



浪潮网络 CN12800-G 系列产品 典型配置手册

手册版本: v6.2

软件版本:

发布日期: 2024-11-8

声 明

Copyright © 2016-2024 浪潮网络科技(山东)有限公司及其许可者。保留一切权利。

非经本公司书面许可，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。

inspur 浪潮 为浪潮网络科技(山东)有限公司的商标。

对于本手册中出现的其他所有商标或注册商标，由各自的所有人拥有。

由于产品版本升级或其他原因，本手册内容会不定期进行更新。

本手册仅作为使用指导，本手册中的所有陈述、信息和建议不构成任何明示或暗示的担保。

浪潮网络科技(山东)有限公司对使用本手册或使用本公司产品导致的任何特殊、附带、偶然或间接的损害不承担责任，包括但不限于商业利润损失、数据或文档丢失产生的损失，因遭受网络攻击、黑客攻击、病毒感染等造成的产品工作异常、信息泄露。

约 定

命令行格式约定

格式	意义
粗体	命令行关键字（命令中保持不变、必须照输的部分）采用粗体表示
斜体	命令行参数（命令中必须由实际值进行替代的部分）采用斜体表示
[]	用“[]”括起来的部分在命令配置时是可选的
{ }	用“{ }”括起来的部分在命令配置时可出现一次或多次
(x y ...)	表示从两个或多个选项中选取一个
#	由“#”号开始的行表示为注释行

各类标志约定

格式	意义
 注意	表示操作中必须注意的信息，如果忽视这类信息，可能导致数据丢失、功能失效、设备损坏或不可预知的结果。
 说明	表示对操作内容的描述进行强调和补充。

目 录

1 常用维护命令行介绍	1
1.1 登陆设备	1
1.2 查看设备信息	2
1.3 软件版本升级	2
1.4 清除配置	9
2 基本二三层转发配置案例	1
2.1 二层转发简介	1
2.2 三层转发简介	2
3 端口聚合典型配置案例	1
3.1 端口聚合简介	1
3.2 动态端口聚合配置案例	3
4 端口镜像典型配置案例	1
4.1 端口镜像简介	1
4.2 本地端口镜像配置案例	2
4.3 反射口实现的远程端口镜像配置案例	4
4.4 出端口实现的远程端口镜像	8
5 端口限速典型配置案例	1
5.1 端口限速简介	1
5.2 配置案例	1
6 端口隔离典型配置案例	1
6.1 端口隔离简介	1
6.2 配置案例	1
7 PVLAN 典型配置案例	1
7.1 PVLAN 简介	1
7.2 配置案例	1
8 QinQ 典型配置案例	1
8.1 QinQ 简介	1
8.2 基本 QinQ 配置案例	1
8.3 灵活 QinQ 配置案例	4
9 ARP 防护典型配置案例	1
9.1 ARP 防护简介	1
9.2 ARP 报文一致性检测配置案例	2

9.3 ARP 用户合法性配置案例	4
9.4 ARP 网关保护配置案例	6
10 路由协议典型配置案例	1
10.1 路由协议简介	1
10.2 静态路由配置案例	2
10.3 RIP 路由配置案例	5
10.4 OSPF 典型配置案例	8
10.5 OSPF 多进程典型配置案例	11
10.6 策略路由配置案例	14
11 DHCP 典型配置案例	1
11.1 DHCP 简介	1
11.2 DHCP Server 配置案例	2
11.3 DHCP 中继配置案例	4
11.4 DHCP Snooping 配置案例	6
12 QoS 典型配置案例	1
12.1 QoS 简介	1
12.2 配置案例	2
13 ACL 典型配置案例	4
13.1 ACL 简介	4
13.2 IPv4 ACL 典型配置案例	6
13.3 IPv6 ACL 典型配置案例	7
13.4 MAC ACL 典型配置案例	10
13.5 MAC/IPv4 绑定 ACL 典型配置案例	13
14 生成树典型配置案例	1
14.1 生成树简介	1
14.2 STP 配置案例	2
14.3 RSTP 配置案例	6
14.4 MSTP 配置案例	8
15 VRRP 典型配置案例	1
15.1 VRRP 简介	1
15.2 VRRP 配置案例	1
16 VSM 典型配置案例	1
16.1 VSM 简介	1
16.2 VSM 主备选举	3
16.3 VSM 的配置同步	3

16.4 VSM 维护	4
16.5 VSM 典型配置案例	5
17 VRF 典型配置案例	1
17.1 VRF 简介	1
17.2 VRF 典型配置案例	1

1 常用维护命令行介绍

1.1 登陆设备

1.1.1 SSH 方式登陆

在交换机上开启 SSH 后，就可以在串口终端上输入设备的管理地址、用户名（初始用户名 **admin**）和密码（初始密码 **Admin@default666**）登录设备。

```
<INSPUR>conf-mode
[INSPUR]ssh enable
[INSPUR]
```

1.1.2 Telnet 方式登陆

- 使用密码登陆

```
<INSPUR>conf-mode
[INSPUR]telnet enable
[INSPUR]
```

在交换机上开启 Telnet 后，就可以在串口终端上输入设备的管理地址和密码登录设备，终端信息显示如下：

```
User Access Verification
Password:
<INSPUR>
```

- 使用用户名和密码登录

```
<INSPUR>conf-mode
[INSPUR]line vty
[INSPUR]authentication mode username
```

在交换机上开启 Telnet 后，就可以在串口终端输入设备的管理地址、用户名（初始用户名 **admin**）和密码（初始密码 **Admin@default666**）登录设备，终端信息显示如下：

```
User Access Verification
Username:admin
Password:Admin@default666
<INSPUR>
```

1.2 查看设备信息

表 1-1 设备信息

项目	说明
进入配置视图	<INSPUR>conf-mode
查看当前可执行的命令，键入“？”	<INSPUR>?
从当前视图退回到上一级视图	[INSPUR]exit <INSPUR>exit
查看当前使用版本	<INSPUR>show version
查看设备当前配置	<INSPUR> show running-config
查看已创建 VLAN	<INSPUR>show vlan
查看 vlan-if 口	<INSPUR>show ip interface brief
查看 boot-file	<INSPUR>show boot-file
重启设备	<INSPUR>reboot
开启和关闭 SSH	[INSPUR]ssh enable [INSPUR]no ssh enable
开启和关闭 Telnet	[INSPUR]telnet enable [INSPUR]no telnet enable
开启和关闭 Telnet 用户名密码方式	[INSPUR]line vty [INSPUR-line] authentication mode username [INSPUR-line] authentication mode none

1.3 软件版本升级

1.3.1 Conboot 模式下操作

升级版本之前确保用户终端和交换机正确连接：使用主机配置的串口线与交换机的 Console 口连

接，使用网线连接主机的网卡和交换机的物理端口。在主机上开启 TFTP 服务器。

设备上电或者重启时，终端上会显示下面的信息，打印到 **will boot in 3** 这一句时，提示我们是否要进入 **Conboot** 菜单，并且提供 **3** 秒的等待时间。在这一秒内键入 **<Ctrl+B>**，系统会提示：

```
please enter the password:
```

输入正确的密码后就能进入 **boot** 菜单，交换机缺省未设置密码，键入 **Enter** 后显示 **Conboot** 菜单，之后就可以根据下面的提示升级设备的软件版本。

- **Conboot** 模式下升级版本。

```
<INSPUR>reboot
System configuration has been modified. Save? (Y/N) [N]: y
Proceed with reboot? (Y/N) [N]: y
System reboot...
The system is going down NOW!
Sent SIGTERM to all processes
Sent SIGKILL to all processes
Requesting system reboot

System start booting...
Booting Basic ConBoot....

*****
* *
* ConBoot, basic Version 1.25.05 *
* *
*****
Power On reset config = 0x000000000D4093B
Compiled Date:Compiled on Wed, 23 Dec 2015 00:43:03 +0800
Type [CTRL+F] to enter board setup 0
Trying configuration: CMD:0x18
Dram page thrash test PASSED.
Running DRAM mats
Press Ctrl+E to enter Basic boot menu...
Checking system image...
Booting Normal Extend ConBoot....,180000
*****
* *
* ConBoot, extend Version 1.25.05 *
* *
```

```
*****
Compiled Date: Compiled on Wed, 23 Dec 2015 00:43:37 +0800
Detected CPU Rev B1 [Revision ID 0x3]
Detected CPU XLS408B [CPU ID 0x4e]
Bridge Device Mask: fff0f907
Capacity of DDRII Memory:1024MB
CPU Frequency = 1000.000000MHz
Detected 8 online CPU(s), map = 0xff
CPLD Version :3.0 2014-12-24
PCB Version :B
NAND device: Manufacturer ID: 0xad, Chip ID: 0xdc (Hynix NAND 512MiB 3,3V 8-bit)
Scanning JFFS2 FS: . done.

configuring gmac0 in byte mode @ 125MHz (1000Mbps): full duplex mode
configuring gmac1 in byte mode @ 125MHz (1000Mbps): full duplex mode
B1_XLS408B @ ATX_I $
Press Ctrl+B to enter extend boot menu...2----->设备启动到此处,
在 3 秒内 Ctrl+B 截断
please enter the password: ----->回车键进入
下面的主菜单
=====<EXTEND-ConBoot-MENU>=====
<1> Boot System
<2> Enter Serial SubMenu
<3> Enter Ethernet SubMenu
<4> File Control
<5> Modify ConBoot Password
<6> Skip Current System Configuration
<7> ConBoot Operation Menu
<8> Skip Current System Password
<0> reboot
=====
enter your choice (0 - 7):3----->键入
3, 进入子菜单
=====<GIGEERNET SUB-MENU>=====
<1> Download Application Program To SDRAM And Run
<2> Modify Gigeernet Parameter
<3> Update Main Application File
<4> Update Backup Application File
<0> Exit To Main Menu
=====
```

```

enter your choice (0 - 4):2----->键
入 2, 修改参数
=====<GIGEERNET PARAMETER SET>=====
note:
'+'=go to next field
'-' = Go to previous field.
Ctrl+D = Quit.
=====
Load File Name: CN12800-G-S111C011D007.bin ----->
需要升级的版本名称
Server IP Address:10.24.9.99 ----->存放上述
版本的主机地址
Local IP Address:10.24.14.15 ----->设备所在
网段的地址即可
Gateway IP Address:10.24.0.1 ----->
设备的网关地址
Net                                                    Mask:255.255.0.0
----->主机掩码
change successfully!
=====<GIGEERNET SUB-MENU>=====
<1> Download Application Program To SDRAM And Run
<2> Modify Gigeernet Parameter
<3> Update Main Application File
<4> Update Backup Application File
<0> Exit To Main Menu
=====
enter your choice (0 - 4):3----->键
入 3, 升级版本
Downloading [CN12800-G-S111C011D007.bin].
Server IP : 10.24.17.1
Bytes downloaded: 45386636
tftpc: download done. Size [45386636] @ Addr [0x20000000]

Checking system image...
System updating, please don't power off!
Writing:*.....
.....
.....
.....
.....
writtenlen = 45386636
    
```

```

update successfully
=====<GIGEERNET SUB-MENU>=====
<1> Download Application Program To SDRAM And Run
<2> Modify Gigeernet Parameter
<3> Update Main Application File
<4> Update Backup Application File
<0> Exit To Main Menu
=====
enter your choice (0 - 4):0----->升级成功后键入 0，
退回到主菜单
=====<EXTEND-ConBoot-MENU>=====
<1> Boot System
<2> Enter Serial SubMenu
<3> Enter Ethernet SubMenu
<4> File Control
<5> Modify ConBoot Password
<6> Skip Current System Configuration
<7> ConBoot Operation Menu
<8> Skip Current System Password
<0> reboot
=====
enter your choice (0 - 7):0----->键入 0，重启设备，设备起来后所使用的版本即
为上述升级版本

```

- **Conboot** 模式下设置主用启动版本。

当设备里存放了多个版本时，可以通过设置某个版本为主用启动版本，使设备下次启动时使用该版本。

```

=====<EXTEND-ConBoot-MENU>=====
<1> Boot System
<2> Enter Serial SubMenu
<3> Enter Ethernet SubMenu
<4> File Control
<5> Modify ConBoot Password
<6> Skip Current System Configuration
<7> ConBoot Operation Menu
<8> Skip Current System Password
<0> reboot
=====
enter your choice (0 - 7):4----->主菜单模式下键入 4，

```

```
进入文件控制菜单
=====<File CONTROL>=====
<1> Display All File(s)
<2> Set Application File type
<3> Delete File
<4> Format Partition
<0> Exit To Main Menu
=====
enter          your          choice          (0
4):2----->键入 2
Display all file(s) in cfa0:
'M'= main 'B'=backup
=====
NO.           filename           size           type
-----
1: CN12800-G-S111C010D003.bin 45381504
2: CN12800-G-S111C011D007.bin 45386636 M
=====
enter          your
chioce:1----->键入版本编
号
=====
Modify the file attribute:
<1> +Main
<2> -Main
<3> +Backup
<4> -Backup
<0> Exit
=====
Enter your choice (0-4):1----->设置所选版本为主，下次启动时使
用该版本
change successfully!
=====<File CONTROL>=====
<1> Display All File(s)
<2> Set Application File type
<3> Delete File
<4> Format Partition
<0> Exit To Main Menu
=====
enter your choice (0 - 4):0----->键入 0,
回退到主菜单
```

- Conboot 模式下删除版本。

```
=====<EXTEND-ConBoot-MENU>=====
<1> Boot System
<2> Enter Serial SubMenu
<3> Enter Ethernet SubMenu
<4> File Control
<5> Modify ConBoot Password
<6> Skip Current System Configuration
<7> ConBoot Operation Menu
<8> Skip Current System Password
<0> reboot
=====
enter your choice (0 - 7):4----->主菜单模式下键入 4，进入文件控制菜单
=====<File CONTROL>=====
<1> Display All File(s)
<2> Set Application File type
<3> Delete File
<4> Format Partition
<0> Exit To Main Menu
=====
enter your choice (0 - 4):3----->键入 3，进入删除版本界面
Display all file(s) in nand0:
'M'= main 'B'=backup
=====
NO.          filename          size          type
-----
1: CN12800-G-S111C010D003.bin 45381504
2: CN12800-G-S111C011D007.bin 45386636 M
=====
enter your choice:2----->选择要删除的版本编号
=====<File CONTROL>=====
<1> Display All File(s)
<2> Set Application File type
<3> Delete File
<4> Format Partition
<0> Exit To Main Menu
=====
enter your choice (0 - 4):0----->删除成功后键入 0，
```

回退到主菜单

1.3.2 命令行下操作

升级版本之前保证存放版本的主机网卡和交换机的物理端口连接正确，主机和设备可以正常通信，并在主机上开启 TFTP 服务器。使用终端登录设备的串口，按照下面的提示升级版本。

- 命令行模式下升级版本。

```
[INSPUR]boot-file get CN12800-G-S111C011D007.bin tftp 10.24.9.99
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total      Spent    Left    Speed
100 37.6M  100 37.6M    0     0   306k      0  0:02:05  0:02:05  --:--:-- 312k
Download successfully!
[INSPUR]
```

- 命令行模式下设置主启动版本。

```
[INSPUR]boot-file main CN12800-G-S111C011D007.bin
[INSPUR]
```

- 命令行模式下设置备份版本，当主版本被删除时，设备重启后就使用备版本。

```
[INSPUR]boot-file backup CN12800-G-S111C011D007.bin
[INSPUR]
```

- 命令行模式下删除版本。

```
[INSPUR]boot-file delete CN12800-G-S111C011D007.bin
[INSPUR]
```

1.4 清除配置

当设备新版本和旧版本跨度较大时，会出现配置不兼容的情况，此时需要清除设备以前的配置，再换新的版本。

```
<INSPUR> configuration clear-all
```

2 基本二三层转发配置案例

2.1 二层转发简介

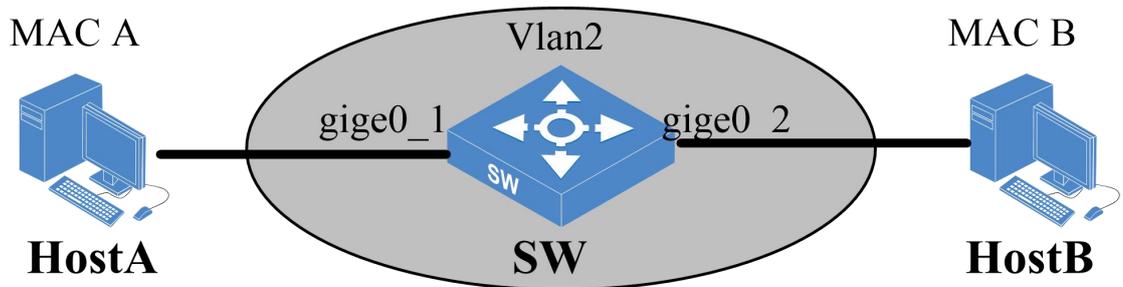
二层转发是建立在 MAC 地址基础上的数据转发，由内部芯片来完成数据转发，转发速度快，性能高。二层转发通过维护 MAC 地址表实现转发，收到报文时将其源 MAC 与端口的对应关系写到 MAC 表里，作为二层转发的依据，转发时根据报文的目的 MAC 查找 MAC 表进行转发。二层报文只能在相同 VLAN 里通过二层转发。

2.1.1 配置需求

公司的内部网络划分了多个不同的 VLAN，同一个 VLAN 内的用户可使用二层转发进行通信。

2.1.2 网络拓扑

图 2-1 二层转发组网图



2.1.3 配置流程

- (1) 在 SW 上创建 VLAN2;
- (2) 添加 gige0_1、gige0_2 到 VLAN2;
- (3) 验证配置。

2.1.4 配置步骤

(1) 在 SW 上创建 VLAN2。

```
<INSPUR>conf-mode
[INSPUR]vlan 2
```

(2) 添加 gige0_1、gige0_2 到 VLAN2。

```
<INSPUR>conf-mode
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_1
[INSPUR-vlan2]port gige0_2
```

(3) 验证配置。

进入用户视图使用命令 **show mac-address-table all** 能查看到 MAC 地址表,其中包含了 MAC A、MAC B 与端口的对应关系, HostA 和 HostB 配置好同一网段的地址后能正常通信。

2.2 三层转发简介

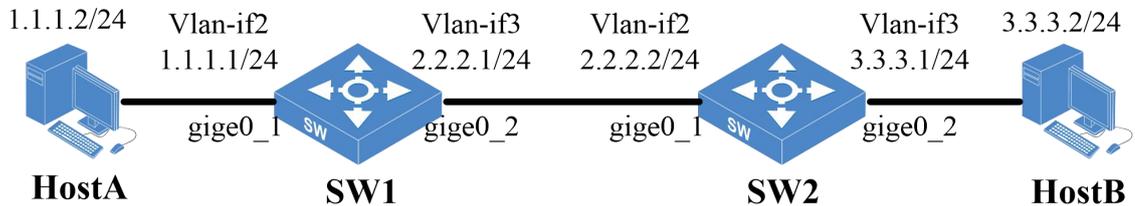
三层转发主要由芯片和 CPU 两大部分完成,芯片主要负责转发功能,内部有二层 MAC 表项和三层转发表项,CPU 主要用于转发控制,维护表项并将三层转发表项下发到芯片。当交换机上未建立任何转发表项的时候,位于不同网段的 PC 不能直接通过芯片转发进行通信,需要 CPU 的参与在三层交换机芯片上建立三层转发表项。表项建立完成后,交换机收到三层报文就会按照查询 MAC 表-->查询三层转发表项的流程将报文转发到目的主机。

2.2.1 配置需求

公司内部划分了多个不同的 VLAN, VLAN 之间二层隔离,不同 VLAN 的用户想要通信,只能通过三层转发来实现。

2.2.2 网络拓扑

图 2-2 三层转发组网图



三层转发需要保证 1.1.1.0 网段和 3.3.3.0 网段的路由可达，可以在 SW1 和 SW2 上配置静态路由或使用 RIP 和 OSPF 等路由协议。

2.2.3 配置流程

- (1) 在 SW1 上创建 VLAN2、VLAN3，添加接口到相应的 VLAN；
- (2) 在 SW1 上配置 vlan-if 口的 IP 地址及静态路由；
- (3) 在 SW2 上创建 VLAN2、VLAN3，添加接口到相应的 VLAN；
- (4) 在 SW2 上配置 vlan-if 口的 IP 地址及静态路由；
- (5) 验证配置。

2.2.4 配置步骤

- (1) 在 SW1 上创建 VLAN2、VLAN3。添加接口到相应的 VLAN，配置 vlan-if 口的 IP 地址。

```
<INSPUR>conf-mode
[INSPUR]vlan 2 to 3
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_1
[INSPUR-vlan2]exit
[INSPUR]vlan 3
[INSPUR-vlan3]port gige0_2
```

- (2) 在 SW1 上配置 vlan-if 口的 IP 地址及静态路由。

```
[INSPUR]interface vlan-if2
[INSPUR-vlan-if2]ip address 1.1.1.1/24
```

```
[INSPUR-vlan-if2]exit
[INSPUR]interface vlan-if3
[INSPUR-vlan-if3]ip address 2.2.2.1/24
[INSPUR-vlan-if3]exit
[INSPUR]ip route 3.3.3.0 255.255.255.0 2.2.2.2
```

(3) 在 SW2 上创建 VLAN2、VLAN3，添加接口到相应的 VLAN。

```
<INSPUR>conf-mode
[INSPUR]vlan 2 to 3
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_1
[INSPUR-vlan2]exit
[INSPUR]vlan 3
[INSPUR-vlan3]port gige0_2
```

(4) 在 SW2 上配置 vlan-if 口的 IP 地址及静态路由。

```
[INSPUR]interface vlan-if2
[INSPUR-vlan-if2]ip address 2.2.2.2/24
[INSPUR-vlan-if2]exit
[INSPUR]interface vlan-if3
[INSPUR-vlan-if3]ip address 3.3.3.1/24
[INSPUR-vlan-if3]exit
[INSPUR]ip route 1.1.1.0 255.255.255.0 2.2.2.1
```

(5) 验证配置。

HostA 和 HostB 配置相应网段的 IP 地址后可以正常通信。

3 端口聚合典型配置案例

3.1 端口聚合简介

端口聚合又称链路聚合，它通过捆绑多条物理链路成为一条逻辑链路，不仅增加了链路的带宽，还使捆绑在一起的链路相互形成动态备份，有效提高了链路的可靠性。当交换机检测到端口聚合中一个成员端口的链路发生故障时，就会停止在此端口上发送报文，并根据负载分担策略将该故障链路上原来的负载分到剩下的链路上，故障链路恢复后又重新开始发送报文。

链路聚合的功能：

- 增加了带宽-将多个链路的容量组合到一个逻辑链路中。
- 自动故障转移/故障恢复-将来自故障链路的通信转移到聚合中的工作链路。
- 负载均衡-传入和外发通信都是根据用户选择的负载均衡策略（如源和目标 MAC 或 IP 地址）进行分配的。
- 改进管理-所有接口作为一个单元进行管理。
- 减少网络地址池消耗-可以将一个 IP 地址指定给整个聚合。

3.1.1 基本概念

(1) 聚合组和成员端口

表 3-1 聚合组和成员端口

项目	说明
聚合组	多个以太网接口捆绑在一起所形成的组合
成员端口	被捆绑在一起的以太网接口就称为该聚合组的成员端口

成员端口的状态

聚合组内的成员端口具有以下几种状态：

表 3-2 成员端口的状态

项目	说明
选中 (Selected) 状态	此状态下的成员端口可以参与用户数据的转发。
非选中 (Unselected) 状态	此状态下的成员端口不能参与用户数据的转发。
管理 KEY	在进行链路聚合时, 系统会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。 自动计算生成一个数值, 该数值就称为管理 KEY。这个信息组合中任何一项的变化都会引起管理 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管理 KEY。

3.1.2 聚合模式

端口聚合有两种模式: 静态聚合和动态聚合

表 3-3 聚合模式

项目	说明
静态聚合	静态聚合模式下, 聚合端口的建立, 成员端口的加入完全由手工来配置, 没有端口聚合控制协议的参与。
动态聚合	动态聚合模式下, 聚合端口的建立, 成员端口的加入, 都是由管理员配置完成的。但与静态聚合不同的是, 动态聚合模式下 LACP 协议报文参与活动接口的选择。当把一组端口加入到聚合组后, 这些端口是否处于选中状态参与数据转发还需要经过 LACP 协议报文的协商确定。只有速率和双工属性相同、连接到同一个设备、有相同基本配置的端口才能被动态汇聚在一起。一个端口也可以创建动态聚合, 称为单端口聚合。

3.1.3 负载分担类型

目前, 交换机的端口聚合负载分担支持以下几种类型:

- 根据源 IP 地址进行负载分担;
- 根据目的 IP 地址进行负载分担;
- 根据源 IP 地址和目的 IP 地址进行负载分担;

- 根据源 MAC 地址进行负载分担；
- 根据目的 MAC 地址进行负载分担；
- 根据源 MAC 地址和目的 MAC 地址进行负载分担；
- 根据端口（enhanced）进行负载分担。
- 根据用户自定义方式进行负载分担。

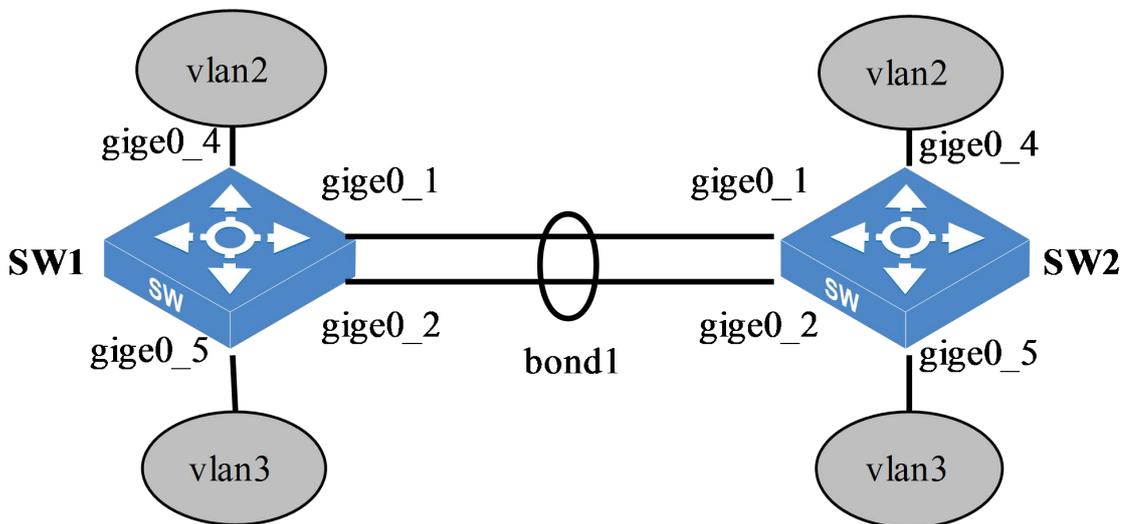
3.2 动态端口聚合配置案例

3.2.1 配置需求

端口聚合后生成的逻辑链路带宽等于物理链路的带宽总和，且多条链路相互备份，有效地提高了链路的可靠性。可用于企业的某些重要链路上，使网络的运行更具保障性。

3.2.2 网络拓扑

图 3-2 端口聚合组网图



3.2.3 配置流程

- (1) 分别在 SW1 和 SW2 上创建 bond1，配置聚合组类型为动态聚合、出端口算法为源 IP 地址和目的 IP 地址，添加聚合成员端口 gige0_1、gige0_2。

- (2) 分别在 SW1 和 SW2 创建 VLAN2-3，配置 bond1 口为 Trunk，允许 VLAN2-3 通过。
- (3) 验证配置。

3.2.4 配置步骤

- (1) 分别在 SW1 和 SW2 上创建 bond1，配置聚合组类型为动态聚合、出端口算法为源 IP 地址+目的 IP 地址，添加聚合成员端口 gige0_1、gige0_2。

```
[INSPUR]interface bond 1
[INSPUR-bond1]bond mode dynamic
[INSPUR-bond1]bond load-sharing mode source-destination-ip
[INSPUR-bond1]exit
[INSPUR]interface gige 0_1
[INSPUR-gige0_1]bond group 1
[INSPUR]interface gige 0_2
[INSPUR-gige0_1]bond group 1
[INSPUR-gige0_1]exit
```

- (2) 分别在 SW1 和 SW2 创建 VLAN2-3，配置 bond1 口为 Trunk，允许 VLAN2-3 通过。

```
[INSPUR]vlan 2 to 3
[INSPUR]interface bond1
[INSPUR-bond5]switchport mode trunk
[INSPUR-bond5]switchport trunk allowed vlan 2-3
[INSPUR-bond5]switchport trunk native vlan 3
```

- (3) 验证配置。

查看 SW1 端口聚合状态

```
<INSPUR>show bond 1 summary

          bond listing:
          -----
bond: 1
-----
Bond state      : L2
MII Status     : up
Bond mode      : dynamic
Load sharing    : source-destination-ip
Bond description:
Minimum Links   :
```

```
Maxports      : 8
Protocol      : LACP
Select mode   : speed
System-priority : 32768
System-id     : 00:24:AC:71:AD:F1
Par system-id : 00:24:AC:B5:47:12
Minimum port  : gige0_1
Select port   : gige0_1,gige0_2
Unselect port :
```

查看 SW2 端口聚合状态

```
<INSPUR>show bond 1 summary

                bond listing:
                -----
bond: 1
-----
Bond state      : L2
MII Status      : up
Bond mode       : dynamic
Load sharing    : source-destination-ip
Bond description:
Minimum Links   :
Maxports        : 8
Protocol        : LACP
Select mode     : speed
System-priority : 32768
System-id       : 00:24:AC:B5:47:12
Par system-id   : 00:24:AC:71:AD:F1
Minimum port    : gige0_1
Select port     : gige0_1,gige0_2
Unselect port   :
```

聚合组状态信息表明了聚合组 1 为根据源 IP+目的 IP 进行负载分担的动态聚合组。VLAN2 和 VLAN3 的数据流量通过聚合链路时，能实现负载分担和链路备份，增加了链路的可靠性。

4 端口镜像典型配置案例

4.1 端口镜像简介

端口镜像的主要功能是将源端口的报文复制一份到目的端口，可用于网络监控和故障排查。在实际应用中，目的端口一般与服务器和 PC 等设备相连，用户可以在服务器上查看源端口入方向和出方向报文，以实现对接端口的监控。当网络出现故障时，利用笔记本与目的端口相连，抓包分析源端口进出的报文，能帮助定位解决问题。

