢弃，设备重新生成以自己为根的BPDU并向外发送，从而引发生成树的重新计算，得到一条新的通路替代发生故障的链路，恢复网络连通性。

不过，重新计算得到的新BPDU不会立刻就传遍整个网络，因此旧的根端口和指定端口由于没有发现网络拓扑变化，将仍按原来的路径继续转发数据。如果新选出的根端口和指定端口立刻就开始数据转发的话，可能会造成暂时性的环路。

## 5. STP的时间参数

在STP的计算过程中，用到了以下三个重要的时间参数：

- **Forward Delay**：用于确定状态迁移的延迟时间。缺省情况下Forward Delay时间为15秒。链路故障会引发网络重新进行生成树的计算，生成树的结构将发生相应的变化。不过重新计算得到的新BPDU无法立刻传遍整个网络，如果新选出的根端口和指定端口立刻就开始数据转发的话，可能会造成暂时性的环路。为此，生成树协议在端口由Blocking状态向Forwarding状态迁移的过程中设置了Listening和Learning状态作为过渡（Listening和Learning状态都会持续Forward Delay时间），并规定状态迁移需要等待Forward Delay时间，以保持与远端

的设备状态切换同步。新选出的根端口和指定端口要经过 2 倍的 Forward Delay 延时后才能进入转发状态，这个延时保证了新的 BPDU 已经传遍整个网络。

- **Hello Time:** 用于设备检测链路是否存在故障。缺省情况下 Hello Time 为 2 秒。生成树协议每隔 Hello Time 时间会发送 BPDU，以确认链路是否存在故障。如果设备在超时时间（超时时间=超时时间因子×3×Hello Time）内没有收到 BPDU，则会由于消息超时而重新计算生成树。
- **Max Age:** 用于判断 BPDU 在设备内的保存时间是否“过时”，设备会将过时的 BPDU 丢弃。缺省情况下 Max Age 时间为 20 秒。在 MSTP 的 CIST 上，设备根据 Max Age 时间来确定端口收到的 BPDU 是否超时。如果端口收到的 BPDU 超时，则需要对该 MSTI 重新计算。Max Age 时间对 MSTP 的 MSTI 无效。

STP 每隔一个 Hello Time 发送一个 BPDU，并且引入 Keepalive 机制。Hello 包的发送可以避免最大失效定时器溢出。如果最大失效定时器溢出，通常表明有连接错误发生。此时，STP 会进入监听状态。STP 要从连接错误中恢复过来，一般需要 50 秒的时间。其中 BPDU 最长的失效时间 20 秒；监听状态持续 15 秒；学习状态持续 15 秒。

为保证网络拓扑的快速收敛，需要配置合适的时间参数。上述三个时间参数之间应满足以下关系，否则会引起网络的频繁震荡：

- $2 \times (\text{Forward Delay} - 1 \text{ 秒}) \geq \text{Max Age}$
- $\text{Max Age} \geq 2 \times (\text{Hello Time} + 1 \text{ 秒})$

## 1.1.2 RSTP简介

RSTP 由 IEEE 制定的 802.1w 标准定义，它在 STP 基础上进行了改进，实现了网络拓扑的快速收敛。其“快速”体现在，当一个端口被选为根端口和指定端口后，其进入转发状态的延时将大大缩短，从而缩短了网络最终达到拓扑稳定所需要的时间。

### 1. RSTP的协议报文

RSTP 也是通过在设备之间传递 BPDU 来确定网络的拓扑结构。RSTP 的 BPDU 格式和 STP 的配置 BPDU 格式非常相似，仅在以下几个信息有所不同：

- BPDU 协议类型变为 0x02，表示为 RSTP 的 BPDU。
- BPDU 协议版本号为 0x02，表示为 RSTP 协议。
- Flags 位字段使用了全 8 位。
- RSTP 在 BPDU 报文的最后增加了 Version1 Length 字段。该字段的值为 0x00，表示本 BPDU 中不包含 Version 1 内容。

在拓扑改变时，RSTP 的拓扑改变处理过程不再使用 TCN BPDU，而使用 Flags 位中 TC 置位的 RST BPDU 取代 TCN BPDU，并通过泛洪方式快速的通知到整个网络。

### 2. RSTP的基本概念

#### (1) 端口角色

RSTP 中根端口和指定端口角色的定义和 STP 相同。与 STP 相比，RSTP 增加了三种端口角色替换端口（Alternate Port）、备份端口（Backup Port）和边缘端口（Edge Port）。

- 替换端口为网桥提供一条到达根桥的备用路径，当根端口或主端口被阻塞后，替换端口将成为新的根端口或主端口。

- 备份端口为网桥提供了到达同一个物理网段的冗余路径，当指定端口失效后，备份端口将转换为新的指定端口。当开启了生成树协议的同台设备上的两个端口互相连接而形成环路时，设备会将其中一个端口阻塞，该端口就是备份端口。
- 边缘端口是不与其他设备或网段连接的端口，边缘端口一般与用户终端设备直接相连。

## (2) 端口状态

RSTP将端口状态缩减为三个，分别为Discarding、Learning和Forwarding状态。STP中的Disabled、Blocking和Listening状态在RSTP中都对应为Discarding状态，如 [表 1-7](#) 所示。

表1-7 RSTP 的端口状态

STP 端口状态	RSTP 端口状态	是否发送 BPDU	是否进行 MAC 地址学习	是否收发用户流量
Disabled	Discarding	否	否	否
Blocking	Discarding	否	否	否
Listening	Discarding	是	否	否
Learning	Learning	是	是	否
Forwarding	Forwarding	是	是	是

## 3. RSTP的基本原理

进行RSTP计算时，端口会在 Discarding 状态完成角色的确定，当端口确定为根端口和指定端口后，经过 Forward Delay 端口会进入 Learning 状态；当端口确定为替代端口，端口会维持在 Discarding 状态。

处于 Learning 状态的端口其处理方式和 STP 相同，开始学习 MAC 地址并在 Forward Delay 后进入 Forwarding 状态开始收发用户流量。

在 RSTP 中，根端口的端口状态快速迁移的条件是：本设备上旧的根端口已经停止转发数据，而且上游指定端口已经开始转发数据。

在 RSTP 中，指定端口的端口状态快速迁移的条件是：指定端口是边缘端口（即该端口直接与用户终端相连，而没有连接到其他设备或共享网段上）或者指定端口与点对点链路（即两台设备直接相连的链路）相连。如果指定端口是边缘端口，则指定端口可以直接进入转发状态；如果指定端口连接着点对点链路，则设备可以通过与下游设备握手，得到响应后即刻进入转发状态。

## 4. RSTP中的BPDU处理

相比于 STP，RSTP 对 BPDU 的发送方式做了改进，RSTP 中网桥可以自行从指定端口发送 RST BPDU，不需要等待来自根桥的 RST BPDU，BPDU 的发送周期为 Hello Time。

由于 RSTP 中网桥可以自行从指定端口发送 RST BPDU，所以在网桥之间可以提供一种保活机制，即在一定时间内网桥没有收到对端网桥发送的 RST BPDU，即可认为和对端网桥的连接中断。

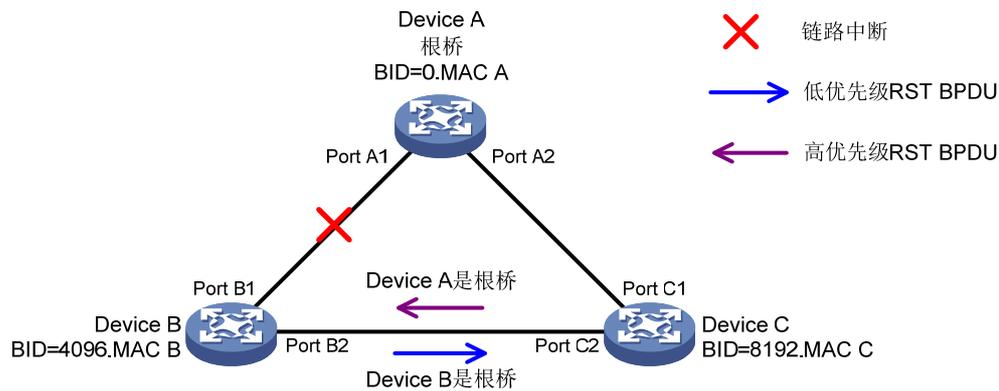
RSTP 规定，若在三个连续的 Hello Time 时间内网桥没有收到对端指定桥发送的 RST BPDU，则网桥端口保存的 RST BPDU 老化，认为与对端网桥连接中断。新的老化机制大大加快了拓扑变化的感知，从而可以实现快速收敛。

在 RSTP 中，如果阻塞状态的端口收到低优先级的 RST BPDU，也可以立即对其做出回应。

如 [图 1-6](#)，网络中 Device A 为根桥，Device C 阻塞和 Device B 相连的端口。当 Device B 和根桥之间的链路中断时，Device B 会发送以自己为根桥的 RST BPDU。Device C 收到 Device B 发送的 RST

BPDU后，经过比较，Device B的值RST BPDU为低优先级的RST BPDU，所以Device C的端口会立即对该RST BPDU做出回应，发送优先级更高的RST BPDU。Device B收到Device C发送的RST BPDU后，将会停止发送RST BPDU，并将和Device C连接的端口确定为根端口。

图1-6 RSTP 对低优先级 RST BPDU 的处理



### 1.1.3 PVST简介

STP 和 RSTP 在局域网内的所有网桥都共享一棵生成树，不能按 VLAN 阻塞冗余链路，所有 VLAN 的报文都沿着一棵生成树进行转发。而 PVST 则可以在每个 VLAN 内都拥有一棵生成树，能够有效地提高链路带宽的利用率。PVST 可以简单理解为在每个 VLAN 上运行一个 RSTP 协议，不同 VLAN 之间的生成树完全独立。

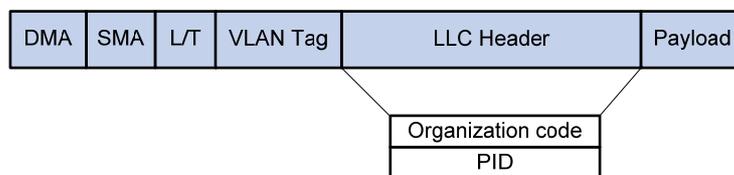
运行 PVST 的 H3C 设备可以与运行 Rapid PVST 或 PVST 的第三方设备互通。当运行 PVST 的 H3C 设备之间互联，或运行 PVST 的 H3C 设备与运行 Rapid PVST 的第三方设备互通时，H3C 设备支持像 RSTP 一样的快速收敛。

#### 1. PVST的协议报文

如 图 1-7，从报文结构对上看，PVST的BPDU和RSTP的BPDU不同在于以下几点：

- 报文的目的地 MAC 地址改变，变为私有 MAC 地址 01-00-0c-cc-cc-cd。
- 报文携带 VLAN 标签，确定该协议报文归属的 VLAN。
- 报文配置消息固定链路头字段添加 Organization code 和 PID 字段。

图1-7 PVST 报文格式



根据端口类型的不同，PVST 所发送的 BPDU 格式也有所差别：

- 对于 Access 端口，PVST 将根据该 VLAN 的状态发送 RSTP 格式的 BPDU。
- 对于 Trunk 端口和 Hybrid 端口，PVST 将在缺省 VLAN 内根据该 VLAN 的状态发送 RSTP 格式的 BPDU，而对于其他本端口允许通过的 VLAN，则发送 PVST 格式的 BPDU。

## 2. PVST的基本概念

PVST 为每个 VLAN 维护一个单独的生成树实例。每个 VLAN 都将运行单个生成树，允许以每个 VLAN 为基础开启或关闭生成树。每个 VLAN 内的生成树实例都有单独的网络拓扑结构，相互之间没有影响。这样既可以消除了 VLAN 内的冗余环路，还可以实现不同 VLAN 间负载分担。

PVST 的端口角色和端口状态和 RSTP 相同，能够实现快速收敛。

## 3. PVST的基本原理

PVST 借助 MSTP 的实例和 VLAN 映射关系模型，将 MSTP 每个实例映射一个 VLAN。PVST 中每个 VLAN 独立运行 RSTP，独立运算。PVST 在缺省 VLAN 上通过 RSTP 报文进行拓扑运算；在其他 VLAN 上通过带 VLAN Tag 的 PVST 报文进行拓扑运算。

### 1.1.4 MSTP简介

#### 1. MSTP的产生背景

##### (1) STP、RSTP 和 PVST 存在的不足

STP 不能快速迁移，即使是在点对点链路或边缘端口，也必须等待两倍的 Forward Delay 的时间延迟，端口才能迁移到转发状态。

RSTP 可以快速收敛，但和 STP 一样还存在如下缺陷：由于局域网内所有 VLAN 都共享一棵生成树，因此所有 VLAN 的报文都沿这棵生成树进行转发，不能按 VLAN 阻塞冗余链路，也无法在 VLAN 间实现数据流量的负载均衡。

对于 PVST 而言，由于每个 VLAN 都需要生成一棵树，因此 PVST BPDU 的通信量将与 Trunk/Hybrid 端口上允许通过的 VLAN 数量成正比。而且当 VLAN 数量较多时，维护多棵生成树的计算量以及资源占用量都将急剧增长，特别是当允许通过很多 VLAN 的 Trunk/Hybrid 端口的链路状态发生改变时，对应生成树的状态都要重新计算，网络设备的 CPU 将不堪重负。

##### (2) MSTP 的特点

MSTP 由 IEEE 制定的 802.1s 标准定义，它可以弥补 STP、RSTP 和 PVST 的缺陷，既可以快速收敛，也能使不同 VLAN 的流量沿各自的路径转发，从而为冗余链路提供了更好的负载分担机制。

MSTP 的特点如下：

- MSTP 把一个交换网络划分成多个域，每个域内形成多棵生成树，生成树之间彼此独立。
- MSTP 通过设置 VLAN 与生成树的对应关系表（即 VLAN 映射表），将 VLAN 与生成树联系起来。并通过“实例”的概念，将多个 VLAN 捆绑到一个实例中，从而达到了节省通信开销和降低资源占用率的目的。
- MSTP 将环路网络修剪成为一个无环的树型网络，避免报文在环路网络中的增生和无限循环，同时还提供了数据转发的多个冗余路径，在数据转发过程中实现 VLAN 数据的负载分担。
- MSTP 兼容 STP 和 RSTP，部分兼容 PVST。

#### 2. MSTP的协议报文

如 [图 1-8](#)，MST BPDU 和 RST BPDU 的前 36 个字节格式是相同的，其中 BPDU 协议版本号为 0x03，表示 MSTP 协议，BPDU 类型为 0x02，表示为 RST/MST BPDU。

图1-8 MSTP 的 BPDU 格式

Parameters	Byte
Protocol ID	2
Protocol Version ID	1
BPDU Type	1
Flags	1
Root ID	8
Root Path Cost	4
Bridge ID	8
Port ID	2
Message Age	2
Max Age	2
Hello Time	2
Forward Delay	2
Version1 Length=0	1
Version3 Length	2
MST Configuration ID	51
CIST IRPC	4
CIST Bridge ID	8
CIST Remaining ID	1
MSTI Configuration Messages	LEN

MSTP 专有字段

RST BPDU 中的 Root ID 字段在 MSTP 中表示 CIST（Common and Internal Spanning Tree，公共和内部生成树）总根 ID，Root Path Cost 字段在 MSTP 中表示 CIST 外部路径开销（External Path Cost, EPC），Bridge ID 字段在 MSTP 中表示 CIST 域根 ID，Port ID 字段在 MSTP 中表示 CIST 指定端口 ID。

从第 37 字节开始是 MSTP 的专有字段：

- **Version3 Length:** 表示 MSTP 专有字段长度，该字段用于接收到 BPDU 后进行校验。
- **MST 配置标识(Configuration ID):** 包含格式选择符(Format Selector)、域名(Configuration Name)、修订级别(Revision Level)和配置摘要(Configuration Digest)四个字段。其中格式选择符字段固定为 0x00，其余三个字段用来判断网桥是否属于某 MST 域。
- **CIST 内部路径开销(Internal Root Path Cost, IRPC):** 表示发送此 BPDU 的网桥到达 CIST 域根的路径开销。
- **CIST Bridge ID:** 表示发送此 BPDU 的网桥 ID。
- **CIST 剩余跳数:** 用来限制 MST 域的规模。从 CIST 域根开始，BPDU 每经过一个网桥的转发，跳数就被减 1；网桥将丢弃收到的跳数为 0 的 BPDU，使出于最大跳数外的网桥无法参与生成树的计算，从而限制了 MST 域的规模。CIST 剩余跳数默认值为 20。
- **MSTI Configuration Messages:** 包含 0 个或最多 64 个 MSTI (Multiple Spanning Tree Instance, 多生成树实例) 配置信息，MSTI 配置信息数量由域内 MST 实例数决定，每一个 MSTI 配置信息长度为 16 字节。

### 3. MSTP的基本概念

图1-9 MSTP 的基本概念示意图

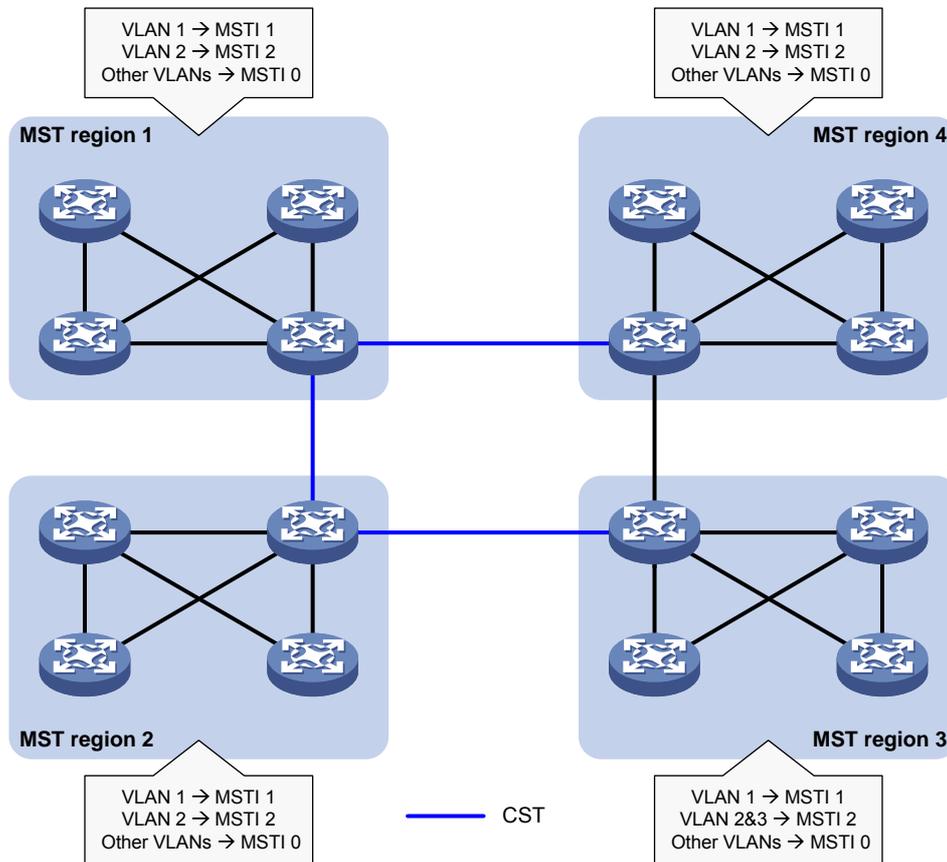
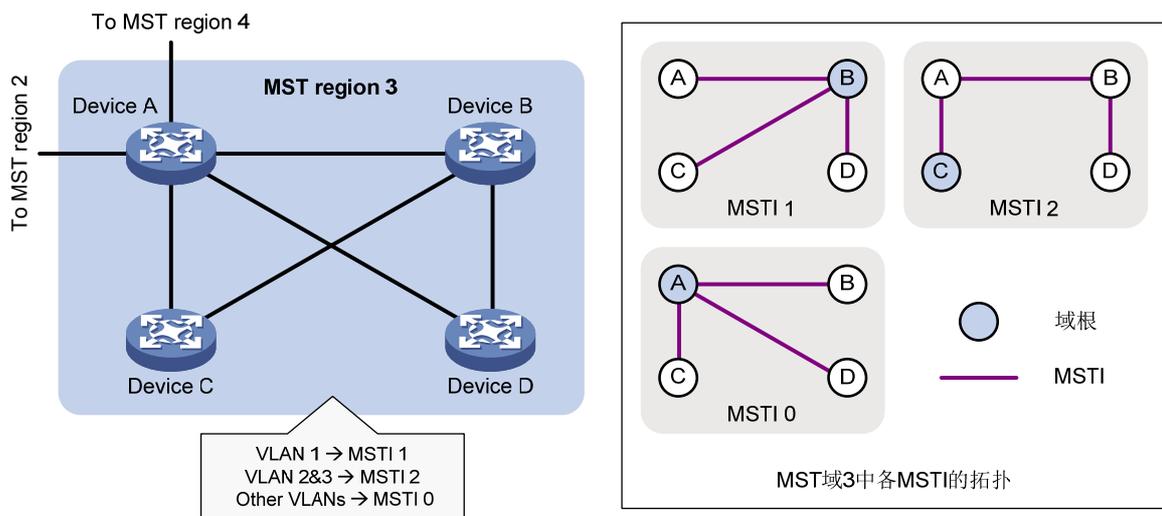


图1-10 MST 域 3 详图



在如 [图 1-9](#) 所示的交换网络中有四个MST域，每个MST域都由四台设备构成，所有设备都运行MSTP；为了看清MST域内的情形，我们以MST域 3 为例放大来看，如 [图 1-10](#) 所示。下面就结合这两张图来介绍一些MSTP中的基本概念：

#### (1) MST 域

MST 域（Multiple Spanning Tree Regions，多生成树域）是由交换网络中的多台设备以及它们之间的网段所构成。这些设备具有下列特点：

- 都开启了生成树协议。
- 域名相同。
- VLAN 与 MSTI 间映射关系的配置相同。
- MSTP 修订级别的配置相同。
- 这些设备之间有物理链路连通。

一个交换网络中可以存在多个MST域，用户可以通过配置将多台设备划分在一个MST域内。如在 [图 1-9](#) 所示的网络中就有MST域 1~MST域 4 这四个MST域，每个域内的所有设备都具有相同的MST域配置。

#### (2) MSTI

一个MST域内可以通过MSTP生成多棵生成树，各生成树之间彼此独立并分别与相应的VLAN对应，每棵生成树都称为一个MSTI（Multiple Spanning Tree Instance，多生成树实例）。如在 [图 1-10](#) 所示的MST域 3 中，包含有三个MSTI：MSTI 1、MSTI 2 和MSTI 0。

#### (3) VLAN 映射表

VLAN映射表是MST域的一个属性，用来描述VLAN与MSTI间的映射关系。如 [图 1-10](#) 中MST域 3 的VLAN映射表就是：VLAN 1 映射到MSTI 1，VLAN 2 和VLAN 3 映射到MSTI 2，其余VLAN映射到MSTI 0。MSTP就是根据VLAN映射表来实现负载分担的。

#### (4) CST

CST（Common Spanning Tree，公共生成树）是一棵连接交换网络中所有MST域的单生成树。如果把每个MST域都看作一台“设备”，CST就是这些“设备”通过STP协议、RSTP协议计算生成的一棵生成树。如 [图 1-9](#) 中的蓝色线条描绘的就是CST。

#### (5) IST

IST（Internal Spanning Tree，内部生成树）是MST域内的一棵生成树，它是一个特殊的MSTI，通常也称为MSTI 0，所有VLAN缺省都映射到MSTI 0 上。如 [图 1-10](#) 中的MSTI 0 就是MST域 3 内的IST。

