计算机网络技术实践

实验报告

实验名称__VLAN 配置及协议流程分析_

姓　　名_____黎昱彤_____

实　验　日　期：　2024.12.1

学　　号____2022211414____

实 验 报 告 日 期 ：　2024.12.1

报　告　退　发：（ 订正 、 重做 ）

目录

1 实验环境

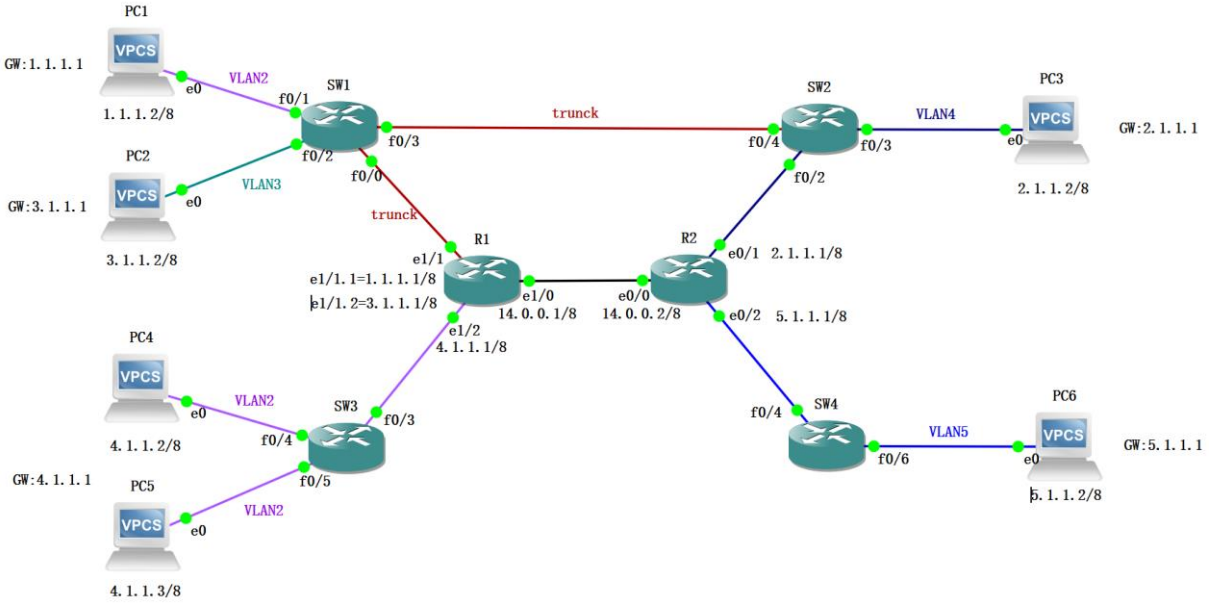
- GNS3
- VMware
- Wireshark
- 交换机用 C3640 路由器模拟

2 实验目的

- 掌握以太网交换机的使用方法，能够在模拟环境中使用以太网交换机组建局域网
- 掌握以太网交换机的 VLAN 划分和配置方法，能够在仿真环境中 使用以太网组建虚拟局域网
- 掌握通过路由器实现不同 VLAN 间互通的方法，学会使用单臂路由通过交换机的 Trunk 接口完成多个 VLAN 间的互通

3 实验配置

以复杂拓扑为例：



- 配置PC IP和网关：

  PC1 # ip 1.1.1.2/8 1.1.1.1

  PC1 # save

- 配置路由器子接口：

（启动物理接口）

  R1 (config) # interface e1/1

  R1 (config-if) # no shutdown

（进入逻辑接口，封装dot1q，配置IP）

  R1 (config) # interface e1/1.1

  R1 (config-if) # encapsulation dot1q 2

  R1 (config-if) # ip address 1.1.1.1 255.0.0.0

- 创建交换机VLAN并接入：

```
SW1#vlan database
SW1(vlan)#vlan 2
VLAN 2 added:
    Name: VLAN0002
SW1(vlan)#vlan 3
VLAN 3 added:
    Name: VLAN0003
SW1(vlan)#exit
APPLY completed.
Exiting....
SW1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
SW1(config)#interface f0/1
SW1(config-if)#switchport access vlan 2
SW1(config-if)#exit
SW1(config)#interface f0/2
SW1(config-if)#switchport access vlan 3
```