4.1.1 端口镜像基本概念

端口镜像组的源端口和目的端口

表 4-1 端口镜像组的源端口和目的端口

项目	说明
源端口	被监控的端口，用户可以将通过该端口的报文复制到目的端进行监控和分析
目的端口	监控端口，接收源端口复制过来的报文，并转发到服务器上，便于对报文进行监控和分析
反射端口	远程镜像源镜像组中的特殊端口，该端口独立使用一个 VLAN，且端口不需要连接网线

4.1.1.1 镜像方向

端口镜像的方向分为三种：

表 4-2 端口镜像的方向

项目	说明
入方向	仅对从源端口收到的报文进行镜像
出方向	仅对从源端口转发出去的报文进行镜像

项目	说明
双向	对从源端口收到和发出的报文都进行镜像

4.1.2 端口镜像分类

端口镜像分为两类：本地端口镜像和远程端口镜像

表 4-3 端口镜像

项目	说明
本地端口镜像	源端口和目的端口均在同一台设备上，将源端口的报文复制一份到目的端口
远程端口镜像	源端口和目的端口分别在不同的设备上，两台设备之间通过二层网络连接，被镜像的报文通过二层网络转发到目的端口



说明

一个端口只能加入一个镜像组，源端口不能配置为本镜像组或其它镜像组的目的端口。

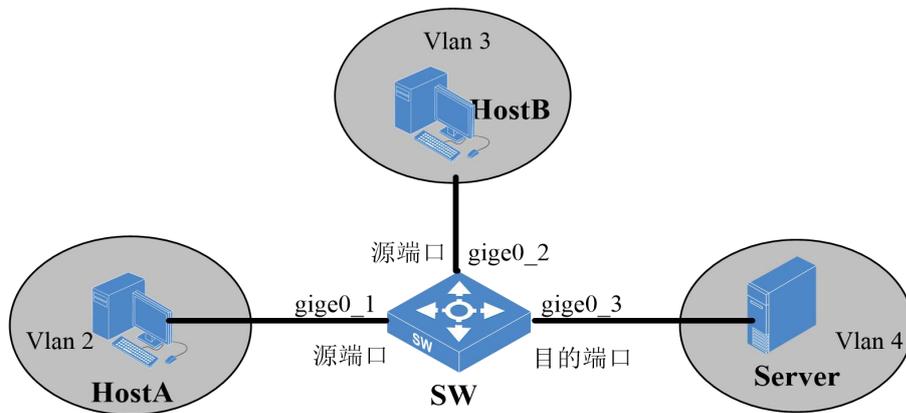
4.2 本地端口镜像配置案例

4.2.1 配置需求

本地端口镜像主要用于监控和分析进出本台设备端口的报文，当用户需要监控某个端口的报文时，将该端口配置为源端口，在目的端口上连接服务器就能进行实时监控。当网络出现故障，需要排查设备时，将可疑端口配置为源端口，在目的端口上连接笔记本抓包分析。

4.2.2 网络拓扑

图 4-2 本地端口镜像组网图



4.2.3 配置流程

- (1) 在 SW 上创建 VLAN2、VLAN3 和 VLAN4。
- (2) 添加 gige0_1 到 VLAN2，gige0_2 到 VLAN3，gige0_3 到 VLAN4。
- (3) 在 SW 上创建本地镜像组 1，源端口为 gige0_1 和 gige0_2，目的端口为 gige0_3，方向为双向。
- (4) 验证配置。

4.2.4 配置步骤

- (1) 在 SW 上创建 VLAN2、VLAN3 和 VLAN4。

```
<INSPUR>conf-mode
[INSPUR]vlan 2 to 4
[INSPUR]
```

- (2) 添加 gige0_1 到 VLAN2，gige0_2 到 VLAN3，gige0_3 到 VLAN4。

```
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_1
[INSPUR-vlan2]exit
[INSPUR]vlan 3
```

```
[INSPUR-vlan3]port gige0_2
[INSPUR-vlan3]exit
[INSPUR]vlan 4
[INSPUR-vlan4]port gige0_3
[INSPUR-vlan4]exit
[INSPUR]
```

(3) 在 SW 上创建本地镜像组 1，源端口为 gige0_1 和 gige0_2，目的端口为 gige0_3，方向为双向。

```
[INSPUR] mirror 1 source interface gige0_1 gige0_2 both
[INSPUR] mirror 1 destination interface gige0_3
```

(4) 验证配置

查看镜像组 1

```
<INSPUR>show mirror local
-----Local mirror groups information-----
Group-id Mirroring-ports          Direction Monitor-ports  Description
0      1      gige0_1,gige0_2              both      gige0_3
```

以上信息表明该镜像组为本地镜像，镜像组 ID 是 1，源端口为 gige0_1,gige0_2，目的端口为 gige0_3，镜像方向是双向。当 gige0_1 和 gige0_2 上有报文进出时，服务器上可以监控到所有报文。

4.3 反射口实现的远程端口镜像配置案例

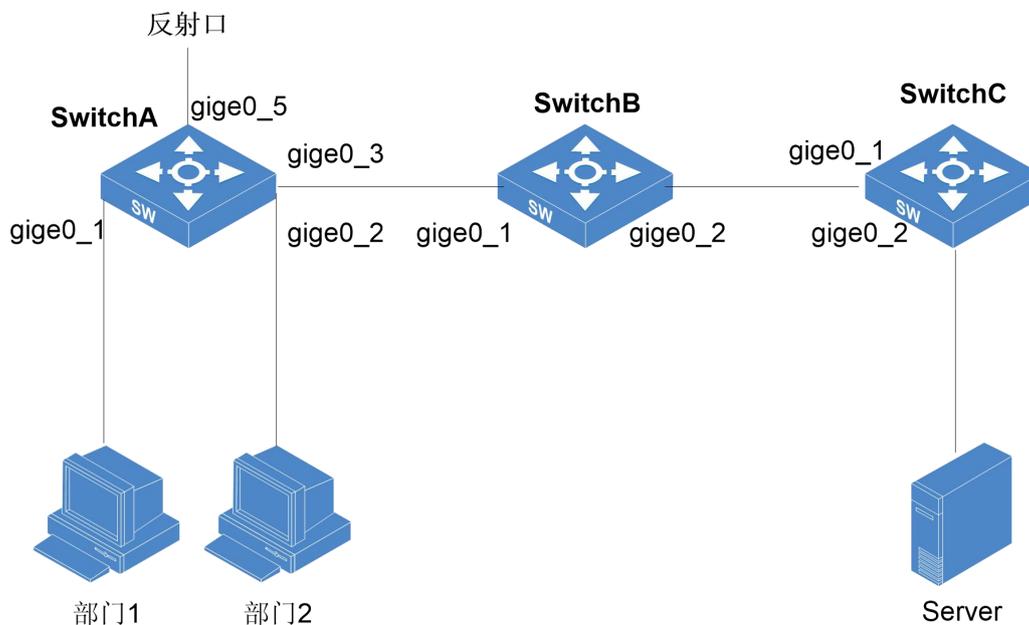
4.3.1 配置需求

某公司内部通过交换机实现各部门之间互连，网络环境描述如下：

- 部门 1 通过端口 gige0_1 接入 SwitchA;
- 部门 2 通过端口 gige0_2 接入 SwitchA;
- SwitchA 的端口 gige0_3 和 SwitchB 的端口 gige0_1 相连;
- SwitchB 的端口 gige0_2 和 SwitchC 的端口 gige0_1 相连;
- 监控设备 server 连接在 SwitchC 的端口 gige0_2 上;

网络管理员系统通过监控设备 server 对部门 1 和部门 2 收发的报文进行监控，使用远程端口镜像实现该需求。

图 4-3 反射口实现的远程端口镜像组网图



4.3.2 配置思路

- (1) SwitchA 为源设备，SwitchB 为中间设备，SwitchC 为目的设备。
- (2) 在 SW1 上创建 vlan2 vlan3，添加 gige0_1 端口到 vlan2，添加 gige0_2 端口到 vlan3，配置 gige0_3 允许 vlan10 通过。
- (3) 在 SwitchA 设备上配置 vlan 10 为远程镜像 vlan，端口 gige0_1 和端口 gige0_2 为镜像源端口，端口 gige0_5 为反射口。
- (4) 配置 SwitchA 的端口 gige0_3、SwitchB 的端口 gige0_1 和 gige0_2、SwitchC 的端口 gige0_1 的端口类型为 trunk 口，并且都允许 vlan10 的报文通过。
- (5) 在 SwitchC 上配置 vlan10 为远程镜像 VLAN，连接数据监测设备的端口 gige0_2 为镜像目的端口。

4.3.3 配置步骤

- (1) 在 SwitchA 上创建 vlan2、vlan3。

```
<SwitchA>conf-mode
[SwitchA]vlan 2 to 3
```

- (2) 在 SwitchA 上添加 gige0_1 端口到 vlan2，添加 gige0_2 端口到 vlan3，配置 gige0_3 允许 VLAN10 通过。

```
[SwitchA]vlan 2
[SwitchA-vlan2]port gige0_1
[SwitchA-vlan2]exit
[SwitchA]vlan 3
[SwitchA-vlan3]port gige0_2
[SwitchA-vlan3]exit
[SwitchA]interface gige0_3
[SwitchA-gige0_3]switchport mode trunk
[SwitchA-gige0_3]switchport trunk allowed vlan 10
```

- (3) 在 SwitchA 上配置远程镜像 VLAN、源端口和出端口。

```
<SwitchA>conf-mode
[SwitchA]mirror 1000 source interface gige0_1 gige0_2 both
[SwitchA]mirror 1000 destination remote-vlan 10 reflector-port gige0_5
```

- (4) 在 SwitchB 上配置端口 trunk 口，允许 vlan10 的报文通过。

```
<SwitchB>conf-mode
[SwitchB]interface gige 0_1
[SwitchB-gige0_1]switchport mode trunk
[SwitchB-gige0_1]switchport trunk allowed vlan 10
[SwitchB-gige0_1]exit
[SwitchB]interface gige 0_2
[SwitchB-gige0_2]switchport mode trunk
[SwitchB-gige0_2]switchport trunk allowed vlan 10
```

- (5) 在 SwitchC 上配置端口 gige0_1 的端口类型为 trunk 口，允许 vlan10 报文通过。

```
<SwitchC>conf-mode
[SwitchC]interface gige 0_1
[SwitchC-gige0_1]switchport mode trunk
[SwitchC-gige0_1]switchport trunk allowed vlan 10
```

- (6) 在 SwitchC 上配置目的设备的远程镜像 vlan 和目的端口。

```
<SwitchC>conf-mode
[SwitchC]vlan 10
[SwitchC-vlan10]port gige0_2
[SwitchC-vlan10]exit
[SwitchC]mirror 2000 source remote-vlan 10
```

4.3.4 验证配置

(1) 在 **SwitchA** 上查看远程源镜像组。

```
[SwitchA]show mirror 1000
Mirror ID: 1000
-----
Mirror Type: remote mirror
Mirror Direction: Both
Source Ports: gige0_1,gige0_2
Reflector Port: gige0_5
Remote Vlan: 10
Mirror Description: None
```

(2) 在 **SwitchC** 上查看远程目的镜像组。

```
[SwitchC]show mirror 2000
Mirror ID: 2000
-----
Remote Vlan: 10
Mirror Description: None
```

(3) 在 **Server** 上使用抓包工具进行抓包，能抓到镜像源端口的出入报文。

4.3.5 配置文件

(1) **SwitchA** 的配置文件

```
mirror 1000 source interface gige0_1,gige0_2 both
mirror 1000 destination remote-vlan 10 reflector-port gige0_5
vlan 1 to 3
!
interface gige0_1
switchport access vlan 2
!
interface gige0_2
```

```
switchport access vlan 3
!
interface gige0_3
switchport mode trunk
switchport trunk allowed vlan 10
!
interface gige0_5
!
```

(2) SwitchB 的配置文件

```
interface gige0_1
switchport mode trunk
switchport trunk allowed vlan 10
! !
interface gige0_2
switchport mode trunk
switchport trunk allowed vlan 10
```

(3) SwitchC 的配置文件

```
mirror 2000 source remote-vlan 10
vlan 10
!
interface gige0_1
switchport mode trunk
switchport trunk allowed vlan 10
!
interface gige0_2
switchport access vlan 10
```

4.4 出端口实现的远程端口镜像

4.4.1 配置需求

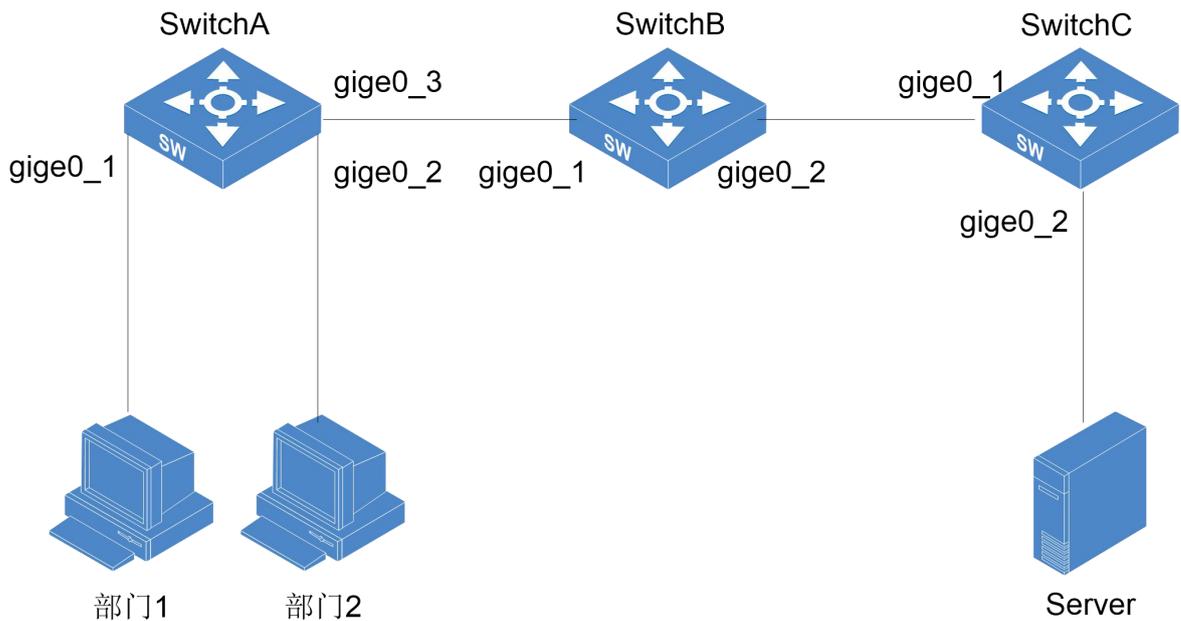
某公司内部通过交换机实现各部门之间互连，网络环境描述如下：

- 部门 1 通过端口 gige0_1 接入 SwitchA;
- 部门 2 通过端口 gige0_2 接入 SwitchA;
- SwitchA 的端口 gige0_3 和 SwitchB 的端口 gige0_1 相连;

- SwitchB 的端口 gige0_2 和 SwitchC 的端口 gige0_1 相连；
- 监控设备 server 连接在 SwitchC 的端口 gige0_2 上；

网络管理员系统通过监控设备 server 对部门 1 和部门 2 收发的报文进行监控，使用远程端口镜像实现该需求。

图 4-4 出端口实现的远程端口镜像组网图



4.4.2 配置思路

- (1) SwitchA 为源设备，SwitchB 为中间设备，SwitchC 为目的设备；
- (2) 在 SW1 上创建 vlan2 vlan3，添加 gige0_1 端口到 vlan2，添加 gige0_2 端口到 vlan3，配置 gige0_3 允许 vlan10 通过；
- (3) 在 SwitchA 设备上配置 vlan 10 为远程镜像 vlan，端口 gige0_1 和端口 gige0_2 为镜像源端口，端口 gige0_3 为出端口；
- (4) 配置 SwitchA 的端口 gige0_3、SwitchB 的端口 gige0_1 和 gige0_2、SwitchC 的端口 gige0_1 的端口类型为 trunk 口，并且都允许 vlan10 的报文通过；
- (5) 在 SwitchC 上配置 vlan10 为远程镜像 VLAN，连接数据监测设备的端口 gige0_2 为镜像目的端口。

4.4.3 配置步骤

- (1) 在 SwitchA 上创建 vlan2、vlan3;

```
<SwitchA>conf-mode
[SwitchA]vlan 2 to 3
```

- (2) 在 SwitchA 上添加 gige0_1 端口到 vlan2, 添加 gige0_2 端口到 vlan3, 配置 gige0_3 允许 VLAN10 通过;

```
[SwitchA]vlan 2
[SwitchA-vlan2]port gige0_1
[SwitchA-vlan2]exit
[SwitchA]vlan 3
[SwitchA-vlan3]port gige0_2
[SwitchA-vlan3]exit
[SwitchA]interface gige0_3
[SwitchA-gige0_3]switchport mode trunk
[SwitchA-gige0_3]switchport trunk allowed vlan 10
```

- (3) 在 SwitchA 上配置远程镜像 VLAN、源端口和出端口;

```
<SwitchA>conf-mode
[SwitchA]mirror 1000 source interface gige0_1 gige0_2 both
[SwitchA]mirror 1000 destination remote-vlan 10 out-port gige0_3
```

- (4) 在 SwitchB 上配置端口 trunk 口, 允许 vlan10 的报文通过;

```
<SwitchB>conf-mode
[SwitchB]interface gige 0_1
[SwitchB-gige0_1]switchport mode trunk
[SwitchB-gige0_1]switchport trunk allowed vlan 10
[SwitchB-gige0_1]exit
[SwitchB]interface gige 0_2
[SwitchB-gige0_2]switchport mode trunk
[SwitchB-gige0_2]switchport trunk allowed vlan 10
```

- (5) 在 SwitchC 上配置端口 gige0_1 的端口类型为 trunk 口, 允许 vlan10 报文通过;

```
<SwitchC>conf-mode
[SwitchC]interface gige 0_1
[SwitchC-gige0_1]switchport mode trunk
[SwitchC-gige0_1]switchport trunk allowed vlan 10
```

- (6) 在 SwitchC 上配置目的设备的远程镜像 vlan 和目的端口;

```
<SwitchC>conf-mode
[SwitchC]vlan 10
[SwitchC-vlan10]port gige0_2
[SwitchC-vlan10]exit
[SwitchC]mirror 2000 source remote-vlan 10
```

4.4.4 验证配置

(1) 在 **SwitchA** 上查看远程源镜像组

```
[SwitchA]show mirror 1000
Mirror ID: 1000
-----
Mirror Type: remote mirror
Mirror Direction: Both
Source Ports: gige0_1,gige0_2
Out Port: gige0_3
Remote Vlan: 10
Mirror Description: None
```

(2) 在 **SwitchC** 上查看远程目的镜像组

```
[SwitchC]show mirror 2000
Mirror ID: 2000
-----
Remote Vlan: 10
Mirror Description: None
```

(3) 在 **Server** 上使用抓包工具进行抓包，能抓到镜像源端口的出入报文。

4.4.5 配置文件

(1) **SwitchA** 的配置文件

```
mirror 1000 source interface gige0_1,gige0_2 both
mirror 1000 destination remote-vlan 10 out-port gige0_3
vlan 1 to 3
!
interface gige0_1
switchport access vlan 2
!
interface gige0_2
```

```
switchport access vlan 3
!  
interface gige0_3  
switchport mode trunk  
switchport trunk allowed vlan 10
```

(2) SwitchB 的配置文件

```
interface gige0_1  
switchport mode trunk  
switchport trunk allowed vlan 10  
!  
interface gige0_2  
switchport mode trunk  
switchport trunk allowed vlan 10
```

(3) SwitchC 的配置文件

```
mirror 2000 source remote-vlan 10  
vlan 10  
!  
interface gige0_1  
switchport mode trunk  
switchport trunk allowed vlan 10  
!  
interface gige0_2  
switchport access vlan 10
```

5 端口限速典型配置案例

5.1 端口限速简介

端口限速是指限制一个端口上报文转发的总速率，分为两个方向：入方向和出方向。

表 5-1 端口限速

项目	说明
入方向端口限速	在报文进入的端口进行限速
出方向端口限速	在报文转发出去的端口进行限速。

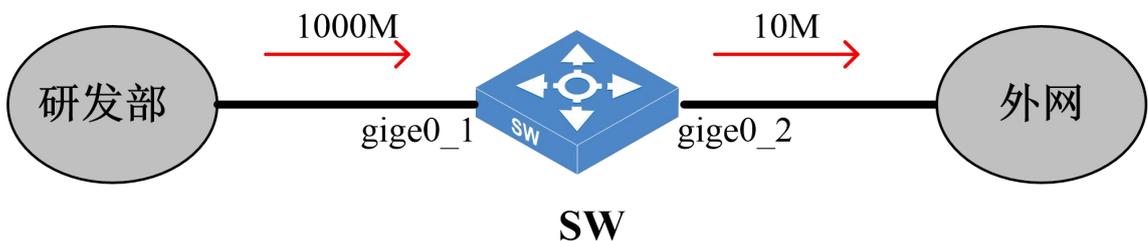
5.2 配置案例

5.2.1 配置需求

当企业想限制某个部门访问外网、服务器等资源的速率时，可以通过端口限速功能实现。

5.2.2 网络拓扑

图 5-2 端口限速组网图



5.2.3 配置流程

- (1) 在 SW 上创建 VLAN2，添加 gige0_1 和 gige0_2 到 VLAN2。

(2) 在 SW 上配置入方向端口限速，入端口为 gige0_1，限制速率为 10Mbits/s，设置突发流量为 1024Kbits。

(3) 验证配置。

5.2.4 配置步骤

(1) 在 SW 上创建 VLAN2，添加 gige0_1 和 gige0_2 到 VLAN2。

```
<INSPUR>conf-mode
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_1
[INSPUR-vlan2]port gige0_2
[INSPUR-vlan2]exit
[INSPUR]
```

(2) 在 SW 上配置入方向端口限速，入端口为 gige0_1，限制速率为 10Mbits/s，设置突发流量为 1024Kbits。

```
[INSPUR]interface gige0_1
[INSPUR-gige0_1]rate-limit input 10000 burst-bucket 1024
[INSPUR-gige0_1]exit
[INSPUR]
```

(3) 验证配置。

配置了端口限速后，研发部访问外网的总带宽为 10Mbits/s。

6 端口隔离典型配置案例

6.1 端口隔离简介

端口隔离是一种二层隔离功能，能实现相同 VLAN 的两个接口进行二层隔离，只需要将相同 VLAN 的两个接口配置为端口隔离即可。需要注意的是，隔离端口与隔离端口之间，单播、组播、广播均不能转发，若需要隔离端口与隔离端口之间互相通讯，则必须通过三层设备进行转发。与 VLAN 之间的隔离相比，端口隔离避免了有限 VLAN 资源的浪费。是一种较为实用的二层隔离技术。

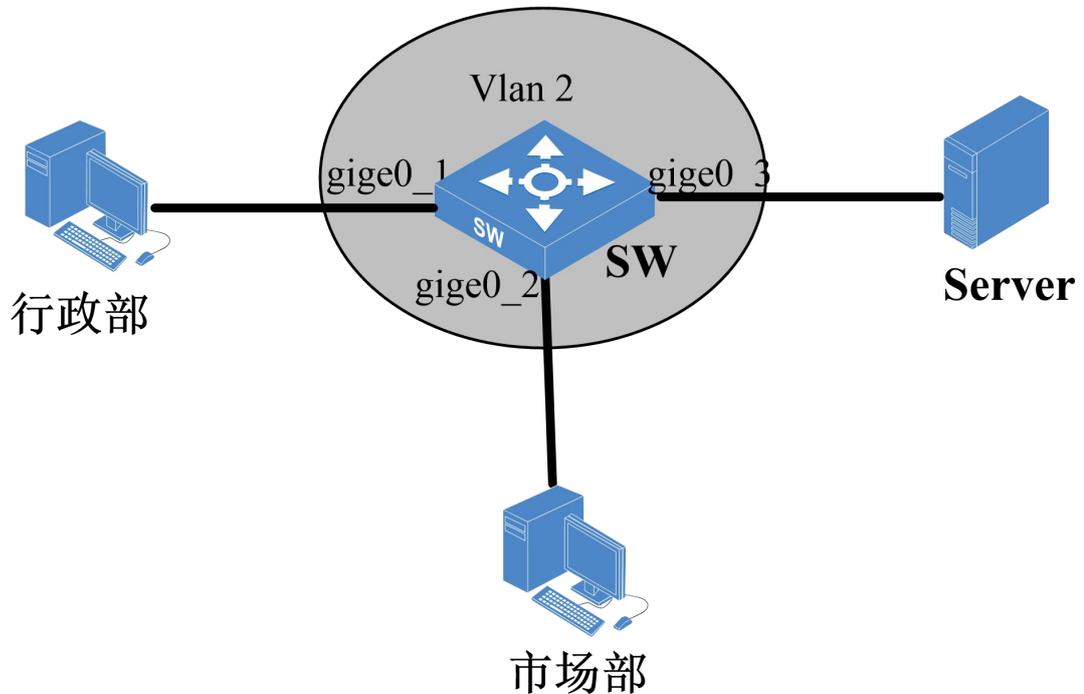
6.2 配置案例

6.2.1 配置需求

端口隔离实现了同一个 VLAN 内的二层报文隔离，当用户需要限制同一个 VLAN 内不同部门之间不能相互访问的同时又能保证各个部门可以访问服务器、外网等资源时，用户只需将这些部门添加到隔离组内即可。

6.2.2 网络拓扑

图 6-1 端口隔离组网图



6.2.3 配置流程

- (1) 在 SW 创建 VLAN2，添加 gige0_1、gige0_2 和 gige0_3 到 VLAN2
- (2) 在 SW 上添加 gige0_1、gige0_2 为隔离组成员端口。
- (3) 验证配置。

6.2.4 配置步骤

- (1) 在 SW 创建 VLAN2，添加 gige0_1、gige0_2 和 gige0_3 到 VLAN2。

```
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_1
[INSPUR-vlan2]port gige0_2
[INSPUR-vlan2]port gige0_3
[INSPUR-vlan2]exit
[INSPUR]
```

(2) 在 SW 上添加 gige0_1、gige0_2 到隔离组。

```
[INSPUR]interface gige 0_1
[INSPUR-gige0_1]switchport protected
[INSPUR-gige0_1]interface gige 0_2
[INSPUR-gige0_2]switchport protected
[INSPUR-gige0_2]
```

(3) 验证配置。

启用端口隔离功能后，行政部和市场部都能访问 Server，但行政部和市场部不能相互访问。

7 PVLAN 典型配置案例

7.1 PVLAN 简介

PVLAN (Private VLAN)，即私有 VLAN。采用两层 VLAN 隔离技术，上层 VLAN 全局可见，下层 VLAN 相互隔离。PVLAN 通常用于企业内部网，用来防止连接到某些接口或接口组的网络设备之间的相互通信，但却允许与默认网关进行通信。尽管各设备处于不同的 PVLAN 中，它们可以使用相同的 IP 子网。

PVLAN 的 VLAN 类型

表 7-1 VLAN 类型

项目	说明
主 VLAN	可以和所有与之关联的团体 VLAN，隔离 VLAN 通信
团体 VLAN	相同团体 VLAN 内的端口可以互相通信，也可与主 VLAN 通信
隔离 VLAN	隔离 VLAN 内的端口不能相互通信，只可以与主 VLAN 内的端口通信，每个主 VLAN 中只能有一个隔离 VLAN

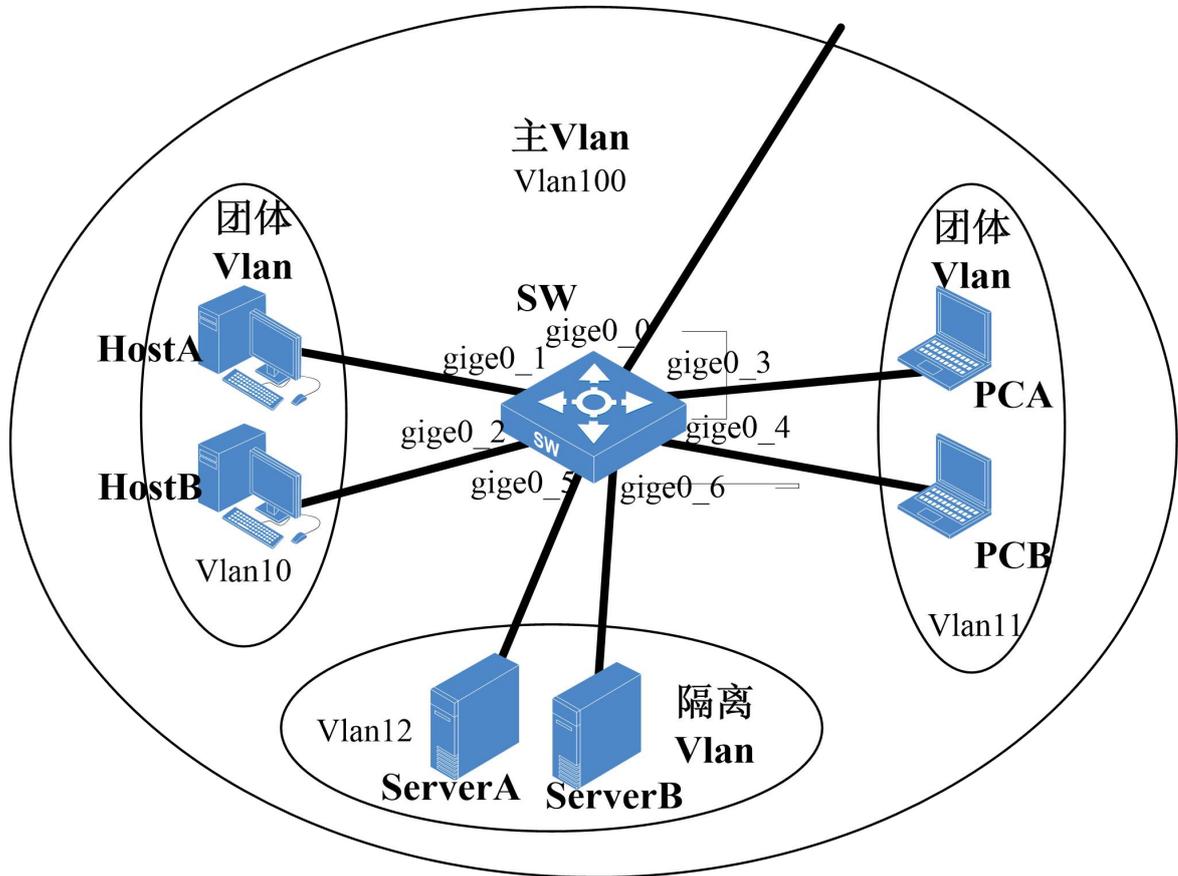
7.2 配置案例

7.2.1 配置需求

某公司内网中有移动用户接入，还有数台服务器。要使服务器之间相互隔离不能通信，并且移动用户也不能与公司内网员工通信，但所有用户和服务器可以正常连接外网。

7.2.2 网络拓扑

图 7-2 PVLAN 组网图



可使用 PVLAN 功能把移动用户和员工用户分别划入团体 VLAN10 和 VLAN11，服务器划入隔离 VLAN12，都可以通过主 VLAN100 连接外网。

7.2.3 配置流程

- (1) 配置 PVLAN，VLAN100 为主 VLAN，VLAN10 和 VLAN11 为团体 VLAN，VLAN12 为隔离 VLAN。
- (2) 把端口划入相应 VLAN 内。
- (3) 验证配置。

