

# H3C SecPath M9000 系列 多业务安全网关

## 二层技术-以太网安全配置指导(V7)

Copyright © 2021-2024 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。

除新华三技术有限公司的商标外，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相关功能原理及配置。

前言部分包含如下内容：

- [读者对象](#)
- [本书约定](#)
- [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”。
[ ]	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。
	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。
	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。

## 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 MAC 地址表 .....	1-1
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 MAC 地址表配置任务简介 .....	1-2
1.3 手工配置 MAC 地址表项.....	1-2
1.3.1 功能简介 .....	1-2
1.3.2 配置限制和指导 .....	1-2
1.3.3 配置准备 .....	1-3
1.3.4 配置静态/动态 MAC 地址表项.....	1-3
1.3.5 配置黑洞 MAC 地址表项 .....	1-3
1.4 配置动态 MAC 地址表项的老化时间 .....	1-4
1.5 关闭 MAC 地址学习功能.....	1-4
1.5.1 功能简介 .....	1-4
1.5.2 关闭全局的 MAC 地址学习功能 .....	1-4
1.5.3 关闭接口的 MAC 地址学习功能 .....	1-5
1.6 配置 MAC 地址数学习上限 .....	1-5
1.7 配置当达到 MAC 地址数学习上限时的报文转发规则 .....	1-6
1.7.1 功能简介 .....	1-6
1.7.2 配置当达到接口的 MAC 地址数学习上限时的报文转发规则.....	1-6
1.8 开启 MAC 地址表告警功能 .....	1-6
1.9 MAC 地址表显示和维护.....	1-7

# 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 MAC 地址查询 MAC 地址表，如果 MAC 地址表中包含与报文目的 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 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。

静态 MAC 地址表项、黑洞 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、黑洞 MAC 地址表项覆盖。静态 MAC 地址表项、黑洞 MAC 地址表项不会彼此覆盖。

## 1.2 MAC地址表配置任务简介

本章中的所有配置均为可选，请根据实际情况选择配置。

- [手工配置 MAC 地址表项](#)
  - [配置静态/动态 MAC 地址表项](#)
  - [配置黑洞 MAC 地址表项](#)
- [配置动态 MAC 地址表项的老化时间](#)
- 配置 MAC 地址学习功能
  - [关闭 MAC 地址学习功能](#)
- [开启 MAC 地址表告警功能](#)

## 1.3 手工配置MAC地址表项

### 1.3.1 功能简介

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	<p>缺省情况下，如果表项中的接口与报文入接口不同，则设备丢弃该报文；表项中的接口与报文入接口相同时，则设备根据目的MAC地址转发该报文</p> <p>关闭报文入接口与静态MAC地址表项匹配检查功能后，设备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</p>
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>

### 1.3.2 配置限制和指导

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 地址表项不会丢失，手工配置的动态 MAC 地址表项会丢失。

设备的保留 MAC 地址和三层以太网接口/三层以太网子接口/三层聚合接口/三层聚合子接口的 MAC 地址不允许配置为静态、动态、黑洞 MAC。

### 1.3.3 配置准备

手工配置 MAC 地址表项时，必须先创建指定接口所属的 VLAN，否则配置失败。

### 1.3.4 配置静态/动态 MAC 地址表项

#### 1. 系统视图下配置静态/动态 MAC 地址表项

- (1) 进入系统视图。

```
system-view
```

- (2) 添加或者修改静态/动态 MAC 地址表项。

```
mac-address { dynamic | static } mac-address interface interface-type  
interface-number vlan vlan-id
```

缺省情况下，未配置静态/动态 MAC 地址表项。

**interface** 参数指定的接口必须属于 *vlan-id* 参数指定的 VLAN。

#### 2. 接口视图下配置静态/动态 MAC 地址表项

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

- 进入二层以太网接口视图。

```
interface interface-type interface-number
```

- 进入二层聚合接口视图。

```
interface bridge-aggregation interface-number
```

- (3) 在接口下添加或者修改静态/动态 MAC 地址表项。

```
mac-address { dynamic | static } mac-address vlan vlan-id
```

缺省情况下，接口下未配置静态/动态 MAC 地址表项。

当前接口必须属于 *vlan-id* 参数指定的 VLAN。

### 1.3.5 配置黑洞 MAC 地址表项

- (1) 进入系统视图。

```
system-view
```

- (2) 添加或者修改黑洞 MAC 地址表项。

```
mac-address blackhole mac-address vlan vlan-id
```

缺省情况下，未配置黑洞 MAC 地址表项。

## 1.4 配置动态MAC地址表项的老化时间

### 1. 功能简介

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。

### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 配置动态 MAC 地址表项的老化时间。

```
mac-address timer { aging seconds | no-aging }
```

缺省情况下,动态 MAC 地址表项的老化时间为 300 秒。

## 1.5 关闭MAC地址学习功能

### 1.5.1 功能简介

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，对于已经存在的动态 MAC 地址表项,待老化时间结束后自然老化。

### 1.5.2 关闭全局的 MAC 地址学习功能

#### 1. 配置限制和指导

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。

#### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 关闭全局的 MAC 地址学习功能。

```
undo mac-address mac-learning enable
```

缺省情况下，全局的 MAC 地址学习功能处于开启状态。

### 1.5.3 关闭接口的 MAC 地址学习功能

#### 1. 功能简介

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。

#### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

- 进入二层以太网接口视图。

```
interface interface-type interface-number
```

- 进入二层聚合接口视图。

```
interface bridge-aggregation interface-number
```

- (3) 关闭接口的 MAC 地址学习功能。

```
undo mac-address mac-learning enable
```

缺省情况下，接口的 MAC 地址学习功能处于开启状态。

### 1.6 配置MAC地址数学习上限

#### 1. 功能简介

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。

#### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

- 进入二层以太网接口视图。

```
interface interface-type interface-number
```

- 进入二层聚合接口视图。

```
interface bridge-aggregation interface-number
```

- (3) 配置接口的 MAC 地址数学习上限。

```
mac-address max-mac-count count
```

缺省情况下，接口的 MAC 地址数学习上限为 1024。

## 1.7 配置当达到MAC地址数学习上限时的报文转发规则

### 1.7.1 功能简介

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。

### 1.7.2 配置当达到接口的 MAC 地址数学习上限时的报文转发规则

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

- 进入二层以太网接口视图。

```
interface interface-type interface-number
```

- 进入二层聚合接口视图。

```
interface bridge-aggregation interface-number
```

- (3) 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。

```
mac-address max-mac-count enable-forwarding
```

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。

## 1.8 开启MAC地址表告警功能

### 1. 功能简介

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”。

### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 开启 MAC 地址表的告警功能。

```
snmp-agent trap enable mac-address
```

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。

## 1.9 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1-2 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</i> <i>interface-number</i> ]   <b>blackhole</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</i> <i>interface-number</i> ]

# 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 以太网链路聚合应用场景 .....	1-1
1.1.2 聚合组、成员端口和聚合接口 .....	1-1
1.1.3 操作 Key .....	1-2
1.1.4 配置分类 .....	1-2
1.1.5 聚合模式 .....	1-2
1.1.6 静态聚合模式 .....	1-3
1.1.7 动态聚合 .....	1-4
1.1.8 动态聚合模式 .....	1-5
1.1.9 聚合边缘接口 .....	1-7
1.1.10 聚合负载分担类型 .....	1-7
1.2 vSystem 相关说明 .....	1-7
1.3 以太网链路聚合配置限制和指导 .....	1-7
1.4 以太网链路聚合配置任务简介 .....	1-7
1.5 配置手工聚合 .....	1-8
1.5.1 配置限制和指导 .....	1-8
1.5.2 配置二层聚合组 .....	1-9
1.5.3 配置三层聚合组 .....	1-11
1.5.4 配置引擎聚合组 .....	1-12
1.6 配置聚合接口基本参数 .....	1-13
1.6.1 限制聚合组内选中端口的数量 .....	1-13
1.6.2 配置聚合接口的描述信息 .....	1-14
1.6.3 配置聚合接口的 MAC 地址 .....	1-15
1.6.4 配置聚合接口允许超长帧通过 .....	1-15
1.6.5 关闭聚合成员端口缺省选中功能 .....	1-16
1.6.6 配置二层聚合接口的忽略 VLAN .....	1-16
1.6.7 配置三层聚合接口 MTU .....	1-17
1.6.8 配置引擎聚合接口的 Blade 类型 .....	1-17
1.6.9 配置聚合接口的期望带宽 .....	1-17
1.6.10 配置聚合接口为聚合边缘接口 .....	1-18
1.6.11 配置聚合接口物理连接状态抑制功能 .....	1-18
1.6.12 关闭聚合接口 .....	1-19

1.6.13 开启三层聚合子接口速率统计功能 .....	1-20
1.6.14 恢复聚合接口的缺省配置 .....	1-20
1.7 配置聚合负载分担 .....	1-21
1.7.1 配置聚合负载分担类型 .....	1-21
1.7.2 配置聚合负载分担采用本地转发优先 .....	1-22
1.8 开启聚合流量隔离功能 .....	1-23
1.9 开启聚合接口联动 RBM 状态的功能 .....	1-24
1.10 配置聚合流量重定向功能 .....	1-24
1.10.1 功能简介 .....	1-24
1.10.2 配置限制和指导 .....	1-24
1.10.3 配置全局的聚合流量重定向功能 .....	1-24
1.11 以太网链路聚合显示和维护 .....	1-25
1.12 以太网链路聚合典型配置举例 .....	1-26
1.12.1 二层静态聚合配置举例 .....	1-26
1.12.2 二层动态聚合配置举例 .....	1-28
1.12.3 二层聚合负载分担配置举例 .....	1-31
1.12.4 三层静态聚合配置举例 .....	1-35
1.12.5 三层动态聚合配置举例 .....	1-37
1.12.6 三层聚合负载分担配置举例 .....	1-40

# 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。

### 1.1.1 以太网链路聚合应用场景

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



在防火墙设备中，可以将多个安全引擎上的引擎口捆绑形成一个引擎聚合组，以增加链路带宽的目的。有关安全引擎组的详细描述，请参见“虚拟化技术配置指导”中的“Context”。

### 1.1.2 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。

#### 1. 聚合组和聚合接口的类型

聚合组/聚合接口可以分为以下类型：

- 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。在创建了三层聚合接口之后，还可继续创建该三层聚合接口的子接口，即三层聚合子接口。三层聚合子接口处理与该子接口编号相同的 VLAN 的报文。
- 引擎聚合组/引擎聚合接口：引擎聚合组的成员端口是系统自动添加，其对应的聚合接口称为引擎聚合接口。

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“[1.1.2 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。满足以下条件时，如果成员端口在经过 **LACP**（**Link Aggregation Control Protocol**，链路聚合控制协议）超时时间之后未收到 **LACP** 报文，则该成员端口会被置为该状态：
  - 聚合接口配置为边缘端口。
  - 处于选中/非选中状态的成员端口经过一次 **down**、**up** 后，该成员端口将被置为独立状态。

### 1.1.3 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**。

### 1.1.4 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：属性类配置和协议类配置。

#### 1. 属性类配置

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
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）。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”

#### 2. 协议类配置

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。

### 1.1.5 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：通过 **LACP** 协议实现，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。

## 1.1.6 静态聚合模式

### 1. 选择参考端口

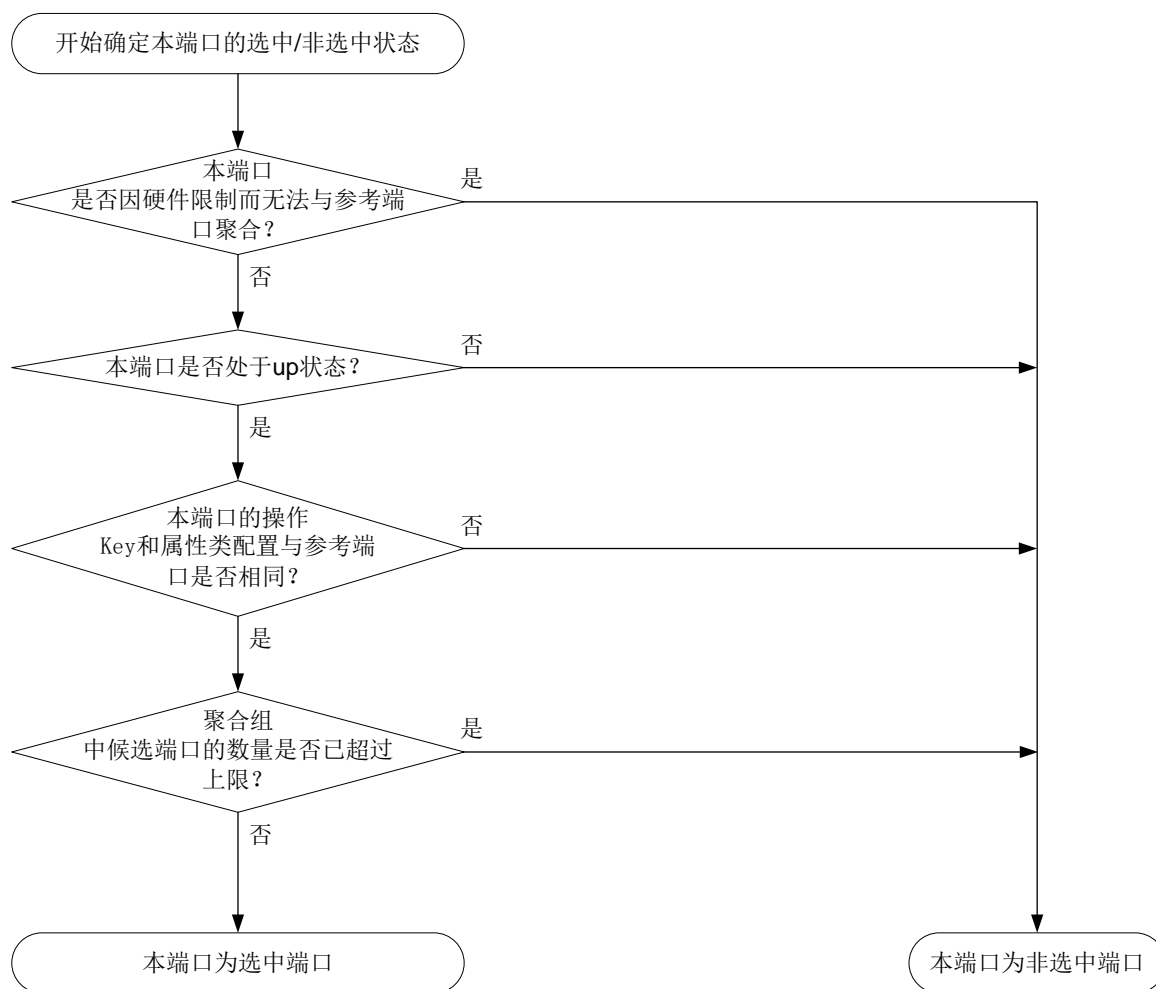
参考端口从本端的成员端口中选出，其操作 **Key** 和属性类配置将作为同一聚合组内的其他成员端口的参照，只有操作 **Key** 和属性类配置与参考端口一致的成员端口才能被选中。

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。

## 1.1.7 动态聚合

### 1. LACP 协议

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。

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。

### 2. 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“虚拟化技术配置指导”中的“IRF”

### 3. 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。

### 4. 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	

5. 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，LACP 超时时间分为短超时（3 秒）和长超时（90 秒）两种。在 LACP 超时时间+3 秒之后（即 6 秒或 93 秒之后），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。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）。

6. 端口加入聚合组的方式

端口加入聚合组的方式为手工动态聚合：两端设备成员端口手工加入动态聚合组。

1.1.8 动态聚合模式

1. 选择参考端口

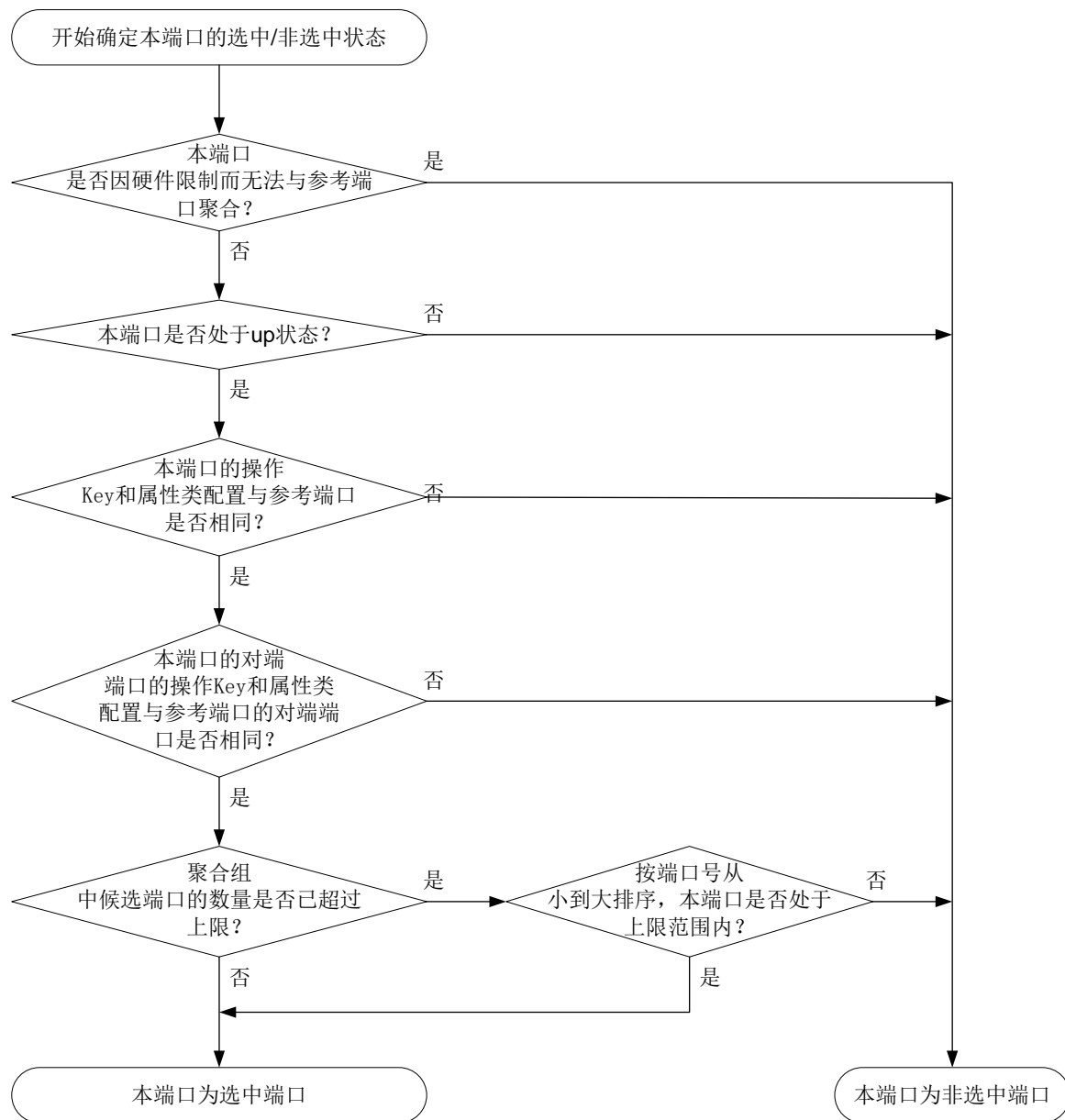
参考端口从聚合链路两端处于 up 状态的成员端口中选出，其操作 Key 和属性类配置将作为同一聚合组内的其他成员端口的参照，只有操作 Key 和属性类配置与参考端口一致的成员端口才能被选中。

- 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。

### 1.1.9 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。

### 1.1.10 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类型：

- 逐流负载分担：按照报文的源/目的 MAC 地址、源/目的服务端口、源/目的 IP 地址、IP 协议类型或 MPLS 标签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。设备还支持按照接口的带宽利用率对数据流进行负载分担。当数据流经过聚合组时，会选择聚合组内带宽利用率最低的接口转发；同一数据流在同一接口转发。
- 逐包负载分担：不区分数据流，而是以报文为单位，将流量分担到不同的成员链路上进行传输。当成员接口下存在 `bandwidth` 配置时后，逐包负载分担时首先根据各个成员接口配置的期望带宽值计算负载分担比例，然后按照比例对收到的报文进行逐包负载分担。

## 1.2 vSystem 相关说明

非缺省 vSystem 支持本特性的部分功能，包括配置三层聚合接口/子接口的描述信息和期望带宽。



说明

非缺省 vSystem 对具体命令的支持情况，请见本特性的命令参考。有关 vSystem 的详细介绍请参见“虚拟化技术配置指导”中的“vSystem”。

## 1.3 以太网链路聚合配置限制和指导

对于手工聚合和自动聚合，建议用户不要混用两种方式，避免端口加入不同的聚合组，从而导致成员端口不被选中。

## 1.4 以太网链路聚合配置任务简介

以太网链路聚合配置任务如下：

- (1) 配置聚合方式
  - [配置手工聚合](#)

(2) (可选) 配置聚合接口基本参数

- [限制聚合组内选中端口的数量](#)
- [配置聚合接口的描述信息](#)
- [配置聚合接口的 MAC 地址](#)
- [配置聚合接口允许超长帧通过](#)
- [关闭聚合成员端口缺省选中功能](#)
- [配置二层聚合接口的忽略 VLAN](#)

二层聚合组中选择选中端口时忽略成员端口的 VLAN 属性。

- [配置三层聚合接口 MTU](#)
- [配置引擎聚合接口的 Blade 类型](#)
- [配置聚合接口的期望带宽](#)
- [配置聚合接口为聚合边缘接口](#)

终端设备未配置动态聚合模式时，使终端设备与网络设备间的链路可以形成备份。

- [配置聚合接口物理连接状态抑制功能](#)
- [关闭聚合接口](#)
- [开启三层聚合子接口速率统计功能](#)
- [恢复聚合接口的缺省配置](#)

(3) (可选) 配置聚合负载分担

- [配置聚合负载分担类型](#)
- [配置聚合负载分担采用本地转发优先](#)

(4) (可选) 开启聚合流量隔离功能

(5) (可选) 配置链路聚合联动功能

- [开启聚合接口联动 RBM 状态的功能](#)

(6) (可选) 配置聚合流量重定向功能

开启聚合流量重定向功能实现聚合链路上流量不中断。

## 1.5 配置手工聚合

### 1.5.1 配置限制和指导

#### 1. 二层聚合组限制

配置了冗余组节点功能的端口将不能加入二层聚合组，有关冗余组节点的详细介绍请参见“虚拟化技术配置指导/冗余备份”中的“冗余组”。

#### 2. 三层聚合组限制

配置了下列功能的端口将不能加入三层聚合组：

- 以太网冗余接口。有关以太网冗余接口的详细介绍请参见“虚拟化技术配置指导”中的“以太网冗余接口”。
- 冗余组节点。有关冗余组的详细介绍请参见“虚拟化技术配置指导”中的“冗余组”。

当三层聚合子接口加入 vSystem 后，不能删除对应的主接口。



### 3. 成员端口限制

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。

以太网接口不能和以太网子接口加入同一个聚合组。

加入聚合组的以太网接口不能再创建子接口；已创建子接口的以太网接口不能加入聚合组。

成员端口为以太网子接口的聚合组对应的聚合接口不能再创建聚合子接口；以太网子接口不能加入已创建聚合子接口的聚合组。

将以太网子接口加入到聚合组前：

- 建议先配置 VLAN 终结命令，以太网子接口加入到聚合组后，将不能修改 VLAN 终结配置。VLAN 终结命令是指 **vlan-type dot1q default**、**vlan-type dot1q untagged**、**vlan-type dot1q vid** 命令，请参见“二层技术-以太网交换命令参考”中的“VLAN 终结”。
- 只有配置了相同 VLAN 终结命令的以太网子接口才能加入同一个聚合组。
- 子接口加入动态聚合组时，如果在子接口上配置 **vlan-type dot1q vid vlan-id-list [ loose ]** 命令，请指定该子接口只终结一个最外层 VLAN ID。即，将 **vlan-id-list** 指定为一个 VLAN ID。

### 4. 聚合组属性类配置和协议类配置限制

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。

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. 聚合模式限制

聚合链路的两端应配置相同的聚合模式。对于不同模式的聚合组，其选中端口存在如下限制：

- 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。

## 1.5.2 配置二层聚合组

### 1. 配置二层静态聚合组

(1) 进入系统视图。

```
system-view
```

(2) 创建二层聚合接口，并进入二层聚合接口视图。

```
interface bridge-aggregation interface-number
```



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。

- (3) 退回系统视图。

**quit**

- (4) 将二层以太网接口加入聚合组。

- a. 进入二层以太网接口视图。

**interface** *interface-type interface-number*

- b. 将二层以太网接口加入聚合组。

**port link-aggregation group** *group-id*

多次执行此步骤可将多个二层以太网接口加入聚合组。

- (5) （可选）配置端口优先级。

**link-aggregation port-priority** *priority*

缺省情况下，端口优先级为 32768。

## 2. 配置二层动态聚合组

- (1) 进入系统视图。

**system-view**

- (2) 配置系统的 LACP 优先级。

**lacp system-priority** *priority*

缺省情况下，系统的 LACP 优先级为 32768。

创建动态聚合组后，不建议修改系统的 LACP 优先级，避免影响动态聚合组成员端口的选中/非选中状态。

- (3) 创建二层聚合接口，并进入二层聚合接口视图。

**interface bridge-aggregation** *interface-number*

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。

- (4) 配置聚合组工作在动态聚合模式下。

**link-aggregation mode** *dynamic*

缺省情况下，聚合组工作在静态聚合模式下。

- (5) 退回系统视图。

**quit**

- (6) 将二层以太网接口加入聚合组。

- a. 进入二层以太网接口视图。

**interface** *interface-type interface-number*

- b. 将二层以太网接口加入聚合组。

**port link-aggregation group** *group-id*

多次执行此步骤可将多个二层以太网接口加入聚合组。

- (7) 配置端口的 LACP 工作模式。

- o 配置端口的 LACP 工作模式为 PASSIVE。

**lacp mode passive**

- 配置端口的 LACP 工作模式为 ACTIVE。

**undo lacp mode**

缺省情况下，端口的 LACP 工作模式为 ACTIVE。

- (8) (可选) 配置端口优先级。

**link-aggregation port-priority *priority***

缺省情况下，端口优先级为 32768。

- (9) (可选) 配置端口的 LACP 超时时间为短超时（3 秒）。

**lacp period short**

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。有关 ISSU 升级的详细介绍请参见“基础配置指导”中的“ISSU 配置”。

### 1.5.3 配置三层聚合组

#### 1. 配置三层静态聚合组

- (1) 进入系统视图。

**system-view**

- (2) 创建三层聚合接口，并进入三层聚合接口视图。

**interface route-aggregation *interface-number***

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。

- (3) 退回系统视图。

**quit**

- (4) 将三层以太网接口加入聚合组。

- a. 进入三层以太网接口视图。

**interface *interface-type interface-number***

- b. 将三层以太网接口加入聚合组。

**port link-aggregation group *group-id***

多次执行此步骤可将多个三层以太网接口加入聚合组。

- (5) (可选) 配置端口优先级。

**link-aggregation port-priority *priority***

缺省情况下，端口优先级为 32768。

#### 2. 配置三层动态聚合组

- (1) 进入系统视图。

**system-view**

- (2) 配置系统的 LACP 优先级。

**lacp system-priority *priority***

缺省情况下，系统的 LACP 优先级为 32768。

创建动态聚合组后，不建议修改系统的 LACP 优先级，避免影响动态聚合组成员端口的选中/非选中状态。

- (3) 创建三层聚合接口，并进入三层聚合接口视图。

```
interface route-aggregation interface-number
```

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。

- (4) 配置聚合组工作在动态聚合模式下。

```
link-aggregation mode dynamic
```

缺省情况下，聚合组工作在静态聚合模式下。

- (5) 退回系统视图。

```
quit
```

- (6) 将三层以太网接口加入聚合组。

- a. 进入三层以太网接口视图。

```
interface interface-type interface-number
```

- b. 将三层以太网接口加入聚合组。

```
port link-aggregation group group-id
```

多次执行此步骤可将多个三层以太网接口加入聚合组。

- (7) 配置端口的 LACP 工作模式。

- o 配置端口的 LACP 工作模式为 PASSIVE。

```
lacp mode passive
```

- o 配置端口的 LACP 工作模式为 ACTIVE。

```
undo lacp mode
```

缺省情况下，端口的 LACP 工作模式为 ACTIVE。

- (8) （可选）配置端口优先级。

```
link-aggregation port-priority priority
```

缺省情况下，端口优先级为 32768。

- (9) （可选）配置端口的 LACP 超时时间为短超时（3 秒）。

```
lacp period short
```

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。有关 ISSU 升级的详细介绍请参见“基础配置指导”中的“ISSU 配置”。

