

# S9800 系列交换机

## 命令典型配置手册

手册版本：v3.1

软件版本：S511C013D001P02PATCH05

发布日期：2022-03-26

# 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 .....	2
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 MAC/IP/端口绑定典型配置案例 .....	1
7.1 MAC/IP/端口绑定简介 .....	1
7.2 配置案例 .....	1
8 PVLAN 典型配置案例 .....	1
8.1 PVLAN 简介 .....	1
8.2 配置案例 .....	1
9 QinQ 典型配置案例 .....	1
9.1 QinQ 简介 .....	1
9.2 基本 QinQ 配置案例 .....	1

9.3 灵活 QinQ 配置案例 .....	4
10 ARP 防护典型配置案例 .....	1
10.1 ARP 防护简介 .....	1
10.2 ARP 报文一致性检测配置案例 .....	2
10.3 ARP 用户合法性配置案例 .....	4
10.4 ARP 网关保护配置案例 .....	6
11 路由协议典型配置案例 .....	1
11.1 路由协议简介 .....	1
11.2 静态路由配置案例 .....	2
11.3 RIP 路由配置案例 .....	4
11.4 OSPF 典型配置案例 .....	7
11.5 OSPF 多进程典型配置案例 .....	10
12 DHCP 典型配置案例 .....	1
12.1 DHCP 简介 .....	1
12.2 DHCP Server 配置案例 .....	2
12.3 DHCP 中继配置案例 .....	4
12.4 DHCP Snooping 配置案例 .....	6
13 QoS 典型配置案例 .....	1
13.1 QoS 简介 .....	1
13.2 配置案例 .....	2
14 802.1x 典型配置案例 .....	1
14.1 802.1x 简介 .....	1
14.2 802.1x 本地认证配置案例 .....	2
14.3 802.1x Radius 认证配置案例 .....	3
15 MAC 认证典型配置案例 .....	1
15.1 MAC 地址认证简介 .....	1
15.2 MAC 地址本地认证配置案例 .....	2
15.3 MAC 地址 Radius 认证配置案例 .....	3
16 Portal 典型配置案例 .....	1
16.1 Portal 认证简介 .....	1
16.2 Portal 认证配置案例 .....	1
17 生成树典型配置案例 .....	1
17.1 生成树简介 .....	1
17.2 STP 配置案例 .....	2
17.3 RSTP 配置案例 .....	6
17.4 MSTP 配置案例 .....	8
18 VRRP 典型配置案例 .....	1

---

18.1 VRRP 简介 .....	1
18.2 VRRP 配置案例 .....	1
19 VSM 典型配置案例 .....	1
19.1 VSM 简介 .....	1
19.2 VSM 主备选举 .....	1
19.3 VSM 的配置同步 .....	2
19.4 VSM 维护 .....	2
19.5 VSM 典型配置案例 .....	3
20 OVC 典型配置案例 .....	1
20.1 OVC 简介 .....	1
20.2 管理平面虚拟化 .....	2
20.3 控制平面虚拟化 .....	2
20.4 数据平面虚拟化 .....	3
20.5 OVC 典型配置案例 .....	4
21 VRF 典型配置案例 .....	1
21.1 VRF 简介 .....	1
21.2 MPLS VPN 简介 .....	1
21.3 VRF 典型配置案例 .....	2

# 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” Press Ctrl+B to enter extend boot

menu”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
Reboot by cpld
Initializing DDR....
sdhci@30400000: 0

*****
*                                                                 *
*                               System Booting                       *
*                                                                 *
*****

Basic Version           : 22.05.02
CPU Type                : Centec CTC5236 (TsingMa) Ver. 1.0
Frequency               : CPU0 (A53),1200 MHz  CPU1 (A53),1200 MHz  Bus:500  MHz
DRAM                   : 2GiB (DDR4, 16-bit, CS=1, ECC on) 2133 Mbps
GPIO_INIT              : Pass
CPLD_INIT              : Pass
I2C_INIT               : Pass
Press Ctrl+E to enter Basic boot menu... 0
boot extend...
Net                    : eth0: ethernet@33410000
Extend Version         : 22.05.02
Compiled Date          : 15:48:19 Mar  2 2022
Board_Id               : 0x864
PCB version            : 0xa
CPLD version           : 1.0 2022-1-20
```

```

EMMC      Initializing: Passed
Capacity      : 3.6GiB
Elabel      Initializing: Passed
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:      INSPUR-S9800-S511C013D001P05.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 [INSPUR-S9800-S511C013D001P05.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: INSPUR-S9800-S511C013D001P04.bin 45381504
2: INSPUR-S9800-S511C013D001P05.bin 45386636 M
=====

```

```

enter
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: INSPUR-S9800-S511C013D001P04.bin 45381504
2: INSPUR-S9800-S511C013D001P05.bin 45386636 M
=====
enter your chioce: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 INSPUR-S9800-S511C013D001P05.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 INSPUR-S9800-S511C013D001P05.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 INSPUR-S9800-S511C013D001P05.bin
```

```
[INSPUR]
```

- 命令行模式下删除版本。

```
[INSPUR]boot-file delete INSPUR-S9800-S511C013D001P05.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

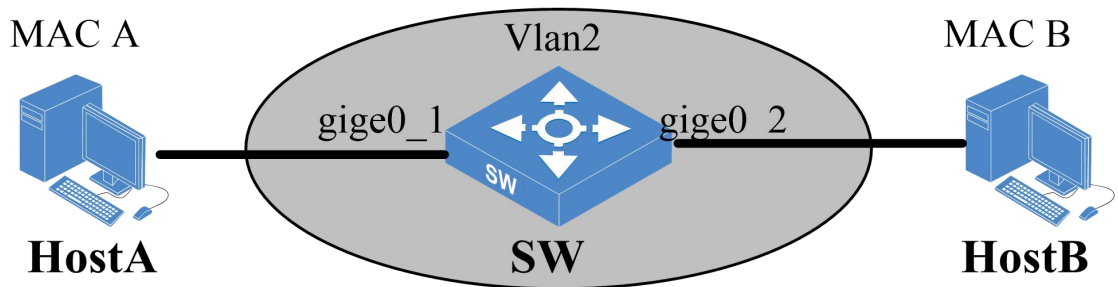
二层转发是建立在 MAC 地址基础上的数据转发，由内部芯片来完成数据转发，转发速度快，性能高。二层转发通过维护 MAC 地址表实现转发，收到报文时将其源 MAC 与端口的对应关系写到 MAC 表里，作为二层转发的依据，转发时根据报文的目的 MAC 查找 MAC 表进行转发。二层报文只能在相同 VLAN 里通过二层转发。

#### 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 dynamic** 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

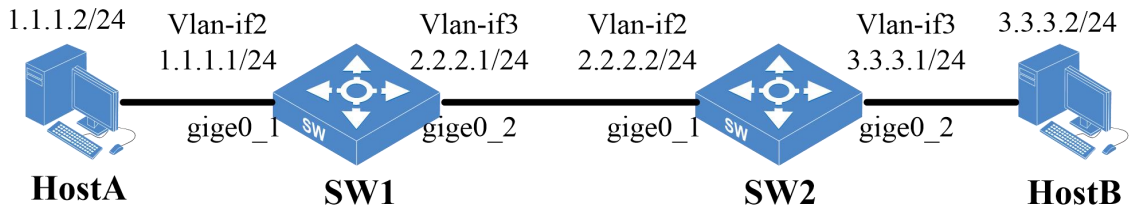
三层转发主要由芯片和 CPU 两大部分完成，芯片主要负责转发功能，内部有二层 MAC 表项和三层转发表项，CPU 主要用于转发控制，维护表项并将三层转发表项下发到芯片。当交换机上未建立任何转发表项的时候，位于不同网段的 PC 不能直接通过芯片转发进行通信，需要 CPU 的参与在三层交换机芯片上建立三层转发表项。表项建立完成后，交换机收到三层报文就会按照查询 MAC 表-->查询三层转发表项的流程将报文转发到目的主机。

### 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。

#### 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（ <b>Selected</b> ）状态	成员端口处于此状态时可以参与用户数据的转发
非选中（ <b>Unselected</b> ）状态	成员端口处于此状态时的不能参与用户数据的转发
主端口（ <b>Master</b> ）	处于 <b>Selected</b> 状态且端口号最小的端口称为主端口（ <b>Master Port</b> ）

#### 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。

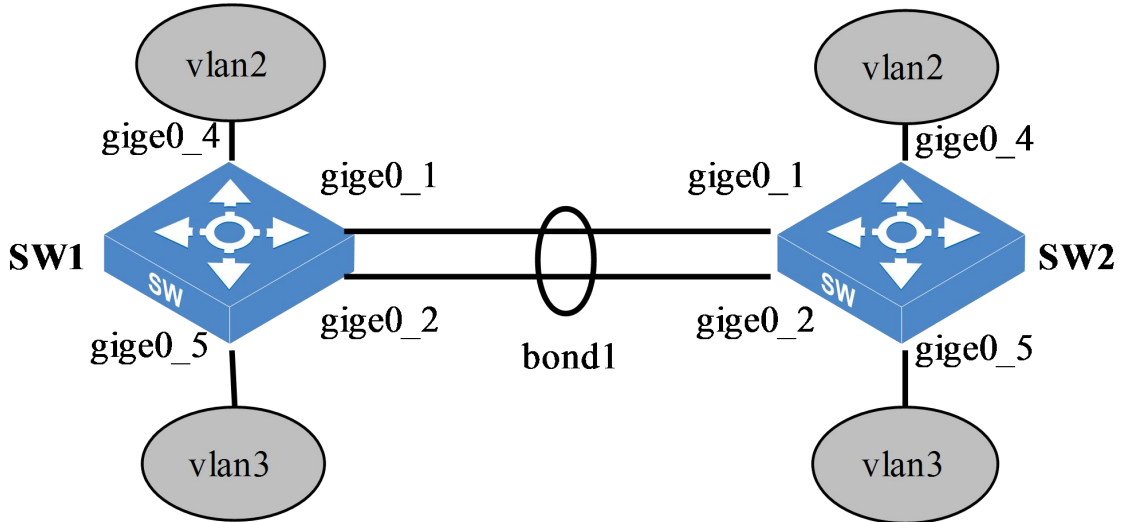
## 3.2 动态端口聚合配置案例

### 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，以实现对其监控。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
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

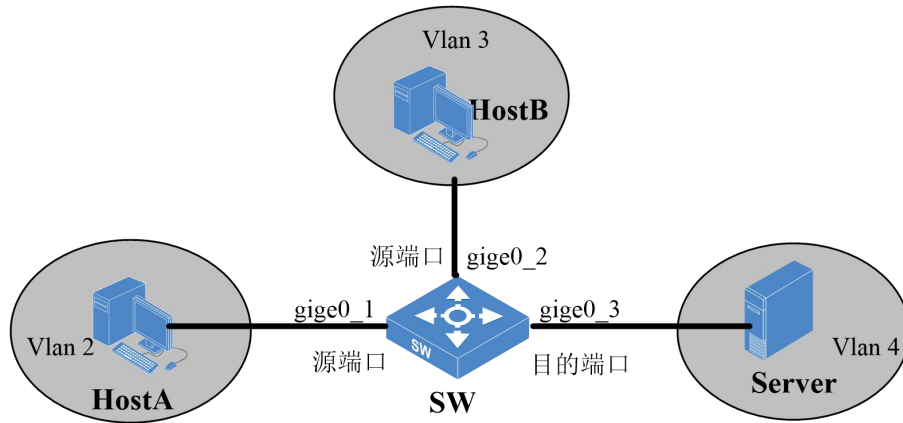
### 4.2.1 配置需求

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-----
Slot 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