7.2.4 配置步骤

- (1) 配置 PVLAN，vlan100 为主 vlan，vlan10 和 vlan11 为团体 vlan，vlan12 为隔离 VLAN。

```
[INSPUR]pvlan primary-vlan 100 isolate-vlan 12 community-vlan-range 10-11
```

- (2) 配置接口加入 VLAN 并使能 PVLAN。

```
[INSPUR]interface gige 0_0
[INSPUR-gige0_0]pvlan promisc-association primary-vlan 100
[INSPUR-gige0_0]exit
[INSPUR]interface gige 0_1
[INSPUR-gige0_1]pvlan host-association secondary-vlan 10
[INSPUR-gige0_1]exit
[INSPUR]interface gige 0_2
[INSPUR-gige0_2]pvlan host-association secondary-vlan 10
[INSPUR-gige0_2]exit
[INSPUR]interface gige 0_3
[INSPUR-gige0_3]pvlan host-association secondary-vlan 11
[INSPUR-gige0_3]exit
[INSPUR]interface gige 0_4
[INSPUR-gige0_4]pvlan host-association secondary-vlan 11
[INSPUR-gige0_4]exit
[INSPUR]interface gige 0_5
[INSPUR-gige0_5]pvlan host-association secondary-vlan 12
[INSPUR-gige0_5]exit
[INSPUR]interface gige 0_6
[INSPUR-gige0_6]pvlan host-association secondary-vlan 12
[INSPUR-gige0_6]exit
```

- (3) 验证配置。

HostA 能与 HostB 通信，PCA 能与 PCB 通信，HostA 与 PCA 不能通信；ServerA 和 ServerB 不能通信，所有 Host、PC 和 Server 都能够连接外网。

8 QinQ 典型配置案例

8.1 QinQ 简介

QinQ (802.1Q-in-802.1Q) 技术是一项扩展 VLAN 空间的技术，通过在 802.1Q 标签报文的基础上再增加一层 802.1Q 的标签头来达到扩展 VLAN 空间的目的，可以使私网 VLAN 透传公网。由于在骨干网中传递的报文有两层 802.1Q Tag 头（一层公网 Tag，一层私网 Tag），即 802.1Q-in-802.1Q，所以称之为 QinQ 协议。

QinQ 是一种二层隧道协议，它是在 802.1Q 标签报文的基础上再增加一层 802.1Q 的标签头，即携带两层 VLAN Tag 穿越公网，从而为用户提供了一种比较简单的二层 VPN 隧道技术。QinQ 的实现方式可分为两种：基本 QinQ 和灵活 QinQ。

表 8-1 QinQ 简介

项目	说明
基本 QinQ	基于端口方式实现，配置了基本 QinQ 功能的端口会为收到的报文添加一层本端口缺省 VLAN 的 Tag。
灵活 QinQ	基于端口与 VLAN 相结合的方式实现。通过匹配流分类，可使同一端口为不同 VLAN 的流量添加不同的外层 VLAN Tag。

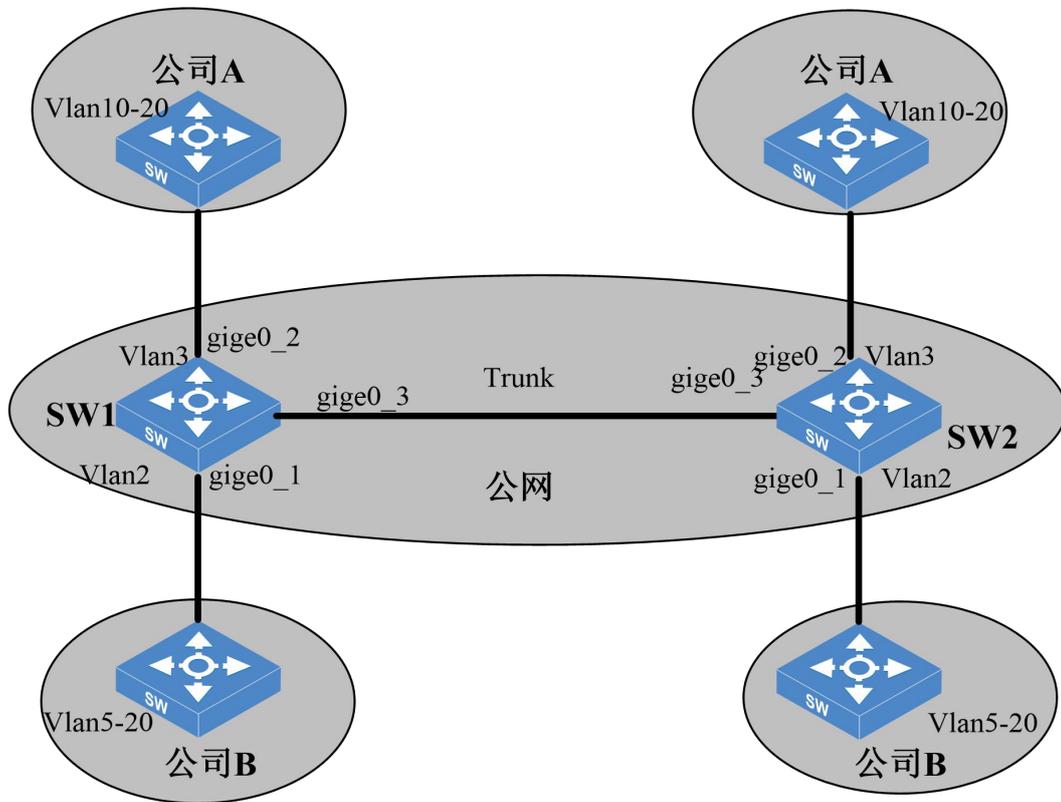
8.2 基本 QinQ 配置案例

8.2.1 配置需求

当运营商的网络需要承载公司 A 和公司 B 的流量，且两个公司在不同的区域有分支机构，运营商需要给公司 A 和公司 B 分配不同的 VLAN，使两个公司的流量隔开并且保证同一个公司的分支机构能互通，运营商可以在与用户连接的设备上使用基本 QinQ 功能来实现。

8.2.2 网络拓扑

图 8-2 基本 QinQ 组网图



8.2.3 配置流程

- (1) 分别在 SW1 和 SW2 上创建 VLAN2-4。
- (2) 分别在 SW1 和 SW2 上配置 gige0_1 为 trunk 允许 VLAN 2 通过，pvid 为 2；gige0_2 为 trunk 允许 VLAN 3 通过，pvid 为 3；配置 gige0_3 为 trunk，允许 VLAN2-4 通过，pvid 为 4。
- (3) 分别在 SW1 和 SW2 上的 gige0_1 口和 gige0_2 上开启基本 QinQ 功能。
- (4) 验证配置。

8.2.4 配置步骤

- (1) 分别在 SW1 和 SW2 上创建 VLAN。

```
[INSPUR]vlan 2 to 4
```

```
[INSPUR]
```

- (2) 分别在 SW1 和 SW2 上配置 gige0_1 为 Trunk，允许 VLAN 2 通过，Native VLAN id 为 2；配置 gige0_2 为 Trunk，允许 VLAN 3 通过，Native VLAN id 为 3；配置 gige0_3 为 Trunk，允许 VLAN2-4 通过，Native VLAN ID 为 4。

```
[INSPUR]interface gige0_1
[INSPUR-gige0_1]switchport mode trunk
[INSPUR-gige0_1]switchport trunk allowed vlan 2
[INSPUR-gige0_1]switchport trunk native vlan 2
[INSPUR-gige0_1]exit
[INSPUR]interface gige0_2
[INSPUR-gige0_2]switchport mode trunk
[INSPUR-gige0_2]switchport trunk allowed vlan 3
[INSPUR-gige0_2]switchport trunk native vlan 3
[INSPUR-gige0_2]exit
[INSPUR]interface gige0_3
[INSPUR-gige0_3]switchport mode trunk
[INSPUR-gige0_3]switchport trunk allowed vlan 2-4
[INSPUR-gige0_3]switchport trunk native vlan 4
[INSPUR-gige0_3]exit
[INSPUR]
```

- (3) 分别在 SW1 和 SW2 上的 gige0_1 口和 gige0_2 上开启基本 QinQ 功能。

```
[INSPUR]interface gige0_1
[INSPUR-gige0_1]qinq enable
[INSPUR]interface gige0_2
[INSPUR-gige0_2]qinq enable
```

- (4) 验证配置。

```
<INSPUR>show qinq
Qinq is enabled on following ports:
gige0_1
gige0_2
```

以上信息表明，gige0_1 和 gige0_2 口上开启了基本 QinQ 功能。公司 A 和公司 B 可以利用运营商分配不同的 VLAN 与异地部门进行正常通信。

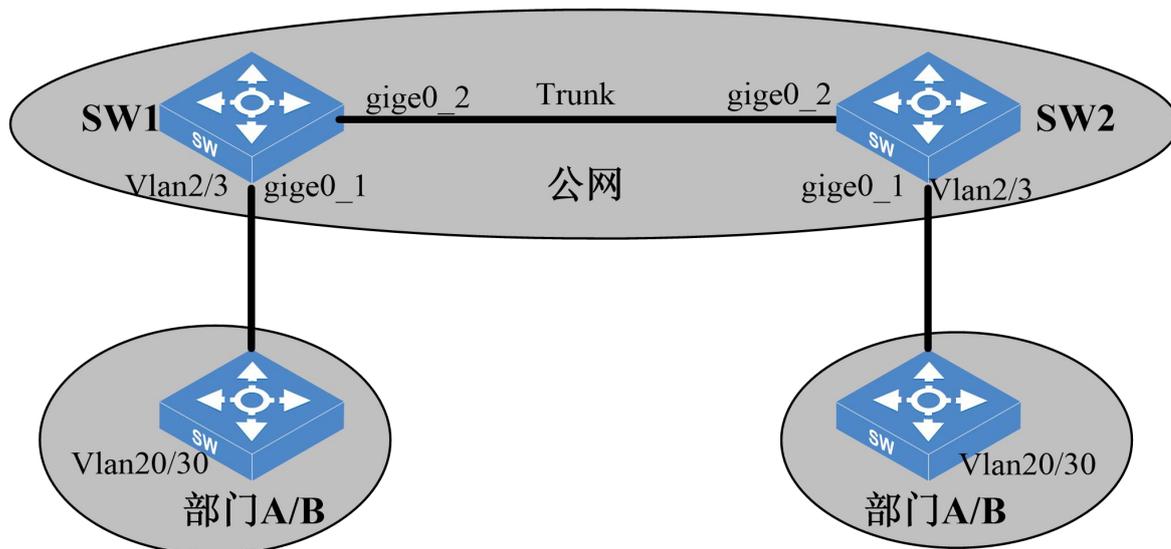
8.3 灵活 QinQ 配置案例

8.3.1 配置需求

当运营商的网络需要承载一个公司的流量，且该公司部门 A 与部门 B 在不同的 VLAN 内，部门 A、B 在较远的地方有分支机构，需要通过公网来互通时，运营商可以在连接用户的边缘设备上使能灵活 QinQ 功能，识别不同部门的报文所带的 Tag 后，为之封装上不同的外层 Tag。

8.3.2 网络拓扑

图 8-3 灵活 QinQ 组网图



8.3.3 配置流程

- (1) 分别在 SW1 和 SW2 上创建 VLAN2-4。
- (2) 分别在 SW1 和 SW2 上配置 gige0_1 为 hybrid，允许 VLAN 2 和 VLAN3 通过，配置 gige0_2 为 Trunk，允许 VLAN2-4 通过，native VLAN id 为 4。
- (3) 分别在 SW1 和 SW2 上的 gige0_1 口上开启灵活 QinQ 功能，当收到的报文 VLAN Tag 为 20 时，封装上外层 Tag VLAN2，当收到的报文 VLAN Tag 为 30 时，封装上外层 Tag VLAN3。
- (4) 验证配置。

8.3.4 配置步骤

- (1) 分别在 SW1 和 SW2 上创建 VLAN。

```
[INSPUR]vlan 2 to 4
```

- (2) 分别在 SW1 和 SW2 上配置 gige0_1 为 Hybrid，允许 VLAN 2 和 VLAN3 通过，配置 gige0_2 为 trunk，允许 VLAN2-4 通过，Native VLAN ID 为 4。

```
[INSPUR]interface gige0_1
[INSPUR-gige0_1]switchport mode hybrid
[INSPUR-gige0_1]switchport hybrid allowed vlan 2-3 untagged
[INSPUR-gige0_1]switchport hybrid native vlan 2
[INSPUR]interface gige0_2
[INSPUR-gige0_2]switchport mode trunk
[INSPUR-gige0_2]switchport trunk allowed vlan 2-4
[INSPUR-gige0_2]switchport trunk native vlan 4
[INSPUR-gige0_2]exit
```

- (3) 分别在 SW1 和 SW2 上的 gige0_1 口上开启灵活 QinQ 功能，当收到的报文 VLAN Tag 为 20 时，封装上外层 Tag VLAN2，当收到的报文 VLAN Tag 为 30 时，封装上外层 Tag VLAN3。

```
[INSPUR]interface gige0_1
[INSPUR-gige0_1]qinq inner-vid 20 outer-vid 2 outer-priority 0
[INSPUR-gige0_1]qinq inner-vid 30 outer-vid 3 outer-priority 0
```

- (4) 验证配置。

SW1 和 SW2 的 gige0_1 口上开启了灵活 QinQ 功能，为部门 A、B 打上不同的外层 Tag。位于异地的部门 A 和部门 B 可以利用运营商分配的不同的 VLAN 进行正常通信。

9 ARP 防护典型配置案例

9.1 ARP 防护简介

9.1.1 ARP 报文有效性检查

为了防止非法用户的 ARP 报文攻击,我们可以使用 ARP 报文有效性检测功能对设备接收到的 ARP 进行检测,丢弃非法 ARP 报文,处理合法 ARP 报文。对于 ARP 信任端口不进行检查,对于 ARP 非信任端口,需要对 MAC 地址和 IP 地址不合法的报文进行过滤。检查模式有源 MAC 地址、目的 MAC 地址或 IP 地址模式。

表 9-1 检查模式

项目	说明
源 MAC 检查模式	检查 ARP 报文中的源 MAC 地址和以太网报文头部中的源 MAC 地址是否一致,一致则认为报文合法,继续处理该报文,否则丢弃。
目的 MAC 检查模式	检查 ARP 应答报文中的目的 MAC 地址是否为全 0 或者全 1,是否和以太网报文头中的目的 MAC 地址一致。全 0、全 1、不一致的报文视为非法报文,直接丢弃。
IP 地址检查模式	检查 ARP 报文中的源 IP 和目的 IP 地址,组播地址、全 0 和全 1 的地址视为不合法报文,直接丢弃。ARP 应答报文需要检查源 IP 地址和目的 IP 地址,ARP 请求报文,只检查源 IP 地址。

9.1.2 ARP 用户合法性检查

网络攻击者会使用合法用户的 IP 地址伪装成合法用户,访问网络资源并且和网络中的合法用户通信,导致网络信息传输出错,重要信息的泄露。ARP 用户合法性检测能识别出非法用户,丢弃非法报文。对于 ARP 信任端口,不进行用户合法性检查;对于 ARP 非信任端口,需要进行用户合法性检查,以防止仿冒用户的攻击。

用户合法性检测是根据 ARP 报文中源 IP 地址和源 MAC 地址检查用户是否是所属 VLAN 所在端口上的合法用户,包括基于静态 ARP 表项的检查和基于 DHCP Snooping 安全表项的检查。通常先

检查静态 ARP 表项后检查 DHCP Snooping 表项。

表 9-2 基于静态 ARP 表项和 DHCP Snooping 表项的检查

项目	说明
基于静态 ARP 表项的检查	优先检查该表项，如果用户 ARP 报文的源 IP 地址和源 MAC 地址匹配静态 ARP 表项，则认为该用户合法，转发其发送 ARP 报文。如果只匹配源 IP 不匹配源 MAC，则认为该用户非法，丢弃其发送的 ARP 报文。如果源 IP 和源 MAC 都不匹配，就继续查找 DHCP Snooping 安全表项。
基于 DHCP Snooping 表项检查	检查过静态 ARP 表项之后再检查 DHCP Snooping 安全表项，只要符合二者中任何一个，就认为该 ARP 报文合法，进行转发。如果所有检查都没有找到匹配的表项，则认为是非法报文，直接丢弃。

9.1.3 ARP 网关保护

ARP 网关保护功能可以防止伪造网关攻击，在端口上使能该功能后，当端口收到 ARP 时，将检查报文的源 IP 地址是否和被保护的网关 IP 地址相同。如果相同，则认为此报文非法，将其丢弃；否则，认为此报文合法，进行处理。

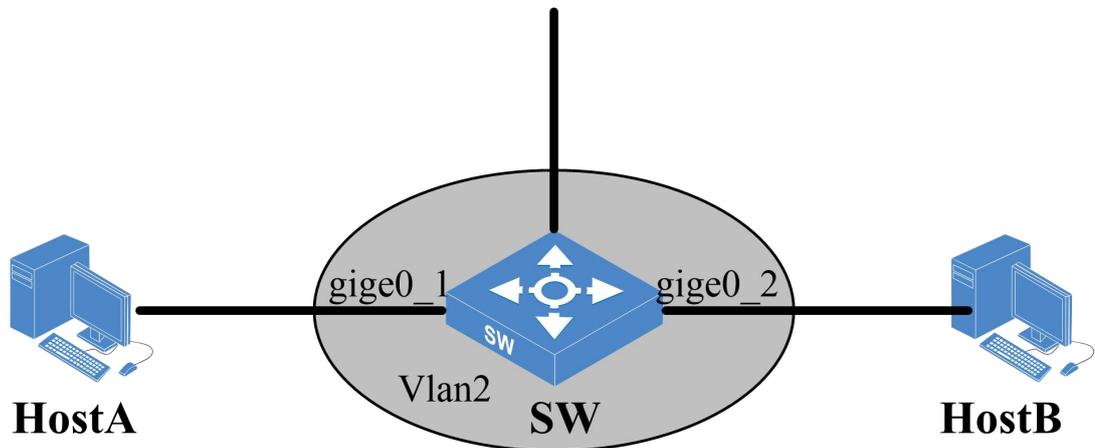
9.2 ARP 报文一致性检测配置案例

9.2.1 配置需求

公司的接入层设备上启用 ARP 报文一致性检测，可以有效防止仿冒用户的攻击。

9.2.2 网络拓扑

图 9-2 ARP 报文一致性检测组网图



9.2.3 配置流程

- (1) 在 SW 上创建 VLAN2，将 gige0_1 和 gige0_2 添加到 VLAN2。
- (2) 在 SW 的 VLAN2 启用 ARP 报文一致性功能。
- (3) 验证配置。

9.2.4 配置步骤

- (1) 在 SW 上创建 VLAN2，将 gige0_1 和 gige0_2 添加到 VLAN2。

```
<INSPUR>conf-mode
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_1
[INSPUR-vlan2]port gige0_2
[INSPUR-vlan2]exit
[INSPUR]
```

- (2) 在 SW 的 VLAN2 启用 ARP 报文一致性功能。

```
[INSPUR]arp inspection vlan 2 untrust interface gige0_1 gige0_2
```

- (3) 验证配置。

SW 上的 gige0_1 和 gige0_2 口收到 ARP 报文后，会检测 ARP 报文的源 MAC 地址和以太网首部

的源 MAC 是否相同，若相同则转发该报文，否则丢弃。

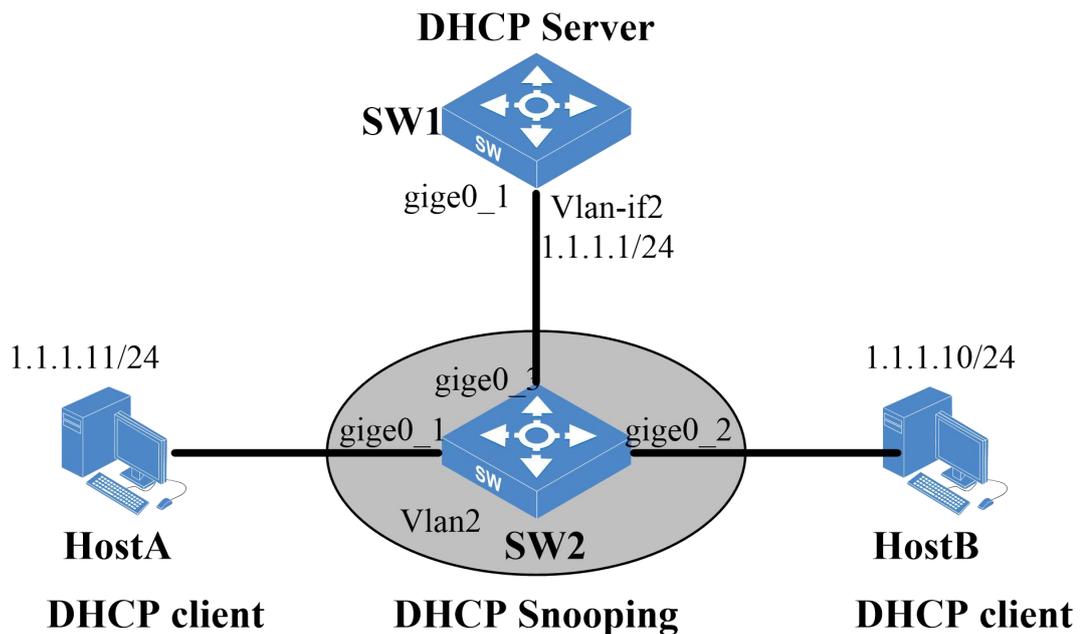
9.3 ARP 用户合法性配置案例

9.3.1 配置需求

公司的接入层交换机上启用 ARP 用户合法性检测功能后，设备在收到 ARP 报文后，会依次查询静态 ARP 表项和 DHCP Snooping 表项。如果查不到匹配的用户，将认为该 ARP 报文是非法用户发的，丢弃 ARP 报文，从而防止了 ARP 仿冒用户攻击。

9.3.2 网络拓扑

图 9-3 ARP 用户合法性检测组网图



9.3.3 配置流程

- (1) 在 SW1 上创建 vlan-if2，配置 IP 地址及 DHCP 地址池。
- (2) 在 SW2 上创建 VLAN2，将端口 gige0_1、gige0_2 和 gige0_3 添加到 VLAN2。
- (3) 在 SW2 上开启 DHCP Snooping 功能并使能记录 IP MAC 地址功能。

- (4) 在 SW2 上启用 ARP Detection 功能，设置 gige0_1 和 gige0_2 为非信任端口。
- (5) 验证配置。

9.3.4 配置步骤

- (1) 在 SW1 上创建 vlan-if2，配置 IP 地址及 DHCP 地址池。

```
<INSPUR>conf-mode
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_1
[INSPUR-vlan2]exit
[INSPUR]interface vlan-if2
[INSPUR-vlan-if2]ip address 1.1.1.1/24
[INSPUR-vlan-if2]exit
[INSPUR]dhcp server pool test
[INSPUR-dhcp-pool-test] binding interface vlan-if2
[INSPUR-dhcp-pool-test]address range 1.1.1.10 1.1.1.100 24
[INSPUR-dhcp-pool-test]default-router 1.1.1.1
[INSPUR-dhcp-pool-test]exit
[INSPUR]dhcp server enable
[INSPUR]
```

- (2) 在 SW2 上创建 VLAN2，将端口 gige0_1、gige0_2 和 gige0_3 添加到 VLAN2。

```
<INSPUR>conf-mode
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_1
[INSPUR-vlan2]port gige0_2
[INSPUR-vlan2]port gige0_3
```

- (3) 在 SW2 上开启 DHCP Snooping 功能。

```
[INSPUR]interface gige0_3
[INSPUR-gige0_3]dhcp snooping trust
[INSPUR]dhcp snooping enable
```

- (4) 在 SW2 上启用 ARP Detection 功能，设置 gige0_1 和 gige0_2 为非信任端口。

```
[INSPUR]arp inspection vlan 2 untrust interface gige0_1 gige0_2
```

- (5) 验证配置。

SW2上启用了 DHCP Snooping 功能, HostA 和 HostB 获取地址后会在 SW2 上形成一个 Snooping 信息列表, 记录了客户端的 MAC 地址、IP 地址以及对应的端口。当 SW2 的 gige0_1 和 gige0_2 口收到 ARP 报文后, 会查询 DHCP Snooping 表项, 若查到匹配用户则转发 ARP 报文, 否则丢弃。

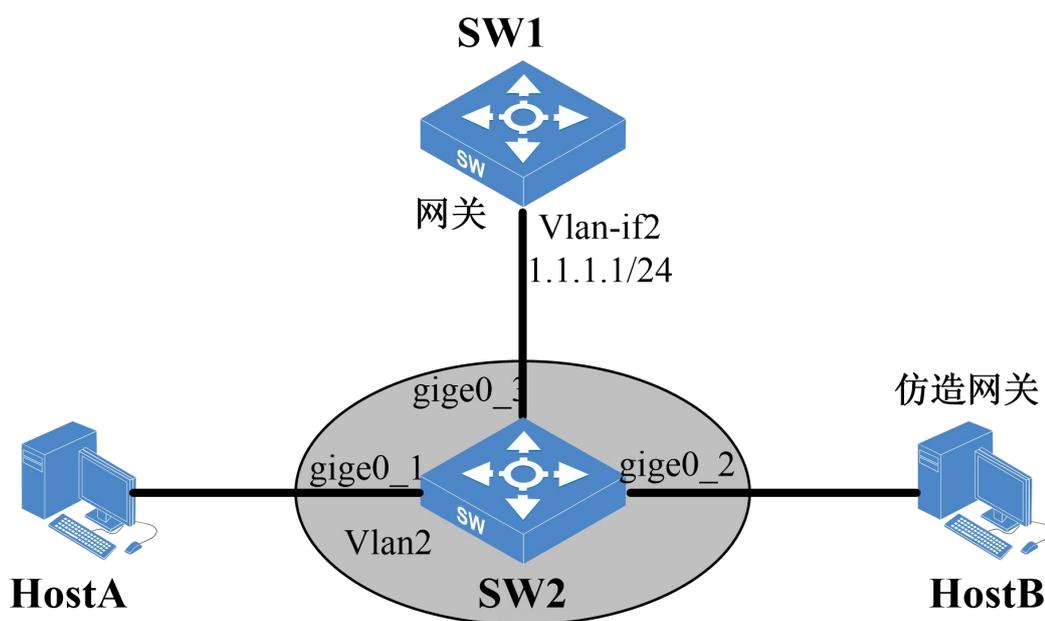
9.4 ARP 网关保护配置案例

9.4.1 配置需求

用户的网络中可能会存在网关攻击的风险, ARP 网关保护功能有效防止网关攻击。

9.4.2 网络拓扑

图 9-4 ARP 网关保护组网图



9.4.3 配置流程

- (1) 在 SW2 上创建 VLAN2, 将端口 gige0_1、gige0_2 和 gige0_3 添加到 VLAN2。
- (2) 在 SW2 上启用 ARP 网关保护功能, 设置 gige0_1 和 gige0_2 为非信任端口。

(3) 验证配置。

9.4.4 配置步骤

(1) 在 SW2 上创建 VLAN2，将端口 gige0_1、gige0_2 和 gige0_3 添加到 VLAN2。

```
<INSPUR>conf-mode
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_1
[INSPUR-vlan2]port gige0_2
[INSPUR-vlan2]port gige0_3
```

(2) 在 SW2 上启用 ARP 网关保护功能，设置 gige0_1 和 gige0_2 为非信任端口。

```
[INSPUR]arp inspection vlan 2 untrust interface gige0_1 gige0_2 filter source
1.1.1.1
```

(3) 验证配置。

在 SW2 上配置网关保护后，当 gige0_2 收到了来自 HostB 发来的伪造网关的 ARP 报文，将会被丢弃，防止 HostA 学习到伪造网关的 MAC 地址，将与网关设备 SW1 的通信报文错误地发送到 HostB 上。

10 路由协议典型配置案例

10.1 路由协议简介

10.1.1 静态路由协议简介

静态路由是在交换机上手动配置的路由信息，不会把路由信息传递给其他设备。静态路由一般适用于比较简单的网络环境，在这样的环境中，网络管理员易于清楚地了解网络的拓扑结构，便于设置正确的路由信息。

10.1.2 RIP 路由协议简介

RIP（Routing Information Protocol，路由信息协议）是一种较为简单的内部网关协议（Interior Gateway Protocol，IGP）。RIP 是最早的距离矢量路由协议，尽管 RIP 缺少许多更为高级的路由协议所具备的复杂功能，但使用的简单性和广泛性使其具有很强的生命力。RIP 一般适用于小型同类网络的一个自治系统（AS）内的路由信息的传递。

10.1.3 OSPF 路由协议简介

OSPF（Open Shortest Path First，开放最短路径优先）是一种典型的链路状态路由协议，采用 OSPF 的路由器彼此交换并保存整个网络的链路信息，从而掌握全网的拓扑结构，独立计算路由。OSPF 作为一种内部网关协议（Interior Gateway Protocol，IGP），用于在同一个自治域（AS）中的路由器之间发布路由信息。区别于距离矢量协议(RIP)，OSPF 具有支持大型网络、路由收敛快、占用网络资源少等优点，在目前应用的路由协议中占有相当重要的地位。

OSPF 支持多进程配置，在同一台设备上可以运行多个 OSPF 进程，进程之间互不影响，彼此独立。不同 OSPF 进程之间的路由交互相当于不同路由协议之间的路由交互。支持多个 OSPF 进程公用一个 RID，路由器的一个接口只能属于某一个 OSPF 进程。

10.1.4 策略路由协议简介

策略路由（Policy-Based Routing 简称 PBR）是一种依据用户制定的策略进行路由选择的机制，与单纯依照 IP 报文的目的地址查找路由表进行转发不同，可灵活应用于安全、负载分担等场景。策略路由不仅能够根据目的地址发送，同时可以与访问控制列表配合使用，所以报文发送还可基于报文入接口、协议类型、报文的 TOS 字段等进行匹配从而执行指定的操作（设置报文的下一跳和缺省下一跳等）

