

# 北京邮电大学



## 计算机网络技术实践

实验报告： 交换机基本配置和 VLAN

学院： 计算机学院（国家示范性软件学院）

专业： 计算机科学与技术

班级： 2022211305

学号： 2022211683

姓名： 张晨阳

2024 年 12 月 18 号

# 目录

1.	环境.....	3
2.	第一部分 .....	3
2.1.	实验拓扑 .....	3
2.2.	实验步骤 .....	5
3.	第二部分 .....	7
3.1.	实验拓扑 .....	7
3.2.	实验步骤 .....	7
4.	第三部分 .....	9
4.1.	实验拓扑 .....	9
4.2.	配置 PC.....	16
4.3.	配置路由器 .....	17
4.4.	配置交换机 .....	19
4.5.	互 ping 结果 .....	21
5.	实验中的问题及心得 .....	22
6.	实验思考 .....	23

# 1.环境

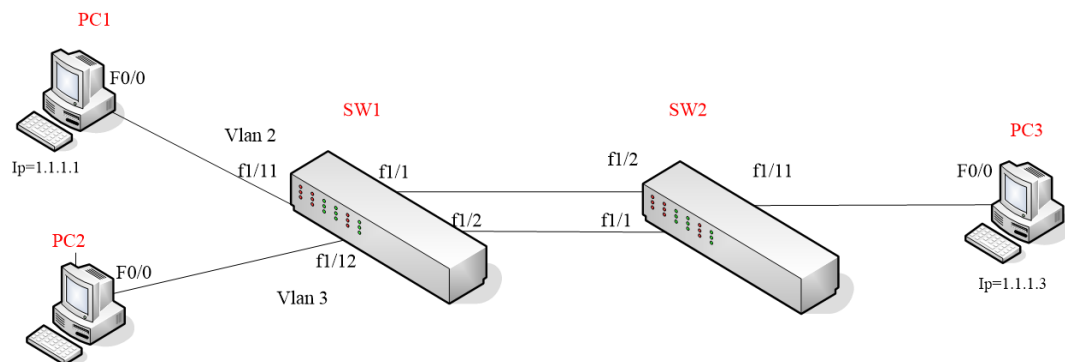
操作系统: Windows11

网络平台: 仿真软件 Dynamips

## 2.第一部分

### 2.1. 实验拓扑

此部分无需自己设计，给出的拓扑结果如下：



拓扑文件为：

```
1. autostart = False
2.
3. [localhost]
4.     port = 7200
5.     udp = 10000
6.     workingdir = ..\tmp\
7.
8.     [[router SW1]]
9.         image = ..\ios\unzip-c3640-js-mz.124-10.bin
10.        model = 3640
11.        console = 3003
12.        ram = 128
13.        confreg = 0x2102
14.        exec_area = 64
15.        mmap = False
16.        slot1 = NM-16ESW
17.        f1/1 = SW2 f1/2
```

```
18.         f1/2 = SW2 f1/1
19.
20.         f1/11 = PC1 f0/0
21.         f1/12 = PC2 f0/0
22.
23.     [[router SW2]]
24.         image = ..\ios\unzip-c3640-js-mz.124-10.bin
25.         model = 3640
26.         console = 3004
27.         ram = 128
28.         confreg = 0x2102
29.         exec_area = 64
30.         mmap = False
31.         slot1 = NM-16ESW
32.         f1/11 = PC3 f0/0
33.
34.
35.     [[router PC1]]
36.         model = 2621
37.         ram = 20
38.         image = ..\ios\unzip-c2600-i-mz.121-3.T.bin
39.         mmap = False
40.         confreg = 0x2102
41.         console = 3006
42.
43.     [[router PC2]]
44.         model = 2621
45.         ram = 20
46.         image = ..\ios\unzip-c2600-i-mz.121-3.T.bin
47.         mmap = False
48.         confreg = 0x2102
49.         console = 3007
50.
51.     [[router PC3]]
52.         model = 2621
53.         ram = 20
54.         image = ..\ios\unzip-c2600-i-mz.121-3.T.bin
55.         mmap = False
56.         confreg = 0x2102
57.         console = 3008
```