## 1.5.4 配置引擎聚合组

### 1. 功能简介

在防火墙设备中，可以将多个安全引擎上的引擎口捆绑形成一个引擎聚合组，以增加链路带宽的目的。

在创建引擎聚合组之前，需要先创建安全引擎组，之后，系统会自动创建一个和安全引擎组编号一致的引擎聚合组。需要注意的是，由于缺省引擎聚合组编号是 1，所以系统将自动分配编号 2。

引擎聚合组创建后，还需要将安全引擎加入安全引擎组，之后，系统会自动将安全引擎上的引擎口加入刚才创建好的引擎聚合组。有关安全引擎组的详细描述，请参见“虚拟化技术配置指导”中的“Context”。

## 2. 配置限制和指导

引擎聚合口对应的引擎聚合组只能工作在静态聚合模式下。

引擎聚合口的创建和删除，在创建和删除安全引擎组时自动完成，由于默认引擎聚合接口 1 所对应的缺省安全引擎组不能被删除，所以默认引擎聚合接口 1 不能被删除。

## 3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 创建安全引擎组并进入安全引擎组视图。

```
blade-controller-team blade-controller-team-name [ id  
blade-controller-team-id ]
```

缺省情况下，设备有一个安全引擎组，名称为 **Default**，编号为 **1**。

安全引擎组创建后，系统自动创建一个编号和安全引擎组编号相同的引擎聚合组。

有关 **blade-controller-team** 命令的详细介绍，请参见“虚拟化技术命令参考”中的“Context”。

- (3) 将安全引擎加入安全引擎组，进而将引擎口加入引擎聚合组。

（独立运行模式）

```
location blade-controller slot slot-number cpu cpu-number
```

（IRF 模式）

```
location blade-controller chassis chassis-number slot slot-number cpu  
cpu-number
```

缺省情况下，安全业务板插入时，安全业务板上的安全引擎会自动加入缺省的安全引擎组。

安全引擎加入安全引擎组后，系统自动将安全引擎对应的引擎口加入之前系统自动创建的引擎聚合组。

## 1.6 配置聚合接口基本参数

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。

### 1.6.1 限制聚合组内选中端口的数量

#### 1. 功能简介

用户可以根据不同的使用场景，灵活修改聚合组中最大和最小选中端口数，来满足不同需求。

- 最小选中端口数应用场景

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**。
- 最大选中端口数应用场景  
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。

## 2. 配置限制和指导

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。

同一聚合组内，最大选中端口数配置值不能小于最小选中端口数配置值。

## 3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入聚合接口视图。

- 进入二层聚合接口视图。

```
interface bridge-aggregation interface-number
```

- 进入三层聚合接口视图。

```
interface route-aggregation interface-number
```

- (3) 配置聚合组中的最小选中端口数。

```
link-aggregation selected-port minimum min-number
```

缺省情况下，聚合组中的最小选中端口数不受限制。

- (4) 配置聚合组中的最大选中端口数。

```
link-aggregation selected-port maximum max-number
```

缺省情况下，聚合组中的最大选中端口数请参考命令手册。

## 1.6.2 配置聚合接口的描述信息

### 1. 功能简介

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。

### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入聚合接口视图。

- 进入二层聚合接口视图。

```
interface bridge-aggregation interface-number
```

- 进入三层聚合接口视图。

```
interface route-aggregation interface-number
```

- 。进入三层聚合子接口视图。

```
interface route-aggregation interface-number.subnumber
```

- (3) 配置当前接口的描述信息。

```
description text
```

缺省情况下，接口的描述信息为“接口名 Interface”。

### 1.6.3 配置聚合接口的 MAC 地址

#### 1. 功能简介

同一设备上所有聚合接口的缺省 MAC 地址都相同，不同设备上聚合接口的缺省 MAC 地址不同。通常情况下，不需要修改聚合接口的 MAC 地址。

对于一些特殊情况，例如，当设备在特定生成树组网中与第三方厂商设备对接时，由于不同二层聚合接口发出的 BPDU 都具有相同的源 MAC 地址，导致第三方厂商设备收到该报文后会将其视为攻击报文而丢弃，从而引发互通问题。此时就需要为不同的二层聚合接口配置不同的 MAC 地址。

三层聚合子接口的 MAC 地址与对应的聚合接口保持一致。当修改聚合口的 MAC 地址时，子接口的 MAC 地址也会同步修改。

#### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入聚合接口视图。

- 。进入三层聚合接口视图。

```
interface route-aggregation interface-number
```

- (3) 配置聚合接口的 MAC 地址。

```
mac-address mac-address
```

缺省情况下，同一设备上所有聚合接口的 MAC 地址都相同，不同设备上聚合接口的 MAC 地址不同，具体的 MAC 地址请以设备实际情况为准。

### 1.6.4 配置聚合接口允许超长帧通过

#### 1. 功能简介

聚合接口在进行文件传输等大吞吐量数据交换的时候，接口收到的长度大于固定值的帧称为超长帧。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（通过 **undo jumboframe enable** 命令配置），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。

#### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入聚合接口视图。

- 进入二层聚合接口视图。

```
interface bridge-aggregation interface-number
```

- 进入三层聚合接口视图。

```
interface route-aggregation interface-number
```

- (3) 允许超长帧通过。

```
jumboframe enable [size]
```

设备仅允许长度为 9216 的超长帧通过，即 *size* 值固定为 9216。

## 1.6.5 关闭聚合成员端口缺省选中功能

### 1. 功能简介

聚合成员端口缺省选中功能是指动态聚合组的成员端口处于 up 状态时，成员端口在经过 LACP 超时时间之后未收到 LACPDU，则会在所有处于 up 状态的成员端口中选择一个作为选中端口。聚合组选择选中端口时比较各成员端口的端口 ID，端口 ID 最小的作为选中端口。

关闭聚合成员端口缺省选中功能后，动态聚合组中处于 up 状态的成员端口未收到 LACPDU 时，将处于非选中状态。

### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 关闭聚合成员端口缺省选中功能。

```
lacp default-selected-port disable
```

缺省情况下，聚合成员端口缺省选中功能处于开启状态。

## 1.6.6 配置二层聚合接口的忽略 VLAN

### 1. 功能简介

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。

### 2. 配置限制和指导

本功能仅对 Hybrid 或者 Trunk 类型的端口所允许通过的 VLAN 范围有效。

### 3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入二层聚合接口视图。

```
interface bridge-aggregation interface-number
```

- (3) 配置二层聚合接口的忽略 VLAN。

```
link-aggregation ignore vlan vlan-id-list
```

缺省情况下，二层聚合接口未配置忽略 VLAN。



## 1.6.7 配置三层聚合接口 MTU

### 1. 功能简介

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。

### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入三层聚合接口/子接口视图。

```
interface route-aggregation { interface-number |  
interface-number.subnumber }
```

- (3) 配置三层聚合接口/子接口的 MTU 值。

```
mtu size
```

缺省情况下，三层聚合接口/子接口的 MTU 值为 1500 字节。

## 1.6.8 配置引擎聚合接口的 Blade 类型

### 1. 功能简介

引擎聚合组根据接口编号可以分为以下聚合组类型：

- 缺省引擎聚合组：聚合组编号为 1～255。
- 负载分担引擎聚合组：聚合组编号大于等于 256。

负载分担引擎聚合组中包含的 Blade 类型接口固定，缺省引擎聚合组里可以包含多种 Blade 类型接口。在缺省引擎聚合组中，如果这些不同类型的接口一起工作时，将造成流量转发错误。通过配置本功能，可以修改缺省引擎聚合组里 Blade 的类型，保证流量正确转发。

### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入引擎聚合接口视图。

```
interface blade-aggregation interface-number
```

- (3) 配置引擎聚合接口的 Blade 类型。

```
link-aggregation blade blade-type
```



注意

更改引擎聚合接口的 Blade 类型时，必须与引擎接口的类型保持一致，否则会导致网络中断。

---

## 1.6.9 配置聚合接口的期望带宽

### 1. 功能简介

期望带宽供业务模块使用，不会对接口实际带宽造成影响。



## 2. 配置步骤

- (1) 进入系统视图。

**system-view**

- (2) 进入聚合接口视图。

- 进入二层聚合接口视图。

**interface bridge-aggregation** *interface-number*

- 进入三层聚合接口视图。

**interface route-aggregation** *interface-number*

- 进入三层聚合子接口视图。

**interface route-aggregation** *interface-number.subnumber*

- (3) 配置当前接口的期望带宽。

**bandwidth** *bandwidth-value*

缺省情况下，接口的期望带宽=接口的波特率÷1000（kbps）。

### 1.6.10 配置聚合接口为聚合边缘接口

#### 1. 配置限制和指导

该配置仅在聚合接口对应的聚合组为动态聚合组时生效。

当聚合接口配置为聚合边缘接口后，聚合流量重定向功能将不能正常使用，聚合流量重定向功能的相关介绍请参见“[1.10 配置聚合流量重定向功能](#)”。

#### 2. 配置步骤

- (1) 进入系统视图。

**system-view**

- (2) 进入聚合接口视图。

- 进入二层聚合接口视图。

**interface bridge-aggregation** *interface-number*

- 进入三层聚合接口视图。

**interface route-aggregation** *interface-number*

- (3) 配置聚合接口为聚合边缘接口。

**lacp edge-port**

缺省情况下，聚合接口不为聚合边缘接口。

### 1.6.11 配置聚合接口物理连接状态抑制功能

#### 1. 功能简介

聚合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，再上报 CPU 进行处理。

## 2. 配置限制和指导

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。

## 3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入聚合接口视图。

- o 进入二层聚合接口视图。

```
interface bridge-aggregation interface-number
```

- o 进入三层聚合接口视图。

```
interface route-aggregation interface-number
```

- (3) 配置聚合接口物理连接状态抑制功能。

```
link-delay [msec] delay-time [mode { up | updown }]
```

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。

不指定 **mode** 参数，表示对接口状态从 **up** 变成 **down** 事件进行抑制。指定 **mode up** 参数，表示对接口状态从 **down** 变成 **up** 事件进行抑制。指定 **mode updown** 参数，表示接口状态从 **up** 变成 **down** 事件或者 **down** 变成 **up** 事件进行抑制。

## 1.6.12 关闭聚合接口

### 1. 配置限制和指导

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。

### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入聚合接口视图。

- o 进入二层聚合接口视图。

```
interface bridge-aggregation interface-number
```

- o 进入三层聚合接口视图。

```
interface route-aggregation interface-number
```

- o 进入三层聚合子接口视图。

```
interface route-aggregation interface-number.subnumber
```

- (3) 关闭当前接口。

```
shutdown
```

缺省情况下，接口处于开启状态。



注意

执行本命令会导致使用该接口建立的链路中断，不能通信，请谨慎使用。

---

### 1.6.13 开启三层聚合子接口速率统计功能

#### 1. 配置限制和指导

开启本功能可能需要耗费大量系统资源，影响系统性能，请谨慎使用。

当三层聚合接口开启子接口速率统计功能后，设备会定时刷新子接口速率统计信息。

配置本功能后，需要等待两个统计周期，才能显示子接口的速率统计信息。统计周期可以通过 **flow-interval** 命令进行设置。有关 **flow-interval** 命令的详细介绍，请参见“接口管理命令参考”中的“以太网接口”。

#### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入三层聚合接口视图。

```
interface route-aggregation interface-number
```

- (3) 开启三层聚合子接口速率统计功能。

```
sub-interface rate-statistic
```

缺省情况下，三层聚合接口的子接口速率统计功能处于关闭状态。

- (4) （可选）查看子接口速率统计结果。

```
display interface
```

### 1.6.14 恢复聚合接口的缺省配置

#### 1. 配置限制和指导



注意

接口下的某些配置恢复到缺省情况后，会对设备上当前运行的业务产生影响。建议您在执行本配置前，完全了解其对网络产生的影响。

---

您可以在执行 **default** 命令后通过 **display this** 命令确认执行效果。对于未能成功恢复缺省的配置，建议您查阅相关功能的命令手册，手工执行恢复该配置缺省情况的命令。如果操作仍然不能成功，您可以通过设备的提示信息定位原因。

## 2. 配置步骤

- (1) 进入系统视图。

**system-view**

- (2) 进入聚合接口视图。

- 进入二层聚合接口视图。

**interface bridge-aggregation** *interface-number*

- 进入三层聚合接口视图。

**interface route-aggregation** *interface-number*

- 进入三层聚合子接口视图。

**interface route-aggregation** *interface-number.subnumber*

- (3) 恢复当前聚合接口的缺省配置。

**default**

## 1.7 配置聚合负载分担

### 1.7.1 配置聚合负载分担类型

#### 1. 功能简介

聚合负载分担类型支持全局配置或在聚合组内配置两种方式：全局的配置对所有聚合组都有效，而聚合组内的配置只对当前聚合组有效。对于一个聚合组来说，优先采用该聚合组内的配置，只有该聚合组内未进行配置时，才采用全局的配置。

#### 2. 全局配置聚合负载分担类型

- (1) 进入系统视图。

**system-view**

- (2) 配置全局采用的聚合负载分担类型。

**link-aggregation global load-sharing mode** { **destination-ip** | **source-ip** }  
\*

缺省情况下，二层报文缺省使用源、目的 MAC 地址进行负载分担，三层报文缺省使用源、目的 IP 地址进行负载分担。

#### 3. 在聚合组内配置聚合负载分担类型

- (1) 进入系统视图。

**system-view**

- (2) 进入聚合接口视图。

- 进入二层聚合接口视图。

**interface bridge-aggregation** *interface-number*

- 进入三层聚合接口视图。

**interface route-aggregation** *interface-number*

- 进入引擎聚合接口视图。

**interface blade-aggregation** *interface-number*

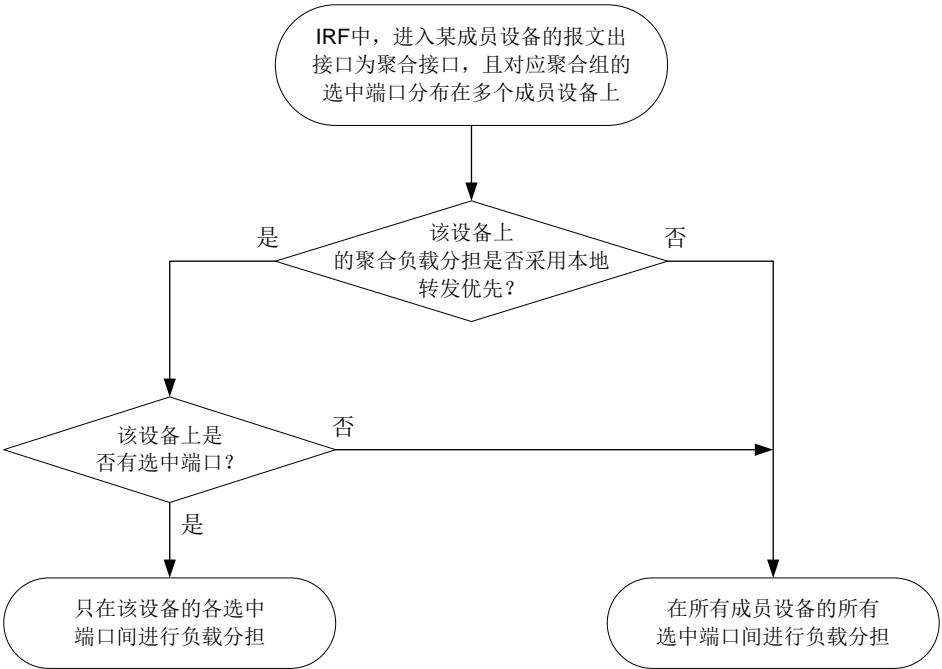
- (3) 配置聚合组内采用的聚合负载分担类型。
- ```
link-aggregation load-sharing mode { { destination-ip | destination-mac  
| destination-port | ip-protocol | mpls-label1 | source-ip | source-mac |  
source-port } * | per-packet }
```
- 缺省情况下，聚合组内采用的聚合负载分担类型与全局的配置相同。

1.7.2 配置聚合负载分担采用本地转发优先

1. 功能简介

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“虚拟化技术配置指导”中的“IRF”。

图1-4 IRF 中成员设备间聚合负载分担处理流程



2. 硬件适配关系

本特性的支持情况与设备型号有关，请以设备的实际情况为准。

| 设备型号                    | 业务板类型          | 说明  |
|-------------------------|----------------|-----|
| M9006<br>M9010<br>M9014 | Blade IV防火墙业务板 | 支持  |
|                         | Blade V防火墙业务板  | 支持  |
|                         | NAT业务板         | 不支持 |
| M9010-GM                | 加密业务板          | 不支持 |
| M9016-V                 | Blade V防火墙业务板  | 支持  |
| M9008-S                 | Blade IV防火墙业务板 | 支持  |

|                                                                       |                 |     |
|-----------------------------------------------------------------------|-----------------|-----|
| M9012-S                                                               | 入侵防御业务板         | 支持  |
|                                                                       | 视频网关业务板         | 支持  |
| M9008-S-V                                                             | Blade IV 防火墙业务板 | 支持  |
| M9000-AI-E4<br>M9000-AI-E8<br>M9000-AI-E16                            | Blade V 防火墙业务板  | 支持  |
| M9000-AK001                                                           | Blade V 防火墙业务板  | 支持  |
| M9000-X06<br>M9000-X06-B<br>M9000-X06-B-G<br>M9000-X06-G<br>M9000-X10 | Blade VI 防火墙业务板 | 不支持 |
| M9000-AI-X06<br>M9000-AI-X10                                          | Blade VI 防火墙业务板 | 不支持 |

### 3. 配置限制和指导

聚合负载分担采用本地转发优先支持全局配置或在聚合组内配置两种方式：全局的配置对所有聚合组都有效，而聚合组内的配置只对当前聚合组有效。对于一个聚合组来说，优先采用该聚合组内的配置，只有该聚合组内未进行配置时，才采用全局的配置。

### 4. 配置全局的聚合负载分担采用本地转发优先

(1) 进入系统视图。

```
system-view
```

(2) 配置全局的聚合负载分担采用本地转发优先。

```
link-aggregation load-sharing mode local-first
```

缺省情况下，聚合负载分担采用本地转发优先。

## 1.8 开启聚合流量隔离功能

### 1. 功能简介

配置本功能后，就表示设备上聚合接口流量被隔离，即该设备所有聚合接口的成员端口都处在非选中状态，对端设备与之相应聚合组的成员端口也处在非选中状态。

### 2. 配置限制和指导

本命令仅对动态聚合接口生效，对于静态聚合接口不生效。

### 3. 配置步骤

(1) 进入系统视图。

```
system-view
```

(2) 开启聚合流量隔离功能。

**link-aggregation lacp isolate**

缺省情况下，聚合流量隔离功能处于关闭状态。

## 1.9 开启聚合接口联动RBM状态的功能

### 1. 功能简介

在主备模式 RBM 组网中，让聚合接口中位于 RBM 主设备上的成员接口作为选中接口，位于 RBM 备设备上的成员接口作为非选中接口。从而，确保流量经过聚合接口交给 RBM 主设备处理。

### 2. 配置限制和指导

聚合接口下配置本功能后，即使该聚合接口中没有选中端口也会保持 Up 状态。

### 3. 配置步骤

- (1) 进入系统视图。

**system-view**

- (2) 进入聚合接口视图。

- 进入二层聚合接口视图。

**interface bridge-aggregation interface-number**

- 进入三层聚合接口视图。

**interface route-aggregation interface-number**

- (3) 开启聚合接口联动 RBM 状态的功能。

**link-aggregation rbm-related**

缺省情况下，聚合接口联动 RBM 状态的功能处于关闭状态。

## 1.10 配置聚合流量重定向功能

### 1.10.1 功能简介

在开启了聚合流量重定向功能后，当手工关闭聚合组内某选中端口或重启聚合组内某选中端口所在的 slot 时，系统可以将该端口上的流量重定向到其他选中端口上，从而实现聚合链路上流量的不中断。其中，已知单播报文可以实现零丢包，非已知单播报文不保证不丢包。聚合流量重定向过程中，对于聚合组中新选中的端口，流量不会重定向到该端口上。

### 1.10.2 配置限制和指导

必须在聚合链路两端都开启聚合流量重定向功能才能实现聚合链路上流量的不中断。

如果同时开启聚合流量重定向功能和生成树功能，在重启 slot 时会出现少量的丢包，因此不建议同时开启上述两个功能。

当聚合接口配置为聚合边缘接口后，聚合流量重定向功能将不能正常使用。

只有动态聚合组支持聚合流量重定向功能。

### 1.10.3 配置全局的聚合流量重定向功能

- (1) 进入系统视图。

**system-view**

- (2) 开启聚合流量重定向功能。

**link-aggregation lacp traffic-redirect-notification enable**

缺省情况下，聚合流量重定向功能处于关闭状态。

## 1.11 以太网链路聚合显示和维护

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。



说明

非缺省 vSystem 不支持部分显示和维护命令，具体情况请见本特性的命令参考。

表1-4 以太网链路聚合显示和维护

| 操作                   | 命令                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 显示聚合接口的相关信息          | <b>display interface</b> [ { <b>blade-aggregation</b>   <b>bridge-aggregation</b>   <b>route-aggregation</b> } [ <b>interface-number</b> ] ] [ <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><br>[ <b>interface</b> [ { <b>blade-aggregation</b>   <b>bridge-aggregation</b>   <b>route-aggregation</b> } <b>interface-number</b> ] ]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 显示聚合组内采用的聚合负载分担的选路信息 | (独立运行模式)<br><b>display link-aggregation load-sharing path interface</b><br>{ <b>bridge-aggregation</b>   <b>route-aggregation</b> }<br><b>interface-number ingress-port interface-type</b><br><b>interface-number</b> [ <b>route</b> ] { { <b>destination-ip</b> <b>ip-address</b>   <b>destination-ipv6</b> <b>ipv6-address</b> }  <br>{ <b>source-ip</b> <b>ip-address</b>   <b>source-ipv6</b> <b>ipv6-address</b> }  <br><b>destination-mac</b> <b>mac-address</b>   <b>destination-port</b> <b>port-id</b>   <b>ethernet-type</b> <b>type-number</b>   <b>ip-protocol</b> <b>protocol-id</b>   <b>source-mac</b> <b>mac-address</b>   <b>source-port</b> <b>port-id</b>   <b>vlan</b> <b>vlan-id</b> } * <b>slot</b> <b>slot-number</b> [ <b>cpu</b> <b>cpu-number</b> ]<br>(IRF模式)<br><b>display link-aggregation load-sharing path interface</b><br>{ <b>bridge-aggregation</b>   <b>route-aggregation</b> }<br><b>interface-number ingress-port interface-type</b><br><b>interface-number</b> [ <b>route</b> ] { { <b>destination-ip</b> <b>ip-address</b>   <b>destination-ipv6</b> <b>ipv6-address</b> }  <br>{ <b>source-ip</b> <b>ip-address</b>   <b>source-ipv6</b> <b>ipv6-address</b> }  <br><b>destination-mac</b> <b>mac-address</b>   <b>destination-port</b> <b>port-id</b>   <b>ethernet-type</b> <b>type-number</b>   <b>ip-protocol</b> <b>protocol-id</b>   <b>source-mac</b> <b>mac-address</b>   <b>source-port</b> <b>port-id</b>   <b>vlan</b> <b>vlan-id</b> } * <b>chassis</b> <b>chassis-number</b> <b>slot</b> <b>slot-number</b> [ <b>cpu</b> <b>cpu-number</b> ] |
| 显示成员端口上链路聚合的详细信息     | <b>display link-aggregation member-port</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |



| 操作                  | 命令                                                                                                                                                                   |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     | [ <i>interface-list</i>   <b>auto</b> ]                                                                                                                              |
| 显示所有聚合组的摘要信息        | <b>display link-aggregation summary</b>                                                                                                                              |
| 显示聚合组成员端口的选中状态及原因   | <b>display link-aggregation troubleshooting</b><br>[ { <b>bridge-aggregation</b>   <b>route-aggregation</b>   <b>schannel-bundle</b> } [ <i>interface-number</i> ] ] |
| 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><br>[ { <b>blade-aggregation</b>   <b>bridge-aggregation</b>   <b>route-aggregation</b> } [ <i>interface-number</i> ] ]       |
| 清除聚合接口上的统计信息        | <b>reset counters interface</b> [ { <b>blade-aggregation</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> ]                                                                                              |

## 1.12 以太网链路聚合典型配置举例

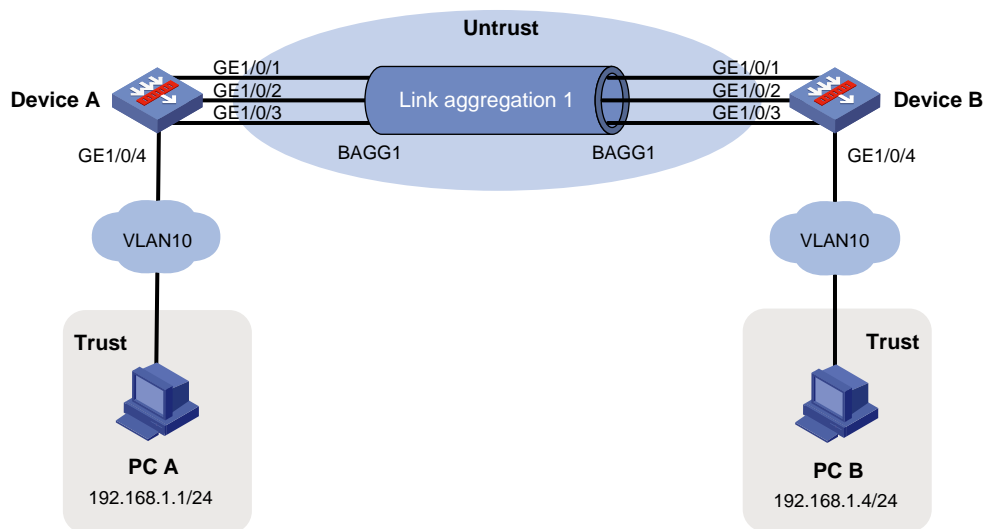
### 1.12.1 二层静态聚合配置举例

#### 1. 组网需求

通过将 Device A 与 Device B 之间的链路进行聚合，有利于设备间通信链路的相互备份和增加通信带宽。

#### 2. 组网图

图1-5 二层静态聚合配置组网图



#### 3. 配置步骤

Device B 的配置与 Device A 相似，下面仅以 Device A 为例。

##### (1) 配置以太网接口的工作模式

# 使接口 GigabitEthernet1/0/1 至 GigabitEthernet1/0/4 工作在二层模式下。

```

<DeviceA> system-view
[DeviceA] interface range gigabitethernet 1/0/1 to gigabitethernet 1/0/4
[DeviceA-if-range] port link-mode bridge
[DeviceA-if-range] quit

```

(2) 配置 VLAN

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

```

[DeviceA] vlan 10
[DeviceA-vlan10] port gigabitethernet 1/0/4
[DeviceA-vlan10] quit

```

(3) 配置二层静态聚合

# 创建二层聚合接口 1 为 Trunk 端口，并允许 VLAN 1 和 10 的报文通过。

```

[DeviceA] interface bridge-aggregation 1
[DeviceA-Bridge-Aggregation1] port link-type trunk
[DeviceA-Bridge-Aggregation1] port trunk permit vlan 1 10
[DeviceA-Bridge-Aggregation1] quit

```

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

```

[DeviceA] interface range gigabitethernet 1/0/1 to gigabitethernet 1/0/3
[DeviceA-if-range] port link-aggregation group 1
[DeviceA-if-range] quit

```

(4) 配置以太网接口的链路类型

# 配置 GigabitEthernet1/0/1 至 GigabitEthernet1/0/3 为 Trunk 端口，并允许 VLAN 1 和 VLAN 10 的报文通过。

```

[DeviceA] interface range gigabitethernet 1/0/1 to gigabitethernet 1/0/3
[DeviceA-if-range] port link-type trunk
[DeviceA-if-range] port trunk permit vlan 1 10
[DeviceA-if-range] quit

```

(5) 配置接口加入安全域

# 请根据组网图中规划的信息，将接口加入对应的安全域，具体配置步骤如下。

```

[DeviceA] security-zone name trust
[Device-security-zone-Trust] import interface gigabitethernet 1/0/4 vlan 10
[Device-security-zone-Trust] quit
[DeviceA] security-zone name untrust
[Device-security-zone-Untrust] import interface bridge-aggregation 1 vlan 1 10
[Device-security-zone-Untrust] quit

```

(6) 配置安全策略

配置安全策略放行 trust 与 untrust 安全域之间的流量，用于设备之间的互通。

# 配置名称为 trust-untrust 的安全策略，使 trust 的安全域到 untrust 的安全域的报文可通。

```

[DeviceA] security-policy ip

```

```
[DeviceA-security-policy-ip] rule name trust-untrust
[DeviceA-security-policy-ip-0-trust-untrust] action pass
[DeviceA-security-policy-ip-0-trust-untrust] source-zone trust
[DeviceA-security-policy-ip-0-trust-untrust] destination-zone untrust
[DeviceA-security-policy-ip-0-trust-untrust] quit
[DeviceA-security-policy-ip] quit
```