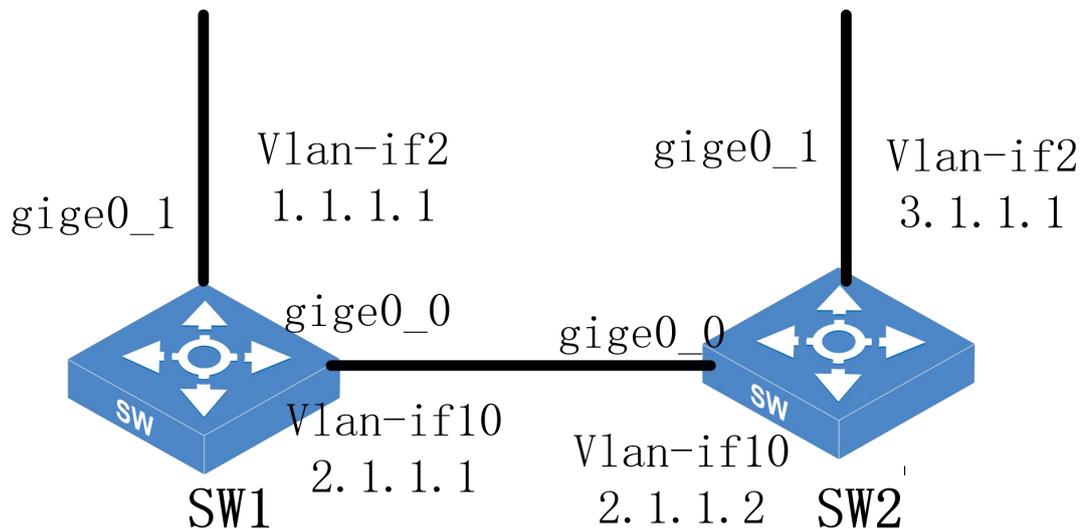
10.2 静态路由配置案例

10.2.1 配置需求

某公司内网仅有几台交换机，组网比较简单，需要实现各个网段之间的通信。

10.2.2 网络拓扑

图 10-1 静态路由组网图



10.2.3 配置流程

- (1) 在 SW1 和 SW2 上添加 VLAN2、VLAN10，把端口划入相应 VLAN，并且配置 vlan-if 的 IP 地址。
- (2) 在 SW1 和 SW2 上配置静态路由。
- (3) 验证配置。

10.2.4 配置步骤

- (1) 在 SW1 和 SW2 上配置 VLAN 及端口

在 SW1 上配置

```
<INSPUR>conf-mode
[INSPUR]vlan 2
[INSPUR-vlan 2]exit
[INSPUR]vlan 10
[INSPUR-vlan 10]exit
[INSPUR]interface gige0_0
[INSPUR-gige0_0]switchport mode access
```

```
[INSPUR-gige0_0]switchport access vlan 10
[INSPUR-gige0_0]exit
[INSPUR] interface gige0_1
[INSPUR-gige0_1]switchport access vlan
[INSPUR-gige0_1]switchport access vlan 2
[INSPUR-gige0_1]exit
[INSPUR]interface vlan-if2
[INSPUR-vlan-if2]ip address 1.1.1.1/24
[INSPUR-vlan-if2]exit
[INSPUR]interface vlan-if10
[INSPUR-vlan-if10]ip address 2.1.1.1/24
[INSPUR-vlan-if10]
```

在 SW2 上配置

```
[INSPUR]vlan 2
[INSPUR-vlan 2]exit
[INSPUR]vlan 10
[INSPUR-vlan 2]exit
[INSPUR]interface gige0_0
[INSPUR-gige0_0]switchport access vlan
[INSPUR-gige0_0]switchport access vlan 10
[INSPUR-gige0_0]exit
[INSPUR]interface gige0_1
[INSPUR-gige0_1]switchport access vlan
[INSPUR-gige0_1]switchport access vlan 2
[INSPUR-gige0_1]exit
[INSPUR]interface vlan-if2
[INSPUR-vlan-if2]ip address 3.1.1.1/24
[INSPUR-vlan-if2]exit
[INSPUR] interface vlan-if10
[INSPUR-vlan-if10]ip address 2.1.1.2/24
[INSPUR-vlan-if10]
```

(2) 在 SW1 和 SW2 上配置静态路由

在 SW1 上配置

```
[INSPUR]ip route 3.1.1.0 255.255.255.0 2.1.1.2
[INSPUR]
```

在 SW2 上配置

```
[INSPUR]ip route 1.1.1.0 255.255.255.0 2.1.1.1
[INSPUR]
```

(3) 验证配置。

```
<INSPUR>show ip route
```

1.1.1.0 网段和 3.1.1.0 网段可以正常通信

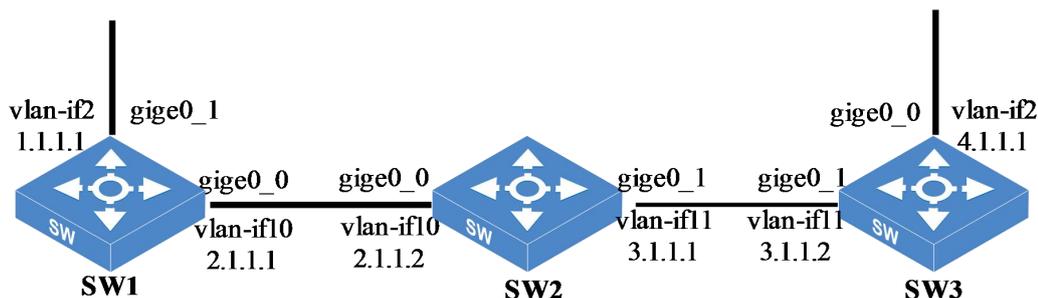
10.3 RIP 路由配置案例

10.3.1 配置需求

某公司内网规模较小，结构设备简单。使用 RIP 路由协议进行路由管理较为简便。

10.3.2 网络拓扑

图 10-2 RIP 路由组网图



10.3.3 配置流程

- (1) 在 SW1、SW2 和 SW3 上创建对应的 VLAN，把端口划入相应 VLAN，并且配置 vlan-if 的 IP 地址。
- (2) 在 SW1、SW2 和 SW3 上配置 RIP 路由。
- (3) 验证配置。

10.3.4 配置步骤

- (1) 在 SW1、SW2 和 SW3 上创建对应的 VLAN，把端口划入相应 VLAN，并且配置 vlan-if 的 IP 地址。

在 SW1 上配置

```
[INSPUR]vlan 2
[INSPUR-vlan2]exit
[INSPUR]vlan 10
[INSPUR-vlan10]exit
[INSPUR]interface gige0_0
[INSPUR-gige0_0]switchport mode access
[INSPUR-gige0_0]switchport access vlan 10
[INSPUR-gige0_0]exit
[INSPUR]interface gige0_1
[INSPUR-gige0_1]switchport mode access
[INSPUR-gige0_1]switchport access vlan 2
[INSPUR]interface vlan-if2
[INSPUR-vlan-if2]ip address 1.1.1.1/24
[INSPUR-vlan-if2]exit
[INSPUR]interface vlan-if10
[INSPUR-vlan-if10]ip address 2.1.1.1/24
[INSPUR-vlan-if10]
```

在 SW2 上配置

```
[INSPUR]vlan 10
[INSPUR-vlan10]exit
[INSPUR]vlan 11
[INSPUR-vlan11]exit
[INSPUR]interface gige0_0
[INSPUR-gige0_0]switchport mode access
[INSPUR-gige0_0]switchport access vlan 10
[INSPUR-gige0_0]exit
[INSPUR]interface gige0_1
[INSPUR-gige0_1]switchport mode access
[INSPUR-gige0_1]switchport access vlan 11
[INSPUR]interface vlan-if10
[INSPUR-vlan-if10]ip address 2.1.1.2/24
[INSPUR-vlan-if10]exit
[INSPUR]interface vlan-if11
[INSPUR-vlan-if11]ip address 3.1.1.1/24
[INSPUR-vlan-if11]
```

在 SW3 上配置

```
[INSPUR]vlan 2
[INSPUR-vlan2]exit
```

```
[INSPUR]vlan 11
[INSPUR-vlan11]exit
[INSPUR]interface gige0_0
[INSPUR-gige0_0]switchport mode access
[INSPUR-gige0_0]switchport access vlan 2
[INSPUR-gige0_0]exit
[INSPUR]interface gige0_1
[INSPUR-gige0_1]switchport mode access
[INSPUR-gige0_1] switchport access vlan 11
[INSPUR] interface vlan-if2
[INSPUR-vlan-if2]ip address 4.1.1.1/24
[INSPUR-vlan-if2]exit
[INSPUR]interface vlan-if11
[INSPUR-vlan-if11]ip address 3.1.1.2/24
[INSPUR-vlan-if11]
```

(2) 在 SW1、SW2 和 SW3 上配置 RIP 路由

在 SW1 上配置

```
[INSPUR]router rip
[INSPUR-rip]network 1.1.1.0/24
[INSPUR-rip]network 2.1.1.0/24
[INSPUR-rip]
```

在 SW2 上配置

```
[INSPUR]router rip
[INSPUR-rip]network 2.1.1.0/24
[INSPUR-rip]network 3.1.1.0/24
[INSPUR-rip]
```

在 SW3 上配置

```
[INSPUR]router rip
[INSPUR-rip]network 3.1.1.0/24
[INSPUR-rip]network 4.1.1.0/24
[INSPUR-rip]
```

(3) 验证配置。

```
<INSPUR>show ip rip
<INSPUR>show ip route
```

1.1.1.0 网段和 4.1.1.0 网段可以正常通信

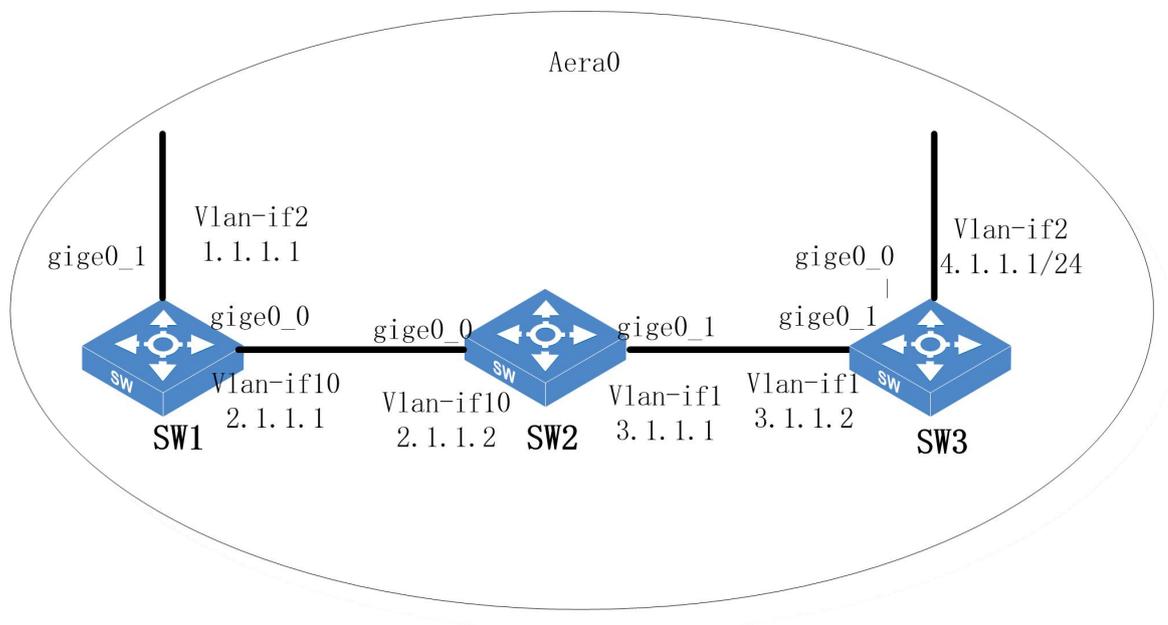
10.4 OSPF 典型配置案例

10.4.1 配置需求

某公司内新增规划部门，增加三台交换机，共有四个网段；整个网络使用 OSPF 路由协议组网，新增的交换机加入内网中，要求各个网段与公司内网正常通信。

10.4.2 网络拓扑

图 10-3 OSPF 路由组网图



10.4.3 配置流程

- (1) 在 SW1、SW2 和 SW3 上创建对应的 VLAN，把端口划入相应的 VLAN，并且配置 vlan-if 的 IP 地址。
- (2) 在 SW1、SW2 和 SW3 上配置 OSPF 路由。
- (3) 验证配置。

10.4.4 配置步骤

- (1) 在 SW1、SW2 和 SW3 上创建对应的 VLAN，把端口划入相应的 VLAN，并且配置 vlan-if 的 IP 地址。

在 SW1 上配置

```
[INSPUR]vlan 2
[INSPUR]vlan 10
[INSPUR]interface gige0_0
[INSPUR-gige0_0]switchport mode access
[INSPUR-gige0_0] switchport access vlan 10
[INSPUR-gige0_0]exit
[INSPUR]interface gige0_1
[INSPUR-gige0_1]switchport mode access
[INSPUR-gige0_1] switchport access vlan 2
[INSPUR]interface vlan-if2
[INSPUR-vlan-if2]ip address 1.1.1.1/24
[INSPUR-vlan-if2]exit
[INSPUR]interface vlan-if10
[INSPUR-vlan-if10]ip address 2.1.1.1/24
[INSPUR-vlan-if10]
```

在 SW2 上配置

```
[INSPUR]vlan 10
[INSPUR]vlan 11
[INSPUR]interface gige0_0
[INSPUR-gige0_0]switchport mode access
[INSPUR-gige0_0]switchport access vlan 10
[INSPUR-gige0_0]exit
[INSPUR]interface gige0_1
[INSPUR-gige0_1]switchport mode access
[INSPUR-gige0_1]switchport access vlan 11
[INSPUR]interface vlan-if10
[INSPUR-vlan-if10]ip address 2.1.1.2/24
[INSPUR-vlan-if10]exit
[INSPUR]interface vlan-if11
[INSPUR-vlan-if11]ip address 3.1.1.1/24
[INSPUR-vlan-if11]
```

在 SW3 上配置

```
[INSPUR]vlan 2
[INSPUR]vlan 11
[INSPUR]interface gige0_0
[INSPUR-gige0_0]switchport mode access
[INSPUR-gige0_0]switchport access vlan 2
[INSPUR-gige0_0]exit
[INSPUR]interface gige0_1
[INSPUR-gige0_1]switchport mode access
[INSPUR-gige0_1]switchport access vlan 11
[INSPUR]interface vlan-if2
[INSPUR-vlan-if2]ip address 4.1.1.1/24
[INSPUR-vlan-if2]exit
[INSPUR]interface vlan-if11
[INSPUR-vlan-if11]ip address 3.1.1.2/24
[INSPUR-vlan-if11]
```

(2) 在 SW1、SW2 和 SW3 上配置 OSPF 路由

在 SW1 上配置

```
[INSPUR]router ospf 1
[INSPUR-ospf-1]network 1.1.1.0/24 area 0
[INSPUR-ospf-1]network 2.1.1.0/24 area 0
[INSPUR-ospf-1]
```

在 SW2 上配置

```
[INSPUR]router ospf 1
[INSPUR-ospf-1]network 2.1.1.0/24 area 0
[INSPUR-ospf-1]network 3.1.1.0/24 area 0
[INSPUR-ospf-1]
```

在 SW3 上配置

```
[INSPUR]router ospf 1
[INSPUR-ospf-1]network 3.1.1.0/24 area 0
[INSPUR-ospf-1]network 4.1.1.0/24 area 0
[INSPUR-ospf-1]
```

(3) 验证配置。

```
<INSPUR>show ip ospf
<INSPUR>show ip route
```

1.1.1.0 网段和 4.1.1.0 网段可以正常通信

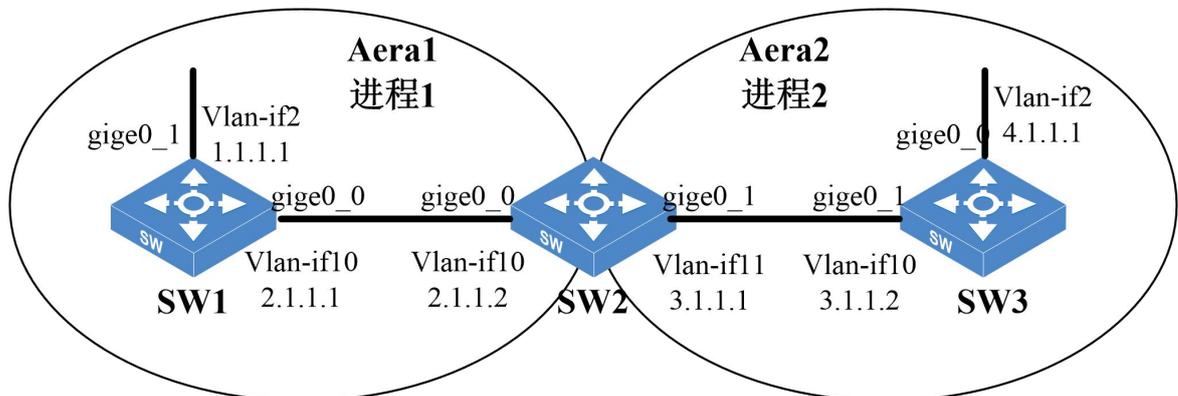
10.5 OSPF 多进程典型配置案例

10.5.1 配置需求

当用户的组网比较复杂，想要实现分区管理，使某些区域的设备独立出来不受其它区域的路由信息影响，可以使用多个 OSPF 进程进行路由分区管理。不同的 OSPF 进程之间路由信息是隔离的，若想要学习到其它进程的路由信息，可以将该进程的路由信息引用过来。

10.5.2 网络拓扑

图 10-4 OSPF 多进程组网图



10.5.3 配置流程

- (1) 在 SW1、SW2 和 SW3 上创建对应的 VLAN，把端口划入相应的 VLAN，并且配置 vlan-if 的 IP 地址。
- (2) 在 SW1、SW2 和 SW3 上配置 OSPF 路由。
- (3) 验证配置。

10.5.4 配置步骤

- (1) 在 SW1、SW2 和 SW3 上创建对应的 VLAN，把端口划入相应的 VLAN，并且配置 vlan-if 的 IP 地址。

在 SW1 上配置

```
[INSPUR]vlan 2
[INSPUR]vlan 10
[INSPUR]interface gige0_0
[INSPUR-gige0_0]switchport mode access
[INSPUR-gige0_0]switchport access vlan 10
[INSPUR-gige0_0]exit
[INSPUR]interface gige0_1
[INSPUR-gige0_1]switchport mode access
[INSPUR-gige0_1]switchport access vlan 2
[INSPUR]interface vlan-if2
[INSPUR-vlan-if2]ip address 1.1.1.1/24
[INSPUR-vlan-if2]exit
[INSPUR]interface vlan-if10
[INSPUR-vlan-if10]ip address 2.1.1.1/24
[INSPUR-vlan-if10]
```

在 SW2 上配置

```
[INSPUR]vlan 10
[INSPUR]vlan 11
[INSPUR]interface gige0_0
[INSPUR-gige0_0]switchport mode access
[INSPUR-gige0_0]switchport access vlan 10
[INSPUR-gige0_0]exit
[INSPUR]interface gige0_1
[INSPUR-gige0_1]switchport mode access
[INSPUR-gige0_1]switchport access vlan 11
[INSPUR]
[INSPUR]interface vlan-if10
[INSPUR-vlan-if10]ip address 2.1.1.2/24
[INSPUR-vlan-if10]exit
[INSPUR]interface vlan-if11
[INSPUR-vlan-if11]ip address 3.1.1.1/24
[INSPUR-vlan-if11]
```

在 SW3 上配置

```
[INSPUR]vlan 2
[INSPUR]vlan 11
[INSPUR]interface gige0_0
[INSPUR-gige0_0]switchport mode access
[INSPUR-gige0_0]switchport access vlan 2
```

```
[INSPUR-gige0_0]exit
[INSPUR]interface gige0_1
[INSPUR-gige0_1]switchport mode access
[INSPUR-gige0_1]switchport access vlan 11
[INSPUR]interface vlan-if2
[INSPUR-vlan-if2]ip address 4.1.1.1/24
[INSPUR-vlan-if2]exit
[INSPUR]interface vlan-if11
[INSPUR-vlan-if11]ip address 3.1.1.2/24
[INSPUR-vlan-if11]
```

(2) 在 SW1、SW2 和 SW3 上配置 OSPF 路由。

在 SW1 上配置

```
[INSPUR]router ospf 1
[INSPUR-ospf-1]network 1.1.1.0/24 area 1
[INSPUR-ospf-1]network 2.1.1.0/24 area 1
[INSPUR-ospf-1]
```

在 SW2 上配置

```
[INSPUR]router ospf 1
[INSPUR-ospf-1]network 2.1.1.0/24 area 1
[INSPUR-ospf-1]exit
[INSPUR]router ospf 2
[INSPUR-ospf-2]network 3.1.1.0/24 area 2
[INSPUR-ospf-2]
```

在 SW3 上配置

```
[INSPUR]router ospf 2
[INSPUR-ospf-2]network 3.1.1.0/24 area 2
[INSPUR-ospf-2]network 4.1.1.0/24 area 2
[INSPUR-ospf-2]
```

(3) 验证配置。

使用了进程 1 和进程 2，使得区域 1 和区域 2 的路由信息隔离开来，两个区域的路由信息互不影响，SW1 与 SW3 不能互通，可以达到用户隔离某些特殊区域的效果。但是当用户想要 SW1 能访问 SW3 时，可以在 SW2 上进入进程 1 的配置视图引用进程 2 的路由：

```
[INSPUR]route ospf 1
[INSPUR-ospf-1]redistribute ospf 2
```

反之，用户想要 SW3 能访问 SW1 时，可以在 SW2 上进入进程 2 的配置视图引用进程 1 的路由：

```
[INSPUR]route ospf 2
[INSPUR-ospf-2]redistribute ospf 1
```

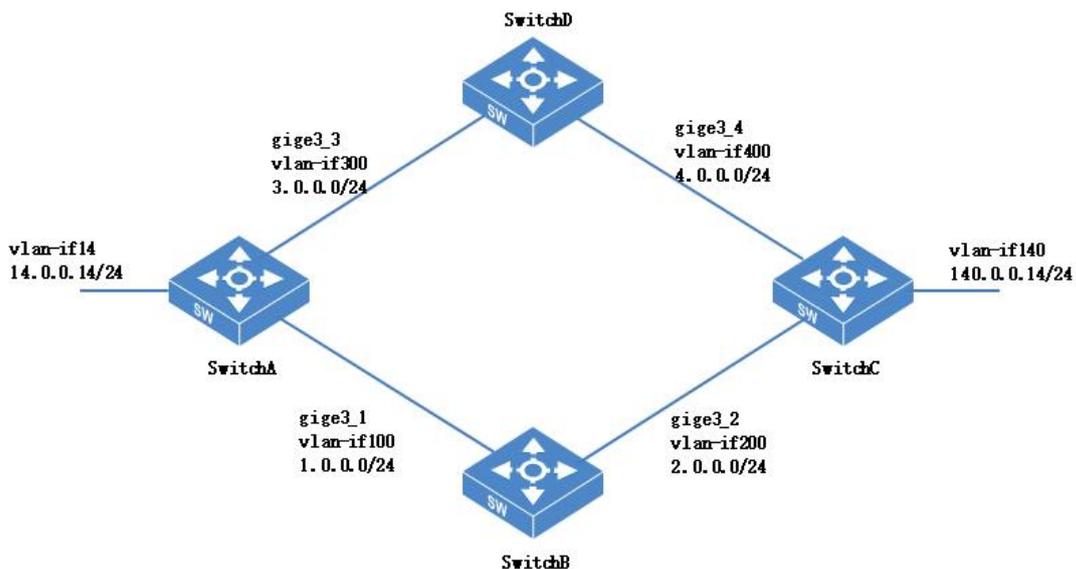
10.6 策略路由配置案例

10.6.1 配置需求

某客户组网搭建时有特殊路径需求，要求针对特定流量走单独路径，从而实现特殊场景能够独立选路，且不影响其他场景业务正常通信。

10.6.2 网络拓扑

图 10-5 策略路由组网图



10.6.3 配置流程

- (1) 在 SwitchA、SwitchB、SwitchC、SwitchD 上创建对应的 VLAN，把端口划入相应的 VLAN，并且配置 vlan-if 的 IP 地址。
- (2) SwitchA、SwitchB 和 SwitchC 之间运行 OSPF 协议。

- (3) SwitchA、SwitchD 和 SwitchC 之间运行 RIP 协议。
- (4) 将 140.0.0.0/24 网段发布到 RIP 和 OSPF 协议中。
- (5) 通过策略路由使从 SwitchA 去往 140.0.0.0/24 的 UDP 流量中目的端口为 6000 的走上面链路（默认走下面链路）。

10.6.4 配置步骤

- (1) 创建 VLAN 并配置各接口，配置 SwitchA。SwitchB、SwitchC 和 SwitchD 的配置与 SwitchA 类似。

在 SwitchA 上配置

```
[INSPUR]sysname SwitchA
[SwitchA]vlan 100
[SwitchA-vlan100]port gige3_1
[SwitchA-vlan100]exit
[SwitchA]vlan 300
[SwitchA-vlan300]port gige3_3
[SwitchA-vlan300]exit
[SwitchA]interface vlan-if100
[SwitchA-vlan-if100] ip address 1.0.0.1/24
[SwitchA-vlan-if100]exit
[SwitchA]interface vlan-if300
[SwitchA-vlan-if300]ip address 3.0.0.1/24
[SwitchA-vlan-if300]exit
[SwitchA-vlan14]port gige3_14
[SwitchA-vlan14]exit
[SwitchA]interface vlan-if14
[SwitchA-vlan-if14]ip address 14.0.0.1/24
[SwitchA-vlan-if14]exit
[SwitchA]
```

在 SwitchB 上配置

```
[INSPUR]sysname SwitchB
[SwitchB]vlan 100
[SwitchB-vlan100]port gige3_1
[SwitchB-vlan100]exit
[SwitchB]vlan 200
```

```
[SwitchB-vlan200]port gige3_2
[SwitchB-vlan200]exit
[SwitchB]interface vlan-if 100
[SwitchB-vlan-if100] ip address 1.0.0.2/24
[SwitchB-vlan-if100]exit
[SwitchB]interface vlan-if 200
[SwitchB-vlan-if200]ip address 2.0.0.1/24
[SwitchB-vlan-if200]exit
[SwitchB]
```

在 SwitchC 上配置

```
[INSPUR]sysname SwitchC
[SwitchC]vlan 200
[SwitchC-vlan200]port gige3_2
[SwitchC-vlan200]exit
[SwitchC]vlan 400
[SwitchC-vlan400]port gige3_4
[SwitchC-vlan400]exit
[SwitchC]vlan 140
[SwitchC-vlan140]port gige3_14
[SwitchC-vlan140]exit
[SwitchC]interface vlan-if 200
[SwitchC-vlan-if200] ip address 2.0.0.2/24
[SwitchC-vlan-if200]exit
[SwitchC]interface vlan-if 400
[SwitchC-vlan-if400]ip address 4.0.0.2/24
[SwitchC-vlan-if400]exit
[SwitchC]interface vlan-if 140
[SwitchC-vlan-if140]ip address 140.0.0.1/24
[SwitchC-vlan-if140]exit
[SwitchC]
```

在 SwitchD 上配置

```
[INSPUR]sysname SwitchD
[SwitchD]vlan 300
[SwitchD-vlan300]port gige3_3
[SwitchD-vlan300]exit
[SwitchD]vlan 400
[SwitchD-vlan400]port gige3_4
[SwitchD-vlan400]exit
[SwitchD]interface vlan-if 300
```

```
[SwitchD-vlan-if300] ip address 3.0.0.2/24
[SwitchD-vlan-if300]exit
[SwitchD]interface vlan-if 400
[SwitchD-vlan-if400]ip address 4.0.0.1/24
[SwitchD-vlan-if400]exit
[SwitchD]
```

(2) SwitchA、SwitchB 和 SwitchC 之间运行 OSPF 协议。

在 SwitchA 上配置

```
[SwitchA]router ospf 1
[SwitchA-ospf-1]ospf router-id 1.1.1.1
[SwitchA-ospf-1]network 1.0.0.0/24 area 0.0.0.0
[SwitchA-ospf-1]exit
[SwitchA]
```

在 SwitchB 上配置

```
[SwitchB]router ospf 1
[SwitchB-ospf-1]ospf router-id 2.2.2.2
[SwitchB-ospf-1]network 1.0.0.0/24 area 0.0.0.0
[SwitchB-ospf-1]network 2.0.0.0/24 area 0.0.0.0
[SwitchB-ospf-1]exit
[SwitchB]
```

在 SwitchC 上配置

```
[SwitchC]router ospf 1
[SwitchC-ospf-1]ospf router-id 3.3.3.3
[SwitchC-ospf-1]network 2.0.0.0/24 area 0.0.0.0
[SwitchC-ospf-1]exit
[SwitchC]
```

(3) SwitchA、SwitchD 和 SwitchC 之间运行 RIP 协议。

配置 SwitchA

```
[SwitchA]router rip
[SwitchA-rip]network vlan-if300
[SwitchA-rip]exit
[SwitchA]
```

配置 SwitchD

```
[SwitchD]router rip
[SwitchD-rip]network vlan-if300
```

```
[SwitchD-rip]network vlan-if400
[SwitchD-rip]exit
[SwitchD]
```

配置 SwitchC

```
[SwitchC]router rip
[SwitchC-rip]network vlan-if400
[SwitchC-rip]exit
[SwitchC]
```

(4) 将 140.0.0.0/24 网段发布到 RIP 和 OSPF 协议中。

```
[SwitchC]router rip
[SwitchC-rip] network vlan-if140
[SwitchC-rip]exit
[SwitchC]route ospf 1
[SwitchC-ospf-1]network 140.0.0.0/24 area 0.0.0.0
[SwitchC-ospf-1]exit
[SwitchC]
```

在 SwitchA 上查看去往 140.0.0.0/24 网段的路由

```
[SwitchA]show ip route detail
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
I - ISIS, B - BGP, > - selected route, * - FIB route, G - GUARD
O 1.0.0.0/24 [110/1] fmap : 0x0 is directly connected, vlan-if100, instance
1, 00:27:17
C>* 1.0.0.0/24 fmap : 0x1 is directly connected, vlan-if100 weight 0
O>* 2.0.0.0/24 [110/2] fmap : 0x1 via 1.0.0.2, instance 1, 00:26:29
C>* 3.0.0.0/24 fmap : 0x1 is directly connected, vlan-if300 weight 0
R>* 4.0.0.0/24 [120/2] fmap : 0x1 via 3.0.0.2, 00:05:50
C>* 10.24.0.0/16 fmap : 0x1 is directly connected, vlan-if1 weight 0
R 140.0.0.0/24 [120/3] fmap : 0x0 via 3.0.0.2, 00:02:40
O>* 140.0.0.0/24 [110/3] fmap : 0x1 via 1.0.0.2, instance 1, 00:07:28
```