- 配置trunk：

  SW1(config)# interface f0/0

SW1(config-if)#switchport trunk encapsulation dot1q

SW1(config-if)#switchport mode trunk

SW1(config-if)#switchport trunk allowed vlan all

复杂拓扑配置结果如下：

```
SW1#show vlan-switch brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/4, Fa0/5, Fa0/6, Fa0/7
                                                Fa0/8, Fa0/9, Fa0/10, Fa0/11
                                                Fa0/12, Fa0/13, Fa0/14, Fa0/15
2    VLAN0002                         active    Fa0/1
3    VLAN0003                         active    Fa0/2
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
```

```
SW1#show interfaces trunk

Port        Mode         Encapsulation  Status        Native vlan
Fa0/0       on           802.1q         trunking      1
Fa0/3       on           802.1q         trunking      1

Port        Vlans allowed on trunk
Fa0/0       1-1005
Fa0/3       1-1005

Port        Vlans allowed and active in management domain
Fa0/0       1-3
Fa0/3       1-3

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/0       1-3
Fa0/3       1-3
```

```
SW2#show vlan-switch brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/0, Fa0/1, Fa0/5, Fa0/6
                                                Fa0/7, Fa0/8, Fa0/9, Fa0/10
                                                Fa0/11, Fa0/12, Fa0/13, Fa0/14
                                                Fa0/15
4    VLAN0004                         active    Fa0/2, Fa0/3
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
```

```
SW3#show vlan-switch brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/0, Fa0/1, Fa0/2, Fa0/6
                                                Fa0/7, Fa0/8, Fa0/9, Fa0/10
                                                Fa0/11, Fa0/12, Fa0/13, Fa0/14
                                                Fa0/15
2    VLAN0002                         active    Fa0/3, Fa0/4, Fa0/5
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
```

```
SW4#show vlan-switch brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/0, Fa0/1, Fa0/2, Fa0/3
                                                Fa0/5, Fa0/7, Fa0/8, Fa0/9
                                                Fa0/10, Fa0/11, Fa0/12, Fa0/13
                                                Fa0/14, Fa0/15
5    VLAN0005                         active    Fa0/4, Fa0/6
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
```

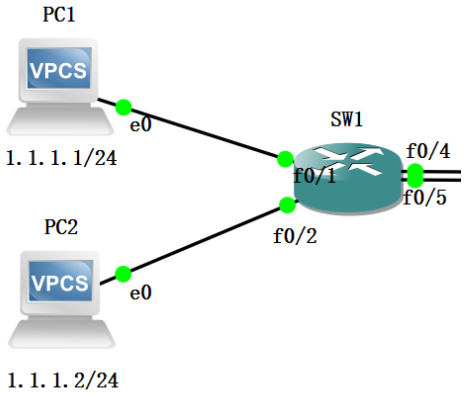
```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, 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 0.0.0.0 to network 0.0.0.0

C    1.0.0.0/8 is directly connected, Ethernet1/1.1
C    3.0.0.0/8 is directly connected, Ethernet1/1.2
C    4.0.0.0/8 is directly connected, Ethernet1/2
C    14.0.0.0/8 is directly connected, Ethernet1/0
S*   0.0.0.0/0 is directly connected, Ethernet1/0
```

```
R2#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 0.0.0.0 to network 0.0.0.0

C    2.0.0.0/8 is directly connected, Ethernet0/1
C    5.0.0.0/8 is directly connected, Ethernet0/2
C    14.0.0.0/8 is directly connected, Ethernet0/0
S*   0.0.0.0/0 is directly connected, Ethernet0/0
```

4 实验内容

4.1 以太网交换机简单组网

1. 用一台交换机连接两台主机组成局域网，并通过在两台主机上的配置完成局域网内部主机之间的互通。



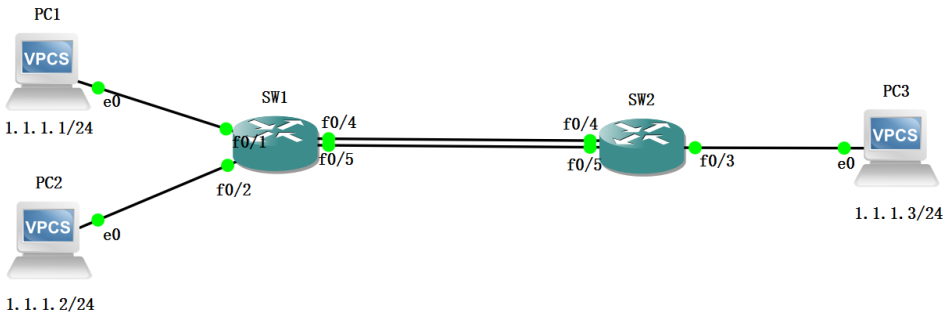
在从 PC1 到 SW1 的链路上启动Wireshark抓包测试：

```
PC1> ping 1.1.1.2

84 bytes from 1.1.1.2 icmp_seq=1 ttl=64 time=0.221 ms
84 bytes from 1.1.1.2 icmp_seq=2 ttl=64 time=0.214 ms
84 bytes from 1.1.1.2 icmp_seq=3 ttl=64 time=0.152 ms
84 bytes from 1.1.1.2 icmp_seq=4 ttl=64 time=0.143 ms
84 bytes from 1.1.1.2 icmp_seq=5 ttl=64 time=0.136 ms
```

2	3.673020	1.1.1.1	1.1.1.2	ICMP	98 Echo (ping) request id=0x5e36, seq=1/256, ttl=64 (reply in 3)
3	3.673129	1.1.1.2	1.1.1.1	ICMP	98 Echo (ping) reply id=0x5e36, seq=1/256, ttl=64 (request in 2)
4	4.674280	1.1.1.1	1.1.1.2	ICMP	98 Echo (ping) request id=0x5f36, seq=2/512, ttl=64 (reply in 5)
5	4.674796	1.1.1.2	1.1.1.1	ICMP	98 Echo (ping) reply id=0x5f36, seq=2/512, ttl=64 (request in 4)
6	4.812826	cc:06:42:99:f0:01	Spanning-tree-(for-…	STP	60 Conf. Root = 32768/0/cc:06:29:10:00:00 Cost = 0 Port = 0x8002
7	5.676017	1.1.1.1	1.1.1.2	ICMP	98 Echo (ping) request id=0x6036, seq=3/768, ttl=64 (reply in 8)
8	5.676148	1.1.1.2	1.1.1.1	ICMP	98 Echo (ping) reply id=0x6036, seq=3/768, ttl=64 (request in 7)
9	6.677325	1.1.1.1	1.1.1.2	ICMP	98 Echo (ping) request id=0x6136, seq=4/1024, ttl=64 (reply in 10)
10	6.677437	1.1.1.2	1.1.1.1	ICMP	98 Echo (ping) reply id=0x6136, seq=4/1024, ttl=64 (request in 9)
11	7.683619	1.1.1.1	1.1.1.2	ICMP	98 Echo (ping) request id=0x6236, seq=5/1280, ttl=64 (reply in 12)
12	7.683762	1.1.1.2	1.1.1.1	ICMP	98 Echo (ping) reply id=0x6236, seq=5/1280, ttl=64 (request in 11)