# 配置名称为 untrust-trust 的安全策略，使 untrust 的安全域到 trust 的安全域的报文可通。

```
[DeviceA] security-policy ip
[DeviceA-security-policy-ip] rule name untrust-trust
[DeviceA-security-policy-ip-1-untrust-trust] action pass
[DeviceA-security-policy-ip-1-untrust-trust] source-zone untrust
[DeviceA-security-policy-ip-1-untrust-trust] destination-zone trust
[DeviceA-security-policy-ip-1-untrust-trust] quit
[DeviceA-security-policy-ip] quit
```

#### 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
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: Shar

  Port                Status  Priority Oper-Key
  -----
  GE1/0/1              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 为负载分担类型的二层静态聚合组，包含有三个选中端口。

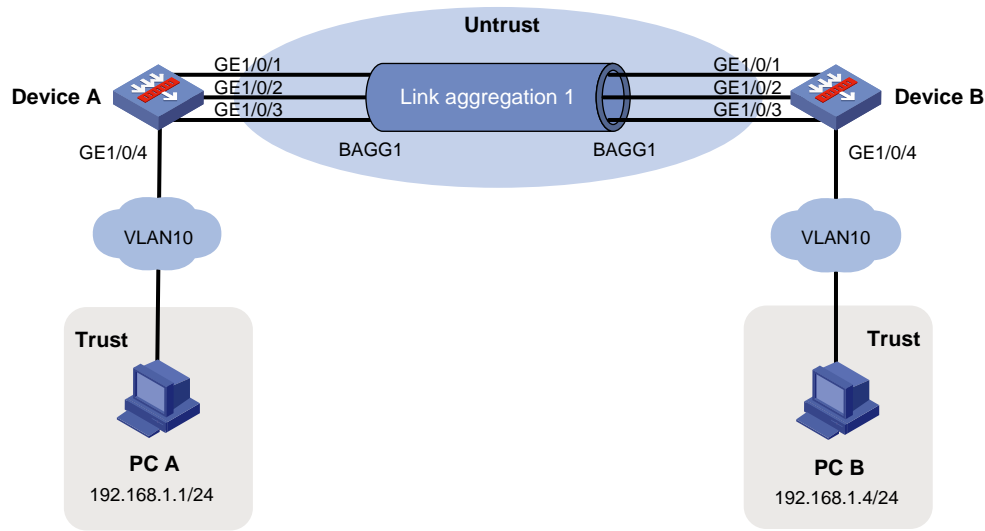
### 1.12.2 二层动态聚合配置举例

#### 1. 组网需求

通过将 Device A 与 Device B 之间的链路进行聚合，有利于设备间通信链路的相互备份和增加通信带宽。

## 2. 组网图

图1-6 二层动态聚合配置组网图



## 3. 配置步骤

Device B 的配置与 Device A 相似，下面仅以 Device A 为例。

### (1) 配置以太网接口的工作模式

# 使接口 GigabitEthernet1/0/1 至 GigabitEthernet1/0/4 工作在二层模式下。

```
<DeviceA> system-view
[DeviceA] interface range gigabitethernet 1/0/1 to gigabitethernet 1/0/4
[DeviceA-if-range] port link-mode bridge
[DeviceA-if-range] quit
```

### (2) 配置 VLAN

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

```
[DeviceA] vlan 10
[DeviceA-vlan10] port gigabitethernet 1/0/4
[DeviceA-vlan10] quit
```

### (3) 配置二层动态聚合

# 创建二层聚合接口 1 为 Trunk 端口，允许 VLAN 1 和 10 的报文通过，并配置该接口为动态聚合模式。

```
[DeviceA] interface bridge-aggregation 1
[DeviceA-Bridge-Aggregation1] port link-type trunk
[DeviceA-Bridge-Aggregation1] port trunk permit vlan 1 10
[DeviceA-Bridge-Aggregation1] link-aggregation mode dynamic
[DeviceA-Bridge-Aggregation1] quit
```

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

```
[DeviceA] interface range gigabitethernet 1/0/1 to gigabitethernet 1/0/3
```

```
[DeviceA-if-range] port link-aggregation group 1
[DeviceA-if-range] quit
```

(4) 配置以太网接口的链路类型

# 配置 GigabitEthernet1/0/1 至 GigabitEthernet1/0/3 为 Trunk 端口，并允许 VLAN 1 和 VLAN 10 的报文通过。

```
[DeviceA] interface range gigabitethernet 1/0/1 to gigabitethernet 1/0/3
[DeviceA-if-range] port link-type trunk
[DeviceA-if-range] port trunk permit vlan 1 10
[DeviceA-if-range] quit
```

(5) 配置接口加入安全域

# 请根据组网图中规划的信息，将接口加入对应的安全域，具体配置步骤如下。

```
[DeviceA] security-zone name trust
[Device-security-zone-Trust] import interface gigabitethernet 1/0/4 vlan 10
[Device-security-zone-Trust] quit
[DeviceA] security-zone name untrust
[Device-security-zone-Untrust] import interface bridge-aggregation 1 vlan 1 10
[Device-security-zone-Untrust] quit
```

(6) 配置安全策略

配置安全策略放行 trust 与 untrust 安全域之间的流量，用于设备之间的互通。

# 配置名称为 trust-untrust 的安全策略，使 trust 的安全域到 untrust 的安全域的报文可通。

```
[DeviceA] security-policy ip
[DeviceA-security-policy-ip] rule name trust-untrust
[DeviceA-security-policy-ip-0-trust-untrust] action pass
[DeviceA-security-policy-ip-0-trust-untrust] source-zone trust
[DeviceA-security-policy-ip-0-trust-untrust] destination-zone untrust
[DeviceA-security-policy-ip-0-trust-untrust] quit
[DeviceA-security-policy-ip] quit
```

# 配置名称为 untrust-trust 的安全策略，使 untrust 的安全域到 trust 的安全域的报文可通。

```
[DeviceA] security-policy ip
[DeviceA-security-policy-ip] rule name untrust-trust
[DeviceA-security-policy-ip-1-untrust-trust] action pass
[DeviceA-security-policy-ip-1-untrust-trust] source-zone untrust
[DeviceA-security-policy-ip-1-untrust-trust] destination-zone trust
[DeviceA-security-policy-ip-1-untrust-trust] quit
[DeviceA-security-policy-ip] quit
```

## 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  
 Flags: A -- LACP\_Activity, B -- LACP\_Timeout, C -- Aggregation,  
 D -- Synchronization, E -- Collecting, F -- Distributing,  
 G -- Defaulted, H -- Expired

Aggregate Interface: Bridge-Aggregation1

Creation Mode: Manual

Aggregation Mode: Dynamic

Loadsharing Type: Shar

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

Local:

| Port    | Status | Priority | Oper-Key | Flag    |
|---------|--------|----------|----------|---------|
| GE1/0/1 | S      | 32768    | 1        | {ACDEF} |
| GE1/0/2 | S      | 32768    | 1        | {ACDEF} |
| GE1/0/3 | S      | 32768    | 1        | {ACDEF} |

Remote:

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

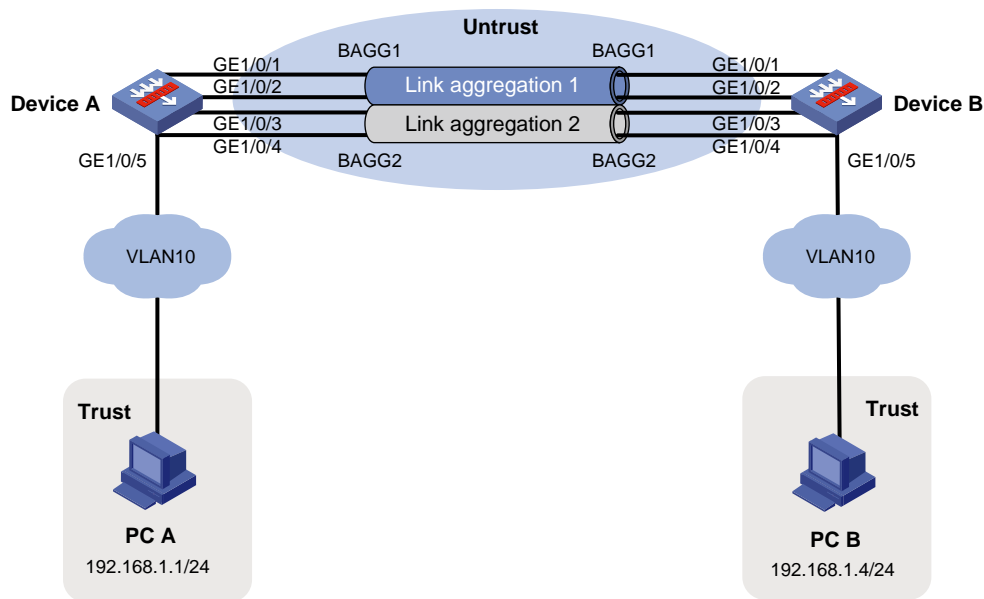
以上信息表明，聚合组 1 为负载分担类型的二层动态聚合组，包含有三个选中端口。

### 1.12.3 二层聚合负载分担配置举例

#### 1. 组网需求

通过将 Device A 与 Device B 之间的链路进行聚合，有利于设备间通信链路的相互备份和增加通信带宽。通过在不同聚合组内配置不同的负载分担方式，来实现数据流量在各成员端口间的负载分担。

#### 2. 组网图



### 3. 配置步骤

Device B 的配置与 Device A 相似，下面仅以 Device A 为例。

#### (1) 配置以太网接口的工作模式

# 使接口 GigabitEthernet1/0/1 至 GigabitEthernet1/0/5 工作在二层模式下。

```
<DeviceA> system-view
[DeviceA] interface range gigabitethernet 1/0/1 to gigabitethernet 1/0/5
[DeviceA-if-range] port link-mode bridge
[DeviceA-if-range] quit
```

#### (2) 配置 VLAN

# 创建 VLAN 10，并将端口 GigabitEthernet1/0/5 加入到 VLAN 10 中。

```
[DeviceA] vlan 10
[DeviceA-vlan10] port gigabitethernet 1/0/5
[DeviceA-vlan10] quit
```

#### (3) 配置二层聚合组

# 创建二层聚合接口 1 为 Trunk 端口，允许 VLAN 1 和 10 的报文通过，并配置该接口对应的聚合组内按照源 MAC 地址进行聚合负载分担。

```
[DeviceA] interface bridge-aggregation 1
[DeviceA-Bridge-Aggregation1] port link-type trunk
[DeviceA-Bridge-Aggregation1] port trunk permit vlan 1 10
[DeviceA-Bridge-Aggregation1] link-aggregation load-sharing mode source-mac
[DeviceA-Bridge-Aggregation1] quit
```

# 将端口 GigabitEthernet1/0/1 至 GigabitEthernet1/0/2 加入到聚合组 1 中。

```
[DeviceA] interface range GigabitEthernet1/0/1 to GigabitEthernet1/0/2
[DeviceA-if-range] port link-aggregation group 1
```

```
[DeviceA-if-range] quit
```

# 创建二层聚合接口 2 为 Trunk 端口，允许 VLAN 1 和 10 的报文通过，并配置该接口对应的聚合组内按照目的 MAC 地址进行聚合负载分担。

```
[DeviceA] interface bridge-aggregation 2
```

```
[DeviceA-Bridge-Aggregation1] port link-type trunk
```

```
[DeviceA-Bridge-Aggregation1] port trunk permit vlan 1 10
```

```
[DeviceA-Bridge-Aggregation1] link-aggregation load-sharing mode destination-mac
```

```
[DeviceA-Bridge-Aggregation1] quit
```

# 将端口 GigabitEthernet1/0/3 至 GigabitEthernet1/0/4 加入到聚合组 2 中。

```
[DeviceA] interface range gigabitethernet 1/0/3to gigabitethernet 1/0/4
```

```
[DeviceA-if-range] port link-aggregation group 2
```

```
[DeviceA-if-range] quit
```

#### (4) 配置以太网接口的链路类型

# 配置 GigabitEthernet1/0/1 至 GigabitEthernet1/0/4 为 Trunk 端口，并允许 VLAN 1 和 VLAN 10 的报文通过。

```
[DeviceA] interface range gigabitethernet 1/0/1 to gigabitethernet 1/0/4
```

```
[DeviceA-if-range] port link-type trunk
```

```
[DeviceA-if-range] port trunk permit vlan 1 10
```

```
[DeviceA-if-range] quit
```

#### (5) 配置接口加入安全域

# 请根据组网图中规划的信息，将接口加入对应的安全域，具体配置步骤如下。

```
[DeviceA] security-zone name trust
```

```
[Device-security-zone-Trust] import interface gigabitethernet 1/0/5 vlan 10
```

```
[Device-security-zone-Trust] quit
```

```
[DeviceA] security-zone name untrust
```

```
[Device-security-zone-Untrust] import interface bridge-aggregation 1 vlan 1 10
```

```
[Device-security-zone-Untrust] import interface bridge-aggregation 2 vlan 1 10
```

```
[Device-security-zone-Untrust] quit
```

#### (6) 配置安全策略

配置安全策略放行 trust 与 untrust 安全域之间的流量，用于设备之间的互通。

# 配置名称为 trust-untrust 的安全策略，使 trust 的安全域到 untrust 的安全域的报文可通。

```
[DeviceA] security-policy ip
```

```
[DeviceA-security-policy-ip] rule name trust-untrust
```

```
[DeviceA-security-policy-ip-0-trust-untrust] action pass
```

```
[DeviceA-security-policy-ip-0-trust-untrust] source-zone trust
```

```
[DeviceA-security-policy-ip-0-trust-untrust] destination-zone untrust
```

```
[DeviceA-security-policy-ip-0-trust-untrust] quit
```

```
[DeviceA-security-policy-ip] quit
```



# 配置名称为 untrust-trust 的安全策略，使 untrust 的安全域到 trust 的安全域的报文可通。

```
[DeviceA] security-policy ip
[DeviceA-security-policy-ip] rule name untrust-trust
[DeviceA-security-policy-ip-1-untrust-trust] action pass
[DeviceA-security-policy-ip-1-untrust-trust] source-zone untrust
[DeviceA-security-policy-ip-1-untrust-trust] destination-zone trust
[DeviceA-security-policy-ip-1-untrust-trust] quit
[DeviceA-security-policy-ip] quit
```

#### 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
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: Shar

  Port           Status  Priority Oper-Key
  -----
  GE1/0/1         S       32768    1
  GE1/0/2         S       32768    1

Aggregate Interface: Bridge-Aggregation2
Aggregation Mode: Static
Loadsharing Type: Shar

  Port           Status  Priority Oper-Key
  -----
  GE1/0/3         S       32768    2
  GE1/0/4         S       32768    2
```

以上信息表明，聚合组 1 和聚合组 2 都是负载分担类型的二层静态聚合组，各包含有两个选中端口。

# 查看 Device A 上所有聚合接口所对应聚合组内采用的聚合负载分担类型。

```
[DeviceA] display link-aggregation load-sharing mode interface

Bridge-Aggregation1 Load-Sharing Mode:
source-mac address
```

```
Bridge-Aggregation2 Load-Sharing Mode:
destination-mac address
```

以上信息表明，二层聚合组 1 按照报文的源 MAC 地址进行聚合负载分担，二层聚合组 2 按照报文的的目的 MAC 地址进行聚合负载分担。

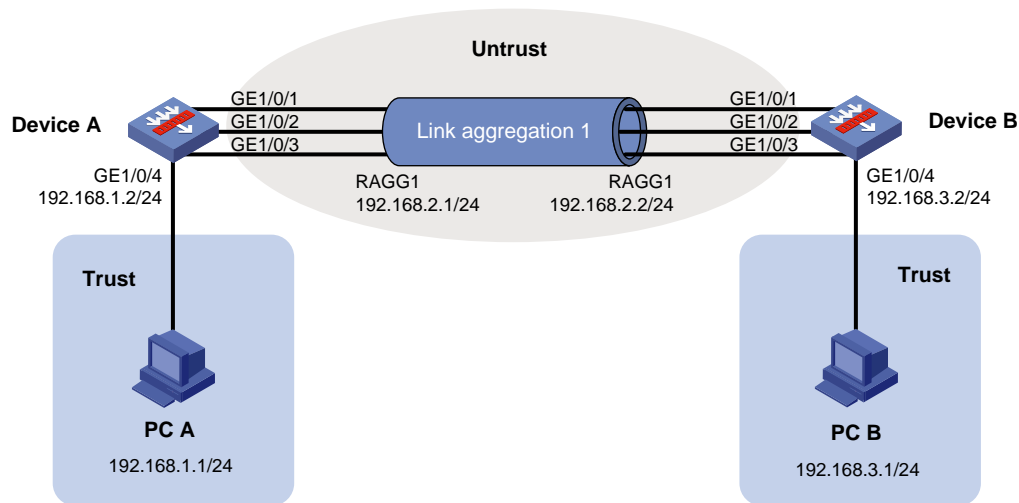
### 1.12.4 三层静态聚合配置举例

#### 1. 组网需求

通过将 Device A 与 Device B 之间的链路进行聚合，有利于设备间通信链路的相互备份和增加通信带宽。

#### 2. 组网图

图1-7 三层静态聚合配置组网图



#### 3. 配置步骤

Device B 的配置与 Device A 相似，下面仅以 Device A 为例。

##### (1) 配置三层静态聚合

# 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 192.168.2.1 24
[DeviceA-Route-Aggregation1] quit
```

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

```
[DeviceA] interface range gigabitethernet 1/0/1 to gigabitethernet 1/0/3
[DeviceA-if-range] port link-aggregation group 1
[DeviceA-if-range] quit
```

##### (2) 配置接口信息

# 配置 GigabitEthernet1/0/4 的 IP 地址为 192.168.1.2/24。

```
[DeviceA] interface gigabitethernet 1/0/4
[DeviceA-GigabitEthernet1/0/4] ip address 192.168.1.2 24
[DeviceA-GigabitEthernet1/0/4] quit
```

(3) 配置静态路由

# 请根据组网图中规划的信息，配置静态路由，本举例假设到达 PC B 的下一跳 IP 地址为 192.168.2.2/24，PC B 的 IP 地址为 192.168.3.1/24，实际使用中请以具体组网情况为准，具体配置步骤如下。

```
[DeviceA] ip route-static 192.168.3.1 24 192.168.2.2
```

(4) 配置接口加入安全域

# 请根据组网图中规划的信息，将接口加入对应的安全域，具体配置步骤如下。

```
[DeviceA] security-zone name trust
[Device-security-zone-Trust] import interface gigabitethernet 1/0/4
[Device-security-zone-Trust] quit
[DeviceA] security-zone name untrust
[Device-security-zone-Untrust] import interface route-aggregation 1
[Device-security-zone-Untrust] quit
```

(5) 配置安全策略

配置安全策略放行 PC A 与 PC B 之间的流量。

# 配置名称为 trust-untrust 的安全策略，使 trust 的安全域到 untrust 的安全域的报文可通。

```
[DeviceA] security-policy ip
[DeviceA-security-policy-ip] rule name trust-untrust
[DeviceA-security-policy-ip-0-trust-untrust] action pass
[DeviceA-security-policy-ip-0-trust-untrust] source-zone trust
[DeviceA-security-policy-ip-0-trust-untrust] destination-zone untrust
[DeviceA-security-policy-ip-0-trust-untrust] source-ip-subnet 192.168.1.0 24
[DeviceA-security-policy-ip-0-trust-untrust] destination-ip-subnet 192.168.3.0 24
[DeviceA-security-policy-ip-0-trust-untrust] quit
[DeviceA-security-policy-ip] quit
```

# 配置名称为 untrust-trust 的安全策略，使 untrust 的安全域到 trust 的安全域的报文可通。

```
[DeviceA] security-policy ip
[DeviceA-security-policy-ip] rule name untrust-trust
[DeviceA-security-policy-ip-1-untrust-trust] action pass
[DeviceA-security-policy-ip-1-untrust-trust] source-zone untrust
[DeviceA-security-policy-ip-1-untrust-trust] destination-zone trust
[DeviceA-security-policy-ip-1-untrust-trust] source-ip-subnet 192.168.3.0 24
[DeviceA-security-policy-ip-1-untrust-trust] destination-ip-subnet 192.168.1.0 24
[DeviceA-security-policy-ip-1-untrust-trust] quit
[DeviceA-security-policy-ip] quit
```

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

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: Shar

  Port           Status  Priority Oper-Key
-----
GE1/0/1          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 为负载分担类型的三层静态聚合组，包含有三个选中端口。

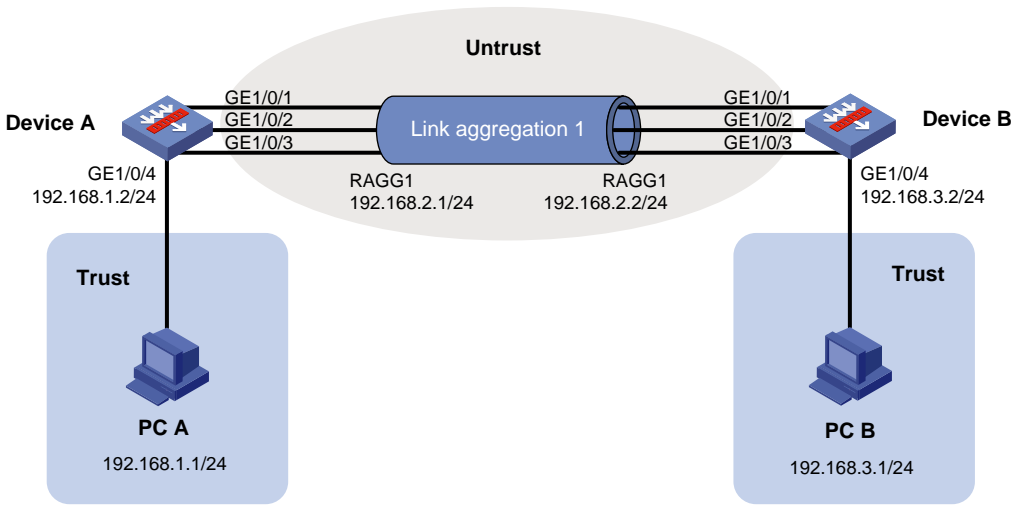
1.12.5 三层动态聚合配置举例

1. 组网需求

通过将 Device A 与 Device B 之间的链路进行聚合，有利于设备间通信链路的相互备份和增加通信带宽。

2. 组网图

图1-8 三层动态聚合配置组网图



### 3. 配置步骤

Device B 的配置与 Device A 相似，下面仅以 Device A 为例。

#### (1) 配置三层动态聚合

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

```
<DeviceA> system-view
[DeviceA] interface route-aggregation 1
[DeviceA-Route-Aggregation1] ip address 192.168.2.1 24
[DeviceA-Route-Aggregation1] link-aggregation mode dynamic
[DeviceA-Route-Aggregation1] quit
```

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

```
[DeviceA] interface range gigabitethernet 1/0/1 to gigabitethernet 1/0/3
[DeviceA-if-range] port link-aggregation group 1
[DeviceA-if-range] quit
```

#### (2) 配置接口信息

# 配置 GigabitEthernet1/0/4 的 IP 地址为 192.168.1.2/24。

```
[DeviceA] interface gigabitethernet 1/0/4
[DeviceA-GigabitEthernet1/0/4] ip address 192.168.1.2 24
[DeviceA-GigabitEthernet1/0/4] quit
```

# 请根据组网图中规划的信息，配置静态路由，本举例假设到达 PC B 的下一跳 IP 地址为 192.168.2.2/24，PC B 的 IP 地址为 192.168.3.1/24，实际使用中请以具体组网情况为准，具体配置步骤如下。

```
[DeviceA] ip route-static 192.168.3.1 24 192.168.2.2
```

#### (3) 配置接口加入安全域

# 请根据组网图中规划的信息，将接口加入对应的安全域，具体配置步骤如下。

```
[DeviceA] security-zone name trust
[Device-security-zone-Trust] import interface gigabitethernet 1/0/4
[Device-security-zone-Trust] quit
[DeviceA] security-zone name untrust
[Device-security-zone-Untrust] import interface route-aggregation 1
[Device-security-zone-Untrust] quit
```

#### (4) 配置安全策略

配置安全策略放行 PC A 与 PC B 之间的流量。

# 配置名称为 trust-untrust 的安全策略，使 trust 的安全域到 untrust 的安全域的报文可通。

```
[DeviceA] security-policy ip
[DeviceA-security-policy-ip] rule name trust-untrust
[DeviceA-security-policy-ip-0-trust-untrust] action pass
[DeviceA-security-policy-ip-0-trust-untrust] source-zone trust
[DeviceA-security-policy-ip-0-trust-untrust] destination-zone untrust
```

```
[DeviceA-security-policy-ip-0-trust-untrust] source-ip-subnet 192.168.1.0 24
[DeviceA-security-policy-ip-0-trust-untrust] destination-ip-subnet 192.168.3.0 24
[DeviceA-security-policy-ip-0-trust-untrust] quit
[DeviceA-security-policy-ip] quit
```

# 配置名称为 untrust-trust 的安全策略，使 untrust 的安全域到 trust 的安全域的报文可通。

```
[DeviceA] security-policy ip
[DeviceA-security-policy-ip] rule name untrust-trust
[DeviceA-security-policy-ip-1-untrust-trust] action pass
[DeviceA-security-policy-ip-1-untrust-trust] source-zone untrust
[DeviceA-security-policy-ip-1-untrust-trust] destination-zone trust
[DeviceA-security-policy-ip-1-untrust-trust] source-ip-subnet 192.168.3.0 24
[DeviceA-security-policy-ip-1-untrust-trust] destination-ip-subnet 192.168.1.0 24
[DeviceA-security-policy-ip-1-untrust-trust] quit
[DeviceA-security-policy-ip] quit
```

#### 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
Flags: A -- LACP_Activity, B -- LACP_Timeout, C -- Aggregation,
       D -- Synchronization, E -- Collecting, F -- Distributing,
       G -- Defaulted, H -- Expired
```

Aggregate Interface: Route-Aggregation1

Creation Mode: Manual

Aggregation Mode: Dynamic

Loadsharing Type: Shar

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

Local:

| Port    | Status | Priority | Oper-Key | Flag    |
|---------|--------|----------|----------|---------|
| GE1/0/1 | S      | 32768    | 1        | {ACDEF} |
| GE1/0/2 | S      | 32768    | 1        | {ACDEF} |
| GE1/0/3 | S      | 32768    | 1        | {ACDEF} |

Remote:

| Actor | Partner | Priority | Oper-Key | SystemID | Flag |
|-------|---------|----------|----------|----------|------|
| ----- |         |          |          |          |      |

|         |   |       |   |                                |
|---------|---|-------|---|--------------------------------|
| GE1/0/1 | 1 | 32768 | 1 | 0x8000, 000f-e267-57ad {ACDEF} |
| GE1/0/2 | 2 | 32768 | 1 | 0x8000, 000f-e267-57ad {ACDEF} |
| GE1/0/3 | 3 | 32768 | 1 | 0x8000, 000f-e267-57ad {ACDEF} |

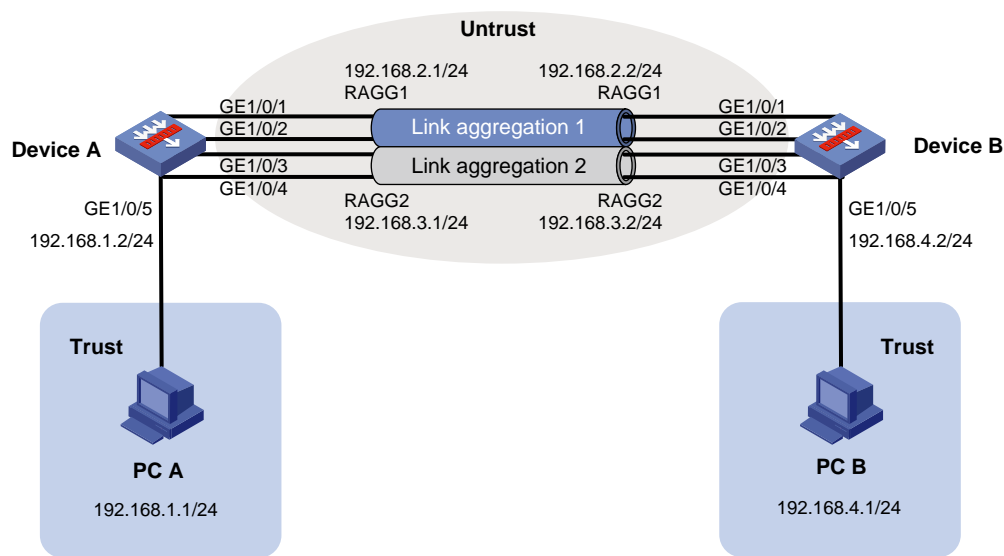
以上信息表明，聚合组 1 为负载分担类型的三层动态聚合组，包含有三个选中端口。

## 1.12.6 三层聚合负载分担配置举例

### 1. 组网需求

通过将 Device A 与 Device B 之间的链路进行聚合，有利于设备间通信链路的相互备份和增加通信带宽。通过在不同聚合组内配置不同的负载分担方式，来实现数据流量在各成员端口间的负载分担。

### 2. 组网图



### 3. 配置步骤

Device B 的配置与 Device A 相似，下面仅以 Device A 为例。

#### (1) 配置三层聚合组

# 创建三层聚合接口 1，配置该接口对应的聚合组内按照源 IP 地址进行聚合负载分担，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 load-sharing mode source-ip
[DeviceA-Route-Aggregation1] ip address 192.168.2.1 24
[DeviceA-Route-Aggregation1] quit
```