(5) 查看 SwitchA 是否已为策略路由分配好 ACL 资源（对应物理槽位的资源需要有 rt-policy 的）

```
[SwitchA] show acl resource slot 3 ingress
Slice information:
  Slice  0: Reserved
  Slice  1: Reserved
  Slice  2: ipv4
  Slice  3: ipv4
  Slice  4: ipv4
```

```
Slice 5: rt-policy
Slice 6: mac
Slice 7: mac
Slice 8: ipv4
Slice 9: mix-stream
Slice 10: mix-stream
Slice 11: mix-stream
Slice 12: Reserved
Slice 13: Reserved
Slice 14: Reserved
Slice 15: Reserved
```

```
[SwitchA]
```



说明 如果没有 **rt-policy** 资源，需手动调整其中一条 **Slice** 资源给策略路由使用

```
[SwitchA]policy-route mode switch
[SwitchA]acl resource slot 4 ingress slice 8 mode rt-policy
[SwitchA]
```

- (6) 配置策略路由使从 **SwitchA** 去往 **140.0.0.0/24** 的 **UDP** 流量中目的端口为 **6000** 的流量走上面 **RIP** 协议链路（默认走下面链路）。

```
[SwitchA]policy-route mode switch
[SwitchA]policy-route switch 123 permit
[SwitchA-policy-route-switch-123] match dst-ip 140.0.0.0/24
[SwitchA-policy-route-switch-123] match protocol UDP dst-port 6000 to 6000
[SwitchA-policy-route-switch-123] set output-interface vlan-if300 next-hop
3.0.0.2
[SwitchA-policy-route-switch-123]commit
[SwitchA-policy-route-switch-123]exit
[SwitchA]
```


11 DHCP 典型配置案例

11.1 DHCP 简介

DHCP (Dynamic Host Configuration Protocol, 动态主机配置协议), 用于给内部网络自动分配 IP 地址, 方便用户网络管理员对所有计算机做管理, 也使 PC 及无线网络的使用更加方便。DHCP 采用客户端/服务器通信模式, 由客户端主动向服务器申请 IP 地址及相应的配置, 以实现 IP 地址等信息的动态配置。

客户端通过动态分配方式获取到的 IP 地址是有租约期限的, 租约时间到之后服务器将收回该地址。但是如果客户端想要继续使用该地址的话, 可以通过主动续约的方式获取该地址的使用权。在地址租约到期前, 客户端主动给服务器发送续约报文, 如果服务器确定该地址能继续给该客户端使用, 就回复客户端续约成功。



说明

DHCP 客户端申请地址的报文是以广播方式发送的, 只能在同一网段内广播, 当客户端和服务不在一个网段时, 可通过 DHCP 中继功能实现跨网段获取 IP 地址。

DHCP 分配地址的方式有两种:

表 11-1 DHCP 分配方式

项目	说明
动态分配	客户端从 DHCP 服务器上分配到的 IP 地址不能永久使用, 是有一个有效期的。
手工分配	手工为某些客户端分配固定的 IP 地址, 将 IP 地址和客户端静态绑定, 可以永久使用。

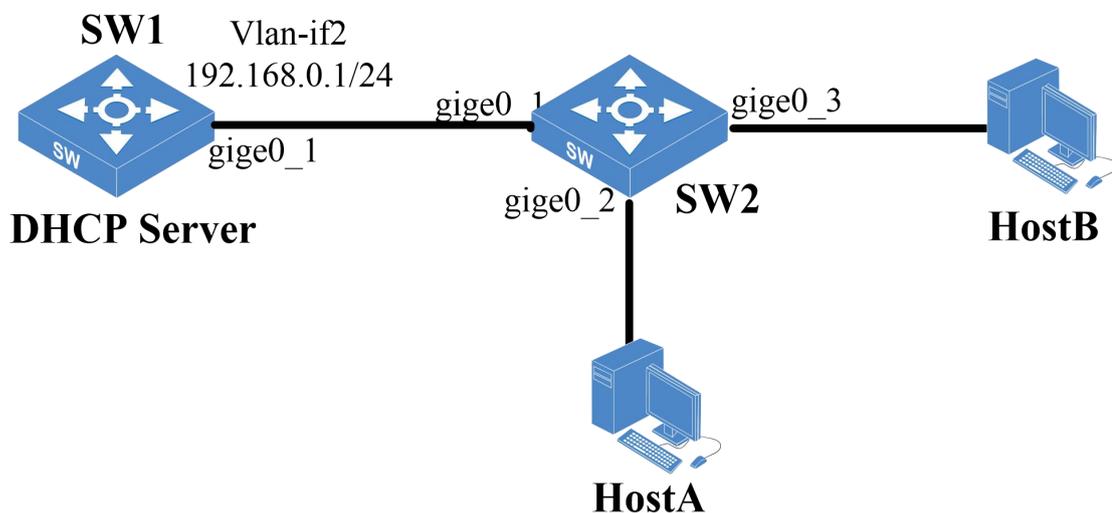
11.2 DHCP Server 配置案例

11.2.1 配置需求

当网络规模较大，手工配置客户端 IP 地址工作量很大时，使用 DHCP Server 动态分配 IP 地址能有效管理网络。

11.2.2 网络拓扑

图 11-2 DHCP Server 组网图



11.2.3 配置流程

- (1) 在 SW1 上创建 vlan-if2，配置 IP 地址为 192.168.0.1/24，将 gige0_1 添加到 Vlan-if2。
- (2) 在 SW1 使能 DHCP Server 并创建动态地址池和静态绑定地址。
- (3) 在 SW2 上创建 vlan2，添加 gige0_1、gige0_2、gige0_3 到 vlan2。
- (4) 验证配置。

11.2.4 配置步骤

- (1) 在 SW1 上创建 vlan2，配置 IP 地址为 192.168.0.1/24，将 gige0_1 添加到 Vlan2。

```
<INSPUR>conf-mode
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_1
[INSPUR-vlan2]exit
[INSPUR]interface vlan-if2
[INSPUR-vlan-if2]ip address 192.168.0.1/24
[INSPUR-vlan-if2]exit
[INSPUR]
```

(2) 在 SW1 使能 DHCP Server 并创建动态地址池和静态绑定地址。

```
[INSPUR]dhcp server pool 192
[INSPUR-dhcp-pool-192]address range 192.168.0.10 192.168.0.100 24
[INSPUR-dhcp-pool-192]binding interface vlan-if2
[INSPUR-dhcp-pool-192]lease 1440
[INSPUR-dhcp-pool-192]default-router 192.168.0.1
[INSPUR-dhcp-pool-192]dns-server 172.153.0.1
[INSPUR-dhcp-pool-192]static-bind ip-address 192.168.0.120 mac-address
00:10:94:00:00:01 client-name administrator
[INSPUR-dhcp-pool-192]exit
[INSPUR]dhcp server enable
[INSPUR]
```

(3) 在 SW2 上创建 vlan2，添加 gige0_1、gige0_2、gige0_3 到 vlan2。

```
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_1
[INSPUR-vlan2]port gige0_2
[INSPUR-vlan2]port gige0_3
```

(4) 验证配置

查看 DHCP Server 动态地址池及静态绑定地址。

```
<INSPUR>show dhcp server pool 192
Pool 192:
Address range : 192.168.0.10 to 192.168.0.100
Mask          : 255.255.255.0
Lease time    : 1 days 0 hours 0 mins
Static bind ip address 192.168.0.120 mac address 00:10:94:00:00:01
<INSPUR>
```

HostA 和 HostB 主动申请地址后，设备上会显示 DHCP Server 分配出去的地址，当申请 IP 地址的主机 MAC 地址匹配静态绑定项时，主机将获取该绑定的 IP 地址。可进入用户视图使用命令 show

dhcp-server ip-in-use 查看地址池已分配出去的 IP 地址。

11.3 DHCP 中继配置案例

11.3.1 配置需求

当 DHCP 的客户端和 DHCP Server 不在同一网段时，DHCP Server 收不到客户端发送地址请求报文，此时可以在客户端和 DHCP Server 之间的设备上启用 DHCP 中继功能，让中继设备转发客户端和服务器的互通报文。

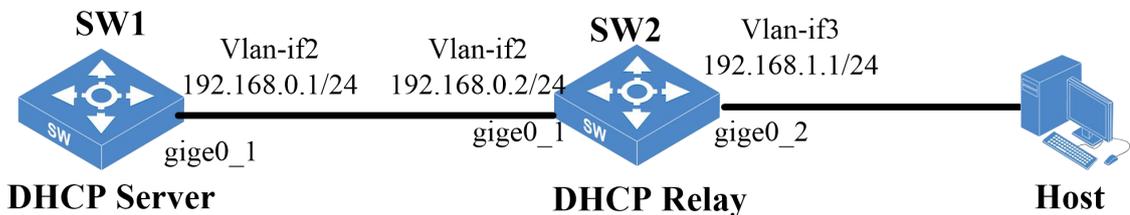


说明

DHCP 客户端的地址请求报文是以广播的方式发送的，只能在同一个网段内广播。在使用 DHCP 中继功能时，需要保证 DHCP 服务器和客户端的路由可达。

11.3.2 网络拓扑

图 11-3 DHCP 中继组网图



SW1 和 Host 网段之间要保证路由可达，可在 SW1 配置一条到客户端 Host 网段的静态路由。

11.3.3 配置流程

- (1) 在 SW1 上创建 vlan-if2，配置 IP 地址为 192.168.0.1/24，添加 gige0_1 到 vlan-if2。
- (2) 在 SW1 使能 DHCP Server 并创建动态地址池，并配置静态路由。
- (3) 在 SW2 上创建 vlan-if2，配置 IP 地址为 192.168.0.2/24，添加 gige0_1 到 vlan-if2，创建 vlan-if3，配置 IP 地址为 192.168.1.1/24，添加 gige0_2 到 vlan-if3。
- (4) 在 SW2 使能 DHCP 中继功能。

(5) 验证配置。

11.3.4 配置步骤

(1) 在 SW1 上创建 vlan-if2，配置 IP 地址为 192.168.0.1/24，添加 gige0_1 到 vlan-if2。

```
<INSPUR>conf-mode
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_1
[INSPUR-vlan2]exit
[INSPUR]interface vlan-if2
[INSPUR-vlan-if2]ip address 192.168.0.1/24
[INSPUR-vlan-if2]exit
[INSPUR]
```

(2) 在 SW1 使能 DHCP Server 并创建动态地址池，并配置静态路由。

```
[INSPUR]dhcp server pool 192
[INSPUR-dhcp-pool-192]address range 192.168.1.20 192.168.0.250 24
[INSPUR-dhcp-pool-192]binding interface vlan-if2
[INSPUR-dhcp-pool-192]lease 1440
[INSPUR-dhcp-pool-192]default-router 192.168.2.1
[INSPUR-dhcp-pool-192]dns-server 172.153.0.1
[INSPUR-dhcp-pool-192]exit
[INSPUR]dhcp server enable
[INSPUR]
[INSPUR]ip route 192.168.1.0 255.255.255.0 192.168.0.2
```

(3) 在 SW2 上创建 vlan-if2，配置 IP 地址为 192.168.0.2/24，添加 gige0_1 到 vlan2，创建 vlan-if3，配置 IP 地址为 192.168.1.1/24，添加 gige0_2 到 vlan-if3。

```
<INSPUR>conf-mode
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_1
[INSPUR-vlan2]exit
[INSPUR]interface vlan-if2
[INSPUR-vlan-if2]ip address 192.168.0.2/24
[INSPUR-vlan-if2]exit
[INSPUR]vlan 3
[INSPUR-vlan3]port gige0_2
[INSPUR-vlan3]exit
[INSPUR]interface vlan-if3
```

```
[INSPUR-vlan-if3]ip address 192.168.1.1/24
[INSPUR-vlan-if3]exit
[INSPUR]
```

(4) 在 SW2 使能 DHCP 中继功能。

```
[INSPUR]interface vlan 3
[INSPUR-vlan-if3]dhcp relay server-address 192.168.0.1
[INSPUR-vlan-if3]exit
[INSPUR]dhcp relay enable
```

(5) 验证配置。

客户端主动发送地址请求报文，若能申请到对应的地址则说明 DHCP 功能生效。

11.4 DHCP Snooping 配置案例

11.4.1 配置需求

DHCP Snooping 是一种安全手段，当网络中存在非法的 DHCP 服务器时，客户端就可能获取到非法服务器的地址，导致网络不通。为了保证 DHCP 客户端能从合法的 DHCP 服务器上获取地址，用户可以在合法 DHCP 服务器和客户端之间的设备上启用 DHCP Snooping 功能，合法服务器和 DHCP Snooping 连接的端口设置为信任端口，其余端口均为非信任端口。当客户端获取到地址后，启用 DHCP snooping 的设备上能记录用户的 IP 地址和 MAC 地址的对应关系。

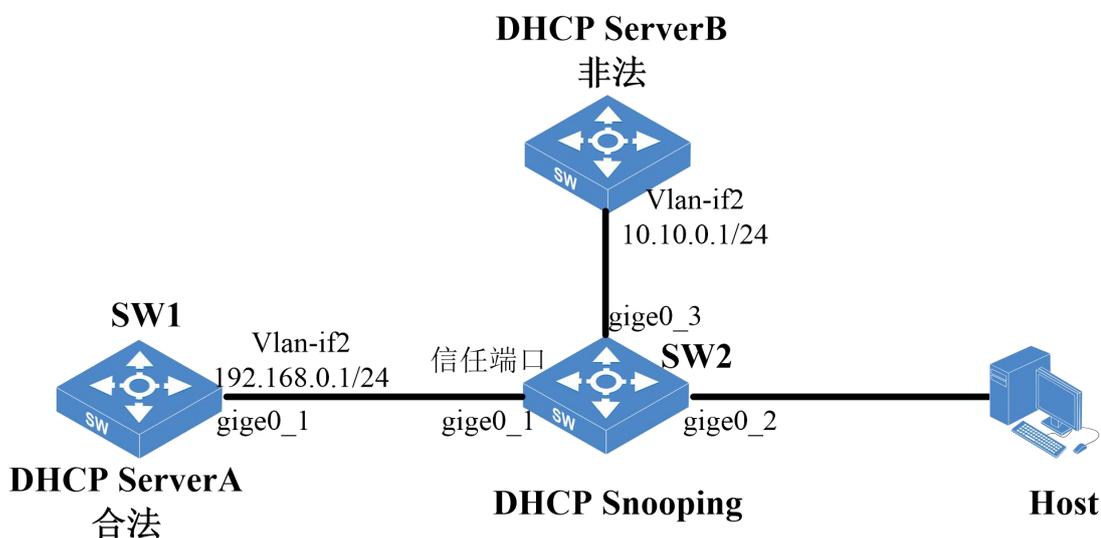


说明

信任端口能正常转发所有 DHCP 报文，非信任端口丢弃部分 DHCP 报文，以保证客户端能从合法服务器上获取 IP 地址。

11.4.2 网络拓扑

图 11-4 DHCP Snooping 组网图



11.4.3 配置流程

- (1) 在 SW1 上创建 vlan-if2，配置 IP 地址为 192.168.0.1/24，添加 gige0_1 到 vlan-if2。
- (2) 在 SW1 使能 DHCP Server 并创建动态地址池。
- (3) 在 SW2 上创建 vlan2，添加 gige0_1、gige0_2、gige0_3 到 vlan2。
- (4) 在 SW2 上启用 DHCP Snooping 功能，配置 gige0_1 端口为信任端口，其余端口为非信任端口，并启用记录 IP MAC 地址功能。
- (5) 验证配置。

11.4.4 配置步骤

- (1) 在 SW1 上创建 vlan-if2，配置 IP 地址为 192.168.0.1/24，添加 gige0_1 到 vlan-if2。

```
<INSPUR>conf-mode
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_1
[INSPUR-vlan2]exit
[INSPUR]interface vlan-if2
[INSPUR-vlan-if2]ip address 192.168.0.1/24
[INSPUR-vlan-if2]exit
[INSPUR]
```

- (2) 在 SW1 使能 DHCP Server 并创建动态地址池。

```
<INSPUR>conf-mode
[INSPUR]dhcp server pool 192
[INSPUR-dhcp-pool-192]address range 192.168.0.10 192.168.0.100 24
[INSPUR-dhcp-pool-192]binding interface vlan-if2
[INSPUR-dhcp-pool-192]lease 1440
[INSPUR-dhcp-pool-192]default-router 192.168.0.1
[INSPUR-dhcp-pool-192]dns-server 172.153.0.1
[INSPUR-dhcp-pool-192]exit
[INSPUR]dhcp server enable
[INSPUR]
```

- (3) 在 SW2 上创建 vlan2，添加 gige0_1、gige0_2、gige0_3 到 vlan2。

```
<INSPUR>conf-mode
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_1
[INSPUR-vlan2]port gige0_2
[INSPUR-vlan2]port gige0_3
```

- (4) 在 SW2 上使能 DHCP Snooping 功能，设置 gige0_1 端口为信任端口，其余端口为非信任端口。

```
<INSPUR>conf-mode
[INSPUR]interface gige0_1
[INSPUR-gige0_1]dhcp snooping trust
[INSPUR]dhcp snooping enable
```

- (5) 验证配置

客户端获取地址后查看 DHCP Snooping 信息列表, 记录了获取地址的主机 MAC 地址和 IP 地址的对应关系。

```
<INSPUR>show dhcp snooping
      Dhcp Snooping information
-----
Port_name          Macaddr           Ipaddr
-----
gige0_2           00:24:ac:13:14:02 192.168.0.10
```

12 QoS 典型配置案例

12.1 QoS 简介

QoS (Quality of Service, 服务质量) 是网络的一种安全机制, 是用来解决网络延迟和阻塞等问题的一种技术。网络带宽总是有限的, 只要存在抢夺网络带宽的情况, 就会出现服务质量的要求。QoS 可以保障业务带宽的优先级最高, 使其数据优先转发。

我司的交换机上, QoS 信任模式支持以下四种:

表 12-1 QoS 信任模式

项目	说明
信任端口优先级	根据报文进入的端口进行优先级映射
信任 COS 优先级	根据报文携带的 COS 优先级查找映射表进行优先级映射
信任 DSCP 优先级	根据报文携带的 DSCP 优先级查找映射表进行优先级映射
信任 IP 优先级	根据报文携带的 IP 优先级查找映射表进行优先级映射

当设备的多个端口收到不同的流量时, 可以通过配置端口的优先级来实现某些端口的流量优先转发。当设备收到的流量携带 COS、DSCP、IP 优先级时, 可在设备上配置优先级和 COS 队列的映射关系, 实现流量的先后转发。交换机上共有 8 个 COS 队列, 分别用 0-7 表示, COS 数值大的队列, 优先级越高。流量从设备转发出去时, 出端口可以通过配置队列调度模式及队列权重值来调整各个队列里的流量转发先后顺序及占用带宽比例。队列调度模式有 SP、WRR、WDRR 三种模式, 在 WRR 和 WDRR 模式里需要配置不同的 COS 队列的转发比例, 即权重值。

表 12-2 队列调度模式

项目	说明
SP 模式	绝对模式, 从优先级高到低的队列排队转发, 保证优先级高的队列先转发, 直到带宽占满, 超出带宽后, 优先级低的报文将被丢包。

项目	说明
WRR 模式	轮询模式，通过配置不同 COS 队列的权重值，使不同队列按比例转发流量
WDRR 模式	流量转发机制与 WRR 模式类似

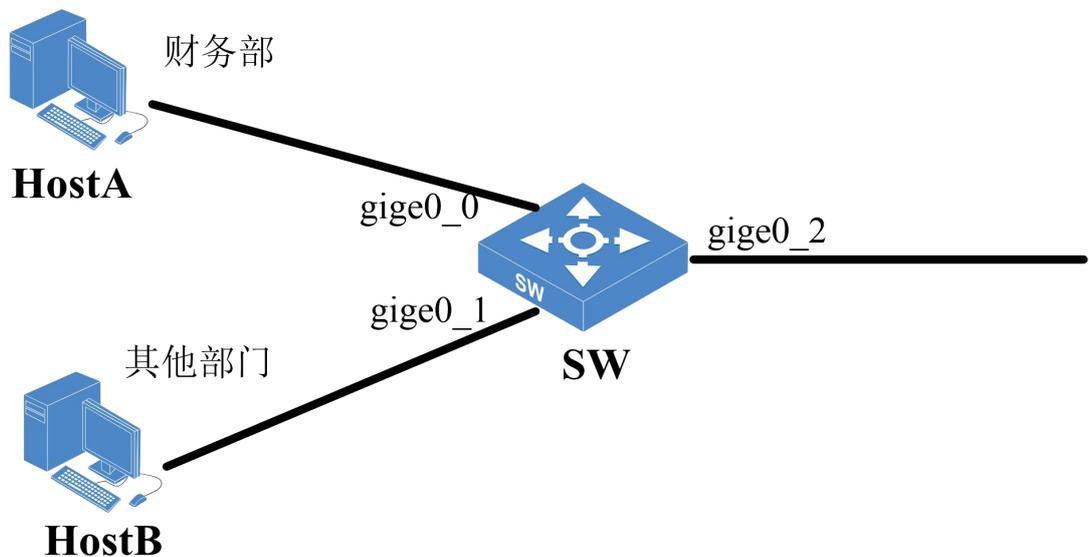
12.2 配置案例

12.2.1 配置需求

某公司内网络带宽分配不均，导致财务部门在传输数据时经常断掉，无法使其正常工作。现要求交换机配置功能使其财务部传输数据优先级最高，保障正常工作。

12.2.2 网络拓扑

图 12-2 QoS 组网图



12.2.3 配置流程

- (1) 在 SW 上把端口 gige0_0 和 gige0_1 分别加入队列 7 和 1。
- (2) 在 SW 上端口 gige0_2 配置 QoS 使用 WRR 模式，配置队列 7 的权重值为 7，配置队列 1 的权重值为 3。

(3) 验证配置。

12.2.4 配置步骤

(1) 在 SW 上把端口 gige0_0 和 gige0_1 分别加入队列 7 和 1。

```
[INSPUR]interface gige 0_0
[INSPUR-gige0_0]qos trust port
[INSPUR-gige0_0]qos map port-cos 7
[INSPUR-gige0_0]interface gige 0_1
[INSPUR-gige0_1]qos trust port
[INSPUR-gige0_1]qos map port-cos 1
[INSPUR-gige0_1]
```

(2) 在 SW 上端口 gige0_2 配置 QoS 使用 WRR 模式，配置队列 7 的权重值为 7，配置队列 1 的权重值为 3。

```
[INSPUR]interface gige0_2
[INSPUR-gige0_2]qos scheduler wrr
[INSPUR-gige0_2]qos wrr queue 7 weight 7
[INSPUR-gige0_2]qos wrr queue 1 weight 3
[INSPUR]
```

(3) 验证配置。

可在 gige0_0 和 gige0_1 下满带宽的传输数据，查看带宽使用状况应该为财务部流量转发 70%带宽数据转发，其他部门 30%带宽数据转发。

13 ACL 典型配置案例

13.1 ACL 简介

ACL（Access Control List，访问控制列表）即通过配置对报文的匹配条件和处理动作而实现报文过滤和控制的功能。当网络设备的物理接口接收到报文，或者将报文从物理接口发送出去之前，根据当前接口上配置的 ACL 表项对报文内容进行解析和匹配，对能够匹配到的报文进行相应动作处理。

一条 ACL 不可能同时支持所有的匹配条件，若将 ACL 按照匹配条件进行分类，每一类作为一种 ACL 模式，支持若干关系密切的几种匹配条件：

- **IPv4 模式：**支持匹配 IPv4 报文的源 IP 地址、目的 IP 地址、IP 协议号、源端口号、目的端口号、IP 优先级、ToS/DSCP 优先级、物理端口。
- **IPv6 模式：**支持匹配 IPv6 报文的源 IPv6 地址、目的 IPv6 地址、IP 协议号、源端口号、目的端口号、IP 优先级、ToS/DSCP 优先级、物理端口。
- **MAC 模式：**支持匹配任意报文的源 MAC 地址、目的 MAC 地址、以太网类型、VLAN ID、802.1p 优先级、物理端口。
- **MAC/IPv4 绑定模式：**支持匹配 IPv4 报文的源 MAC 地址、源 IP 地址、以太网类型、VLAN ID、802.1p 优先级、物理端口。
- **MAC/IPv6 绑定模式：**支持匹配 IPv6 报文的源 MAC 地址、源 IPv6 地址、以太网类型、VLAN ID、802.1p 优先级、物理端口。

目前我司支持的 ACL 匹配条件大致如下（不同款型可能会有差异，请参考规格表实际支持情况）：

ACL 匹配条件	意义	匹配方法
源 MAC/掩码	匹配报文的源 MAC 地址	报文源 MAC 和掩码进行按位与操作得 MAC1，ACL 源 MAC 与掩码进行按位与得 MAC2，若 MAC1 与 MAC2 相同则为匹配
目的 MAC/掩码	匹配报文的目的 MAC 地址	报文目的 MAC 和掩码进行按位与操作得 MAC1，ACL 目的 MAC 与掩码进行按位与得 MAC2，若 MAC1 与 MAC2 相同则为匹配

ACL 匹配条件	意义	匹配方法
源 IPv4/掩码	匹配报文的源 IP 地址	报文源 IP 和掩码进行按位与操作得 IP1, ACL 源 IP 与掩码进行按位与得 IP2, 若 IP1 与 IP2 相同则为匹配
目的 IPv4/掩码	匹配报文的的目的 IP 地址	报文目的 IP 和掩码进行按位与操作得 IP1, ACL 目的 IP 与掩码进行按位与得 IP2, 若 IP1 与 IP2 相同则为匹配
源 IPv6/掩码	匹配报文的源 IPv6 地址	报文源 IPv6 和掩码进行按位与操作得 ip1, ACL 源 IPv6 与掩码进行按位与得 ip2, 若 ip1 与 ip2 相同则为匹配
目的 IPv6/掩码	匹配报文的的目的 IPv6 地址	报文目的 IPv6 和掩码进行按位与操作得 ip1, ACL 目的 IPv6 与掩码进行按位与得 ip2, 若 ip1 与 ip2 相同则为匹配
IP 协议号	匹配报文的 IP 协议号	报文 IP 协议号等于 IP 协议号则为匹配
四层源端口号	匹配 IP 报文的四层源端口号	报文的四层源端口号大于等于 ACL 四层源端口号最小值且小于等于四层源端口号最大值则为匹配
四层目的端口号	匹配 IP 报文的四层目的端口号	报文的四层目的端口号大于等于 ACL 四层目的端口号最小值且小于等于四层目的端口号最大值则为匹配
VLAN ID	匹配报文外层 VLAN 标签中的 VLAN ID。只有一层 VLAN 标签时, 即检查该层 VLAN 标签	报文的外层 VLAN 标签中的 VLAN ID 大于等于 ACL 的 VLAN ID 最小值且小于等于 ACL 的 VLAN ID 最大值则为匹配。
TOS/DSCP 优先级	匹配报文 ToS/DSCP 优先级	报文的 ToS/DSCP 优先级等于 ACL ToS/DSCP 优先级则为匹配
以太网类型	检查报文以太网类型	报文以太网类型等于 ACL 以太网类型则为匹配
物理端口	匹配报文入端口或出端口	报文入端口(VLANACL 及入方向 ACL)或出端口(出方向 ACL) 属于 ACL 物理端口中的一个则为匹配。只能选择直连在本交换芯片上的物理口。



说明

- 如果某种场景需要较多某模式的 ACL 资源, 则可通过 ACL 硬件资源调配功能调整片区(slice)模式, 将不在使用的片区改为所需模式。例如: 当设备主要用于 MAC/IPv4 绑定功能时, 需要较多 MAC/IPv4 绑定模式的 ACL 资源, 此时可将多余的 IPv4 模式的片区调整为 MAC/IPv4 绑定模式, 即 `acl resource slot xx ingress/ egress slice xx mode (mac | ipv4 | ipv6 | mac-ipv4 | mac-ipv6 | portal | portal-ipv6 | rt-policy | ipv4-vlan | ipv6-vlan)` 进行调整。

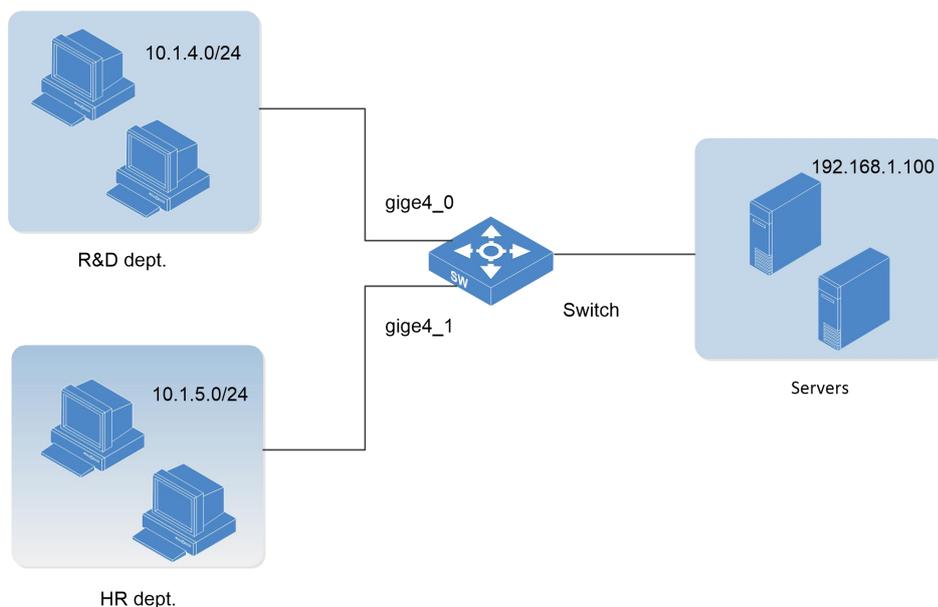
13.2 IPv4 ACL 典型配置案例

13.2.1 配置需求

某公司内部网络现要求对服务器的访问做以下限制：无论任何时间，只有人力资源部的某台电脑才能够访问公司文档资源 Server 服务器，其他部门主机不能访问此资源 Server 服务器，防止敏感材料外泄风险。

13.2.2 网络拓扑

图 13-1 IPv4 ACL 过滤指定流量组网图



13.2.3 配置流程

- (1) 在 Switch 上把和研发部主机相连的交换机端口 `gige4_0` 下配置一条入方向 IPv4 ACL。
- (2) 匹配研发部主机的 IP 网段（`10.1.4.0/24`）和服务器的 IP 地址 `192.168.1.100`，对匹配的报文进行过滤丢弃。
- (3) 验证配置。

