实验四：交换机和VLAN实验

计算机网络技术实践

实验报告

实验名称 交换机和VLAN实验

姓　名_____

实 验 日 期： 12.02

学　号 20222113

实验报告日期： 12.22

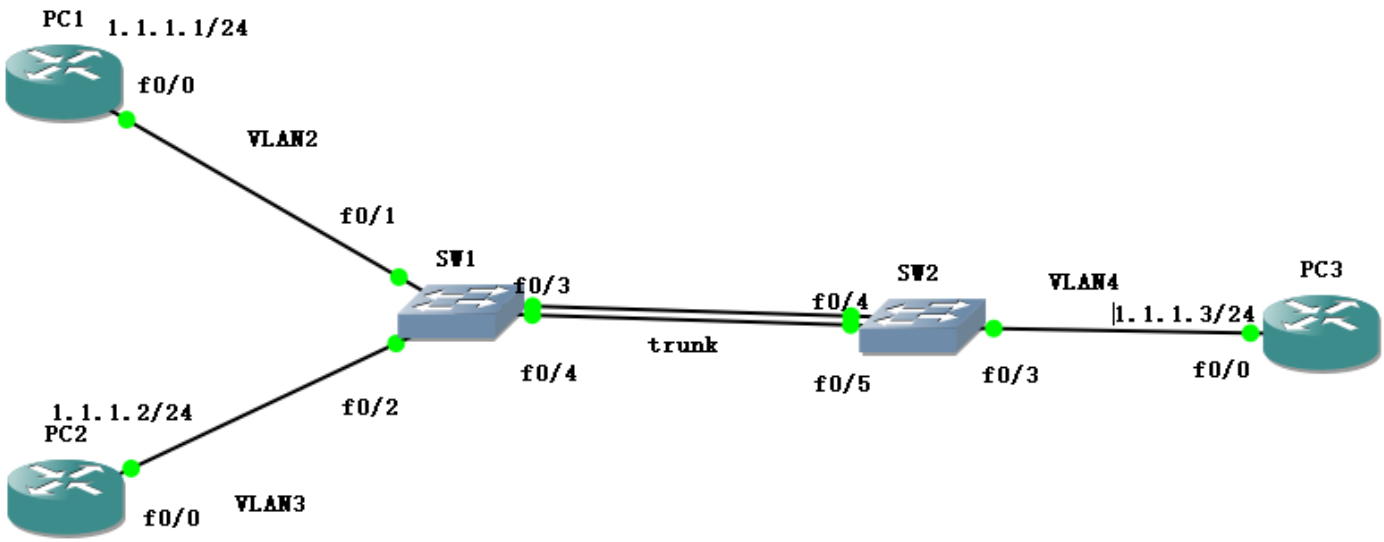
报 告 退 发:

一、实验环境

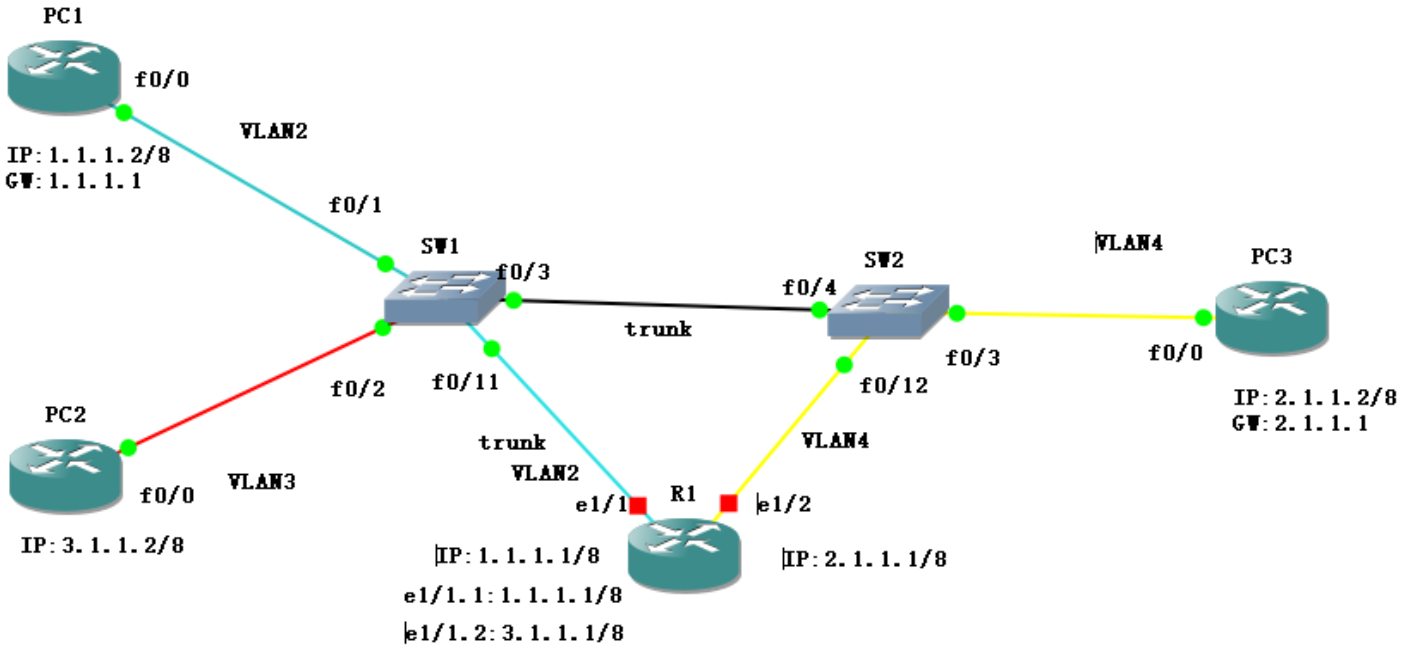
1. 前端GNS3负责图形拓扑设计
2. 计算资源为 Windows 11.0 与 VMWare Ubuntu 20.04.6，后端Dynamips仿真环境负责硬件模拟，cisco 2610 模拟PC，cisco 3640 模拟交换机SW
3. 抓包采用Wireshark 4.4.2