# 将端口 GigabitEthernet1/0/1 至 GigabitEthernet1/0/2 加入到聚合组 1 中。

```
[DeviceA] interface range gigabitethernet 1/0/1 to gigabitethernet 1/0/2
[DeviceA-if-range] port link-aggregation group 1
[DeviceA-if-range] quit
```

# 创建三层聚合接口 2，配置该接口对应的聚合组内按照目的 IP 地址进行聚合负载分担，并为其配置 IP 地址和子网掩码。

```
[DeviceA] interface route-aggregation 2
[DeviceA-Route-Aggregation2] link-aggregation load-sharing mode destination-ip
[DeviceA-Route-Aggregation2] ip address 192.168.3.1 24
[DeviceA-Route-Aggregation2] quit
```

# 将端口 GigabitEthernet1/0/3 至 GigabitEthernet1/0/4 加入到聚合组 2 中。

```
[DeviceA] interface range gigabitethernet 1/0/3 to gigabitethernet 1/0/4
[DeviceA-if-range] port link-aggregation group 2
[DeviceA-if-range] quit
```

## (2) 配置接口信息

# 配置 GigabitEthernet1/0/5 的 IP 地址为 192.168.1.2/24。

```
[DeviceA] interface gigabitethernet 1/0/5
[DeviceA-GigabitEthernet1/0/5] ip address 192.168.1.2 24
[DeviceA-GigabitEthernet1/0/5] quit
```

# 请根据组网图中规划的信息，配置静态路由，本举例假设到达 PC B 的下一跳 IP 地址为 192.168.2.2/24 和 192.168.3.2/24，PC B 的 IP 地址为 192.168.4.1/24，实际使用中请以具体组网情况为准，具体配置步骤如下。

```
[DeviceA] ip route-static 192.168.4.1 24 192.168.2.2
[DeviceA] ip route-static 192.168.4.1 24 192.168.3.2
```

## (3) 配置接口加入安全域

# 请根据组网图中规划的信息，将接口加入对应的安全域，具体配置步骤如下。

```
[DeviceA] security-zone name trust
[Device-security-zone-Trust] import interface gigabitethernet 1/0/5
[Device-security-zone-Trust] quit
[DeviceA] security-zone name untrust
[Device-security-zone-Untrust] import interface route-aggregation 1
[Device-security-zone-Untrust] import interface route-aggregation 2
[Device-security-zone-Untrust] quit
```

## (4) 配置安全策略

配置安全策略放行 PC A 与 PC B 之间的流量。

# 配置名称为 trust-untrust 的安全策略，使 trust 的安全域到 untrust 的安全域的报文可通。

```
[DeviceA] security-policy ip
[DeviceA-security-policy-ip] rule name trust-untrust
[DeviceA-security-policy-ip-0-trust-untrust] action pass
[DeviceA-security-policy-ip-0-trust-untrust] source-zone trust
[DeviceA-security-policy-ip-0-trust-untrust] destination-zone untrust
[DeviceA-security-policy-ip-0-trust-untrust] source-ip-subnet 192.168.1.0 24
```



```
[DeviceA-security-policy-ip-0-trust-untrust] destination-ip-subnet 192.168.4.0 24
[DeviceA-security-policy-ip-0-trust-untrust] quit
[DeviceA-security-policy-ip] quit
```

# 配置名称为 untrust-trust 的安全策略，使 untrust 的安全域到 trust 的安全域的报文可通。

```
[DeviceA] security-policy ip
[DeviceA-security-policy-ip] rule name untrust-trust
[DeviceA-security-policy-ip-1-untrust-trust] action pass
[DeviceA-security-policy-ip-1-untrust-trust] source-zone untrust
[DeviceA-security-policy-ip-1-untrust-trust] destination-zone trust
[DeviceA-security-policy-ip-1-untrust-trust] source-ip-subnet 192.168.4.0 24
[DeviceA-security-policy-ip-1-untrust-trust] destination-ip-subnet 192.168.1.0 24
[DeviceA-security-policy-ip-1-untrust-trust] quit
[DeviceA-security-policy-ip] quit
```

#### 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
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: Shar

| Port    | Status | Priority | Oper-Key |
|---------|--------|----------|----------|
| GE1/0/1 | S      | 32768    | 1        |
| GE1/0/2 | S      | 32768    | 1        |

Aggregate Interface: Route-Aggregation2

Aggregation Mode: Static

Loadsharing Type: Shar

| Port    | Status | Priority | Oper-Key |
|---------|--------|----------|----------|
| GE1/0/3 | S      | 32768    | 2        |
| GE1/0/4 | S      | 32768    | 2        |

以上信息表明,聚合组 1 和聚合组 2 都是负载分担类型的三层静态聚合组,各包含有两个选中端口。

# 查看 Device A 上所有聚合接口所对应聚合组内采用的聚合负载分担类型。

```
[DeviceA] display link-aggregation load-sharing mode interface
```

```
Route-Aggregation1 Load-Sharing Mode:
```

```
source-ip address
```

```
Route-Aggregation2 Load-Sharing Mode:
```

```
destination-ip address
```

以上信息表明,三层聚合组 1 按照报文的源 IP 地址进行聚合负载分担,三层聚合组 2 按照报文的目的 IP 地址进行聚合负载分担。

# 目 录

|                                  |     |
|----------------------------------|-----|
| 1 VLAN .....                     | 1-1 |
| 1.1 VLAN 简介 .....                | 1-1 |
| 1.1.1 VLAN 报文封装 .....            | 1-1 |
| 1.1.2 基于端口的 VLAN .....           | 1-2 |
| 1.1.3 不同 VLAN 间的三层互通 .....       | 1-3 |
| 1.1.4 协议规范 .....                 | 1-3 |
| 1.2 vSystem 相关说明 .....           | 1-3 |
| 1.3 配置 VLAN .....                | 1-4 |
| 1.3.1 配置限制和指导 .....              | 1-4 |
| 1.3.2 创建 VLAN .....              | 1-4 |
| 1.4 配置基于端口的 VLAN .....           | 1-4 |
| 1.4.1 配置限制和指导 .....              | 1-4 |
| 1.4.2 配置基于 Access 端口的 VLAN ..... | 1-5 |
| 1.4.3 配置基于 Trunk 端口的 VLAN .....  | 1-5 |
| 1.4.4 配置基于 Hybrid 端口的 VLAN ..... | 1-6 |
| 1.5 配置 VLAN 组 .....              | 1-7 |
| 1.6 配置 VLAN 接口 .....             | 1-7 |
| 1.6.1 VLAN 接口配置任务简介 .....        | 1-7 |
| 1.6.2 配置准备 .....                 | 1-7 |
| 1.6.3 创建 VLAN 接口 .....           | 1-7 |
| 1.6.4 恢复 VLAN 接口的缺省配置 .....      | 1-8 |
| 1.7 VLAN 显示和维护 .....             | 1-8 |

# 1 VLAN

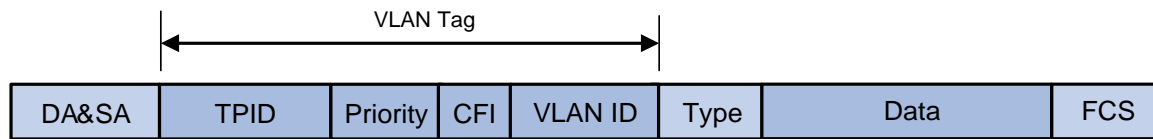
## 1.1 VLAN简介

VLAN（Virtual Local Area Network，虚拟局域网）技术把一个物理 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，从而增强了局域网的安全性。划分 VLAN 后，广播报文被限制在同一个 VLAN 内，即每个 VLAN 是一个广播域，有效地限制了广播域的范围。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。

### 1.1.1 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 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1-1 VLAN Tag 的组成字段



如图 1-1 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“VLAN 终结”。
- **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。



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。

## 1.1.2 基于端口的 VLAN

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。

### 1. 端口的链路类型

端口的链路类型分为三种，端口的链路类型决定了端口能否加入多个 VLAN。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。

### 2. 端口缺省 VLAN

端口缺省 VLAN 简称为 PVID（Port VLAN ID）。当端口收到 Untagged 报文时，会认为该报文所属的 VLAN 为 PVID。

Access 端口的 PVID 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 PVID。

### 3. 端口对报文的处理方式

端口对报文的接收和发送的处理有几种不同情况，具体情况请参看[表 1-1](#)。

表1-1 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PVID的Tag | <ul style="list-style-type: none"><li>• 当报文的 VLAN ID 与端口的 PVID 相同时，接收该报文</li><li>• 当报文的 VLAN ID 与端口的 PVID 不同时，丢弃该报文</li></ul> | 去掉Tag，发送该报文 |

| 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                                                                                                                               |                                                                                                                                                                                      |
| Trunk端口  | <ul style="list-style-type: none"> <li>当端口的 PVID 在端口允许通过的 VLAN ID 列表里时，接收该报文，给报文添加 PVID 的 Tag</li> <li>当端口的 PVID 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 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 PVID 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 PVID 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 |                                                                                                                                                          |                                                                                                                                               | 当报文的 VLAN ID 是端口允许通过的 VLAN ID 时，发送该报文，并可以配置端口在发送该 VLAN 的报文时是否携带 Tag                                                                                                                  |

### 1.1.3 不同 VLAN 间的三层互通

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。

### 1.1.4 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 vSystem相关说明

非缺省 vSystem 支持本特性的部分功能，具体包括：

- 进入 VLAN 接口。
- 配置 VLAN 接口的描述信息。
- 配置 VLAN 接口的期望带宽。



说明

非缺省 vSystem 对具体命令的支持情况，请见本特性的命令参考。有关 vSystem 的详细介绍请参见“虚拟化技术配置指导”中的“vSystem”。

## 1.3 配置VLAN

### 1.3.1 配置限制和指导

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。

### 1.3.2 创建 VLAN

- (1) 进入系统视图。

```
system-view
```

- (2) 创建 VLAN。请至少选择其中一项进行配置。

- 创建一个 VLAN，并进入 VLAN 视图。

```
vlan vlan-id
```

- 批量创建 VLAN，然后进入 VLAN 视图。

```
vlan { vlan-id1 to vlan-id2 | all }
```

```
vlan vlan-id
```

缺省情况下，系统只有一个缺省 VLAN（VLAN 1）。

- (3) （可选）指定 VLAN 的名称。

```
name text
```

缺省情况下，VLAN 的名称为“VLAN *vlan-id*”，其中 *vlan-id* 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”。

- (4) （可选）配置 VLAN 的描述信息。

```
description text
```

缺省情况下，VLAN 的描述信息为“VLAN *vlan-id*”，其中 *vlan-id* 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”。

## 1.4 配置基于端口的VLAN

### 1.4.1 配置限制和指导

- 当执行 **undo vlan** 命令删除的 VLAN 是某个端口的 PVID 时，对 Access 端口，端口的 PVID 会恢复到 VLAN 1；对 Trunk 或 Hybrid 端口，端口的 PVID 配置不会改变，即它们可以使用已经不存在的 VLAN 作为端口 PVID。
- 建议本端设备端口的 PVID 和相连的对端设备端口的 PVID 保持一致。
- 建议保证端口的 PVID 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 PVID 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。

## 1.4.2 配置基于 Access 端口的 VLAN

### 1. 简介

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。

### 2. 在 VLAN 视图下配置基于 Access 端口的 VLAN

- (1) 进入系统视图。

```
system-view
```

- (2) 进入 VLAN 视图。

```
vlan vlan-id
```

- (3) 向当前 VLAN 中添加一个或一组 Access 端口。

```
port interface-list
```

缺省情况下，系统将所有端口都加入到 VLAN 1。

### 3. 在接口视图下配置基于 Access 端口的 VLAN

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

- 进入二层以太网接口视图。

```
interface interface-type interface-number
```

- 进入二层聚合接口视图。

```
interface bridge-aggregation interface-number
```

- (3) 配置端口的链路类型为 Access 类型。

```
port link-type access
```

缺省情况下，端口的链路类型为 Access。

- (4) 将 Access 端口加入到指定 VLAN。

```
port access vlan vlan-id
```

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

### 1. 简介

Trunk 端口可以加入多个 VLAN。基于 Trunk 端口的 VLAN 只能在接口视图下配置。

### 2. 配置限制和指导

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 PVID 后，必须使用 **port trunk permit vlan** 命令配置允许 PVID 的报文通过，接口才能转发 PVID 的报文。



### 3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

- 进入二层以太网接口视图。

```
interface interface-type interface-number
```

- 进入二层聚合接口视图。

```
interface bridge-aggregation interface-number
```

- (3) 配置端口的链路类型为 Trunk 类型。

```
port link-type trunk
```

缺省情况下，端口的链路类型为 Access 类型。

- (4) 允许指定的 VLAN 通过当前 Trunk 端口。

```
port trunk permit vlan { vlan-id-list | all }
```

缺省情况下，Trunk 端口只允许 VLAN 1 的报文通过。

- (5) （可选）配置 Trunk 端口的 PVID。

```
port trunk pvid vlan vlan-id
```

缺省情况下，Trunk 端口的 PVID 为 VLAN 1。

## 1.4.4 配置基于 Hybrid 端口的 VLAN

### 1. 简介

Hybrid 端口可以加入多个 VLAN。基于 Hybrid 端口的 VLAN 只能在接口视图下配置。将 Hybrid 端口加入 VLAN 时，指定 VLAN 必须已经存在。

### 2. 配置限制和指导

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。

配置端口 PVID 后，必须使用 **port hybrid vlan** 命令配置允许 PVID 的报文通过，出接口才能转发 PVID 的报文。

### 3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

- 进入二层以太网接口视图。

```
interface interface-type interface-number
```

- 进入二层聚合接口视图。

```
interface bridge-aggregation interface-number
```

- (3) 配置端口的链路类型为 Hybrid 类型。

```
port link-type hybrid
```

缺省情况下，端口的链路类型为 Access 类型。

- (4) 允许指定的 VLAN 通过当前 Hybrid 端口。

```
port hybrid vlan vlan-id-list { tagged | untagged }
```

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。

- (5) （可选）配置 Hybrid 端口的 PVID。

```
port hybrid pvid vlan vlan-id
```

缺省情况下，Hybrid 端口的 PVID 为该端口在链路类型为 Access 时的所属 VLAN。

## 1.5 配置VLAN组

### 1. 功能简介

VLAN 组是一组 VLAN 的集合。VLAN 组内可以添加多个 VLAN 列表，一个 VLAN 列表表示一组 VLAN ID 连续的 VLAN。

### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 创建一个 VLAN 组，并进入 VLAN 组视图。

```
vlan-group group-name
```

- (3) 在 VLAN 组内添加 VLAN 成员。

```
vlan-list vlan-id-list
```

缺省情况下，当前 VLAN 组中不存在 VLAN 列表。

可以多次在当前 VLAN 组内添加 VLAN 成员。

## 1.6 配置VLAN接口

### 1.6.1 VLAN 接口配置任务简介

VLAN 接口配置任务如下：

- (1) [创建 VLAN 接口](#)  
(2) （可选）[恢复 VLAN 接口的缺省配置](#)

### 1.6.2 配置准备

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。

### 1.6.3 创建 VLAN 接口

- (1) 进入系统视图。

```
system-view
```

- (2) 创建 VLAN 接口，并进入 VLAN 接口视图。

```
interface vlan-interface interface-number
```

- (3) 配置 VLAN 接口的 IP 地址。

**ip address** *ip-address* { *mask* | *mask-length* } [ *sub* ]

缺省情况下，未配置 VLAN 接口的 IP 地址。

- (4) （可选）配置 VLAN 接口的描述信息。

**description** *text*

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”。

- (5) （可选）配置 VLAN 接口的 MTU 值。

**mtu** *size*

缺省情况下，VLAN 接口的 MTU 值为 1500。

- (6) （可选）配置 VLAN 接口的期望带宽。

**bandwidth** *bandwidth-value*

缺省情况下，接口的期望带宽 = 接口的波特率 ÷ 1000（kbps）。

- (7) 取消手工关闭 VLAN 接口。

**undo shutdown**

缺省情况下，未手工关闭 VLAN 接口。

## 1.6.4 恢复 VLAN 接口的缺省配置

### 1. 配置限制和指导

您可以在执行 **default** 命令后通过 **display this** 命令确认执行效果。对于未能成功恢复缺省的配置，建议您查阅相关功能的命令手册，手工执行恢复该配置缺省情况的命令。如果操作仍然不能成功，您可以通过设备的提示信息定位原因。

### 2. 配置步骤

- (1) 进入系统视图。

- (2) **system-view**

- (3) 进入 VLAN 接口视图。

**interface** *vlan-interface* *interface-number*

- (4) 恢复 VLAN 接口的缺省配置。

**default**



注意

接口下的某些配置恢复到缺省情况后，会对设备上当前运行的业务产生影响。建议您在执行该命令前，完全了解其对网络产生的影响。

---

## 1.7 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。



说明

非缺省 vSystem 不支持部分显示和维护命令，具体情况请见本特性的命令参考。

表1-2 VLAN 显示和维护

| 操作                     | 命令                                                                                                                                   |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| 显示VLAN接口相关信息           | <b>display interface</b> [ <b>vlan-interface</b> [ <i>interface-number</i> ] ] [ <b>brief</b> [ <b>description</b>   <b>down</b> ] ] |
| 显示设备上存在的Hybrid或Trunk端口 | <b>display port</b> { <b>hybrid</b>   <b>trunk</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>static</b> ]                |
| 显示设备上所有已创建VLAN的概要信息    | <b>display vlan brief</b>                                                                                                            |
| 显示创建的VLAN组及其VLAN成员列表   | <b>display vlan-group</b> [ <i>group-name</i> ]                                                                                      |
| 清除VLAN接口的统计信息          | <b>reset counters interface</b> [ <b>vlan-interface</b> [ <i>interface-number</i> ] ]                                                |

# 目 录

|                                       |     |
|---------------------------------------|-----|
| 1 VLAN 终结 .....                       | 1-1 |
| 1.1 VLAN 终结简介 .....                   | 1-1 |
| 1.1.1 VLAN 终结分类 .....                 | 1-1 |
| 1.1.2 VLAN 终结工作机制 .....               | 1-1 |
| 1.1.3 VLAN 终结应用场景 .....               | 1-2 |
| 1.2 VLAN 终结配置限制和指导 .....              | 1-4 |
| 1.3 VLAN 终结配置任务简介 .....               | 1-4 |
| 1.4 配置模糊的 Dot1q 终结 .....              | 1-5 |
| 1.5 配置明确的 Dot1q 终结 .....              | 1-5 |
| 1.6 配置模糊的 QinQ 终结 .....               | 1-6 |
| 1.6.1 功能简介 .....                      | 1-6 |
| 1.6.2 指定最外两层 VLAN ID 进行 QinQ 终结 ..... | 1-6 |
| 1.7 配置明确的 QinQ 终结 .....               | 1-7 |
| 1.7.1 功能简介 .....                      | 1-7 |
| 1.7.2 指定最外两层 VLAN ID 进行 QinQ 终结 ..... | 1-7 |
| 1.8 配置 Untagged 终结 .....              | 1-7 |
| 1.9 配置 Default 终结 .....               | 1-8 |
| 1.10 配置 VLAN 终结支持广播/组播 .....          | 1-8 |
| 1.11 配置 VLAN Tag 的 TPID 值 .....       | 1-9 |

# 1 VLAN 终结

## 1.1 VLAN终结简介

VLAN 终结是指对接收到的报文，按照报文携带的 VLAN Tag 信息匹配对应的接口后，去除报文 VLAN Tag，再将报文进行三层转发或交由其他业务处理。转发出去的报文是否带有 VLAN Tag 由出接口决定，对从配置了 VLAN 终结的接口发送的报文，按照该接口上的终结配置，将相应的 VLAN Tag 添加到报文中后发送该报文。

### 1.1.1 VLAN 终结分类

根据对所终结的报文的处理方式，VLAN 终结分为：

- **Dot1q 终结：**用来终结带有一层及以上 VLAN Tag 的报文（要求最外层 VLAN ID 必须匹配配置值），从配置了 Dot1q 终结的接口发送的报文，都添加一层 VLAN Tag。
- **QinQ 终结：**用来终结带有两层及以上 VLAN Tag 的报文（要求最外两层 VLAN ID 必须匹配配置值），从配置了 QinQ 终结的接口发送的报文，都添加两层 VLAN Tag。
- **Untagged 终结：**用来终结收到的不带 VLAN Tag 的报文，从配置了 Untagged 终结的接口发送的报文，都不添加 VLAN Tag。
- **Default 终结：**用来终结同一主接口上其他子接口上无法处理的报文，从配置了 Default 终结的接口发送的报文，都不添加 VLAN Tag。



说明

为便于描述，本特性部分内容对带有两层及以上 VLAN Tag 的报文，将其最外两层 VLAN Tag 按从外层到内层的方向，分别用第一层 VLAN Tag、第二层 VLAN Tag 表示，对 VLAN ID 的描述类似。

### 1.1.2 VLAN 终结工作机制

子接口（例如三层以太网子接口/三层聚合子接口）、VLAN 接口可以终结匹配最外层 VLAN ID 的报文或匹配最外两层 VLAN ID 的报文。其中，VLAN 接口只能终结最外层 VLAN ID 与接口编号相同的 VLAN 报文，例如 Vlan-interface10 只能终结最外层 VLAN ID 为 10 的报文。

主接口（例如三层以太网接口/三层聚合接口）本身不能对 VLAN 报文做终结处理，在主接口创建子接口后，由子接口来处理。

配置 VLAN 终结后，设备对收到的报文按如下优先级顺序匹配接口：

- 配置了带 loose 属性的 QinQ 终结的子接口
- 配置了 Dot1q 终结或者缺省支持 Dot1q 终结的子接口
- 配置了带 loose 属性的 Dot1q 终结的子接口
- 配置了 Untagged 终结的子接口
- 配置了 Default 终结的子接口

- 主接口

当主接口的某个子接口配置了 **Untagged** 终结时，不带 VLAN Tag 的报文只能由主接口下的子接口处理，而不会匹配到主接口。

当主接口的某个子接口配置了 **Default** 终结时，报文只能由主接口下的子接口处理，而不会匹配到主接口。

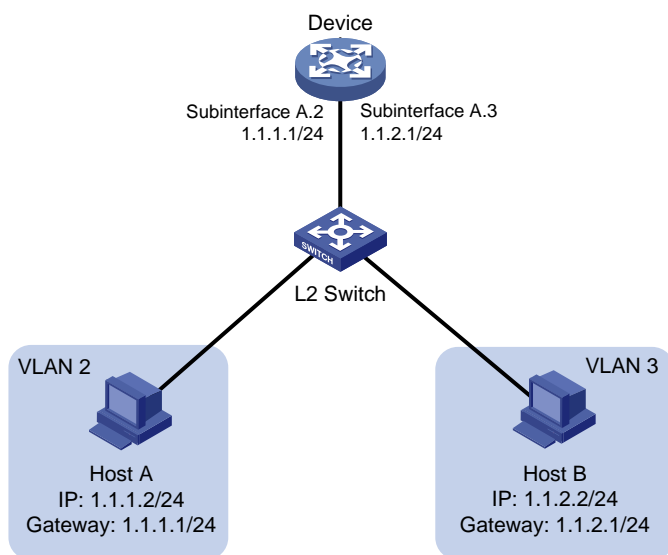
与 VLAN 接口绑定的主接口在收到 VLAN 报文后，根据 VLAN 接口的配置对报文进行处理。

### 1.1.3 VLAN 终结应用场景

#### 1. 指定 VLAN 间的互通

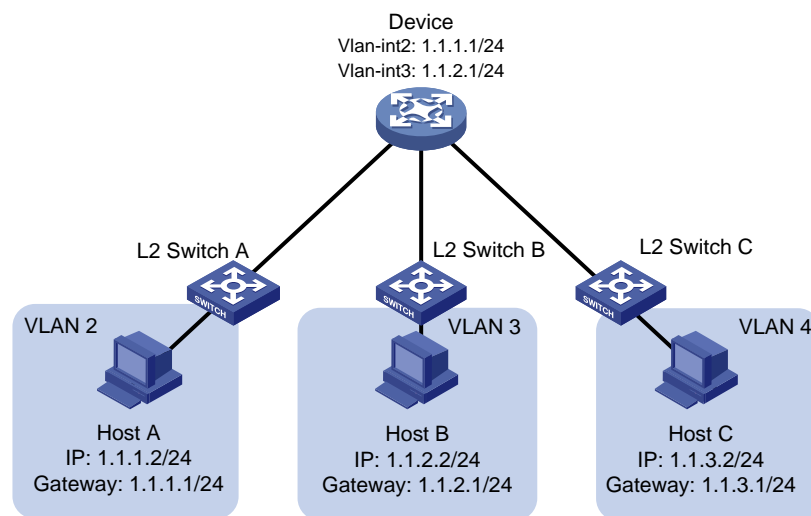
划分 VLAN 后，不同 VLAN 间的主机不能直接通信，使用三层路由技术可以实现所有 VLAN 间报文的互通。此时如果要对互通的 VLAN 范围做限制，即要求只有指定的部分 VLAN 间可以互通，可以借助 VLAN 终结功能来实现。目前可以通过子接口或 VLAN 接口实现指定 VLAN 间的互通。

图1-1 VLAN 终结用于不同 VLAN 之间互通（通过三层以太网子接口）



如[图 1-1](#)所示，Host A 属于 VLAN 2，Host B 属于 VLAN 3。将 Host A 的网关地址指定为 1.1.1.1/24，Host B 的网关地址指定为 1.1.2.1/24，在 Device 上创建三层以太网子接口 Subinterface A.2 和 Subinterface A.3 并配置 Dot1q 终结，可实现 Host A 和 Host B 之间的互通。

图1-2 VLAN 终结用于不同 VLAN 之间互通（通过 VLAN 接口）



如图 1-2 所示，Host A 属于 VLAN 2，Host B 属于 VLAN 3，Host C 属于 VLAN 4。将 Host A 的网关地址指定为 1.1.1.1/24，Host B 的网关地址指定为 1.1.2.1/24，然后在 Device 上创建 Vlan-interface2 和 Vlan-interface3 并分别配置 IP 地址为 Host A 和 Host B 的网关地址，可实现 Host A 和 Host B 的三层报文互通：当 Vlan-interface2 收到 Host A 的 VLAN 2 报文时，去除 VLAN Tag，再转发给 Vlan-interface3，Vlan-interface3 在发送该报文时添加该接口 VLAN 的 Tag（VLAN 3），使该报文能发送到 VLAN 3 内的 Host B；反之亦然。

而 Host C 由于没有对应的 VLAN 接口终结 VLAN 4 的报文，不能与 Host A 或 Host B 互通。



说明

VLAN 接口缺省支持终结 VLAN ID 等于该接口编号的最外层 VLAN Tag。

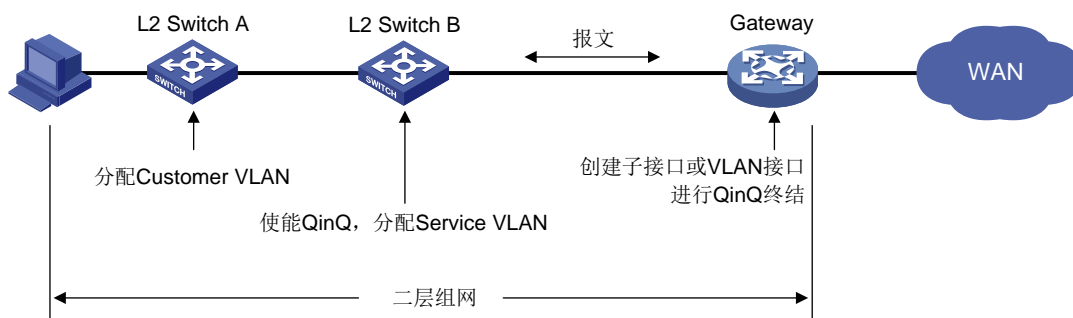
## 2. 局域网和广域网的互联

局域网内的报文大多数都带有 VLAN Tag，但一些广域网协议（例如 FR 和 PPP）并不能识别 VLAN 报文。这种情况下，如果局域网的 VLAN 报文要转发到广域网，需要在本地记录并去掉报文的 VLAN 信息后再转发，可以借助 VLAN 终结功能来实现。目前可以通过子接口或 VLAN 接口实现局域网和广域网的互联。

如下图所示，用户网络的私网 VLAN 为 Customer VLAN，运营商为用户网络分配的公网 VLAN 为 Service VLAN。当用户网内 Customer VLAN 的报文进入运营商网络时，报文外面就会被封装上 Service VLAN 的 VLAN Tag，在运营商网络基于 Service VLAN 进行转发。如果报文要发往外部广域网，则需要在出口网关上对该报文进行 QinQ 终结处理，去掉两层 VLAN Tag，再发送到广域网。