#### (6) CIST

CIST（Common and Internal Spanning Tree，公共和内部生成树）是一棵连接交换网络内所有设备的单生成树，所有MST域的IST再加上CST就共同构成了整个交换网络的一棵完整的单生成树，即CIST。如 [图 1-9](#) 中各MST域内的IST（即MSTI 0）再加上MST域间的CST就构成了整个网络的CIST。

#### (7) 域根

域根（Regional Root）就是MST域内IST或MSTI的根桥。MST域内各生成树的拓扑不同，域根也可能不同。如在 [图 1-10](#) 所示的MST域 3 中，MSTI 1 的域根为Device B，MSTI 2 的域根为Device C，而MSTI 0（即IST）的域根则为Device A。

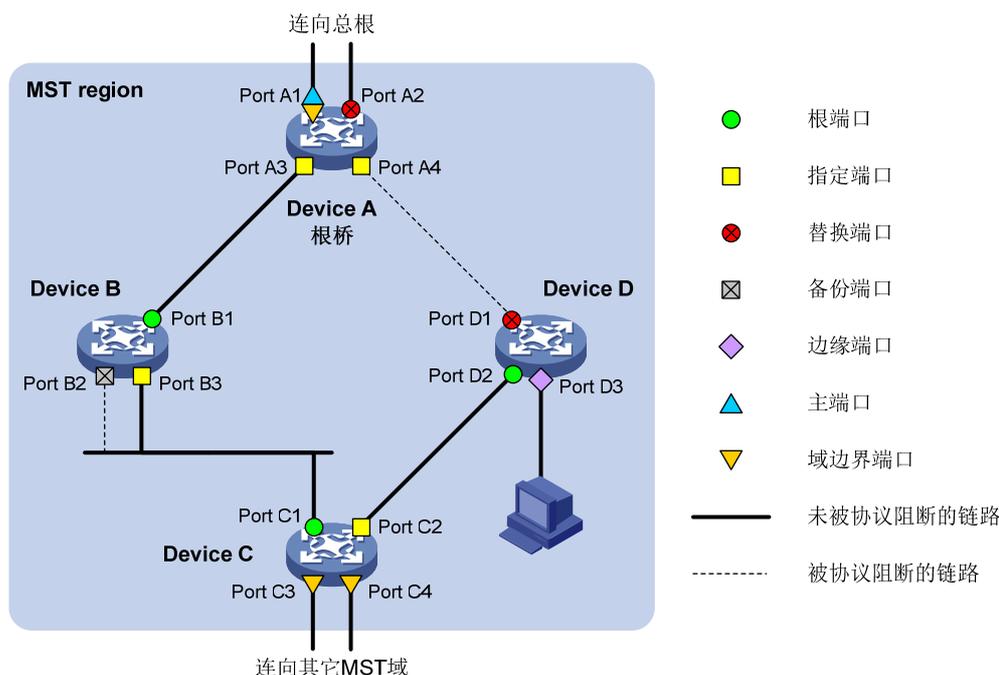
#### (8) 总根

总根（Common Root Bridge）就是CIST的根桥。如 图 1-9 中CIST的总根就是MST域 1 中的某台设备。

### (9) 端口角色

端口在不同的MSTI中可以担任不同的角色。如 图 1-11 所示，在由Device A、Device B、Device C和Device D共同构成的MST域中，Device A的端口Port A1 和Port A2 连向总根方向，Device B的端口Port B2 和Port B3 相连而构成环路，Device C的端口Port C3 和Port C4 连向其他MST域，Device D的端口Port D3 直接连接用户主机。

图1-11 端口角色示意图



如 图 1-11 所示，MSTP计算过程中涉及到的主要端口角色有以下几种：

- 根端口（Root Port）：在非根桥上负责向根桥方向转发数据的端口就称为根端口，根桥上没有根端口。
- 指定端口（Designated Port）：负责向下游网段或设备转发数据的端口就称为指定端口。
- 替换端口（Alternate Port）：是根端口和主端口的备份端口。当根端口或主端口被阻塞后，替换端口将成为新的根端口或主端口。
- 备份端口（Backup Port）：是指定端口的备份端口。当指定端口失效后，备份端口将转换为新的指定端口。当开启了生成树协议的一台设备上的两个端口互相连接而形成环路时，设备会将其中一个端口阻塞，该端口就是备份端口。
- 边缘端口（Edge Port）：不与其他设备或网段连接的端口就称为边缘端口，边缘端口一般与用户终端设备直接相连。
- 主端口（Master Port）：是将 MST 域连接到总根的端口（主端口不一定在域根上），位于整个域到总根的最短路径上。主端口是 MST 域中的报文去往总根的必经之路。主端口在 IST/CIST 上的角色是根端口，而在其他 MSTI 上的角色则是主端口。

- 域边界端口（Boundary Port）：是位于 MST 域的边缘、并连接其他 MST 域或 MST 域与运行 STP/RSTP 的区域的端口。主端口同时也是域边界端口。在进行 MSTP 计算时，域边界端口在 MSTI 上的角色与 CIST 的角色一致，但主端口除外——主端口在 CIST 上的角色为根端口，在其他 MSTI 上的角色才是主端口。

(10) 端口状态

MSTP中的端口状态可分为三种，如 [表 1-8](#) 所示。同一端口在不同的MSTI中的端口状态可以不同。

表1-8 MSTP 的端口状态

状态	描述
Forwarding	该状态下的端口可以接收和发送BPDU，也转发用户流量
Learning	是一种过渡状态，该状态下的端口可以接收和发送BPDU，但不转发用户流量
Discarding	该状态下的端口可以接收和发送BPDU，但不转发用户流量

端口状态和端口角色是没有必然联系的，[表 1-9](#) 给出了各种端口角色能够具有的端口状态（“√”表示此端口角色能够具有此端口状态；“-”表示此端口角色不能具有此端口状态）。

表1-9 各种端口角色具有的端口状态

端口角色（右） 端口状态（下）	根端口/主端口	指定端口	替换端口	备份端口
Forwarding	√	√	-	-
Learning	√	√	-	-
Discarding	√	√	√	√

#### 4. MSTP的基本原理

MSTP 将整个二层网络划分为多个 MST 域，各域之间通过计算生成 CST；域内则通过计算生成多棵生成树，每棵生成树都被称为是一个 MSTI，其中的 MSTI 0 也称为 IST。MSTP 同 STP 一样，使用 BPDU 进行生成树的计算，只是 BPDU 中携带的是设备上 MSTP 的配置信息。

(1) CIST 生成树的计算

通过比较 BPDU 后，在整个网络中选择一个优先级最高的设备作为 CIST 的根桥。在每个 MST 域内 MSTP 通过计算生成 IST；同时 MSTP 将每个 MST 域作为单台设备对待，通过计算在域间生成 CST。CST 和 IST 构成了整个网络的 CIST。

(2) MSTI 的计算

在MST域内，MSTP根据VLAN与MSTI的映射关系，针对不同的VLAN生成不同的MSTI。每棵生成树独立进行计算，计算过程与STP计算生成树的过程类似，请参见“[1.1.1 3. STP的基本原理](#)”。

MSTP 中，一个 VLAN 报文将沿着如下路径进行转发：

- 在 MST 域内，沿着其对应的 MSTI 转发；
- 在 MST 域间，沿着 CST 转发。

## 5. MSTP在设备上的实现

MSTP 同时兼容 STP 和 RSTP。STP 和 RSTP 的协议报文都可以被运行 MSTP 协议的设备识别并应用于生成树计算。设备除了提供 MSTP 的基本功能外，还从用户的角度出发，提供了如下便于管理的特殊功能：

- 根桥保持；
- 根桥备份；
- 根保护功能；
- BPDU 保护功能；
- 环路保护功能；
- 防 TC-BPDU 攻击保护功能；
- 端口角色限制功能；
- TC-BPDU 传播限制功能；

### 1.1.5 快速收敛机制

在 STP 中，为避免临时环路，端口从开启到进入转发状态需要等待默认 30 秒的时间，如果想要缩短这个时间，只能手工方式将 Forward Delay 设置为较小值。但是 Forward Delay 是由 Hello Time 和网络直径共同决定的一个参数，如果将 Forward Delay 设置太小，可能会导致临时环路的产生，影响网络的稳定性。

目前，RSTP/PVST/MSTP 都支持快速收敛机制。快速收敛机制包括边缘端口机制、根端口快速切换机制、指定端口快速切换机制。其中指定端口快速切换机制也称为 P/A（Proposal/Agreement, 请求/回应）机制。

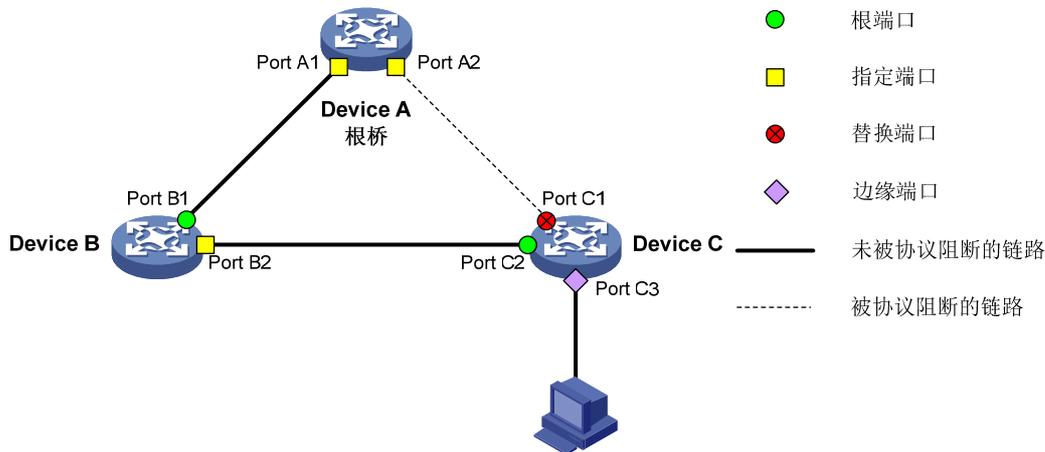
#### 1. 边缘端口机制

当端口直接与用户终端相连，而没有连接到其他网桥或局域网网段上时，该端口即为边缘端口。

边缘端口连接的是终端，当网络拓扑变化时，边缘端口不会产生临时环路，所以边缘端口可以略过两个 Forward Delay 的时间，直接进入 Forwarding 状态，无需任何延时。

由于网桥无法自动判断端口是否直接与终端相连，所以用户需要手工将与端口连接的端口配置为边缘端口。

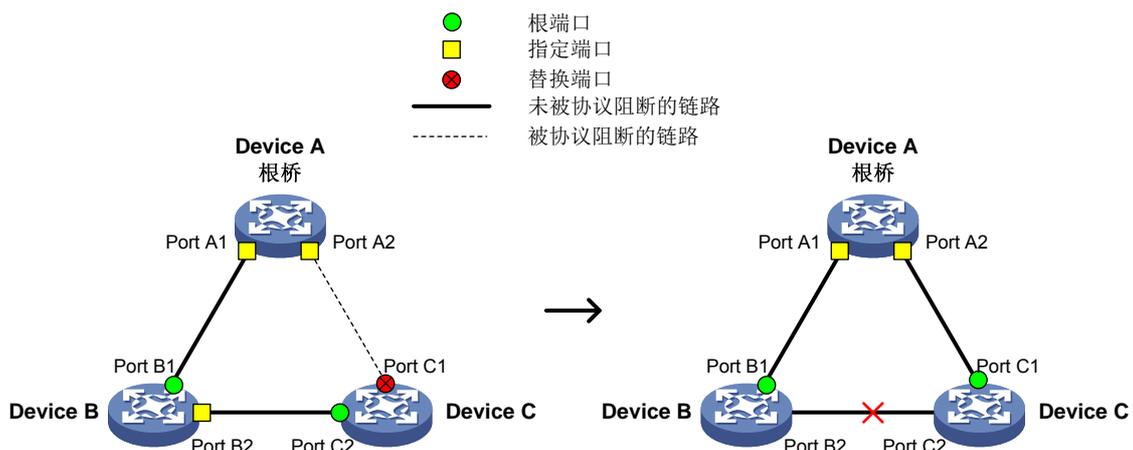
图1-12 边缘端口示意图



## 2. 根端口快速切换机制

当旧的根端口进入阻塞状态，网桥会选择优先级最高的替代端口作为新的根端口，如果当前新根端口连接的对端网桥的指定端口处于 **Forwarding** 状态，则新根端口可以立刻进入 **Forwarding** 状态。

图1-13 根端口快速切换示意图



如 [图 1-13](#)，Device C有两个端口，一个为根端口另一个为替代端口，当根端口链路中断时，替代端口会立刻成为新的根端口并进入**Forwarding**状态，期间不需要延时。

## 3. P/A机制

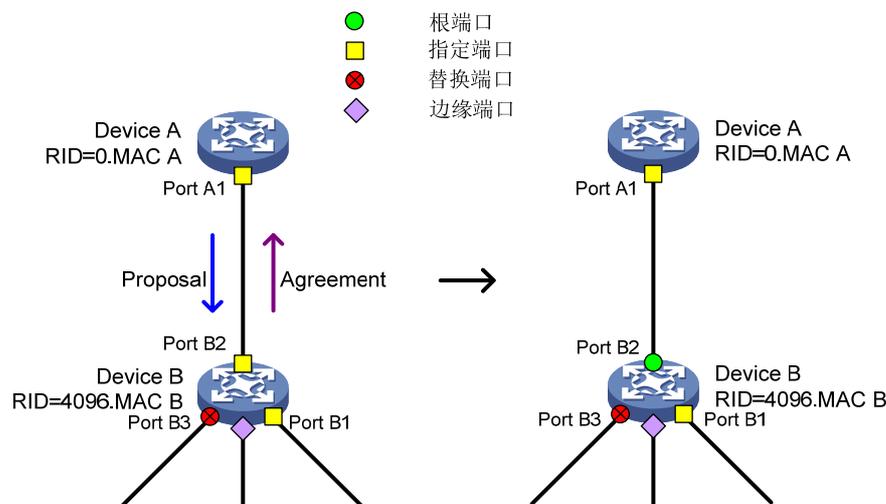
P/A 机制是指指定端口可以通过与对端网桥进行一次握手，即可快速进入转发状态，期间不需要任何定时器。P/A 机制的前提条件是：握手必须在点到点链路上进行。有点到点链路作为前提，P/A 机制可以实现网络拓扑的逐链路收敛，而不必像 STP，需要被动等待 30 秒的时间以确保全网实现收敛。

### (1) RSTP/PVST 的 P/A 机制

当新链路连接或故障链路恢复时，链路两端的端口初始都为指定端口并处于阻塞状态。当指定端口处于 **Discarding** 状态和 **Learning** 状态，其所发送的 BPDU 中 **Proposal** 位将被置位，端口角色为指

定端口。收到 Proposal 置位的 BPDUs 后，网桥会判断接收端口是否为根端口，如果是，网桥会启动同步过程。同步过程指网桥阻塞除边缘端口之外的所有端口，在本网桥层面消除环路产生的可能。

图1-14 RSTP/PVST 的 P/A 机制实现快速收敛



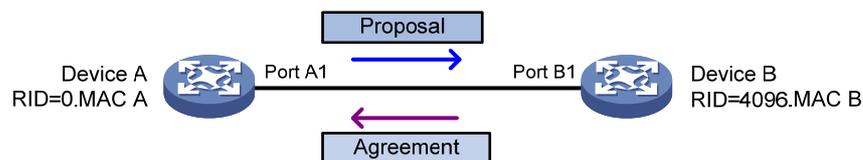
如 图 1-14，当 Device A 和 Device B 之间的链路连接后，P/A 机制处理过程如下：

- Device A 从端口 Port A1 发送 Proposal 置位的 BPDUs 给 Device B。
- Device B 收到 Proposal BPDUs 后，判断端口 Port B2 为根端口，启动同步过程阻塞指定端口 Port B1 和替代端口 Port B3 避免环路产生，然后将根端口 Port B2 设置为转发状态，并向 Device A 回复 Agreement BPDUs。
- Device A 收到 Agreement BPDUs 后，指定端口 Port A1 立即进入转发状态。
- Device A 的端口 Port A1 和 Device B 的端口 Port B2 均进入转发状态，P/A 收敛过程结束。

## (2) MSTP 的 P/A 机制

在 MSTP 中，上游网桥发送的 Proposal BPDUs 中的 Proposal 位和 Agreement 位均置位，下游网桥收到 Proposal 位和 Agreement 位均置位的 BPDUs 后，执行同步操作然后回应 Agreement 置位的 BPDUs，使得上游指定端口快速进入转发状态。

图1-15 MSTP 的 P/A 机制实现快速收敛



如 图 1-15，Device A 和 Device B 之间的 P/A 机制处理过程如下：

- Device A 从端口 Port A1 发送 Proposal 位和 Agreement 位均置位的 BPDUs 给 Device B。
- Device B 收到 Proposal 位和 Agreement 位均置位的 BPDUs 后，判断端口 Port B1 为根端口，执行同步操作然后将根端口 Port B1 设置为转发状态，并向 Device A 回复 Agreement BPDUs。
- Device A 收到 Agreement BPDUs 后，指定端口 Port A1 立即进入转发状态。
- Device A 的端口 Port A1 和 Device B 的端口 Port B1 均进入转发状态，P/A 收敛过程结束。

从 RSTP/PVST 和 MSTP 的 P/A 机制处理过程可以看到，P/A 机制没有依赖任何定时器，可以实现快速的收敛。

需要注意的是，如果指定端口发出的 Proposal BPDU 后没有收到 Agreement BPDU，则该端口将切换到 STP 方式，需要等待 30 秒时间才能进入转发状态。

### 1.1.6 协议规范

与生成树相关的协议规范有：

- IEEE 802.1D: Media Access Control (MAC) Bridges
- IEEE 802.1w: Part 3: Media Access Control (MAC) Bridges—Amendment 2: Rapid Reconfiguration
- IEEE 802.1s: Virtual Bridged Local Area Networks—Amendment 3: Multiple Spanning Trees
- IEEE 802.1Q-REV/D1.3: Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks—Clause 13: Spanning tree Protocols

## 1.2 生成树配置任务简介

生成树协议包括 STP、RSTP、PVST 和 MSTP 四种类型。在配置生成树之前，首先需明确要使用的生成树协议类型，并规划好各设备在其中的角色（根桥或叶子节点）；然后根据所选择的协议类型及规划好的设备角色，依照本节中的表格进行配置。配置时，请注意以下几点：

- 当同时开启 MVRP（Multiple VLAN Registration Protocol，多 VLAN 注册协议）和生成树协议时，MVRP 报文将沿 MSTI 传播。因此当同时开启了 MVRP 和生成树协议时，如果希望通过 MVRP 在网络中发布某个 VLAN，则配置生成树协议的 VLAN 映射表时要保证将该 VLAN 映射到 MSTI 上。有关 MVRP 的详细介绍，请参见“二层技术-以太网交换配置指导”中的“MVRP”。
- 在生成树和分布式聚合结合使用的组网中，用户需要保证分布式聚合系统中成员设备生成树全局配置、DR 口及 IPP 口生成树端口配置完全一致，否则可能会导致网络有震荡。分布式聚合系统中 IPP 口不参与生成树的拓扑计算。DR 口的生成树信息需要在分布式聚合系统的 Primary 设备上查看。
- 生成树协议与以下功能互斥：RRPP 功能、Smart Link 功能。
- 系统视图下的配置全局生效；二层以太网接口视图下的配置只对当前端口生效；二层聚合接口视图下的配置只对当前接口生效；聚合成员端口上的配置，只有当成员端口退出聚合组后才能生效。
- 在二层聚合接口上开启生成树协议后，生成树的相关计算只在二层聚合接口上进行，聚合成员端口不再参与生成树计算。二层聚合接口的所有选中成员端口上生成树协议的开启/关闭状态以及端口转发状态与二层聚合接口保持一致。尽管聚合成员端口不参与生成树计算，但端口上的生成树相关配置仍然保留，当端口退出聚合组时，该端口将采用这些配置参与生成树计算。

## 1.2.1 STP配置任务简介

表1-10 STP 配置任务简介

配置任务		说明	详细配置
配置根桥	配置生成树的工作模式	必选 通过本配置将生成树的工作模式配置为STP模式	<a href="#">1.3.1</a>
	配置根桥和备份根桥	可选	<a href="#">1.3.3</a>
	配置设备的优先级	可选	<a href="#">1.3.4</a>
	配置交换网络的网络直径	可选	<a href="#">1.3.6</a>
	配置生成树的时间参数	可选	<a href="#">1.3.7</a>
	配置超时时间因子	可选	<a href="#">1.3.8</a>
	配置端口发送BPDU的速率	可选	<a href="#">1.3.9</a>
	打开端口状态变化信息显示开关	可选	<a href="#">1.3.15</a>
	开启生成树协议	必选	<a href="#">1.3.16</a>
配置叶子节点	配置生成树的工作模式	必选 通过本配置将生成树的工作模式配置为STP模式	<a href="#">1.3.1</a>
	配置设备的优先级	可选	<a href="#">1.3.4</a>
	配置超时时间因子	可选	<a href="#">1.3.8</a>
	配置端口发送BPDU的速率	可选	<a href="#">1.3.9</a>
	配置端口的路径开销	可选	<a href="#">1.3.11</a>
	配置端口的优先级	可选	<a href="#">1.3.12</a>
	打开端口状态变化信息显示开关	可选	<a href="#">1.3.15</a>
	开启生成树协议	必选	<a href="#">1.3.16</a>
配置TC Snooping功能		可选	<a href="#">1.3.21</a>
配置生成树保护功能		可选	<a href="#">1.3.22</a>
配置被BPDU保护功能关闭的端口不再自动恢复		可选	<a href="#">1.3.24</a>
配置生成树的网管功能		可选	<a href="#">1.3.25</a>

## 1.2.2 RSTP配置任务简介

表1-11 RSTP 配置任务简介

配置任务		说明	详细配置
配置根桥	配置生成树的工作模式	必选 通过本配置将生成树的工作模式配置为RSTP模式	<a href="#">1.3.1</a>
	配置根桥和备份根桥	可选	<a href="#">1.3.3</a>
	配置设备的优先级	可选	<a href="#">1.3.4</a>
	配置交换网络的网络直径	可选	<a href="#">1.3.6</a>
	配置生成树的时间参数	可选	<a href="#">1.3.7</a>
	配置超时时间因子	可选	<a href="#">1.3.8</a>
	配置端口发送BPDU的速率	可选	<a href="#">1.3.9</a>
	配置端口为边缘端口	可选	<a href="#">1.3.10</a>
	配置端口的链路类型	可选	<a href="#">1.3.13</a>
	打开端口状态变化信息显示开关	可选	<a href="#">1.3.15</a>
	开启生成树协议	必选	<a href="#">1.3.16</a>
配置叶子节点	配置生成树的工作模式	必选 通过本配置将生成树的工作模式配置为RSTP模式	<a href="#">1.3.1</a>
	配置设备的优先级	可选	<a href="#">1.3.4</a>
	配置超时时间因子	可选	<a href="#">1.3.8</a>
	配置端口发送BPDU的速率	可选	<a href="#">1.3.9</a>
	配置端口为边缘端口	可选	<a href="#">1.3.10</a>
	配置端口的路径开销	可选	<a href="#">1.3.11</a>
	配置端口的优先级	可选	<a href="#">1.3.12</a>
	配置端口的链路类型	可选	<a href="#">1.3.13</a>
	打开端口状态变化信息显示开关	可选	<a href="#">1.3.15</a>
开启生成树协议	必选	<a href="#">1.3.16</a>	
执行mCheck操作	可选	<a href="#">1.3.17</a>	
配置TC Snooping功能	可选	<a href="#">1.3.21</a>	
配置生成树保护功能	可选	<a href="#">1.3.22</a>	
配置被BPDU保护功能关闭的端口不再自动恢复	可选	<a href="#">1.3.24</a>	
配置生成树的网管功能	可选	<a href="#">1.3.25</a>	

