

计算机网络技术实践

实验报告



实验名称： 实验四—VLAN 组网配置实验

姓 名： 陈朴炎 实 验 日 期： 2023.11.19

学 号： 2021211138 实验报告日期： 2023.12.19

报告退发： （ 订正、 重做 ）

目录

一、 环境.....	3
1.1 运行环境.....	3
1.2 网络拓扑.....	3
1.2.1 网络拓扑图.....	3
1.2.2 网络拓扑说明.....	4
二、 实验目的.....	4
三、 实验内容及步骤.....	4
3.1 配置 PC.....	4
3.2 配置交换机及前两部分实验.....	5
3.2.1 配置交换机.....	5
3.2.2 实验四第一部分.....	9
3.2.3 实验四第二部分.....	10
3.3 配置路由器.....	11
3.3.1 方法 1 和 access 口相连.....	11
3.3.2 方法 2 和 trunk 口相连.....	13
3.3.3 配置路由器之间的动态路由.....	15
四、 实验结果.....	18
4.1 第一部分.....	18
4.2 第二部分.....	20
4.3 第三部分 不同 VLAN 互联.....	22
4.3.1 方法 1.....	22
4.3.2 方法 2.....	24
4.3.3 跨路由器互联.....	26
五、 实验中的问题及心得.....	27
六、 实验思考.....	30
6.1 同个局域网配置不同 IP 网段.....	30
6.2 分析数据包传输流程.....	31
6.3 物理网络、VLAN、IP 网段的关系.....	33

一、 环境

1.1 运行环境

本次实验在 Windows 11 上完成，使用 GNS3 虚拟仿真平台。

GNS3 版本：22.44.1

GNS3 的服务器运行在虚拟机上，使用 GNS3 VM.ova 作为映像。

运行虚拟机的程序为 VM Ware workstation pro 17.x

VM Version: 0.15.0

Qemu version 4.2.1

Ubuntu version focal

KVM support available: True

IP: 192.168.177.127 port: 80

1.2 网络拓扑

1.2.1 网络拓扑图

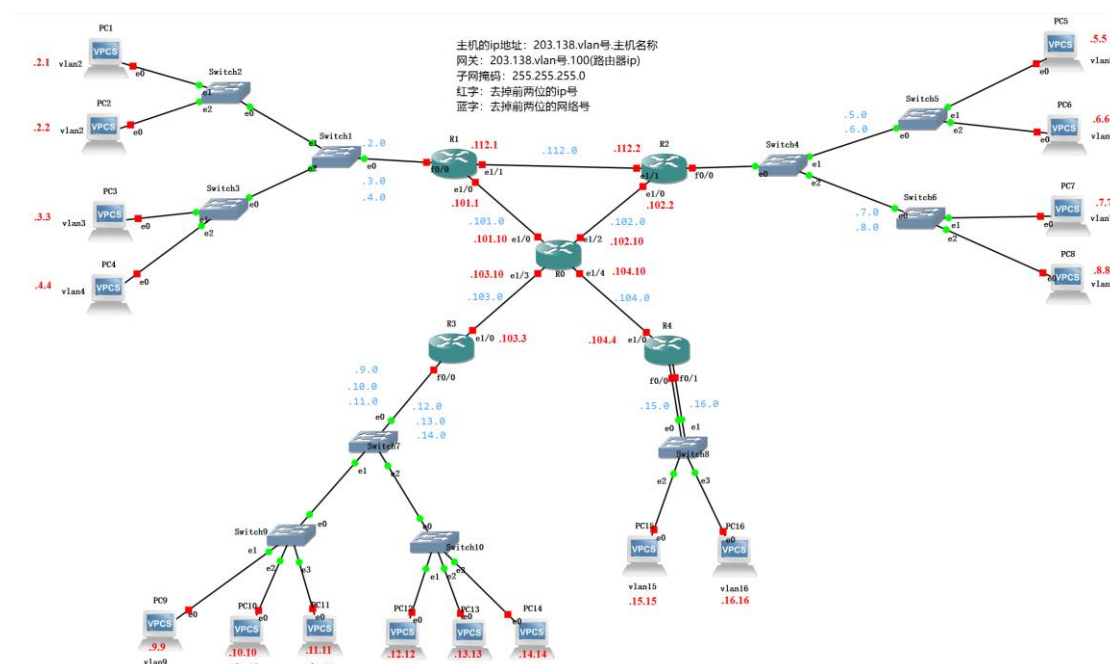


图 1-1 网络拓扑图

1.2.2 网络拓扑说明

网络拓扑中有 5 个路由器，10 个交换机，16 个 PC 机，15 个 VLAN。

整个网络拓扑的子网掩码都是 255.255.255.0

网络号的前两位都为 203.138，在网络拓扑图中，用蓝色的数字标明。

网络拓扑图中，红色的为各个主机、端口的 IP 号后两位。

其中，主机的 IP 号为 203.138.vlan 号.主机号。vlan 号标注在了 PC 的旁边，比如：PC2 的 vlan 号为 2，PC1 的 vlan 号也为 2。PC1 的主机号就为 1，PC2 的主机号就为 2 以此类推。

路由器互联部分的网络号为 203.138.“1RnRm”.0，比如说，R1 和 R2 相连的那部分网络号就为 203.138.112.0，第三位的后两位为相连的路由器的标号。

路由器和 PC 连的端口为各个子网的网关，IP 号的最后一位为 100。

二、实验目的

能够清楚的知道不同 VLAN 配置的区别和原理

能够说明一个数据包从一台主机到另一台主机的发送传输流程

能够设计网络拓扑，配置各台机器，并令网络联通

三、实验内容及步骤

3.1 配置 PC

双击 PC，对 PC 输入如下命令，配置 PC 的 ip：

```
ip 203.138.vlan号.PC号/24 203.138.vlan号.100
```

比如 PC1, 就输入 `ip 203.138.2.1/24 203.138.2.100`

203.138.2.1 为 PC 的 IP 号, /24 为子网掩码的位数, 203.138.2.100 为

这个网段的网关。如下图所示

```
PC1> ip 203.138.2.1/24 203.138.2.100
Checking for duplicate address...
PC1 : 203.138.2.1 255.255.255.0 gateway 203.138.2.100
```

图 3-1 配置 PC 的 ip 图

同理, 配置其他的 16 个 PC 机, 过程都相同, 不再重复罗列。

配置完后, 使用 save 命令保存 PC 的配置, 如下图所示:

```
PC16> save
Saving startup configuration to startup.vpc
. done
PC16>
```

图 3-2 保存 PC 配置

3.2 配置交换机及前两部分实验

3.2.1 配置交换机

双击交换机, 出现如下配置界面

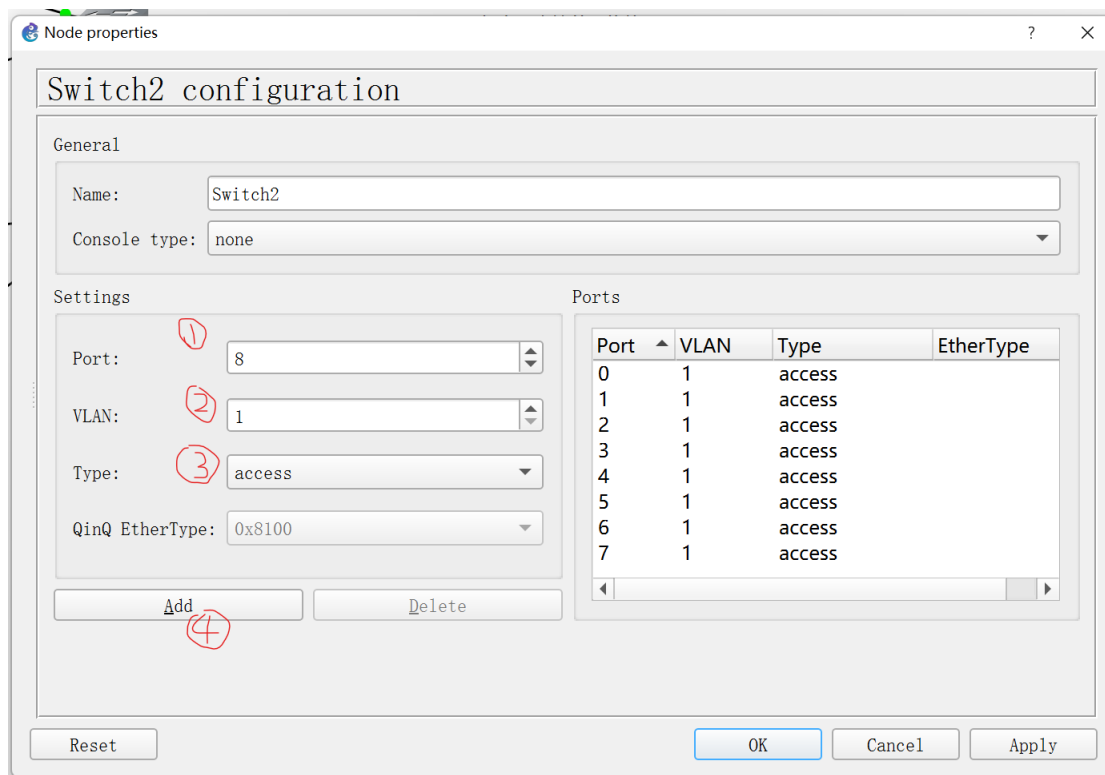


图 3-3 交换机配置界面

首先选择端口号，以 Switch2 为例，Switch2 的 e1 端口连 PC1，e2 端口连 PC2，e0 端口连 Switch1。在这里，我把交换机和 PC 连接的端口类型都设置为 access 口，而交换机跟交换机/路由器连的端口类型都设置为 trunk 口（除了 Switch8 和 R4）。配置交换机的步骤如下：