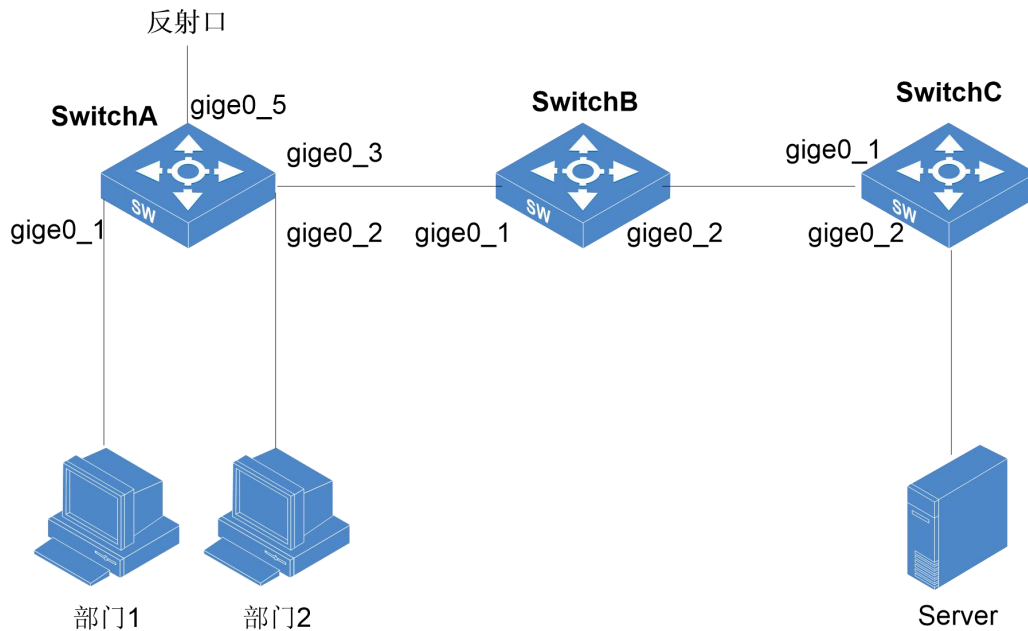
### 4.3.1 配置需求

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

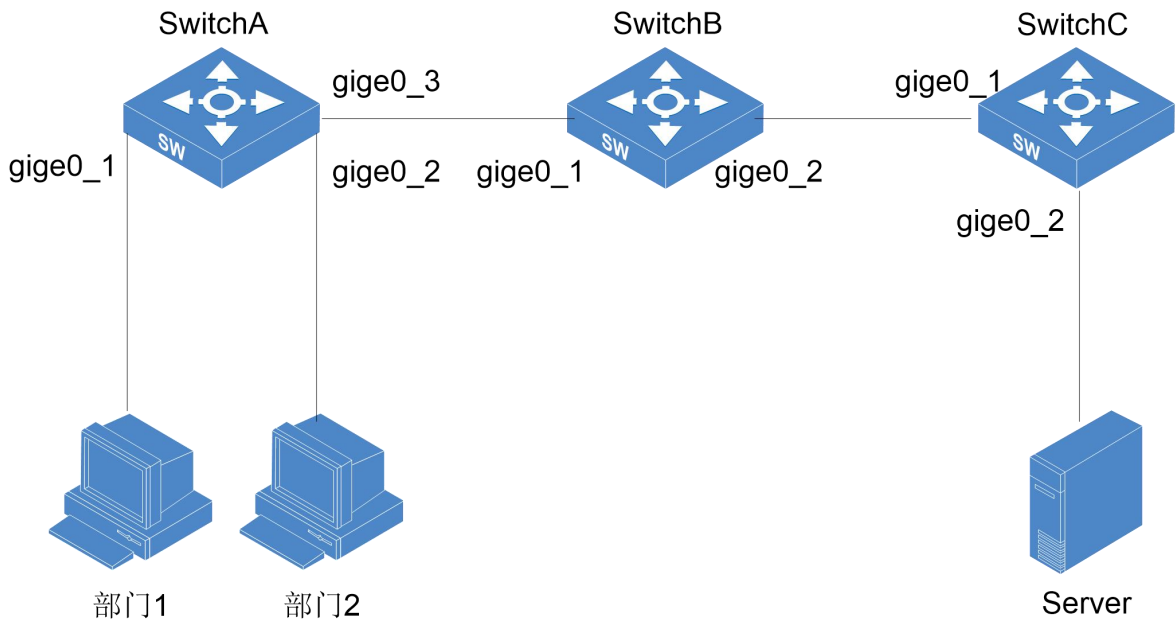
某公司内部通过交换机实现各部门之间互连，网络环境描述如下：

- 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。

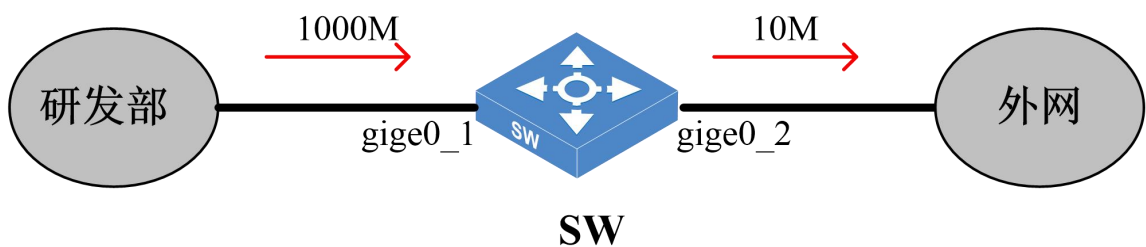
## 5.2 配置案例

### 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。

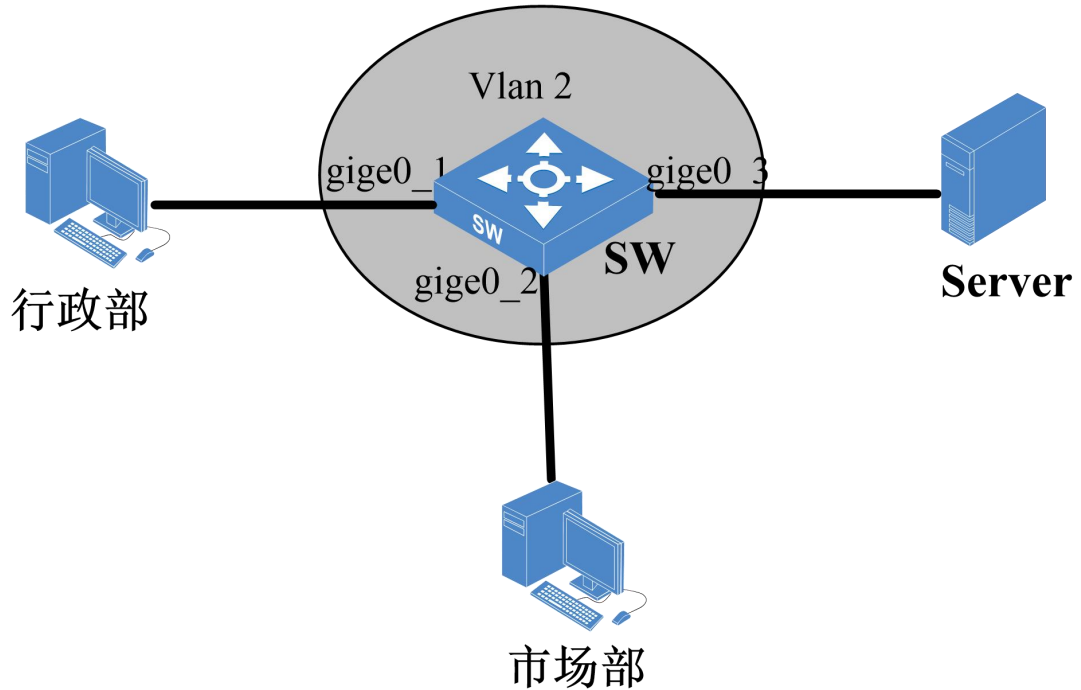
## 6.2 配置案例

### 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 MAC/IP/端口绑定典型配置案例

## 7.1 MAC/IP/端口绑定简介

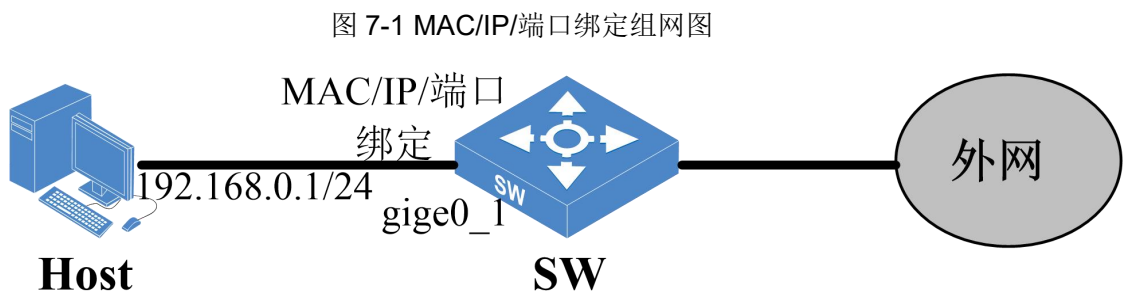
MAC/IP/端口绑定是指将用户主机的 MAC 地址和 IP 地址与所连接的交换机端口进行绑定。该端口收到的报文匹配绑定项，就转发，否则将丢弃报文。这样可以防止用户随意修改主机 IP 地址，导致管理不便的问题。

## 7.2 配置案例

### 7.2.1 配置需求

公司想要某些员工使用固定的 IP 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。

### 7.2.2 网络拓扑



### 7.2.3 配置流程

- (1) SW 上配置主机 MAC 地址和 IP 地址与 gige0\_1 端口绑定，匹配该绑定项的报文转发，否则丢包。
- (2) 验证配置。



## 7.2.4 配置步骤

- (1) SW1 上配置主机 MAC 地址和 IP 地址与 gige0\_1 端口绑定，匹配该绑定项的报文转发，否则丢包。

```
[INSPUR]acl mode mac-ipv4 ingress
[INSPUR-acl-mac-ipv4-ingress]rule 1 source-mac 00:24:AC:be:00:01 source-ipv4
192.168.0.1 interface gige0_1 action permit
[INSPUR-acl-mac-ipv4-ingress]rule 2 interface gige0_1 action drop
```

- (2) 验证配置。

Host 用原来的 IP 地址并且保持与 SW1 的 gige0\_1 接口连接，能访问外网。Host 修改 IP 地址或改变与 SW 相连的接口时，不能访问外网，则表明 MAC/IP/端口绑定生效。