## 1.2.3 PVST配置任务简介

表1-12 PVST 配置任务简介

配置任务		说明	详细配置
配置根桥	配置生成树的工作模式	必选 生成树缺省工作在MSTP模式下，通过本配置将其工作模式配置为PVST模式	<a href="#">1.3.1</a>
	配置根桥和备份根桥	可选	<a href="#">1.3.3</a>
	配置设备的优先级	可选	<a href="#">1.3.4</a>
	配置交换网络的网络直径	可选	<a href="#">1.3.6</a>
	配置生成树的时间参数	可选	<a href="#">1.3.7</a>
	配置超时时间因子	可选	<a href="#">1.3.8</a>
	配置端口发送BPDU的速率	可选	<a href="#">1.3.9</a>
	配置端口为边缘端口	可选	<a href="#">1.3.10</a>
	配置端口的链路类型	可选	<a href="#">1.3.13</a>
	打开端口状态变化信息显示开关	可选	<a href="#">1.3.15</a>
开启生成树协议	必选	<a href="#">1.3.16</a>	
配置叶子节点	配置生成树的工作模式	必选 生成树缺省工作在MSTP模式下，通过本配置将其工作模式配置为PVST模式	<a href="#">1.3.1</a>
	配置设备的优先级	可选	<a href="#">1.3.4</a>
	配置超时时间因子	可选	<a href="#">1.3.8</a>
	配置端口发送BPDU的速率	可选	<a href="#">1.3.9</a>
	配置端口为边缘端口	可选	<a href="#">1.3.10</a>
	配置端口的路径开销	可选	<a href="#">1.3.11</a>
	配置端口的优先级	可选	<a href="#">1.3.12</a>
	配置端口的链路类型	可选	<a href="#">1.3.13</a>
打开端口状态变化信息显示开关	可选	<a href="#">1.3.15</a>	
开启生成树协议	必选	<a href="#">1.3.16</a>	
执行mCheck操作	可选	<a href="#">1.3.17</a>	
关闭PVST的PVID不一致保护功能	可选	<a href="#">1.3.18</a>	
配置生成树保护功能	可选	<a href="#">1.3.22</a>	
配置在PVST模式下设备检测或接收到TC报文时打印日志信息	可选	<a href="#">1.3.23</a>	
配置被BPDU保护功能关闭的端口不再自动恢复	可选	<a href="#">1.3.24</a>	
配置生成树的网管功能	可选	<a href="#">1.3.25</a>	

## 1.2.4 MSTP配置任务简介

表1-13 MSTP 配置任务简介

	配置任务	说明	详细配置
配置根桥	配置生成树的工作模式	必选 通过本配置将生成树的工作模式配置为MSTP模式	<a href="#">1.3.1</a>
	配置MST域	必选	<a href="#">1.3.2</a>
	配置根桥和备份根桥	可选	<a href="#">1.3.3</a>
	配置设备的优先级	可选	<a href="#">1.3.4</a>
	配置MST域的最大跳数	可选	<a href="#">1.3.5</a>
	配置交换网络的网络直径	可选	<a href="#">1.3.6</a>
	配置生成树的时间参数	可选	<a href="#">1.3.7</a>
	配置超时时间因子	可选	<a href="#">1.3.8</a>
	配置端口发送BPDU的速率	可选	<a href="#">1.3.9</a>
	配置端口为边缘端口	可选	<a href="#">1.3.10</a>
	配置端口的链路类型	可选	<a href="#">1.3.13</a>
	配置端口收发的MSTP报文格式	可选	<a href="#">1.3.14</a>
	打开端口状态变化信息显示开关	可选	<a href="#">1.3.15</a>
	开启生成树协议	必选	<a href="#">1.3.16</a>
配置叶子节点	配置生成树的工作模式	必选 通过本配置将生成树的工作模式配置为MSTP模式	<a href="#">1.3.1</a>
	配置MST域	必选	<a href="#">1.3.2</a>
	配置设备的优先级	可选	<a href="#">1.3.4</a>
	配置超时时间因子	可选	<a href="#">1.3.8</a>
	配置端口发送BPDU的速率	可选	<a href="#">1.3.9</a>
	配置端口为边缘端口	可选	<a href="#">1.3.10</a>
	配置端口的路径开销	可选	<a href="#">1.3.11</a>
	配置端口的优先级	可选	<a href="#">1.3.12</a>
	配置端口的链路类型	可选	<a href="#">1.3.13</a>
	配置端口收发的MSTP报文格式	可选	<a href="#">1.3.14</a>
	打开端口状态变化信息显示开关	可选	<a href="#">1.3.15</a>
开启生成树协议	必选	<a href="#">1.3.16</a>	

配置任务	说明	详细配置
执行mCheck操作	可选	<a href="#">1.3.17</a>
配置摘要侦听功能	可选	<a href="#">1.3.19</a>
配置No Agreement Check功能	可选	<a href="#">1.3.20</a>
配置TC Snooping功能	可选	<a href="#">1.3.21</a>
配置生成树保护功能	可选	<a href="#">1.3.22</a>
配置被BPDU保护功能关闭的端口不再自动恢复	可选	<a href="#">1.3.24</a>
配置生成树的网管功能	可选	<a href="#">1.3.25</a>

## 1.3 配置生成树

### 1.3.1 配置生成树的工作模式

生成树的工作模式有以下几种：

- **STP 模式：**设备的所有端口都将向外发送 STP BPDU。如果端口的对端设备只支持 STP，可选择此模式。
- **RSTP 模式：**设备的所有端口都向外发送 RSTP BPDU。当端口收到对端设备发来的 STP BPDU 时，会自动迁移到 STP 模式；如果收到的是 MSTP BPDU，则不会进行迁移。
- **PVST 模式：**设备的所有端口都向外发送 PVST BPDU，每个 VLAN 对应一棵生成树。进行 PVST 组网时，若网络中所有设备的生成树维护量（开启生成树协议的 VLAN 数×开启生成树协议的端口数）达到一定数量，会导致 CPU 负荷过重，不能正常处理报文，引起网络震荡。本系列设备支持的生成树维护量为 128。
- **MSTP 模式：**设备的所有端口都向外发送 MSTP BPDU。当端口收到对端设备发来的 STP BPDU 时，会自动迁移到 STP 模式；如果收到的是 RSTP BPDU，则不会进行迁移。

MSTP 模式兼容 RSTP 模式，RSTP 模式兼容 STP 模式，PVST 模式与其他模式的兼容性如下：

- 对于 Access 端口：PVST 模式在任意 VLAN 中都能与其他模式互相兼容。
- 对于 Trunk 端口或 Hybrid 端口：PVST 模式仅在缺省 VLAN 中能与其他模式互相兼容。

表1-14 配置生成树的工作模式

操作	命令	说明
进入系统视图	<b>system-view</b>	-
配置生成树的工作模式	<b>stp mode { mstp   pvst   rstp   stp }</b>	缺省情况下，生成树的工作模式为MSTP模式

### 1.3.2 配置MST域

两台或多台开启了生成树协议的设备若要属于同一个 MST 域，必须同时满足以下两个条件：第一是选择因子（取值为 0，不可配）、域名、修订级别和 VLAN 映射表的配置都相同；第二是这些设备之间的链路相通。

在配置 MST 域的相关参数（特别是 VLAN 映射表）时，会引发生成树的重新计算，从而引起网络拓扑的振荡。为了减少网络振荡，新配置的 MST 域参数并不会马上生效，而是在使用 **active region-configuration** 命令激活，或使用命令 **stp global enable** 全局开启生成树协议后才会生效。需要注意的是，在 STP/RSTP/PVST 模式下，MST 域的相关配置不会生效。

表1-15 配置 MST 域

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入MST域视图	<b>stp region-configuration</b>	-
配置MST域的域名	<b>region-name name</b>	缺省情况下，MST域的域名为设备的MAC地址
配置VLAN映射表	<b>instance instance-id vlan vlan-id-list</b>	二者选其一 缺省情况下，所有VLAN都映射到 CIST（即MSTI 0）上
	<b>vlan-mapping modulo modulo</b>	
配置MSTP的修订级别	<b>revision-level level</b>	缺省情况下，MSTP的修订级别为0
（可选）显示MST域的预配置信息	<b>check region-configuration</b>	-
激活MST域的配置	<b>active region-configuration</b>	-

### 1.3.3 配置根桥和备份根桥

可以通过计算来自动确定生成树的根桥，用户也可以手工将设备配置为指定生成树的根桥或备份根桥：

- 设备在各生成树中的角色互相独立，在作为一棵生成树的根桥或备份根桥的同时，也可以作为其他生成树的根桥或备份根桥；但在同一棵生成树中，一台设备不能既作为根桥，又作为备份根桥。
- 在一棵生成树中，生效的根桥只有一个；当两台或两台以上的设备被指定为同一棵生成树的根桥时，系统将选择 MAC 地址最小的设备作为根桥。
- 可以在每棵生成树中指定一个或多个备份根桥。当根桥出现故障或被关机时，如果配置了一个备份根桥，则该备份根桥可以取代根桥成为指定生成树的根桥；如果配置了多个备份根桥，则 MAC 地址最小的备份根桥将成为指定生成树的根桥。但此时若配置了新的根桥，则备份根桥将不会成为根桥。
- 用户可以为每棵生成树指定一个根桥，而无需关心设备的优先级配置。当设备一旦被配置为根桥或者备份根桥之后，便不能再修改该设备的优先级。也可以通过配置设备的优先级为 0 来实现将当前设备指定为根桥的目的。有关设备优先级的配置，请参见“[1.3.4 配置设备的优先级](#)”。

## 1. 配置根桥

请在欲配置为根桥的设备上进行如下配置。

表1-16 配置根桥

操作	命令	说明
进入系统视图	<b>system-view</b>	-
配置设备为根桥	STP/RSTP模式: <b>stp root primary</b>	缺省情况下，设备不是根桥
	PVST模式: <b>stp vlan <i>vlan-id-list</i> root primary</b>	
	MSTP模式: <b>stp [ instance <i>instance-list</i> ] root primary</b>	

## 2. 配置备份根桥

请在欲配置为备份根桥的设备上进行如下配置。

表1-17 配置备份根桥

操作	命令	说明
进入系统视图	<b>system-view</b>	-
配置设备为备份根桥	STP/RSTP模式: <b>stp root secondary</b>	缺省情况下，设备不是备份根桥
	PVST模式: <b>stp vlan <i>vlan-id-list</i> root secondary</b>	
	MSTP模式: <b>stp [ instance <i>instance-list</i> ] root secondary</b>	

### 1.3.4 配置设备的优先级

设备的优先级参与生成树计算，其大小决定了该设备是否能够被选作生成树的根桥。数值越小表示优先级越高，通过配置较小的优先级，可以达到指定某台设备成为生成树根桥的目的。可以在不同的生成树中为设备配置不同的优先级。如果设备的优先级相同，则 MAC 地址最小的设备将被选择为根。当指定设备为根桥或者备份根桥之后，不允许再修改该设备的优先级。

表1-18 配置设备的优先级

操作	命令	说明
进入系统视图	<b>system-view</b>	-
配置设备的优先级	STP/RSTP模式: <b>stp priority <i>priority</i></b>	缺省情况下，设备的优先级为32768

操作	命令	说明
	PVST模式: <b>stp vlan <i>vlan-id-list</i> priority <i>priority</i></b>	
	MSTP模式: <b>stp [ <i>instance instance-list</i> ] priority <i>priority</i></b>	

### 1.3.5 配置MST域的最大跳数

MST 域的最大跳数限制了 MST 域的规模,在域根上配置的最大跳数将作为该 MST 域的最大跳数。从 MST 域内的生成树的根桥开始,域内的 BPDU 每经过一台设备的转发,跳数就被减 1;设备将丢弃跳数为 0 的 BPDU,以使处于最大跳数外的设备无法参与生成树的计算,从而限制了 MST 域的规模。

本配置只需在根桥设备上,非根桥设备将采用根桥设备的配置值。

用户可以根据设计的 MST 域内拓扑的层数来配置 MST 域的最大跳数, MST 域的最大跳数要大于 MST 域内拓扑的最大层数。

表1-19 配置 MST 域的最大跳数

操作	命令	说明
进入系统视图	<b>system-view</b>	-
配置MST域的最大跳数	<b>stp max-hops <i>hops</i></b>	缺省情况下, MST域的最大跳数为20

### 1.3.6 配置交换网络的网络直径

交换网络中任意两台终端设备都通过特定路径彼此相连,这些路径由一系列的设备构成。网络直径就是指对于交换网络中的任意两台网络边缘设备,其中一台经过根桥到达另一台所经过的最大设备数。网络直径越大,说明网络的规模越大。

在 STP/RSTP/MSTP 模式下,每个 MST 域将被视为一台设备,且网络直径配置只对 CIST 有效(即只能在总根上生效),而对 MSTI 无效。在 PVST 模式下,网络直径的配置只能在指定 VLAN 的根桥上生效。

通过本配置,可以根据网络直径调整设备的 Hello Time、Forward Delay 和 Max Age 三个时间参数到合适的值。

表1-20 配置交换网络的网络直径

操作	命令	说明
进入系统视图	<b>system-view</b>	-
配置交换网络的网络直径	STP/RSTP/MSTP模式: <b>stp bridge-diameter <i>diameter</i></b>  PVST模式: <b>stp vlan <i>vlan-id-list</i> bridge-diameter <i>diameter</i></b>	缺省情况下, 交换网络的网络直径为7

### 1.3.7 配置生成树的时间参数

在生成树的计算过程中，用到了以下三个时间参数：

- (1) **Forward Delay**：用于确定状态迁移的延迟时间。为了防止产生临时环路，生成树协议在端口由 **Discarding** 状态向 **Forwarding** 状态迁移的过程中设置了 **Learning** 状态作为过渡，并规定状态迁移需要等待 **Forward Delay** 时间，以保持与远端的设备状态切换同步。
- (2) **Hello Time**：用于检测链路是否存在故障。生成树协议每隔 **Hello Time** 时间会发送 BPDUs，以确认链路是否存在故障。如果设备在超时时间（超时时间 = 超时时间因子 × 3 × **Hello Time**）内没有收到 BPDUs，则会由于消息超时而重新计算生成树。
- (3) **Max Age**：用于确定 BPDUs 是否超时。在 MSTP 的 CIST 上，设备根据 **Max Age** 时间来确定端口收到的 BPDUs 是否超时。如果端口收到的 BPDUs 超时，则需要对该 MSTI 重新计算。**Max Age** 时间对 MSTP 的 MSTI 无效。

为保证网络拓扑的快速收敛，需要配置合适的时间参数。上述三个时间参数之间应满足以下关系，否则会引起网络的频繁震荡：

- $2 \times (\text{Forward Delay} - 1 \text{ 秒}) \geq \text{Max Age}$
- $\text{Max Age} \geq 2 \times (\text{Hello Time} + 1 \text{ 秒})$

配置生成树时间参数时，需要注意：

- **Forward Delay** 的长短与交换网络的网络直径有关。一般来说，网络直径越大，**Forward Delay** 就应该越长。如果 **Forward Delay** 过短，可能引入临时的冗余路径；如果 **Forward Delay** 过长，网络可能较长时间不能恢复连通。建议用户采用自动计算值。
- 合适的 **Hello Time** 可以保证设备能够及时发现网络中的链路故障，又不会占用过多的网络资源。如果 **Hello Time** 过长，在链路发生丢包时，设备会误以为链路出现了故障，从而引发设备重新计算生成树；如果 **Hello Time** 过短，设备将频繁发送重复的 BPDUs，增加了设备的负担，浪费了网络资源。建议用户采用自动计算值。
- 如果 **Max Age** 过短，设备会频繁地计算生成树，而且有可能将网络拥塞误认成链路故障；如果 **Max Age** 过长，设备很可能不能及时发现链路故障，不能及时重新计算生成树，从而降低网络的自适应能力。建议用户采用自动计算值。

通常情况下，不建议通过手工配置直接调整上述三个时间参数。由于这三个时间参数的取值与网络规模有关，生成树协议会自动根据网络直径计算出这三个时间参数的最优值，因此在网络拓扑变化时，建议在设备上通过执行 **stp bridge-diameter** 命令调整网络直径，使设备自动调整这三个时间参数的值。当网络直径取缺省值时，这三个时间参数也分别取其各自的缺省值。

本配置只需在根桥设备上进行，整个交换网络中的所有设备都将采用根桥设备的配置值。

表1-21 配置生成树的时间参数

操作	命令	说明
进入系统视图	<b>system-view</b>	-
配置Forward Delay时间参数	STP/RSTP/MSTP模式： <b>stp timer forward-delay time</b>	缺省情况下，Forward Delay为15秒

操作	命令	说明
	PVST模式: <b>stp vlan <i>vlan-id-list</i> timer forward-delay <i>time</i></b>	
配置Hello Time时间参数	STP/RSTP/MSTP模式: <b>stp timer hello <i>time</i></b>	缺省情况下, Hello Time为2秒
	PVST模式: <b>stp vlan <i>vlan-id-list</i> timer hello <i>time</i></b>	
配置Max Age时间参数	STP/RSTP/MSTP模式: <b>stp timer max-age <i>time</i></b>	缺省情况下, Max Age为20秒
	PVST模式: <b>stp vlan <i>vlan-id-list</i> timer max-age <i>time</i></b>	

### 1.3.8 配置超时时间因子

超时时间因子用来确定设备的超时时间： $\text{超时时间} = \text{超时时间因子} \times 3 \times \text{Hello Time}$ 。

当网络拓扑结构稳定后，非根桥设备会每隔 Hello Time 时间向周围相连设备转发根桥发出的 BPDU 以确认链路是否存在故障。通常如果设备在 9 倍的 Hello Time 时间内没有收到上游设备发来的 BPDU，就会认为上游设备已经故障，从而重新进行生成树的计算。

对于以下情况，建议将设备的超时时间因子配置为 5~7。

- 有时本端设备在较长时间内收不到对端设备发来的 BPDU，可能是由于对端设备的繁忙导致的（例如，对端设备配置了大量二层接口时），在这种情况下一般不应重新进行生成树的计算，需要延长本端设备的超时时间。
- 稳定的网络中，可以通过延长超时时间来减少网络资源的浪费。

表1-22 配置超时时间因子

操作	命令	说明
进入系统视图	<b>system-view</b>	-
配置设备的超时时间因子	<b>stp timer-factor <i>factor</i></b>	缺省情况下, 设备的超时时间因子为3

### 1.3.9 配置端口发送BPDU的速率

每 Hello Time 时间内端口能够发送的 BPDU 的最大数目 = 端口发送 BPDU 的速率 + Hello Time 时间值。端口发送 BPDU 的速率与端口的物理状态和网络结构有关，用户可以根据实际的网络情况对其进行配置。

端口发送 BPDU 的速率越高，每个 Hello Time 内可发送的 BPDU 数量就越多，占用的系统资源也越多。适当配置发送速率一方面可以限制端口发送 BPDU 的速度，另一方面还可以防止在网络拓扑动荡时，生成树协议占用过多的带宽资源。建议用户采用缺省配置。

表1-23 配置端口的最大发送速率

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二层以太网接口视图/二层聚合接口视图	<b>interface interface-type interface-number</b>	-
配置端口的发送BPDU的速率	<b>stp transmit-limit limit</b>	缺省情况下，端口发送BPDU的速率为10

### 1.3.10 配置端口为边缘端口

当端口直接与用户终端相连，而没有连接到其他设备或共享网段上，则该端口被认为是边缘端口。网络拓扑变化时，边缘端口不会产生临时环路。

由于设备无法知道端口是否直接与终端相连，所以需要用户手工将端口配置为边缘端口。如果用户将某个端口配置为边缘端口，那么当该端口由阻塞状态向转发状态迁移时，这个端口可以实现快速迁移，而无需等待延迟时间。

对于直接与终端相连的端口，请将该端口设置为边缘端口，同时开启 BPDU 保护功能。这样既能够使该端口快速迁移到转发状态，也可以保证网络的安全。

配置端口为边缘端口时，需要注意：

- 在同一个端口上，不允许同时配置边缘端口和环路保护功能。
- 在端口没有开启 BPDU 保护的情况下，如果被设置为边缘端口的端口上收到来自其他端口的 BPDU，则该端口会重新变为非边缘端口。此时，只有重启端口才能将该端口恢复为边缘端口。

表1-24 配置端口为边缘端口

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二层以太网接口视图/二层聚合接口视图	<b>interface interface-type interface-number</b>	-
配置当前端口为边缘端口	<b>stp edged-port</b>	缺省情况下，端口为非边缘端口

### 1.3.11 配置端口的路径开销

路径开销（Path Cost）是与端口相连的链路速率相关的参数。在支持生成树协议的设备上，端口在不同的 MSTI 中可以拥有不同的路径开销。设置合适的路径开销可以使不同 VLAN 的流量沿不同的物理链路转发，从而实现按 VLAN 负载分担的功能。

设备可以自动计算端口的缺省路径开销，用户也可以直接配置端口的路径开销。

#### 1. 配置缺省路径开销的计算标准

缺省路径开销的计算标准有以下三种，用户可以通过本配置来改变设备自动计算端口的缺省路径开销时所采用的计算标准：

- **dot1d-1998**：表示按照 IEEE 802.1D-1998 标准来计算缺省路径开销。

- **dot1t**: 表示按照 IEEE 802.1t 标准来计算缺省路径开销。
- **legacy**: 表示按照私有标准来计算缺省路径开销。

需要注意的是，改变缺省路径开销的计算标准，将使端口的路径开销值恢复为缺省值。

表1-25 配置缺省路径开销的计算标准

操作	命令	说明
进入系统视图	<b>system-view</b>	-
配置缺省路径开销的计算标准	<b>stp pathcost-standard { dot1d-1998   dot1t   legacy }</b>	缺省情况下，缺省路径开销的计算标准为 legacy

链路速率与路径开销值的对应关系如 [表 1-26](#) 所示。

表1-26 链路速率与端口路径开销值的对应关系表

链路速率	端口类型	端口的路径开销值		
		IEEE 802.1D-1998	IEEE 802.1t	私有标准
0	-	65535	200,000,000	200,000
10Mbps	单个端口	100	2,000,000	2,000
	聚合接口（含两个选中端口）		1,000,000	1,800
	聚合接口（含三个选中端口）		666,666	1,600
	聚合接口（含四个选中端口）		500,000	1,400
100Mbps	单个端口	19	200,000	200
	聚合接口（含两个选中端口）		100,000	180
	聚合接口（含三个选中端口）		66,666	160
	聚合接口（含四个选中端口）		50,000	140
1000Mbps	单个端口	4	20,000	20
	聚合接口（含两个选中端口）		10,000	18
	聚合接口（含三个选中端口）		6,666	16
	聚合接口（含四个选中端口）		5,000	14
10Gbps	单个端口	2	2,000	2
	聚合接口（含两个选中端口）		1,000	1
	聚合接口（含三个选中端口）		666	1
	聚合接口（含四个选中端口）		500	1
20Gbps	单个端口	1	1,000	1
	聚合接口（含两个选中端口）		500	1
	聚合接口（含三个选中端口）		333	1
	聚合接口（含四个选中端口）		250	1

链路速率	端口类型	端口的路径开销值		
		IEEE 802.1D-1998	IEEE 802.1t	私有标准
40Gbps	单个端口	1	500	1
	聚合接口（含两个选中端口）		250	1
	聚合接口（含三个选中端口）		166	1
	聚合接口（含四个选中端口）		125	1
100Gbps	单个端口	1	200	1
	聚合接口（含两个选中端口）		100	1
	聚合接口（含三个选中端口）		66	1
	聚合接口（含四个选中端口）		50	1