13.2.4 配置步骤

(1) 在 Switch 上进入入方向 ACL 的 IPv4 ACL 配置模式。

```
<INSPUR>conf-mode
[INSPUR]acl mode ipv4 ingress
[INSPUR-acl-ipv4-ingress]
```

(2) 策略匹配物理端口 gige4_0、源 IPv4 网段 (10.1.4.0/24)、目的 IPv4 地址 192.168.1.100，对匹配的报文进行过滤。

```
[INSPUR-acl-ipv4-ingress]
[INSPUR-acl-ipv4-ingress]rule T1 source 10.1.4.0/24 destination host
192.168.1.100 interface gige4_0 action drop
[INSPUR-acl-ipv4-ingress]exit
[INSPUR]
```

(3) 在 Switch 上执行 **show acl mode ipv4 ingress rule T1** 命令查看 ACL 策略状态

```
[INSPUR] show acl mode ipv4 ingress rule T1
-----Rule T1's priority is 1 and takes 1 resource(s).

  Source IPv4:
      10.1.4.0/24.

  Destination IPv4:
      host 192.168.1.100.

  In ports: gige4_0.

  Action:
      Drop if matched.

[INSPUR]
```

那么在任时间段内，研发部都无法访问服务器 192.168.1.100，而人力资源部则不受任何影响。

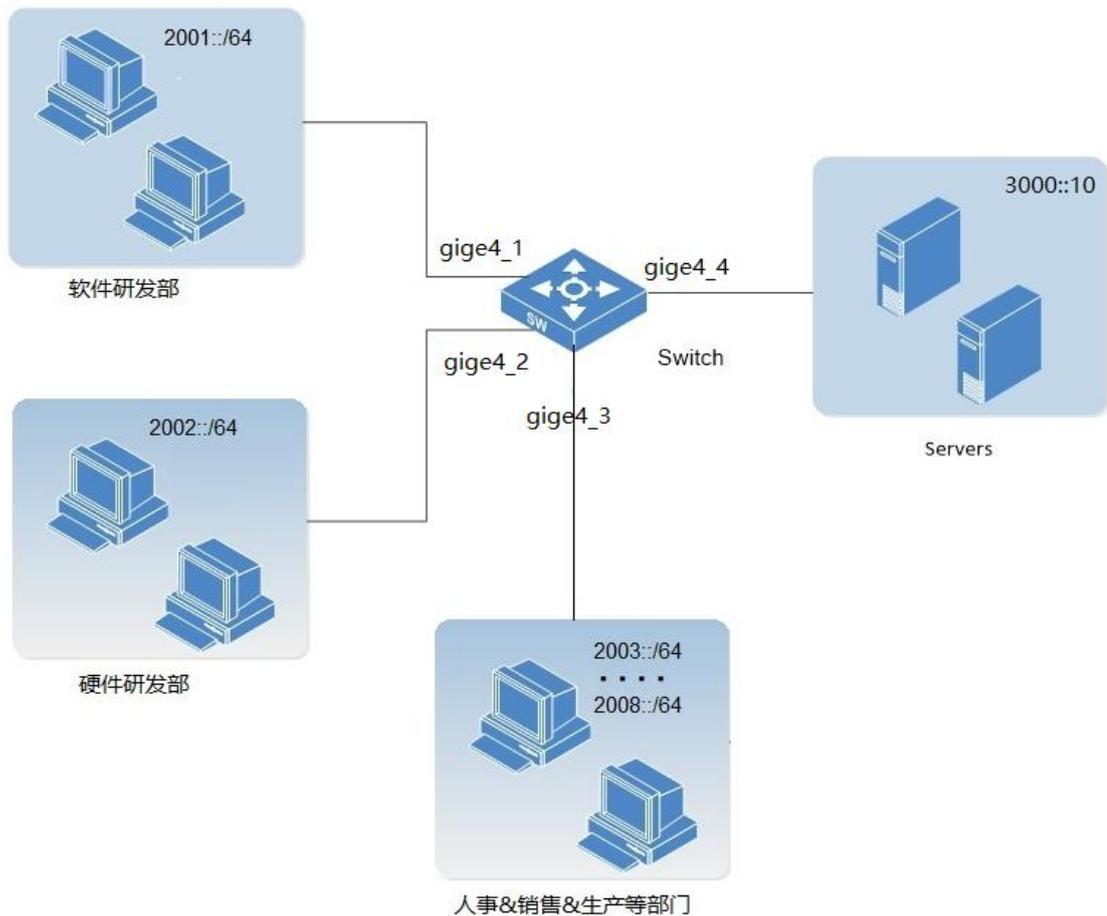
13.3 IPv6 ACL 典型配置案例

13.3.1 配置需求

某公司内部网络要求对代码服务器的访问做以下限制：无论任何时间，只有研发部门能够访问公司代码服务器，其他部门主机不能访问代码服务器，防止敏感信息外泄。

13.3.2 网络拓扑

图 13-2 IPv6 ACL 过滤指定流量组网图



13.3.3 配置流程

- (1) 查看设备配置端口的槽位是否已分配 slice 资源给 IPv6 ACL，若没有，则需将 Switch 的 4 槽部分入方向 ACL 的 slice 资源分配给 IPv6 ACL。
- (2) 在 Switch 上把和除研发部门外与交换机相连的端口 gige4_3 下配置一条入方向 IPv6 ACL。
- (3) 匹配部门的 IP 网段（2003::/64~2008::/64）和服务器的 IP 地址 3000::10，对匹配的报文进行过滤丢弃。
- (4) 验证配置。

13.3.4 配置步骤

- (1) 将 Switch 的 4 槽的入方向 ACL 资源中第 3 个 slice 资源分配给 IPv6 ACL。

将 Switch 的 4 槽的 slice3 分配给 IPv6 ACL（框式交换机每个槽位可配置的 ACL 数量都是有限的，用 slice 来定义资源片区。比如单槽位入方向 ACL 的总资源数分为多个 slice，一个 slice 代表一定的入方向 ACL 资源数，一般为 128 条、256 条等。若某种模式的入方向 ACL 没有分配到一个 slice，则无法进行配置）

```
<INSPUR>conf-mode
[INSPUR]acl resource slot 4 ingress slice 3 mode ipv6
```

- (2) 在 Switch 上进入入方向 ACL 的 IPv6 ACL 配置模式。

```
<INSPUR>conf-mode
[INSPUR]acl mode ipv6 ingress
[INSPUR-acl-ipv6-ingress]
```

- (3) 策略匹配物理端口 gige4_3、源网段（2003::/64~2008::/64）、目的 IPv6 地址 3000::10，对匹配的报文进行过滤。

```
[INSPUR-acl-ipv6-ingress]
[INSPUR-acl-ipv6-ingress] rule R1 source 2003::/64 2004::/64 2005::/64
2006::/64 2007::/64 2008::/64 destination host 3000::10 interface gige4_3
action drop
[INSPUR-acl-ipv6-ingress]exit
[INSPUR]
```

- (4) 在 Switch 上执行 **show acl mode ipv6 ingress rule R1** 命令查看 ACL 策略状态

```
[INSPUR]show acl mode ipv6 ingress rule R1
-----Rule R1's priority is 3 and takes 1 resource(s).
  Source IPv6:
    IPv6/prefix: 2003::/64.
    IPv6/prefix: 2004::/64.
    IPv6/prefix: 2005::/64.
    IPv6/prefix: 2006::/64.
    IPv6/prefix: 2007::/64.
    IPv6/prefix: 2008::/64.
  Destination IPv6:
    Host IPv6: 3000::10.
  In ports: gige4_3.
```

```
Action:  
Drop if matched.  
[INSPUR]
```

那么在任何时间段内，除了研发部其他部门都无法访问代码服务器 3000::10，而研发部则不受任何影响。

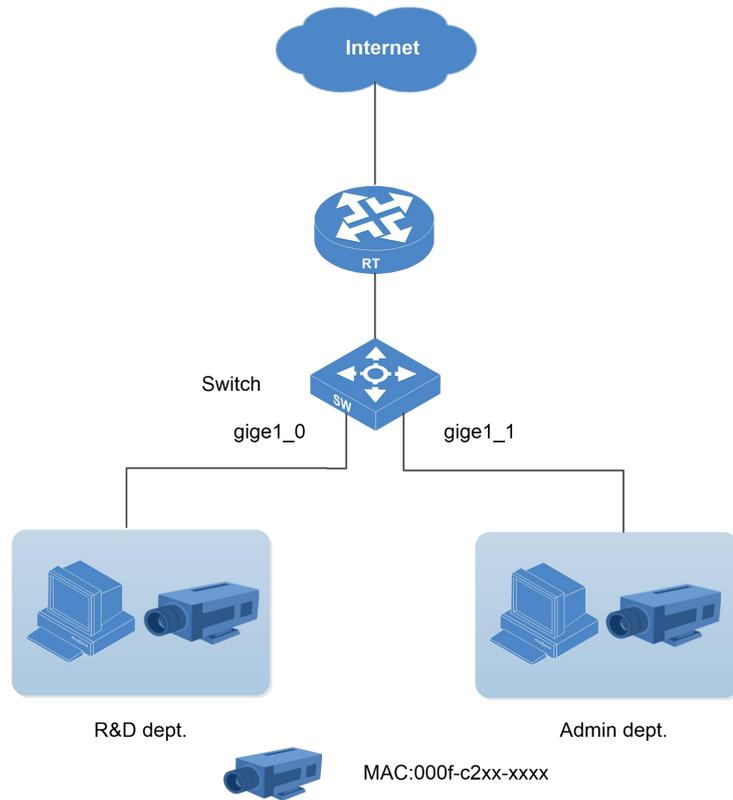
13.4 MAC ACL 典型配置案例

13.4.1 配置需求

某公司研发部和管理部均部署了网络视频设备，这些视频设备的 MAC 地址为 000f-c2xx-xxxx，现要求限制这些设备不能在每天的 8:30 到 18:00 的上班时间段向外网发送数据。

13.4.2 网络拓扑

图 13-3 通过配置入方向 ACL 的 MAC ACL 过滤指定的流量



13.4.3 配置流程

在受限设备的 IP 地址不定时，可以通过 MAC 地址来进行匹配；而对于具有相同 MAC 地址前缀的多台设备，也可以通过 MAC 地址掩码的方式来进行同时匹配

- (1) 查看设备配置端口的槽位是否已分配 slice 资源给 MAC ACL，若没有，则需将 Switch 的 1 槽部分入方向 ACL 的 slice 资源分配给 MAC ACL。
- (2) 在 1 槽的两个接口 gige1_0 和 gige1_1 下配置入方向 ACL，匹配设备的 mac 地址进行过滤，使用 MAC ACL 实现。
- (3) 验证配置。

13.4.4 配置步骤

- (1) 将 Switch 的 1 槽的部分入方向 ACL 资源分配给 MAC ACL。

将 Switch 的 1 槽的 slice3 分配给 MAC ACL（框式交换机每个槽位可配置的 ACL 数量都是有限的，用 slice 来定义资源片区。比如单槽位入方向 ACL 的总资源数分为多个 slice，一个 slice 代表一定的入方向 ACL 资源数，一般为 128 条、256 条等。若某种模式的入方向 ACL 没有分配到一个 slice，则无法进行配置）

```
<INSPUR>conf-mode
[INSPUR]acl resource slot 1 ingress slice 3 mode mac
```

- (2) 进入 1 槽的入方向 ACL 的 MAC ACL 配置模式，在 1 槽的两个接口 gige1_0 和 gige1_1 下配置一条 MAC ACL，匹配设备的 mac 地址 000f-c2xx-xxxx 进行过滤，并设置 ACL 生效时间段为每天的 8:30 到 18:00。

```
[INSPUR]acl mode mac ingress
[INSPUR-acl-mac-ingress]rule R1 source 000F-C200-0000/FFFF-FF00-0000
interface gige1_0,gige1_1 action drop time-range every-week mon tues wed thur
fri sat sun time-begin 08:30 time-end 18:00
[INSPUR-acl-mac-ingress]exit
[INSPUR]
```

- (3) 在 Switch 上执行 **show acl mode mac slot 1 ingress all** 命令查看 ACL 策略状态

```
[INSPUR] show acl mode mac slot 1 ingress all
There are 1 ACL rules of mode mac ingress in slot 1
-----Rule 11's priority is 1 and takes 1 resource(s).
  Source MAC/mask: 00:0F:C2:00:00:00/FF:FF:FF:00:00:00.
  In ports: gige1_0,gige1_1.
  Action:
  Drop if matched.
  time range: monday tuesday wednesday thursday friday saturday sunday from
08:30 to 18:00
[INSPUR]
```

那么每天的 8:30 到 18:00 时间段，视频设备不可以与外网中的设备进行通信；在其它时间段视频设备可以与外网的设备进行通信。

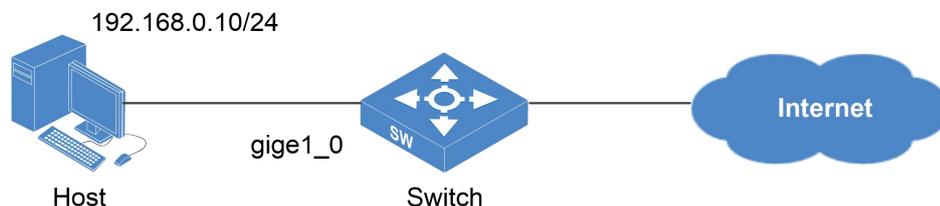
13.5 MAC/IPv4 绑定 ACL 典型配置案例

13.5.1 配置需求

MAC/IPv4 绑定是指通过配置 MAC/IPv4 绑定 ACL 将用户的主机 MAC 地址和 IPv4 地址与所连接的交换机端口进行绑定，过滤不匹配绑定项的报文，达到管理用户的目的。假定公司想要某些员工使用固定的 IPv4 地址，不能随意修改，若随意修改地址后，该员工就不能访问服务器、外网等资源，可通过配置 MAC/IPv4 绑定 ACL 来实现。配置后，如果该端口收到的报文匹配绑定项则转发报文，否则将丢弃报文，以此过滤掉不合法的用户报文。这样可以防止用户随意修改主机 IPv4 地址，方便对用户进行管理。

13.5.2 网络拓扑

图 13-4 MAC/IPv4 绑定 ACL 组网图



13.5.3 配置流程

主机 MAC 地址为 00:10:01:B1:C2:D3，固定 IPv4 地址为 192.168.0.10/24。

- (1) 查看设备配置端口的槽位是否已分配 slice 资源给 MAC/IPv4 ACL，若没有，则需将 Switch 的 1 槽部分入方向 ACL 的 slice 资源分配给 MAC/IPv4 ACL。
- (2) 在和主机相连的接入交换机的 gige1_0 接口下配置主机 MAC 地址和 IPv4 地址绑定 ACL。匹配该绑定策略的报文进行转发，否则进行丢包。

13.5.4 配置步骤

- (1) 将 Switch 的 1 槽的部分入方向 ACL 资源分配给 MAC/IPv4 ACL。

将 Switch 的 1 槽的 slice3 分配给 MAC/IPv4 ACL（框式交换机每个槽位可配置的 ACL 数量都是有限的，用 slice 来定义资源片区。比如单槽位入方向 ACL 的总资源数分为多个 slice，一个 slice 代表一定的入方向 ACL 资源数，一般为 128 条、256 条等。若某种模式的入方向 ACL 没有分配到一个 slice，则无法进行配置）

```
<INSPUR>conf-mode
[INSPUR]acl resource slot 1 ingress slice 3 mode mac-ipv4
```

- (2) 在 Switch 的 gige1_0 下配置一条 MAC/IPv4 绑定的入方向 ACL，匹配主机的 MAC 地址 00:10:01:B1:C2:D3，IPv4 地址 192.168.0.10/24，动作为通过。

```
<INSPUR>conf-mode
[INSPUR]acl mode mac-ipv4 ingress
[INSPUR-acl-mac-ipv4-ingress]rule R1 source-mac 0010-01B1-C2D3 source-ipv4
192.168.0.10 interface gige1_0 action permit
```

- (3) 在 gige1_0 下配置第二条 MAC/IPv4 绑定的入方向 ACL，匹配所有 MAC 地址和 IPv4 地址，动作为丢包。

```
[INSPUR-acl-mac-ipv4-ingress]rule R2 interface gige1_0 action drop
[INSPUR-acl-mac-ipv4-ingress]exit
[INSPUR]
```

- (4) 在 Switch 上执行 **show acl mode mac-ipv4 slot 1 ingress all** 命令查看 ACL 策略状态

```
[INSPUR]show acl mode mac-ipv4 slot 1 ingress all
There are 2 ACL rules of mode mac-ipv4 ingress in slot 1
-----Rule R1's priority is 1 and takes 1 resource(s).
  Source MAC/mask: 00:10:01:B1:C2:D3/FF:FF:FF:FF:FF:FF.
  MAC/IPv4 Source IPv4: 192.168.0.10.
  In ports: gige0_0.
  Action:
  Permit if matched.
-----Rule R2's priority is 2 and takes 1 resource(s).
  In ports: gige0_0.
  Action:
  Drop if matched.
[INSPUR]
```

那么在接口 gige1_0 的接入电脑只有当 MAC 地址为 00:10:01:B1:C2:D3、IPv4 地址为 192.168.0.10 时才可访问网络；当修改 IP 地址或改变与 SW 相连的接口时，不能访问网络。

14 生成树典型配置案例

14.1 生成树简介

生成树协议 (Spanning Tree Protocol) 是一种二层网络的防环协议, 通过与其他交换机交换 BPDU 消息来检测环路, 然后删除环路。当生成树协议感知到网络中存在环路时, 它会在环路上选择一个恰当的位置阻塞链路中的端口, 以阻止端口接收和转发报文, 通过这种方式消除环路上可能产生的广播风暴。生成树协议根据网络中的拓扑结构, 将网络中的节点按照一定的算法生成一个树形的拓扑结构, 从而避免网络中环路的存在。当网络中拓扑结构发生变化时, 生成树算法会根据新的网络拓扑重新计算树, 生成新的树形结构, 这样提供了环路保护的功能。

生成树的工作主要分为三大部分: 选举过程、拓扑计算、确定端口行为。选举出根桥后, 在根桥的统一指挥下进行树形拓扑计算, 以根桥作为树根向外延伸, 树形拓扑计算完后端口角色就已经确定了。其中根端口和指定端口参与报文转发, 阻塞的端口不转发报文。

图 14-1 生成树协议

项目	说明
根桥	由选举产生或者手动指定, 用于指挥整个网络设备的工作, 是生成的树形结构的树根。
根端口	非跟桥设备到根桥最优配置的一个端口。
指定端口	根桥上的端口全都是指定端口, 非根桥上除了根端口以外的转发数据的端口都是指定端口。

快速生成树协议 RSTP (Rapid Spanning Tree Protocol) 针对 STP 协议收敛时间太长, IEEE 定义了 802.1w RSTP 协议。该协议通过引入边缘端口, 替换端口, 备份端口等概念, 使得端口状态的改变在某些情况下可以快速进行切换, 从而实现生成树的快速收敛。

图 14-2 快速生成树协议

项目	说明
边缘端口	根据实际需要配置的一种指定端口,用于连接 PC 和不需要运行 STP 的下游交换机,当边缘端口上开启 BPDU 保护功能时,收到 BPDU 报文后就会自动关闭该端口
替换端口	是根端口的备份端口,当根端口发生故障时它可以迅速代替成为新的根端口,进入转发状态
备份端口	指被本交换机抑制的端口

MSTP 协议是一个多生成树 (Multi Spanning Tree Protocol) 协议,相对 RSTP 来说,主要是引入了实例和域的概念。域的概念是为了将网络中具有不同配置的网段进行分割开,在网段内部实行统一的配置,可以在域内进行独立的生成树构造。域之间则使用一个单一生成树将所有的域连接起来(该生成树被称为 CST,公共生成树),确保全链接和无环。在域的内部可以构造多个生成树实例,同时可以将不同的 VLAN 映射到不同的生成树实例上。在每个域的内部都有一个实例 ID 为 0 的实例,该实例与 CST 共同组成了 CIST(公共内部生成树)。该生成树将整个网络中的域和域内部的桥设备和网段连成一个全接无环的树。

图 14-3 多生成树协议

项目	说明
实例	是单个或者多个 VLAN 的一种集合
域	是由交换机网络中的多台设备及它们之间的网段构成

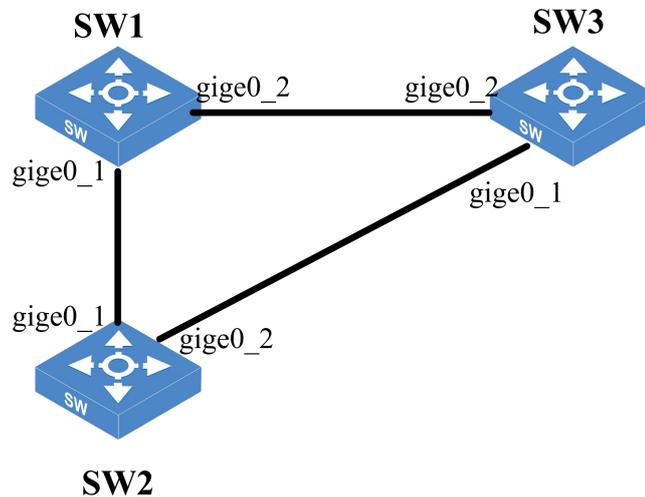
14.2 STP 配置案例

14.2.1 配置需求

当网络比较复杂时,用户不能确定网络中是否存在环路,二层网络里所有通过广播方式转发的报文都可能在环路上产生风暴,一旦环路上发生广播风暴就几乎不会自己停下来,除非人为干预使环路消失。在二层网络中启用生成树协议,按照一定的算法阻塞环路上的端口,使环路消失,解决风暴隐患。

14.2.2 网络拓扑

图 14-4 STP 组网图



14.2.3 配置流程

- (1) 在 SW1 上创建 vlan2，添加 gige0_1、gige0_2 到 vlan2，开启 STP 功能，配置网桥优先级为 0（最高优先级），添加启用 STP 的端口 gige0_1、gige0_2。
- (2) 分别在 SW2 和 SW3 上创建 vlan2，添加 gige0_1、gige0_2 到 vlan2，开启 STP 功能，配置网桥优先级为 4096，添加启用 STP 的端口 gige0_1、gige0_2。
- (3) 验证配置。

14.2.4 配置步骤

- (1) 在 SW1 上创建 vlan2，添加 gige0_1、gige0_2 到 vlan2，开启 STP 功能，配置网桥优先级为 0（最高优先级），添加启用 STP 的端口 gige0_1、gige0_2。

```
<INSPUR>conf-mode
[INSPUR]
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_1
[INSPUR-vlan2]port gige0_2
[INSPUR-vlan2]exit
```

```
[INSPUR]spanning-tree enable
[INSPUR]spanning-tree mode stp
[INSPUR] spanning-tree bridge-priority 0
[INSPUR]interface gige 0_1
[INSPUR-gige0_1]
[INSPUR-gige0_1]spanning-tree enable
[INSPUR-gige0_1]exit
[INSPUR]interface gige0_2
[INSPUR-gige0_2]spanning-tree enable
```

- (2) 分别在 SW2 和 SW3 上创建 vlan2，添加 gige0_1、gige0_2 到 vlan2，开启 STP 功能，配置网桥优先级为 4096 添加启用 STP 的端口 gige0_1、gige0_2。

```
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_1
[INSPUR-vlan2]port gige0_2
[INSPUR-vlan2]exit
[INSPUR]spanning-tree enable
[INSPUR]spanning-tree mode stp
[INSPUR] spanning-tree bridge-priority 4096
[INSPUR]interface gige0_1
[INSPUR-gige0_1]
[INSPUR-gige0_1]spanning-tree enable
[INSPUR-gige0_1]exit
[INSPUR]interface gige0_2
[INSPUR-gige0_2]spanning-tree enable
```

- (3) 验证配置

查看 STP 结果。

SW1 状态

```
<INSPUR>show spanning-tree
MST0
Spanning tree enabled protocol STP
Root ID Priority 0
Address 00:24:AC:71:AD:85
This bridge is the root
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 0
Address 00:24:AC:71:AD:85
```

```
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Configuration Digest Oxac36177f 50283cd4 b83821d8 ab26de62

Interface Role Sts Cost Prio P2p
-----
gige0_1 designated forwarding 20000 128 yes
gige0_2 designated forwarding 20000 128 yes

<INSPUR>
```

SW2 上的状态

```
<INSPUR>show spanning-tree
MST0
Spanning tree enabled protocol STP
Root ID Priority 0
Address 00:24:AC:71:AD:85
Cost 20000
Port gige0_1
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 4096
Address 00:24:AC:71:AD:86
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Configuration Digest Oxac36177f 50283cd4 b83821d8 ab26de62

Interface Role Sts Cost Prio P2p
-----
gige0_1 root forwarding 20000 128 yes
gige0_2 alternate blocking 20000 128 yes

<INSPUR>
```

SW3 上的状态

```
<INSPUR>show spanning-tree
MST0
Spanning tree enabled protocol STP
Root ID Priority 0
Address 00:24:AC:71:AD:85
Cost 20000
Port gige0_1
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
```

```

Bridge ID Priority 4096
Address 00:24:AC:71:AD:88
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Configuration Digest Oxac36177f 50283cd4 b83821d8 ab26de62

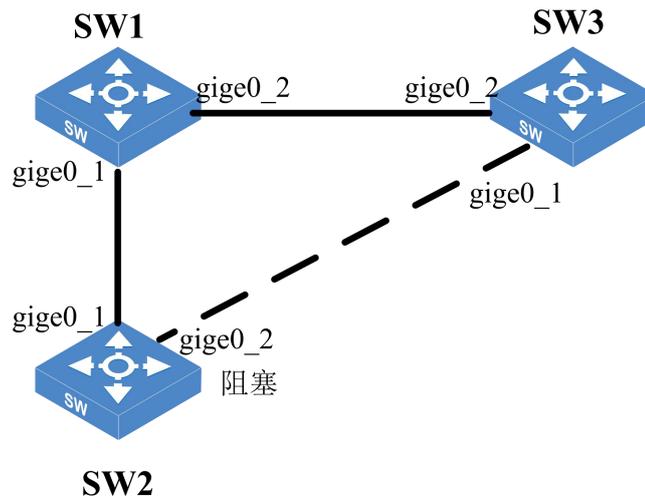
Interface Role Sts Cost Prio P2p
-----
gige0_1 designated forwarding 20000 128 yes
gige0_2 root forwarding 20000 128 yes

<INSPUR>

```

以上信息表明生成树协议阻塞了 SW2 上的 gige0_2 口，使原来的环路变成一个树形结构，如图。

图 14-5 STP 树形图



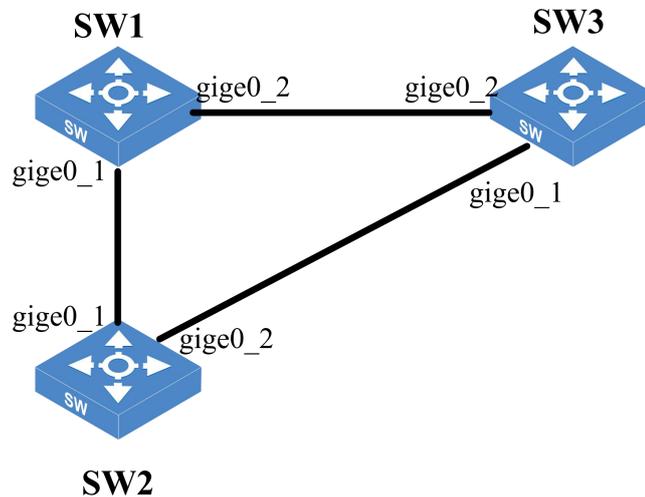
14.3 RSTP 配置案例

14.3.1 配置需求

用户的网络中可能存在环路，一旦环路上发生广播风暴就几乎不会自己停下来，RSTP 与 STP 协议都能按照一定的算法阻塞环路上的端口使环路消失，且 RSTP 的收敛时间更短，当用户对收敛时间要求较高时使用 RSTP 更好。

14.3.2 网络拓扑

图 14-6 RSTP 组网图



14.3.3 配置流程

- (1) 在 SW1 上创建 vlan2，添加 gige0_1、gige0_2 到 vlan2，开启 RSTP 功能，配置网桥优先级为 0（最高优先级），添加启用 RSTP 的端口 gige0_1、gige0_2。
- (2) 分别在 SW2 和 SW3 上创建 vlan2，添加 gige0_1、gige0_2 到 vlan2，开启 RSTP 功能，配置网桥优先级为 4096 添加启用 RSTP 的端口 gige0_1、gige0_2。
- (3) 验证配置。

14.3.4 配置步骤

- (1) 在 SW1 上创建 vlan2，添加 gige0_1、gige0_2 到 vlan2，开启 RSTP 功能，配置网桥优先级为 0（最高优先级），添加启用 RSTP 的端口 gige0_1、gige0_2。

```
<INSPUR>conf-mode
[INSPUR]
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_1
[INSPUR-vlan2]port gige0_2
[INSPUR-vlan2]exit
```

```
[INSPUR]spanning-tree enable
[INSPUR]spanning-tree mode rstp
[INSPUR] spanning-tree bridge-priority 0
[INSPUR]interface gige0_1
[INSPUR-gige0_1]
[INSPUR-gige0_1]spanning-tree enable
[INSPUR-gige0_1]exit
[INSPUR]interface gige 0_2
[INSPUR-gige0_2]spanning-tree enable
```

- (2) 分别在 SW2 和 SW3 上创建 vlan2，添加 gige0_1、gige0_2 到 vlan2，开启 RSTP 功能，配置网桥优先级为 4096 添加启用 RSTP 的端口 gige0_1、gige0_2。

```
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_1
[INSPUR-vlan2]port gige0_2
[INSPUR-vlan2]exit
[INSPUR]spanning-tree enable
[INSPUR]spanning-tree mode rstp
[INSPUR]spanning-tree bridge-priority 4096
[INSPUR]interface gige0_1
[INSPUR-gige0_1]
[INSPUR-gige0_1]spanning-tree enable
[INSPUR-gige0_1]exit
[INSPUR]interface gige0_2
[INSPUR-gige0_2]spanning-tree enable
```