图1-3 VLAN 终结用于 LAN 和 WAN 互联



## 1.2 VLAN终结配置限制和指导



注意

如果要在开启了 Portal 认证功能的接口上更改 VLAN 终结方式，必须先把该接口的在线 Portal 用户全部下线，否则会导致这些在线 Portal 用户无法正常下线也无法重新认证上线。有关 Portal 的介绍，请参见“安全配置指导”中的“Portal”。

在子接口视图下修改已有的 VLAN 终结配置后，该子接口会 down/up 一次，设备 ARP 表中与该子接口相关的动态表项也会被全部删除。

## 1.3 VLAN终结配置任务简介

VLAN 终结配置任务如下：

### (1) 配置 VLAN 终结

请根据实际组网情况选择以下一种方式：

- [配置模糊的 Dot1q 终结](#)
- [配置明确的 Dot1q 终结](#)
- [配置模糊的 QinQ 终结](#)
- [配置明确的 QinQ 终结](#)
- [配置 Untagged 终结](#)
- [配置 Default 终结](#)

### (2) （可选）[配置 VLAN 终结支持广播/组播](#)

执行该任务后，配置了 VLAN 终结功能的接口才能发送广播/组播报文。

### (3) （可选）[配置 VLAN Tag 的 TPID 值](#)

## 1.4 配置模糊的Dot1q终结

### 1. 功能简介

模糊的 Dot1q 终结只允许接口接收最外层 VLAN ID 在指定范围内的 VLAN 报文，不属于该范围的 VLAN 报文则不允许通过该接口。接口收到报文后，将报文最外层 VLAN Tag 剥离。发送报文时，会给报文添加一层 VLAN Tag，VLAN ID 字段取值为：

- 对于 PPPoE 报文，通过查找 PPPoE 会话表项获取相应的 VLAN ID。
- 对于 DHCP Relay 转发的 DHCP Server 端报文，通过查找 DHCP 会话表项获取相应的 VLAN ID。
- 对于 MPLS 和其他 IPv4 报文，通过查找 ARP 表项获取相应的 VLAN ID。

### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

- 进入三层以太网子接口视图。

```
interface interface-type interface-number.subnumber
```

- 进入三层聚合子接口视图。

```
interface route-aggregation interface-number.subnumber
```

- 进入以太网冗余子接口视图。

```
interface reth interface-number.subnumber
```

- (3) 开启子接口的 Dot1q 终结功能，并指定子接口能够终结的 VLAN 报文的最外层 VLAN ID 范围。

```
vlan-type dot1q vid vlan-id-list [ loose ]
```

缺省情况下，子接口的 Dot1q 终结功能处于关闭状态。

## 1.5 配置明确的Dot1q终结

### 1. 功能简介

明确的 Dot1q 终结只允许接口接收最外层 VLAN ID 为指定值的 VLAN 报文，其他 VLAN 报文则不允许通过该接口。接口收到报文后，将报文最外层 VLAN Tag 剥离。发送报文时，给报文添加一层 VLAN Tag，VLAN ID 为指定值。

### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

- 进入三层以太网子接口视图。

```
interface interface-type interface-number.subnumber
```

- 进入三层聚合子接口视图。

```
interface route-aggregation interface-number.subnumber
```

- 进入以太网冗余子接口视图。

```
interface reth interface-number.subnumber
```

- (3) 开启子接口的 Dot1q 终结功能，并指定子接口能够终结的 VLAN 报文最外层 VLAN ID。

```
vlan-type dot1q vid vlan-id [ loose ]
```

缺省情况下，子接口的 Dot1q 终结功能处于关闭状态。

## 1.6 配置模糊的 QinQ 终结

### 1.6.1 功能简介

模糊的 QinQ 终结只允许接口接收最外两层 VLAN ID 均在指定范围内的报文，不属于该范围的 VLAN 报文则不允许通过该接口。接口收到报文后，将报文最外两层 VLAN Tag 剥离。发送报文时，会给报文添加两层 VLAN Tag。添加 VLAN Tag 后，报文的最外两层 VLAN ID 取值为：

- 对于 PPPoE 报文，通过查找 PPPoE 会话表项获取相应的 VLAN ID。
- 对于 DHCP Relay 转发的 DHCP Server 端报文，通过查找 DHCP 中继用户地址表项获取相应的 VLAN ID。

### 1.6.2 指定最外两层 VLAN ID 进行 QinQ 终结

#### 1. 配置限制和指导

- 在同一主接口的不同子接口下配置 QinQ 终结功能时，如果指定的第一层 VLAN ID 相同，则第二层 VLAN ID 必须不同；如果指定的第一层 VLAN ID 不同，则第二层 VLAN ID 可以相同。
- 不同主接口下的子接口可以终结的 VLAN 报文可以相同也可以不同。
- 在同一子接口下多次执行 **vlan-type dot1q vid second-dot1q** 命令时，最终生效的第一层 VLAN ID 和第二层 VLAN ID 是多次配置的集合。

#### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

- 进入三层以太网子接口视图。

```
interface interface-type interface-number.subnumber
```

- 进入三层聚合子接口视图。

```
interface route-aggregation interface-number.subnumber
```

- 进入以太网冗余子接口视图。

```
interface reth interface-number.subnumber
```

- (3) 开启子接口的 QinQ 终结功能，并指定子接口可以终结的 VLAN 报文的最外两层 VLAN ID。

```
vlan-type dot1q vid vlan-id-list second-dot1q { vlan-id-list | any }  
[ loose ]
```

缺省情况下，子接口的 QinQ 终结功能处于关闭状态。

## 1.7 配置明确的QinQ终结

### 1.7.1 功能简介

明确的 QinQ 终结只允许接口接收最外两层 VLAN ID 均为指定值的报文，其他 VLAN 报文则不允许通过该接口。接口收到报文后，将报文最外两层 VLAN Tag 剥离。发送报文时，会给报文添加两层 VLAN Tag，两层 VLAN ID 均为指定值。

### 1.7.2 指定最外两层 VLAN ID 进行 QinQ 终结

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

- 进入三层以太网子接口视图。

```
interface interface-type interface-number.subnumber
```

- 进入三层聚合子接口视图。

```
interface route-aggregation interface-number.subnumber
```

- 进入以太网冗余子接口视图。

```
interface reth interface-number.subnumber
```

- (3) 开启子接口的 QinQ 终结功能，并指定子接口可以终结的 VLAN 报文的最外两层 VLAN ID。

```
vlan-type dot1q vid vlan-id second-dot1q vlan-id [ loose ]
```

缺省情况下，子接口的 QinQ 终结功能处于关闭状态。

L2VE 子接口不支持配置 **loose** 参数。

## 1.8 配置Untagged终结

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

- 进入三层以太网子接口视图。

```
interface interface-type interface-number.subnumber
```

- 进入三层聚合子接口视图。

```
interface route-aggregation interface-number.subnumber
```

- 进入以太网冗余子接口视图。

```
interface reth interface-number.subnumber
```

- (3) 开启子接口的 Untagged 终结功能。

```
vlan-type dot1q untagged
```

缺省情况下，子接口的 Untagged 终结功能处于关闭状态。

## 1.9 配置Default终结

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

- 进入三层以太网子接口视图。

```
interface interface-type interface-number.subnumber
```

- 进入三层聚合子接口视图。

```
interface route-aggregation interface-number.subnumber
```

- 进入以太网冗余子接口视图。

```
interface reth interface-number.subnumber
```

- (3) 开启子接口的 Default 终结功能。

```
vlan-type dot1q default
```

缺省情况下，子接口的 Default 终结功能处于关闭状态。

## 1.10 配置VLAN终结支持广播/组播

### 1. 功能简介

当接口下配置了模糊的 Dot1q 终结功能后，缺省情况下不允许发送广播/组播报文。只有配置了 VLAN 终结支持广播/组播功能，该接口才能发送广播/组播报文。

本功能允许接口遍历模糊终结范围内的 VLAN ID，给报文分别添加这些 VLAN ID 对应的 VLAN Tag 后，再发送报文。

### 2. 配置限制和指导

IPv6 网络中，当接口下配置了模糊的 Dot1q 终结功能后，建议配置 **vlan-termination broadcast ra** 命令，以允许接口遍历模糊终结的范围发送 RA（Router Advertisement，路由器通告消息）组播报文，其他类型的广播/组播报文则不允许发送。该命令与 **vlan-termination broadcast enable** 命令相比，能有效减少设备 CPU 负担。

### 3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

- 进入三层以太网子接口视图。

```
interface interface-type interface-number.subnumber
```

- 进入三层聚合子接口视图。

```
interface route-aggregation interface-number.subnumber
```

- 进入以太网冗余子接口视图。

```
interface reth interface-number.subnumber
```

- (3) 配置允许接口发送广播/组播报文。请选择其中一项进行配置。

- 允许接口发送广播和组播报文。

**vlan-termination broadcast enable**

- 允许接口发送 RA 组播报文（IPv6 环境）。

**vlan-termination broadcast ra**

缺省情况下，接口配置了模糊的 Dot1q 终结或者模糊的 QinQ 终结功能后，不允许发送广播/组播报文。

## 1.11 配置VLAN Tag的TPID值

### 1. 功能简介

在子接口/VLAN 接口上使用 VLAN 终结功能时，可以通过以下配置指定接口接收和发送报文的最外层 VLAN Tag 的 TPID 值。在配置 TPID 值后，当接收报文时，只有报文最外层 VLAN Tag 的 TPID 值为 0x8100 或者指定值的报文才会作为 VLAN 报文来处理；发送报文时，会给报文最外层 VLAN Tag 的 TPID 值填入指定值，如果报文带有两层及以上 VLAN Tag，则给报文其他层 VLAN Tag 的 TPID 值都填入 0x8100。

### 2. 配置步骤

- (1) 进入系统视图。

**system-view**

- (2) 进入接口视图。

- 进入三层以太网接口视图。

**interface interface-type interface-number**

- 进入三层聚合接口视图。

**interface route-aggregation interface-number**

- 进入以太网冗余接口视图。

**interface reth interface-number**

- (3) 配置接口接收和发送的报文最外层 VLAN Tag 的 TPID 值。

**dot1q ethernet-type hex-value**

缺省情况下，接口接收和发送的报文最外层 VLAN Tag 的 TPID 值均为 0x8100。

# 目 录

|                             |            |
|-----------------------------|------------|
| <b>1 生成树协议概述 .....</b>      | <b>1-1</b> |
| 1.1 STP 简介 .....            | 1-1        |
| 1.1.1 STP 的协议报文 .....       | 1-1        |
| 1.1.2 STP 的基本概念 .....       | 1-3        |
| 1.1.3 STP 的拓扑计算过程 .....     | 1-4        |
| 1.1.4 STP 算法实现举例 .....      | 1-6        |
| 1.1.5 STP 的 BPDU 传递机制 ..... | 1-8        |
| 1.1.6 STP 的时间参数 .....       | 1-9        |
| 1.2 RSTP 简介 .....           | 1-9        |
| 1.2.1 RSTP 的协议报文 .....      | 1-10       |
| 1.2.2 RSTP 的基本概念 .....      | 1-10       |
| 1.2.3 RSTP 的工作原理 .....      | 1-10       |
| 1.2.4 RSTP 中的 BPDU 处理 ..... | 1-11       |
| 1.3 PVST 简介 .....           | 1-11       |
| 1.3.1 PVST 的协议报文 .....      | 1-12       |
| 1.3.2 PVST 的工作原理 .....      | 1-12       |
| 1.4 MSTP 简介 .....           | 1-12       |
| 1.4.1 MSTP 的优点 .....        | 1-12       |
| 1.4.2 MSTP 的协议报文 .....      | 1-13       |
| 1.4.3 MSTP 的基本概念 .....      | 1-14       |
| 1.4.4 MSTP 的工作原理 .....      | 1-18       |
| 1.4.5 MSTP 在设备上的实现 .....    | 1-18       |
| 1.5 快速收敛机制 .....            | 1-19       |
| 1.5.1 边缘端口机制 .....          | 1-19       |
| 1.5.2 根端口快速切换机制 .....       | 1-20       |
| 1.5.3 P/A 机制 .....          | 1-20       |
| 1.6 协议规范 .....              | 1-22       |
| <b>2 配置生成树协议 .....</b>      | <b>2-1</b> |
| 2.1 生成树协议配置限制和指导 .....      | 2-1        |
| 2.1.1 接口相关配置限制和指导 .....     | 2-1        |
| 2.2 生成树协议配置任务简介 .....       | 2-1        |
| 2.2.1 STP 配置任务简介 .....      | 2-1        |

|                                        |      |
|----------------------------------------|------|
| 2.2.2 RSTP 配置任务简介 .....                | 2-2  |
| 2.2.3 PVST 配置任务简介 .....                | 2-3  |
| 2.2.4 MSTP 配置任务简介 .....                | 2-4  |
| 2.3 配置生成树的工作模式 .....                   | 2-5  |
| 2.4 配置 MST 域 .....                     | 2-6  |
| 2.5 配置根桥和备份根桥 .....                    | 2-7  |
| 2.5.1 配置限制和指导 .....                    | 2-7  |
| 2.5.2 配置根桥 .....                       | 2-7  |
| 2.5.3 配置备份根桥 .....                     | 2-8  |
| 2.6 配置设备的优先级 .....                     | 2-8  |
| 2.7 配置 MST 域的最大跳数 .....                | 2-8  |
| 2.8 配置交换网络的网络直径 .....                  | 2-9  |
| 2.9 配置生成树的时间参数 .....                   | 2-9  |
| 2.10 配置超时时间因子 .....                    | 2-11 |
| 2.11 配置端口发送 BPDU 的速率 .....             | 2-11 |
| 2.12 配置端口为边缘端口 .....                   | 2-12 |
| 2.13 配置端口的路径开销 .....                   | 2-12 |
| 2.13.1 功能简介 .....                      | 2-12 |
| 2.13.2 配置缺省路径开销的计算标准 .....             | 2-13 |
| 2.13.3 配置端口的路径开销 .....                 | 2-15 |
| 2.14 配置端口的优先级 .....                    | 2-15 |
| 2.15 配置端口的链路类型 .....                   | 2-16 |
| 2.16 配置端口收发的 MSTP 报文格式 .....           | 2-17 |
| 2.17 打开端口状态变化信息显示开关 .....              | 2-17 |
| 2.18 开启生成树协议 .....                     | 2-18 |
| 2.18.1 配置限制和指导 .....                   | 2-18 |
| 2.18.2 开启生成树协议（STP/RSTP/MSTP 模式） ..... | 2-18 |
| 2.18.3 开启生成树协议（PVST 模式） .....          | 2-18 |
| 2.19 执行 mCheck 操作 .....                | 2-19 |
| 2.19.1 功能简介 .....                      | 2-19 |
| 2.19.2 配置限制和指导 .....                   | 2-19 |
| 2.19.3 全局执行 mCheck 操作 .....            | 2-19 |
| 2.19.4 在端口上执行 mCheck 操作 .....          | 2-19 |
| 2.20 关闭 PVST 的 PVID 不一致保护功能 .....      | 2-19 |
| 2.21 配置摘要侦听功能 .....                    | 2-20 |
| 2.22 配置 No Agreement Check 功能 .....    | 2-21 |



|                                    |      |
|------------------------------------|------|
| 2.23 配置 TC Snooping 功能 .....       | 2-23 |
| 2.24 配置生成树保护功能 .....               | 2-24 |
| 2.24.1 生成树保护功能配置任务简介 .....         | 2-24 |
| 2.24.2 配置 BPDU 保护功能 .....          | 2-24 |
| 2.24.3 配置根保护功能 .....               | 2-25 |
| 2.24.4 配置环路保护功能 .....              | 2-26 |
| 2.24.5 配置端口角色限制功能 .....            | 2-26 |
| 2.24.6 配置 TC-BPDU 传播限制功能 .....     | 2-27 |
| 2.24.7 配置防 TC-BPDU 攻击保护功能 .....    | 2-27 |
| 2.24.8 配置 MSTP 的 PVST 报文保护功能 ..... | 2-28 |
| 2.24.9 Dispute 保护功能 .....          | 2-28 |
| 2.25 配置生成树的网管功能 .....              | 2-29 |
| 2.26 生成树显示和维护 .....                | 2-29 |

# 1 生成树协议概述

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。最初的生成树协议为 STP（Spanning Tree Protocol，生成树协议），之后又发展出 RSTP（Rapid Spanning Tree Protocol，快速生成树协议）、PVST（Per-VLAN Spanning Tree，每 VLAN 生成树）和 MSTP（Multiple Spanning Tree Protocol，多生成树协议）。

## 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。

### 1.1.1 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。本文中将生成树协议的协议报文均简称为 BPDU。

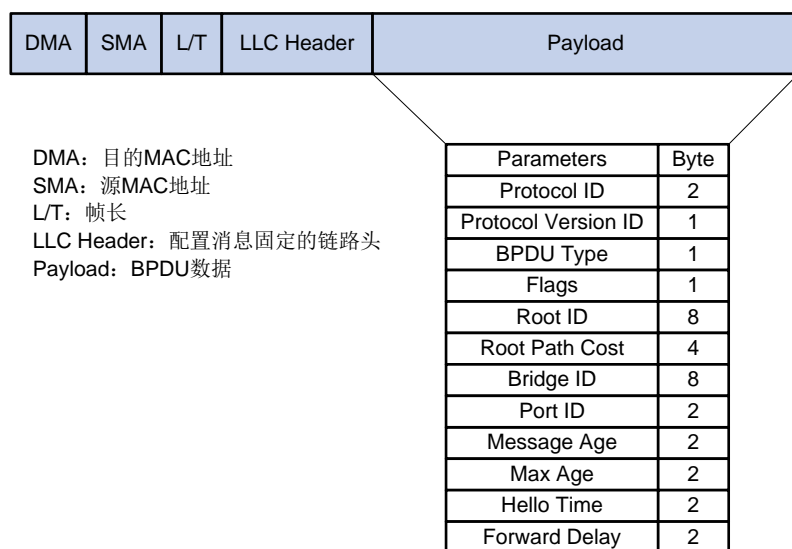
STP 通过在设备之间传递 BPDU 来确定网络的拓扑结构。BPDU 中包含了足够的信息来保证设备完成生成树的计算过程。STP 协议的 BPDU 分为以下两类：

- 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：

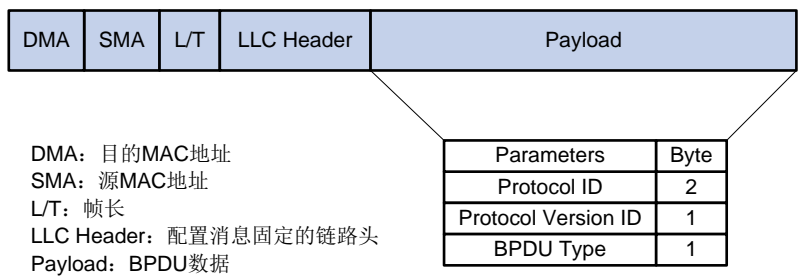
- 协议类型（Protocol ID）：固定为 0x0000，表示生成树协议。
- 协议版本号（Protocol Version ID）：目前生成树有三个版本，STP 的协议版本号为 0x00。
- BPDU 类型：配置 BPDU 类型为 0x00。
- BPDU Flags 位：BPDU 标志位，表示是哪种 BPDU。由 8 位组成，最低位（0 位）为 TC（Topology Change，拓扑改变）标志位；最高位（7 位）为 TCA（Topology Change Acknowledge，拓扑改变确认）标志位；其他 6 位保留。
- 根桥（Root Bridge）ID：由根桥的优先级和 MAC 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。

### 1.1.2 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

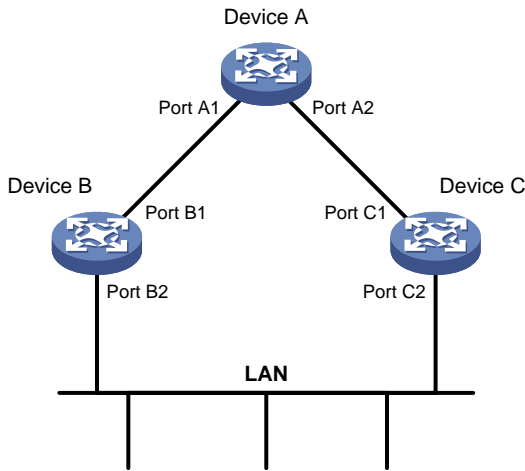
有关指定桥与指定端口的含义, 请参见[表 1-1](#) 的说明。

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。

### 1.1.3 STP 的拓扑计算过程

STP 的拓扑计算过程如下：设备通过比较不同端口收到的 BPDU 报文的优先级高低，选举出根桥、根端口、指定端口，完成生成树的计算，建立对应的树形拓扑。

#### 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，需要在网络中所有的 STP 设备中选择一个根桥，根桥的选择方式有以下两种：

- 自动选举：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。
- 手工指定：用户手工将设备配置为指定生成树的根桥或备份根桥。
  - 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。
  - 用户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。

### 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 <a href="#">表1-4</a> 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：

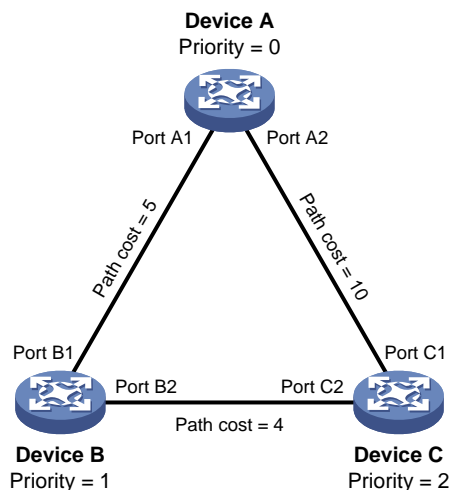
- 根桥 ID 较小的 BPDU 优先级较高；
- 若根桥 ID 相同，则比较根路径开销：将 BPDU 中的根路径开销与本端口对应的路径开销相加，二者之和较小的 BPDU 优先级较高；
- 若根路径开销也相同，则依次比较指定桥 ID、指定端口 ID、接收该 BPDU 的端口 ID 等，上述值较小的 BPDU 优先级较高。

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。

### 1.1.4 STP 算法实现举例

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。

#### 1. 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} |

| 设备       | 端口名称    | 端口的 BPDU           |
|----------|---------|--------------------|
| 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}。

## 2. 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:<br/>{0, 0, 0, Port A1}</li> <li>Port A2:<br/>{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:<br/>{0, 0, 0, Port A1}</li> <li>Port B2:<br/>{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:<br/>{0, 0, 0, Port A1}</li> <li>指定端口 Port B2:<br/>{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:<br/>{0, 0, 0, Port A2}</li> <li>Port C2:<br/>{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:<br/>{0, 0, 0, Port A2}</li> <li>指定端口 Port C2:<br/>{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</li> </ul>                                                                                                                                                                                                                             | <ul style="list-style-type: none"> <li>Port C1:</li> </ul>                                                                           |

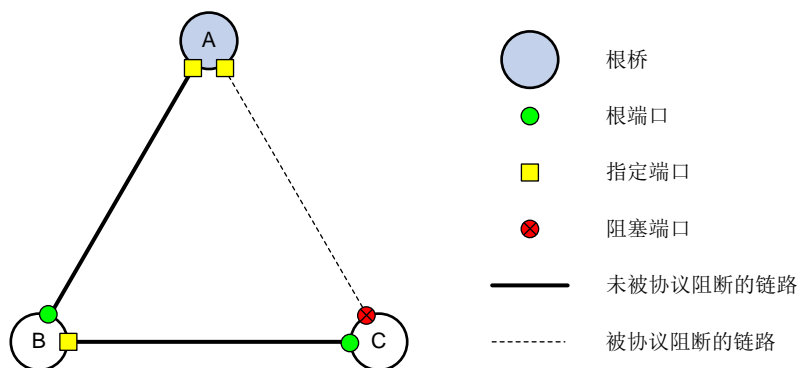


| 设备 | 比较过程                                                                                                                                                                                                                                                                                                                                                                                                                          | 比较后端口的 BPDU                                                                   |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|
|    | 自己的 BPDU {0, 10, 2, Port C2} 更优，于是更新自己的 BPDU。<br>• 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。                                                                                                                                                                                                                                                                                                  | {0, 0, 0, Port A2}<br>• Port C2:<br>{0, 5, 1, Port B2}                        |
|    | • 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。<br>• 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。 | • 阻塞端口 Port C1:<br>{0, 0, 0, Port A2}<br>• 根端口 Port C2:<br>{0, 5, 1, Port B2} |

### 3. 计算出的生成树

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。

#### 1.1.5 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。

### 1.1.6 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 Listening 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；Listening 状态持续 15 秒；Learning 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：

- $2 \times (\text{Forward Delay} - 1 \text{ 秒}) \geq \text{Max Age}$
- $\text{Max Age} \geq 2 \times (\text{Hello Time} + 1 \text{ 秒})$

## 1.2 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。

### 1.2.1 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 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。

### 1.2.2 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        |

### 1.2.3 RSTP 的工作原理

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；当端口确定为替换端口，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。

## 1.2.4 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。

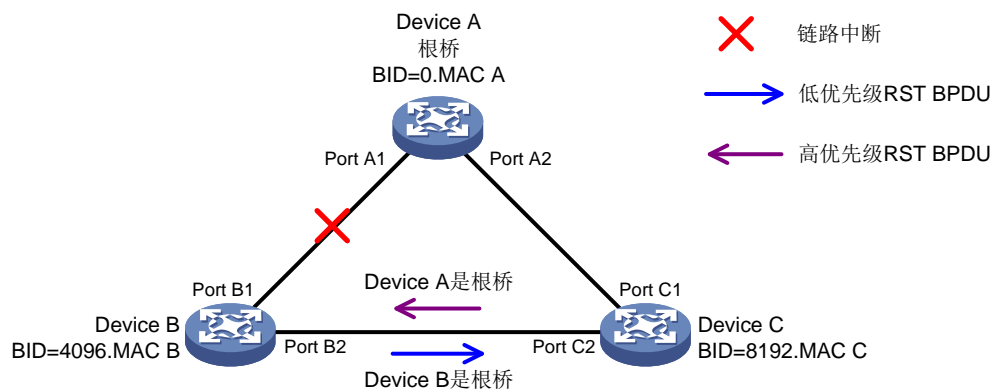
由于 RSTP 中网桥可以自行从指定端口发送 RST BPDU，所以在网桥之间可以提供一种保活机制，即在一定时间内网桥没有收到对端网桥发送的 RST BPDU，即可认为和对端网桥的连接中断。

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



## 1.3 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。

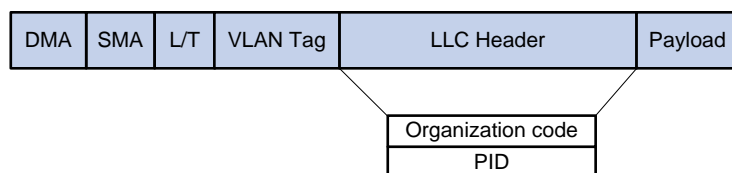
运行 PVST 的 H3C 设备可以与运行 Rapid PVST 或 PVST 的第三方设备互通。当运行 PVST 的 H3C 设备之间互联，或运行 PVST 的 H3C 设备与运行 Rapid PVST 的第三方设备互通时，H3C 设备支持像 RSTP 一样的快速收敛。

### 1.3.1 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 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。

### 1.3.2 PVST 的工作原理

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，并允许以每个 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 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。

PVST 的端口角色和端口状态和 RSTP 相同，能够实现快速收敛，请参见“[1.2.2 RSTP 的基本概念](#)”。

## 1.4 MSTP 简介

### 1.4.1 MSTP 的优点

MSTP 由 IEEE 制定的 802.1s 标准定义，相比于 STP、RSTP 和 PVST，MSTP 的优点如下：

- 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。生成树间独立计算，实现快速收敛。

- 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，不同 VLAN 的流量沿各自的路径转发，实现 VLAN 数据的负载分担。
- MSTP 兼容 STP 和 RSTP，部分兼容 PVST。

### 1.4.2 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 剩余跳数缺省值为 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。