### 说明

- 在计算聚合接口的路径开销时，IEEE 802.1D-1998 标准不考虑聚合接口所对应聚合组内选中端口的数量；而 IEEE 802.1t 标准则对此予以考虑，其计算公式为：端口的路径开销 =  $200000000 \div \text{链路速率}$ （单位为 100Kbps），其中链路速率为聚合接口所对应聚合组内选中端口的速率之和。
- 当端口的链路速率大于 10Gbps、且缺省路径开销的计算标准为 IEEE 802.1D-1998 或私有标准时，单个端口和聚合接口的路径开销值都会取所选标准规定的最小值，这将影响转发路径选择的合理性。在这种情况下，建议将缺省路径开销的计算标准配置为 IEEE 802.1t，或手工配置端口的路径开销（请参见“[1.3.11 2. 配置端口的路径开销](#)”）。

## 2. 配置端口的路径开销

需要注意的是，当端口的路径开销值改变时，系统将重新计算端口的角色并进行状态迁移。

表1-27 配置端口的路径开销

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二层以太网接口视图/二层聚合接口视图	<b>interface interface-type interface-number</b>	-
配置端口的路径开销	STP/RSTP模式： <b>stp cost cost-value</b>	缺省情况下，自动按照相应的标准计算各生成树上的路径开销
	PVST模式： <b>stp vlan vlan-id-list cost cost-value</b>	
	MSTP模式： <b>stp [ instance instance-list ] cost cost-value</b>	

### 3. 配置举例

# 在 MSTP 模式下，配置按照 IEEE 802.1D-1998 标准来计算缺省路径开销，并配置端口 GigabitEthernet1/0/3 在 MSTI 2 上的路径开销值为 200。

```
<Sysname> system-view
[Sysname] stp pathcost-standard dot1d-1998
Cost of every port will be reset and automatically re-calculated after you change the current
pathcost standard. Continue?[Y/N]:y
Cost of every port has been re-calculated.
[Sysname] interface gigabitethernet 1/0/3
[Sysname-GigabitEthernet1/0/3] stp instance 2 cost 200
```

# 在 PVST 模式下，配置设备按照 IEEE 802.1D-1998 标准来计算缺省路径开销，并配置端口 GigabitEthernet1/0/3 在 PVST VLAN 20~30 上的路径开销为 2000。

```
<Sysname> system-view
[Sysname] stp pathcost-standard dot1d-1998
Cost of every port will be reset and automatically re-calculated after you change the current
pathcost standard. Continue?[Y/N]:y
Cost of every port has been re-calculated
[Sysname] interface gigabitethernet 1/0/3
[Sysname-GigabitEthernet1/0/3] stp vlan 20 to 30 cost 2000
```

#### 1.3.12 配置端口的优先级

端口优先级是确定该端口是否会被选为根端口的重要依据，同等条件下优先级高的端口将被选为根端口。在支持生成树协议的设备上，端口可以在不同的生成树中拥有不同的优先级，同一端口可以在不同的生成树中担任不同的角色，从而使不同 VLAN 的数据沿不同的物理路径传播，实现按 VLAN 进行负载分担的功能。用户可以根据组网的实际需要来设置端口的优先级。

需要注意的是，当端口的优先级改变时，系统将重新计算端口的角色并进行状态迁移。

表1-28 配置端口的优先级

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二层以太网接口视图/二层聚合接口视图	<b>interface</b> <i>interface-type interface-number</i>	-
配置端口的优先级	STP/RSTP模式： <b>stp port priority</b> <i>priority</i>	缺省情况下，端口的优先级为128
	PVST模式： <b>stp vlan</b> <i>vlan-id-list port priority</i> <i>priority</i>	
	MSTP模式： <b>stp [ instance</b> <i>instance-list</i> <b>port priority</b> <i>priority</i>	

### 1.3.13 配置端口的链路类型

点对点链路是两台设备之间直接连接的链路。与点对点链路相连的两个端口如果为根端口或者指定端口，则端口可以通过传送同步报文（Proposal 报文和 Agreement 报文）快速迁移到转发状态，减少了不必要的转发延迟时间。

如果某端口是二层聚合接口或其工作在全双工模式下，则可以将该端口配置为与点对点链路相连。通常建议使用缺省配置，由系统进行自动检测。

在 PVST 或 MSTP 模式下，如果某端口被配置为与点对点链路（或非点对点链路）相连，那么该配置对该端口所属的所有 VLAN 或 MSTI 都有效。

需要注意的是，如果某端口被配置为与点对点链路相连，但与该端口实际相连的物理链路不是点对点链路，则有可能引入临时回路。

表1-29 配置端口的链路类型

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二层以太网接口视图/二层聚合接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
配置端口的链路类型	<b>stp point-to-point { auto   force-false   force-true }</b>	缺省情况下，端口的链路类型为 <b>auto</b> ，即由系统自动检测与本端口相连的链路是否为点对点链路

### 1.3.14 配置端口收发的MSTP报文格式

端口可以收发的 MSTP 报文格式有两种：

- **dot1s**: 符合 802.1s 协议的标准格式；
- **legacy**: 与非标准格式兼容的格式。

端口默认配置为自动识别方式（**auto**），即可以自动识别这两种格式的 MSTP 报文，并根据识别结果确定发送报文的格式，从而实现与对端设备的互通。

用户也可以通过配置改变端口发送的 MSTP 报文格式，使端口只发送与所配格式相符的 MSTP 报文，实现与对端只识别特定格式报文的设备互通。

当端口处于 **auto** 模式时，默认发送 802.1s 标准的报文。在此模式下，为避免因收到不同格式的 MSTP 报文而导致端口发送的报文格式频繁变化，端口一旦收到私有格式报文就将一直以该格式发送报文。若想使该端口恢复发送 802.1s 标准的报文，可对其依次执行关闭/开启操作。

需要注意的是，如果当前配置的 MSTI 大于 48，端口将只发送 802.1s 标准的 MSTP 报文。

表1-30 配置端口收发的 MSTP 报文格式

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二层以太网接口视图/二层聚合接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-

操作	命令	说明
配置端口收发的MSTP报文格式	<b>stp compliance { auto   dot1s   legacy }</b>	缺省情况下，端口会自动识别收到的MSTP报文格式并根据识别结果确定发送的报文格式

### 1.3.15 打开端口状态变化信息显示开关

在开启了生成树协议的大型网络中，用户可以通过打开端口状态变化信息显示开关，使系统输出端口状态变化的相关信息，方便用户对端口状态进行实时监控。

表1-31 打开端口状态变化信息显示开关

操作	命令	说明
进入系统视图	<b>system-view</b>	-
打开端口状态变化信息显示开关	STP/RSTP模式： <b>stp port-log instance 0</b>	缺省情况下，端口状态变化信息显示开关处于关闭状态
	PVST模式： <b>stp port-log vlan <i>vlan-id-list</i></b>	
	MSTP模式： <b>stp port-log { all   instance <i>instance-list</i> }</b>	

### 1.3.16 开启生成树协议

只有开启了生成树协议，生成树的其他配置才会生效。在 STP/RSTP/MSTP 模式下，必须保证全局和端口上的生成树协议均处于开启状态；在 PVST 模式下，必须保证全局、VLAN 和端口上的生成树协议均处于开启状态。

需要注意的是，可以通过 **undo stp enable** 命令关闭指定端口的生成树协议，使其不参与生成树计算，以节省设备的 CPU 资源。但必须保证指定的端口关闭生成树协议后，网络中不能出现环路。

表1-32 开启生成树协议（STP/RSTP/MSTP 模式）

操作	命令	说明
进入系统视图	<b>system-view</b>	-
全局开启生成树协议	<b>stp global enable</b>	空配置启动时，使用软件功能缺省值，全局生成树协议处于关闭状态 出厂配置启动时，使用软件功能出厂值，全局生成树协议处于开启状态 关于空配置启动和出厂配置启动的详细介绍，请参见“基础配置指导”中的“配置文件管理”
进入二层以太网接口视图 /二层聚合接口视图	<b>interface <i>interface-type</i> <i>interface-number</i></b>	-
在端口上开启生成树协议	<b>stp enable</b>	缺省情况下，所有端口上的生成树协议均处于开启状态

表1-33 开启生成树协议（PVST 模式）

操作	命令	说明
进入系统视图	<b>system-view</b>	-
全局开启生成树协议	<b>stp global enable</b>	空配置启动时，使用软件功能缺省值，全局生成树协议处于关闭状态 出厂配置启动时，使用软件功能出厂值，全局生成树协议处于开启状态 关于空配置启动和出厂配置启动的详细介绍，请参见“基础配置指导”中的“配置文件管理”
在VLAN中开启生成树协议	<b>stp vlan <i>vlan-id-list</i> enable</b>	缺省情况下，生成树协议在VLAN中处于开启状态
进入二层以太网接口视图 /二层聚合接口视图	<b>interface <i>interface-type</i> <i>interface-number</i></b>	-
在端口上开启生成树协议	<b>stp enable</b>	缺省情况下，所有端口上的生成树协议均处于开启状态

### 1.3.17 执行mCheck操作

生成树的工作模式有 STP 模式、RSTP 模式、PVST 模式和 MSTP 模式四种。在运行 RSTP、PVST 或 MSTP 的设备上，若某端口连接着运行 STP 协议的设备，该端口收到 STP 报文后会自动迁移到 STP 模式；但当对端运行 STP 协议的设备关机或撤走，而该端口又无法感知的情况下，该端口将无法自动迁移回原有模式，此时需要通过执行 mCheck 操作将其手工迁移回原有模式。

当运行 STP 的设备 A、未开启生成树协议的设备 B 和运行 RSTP/PVST/MSTP 的设备 C 三者顺次相连时，设备 B 将透传 STP 报文，设备 C 上连接设备 B 的端口将迁移到 STP 模式。在设备 B 上开启生成树协议后，若想使设备 B 与设备 C 之间运行 RSTP/PVST/MSTP 协议，除了要在设备 B 上配置生成树的工作模式为 RSTP/PVST/MSTP 外，还要在设备 B 与设备 C 相连的端口上都执行 mCheck 操作。

可以在全局或在端口上执行 mCheck 操作。

执行 mCheck 操作时，需要注意：

- 只有当生成树的工作模式为 RSTP 模式、PVST 模式或 MSTP 模式时执行 mCheck 操作才有效。

#### 1. 全局执行mCheck操作

表1-34 全局执行 mCheck 操作

操作	命令	说明
进入系统视图	<b>system-view</b>	-
全局执行mCheck操作	<b>stp global mcheck</b>	-

## 2. 在端口上执行mCheck操作

表1-35 在端口上执行 mCheck 操作

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二层以太网接口视图/二层聚合接口视图	<b>interface interface-type interface-number</b>	-
在端口上执行mCheck操作	<b>stp mcheck</b>	-

### 1.3.18 关闭PVST的PVID不一致保护功能

在当链路相连的两端 PVID 不一致时，PVST 的计算可能出现错误，为了防止这样的错误，系统默认会开启 PVID 不一致保护功能，即做 PVID 不一致的检查。若端口 PVID 不一致保护功能触发后，端口在 PVID 不一致的 VLAN 中，会变为阻塞状态。

在某些特定的组网场景中，比如网络中的接入层设备采用同样的配置，其接口 PVID 一致，而网络管理员在汇聚层设备的下行口（即连接接入层设备的接口）上做了不同的 PVID 配置，该配置与接入层设备的上行口（即连接汇聚层设备的接口）的 PVID 配置不一致时，有可能引起生成树的阻塞，为避免这种情况的发生，保持流量的转发，可以关闭 PVID 不一致保护功能。

关闭 PVST 的 PVID 不一致保护功能后，如果链路两端端口 PVID 不一致，为了避免生成树的计算错误，需要注意：

- 除了缺省 VLAN，本端所在设备不能创建对端 PVID 对应的 VLAN，同样，对端也不能创建本端 PVID 对应的 VLAN。
- 本端端口的链路类型是 Hybrid 时，建议本端所在设备不创建以 Untagged 方式允许通过的 VLAN，同样，对端也不创建本端 Untagged 方式允许通过的 VLAN。
- 建议链路对端设备也关闭 PVST 的 PVID 不一致保护功能。
- 本配置在 PVST 工作模式下才能生效。

表1-36 关闭 PVST 的 PVID 不一致保护功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
关闭PVST的PVID不一致保护功能	<b>stp ignore-pvid-inconsistency</b>	缺省情况下，PVST的PVID不一致保护功能处于开启状态

### 1.3.19 配置摘要侦听功能

#### 1. 功能简介

根据 IEEE 802.1s 规定，只有在 MST 域配置（包括域名、修订级别和 VLAN 映射关系）完全一致的情况下，相连的设备才被认为是在同一个域内。当设备开启了生成树协议以后，设备之间通过识别 BPDU 数据报文内的配置 ID 来判断相连的设备是否与自己处于相同的 MST 域内；配置 ID 包含

域名、修订级别、配置摘要等内容，其中配置摘要长 16 字节，是由 HMAC-MD5 算法将 VLAN 与 MSTI 的映射关系加密计算而成。

在网络中，由于一些厂商的设备在对生成树协议的实现上存在差异，即用加密算法计算配置摘要时采用私有的密钥，从而导致即使 MST 域配置相同，不同厂商的设备之间也不能实现在 MST 域内的互通。

通过在我方设备与对生成树协议的实现存在差异的第三方厂商设备相连的端口上开启摘要侦听功能，可以实现我方设备与这些厂商设备在 MST 域内的完全互通。

## 2. 配置限制和指导

- 摘要侦听功能在端口生效后，由于不再通过配置摘要的比较计算来判断是否在同一个域内，因此需要保证互连设备的域配置中 VLAN 与 MSTI 映射关系的配置相同。
- 全局开启摘要侦听功能后，如果要修改 VLAN 与 MSTI 间的映射关系，或执行 **undo stp region-configuration** 命令取消当前域配置，均可能因与邻接设备的 VLAN 和 MSTI 映射关系不一致而导致环路或流量中断，因此请谨慎操作。
- 只有当全局和端口上都开启了摘要侦听功能后，该功能才能生效。开启摘要侦听功能时，建议先在所有与第三方厂商设备相连的端口上开启该功能，再全局开启该功能，以一次性让所有端口的配置生效，从而减少对网络的冲击。
- 请不要在 MST 域的边界端口上开启摘要侦听功能，否则可能会导致环路。
- 建议配置完摘要侦听功能后再开启生成树协议。在网络稳定的情况下不要进行摘要侦听功能的配置，以免造成临时的流量中断。

## 3. 配置准备

我方设备与第三方厂商设备相连，网络配置正确，生成树协议正常运行。

## 4. 配置步骤

只有当我方设备与对生成树协议的实现存在差异的第三方厂商设备（即采用私有密钥来计算配置摘要）互连时，才有必要配置本功能。

表1-37 配置摘要侦听功能

配置步骤	命令	说明
进入系统视图	<b>system-view</b>	-
进入二层以太网接口视图/二层聚合接口视图	<b>interface interface-type interface-number</b>	-
在端口上开启摘要侦听功能	<b>stp config-digest-snooping</b>	缺省情况下，端口上的摘要侦听功能处于关闭状态
退回系统视图	<b>quit</b>	-
全局开启摘要侦听功能	<b>stp global config-digest-snooping</b>	缺省情况下，摘要侦听功能处于全局关闭状态

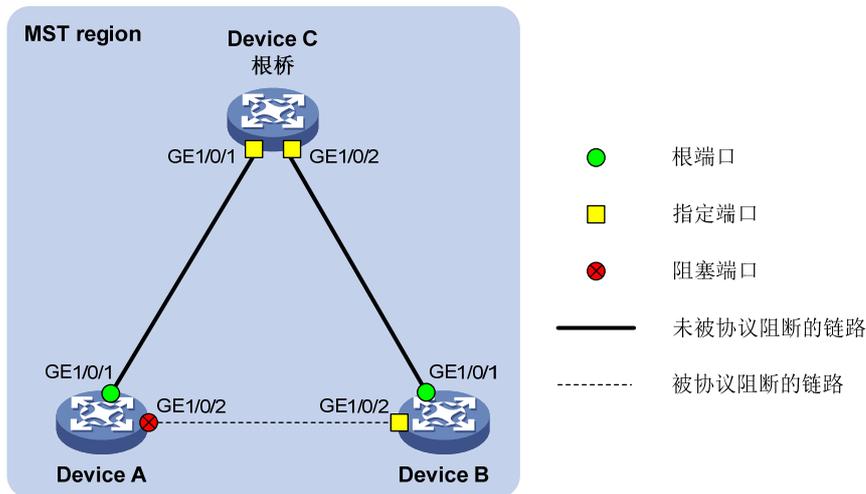
## 5. 配置举例

### (1) 组网需求

- Device A 和 Device B 分别与对生成树协议的实现存在差异的第三方厂商设备 Device C 相连并配置在同一域内。
- 分别在 Device A 和 Device B 各自与 Device C 相连的端口上开启摘要侦听功能，实现 Device A、Device B 和 Device C 在 MST 域内的互通。

(2) 组网图

图1-16 摘要侦听功能配置组网图



(3) 配置步骤

# 在 Device A 的端口 GigabitEthernet1/0/1 上开启摘要侦听功能，并全局开启摘要侦听功能。

```
<DeviceA> system-view
[DeviceA] interface gigabitethernet 1/0/1
[DeviceA-GigabitEthernet1/0/1] stp config-digest-snooping
[DeviceA-GigabitEthernet1/0/1] quit
[DeviceA] stp global config-digest-snooping
```

# 在 Device B 的端口 GigabitEthernet1/0/1 上开启摘要侦听功能，并全局开启摘要侦听功能。

```
<DeviceB> system-view
[DeviceB] interface gigabitethernet 1/0/1
[DeviceB-GigabitEthernet1/0/1] stp config-digest-snooping
[DeviceB-GigabitEthernet1/0/1] quit
[DeviceB] stp global config-digest-snooping
```

### 1.3.20 配置No Agreement Check功能

#### 1. 功能简介

RSTP 和 MSTP 的指定端口快速迁移机制使用两种协议报文：

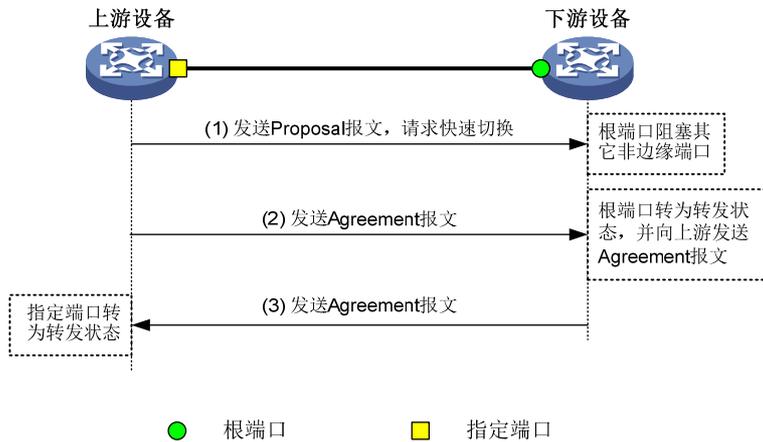
- Proposal 报文：指定端口请求快速迁移的报文。
- Agreement 报文：同意对端进行快速迁移的报文。

RSTP 和 MSTP 均要求上游设备的指定端口在接收到下游设备的 Agreement 报文后才能进行快速迁移。不同之处如下：

- 对于 MSTP，上游设备先向下游设备发送 Agreement 报文，而下游设备的根端口只有在收到了上游设备的 Agreement 报文后才会向上游设备回应 Agreement 报文。
- 对于 RSTP，下游设备无需等待上游设备发送 Agreement 报文就可向上游设备发送 Agreement 报文。

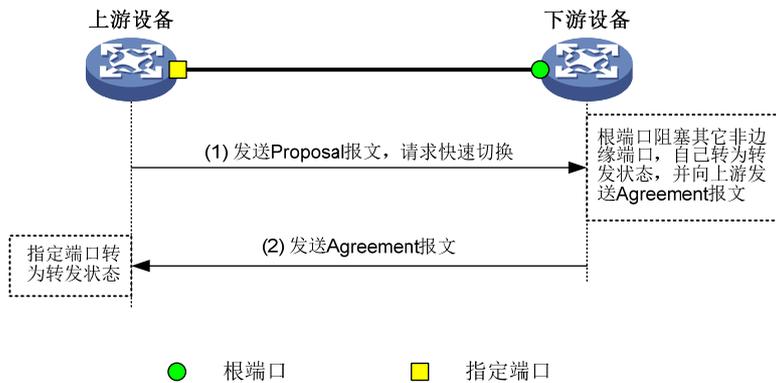
如 图 1-17 所示，是MSTP的指定端口快速迁移机制。

图1-17 MSTP 指定端口快速迁移机制



如 图 1-18 所示，是RSTP的指定端口快速迁移机制。

图1-18 RSTP 指定端口快速迁移机制



当我方设备与作为上游设备且与对生成树协议的实现存在差异的第三方厂商设备互联时，二者在快速迁移的配合上可能会存在一定的限制。例如：上游设备指定端口的状态迁移实现机制与 RSTP 类似；而下游设备运行 MSTP 并且不工作在 RSTP 模式时，由于下游设备的根端口接收不到上游设备的 Agreement 报文，它不会向上游设备发 Agreement 报文，所以上游设备的指定端口无法实现状态的快速迁移，只能在 2 倍的 Forward Delay 延时后变成转发状态。

通过在我方设备与对生成树协议的实现存在私有性差异的上游第三方厂商设备相连的端口上开启 No Agreement Check 功能，可避免这种情况的出现，使得上游的第三方厂商设备的指定端口能够进行状态的快速迁移。

## 2. 配置准备

- 设备与作为上游设备且支持生成树协议的第三方厂商设备互连，并且端口之间为点对点链路。
- 为我方设备与第三方厂商设备配置相同的域名、域配置修订级别和 VLAN 与 MSTI 的映射关系，以确保它们在同一域内。

## 3. 配置功能

请在设备的根端口上进行如下配置，且本功能只有在根端口上配置才会生效。

表1-38 配置 No Agreement Check 功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二层以太网接口视图/二层聚合接口视图	<b>interface interface-type interface-number</b>	-
开启No Agreement Check功能	<b>stp no-agreement-check</b>	缺省情况下，No Agreement Check功能处于关闭状态

## 4. 配置举例

### (1) 组网需求

- Device A 与对生成树协议的实现存在差异的第三方厂商设备 Device B 互连并配置在同一域内。
- Device B 作为域根，Device A 作为下游设备。

### (2) 组网图

图1-19 No Agreement Check 功能配置组网图



### (3) 配置步骤

# 在 Device A 的端口 GigabitEthernet1/0/1 上开启 No Agreement Check 功能。

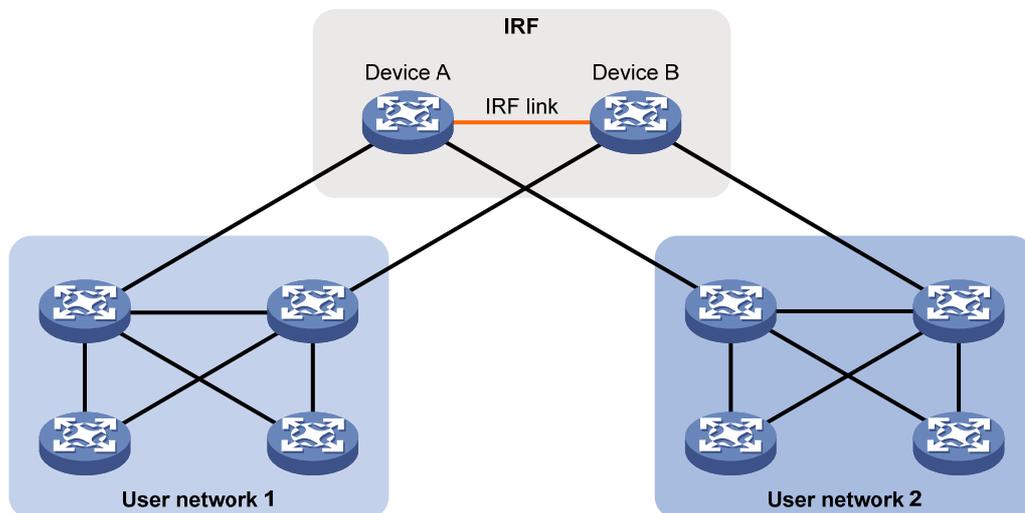
```
<DeviceA> system-view
[DeviceA] interface gigabitethernet 1/0/1
[DeviceA-GigabitEthernet1/0/1] stp no-agreement-check
```