- ① 选择端口
- ② 更改 VLAN 号
- ③ 选择端口的类型
- ④ 点击 Add
- ⑤ 点击 Apply → OK

前四步做完之后，Switch2 的配置结果如下，Type 为 dot1q 表示该端口类型为 trunk 口。我把 0 号端口设置为 trunk，1、2 号端口设置为 access 口，

并且 1、2 端口的 VLAN 号为 2。trunk 口的 vlan 号最好设置成 1，不然会出现问题（后面会讨论）。

Node properties

Switch2 configuration

General

Name: Switch2

Console type: none

Settings

Port: 8

VLAN: 1

Type: dot1q

QinQ EtherType: 0x8100

Ports

Port	VLAN	Type	EtherType
0	1	dot1q	
1	2	access	
2	2	access	
3	1	access	
4	1	access	
5	1	access	
6	1	access	
7	1	access	

Buttons: Add, Delete, Reset, OK, Cancel, Apply

图 3-4 Switch2 配置图

Switch1 configuration

General

Name: Switch1

Console type: none

Settings

Port: 8

VLAN: 1

Type: access

QinQ EtherType: 0x8100

Ports

Port	VLAN	Type	EtherType
0	1	dot1q	
1	1	dot1q	
2	1	dot1q	
3	1	access	
4	1	access	
5	1	access	
6	1	access	
7	1	access	

Buttons: Add, Delete

图 3-5 Switch1 配置图

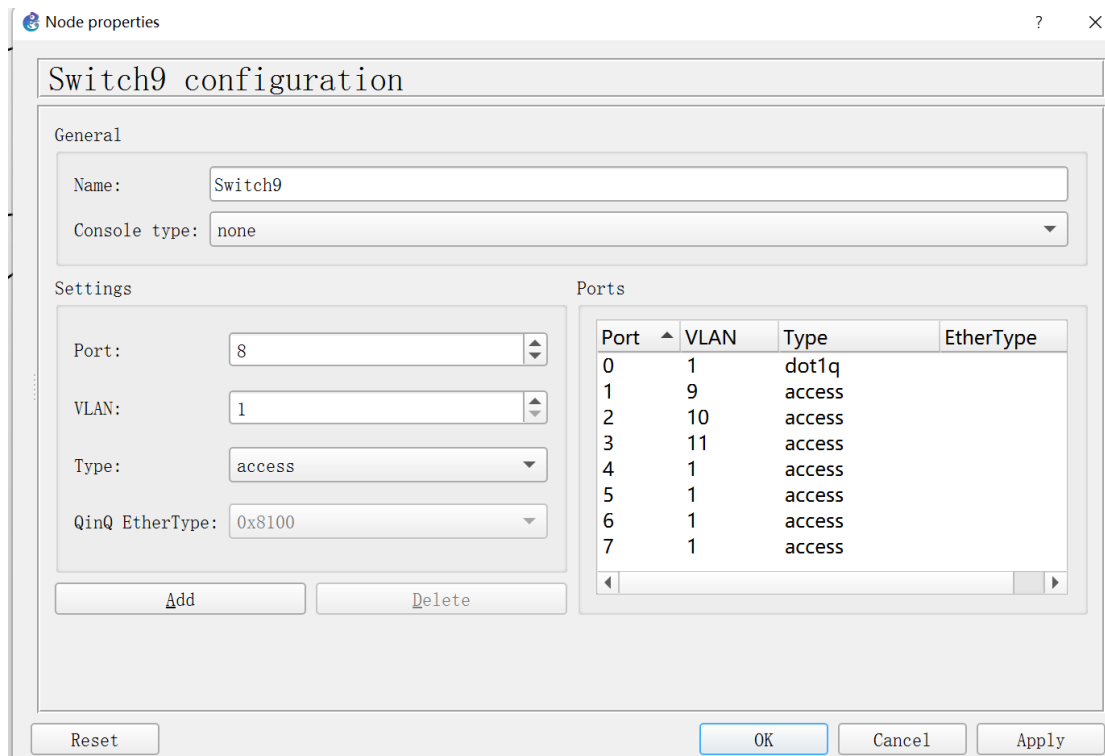


图 3-6 Switch9 配置图

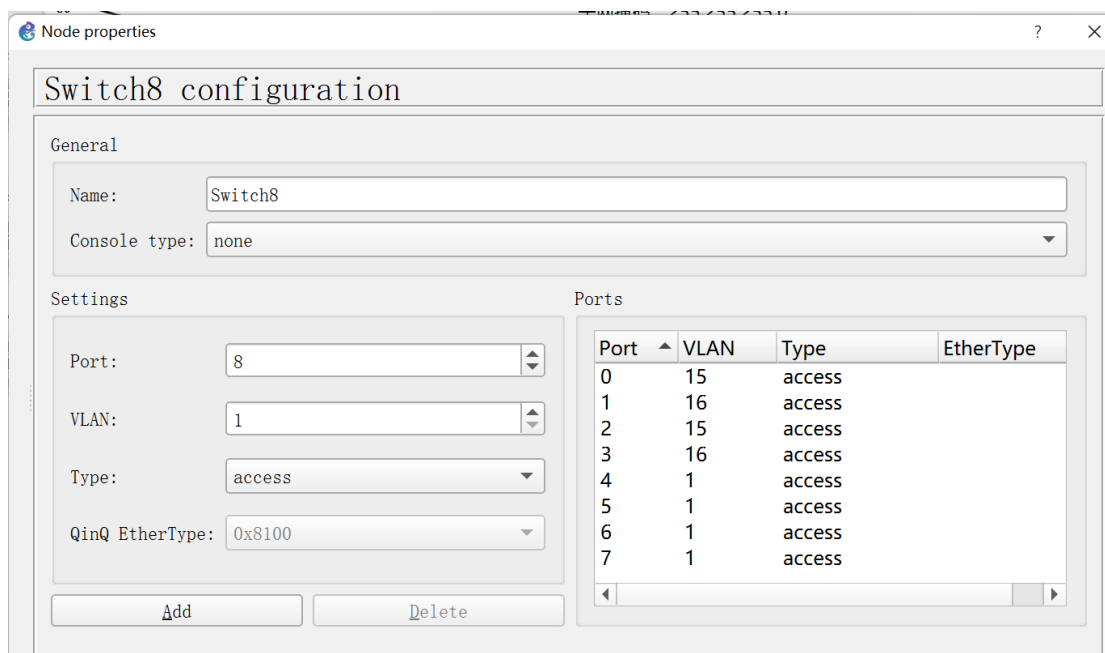


图 3-7 Switch8 配置图

以上是几个在我的网络拓扑图中重要/典型的交换机配置，其他交换机配置和它们类似。

3.2.2 实验四第一部分

现在做实验 4 的第一部分：

启动 Switch1、2, PC1、2。PC1 的 IP 配置[同 3.1.1 节所示](#), PC2 的 ip 配置为 203.138.2.2。将 Switch2 的端口 e1 设置为 vlan2, e2 设置为 vlan3, 如下图所示：

Switch2 configuration

General

Name: Switch2

Console type: none

Settings

Port: 8

VLAN: 3

Type: access

QinQ EtherType: 0x8100

Ports

Port	VLAN	Type	EtherType
0	1	dot1q	
1	2	access	
2	3	access	
3	1	access	
4	1	access	
5	1	access	
6	1	access	
7	1	access	

Add Delete

图 3-8 第一部分 Switch2 配置 1

查看 PC1 和 PC2 能否 ping 通：

```
PC1>
PC1> ping 203.138.2.2
host (203.138.2.2) not reachable
PC1> █
```

图 3-9 PC1 ping PC2

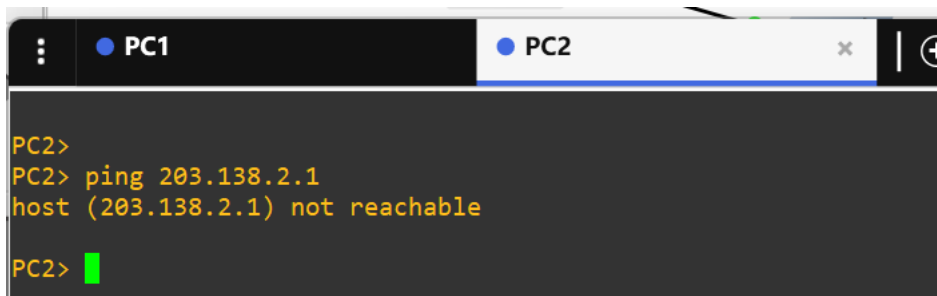


图 3-10 PC2 ping PC1

不能 ping 通。

将 Switch2 的 e2 端口配置成 vlan2，查看能否 ping 通：

A configuration window for Switch2. The 'Name' field is 'Switch2' and 'Console type' is 'none'. The 'Settings' tab is active, showing 'Port: 8', 'VLAN: 2', 'Type: access', and 'QinQ EtherType: 0x8100'. The 'Ports' tab is also visible, showing a table of ports and their configurations.