### 1.4.3 MSTP 的基本概念

图1-9 MSTP 的基本概念示意图

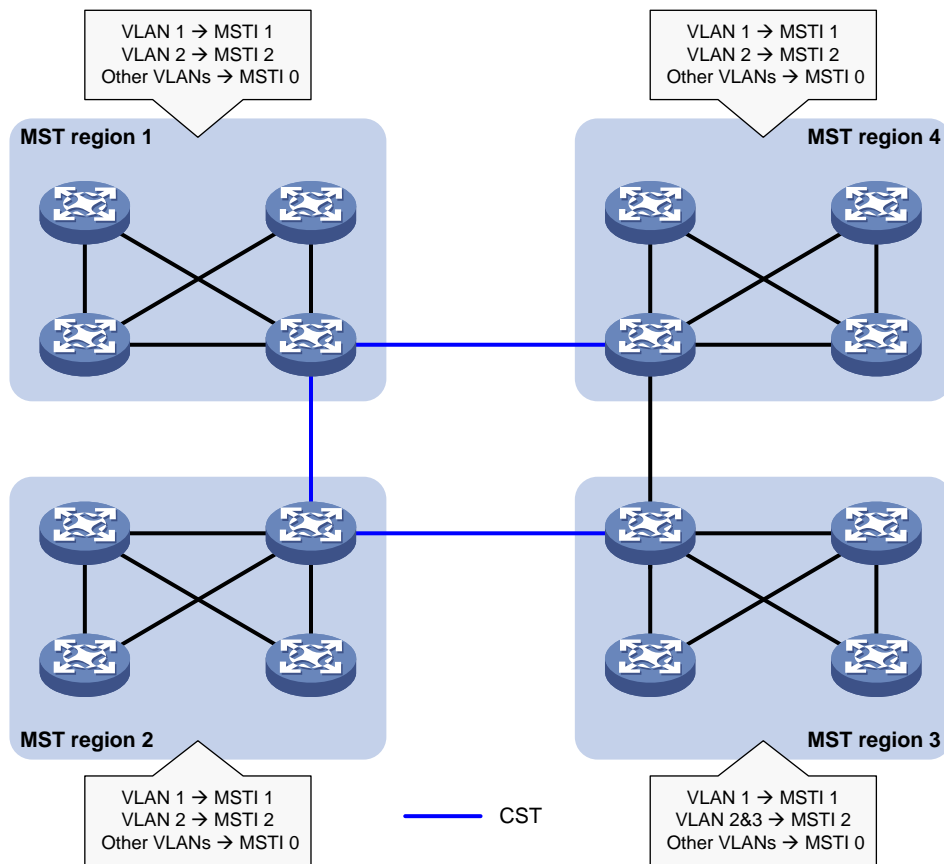
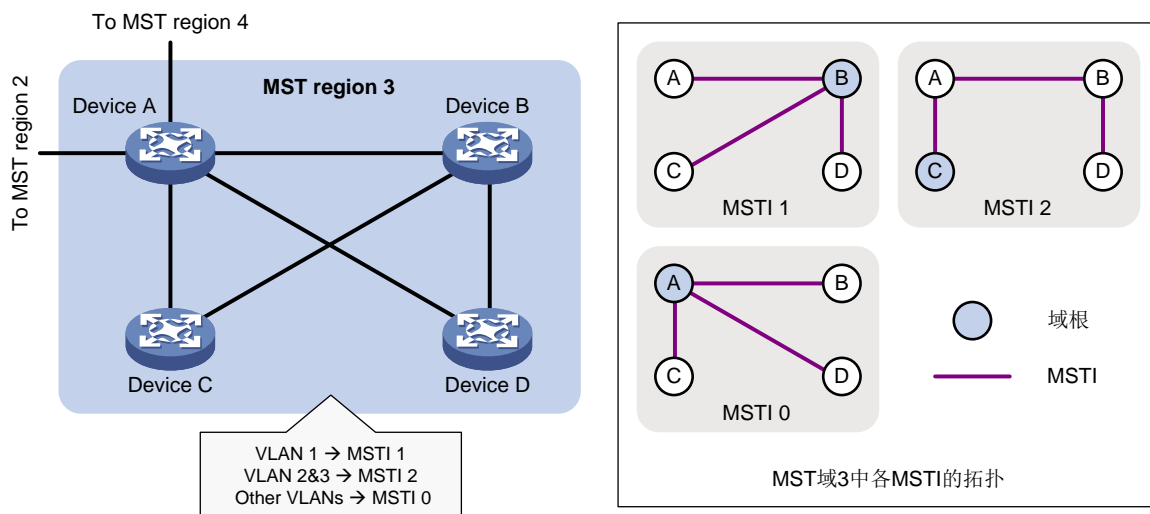


图1-10 MST 域 3 详图



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。

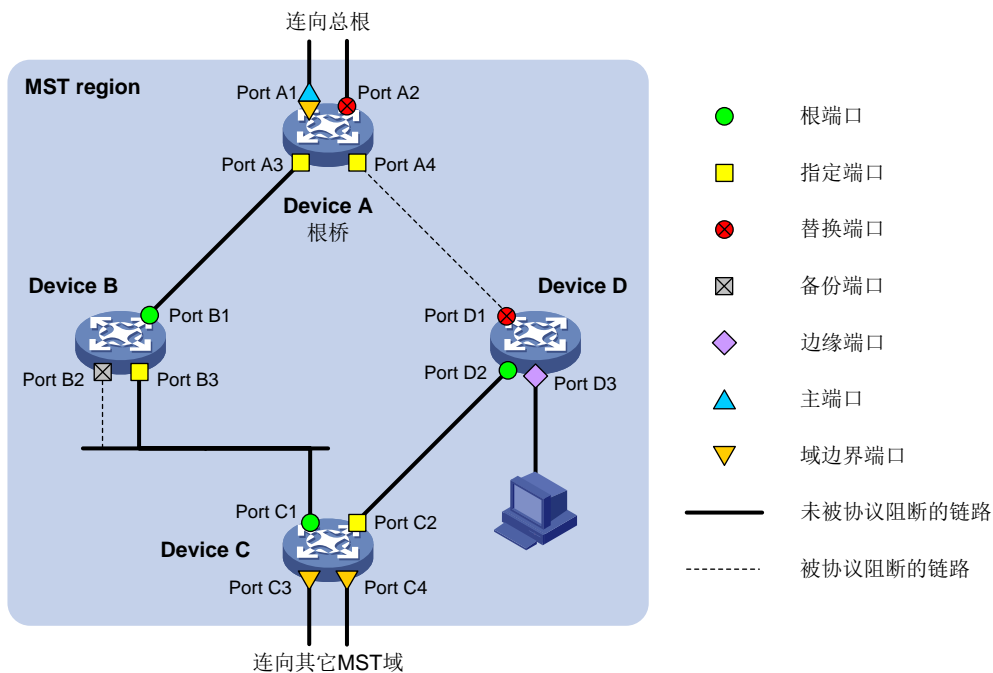
#### 8. 总根

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同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）<br>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         | √       | √    | √    | √    |

#### 1.4.4 MSTP 的工作原理

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“[1.1.3 STP 的拓扑计算过程](#)”。

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。

#### 1.4.5 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。
- 支持接口板的热插拔，同时支持主控板与备用主控板的倒换。

## 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。

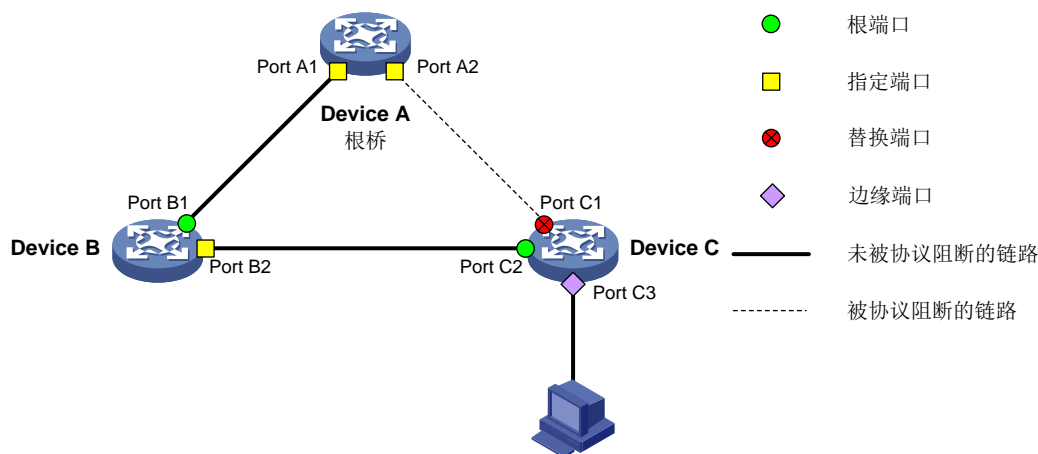
### 1.5.1 边缘端口机制

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终端连接的端口配置为边缘端口。

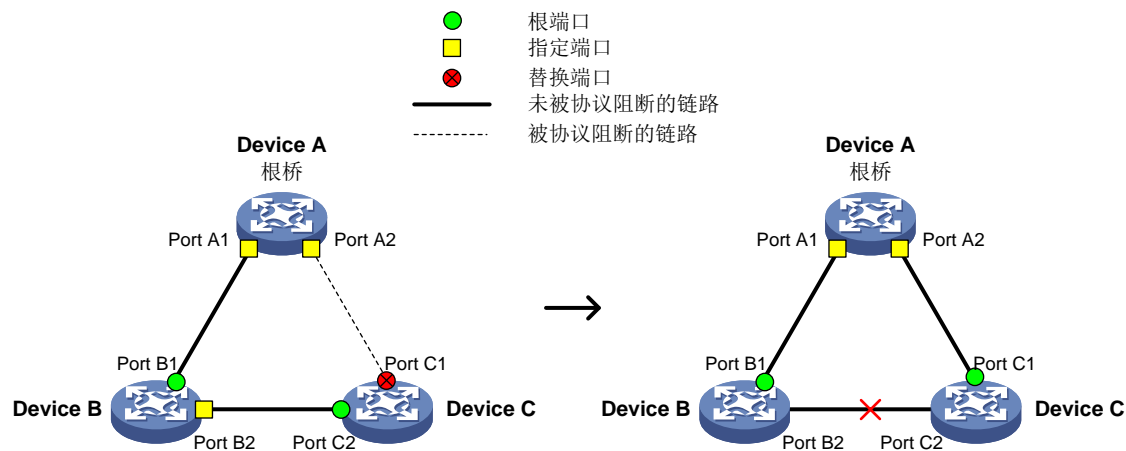
图1-12 边缘端口示意图



## 1.5.2 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换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换端口，当根端口链路中断时，替换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。

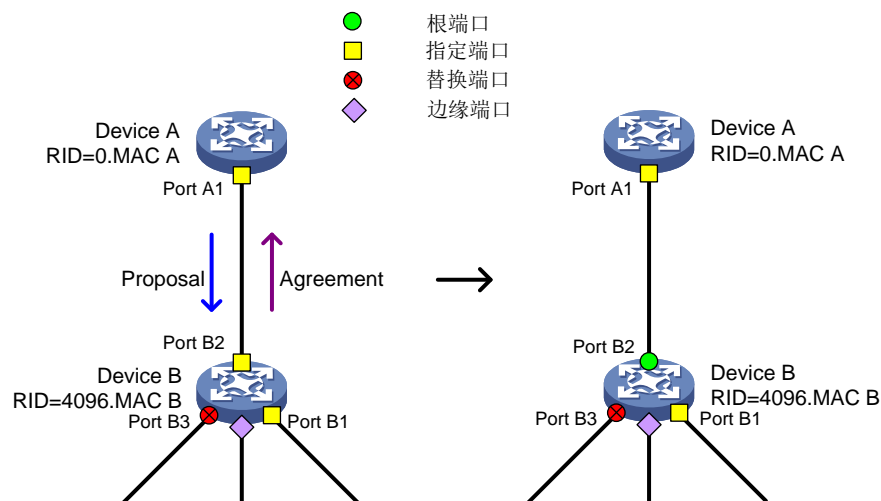
## 1.5.3 P/A 机制

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 BPDU 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



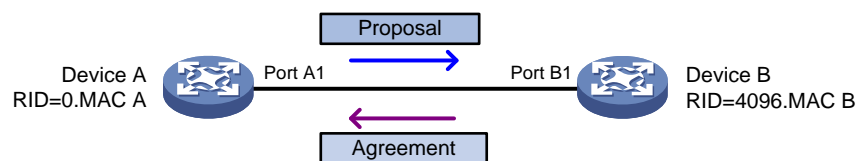
如图 1-14，当 Device A 和 Device B 之间的链路连接后，P/A 机制处理过程如下：

- 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 和替换端口 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。

## 1.6 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

## 2 配置生成树协议

### 2.1 生成树协议配置限制和指导

#### 2.1.1 接口相关配置限制和指导

生成树的部分功能支持在二层以太网接口视图、二层聚合接口视图配置，本文后续将概括称为接口视图。

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。

### 2.2 生成树协议配置任务简介

#### 2.2.1 STP 配置任务简介

##### 1. 配置根桥

STP 模式下，根桥上的配置任务如下：

- (1) [配置生成树的工作模式](#)  
通过本配置将生成树的工作模式配置为 STP 模式。
- (2) （可选）[配置根桥和备份根桥](#)
- (3) （可选）[配置设备的优先级](#)
- (4) （可选）配置影响 STP 拓扑收敛的参数
  - [配置交换网络的网络直径](#)
  - [配置生成树的时间参数](#)
  - [配置超时时间因子](#)
  - [配置端口发送 BPDU 的速率](#)
- (5) （可选）[打开端口状态变化信息显示开关](#)
- (6) [开启生成树协议](#)
- (7) （可选）配置生成树高级功能
  - [配置 TC Snooping 功能](#)
  - [配置生成树保护功能](#)
  - [配置生成树的网管功能](#)

##### 2. 配置叶子节点

STP 模式下，叶子节点上的配置任务如下：



- (1) [配置生成树的工作模式](#)  
通过本配置将生成树的工作模式配置为 STP 模式。
- (2) (可选) [配置设备的优先级](#)
- (3) (可选) 配置影响 STP 拓扑收敛的参数
  - [配置超时时间因子](#)
  - [配置端口发送 BPDU 的速率](#)
  - [配置端口的路径开销](#)
  - [配置端口的优先级](#)
- (4) (可选) [打开端口状态变化信息显示开关](#)
- (5) [开启生成树协议](#)
- (6) (可选) 配置生成树高级功能
  - [配置 TC Snooping 功能](#)
  - [配置生成树保护功能](#)
  - [配置生成树的网管功能](#)

## 2.2.2 RSTP 配置任务简介

### 1. 配置根桥

RSTP 模式下，根桥上的配置任务如下：

- (1) [配置生成树的工作模式](#)  
通过本配置将生成树的工作模式配置为 RSTP 模式。
- (2) (可选) [配置根桥和备份根桥](#)
- (3) (可选) [配置设备的优先级](#)
- (4) (可选) 配置影响 RSTP 拓扑收敛的参数
  - [配置交换网络的网络直径](#)
  - [配置生成树的时间参数](#)
  - [配置超时时间因子](#)
  - [配置端口发送 BPDU 的速率](#)
  - [配置端口为边缘端口](#)
  - [配置端口的链路类型](#)
- (5) (可选) [打开端口状态变化信息显示开关](#)
- (6) [开启生成树协议](#)
- (7) (可选) 配置生成树高级功能
  - [执行 mCheck 操作](#)
  - [配置 TC Snooping 功能](#)
  - [配置生成树保护功能](#)
  - [配置生成树的网管功能](#)

### 2. 配置叶子节点

RSTP 模式下，叶子节点上的配置任务如下：

- (1) [配置生成树的工作模式](#)  
通过本配置将生成树的工作模式配置为 RSTP 模式。
- (2) (可选) [配置设备的优先级](#)
- (3) (可选) 配置影响 RSTP 拓扑收敛的参数
  - [配置超时时间因子](#)
  - [配置端口发送 BPDU 的速率](#)
  - [配置端口为边缘端口](#)
  - [配置端口的路径开销](#)
  - [配置端口的优先级](#)
  - [配置端口的链路类型](#)
- (4) (可选) [打开端口状态变化信息显示开关](#)
- (5) [开启生成树协议](#)
- (6) (可选) 配置生成树高级功能
  - [执行 mCheck 操作](#)
  - [配置 TC Snooping 功能](#)
  - [配置生成树保护功能](#)
  - [配置生成树的网管功能](#)

## 2.2.3 PVST 配置任务简介

### 1. 配置根桥

PVST 模式下，根桥上的配置任务如下：

- (1) [配置生成树的工作模式](#)  
通过本配置将生成树的工作模式配置为 PVST 模式。
- (2) (可选) [配置根桥和备份根桥](#)
- (3) (可选) [配置设备的优先级](#)
- (4) (可选) 配置影响 PVST 拓扑收敛的参数
  - [配置交换网络的网络直径](#)
  - [配置生成树的时间参数](#)
  - [配置超时时间因子](#)
  - [配置端口发送 BPDU 的速率](#)
  - [配置端口为边缘端口](#)
  - [配置端口的链路类型](#)
- (5) (可选) [打开端口状态变化信息显示开关](#)
- (6) [开启生成树协议](#)
- (7) (可选) 配置生成树高级功能
  - [执行 mCheck 操作](#)
  - [关闭 PVST 的 PVID 不一致保护功能](#)
  - [配置生成树保护功能](#)

- [配置生成树的网管功能](#)

## 2. 配置叶子节点

PVST 模式下，叶子节点上的配置任务如下：

- (1) [配置生成树的工作模式](#)  
通过本配置将生成树的工作模式配置为 PVST 模式。
- (2) （可选）[配置设备的优先级](#)
- (3) （可选）配置影响 PVST 拓扑收敛的参数
  - [配置超时时间因子](#)
  - [配置端口发送 BPDU 的速率](#)
  - [配置端口为边缘端口](#)
  - [配置端口的路径开销](#)
  - [配置端口的优先级](#)
  - [配置端口的链路类型](#)
- (4) （可选）[打开端口状态变化信息显示开关](#)
- (5) [开启生成树协议](#)
- (6) （可选）配置生成树高级功能
  - [执行 mCheck 操作](#)
  - [关闭 PVST 的 PVID 不一致保护功能](#)
  - [配置生成树保护功能](#)
  - [配置生成树的网管功能](#)

### 2.2.4 MSTP 配置任务简介

#### 1. 配置根桥

MSTP 模式下，根桥上的配置任务如下：

- (1) [配置生成树的工作模式](#)  
通过本配置将生成树的工作模式配置为 MSTP 模式。
- (2) [配置 MST 域](#)
- (3) （可选）[配置根桥和备份根桥](#)
- (4) （可选）[配置设备的优先级](#)
- (5) （可选）配置影响 MSTP 拓扑收敛的参数
  - [配置 MST 域的最大跳数](#)
  - [配置交换网络的网络直径](#)
  - [配置生成树的时间参数](#)
  - [配置超时时间因子](#)
  - [配置端口发送 BPDU 的速率](#)
  - [配置端口为边缘端口](#)
  - [配置端口的链路类型](#)
- (6) （可选）[配置端口收发的 MSTP 报文格式](#)

- (7) (可选) [打开端口状态变化信息显示开关](#)
- (8) [开启生成树协议](#)
- (9) (可选) 配置生成树高级功能
  - [执行 mCheck 操作](#)
  - [配置摘要侦听功能](#)
  - [配置 No Agreement Check 功能](#)
  - [配置 TC Snooping 功能](#)
  - [配置生成树保护功能](#)
  - [配置生成树的网管功能](#)

## 2. 配置叶子节点

MSTP 模式下，叶子节点上的配置任务如下：

- (1) [配置生成树的工作模式](#)  
通过本配置将生成树的工作模式配置为 MSTP 模式。
- (2) [配置 MST 域](#)
- (3) (可选) [配置设备的优先级](#)
- (4) (可选) 配置影响 MSTP 拓扑收敛的参数
  - [配置超时时间因子](#)
  - [配置端口发送 BPDU 的速率](#)
  - [配置端口为边缘端口](#)
  - [配置端口的路径开销](#)
  - [配置端口的优先级](#)
  - [配置端口的链路类型](#)
- (5) (可选) [配置端口收发的 MSTP 报文格式](#)
- (6) (可选) [打开端口状态变化信息显示开关](#)
- (7) [开启生成树协议](#)
- (8) (可选) 配置生成树高级功能
  - [执行 mCheck 操作](#)
  - [配置摘要侦听功能](#)
  - [配置 No Agreement Check 功能](#)
  - [配置 TC Snooping 功能](#)
  - [配置生成树保护功能](#)
  - [配置生成树的网管功能](#)

## 2.3 配置生成树的工作模式

### 1. 功能简介

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。
- 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。

## 2. 配置限制和指导

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。

## 3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 配置生成树的工作模式。

```
stp mode { mstp | pvst | rstp | stp }
```

缺省情况下，生成树的工作模式为 MSTP 模式。

## 2.4 配置MST域

### 1. 功能简介

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震荡。为了减少网络震荡，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。

### 2. 配置限制和指导

在 STP/RSTP/PVST 模式下，MST 域的相关配置不会生效。

### 3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入 MST 域视图。

```
stp region-configuration
```

- (3) 配置 MST 域的域名。

```
region-name name
```

缺省情况下，MST 域的域名为设备的 MAC 地址。

- (4) 配置 VLAN 映射表。请选择其中一项进行配置。

- 将指定 VLAN 映射到指定的 MSTI 上。

```
instance instance-id vlan vlan-id-list
```

- 快速配置 VLAN 映射表。

```
vlan-mapping modulo modulo
```

缺省情况下，所有 VLAN 都映射到 CIST（即 MSTI 0）上。

- (5) 配置 MSTP 的修订级别。

```
revision-level level
```

缺省情况下，MSTP 的修订级别为 0。

- (6) （可选）显示 MST 域的预配置信息。

```
check region-configuration
```

- (7) 激活 MST 域的配置。

```
active region-configuration
```

## 2.5 配置根桥和备份根桥

### 2.5.1 配置限制和指导

生成树协议可以根据桥 ID 自动计算确定生成树的根桥，也可以手工将设备配置为指定生成树的根桥或备份根桥。手工指定时，需要注意：

- 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。
- 用户指定根桥后不会再根据设备的优先级选举根桥。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“[2.6 配置设备的优先级](#)”。

### 2.5.2 配置根桥

- (1) 进入系统视图。

```
system-view
```

- (2) 配置设备为根桥。

- STP/RSTP 模式：

```
stp root primary
```

- PVST 模式：

```
stp vlan vlan-id-list root primary
```

- MSTP 模式：

```
stp [ instance instance-list ] root primary
```

缺省情况下，设备不是根桥。

### 2.5.3 配置备份根桥

- (1) 进入系统视图。

**system-view**

- (2) 配置设备为备份根桥。

- STP/RSTP 模式:

**stp root secondary**

- PVST 模式:

**stp vlan vlan-id-list root secondary**

- MSTP 模式:

**stp [ instance instance-list ] root secondary**

缺省情况下，设备不是备份根桥。

## 2.6 配置设备的优先级

### 1. 功能简介

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。

### 2. 配置步骤

- (1) 进入系统视图。

**system-view**

- (2) 配置设备的优先级。

- STP/RSTP 模式:

**stp priority priority**

- PVST 模式:

**stp vlan vlan-id-list priority priority**

- MSTP 模式:

**stp [ instance instance-list ] priority priority**

缺省情况下，设备的优先级为 32768。

## 2.7 配置MST域的最大跳数

### 1. 功能简介

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。

## 2. 配置限制和指导

本配置只需在根桥设备上进行，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。

## 3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 配置 MST 域的最大跳数。

```
stp max-hops hops
```

缺省情况下，MST 域的最大跳数为 20。

## 2.8 配置交换网络的网络直径

### 1. 功能简介

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。

### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 配置交换网络的网络直径。

- o STP/RSTP/MSTP 模式：

```
stp bridge-diameter diameter
```

- o PVST 模式：

```
stp vlan vlan-id-list bridge-diameter diameter
```

缺省情况下，交换网络的网络直径为 7。

## 2.9 配置生成树的时间参数

### 1. 功能简介

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 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。
- (3) **Max Age**: 用于确定 BPDU 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 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。

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:

- $2 \times (\text{Forward Delay} - 1 \text{ 秒}) \geq \text{Max Age}$
- $\text{Max Age} \geq 2 \times (\text{Hello Time} + 1 \text{ 秒})$

## 2. 配置限制和指导

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 BPDU, 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。

## 3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 配置 **Forward Delay** 时间参数。

- STP/RSTP/MSTP 模式:

```
stp timer forward-delay time
```

- PVST 模式:

```
stp vlan vlan-id-list timer forward-delay time
```

缺省情况下, **Forward Delay** 为 15 秒。

- (3) 配置 **Hello Time** 时间参数。

- STP/RSTP/MSTP 模式:

```
stp timer hello time
```

- PVST 模式:

```
stp vlan vlan-id-list timer hello time
```

缺省情况下, Hello Time 为 2 秒。

- (4) 配置 Max Age 时间参数。

- STP/RSTP/MSTP 模式:

```
stp timer max-age time
```

- PVST 模式:

```
stp vlan vlan-id-list timer max-age time
```

缺省情况下, Max Age 为 20 秒。

## 2.10 配置超时时间因子

### 1. 功能简介

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。

### 2. 配置限制和指导

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。

### 3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 配置设备的超时时间因子。

```
stp timer-factor factor
```

缺省情况下, 设备的超时时间因子为 3。

## 2.11 配置端口发送BPDU的速率

### 1. 功能简介

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。

### 2. 配置限制和指导

端口发送 BPDU 的速率与端口的物理状态和网络结构有关, 建议用户采用缺省配置。

### 3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

```
interface interface-type interface-number
```

- (3) 配置端口的发送 BPDU 的速率。

```
stp transmit-limit limit
```

缺省情况下，端口发送 BPDU 的速率为 10。

## 2.12 配置端口为边缘端口

### 1. 功能简介

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。

### 2. 配置限制和指导

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。

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。

### 3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

```
interface interface-type interface-number
```

- (3) 配置当前端口为边缘端口。

```
stp edged-port
```

缺省情况下，端口为非边缘端口。

## 2.13 配置端口的路径开销

### 2.13.1 功能简介

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。

### 2.13.2 配置缺省路径开销的计算标准

#### 1. 功能简介

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。

不同速率链路的路径开销值请参见下列各表。

表2-1 100M 及以下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     |

表2-2 1000M 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 |
| 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   |

表2-3 10G 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 |
| 10Gbps | 单个端口          | 2                | 2,000       | 2    |
|        | 聚合接口（含两个选中端口） |                  | 1,000       | 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 |
|      | 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    |

表2-4 20G 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 |
| 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    |

表2-5 40G 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 |
| 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    |

表2-6 100G 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 |
| 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    |

## 2. 配置限制和指导

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。

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“[2.13.3 配置端口的路径开销](#)”）。

### 3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 配置缺省路径开销的计算标准。

```
stp pathcost-standard { dot1d-1998 | dot1t | legacy }
```

缺省情况下，缺省路径开销的计算标准为 **legacy**。

## 2.13.3 配置端口的路径开销

### 1. 配置限制和指导

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。

### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

```
interface interface-type interface-number
```

- (3) 配置端口的路径开销。

- STP/RSTP 模式：

```
stp cost cost-value
```

- PVST 模式：

```
stp vlan vlan-id-list cost cost-value
```

- MSTP 模式：

```
stp [ instance instance-list ] cost cost-value
```

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。

## 2.14 配置端口的优先级

### 1. 功能简介

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。

### 2. 配置限制和指导

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，引起网络拓扑变化，请用户做好相关准备工作。

### 3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

```
interface interface-type interface-number
```

- (3) 配置端口的优先级。

- STP/RSTP 模式：

```
stp port priority priority
```

- PVST 模式：

```
stp vlan vlan-id-list port priority priority
```

- MSTP 模式：

```
stp [ instance instance-list ] port priority priority
```

缺省情况下，端口的优先级为 128。

## 2.15 配置端口的链路类型

### 1. 功能简介

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。

### 2. 配置限制和指导

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。

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，则有可能引入临时环路。

### 3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

```
interface interface-type interface-number
```

- (3) 配置端口的链路类型。

```
stp point-to-point { auto | force-false | force-true }
```

缺省情况下，端口的链路类型为 **auto**，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。



## 2.16 配置端口收发的MSTP报文格式

### 1. 功能简介

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。

### 2. 配置限制和指导

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。

### 3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

```
interface interface-type interface-number
```

- (3) 配置端口收发的 MSTP 报文格式。

```
stp compliance { auto | dot1s | legacy }
```

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。

## 2.17 打开端口状态变化信息显示开关

### 1. 功能简介

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。

### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 打开端口状态变化信息显示开关。

- STP/RSTP 模式：

```
stp port-log instance 0
```

- PVST 模式：

```
stp port-log vlan vlan-id-list
```

- MSTP 模式：

```
stp port-log { all | instance instance-list }
```



缺省情况下，端口状态变化信息显示开关处于关闭状态。

## 2.18 开启生成树协议

### 2.18.1 配置限制和指导

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。

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。

### 2.18.2 开启生成树协议（STP/RSTP/MSTP 模式）

- (1) 进入系统视图。

```
system-view
```

- (2) 全局开启生成树协议。

```
stp global enable
```

缺省情况下，生成树协议的全局状态为关闭。

- (3) 进入接口视图。

```
interface interface-type interface-number
```

- (4) 在端口上开启生成树协议。

```
stp enable
```

缺省情况下，所有端口上的生成树协议均处于开启状态。

### 2.18.3 开启生成树协议（PVST 模式）

- (1) 进入系统视图。

```
system-view
```

- (2) 全局开启生成树协议。

```
stp global enable
```

缺省情况下，生成树协议的全局状态为关闭。

- (3) 在 VLAN 中开启生成树协议。

```
stp vlan vlan-id-list enable
```

缺省情况下，生成树协议在 VLAN 中处于开启状态。

- (4) 进入接口视图。

```
interface interface-type interface-number
```

- (5) 在端口上开启生成树协议。

```
stp enable
```

缺省情况下，所有端口上的生成树协议均处于开启状态。

## 2.19 执行mCheck操作

### 2.19.1 功能简介

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。

### 2.19.2 配置限制和指导

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。

### 2.19.3 全局执行 mCheck 操作

- (1) 进入系统视图。

```
system-view
```

- (2) 全局执行 mCheck 操作。

```
stp global mcheck
```

### 2.19.4 在端口上执行 mCheck 操作

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

```
interface interface-type interface-number
```