## 1.3.21 配置TC Snooping功能

TC Snooping功能的典型应用环境如 图 1-20 所示。在该组网中，由Device A和Device B组成的IRF设备未开启生成树协议，而用户网络 1 和用户网络 2 中的所有设备均开启了生成树协议。用户网络 1 和用户网络 2 均通过双上行链路与IRF设备相连以提高链路可靠性，IRF设备可以透明传输每个用户网络中的BPDU。

在该组网中，当用户网络的拓扑结构发生改变时，由于 IRF 设备对 BPDU 进行了透明传输而不参与生成树计算，因而其本身可能需经过较长时间才能重新学到正确的 MAC 地址表项和 ARP 表项，在此期间可能导致网络中断。

图1-20 TC Snooping 功能典型应用组网图



为了避免这种情况，可以通过在 IRF 设备上开启 TC Snooping 功能，使其在收到 TC-BPDU（网络拓扑发生变化的通知报文）后，主动更新接收该报文的端口所属的 VLAN 所对应的 MAC 地址表和 ARP 表，从而保证业务流量的正常转发。有关 MAC 地址表和 ARP 表的详细介绍，请分别参见“二层技术-以太网交换配置指导”中的“MAC 地址表”和“三层技术-IP 业务配置指导”中的“ARP”。

配置 TC Snooping 功能时，需要注意：

- TC Snooping 功能与生成树协议互斥，因此在开启 TC Snooping 功能之前必须全局关闭生成树协议。
- TC Snooping 功能不支持 PVST 格式的 TC-BPDU，因此在 PVST 模式下不支持该功能。

表1-39 配置 TC Snooping 功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
全局关闭生成树协议	<b>undo stp global enable</b>	空配置启动时，使用软件功能缺省值，全局生成树协议处于关闭状态 出厂配置启动时，使用软件功能出厂值，全局生成树协议处于开启状态 关于空配置启动和出厂配置启动的详细介绍，请参见“基础配置指导”中的“配置文件管理”
开启TC Snooping功能	<b>stp tc-snooping</b>	缺省情况下，TC Snooping功能处于关闭状态

### 1.3.22 配置生成树保护功能

生成树保护功能包括以下几种：

- BPDU 保护功能
- 根保护功能
- 环路保护功能
- 端口角色限制功能
- TC-BPDU 传播限制功能
- 防 TC-BPDU 攻击保护功能
- BPDU 拦截功能
- MSTP 的 PVST 报文保护功能

### 1. 配置BPDU保护功能

对于接入层设备，接入端口一般直接与用户终端（如 PC）或文件服务器相连，此时接入端口被设置为边缘端口以实现这些端口的快速迁移；当这些端口接收到 BPDU 时系统会自动将这些端口设置为非边缘端口，重新计算生成树，引起网络拓扑结构的变化。这些端口正常情况下应该不会收到 STP 的 BPDU。如果有人伪造 BPDU 恶意攻击设备，就会引起网络震荡。

生成树协议提供了 BPDU 保护功能来防止这种攻击：设备上开启了 BPDU 保护功能后，如果边缘端口收到了 BPDU，系统就将这些端口关闭，同时通知网管这些端口已被生成树协议关闭。被关闭的端口在经过一定时间间隔之后将被重新激活，这个时间间隔可通过 **shutdown-interval** 命令配置。有关该命令的详细介绍，请参见“基础配置命令参考”中的“设备管理”。

BPDU 保护功能支持在全局或指定端口配置。

需要注意的是，BPDU 保护功能对开启了环回测试功能的端口无效。有关环回测试功能的相关介绍，请参见“接口管理配置指导”中的“以太网接口”。

#### (1) 配置全局的 BPDU 保护功能

开启全局的 BPDU 保护功能后，所有边缘端口的 BPDU 保护功能处于开启状态。

表1-40 配置全局的 BPDU 保护功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
开启全局的BPDU保护功能	<b>stp bpdu-protection</b>	缺省情况下，全局的BPDU保护功能处于关闭状态

#### (2) 配置端口的 BPDU 保护功能

端口开启 BPDU 保护功能后，不论全局的 BPDU 保护功能是否开启，端口的 BPDU 保护功能均处于开启状态。端口关闭 BPDU 保护功能后，不论全局的 BPDU 保护功能是否开启，端口的 BPDU 保护功能均处于关闭状态。

表1-41 配置端口的 BPDU 保护功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二层以太网接口视图 /二层聚合接口视图	<b>interface interface-type interface-number</b>	-

操作	命令	说明
配置端口的BPDU保护功能	<b>stp port bpdu-protection</b> { enable   disable }	缺省情况下, 如果全局的BPDU保护功能处于开启状态, 则边缘端口的BPDU保护功能处于开启状态; 如果全局的BPDU保护功能处于关闭状态, 则边缘端口的BPDU保护功能处于关闭状态 请在直连用户终端的端口上配置, 勿在连接其他设备或共享网段的端口上配置

## 2. 配置根保护功能

生成树的根桥和备份根桥应该处于同一个域内, 特别是对于 CIST 的根桥和备份根桥, 网络设计时一般会把 CIST 的根桥和备份根桥放在一个高带宽的核心域内。但是, 由于维护人员的错误配置或网络中的恶意攻击, 网络中的合法根桥有可能会收到优先级更高的 BPDU, 这样当前合法根桥会失去根桥的地位, 引起网络拓扑结构的错误变动。这种不合法的变动, 会导致原来应该通过高速链路的流量被牵引到低速链路上, 导致网络拥塞。

为了防止这种情况发生, 生成树协议提供了根保护功能: 对于开启了根保护功能的端口, 其在所有 MSTI 上的端口角色只能为指定端口。一旦该端口收到某 MSTI 优先级更高的 BPDU, 立即将该 MSTI 端口设置为侦听状态, 不再转发报文(相当于将此端口相连的链路断开)。当在 2 倍的 Forward Delay 时间内没有收到更优的 BPDU 时, 端口会恢复原来的正常状态。

请在设备的指定端口上进行如下配置。

需要注意的是, 在同一个端口上, 不允许同时配置根保护功能和环路保护功能。

表1-42 配置根保护功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二层以太网接口视图/二层聚合接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
开启端口的根保护功能	<b>stp root-protection</b>	缺省情况下, 端口上的根保护功能处于关闭状态

## 3. 配置环路保护功能

依靠不断接收上游设备发送的 BPDU, 设备可以维持根端口和其他阻塞端口的状态。但是由于链路拥塞或者单向链路故障, 这些端口会收不到上游设备的 BPDU, 此时下游设备会重新选择端口角色, 收不到 BPDU 的下游设备端口会转变为指定端口, 而阻塞端口会迁移到转发状态, 从而交换网络中会产生环路。环路保护功能会抑制这种环路的产生。

在开启了环路保护功能的端口上, 其所有 MSTI 的初始状态均为 Discarding 状态: 如果该端口收到了 BPDU, 这些 MSTI 可以进行正常的状态迁移; 否则, 这些 MSTI 将一直处于 Discarding 状态以避免环路的产生。

请在设备的根端口和替换端口上进行如下配置。

配置环路保护功能时, 需要注意:

- 请不要在与用户终端相连的端口上开启环路保护功能, 否则该端口会因收不到 BPDU 而导致其所有 MSTI 将一直处于 Discarding 状态。

- 在同一个端口上，不允许同时配置边缘端口和环路保护功能，或者同时配置根保护功能和环路保护功能。

表1-43 配置环路保护功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二层以太网接口视图/二层聚合接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
开启端口的环路保护功能	<b>stp loop-protection</b>	缺省情况下，端口的环路保护功能处于关闭状态

#### 4. 配置端口角色限制功能

用户接入网络中设备桥 ID 的变化会引起核心网络生成树拓扑的改变。为了避免这种情况，可以在端口上开启端口角色限制功能，此后当该端口收到最优根消息时将不再当选为根端口，而是成为替换端口。

请在与用户接入网络相连的端口上进行如下配置。

需要注意的是，开启端口角色限制功能后可能影响生成树拓扑的连通性，请慎重配置。

表1-44 配置端口角色限制功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二层以太网接口视图/二层聚合接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
开启端口角色限制功能	<b>stp role-restriction</b>	缺省情况下，端口角色限制功能处于关闭状态

#### 5. 配置TC-BPDU传播限制功能

用户接入网络的拓扑改变会引起核心网络的转发地址更新，当用户接入网络的拓扑因某种原因而不稳定时，就会对核心网络形成冲击。为了避免这种情况，可以在端口上开启 TC-BPDU 传播限制功能，此后当该端口收到 TC-BPDU 时，不会再向其他端口传播。

请在与用户接入网络相连的端口上进行如下配置。

需要注意的是，开启 TC-BPDU 传播限制功能后，当拓扑改变时原有转发地址表项可能无法更新，请慎重配置。

表1-45 配置 TC-BPDU 传播限制功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二层以太网接口视图/二层聚合接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
开启TC-BPDU传播限制功能	<b>stp tc-restriction</b>	缺省情况下，TC-BPDU传播限制功能处于关闭状态

## 6. 配置防TC-BPDU攻击保护功能

设备在收到 TC-BPDU 后，会执行转发地址表项的刷新操作。在有人伪造 TC-BPDU 恶意攻击设备时，设备短时间内会收到很多的 TC-BPDU，频繁的刷新操作给设备带来很大负担，给网络的稳定带来很大隐患。而通过在设备上开启防 TC-BPDU 攻击保护功能，就可以避免转发地址表项的频繁刷新。

当开启了防 TC-BPDU 攻击保护功能后，如果设备在单位时间（固定为十秒）内收到 TC-BPDU 的次数大于 **stp tc-protection threshold** 命令所指定的最高次数（假设为 N 次），那么该设备在这段时间之内将只进行 N 次刷新转发地址表项的操作，而对于超出 N 次的那些 TC-BPDU，设备会在这段时间过后再统一进行一次地址表项刷新的操作，这样就可以避免频繁地刷新转发地址表项。

建议不要关闭防 TC-BPDU 攻击保护功能。

表1-46 配置防 TC-BPDU 攻击保护功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
开启防TC-BPDU攻击保护功能	<b>stp tc-protection</b>	缺省情况下，防TC-BPDU攻击保护功能处于开启状态
（可选）配置在单位时间（固定为十秒）内，设备收到TC-BPDU后立即刷新转发地址表项的最高次数	<b>stp tc-protection threshold number</b>	缺省情况下，在单位时间（固定为十秒）内，设备收到TC-BPDU后立即刷新转发地址表项的最高次数为6

## 7. 配置BPDU拦截功能

在开启了生成树协议的网络中，由于设备收到 BPDU 后会进行 STP 计算并向其他设备转发，因此恶意用户可借此进行 BPDU 攻击：通过不停地发送 BPDU，使网络中的所有设备都不停地进行 STP 计算，从而导致设备的 CPU 占用率过高或 BPDU 的协议状态错误等问题。

为了避免这种情况，用户可以在端口上配置 BPDU 拦截功能。开启了该功能的端口将不再接收任何 BPDU，从而能够防止设备遭受 BPDU 攻击，保证 STP 计算的正确性。

表1-47 配置 BPDU 拦截功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二层以太网接口视图	<b>interface interface-type interface-number</b>	-
开启端口的BPDU拦截功能	<b>bpdu-drop any</b>	缺省情况下，端口的BPDU拦截功能处于关闭状态

## 8. 配置MSTP的PVST报文保护功能

对于开启 MSTP 的设备，并不识别 PVST 报文，所以开启 MSTP 的设备会将 PVST 报文当做数据报文转发。在另一个并不相干的网络中，开启 PVST 的设备收到该报文，处理后可能导致该网络的拓扑计算出现错误。

对于这个问题，可以通过配置 MSTP 的 PVST 报文保护功能来解决。在 MSTP 模式下，设备上开启了 PVST 报文保护功能后，如果端口收到了 PVST 报文，系统就将这些端口关闭。

需要注意的是，本配置在 MSTP 工作模式下才能生效。

表1-48 配置 MSTP 的 PVST 报文保护功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
开启MSTP的PVST报文保护功能	<b>stp pvst-bpdu-protection</b>	缺省情况下，MSTP的PVST报文保护功能处于关闭状态

### 1.3.23 配置在PVST模式下设备检测或接收到TC报文时打印日志信息

配置在 PVST 模式下，设备检测或接收到 TC 报文时打印日志信息。

表1-49 配置在 PVST 模式下设备检测或接收到 TC 报文时打印日志信息

操作	命令	说明
进入系统视图	<b>system-view</b>	-
配置在PVST模式下设备检测或接收到TC报文时打印日志信息	<b>stp log enable tc</b>	缺省情况下，PVST模式下设备检测或接收到TC报文后，不打印日志信息

### 1.3.24 配置被BPDU保护功能关闭的端口不再自动恢复

设备上使能了 BPDU 保护功能后，如果边缘端口收到了 BPDU，系统就将这些端口关闭，同时通知网管这些端口已被生成树协议关闭。被关闭的端口在经过一定时间间隔之后将被重新激活，这个时间间隔可通过 **shutdown-interval** 命令配置。有关该命令的详细介绍，请参见“基础配置命令参考”中的“设备管理”。

- 配置 **stp port shutdown permanent** 命令后，端口被 BPDU 保护功能关闭，再执行 **undo stp port shutdown permanent** 命令，端口不会 UP，端口保持关闭状态，需要需要执行 **undo shutdown** 命令才能恢复。
- 端口被 BPDU 保护功能关闭，再配置 **stp port shutdown permanent** 命令，此时端口会在端口状态检测定时器超时后 UP，当再次被 BPDU 保护功能关闭时，端口才不会恢复，保持关闭状态。

表1-50 配置被 BPDU 保护功能关闭的端口不再自动恢复

操作	命令	说明
进入系统视图	<b>system-view</b>	-
配置被BPDU保护功能关闭的端口不再自动恢复	<b>stp port shutdown permanent</b>	缺省情况下，被BPDU保护功能关闭的端口会自动恢复

### 1.3.25 配置生成树的网管功能

开启生成树的告警功能之后，生成树会生成告警信息，用于报告本模块的重要事件。生成的告警信息将发送至 SNMP 模块，通过配置 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1-51 开启生成树的告警功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
开启生成树的告警功能	<b>snmp-agent trap enable stp [ new-root   tc ]</b>	缺省情况下，生成树的new-root告警功能处于关闭状态。在MSTP模式下，生成树的TC告警功能在MSTI 0中处于开启状态，在其他MSTI中处于关闭状态；在PVST模式下，生成树的TC告警功能在所有VLAN中处于关闭状态

## 1.4 生成树显示和维护

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

在用户视图下执行 **reset** 命令可以清除生成树的统计信息。

表1-52 生成树显示和维护

操作	命令
显示被生成树保护功能阻塞的端口信息	<b>display stp abnormal-port</b>
显示端口上的BPDU统计信息	<b>display stp bpdu-statistics [ interface interface-type interface-number [ instance instance-list ] ]</b>
显示被生成树保护功能down掉的端口信息	<b>display stp down-port</b>
显示生成树端口角色计算的历史信息	<b>display stp [ instance instance-list   vlan vlan-id-list ] history [ slot slot-number ]</b>
显示生成树所有端口收发的TC或TCN报文数	<b>display stp [ instance instance-list   vlan vlan-id-list ] tc [ slot slot-number ]</b>
显示生成树的状态和统计信息	<b>display stp [ instance instance-list   vlan vlan-id-list ] [ interface interface-list   slot slot-number ] [ brief ]</b>
显示生效的MST域配置信息	<b>display stp region-configuration</b>
显示所有生成树的根桥信息	<b>display stp root</b>
清除生成树的统计信息	<b>reset stp [ interface interface-list ]</b>

## 1.5 生成树典型配置举例

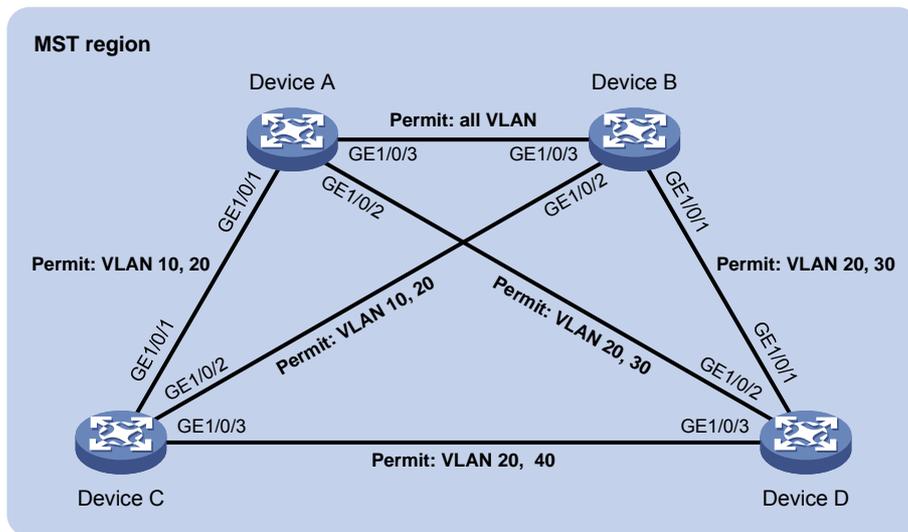
### 1.5.1 MSTP典型配置举例

#### 1. 组网需求

- 网络中所有设备都属于同一个 MST 域。Device A 和 Device B 为汇聚层设备，Device C 和 Device D 为接入层设备。
- 通过配置 MSTP，使不同 VLAN 的报文按照不同的 MSTI 转发：VLAN 10 的报文沿 MSTI 1 转发，VLAN 30 沿 MSTI 3 转发，VLAN 40 沿 MSTI 4 转发，VLAN 20 沿 MSTI 0 转发。
- 由于 VLAN 10 和 VLAN 30 在汇聚层设备终结、VLAN 40 在接入层设备终结，因此配置 MSTI 1 和 MSTI 3 的根桥分别为 Device A 和 Device B，MSTI 4 的根桥为 Device C。

#### 2. 组网图

图1-21 MSTP 典型配置组网图



#### 3. 配置步骤

##### (1) 配置 VLAN 和端口

请按照 [图 1-21](#) 在 Device A 和 Device B 上分别创建 VLAN 10、20 和 30，在 Device C 上创建 VLAN 10、20 和 40，在 Device D 上创建 VLAN 20、30 和 40；将各设备的各端口配置为 Trunk 端口并允许相应的 VLAN 通过，具体配置过程略。

##### (2) 配置 Device A

# 配置 MST 域的域名为 example，将 VLAN 10、30、40 分别映射到 MSTI 1、3、4 上，并配置 MSTP 的修订级别为 0。

```
<DeviceA> system-view
[DeviceA] stp region-configuration
[DeviceA-mst-region] region-name example
[DeviceA-mst-region] instance 1 vlan 10
[DeviceA-mst-region] instance 3 vlan 30
[DeviceA-mst-region] instance 4 vlan 40
```

```
[DeviceA-mst-region] revision-level 0
# 激活 MST 域的配置。
[DeviceA-mst-region] active region-configuration
[DeviceA-mst-region] quit
# 配置本设备为 MSTI 1 的根桥。
[DeviceA] stp instance 1 root primary
# 全局开启生成树协议。
[DeviceA] stp global enable
```

### (3) 配置 Device B

# 配置 MST 域的域名为 **example**，将 VLAN 10、30、40 分别映射到 MSTI 1、3、4 上，并配置 MSTP 的修订级别为 0。

```
<DeviceB> system-view
[DeviceB] stp region-configuration
[DeviceB-mst-region] region-name example
[DeviceB-mst-region] instance 1 vlan 10
[DeviceB-mst-region] instance 3 vlan 30
[DeviceB-mst-region] instance 4 vlan 40
[DeviceB-mst-region] revision-level 0
```

# 激活 MST 域的配置。

```
[DeviceB-mst-region] active region-configuration
[DeviceB-mst-region] quit
# 配置本设备为 MSTI 3 的根桥。
[DeviceB] stp instance 3 root primary
```

# 全局开启生成树协议。

```
[DeviceB] stp global enable
```

### (4) 配置 Device C

# 配置 MST 域的域名为 **example**，将 VLAN 10、30、40 分别映射到 MSTI 1、3、4 上，并配置 MSTP 的修订级别为 0。

```
<DeviceC> system-view
[DeviceC] stp region-configuration
[DeviceC-mst-region] region-name example
[DeviceC-mst-region] instance 1 vlan 10
[DeviceC-mst-region] instance 3 vlan 30
[DeviceC-mst-region] instance 4 vlan 40
[DeviceC-mst-region] revision-level 0
```

# 激活 MST 域的配置。

```
[DeviceC-mst-region] active region-configuration
[DeviceC-mst-region] quit
# 配置本设备为 MSTI 4 的根桥。
[DeviceC] stp instance 4 root primary
```

# 全局开启生成树协议。

```
[DeviceC] stp global enable
```

### (5) 配置 Device D

# 配置 MST 域的域名为 **example**，将 VLAN 10、30、40 分别映射到 MSTI 1、3、4 上，并配置 MSTP 的修订级别为 0。

```
<DeviceD> system-view
[DeviceD] stp region-configuration
[DeviceD-mst-region] region-name example
[DeviceD-mst-region] instance 1 vlan 10
[DeviceD-mst-region] instance 3 vlan 30
[DeviceD-mst-region] instance 4 vlan 40
[DeviceD-mst-region] revision-level 0
```

# 激活 MST 域的配置。

```
[DeviceD-mst-region] active region-configuration
[DeviceD-mst-region] quit
```

# 全局开启生成树协议。

```
[DeviceD] stp global enable
```

#### 4. 验证配置

---



说明

在本例中，假定 Device B 的根桥 ID 最小，因此该设备将在 MSTI 0 中被选举为根桥。

---

当网络拓扑稳定后，通过使用 **display stp brief** 命令可以查看各设备上生成树的简要信息。例如：

# 查看 Device A 上生成树的简要信息。

```
[DeviceA] display stp brief
```

MST ID	Port	Role	STP State	Protection
0	GigabitEthernet1/0/1	ALTE	DISCARDING	NONE
0	GigabitEthernet1/0/2	DESI	FORWARDING	NONE
0	GigabitEthernet1/0/3	ROOT	FORWARDING	NONE
1	GigabitEthernet1/0/1	DESI	FORWARDING	NONE
1	GigabitEthernet1/0/3	DESI	FORWARDING	NONE
3	GigabitEthernet1/0/2	DESI	FORWARDING	NONE
3	GigabitEthernet1/0/3	ROOT	FORWARDING	NONE

# 查看 Device B 上生成树的简要信息。

```
[DeviceB] display stp brief
```

MST ID	Port	Role	STP State	Protection
0	GigabitEthernet1/0/1	DESI	FORWARDING	NONE
0	GigabitEthernet1/0/2	DESI	FORWARDING	NONE
0	GigabitEthernet1/0/3	DESI	FORWARDING	NONE
1	GigabitEthernet1/0/2	DESI	FORWARDING	NONE
1	GigabitEthernet1/0/3	ROOT	FORWARDING	NONE
3	GigabitEthernet1/0/1	DESI	FORWARDING	NONE
3	GigabitEthernet1/0/3	DESI	FORWARDING	NONE