# 8 PVLAN 典型配置案例

## 8.1 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 8-1 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

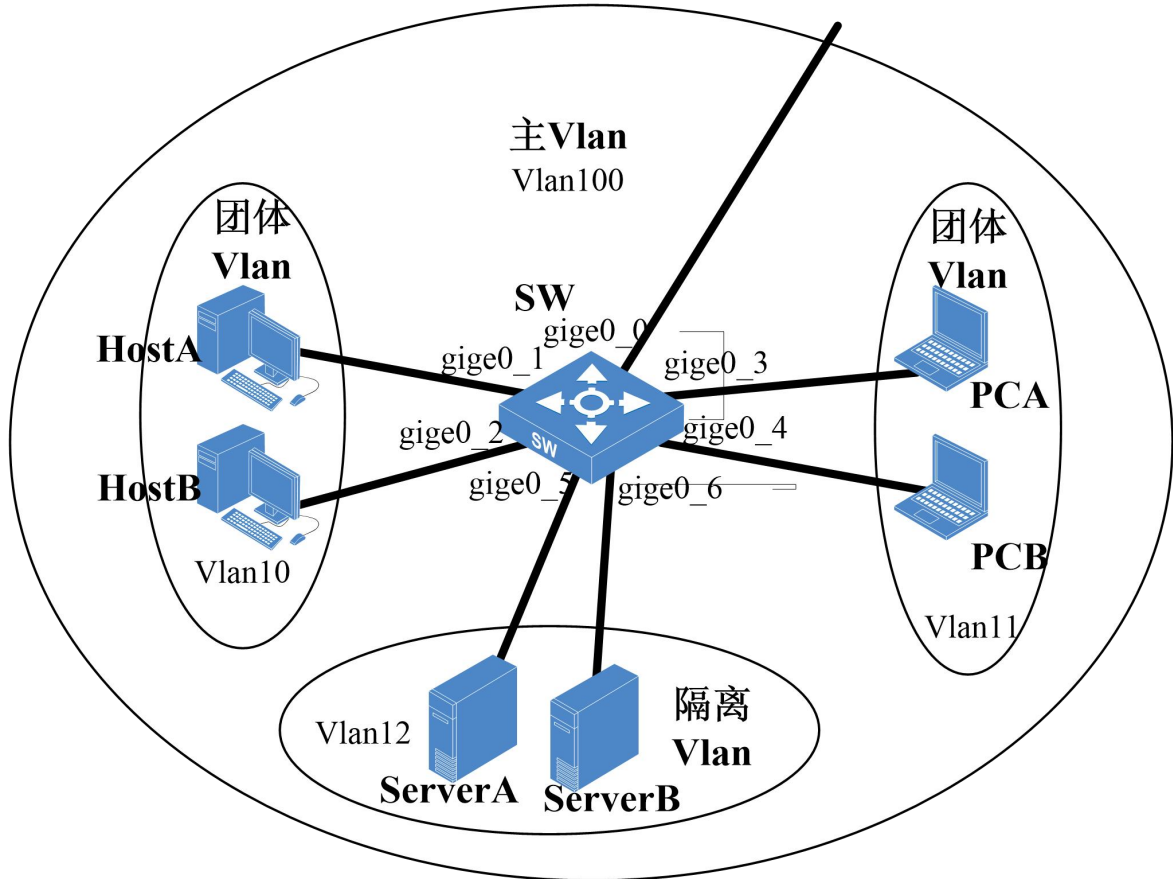
## 8.2 配置案例

### 8.2.1 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。

## 8.2.2 网络拓扑

图 8-2 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。

## 8.2.3 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。

## 8.2.4 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。

# 9 QinQ 典型配置案例

## 9.1 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 9-1 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。

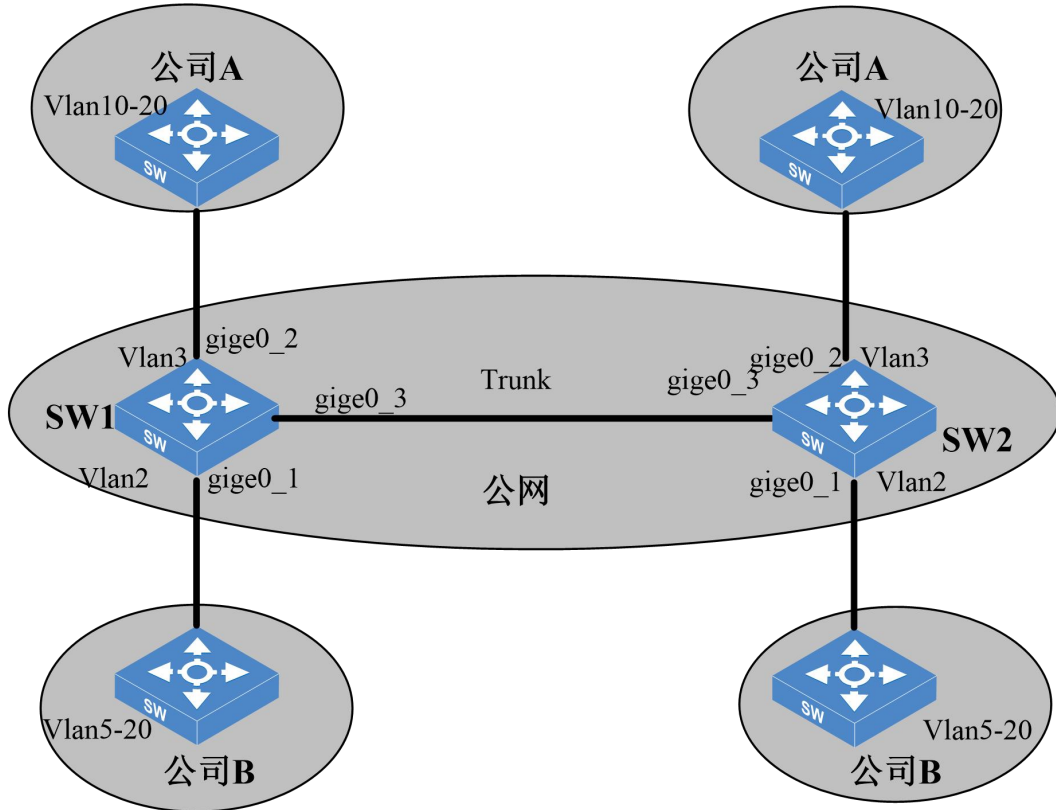
## 9.2 基本 QinQ 配置案例

### 9.2.1 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。

## 9.2.2 网络拓扑

图 9-2 基本 QinQ 组网图



## 9.2.3 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。

## 9.2.4 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。

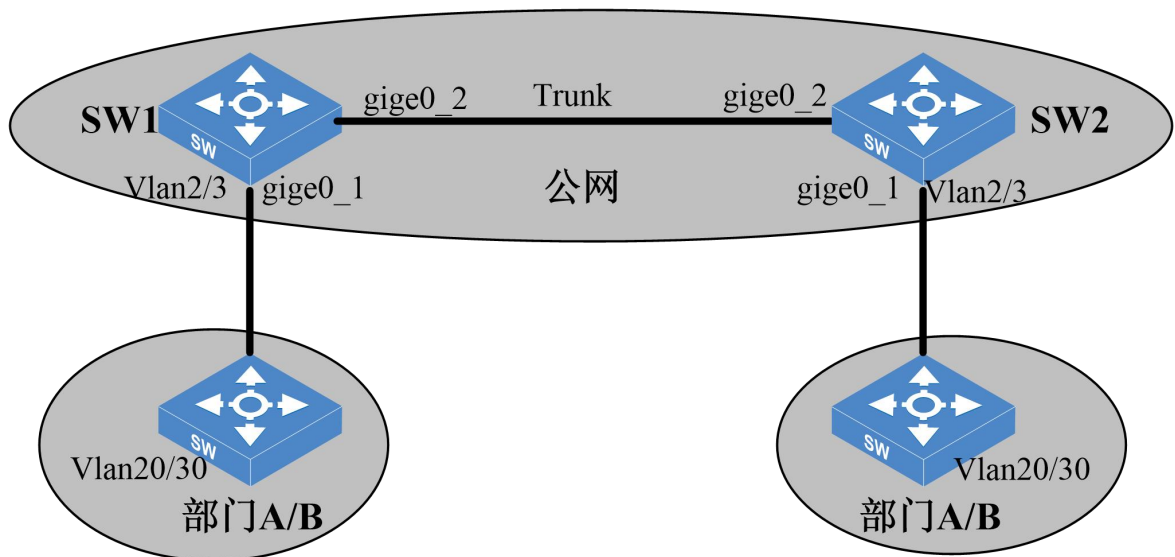
## 9.3 灵活 QinQ 配置案例

### 9.3.1 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。

### 9.3.2 网络拓扑

图 9-3 灵活 QinQ 组网图



### 9.3.3 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。



### 9.3.4 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。

# 10 ARP 防护典型配置案例

## 10.1 ARP 防护简介

### 10.1.1 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 10-1 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。

### 10.1.2 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 10-2 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，就认为该用户非法，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。

### 10.1.3 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。

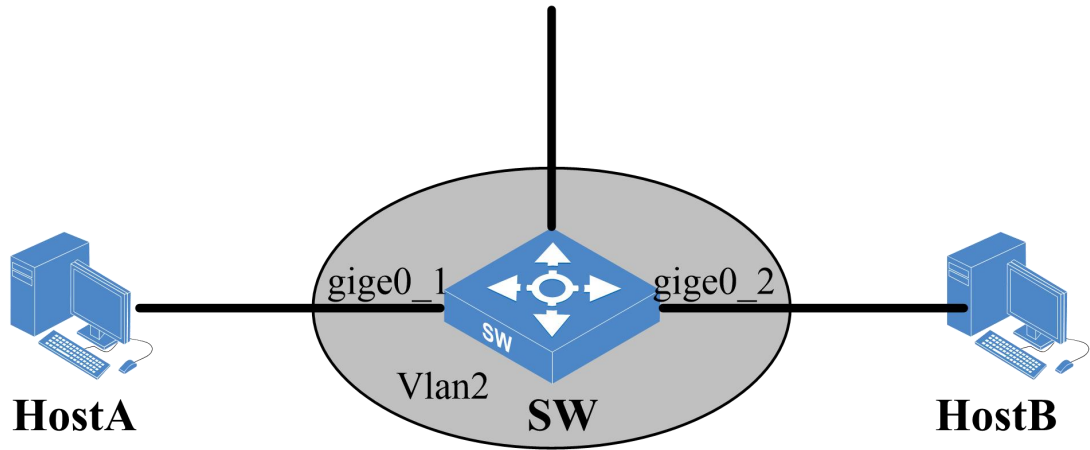
## 10.2 ARP 报文一致性检测配置案例

### 10.2.1 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。

## 10.2.2 网络拓扑

图 10-2 ARP 报文一致性检测组网图



## 10.2.3 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。

## 10.2.4 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。

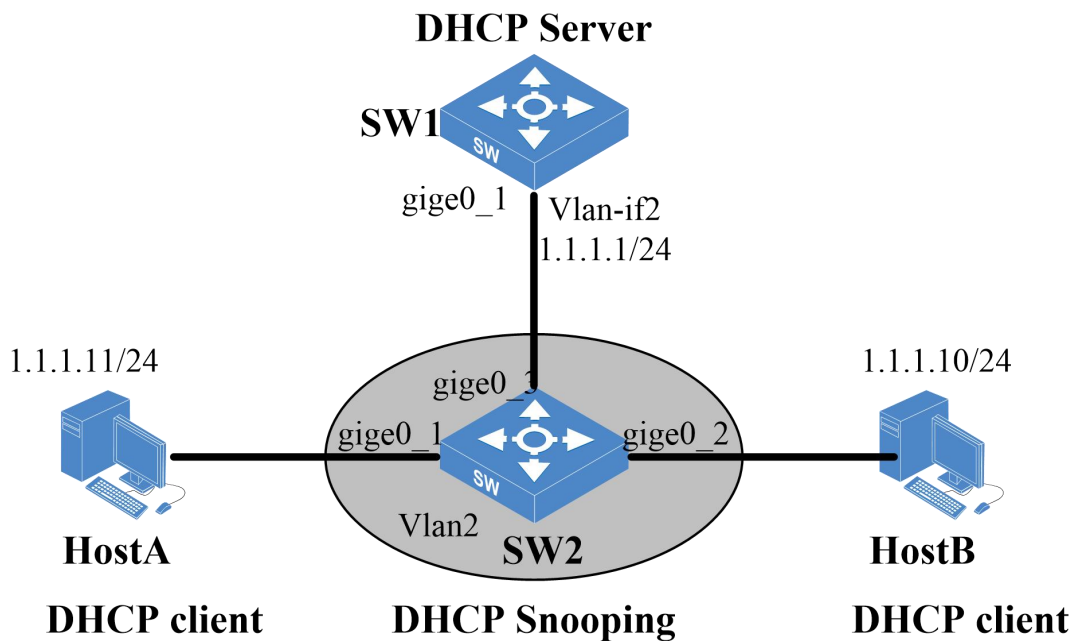
## 10.3 ARP 用户合法性配置案例

### 10.3.1 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。

### 10.3.2 网络拓扑

图 10-3 ARP 用户合法性检测组网图



### 10.3.3 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。

### 10.3.4 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]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 端口收到 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。