二、实验拓扑

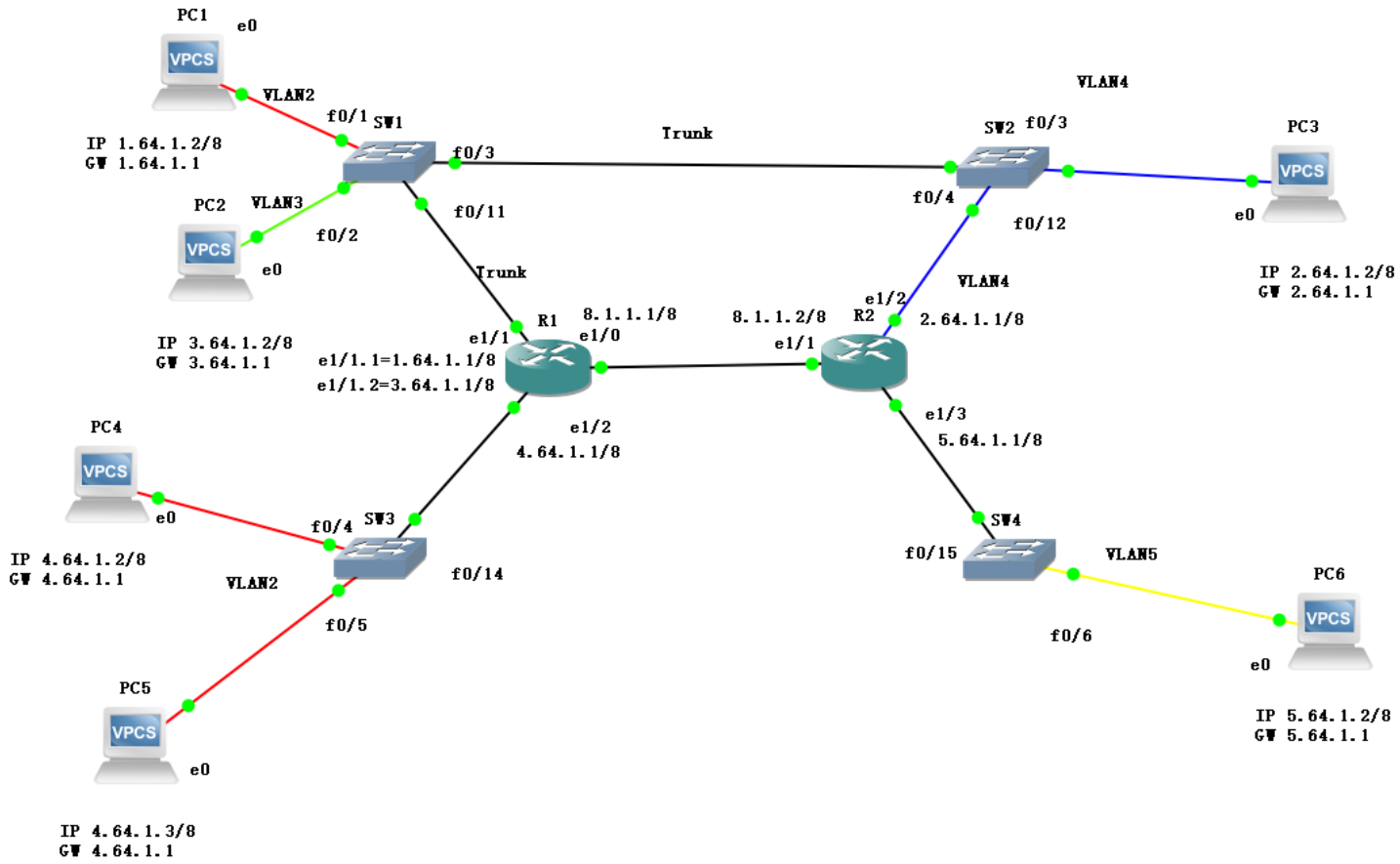
拓扑1：



拓扑2：



拓扑3：



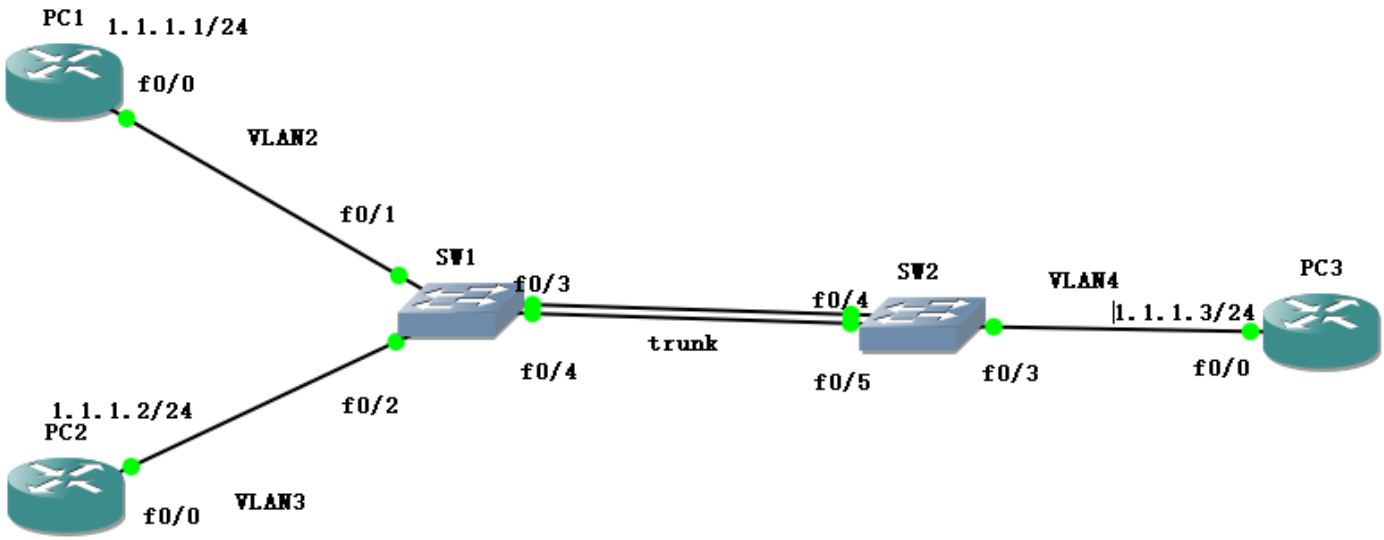
三、实验目的

- 熟练掌握以太网交换机的使用方法，能够在模拟环境中配置和使用以太网交换机来组建局域网。
- 对以太网交换机的VLAN划分与配置有深入了解，能够在仿真环境中实现VLAN的创建和管理，灵活配置多个虚拟局域网以满足不同需求。

- 通过路由器实现不同VLAN之间的互通，能够配置单臂路由，利用交换机的Trunk接口实现多个VLAN之间的数据通信和路由功能。

四、实验内容与分析

网络拓扑1实验



1.完成局域网内的主机互通

使用如下配置指令配置PC1

```
PC1#configure terminal
PC1(config)#interface f0/0
PC1(config-if)#ip address 1.1.1.1 255.255.255.0
PC1(config-if)#no shutdown
PC1#write
```

效果如下

```
PC1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
PC1(config)#int f0/0
PC1(config-if)#ip addr 1.1.1.1 255.255.255.0
PC1(config-if)#no shut
PC1(config-if)#
PC1#
```

配置之后三台PC可以互通，这里用PC1pingPC3，如下

```
PC1#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 60 percent (3/5), round-trip min/avg/max = 20/20/20 ms
```

2.完成不同网段的主机互通

为了实现不同网段的构造，我们把PC3的IP变更为1.1.2.3/24，转而在PC3上pingPC1

我们对PC3的配置和ping命令如下

```
PC3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
PC3(config)#int f0/0
PC3(config-if)#ip addr 1.1.2.3 255.255.255.0
PC3(config-if)#no shut
PC3(config-if)#exit
PC3(config)#exit
PC3#write
Building configuration...
[OK]
PC3#
*Mar  1 00:17:48.895: %SYS-5-CONFIG_I: Configured from console by console
PC3#ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

可知不同网段的PC无法互相通信

3.完成交换机间双线同传

环路是交换网络中常见的网络问题，它通常发生在多个路径连接同一对交换机时，可能导致广播风暴、MAC地址表混乱以及网络拥塞。生成树协议（STP, Spanning Tree Protocol）可以避免环路的发生，确保网络的稳定运行；除此以外链路聚合可以分配流量实现双线同传，同样也可以避免环路发生。环路会引发诸如广播风暴等问题，**STP和链路聚合**，通过以下方式有效地防止这些问题：

1. **根桥选举**：在STP协议中，所有交换机通过选举机制选出一个根桥，其余交换机根据根桥的位置来构建生成树，确保网络拓扑没有环路。
2. **端口角色分配**：
   - **根端口（Root Port, RP）**：每个非根桥交换机上到根桥的路径成本最低的端口，负责转发数据包。
   - **指定端口（Designated Port, DP）**：在每个网络段上，具有最低路径成本的端口用于转发流量。
   - **阻塞端口（Blocking Port, BP）**：为了防止环路，某些端口会被置于阻塞状态，不进行数据转发，只负责监听STP的BPDU（桥协议数据单元）。

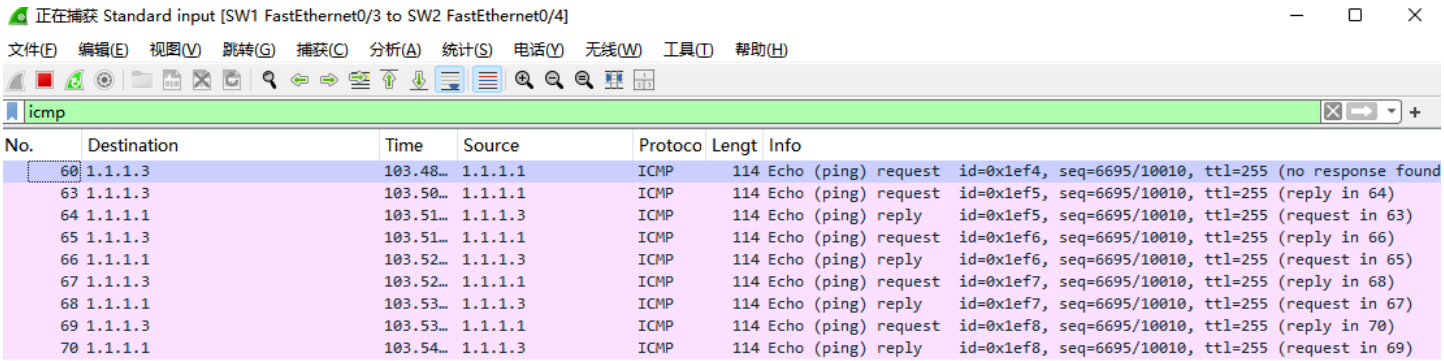
3. **STP的工作机制：**
    - 默认情况下，STP会在交换机之间的多条链路中选择一条作为活动链路，其他链路则被阻塞以避免环路。
    - 当活动链路发生故障时，STP会自动激活阻塞链路，确保网络的冗余和高可用性。

在实际网络中，例如在SW1与SW2之间有两条物理链路时，STP会确保在任何时刻，只有一条链路用于数据传输，另一条链路会被阻塞，直到出现链路故障或网络拓扑变化时，STP会重新计算并切换链路。

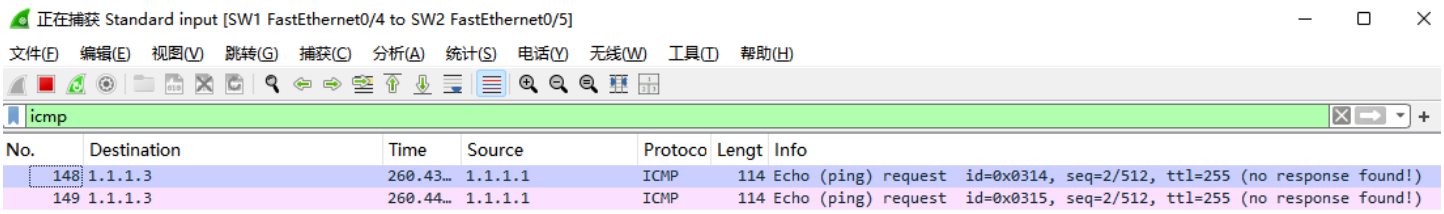
通过Wireshark抓包进行验证时，假设PC1和PC3之间进行ping测试，两条链路的数据传输情况可以通过抓包分析。

在正常情况下，只有一条链路会承载数据流量，另一条链路将处于阻塞状态。这种机制有效避免了环路的发生，保证了网络的稳定性和数据的可靠传输。

SW1-f0/3----SW2-f0/4



SW1-f0/4----SW2-f0/5



这里我们可以分析得到，SW1被选择为根桥，其端口处于转发状态，SW2不是根桥，所以f0/5端口被阻塞

我们接着使用show命令查看STP生成树的详细信息来辅佐验证

SW1-ALL都属于转发状态:

```
SW1#show spanning-tree

 VLAN1 is executing the ieee compatible Spanning Tree protocol
  Bridge Identifier has priority 32768, address cc04.0b4e.0000
  Configured hello time 2, max age 20, forward delay 15
  We are the root of the spanning tree
  Topology change flag not set, detected flag not set
  Number of topology changes 1 last change occurred 00:33:34 ago
          from FastEthernet0/1
  Times:  hold 1, topology change 35, notification 2
          hello 2, max age 20, forward delay 15
  Timers: hello 0, topology change 0, notification 0, aging 300

 Port 2 (FastEthernet0/1) of VLAN1 is forwarding
   Port path cost 19, Port priority 128, Port Identifier 128.2.
   Designated root has priority 32768, address cc04.0b4e.0000
   Designated bridge has priority 32768, address cc04.0b4e.0000
   Designated port id is 128.2, designated path cost 0
   Timers: message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   BPDU: sent 1021, received 0

 Port 3 (FastEthernet0/2) of VLAN1 is forwarding
   Port path cost 19, Port priority 128, Port Identifier 128.3.
   Designated root has priority 32768, address cc04.0b4e.0000
   Designated bridge has priority 32768, address cc04.0b4e.0000
   Designated port id is 128.3, designated path cost 0
   Timers: message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   BPDU: sent 1021, received 0

 Port 4 (FastEthernet0/3) of VLAN1 is forwarding
   Port path cost 19, Port priority 128, Port Identifier 128.4.
   Designated root has priority 32768, address cc04.0b4e.0000
   Designated bridge has priority 32768, address cc04.0b4e.0000
   Designated port id is 128.4, designated path cost 0
   Timers: message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   BPDU: sent 1026, received 2

 Port 5 (FastEthernet0/4) of VLAN1 is forwarding
   Port path cost 19, Port priority 128, Port Identifier 128.5.
   Designated root has priority 32768, address cc04.0b4e.0000
   Designated bridge has priority 32768, address cc04.0b4e.0000
   Designated port id is 128.5, designated path cost 0
   Timers: message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   BPDU: sent 1025, received 1
```

SW2-f0/5被阻塞：

```
SW2#show spanning-tree

VLAN1 is executing the ieee compatible Spanning Tree protocol
 Bridge Identifier has priority 32768, address cc05.0b6e.0000
 Configured hello time 2, max age 20, forward delay 15
 Current root has priority 32768, address cc04.0b4e.0000
 Root port is 5 (FastEthernet0/4), cost of root path is 19
 Topology change flag not set, detected flag not set
 Number of topology changes 1 last change occurred 00:33:54 ago
         from FastEthernet0/3
 Times:  hold 1, topology change 35, notification 2
         hello 2, max age 20, forward delay 15
 Timers: hello 0, topology change 0, notification 0, aging 300

Port 4 (FastEthernet0/3) of VLAN1 is forwarding
   Port path cost 19, Port priority 128, Port Identifier 128.4.
   Designated root has priority 32768, address cc04.0b4e.0000
   Designated bridge has priority 32768, address cc05.0b6e.0000
   Designated port id is 128.4, designated path cost 19
   Timers: message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   BPDU: sent 1047, received 0

Port 5 (FastEthernet0/4) of VLAN1 is forwarding
   Port path cost 19, Port priority 128, Port Identifier 128.5.
   Designated root has priority 32768, address cc04.0b4e.0000
   Designated bridge has priority 32768, address cc04.0b4e.0000
   Designated port id is 128.4, designated path cost 0
   Timers: message age 4, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   BPDU: sent 2, received 1046

Port 6 (FastEthernet0/5) of VLAN1 is blocking
   Port path cost 19, Port priority 128, Port Identifier 128.6.
   Designated root has priority 32768, address cc04.0b4e.0000
   Designated bridge has priority 32768, address cc04.0b4e.0000
   Designated port id is 128.5, designated path cost 0
   Timers: message age 1, forward delay 0, hold 0
   Number of transitions to forwarding state: 0
   BPDU: sent 1, received 1052
```

4. **链路聚合的工作机制：**
   - 如果我们启用链路聚合，交换机会自动分配合适比例的流量在连接的两条线路上。

```
SW1(config)#interface range f0/3 - 4
SW1(config-if-range)#channel-group 1 mode on
SW1(config)#interface Port-channel1
SW1(config-if)#switchport mode trunk

SW2(config)#interface range f0/4 - 5
SW2(config-if-range)#channel-group 1 mode on
SW2(config)#interface Port-channel1
SW2(config-if)#switchport mode trunk
```

再执行PC1pingPC3，发现流量被分配到了两条线路上

SW1-f0/3----SW2-f0/4

```
526 1.1.1.3    762.09… 1.1.1.1    ICMP    114 Echo (ping) request  id=0x03ed, seq=4232/34832, ttl=255 (no response fou
531 1.1.1.3    763.95… 1.1.1.1    ICMP    114 Echo (ping) request  id=0x03ee, seq=4232/34832, ttl=255 (reply in 532)
532 1.1.1.1    763.96… 1.1.1.3    ICMP    114 Echo (ping) reply    id=0x03ee, seq=4232/34832, ttl=255 (request in 531)
533 1.1.1.3    763.97… 1.1.1.1    ICMP    114 Echo (ping) request  id=0x03ef, seq=4232/34832, ttl=255 (no response fou
534 1.1.1.1    763.98… 1.1.1.3    ICMP    114 Echo (ping) reply    id=0x03f0, seq=4232/34832, ttl=255
535 1.1.1.3    763.99… 1.1.1.1    ICMP    114 Echo (ping) request  id=0x03f1, seq=4232/34832, ttl=255 (no response fou
```

SW1-f0/4----SW2-f0/5

```
417 1.1.1.1    762.00… 1.1.1.3    ICMP    114 Echo (ping) reply    id=0x03ef, seq=4232/34832, ttl=255
418 1.1.1.3    762.01… 1.1.1.1    ICMP    114 Echo (ping) request  id=0x03f0, seq=4232/34832, ttl=255 (no response found
419 1.1.1.1    762.02… 1.1.1.3    ICMP    114 Echo (ping) reply    id=0x03f1, seq=4232/34832, ttl=255
```

之后我们输入如下指令关闭链路聚合