PC1和PC2能够互相ping通。

2. 用两台交换机将三台主机组成一个局域网，并通过在三台主机上的配置完成局域网内部主机之间的互通



测试：

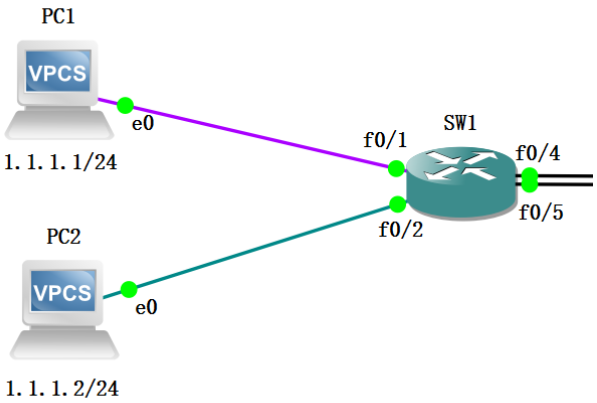
```
PC1> ping 1.1.1.3

host (1.1.1.3) not reachable
```

PC1和PC3不能够互相ping通。

4.2 以太网交换机划分 VLAN

使用两台交换机 SW1 和 SW2 将 3 台主机 PC1、PC2、PC3 互相连接在一起，两台交换机之间采用双链路进行连接，实现链路备份。

1. SW1 连接两台主机 PC1 和 PC2 组成局域网，在 SW1 上建立 VLAN2 和 VLAN3，并将 PC1 和 PC2 配置在同一网段上



测试：

```
PC1> ping 1.1.1.2

host (1.1.1.2) not reachable
```

PC1和PC2不能够互相ping通。

将SW1的端口f0/2配置成VLAN 2

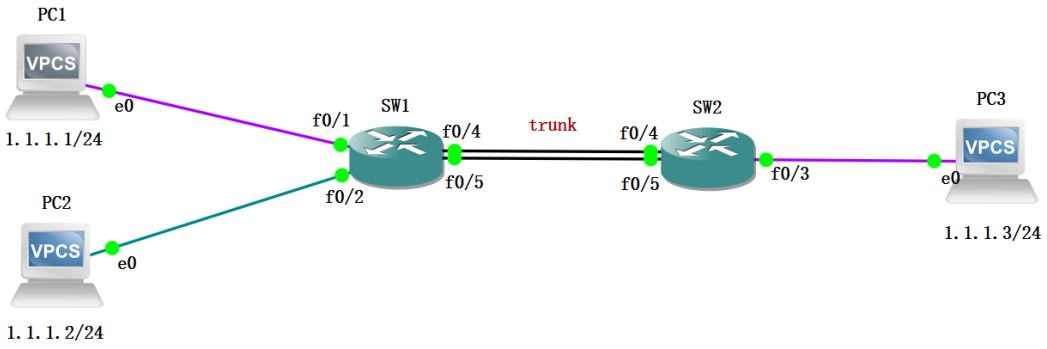
在从 PC1 到 SW1 的链路上启动Wireshark抓包测试：

```
PC1> ping 1.1.1.2

84 bytes from 1.1.1.2 icmp_seq=1 ttl=64 time=0.267 ms
84 bytes from 1.1.1.2 icmp_seq=2 ttl=64 time=0.203 ms
84 bytes from 1.1.1.2 icmp_seq=3 ttl=64 time=0.177 ms
84 bytes from 1.1.1.2 icmp_seq=4 ttl=64 time=0.205 ms
84 bytes from 1.1.1.2 icmp_seq=5 ttl=64 time=0.307 ms
```

8	12.212079	Private_66:68:02	Broadcast	ARP	64 Who has 1.1.1.2? Tell 1.1.1.1
9	12.212413	Private_66:68:04	Private_66:68:02	ARP	64 1.1.1.2 is at 00:50:79:66:68:04
10	12.213183	1.1.1.1	1.1.1.2	ICMP	98 Echo (ping) request id=0x585c, seq=1/256, ttl=64 (reply in 11)
11	12.213395	1.1.1.2	1.1.1.1	ICMP	98 Echo (ping) reply id=0x585c, seq=1/256, ttl=64 (request in 10)
12	13.215080	1.1.1.1	1.1.1.2	ICMP	98 Echo (ping) request id=0x595c, seq=2/512, ttl=64 (reply in 13)
13	13.215214	1.1.1.2	1.1.1.1	ICMP	98 Echo (ping) reply id=0x595c, seq=2/512, ttl=64 (request in 12)
14	14.029398	cc:01:56:85:f0:01	Spanning-tree-(for-…	STP	60 Conf. TC + Root = 32768/0/cc:01:56:85:00:01 Cost = 0 Port = 0x8002
15	14.216731	1.1.1.1	1.1.1.2	ICMP	98 Echo (ping) request id=0x5a5c, seq=3/768, ttl=64 (reply in 16)
16	14.216846	1.1.1.2	1.1.1.1	ICMP	98 Echo (ping) reply id=0x5a5c, seq=3/768, ttl=64 (request in 15)
17	15.218474	1.1.1.1	1.1.1.2	ICMP	98 Echo (ping) request id=0x5b5c, seq=4/1024, ttl=64 (reply in 18)
18	15.218601	1.1.1.2	1.1.1.1	ICMP	98 Echo (ping) reply id=0x5b5c, seq=4/1024, ttl=64 (request in 17)
19	16.108593	cc:01:56:85:f0:01	Spanning-tree-(for-…	STP	60 Conf. TC + Root = 32768/0/cc:01:56:85:00:01 Cost = 0 Port = 0x8002
20	16.220901	1.1.1.1	1.1.1.2	ICMP	98 Echo (ping) request id=0x5c5c, seq=5/1280, ttl=64 (reply in 21)
21	16.221115	1.1.1.2	1.1.1.1	ICMP	98 Echo (ping) reply id=0x5c5c, seq=5/1280, ttl=64 (request in 20)