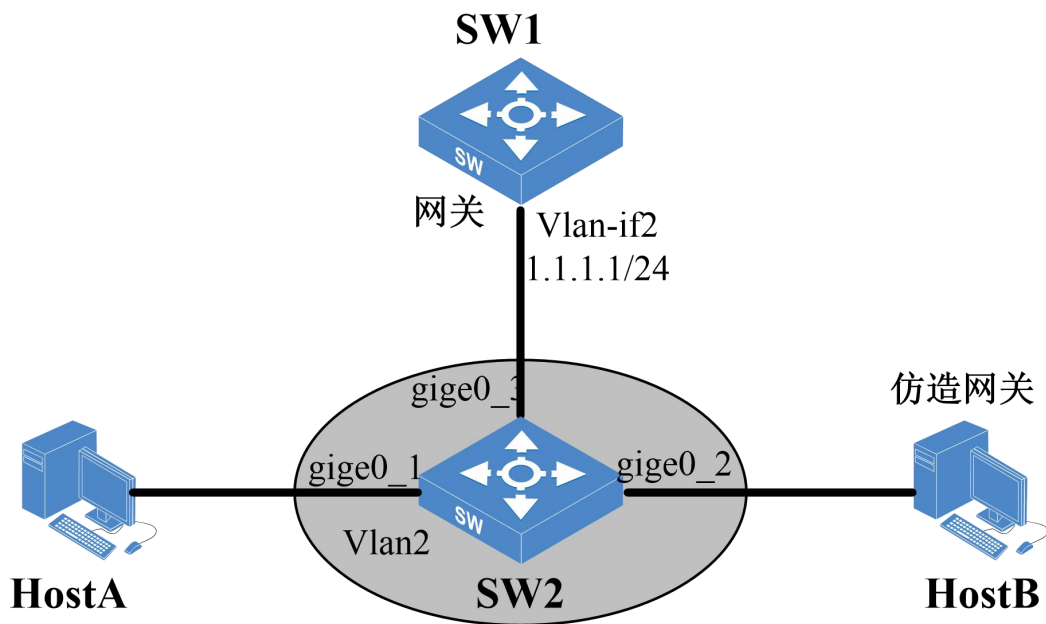
## 10.4 ARP 网关保护配置案例

### 10.4.1 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。

### 10.4.2 网络拓扑

图 10-4 ARP 网关保护组网图



### 10.4.3 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。

## 10.4.4 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。



# 11 路由协议典型配置案例

## 11.1 路由协议简介

### 11.1.1 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。

### 11.1.2 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。

### 11.1.3 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。

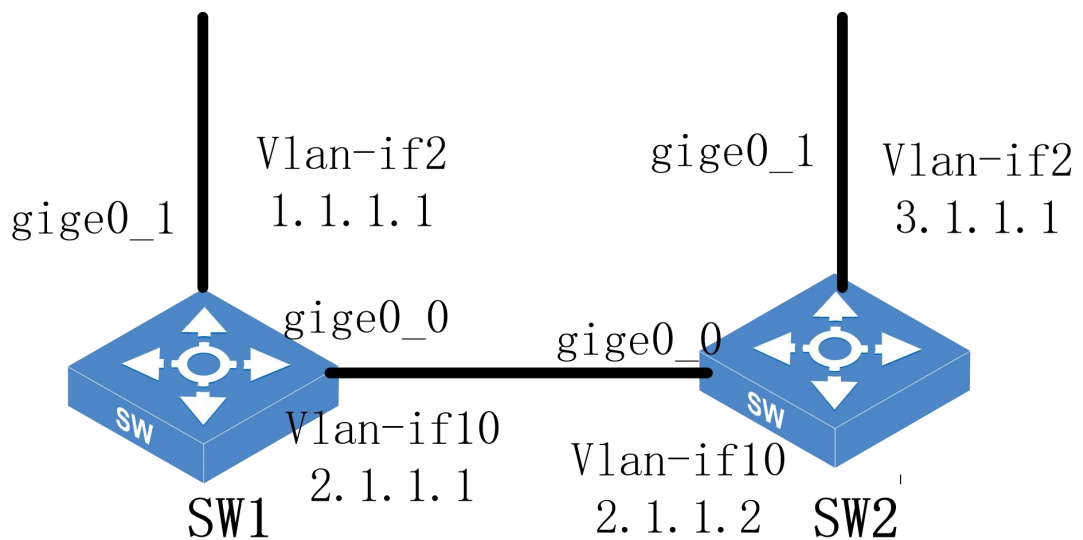
## 11.2 静态路由配置案例

### 11.2.1 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。

### 11.2.2 网络拓扑

图 11-1 静态路由组网图



### 11.2.3 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。

### 11.2.4 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

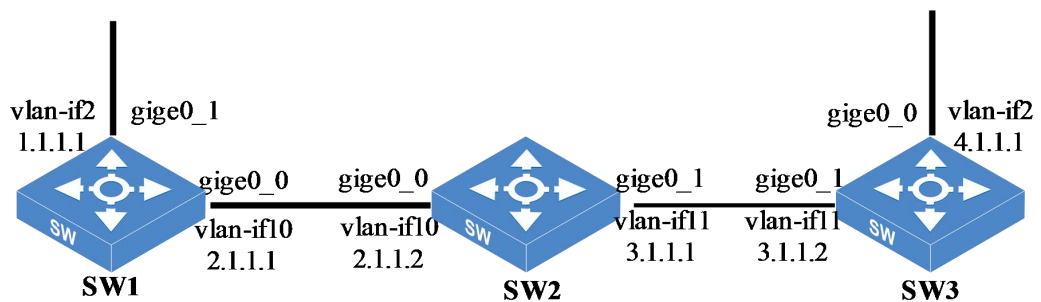
## 11.3 RIP 路由配置案例

### 11.3.1 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。

### 11.3.2 网络拓扑

图 11-2 RIP 路由组网图



### 11.3.3 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。

### 11.3.4 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

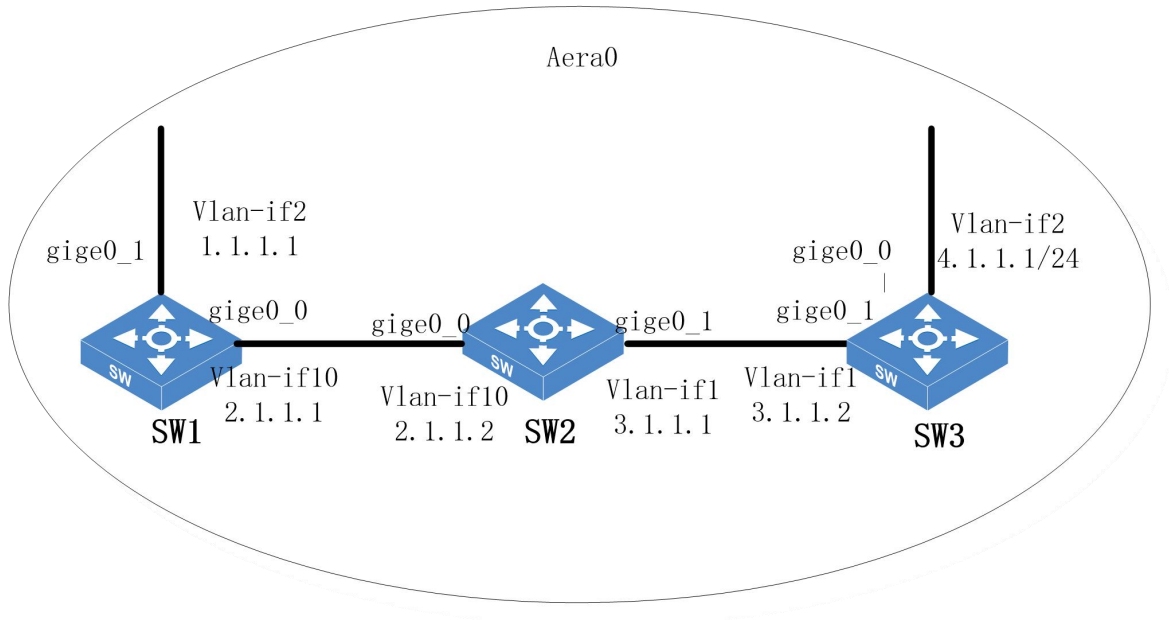
## 11.4 OSPF 典型配置案例

### 11.4.1 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。

## 11.4.2 网络拓扑

图 11-3 OSPF 路由组网图



## 11.4.3 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。

## 11.4.4 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

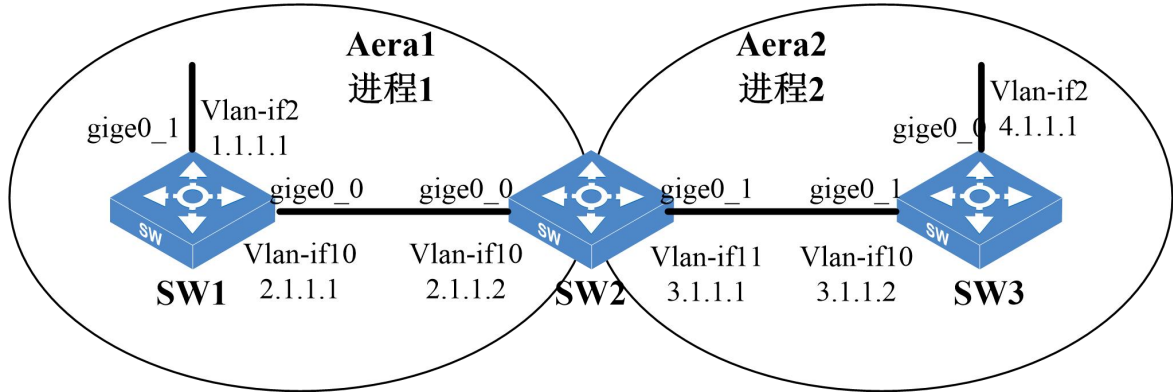
## 11.5 OSPF 多进程典型配置案例

### 11.5.1 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。

## 11.5.2 网络拓扑

图 11-4 OSPF 多进程组网图



## 11.5.3 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。

## 11.5.4 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
```

# 12 DHCP 典型配置案例

## 12.1 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, 当客户端和服务器不在一个网段时, 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 12-1 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。

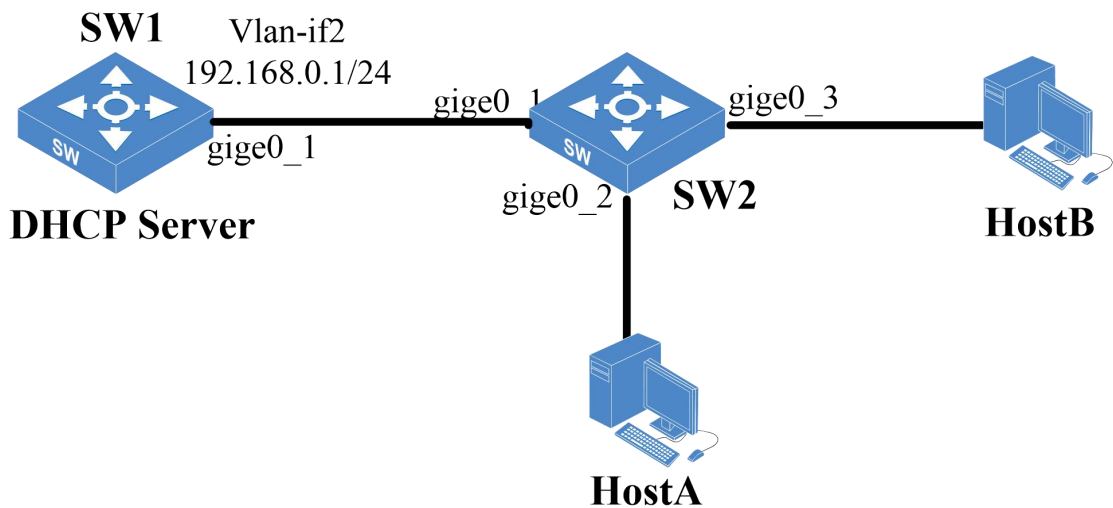
## 12.2 DHCP Server 配置案例

### 12.2.1 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。

### 12.2.2 网络拓扑

图 12-2 DHCP Server 组网图



### 12.2.3 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。

### 12.2.4 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。

## 12.3 DHCP 中继配置案例

### 12.3.1 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。

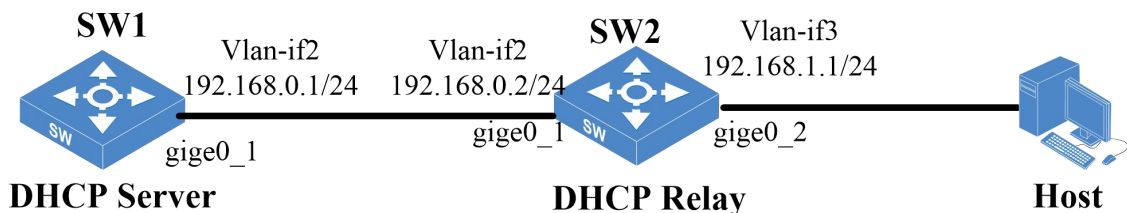


#### 说明

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。

### 12.3.2 网络拓扑

图 12-3 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。

### 12.3.3 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。

### 12.3.4 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。

## 12.4 DHCP Snooping 配置案例

### 12.4.1 配置需求

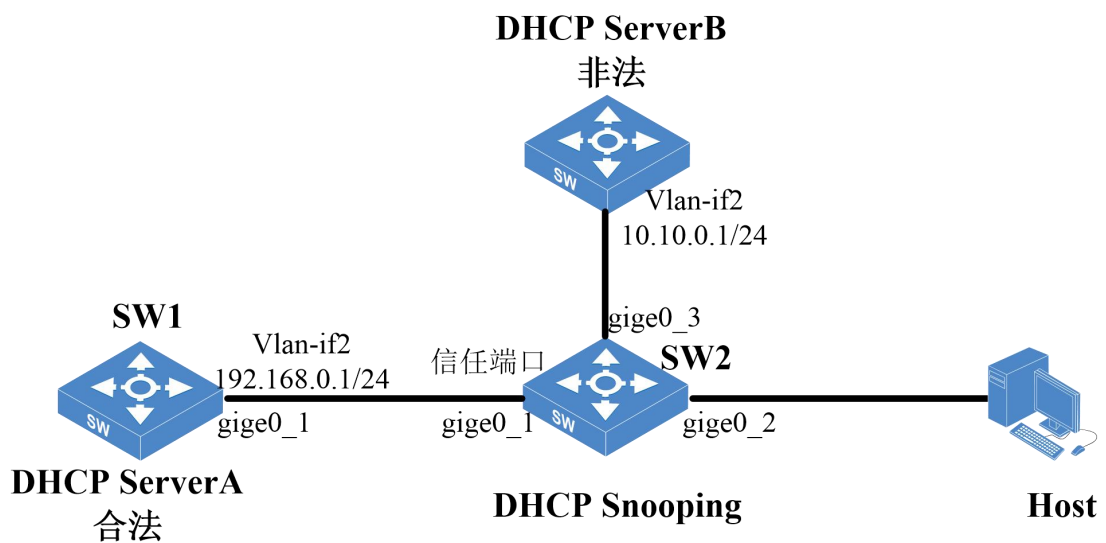
DHCP Snooping 是一种安全手段，当网络中存在非法的 DHCP 服务器时，客户端就可能获取到非法服务器的地址，导致网络不通。为了保证 DHCP 客户端能从合法的 DHCP 服务器上获取地址，用户可以在合法 DHCP 服务器和客户端之间的设备上启用 DHCP Snooping 功能，合法服务器和 DHCP Snooping 连接的端口设置为信任端口，其余端口均为非信任端口。当客户端获取到地址后，启用 DHCP snooping 的设备上能记录用户的 IP 地址和 MAC 地址的对应关系。

## 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。

## 12.4.2 网络拓扑

图 12-4 DHCP Snooping 组网图



## 12.4.3 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。

## 12.4.4 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
```

# 13 QoS 典型配置案例

## 13.1 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 13-1 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 13-2 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

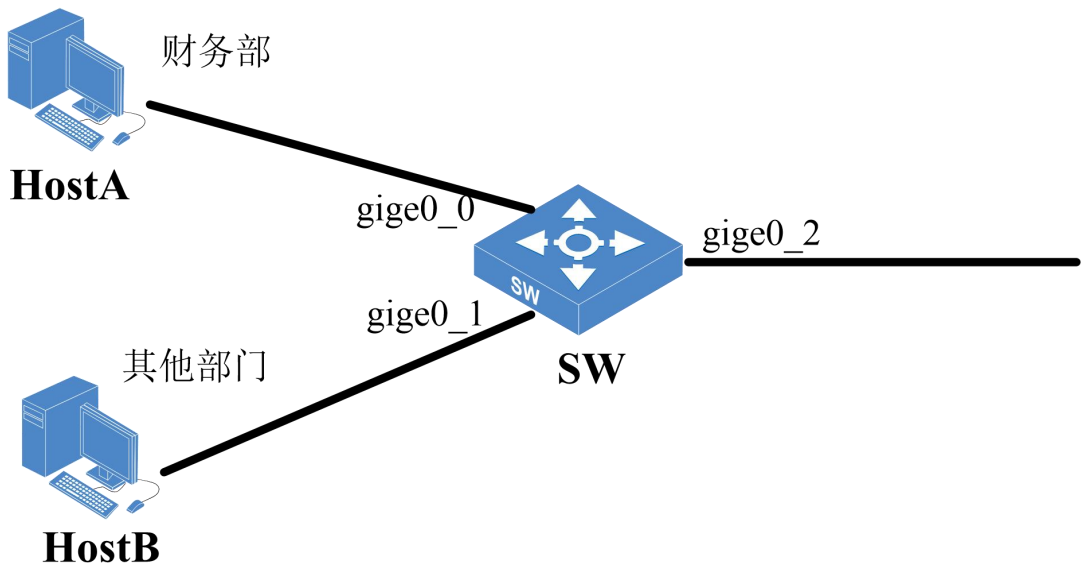
## 13.2 配置案例

### 13.2.1 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。

### 13.2.2 网络拓扑

图 13-2 QoS 组网图



### 13.2.3 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。

### 13.2.4 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。

# 14 802.1x 典型配置案例

## 14.1 802.1x 简介

随着移动办公及驻地网运营等应用的大规模发展，服务提供者需要对用户的接入进行控制和配置。尤其是 WLAN 的应用和 LAN 接入在电信网上大规模开展，有必要对端口加以控制以实现用户级的接入控制，802.1x 就是 IEEE 为了解决基于端口的接入控制 (Port-Based Network Access Control) 而定义的一个标准。

### 14.1.1 基本概念

802.1x 协议是基于 Client/Server 的访问控制和认证协议，也是一种基于端口的网络接入控制协议。客户端接入端口通过认证后可以访问外部资源。

### 14.1.2 认证方式

图 14-1 认证方式

项目	说明
本地认证	使用本地用户名和密码进行认证
Radius 认证	使用 Radius 方式认证，在 Radius 服务器上配置用户名和密码
LDAP 认证	使用 LDAP 认证方式认证

### 14.1.3 端口接入控制模式

图 14-2 端口接入控制模式

项目	说明
强制授权模式	该端口不需做认证就可正常访问外部资源
强制未授权模式	该端口不能做认证，不能访问外部资源
基于端口模式	该端口下只需一个客户端认证，其他客户端可正常访问外部资源

项目	说明
基于 MAC 模式	该端口下所用接入客户端都需要认证，才能访问外部资源，设备默认为 MAC 模式

## 14.1.4 Radius 认证分类

图 14-3 Radius 认证分类

项目	说明
Radius 中继方式 (Relay)	将 EAP 承载在其它高层协议中，如 EAP over Radius，以便扩展认证协议报文穿越复杂的网络到达认证服务器
Radius 终结方式 (End)	将 EAP 报文在设备端终结并映射到 Radius 报文中，利用标准 Radius 协议完成认证、授权和计费