```
SW1(config)#interface range f0/3 - 4
SW1(config-if-range)#no channel-group 1

SW2(config)#interface range f0/4 - 5
SW2(config-if-range)#no channel-group 1
```

4.完成跨越两台交换机的主机互通

我们查看一下SW1的MAC地址表，由于没有直连PC3，并没有PC3的MAC地址，但我们仍然可以直接ping通PC1和PC3，接下来我们解析一下原因

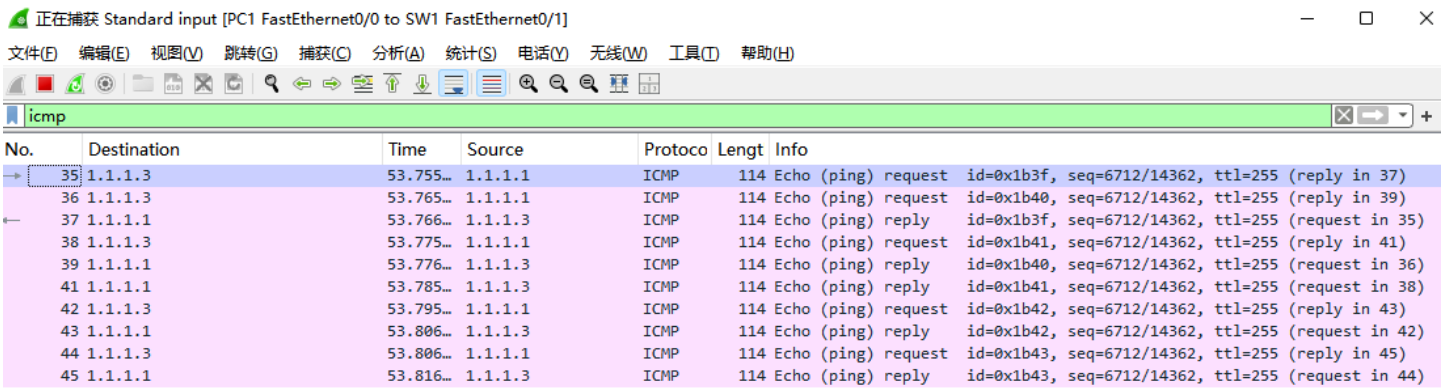
```
SW1#show mac
Destination Address  Address Type  VLAN  Destination Port
-------------------  ------------  ----  --------------------
cc04.0b4e.0000          Self          1     Vlan1
c801.0aa5.0000          Dynamic       1     FastEthernet0/1
c802.0ac7.0000          Dynamic       1     FastEthernet0/2
```

按照上文我们对STP协议的了解，我们猜测会有这样的过程：

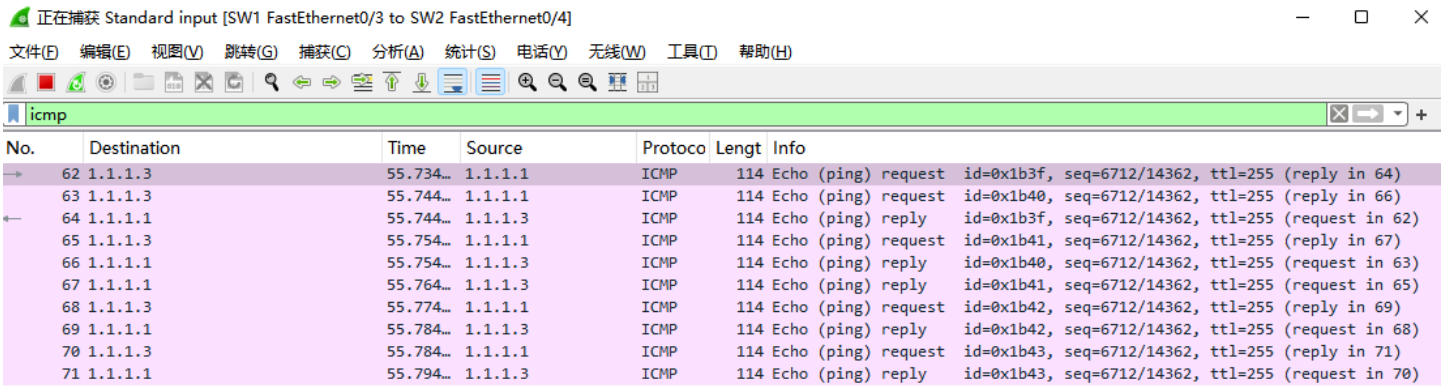
- SW1作为根桥，在遇到未知MAC的帧，从所有处于转发态的接口转发报文，接下来，PC1的报文会从f0/4接口转发，SW2收到之后，因为SW2已知PC3的MAC地址，最终SW2会把数据包发送到PC3

我们接下来抓包验证一下

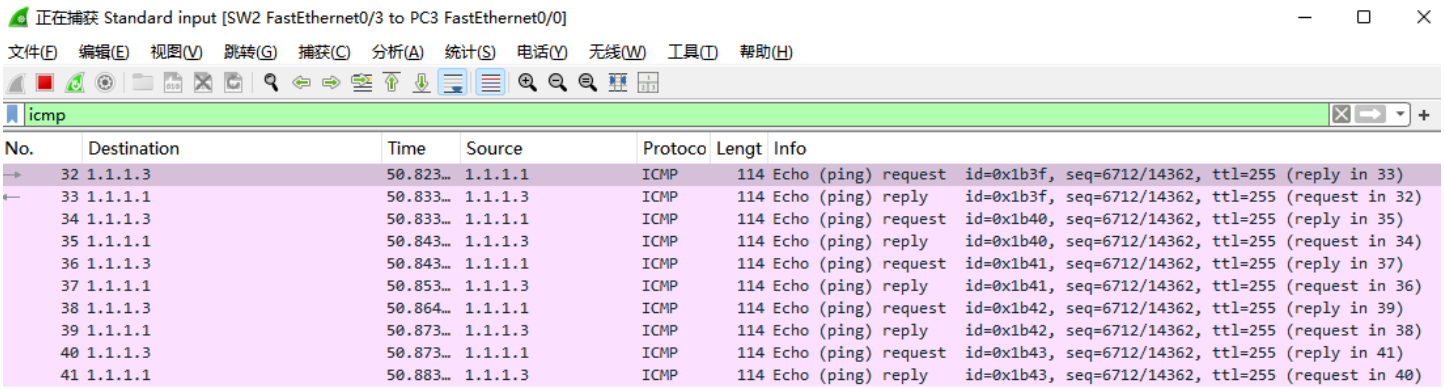
PC1-SW1



SW1-SW2



SW2-PC3



完成一次传输后，SW1会更新MAC表，下次从MAC表中直接查询转发端口，不再泛洪转发

```
SW1#show mac
Destination Address  Address Type  VLAN  Destination Port
-------------------  ------------  ----  --------------------
cc04.0b4e.0000          Self          1     Vlan1
c801.0aa5.0000          Dynamic       1     FastEthernet0/1
c802.0ac7.0000          Dynamic       1     FastEthernet0/2
c803.0b08.0000          Dynamic       1     FastEthernet0/3
```

可以看到SW1已经更新了PC的地址，通过接口f0/3进行转发

5.实现交换机划分VLAN

利用以下指令，在SW1上将端口f0/1配成VLAN 2，将端口f0/2配成VLAN 3