PC1和PC2能够互相ping通。

2. SW1和SW2连接3台主机组成局域网，在SW1和SW2上建立VLAN2，并将PC1和PC3所连接的接口都配置到VLAN2，将PC1和PC3配置在同一网段。



测试：

```
PC1> ping 1.1.1.3

84 bytes from 1.1.1.3 icmp_seq=1 ttl=64 time=0.435 ms
84 bytes from 1.1.1.3 icmp_seq=2 ttl=64 time=0.237 ms
84 bytes from 1.1.1.3 icmp_seq=3 ttl=64 time=0.188 ms
84 bytes from 1.1.1.3 icmp_seq=4 ttl=64 time=0.188 ms
84 bytes from 1.1.1.3 icmp_seq=5 ttl=64 time=0.428 ms
```

PC1和PC3能够互相ping通。

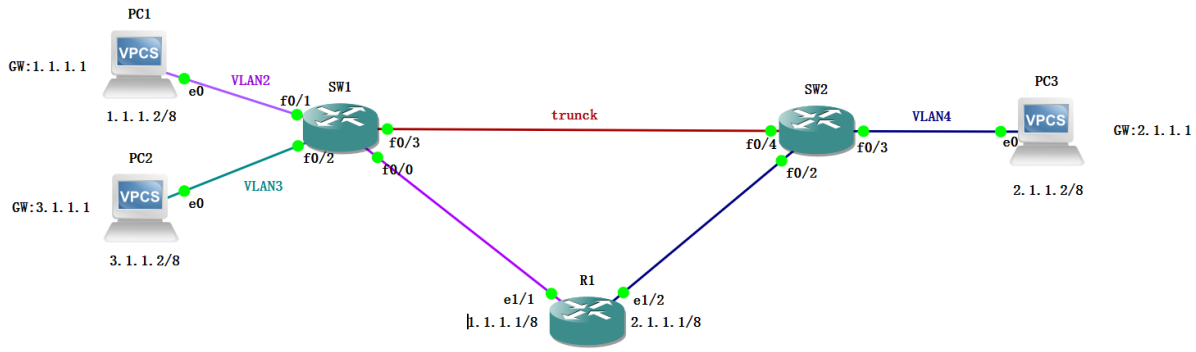
4.3 VLAN 互通

实验拓扑：

交换机 SW1 将主机 PC1 和 PC2 组成局域网，交换机 SW2 将主机 PC3 组成局域网，SW1 和 SW2 之间通过 Trunk 接口相连

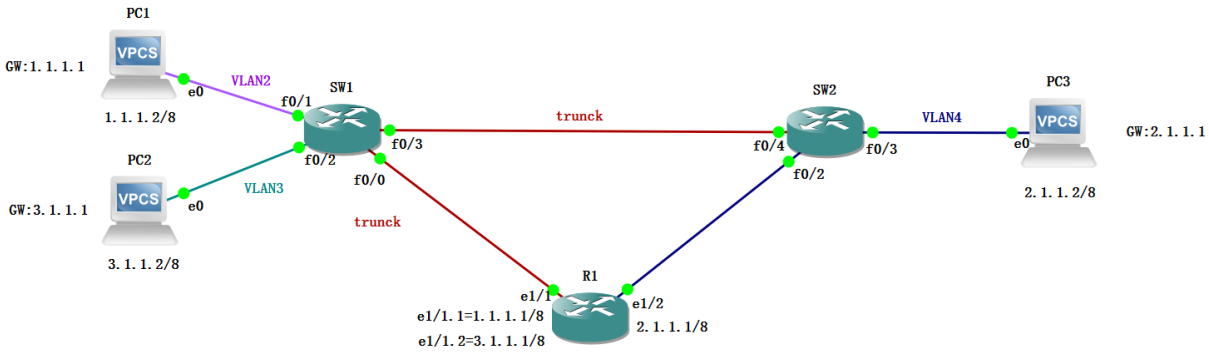
添加路由器 R1，SW1 和 SW2 分别将 R1 接入 VLAN2 和 VLAN4，使得位于不同 VLAN 的 PC1 和 PC3 能够互通