- (3) 在端口上执行 mCheck 操作。

```
stp mcheck
```

## 2.20 关闭PVST的PVID不一致保护功能

### 1. 功能简介

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。

## 2. 配置限制和指导

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。

## 3. 配置步骤

(1) 进入系统视图。

```
system-view
```

(2) 关闭 PVST 的 PVID 不一致保护功能。

```
stp ignore-pvid-inconsistency
```

缺省情况下，PVST 的 PVID 不一致保护功能处于开启状态。

## 2.21 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

配置本任务前，请确保生成树协议在我方设备与第三方厂商设备上均正常运行。

### 4. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

```
interface interface-type interface-number
```

- (3) 在端口上开启摘要侦听功能。

```
stp config-digest-snooping
```

缺省情况下，端口上的摘要侦听功能处于关闭状态。

- (4) 退回系统视图。

```
quit
```

- (5) 全局开启摘要侦听功能。

```
stp global config-digest-snooping
```

缺省情况下，摘要侦听功能处于全局关闭状态。

## 2.22 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：

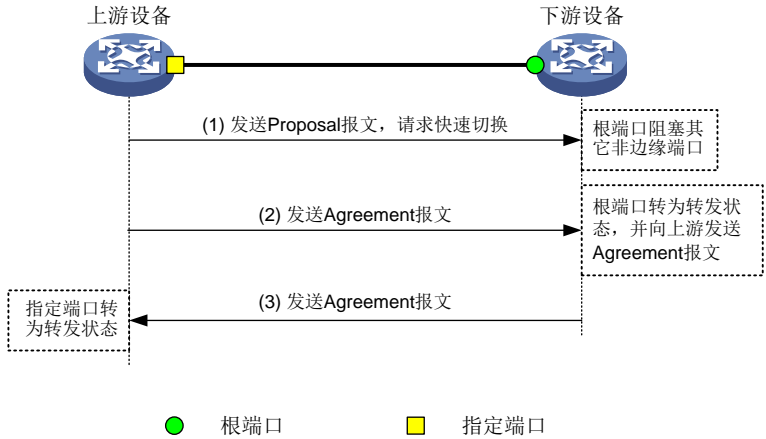
- Proposal 报文：指定端口请求快速迁移的报文。
- Agreement 报文：同意对端进行快速迁移的报文。

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。

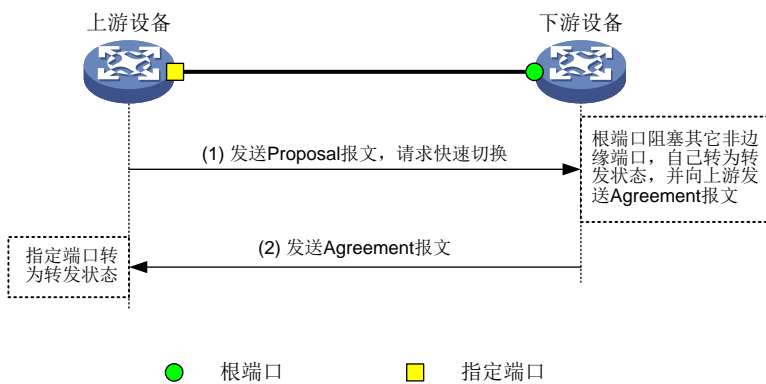
如图 2-1 所示，是 MSTP 的指定端口快速迁移机制。

图2-1 MSTP 指定端口快速迁移机制



如图 2-2 所示，是 RSTP 的指定端口快速迁移机制。

图2-2 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. 配置限制和指导

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。

## 3. 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，以确保它们在同一个域内。

#### 4. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

```
interface interface-type interface-number
```

- (3) 开启 No Agreement Check 功能。

```
stp no-agreement-check
```

缺省情况下，No Agreement Check 功能处于关闭状态。

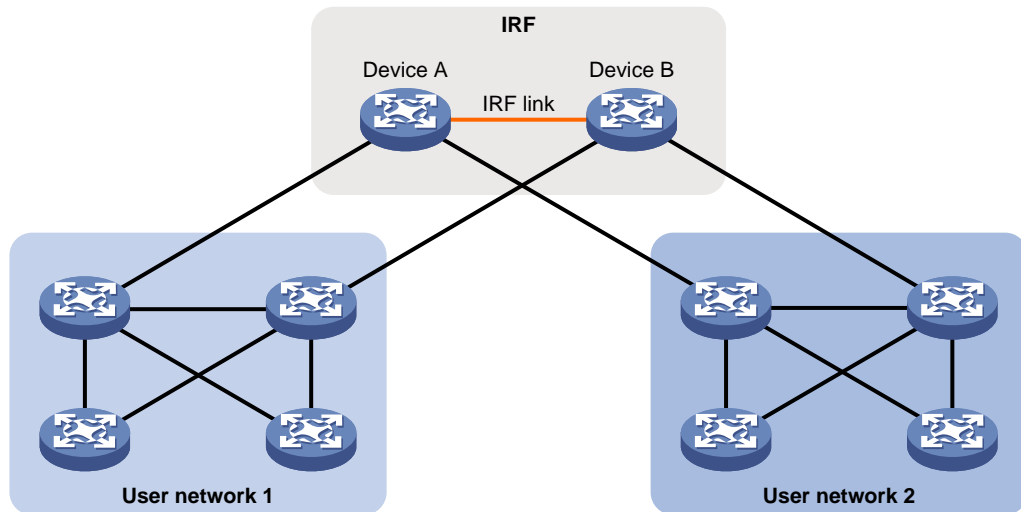
## 2.23 配置TC Snooping功能

### 1. 功能简介

TC Snooping 功能的典型应用环境如图 2-3 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2-3 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”。

## 2. 配置限制和指导

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。

## 3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 全局关闭生成树协议。

```
undo stp global enable
```

缺省情况下，生成树协议在全局中处于关闭状态。

- (3) 开启 TC Snooping 功能。

```
stp tc-snooping
```

缺省情况下，TC Snooping 功能处于关闭状态。

## 2.24 配置生成树保护功能

### 2.24.1 生成树保护功能配置任务简介

本节中的所有配置均为可选，请根据实际情况选择配置。

- [配置 BPDU 保护功能](#)
- [配置根保护功能](#)
- [配置环路保护功能](#)
- [配置端口角色限制功能](#)
- [配置 TC-BPDU 传播限制功能](#)
- [配置防 TC-BPDU 攻击保护功能](#)
- [配置 MSTP 的 PVST 报文保护功能](#)
- [Dispute 保护功能](#)

### 2.24.2 配置 BPDU 保护功能

#### 1. 功能简介

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”。



## 2. 配置限制和指导

配置端口的 BPDU 保护功能时，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。

本功能只对 **stp edged-port** 命令手工指定的边缘端口生效。

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”。

## 3. 配置 BPDU 保护功能

- (1) 进入系统视图。

```
system-view
```

- (2) 开启全局的 BPDU 保护功能。

```
stp bpdu-protection
```

缺省情况下，全局的 BPDU 保护功能处于关闭状态。

## 2.24.3 配置根保护功能

### 1. 功能简介

请在设备的指定端口上配置本功能。

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。

### 2. 配置限制和指导

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。

### 3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

```
interface interface-type interface-number
```

- (3) 开启端口的根保护功能。

```
stp root-protection
```

缺省情况下，端口上的根保护功能处于关闭状态。



## 2.24.4 配置环路保护功能

### 1. 功能简介

请在设备的根端口和替换端口上配置本功能。

依靠不断接收上游设备发送的 BPDUs，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 BPDUs，此时下游设备会重新选择端口角色，收不到 BPDUs 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 BPDUs，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。

### 2. 配置限制和指导

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 BPDUs 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。

以下端口配置环路保护功能后，该端口不会因收不到 BPDUs 而导致其一直处于 Discarding 状态，而是进行端口状态迁移，经过两个 Forward Delay 时长后再次变为 Forwarding 状态：

- 端口状态从 down 变成 up。
- 处于 up 状态的端口，生成树功能状态从关闭变成开启。

### 3. 配置步骤

(1) 进入系统视图。

```
system-view
```

(2) 进入接口视图。

```
interface interface-type interface-number
```

(3) 开启端口的环路保护功能。

```
stp loop-protection
```

缺省情况下，端口的环路保护功能处于关闭状态。

## 2.24.5 配置端口角色限制功能

### 1. 功能简介

请在与用户接入网络相连的端口上配置本功能。

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。

### 2. 配置限制和指导

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。

### 3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

```
interface interface-type interface-number
```

- (3) 开启端口角色限制功能。

```
stp role-restriction
```

缺省情况下，端口角色限制功能处于关闭状态。

## 2.24.6 配置 TC-BPDU 传播限制功能

### 1. 功能简介

请在与用户接入网络相连的端口上配置本功能。

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。

### 2. 配置限制和指导

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。

### 3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

```
interface interface-type interface-number
```

- (3) 开启 TC-BPDU 传播限制功能。

```
stp tc-restriction
```

缺省情况下，TC-BPDU 传播限制功能处于关闭状态。

## 2.24.7 配置防 TC-BPDU 攻击保护功能

### 1. 功能简介

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。

## 2. 配置限制和指导

建议不要关闭防 TC-BPDU 攻击保护功能。

## 3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 开启防 TC-BPDU 攻击保护功能。

```
stp tc-protection
```

缺省情况下，防 TC-BPDU 攻击保护功能处于开启状态。

- (3) （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。

```
stp tc-protection threshold number
```

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。

## 2.24.8 配置 MSTP 的 PVST 报文保护功能

### 1. 功能简介

本配置在 MSTP 工作模式下才能生效。

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。

### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 开启 MSTP 的 PVST 报文保护功能。

```
stp pvst-bpdu-protection
```

缺省情况下，MSTP 的 PVST 报文保护功能处于关闭状态。

## 2.24.9 Dispute 保护功能

### 1. 功能简介

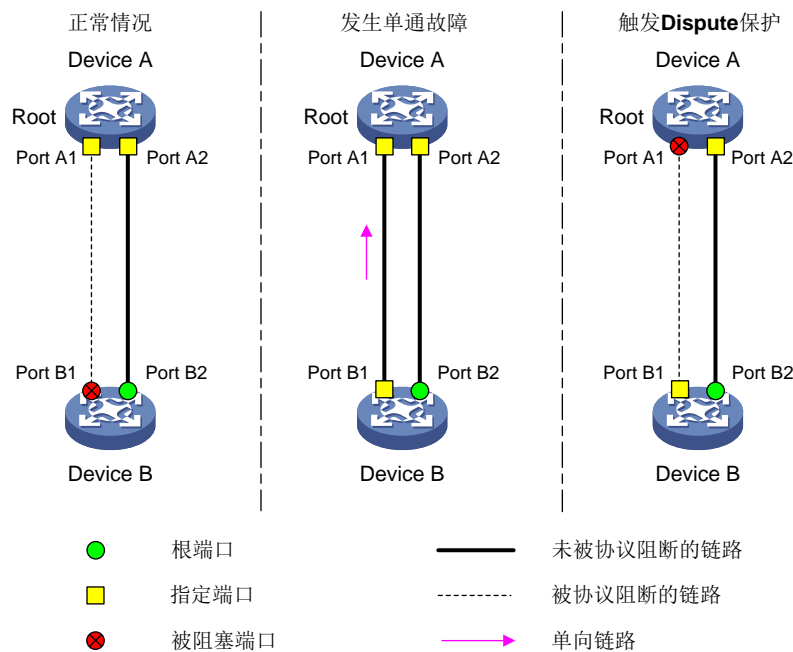
当端口收到指定端口发出的低优先级消息，且发送端口处于 Forwarding 或 Learning 状态时，会触发 Dispute 保护，阻塞端口以防止环路。

Dispute 保护功能是设备默认启用的特性，不需要配置。

如图 2-4 所示，正常情况下，Device A 是根桥，经过生成树计算后，Port B1 被阻塞。如果 Port A1 发生单通故障，即 Port A1 不能发送报文，只能接收报文。Port B1 在一定时间内未收到 Port A1 发送的 BPDU，则 Device B 认为自己是根桥，由 Port B1 发送低优先级 BPDU 到 Port A1。此时，Port

A2 和 Port B2 之间链路正常，Device B 会接收到自己发送 BPDU，导致产生环路。因此当链路出现单通故障后，会触发 Dispute 保护功能，阻塞端口，防止环路。

图2-4 Dispute 保护触发场景



## 2.25 配置生成树的网管功能

### 1. 功能简介

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”。

### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 开启生成树的告警功能。

```
snmp-agent trap enable stp [ new-root | tc ]
```

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。

## 2.26 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2-7 生成树显示和维护

| 操作                    | 命令                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 显示生成树的状态和统计信息         | (独立运行模式)<br><b>display stp</b> [ <b>instance</b> <i>instance-list</i>   <b>vlan</b> <i>vlan-id-list</i> ] [ <b>interface</b> <i>interface-list</i>   <b>slot</b> <i>slot-number</i> [ <b>cpu</b> <i>cpu-number</i> ] ] [ <b>brief</b> ]<br>(IRF模式)<br><b>display stp</b> [ <b>instance</b> <i>instance-list</i>   <b>vlan</b> <i>vlan-id-list</i> ] [ <b>interface</b> <i>interface-list</i>   <b>chassis</b> <i>chassis-number</i> <b>slot</b> <i>slot-number</i> [ <b>cpu</b> <i>cpu-number</i> ] ] [ <b>brief</b> ] |
| 显示生成树端口角色计算的历史信息      | (独立运行模式)<br><b>display stp</b> [ <b>instance</b> <i>instance-list</i>   <b>vlan</b> <i>vlan-id-list</i> ] <b>history</b> [ <b>slot</b> <i>slot-number</i> [ <b>cpu</b> <i>cpu-number</i> ] ]<br>(IRF模式)<br><b>display stp</b> [ <b>instance</b> <i>instance-list</i>   <b>vlan</b> <i>vlan-id-list</i> ] <b>history</b> [ <b>chassis</b> <i>chassis-number</i> <b>slot</b> <i>slot-number</i> [ <b>cpu</b> <i>cpu-number</i> ] ]                                                                                       |
| 显示生成树所有端口收发的TC或TCN报文数 | (独立运行模式)<br><b>display stp</b> [ <b>instance</b> <i>instance-list</i>   <b>vlan</b> <i>vlan-id-list</i> ] <b>tc</b> [ <b>slot</b> <i>slot-number</i> [ <b>cpu</b> <i>cpu-number</i> ] ]<br>(IRF模式)<br><b>display stp</b> [ <b>instance</b> <i>instance-list</i>   <b>vlan</b> <i>vlan-id-list</i> ] <b>tc</b> [ <b>chassis</b> <i>chassis-number</i> <b>slot</b> <i>slot-number</i> [ <b>cpu</b> <i>cpu-number</i> ] ]                                                                                                 |
| 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</b> [ <b>interface</b> <i>interface-type</i> <i>interface-number</i> [ <b>instance</b> <i>instance-list</i> ] ]                                                                                                                                                                                                                                                                                                                                                                         |
| 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</b> [ <b>interface</b> <i>interface-list</i> ]                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

# 目 录

|                                                |      |
|------------------------------------------------|------|
| 1 LLDP.....                                    | 1-1  |
| 1.1 LLDP 简介 .....                              | 1-1  |
| 1.1.1 LLDP 代理和桥模式 .....                        | 1-1  |
| 1.1.2 LLDP 报文 .....                            | 1-2  |
| 1.1.3 LLDPDU .....                             | 1-3  |
| 1.1.4 TLV .....                                | 1-3  |
| 1.1.5 管理地址 .....                               | 1-6  |
| 1.1.6 LLDP 的工作模式 .....                         | 1-6  |
| 1.1.7 LLDP 报文的收发工作机制 .....                     | 1-6  |
| 1.1.8 协议规范 .....                               | 1-7  |
| 1.2 LLDP 配置限制和指导 .....                         | 1-7  |
| 1.3 LLDP 配置任务简介 .....                          | 1-7  |
| 1.4 开启 LLDP 功能 .....                           | 1-8  |
| 1.5 配置 LLDP 桥模式 .....                          | 1-8  |
| 1.6 配置 LLDP 工作模式 .....                         | 1-8  |
| 1.7 配置接口初始化延迟时间 .....                          | 1-9  |
| 1.8 配置允许发布的 TLV 类型 .....                       | 1-9  |
| 1.9 配置管理地址及其封装格式 .....                         | 1-12 |
| 1.10 配置 LLDP 报文的封装格式 .....                     | 1-12 |
| 1.11 调整 LLDP 报文发送参数 .....                      | 1-13 |
| 1.12 配置 LLDP 报文接收超时时间 .....                    | 1-14 |
| 1.13 配置轮询功能 .....                              | 1-14 |
| 1.14 关闭 LLDP 的 PVID 不一致检查功能 .....              | 1-15 |
| 1.15 配置 LLDP Trap 和 LLDP-MED Trap 功能 .....     | 1-15 |
| 1.16 配置 LLDP 报文的源 MAC 地址为指定的 MAC 地址 .....      | 1-16 |
| 1.17 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 |
| 1.18 LLDP 显示和维护 .....                          | 1-17 |

# 1 LLDP

## 1.1 LLDP简介

LLDP (Link Layer Discovery Protocol, 链路层发现协议) 提供了一种标准的链路层发现方式, 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。LLDP 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) 封装在 LLDPDU (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 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。

### 1.1.1 LLDP 代理和桥模式

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:

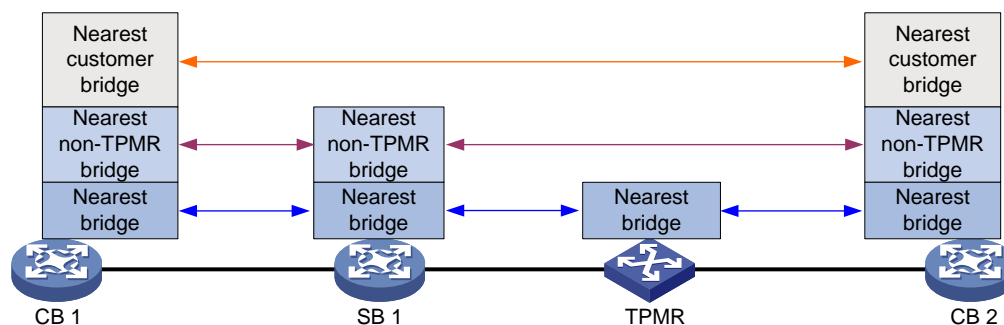
**Nearest Bridge:** 最近桥代理。

**Nearest non-TPMR Bridge:** 最近非 TPMR 桥代理。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。

**Nearest Customer Bridge:** 最近客户桥代理。

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。LLDP 不同类型的代理邻居关系如图 1-1 所示。

图1-1 LLDP 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。

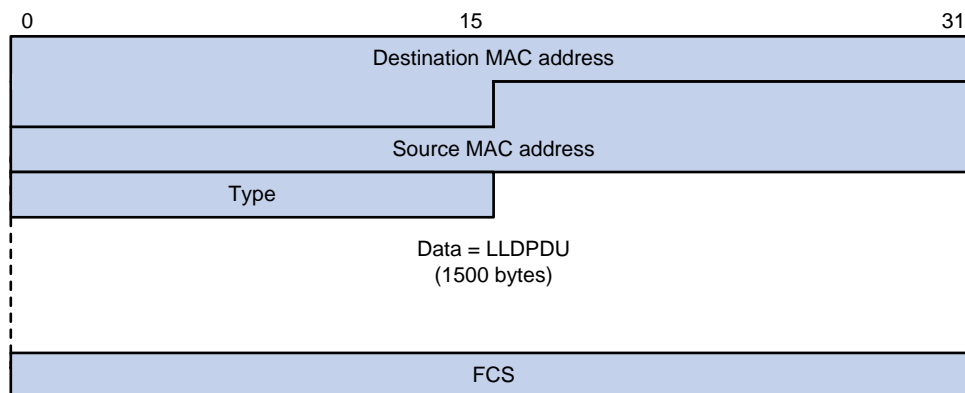
- LLDP 工作于客户桥模式时, 设备可支持最近桥代理、最近非 TPMR 桥代理和最近客户桥代理, 即设备对报文目的 MAC 地址为上述代理的 MAC 地址的 LLDP 报文进行处理, 对报文目的 MAC 地址为其他 MAC 地址的 LLDP 报文进行 VLAN 内透传。
- LLDP 工作于服务桥模式时, 设备可支持最近桥代理和最近非 TPMR 桥代理, 即设备对报文目的 MAC 地址为上述代理的 MAC 地址的 LLDP 报文进行处理, 对报文目的 MAC 地址为其他 MAC 地址的 LLDP 报文进行 VLAN 内透传。

## 1.1.2 LLDP 报文

封装 LLDPDU 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



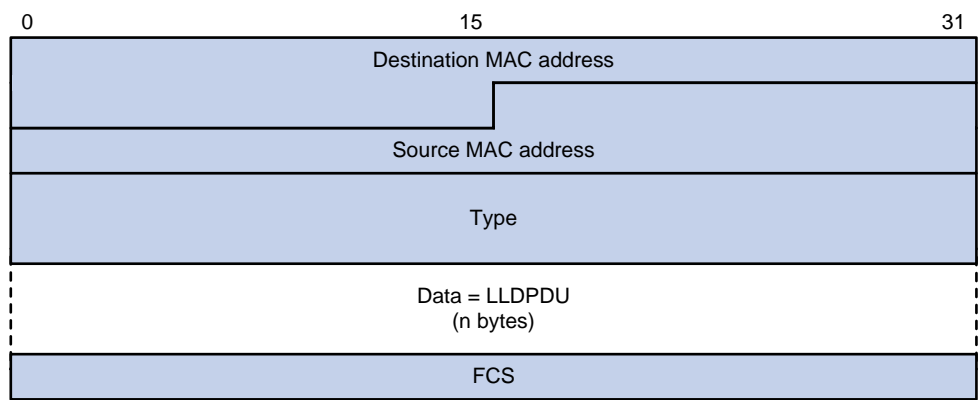
如图 1-2 所示，Ethernet II 格式封装的 LLDP 报文包含如下字段：

- **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。
  - 最近桥代理类型的 LLDP 报文使用组播 MAC 地址 0x0180-c200-000e。
  - 最近客户桥代理类型的 LLDP 报文使用组播 MAC 地址 0x0180-c200-0000。
  - 最近非 TPMR 桥代理类型的 LLDP 报文使用组播 MAC 地址 0x0180-c200-0003。
- **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



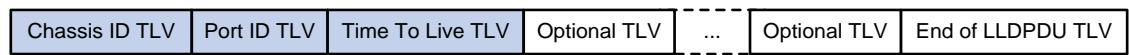
如图 1-3 所示，SNAP 格式封装的 LLDP 报文包含如下字段：

- **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。

1.1.3 LLDPDU

LLDPDU 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。

1.1.4 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                                       | 是      |
| 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。

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。

表1-2 IEEE 802.1 组织定义的 TLV

| TLV 名称                           | 说明                                                                                                                          |
|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| Port VLAN ID(PVID)               | 端口PVID                                                                                                                      |
| 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                             | （暂不支持）数据中心桥能力交换协议（Data Center Bridging Exchange Protocol）                                                                   |
| EVB模块                            | （暂不支持）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。 |
| 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                                                                                                            |

| TLV 名称             | 说明                                              |
|--------------------|-------------------------------------------------|
| ETS Configuration  | （暂不支持）增强传输选择（Enhanced Transmission Selection）配置 |
| ETS Recommendation | （暂不支持）增强传输选择推荐                                  |
| PFC                | （暂不支持）基于优先级的流量控制（Priority-based Flow Control）   |
| APP                | （暂不支持）应用协议（Application Protocol）                |
| QCN                | （暂不支持）量化拥塞通知（Quantized Congestion Notification） |

### 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。

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。

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                | （暂不支持）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                                                                                                                                                                                         |

### 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。

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。

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        | （暂不支持）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 |

| TLV 名称                  | 说明                                       |
|-------------------------|------------------------------------------|
| 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            |

### 1.1.5 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。

### 1.1.6 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。

### 1.1.7 LLDP 报文的收发工作机制

#### 1. 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“QoS”。

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。

## 2. 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。

### 1.1.8 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配置限制和指导

如表 1-5 所示，LLDP 以下配置任务支持在多个接口视图配置。

表1-5 LLDP 配置任务对应的接口视图

| 配置任务                        | 支持配置的接口视图                                                                                                                                            |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| 开启LLDP功能                    | 支持配置的接口视图： <ul style="list-style-type: none"><li>• 二层以太网接口视图</li><li>• 三层以太网接口视图</li><li>• 管理以太网接口视图</li><li>• 二层聚合接口视图</li><li>• 三层聚合接口视图</li></ul> |
| 配置LLDP工作模式                  |                                                                                                                                                      |
| 配置允许发布的TLV类型                |                                                                                                                                                      |
| 配置管理地址及其封装格式                |                                                                                                                                                      |
| 配置LLDP报文的封装格式               |                                                                                                                                                      |
| 配置轮询功能                      |                                                                                                                                                      |
| 配置LLDP Trap和LLDP-MED Trap功能 |                                                                                                                                                      |

## 1.3 LLDP配置任务简介

LLDP 配置任务如下：

- (1) [开启 LLDP 功能](#)
- (2) [配置 LLDP 桥模式](#)
- (3) [配置 LLDP 工作模式](#)
- (4) （可选）[配置接口初始化延迟时间](#)
- (5) （可选）配置 LLDP 报文相关参数
  - [配置允许发布的 TLV 类型](#)
  - [配置管理地址及其封装格式](#)

- [配置 LLDP 报文的封装格式](#)
- [调整 LLDP 报文发送参数](#)
- [配置 LLDP 报文接收超时时间](#)
- (6) (可选) [配置轮询功能](#)
- (7) (可选) [关闭 LLDP 的 PVID 不一致检查功能](#)
- (8) (可选) [配置 LLDP Trap 和 LLDP-MED Trap 功能](#)
- (9) (可选) [配置 LLDP 报文的源 MAC 地址为指定的 MAC 地址](#)
- (10) (可选) [配置设备支持通过 LLDP 生成对端管理地址的 ARP 或 ND 表项](#)

## 1.4 开启LLDP功能

### 1. 配置限制和指导

只有当全局和接口上都开启了 LLDP 功能后，该功能才会生效。  
聚合接口上的 LLDP 开启/关闭配置不会被同步到其成员接口中。

### 2. 配置步骤

- (1) 进入系统视图。  
**system-view**
- (2) 全局开启 LLDP 功能。  
**lldp global enable**  
缺省情况下,设备未全局开启 LLDP 功能。
- (3) 进入接口视图。  
**interface interface-type interface-number**
- (4) 在接口上开启 LLDP 功能。  
**lldp enable**  
缺省情况下，LLDP 功能在接口上处于开启状态。

## 1.5 配置LLDP桥模式

- (1) 进入系统视图。  
**system-view**
- (2) 配置 LLDP 桥模式。
  - 配置 LLDP 桥模式为服务桥模式。  
**lldp mode service-bridge**
  - 配置 LLDP 桥模式为客户桥模式。  
**undo lldp mode service-bridge**缺省情况下，LLDP 桥模式为客户桥模式。

## 1.6 配置LLDP工作模式

- (1) 进入系统视图。

**system-view**

- (2) 进入接口视图。

**interface** *interface-type interface-number*

- (3) 配置 LLDP 的工作模式。

- 在二/三层以太网接口视图或管理以太网接口视图下：

```
lldp [ agent { nearest-customer | nearest-nontpmr } ] admin-status  
{ disable | rx | tx | txrx }
```

以太网接口视图下，未指定 **agent** 参数时，表示配置最近桥代理的工作模式。

- 在二/三层聚合接口视图下：

```
lldp agent { nearest-customer | nearest-nontpmr } admin-status  
{ disable | rx | tx | txrx }
```

聚合接口视图下，只支持配置最近客户桥代理和最近非 TPMR 代理的工作模式。

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。

## 1.7 配置接口初始化延迟时间

### 1. 功能简介

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。

### 2. 配置步骤

- (1) 进入系统视图。

**system-view**

- (2) 配置接口初始化的延迟时间。

```
lldp timer reinit-delay delay
```

缺省情况下，接口初始化的延迟时间为 2 秒。

## 1.8 配置允许发布的TLV类型

- (1) 进入系统视图。

**system-view**

- (2) 进入接口视图。

**interface** *interface-type interface-number*

- (3) 配置接口上允许发布的 TLV 类型。

- 在二层以太网接口视图下：

```
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 | power-over-ethernet | location-id { civic-address
device-type country-code { ca-type ca-value } <1-10> | elin-address
tel-number } } }
```