- (3) 验证配置。

进入用户视图使用命令 **show spanning-tree** 查看 RSTP 状态，RSTP 协议会根据一定的算法阻塞某个端口，使环路消失。

14.4 MSTP 配置案例

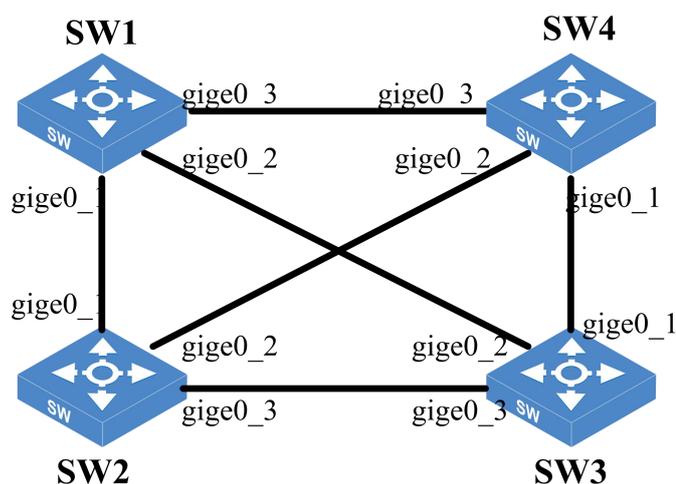
14.4.1 配置需求

MSTP 协议是多生成树协议，与 STP 相比，MSTP 的收敛时间短，能让端口快速处于转发状态。与 RSTP 相比，MSTP 可以将网络划分成不同的域，不同的 vlan 映射到不同的实例上，每个实例对应一颗独立的生成树。提供链路冗余和负载分担。当用户的组网比较复杂，且对收敛时间要求

严格时，使用 MSTP 功能是一个比较好的选择。

14.4.2 网络拓扑

图 14-7 MSTP 组网图



- 以上所有交换机属于同一个域。
- 图中的所有接口均为 Trunk，允许 vlan2-3 通过。
- 实例 1 和实例 2 的根网桥分别为 SW1 和 SW4。
- vlan2 沿着实例 1 转发，vlan 3 沿着实例 2 转发。

14.4.3 配置流程

- (1) 在 SW1 配置上实例 1 的保护 VLAN 为 vlan2，网桥优先级为 0，实例 2 的保护 VLAN 为 vlan3，网桥优先级为 4096，成员端口均为 gige0_1、gige0_2 和 gige0_3。
- (2) 分别在 SW2 和 SW3 上配置实例 1 的保护 VLAN 为 vlan2，网桥优先级为 4096，实例 2 的保护 VLAN 为 vlan3，网桥优先级为 4096，成员端口均为 gige0_1、gige0_2 和 gige0_3。
- (3) 在 SW4 上配置实例 1 的保护 VLAN 为 vlan2，网桥优先级为 4096，实例 2 的保护 VLAN 为 vlan3，网桥优先级为 0，成员端口均为 gige0_1、gige0_2 和 gige0_3。
- (4) 验证配置。

14.4.4 配置步骤

- (1) 在 SW1 配置上实例 1 的保护 VLAN 为 vlan2, 网桥优先级为 0, 实例 2 的保护 VLAN 为 vlan3, 网桥优先级为 4096, 成员端口均为 gige0_1、gige0_2 和 gige0_3。

```
<INSPUR>conf-mode
[INSPUR]vlan 2 to 3
[INSPUR]interface gige0_1
[INSPUR-gige0_1]switchport mode trunk
[INSPUR-gige0_1]switchport trunk allowed vlan 2-3
[INSPUR-gige0_1]switchport trunk native vlan 3
[INSPUR]interface gige0_2
[INSPUR-gige0_2]switchport mode trunk
[INSPUR-gige0_2]switchport trunk allowed vlan 2-3
[INSPUR-gige0_2]switchport trunk native vlan 3
[INSPUR]interface gige0_3
[INSPUR-gige0_3]switchport mode trunk
[INSPUR-gige0_3]switchport trunk allowed vlan 2-3
[INSPUR-gige0_3]switchport trunk native vlan 3
[INSPUR]spanning-tree enable
[INSPUR]spanning-tree mode mst
[INSPUR]interface gige0_1
[INSPUR-gige0_1]spanning-tree enable
[INSPUR-gige0_1]exit
[INSPUR]interface gige0_2
[INSPUR-gige0_2]spanning-tree enable
[INSPUR]interface gige0_3
[INSPUR-gige0_3]spanning-tree enable
[INSPUR]spanning-tree mst configuration
[INSPUR-MSTP]instance 1 vlan 2
[INSPUR]spanning-tree mst 1 bridge-priority 0
[INSPUR-MSTP]instance 2 vlan 3
[INSPUR]spanning-tree mst 2 bridge-priority 4096
[INSPUR]
```

- (2) 分别在 SW2 和 SW3 上配置实例 1 的保护 VLAN 为 vlan2, 网桥优先级为 4096, 实例 2 的保护 VLAN 为 vlan3, 网桥优先级为 4096, 成员端口均为 gige0_1、gige0_2 和 gige0_3。

```
<INSPUR>conf-mode
[INSPUR]vlan 2 to 3
[INSPUR]interface gige0_1
```

```
[INSPUR-gige0_1]switchport mode trunk
[INSPUR-gige0_1]switchport trunk allowed vlan 2-3
[INSPUR-gige0_1]switchport trunk native vlan 3
[INSPUR]interface gige0_2
[INSPUR-gige0_2]switchport mode trunk
[INSPUR-gige0_2]switchport trunk allowed vlan 2-3
[INSPUR-gige0_2]switchport trunk native vlan 3
[INSPUR]interface gige0_3
[INSPUR-gige0_3]switchport mode trunk
[INSPUR-gige0_3]switchport trunk allowed vlan 2-3
[INSPUR-gige0_3]switchport trunk native vlan 3
[INSPUR-gige0_3]exit
[INSPUR]spanning-tree enable
[INSPUR]spanning-tree mode mst
[INSPUR]interface gige0_1
[INSPUR-gige0_1]spanning-tree enable
[INSPUR-gige0_1]exit
[INSPUR]interface gige0_2
[INSPUR-gige0_2]spanning-tree enable
[INSPUR]interface gige0_3
[INSPUR-gige0_3]spanning-tree enable
[INSPUR]spanning-tree mst configuration
[INSPUR-MSTP]instance 1 vlan 2
[INSPUR]spanning-tree mst 1 bridge-priority 4096
[INSPUR-MSTP]instance 2 vlan 3
[INSPUR]spanning-tree mst 2 bridge-priority 4096
[INSPUR]
```

- (3) 在 SW4 上配置实例 1 的保护 VLAN 为 vlan2，网桥优先级为 4096，实例 2 的保护 VLAN 为 vlan3，网桥优先级为 0，成员端口均为 gige0_1、gige0_2 和 gige0_3。

```
<INSPUR>conf-mode
[INSPUR]vlan 2 to 3
[INSPUR]interface gige0_1
[INSPUR-gige0_1]switchport mode trunk
[INSPUR-gige0_1]switchport trunk allowed vlan 2-3
[INSPUR-gige0_1]switchport trunk native vlan 3
[INSPUR]interface gige0_2
[INSPUR-gige0_2]switchport mode trunk
[INSPUR-gige0_2]switchport trunk allowed vlan 2-3
[INSPUR-gige0_2]switchport trunk native vlan 3
[INSPUR]interface gige0_3
```

```
[INSPUR-gige0_3]switchport mode trunk
[INSPUR-gige0_3]switchport trunk allowed vlan 2-3
[INSPUR-gige0_3]switchport trunk native vlan 3
[INSPUR]spanning-tree enable
[INSPUR]spanning-tree mode mst
[INSPUR]interface gige0_1
[INSPUR-gige0_1]spanning-tree enable
[INSPUR-gige0_1]exit
[INSPUR]interface gige0_2
[INSPUR-gige0_2]spanning-tree enable
[INSPUR]interface gige0_3
[INSPUR-gige0_3]spanning-tree enable
[INSPUR]spanning-tree mst configuration
[INSPUR-MSTP]instance 1 vlan 2
[INSPUR]spanning-tree mst 1 bridge-priority 4096
[INSPUR-MSTP]instance 2 vlan 3
[INSPUR]spanning-tree mst 2 bridge-priority 0
[INSPUR]
```

(4) 查看 MSTP 结果。

SW1 状态

```
<INSPUR>show spanning-tree
MST0
Spanning tree enabled protocol MSTP
Root ID Priority 32768
Address 00:11:55:44:33:99
Cost 6666
Port bond1
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32768
Address 00:24:AC:71:AD:85
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Configuration Digest Oxb41829f9 3a54f b74ef7a8 587ff58d

Interface Role Sts Cost Prio P2p
-----
gige0_1 root forwarding 20000 128 yes
gige0_2 alternate blocking 20000 128 yes
gige0_3 designated forwarding 20000 128 yes
```

```
MST1
Spanning tree enabled protocol MSTP
Root ID Priority 0
Address 00:24:AC:71:AD:85
This bridge is the root
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 0
Address 00:24:AC:71:AD:85
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Configuration Digest Oxb41829f9 3a54f b74ef7a8 587ff58d

Interface Role Sts Cost Prio P2p
-----
gige0_1 designated forwarding 20000 128 yes
gige0_2 designated forwarding 20000 128 yes
gige0_3 designated forwarding 20000 128 yes

MST2
Spanning tree enabled protocol MSTP
Root ID Priority 0
Address 00:11:55:44:33:99
Cost 6666
Port bond1
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 4096
Address 00:24:AC:71:AD:85
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Configuration Digest Oxb41829f9 3a54f b74ef7a8 587ff58d

Interface Role Sts Cost Prio P2p
-----
gige0_1 alternate blocking 20000 128 yes
gige0_2 alternate blocking 20000 128 yes
gige0_3 root forwarding 20000 128 yes
<INSPUR>
```

SW2 状态

```
<INSPUR>show spanning-tree
```

```
MST0
Spanning tree enabled protocol MSTP
Root ID Priority 32768
Address 00:11:55:44:33:99
Cost 6666
Port bond1
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32768
Address 00:11:55:44:33:99
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Configuration Digest Oxb41829f9 3a54f b74ef7a8 587ff58d

Interface Role Sts Cost Prio P2p
-----
gige0_1 designated forwarding 20000 128 yes
gige0_2 designated forwarding 20000 128 yes
gige0_3 designated forwarding 20000 128 yes

MST1
Spanning tree enabled protocol MSTP
Root ID Priority 4096
Address 00:24:AC:D4:BB:40
This bridge is the root
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 0
Address 00:24:ac:3a:95:bb
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Configuration Digest Oxb41829f9 3a54f b74ef7a8 587ff58d

Interface Role Sts Cost Prio P2p
-----
gige0_1 root forwarding 20000 128 yes
gige0_2 designated forwarding 20000 128 yes
gige0_3 designated forwarding 20000 128 yes

MST2
Spanning tree enabled protocol MSTP
Root ID Priority 0
```

```
Address 00:24:AC:D4:BB:40
Cost 6666
Port bond1
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 4096
Address 00:11:55:44:33:99
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Configuration Digest Oxb41829f9 3a54f b74ef7a8 587ff58d

Interface Role Sts Cost Prio P2p
-----
gige0_1 designated forwarding 20000 128 yes
gige0_2 root forwarding 20000 128 yes
gige0_3 designated forwarding 20000 128 yes
<INSPUR>
```

SW3 状态

```
<INSPUR>show spanning-tree
MST0
Spanning tree enabled protocol MSTP
Root ID Priority 32768
Address 00:11:55:44:33:99
Cost 6666
Port bond1
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32768
Address 00:24:AC:D4:BB:40
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Configuration Digest Oxb41829f9 3a54f b74ef7a8 587ff58d

Interface Role Sts Cost Prio P2p
-----
gige0_1 designated forwarding 20000 128 yes
gige0_2 designated forwarding 20000 128 yes
gige0_3 root forwarding 20000 128 yes

MST1
Spanning tree enabled protocol MSTP
```

```
Root ID Priority 0
Address 00:24:AC:71:AD:85
This bridge is the root
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 4096
Address 00:24:AC:D4:BB:40
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Configuration Digest Oxb41829f9 3a54f b74ef7a8 587ff58d

Interface Role Sts Cost Prio P2p
-----
gige0_1 designated forwarding 20000 128 yes
gige0_2 root forwarding 20000 128 yes
gige0_3 alternate blocking 20000 128 yes

MST2
Spanning tree enabled protocol MSTP
Root ID Priority 0
Address 00:11:55:44:33:99
Cost 6666
Port bond1
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 4096
Address 00:24:AC:D4:BB:40
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Configuration Digest Oxb41829f9 3a54f b74ef7a8 587ff58d

Interface Role Sts Cost Prio P2p
-----
gige0_1 root forwarding 20000 128 yes
gige0_2 designated forwarding 20000 128 yes
gige0_3 alternate blocking 20000 128 yes
<INSPUR>
```

SW4 状态

```
<INSPUR>show spanning-tree
MST0
Spanning tree enabled protocol MSTP
Root ID Priority 32768
```

```
Address 00:11:55:44:33:99
Cost 6666
Port bond1
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32768
Address 00:24:ac:3a:95:bb
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Configuration Digest Oxb41829f9 3a54f b74ef7a8 587ff58d

Interface Role Sts Cost Prio P2p
-----
gige0_1 alternate blocking 20000 128 yes
gige0_2 root forwarding 20000 128 yes
gige0_3 alternate blocking 20000 128 yes

MST1
Spanning tree enabled protocol MSTP
Root ID Priority 4096
Address 00:24:AC:71:AD:85
This bridge is the root
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 4096
Address 00:24:ac:3a:95:bb
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Configuration Digest Oxb41829f9 3a54f b74ef7a8 587ff58d

Interface Role Sts Cost Prio P2p
-----
gige0_1 alternate blocking 20000 128 yes
gige0_2 alternate blocking 20000 128 yes
gige0_3 root forwarding 20000 128 yes

MST2
Spanning tree enabled protocol MSTP
Root ID Priority 0
Address 00:24:ac:3a:95:bb
Cost 6666
Port bond1
```

```

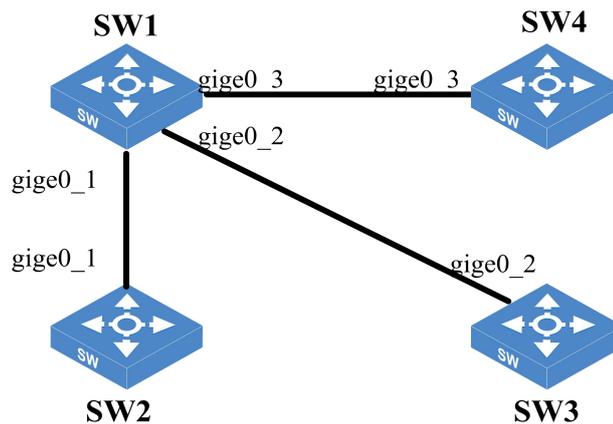
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 0
Address 00:24:ac:3a:95:bb
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Configuration Digest Oxb41829f9 3a54f b74ef7a8 587ff58d

Interface Role Sts Cost Prio P2p
-----
-----
gige0_1 designated forwarding 20000 128 yes
gige0_2 designated forwarding 20000 128 yes
gige0_3 designated forwarding 20000 128 yes
<INSPUR>
    
```

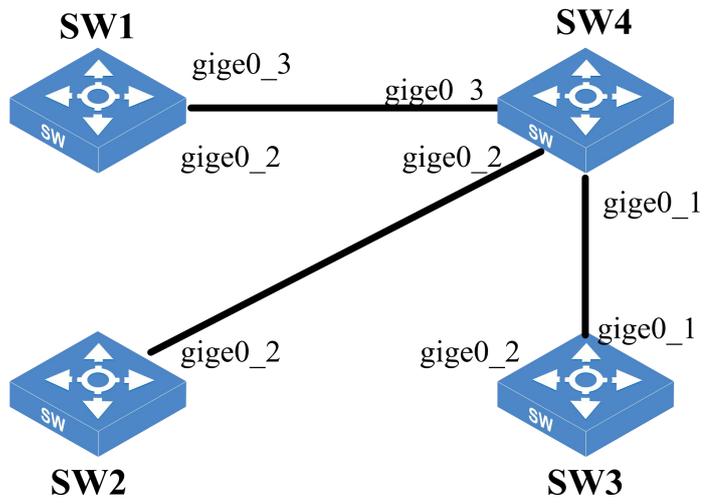
以上结果表明，在实例 1 里，SW1 的优先级最高，作为 MSTP 的树根，其形成的拓扑图如下：

图 14-8 实例 1 树形图



以上结果表明，在实例 2 里，SW4 的优先级最高，作为 MSTP 的树根，其形成的拓扑图如下：

图 14-9 实例 2 树形图



15 VRRP 典型配置案例

15.1 VRRP 简介

虚拟路由冗余协议(Virtual Router Redundancy Protocol, 简称 VRRP)是由 IETF 提出的解决局域网中配置静态网关出现单点失效现象的路由协议。是一种路由容错协议,也可以叫做备份路由协议。一个局域网内的所有主机都设置缺省路由,当主机发出的目的地址不在本网段时,报文将被通过缺省路由发往外部路由器,从而实现了主机与外部网络的通信。当缺省路由器 down 掉(即端口关闭)之后,内部主机将无法与外部通信,如果路由器设置了 VRRP 时,那么这时,虚拟路由将启用备份路由器,从而实现全网通信。

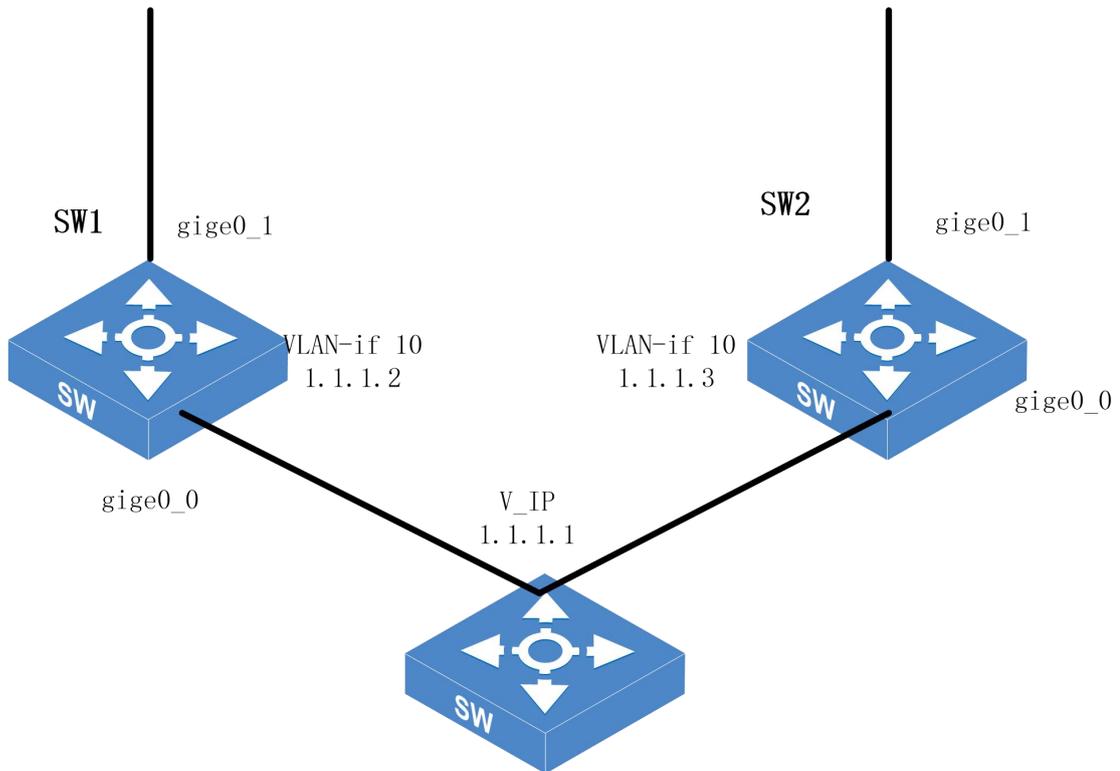
15.2 VRRP 配置案例

15.2.1 配置需求

某公司内网需要网关设备有冗余的功能,当一台设备坏掉后,另一台可以正常工作,不影响公司业务运行。

15.2.2 网络拓扑

图 15-1 VRRP 组网图



15.2.3 配置流程

- (1) 在 SW1 和 SW2 上配置 vlan10，并且配置端口及 vlan-if IP。
- (2) 在 SW1 和 SW2 上开启 VRRP，配置虚拟 IP，设置优先级。
- (3) 验证配置。

15.2.4 配置步骤

- (1) 在 SW1 和 SW2 上配置 vlan10，并且配置端口及 vlan-if IP。

在 SW1 上配置

```
[INSPUR]vlan 10
[INSPUR-vlan10]port gige0_0
[INSPUR-vlan10]exit
```

```
[INSPUR]inter vlan-if10
[INSPUR-vlan-if10]ip address 1.1.1.2/24
[INSPUR-vlan-if10]exit
[INSPUR]
```

在 SW2 上配置:

```
[INSPUR]vlan 10
[INSPUR-vlan10]port gige0_0
[INSPUR-vlan10]exit
[INSPUR]inter vlan-if10
[INSPUR-vlan-if10]ip address 1.1.1.3/24
[INSPUR-vlan-if10]
```

(2) 在 SW1 和 SW2 上开启 VRRP，配置虚拟 IP，设置优先级及强制模式

在 SW1 上配置:

```
[INSPUR-vlan-if10]vrrp 1 ip 1.1.1.1
[INSPUR-vlan-if10]vrrp 1 priority 150
[INSPUR-vlan-if10]vrrp 1 preempt delay 10
```

在 SW2 上配置:

```
[INSPUR-vlan-if10]vrrp 1 ip 1.1.1.1
[INSPUR-vlan-if10]vrrp 1 priority 100
[INSPUR-vlan-if10]vrrp 1 preempt delay 10
```

(3) 验证配置。

```
[INSPUR-vlan-if10]show vrrp
```

用户可以设置网关为 1.1.1.1，能与网关正常通信

16 VSM 典型配置案例

16.1 VSM 简介

VSM (Virtual Switch Matrix) 是一种网络交换技术, 通过将多个物理交换机连接起来, 形成一个虚拟的逻辑交换机, 以提供更高级别的网络功能和管理。

VSM 中的每个设备都被称为成员设备, 它们都具有相同的配置和功能。根据功能不同, 成员设备可以分为两种模式: **Master** 和 **Slave**。

- **Master** 设备是 VSM 中的主控制器, 负责整个交换机系统的配置、管理和监视。它负责向所有成员设备下发配置命令, 监控网络状态, 并报告任何故障和问题。
- **Slave** 设备是 VSM 中的数据平面设备, 负责实际的数据转发和交换操作。它们执行 **Master** 下发的配置命令, 并处理网络流量, 进行数据包的转发和交换。

在 VSM 中, 所有的成员设备都保持相同的配置, 以确保整个交换机系统的一致性和可靠性。**Master** 设备负责管理和协调所有的 **Slave** 设备, 通过不断地与 **Slave** 设备进行通信和同步, 以保持整个系统的一致性。如果 **Master** 设备故障或离线, 一个新的 **Master** 设备将被选举出来, 以确保系统的可用性。

表 16-1 VSM 简介

项目	说明
Master	<p>负责管理整个 VSM。Master 设备是 VSM 中的主控制器,负责管理整个 VSM 系统。它负责配置、监控和管理 VSM 中的所有成员设备。以下是 Master 设备的一些主要责任:</p> <ul style="list-style-type: none"> • 系统配置和管理: Master 设备负责对整个 VSM 系统进行配置和管理。它可以设置 VSM 的全局参数,如时间同步、日志记录、用户权限等。它还可以进行网络拓扑的配置,包括添加、删除和管理成员设备。 • 配置下发: Master 设备负责向所有成员设备下发配置命令和策略。它可以通过命令行界面或图形用户界面与成员设备进行通信,并将配置命令传递给它们。这样,所有设备都可以保持相同的配置状态,从而确保整个系统的一致性。 • 状态监控: Master 设备监控 VSM 中成员设备的状态和性能。它可以定期收集和记录成员设备的统计信息,并对其进行分析和报告。通过监控设备的运行状态和资源利用情况,Master 设备可以检测到任何故障或问题,并采取相应的措施进行修复和优化。 • 故障和问题报告: Master 设备负责检测并报告 VSM 系统中的故障和问题。它可以识别设备离线、链路中断、配置错误等问题,并生成相应的警报和报告。此外,Master 设备还可以提供诊断和故障排除指南,帮助管理员解决问题。
Slave	<p>Slave 设备可以作为 Master 设备的备份设备运行。当 Master 设备故障或失效时,Slave 设备可以接管 Master 设备的职责和功能,确保系统的连续性和稳定性。以下是 Slave 设备的一些主要责任:</p> <ul style="list-style-type: none"> • 监控 Master 设备状态: Slave 设备负责监控 Master 设备的状态和可用性。它可以定期向 Master 设备发送心跳信号,以确保 Master 设备正常工作。如果 Slave 设备检测到 Master 设备故障或失效,它会自动接管控制权。 • 同步配置: Slave 设备可以与 Master 设备保持实时的配置同步。当 Master 设备的配置发生变化时,Slave 设备会自动更新自己的配置,以确保两个设备之间的一致性。 • 接管 Master 职责: 一旦 Slave 设备检测到 Master 设备的故障,它会立即接管 Master 设备的职责和功能。它会开始向成员设备下发配置命令和策略,监控系统的状态和性能,并报告任何故障或问题。 • 提供冗余备份: Slave 设备作为 Master 设备的备份,在系统发生故障时可以提供冗余的备份功能。它可以接管 Master 设备的功能并继续提供服务,从而确保系统的连续性和可用性。

当 Master 故障时,Slave 将自动变成新的 Master 接替原 Master 工作,Master 和 Slave 均由角色选举产生,一个 VSM 中同时只能存在一台 Master,其它成员设备都是 Slave。

VSM 中的几个概念:

表 16-2 VSM 中的几个概念

项目	说明
VSM 标识	<p>VSM 标识 (VSM ID) 是在 VSM 中使用的一个唯一标识符。每台设备都被分配一个 VSM ID, 用以进行 VSM 模式的选举。VSM ID 的取值范围是 0 到 7, 可以选择任意一个值作为设备的 VSM ID。</p> <p>设备之间通过比较 VSM ID 来确定哪台设备将成为主 VSM (Active VSM), 而其他设备将成为备用 VSM (Standby VSM)。主 VSM 负责处理控制平面的操作, 而备用 VSM 则处于备份状态, 以便在主 VSM 故障时能够快速接管控制平面的操作。</p>
VSM 通道	<p>通过 VSM 通道来实现两台设备的通信和跨设备的报文转发。通过 VSM 通道, 两台设备可以直接进行通信, 无需经过其他网络设备的介入。这种直接的通信方式可以提供低延迟和高带宽的传输性能, 适用于对实时性要求较高的应用场景。</p>

16.2 VSM 主备选举

VSM 系统由两台或多台成员设备组成, 每台成员设备具有一个确定的 Mode, 即 Master 或者 Slave, 确定成员设备 Mode 的过程称为主备选举, 在 VSM 刚开始形成时, 每台设备的 VSM Mode 默认都为 Slave。主备选举会在 VSM 建立、新设备加入等情况下产生。主备选举规则如下:

- 当前已经有且只有一台 Master 成员的时候不需要选举;
- 当两台设备的 Mode 一样时进行选举;
- 当 VSM 中设备都为 Slave 时, VSM ID 最小的那台设备为 Master, 其他的都为 Slave;
- 当 VSM 中设备都为 Master 时, VSM ID 最小的设备为 Master, 其他的设备将会以 Slave 模式重启;

16.3 VSM 的配置同步

VSM 的配置同步包括两个步骤: 初始化时的批量同步和稳定运行时的实时同步:

表 16-3 VSM 的配置同步

项目	说明
批量同步	当两台设备组合形成 VSM 时，先选举出 Master 设备。Master 设备使用自己的启动配置文件启动，启动过程中将配置批量同步给 Slave 设备，Slave 设备完成初始化，VSM 形成；在 VSM 运行过程中，有新的成员设备加入时，也会进行批量同步。新设备重启以 Slave 的身份加入 VSM，Master 会将当前的配置批量同步给新设备。新设备以同步过来的配置完成初始化。
实时同步	所有设备初始化完成后，VSM 作为单一网络设备在网络中运行。Master 设备作为 VSM 系统的管理中枢，负责将用户的配置同步给 Slave 设备，从而使 VSM 内的设备的配置随时保持高度统一。

16.4 VSM 维护

VSM 维护的主要功能是监控成员设备的加入和离开，来进行设备之间的主备切换，选举出新的 Master 设备来对 VSM 进行管理。

(1) 成员设备的加入

VSM 维护过程中，会不停的检测其他设备发送过来的心跳信息，当发现有新的成员设备加入时会根据新加入设备的状态采取不同的处理：

- 新加入的设备本身未形成 VSM（比如，新加入的设备配置了 VSM 功能，之后断电，再使用 VSM 电缆连接到已有 VSM 系统，上电重启），则该设备会被选为 Slave。
- 加入的设备本身已经形成了 VSM（比如，新加入的设备配置了 VSM 功能，已经作为 VSM 系统运行，之后使用 VSM 连接到已有 VSM 系统），此时 VSM 中会有两台 Master 设备（请注意，通常情况下，不建议使用这种方式形成 VSM）。在这种情况下，两个 VSM 会进行主备选举，选举遵循主备选举的规则，选举出 Slave 的设备重启后该设备以 Slave 的角色重新加入 VSM。

成员设备加入可能原因有：人为增加 VSM 系统中的成员；故障恢复，当设备故障或链路故障恢复时，恢复的设备会重新加入 VSM。

(2) 成员设备的离开

VSM 通过以下方式能够准确判断是否有成员设备离开，是否需要主备模式的切换：

- 当 VSM 中的成员设备之间的 VSM 通道 down 掉时，这时级联口会 down 掉，VSM 中的其他成员设备会快速检测到该设备离开（不用等到心跳信息超时）。

- 当 VSM 通道出现异常情况时，VSM 中的所有成员都接收不到其他成员的心跳信息，超过超时时间，判定除本设备外的其他设备离开。

获取到离开消息的成员设备会根据本地维护的 VSM 信息来判断离开的是 Master 还是 Slave，如果离开的是 Master，则触发新的主备选举，再更新本地的 VSM 信息；如果离开的是 Slave，则直接更新本地的 VSM 信息。成员设备离开可能原因有：人为取走成员设备；成员设备故障；链接故障。