```
SW1#vlan database
SW1(vlan)#vlan 2
SW1(vlan)#vlan 3

SW1#configure terminal
SW1(config)#interface f0/1
SW1(config-if)#switchport access vlan 2
SW1(config)#interface f0/2
SW1(config-if)#switchport access vlan 3
```

此时PC1与PC2不在一个VLAN内，通信被阻断

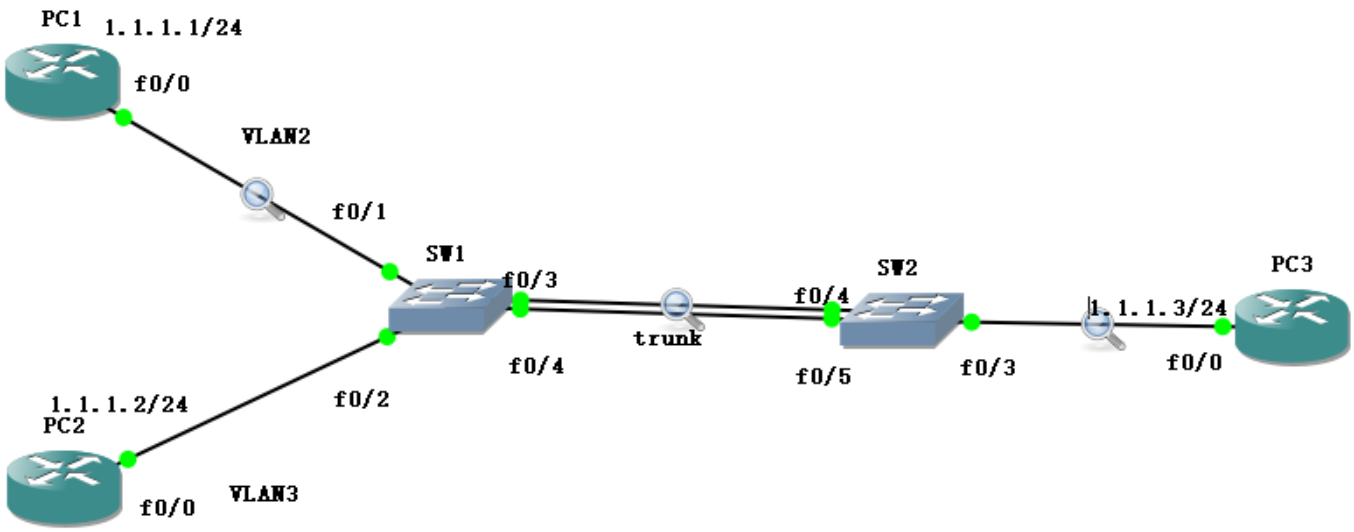
```
PC1#ping 1.1.1.2

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)
```

我们接着把SW1端口f0/3、f0/4和SW2端口f0/4、f0/5配置成trunk模式

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

实际拓扑如下



意味着SW1-SW2链路允许所有的数据帧通过，也就是说PC1与PC3仍然可以正常通信

```
PC1#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 100 percent (5/5), round-trip min/avg/max = 20/156/676 ms
```

验证如上

6.分析ACCESS与TRUNK模式的异同

3.6.1 ACCESS模式

ACCESS模式通常用于连接终端设备（如PC）。在这种模式下，每个端口只能属于一个特定的VLAN，并且该端口上传输的所有数据帧都被视为该VLAN的一部分。

以下是ACCESS模式的几个关键特点：

1. **单一VLAN**：每个ACCESS端口只能关联一个VLAN，意味着所有通过该端口的流量都会被标记为该VLAN的流量。
2. **无VLAN标签**：在ACCESS模式下，数据帧在交换机端口之间传输时不会携带VLAN标签，因此设备之间的通信是透明的，对VLAN的划分不可见。
3. **用途**：该模式主要用于连接不支持VLAN标记的终端设备，例如普通PC等，这些设备无法识别VLAN标签，因此必须通过ACCESS端口与交换机连接。

在本拓扑中，连接交换机与PC之间的链路均配置为ACCESS模式。在PC1与PC3进行ping测试时，可以观察到，数据包在从PC1到SW1、再到SW2，最终到PC3的传输过程中，不会带有VLAN标签。这表明，整个通信过程中，流量始终作为默认VLAN的成员进行处理，且未涉及VLAN标签的识别或传输

PC1-SW1

```
> Frame 666: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface -, id 0
> Ethernet II, Src: c8:01:0a:a5:00:00 (c8:01:0a:a5:00:00), Dst: c8:03:0b:08:00:00 (c8:03:0b:08:00
> Internet Protocol Version 4, Src: 1.1.1.1, Dst: 1.1.1.3
> Internet Control Message Protocol
```

SW2-PC3

```
> Frame 536: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface -, id 0
> Ethernet II, Src: c8:03:0b:08:00:00 (c8:03:0b:08:00:00), Dst: c8:01:0a:a5:00:00 (c8:01:0a:a5:00
> Internet Protocol Version 4, Src: 1.1.1.3, Dst: 1.1.1.1
> Internet Control Message Protocol
```

3.6.2 TRUNK模式

TRUNK模式用于交换机之间，或交换机与路由器之间的链路，允许多个VLAN的流量通过同一物理链路传输。与ACCESS模式不同，TRUNK模式可以承载来自多个VLAN的数据流量，并且在数据传输过程中，每个数据帧都会带有VLAN标签。

以下是TRUNK模式的主要特点：

1. **多VLAN支持**：一个TRUNK端口可以同时传输来自多个VLAN的流量，这使得网络中的不同VLAN能够共享一条物理链路，极大提高了链路的利用率。

2. **VLAN标签**：在TRUNK链路上传输的数据帧会带上VLAN标签，如下图，使用802.1Q标准进行封装。这个标签指示了数据帧所属的VLAN，使得网络中的交换机能够根据标签正确地转发数据。
3. **用途**：TRUNK模式通常用于交换机之间的连接，或者连接支持VLAN的路由器接口。它使得多个VLAN的流量能够在同一链路上传输，特别适用于需要跨多个VLAN进行通信的网络环境。

在本拓扑中，SW1与SW2之间的连接配置为TRUNK模式。在PC1向PC3发送ping请求时，经过SW1和SW2之间的链路时，数据包会携带VLAN标签。这是因为TRUNK链路需要识别多个VLAN的流量，因此每个数据帧上都会附加上相应的VLAN标识

SW1-SW2

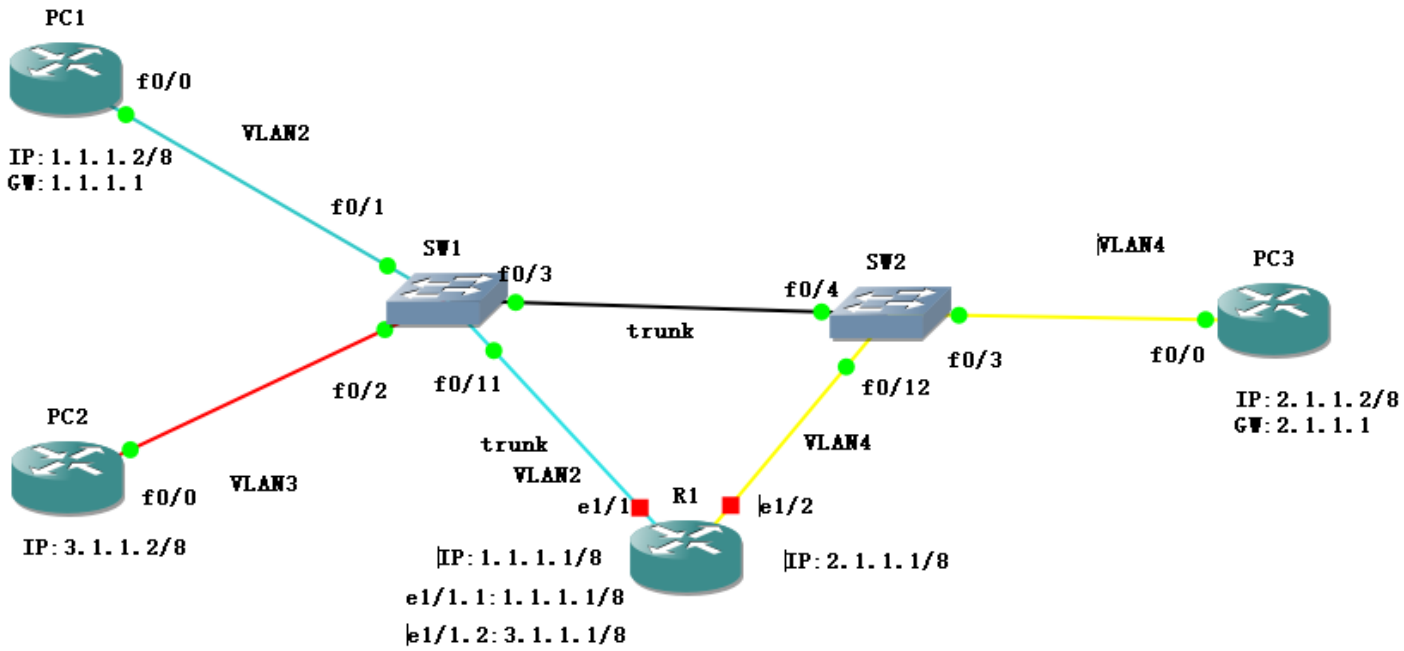
```
> Frame 1570: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface -, id 0
> Ethernet II, Src: c8:01:0a:a5:00:00 (c8:01:0a:a5:00:00), Dst: c8:03:0b:08:00:00 (c8:03:0b:08:00
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 2
> Internet Protocol Version 4, Src: 1.1.1.1, Dst: 1.1.1.3
> Internet Control Message Protocol
```

3.6.3 TRUNK模式

特性	ACCESS模式	TRUNK模式
用途	连接终端设备（如PC、打印机）到交换机	用于交换机之间或交换机与路由器之间的链路
VLAN划分	每个端口只能属于一个VLAN	一个端口可以承载多个VLAN的流量
帧标记	数据帧不包含VLAN标签	数据帧包含VLAN标签
链路冗余支持	仅传输单一VLAN的流量，不支持带宽聚合，通常配置为单链路	可同时传输多个VLAN的流量，支持链路聚合协议，可以提供更高的带宽和冗余性
流量类型	处理单一VLAN的流量，用于Layer 2交换	处理多个VLAN的流量，支持Layer 2的VLAN隔离和Layer 3的路由
广播域	每个ACCESS端口对应一个广播域	一个TRUNK端口可以跨多个广播域传输流量
安全性	限制单一VLAN，减少未授权访问的VLAN间风险	支持多个VLAN，需要适当的VLAN配置和访问控制来确保安全
封装协议	不涉及协议封装	使用802.1Q或ISL协议封装数据帧
MTU	数据帧大小无额外开销	因封装VLAN标签，数据帧增加4字节开销，可能受MTU限制影响
管理复杂度	简单配置，适合终端接入场景	配置相对复杂，需要管理允许通过的VLAN列表和封装方式

1. ACCESS模式主要用于连接终端设备，每个端口只能属于一个VLAN，数据帧不带标签，配置简单但仅支持单一VLAN流量；
2. TRUNK模式用于交换机之间或交换机与路由器之间的连接，可承载多个VLAN流量，数据帧带有802.1Q标签，支持链路聚合，适合复杂网络环境。两者结合使用，实现终端接入与VLAN间通信。

网络拓扑2实验



1.使用同一路由器接入两个VLAN

我们按照上图拓扑图注配置VLAN，执行PC1pingPC3结果如下