# 查看 Device C 上生成树的简要信息。

```
[DeviceC] display stp brief
```

MST ID	Port	Role	STP State	Protection
--------	------	------	-----------	------------

```

0          GigabitEthernet1/0/1      DESI FORWARDING NONE
0          GigabitEthernet1/0/2      ROOT FORWARDING NONE
0          GigabitEthernet1/0/3      DESI FORWARDING NONE
1          GigabitEthernet1/0/1      ROOT FORWARDING NONE
1          GigabitEthernet1/0/2      ALTE DISCARDING NONE
4          GigabitEthernet1/0/3      DESI FORWARDING NONE

```

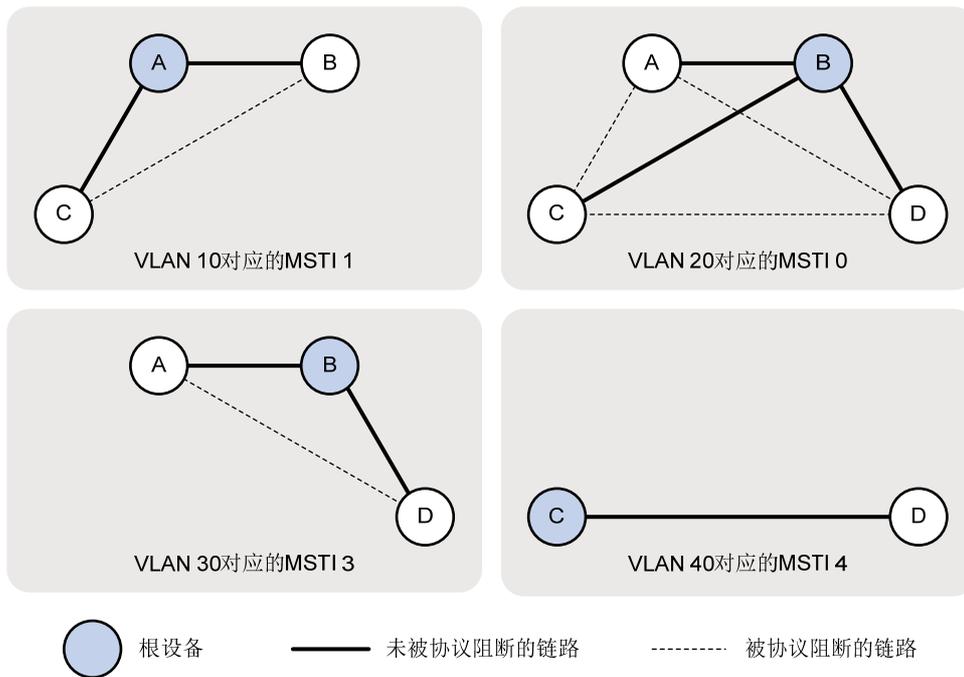
# 查看 Device D 上生成树的简要信息。

```
[DeviceD] display stp brief
```

MST ID	Port	Role	STP State	Protection
0	GigabitEthernet1/0/1	ROOT	FORWARDING	NONE
0	GigabitEthernet1/0/2	ALTE	DISCARDING	NONE
0	GigabitEthernet1/0/3	ALTE	DISCARDING	NONE
3	GigabitEthernet1/0/1	ROOT	FORWARDING	NONE
3	GigabitEthernet1/0/2	ALTE	DISCARDING	NONE
4	GigabitEthernet1/0/3	ROOT	FORWARDING	NONE

根据上述显示信息，可以绘出各VLAN所对应MSTI的拓扑，如 [图 1-22](#) 所示。

图1-22 各 VLAN 所对应 MSTI 的拓扑图



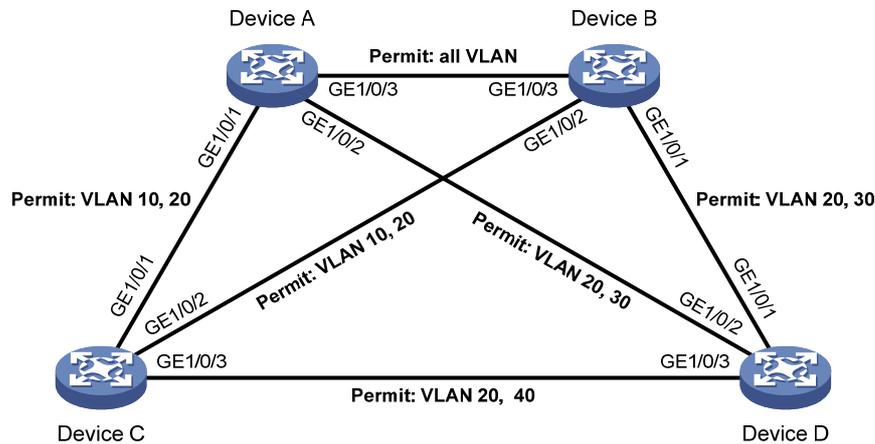
## 1.5.2 PVST典型配置举例

### 1. 组网需求

- Device A 和 Device B 为汇聚层设备，Device C 和 Device D 为接入层设备。
- 通过配置 PVST，使 VLAN 10、20、30 和 40 中的报文分别按照其各自 VLAN 所对应的生成树转发。
- 由于 VLAN 10、20 和 30 在汇聚层设备终结、VLAN 40 在接入层设备终结，因此配置 VLAN 10 和 20 的根桥为 Device A，VLAN 30 的根桥为 Device B，VLAN 40 的根桥为 Device C。

## 2. 组网图

图1-23 PVST 典型配置组网图



## 3. 配置步骤

### (1) 配置 VLAN 和端口

请按照 [图 1-23](#) 在 Device A 和 Device B 上分别创建 VLAN 10、20 和 30，在 Device C 上创建 VLAN 10、20 和 40，在 Device D 上创建 VLAN 20、30 和 40；将各设备的各端口配置为 Trunk 端口并允许相应的 VLAN 通过，具体配置过程略。

### (2) 配置 Device A

# 配置生成树的工作模式为 PVST 模式。

```
<DeviceA> system-view
[DeviceA] stp mode pvst
```

# 配置本设备为 VLAN 10 和 VLAN 20 的根桥。

```
[DeviceA] stp vlan 10 20 root primary
```

# 全局开启生成树协议，并开启 VLAN 10、20 和 30 中的生成树协议。

```
[DeviceA] stp global enable
[DeviceA] stp vlan 10 20 30 enable
```

### (3) 配置 Device B

# 配置生成树的工作模式为 PVST 模式。

```
<DeviceB> system-view
[DeviceB] stp mode pvst
```

# 配置本设备为 VLAN 30 的根桥。

```
[DeviceB] stp vlan 30 root primary
```

# 全局开启生成树协议，并开启 VLAN 10、20 和 30 中的生成树协议。

```
[DeviceB] stp global enable
[DeviceB] stp vlan 10 20 30 enable
```

### (4) 配置 Device C

# 配置生成树的工作模式为 PVST 模式。

```
<DeviceC> system-view
[DeviceC] stp mode pvst
```

# 配置本设备为生成树 VLAN 40 的根桥。

```
[DeviceC] stp vlan 40 root primary
```

# 全局开启生成树协议，并开启 VLAN 10、20 和 40 中的生成树协议。

```
[DeviceC] stp global enable
```

```
[DeviceC] stp vlan 10 20 40 enable
```

#### (5) 配置 Device D

# 配置生成树的工作模式为 PVST 模式。

```
<DeviceD> system-view
```

```
[DeviceD] stp mode pvst
```

# 全局开启生成树协议，并开启 VLAN 20、30 和 40 中的生成树协议。

```
[DeviceD] stp global enable
```

```
[DeviceD] stp vlan 20 30 40 enable
```

### 4. 验证配置

当网络拓扑稳定后，通过使用 **display stp brief** 命令可以查看各设备上生成树的简要信息。例如：

# 查看 Device A 上生成树的简要信息。

```
[DeviceA] display stp brief
```

VLAN ID	Port	Role	STP State	Protection
10	GigabitEthernet1/0/1	DESI	FORWARDING	NONE
10	GigabitEthernet1/0/3	DESI	FORWARDING	NONE
20	GigabitEthernet1/0/1	DESI	FORWARDING	NONE
20	GigabitEthernet1/0/2	DESI	FORWARDING	NONE
20	GigabitEthernet1/0/3	DESI	FORWARDING	NONE
30	GigabitEthernet1/0/2	DESI	FORWARDING	NONE
30	GigabitEthernet1/0/3	ROOT	FORWARDING	NONE

# 查看 Device B 上生成树的简要信息。

```
[DeviceB] display stp brief
```

VLAN ID	Port	Role	STP State	Protection
10	GigabitEthernet1/0/2	DESI	FORWARDING	NONE
10	GigabitEthernet1/0/3	ROOT	FORWARDING	NONE
20	GigabitEthernet1/0/1	DESI	FORWARDING	NONE
20	GigabitEthernet1/0/2	DESI	FORWARDING	NONE
20	GigabitEthernet1/0/3	ROOT	FORWARDING	NONE
30	GigabitEthernet1/0/1	DESI	FORWARDING	NONE
30	GigabitEthernet1/0/3	DESI	FORWARDING	NONE

# 查看 Device C 上生成树的简要信息。

```
[DeviceC] display stp brief
```

VLAN ID	Port	Role	STP State	Protection
10	GigabitEthernet1/0/1	ROOT	FORWARDING	NONE
10	GigabitEthernet1/0/2	ALTE	DISCARDING	NONE
20	GigabitEthernet1/0/1	ROOT	FORWARDING	NONE
20	GigabitEthernet1/0/2	ALTE	DISCARDING	NONE
20	GigabitEthernet1/0/3	DESI	FORWARDING	NONE
40	GigabitEthernet1/0/3	DESI	FORWARDING	NONE

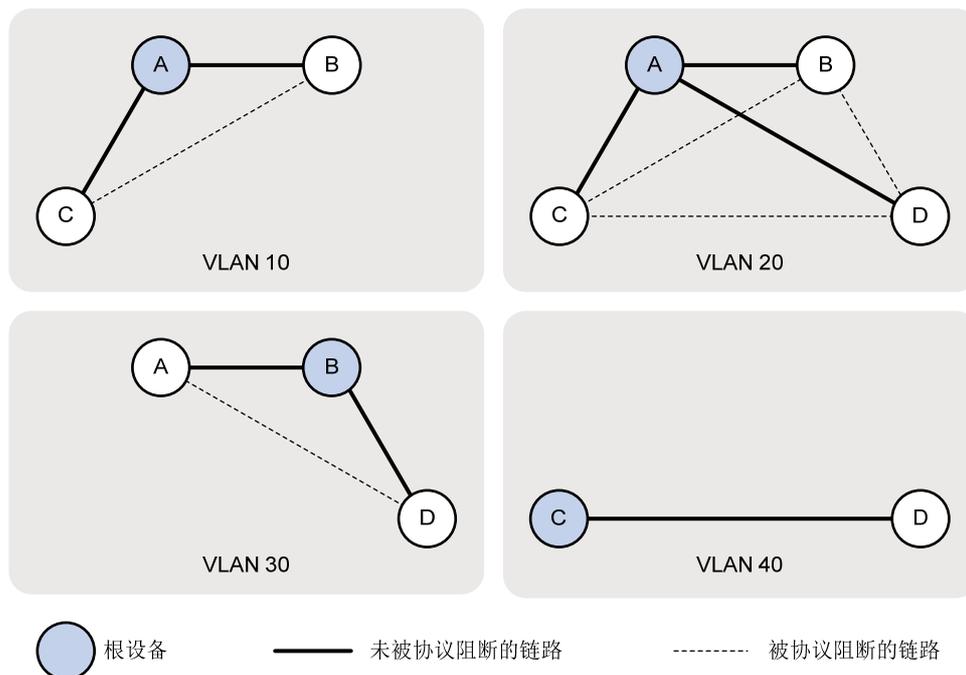
# 查看 Device D 上生成树的简要信息。

```
[DeviceD] display stp brief
```

VLAN ID	Port	Role	STP State	Protection
20	GigabitEthernet1/0/1	ALTE	DISCARDING	NONE
20	GigabitEthernet1/0/2	ROOT	FORWARDING	NONE
20	GigabitEthernet1/0/3	ALTE	DISCARDING	NONE
30	GigabitEthernet1/0/1	ROOT	FORWARDING	NONE
30	GigabitEthernet1/0/2	ALTE	DISCARDING	NONE
40	GigabitEthernet1/0/3	ROOT	FORWARDING	NONE

根据上述显示信息，可以绘出各VLAN所对应生成树的拓扑，如 [图 1-24](#) 所示。

图1-24 各 VLAN 所对应生成树的拓扑图



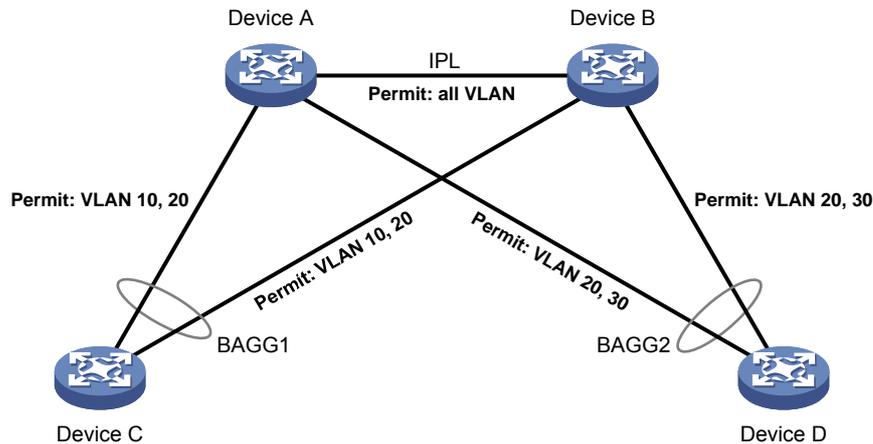
### 1.5.3 生成树支持分布式聚合典型配置举例

#### 1. 组网需求

- Device A 和 Device B 配置分布式聚合，其中 DeviceA 为分布式聚合 Primary 设备，DeviceB 为分布式聚合 Secondary 设备。
- Device A 和 Device B 为汇聚层设备，Device C 和 Device D 为接入层设备。
- 通过配置 PVST，使 VLAN 10、20 和 30 中的报文分别按照其各自 VLAN 所对应的生成树转发。
- 由于 VLAN 10、20 和 30 在汇聚层设备终结，因此 Device A 和 Device B 组成的分布式聚合系统为 VLAN 10、20 和 30 的根桥。

## 2. 组网图

图1-25 生成树支持分布式聚合典型配置组网图



## 3. 配置步骤

### (1) 配置 VLAN 和端口

请按照 图 1-25 在 Device A 和 Device B 上分别创建 VLAN 10、20 和 30，在 Device C 上创建 VLAN 10 和 VLAN 20，在 Device D 上创建 VLAN 20 和 VLAN 30；将各设备的各端口配置为 Trunk 端口并允许相应的 VLAN 通过，具体配置过程略。

### (2) 配置 Device A 和 Device B 分布式聚合，具体配置过程略。

分布式聚合的详细介绍请参考“二层技术-以太网交换配置指导”中的“DRNI”。

### (3) 配置 Device A

# 配置生成树的工作模式为 PVST 模式。

```
<DeviceA> system-view
[DeviceA] stp mode pvst
```

# 配置本设备为 VLAN 10 和 VLAN 20 的根桥。

```
[DeviceA] stp vlan 10 20 root primary
```

# 全局开启生成树协议，并开启 VLAN 10、20 和 30 中的生成树协议。

```
[DeviceA] stp global enable
[DeviceA] stp vlan 10 20 30 enable
```

### (4) 配置 Device B

Device B 的配置与 Device A 相同，配置过程略。

### (5) 配置 Device C

# 配置生成树的工作模式为 PVST 模式。

```
<DeviceC> system-view
[DeviceC] stp mode pvst
```

# 全局开启生成树协议，并开启 VLAN 10 和 VLAN 20 中的生成树协议。

```
[DeviceC] stp global enable
[DeviceC] stp vlan 10 20 enable
```

### (6) 配置 Device D

# 配置生成树的工作模式为 PVST 模式。

```
<DeviceD> system-view
[DeviceD] stp mode pvst
# 全局开启生成树协议，并开启 VLAN 20 和 VLAN 30 中的生成树协议。
[DeviceD] stp global enable
[DeviceD] stp vlan 20 30 enable
```

#### 4. 验证配置

当网络拓扑稳定后，通过使用 **display stp brief** 命令可以查看各设备上生成树的简要信息。例如：

# 在分布式聚合 Primary 设备 Device A 上查看分布式聚合系统的生成树简要信息。

```
[DeviceA] display stp brief
```

VLAN ID	Port	Role	STP State	Protection
10	Bridge-Aggregation1	DESI	FORWARDING	NONE
20	Bridge-Aggregation1	DESI	FORWARDING	NONE
20	Bridge-Aggregation2	DESI	FORWARDING	NONE
30	Bridge-Aggregation2	DESI	FORWARDING	NONE

# 查看 Device C 上生成树的简要信息。

```
[DeviceC] display stp brief
```

VLAN ID	Port	Role	STP State	Protection
10	Bridge-Aggregation1	ROOT	FORWARDING	NONE
20	Bridge-Aggregation1	ROOT	FORWARDING	NONE

# 查看 Device D 上生成树的简要信息。

```
[DeviceD] display stp brief
```

VLAN ID	Port	Role	STP State	Protection
20	Bridge-Aggregation2	ROOT	FORWARDING	NONE
30	Bridge-Aggregation2	ROOT	FORWARDING	NONE

# 目 录

<b>1 LLDP</b> .....	<b>1-1</b>
1.1 LLDP简介 .....	1-1
1.1.1 LLDP产生背景 .....	1-1
1.1.2 LLDP基本概念 .....	1-1
1.1.3 LLDP工作机制 .....	1-6
1.1.4 协议规范 .....	1-6
1.2 LLDP配置任务简介 .....	1-7
1.3 配置LLDP基本功能 .....	1-7
1.3.1 开启LLDP功能 .....	1-7
1.3.2 配置LLDP桥模式 .....	1-8
1.3.3 配置LLDP工作模式 .....	1-8
1.3.4 配置接口初始化延迟时间 .....	1-9
1.3.5 配置轮询功能 .....	1-9
1.3.6 配置允许发布的TLV类型 .....	1-10
1.3.7 配置管理地址及其封装格式 .....	1-12
1.3.8 调整LLDP相关参数 .....	1-12
1.3.9 配置LLDP报文的封装格式 .....	1-13
1.3.10 关闭LLDP的PVID不一致检查功能 .....	1-13
1.4 配置LLDP兼容CDP功能 .....	1-14
1.4.1 配置准备 .....	1-14
1.4.2 配置LLDP兼容CDP功能 .....	1-14
1.5 配置LLDP Trap和LLDP-MED Trap功能 .....	1-15
1.6 配置LLDP报文的源MAC地址为指定三层以太网接口的MAC地址 .....	1-16
1.7 配置设备支持通过LLDP生成对端管理地址的ARP或ND表项 .....	1-16
1.8 LLDP显示和维护 .....	1-16
1.9 LLDP典型配置举例 .....	1-17
1.9.1 LLDP基本功能配置举例 .....	1-17

# 1 LLDP

## 1.1 LLDP简介

### 1.1.1 LLDP产生背景

目前，网络设备的种类日益繁多且各自的配置错综复杂，为了使不同厂商的设备能够在网络中相互发现并交互各自的系统及配置信息，需要有一个标准的信息交流平台。

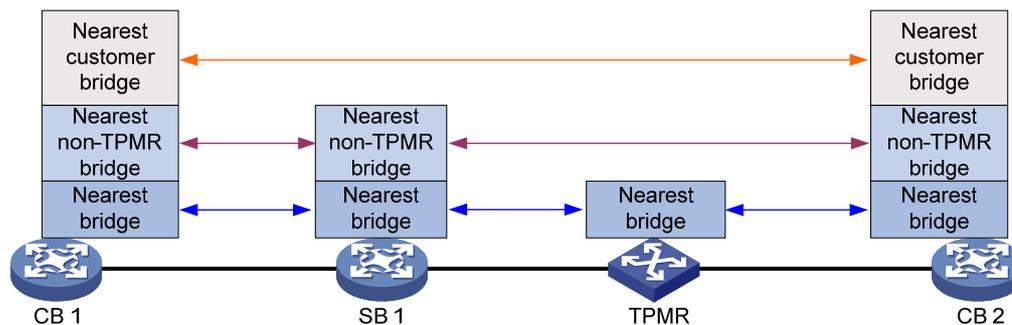
LLDP（Link Layer Discovery Protocol，链路层发现协议）就是在这样的背景下产生的，它提供了一种标准的链路层发现方式，可以将本端设备的信息（包括主要能力、管理地址、设备标识、接口标识等）组织成不同的 TLV（Type/Length/Value，类型/长度/值），并封装在 LLDPU（Link Layer Discovery Protocol Data Unit，链路层发现协议数据单元）中发布给与自己直连的邻居，邻居收到这些信息后将其以标准 MIB（Management Information Base，管理信息库）的形式保存起来，以供网络管理系统查询及判断链路的通信状况。有关 MIB 的详细介绍，请参见“网络管理和监控配置指导”中的“SNMP”。

### 1.1.2 LLDP基本概念

#### 1. LLDP代理

LLDP代理是LLDP协议运行实体的一个抽象映射。一个接口下，可以运行多个LLDP代理。目前LLDP定义的代理类型包括：Nearest Bridge（最近桥代理）、Nearest non-TPMR Bridge（最近非TPMR桥代理）和Nearest Customer Bridge（最近客户桥代理）。其中TPMR（Two-Port MAC Relay，双端口MAC中继），是一种只有两个可供外部访问桥端口的桥，支持MAC桥的功能子集。TPMR对于所有基于帧的介质无关协议都是透明的，但如下协议除外：以TPMR为目的地的协议、以保留MAC地址为目的地址但TPMR定义为不予转发的协议。LLDP在相邻的代理之间进行协议报文交互，并基于代理创建及维护邻居信息。如 图 1-1 所示，是LLDP不同类型的代理邻居关系示意图。其中，CB（Customer Bridge，客户桥）和SB（Service Bridge，服务桥）表示LLDP的两种桥模式。

图1-1 LLDP 邻居关系示意图

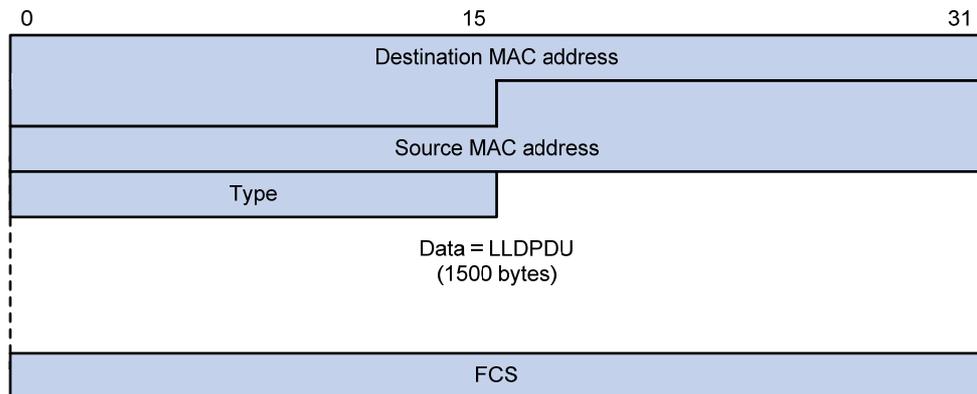


#### 2. LLDP报文

封装有 LLDPU 的报文称为 LLDP 报文，其封装格式有两种：Ethernet II 和 SNAP（Subnetwork Access Protocol，子网访问协议）。