Port	VLAN	Type	EtherType
0	1	dot1q	
1	2	access	
2	2	access	
3	1	access	
4	1	access	
5	1	access	
6	1	access	
7	1	access	

图 3-11 修改 Switch2 配置

重新查看 PC1 和 PC2 能否 ping 通。实验结果在 [4.1 第一部分](#)。

3.2.3 实验四第二部分

用拓扑图的左上角部分 PC1、PC3、Switch1、Switch2、Switch3 进行实验。

将 Switch3 配置信息更改，e1 端口配置成 vlan2，如下：

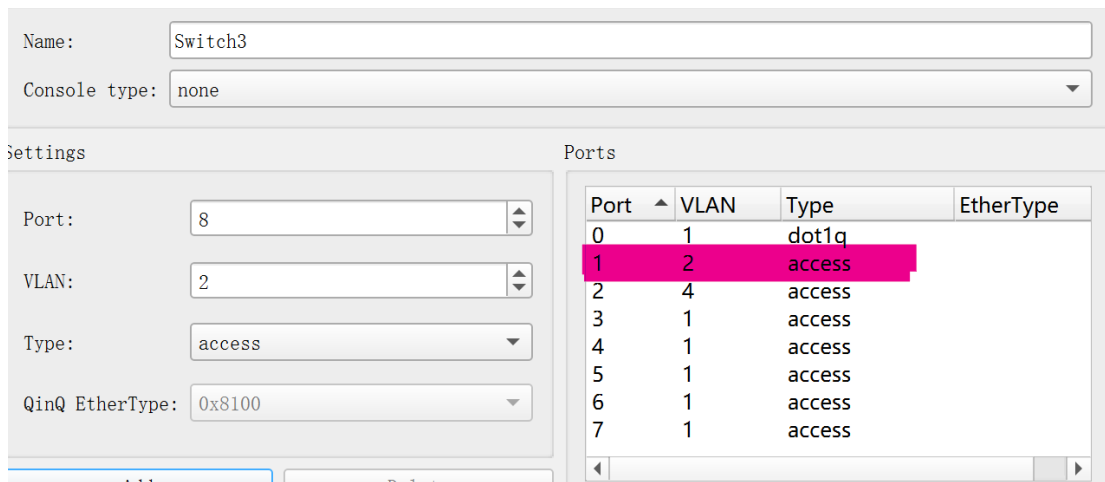


图 3-12 修改 Switch3 端口标签

修改 PC3 的 ip 号为 203.138.2.3, 配置如下:

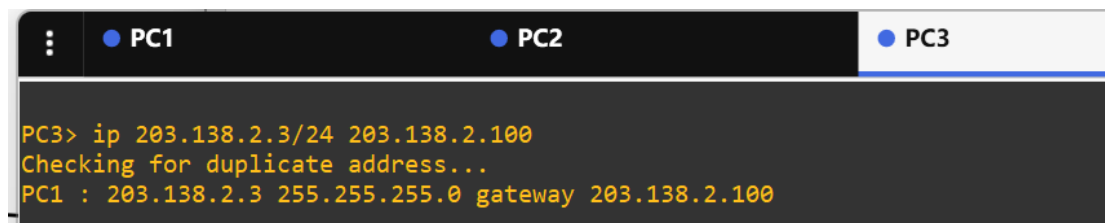


图 3-13 PC3 的 ip 配置图

而 Switch1、2、3 相连的部分都为 trunk 口。现在尝试 PC1 、PC3 互相 ping, 实验结果及分析在 [4.2 第二部分](#)。

3.3 配置路由器

路由器的端口有两种配置方法, 一种是一个端口一个 vlan 号, 另一种是一个端口划分成多个子端口, 一个子端口一个 vlan 号。下面我将详细说明两种配置方法的实验步骤。

3.3.1 方法 1 和 access 口相连

由于交换机的 access 口只允许同一个 vlan 标签的数据包传输, 因此需要为路由器连接的每一个 vlan 配置一个端口。

配置一个端口一个 vlan 号的路由器，步骤如下。

在拓扑图中，我使用 R4 作为方法 1 的路由器。R4 的 f0/0 为 PC15 的网关，R4 的 f0/1 为 PC16 的网关。

首先双击 R4 路由器，进入配置界面。

输入如下配置信息：

```
R4# conf t
R4(config)# in f0/0
R4(config-if)# ip add 203.138.15.100 255.255.255.0
R4(config-if)# no shutdown
R4(config-if)# exit
```

```
R4#conf
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
R4(config)#in f0/0
R4(config-if)#encapsulation dot1q 15
^
% Invalid input detected at '^' marker.
R4(config-if)#ip add 203.138.15.100 255.255.255.0
R4(config-if)#no shutdown
R4(config-if)#exit
R4(config)#
```

图 3-14 配置 R4 f0/0 端口示意图

```
R4(config)#in f0/1
R4(config-if)#ip 203.138.16.100 255.255.255.0
^
% Invalid input detected at '^' marker.
R4(config-if)#ip add 203.138.16.100 255.255.255.0
R4(config-if)#no shutdown
R4(config-if)#exit
R4(config)#
*Dec 19 15:18:28.659: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*Dec 19 15:18:29.663: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
R4(config)#
```

图 3-15 配置 R4 f0/1 端口示意图

查看 R4 此时的路由表，如下：

```
Gateway of last resort is not set

C    203.138.16.0/24 is directly connected, FastEthernet0/1
C    203.138.15.0/24 is directly connected, FastEthernet0/0
R4#
```

图 3-16 R4 路由表

配置好后，尝试 PC15 和 PC16 互通，实验结果在 [4.3.1 方法 1](#)。

3.3.2 方法 2 和 trunk 口相连

当路由器的一个端口和交换机的 trunk 口相连的时候，路由器需要为这个端口划分多个子端口，来让它的一个端口作为不同虚拟局域网的网关。配置过程如下所示，这里我选择 R1 为例，R1 的 f0/0 连接了 vlan2、vlan3、vlan4，因此需要为 R1 划分三个子端口：f0/0.2，f0/0.3，f0/0.4，并为每一个子端口配置 ip 地址和协议。步骤如下：

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#in f0/0.2
R1(config-subif)#enc
R1(config-subif)#encapsulation dot1q 2
R1(config-subif)#ip add 203.138.2.100 255.255.255.0
R1(config-subif)#exit
```

conf t 表示进入配置环境，in f0/0.2 表示进入到接口 f0/0 的 .2 子接口配置环境。encapsulation dot1q 2 表示为该子接口封装上 dot1q 协议，其中标签为 2。最后加上该子接口的 ip 地址和子网掩码。no shutdown 用来开启该接口。

```
administratively down
R1#
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#in f0/0.2
R1(config-subif)#enc
R1(config-subif)#encapsulation dot1q 2
R1(config-subif)#ip add 203.138.2.100 255.255.255.0
R1(config-subif)#no shutdown
R1(config-subif)#exit
R1(config)#
```

图 3-17 R1 的 f0/0.2 配置过程

同理，配置 R1 的 f0/0.3、f0/0.4，过程如下：

```
Enter configuration commands, one per line. End with CN
R1(config)#in f0/0.3
R1(config-subif)#enca
R1(config-subif)#encapsulation dot
R1(config-subif)#encapsulation dot1Q 3
R1(config-subif)#ip add 203.138.3.100 255.255.255.0
R1(config-subif)#exit
```

图 4-18 R1 的 f0/0.3 配置过程

```
R1(config-subif)#exit
R1(config)#in f0/0.4
R1(config-subif)#en
R1(config-subif)#encapsulation dot
R1(config-subif)#encapsulation dot1Q 4
R1(config-subif)#ip add 203.138.4.100 255.255.255.0
R1(config-subif)#exit
```

图 4-19 R1 的 f0/0.4 配置过程

最后，要进入子接口对应的物理接口，把该端口打开，如下：

```
R1(config)#in f0/0
R1(config-if)#no shutdown
R1(config-if)#ex
R1(config)#
```

图 4-20 打开 R1 的 f0/0 端口

检查配置是否成功：用 PC3 去 ping PC1、PC4

```
PC3> ping 203.138.4.4
84 bytes from 203.138.4.4 icmp_seq=1 ttl=63 time=48.151 ms
84 bytes from 203.138.4.4 icmp_seq=2 ttl=63 time=31.880 ms
84 bytes from 203.138.4.4 icmp_seq=3 ttl=63 time=31.648 ms
84 bytes from 203.138.4.4 icmp_seq=4 ttl=63 time=30.124 ms
84 bytes from 203.138.4.4 icmp_seq=5 ttl=63 time=30.713 ms

PC3> ping 203.138.2.1
84 bytes from 203.138.2.1 icmp_seq=1 ttl=63 time=47.481 ms
84 bytes from 203.138.2.1 icmp_seq=2 ttl=63 time=31.021 ms
84 bytes from 203.138.2.1 icmp_seq=3 ttl=63 time=30.481 ms
```

图 4-21 PC3 ping PC4、PC1 测试结果

结果显示，路由器 f0/0 接口配置成功。查看 R1 的路由表，如下：

```
C    203.138.4.0/24 is directly connected, FastEthernet0/0.4
C    203.138.3.0/24 is directly connected, FastEthernet0/0.3
C    203.138.2.0/24 is directly connected, FastEthernet0/0.2
R1#wr
```

图 4-22 R1 路由表 1

按 R1 的方法配置 R2、R3，过程重复，不再罗列。

这里测试使用 PC2 和 PC4，查看它们之间能否互联。实验结果请看 [4.3.2](#)

[不同 vlan 互联——方法 2 实验结果](#)

3.3.3 配置路由器之间的动态路由

路由器之间需要动态路由协议，这样才能让不同路由器连的不同 vlan 的 PC 机互联。这里我选择使用 ospf 协议，配置过程如下：

同样的，以 R1 为例

蓝字：去掉前两位的网络号

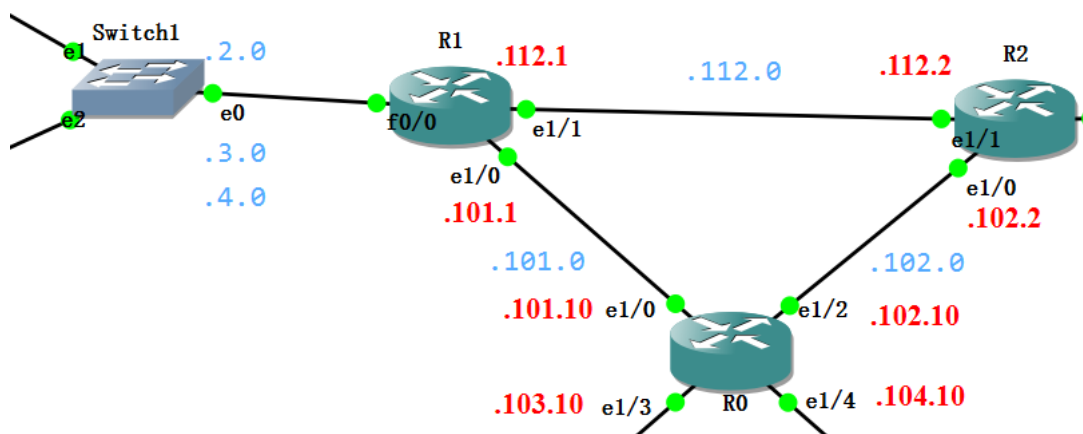


图 4-23 R1 的连接情况

R1 的 e1/1 端口要配 OSPF 协议，e1/0 端口也要配 OSPF 协议。由于我的路由器卡槽插的是 PA-8E，所以不用配置 clockrate 和封装 ppp 协议。配置步骤如下：