```
PC1#ping 2.1.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/448/1916 ms
```

SW的ACCESS方法不会向数据包中添加tag，可以在抓包时看到没有VLAN tag

PC1-SW1-R1是因为PC1的包来自SW1的VLAN2端口，R1与SW1的VLAN2端口连接，这条链路可行

R1-SW2-PC3是因为R1的包来自SW2的VLAN4端口，PC3与SW2的VLAN4端口连接，这条链路也可行

以下是抓包证明

PC1-SW1

```
> Frame 45: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface -,
> Ethernet II, Src: c8:01:73:ec:00:00 (c8:01:73:ec:00:00), Dst: cc:06:6a:9c:00:11 (cc:06
> Internet Protocol Version 4, Src: 1.1.1.2, Dst: 2.1.1.2
> Internet Control Message Protocol
```

SW1-R1

```
> Frame 41: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface -,
> Ethernet II, Src: c8:01:73:ec:00:00 (c8:01:73:ec:00:00), Dst: cc:06:6a:9c:00:11 (cc:06
> Internet Protocol Version 4, Src: 1.1.1.2, Dst: 2.1.1.2
> Internet Control Message Protocol
```

R1-SW2

```
> Frame 35: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface -,
> Ethernet II, Src: cc:06:6a:9c:00:12 (cc:06:6a:9c:00:12), Dst: c8:03:7a:18:00:00 (c8:03
> Internet Protocol Version 4, Src: 1.1.1.2, Dst: 2.1.1.2
> Internet Control Message Protocol
```

SW2-PC3

```
> Frame 26: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface -,
> Ethernet II, Src: cc:06:6a:9c:00:12 (cc:06:6a:9c:00:12), Dst: c8:03:7a:18:00:00 (c8:03
> Internet Protocol Version 4, Src: 1.1.1.2, Dst: 2.1.1.2
> Internet Control Message Protocol
```

但是PC1和PC2不能通信

```
PC1#ping 3.1.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.1.1.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

2.使用单臂路由器实现VLAN互通

修改SW1-R1之间的链路为trunk模式，使得所有VLAN tag都可以通行，指令如下

```
1 SW1(config-if)#switchport mode trunk
2 SW1(config-if)#switchport trunk allowed vlan all
```

为了实现路由器与交换机之间的多 VLAN 通信，我们可以在路由器的物理接口上配置子接口。以 R1 路由器和 SW1 交换机为例，以下是具体实现方式：

1. 将路由器 R1 的物理接口 `e1/1` 划分为两个子接口 `e1/1.1` 和 `e1/1.2`，并分别为它们配置 802.1Q 封装协议的 VLAN 标签（dot1q 2 和 dot1q 3）。
2. 配置子接口 `e1/1.1` 的封装协议为 VLAN 2。这意味着该子接口能够接收带有 VLAN 2 标签的数据帧，并会在进入路由器内部前自动去除 VLAN 2 标签。对于从路由器发出的数据帧，`e1/1.1` 会重新加上 VLAN 2 的标签后发送给交换机。

3. 配置子接口 `e1/1.2` 的封装协议为 VLAN 3，同样可以处理带有 VLAN 3 标签的数据帧，去除标签后在路由器内部处理，并在发送前重新加上 VLAN 3 标签。
4. 将 VLAN 2 的终端设备（如 PC1）与子接口 `e1/1.1` 配置在同一个网段，同时将其网关配置为 `e1/1.1` 的 IP 地址。类似地，将 VLAN 3 的终端设备（如 PC2）与子接口 `e1/1.2` 配置在同一个网段，网关为 `e1/1.2` 的 IP 地址。通过这样的配置，PC1 和 PC2 就能够分别与路由器通信。

```
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

R1(config)#interface e1/1.2
R1(config-if)#encapsulation dot1q 3
R1(config-if)#ip address 3.1.1.1 255.0.0.0
```

此外，通过路由器的路由功能，R1 会自动生成直连路由条目，分别指向 VLAN 2 和 VLAN 3 的网络（对应的子接口为 `e1/1.1` 和 `e1/1.2`）。如果 R1 上还配置了其他接口（如 `e1/2`）与 VLAN 4 相连，则 R1 会生成到 VLAN 4 的直连路由。通过这三条直连路由，路由器可以实现 VLAN 2、VLAN 3 和 VLAN 4 之间的互联。

最终，PC1（VLAN 2）、PC2（VLAN 3）和 PC3（VLAN 4）可以通过路由器进行互相通信，实现跨 VLAN 的互通

```
PC1#ping 2.1.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.1.1.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 92/347
PC1#ping 3.1.1.2
```

```
Sending 5, 100-byte ICMP Echos to 3.1.1.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 8/18/
```

PC1-f0/0----SW1-f0/1 使用ACCESS模式，没有VLAN tag

```
> Frame 35: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface -,
> Ethernet II, Src: c8:01:73:ec:00:00 (c8:01:73:ec:00:00), Dst: cc:06:6a:9c:00:11 (cc:06
> Internet Protocol Version 4, Src: 1.1.1.2, Dst: 3.1.1.2
> Internet Control Message Protocol
```

SW1-f0/11----R1-e1/1.1 使用TRUNK模式，添加VLAN2字段，又R1-e1/1.1接口配置了 dot1q 2，所以路由器可以接受、删除VLAN 2的数据包

```
> Frame 112: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface -,
> Ethernet II, Src: c8:01:73:ec:00:00 (c8:01:73:ec:00:00), Dst: cc:06:6a:9c:00:11 (cc:06
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 2
> Internet Protocol Version 4, Src: 1.1.1.2, Dst: 3.1.1.2
> Internet Control Message Protocol
```