## 2.2. 实验步骤

### 1. 启动模拟器并指定 idlepc 值

第一次打开一类机型的设备中的一台的时候，我们需要为其指定其 idlepc 值用于后续的匹配，获取 idlepc 值并将之保存。

由于此前实验三已进行过该操作，所以在此不多赘述。直接给出相应的指令：

```
idlepc get R1
idlepc save R1 db
```

其中，idlepc 选择标记\*符号的即可。

最后启动所有设备。

### 2. 配置 IP

将 PC 的 F0/0 接口配置为其对应 IP。下面以 PC1 为例，剩余 PC 配置方法同理：

```
Router> en
Router# conf
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface f0/0
Router(config-if)# ip add 1.1.1.1 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# ip route 0.0.0.0 0.0.0.0 f0/0
Router(config)# exit
Router# wr
Building configuration...
```

### 3. 配置 Vlan

我们需要将 SW1 的端口 f1/11 配成 Vlan 2，端口 f1/12 配成 Vlan 3。

相关配置指令如下：

```
Router> en
Router# vlan database
Router(vlan)# vlan 2
VLAN 2 added:
    Name: VLAN0002
Router(vlan)# vlan 3
VLAN 3 added:
    Name: VLAN0003
Router(vlan)# exit
```

```
Router# conf
Router(config)# interface vlan 2
Router(config-if)# exit
Router(config)# interface vlan 3
Router(config-if)# exit

Router(config)# interface f1/11
Router(config-if)# switchport access vlan 2
Router(config-if)# exit
Router(config)# interface f1/12
Router(config-if)# switchport access vlan 3
Router(config-if)# exit
```

其中，interface 语句用于激活 Vlan。

#### 4. 不同 VLAN 互相 ping 通

我们在 PC1 执行命令：ping 1.1.1.2

结果如下：

```
Router# ping 1.1.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

发现此时处于不同 VLAN 下的两个 PC 无法 ping 通。

#### 5. 相同 VLAN 互相 ping 通

首先要修改 f1/12 接口的 VLAN 为 2：

```
Router(config)# interface f1/12
Router(config-if)# switchport access vlan 2
Router(config-if)# exit
```

同样的，我们在 PC1 执行命令：ping 1.1.1.2

结果如下：

```
Router# ping 1.1.1.2

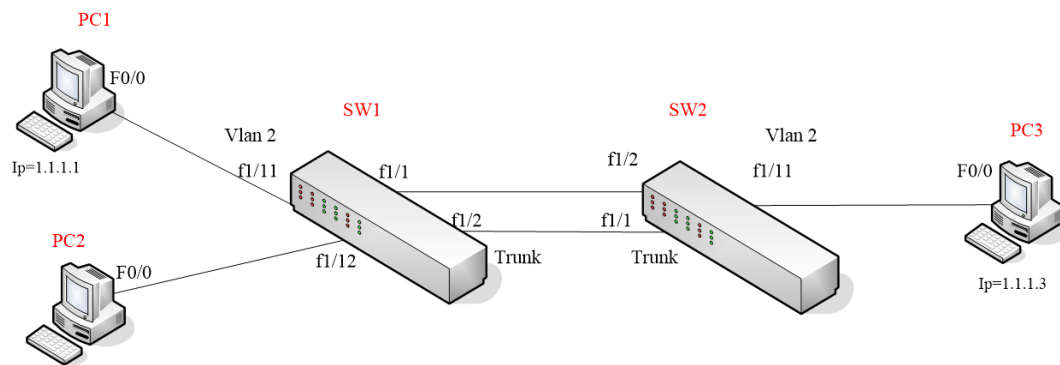
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.2, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/11/20 ms
```

发现此时可以 ping 通。

## 3. 第二部分

### 3.1. 实验拓扑

该部分的实验拓扑与上一部分相同：



只是连接方式的区别，故不再展示拓扑文件。

### 3.2. 实验步骤

#### 1. 配置 SW2 和 PC3

首先 telnet 到 PC3 并分配 IP 为 1.1.1.3

```
Router(config)# interface f0/0
Router(config-if)# ip add 1.1.1.3 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# ip route 0.0.0.0 0.0.0.0 f0/0
```

然后配置 SW2，将对应的 f1/11 接口配置好 VLAN2：