```
R1#conf t
R1(config)#in e1/1
R1(config-if)#ip ospf hel
R1(config-if)#ip ospf hello-interval 5
R1(config-if)#ip ospf dead
```

```
R1(config-if)#ip ospf dead-interval 20
R1(config-if)#no shutdown
R1(config-if)#
```

这些命令的解释如下：

conf t 表示进入到该路由器的配置界面。in e1/1 表示进入到 e1/1 接口的配置环境中。ip ospf hello-interval 5 表示该接口每隔 5 秒都会像隔壁邻居发送一个 hello 包。ip ospf dead-interval 20 表示若是 20 秒收不到 hello 包就代表宕机。最后 no shutdown 启动该接口。

```
% Invalid input detected at ... marker.
R1(config-if)#ip ospf hel
R1(config-if)#ip ospf hello-interval 5
R1(config-if)#ip ospf dead
R1(config-if)#ip ospf dead-interval 20
R1(config-if)#no shutdown
R1(config-if)#
*Dec 19 16:18:16.715: %LINK-3-UPDOWN: Interface Ethernet1/1, changed state to up
*Dec 19 16:18:17.715: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet1/1, changed state to up
R1(config-if)#
```

图 4-24 R1 e1/1 端口配置 ospf 协议示意图

同理，还要配置 R1 e1/0 端口的 ospf 协议，

```
R1(config)#in e1/0
R1(config-if)#ip add 203.138.101.1 255.255.255.0
R1(config-if)#ip ospf
R1(config-if)#ip ospf hel
R1(config-if)#ip ospf hello-interval 5
R1(config-if)#ip ospf dea
R1(config-if)#ip ospf dead-interval 20
R1(config-if)#no shutdown
R1(config-if)#exit
*Dec 19 17:31:09.919: %LINK-3-UPDOWN: Interface Ethernet1/0, changed state to up
*Dec 19 17:31:10.919: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet1/0, changed state to up
```

图 4-25 R1 e1/0 端口配置 ospf 协议示意图

配完端口，还需要配置 ospf 的路由邻居，配置命令如下：

```
router ospf 10
network 203.138.2.0 255.255.255.0 area 0
```

router ospf 10 表示开启 10 号进程来运行 ospf 路由协议

network 命令后面跟着的网络是和该路由直连的网络，255.255.255.0 为子网掩码。

area 0 表示该网络被划分在区域 0


```

R1(config)#router ospf 10
R1(config-router)#network 203.138.2.0 255.255.255.0 area 0
R1(config-router)#network 203.138.3.0 255.255.255.0 area 0
R1(config-router)#network 203.138.4.0 255.255.255.0 area 0
R1(config-router)#network 203.138.112.0 255.255.255.0 area 0
R1(config-router)#network 203.138.101.0 255.255.255.0 area 0
R1(config-router)#exit
R1(config)#

```

图 4-26 R1 配置 ospf 路由协议

这样，R1 就配置完成了。用同样的方式配置 R0、R2、R3、R4，过程重复，不再一一罗列。

配置完成所有的路由器的动态路由协议后，查看中枢路由器 R0 的路由表，如下图所示：

```

R0#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

O    203.138.112.0/24 [110/20] via 203.138.102.2, 00:03:09, Ethernet1/2
     [110/20] via 203.138.101.1, 00:03:09, Ethernet1/0
O    203.138.7.0/24 [110/11] via 203.138.102.2, 00:03:09, Ethernet1/2
O    203.138.6.0/24 [110/11] via 203.138.102.2, 00:03:09, Ethernet1/2
O    203.138.5.0/24 [110/11] via 203.138.102.2, 00:03:09, Ethernet1/2
O    203.138.4.0/24 [110/11] via 203.138.101.1, 00:03:09, Ethernet1/0
O    203.138.3.0/24 [110/11] via 203.138.101.1, 00:03:09, Ethernet1/0
C    203.138.101.0/24 is directly connected, Ethernet1/0
O    203.138.2.0/24 [110/11] via 203.138.101.1, 00:03:09, Ethernet1/0
O    203.138.16.0/24 [110/11] via 203.138.104.4, 00:00:30, Ethernet1/4
C    203.138.103.0/24 is directly connected, Ethernet1/3
C    203.138.102.0/24 is directly connected, Ethernet1/2
O    203.138.15.0/24 [110/11] via 203.138.104.4, 00:00:30, Ethernet1/4
C    203.138.104.0/24 is directly connected, Ethernet1/4
O    203.138.14.0/24 [110/11] via 203.138.103.3, 00:03:10, Ethernet1/3
O    203.138.13.0/24 [110/11] via 203.138.103.3, 00:03:10, Ethernet1/3
O    203.138.12.0/24 [110/11] via 203.138.103.3, 00:03:10, Ethernet1/3
O    203.138.11.0/24 [110/11] via 203.138.103.3, 00:03:10, Ethernet1/3
O    203.138.10.0/24 [110/11] via 203.138.103.3, 00:03:10, Ethernet1/3
O    203.138.9.0/24 [110/11] via 203.138.103.3, 00:03:10, Ethernet1/3
O    203.138.8.0/24 [110/11] via 203.138.102.2, 00:03:10, Ethernet1/2
R0#

```

图 4-30 R0 的路由表

路由表中有 21 个表项，表示有 21 个网段，以 C 为标记的是直连的网络，以 O 为标记的是 ospf 协议动态学习得来的网络。从 R0 的路由表可以看出，R0 可以到达网络拓扑图中的任意一个网络。

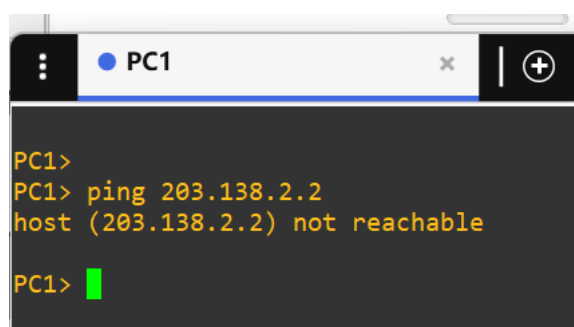
配置所有路由器后，尝试用 PC1 ping 通 PC16 和 PC8。结果见 [4.3.3 跨](#)

[路由器互联。](#)

四、实验结果

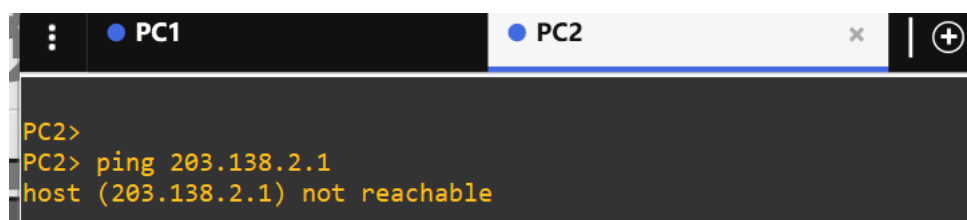
4.1 第一部分

第一次：Switch2 的 e1 端口 vlan 号为 2，e2 端口的 vlan 号为 3，PC1 和 PC2 的互相 ping 的结果如下：



```
PC1>
PC1> ping 203.138.2.2
host (203.138.2.2) not reachable
PC1> █
```

图 4-1 第一次 PC1 ping PC2

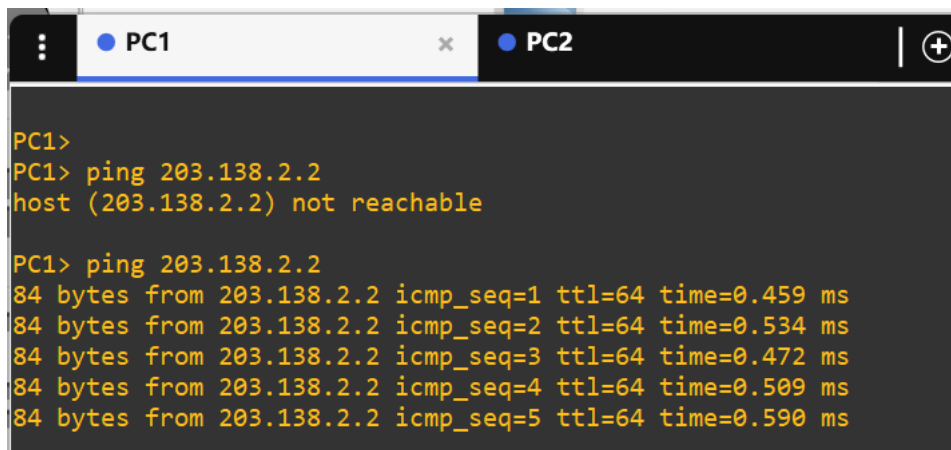