方法 1：用一台路由器，分别接入两个 VLAN



测试：PC1 和 PC3 之间能互相 ping 通，但是和 PC2 之间无法 ping 通

方法 2：单臂路由器：用一条物理线路互联多个 VLAN



测试：

```
PC1> ping 3.1.1.2

84 bytes from 3.1.1.2 icmp_seq=1 ttl=63 time=20.301 ms
84 bytes from 3.1.1.2 icmp_seq=2 ttl=63 time=16.097 ms
84 bytes from 3.1.1.2 icmp_seq=3 ttl=63 time=18.858 ms
84 bytes from 3.1.1.2 icmp_seq=4 ttl=63 time=16.899 ms
84 bytes from 3.1.1.2 icmp_seq=5 ttl=63 time=16.837 ms

PC1> ping 2.1.1.2

2.1.1.2 icmp_seq=1 timeout
84 bytes from 2.1.1.2 icmp_seq=2 ttl=63 time=12.855 ms
84 bytes from 2.1.1.2 icmp_seq=3 ttl=63 time=17.024 ms
84 bytes from 2.1.1.2 icmp_seq=4 ttl=63 time=16.718 ms
84 bytes from 2.1.1.2 icmp_seq=5 ttl=63 time=20.001 ms
```

```
PC2> ping 2.1.1.2

84 bytes from 2.1.1.2 icmp_seq=1 ttl=63 time=31.038 ms
84 bytes from 2.1.1.2 icmp_seq=2 ttl=63 time=11.859 ms
84 bytes from 2.1.1.2 icmp_seq=3 ttl=63 time=12.884 ms
84 bytes from 2.1.1.2 icmp_seq=4 ttl=63 time=12.914 ms
84 bytes from 2.1.1.2 icmp_seq=5 ttl=63 time=17.999 ms
```

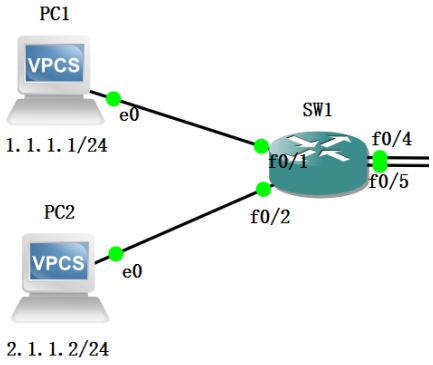
PC1、PC2、PC3 之间能够互通。

5 实验问题思考

5.1 以太网交换机简单组网

思考题一：

PC1和PC2的IP地址如果不配置在一个网段中，是否能够通信？



测试：

```
PC2> ping 1.1.1.1

host (255.255.255.0) not reachable
```

不能直接通信。

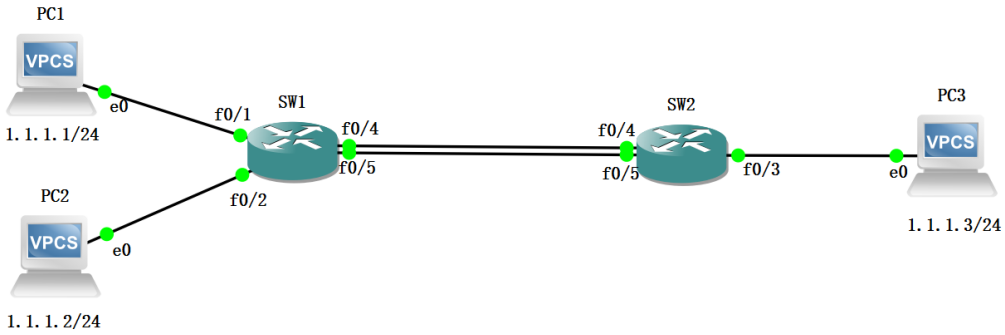
思考题二：

交换机之间用两条线连通，在同一时刻这两条线是否都能够传输数据？

否。默认情况下，交换机之间两条物理链路不能同时用于传输数据。这是因为交换机间的STP会选择一条链路作为活动链路，而把另一条链路置于阻塞状态。这样做是为了避免环路，确保数据帧不会在网络中形成循环并且一直回环。

思考题三：

PC1 和 PC3 如何能够跨越两台交换机通信？



链路聚合将两条物理链路合并为一个逻辑链路，两条链路就可以同时参与数据传输，达到带宽增加的效果。

手动聚合（静态模式）

SW1(config)# interface range f0/4 - 5

SW1(config-if-range)# channel-group 1 mode on

SW1#show ip interface brief

```
Port-channel1              unassigned      YES unset  up                    up
```

SW1(config)#interface port-channel 1

SW1(config-if)#switchport access vlan 2

SW2 同理

测试：

```
PC1> ping 1.1.1.3

84 bytes from 1.1.1.3 icmp_seq=1 ttl=64 time=0.257 ms
84 bytes from 1.1.1.3 icmp_seq=2 ttl=64 time=0.256 ms
84 bytes from 1.1.1.3 icmp_seq=3 ttl=64 time=0.206 ms
84 bytes from 1.1.1.3 icmp_seq=4 ttl=64 time=3.230 ms
84 bytes from 1.1.1.3 icmp_seq=5 ttl=64 time=1.982 ms
```

PC1 和 PC3 之间能互相 ping 通。

5.2 以太网交换机划分 VLAN

思考题一：

在同一个局域网或者同一个VLAN中的主机是否必须配置在同一个网段？在不同局域网或不同VLAN中的主机是否必须配置在不同的网段？

同一个局域网或者同一个VLAN的主机配置在同一个网段中就可以实现在本局域网通信，如果配置在不同的网段，则需要路由器来完成不同网段之间的互通。在不同局域网或者不同VLAN中的主机如果配置在同一网段肯定无法互通，只有配置在不同的网段，并通过路由器互联后才可以互通。