```
Router> en
Router# vlan database
Router(vlan)# vlan 2
VLAN 2 added:
  Name: VLAN0002
Router(vlan)# exit

Router# conf
Router(config)# interface vlan 2
Router(config-if)# exit
```

```
Router(config)# interface f1/11
Router(config-if)# switchport access vlan 2
Router(config-if)# exit
```

## 2. PC1 与 PC3 互 ping

我们在 PC1 执行命令：ping 1.1.1.3

结果如下：

```
Router# ping 1.1.1.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.3, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

此时无论是 PC1 ping PC3 还是 PC3 ping PC1 都无法 ping 通。

## 3. 配置 trunk 模式

为了使得 PC 可以互相 ping 通，我们还需要设置 SW 互联的接口为 trunk 模式。下面是 SW2 的配置过程：

```
Router(config)# interface f1/1
Router(config-if)# Switchport mode Trunk
Router(config-if)# Switchport trunk allowed vlan all
Router(config-if)# exit
```

## 4. PC1 与 PC3 互 ping

此时我们在 PC1 重新 ping PC3：ping 1.1.1.3

结果如下：

```
Router# ping 1.1.1.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.3, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 4/12/32 ms
```

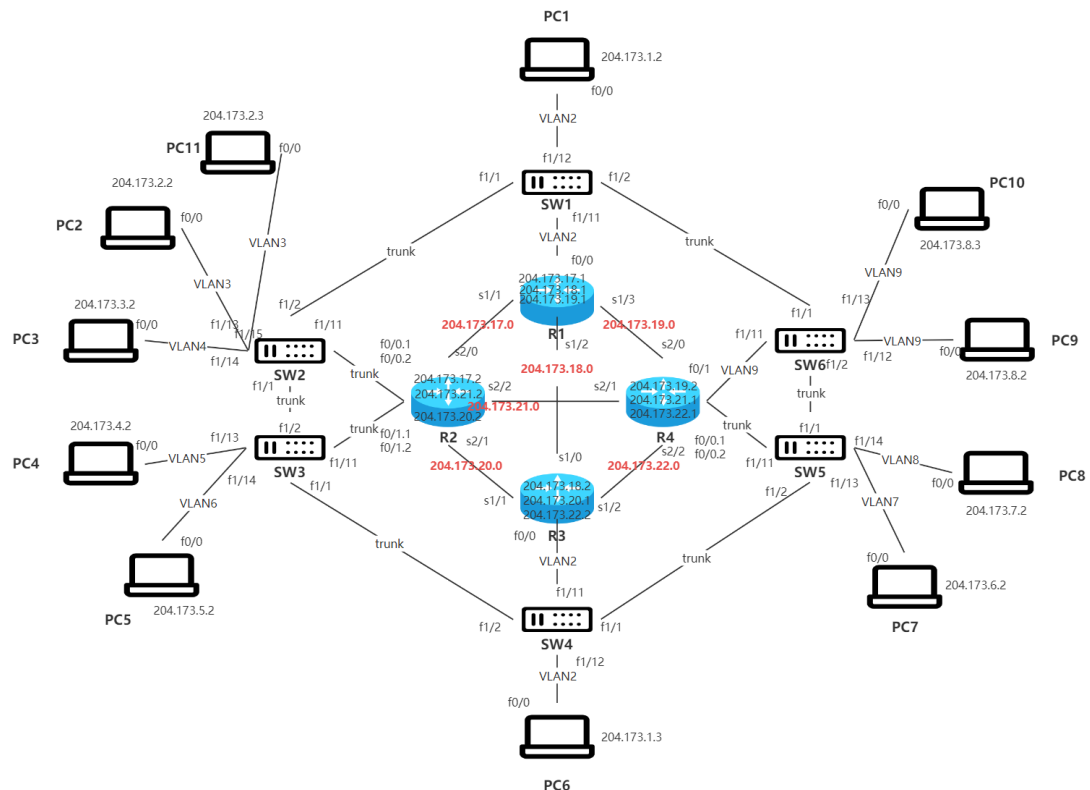
说明了模式 trunk 的可通过性。PC1 与 PC3 已经相连。



## 4.第三部分

### 4.1. 实验拓扑

我自己设计的拓扑结构如下：包含 11 台 PC，6 台交换机，4 台路由器，8 个 VLAN：



所有设备的 VLAN、IP 地址、接口连接、连接模式均已标在图中。

为了便于报告的编写以及报告阅读的方便，且因为已经验收通过，故我们在报告里只以举例的形式给出我们的拓扑图的配置流程和实验结果。

首先给出我们完整的拓扑文件：

```
1. autostart = false
2.
3. [localhost]
4. port = 7200
5. udp = 10000
6. workingdir = .\tmp\
7.
8. [[router R1]]
9. image = ..\ios\unzip-c7200-is-mz.122-37.bin
```

```
10.     model = 7200
11.     console = 3001
12.     npe = npe-400
13.     ram = 64
14.     confreg = 0x2102
15.     exec_area = 64
16.     mmap = false
17.     slot0 = PA-C7200-IO-FE
18.     slot1 = PA-4T
19.     f0/0 = SW1 f1/11
20.     s1/1 = R2 s2/0
21.     s1/2 = R3 s1/0
22.     s1/3 = R4 s2/0
23.
24.
25.     [[router R2]]
26.     image = ..\ios\unzip-c7200-is-mz.122-37.bin
27.     model = 7200
28.     console = 3002
29.     npe = npe-400
30.     ram = 64
31.     confreg = 0x2102
32.     exec_area = 64
33.     mmap = false
34.     slot0 = PA-C7200-IO-2FE
35.     slot1 = PA-2FE-TX
36.     slot2 = PA-4T
37.     f0/0 = SW2 f1/11
38.     f0/1 = SW3 f1/11
39.     s2/0 = R1 s1/1
40.     s2/1 = R3 s1/1
41.     s2/2 = R4 s2/1
42.
43.     [[router R3]]
44.     image = ..\ios\unzip-c7200-is-mz.122-37.bin
45.     model = 7200
46.     console = 3003
47.     npe = npe-400
48.     ram = 64
49.     confreg = 0x2102
50.     exec_area = 64
51.     mmap = false
52.     slot0 = PA-C7200-IO-FE
53.     slot1 = PA-4T
```

```
54.    f0/0 = SW4 f1/11
55.    s1/0 = R1 s1/2
56.    s1/1 = R2 s2/1