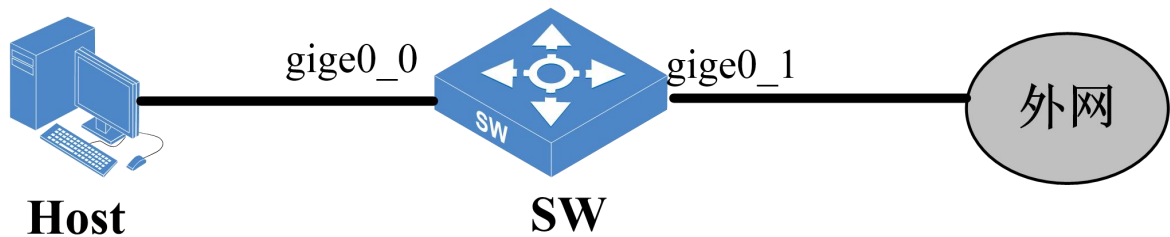
## 14.2 802.1x 本地认证配置案例

### 14.2.1 配置需求

某公司需要限制员工和访客访问内部资源及外网，员工必须通过帐号认证，才能正常访问资源和通信，访客没有帐号不能访问公司内部资源和外网等。

### 14.2.2 网络拓扑

图 14-4 802.1x 组网图



### 14.2.3 配置流程

- (1) 在 SW 上全局开启 802.1x 认证功能。
- (2) 在 SW 上选择本地认证方式，并在本地配置用户名和密码。

- (3) 在端口 `gige0_0` 上使能 802.1x 认证。
- (4) 验证配置。

## 14.2.4 配置步骤

- (1) 在 SW 上全局开启 802.1x 认证功能

```
<INSPUR>conf-mode
[INSPUR]dot1x enable
[INSPUR]
```

- (2) 在 SW 上选择本地认证方式，并在本地配置用户名和密码。

```
[INSPUR] dot1x auth-method local
[INSPUR] dot1x local-user test123
[INSPUR-luser-test123]password cipher test123456
[INSPUR]
```

- (3) 在端口 `gige0_0` 上使能 802.1x 认证。

```
[INSPUR] interface gige 0_0
[INSPUR-gige0_0]dot1x enable
```

- (4) 验证配置。

在 Host 上安装 802.1x 认证的客户端，使用相应的用户名和密码进行拨号认证，认证通过后，Host 能访问外网。进入用户视图使用命令行 `show dot1x online-users` 可以查看在线用户。

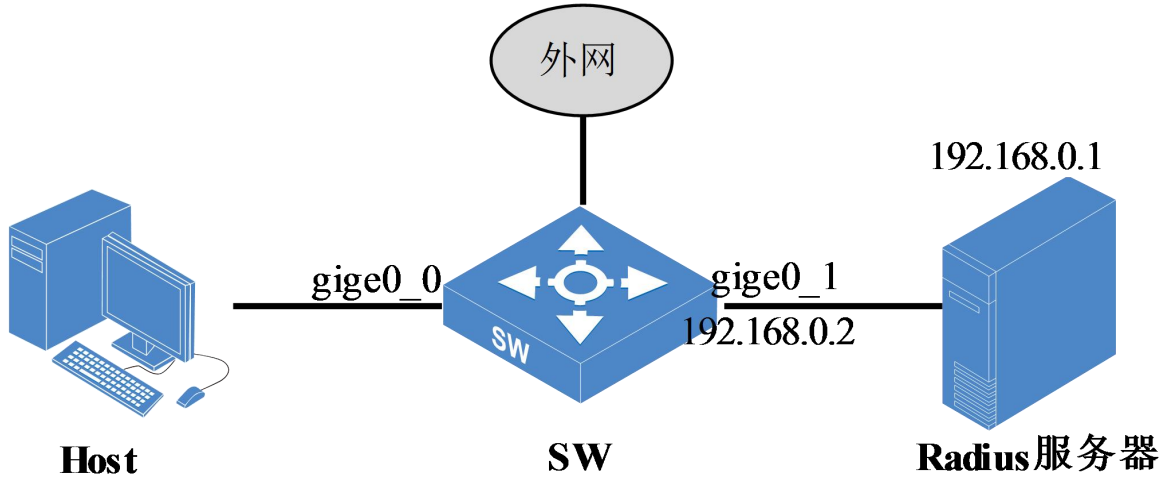
## 14.3 802.1x Radius 认证配置案例

### 14.3.1 配置需求

某公司需要限制员工和访客访问内部资源及外网，员工必须通过帐号认证，才能正常访问资源和通信，访客没有帐号不能访问公司内部资源和外网等。

## 14.3.2 网络拓扑

图 14-5 802.1x Radius 认证组网图



## 14.3.3 配置流程

- (1) 在 SW 上全局开启 802.1x 认证功能。
- (2) 在 SW 上配置 Radius 认证方式，使用中继认证过程，配置 Radius 服务器。
- (3) 在端口 gige0\_0 上使能 802.1x 认证。
- (4) 配置 Radius 服务器信息。
- (5) 验证配置。

## 14.3.4 配置步骤

- (1) 在 SW 上全局开启 802.1x 认证功能。

```
<INSPUR>conf-mode
[INSPUR]dot1x enable
[INSPUR]
```

- (2) 在 SW 上配置 Radius 认证方式，使用中继认证过程，配置 Radius 服务器。

```
[INSPUR] dot1x auth-method radius relay
[INSPUR] dot1x radius-server primary 192.168.0.1 nas-ip 192.168.0.2 key
test123 port 1812
```

(3) 在端口 `gige0_0` 上使能 802.1x 认证。

```
[INSPUR] interface gige0_0  
[INSPUR-gige0_0]dot1x enable
```

(4) 配置 Radius 服务器信息。key 配置成 `test123`。

(5) 验证配置。

在 Host 上安装 802.1x 认证的客户端，使用相应的用户名和密码进行拨号认证，认证通过后，Host 能访问外网，进入用户视图使用命令 `show dot1x online-users` 可以查看在线用户。

# 15 MAC 认证典型配置案例

## 15.1 MAC 地址认证简介

MAC 地址认证是一种对用户网络访问权限进行控制的认证方法，基于端口和 MAC 地址进行认证，不需要用户安装任何客户端软件。设备在首次检测到用户的 MAC 地址以后，即启动对该用户的认证操作。认证过程中，也不需要用户手动输入用户名或者密码。用户认证通过后，就可以访问网络资源，如果用户认证失败，将会被添加为静默 MAC，在静默时间内设备不会处理该用户的认证报文，待静默时间超后又重新认证。

### 15.1.1 认证类型

图 15-1 认证类型

项目	说明
本地认证	使用本地配置的用户名和密码进行认证
Radius 认证	使用 Radius 服务器配置用户名和密码进行认证

### 15.1.2 认证用户名格式

图 15-2 认证用户名格式

项目	说明
MAC 地址用户名	使用用户的 MAC 地址作为认证的用户名和密码
固定用户名	不论用户的 MAC 地址为何值，所有用户均使用在设备上预先配置的用户名和密码进行认证。



### 说明

使用 MAC 地址用户名进行 Radius 认证时，只需在 Radius 服务器上配置用户名和密码即可；使用固定用户名进行 Radius 认证时需要在本地和 Radius 服务器上同时配置用户名和密码。

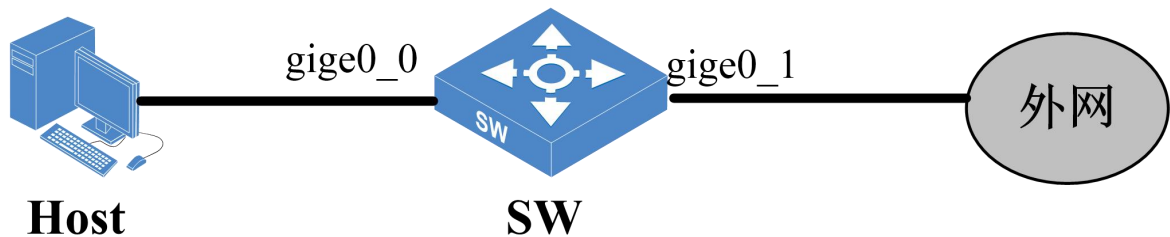
## 15.2 MAC 地址本地认证配置案例

### 15.2.1 配置需求

某公司需要限制访客连接外网，而且对员工需要透明操作，可通过 MAC 认证功能，对员工进行 MAC 地址认证，员工无需手动输入用户名密码即可连接外网，这样限制访客用户连接外网。

### 15.2.2 网络拓扑

图 15-3 MAC 地址本地认证组网图



### 15.2.3 配置流程

- (1) 在 SW 上全局开启 MAC 认证功能。
- (2) 在 SW 上配置本地认证方式，使用固定用户名进行认证，并在本地配置用户名和密码。
- (3) 在端口 gige0\_0 上使能 MAC 地址认证。
- (4) 验证配置。

### 15.2.4 配置步骤

- (1) 在 SW 上全局开启 MAC 认证功能。

```
[INSPUR]mac-authentication enable
```



(2) 在 SW 上配置本地认证方式，使用固定用户名进行认证，并在本地配置用户名和密码。

```
[INSPUR]mac-authentication auth-method local
[INSPUR]mac-authentication auth-username fixed
[INSPUR]mac-authentication local-user zhangsan
[INSPUR-luser-zhangsan]password 123456
[INSPUR-luser-zhangsan]mac-address 11:11:11:11:11:11
[INSPUR-luser-zhangsan]exit
```

(3) 在端口 gige0\_0 上使能 MAC 地址认证。

```
[INSPUR]interface gige0_0
[INSPUR-gige0_0] mac-authentication enable
```

(4) 验证配置。

在 Host 无需安装客户端进行认证，认证通过后 Host 能访问外网。在配置视图下使用 **show mac-authentication access-user** 可以查看在线用户。

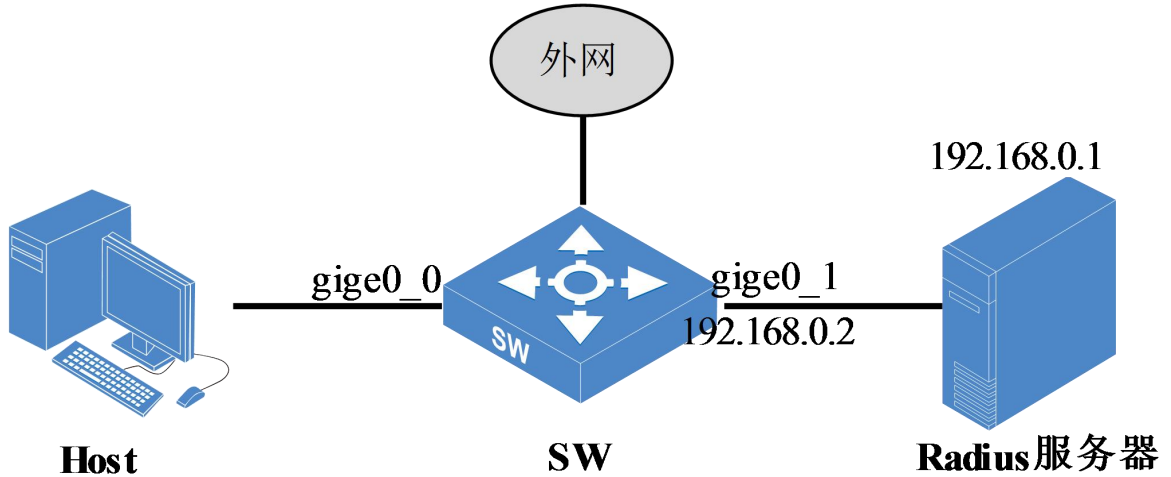
## 15.3 MAC 地址 Radius 认证配置案例

### 15.3.1 配置需求

某公司需要限制访客连接外网，而且对员工需要透明操作，可通过 MAC 认证功能，对员工进行 MAC 地址认证，员工无需手动输入用户名密码即可连接外网，这样限制访客用户连接外网。

## 15.3.2 网络拓扑

图 15-4 MAC 地址 Radius 认证组网图



## 15.3.3 配置流程

- (1) 在 SW 上全局开启 MAC 认证功能。
- (2) 在 SW 上配置 Radius 认证方式，使用 MAC 地址用户名进行认证，配置 Radius 服务器。
- (3) 在端口 gige0\_0 上使能 MAC 地址认证。
- (4) 配置 Radius 服务器信息。
- (5) 验证配置。

## 15.3.4 配置步骤

- (1) 在 SW 上全局开启 MAC 认证功能。

```
[INSPUR]mac-authentication enable
```

- (2) 在 SW 上配置 Radius 认证方式，使用 MAC 地址用户名进行认证，配置 Radius 服务器。

```
[INSPUR]mac-authentication auth-method radius
[INSPUR]mac-authentication auth-username mac
[INSPUR]mac-authentication radius-server 192.168.0.1 key test123
[INSPUR]mac-authentication local-user f0:de:f1:ea:7f:5e
[INSPUR-luser-f0:de:f1:ea:7f:5e]password simple f0:de:f1:ea:7f:5e
[INSPUR-luser-f0:de:f1:ea:7f:5e]mac-address f0:de:f1:ea:7f:5e
```

(3) 在端口 `gige0_0` 上使能 MAC 地址认证。

```
[INSPUR] interface gige0_0  
[INSPUR-gige0_0]mac-authentication enable
```

(4) 配置 Radius 服务器信息。

(5) 验证配置。

Host 无需添加客户端进行认证，认证通过后能访问外网，进入用户视图使用命令 `show macauth users` 可以查看在线用户。

# 16 Portal 典型配置案例

## 16.1 Portal 认证简介

Portal 认证是一种限制用户上网的方式,有 Web 认证和终端认证两种方式。使用 Web 认证方式时,用户上网打开网页,便会被推送到特定的认证页面,用户需要输入用户名和密码,认证成功后才能够正常上网;使用终端认证方式时,用户使用客户端输入相应的用户名和密码,认证成功后才能够访问外网。

我司的 Portal 认证是通过下发 Portal-acl 来控制认证的,Portal-acl 里可配置用户源 IP 地址、目的 IP 地址、端口及动作(认证或通过)。这样就可以灵活控制源 IP 为某个网段的用户进行认证(或通过),也可以配置去往某些目的 IP 网段的用户进行认证(或通过)。