思考题二：

两个交换机之间的双链路备份是否会产生环路？当配置了VLAN后，如何避免产生环路？

两个交换机之间如果建立了两条连接，则其中的一条会被STP生成树协议所禁止，以保证不会出现环路。当配置了VLAN后，需要避免的是同一个VLAN内会产生环路，所以STP生成树就不会简单地禁止某一条链路，而是当两条链路上都允许某个VLAN的数据帧通过时，在一条链路上禁止该VLAN的数据帧通过，这时就会出现两条物理链路同时工作的情况。但是，任意时刻，对于某个VLAN来说，只能从其中的一条上通过，另外一条会被禁止。

思考题三：

两个交换机之间如果需要VLAN能够互通的话，可以采用Trunk方式和Access方式进行互联，它们的区别在哪？

通常在两个交换机之间不建议采用Access方式进行VLAN的互通。因为Access接口只能配置属于一个VLAN，也就是只允许一个VLAN的数据帧通过，因此当两台交换机之间需要实现多个VLAN的互通时，就需要占用多对端口，连接多个Access物理链路，非常浪费资源，也不方便。而Trunk接口可以同时允许多个VLAN的数据帧通过，只需要一条链路就可以实现所有 VLAN之间的互通。

思考题四：

当一个接口配置为Trunk模式并且允许所有VLAN通过时，是否真的允许所有的VLAN数据通过？

允许本交换机上已经创建的所有VLAN的数据帧通过。（本交换机上没有创建的VLAN的数据帧是不能通过的）

思考题五：

在同一个VLAN中，将两台PC配置为1.1.0.1 255.0.0.0和1.1.1.1 255.255.0.0，这两个PC是否能够互通？

是。虽然网络号一个是1.0.0.0，一个是1.1.0.0，但是由于PC在向对方发送数据包时，只知道对方的IP地址，并不知道对方的子网掩码，使用自己的子网掩码计算对方的网络号，而这样计算出来的两个PC的网络号是相同的，所以能够互通。