57.    s1/2 = R4 s2/2
58.
59.    [[router R4]]
60.    image = ..\ios\unzip-c7200-is-mz.122-37.bin
61.    model = 7200
62.    console = 3004
63.    npe = npe-400
64.    ram = 64
65.    confreg = 0x2102
66.    exec_area = 64
67.    mmap = false
68.    slot0 = PA-C7200-IO-2FE
69.    slot1 = PA-2FE-TX
70.    slot2 = PA-4T
71.    f0/0 = SW5 f1/11
72.    f0/1 = SW6 f1/11
73.    s2/0 = R1 s1/3
74.    s2/1 = R2 s2/2
75.    s2/2 = R3 s1/2
76.
77.    [[router SW1]]
78.        image = ..\ios\unzip-c3640-js-mz.124-10.bin
79.        model = 3640
80.        console = 3005
81.        ram = 128
82.        confreg = 0x2102
83.        exec_area = 64
84.        mmap = False
85.        slot1 = NM-16ESW
86.        f1/11 = R1 f0/0
87.        f1/1 = SW2 f1/2
88.        f1/2 = SW6 f1/1
89.        f1/12 = PC1 f0/0
90.
91.    [[router SW2]]
92.        image = ..\ios\unzip-c3640-js-mz.124-10.bin
93.        model = 3640
94.        console = 3006
95.        ram = 128
96.        confreg = 0x2102
97.        exec_area = 64
```

```
98.         mmap = False
99.         slot1 = NM-16ESW
100.        f1/11 = R2 f0/0
101.        f1/1  = SW3 f1/2
102.        f1/2  = SW1 f1/1
103.        f1/13 = PC2 f0/0
104.        f1/14 = PC3 f0/0
105.        f1/15 = PC11 f0/0
106.
107.    [[router SW3]]
108.        image = ../ios/unzip-c3640-js-mz.124-10.bin
109.        model = 3640
110.        console = 3007
111.        ram = 128
112.        confreg = 0x2102
113.        exec_area = 64
114.        mmap = False
115.        slot1 = NM-16ESW
116.        f1/1  = SW4 f1/2
117.        f1/2  = SW2 f1/1
118.        f1/13 = PC4 f0/0
119.        f1/14 = PC5 f0/0
120.
121.    [[router SW4]]
122.        image = ../ios/unzip-c3640-js-mz.124-10.bin
123.        model = 3640
124.        console = 3008
125.        ram = 128
126.        confreg = 0x2102
127.        exec_area = 64
128.        mmap = False
129.        slot1 = NM-16ESW
130.        f1/11 = R3 f0/0
131.        f1/1  = SW5 f1/2
132.        f1/2  = SW3 f1/1
133.        f1/12 = PC6 f0/0
134.
135.    [[router SW5]]
136.        image = ../ios/unzip-c3640-js-mz.124-10.bin
137.        model = 3640
138.        console = 3009
139.        ram = 128
140.        confreg = 0x2102
141.        exec_area = 64
```

```
142.         mmap = False
143.         slot1 = NM-16ESW
144.         f1/11 = R4 f0/0
145.         f1/1 = SW6 f1/2
146.         f1/2 = SW4 f1/1
147.         f1/13 = PC7 f0/0
148.         f1/14 = PC8 f0/0
149.
150.     [[router SW6]]
151.         image = ..\ios\unzip-c3640-js-mz.124-10.bin