### 16.1.1 Portal 认证模式及方式

表 16-1 Portal 认证模式及方式

项目	说明
Web 认证	用户使用网页输入用户名密码进行上网
终端认证	用户使用客户端软件输入用户名密码登录上网
本地认证方式	用户使用设备上的用户名密码登录
Radius 认证方式	用户使用 Radius 服务器上的用户名密码登录

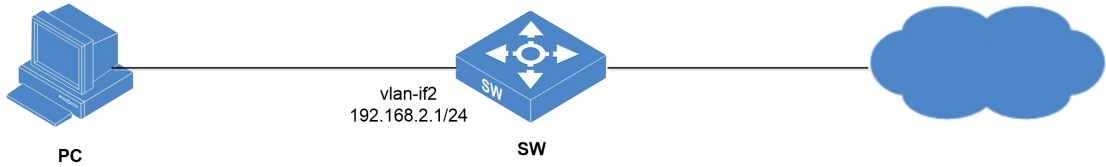
## 16.2 Portal 认证配置案例

### 16.2.1 配置需求

某公司需要限制员工上网,有用户名密码账号的员工可以认证上网,没有账号的不能够上网,员工可以使用客户端登录认证。

## 16.2.2 网络拓扑

图 16-2 Web 认证组网图



## 16.2.3 配置流程

- (1) 在 SW 上配置 vlan 及 ip 地址。
- (2) 在 SW 上配置 Portal 认证策略。
- (3) 在 SW 上添加本地用户。
- (4) 验证配置。

## 16.2.4 配置步骤

- (1) 在 SW 上添加 vlan2，划入端口及配置 vlan-if IP 地址。

```
[INSPUR]vlan 2
[INSPUR-vlan2]interface vlan-if 2
[INSPUR-vlan-if2]ip address 192.168.2.1/24
[INSPUR-vlan-if2]interface gige 0_2
[INSPUR-gige0_2]switchport access vlan 2
[INSPUR-gige0_2]
```

- (2) 在 SW 上配置 Portal 认证策略。

```
[INSPUR] portal policy Default source-address 192.168.2.0/24;
destination-address any action auth auth-server Local portal-server local
redirect success-page enable
```

- (3) 在 SW 上添加本地用户 test。

```
[INSPUR]user-manage user test
[INSPUR]user-manage user test password enable
[INSPUR]user-manage user test password enable
[INSPUR]user-manage user test password simple 123456
```

(4) 验证配置。

PC 访问外网测试，会推送输入用户名密码页面，输入用户名密码后，可以正常访问网络。

# 17 生成树典型配置案例

## 17.1 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 17-1 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 17-2 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 17-3 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

## 17.2 STP 配置案例

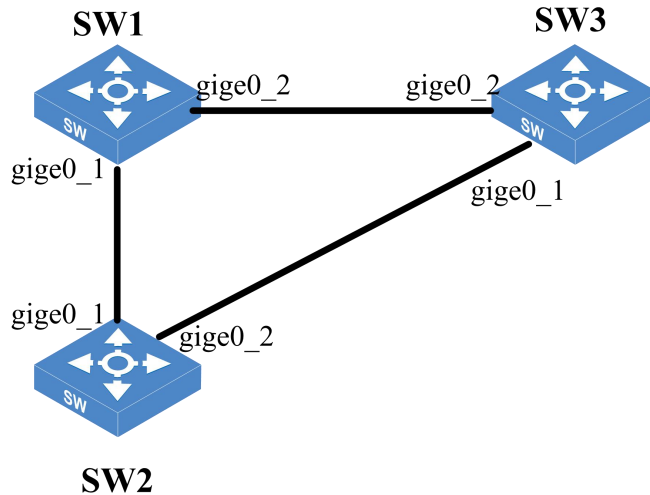
### 17.2.1 配置需求

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。



## 17.2.2 网络拓扑

图 17-4 STP 组网图



## 17.2.3 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。

## 17.2.4 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

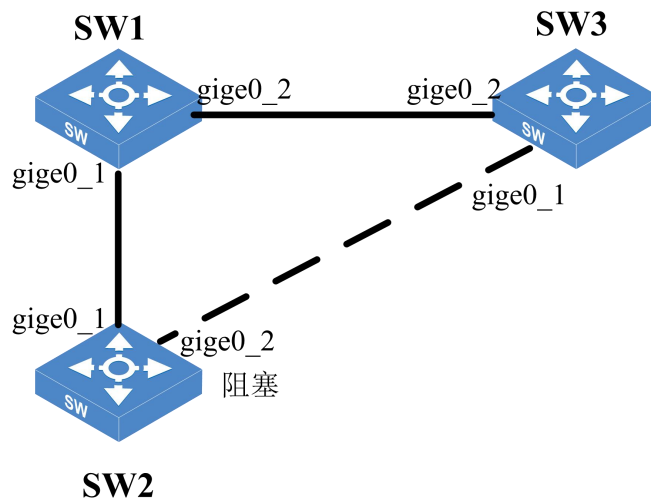
Interface Role Sts Cost Prio P2p
-----
-----
gige0_1 designated forwarding 20000 128 yes
gige0_2 root forwarding 20000 128 yes

<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 17-5 STP 树形图



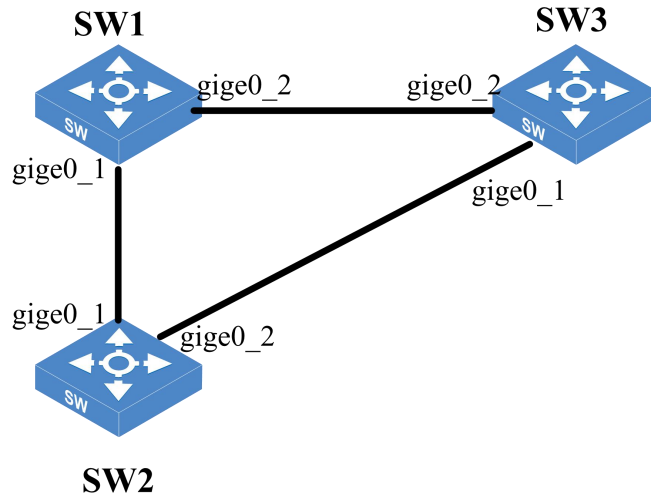
## 17.3 RSTP 配置案例

### 17.3.1 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。

## 17.3.2 网络拓扑

图 17-6 RSTP 组网图



## 17.3.3 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。

## 17.3.4 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。

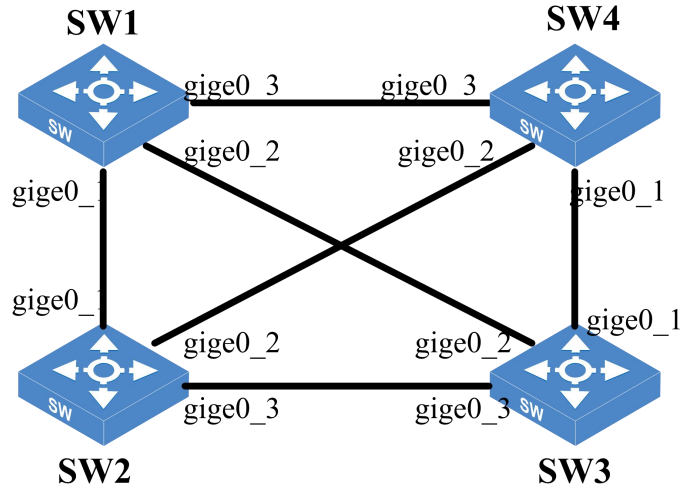
## 17.4 MSTP 配置案例

### 17.4.1 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。

## 17.4.2 网络拓扑

图 17-7 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。

## 17.4.3 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。

## 17.4.4 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
```

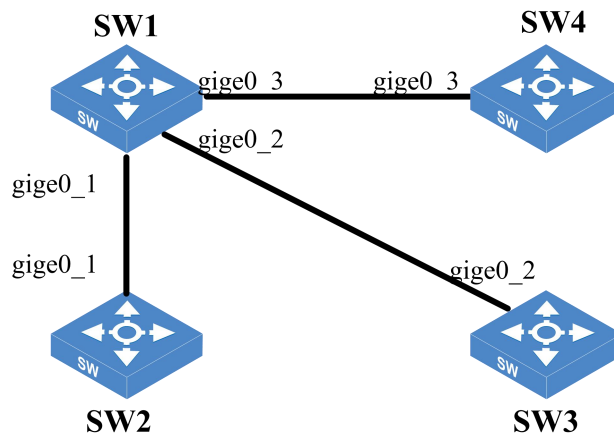
```
Configuration Digest Oxb41829f9 3a54f b74ef7a8 587ff58d
```

```
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>
```

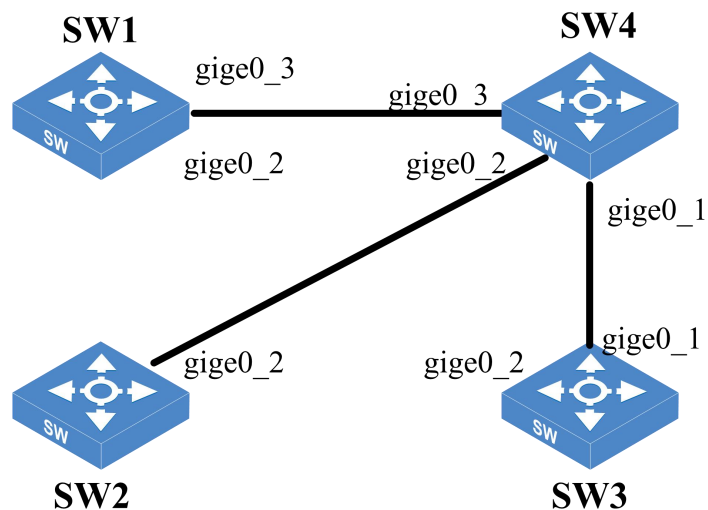
以上结果表明，在实例 1 里，SW1 的优先级最高，作为 MSTP 的树根，其形成的拓扑图如下：

图 17-8 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 17-9 实例 2 树形图





# 18 VRRP 典型配置案例

## 18.1 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。

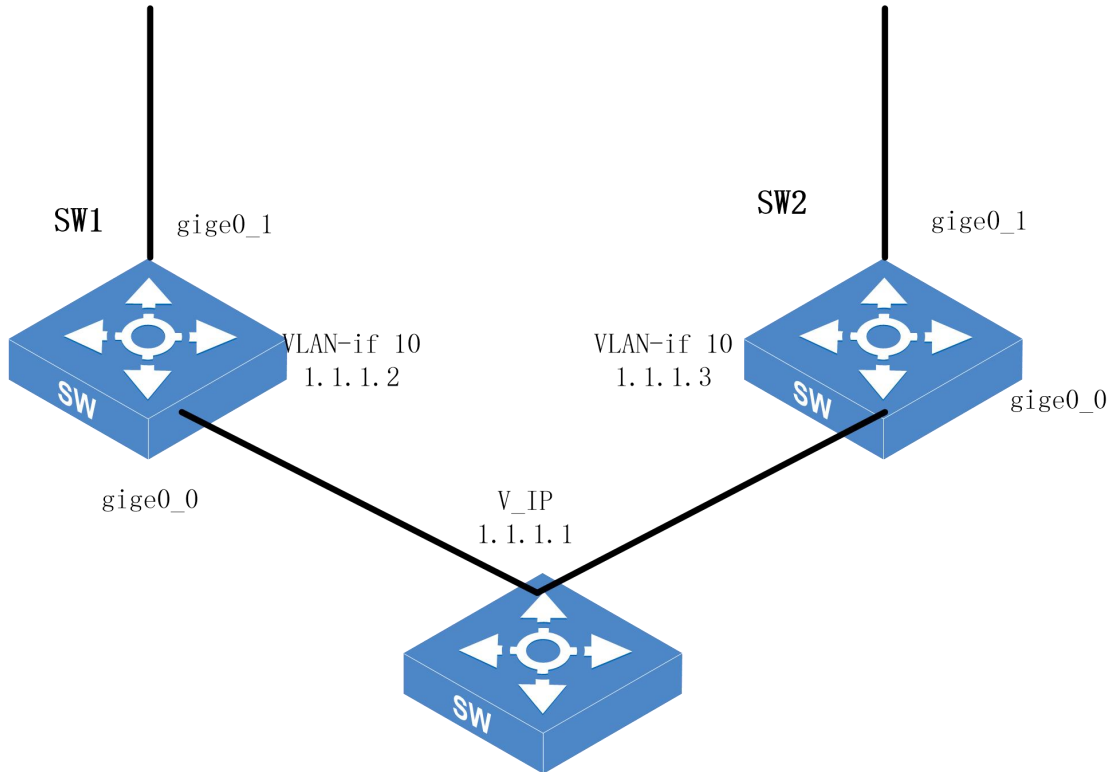
## 18.2 VRRP 配置案例

### 18.2.1 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。

## 18.2.2 网络拓扑

图 18-1 VRRP 组网图



## 18.2.3 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。

## 18.2.4 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，设置优先级及抢占模式

# 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

# 19 VSM 典型配置案例

## 19.1 VSM 简介

VSM（Virtual Switch Matrix，虚拟交换矩阵）是将两台或多台设备通过物理端口连接起来形成一台虚拟的逻辑设备。VSM 中的每台设备都称为成员设备，成员设备的所有配置都保持一致，成员设备按照功能不同可分为两种 Mode：Master 和 Slave

表 19-1 VSM 简介

项目	说明
Master	负责管理整个 VSM
Slave	作为 Master 的备份设备运行

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 19-2 VSM 中的几个概念

项目	说明
VSM 标识	即 VSM ID，在 VSM 中，每台设备都通过 VSM ID 来进行唯一标识，通过 VSM ID 来进行 VSM Mode 的选举，交换机上 VSM ID 值为 0-7。
VSM 级联板	专门用来做 VSM 级联的单板，交换机支持专门的插卡做级联板。
VSM 级联口	VSM 级联板上专用于 VSM 的万兆光口，两台设备通过 VSM 级联口形成 VSM 通道；使用多个物理口做级联时，对接同一台设备的级联口名称必须一致。
VSM 通道	通过 VSM 通道来实现两台设备的通信和跨设备的报文转发。

## 19.2 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；

## 19.3 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 19-3 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。

## 19.4 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。

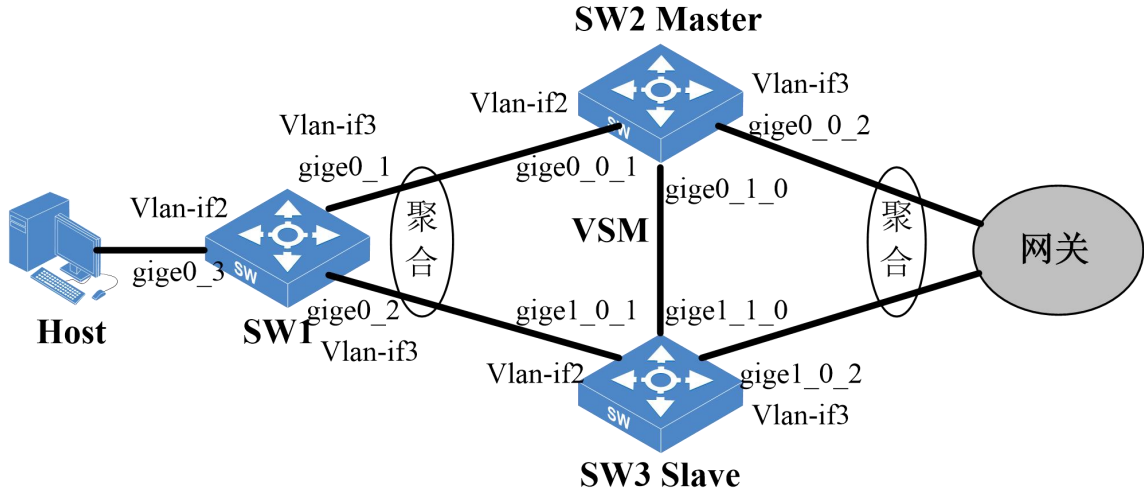
## 19.5 VSM 典型配置案例

### 19.5.1 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。

## 19.5.2 网络拓扑

图 19-2 VSM 组网图



- 开启 VSM 后，设备上显示的端口名为 VSM ID+槽号+端口号，如 gige0\_1\_0。
- 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 吧，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章节。

## 19.5.3 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。

## 19.5.4 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 tengige1_0 downlink-port-list null
The configuration will cause rebooting and take effect after that.
Are you sure? (Y/N) [N]: y
```