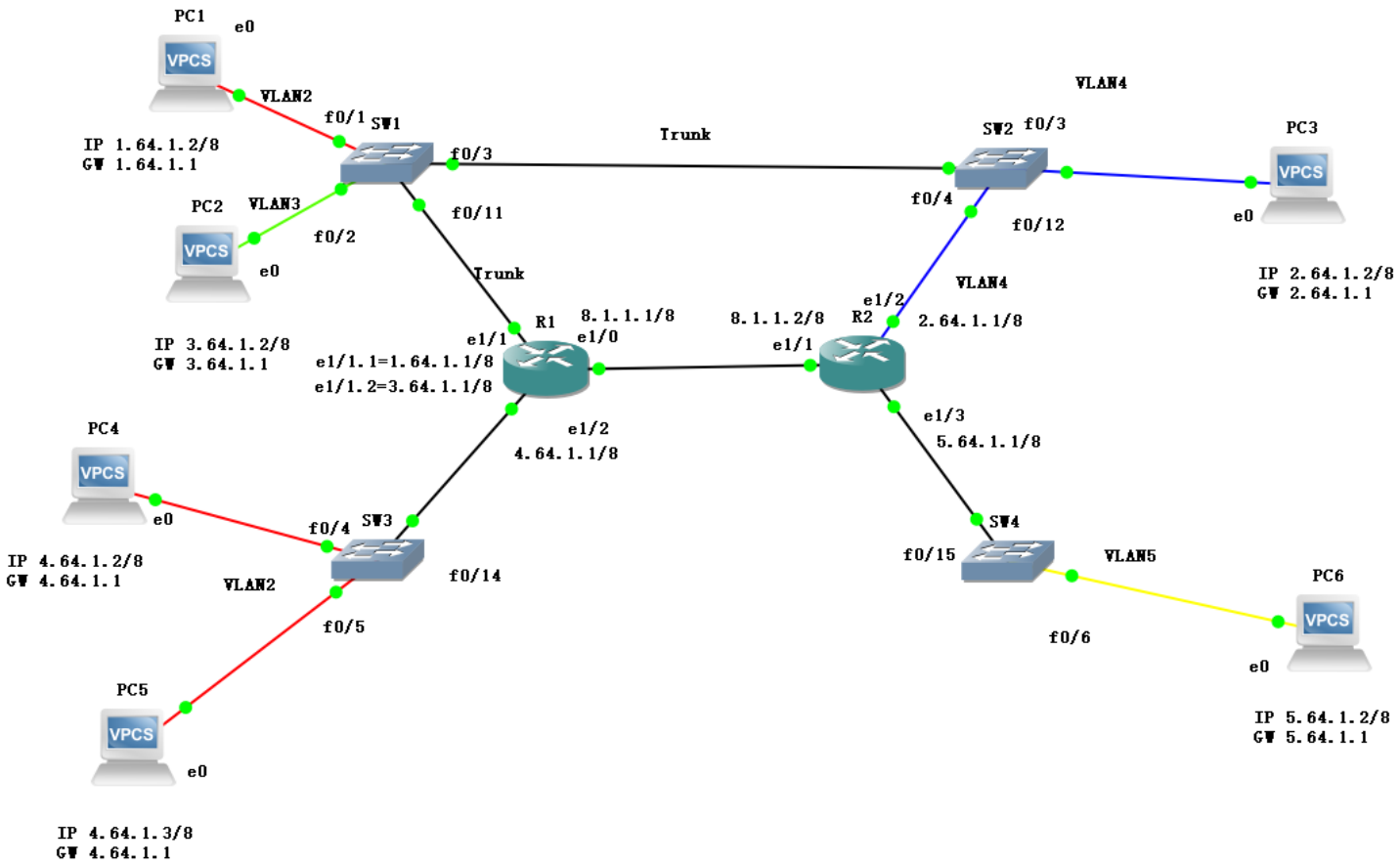
R1-e1/1.2----SW1-f0/12 使用TRUNK模式，且R1-e1/1.2接口配置了dot1q 3，所以数据包会被添加VLAN 3再发送

```
> Frame 113: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface -,
> Ethernet II, Src: cc:06:6a:9c:00:11 (cc:06:6a:9c:00:11), Dst: c8:02:36:90:00:00 (c8:02
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 3
> Internet Protocol Version 4, Src: 1.1.1.2, Dst: 3.1.1.2
> Internet Control Message Protocol
```

SW1-f0/2----PC2-f0/0 使用ACCESS模式，SW1接受到来自VLAN 3的数据包后，删除VLAN tag，将其转发到PC3

```
> Frame 27: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface -,
> Ethernet II, Src: cc:06:6a:9c:00:11 (cc:06:6a:9c:00:11), Dst: c8:02:36:90:00:00 (c8:02
> Internet Protocol Version 4, Src: 1.1.1.2, Dst: 3.1.1.2
> Internet Control Message Protocol
```

网络拓扑3实验



1.完成复杂拓扑互通

我们使用单臂路由配置的方法配置路由器R1的e1/1接口，使用接入VLAN的方式配置R1的e1/2，R2的e1/2与e1/3接口，即可实现复杂拓扑互通

我们这里截图ping命令的响应状况作为验证：

```
PC1> ping 3.64.1.2
3.64.1.2 icmp_seq=1 timeout
84 bytes from 3.64.1.2 icmp_seq=2 ttl=63 time=19.085 ms
84 bytes from 3.64.1.2 icmp_seq=3 ttl=63 time=16.330 ms
84 bytes from 3.64.1.2 icmp_seq=4 ttl=63 time=20.646 ms
84 bytes from 3.64.1.2 icmp_seq=5 ttl=63 time=21.352 ms

PC1> ping 2.64.1.2
2.64.1.2 icmp_seq=1 timeout
2.64.1.2 icmp_seq=2 timeout
2.64.1.2 icmp_seq=3 timeout
84 bytes from 2.64.1.2 icmp_seq=4 ttl=62 time=29.108 ms
84 bytes from 2.64.1.2 icmp_seq=5 ttl=62 time=26.349 ms

PC1> ping 4.64.1.2
4.64.1.2 icmp_seq=1 timeout
84 bytes from 4.64.1.2 icmp_seq=2 ttl=63 time=15.033 ms
84 bytes from 4.64.1.2 icmp_seq=3 ttl=63 time=21.016 ms
84 bytes from 4.64.1.2 icmp_seq=4 ttl=63 time=21.001 ms
84 bytes from 4.64.1.2 icmp_seq=5 ttl=63 time=11.469 ms

PC1> ping 4.64.1.3
4.64.1.3 icmp_seq=1 timeout
84 bytes from 4.64.1.3 icmp_seq=2 ttl=63 time=19.864 ms
84 bytes from 4.64.1.3 icmp_seq=3 ttl=63 time=21.672 ms
84 bytes from 4.64.1.3 icmp_seq=4 ttl=63 time=19.282 ms
84 bytes from 4.64.1.3 icmp_seq=5 ttl=63 time=16.018 ms
```

```
PC1> ping 5.64.1.2
5.64.1.2 icmp_seq=1 timeout
5.64.1.2 icmp_seq=2 timeout
5.64.1.2 icmp_seq=3 timeout
5.64.1.2 icmp_seq=4 timeout
84 bytes from 5.64.1.2 icmp_seq=5 ttl=62 time=30.499 ms
```

可见PC1发出的ping包可以到达所有PC端口

五、实验思考题

所有题目已经在第四章内解决，这里作补充说明

1. VLAN、物理网络与IP网段的关系

VLAN（虚拟局域网）、物理网络和IP网段是网络设计中密切相关的三个关键概念，它们共同决定了网络的逻辑分布、物理实现和通信方式。

1.1 VLAN与IP网段的对应关系

- **一一对应**：通常一个VLAN对应一个IP子网，VLAN内的所有设备都分配在相同的IP网段中。这种配置方式简化了网络管理，并通过路由实现跨VLAN的通信。
- **隔离通信**：不同VLAN（即不同IP网段）的设备默认情况下无法直接通信，必须通过路由器或三层交换机进行数据转发。

1.2 VLAN与物理网络的关系

- **逻辑划分**：通过配置交换机，可以将物理网络中的端口划分到不同的VLAN中，形成独立的广播域。这种划分基于逻辑而非物理连接，提升了网络设计的灵活性。
- **资源共享**：多个VLAN可以共用同一套物理设备（如交换机和链路），而彼此之间的广播流量仍保持隔离，从而提高了资源利用率和安全性。

1.3 IP网段与物理网络的协同

- **逻辑分组与连接**：物理网络提供设备间的实际连接，而IP网段从逻辑上分组这些设备，使得网络层级结构更加清晰。
- **路由优化**：合理的子网划分可以减少路由器的路由表规模，提升转发效率，同时增强网络安全性。