152.         model = 3640
153.         console = 3010
154.         ram = 128
155.         confreg = 0x2102
156.         exec_area = 64
157.         mmap = False
158.         slot1 = NM-16ESW
159.         f1/1 = SW1 f1/2
160.         f1/2 = SW5 f1/1
161.         f1/12 = PC9 f0/0
162.         f1/13 = PC10 f0/0
163.
164.     [[router PC1]]
165.         model = 2621
166.         ram = 20
167.         image = ..\ios\unzip-c2600-i-mz.121-3.T.bin
168.         mmap = False
169.         confreg = 0x2102
170.         console = 3011
171.
172.     [[router PC2]]
173.         model = 2621
174.         ram = 20
175.         image = ..\ios\unzip-c2600-i-mz.121-3.T.bin
176.         mmap = False
177.         confreg = 0x2102
178.         console = 3012
179.
180.     [[router PC3]]
181.         model = 2621
182.         ram = 20
183.         image = ..\ios\unzip-c2600-i-mz.121-3.T.bin
184.         mmap = False
185.         confreg = 0x2102
```

```
186.     console = 3013
187.
188.     [[router PC4]]
189.     model = 2621
190.     ram = 20
191.     image = ../ios/unzip-c2600-i-mz.121-3.T.bin
192.     mmap = False
193.     confreg = 0x2102
194.     console = 3014
195.
196.     [[router PC5]]
197.     model = 2621
198.     ram = 20
199.     image = ../ios/unzip-c2600-i-mz.121-3.T.bin
200.     mmap = False
201.     confreg = 0x2102
202.     console = 3015
203.
204.     [[router PC6]]
205.     model = 2621
206.     ram = 20
207.     image = ../ios/unzip-c2600-i-mz.121-3.T.bin
208.     mmap = False
209.     confreg = 0x2102
210.     console = 3016
211.
212.     [[router PC7]]
213.     model = 2621
214.     ram = 20
215.     image = ../ios/unzip-c2600-i-mz.121-3.T.bin
216.     mmap = False
217.     confreg = 0x2102
218.     console = 3017
219.
220.     [[router PC8]]
221.     model = 2621
222.     ram = 20
223.     image = ../ios/unzip-c2600-i-mz.121-3.T.bin
224.     mmap = False
225.     confreg = 0x2102
226.     console = 3018
227.
228.     [[router PC9]]
229.     model = 2621
```

```
230.     ram = 20
231.     image = ../ios/unzip-c2600-i-mz.121-3.T.bin
232.     mmap = False
233.     confreg = 0x2102
234.     console = 3019
235.
236.     [[router PC10]]
237.     model = 2621
238.     ram = 20
239.     image = ../ios/unzip-c2600-i-mz.121-3.T.bin
240.     mmap = False
241.     confreg = 0x2102
242.     console = 3020
243.
244.     [[router PC11]]
245.     model = 2621
246.     ram = 20
247.     image = ../ios/unzip-c2600-i-mz.121-3.T.bin
248.     mmap = False
249.     confreg = 0x2102
250.     console = 3021
```