### (1) Ethernet II 格式封装的 LLDP 报文

图1-2 Ethernet II 格式封装的 LLDP 报文

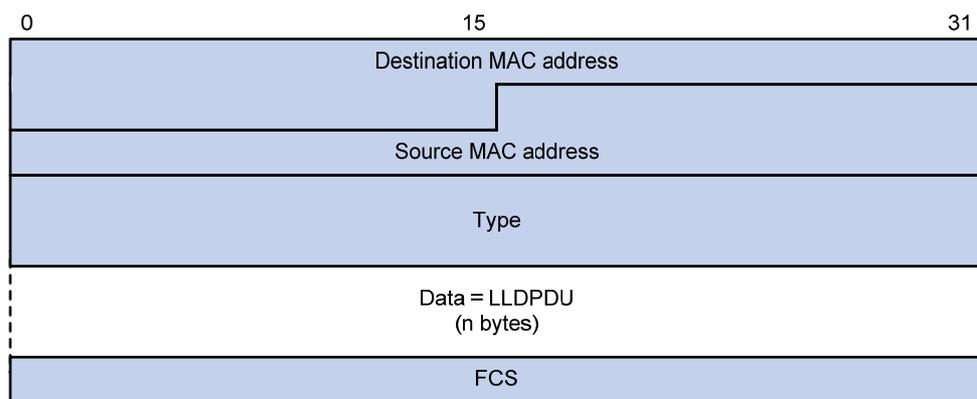


如 [图 1-2](#)所示，是以Ethernet II格式封装的LLDP报文，其中各字段的含义如下：

- **Destination MAC address:** 目的 MAC 地址。为区分同一接口下不同类型代理发送及接收的 LLDP 报文，LLDP 协议规定了不同的组播 MAC 地址作为不同类型代理的 LLDP 报文的目的地 MAC 地址。其中固定的组播 MAC 地址 0x0180-c200-000E 供最近桥代理类型的 LLDP 报文使用，0x0180-c200-0000 供最近客户桥代理类型的 LLDP 报文使用，0x0180-c200-0003 供最近非 TPMR 桥代理类型的 LLDP 报文使用。
- **Source MAC address:** 源 MAC 地址，为端口 MAC 地址。
- **Type:** 报文类型，为 0x88CC。
- **Data:** 数据内容，为 LLDPDU。
- **FCS:** 帧检验序列，用来对报文进行校验。

### (2) SNAP 格式封装的 LLDP 报文

图1-3 SNAP 格式封装的 LLDP 报文



如 [图 1-3](#)所示，是以SNAP格式封装的LLDP报文，其中各字段的含义如下：

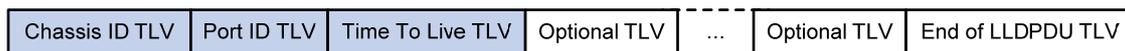
- **Destination MAC address:** 目的 MAC 地址，与 Ethernet II 格式封装的 LLDP 报文目的 MAC 地址相同。
- **Source MAC address:** 源 MAC 地址，为端口 MAC 地址。

- Type: 报文类型，为 0xAAAA-0300-0000-88CC。
- Data: 数据内容，为 LLDPDU。
- FCS: 帧检验序列，用来对报文进行校验。

### 3. LLDPDU

LLDPDU 就是封装在 LLDP 报文数据部分的数据单元。在组成 LLDPDU 之前，设备先将本地信息封装成 TLV 格式，再由若干个 TLV 组合成一个 LLDPDU 封装在 LLDP 报文的数据部分进行传送。

图1-4 LLDPDU 的封装格式



如 [图 1-4](#) 所示，蓝色的 Chassis ID TLV、Port ID TLV、Time To Live TLV 是每个 LLDPDU 都必须携带的，其余的 TLV 则为可选携带。每个 LLDPDU 最多可携带 32 种 TLV。

### 4. TLV

TLV 是组成 LLDPDU 的单元，每个 TLV 都代表一个信息。LLDP 可以封装的 TLV 包括基本 TLV、802.1 组织定义 TLV、802.3 组织定义 TLV 和 LLDP-MED（Link Layer Discovery Protocol Media Endpoint Discovery，链路层发现协议媒体终端发现）TLV。

基本 TLV 是网络设备管理基础的一组 TLV，802.1 组织定义 TLV、802.3 组织定义 TLV 和 LLDP-MED TLV 则是由标准组织或其他机构定义的 TLV，用于增强对网络设备的管理，可根据实际需要选择是否在 LLDPDU 中发送。

#### (1) 基本 TLV

在基本 TLV 中，有几种 TLV 对于实现 LLDP 功能来说是必选的，即必须在 LLDPDU 中发布，如 [表 1-1](#) 所示。

表1-1 基本 TLV

TLV 名称	说明	是否必须发布
Chassis ID	发送设备的桥 MAC 地址	是
Port ID	标识 LLDPDU 发送端的端口。如果 LLDPDU 中携带有 LLDP-MED TLV，其内容为端口的 MAC 地址；否则，其内容为端口的名称	是
Time To Live	本设备信息在邻居设备上的存活时间	是
End of LLDPDU	LLDPDU 的结束标识，是 LLDPDU 的最后一个 TLV	否
Port Description	端口的描述	否
System Name	设备的名称	否
System Description	系统的描述	否
System Capabilities	系统的主要功能以及已开启的功能项	否
Management Address	管理地址，以及该地址所对应的接口号和 OID（Object Identifier，对象标识符）	否

#### (2) 802.1 组织定义 TLV

IEEE 802.1 组织定义TLV的内容如 [表 1-2](#) 所示。

表1-2 IEEE 802.1 组织定义的 TLV

TLV 名称	说明
Port VLAN ID(PVID)	端口VLAN ID
Port and protocol VLAN ID(PPVID)	端口协议VLAN ID
VLAN Name	端口所属VLAN的名称
Protocol Identity	端口所支持的协议类型
DCBX	数据中心桥能力交换协议（Data Center Bridging Exchange Protocol）  说明 本系列交换机不支持 DCBX TLV。
EVB模块	边缘虚拟桥接（Edge Virtual Bridging）模块，具体包括EVB TLV和CDCP（S-Channel Discovery and Configuration Protocol，S通道发现和配置协议）TLV这两种TLV  说明 本系列交换机不支持 EVB TLV。
Link Aggregation	端口是否支持链路聚合以及是否已开启链路聚合
Management VID	管理VLAN
VID Usage Digest	包含VLAN ID使用摘要的数据
ETS Configuration	增强传输选择（Enhanced Transmission Selection）配置
ETS Recommendation	增强传输选择推荐
PFC	基于优先级的流量控制（Priority-based Flow Control）
APP	应用协议（Application Protocol）
QCN	量化拥塞通知（Quantized Congestion Notification）  说明 本系列交换机不支持 QCN TLV。

 说明

- 目前，H3C 设备不支持发送 Protocol Identity TLV 和 VID Usage Digest TLV，但可以接收这两种类型的 TLV。
- 三层以太网接口仅支持 Link Aggregation TLV。

### (3) 802.3 组织定义 TLV

IEEE 802.3 组织定义TLV的内容如 [表 1-3](#) 所示。

表1-3 IEEE 802.3 组织定义的 TLV

TLV 名称	说明
MAC/PHY Configuration/Status	端口支持的速率和双工状态、是否支持端口速率自动协商、是否已开启自动协商功能以及当前的速率和双工状态
Power Via MDI	端口的供电能力，包括PoE（Power over Ethernet，以太网供电）的类型（包括PSE（Power Sourcing Equipment，供电设备）和PD（Powered Device，受电设备）两种）、PoE端口的远程供电模式、是否支持PSE供电、是否已开启PSE供电、供电方式是否可控、供电类型、功率来源、功率优先级、PD请求功率值、PSE分配功率值
Maximum Frame Size	端口支持的最大帧长度
Power Stateful Control	端口的电源状态控制，包括PSE/PD所采用的电源类型、供/受电的优先级以及供/受电的功率
Energy-Efficient Ethernet	节能以太网



说明

Power Stateful Control TLV 是在 IEEE P802.3at D1.0 版本中被定义的，之后的版本不再支持该 TLV。H3C 设备只有在收到 Power Stateful Control TLV 后才会发送该类型的 TLV。

#### (4) LLDP-MED TLV

LLDP-MED TLV为VoIP（Voice over IP，在IP网络上传送语音）提供了许多高级的应用，包括基本配置、网络策略配置、地址信息以及目录管理等，满足了语音设备的不同生产厂商在投资收效、易部署、易管理等方面的要求，并解决了在以太网中部署语音设备的问题，为语音设备的生产者、销售者以及使用者提供了便利。LLDP-MED TLV的内容如 [表 1-4](#) 所示。

表1-4 LLDP-MED TLV

TLV 名称	说明
LLDP-MED Capabilities	网络设备所支持的LLDP-MED TLV类型
Network Policy	网络设备或终端设备上端口的VLAN类型、VLAN ID以及二三层与具体应用类型相关的优先级等
Extended Power-via-MDI	网络设备或终端设备的扩展供电能力，对Power Via MDI TLV进行了扩展
Hardware Revision	终端设备的硬件版本
Firmware Revision	终端设备的固件版本
Software Revision	终端设备的软件版本
Serial Number	终端设备的序列号
Manufacturer Name	终端设备的制造厂商名称
Model Name	终端设备的模块名称
Asset ID	终端设备的资产标识符，以便目录管理和资产跟踪
Location Identification	网络设备的位置标识信息，以供终端设备在基于位置的应用中使用



说明

如果禁止发布 802.3 的组织定义的 MAC/PHY Configuration/Status TLV，则 LLDP-MED TLV 将不会被发布，不论其是否被允许发布；如果禁止发布 LLDP-MED Capabilities TLV，则其他 LLDP-MED TLV 将不会被发布，不论其是否被允许发布。

---

## 5. 管理地址

管理地址是供网络管理系统标识网络设备并进行管理的地址。管理地址可以明确地标识一台设备，从而有利于网络拓扑的绘制，便于网络管理。管理地址被封装在 LLDP 报文的 Management Address TLV 中向外发布。

### 1.1.3 LLDP工作机制

#### 1. LLDP的工作模式

在指定类型的 LLDP 代理下，LLDP 有以下四种工作模式：

- TxRx：既发送也接收 LLDP 报文。
- Tx：只发送不接收 LLDP 报文。
- Rx：只接收不发送 LLDP 报文。
- Disable：既不发送也不接收 LLDP 报文。

当端口的 LLDP 工作模式发生变化时，端口将对协议状态机进行初始化操作。为了避免端口工作模式频繁改变而导致端口不断执行初始化操作，可配置端口初始化延迟时间，当端口工作模式改变时延迟一段时间再执行初始化操作。

#### 2. LLDP报文的发送机制

在指定类型 LLDP 代理下，当端口工作在 TxRx 或 Tx 模式时，设备会周期性地向邻居设备发送 LLDP 报文。如果设备的本地配置发生变化则立即发送 LLDP 报文，以将本地信息的变化情况尽快通知给邻居设备。但为了防止本地信息的频繁变化而引起 LLDP 报文的大量发送，使用令牌桶机制对 LLDP 报文发送作限速处理。有关令牌桶的详细介绍，请参见“ACL 和 QoS 配置指导”中的“流量监管、流量整形和接口限速”。

当设备的工作模式由 Disable/Rx 切换为 TxRx/Tx，或者发现了新的邻居设备（即收到一个新的 LLDP 报文且本地尚未保存发送该报文设备的信息）时，该设备将自动启用快速发送机制，即将 LLDP 报文的发送周期设置为快速发送周期，并连续发送指定数量的 LLDP 报文后再恢复为正常的发送周期。

#### 3. LLDP报文的接收机制

当端口工作在 TxRx 或 Rx 模式时，设备会对收到的 LLDP 报文及其携带的 TLV 进行有效性检查，通过检查后再将邻居信息保存到本地，并根据 Time To Live TLV 中 TTL（Time to Live，生存时间）的值来设置邻居信息在本地设备上的老化时间，若该值为零，则立刻老化该邻居信息。

### 1.1.4 协议规范

与 LLDP 相关的协议规范有：

- IEEE 802.1AB-2005: Station and Media Access Control Connectivity Discovery
- IEEE 802.1AB 2009: Station and Media Access Control Connectivity Discovery
- ANSI/TIA-1057: Link Layer Discovery Protocol for Media Endpoint Devices
- IEEE Std 802.1Qaz-2011: Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks-Amendment 18: Enhanced Transmission Selection for Bandwidth Sharing Between Traffic Classes

## 1.2 LLDP配置任务简介

表1-5 LLDP 配置任务简介

配置任务		说明	详细配置
配置LLDP基本功能	开启LLDP功能	必选	<a href="#">1.3.1</a>
	配置LLDP桥模式	可选	<a href="#">1.3.2</a>
	配置LLDP工作模式	可选	<a href="#">1.3.3</a>
	配置接口初始化延迟时间	可选	<a href="#">1.3.4</a>
	配置轮询功能	可选	<a href="#">1.3.5</a>
	配置允许发布的TLV类型	可选	<a href="#">1.3.6</a>
	配置管理地址及其封装格式	可选	<a href="#">1.3.7</a>
	调整LLDP相关参数	可选	<a href="#">1.3.8</a>
	配置LLDP报文的封装格式	可选	<a href="#">1.3.9</a>
	关闭LLDP的PVID不一致检查功能	可选	<a href="#">1.3.10</a>
配置LLDP兼容CDP功能		可选	<a href="#">1.4</a>
配置LLDP Trap和LLDP-MED Trap功能		可选	<a href="#">1.5</a>
配置LLDP报文的源MAC地址为指定三层以太网的MAC地址		可选	<a href="#">1.6</a>
配置设备支持通过LLDP生成对端管理地址的ARP或ND表项		可选	<a href="#">1.7</a>

## 1.3 配置LLDP基本功能

### 1.3.1 开启LLDP功能

只有当全局和接口上都开启了 LLDP 功能后，该功能才会生效。

需要注意的是，当 LLDP 与 OpenFlow 配合使用时，需要在 Openflow 网络设备上全局开启 LLDP 功能，但为了此时 LLDP 不影响 OpenFlow 控制器发现拓扑，建议在 OpenFlow 实例内的接口上关闭 LLDP 功能。有关 OpenFlow 的详细介绍，请参见“OpenFlow 配置指导”中的“OpenFlow”。

表1-6 开启 LLDP 功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
全局开启LLDP功能	<b>lldp global enable</b>	空配置启动时，使用软件功能缺省值，LLDP功能在全局处于关闭状态 缺省配置启动时，使用软件功能出厂值，LLDP功能在全局处于开启状态 关于空配置启动和缺省配置启动的详细介绍，请参见“基础配置指导”中的“配置文件管理”
进入二/三层以太网接口视图、管理以太网接口视图、二/三层聚合接口视图	<b>interface interface-type interface-number</b>	-
在接口上开启LLDP功能	<b>lldp enable</b>	缺省情况下，LLDP功能在接口上处于开启状态

### 1.3.2 配置LLDP桥模式

LLDP 桥模式有客户桥模式和服务桥模式两种：

- 工作于客户桥模式时，设备可支持最近桥代理、最近非 TPMR 桥代理和最近客户桥代理，即设备对报文目的 MAC 地址为上述代理的 MAC 地址的 LLDP 报文进行处理，对报文目的 MAC 地址为其他 MAC 地址的 LLDP 报文进行 VLAN 内透传。
- 工作于服务桥模式时，设备可支持最近桥代理和最近非 TPMR 桥代理，即设备对报文目的 MAC 地址为上述代理的 MAC 地址的 LLDP 报文进行处理，对报文目的 MAC 地址为其他 MAC 地址的 LLDP 报文进行 VLAN 内透传。

表1-7 配置 LLDP 桥模式

操作	命令	说明
进入系统视图	<b>system-view</b>	-
配置LLDP桥模式	<b>lldp mode service-bridge</b>	缺省情况下，LLDP桥模式为客户桥模式

### 1.3.3 配置LLDP工作模式

LLDP 工作模式分为以下四种：

- **TxRx**：既发送也接收 LLDP 报文。
- **Tx**：只发送不接收 LLDP 报文。
- **Rx**：只接收不发送 LLDP 报文。
- **Disable**：既不发送也不接收 LLDP 报文。

表1-8 配置 LLDP 工作模式

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二/三层以太网接口视图、管理以太网接口视图、二/三层聚合接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
配置LLDP的工作模式	<p>在二/三层以太网接口视图或管理以太网接口视图下：</p> <p><b>lldp [ agent { nearest-customer   nearest-nontpmr } ] admin-status { disable   rx   tx   txrx }</b></p> <p>在二/三层聚合接口视图下：</p> <p><b>lldp agent { nearest-customer   nearest-nontpmr } admin-status { disable   rx   tx   txrx }</b></p>	<p>缺省情况下，最近桥代理类型的LLDP工作模式为TxRx，最近客户桥代理和最近非TPMR桥代理类型的LLDP工作模式为Disable</p> <p>以太网接口视图下，未指定<b>agent</b>参数时，表示配置最近桥代理的工作模式</p> <p>聚合接口视图下，只支持配置最近桥客户桥代理和最近非TPMR代理的工作模式</p>

### 1.3.4 配置接口初始化延迟时间

当接口上 LLDP 的工作模式发生变化时，接口将对协议状态机进行初始化操作，通过配置接口初始化的延迟时间，可以避免由于工作模式频繁改变而导致接口不断地进行初始化。

表1-9 配置接口初始化延迟时间

操作	命令	说明
进入系统视图	<b>system-view</b>	-
配置接口初始化的延迟时间	<b>lldp timer reinit-delay</b> <i>delay</i>	缺省情况下，接口初始化的延迟时间为2秒

### 1.3.5 配置轮询功能

在开启了轮询功能后，LLDP 将以轮询间隔周期性地查询本设备的相关配置是否发生改变，如果发生改变将触发 LLDP 报文的发送，以将本设备的配置变化迅速通知给其他设备。

表1-10 配置轮询功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二/三层以太网接口视图、管理以太网接口视图、二/三层聚合接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-

操作	命令	说明
开启轮询功能并配置轮询间隔	在二/三层以太网接口视图或管理以太网接口视图下： <b>lldp [ agent { nearest-customer   nearest-nontpmr } ]</b> <b>check-change-interval interval</b> 在二/三层聚合接口视图下： <b>lldp agent { nearest-customer   nearest-nontpmr }</b> <b>check-change-interval interval</b>	缺省情况下，轮询功能处于关闭状态

### 1.3.6 配置允许发布的TLV类型

表1-11 配置允许发布的 TLV 类型

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二/三层以太网接口视图、管理以太网接口视图、二/三层聚合接口视图	<b>interface interface-type interface-number</b>	-
配置接口上允许发布的TLV类型（二层以太网接口视图）	<b>lldp tlv-enable { basic-tlv { all   port-description   system-capability   system-description   system-name   management-address-tlv [ ipv6 ] [ ip-address ] }   dot1-tlv { all   port-vlan-id   link-aggregation   protocol-vlan-id [ vlan-id ]   vlan-name [ vlan-id ]   management-vid [ mvlan-id ] }   dot3-tlv { all   mac-physic   max-frame-size   power }   med-tlv { all   capability   inventory   network-policy [ vlan-id ]   power-over-ethernet   location-id { civic-address device-type country-code { ca-type ca-value }&amp;&lt;1-10&gt;   elin-address tel-number } } }</b> <b>lldp agent nearest-nontpmr tlv-enable { basic-tlv { all   port-description   system-capability   system-description   system-name   management-address-tlv [ ipv6 ] [ ip-address ] }   dot1-tlv { all   port-vlan-id   link-aggregation } }</b> <b>lldp agent nearest-customer tlv-enable { basic-tlv { all   port-description   system-capability   system-description   system-name   management-address-tlv [ ipv6 ] [ ip-address ] }   dot1-tlv { all   port-vlan-id   link-aggregation } }</b>	缺省情况下： <ul style="list-style-type: none"> <li>• 最近桥代理允许发布除 Location-id TLV、Port And Protocol VLAN ID TLV、VLAN Name TLV、Management VLAN ID TLV 之外所有类型的 TLV</li> <li>• 最近非 TPMR 桥代理不发布任何 TLV</li> <li>• 最近客户桥代理允许发布基本 TLV 和 IEEE 802.1 组织定义 TLV</li> </ul>

操作	命令	说明
配置接口上允许发布的TLV类型（三层以太网接口视图）	<pre> <b>lldp tlv-enable</b> { <b>basic-tlv</b> { <b>all</b>   <b>port-description</b>   <b>system-capability</b>   <b>system-description</b>   <b>system-name</b>   <b>management-address-tlv</b> [ <b>ipv6</b> ] [ <i>ip-address</i> ]   <b>interface loopback</b> <i>interface-number</i> ] }   <b>dot1-tlv</b> { <b>all</b>   <b>link-aggregation</b> }   <b>dot3-tlv</b> { <b>all</b>   <b>mac-physic</b>   <b>max-frame-size</b>   <b>power</b> }   <b>med-tlv</b> { <b>all</b>   <b>capability</b>   <b>inventory</b>   <b>power-over-ethernet</b>   <b>location-id</b> { <b>civic-address</b> <i>device-type country-code</i> { <i>ca-type ca-value</i> } &amp;&lt;1-10&gt;   <b>elin-address</b> <i>tel-number</i> } } } <b>lldp agent</b> { <b>nearest-nontpmr</b>   <b>nearest-customer</b> } <b>tlv-enable</b> { <b>basic-tlv</b> { <b>all</b>   <b>port-description</b>   <b>system-capability</b>   <b>system-description</b>   <b>system-name</b>   <b>management-address-tlv</b> [ <b>ipv6</b> ] [ <i>ip-address</i> ] }   <b>dot1-tlv</b> { <b>all</b>   <b>link-aggregation</b> } } </pre>	<p>缺省情况下：</p> <ul style="list-style-type: none"> <li>最近桥代理允许发布除 Network Policy TLV 之外所有类型的 TLV，其中 IEEE 802.1 组织定义的 TLV 只支持 Link Aggregation TLV</li> <li>最近非 TPMR 桥代理不发布任何 TLV</li> <li>最近客户桥代理允许发布基本 TLV 和 IEEE 802.1 组织定义 TLV，其中 IEEE 802.1 组织定义的 TLV 只支持 Link Aggregation TLV</li> </ul>
配置接口上允许发布的TLV类型（管理以太网接口视图）	<pre> <b>lldp tlv-enable</b> { <b>basic-tlv</b> { <b>all</b>   <b>port-description</b>   <b>system-capability</b>   <b>system-description</b>   <b>system-name</b>   <b>management-address-tlv</b> [ <b>ipv6</b> ] [ <i>ip-address</i> ] }   <b>dot1-tlv</b> { <b>all</b>   <b>link-aggregation</b> }   <b>dot3-tlv</b> { <b>all</b>   <b>mac-physic</b>   <b>max-frame-size</b>   <b>power</b> }   <b>med-tlv</b> { <b>all</b>   <b>capability</b>   <b>inventory</b>   <b>power-over-ethernet</b>   <b>location-id</b> { <b>civic-address</b> <i>device-type country-code</i> { <i>ca-type ca-value</i> } &amp;&lt;1-10&gt;   <b>elin-address</b> <i>tel-number</i> } } } <b>lldp agent</b> { <b>nearest-nontpmr</b>   <b>nearest-customer</b> } <b>tlv-enable</b> { <b>basic-tlv</b> { <b>all</b>   <b>port-description</b>   <b>system-capability</b>   <b>system-description</b>   <b>system-name</b>   <b>management-address-tlv</b> [ <b>ipv6</b> ] [ <i>ip-address</i> ] }   <b>dot1-tlv</b> { <b>all</b>   <b>link-aggregation</b> } } </pre>	<p>缺省情况下：</p> <ul style="list-style-type: none"> <li>最近桥代理允许发布除 Network Policy TLV 之外所有类型的 TLV，其中 IEEE 802.1 组织定义的 TLV 只支持 Link Aggregation TLV</li> <li>最近非 TPMR 桥代理不发布任何 TLV</li> <li>最近客户桥代理允许发布基本 TLV 和 IEEE 802.1 组织定义 TLV，其中 IEEE 802.1 组织定义的 TLV 只支持 Link Aggregation TLV</li> </ul>
配置接口上允许发布的TLV类型（二层聚合接口视图）	<pre> <b>lldp agent nearest-nontpmr tlv-enable</b> { <b>basic-tlv</b> { <b>all</b>   <b>management-address-tlv</b> [ <b>ipv6</b> ] [ <i>ip-address</i> ]   <b>port-description</b>   <b>system-capability</b>   <b>system-description</b>   <b>system-name</b> }   <b>dot1-tlv</b> { <b>all</b>   <b>port-vlan-id</b> } } <b>lldp agent nearest-customer tlv-enable</b> { <b>basic-tlv</b> { <b>all</b>   <b>management-address-tlv</b> [ <b>ipv6</b> ] [ <i>ip-address</i> ]   <b>port-description</b>   <b>system-capability</b>   <b>system-description</b>   <b>system-name</b> }   <b>dot1-tlv</b> { <b>all</b>   <b>port-vlan-id</b> } } <b>lldp tlv-enable dot1-tlv</b> { <b>protocol-vlan-id</b> [ <i>vlan-id</i> ]   <b>vlan-name</b> [ <i>vlan-id</i> ]   <b>management-vid</b> [ <i>mvlan-id</i> ] } </pre>	<p>不存在最近桥代理</p> <p>缺省情况下：</p> <ul style="list-style-type: none"> <li>不存在最近桥代理</li> <li>最近客户桥代理允许发布基本 TLV 和 IEEE 802.1 组织定义 TLV，其中 IEEE 802.1 组织定义的 TLV 只支持 Port And Protocol VLAN ID TLV、VLAN Name TLV 及 Management VLAN ID TLV</li> </ul>
配置接口上允许发布的TLV类型（三层聚合接口视图）	<pre> <b>lldp agent</b> { <b>nearest-customer</b>   <b>nearest-nontpmr</b> } <b>tlv-enable basic-tlv</b> { <b>all</b>   <b>management-address-tlv</b> [ <b>ipv6</b> ] [ <i>ip-address</i> ]   <b>port-description</b>   <b>system-capability</b>   <b>system-description</b>   <b>system-name</b> } </pre>	<p>不存在最近桥代理</p> <p>缺省情况下：</p> <ul style="list-style-type: none"> <li>不存在最近桥代理</li> <li>最近非 TPMR 桥代理不发布任何 TLV</li> <li>最近客户桥代理只允许发布基本 TLV</li> </ul>