```
PC2>
PC2> ping 203.138.2.1
host (203.138.2.1) not reachable
```

图 4-2 第一次 PC2 ping PC1

PC1 和 PC2 不能互相 ping 通。

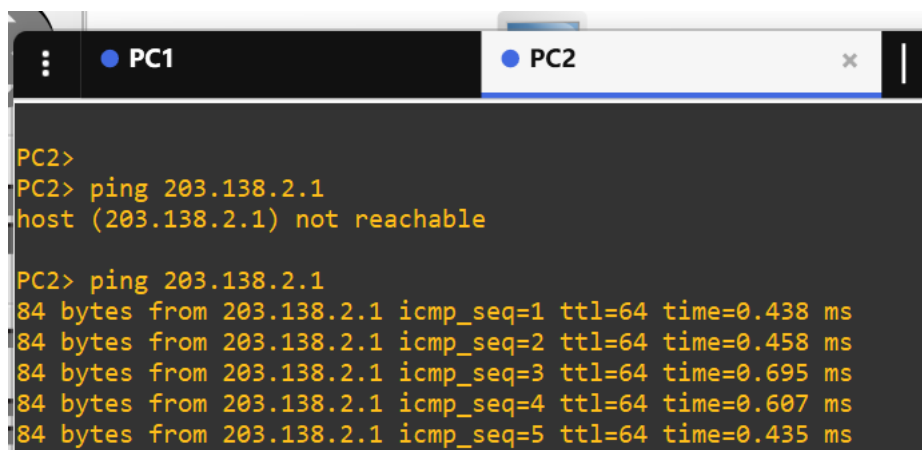
第二次：Switch2 的 e1 端口和 e2 端口 vlan 号都为 2，结果如下：



```
PC1>
PC1> ping 203.138.2.2
host (203.138.2.2) not reachable

PC1> ping 203.138.2.2
84 bytes from 203.138.2.2 icmp_seq=1 ttl=64 time=0.459 ms
84 bytes from 203.138.2.2 icmp_seq=2 ttl=64 time=0.534 ms
84 bytes from 203.138.2.2 icmp_seq=3 ttl=64 time=0.472 ms
84 bytes from 203.138.2.2 icmp_seq=4 ttl=64 time=0.509 ms
84 bytes from 203.138.2.2 icmp_seq=5 ttl=64 time=0.590 ms
```

图 4-3 第二次 PC1 ping PC2



```
PC2>
PC2> ping 203.138.2.1
host (203.138.2.1) not reachable

PC2> ping 203.138.2.1
84 bytes from 203.138.2.1 icmp_seq=1 ttl=64 time=0.438 ms
84 bytes from 203.138.2.1 icmp_seq=2 ttl=64 time=0.458 ms
84 bytes from 203.138.2.1 icmp_seq=3 ttl=64 time=0.695 ms
84 bytes from 203.138.2.1 icmp_seq=4 ttl=64 time=0.607 ms
84 bytes from 203.138.2.1 icmp_seq=5 ttl=64 time=0.435 ms
```

图 4-4 第二次 PC2 ping PC1

实验结果：第一次，由于 PC1 和 PC2 连接的端口是同一个交换机下不同的 vlan 标签，导致 PC1 和 PC2 不能互相 ping 通；第二次，由于 PC1 和 PC2 连接的端口是同一个交换机下的同一个 vlan 标签，因此 PC1 和 PC2 可以互相 ping 通。

结果分析：

在第一次互相 ping 的时候，以 PC1 ping PC2 为例：交换机收到 PC1 的报文，此时交换机的 e1 端口给报文打上 vlan2 的标签。交换机会查看自己的各个端口有没有可以转发的端口，发现只有一个 trunk 口可以转发，其他的端口标

号都不是 vlan2, 就将 PC1 的报文发送给 e0 的 trunk 口, 而不进入 e2 的 vlan3 access 口。

在第二次互相 ping 的时候, 以 PC1 ping PC2 为例: 交换机的 e1 的 access 口收到 PC1 的报文, 此时报文没有标签, 交换机为它打上标签, 然后查找自己的各个端口有没有满足 vlan2 的标签的端口, 发现有 access 端口 e2, 有 trunk 端口 e0, 交换机将 PC1 的 ping 包发送给这两个端口。通过 access 口转发的数据包会将标签去掉再转发。

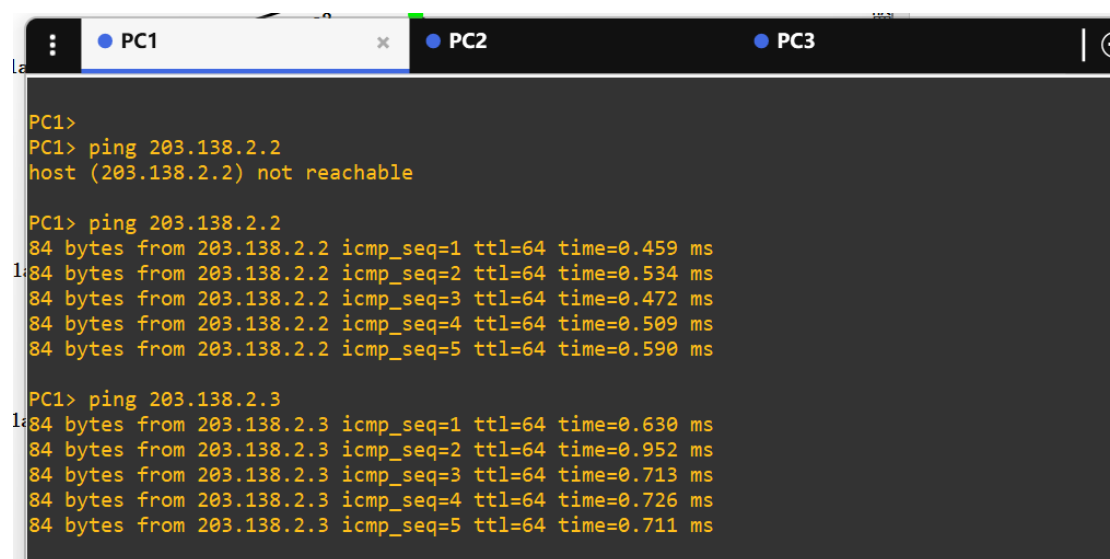
Access 端口的收发报文逻辑如下:

Access 端口收报文: 收到一个报文, 判断是否有 VLAN 信息: 如果没有则打上端口的 PVID, 并进行交换转发, 如果有则直接丢弃 (缺省)

Access 端口发报文: 将报文的 VLAN 信息剥离, 直接发送出去

4.2 第二部分

尝试用 PC1 ping PC3, 结果如下:



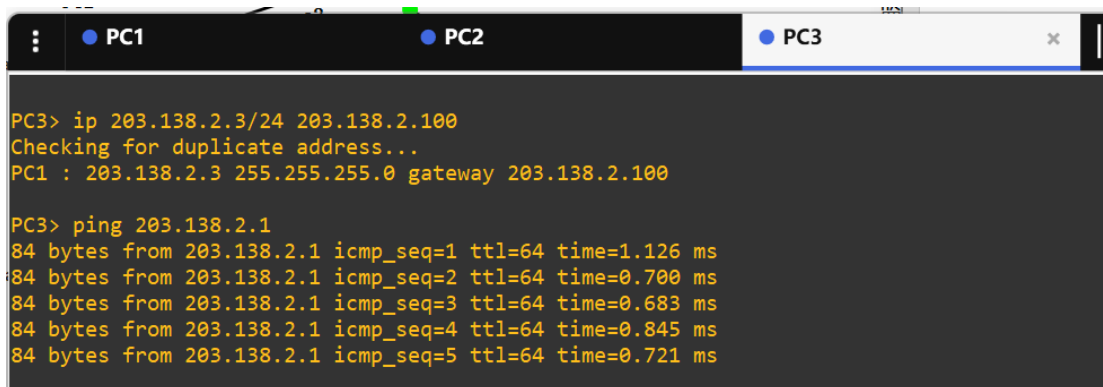
```
PC1>
PC1> ping 203.138.2.2
host (203.138.2.2) not reachable

PC1> ping 203.138.2.2
84 bytes from 203.138.2.2 icmp_seq=1 ttl=64 time=0.459 ms
84 bytes from 203.138.2.2 icmp_seq=2 ttl=64 time=0.534 ms
84 bytes from 203.138.2.2 icmp_seq=3 ttl=64 time=0.472 ms
84 bytes from 203.138.2.2 icmp_seq=4 ttl=64 time=0.509 ms
84 bytes from 203.138.2.2 icmp_seq=5 ttl=64 time=0.590 ms

PC1> ping 203.138.2.3
84 bytes from 203.138.2.3 icmp_seq=1 ttl=64 time=0.630 ms
84 bytes from 203.138.2.3 icmp_seq=2 ttl=64 time=0.952 ms
84 bytes from 203.138.2.3 icmp_seq=3 ttl=64 time=0.713 ms
84 bytes from 203.138.2.3 icmp_seq=4 ttl=64 time=0.726 ms
84 bytes from 203.138.2.3 icmp_seq=5 ttl=64 time=0.711 ms
```

图 4-5 PC1 ping PC3 结果图

尝试用 PC3 ping PC1, 结果如下:



```
PC3> ip 203.138.2.3/24 203.138.2.100
Checking for duplicate address...
PC1 : 203.138.2.3 255.255.255.0 gateway 203.138.2.100