## 4.2. 配置 PC

对于每一台 PC，配置流程基本相同：

1. 选择接口；
2. 配置 IP 地址；
3. 配置默认静态路由；

我们以 PC1 为例：

```
Router> en
Router# conf
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface f0/0
Router(config-if)# ip add 204.173.1.2 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# ip route 0.0.0.0 0.0.0.0 f0/0
Router(config)# exit
Router# wr
```



### 4.3. 配置路由器

我们的路由器之间使用 OSPF 协议进行配置。

由于路由器与交换机之间有两种模式，故我们这里以 R4 为例：与 SW6 通过 VLAN9 相连，与 SW5 通过子接口和 trunk 模式相连。

配置流程为：

1. 配置与其他路由器相连的接口和 IP；
2. 配置与交换机连接的子接口（trunk 模式）
3. 配置与交换机连接的接口（VLAN 模式）
4. 配置 OSPF 所需要的路由器连接的网段号。

配置相关命令如下：

```
Router>en
Router#conf
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface s 2/0
Router(config-if)#clock rate 115200
Router(config-if)#ip add 204.173.19.2 255.255.255.0
Router(config-if)#ip ospf hello-interval 5
Router(config-if)#ip ospf dead-interval 20
Router(config-if)#encapsulation PPP
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface s 2/1
Router(config-if)#clock rate 115200
Router(config-if)#ip add 204.173.21.1 255.255.255.0
Router(config-if)#ip ospf hello-interval 5
Router(config-if)#ip ospf dead-interval 20
Router(config-if)#encapsulation PPP
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface s 2/2
Router(config-if)#clock rate 115200
Router(config-if)#ip add 204.173.22.1 255.255.255.0
Router(config-if)#ip ospf hello-interval 5
Router(config-if)#ip ospf dead-interval 20
Router(config-if)#encapsulation PPP
Router(config-if)#no shutdown
Router(config-if)#exit
```

```
Router(config)#interface f 0/0.1
Router(config-if)#encapsulation dot1q 7
Router(config-if)#ip add 204.173.6.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface f 0/0.2
Router(config-if)#encapsulation dot1q 8
Router(config-if)#ip add 204.173.7.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface f 0/1
Router(config-if)#ip add 204.173.8.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#router ospf 20
Router(config-router)#network 204.173.19.0 255.255.255.0 area 0
Router(config-router)#network 204.173.21.0 255.255.255.0 area 0
Router(config-router)#network 204.173.22.0 255.255.255.0 area 0
Router(config-router)#network 204.173.6.0 255.255.255.0 area 0
Router(config-router)#network 204.173.7.0 255.255.255.0 area 0
Router(config-router)#network 204.173.8.0 255.255.255.0 area 0
Router(config-router)#exit
Router(config)#exit
```