5.3 VLAN 互通

思考题一：

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：在路由器上配置多网段（逻辑接口）并启用静态路由

适合小型网络，易于部署和维护

方法二：通过三层交换机进行 VLAN 划分和路由

适合需要精细隔离、灵活拓展和高效管理的大型网络

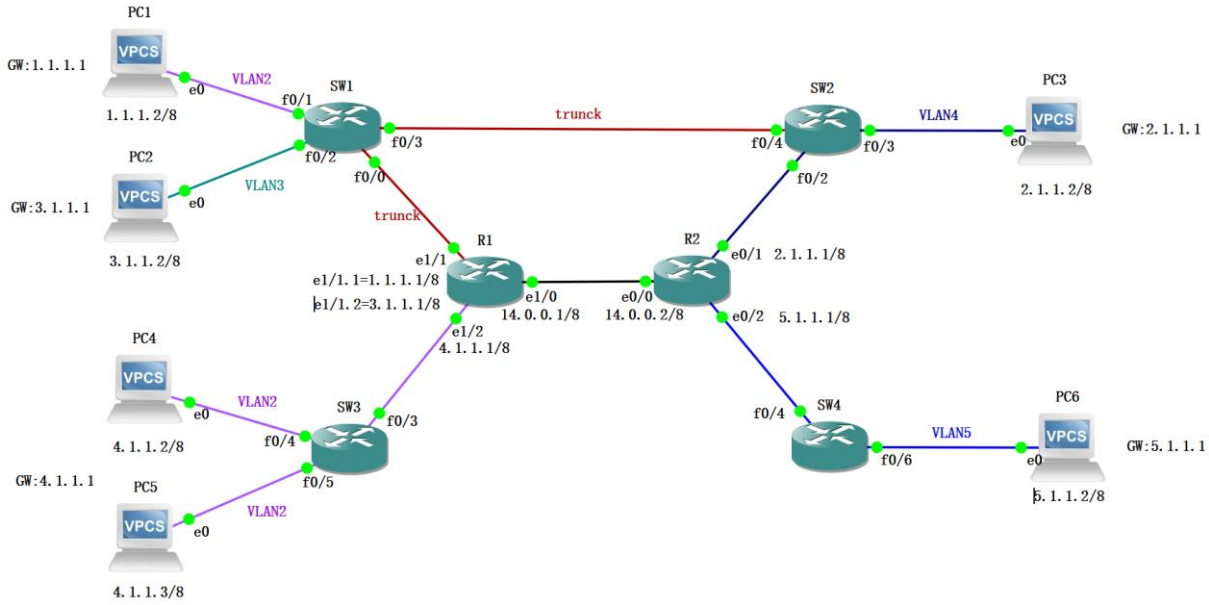
方法三：通过使用具有双网卡的设备，将两个网段连接并启用IP转发功能

适合小型网络或临时需求，实现成本低，配置灵活

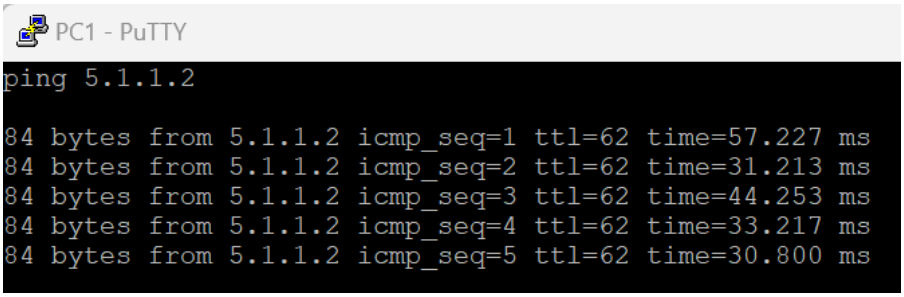
思考题二：

选择两个不同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VLAN标签的变化）

在复杂拓扑下进行实验：



PC1 属于 VLAN 2；PC6 属于 VLAN 5



PC1 ping PC6 流程：

在PC1到SW1的链路上用Wireshark捕获数据包

19	34.796649	cc:03:06:0a:f0:01	CDP/VTP/DTP/PAgP/UD…	CDP	337 Device ID: SW1 Port ID: FastEthernet0/1
20	35.914501	cc:03:06:0a:f0:01	Spanning-tree-(for-…	STP	60 Conf. Root = 32768/0/cc:03:28:b2:00:01 Cost = 0 Port = 0x8002
21	36.079568	Private_66:68:00	Broadcast	ARP	64 Who has 1.1.1.1? Tell 1.1.1.2
22	36.088058	cc:01:21:bd:00:11	Private_66:68:00	ARP	60 1.1.1.1 is at cc:01:21:bd:00:11
23	36.088220	1.1.1.2	5.1.1.2	ICMP	98 Echo (ping) request id=0xfb0f, seq=1/256, ttl=64 (reply in 24)
24	36.145335	5.1.1.2	1.1.1.2	ICMP	98 Echo (ping) reply id=0xfb0f, seq=1/256, ttl=62 (request in 23)
25	37.145941	1.1.1.2	5.1.1.2	ICMP	98 Echo (ping) request id=0xfc0f, seq=2/512, ttl=64 (reply in 26)
26	37.177050	5.1.1.2	1.1.1.2	ICMP	98 Echo (ping) reply id=0xfc0f, seq=2/512, ttl=62 (request in 25)
27	37.987535	cc:03:06:0a:f0:01	Spanning-tree-(for-…	STP	60 Conf. Root = 32768/0/cc:03:28:b2:00:01 Cost = 0 Port = 0x8002
28	38.180733	1.1.1.2	5.1.1.2	ICMP	98 Echo (ping) request id=0xfd0f, seq=3/768, ttl=64 (reply in 29)
29	38.224240	5.1.1.2	1.1.1.2	ICMP	98 Echo (ping) reply id=0xfd0f, seq=3/768, ttl=62 (request in 28)
30	39.224845	1.1.1.2	5.1.1.2	ICMP	98 Echo (ping) request id=0xfe0f, seq=4/1024, ttl=64 (reply in 31)
31	39.257966	5.1.1.2	1.1.1.2	ICMP	98 Echo (ping) reply id=0xfe0f, seq=4/1024, ttl=62 (request in 30)
32	40.081306	cc:03:06:0a:f0:01	Spanning-tree-(for-…	STP	60 Conf. Root = 32768/0/cc:03:28:b2:00:01 Cost = 0 Port = 0x8002
33	40.258583	1.1.1.2	5.1.1.2	ICMP	98 Echo (ping) request id=0xff0f, seq=5/1280, ttl=64 (reply in 34)
34	40.289278	5.1.1.2	1.1.1.2	ICMP	98 Echo (ping) reply id=0xff0f, seq=5/1280, ttl=62 (request in 33)

PC1和SW1之间的接口配置为 Access 模式，数据包从PC1发出时不会有VLAN标签。交换机会将数据包根据配置的VLAN进行转发，但不会在帧中封装VLAN标签。

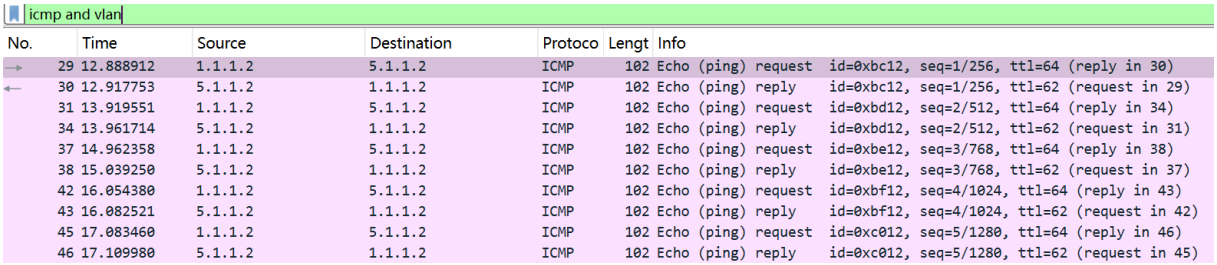
1. PC1 发起 Ping 请求

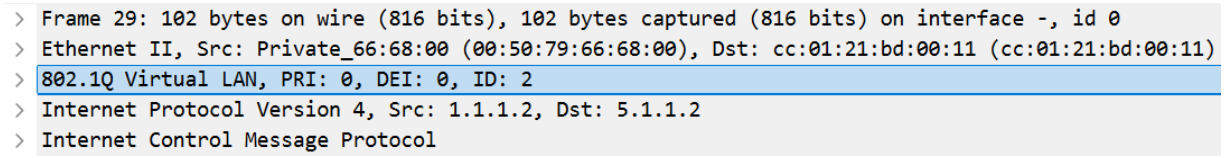
   PC1发送一个 ICMP Echo Request 数据包，目标IP为5.1.1.2（PC6 的地址）