缺省情况下，最近桥代理允许发布除 Location-id TLV、Port And Protocol VLAN ID TLV、VLAN Name TLV 和 Management VLAN ID TLV 之外所有类型的 TLV。

```
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 } }
lldp tlv-enable dot1-tlv { protocol-vlan-id [ vlan-id ] | vlan-name
[ vlan-id ] | management-vid [ mvlan-id ] }
```

缺省情况下，最近非 TPMR 桥代理不发布任何 TLV。

```
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 } }
lldp tlv-enable dot1-tlv { protocol-vlan-id [ vlan-id ] | vlan-name
[ vlan-id ] | management-vid [ mvlan-id ] }
```

缺省情况下，最近客户桥代理允许发布基本 TLV 和 IEEE 802.1 组织定义 TLV。

- 在三层以太网接口视图下：

```
lldp tlv-enable { basic-tlv { all | port-description |
system-capability | system-description | system-name |
management-address-tlv [ ipv6 ] [ ip-address | interface loopback
interface-number ] } | dot1-tlv { all | link-aggregation } | dot3-tlv
{ all | mac-physic | max-frame-size | power } | med-tlv { all | capability
| inventory | power-over-ethernet | location-id { civic-address
device-type country-code { ca-type ca-value } <1-10> | elin-address
tel-number } } }
```

缺省情况下，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。

```
lldp agent { nearest-nontpmr | nearest-customer } tlv-enable
{ basic-tlv { all | port-description | system-capability |
system-description | system-name | management-address-tlv [ ipv6 ]
[ ip-address ] } | dot1-tlv { all | link-aggregation } }
```

缺省情况下，最近非 TPMR 桥代理不发布任何 TLV；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。

- 在管理以太网接口视图下：

```
lldp tlv-enable { basic-tlv { all | port-description |
system-capability | system-description | system-name |
```



```
management-address-tlv [ ipv6 ] [ ip-address ] } | dot1-tlv { all |
link-aggregation } | dot3-tlv { all | mac-physic | max-frame-size |
power } | med-tlv { all | capability | inventory | power-over-ethernet |
location-id { civic-address device-type country-code { ca-type
ca-value } <1-10> | elin-address tel-number } } }
```

缺省情况下，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。

```
lldp agent { nearest-nontpmr | nearest-customer } tlv-enable
{ basic-tlv { all | port-description | system-capability |
system-description | system-name | management-address-tlv [ ipv6 ]
[ ip-address ] } | dot1-tlv { all | link-aggregation } }
```

缺省情况下，最近非 TPMR 桥代理不发布任何 TLV；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。

- 在二层聚合接口视图下：

```
lldp tlv-enable dot1-tlv { protocol-vlan-id [ vlan-id ] | vlan-name
[ vlan-id ] | management-vid [ mvlan-id ]
lldp agent nearest-nontpmr tlv-enable { basic-tlv { all |
management-address-tlv [ ipv6 ] [ ip-address ] | port-description |
system-capability | system-description | system-name } | dot1-tlv
{ all | port-vlan-id } }
```

缺省情况下，最近非 TPMR 桥代理不发布任何 TLV。

```
lldp agent nearest-customer tlv-enable { basic-tlv { all |
management-address-tlv [ ipv6 ] [ ip-address ] | port-description |
system-capability | system-description | system-name } | dot1-tlv
{ all | port-vlan-id } }
```

缺省情况下，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。

不存在最近桥代理。

- 在三层聚合接口视图下：

```
lldp agent { nearest-customer | nearest-nontpmr } tlv-enable basic-tlv
{ all | management-address-tlv [ ipv6 ] [ ip-address ] |
port-description | system-capability | system-description |
system-name }
```

缺省情况下，最近非 TPMR 桥代理不发布任何 TLV；最近客户桥代理只允许发布基本 TLV。

不存在最近桥代理。

## 1.9 配置管理地址及其封装格式

### 1. 功能简介

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。

### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

```
interface interface-type interface-number
```

- (3) 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：

```
lldp [ agent { nearest-customer | nearest-nontpmr } ] tlv-enable  
basic-tlv management-address-tlv [ ipv6 ] [ ip-address ]
```

- 在三层以太网接口视图下：

```
lldp [ agent { nearest-customer | nearest-nontpmr } ] tlv-enable  
basic-tlv management-address-tlv [ ipv6 ] [ ip-address | interface  
loopback interface-number ]
```

- 在二/三层聚合接口视图下：

```
lldp agent { nearest-customer | nearest-nontpmr } tlv-enable  
basic-tlv management-address-tlv [ ipv6 ] [ ip-address ]
```

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。

- (4) 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：

```
lldp [ agent { nearest-customer | nearest-nontpmr } ]  
management-address-format string
```

- 在二/三层聚合接口视图下：

```
lldp agent { nearest-customer | nearest-nontpmr }  
management-address-format string
```

缺省情况下，管理地址在 TLV 中的封装格式为数字格式。

## 1.10 配置LLDP报文的封装格式

### 1. 功能简介

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。

## 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

```
interface interface-type interface-number
```

- (3) 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：

```
lldp [ agent { nearest-customer | nearest-nontpmr } ] encapsulation  
snap
```

- 在二/三层聚合接口视图下：

```
lldp agent { nearest-customer | nearest-nontpmr } encapsulation snap
```

缺省情况下，LLDP 报文的封装格式为 Ethernet II 格式。

## 1.11 调整LLDP报文发送参数

### 1. 功能简介

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  $TTL = \min(65535, (TTL \text{ 乘数} \times \text{LLDP 报文的发送时间间隔} + 1))$ ，即取 65535 与  $(TTL \text{ 乘数} \times \text{LLDP 报文的发送时间间隔} + 1)$  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。

### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 配置 TTL 乘数。

```
lldp hold-multiplier value
```

缺省情况下，TTL 乘数为 4。

- (3) 配置 LLDP 报文的发送时间间隔。

```
lldp timer tx-interval interval
```

缺省情况下，LLDP 报文的发送时间间隔为 30 秒。

- (4) 配置 LLDP 报文发包限速的令牌桶大小。

```
lldp max-credit credit-value
```

缺省情况下，发包限速令牌桶大小为 5。

- (5) 配置快速发送 LLDP 报文的个数。

```
lldp fast-count count
```

缺省情况下，快速发送 LLDP 报文的个数为 4 个。

- (6) 配置快速发送 LLDP 报文的时间间隔。

```
lldp timer fast-interval interval
```

缺省情况下，快速发送 LLDP 报文的发送时间间隔为 1 秒。

## 1.12 配置LLDP报文接收超时时间

### 1. 功能简介

当需要检测设备是否存在直连邻居时，可以配置本功能启动 LLDP 报文接收超时定时器。在经过超时时间后，如果接口仍未收到 LLDP 报文，则认为该接口不存在 LLDP 邻居，并将该事件发送给 NETCONF 处理。

### 2. 配置限制和指导

LLDP 报文接收超时时间需要大于邻居设备 LLDP 报文的发送间隔，避免误配置导致检测到 LLDP 邻居不存在。

### 3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 配置 LLDP 报文接收超时时间。

```
lldp timer rx-timeout timeout
```

缺省情况下，未配置 LLDP 报文接收超时时间，不上报无 LLDP 邻居事件。

## 1.13 配置轮询功能

### 1. 功能简介

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。

### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

```
interface interface-type interface-number
```

- (3) 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：

```
lldp [ agent { nearest-customer | nearest-nontpmr } ]  
check-change-interval interval
```

- 在二/三层聚合接口视图下：

```
lldp agent { nearest-customer | nearest-nontpmr }  
check-change-interval interval
```

缺省情况下，轮询功能处于关闭状态。

## 1.14 关闭LLDP的PVID不一致检查功能

### 1. 功能简介

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。

### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 关闭 LLDP 的 PVID 不一致检查功能。

```
lldp ignore-pvid-inconsistency
```

缺省情况下，LLDP 的 PVID 不一致检查功能处于开启状态。

## 1.15 配置LLDP Trap和LLDP-MED Trap功能

### 1. 功能简介

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 信息的发送时间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。

### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 进入接口视图。

```
interface interface-type interface-number
```

- (3) 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：

```
lldp [ agent { nearest-customer | nearest-nontpmr } ] notification  
remote-change enable
```

- 在二/三层聚合接口视图下：

```
lldp agent { nearest-customer | nearest-nontpmr } notification  
remote-change enable
```

缺省情况下，LLDP Trap 功能处于关闭状态。

- (4) 在二/三层以太网接口视图或管理以太网接口视图下开启 LLDP-MED Trap 功能。

```
lldp notification med-topology-change enable
```

缺省情况下，LLDP-MED Trap 功能处于关闭状态。

- (5) 退回系统视图。

**quit**

- (6) (可选) 配置 LLDP Trap 和 LLDP-MED Trap 信息的发送时间间隔。

**lldp timer notification-interval** *interval*

缺省情况下, LLDP Trap 和 LLDP-MED Trap 信息的发送时间间隔均为 30 秒。

## 1.16 配置LLDP报文的源MAC地址为指定的MAC地址

### 1. 功能简介

本功能用来配合设备支持通过 LLDP 生成对端管理地址的 ARP 或 ND 表项功能使用, 以保证设备发送报文的源 MAC 地址为三层以太网子接口的 MAC 地址, 而不是当前接口的 MAC 地址, 确保对端学习到正确的 ARP/ND 表项。

配置本特性后, LLDP 报文的源 MAC 地址为终结了本功能指定的 VLAN 的三层以太网子接口的 MAC 地址。如果该 VLAN 未被任何三层以太网子接口终结, 则 LLDP 报文源 MAC 地址为主接口的 MAC 地址。有关 VLAN 终结的详细介绍, 请参见“二层技术-以太网交换配置指导”中的“VLAN 终结”。

### 2. 配置步骤

- (1) 进入系统视图。

**system-view**

- (2) 进入三层以太网接口视图。

**interface** *interface-type interface-number*

- (3) 配置 LLDP 报文源 MAC 地址为三层以太网子接口的 MAC 地址。

**lldp source-mac** *vlan vlan-id*

缺省情况下, LLDP 报文源 MAC 地址为当前接口的 MAC 地址。

## 1.17 配置设备支持通过LLDP生成对端管理地址的ARP或ND表项

### 1. 功能简介

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。

本功能需要与配置 LLDP 报文的源 MAC 地址为指定的 MAC 地址功能配合使用, 使设备发送报文的源 MAC 地址为三层以太网子接口的 MAC 地址, 而不是当前接口的 MAC 地址, 确保 LLDP 邻居能学习到正确的 ARP/ND 表项。

配置本功能时, 如果携带了 **vlan vlan-id** 参数, 则生成的 ARP 表项或 ND 表项中的出接口为终结了该 VLAN 的三层以太网子接口; 如果该 VLAN 未被任何三层以太网子接口终结, 则生成的表项中的出接口为主接口。如果不指定 **vlan vlan-id** 参数, 则生成的表项中的出接口为主接口。有关 VLAN 终结的详细介绍, 请参见“二层技术-以太网交换配置指导”中的“VLAN 终结”。

### 2. 配置步骤

- (1) 进入系统视图。

**system-view**

- (2) 进入三层以太网接口视图。

```
interface interface-type interface-number
```

- (3) 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。

```
lldp management-address { arp-learning | nd-learning } [ vlan vlan-id ]
```

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。

## 1.18 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1-6 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 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 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> ] ] |
| 显示LLDP的统计信息      | <b>display lldp statistics</b> [ <b>global</b>   [ <b>interface</b> <i>interface-type 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 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 interface-number</i> ] [ <b>agent</b> { <b>nearest-bridge</b>   <b>nearest-customer</b>   <b>nearest-nontpmr</b> } ] ]                                                                                          |

# 目 录

|                                         |     |
|-----------------------------------------|-----|
| 1 普通二层转发 .....                          | 1-1 |
| 1.1 普通二层转发简介 .....                      | 1-1 |
| 1.2 vSystem 相关说明 .....                  | 1-1 |
| 1.3 普通二层转发显示和维护 .....                   | 1-1 |
| 2 快速二层转发 .....                          | 2-1 |
| 2.1 快速二层转发简介 .....                      | 2-1 |
| 2.2 vSystem 相关说明 .....                  | 2-1 |
| 2.3 关闭快速二层转发时对 VLAN ID 检查功能 .....       | 2-1 |
| 2.4 快速二层转发显示和维护 .....                   | 2-1 |
| 3 Bridge 转发 .....                       | 3-1 |
| 3.1 Bridge 转发简介 .....                   | 3-1 |
| 3.1.1 Bridge 转发模式 .....                 | 3-1 |
| 3.1.2 Inline 转发 .....                   | 3-1 |
| 3.2 配置 Bridge 转发 .....                  | 3-2 |
| 3.2.1 配置 Inline 转发 .....                | 3-2 |
| 3.2.2 配置 Bypass 功能 .....                | 3-3 |
| 4 快速 Bridge 转发 .....                    | 4-1 |
| 4.1 快速 Bridge 转发简介 .....                | 4-1 |
| 4.2 关闭快速 Bridge 转发时对 VLAN ID 检查功能 ..... | 4-1 |
| 4.3 快速 Bridge 转发显示和维护 .....             | 4-1 |



# 1 普通二层转发

## 1.1 普通二层转发简介

如果设备接收到的报文的目的 MAC 地址匹配三层接口的 MAC 地址, 则通过设备的三层接口进行三层转发; 否则通过设备的二层接口进行二层转发。

二层转发根据报文的目的 MAC 地址查找 MAC 地址表, 得到报文的出接口, 然后将报文发送出去。普通二层转发是设备默认启用的特性, 不需要配置。

## 1.2 vSystem相关说明

vSystem 支持本特性的所有功能。有关 vSystem 的详细介绍请参见“虚拟化技术配置指导”中的“vSystem”。

## 1.3 普通二层转发显示和维护

在任意视图下执行 **display** 命令可以显示二层转发过程中的统计信息, 查看转发的结果。

在用户视图下执行 **reset** 命令可以清除二层转发的统计信息。

表1-1 普通二层转发显示和维护

| 操作         | 命令                                                                                                          |
|------------|-------------------------------------------------------------------------------------------------------------|
| 显示二层转发统计信息 | <b>display mac-forwarding statistics</b> [ <b>interface</b> <i>interface-type</i> <i>interface-number</i> ] |
| 清除二层转发统计信息 | <b>reset mac-forwarding statistics</b>                                                                      |

## 2 快速二层转发

### 2.1 快速二层转发简介

快速二层转发采用高速缓存来处理报文，并采用基于数据流的技术，可以大大提高转发效率。

快速二层转发用源 IP 地址、源端口号、目的 IP 地址、目的端口号、协议号、输入接口、输出接口和 VLAN 来标识一条数据流。在二层转发过程中，会根据设备规则，对需要进行三层业务处理的报文，获取其 IP 地址等信息，生成 IP 快速转发表。当一条数据流的第一个报文转发后，会在高速缓存中生成相应的转发信息，该数据流后续报文的转发就可以通过直接查找快速转发表进行转发。这样便大大缩减了报文的排队流程，减少报文的转发时间，提高报文的转发速率。

快速二层转发是设备默认启用的特性，不需要配置。

### 2.2 vSystem相关说明

非缺省 vSystem 不支持本特性中的配置关闭快速二层转发时对 VLAN ID 检查功能。

### 2.3 关闭快速二层转发时对VLAN ID检查功能

#### 1. 功能简介

报文携带的 VLAN ID 是设备判断其所属 TCP 会话的依据之一。在防火墙双机热备的组网环境下，有时需要报文在主备设备间传递后仍可以匹配到同一个会话中，而主备设备上报文入接口所属 VLAN 可能不同，此时可以关闭对 VLAN ID 的检查以保证报文在主备设备之间传递之后能够匹配到同一会话。

#### 2. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 关闭快速二层转发时对 VLAN ID 检查功能。

```
undo mac fast-forwarding check-vlan-id
```

缺省情况下，快速二层转发时对 VLAN ID 字段的检查功能处于开启状态。

### 2.4 快速二层转发显示和维护

在任意视图下执行 **display** 命令可以显示快速二层转发表信息。

表2-1 快速二层转发显示和维护

| 操作            | 命令                                                                                                                                                                                                                                                                                                                                                                                |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 显示IP快速转发表信息   | <p>(独立运行模式)</p> <pre><b>display mac-forwarding cache ip</b> [ <i>ip-address</i> ] [ <b>slot</b> <i>slot-number</i> ] [ <b>cpu</b> <i>cpu-number</i> ] ]</pre> <p>(IRF模式)</p> <pre><b>display mac-forwarding cache ip</b> [ <i>ip-address</i> ] [ <b>chassis</b> <i>chassis-number</i> <b>slot</b> <i>slot-number</i> ] [ <b>cpu</b> <i>cpu-number</i> ] ]</pre>                   |
| 显示分片报文快速转发表信息 | <p>(独立运行模式)</p> <pre><b>display mac-forwarding cache ip fragment</b> [ <i>ip-address</i> ] [ <b>slot</b> <i>slot-number</i> ] [ <b>cpu</b> <i>cpu-number</i> ] ]</pre> <p>(IRF模式)</p> <pre><b>display mac-forwarding cache ip fragment</b> [ <i>ip-address</i> ] [ <b>chassis</b> <i>chassis-number</i> <b>slot</b> <i>slot-number</i> ] [ <b>cpu</b> <i>cpu-number</i> ] ]</pre> |
| 显示IPv6快速转发表信息 | <p>(独立运行模式)</p> <pre><b>display mac-forwarding cache ipv6</b> [ <i>ipv6-address</i> ] [ <b>slot</b> <i>slot-number</i> ] [ <b>cpu</b> <i>cpu-number</i> ] ]</pre> <p>(IRF模式)</p> <pre><b>display mac-forwarding cache ipv6</b> [ <i>ipv6-address</i> ] [ <b>chassis</b> <i>chassis-number</i> <b>slot</b> <i>slot-number</i> ] [ <b>cpu</b> <i>cpu-number</i> ] ]</pre>           |

# 3 Bridge 转发

## 3.1 Bridge转发简介

Bridge 转发是指用户创建 Bridge 实例后，根据 Bridge 实例中添加成员类型不同，实现报文基于 VLAN 或者端口的安全转发功能。

### 3.1.1 Bridge 转发模式

根据报文的转发特征，Bridge 转发有下列几种转发模式：

- 反射模式：报文从同一接口收发。
- 透传模式：报文从一个接口接收，从另一个接口发送。
- 黑洞模式：报文从一个接口接收，处理完后被丢弃。

### 3.1.2 Inline 转发

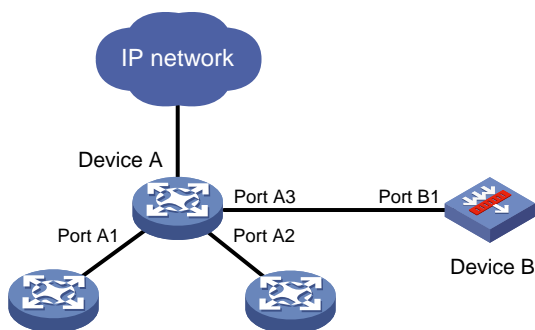
反射模式 Bridge 转发、透传模式 Bridge 转发和黑洞模式 Bridge 转发统称为 Inline 转发。

Inline 转发是在数据链路层对流量进行安全监控的一种技术。目前这种技术主要应用在安全产品上，通过 QoS 策略将经过设备的二层网络流量引流到安全产品上，由安全产品过滤后再进行转发。

#### 1. 反射/黑洞模式 Bridge 转发组网部署

反射/黑洞模式 Bridge 转发中，Device B 与 Device A 通过一个物理接口通信。对于反射模式 Bridge 转发，Device B 通过同一接口完成报文的收发；对于黑洞模式 Bridge 转发，Device B 收到报文后，先处理完安全业务，然后丢弃该报文。反射/黑洞模式 Bridge 转发一般适用于 Device B 旁挂的组网部署，Device A 可直接接入网络。反射/黑洞模式 Bridge 转发如图 3-1 所示。

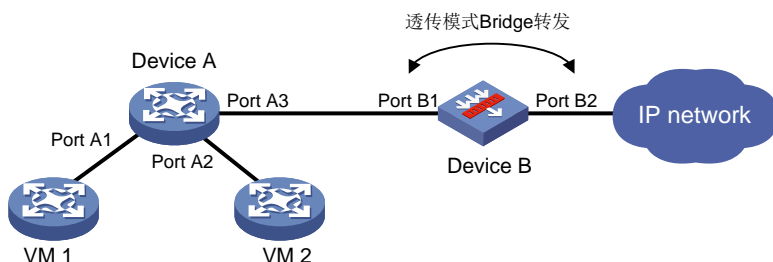
图3-1 反射/黑洞模式 Bridge 转发



#### 2. 透传模式 Bridge 转发组网部署

透传模式 Bridge 转发中，Device B 在两个接口间转发报文，其中一个接口用于收报文，另外一个接口用于发报文。透传模式 Bridge 转发一般适用于 Device B 直连的组网部署，Device A 通过 Device B 接入网络。透传模式 Bridge 转发如图 3-2 所示。

图3-2 透传模式 Bridge 转发



### 3. Inline 转发报文处理流程

在图 3-1、图 3-2 中，以 VM（Virtual Machine，虚拟机）间的报文交互为例，Inline 转发过程如下：

- (1) VM 1 与 VM 2 的内部流量通过 Device A 转发，Device A 将收到的报文引流到与之相连的 Device B 上。
- (2) Device B 将收到的 IP 报文交给安全业务进行处理，其他报文直接返回给 Device A。
- (3) Device B 处理通过安全业务过滤的报文，根据报文信息建立对应的转发表项，并将报文返回给 Device A。

## 3.2 配置Bridge转发

### 3.2.1 配置 Inline 转发

#### 1. 配置限制和指导

透传模式 Bridge 转发实例可以通过命令行手工创建，也可以通过插入硬件 Bypass 子卡后由设备自动生成。

在安全设备上配置 Inline 转发时，需要关闭与其连接的交换机上对应互连口的 MAC 地址学习功能，否则可能造成 MAC 地址频繁迁移。

向 Bridge 转发实例中添加接口时需要注意：

- 每个反射/黑洞模式 Bridge 转发实例只能添加一个接口。
- 每个手工创建的透传模式 Bridge 转发实例只能添加两个接口，且这两个接口的类型必须保持一致。
- 对于自动创建的透传模式 Bridge 转发实例，设备在自动创建 Bridge 转发实例后，会自动将子卡上的一对接口添加到此 Bridge 转发实例中，且自动创建的透传模式 Bridge 转发实例不支持手工添加或删除接口。
- 将三层以太网接口加入 Bridge 转发实例后，该接口将只能用于二层转发，不能进行三层转发。因此，请不要在 Bridge 转发实例的三层以太网接口上配置 IP 地址等三层业务相关的功能，即使配置了也无法生效。

#### 2. 配置步骤

- (1) 进入系统视图。

**system-view**

- (2) （可选）配置 Inline 转发时忽略报文的隧道封装。

**bridge tunnel-encapsulation skip**

缺省情况下，Inline 转发时未忽略报文的隧道封装。

- (3) 手工创建不同模式的转发实例，并进入 Bridge 视图。

- 创建反射模式 Bridge 转发实例，并进入 Bridge 视图。

**bridge bridge-index reflect**

- 创建透传模式 Bridge 转发实例，并进入 Bridge 视图。

**bridge bridge-index forward**

- 创建黑洞模式 Bridge 转发实例，并进入 Bridge 视图。

**bridge bridge-index blackhole**

- (4) 向 Bridge 转发实例中添加接口。

**add interface interface-type interface-number**

缺省情况下，手工创建的 Bridge 转发实例中未添加任何接口。

### 3.2.2 配置 Bypass 功能

#### 1. 功能简介

在 Inline 转发模式下，配置 Bypass 功能，用户流量可以不经过安全业务或者安全设备处理，直接被处理。

内部 Bypass 功能指用户流量经过安全设备 Device，但不进行安全业务处理。安全设备会根据配置的 Inline 转发模式，选择对应的接口将用户流量直接转发或者丢弃。

#### 2. 配置限制和指导

多次配置 **bypass enable** 命令，最后一次执行的配置生效。

#### 3. 配置内部 Bypass 功能

- (1) 进入系统视图。

**system-view**

- (2) 进入 Bridge 视图。

- 进入反射模式 Bridge 视图。

**bridge bridge-index reflect**

- 进入自动创建的透传模式 Bridge 视图。

**bridge bridge-index**

- 进入手工创建的透传模式 Bridge 视图。

**bridge bridge-index forward**

- 进入黑洞模式 Bridge 视图。

**bridge bridge-index blackhole**

- (3) 配置内部 Bypass 功能。

**bypass enable**

缺省情况下，Bypass 功能处于关闭状态。

# 4 快速 Bridge 转发

## 4.1 快速Bridge转发简介

快速 Bridge 转发采用高速缓存来处理报文，采用了基于数据流的技术，可以大大提高转发效率。

快速 Bridge 转发用源 IP 地址、源端口号、目的 IP 地址、目的端口号、协议号、输入接口、输出接口和 VLAN 来标识一条数据流。在二层转发过程中，会根据设备规则，对需要进行三层业务处理的报文，获取其 IP 地址等信息，生成 IP 快速转发表。当一条数据流的第一个报文转发后，会在高速缓存中生成相应的转发信息，该数据流后续报文的转发就可以通过直接查找快速转发表进行转发。这样便大大缩减了报文的排队流程，减少报文的转发时间，提高报文的转发速率。

快速 Bridge 转发是设备默认启用的特性，不需要配置。

## 4.2 关闭快速Bridge转发时对VLAN ID检查功能

### 1. 功能简介

报文携带的 VLAN ID 是设备判断其所属 TCP 会话的依据之一。在防火墙双机热备的组网环境下，有时需要报文在主备设备间传递后仍可以匹配到同一个会话中，而主备设备上报文入接口所属 VLAN 可能不同，此时可以关闭对 VLAN ID 的检查以保证报文在主备设备之间传递之后能够匹配到同一会话。

### 2. 配置限制和指导

仅 Inline 转发会对 VLAN ID 进行检查，设备使用跨 VLAN 模式 Bridge 进行转发时无需配置本功能，因为跨 VLAN 模式 Bridge 转发不对 VLAN ID 进行检查，本功能对跨 VLAN 模式 Bridge 转发不生效。在防火墙双机热备的组网环境下，无论是否配置本功能，使用跨 VLAN 模式 Bridge 进行快速转发时，报文在主备设备间传递后仍可以匹配到同一个会话中。

### 3. 配置步骤

- (1) 进入系统视图。

```
system-view
```

- (2) 关闭快速 Bridge 转发时对 VLAN ID 检查功能。

```
undo bridge fast-forwarding check-vlan-id
```

缺省情况下，快速二层转发时对 VLAN ID 字段的检查功能处于开启状态。

## 4.3 快速Bridge转发显示和维护

在任意视图下执行 **display** 命令可以显示快速 Bridge 转发表信息。

表4-1 快速 Bridge 转发显示和维护

| 操作                       | 命令                                                                                                                                                                                                                                                                                                                                                                  |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 显示Bridge转发创建的IP快速转发表信息   | <p>(独立运行模式)</p> <p><b>display bridge cache ip inline</b> [ <i>ip-address</i> ] [ <b>slot</b> <i>slot-number</i> [ <b>cpu</b> <i>cpu-number</i> ] ]</p> <p>(IRF模式)</p> <p><b>display bridge cache ip inline</b> [ <i>ip-address</i> ] [ <b>chassis</b> <i>chassis-number</i> <b>slot</b> <i>slot-number</i> [ <b>cpu</b> <i>cpu-number</i> ] ]</p>                   |
| 显示Bridge转发创建的分片报文快速转发表信息 | <p>(独立运行模式)</p> <p><b>display bridge cache ip fragment inline</b> [ <i>ip-address</i> ] [ <b>slot</b> <i>slot-number</i> [ <b>cpu</b> <i>cpu-number</i> ] ]</p> <p>(IRF模式)</p> <p><b>display bridge cache ip fragment inline</b> [ <i>ip-address</i> ] [ <b>chassis</b> <i>chassis-number</i> <b>slot</b> <i>slot-number</i> [ <b>cpu</b> <i>cpu-number</i> ] ]</p> |
| 显示Bridge转发创建的IPv6快速转发表信息 | <p>(独立运行模式)</p> <p><b>display bridge cache ipv6 inline</b> [ <i>ipv6-address</i> ] [ <b>slot</b> <i>slot-number</i> [ <b>cpu</b> <i>cpu-number</i> ] ]</p> <p>(IRF模式)</p> <p><b>display bridge cache ipv6 inline</b> [ <i>ipv6-address</i> ] [ <b>chassis</b> <i>chassis-number</i> <b>slot</b> <i>slot-number</i> [ <b>cpu</b> <i>cpu-number</i> ] ]</p>           |