上述 IP 地址均已在拓扑图中标出，可按照图片理解可设备的连接情况。

## 4.4. 配置交换机

对于交换机的配置，主要流程为：

1. 加入 VLAN 到 database 中；
2. 激活 VLAN；
3. 配置接口对应的 VLAN；
4. 配置 trunk 模式的接口。

我们以 SW1 为例即可：

```
Router>en
Router#vlan database
Router(vlan)#vlan 2
VLAN 2 added:
    Name: VLAN0002
Router(vlan)#vlan 3
VLAN 3 added:
    Name: VLAN0003
.....
VLAN 9 added:
    Name: VLAN0009
Router(vlan)#exit
APPLY completed.
Exiting....

Router#conf
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface vlan 2
Router(config-if)#exit
Router(config)#interface vlan 3
Router(config-if)#exit
Router(config)#interface vlan 4
Router(config-if)#exit
Router(config)#interface vlan 5
Router(config-if)#exit
Router(config)#interface vlan 6
Router(config-if)#exit
Router(config)#interface vlan 7
Router(config-if)#exit
Router(config)#interface vlan 8
Router(config-if)#exit
```

```
Router(config)#interface vlan 9
Router(config-if)#exit

Router(config)#interface f1/11
Router(config-if)#switchport access vlan 2
Router(config-if)#exit
Router(config)#interface f1/12
Router(config-if)#switchport access vlan 3
Router(config-if)#exit
Router(config)#interface f1/1
Router(config-if)#switchport mode Trunk
Router(config-if)#switchport trunk allowed vlan all
Router(config-if)#exit
Router(config)#interface f1/2
Router(config-if)#switchport mode Trunk
Router(config-if)#switchport trunk allowed vlan all
Router(config-if)#exit
```

重复上述流程直到配置完所有的设备。

## 4.5. 互 ping 结果

由于本报告完成时间晚于验收时间，在验收过程中，关闭了 SW4-SW5-SW6-SW1 这条线，验收时各 PC 均 ping 通，修改结构前与修改结构后均通过。

我们在这里只给出一个示例：PC7 与 PC10 互 ping，以证明我们的结构配置没有问题。

PC7:

```
Dynamips(16): PC7, Console port
Connected to Dynamips VM "PC7" (ID 16, type c2600) - Console port

Router>
Router>en
Router#ping 204.173.8.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 204.173.8.3, timeout is 2 seconds:
..!!!
Success rate is 60 percent (3/5), round-trip min/avg/max = 64/94/112 ms
Router#
```

PC10:

```
Connected to Dynamips VM "PC10" (ID 19, type c2600) - Console port

Router>
Router>
Router>
Router>
Router>en
Router#ping 204.173.6.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 204.173.6.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 48/52/56 ms
Router#
```

## 5.实验中的问题及心得

对于前两个部分，比较简单，没有什么困难或问题。

而对于我自己设计拓扑结构实现第三部分的实验，可以说是挫折颇多。

首先，我明白了拓扑结构并不是越复杂越好，因为可能存在各种原因（包括现实配置），导致我们的设备/环境崩溃，甚至有可能无法承载过于复杂的结构。应该在多次尝试之后，选择合适的拓扑结构，并细心设计接口、网络。

其次，对于配置流程，我得到了教训：要一边配置一边记录，而不是全部完成后，甚至验收完成后才开始撰写报告，导致重复劳动，重复配置已经实现好的网络结构。我也意识到，配置的时候需要细心，应该把每一步记录下来，也便于后续检查自己的配置是不是有问题。

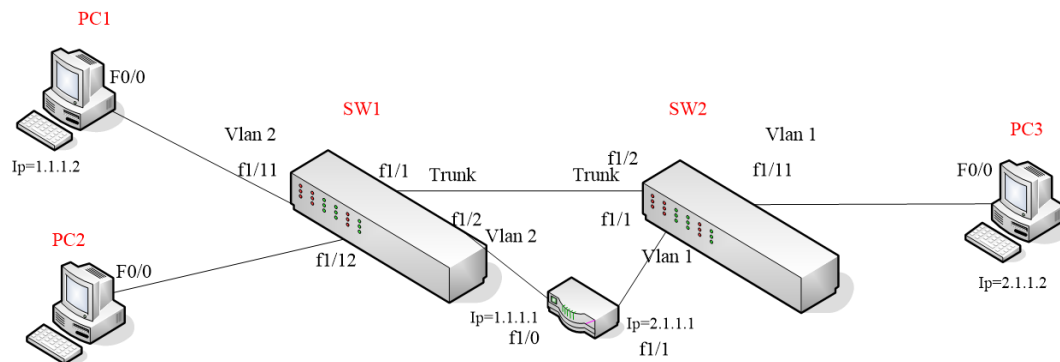
除此之外，我还多次使用了多种 debug 信息，让我跟踪我的 PC 发出的包、路由器的相关配置。