# SW3 上的配置

```
<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
```

(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 gige0_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 gige0_1
[INSPUR-vlan3]port gige0_2
[INSPUR-vlan3]exit
[INSPUR]interface vlan-if3
[INSPUR-vlan-if3]ip address 2.2.2.1/24
[INSPUR-vlan-if3]exit
[INSPUR]
```

# 在 SW2 上配置

```
<INSPUR>conf-mode
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_0_1
[INSPUR-vlan2]port gige1_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 gige0_0_2
[INSPUR-vlan3]port gige1_0_2
[INSPUR-vlan3]exit
[INSPUR]interface vlan-if3
[INSPUR-vlan-if3]ip address 3.3.3.1/24
[INSPUR-vlan-if3]exit
[INSPUR]
```

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 gige0_1
[INSPUR-gige0_1]bond group 1
[INSPUR-gige0_1]exit
[INSPUR]interface gige0_2
[INSPUR-gige0_2]bond group 1
[INSPUR-gige0_2]exit
[INSPUR]
```

# 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 gige0_0_1
[INSPUR-gige0_0_1]bond group 1
```

```
[INSPUR-gige0_0_1]exit
[INSPUR]interface gige1_0_1
[INSPUR-gige1_0_1]bond group 1
[INSPUR-gige1_0_1]exit
[INSPUR]
```

#### # 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 gige0_0_2
[INSPUR-gige0_0_2]bond group 1
[INSPUR-gige0_0_2]exit
[INSPUR]interface gige1_0_2
[INSPUR-gige1_0_2]bond group 1
[INSPUR-gige1_0_2]exit
[INSPUR]
```

#### (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。

# 20 ovc 典型配置案例

## 20.1 OVC 简介

OVC（OS-Level Virtual Context，操作系统级虚拟环境）技术是一种将一台物理设备虚拟成多台逻辑设备的虚拟化技术。经过 OVC 虚拟化之后，同一台物理设备上的多个逻辑设备都拥有独立的硬件、软件、转发表项、管理平面和日志，各逻辑设备的运行互不影响。OVC 技术实现了资源和管理虚拟化，物理设备资源虚拟化后，业务的快速部署和调整不再受限于物理设备本身，实现了节约建设和运维成本、灵活按需部署、完全故障隔离等优点，有效地解决了多业务安全隔离和资源按需分配的问题。为网络和安全向动态的、弹性的云服务模式转变创造了基础条件。

OVC 可分为公共 OVC 和普通 OVC 两种：

表 20-1 OVC

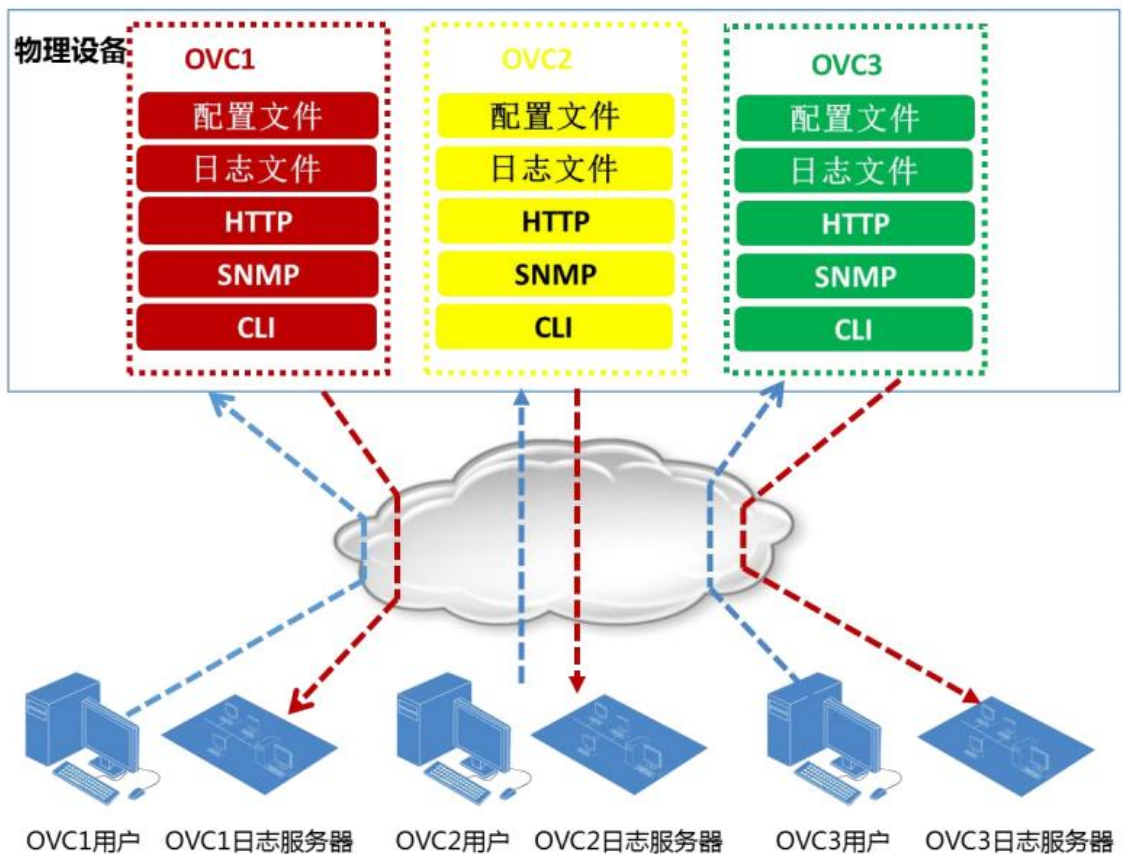
项目	说明
公共 OVC	系统初始状态存在的默认 OVC 实例，称为公共 OVC，所有资源供公共 OVC 统一使用
普通 OVC	公共 OVC 外的其他 OVC 实例称为普通 OVC。创建普通 OVC 后，系统内任何没有划到普通 OVC 的资源都属于公共 OVC

OVC 技术是操作系统级别的虚拟化技术，能够实现 1:N 的虚拟化。通过操作系统级虚拟化技术，可以为每个 OVC 分配独立的端口、CPU、内存资源、会话数、新建、并发、吞吐量、路由表项数量、安全策略数量等一系列的软、硬件资源，灵活定制 OVC 的实际规格。OVC 虚拟化技术使系统可以针对每个虚拟设备进行独立的进程管理、内存管理、磁盘管理，各虚拟设备之间没有切换和调度带来的资源消耗和性能损耗，同时通过操作系统虚拟化的支撑，可以实现各个 OVC 从管理平面、控制平面、数据平面、业务平面全方位隔离，形成各个完全独立的逻辑设备。操作系统内核完成 OVC 虚拟设备间的调度，并按预先设定的资源模板为各 OVC 虚拟设备分配硬件资源。

## 20.2 管理平面虚拟化

如图所示，OVC 实现物理设备的 1:N 虚拟化后，每个 OVC 可以看成独立的设备，用户可通过属于各 OVC 的网络接口访问并管理本 OVC。每个 OVC 拥有独立的 HTTP/CLI/SNMP/SYSLOG 等配置管理协议进程，配置文件单独存放，可以独立进行重启和配置恢复。每个 OVC 拥有独立的管理人员和日志文件，系统日志和操作日志可以独立输出到日志监控服务器。每个 OVC 由相应管理员自主管理，各个 OVC 之间互不可见。

图 20-2 OVC 配置管理



## 20.3 控制平面虚拟化

每个 OVC 会启动各自的管理进程对其所拥有的系统资源进行管理，也会启动各自的协议进程（如 OSPF、ISIS、BGP 等路由协议等）以维持各自的协议运行。每个 OVC 运行独立的协议进程，各进程间互不干扰。

如图 23-2 所示, OVC1 启用了 OSPF/ISIS, OVC2 启用了 OSPF/RIP/BGP, OVC3 启用了 ISIS/BGP, 他们分别拥有独立的进程, 任何 OVC 的协议进程故障不会影响其他 OVC 对应协议进程的正常运行。

图 20-3 控制平面虚拟化



控制平面虚拟化带来的好处是 OVC 间的故障隔离。如图 23-3 所示, OVC2 内的 OSPF 进程崩溃导致该 OVC 的 OSPF 协议无法正常运行, 而其它 OVC 内的 OSPF 进程仍然可以正常运行, 完全不受其影响。

图 20-4 OVC 间的故障隔离



## 20.4 数据平面虚拟化

当创建 OVC 时系统为之划分接口资源, 这些接口由各自的虚拟数据平面管理, 不同的 OVC 之间完全隔离。当流量从属于某一 OVC 的接口进入系统时, 只会查询属于本 OVC 的转发表项, 也只

能从属于该 OVC 的接口转发出去，同时各种路由协议只能在这些接口资源上运行，确保每个 OVC 的转发表项只包含属于本 OVC 的接口，从而使不同 OVC 的路由和转发得到完全的隔离。

在安全设备上，报文转发时还需要建立会话表项，用于记录一些状态信息，为了保证每个 OVC 转发信息的完全隔离，每个 OVC 拥有独立的会话表，报文转发时只会查询、维护属于本 OVC 的会话表，各个 OVC 会话互不干扰，确保各个 OVC 的地址空间和转发信息完全独立。

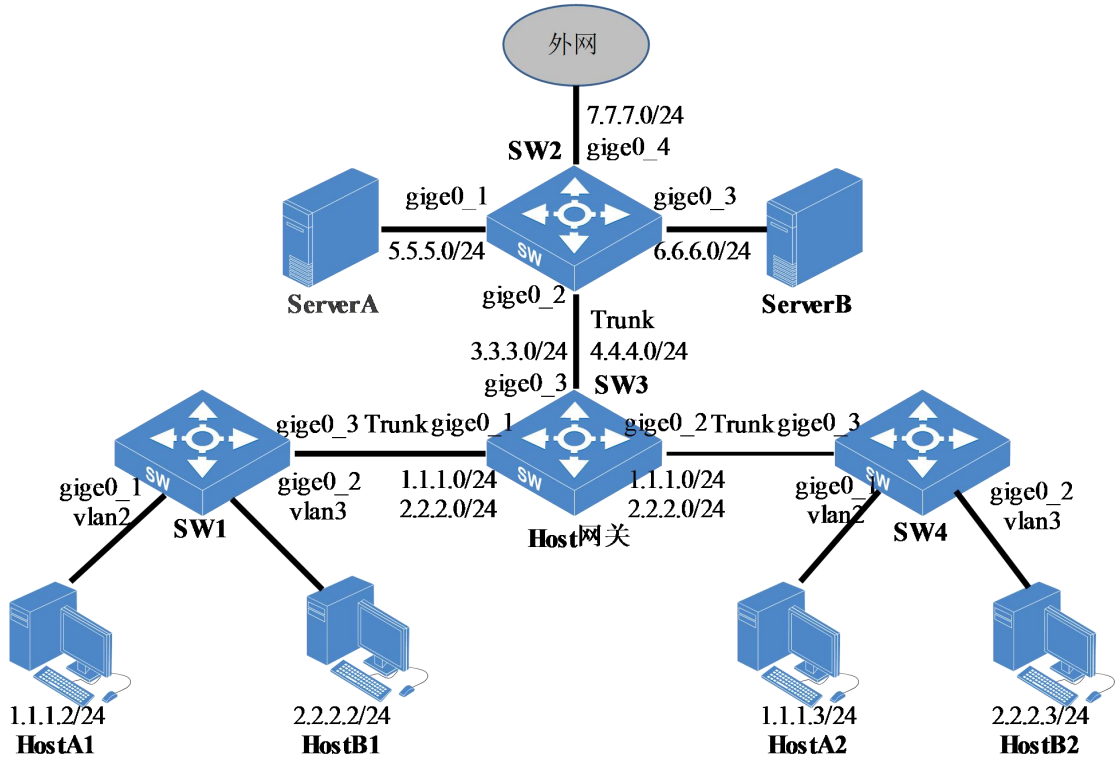
## 20.5 OVC 典型配置案例

### 20.5.1 配置要求

当企业的内网资料需要保密，能访问内网资料的主机不允许访问外网，但还需要保证其他主机能访问外网时，企业可以使用 OVC 功能将其隔离为两个网络，实现两个网络之间相互独立、互不干扰。