2. SW1 处理数据包

   SW1 接收到该数据包时，由于 PC1 属于 VLAN 2，SW1 会将该数据包的以太网帧头中加入 VLAN 2 的标签（使用 802.1Q 封装）

在SW1到R1的 Trunk 链路上用Wireshark捕获数据包，filter: icmp and vlan





   SW1 会根据数据包的目的MAC地址查找转发表，并将数据包从与 R1 连接的 trunk 端口发送出去

3. Trunk Link

   通过 trunk 链路，数据包会携带 VLAN 标签（VLAN 2）进入 R1。此时，R1 通过 VLAN 标签判断该数据包属于 VLAN 2，并将数据包送到正确的接口进行处理

4. R1 处理数据包

   R1 看到数据包来自 VLAN 2，目标是 PC6 所在的 VLAN 5，因此，R1 需要进行 VLAN 间路由

R1 的接口会解封 VLAN 标签，恢复原始的 IP 数据包，因为 R1 在处理 Layer 3 的路由，而不是继续在 Layer 2 中传输；然后 R1 将该数据包根据目标 IP 地址（5.1.1.2）进行路由

5. R1 转发数据包到 R2

   R1 和 R2 之间是直接连接的，数据包会被发送到 R1 上配置的下一跳接口，并通过 R1 和 R2 之间的链路进行传输

6. R2 处理数据包

   R2 接收到从 R1 发来的数据包后，检查目标 IP 地址（5.1.1.2），确认该地址属于 VLAN 5。根据路由表，R2 会决定将数据包转发到 SW4（VLAN 5的交换机）

   R2 与 SW4 之间的通信是在 VLAN 5 中进行的

在SW1到R1的 Trunk 链路上用Wireshark捕获数据包，filter: icmp

```
> Frame 10: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface -, id 0
v Ethernet II, Src: cc:02:21:db:00:02 (cc:02:21:db:00:02), Dst: Private_66:68:05 (00:50:79:66:68:05)
    > Destination: Private_66:68:05 (00:50:79:66:68:05)
    > Source: cc:02:21:db:00:02 (cc:02:21:db:00:02)
      Type: IPv4 (0x0800)
      [Stream index: 2]
> Internet Protocol Version 4, Src: 1.1.1.2, Dst: 5.1.1.2
> Internet Control Message Protocol
```

   R2 并不需要将数据包加上 VLAN 标签

7. SW4 处理数据包

   SW4 接收到数据包后，查看数据包的目标 MAC 地址，确定数据包应该被转发到 PC6 所连接的端口

8. PC6 接收数据包

   PC6 接收到 ICMP Echo Request 数据包后，会生成一个 ICMP Echo Reply 响应，并将其发送回 PC1（反向过程）

思考题三：

阐述VLAN、物理网络及IP网段的关系。

VLAN 是一种逻辑上的网络划分方式，允许在同一物理网络中创建多个虚拟网络。VLAN是通过网络设备（如交换机）进行配置的，它根据不同的需求将一个物理网络划分为多个逻辑子网。

物理网络 是指实际存在的硬件网络设施，包括交换机、路由器、计算机、服务器等设备，以及连接这些设备的物理线路（如以太网电缆、光纤等）。

IP网段 是网络层的一个概念，它基于IP地址的结构对网络进行逻辑划分和管理。一个IP网段表示一组具有相同网络地址的IP地址，设备在同一个网段内时，可以直接进行通信，而不需要经过路由器。

VLAN是在物理网络的基础上进行逻辑划分的。虽然物理交换机提供设备间的硬件连接，VLAN则通过逻辑分组使设备之间的通信仅限于相同VLAN内。VLAN在数据链路层，IP网段在网络层，每个VLAN通常会对应一个或多个IP网段。通过配置不同VLAN的IP网段，可以使得同一物理交换机上的设备进行逻辑隔离，并且跨VLAN的设备需要通过三层设备（如路由器或三层交换机）进行路由操作。在路由过程中，数据包通过三层设备从一个VLAN的IP网段转发到另一个VLAN的IP网段。

总结：物理网络提供了通信的硬件基础，VLAN利用这一基础实现数据链路层的逻辑隔离，IP网段则在网络层通过地址分配和路由规则管理设备间的逻辑通信。

6 实验总结

本次实验让我学会了如何配置 VLAN。因为按顺序做实验写的思考题，在查找资料解决 4.1 思考题三的实验和 4.2 的内容有些重复了，我在思考题里所做的聚合配置在实验环境下其实没有必要。但在实际网络中，LACP 在高流量、高可靠性需求场景下是非常有用的，能够提高网络效率。

实验中我特别关注了 Access 和 Trunk 端口的配置。Access 端口通常用于连接终端设备，配置直接。Trunk 端口为连接不同交换机的链路设计，通过标记机制，将多个 VLAN 的流量传递到另一台交换机。通过 VLAN 标记，Trunk 端口能够识别和区分来自不同 VLAN 的流量。实验中实现了多个 VLAN 的跨交换机通信，让我深入理解了 Trunk 端口在中大型网络中的重要性。Trunk 端口不仅可以减少物理链路的数量，还能提高网络的

可扩展性和管理效率。此外，本次实验中涉及的复杂拓扑配置也让我熟练了排除网络故障的能力。