PC3> ping 203.138.2.1
84 bytes from 203.138.2.1 icmp_seq=1 ttl=64 time=1.126 ms
84 bytes from 203.138.2.1 icmp_seq=2 ttl=64 time=0.700 ms
84 bytes from 203.138.2.1 icmp_seq=3 ttl=64 time=0.683 ms
84 bytes from 203.138.2.1 icmp_seq=4 ttl=64 time=0.845 ms
84 bytes from 203.138.2.1 icmp_seq=5 ttl=64 time=0.721 ms
```

图 4-6 PC3 ping PC1 结果图

从结果显示, PC1 和 PC3 可以互相 ping 通。现在来分析 PC1 ping PC3 数据包的传输过程以及交换机的工作流程:

PC1 发送数据包给交换机 Switch2 的 e1 端口, e1 端口是 access 口, vlan 号为 2。e1 端口检查 PC1 的数据包上有没有 vlan 标签——没有, 打上 vlan1 的标签, 并进行转发。Switch2 里允许 vlan2 通过的有 e2 和 e0 端口, 这两个端口转发该数据包。e0 端口为 trunk 口, 标签为 vlan1, e0 端口在发送这个数据包时, 首先将数据包的标签和自己的标签进行比较——不一样, 直接转发。因此数据包从 Switch2 转发给 Switch1。Switch1 的 e1 端口收到这个数据包。e1 端口为 trunk 口, trunk 口接收数据包时会先判断有没有 vlan 信息, 发现有标签信息——判断 vlan2 能进入, 并给 e2 端口转发, e2 端口直接转发该数据包给 Switch3。Switch3 收到这个数据包后检查数据包上的标签: vlan2 标签, 并将该数据包从 vlan2 的 access 口 e0 转发。

其中, Trunk 口的收发报文的工作流程可以总结如下:

在 Trunk 端口上发送报文时, 先会将要发送报文的 vlan 标记与 Trunk 端口的 PVID 进行比较, 如果 PVID 相等, 则从报文中去掉 VLAN 标记再发送; 如果与

PVID 不相等, 则直接发送。这样一来, 如果将交换机级联端口都设置为 Trunk, 并允许所有 vlan 通过后, 默认情况下除了 vlan1 外的所有来自其他 vlan 中的报文将直接发送 (因为这些 vlan 不是 trunk 端口的默认 vlan), 而作为 trunk 端口默认 vlan 的 vlan1, 则需要通过去掉报文中的 vlan 信息后再发送。

在 Trunk 端口收到一个报文时, 会首先判断是否有 vlan 信息: 如果没有 vlan 标记, 则打上该 trunk 端口的 pvid, 视同该帧是来自 pvid 所对应的 vlan 转发到 PVID 所对应的 vlan 接口上; 如果有 vlan 标记, 判断该 Trunk 端口是否允许该 VLAN 的报文进入, 如果允许则直接转发, 否则丢弃。

4.3 第三部分 不同 VLAN 互联

4.3.1 方法 1

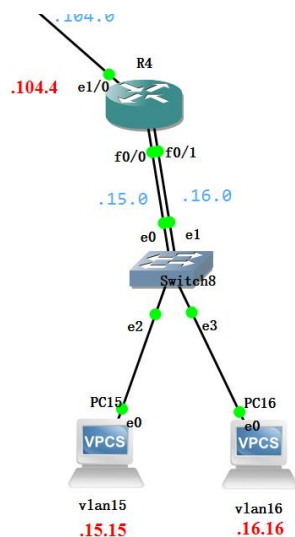


图 4-7 方法 1

使用方法 1 配置路由器, 如上图所示, 尝试 ping 通 PC15 和 PC16, 实验结果如下所示:

```
PC15> ping 203.138.16.16
84 bytes from 203.138.16.16 icmp_seq=1 ttl=63 time=46.448 ms
84 bytes from 203.138.16.16 icmp_seq=2 ttl=63 time=31.072 ms
84 bytes from 203.138.16.16 icmp_seq=3 ttl=63 time=31.432 ms
84 bytes from 203.138.16.16 icmp_seq=4 ttl=63 time=31.383 ms
84 bytes from 203.138.16.16 icmp_seq=5 ttl=63 time=31.279 ms
```

图 4-8 方法 1 PC15 ping PC16 实验结果

```
PC16> ping 203.138.15.15
203.138.15.15 icmp_seq=1 timeout
203.138.15.15 icmp_seq=2 timeout
84 bytes from 203.138.15.15 icmp_seq=3 ttl=63 time=31.386 ms
84 bytes from 203.138.15.15 icmp_seq=4 ttl=63 time=30.835 ms
84 bytes from 203.138.15.15 icmp_seq=5 ttl=63 time=30.643 ms
```

图 4-9 方法 1 PC16 ping PC15 实验结果

```
Gateway of last resort is not set

C    203.138.16.0/24 is directly connected, FastEthernet0/1
C    203.138.15.0/24 is directly connected, FastEthernet0/0
R4#
```

图 4-10 R4 路由表

PC15 和 PC16 成功 ping 通。

结果分析：

以 PC15 发送数据包给 PC16 为例，数据包的收发过程如下：

PC15 发送数据包给 Switch8 的 e2 端口，e2 端口打上 vlan15 标签，并通过 e0 端口转发，e0 端口是剥离了 vlan15 的标签后发送。路由器 f0/0 收到数据包后，通过子网掩码和目的地按位与并查询路由表，判断要用哪个端口转发——f0/1 端口。数据包由 f0/1 端口发送后 Switch8 的 e1 端口收到报文后，access 口 e1 打上 vlan16 的标签，并发送给 e3，e3 去标签转发给 PC16。

4.3.2 方法 2

这一小部分实验通过 R1 来完成，R1 底下的网络拓扑如图所示：

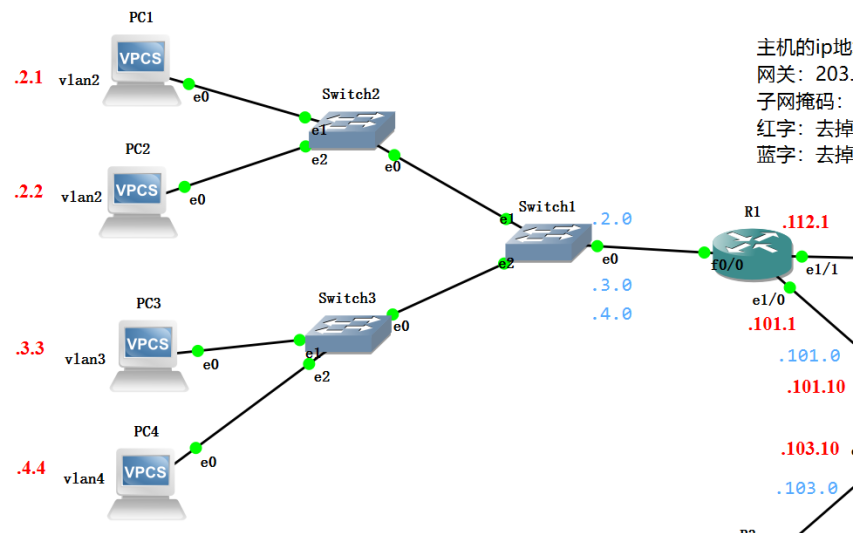


图 4-11 R1 的网络结构图

需要测试的 PC2 和 PC4，他们隶属于不同的 vlan，且路由器和交换机通过 trunk 口，dot1q 协议连接，路由器的物理接口被划分成多个子接口。

实验结果如下所示：

```
PC2> ping 203.138.4.4
203.138.4.4 icmp_seq=1 timeout
203.138.4.4 icmp_seq=2 timeout
84 bytes from 203.138.4.4 icmp_seq=3 ttl=63 time=30.610 ms
84 bytes from 203.138.4.4 icmp_seq=4 ttl=63 time=29.825 ms
84 bytes from 203.138.4.4 icmp_seq=5 ttl=63 time=30.039 ms
PC2>
```

图 4-12 PC2 ping PC4 实验结果

```
PC4> ping 203.138.2.2
84 bytes from 203.138.2.2 icmp_seq=1 ttl=63 time=29.689 ms
84 bytes from 203.138.2.2 icmp_seq=2 ttl=63 time=30.366 ms
84 bytes from 203.138.2.2 icmp_seq=3 ttl=63 time=30.767 ms
84 bytes from 203.138.2.2 icmp_seq=4 ttl=63 time=30.450 ms
84 bytes from 203.138.2.2 icmp_seq=5 ttl=63 time=30.960 ms
PC4>
```


图 4-13 PC4 ping PC2 实验结果

可以看到, PC2 和 PC4 都互相 ping 通了, 不同的 vlan 通过一个路由器实现了互联互通。

结果分析:

以 PC2 ping PC4 为例, 成功 ping 通的数据包的传输流程如下:

PC2 发送数据包给 Switch2, Switch2 的 e2 接口接收数据包, e2 为 access 口, access 接收到无标签包会打上标签, 并给 trunk 口 e0 转发。e0 转发时由于标签 2 和标签 1 不一样, 因此直接发送。Switch1 收到标签 2 的报文, 直接从 e0 端口转发给 R1。R1 收到后首先查看是否有 vlan 标签——有, 那就看 f0/0 哪个子接口能够处理该帧——f0/0.2 封装的是 dot1q 2, 可以处理。因此将这个数据帧的标签从帧里剥离。然后通过路由表, 查看要从哪个接口转发。在路由器转发之前, 路由器会将该帧重新封装, 由于要发往 203.138.4.0 网络, 因此打上标签 4, 再发给 Switch1。Switch1 的 e0 收到标签为 4 的数据帧直接转发。Switch3 的 e0 口收到后交给 e2 转发, e2 口剥离标签后发送给 PC4。

PC4 回复时先发送回应报文给 Switch3 的 e2 口, e2 口为 access 口, 收到无标签的报文会打上 vlan 4 的标签, 然后交给能处理 vlan4 的端口处理该报文。e0 端口可以处理 vlan4, 因为它是 trunk 口, e0 转发时先查看该报文标签和自己的一不一样——一个 4 一个 1, 不一样, 直接转发给 Switch1, Switch1 都是 trunk 口, 而且都是 vlan 1 的标签, 和 vlan4 不一样, 因此将直接转发给路由器 R1。R1 收到报文后, 首先会解析它的 vlan 标签, 发现是标签 4, 于是看自己哪个逻辑子端口可以处理这个标签——f0/0.4 可以处理。路由器通过报

文解析出目的地址后，查路由表转发该报文。在转发前，会给该报文打上目的地 vlan2 的标签。类似的，该报文经过几个 trunk 口后到达 Switch1 的 e2 端口，e2 为 trunk 口，会剥离标签再转发给 PC2。

要注意的是，vlan 标签并不会进入到路由器中。

4.3.3 跨路由器互联

配置好所有的路由器后，尝试使用 PC1 去 ping PC16、PC8，结果如下：