### 1.3.7 配置管理地址及其封装格式

管理地址被封装在 Management Address TLV 中向外发布，封装格式可以是数字或字符串。如果邻居将管理地址以字符串格式封装在 TLV 中，用户可在本地设备上也将封装格式改为字符串，以保证与邻居设备的正常通信。

表1-12 配置管理地址及其封装格式

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二/三层以太网接口视图、管理以太网接口视图或二/三层聚合接口视图	<b>interface interface-type interface-number</b>	-
允许在LLDP报文中发布管理地址并配置所发布的管理地址	在二层以太网接口视图/管理以太网接口视图下： <b>lldp [ agent { nearest-customer   nearest-nontpmr } ] tlv-enable basic-tlv management-address-tlv [ ipv6 ] [ ip-address ]</b> 在三层以太网接口视图下： <b>lldp [ agent { nearest-customer   nearest-nontpmr } ] tlv-enable basic-tlv management-address-tlv [ ipv6 ] [ ip-address ]   interface loopback interface-number</b> 在二/三层聚合接口视图下： <b>lldp agent { nearest-customer   nearest-nontpmr } tlv-enable basic-tlv management-address-tlv [ ipv6 ] [ ip-address ]</b>	缺省情况下，最近桥代理和最近客户桥代理类型的LLDP允许在LLDP报文中发布管理地址，最近非TPMR桥代理类型LLDP不允许在LLDP报文中发布管理地址 对于LLDP报文中所要发布的IPv6格式的管理地址，仅支持数字格式的封装格式
配置管理地址在TLV中的封装格式为字符串格式	在二/三层以太网接口视图或管理以太网接口视图下： <b>lldp [ agent { nearest-customer   nearest-nontpmr } ] management-address-format string</b> 在二/三层聚合接口视图下： <b>lldp agent { nearest-customer   nearest-nontpmr } management-address-format string</b>	缺省情况下，管理地址在TLV中的封装格式为数字格式

### 1.3.8 调整LLDP相关参数

LLDP 报文所携 Time To Live TLV 中 TTL 的值用来设置邻居信息在本地设备上的老化时间，由于  $TTL = \text{Min}(65535, (TTL \text{ 乘数} \times \text{LLDP 报文的发送间隔} + 1))$ ，即取 65535 与  $(TTL \text{ 乘数} \times \text{LLDP 报文的发送间隔} + 1)$  中的最小值，因此通过调整 TTL 乘数可以控制本设备信息在邻居设备上的老化时间。

表1-13 调整 LLDP 相关参数

操作	命令	说明
进入系统视图	<b>system-view</b>	-
配置TTL乘数	<b>lldp hold-multiplier value</b>	缺省情况下，TTL乘数为4

操作	命令	说明
配置LLDP报文的发送间隔	<b>lldp timer tx-interval interval</b>	缺省情况下，LLDP报文的发送间隔为30秒
配置LLDP报文发包限速的令牌桶大小	<b>lldp max-credit credit-value</b>	缺省情况下，发包限速令牌桶大小为5
配置快速发送LLDP报文的个数	<b>lldp fast-count count</b>	缺省情况下，快速发送LLDP报文的个数为4个
配置快速发送LLDP报文的间隔	<b>lldp timer fast-interval interval</b>	缺省情况下，快速发送LLDP报文的发送间隔为1秒

### 1.3.9 配置LLDP报文的封装格式

LLDP 报文的封装格式有 Ethernet II 和 SNAP 两种：

- 当采用 Ethernet II 封装格式时，开启了 LLDP 功能的接口所发送的 LLDP 报文将以 Ethernet II 格式封装。
- 当采用 SNAP 封装格式时，开启了 LLDP 功能的接口所发送的 LLDP 报文将以 SNAP 格式封装。

需要注意的是，LLDP 早期版本要求只有配置为相同的封装格式才能处理该格式的 LLDP 报文，因此为了确保与运行 LLDP 早期版本的设备成功通信，必须配置为与之相同的封装格式。

表1-14 配置 LLDP 报文的封装格式

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二/三层以太网接口视图、管理以太网接口视图、二/三层聚合接口视图	<b>interface interface-type interface-number</b>	-
配置LLDP报文的封装格式为SNAP格式	在二/三层以太网接口视图或管理以太网接口视图下： <b>lldp [ agent { nearest-customer   nearest-nontpmr } ] encapsulation snap</b> 在二/三层聚合接口视图下： <b>lldp agent { nearest-customer   nearest-nontpmr } encapsulation snap</b>	缺省情况下，LLDP报文的封装格式为Ethernet II格式

### 1.3.10 关闭LLDP的PVID不一致检查功能

一般组网情况下，要求链路两端的 PVID 保持一致。设备会对收到的 LLDP 报文中的 PVID TLV 进行检查，如果发现报文中的 PVID 与本端 PVID 不一致，则认为网络中可能存在错误配置，LLDP 会打印日志信息，提示用户。

但在一些特殊情况下，可以允许链路两端的 PVID 配置不一致。例如为了简化接入设备的配置，各接入设备的上行口采用相同的 PVID，而对端汇聚设备的各接口采用不同的 PVID，从而使各接入设备的流量进入不同 VLAN。此时，可以关闭 LLDP 的 PVID 不一致性检查功能。

表1-15 关闭 LLDP 的 PVID 不一致检查功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
关闭LLDP的PVID不一致检查功能	<b>lldp ignore-pvid-inconsistency</b>	缺省情况下，LLDP的PVID不一致检查功能处于开启状态

## 1.4 配置LLDP兼容CDP功能

当设备与只支持 CDP（Cisco Discovery Protocol，思科发现协议）不支持 LLDP 的 Cisco 设备直连时，可以通过配置 LLDP 兼容 CDP 功能与直连设备交互信息。

设备上配置 LLDP 兼容 CDP 功能后，可以利用 LLDP 来接收、识别从直连设备接收的 CDP 报文，并向直连设备发送 CDP 报文。设备向 CDP 邻居设备发送的报文中携带设备 ID，与邻居设备相连端口的 ID、端口 IP 地址以及生存时间信息。其中，端口 IP 地址为该端口允许通过的、对应 VLAN 接口上配置有 IP 地址且处于 up 状态的最小 VLAN 的主 IP 地址，如果该端口允许通过的所有 VLAN 所对应的 VLAN 接口上都未配置 IP 地址或均处于 down 状态，则不发布端口 IP 地址。设备可以识别的 CDP 邻居设备的信息请参见 **display lldp neighbor-information** 命令显示信息中的 CDP neighbor-information 相关字段，有关该命令的详细介绍请参见“二层技术-以太网交换命令参考”中的“LLDP”。

### 1.4.1 配置准备

在配置 LLDP 兼容 CDP 功能之前，需完成以下任务：

- 全局开启 LLDP 功能。
- 在设备与支持 CDP 的设备相连接的接口上开启 LLDP 功能，并配置接口的 LLDP 工作模式为 TxRx。

### 1.4.2 配置LLDP兼容CDP功能

LLDP 兼容 CDP 功能有以下几种工作模式：

- TxRx：既发送也接收 CDP 报文。
- Rx：接收但不发送 CDP 报文。
- Disable：既不发送也不接收 CDP 报文。

要使 LLDP 兼容 CDP 功能生效，必须先在全局开启 LLDP 兼容 CDP 功能，并将 LLDP 兼容 CDP 功能的工作模式配置为 TxRx。



提示

由于 CDP 报文所携 Time To Live TLV 中 TTL 的最大值为 255，而 CDP 报文的发送间隔由 LLDP 报文的发送间隔控制，因此为保证 LLDP 兼容 CDP 功能的正常运行，建议配置 LLDP 报文的发送间隔值不大于实际 TTL 的 1/3。

表1-16 配置 LLDP 兼容 CDP 功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
开启LLDP兼容CDP功能	<b>lldp compliance cdp</b>	缺省情况下，LLDP兼容CDP功能处于关闭状态
进入二层/三层以太网接口视图或管理以太网接口视图	<b>interface interface-type interface-number</b>	-
配置LLDP兼容CDP功能的工作模式为TxRx	<b>lldp compliance admin-status cdp txrx</b>	缺省情况下，LLDP兼容CDP功能的工作模式为Disable

## 1.5 配置LLDP Trap和LLDP-MED Trap功能

开启 LLDP Trap 或 LLDP-MED Trap 功能后，设备可以通过向网管系统发送 Trap 信息以通告如发现新的 LLDP 邻居或 LLDP-MED 邻居、与原来邻居的通信链路发生故障等重要事件。

LLDP Trap 和 LLDP-MED Trap 信息的发送间隔是指设备向网管系统发送 Trap 信息的最小时间间隔，通过调整该时间间隔，可以避免由于邻居信息频繁变化而导致 Trap 信息的频繁发送。

表1-17 配置 LLDP Trap 和 LLDP-MED Trap 功能

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入二层以太网接口视图/二层聚合接口视图/三层以太网接口视图/三层聚合接口视图/管理以太网接口视图	<b>interface interface-type interface-number</b>	-
开启LLDP Trap功能	在二/三层以太网接口视图或管理以太网接口视图下： <b>lldp [ agent { nearest-customer   nearest-nontpmr } ] notification remote-change enable</b> 在二/三层聚合接口视图下： <b>lldp agent { nearest-customer   nearest-nontpmr } notification remote-change enable</b>	缺省情况下，LLDP Trap功能处于关闭状态
开启LLDP-MED Trap功能	在二/三层以太网接口视图或管理以太网接口视图下： <b>lldp notification med-topology-change enable</b>	缺省情况下，LLDP-MED Trap功能处于关闭状态
退回系统视图	<b>quit</b>	-
(可选) 配置LLDP Trap和LLDP-MED Trap信息的发送间隔	<b>lldp timer notification-interval interval</b>	缺省情况下，LLDP Trap和LLDP-MED Trap信息的发送间隔均为30秒

## 1.6 配置LLDP报文的源MAC地址为指定三层以太网接口的MAC地址

配置本特性后，LLDP报文的源MAC地址为指定三层以太网接口的MAC地址。

表1-18 配置LLDP报文的源MAC地址为指定三层以太网接口的MAC地址

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入三层以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
配置LLDP报文源MAC地址为指定三层以太网接口的MAC地址	<b>lldp source-mac</b> <b>vlan</b> <i>vlan-id</i>	缺省情况下，LLDP报文源MAC地址为主接口的MAC地址

## 1.7 配置设备支持通过LLDP生成对端管理地址的ARP或ND表项

配置本特性后，当接口收到携带IPv4格式Management Address TLV的LLDP报文后，会生成该报文携带的管理地址与报文源MAC地址组成的ARP表项；当接口收到携带IPv6格式Management Address TLV的LLDP报文后，会生成该报文携带的管理地址与报文源MAC地址组成的ND表项。

表1-19 配置设备支持通过LLDP生成对端管理地址的ARP或ND表项

操作	命令	说明
进入系统视图	<b>system-view</b>	-
进入三层以太网接口视图	<b>interface</b> <i>interface-type</i> <i>interface-number</i>	-
配置接口收到携带Management Address TLV的LLDP报文后生成ARP表项或ND表项	<b>lldp management-address</b> { <b>arp-learning</b>   <b>nd-learning</b> } [ <b>vlan</b> <i>vlan-id</i> ]	缺省情况下，接口收到携带Management Address TLV的LLDP报文后不生成ARP表项和ND表项 ARP表项和ND表项的生成互不影响，可同时配置

## 1.8 LLDP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LLDP的运行情况，通过查看显示信息验证配置的效果。

表1-20 LLDP显示和维护

操作	命令
显示LLDP本地信息	<b>display lldp local-information</b> [ <b>global</b>   <b>interface</b> <i>interface-type</i> <i>interface-number</i> ]
显示由邻居设备发来的LLDP信息	<b>display lldp neighbor-information</b> [ [ <b>interface</b> <i>interface-type</i> <i>interface-number</i> ] [ <b>agent</b> { <b>nearest-bridge</b>   <b>nearest-customer</b>   <b>nearest-nontpmr</b> } ] [ <b>verbose</b> ] ]   <b>list</b> [ <b>system-name</b> <i>system-name</i> ] ]

操作	命令
显示LLDP的统计信息	<b>display lldp statistics</b> [ <b>global</b>   [ <b>interface</b> <i>interface-type</i> <i>interface-number</i> ] [ <b>agent</b> { <b>nearest-bridge</b>   <b>nearest-customer</b>   <b>nearest-nontpmr</b> } ] ]
显示LLDP的状态信息	<b>display lldp status</b> [ <b>interface</b> <i>interface-type</i> <i>interface-number</i> ] [ <b>agent</b> { <b>nearest-bridge</b>   <b>nearest-customer</b>   <b>nearest-nontpmr</b> } ]
显示接口上可发送的可选TLV信息	<b>display lldp tlv-config</b> [ <b>interface</b> <i>interface-type</i> <i>interface-number</i> ] [ <b>agent</b> { <b>nearest-bridge</b>   <b>nearest-customer</b>   <b>nearest-nontpmr</b> } ]

## 1.9 LLDP典型配置举例

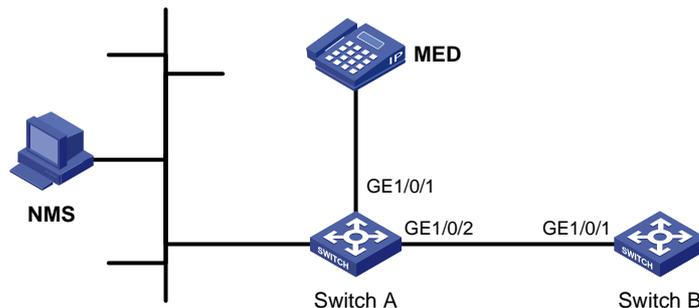
### 1.9.1 LLDP基本功能配置举例

#### 1. 组网需求

- NMS（Network Management System，网络管理系统）与 Switch A 相连，Switch A 通过接口 GigabitEthernet1/0/1 和 GigabitEthernet1/0/2 分别与 MED 设备和 Switch B 相连。
- 通过在 Switch A 和 Switch B 上配置 LLDP 功能，使 NMS 可以对 Switch A 与 MED 设备之间、以及 Switch A 与 Switch B 之间链路的通信情况进行判断。

#### 2. 组网图

图1-5 LLDP 基本功能配置组网图



#### 3. 配置步骤

##### (1) 配置 Switch A

# 全局开启 LLDP 功能。

```
<SwitchA> system-view
[SwitchA] lldp global enable
```

# 在接口 GigabitEthernet1/0/1 和 GigabitEthernet1/0/2 上分别开启 LLDP 功能(此步骤可省略, LLDP 功能在接口上缺省开启), 并配置 LLDP 工作模式为 Rx。

```
[SwitchA] interface gigabitethernet 1/0/1
[SwitchA-GigabitEthernet1/0/1] lldp enable
[SwitchA-GigabitEthernet1/0/1] lldp admin-status rx
[SwitchA-GigabitEthernet1/0/1] quit
[SwitchA] interface gigabitethernet 1/0/2
```

```
[SwitchA-GigabitEthernet1/0/2] lldp enable
[SwitchA-GigabitEthernet1/0/2] lldp admin-status rx
[SwitchA-GigabitEthernet1/0/2] quit
```

## (2) 配置 Switch B

# 全局开启 LLDP 功能。

```
<SwitchB> system-view
[SwitchB] lldp global enable
```

# 在接口 GigabitEthernet1/0/1 上开启 LLDP 功能（此步骤可省略，LLDP 功能在接口上缺省开启），并配置 LLDP 工作模式为 Tx。

```
[SwitchB] interface gigabitethernet 1/0/1
[SwitchB-GigabitEthernet1/0/1] lldp enable
[SwitchB-GigabitEthernet1/0/1] lldp admin-status tx
[SwitchB-GigabitEthernet1/0/1] quit
```

## 4. 验证配置

# 显示 Switch A 上全局和所有接口的 LLDP 状态信息。

```
[SwitchA] display lldp status
Global status of LLDP: Enable
Bridge mode of LLDP: customer-bridge
The current number of LLDP neighbors: 2
The current number of CDP neighbors: 0
LLDP neighbor information last changed time: 0 days, 0 hours, 4 minutes, 40 seconds
Transmit interval          : 30s
Fast transmit interval     : 1s
Transmit max credit        : 5
Hold multiplier            : 4
Reinit delay               : 2s
Trap interval              : 30s
Fast start times           : 4
```

LLDP status information of port 1 [GigabitEthernet1/0/1]:

```
LLDP agent nearest-bridge:
Port status of LLDP       : Enable
Admin status              : Rx_Only
Trap flag                 : No
MED trap flag            : No
Polling interval         : 0s
Number of LLDP neighbors  : 1
Number of MED neighbors   : 1
Number of CDP neighbors   : 0
Number of sent optional TLV : 21
Number of received unknown TLV : 0
```

LLDP agent nearest-nontpnr:

```
Port status of LLDP       : Enable
Admin status              : Disable
Trap flag                 : No
MED trap flag            : No
```

Polling interval : 0s  
Number of LLDP neighbors : 0  
Number of MED neighbors : 0  
Number of CDP neighbors : 0  
Number of sent optional TLV : 1  
Number of received unknown TLV : 0

LLDP agent nearest-customer:

Port status of LLDP : Enable  
Admin status : Disable  
Trap flag : No  
MED trap flag : No  
Polling interval : 0s  
Number of LLDP neighbors : 0  
Number of MED neighbors : 0  
Number of CDP neighbors : 0  
Number of sent optional TLV : 16  
Number of received unknown TLV : 0

LLDP status information of port 2 [GigabitEthernet1/0/2]:

LLDP agent nearest-bridge:

Port status of LLDP : Enable  
Admin status : Rx\_Only  
Trap flag : No  
MED trap flag : No  
Polling interval : 0s  
Number of LLDP neighbors : 1  
Number of MED neighbors : 0  
Number of CDP neighbors : 0  
Number of sent optional TLV : 21  
Number of received unknown TLV : 3

LLDP agent nearest-nontpmr:

Port status of LLDP : Enable  
Admin status : Disable  
Trap flag : No  
MED trap flag : No  
Polling interval : 0s  
Number of LLDP neighbors : 0  
Number of MED neighbors : 0  
Number of CDP neighbors : 0  
Number of sent optional TLV : 1  
Number of received unknown TLV : 0

LLDP agent nearest-customer:

Port status of LLDP : Enable  
Admin status : Disable  
Trap flag : No

```
MED trap flag           : No
Polling interval        : 0s
Number of LLDP neighbors : 0
Number of MED neighbors  : 0
Number of CDP neighbors  : 0
Number of sent optional TLV : 16
Number of received unknown TLV : 0
```

由此可见，Switch A 的接口 GigabitEthernet1/0/1 上连接了一个 MED 邻居设备，GigabitEthernet1/0/2 上则连接了一个非 MED 邻居设备，且这两个接口的 LLDP 工作模式都为 Rx，即只接收而不发送 LLDP 报文。

# 将 Switch A 和 Switch B 间的链路断掉后，再显示 Switch A 上所有接口的 LLDP 状态信息。

```
[SwitchA] display lldp status
Global status of LLDP: Enable
The current number of LLDP neighbors: 1
The current number of CDP neighbors: 0
LLDP neighbor information last changed time: 0 days, 0 hours, 5 minutes, 20 seconds
Transmit interval           : 30s
Fast transmit interval      : 1s
Transmit max credit         : 5
Hold multiplier             : 4
Reinit delay                : 2s
Trap interval               : 30s
Fast start times            : 4
```

```
LLDP status information of port 1 [GigabitEthernet1/0/1]:
```

```
LLDP agent nearest-bridge:
Port status of LLDP       : Enable
Admin status              : Rx_Only
Trap flag                 : No
MED trap flag             : No
Polling interval          : 0s
Number of LLDP neighbors  : 1
Number of MED neighbors   : 1
Number of CDP neighbors   : 0
Number of sent optional TLV : 0
Number of received unknown TLV : 5
```

```
LLDP agent nearest-nontpnr:
```

```
Port status of LLDP       : Enable
Admin status              : Disabl
Trap flag                 : No
MED trap flag             : No
Polling interval          : 0s
Number of LLDP neighbors  : 0
Number of MED neighbors   : 0
Number of CDP neighbors   : 0
Number of sent optional TLV : 1
Number of received unknown TLV : 0
```

LLDP status information of port 2 [GigabitEthernet1/0/2]:

LLDP agent nearest-bridge:

```
Port status of LLDP      : Enable
Admin status             : Rx_Only
Trap flag                : No
MED trap flag            : No
Polling interval         : 0s
Number of LLDP neighbors : 0
Number of MED neighbors  : 0
Number of CDP neighbors  : 0
Number of sent optional TLV : 0
Number of received unknown TLV : 0
```

LLDP agent nearest-nontpnr:

```
Port status of LLDP      : Enable
Admin status             : Disable
Trap flag                : No
MED trap flag            : No
Polling interval         : 0s
Number of LLDP neighbors : 0
Number of MED neighbors  : 0
Number of CDP neighbors  : 0
Number of sent optional TLV : 1
Number of received unknown TLV : 0
```

LLDP agent nearest-customer:

```
Port status of LLDP      : Enable
Admin status             : Disable
Trap flag                : No
MED trap flag            : No
Polling interval         : 0s
Number of LLDP neighbors : 0
Number of MED neighbors  : 0
Number of CDP neighbors  : 0
Number of sent optional TLV : 16
Number of received unknown TLV : 0
```

由此可见，Switch A 的接口 GigabitEthernet1/0/2 上已经没有任何邻居设备了。