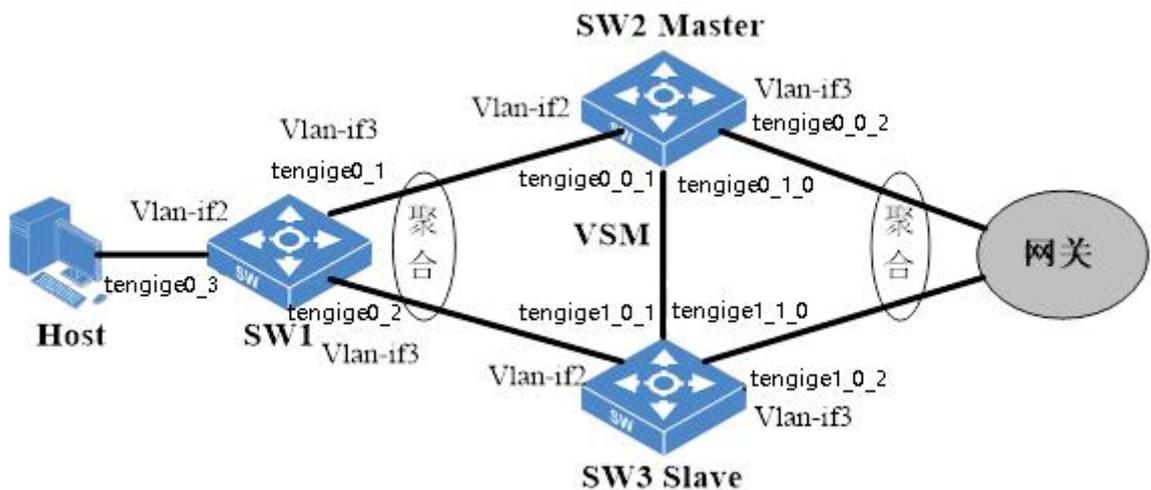
16.5 VSM 典型配置案例

16.5.1 配置要求

VSM 采用了一系列的冗余备份技术来保证 VSM 系统的高可靠性，用户可以将 VSM 设备用于接入层、汇聚层和数据中心，从而尽量缩短因日常维护操作和突发的系统崩溃所导致的停机时间，减少网络故障带来的影响。

16.5.2 网络拓扑

图 16-2 VSM 组网图



- 开启 VSM 后，设备会自动选举出 Master，VSM ID 较小的设备将被选为 Master，进入用户视图使用 `show vsm` 命令查看哪台设备为 Master，所有配置均在 Master 上配置。开启或者关闭 VSM 功能，设备会清除配置重启。
- 需要保证 SW1 和网关之间路由可达，可以配置静态路由或者动态路由协议，详细配置见路由协议一章。

16.5.3 配置流程

- (1) 在 SW2 和 SW3 上开启 VSM 功能，配置 SW2 的 VSM ID 为 0，SW3 的 VSM ID 为 1。
- (2) 分别在 SW1、SW2 上创建相应的 vlan-if 口并配置 IP 地址。
- (3) 在 SW1 和 SW2 上配置端口聚合。
- (4) 验证配置。

16.5.4 配置步骤

- (1) 分别在 SW2 和 SW3 上开启 VSM 功能。

在 SW2 上配置

```
<INSPUR>conf-mode
[INSPUR] vsm enable id 0 uplink-port-list tengigel_0 downlink-port-list null
The configuration will cause rebooting and take effect after that.
Are you sure? (Y/N) [N]: y
```

SW2 设备页面配置

从导航栏选择 Main => 基础配置 => VSM => VSM 配置，进入 VSM 配置页面，如下图所示。

启用; VSM ID: 0; 上行级联口: tengige1_0; 下行级联口: 请配置; 本框优先转发: 启用; 分裂监测: 启用."/>

在 SW3 上配置

```
<INSPUR>conf-mode
[INSPUR]vsm enable id 1 uplink-port-list null downlink-port-list tengige1_0
The configuration will cause rebooting and take effect after that.
Are you sure? (Y/N) [N]: y
```

SW3 设备页面配置：

从导航栏选择 Main => 基础配置 => VSM => VSM 配置，进入 VSM 配置页面，如下图所示。

 启用; VSM ID: 1; 上行级联口: tengige1_0; 下行级联口: 请配置; 本框优先转发: 启用; 分裂监测: 启用."/>

(2) 分别在 SW1、SW2 上创建相应的 vlan-if 口并配置 IP 地址。

在 SW1 上配置

```
<INSPUR>conf-mode
[INSPUR]vlan 2
[INSPUR-vlan2]port tengige0_3
[INSPUR-vlan2]exit
```

```
[INSPUR]interface vlan-if2
[INSPUR-vlan-if2]ip address 1.1.1.1/24
[INSPUR-vlan-if2]exit
[INSPUR]vlan 3
[INSPUR-vlan3]port tengige0_1
[INSPUR-vlan3]port tengige0_2
[INSPUR-vlan3]exit
[INSPUR]interface vlan-if3
[INSPUR-vlan-if3]ip address 2.2.2.1/24
[INSPUR-vlan-if3]exit
[INSPUR]
```

SW1 页面上的配置：

从导航栏选择 Main => VLAN 管理 => VLAN 接口，进入 VLAN 配置页面，如下图所示。

VLAN ID	描述	包含端口	流量统计	操作
1	VLAN 0001	tengige0_4,tengige0_5,tengige0_6	<input type="checkbox"/> 启用	
2	VLAN 0002	tengige0_3	<input type="checkbox"/> 启用	
3	VLAN 0003	tengige0_1,tengige0_2	<input type="checkbox"/> 启用	

VLAN ID	接口名称	接口描述	接口IP/掩码	自定义MAC地址	设置MTU	状态	操作
1	vlan-if1	vlan-if1	无	无	1500	<input type="checkbox"/> 禁用	
2	vlan-if2	vlan-if2	主地址(IPv4): 1.1.1.1/24	无	1500	<input type="checkbox"/> 禁用	
3	vlan-if3	vlan-if3	主地址(IPv4): 2.2.2.1/24	无	1500	<input type="checkbox"/> 禁用	

在 SW2 上配置

```
<INSPUR>conf-mode
[INSPUR]vlan 2
[INSPUR-vlan2]port tengige0_0_1
[INSPUR-vlan2]port tengige1_0_1
[INSPUR-vlan2]exit
[INSPUR]interface vlan-if2
[INSPUR-vlan-if2]ip address 2.2.2.2/24
[INSPUR-vlan-if2]exit
[INSPUR]vlan 3
```

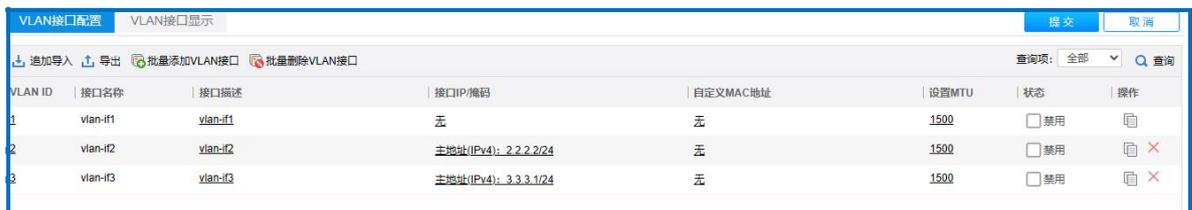
```
[INSPUR-vlan3]port tengige0_0_2
[INSPUR-vlan3]port tengige1_0_2
[INSPUR-vlan3]exit
[INSPUR]interface vlan-if3
[INSPUR-vlan-if3]ip address 3.3.3.1/24
[INSPUR-vlan-if3]exit
[INSPUR]
```

SW2 页面上的配置：

从导航栏选择 Main => VLAN 管理 => VLAN 接口，进入 VLAN 配置页面，如下图所示。



VLAN ID	描述	包含端口	流量统计	操作
1	VLAN 0001	tengige0_0_3, tengige0_0_4	<input type="checkbox"/> 启用	
2	VLAN 0002	tengige0_0_1, tengige1_0_1	<input type="checkbox"/> 启用	
3	VLAN 0003	tengige0_0_2, tengige1_0_2	<input type="checkbox"/> 启用	



VLAN ID	接口名称	接口描述	接口IP掩码	自定义MAC地址	设置MTU	状态	操作
1	vlan-if1	vlan-if1	无	无	1500	<input type="checkbox"/> 禁用	
2	vlan-if2	vlan-if2	主地址(IPv4): 2.2.2.2/24	无	1500	<input type="checkbox"/> 禁用	
3	vlan-if3	vlan-if3	主地址(IPv4): 3.3.3.1/24	无	1500	<input type="checkbox"/> 禁用	

SW3 的配置由 SW2 的配置同步过来。

(3) 在 SW1 和 SW2 上配置端口聚合。

在 SW1 上配置

```
<INSPUR>conf-mode
[INSPUR]interface bond 1
[INSPUR-bond1]bond mode dynamic
[INSPUR-bond1]bond load-sharing mode source-destination-ip
[INSPUR-bond1]switchport mode access
[INSPUR-bond1]switchport access vlan 3
[INSPUR-bond1]exit
[INSPUR]interface tengige0_1
[INSPUR-tengige0_1]bond group 1
[INSPUR-tengige0_1]exit
```

```
[INSPUR]interface tengige0_2
[INSPUR-tengige0_2]bond group 1
[INSPUR-tengige0_2]exit
[INSPUR]
```

页面上配置：

1、端口聚合



2、VLAN 端口配置



在 SW2 上配置 bond1

```
<INSPUR>conf-mode
[INSPUR]interface bond 1
[INSPUR-bond1]bond mode dynamic
[INSPUR-bond1]bond load-sharing mode source-destination-ip
[INSPUR-bond1]switchport mode access
[INSPUR-bond1]switchport access vlan 3
[INSPUR-bond1]exit
[INSPUR]interface tengige0_0_1
[INSPUR-tengige0_0_1]bond group 1
```

```
[INSPUR-tengige0_0_1]exit
[INSPUR]interface tengige1_0_1
[INSPUR-tengige1_0_1]bond group 1
[INSPUR-tengige1_0_1]exit
[INSPUR]
```

页面上配置：

1、端口聚合

序号	聚合组ID	聚合组名字	聚合组描述	聚合组类型	出端口算法	端口列表	高级配置	操作
1	1	bond1	1	动态聚合	源IP+目的IP	tengige0_0_1,tengige1_0_1	无	🔍 ✖

2、VLAN 端口配置

端口名称	类型	所属VLAN	默认VLAN
bond1	access(系统设置)	3	3

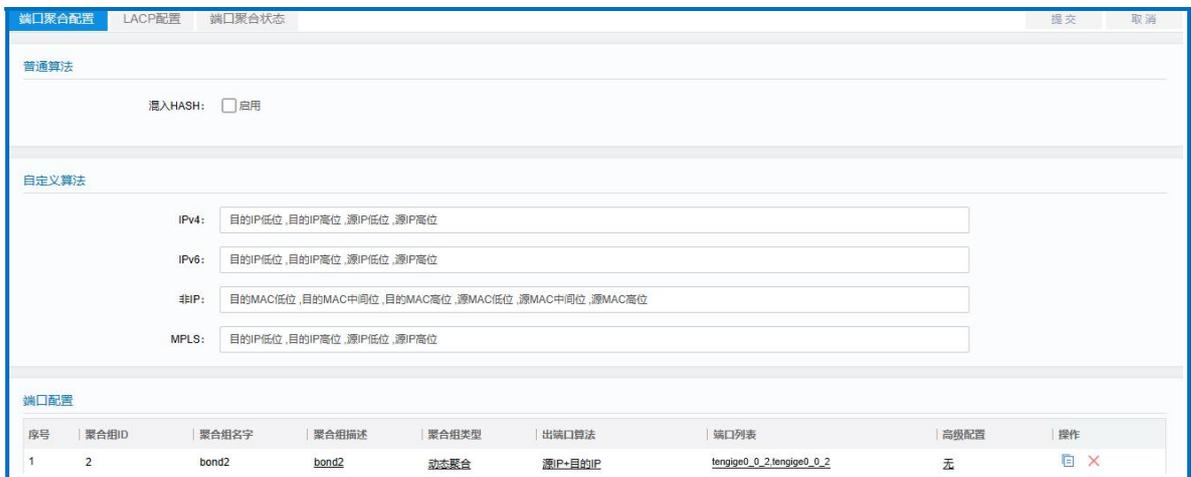
在 SW2 上配置 bond2

```
<INSPUR>conf-mode
[INSPUR]interface bond 2
[INSPUR-bond2]bond mode dynamic
[INSPUR-bond2]bond load-sharing mode source-destination-ip
[INSPUR-bond2]switchport mode access
[INSPUR-bond2]switchport access vlan 3
[INSPUR-bond2]exit
[INSPUR]interface tengige0_0_2
```

```
[INSPUR-tengige0_0_2]bond group 1
[INSPUR-tengige0_0_2]exit
[INSPUR]interface tengige1_0_2
[INSPUR-tengige1_0_2]bond group 1
[INSPUR-tengige1_0_2]exit
[INSPUR]
```

页面上配置：

1、端口聚合



2、VLAN 端口配置



(4) 验证配置。

正常情况下，Master 和 Slave 设备根据出端口算法进行转发报文，Host 能通过网关访问外网；当 Master 设备出现故障时，Slave 设备自动变成 Master 独自承担报文转发，Host 也能访问外网；当 Slave 设备出现故障时，Master 独自承担报文转发，Host 仍然能访问外网。这样就实现了链路冗余，提高了网络的可靠性。

17 VRF 典型配置案例

17.1 VRF 简介

VRF（虚拟路由转发）主要用于路由隔离，解决地址重叠问题。每一个 VRF 可以看作一台虚拟的交换机，该虚拟交换机包括以下元素：

- 一张独立的路由转发表，独立的地址空间；
- 一组属于这个 VRF 的接口集合；
- 一组只用于 VRF 的路由协议。

在 VRF 中定义的和 VPN 业务有关的两个重要参数是 RT 和 RD：

表 17-1 VRF 简介

项目	说明
RT (Route Target)	主要用于控制 VPN 路由的发布和安装策略。分为 import 和 export 两种属性，前者表示了我对那些路由感兴趣，而后者表示了我发出的路由的属性。当 PE 发布路由时，将使用路由所属 VRF 的 RT export 规则，直接发送给其他的 PE 设备。对端 PE 接收路由时，首先接收所有的路由，并根据每个 VRF 配置的 RT 的 import 规则进行检查，如果与路由中的 RT 属性匹配，则将该路由加入到相应的 VRF 中。
RD (Route Distinguisher)	用于说明路由属于哪个 VPN 的标志，理论上可以为每个 VRF 配置一个 RD，通常建议为每个 VPN 的 VRF 都配置相同的 RD，并且要保证这个 RD 全球唯一。如果两个 VRF 中存在相同的地址，但是由于 RD 不同，这两个路由在 PE 间发布过程中也不会混淆，把 RD 和路由一起发送，对端 PE 可以根据 RD 确定路由所属的 VPN，从而把路由安装到正确的 VRF 中。

17.2 VRF 典型配置案例

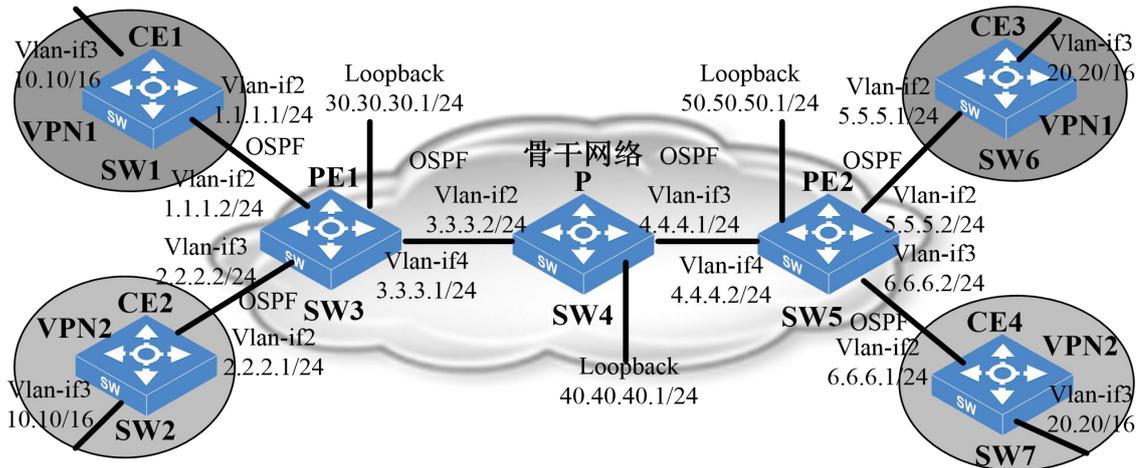
17.2.1 配置要求

当运营商的网络承载几个公司（如公司 A 和公司 B）的流量，公司 A 和公司 B 的异地部门能相互访问的同时禁止 A 和 B 互访。此时运营商网络的 PE 设备会遇到本地路由冲突、路由在网络中的

传播以及 PE 向 CE 的报文转发等问题，在 PE 设备上使用 VRF 功能就可以有效解决这些问题。

17.2.2 网络拓扑

图 17-2 VRF 与 MPLS 结合使用组网图



CE1 是 Site1 的一台交换机，CE2 是 Site2 的一台交换机，CE3 是 Site3 的一台交换机，CE4 是 Site4 的一台交换机。

- Site1、Site3 分别是 VPN1 的两个站点，Site2、Site4 分别是 VPN2 的两个站点。
- PE、CE 与 P 间使用 OSPF 路由协议。

17.2.3 配置流程

- (1) 分别在 SW1、SW2、SW3、SW4、SW5、SW6、SW7 上创建相应的 vlan-if 口，并配置相应的 IP 地址。
- (2) 分别在 SW3 和 SW5 上配置 VRF。
- (3) 分别在 SW1、SW2、SW3、SW4、SW5、SW6、SW7 上配置 OSPF 路由。
- (4) 分别在 SW3 和 SW5 上配置 IBGP 并建立 VPNv4 邻居。
- (5) 分别在 SW3、SW4 和 SW5 上配置 MPLS。
- (6) 验证配置。

17.2.4 配置步骤

- (1) 分别在 SW1、SW2、SW3、SW4、SW5、SW6、SW7 上创建相应的 vlan-if 口，并配置相应的 IP 地址。

在 SW1 上配置

```
[INSPUR]vlan 2 to 3
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_2
[INSPUR-vlan2]exit
[INSPUR]vlan 3
[INSPUR-vlan3]port gige0_3
[INSPUR-vlan3]exit
[INSPUR]interface vlan-if2
[INSPUR-vlan-if2]ip address 1.1.1.1/24
[INSPUR-vlan-if2]exit
[INSPUR]interface vlan-if3
[INSPUR-vlan-if3]ip address 10.10.1.1/16
[INSPUR-vlan-if3]exit
[INSPUR]
```

在 SW2 上配置

```
[INSPUR]vlan 2 to 3
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_2
[INSPUR-vlan2]exit
[INSPUR]vlan 3
[INSPUR-vlan3]port gige0_3
[INSPUR-vlan3]exit
[INSPUR]interface vlan-if2
[INSPUR-vlan-if2]ip address 2.2.2.1/24
[INSPUR-vlan-if2]exit
[INSPUR]interface vlan-if3
[INSPUR-vlan-if3]ip address 10.10.1.1/16
[INSPUR-vlan-if3]exit
[INSPUR]
```

在 SW3 上配置

```
[INSPUR]vlan 2 to 4
[INSPUR]vlan 2
```

```
[INSPUR-vlan2]port gige0_2
[INSPUR-vlan2]exit
[INSPUR]vlan 3
[INSPUR-vlan3]port gige0_3
[INSPUR-vlan3]exit
[INSPUR]vlan 4
[INSPUR-vlan4]port gige0_4
[INSPUR-vlan4]exit
[INSPUR]interface vlan-if2
[INSPUR-vlan-if2]ip address 1.1.1.2/24
[INSPUR-vlan-if2]exit
[INSPUR]interface vlan-if3
[INSPUR-vlan-if3]ip address 2.2.2.2/24
[INSPUR-vlan-if3]exit
[INSPUR]interface vlan-if4
[INSPUR-vlan-if4]ip address 3.3.3.1/24
[INSPUR-vlan-if4]exit
[INSPUR]interface loopback 1
[INSPUR-loopback1]ip address 30.30.30.1/24
[INSPUR-loopback1]exit
[INSPUR]
```

在 SW4 上配置

```
[INSPUR]vlan 2 to 3
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_2
[INSPUR-vlan2]exit
[INSPUR]vlan 3
[INSPUR-vlan3]port gige0_3
[INSPUR-vlan3]exit
[INSPUR]interface vlan-if2
[INSPUR-vlan-if2]ip address 3.3.3.2/24
[INSPUR-vlan-if2]exit
[INSPUR]interface vlan-if3
[INSPUR-vlan-if3]ip address 4.4.4.1/24
[INSPUR-vlan-if3]exit
[INSPUR]interface loopback 1
[INSPUR-loopback1]ip address 40.40.40.1/24
[INSPUR-loopback1]exit
[INSPUR]
```

在 SW5 上配置

```
[INSPUR]vlan 2 to 4
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_2
[INSPUR-vlan2]exit
[INSPUR]vlan 3
[INSPUR-vlan3]port gige0_3
[INSPUR-vlan3]exit
[INSPUR]vlan 4
[INSPUR-vlan4]port gige0_4
[INSPUR-vlan4]exit
[INSPUR]interface vlan-if2
[INSPUR-vlan-if2]ip address 5.5.5.2/24
[INSPUR-vlan-if2]exit
[INSPUR]interface vlan-if3
[INSPUR-vlan-if3]ip address 6.6.6.2/24
[INSPUR-vlan-if3]exit
[INSPUR]interface vlan-if4
[INSPUR-vlan-if4]ip address 4.4.4.2/24
[INSPUR-vlan-if4]exit
[INSPUR]interface loopback 1
[INSPUR-loopback1]ip address 50.50.50.1/24
[INSPUR-loopback1]exit
[INSPUR]
```

在 SW6 上配置

```
[INSPUR]vlan 2 to 3
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_2
[INSPUR-vlan2]exit
[INSPUR]vlan 3
[INSPUR-vlan3]port gige0_3
[INSPUR-vlan3]exit
[INSPUR]interface vlan-if2
[INSPUR-vlan-if2]ip address 5.5.5.1/24
[INSPUR-vlan-if2]exit
[INSPUR]interface vlan-if3
[INSPUR-vlan-if3]ip address 20.20.1.1/16
[INSPUR-vlan-if3]exit
[INSPUR]
```

在 SW7 上配置

```
[INSPUR]vlan 2 to 3
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_2
[INSPUR-vlan2]exit
[INSPUR]vlan 3
[INSPUR-vlan3]port gige0_3
[INSPUR-vlan3]exit
[INSPUR]interface vlan-if2
[INSPUR-vlan-if2]ip address 6.6.6.1/24
[INSPUR-vlan-if2]exit
[INSPUR]interface vlan-if3
[INSPUR-vlan-if3]ip address 20.20.1.1/16
[INSPUR-vlan-if3]exit
[INSPUR]
```

(2) 分别在 SW3 和 SW5 上配置 VRF 。

在 SW3 上配置 VRF

```
[INSPUR]vrf enable
[INSPUR]vrf vrf1
[INSPUR-vrf-vrf1]rd 100:1
[INSPUR-vrf-vrf1]router-target import 100:2
[INSPUR-vrf-vrf1]router-target export 100:3
[INSPUR-vrf-vrf1]exit
[INSPUR] interface vlan-if 2
[INSPUR-vlan-if2] bind vrf vrf1
[INSPUR-vlan-if2] ip address 1.1.1.2/24
[INSPUR-vlan-if2] exit
[INSPUR]vrf vrf2
[INSPUR-vrf-vrf2]rd 100:10
[INSPUR-vrf-vrf2]router-target import 100:11
[INSPUR-vrf-vrf2]router-target export 100:12
[INSPUR-vrf-vrf2]exit
[INSPUR] interface vlan-if3
[INSPUR-vlan-if3] bind vrf vrf2
[INSPUR-vlan-if3]ip address 2.2.2.2/24
[INSPUR-vlan-if3] exit
[INSPUR]
```

在 SW5 上配置 VRF

```
[INSPUR]vrf enable
[INSPUR]vrf vrf1
```

```
[INSPUR-vrf-vrf1] rd 100:4
[INSPUR-vrf-vrf1]router-target import 100:3
[INSPUR-vrf-vrf1]router-target export 100:2
[INSPUR-vrf-vrf1]exit
[INSPUR] interface vlan-if2
[INSPUR-vlan-if2] bind vrf vrf1
[INSPUR-vlan-if2]ip address 5.5.5.2/24
[INSPUR-vlan-if2] exit
[INSPUR]vrf vrf2
[INSPUR-vrf-vrf2]rd 100:20
[INSPUR-vrf-vrf2]router-target import 100:12
[INSPUR-vrf-vrf2]router-target export 100:11
[INSPUR-vrf-vrf2]exit
[INSPUR] interface vlan-if3
[INSPUR-vlan-if3] bind vrf vrf2
[INSPUR-vlan-if3]ip address 6.6.6.2/24
[INSPUR-vlan-if3] exit
[INSPUR]
```

(3) 分别在 SW1、SW2、SW3、SW4、SW5、SW6、SW7 上配置 OSPF 路由。

在 SW1 上配置 OSPF

```
[INSPUR]router ospf 1
[INSPUR-ospf-1]network 1.1.1.0/24 area 0
[INSPUR-ospf-1]network 10.10.0.0/16 area 0
[INSPUR-ospf-1]exit
[INSPUR]
```

在 SW2 上配置 OSPF

```
[INSPUR]router ospf 1
[INSPUR-ospf-1]network 2.2.2.0/24 area 0
[INSPUR-ospf-1]network 10.10.0.0/16 area 0
[INSPUR-ospf-1]exit
[INSPUR]
```

在 SW3 上配置 OSPF

```
[INSPUR] router ospf 1 vrfname vrf1
[INSPUR-ospf-1]network 1.1.1.0/24 area 0
[INSPUR-ospf-1]exit
[INSPUR]router ospf 2 vrfname vrf2
[INSPUR-ospf-2]network 2.2.2.0/24 area 0
[INSPUR-ospf-2]exit
```

```
[INSPUR]router ospf 3
[INSPUR-ospf-3]network 3.3.3.0/24 area 0
[INSPUR-ospf-3]network 30.30.30.0/24 area 0
[INSPUR-ospf-3]exit
[INSPUR]
```

在 SW4 上配置 OSPF

```
[INSPUR]router ospf 1
[INSPUR-ospf-1]network 3.3.3.0/24 area 0
[INSPUR-ospf-1]network 4.4.4.0/24 area 0
[INSPUR-ospf-1]network 40.40.40.0/24 area 0
[INSPUR-ospf-1]exit
[INSPUR]
```

在 SW5 上配置 OSPF

```
[INSPUR]router ospf 1 vrfname vrf1
[INSPUR-ospf-1]network 5.5.5.0/24 area 0
[INSPUR-ospf-1]exit
[INSPUR]router ospf 2 vrfname vrf2
[INSPUR-ospf-2]network 6.6.6.0/24 area 0
[INSPUR-ospf-2]exit
[INSPUR]router ospf 3
[INSPUR-ospf-3]network 4.4.4.0/24 area 0
[INSPUR-ospf-3]network 50.50.50.0/24 area 0
[INSPUR-ospf-3]exit
[INSPUR]
```

在 SW6 上配置 OSPF

```
[INSPUR]router ospf 1
[INSPUR-ospf-1]network 5.5.5.0/24 area 0
[INSPUR-ospf-1]network 20.20.0.0/16 area 0
[INSPUR-ospf-1]exit
[INSPUR]
```

在 SW7 上配置 OSPF

```
[INSPUR]router ospf 1
[INSPUR-ospf-1]network 6.6.6.0/24 area 0
[INSPUR-ospf-1]network 20.20.0.0/16 area 0
[INSPUR-ospf-1]exit
[INSPUR]
```

(4) 分别在 SW3 和 SW5 上配置 IBGP 并建立 VPNv4 邻居。

在 SW3 上配置 IBGP 并建立 VPNv4 邻居

```
[INSPUR]router bgp 1
[INSPUR-bgp]neighbor 50.50.50.1 remote-as 1
[INSPUR-bgp]neighbor 50.50.50.1 update-source loopback1
[INSPUR-bgp]neighbor 50.50.50.1 activate
[INSPUR-bgp]address-family vpnv4 unicast
[INSPUR-bgp-vpnv4]neighbor 50.50.50.1 activate
[INSPUR-bgp-vpnv4]exit
[INSPUR-bgp]address-family ipv4 vrf vrf1
INSPUR(config-router-af-vrf)# redistribute ospf 1
INSPUR(config-router-af-vrf)# exit
[INSPUR-bgp]address-family ipv4 vrf vrf2
INSPUR(config-router-af-vrf)# redistribute ospf 2
INSPUR(config-router-af-vrf)# exit
[INSPUR-bgp]
```

在 SW5 上配置 IBGP 并建立 VPNv4 邻居

```
[INSPUR]router bgp 1
[INSPUR-bgp]neighbor 30.30.30.1 remote-as 1
[INSPUR-bgp]neighbor 30.30.30.1 update-source loopback1
[INSPUR-bgp]neighbor 30.30.30.1 activate
[INSPUR-bgp]address-family vpnv4 unicast
[INSPUR-bgp-vpnv4]neighbor 30.30.30.1 activate
[INSPUR-bgp-vpnv4]exit
[INSPUR-bgp]address-family ipv4 vrf vrf1
INSPUR(config-router-af-vrf)# redistribute ospf 1
INSPUR(config-router-af-vrf)# exit
[INSPUR-bgp]address-family ipv4 vrf vrf2
INSPUR(config-router-af-vrf)# redistribute ospf 2
INSPUR(config-router-af-vrf)# exit
[INSPUR-bgp]
```

(5) 分别在 SW3、SW4 和 SW5 上配置 MPLS。

SW3 上配置 MPLS

```
[INSPUR]mpls ip
[INSPUR] mpls ldp router-id 30.30.30.1
[INSPUR] interface vlan-if 4
[INSPUR-vlan-if4] mpls ip
```

SW4 上配置 MPLS

```
[INSPUR]mpls ip
[INSPUR] mpls ldp router-id 40.40.40.1
[INSPUR] interface vlan-if 2
[INSPUR-vlan-if2] mpls ip
[INSPUR-vlan-if2]exit
[INSPUR] interface vlan-if 3
[INSPUR-vlan-if3] mpls ip
```

SW4 上配置 MPLS

```
[INSPUR]mpls ip
[INSPUR] mpls ldp router-id 50.50.50.1
[INSPUR] interface vlan-if 4
[INSPUR-vlan-if4] mpls ip
```

(6) 验证配置。

VPN1 的两个站点内的主机可以互访，但不能访问 VPN2 内的主机。同理，VPN2 的两个站点内的主机可以互访，但不能访问 VPN1 内的主机。从而实现了两个 VPN 间的逻辑划分和安全隔离。