```
PC1> ping 203.138.16.16
203.138.16.16 icmp_seq=1 timeout
203.138.16.16 icmp_seq=2 timeout
84 bytes from 203.138.16.16 icmp_seq=3 ttl=61 time=90.724 ms
84 bytes from 203.138.16.16 icmp_seq=4 ttl=61 time=91.223 ms
84 bytes from 203.138.16.16 icmp_seq=5 ttl=61 time=91.406 ms

PC1> ping 203.138.8.8
203.138.8.8 icmp_seq=1 timeout
203.138.8.8 icmp_seq=2 timeout
84 bytes from 203.138.8.8 icmp_seq=3 ttl=62 time=60.281 ms
84 bytes from 203.138.8.8 icmp_seq=4 ttl=62 time=60.786 ms
84 bytes from 203.138.8.8 icmp_seq=5 ttl=62 time=60.288 ms
```

图 4-14 PC1 ping PC16、PC8 实验结果

如上图所示，PC1 成功 ping 通了 PC16、PC8。

结果分析：

以成功的 PC1 ping PC16 为例。数据包从 PC1 到 R1 的传输过程在上文中已经复述多次，不再赘述。这里主要描述一下路由器处理带 vlan 标签的数据报文的过程。

路由器的物理接口收到一个带有 dot1Q VLAN 标签的数据包。这个标签包含 VLAN 信息，指示数据包属于哪个 VLAN。之后 R1 根据 dot1q 协议提取出报头的 VLAN 标签，确定数据包所属的 VLAN。R1 根据解析出的 VLAN ID 将数据包分配给相应的子接口。每个子接口通常与一个特定的 VLAN 相关联，将

数据包引导到正确的逻辑接口 f0/0.2 上。由于 R1 使用 OSPF 协议维护一个路由表，其中包含了网络的路由信息。当数据包到达正确的子接口后，路由器会查找路由表，确定应该将数据包发送到哪个下一跳或目标。路由表如下图所示：

```
C 203.138.112.0/24 is directly connected, Ethernet1/1
O 203.138.7.0/24 [110/11] via 203.138.112.2, 00:22:29, Ethernet1/1
O 203.138.6.0/24 [110/11] via 203.138.112.2, 00:22:29, Ethernet1/1
O 203.138.5.0/24 [110/11] via 203.138.112.2, 00:22:29, Ethernet1/1
C 203.138.4.0/24 is directly connected, FastEthernet0/0.4
C 203.138.3.0/24 is directly connected, FastEthernet0/0.3
C 203.138.101.0/24 is directly connected, Ethernet1/0
C 203.138.2.0/24 is directly connected, FastEthernet0/0.2
O 203.138.16.0/24 [110/21] via 203.138.101.10, 00:11:28, Ethernet1/0
O 203.138.103.0/24 [110/20] via 203.138.101.10, 00:14:05, Ethernet1/0
O 203.138.102.0/24 [110/20] via 203.138.112.2, 00:22:29, Ethernet1/1
[110/20] via 203.138.101.10, 00:14:05, Ethernet1/0
O 203.138.15.0/24 [110/21] via 203.138.101.10, 00:11:28, Ethernet1/0
O 203.138.104.0/24 [110/20] via 203.138.101.10, 00:13:56, Ethernet1/0
O 203.138.14.0/24 [110/21] via 203.138.101.10, 00:13:56, Ethernet1/0
O 203.138.13.0/24 [110/21] via 203.138.101.10, 00:13:56, Ethernet1/0
O 203.138.12.0/24 [110/21] via 203.138.101.10, 00:13:57, Ethernet1/0
O 203.138.11.0/24 [110/21] via 203.138.101.10, 00:13:57, Ethernet1/0
O 203.138.10.0/24 [110/21] via 203.138.101.10, 00:13:57, Ethernet1/0
O 203.138.9.0/24 [110/21] via 203.138.101.10, 00:13:57, Ethernet1/0
O 203.138.8.0/24 [110/11] via 203.138.112.2, 00:22:30, Ethernet1/1
R1#
```

图 4-15 R1 的路由表

R1 根据路由表的查找结果，决定将数据包转发到相应的下一跳或目标——203.138.101.10 e1/0。转发之前，R1 将数据包重新封装，由于在路由器之间传输不需要 vlan 标签，因此将其发送到相应的物理接口。

数据包经过 R1、R0 到达 R4 后，同样，R4 会查找路由表，并重装这个数据帧，打上 16 的标签并转发。

五、实验中的问题及心得

(1) 一开始我使用 GNS3 作为我的仿真工具。但是在配 GNS3 的时候卡了半天。首先是 GNS3 的虚拟机老是装不成功，我又决定不装虚拟机，直接在本机上跑。但是老是出现 wait for localhost 的弹窗，让我烦不胜烦。我删掉重装

了好几次 GNS3，最后沉下心来成功安装了虚拟机，并且设置好了环境，也导入了所需要的路由器，这才得以安心实验。

(2) 在 GNS3 软件上，我被一个小 bug 折磨了好久。一开始，我想知道 GNS3 的简单工作流程，我设计了一个简单的拓扑，两台 PC 机和一台路由器，我设置好了 IP 和网关，但是两台 PC 怎么样也 ping 不通。我上网找也没找到有相关的问题，我还以为是我安装软件忘记配置什么东西了。

我想着抓包分析一下包在哪里被丢掉了，但是在抓包时我又犯难了。打开 Wireshark，几乎所有的网口我都尝试了一遍，但是都没有找到我想看到的包。机缘巧合之下，我在 GNS3 界面中右键点击了一条相连的线，看到了 start capture 的选项，这回才成功抓包了。

对于一开始设计的简单网络，我看了从 PC1 到 PC2 的数据包和 PC2 到 PC2 的数据包，发现 PC2 去 ping PC1 的时候根本没发数据包出来。我检查了好几次 PC2 的配置，觉得都没问题，有理由怀疑 PC2 坏了。我又拉出了一个 PC3，放到该网络里，这回 PC1 ping PC3 就成功了，PC3 ping PC1 也成功了，但是 PC2 还是 ping 不通，那就说明是 PC2 坏了，应该是这个软件的小 bug。

(3) 搞好 GNS3 后，我配置好路由器的路由协议后，路由器死活都没有动态路由信息，我而且每个路由器都没有。我测试了在同一个路由器下不通 vlan 号之间的互联，是成功的，但是跨路由器的时候就失败了。当时请教了很多，有人认为是我路由器中间又用交换机连起来导致的，但是我把拓扑结构修改后还是不行。最后电脑重启，重启程序，解决了，服了。

(4) 还有一个问题是由于我没有搞懂 trunk 口转发的过程，导致在相同的局域

网内，相同的 vlan 内互相 ping 不通，如下：

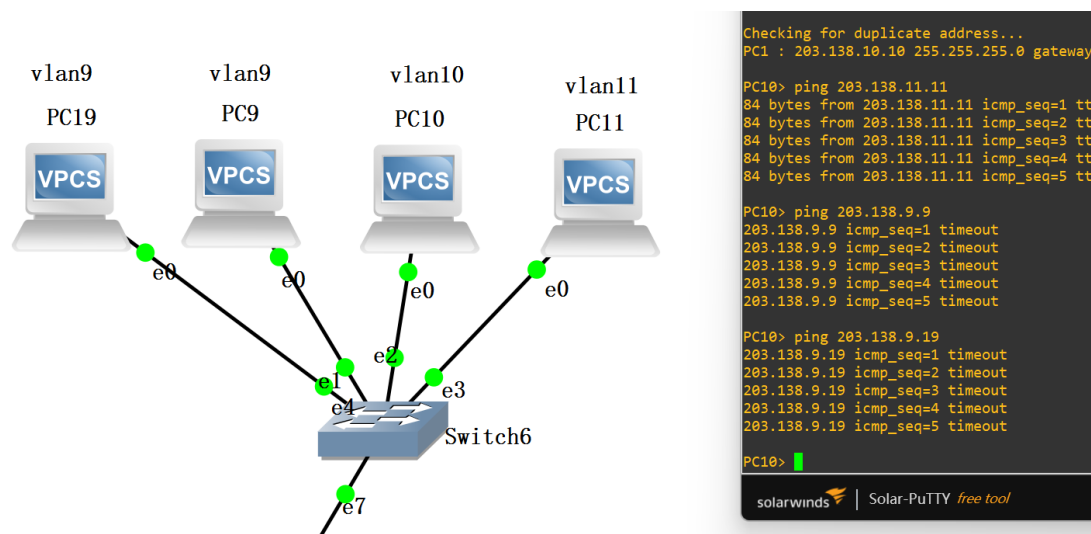


图 5-1 相同局域网内联不通示意图

我想用同一个 vlan 号的 PC10 去 ping PC9，但是 ping 不通。原因如下：

我把 Switchc6 的 e7 端口 vlan 标签号设置成 vlan9 的标签。IP 包从 PC10 到路由器，从路由器到 PC9 之间，从路由器发回的包经过 e7 trunk 口，由于 e7 的标签为 9，和报文里的 vlan 标签相同，因此 trunk 口会去标签再转发。这样 trunk 口收到的报文就不知道发往哪一个端口了（因为我的其他端口都是 access 口），就导致 PC9 接收不到 PC10 的请求。

(5) 这次实验使我大致掌握了 GNS3 的使用

(6) 我现在清楚的知道了数据包在一个实际 vlan 网络的传输流程，也知道了交换机和路由器处理带 vlan 标签的数据包的方法及过程。trunk 口和 access 口不同，且路由器也有两种连接 vlan 的方式，这些我都掌握了。

六、实验思考

6.1 同个局域网配置不同 IP 网段

题目：如何在同一个局域网中，配置两个 IP 网段？（要求这两个网段的设备可以互相 ping 通，采用两种以上的配置方法）

解答如下：

第一种：

首先我们要知道两点：

1. Ping 的时候没有子网掩码，不知道目标的确切网络号。
2. 路由表在查表时不是根据目标的子网掩码来找目标的网络，而是通过某一表项的子网掩码来推算网络号。

根据这两点可以获得我们的一个答案，请看下例。

有如下两个 ip 地址和子网掩码：

PC1: 1.1.0.2 255.0.0.0

PC2: 1.1.1.2 255.255.0.0

上例中 PC1 ping PC2 用自己的子网掩码推算 PC2 的网络是不是一样：用 255.0.0.0 跟 ip2 来与运算：发现一样，因此任务 PC2 和自己在同一个网络里，就直接发送数据包过去。发过去了 PC2 发回来的时候也是，用 ip1 和 PC2 的子网掩码相与，发现网络号一样，便传回去了。

第二种：

在 PC 机上加一条接口转发的路由信息，就能 ping 通。

PC2 2.1.1.1 255.0.0.0

PC3 3.1.1.1 255.0.0.0

PC2 ip route 3.0.0.0 255.0.0.0 f0/0