当然，在这个过程中，我也寻求了同学的帮助，自己钻进牛角尖是无法发现自己的配置漏洞的，需要一定的思想碰撞，让我的拓扑结构更加完美。

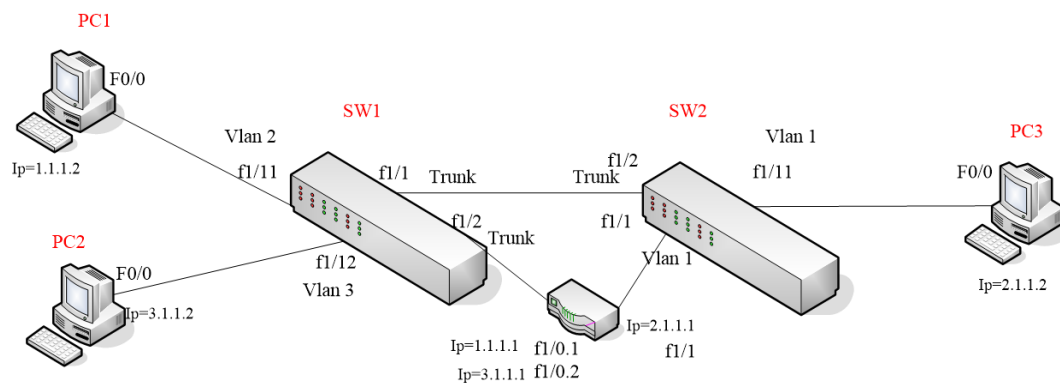
## 6.实验思考

1. 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）

方法一：用一台路由器分别接入两个 VLAN，两个 VLAN 分别对应两个网段的设备，使得不同 VLAN 的设备可以互通；



方法二：采用单臂路由的方式，在两个 IP 网段的包发送给路由器的同一端口，路由器设置子端口来接收不同的网段的包，使得不同 VLAN 的设备可以互通。



2. 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）

PC1 ping PC7:

这是验收过程中，助教提的问题。

前提是已经修改了我的拓扑结构，关闭了 SW1-SW6-SW5-SW4。

那么，PC1 发出包，带有 VLAN2 的标签进入 SW1，SW1 继续将该包去掉标签，发给路由器 R1，R1 接收到该包，查询路由表，将包转发给 R4，R4 接收到包后，也查询自己的路由表，包的标签此时为 VLAN7，通过子接口发给 SW5，SW5 将包转发给 PC7，完成一次传输。之后 PC7 的包也以反向的路径传回 PC1。



3. 请阐述物理以太网、VLAN 及 IP 网段的关系，说明路由器是如何把不同 VLAN 连通的。

- **物理以太网：**指的是计算机网络中硬件层面构建的物理连接，如网线、交换机和路由器等。这些硬件设备通过电缆（如铜线或光纤）将计算机或其他设备互相连接，形成基础的物理网络。
- **VLAN：**通过软件配置在物理以太网络上的一种逻辑网络划分。VLAN 允许将同一物理网络中的设备分组为多个虚拟网络，从而提高网络的管理灵活性和安全性。即使设备位于不同的物理位置或通过不同的交换机连接，使用相同 VLAN 标识符（ID）的设备就像在同一局域网内一样，能够互相通信。
- **IP 网段：**IP 网段（也称为子网）是网络层中的概念，用于为设备分配 IP 地址。在 VLAN 的上下文中，每个 VLAN 通常会对应一个 IP 网段。VLAN 之间的通信通常是通过路由器来完成的，而路由器根据 IP 地址来识别不同的网络并进行路由选择。
- **路由器如何连接不同 VLAN**

路由器通过**三层交换**或**路由功能**来连接不同的 VLAN。具体而言，路由器在多个 VLAN 之间进行 IP 层的转发。这通常是通过**子接口**或 SVI 来实现的。

- **子接口：**在路由器上，为每个 VLAN 创建一个子接口，每个子接口关联一个 VLAN 并分配一个 IP 地址。路由器根据设备的目标 IP 地址，决定是否转发数据包，并将数据包从一个 VLAN 转发到另一个 VLAN。
- **SVI (Switched Virtual Interface)：**在多层交换机中，可以为每个 VLAN 配置一个虚拟接口，每个 VLAN 通过 SVI 与路由器或其他 VLAN 进行通信。