## 20.5.2 网络拓扑

图 20-5 OVC 组网图



## 20.5.3 配置流程

- (1) 在 SW1、SW2、SW3、SW4 上创建 VLAN，并给相应端口配置 VLAN。
- (2) 分别在 SW2 和 SW3 上配置 OVC 功能。
- (3) 分别在 SW2 和 SW3 上创建相应的 vlan-if 口，并配置相应的 IP 地址。
- (4) 分别在 SW2 和 SW3 上配置 OSPF 路由。
- (5) 验证配置。

## 20.5.4 配置步骤

- (1) 在 SW1、SW2、SW3、SW4 上创建 VLAN，并给相应端口配置 VLAN。

# 分别在 SW1 和 SW4 上配置

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

#### # 在 SW2 上配置

```
<INSPUR>conf-mode
[INSPUR]vlan 2 to 6
[INSPUR]vlan 2
[INSPUR-vlan2]port gige0_1
[INSPUR-vlan2]exit
[INSPUR]vlan 3
[INSPUR-vlan3]port gige0_3
[INSPUR]vlan 6
[INSPUR-vlan6]port gige0_4
[INSPUR-vlan6]exit
[INSPUR]interface gige0_2
[INSPUR-gige0_2]switchport mode trunk
[INSPUR-gige0_2]switchport trunk allowed vlan 4-5
[INSPUR-gige0_2]switchport trunk native vlan 5
[INSPUR-gige0_2]exit
```

#### # 在 SW3 上配置

```
[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-gige0_1]exit
[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-gige0_2]exit
```



```
[INSPUR]interface gige0_3
[INSPUR-gige0_3]switchport mode trunk
[INSPUR-gige0_3]switchport trunk allowed vlan 4-5
[INSPUR-gige0_3]switchport trunk native vlan 5
[INSPUR-gige0_3]exit
```

(2) 分别在 SW2 和 SW3 上配置 OVC 功能。

#### # 在 SW2 上配置 OVC

```
[INSPUR]ovc enable
[INSPUR]ovc ovc1
[INSPUR-ovc-ovc1]exit
[INSPUR]interface vlan-if 2
[INSPUR-vlan-if2]bind ovc ovc1
[INSPUR-vlan-if2]exit
[INSPUR]interface vlan-if 4
[INSPUR-vlan-if4]bind ovc ovc1
[INSPUR-vlan-if4]exit
[INSPUR]ovc ovc2
[INSPUR-ovc-ovc2] exit
[INSPUR]interface vlan-if 3
[INSPUR-vlan-if3]bind ovc ovc2
[INSPUR-vlan-if3]exit
[INSPUR]interface vlan-if 5
[INSPUR-vlan-if5]bind ovc ovc2
[INSPUR-vlan-if5]exit
[INSPUR]interface vlan-if 6
[INSPUR-vlan-if6]bind ovc ovc2
[INSPUR-vlan-if6]exit
[INSPUR]
```

#### # 在 SW3 上配置 OVC

```
[INSPUR]ovc enable
[INSPUR]ovc ovc1
[INSPUR-ovc-ovc1]exit
[INSPUR]interface vlan-if2
[INSPUR-vlan-if2]bind ovc ovc1
[INSPUR-vlan-if2]exit
[INSPUR]interface vlan-if4
[INSPUR-vlan-if4]bind ovc ovc1
[INSPUR- vlan-if4]exit
[INSPUR]ovc ovc2
```

```
[INSPUR-ovc-ovc2]exit
[INSPUR] interface vlan-if3
[INSPUR-vlan-if3]bind ovc ovc2
[INSPUR-vlan-if3]exit
[INSPUR] interface vlan-if5
[INSPUR-vlan-if5]bind ovc ovc2
[INSPUR-vlan-if3]exit
[INSPUR]
```

(3) 分别在 SW2 和 SW3 上的 OVC 中配置相应的 vlan-if 口的 IP 地址。

# 在 SW2 上配置

```
<INSPUR> switch-ovc ovc1
Now change to new ovc: ovc1.
<INSPUR-ovc1>conf-mode
[INSPUR-ovc1]interface vlan-if2
[INSPUR-vlan-if2-ovc1]ip add 5.5.5.1/24
[INSPUR-vlan-if2-ovc1]exit
[INSPUR-ovc1]interface vlan-if5
[INSPUR-vlan-if5-ovc1]ip add 4.4.4.2/16
[INSPUR-vlan-if5-ovc1]end
<INSPUR-ovc1>exit
Connection closed by foreign host.
<INSPUR> switch-ovc ovc2
Now change to new ovc: ovc2.
<INSPUR-ovc2>conf-mode
[INSPUR-ovc2]interface vlan-if4
[INSPUR-vlan-if4-ovc2]ip add 3.3.3.2/24
[INSPUR-vlan-if4-ovc2]exit
[INSPUR-ovc2]interface vlan-if3
[INSPUR-vlan-if3-ovc2]ip add 6.6.6.1/24
[INSPUR-vlan-if3-ovc2]exit
[INSPUR-ovc2]interface vlan-if6
[INSPUR-vlan-if6-ovc2]ip add 7.7.7.1/24
[INSPUR-vlan-if6-ovc2]exit
```

# 在 SW3 上配置

```
<INSPUR> switch-ovc ovc1
Now change to new ovc: ovc1.
<INSPUR-ovc1>conf-mode
[INSPUR-ovc1]interface vlan-if2
[INSPUR-vlan-if2-ovc1]ip add 1.1.1.1/24
```

```

[INSPUR-vlan-if2-ovc1]exit
[INSPUR-ovc1]interface vlan-if4
[INSPUR-vlan-if4-ovc1]ip add 3.3.3.1/24
[INSPUR-vlan-if-ovc1] end
<INSPUR-ovc1>exit
Connection closed by foreign host.
<INSPUR> switch-ovc ovc2
Now change to new ovc: ovc2.
<INSPUR-ovc2>conf-mode
[INSPUR-ovc2]interface vlan-if3
[INSPUR-vlan-if3-ovc2]ip add 2.2.2.1/24
[INSPUR-vlan-if3-ovc2]exit
[INSPUR-ovc2]interface vlan-if5
[INSPUR-vlan-if5-ovc2]ip add 4.4.4.1/24
[INSPUR-vlan-if5-ovc2]exit
[INSPUR-ovc2]

```

(4) 分别在 SW2 和 SW3 上配置 OSPF 路由。

# 在 SW2 上配置 OSPF

```

<INSPUR> switch-ovc ovc1
Now change to new ovc: ovc1.
<INSPUR-ovc1>conf-mode
[INSPUR-ovc1]router ospf 1
[INSPUR-ospf-1-ovc1]network 5.5.5.0/24 area 0
[INSPUR-ospf-1-ovc1]network 3.3.3.0/24 area 0
[INSPUR-ospf-1-ovc1]exit
[INSPUR-ovc1]exit
<INSPUR-ovc1>exit
Connection closed by foreign host.
<INSPUR> switch-ovc ovc2
Now change to new ovc: ovc2.
<INSPUR-ovc2>conf-mode
[INSPUR-ovc2]route ospf 1
[INSPUR-ospf-1-ovc2]network 4.4.4.0/24 area 0
[INSPUR-ospf-1-ovc2]network 6.6.6.0/24 area 0
[INSPUR-ospf-1-ovc2]network 7.7.7.0/24 area 0
[INSPUR-ospf-1-ovc2]exit
[INSPUR-ovc2]exit
<INSPUR-ovc2>exit
Connection closed by foreign host.
<INSPUR>

```

**# 在 SW3 上配置 OSPF**

```
<INSPUR> switch-ovc ovc1
Now change to new ovc: ovc1.
<INSPUR-ovc1>conf-mode
[INSPUR-ovc1]route ospf 1
[INSPUR-ospf-1-ovc1]network 1.1.1.0/24 area 0
[INSPUR-ospf-1-ovc1]network 3.3.3.0/24 area 0
[INSPUR-ospf-1-ovc1]exit
[INSPUR-ovc1]exit
<INSPUR-ovc1>exit
Connection closed by foreign host.
<INSPUR> switch-ovc ovc2
Now change to new ovc: ovc2.
<INSPUR-ovc2>conf-mode
[INSPUR-ovc2]route ospf 1
[INSPUR-ospf-1-ovc2]network 2.2.2.0/24 area 0
[INSPUR-ospf-1-ovc2]network 4.4.4.0/24 area 0
[INSPUR-ospf-1-ovc2]exit
[INSPUR-ovc2]exit
<INSPUR-ovc2>exit
Connection closed by foreign host.
<INSPUR>
```

**(5) 验证配置。****# 在 SW2 上查看配置的 OVC**

```
<INSPUR>show ovc ovc1
VFW          : PublicSystem
Manage service: managable
Interface list:
vlan-if2
vlan-if4
<INSPUR>show ovc ovc2
VFW          : PublicSystem
Manage service: managable
Interface list:
vlan-if3
vlan-if5
vlan-if6
<INSPUR>
```

**# 在 SW3 上查看配置的 OVC**

```
<INSPUR>show ovc ovc1
VFW          : PublicSystem
Manage service: managable
Interface list:
vlan-if2
vlan-if4
<INSPUR>show ovc ovc2
VFW          : PublicSystem
Manage service: managable
Interface list:
vlan-if3
vlan-if5
<INSPUR>
```

通过在 SW2 和 SW3 上配置 OVC 功能，网络 A 和网络 B 之间相互隔离，不能互访。网络 A 的主机 HostA1 和 HostA2 可以互访，并都能访问服务器 ServerA，但是不能访问外网。网络 B 的主机 HostB1 和 HostB2 可以互访，且都能访问服务器 ServerB，还能正常访问外网。

# 21 VRF 典型配置案例

## 21.1 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。

每台交换机上可维护一个或多个 VRF，多个 VRF 实例相互独立，互不干扰。VRF 是 MPLS VPN 中经常使用的技术，与之密切相关，所以在此将对 MPLS VPN 做简单的介绍。

在 VRF 中定义的和 VPN 业务有关的两个重要参数是 RT 和 RD：

表 21-1 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。

## 21.2 MPLS VPN 简介

MPLS VPN 是一种基于 MPLS 技术的 VPN，是在路由器和交换机上应用的 MPLS 技术实现的虚拟专用网络，可以灵活满足多种业务需求，如一个公司的异地部门的通信。目前 MPLS L3 VPN，

即 BGP/MPLS VPN 技术比较成熟，已经形成标准，其中包含 PE、P、CE、Site 四个部分。

表 21-2 MPLS VPN 简介

项目	说明
PE	骨干网边缘设备，主要作用是存储 VRF，向直连的 CE 学习路由，然后通过 IBGP 与其他 PE 交换所学习的 VPN 路由，是 MPLS 的主要实现者。
P	骨干网核心设备，不与 CE 直连，主要负责 MPLS 转发。
CE	用户边缘设备，通常是 VPN Site 中的一个交换机或路由器。主要作用是与 PE 交互 VPN 路由用户路由信息，向 PE 发布本地路由并从 PE 学习远端站点路由。
Site	VPN 中的一个孤立的 IP 网络，各个 Site 之间通过运营商骨干网络实现连通。

用户接入 MPLS VPN 的方式是每个 Site 提供一个或多个 CE，同骨干网的 PE 连接。在 PE 上为这个 Site 配置 VRF，将连接 PE-CE 的物理接口、逻辑接口绑定到 VRF 上。

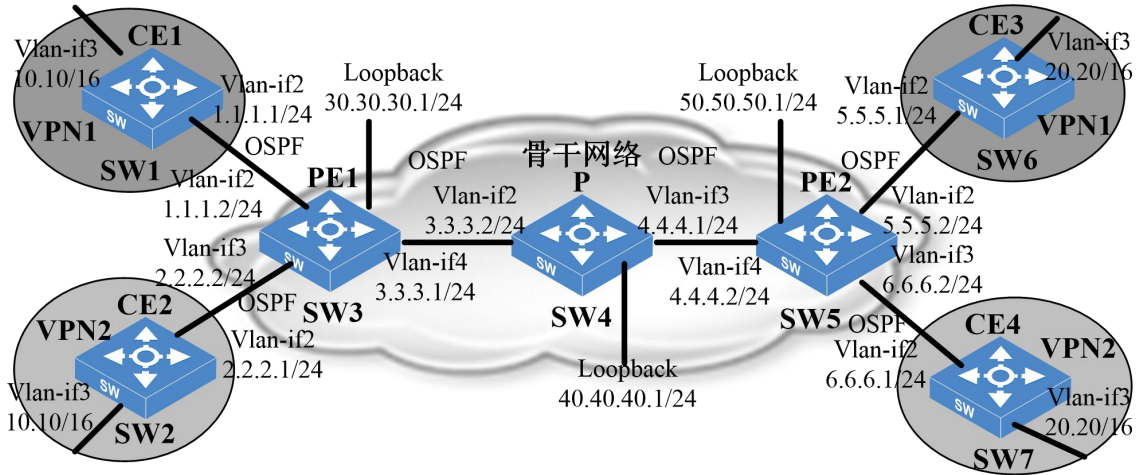
## 21.3 VRF 典型配置案例

### 21.3.1 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。

## 21.3.2 网络拓扑

图 21-2 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。

## 21.3.3 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。



### 21.3.4 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。