```
PC3 ip route 2.0.0.0 255.0.0.0 f0/0
```

这里假设 PC2 ping PC3。PC2 初始时不知道 ping PC3 的下一跳地址要往哪里发送，因此会发送一个 arp 广播包。arp 解析的 ip 地址就是对方的 ip。由于在同一个网络里，arp 广播使得 PC3 得到 PC2 的 mac，告诉 PC2 可以往我这里。PC2 成功发送数据包到 PC3。可是当 PC3 要发回去时，不知道 PC2，又一次发送 arp 广播包，PC2 通过广播包获取 PC3 的请求，并告诉 PC3 往我这里发送。这样 PC3 的回应报文也能成功被 PC2 接收。

6.2 分析数据包传输流程

题目：选择自己拓扑中两个不同 VLAN 中的 PC 机，中间要经过 trunk 链路连接的的路由器，阐述互相 ping 时的完整传输流程。（包括交换机和路由器的简单处理过程，并且要指出数据包中 VLAN 标签的变化过程）

解答：

这一小部分实验通过 R1 来完成，R1 底下的网络拓扑如图所示：

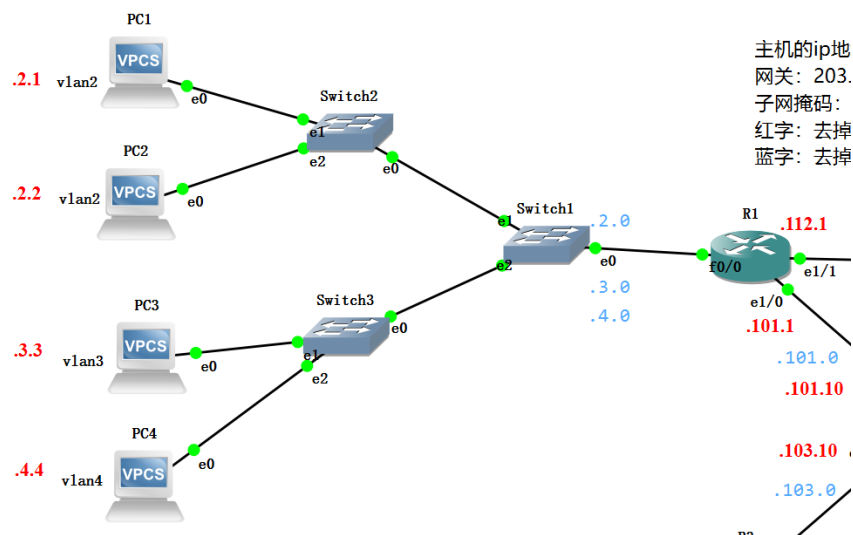


图 6-1 R1 的网络结构图

需要测试的 PC2 和 PC4，他们隶属于不同的 vlan，且路由器和交换机通过

trunk 口，dot1q 协议连接，路由器的物理接口被划分成多个子接口。

实验结果如下所示：

```
PC2> ping 203.138.4.4
203.138.4.4 icmp_seq=1 timeout
203.138.4.4 icmp_seq=2 timeout
84 bytes from 203.138.4.4 icmp_seq=3 ttl=63 time=30.610 ms
84 bytes from 203.138.4.4 icmp_seq=4 ttl=63 time=29.825 ms
84 bytes from 203.138.4.4 icmp_seq=5 ttl=63 time=30.039 ms

PC2> █
```

图 6-2 PC2 ping PC4 实验结果

```
PC4> ping 203.138.2.2
84 bytes from 203.138.2.2 icmp_seq=1 ttl=63 time=29.689 ms
84 bytes from 203.138.2.2 icmp_seq=2 ttl=63 time=30.366 ms
84 bytes from 203.138.2.2 icmp_seq=3 ttl=63 time=30.767 ms
84 bytes from 203.138.2.2 icmp_seq=4 ttl=63 time=30.450 ms
84 bytes from 203.138.2.2 icmp_seq=5 ttl=63 time=30.960 ms

PC4> █
```

图 6-3 PC4 ping PC2 实验结果

可以看到，PC2 和 PC4 都互相 ping 通了，不同的 vlan 通过一个路由器实现了互联互通。

结果分析：

以 PC2 ping PC4 为例，成功 ping 通的数据包的传输流程如下：

PC2 发送数据包给 Switch2, Switch2 的 e2 接口接收数据包, e2 为 access 口，access 接收到无标签包会打上标签 2，并给 trunk 口 e0 转发。e0 转发时由于标签 2 和标签 1 不一样，因此直接发送。Switch1 收到标签 2 的报文，直接从 e0 端口转发给 R1。R1 收到后首先查看是否有 vlan 标签——有，那就看 f0/0 哪个子接口能够处理该帧——f0/0.2 封装的是 dot1q 2，可以处理。因此将这个数据帧的标签从帧里剥离。然后通过路由表，查看要从哪个接口转发。

在路由器转发之前, 路由器会将该帧重新封装, 由于要发往 203.138.4.0 网络, 因此打上标签 4, 再发给 Switch1。Switch1 的 e0 收到标签为 4 的数据帧直接转发。Switch3 的 e0 口收到后交给 e2 access 口转发, e2 口剥离标签后发送给 PC4。

PC4 回复时先发送回应报文给 Switch3 的 e2 口, e2 口为 access 口, 收到无标签的报文会打上 vlan 4 的标签, 然后交给能处理 vlan4 的端口处理该报文。e0 端口可以处理 vlan4, 因为它是 trunk 口, e0 转发时先查看该报文标签和自己的一不一样——一个 4 一个 1, 不一样, 直接转发给 Switch1, Switch1 都是 trunk 口, 而且都是 vlan 1 的标签, 和 vlan4 不一样, 因此将直接转发给路由器 R1。R1 收到报文后, 首先会解析它的 vlan 标签, 发现是标签 4, 于是看自己哪个逻辑子端口可以处理这个标签——f0/0.4 可以处理。路由器通过报文解析出目的地址后, 查路由表转发该报文。在转发前, 会给该报文打上目的地 vlan2 的标签。类似的, 该报文经过几个 trunk 口后到达 Switch1 的 e2 端口, e2 为 trunk 口, 会剥离标签再转发给 PC2。

要注意的是, vlan 标签并不会进入到路由器中。

6.3 物理网络、VLAN、IP 网段的关系

题目: 请阐述物理网络、VLAN 及 IP 网段的关系, 说明路由器是如何把不同物理网络连通的。

解答:

Vlan 是在同一个物理网络再进行划分

不同的物理网路可以划分 vlan1、2、3、4, 不同物理网路相同 vlan 标签的

虚拟网络是独立的

每个 VLAN 可以有自己的 IP 网段，即一个 VLAN 对应一个 IP 网段。

路由器通常被用于连接不同的物理网络或不同的 VLAN，以实现跨网络的通信。

路由器通常有多个物理接口，每个接口连接到一个不同的物理网络。这些物理接口可以是以太网口、无线接口等。如果在路由器上使用 VLAN 划分，路由器的接口可以配置为多个子接口，每个子接口关联一个 VLAN。这样，路由器可以在同一个物理接口上连接多个逻辑上独立的 VLAN。对于每个子接口，路由器会分配一个 IP 地址，该 IP 地址属于子接口所关联的 VLAN 的 IP 网段。这样，路由器就能够在不同的 IP 网段之间进行路由。路由器维护一个路由表，用于决定将数据包从一个接口转发到另一个接口的方式。当路由器收到一个数据包时，它会根据目标 IP 地址查找路由表，确定应该将数据包转发到哪个接口。路由器会检查目标 IP 地址所属的网络，然后将数据包转发到适当的接口，实现不同物理网络之间的通信。