---

2. 不同网段设备互通的实现方法

在实验中，针对不同IP网段的设备通信，常见的实现方法包括：**通过路由器子接口配置（Router-on-a-Stick）和使用多个物理接口**。

2.1 通过路由器子接口实现互通

方法概述：通过路由器的一个物理接口划分为多个子接口，每个子接口绑定一个VLAN，完成多个VLAN间的通信。

配置步骤：

1. **交换机端口配置**：
   - 将连接路由器的交换机端口配置为Trunk模式，允许多个VLAN流量通过。
   - 将连接终端设备的交换机端口配置为Access模式，并将端口分配到相应的VLAN中。
2. **路由器子接口配置**：
   - 在路由器的单个物理接口上创建子接口，每个子接口绑定一个VLAN ID。
   - 配置子接口的IP地址，并启用dot1q封装协议。

```
Router#interface f1/1.1
Router#encapsulation dot1Q 2
```

```
3 Router#ip address 1.1.1.1 255.0.0.0
```

3. **终端设备配置**：
    - 确保终端设备的IP地址属于对应的子网，并将默认网关设置为路由器子接口的IP地址。

实验结论：

- 配置完成后，不同VLAN（或IP网段）的设备可以通过路由器成功互通。
- 此方法高效利用路由器的物理接口资源，适用于中小型网络。

2.2 通过多个物理接口实现互通

方法概述：为每个VLAN使用路由器的独立物理接口，与交换机的不同VLAN分别连接，从而实现跨网段通信。

配置步骤：

1. **交换机端口配置**：
    - 将连接终端设备的端口配置为Access模式，并加入对应的VLAN。
    - 将连接路由器的端口也配置为Access模式，分别对应不同的VLAN。
2. **路由器接口配置**：
    - 每个物理接口分配一个IP地址，与对应的VLAN绑定。

```
1 Router#interface f1/2
2 Router#ip address 2.1.1.1 255.0.0.0
3 Router#no shutdown
```

3. **终端设备配置**：
    - 终端设备的配置与子接口方法类似，将默认网关指向路由器物理接口的IP地址。

实验结论：

- 每个VLAN需要占用一个物理接口，虽然可以实现跨网段通信，但硬件资源需求较高。
- 此方法适用于VLAN数量较少、对硬件资源消耗不敏感的小型网络。

---

3. 端口模式的选择：Trunk模式与Access模式

3.1 Trunk模式

- **定义**：Trunk端口用于传输多个VLAN的数据帧，并通过VLAN标签区分不同的VLAN流量。
- **优点**：
  - **资源节约**：多个VLAN可以共享一条物理链路，减少端口资源浪费。
  - **管理便利**：新增VLAN无需额外物理链路，灵活性高。
- **实验发现**：配置Trunk模式后，未创建的VLAN流量无法通过，增强了网络的安全性和稳定性。

3.2 Access模式

- **定义**：Access端口只允许一个VLAN的数据帧通过，通常用于连接终端设备（如PC）。
- **缺点**：
  - **资源消耗高**：如果在交换机之间传输多个VLAN流量，需要大量Access端口。
  - **扩展性差**：新增VLAN时需频繁调整交换机端口配置。
- **实验发现**：Access模式适合终端设备接入，但在交换机之间的通信中不推荐使用。

---

4. 实验中遇到的问题与解决方法

4.1 物理接口未启用导致通信失败

- **问题描述：**
  配置路由器子接口后，设备无法互相 `ping` 通。
- **原因分析：**
  物理接口未启用，导致子接口无法工作。
- **解决方法：**
  在路由器的物理接口上执行 `no shutdown` 命令，启用物理接口。

4.2 VLAN 标签封装错误

- **问题描述：**
  配置 Trunk 接口后，设备间通信失败。
- **原因分析：**
  交换机之间的 Trunk 接口封装协议不一致。

- **解决方法：**
  确保所有 Trunk 接口的封装协议为 `dot1q`，并检查允许通过的 VLAN 列表是否一致。

4.3 子网掩码配置不当

- **问题描述：**
  不同 IP 网段设备无法互通。
- **原因分析：**
  子网掩码配置错误，导致路由器无法正确识别设备所属子网。
- **解决方法：**
  检查并更正设备的子网掩码配置，确保各设备分配在正确的 IP 网段。

六、实验收获

一、MAC地址表的工作机制

1.1 MAC地址学习与存储

- **连接设备的识别**：
  - 当一台PC连接到交换机的某个端口时，交换机会通过接收来自该PC的数据帧来识别设备的MAC地址。
- **地址表的构建**：
  - 交换机会将学习到的MAC地址与对应的端口进行关联，并存储在MAC地址表（也称为CAM表）中。
- **高效转发**：
  - 这种机制使得交换机能够高效地将数据帧转发到目标设备所在的正确端口，从而减少不必要的广播流量，提高网络性能。

1.2 未知MAC地址的处理

- **洪泛策略（Flooding）**：
  - 对于目标MAC地址尚未出现在MAC地址表中的数据帧，交换机会采取洪泛策略，即将该帧通过所有处于转发状态的端口发送出去，除了接收该帧的端口。
- **确保通信**：
  - 这种机制确保了数据帧能够最终到达目标设备，即使在初始通信过程中，交换机尚未学习到目标设备的MAC地址。

1.3 MAC地址表的动态更新

- **动态学习**：

- 交换机会根据数据帧的来源端口动态更新MAC地址表。当设备移动到不同的端口连接时，交换机会自动更新其MAC地址与新端口的关联。

- **表项老化**：
  - 为了保持MAC地址表的准确性和实时性，交换机通常会对表项设置老化时间，未被更新的表项会被删除，以防止表项过时。

二、链路聚合（Link Aggregation）的实现

2.1 链路聚合的基本原理

- **逻辑链路的构建**：
  - 链路聚合通过将多条物理链路捆绑成一条逻辑链路，实现多链路的并行传输，提升整体带宽和冗余性。
- **性能与容错**：
  - 这种技术不仅提高了网络的传输效率，还增强了网络的容错能力，即使某一条链路出现故障，其他链路仍能维持数据传输。

2.2 配置链路聚合的步骤

- **聚合组的编号**：
  - 在两台交换机（如SW1和SW2）上，通过配置相同的聚合组编号（如Port-channel1），并启用相同的链路聚合协议（如LACP）。
- **物理链路的捆绑**：
  - 将多条物理链路（例如FastEthernet0/4和FastEthernet0/5）捆绑在一起，形成一个逻辑链路。
- **STP的识别**：
  - 配置完成后，STP将识别链路聚合组为一条逻辑链路，从而避免环路问题，同时允许多条物理链路同时传输数据，实现带宽叠加和负载均衡。

2.3 链路聚合的优势

- **带宽叠加**：
  - 通过捆绑多条链路，逻辑链路的总带宽等于各物理链路带宽之和，显著提升数据传输速率。
- **负载均衡**：
  - 数据流量可以在多条链路间均匀分配，避免单一链路过载，提升网络的整体性能。
- **冗余性增强**：
  - 链路聚合提供了冗余路径，即使其中一条链路发生故障，其他链路仍能保持网络的连通性，确保业务的连续性和可靠性。

三、生存树协议（STP）的应用

3.1 防止网络环路

- **环路的危害**：
  - 在交换机互联构建的复杂网络中，存在多条链路连接同一对交换机，容易形成环路。环路会导致广播风暴、MAC地址表混乱等严重问题，影响网络的稳定性和性能。
- **STP的作用**：
  - STP通过动态选择和阻塞冗余链路，确保网络中只有一条无环的路径存在，从而有效防止环路的形成。

3.2 端口角色与状态

- **根桥选举**：
  - STP根据桥标识（Bridge ID）的优先级，选举出根桥（Root Bridge），并确定每个交换机到根桥的最佳路径。
- **端口角色**：
  - **根端口（Root Port）**：每个非根桥上距离根桥最近的端口，负责向根桥转发数据。
  - **指定端口（Designated Port）**：每一段网络中负责转发数据的端口。
  - **阻塞端口（Blocking Port）**：用于阻止数据帧通过，防止环路形成。
- **端口状态**：
  - **监听（Listening）**、**学习（Learning）**、**转发（Forwarding）和阻塞（Blocking）**，每个状态决定了端口在网络中的行为。

四、交换机划分VLAN

4.1 VLAN的基本概念与作用

- **VLAN定义**：
  - VLAN（虚拟局域网）是在物理交换机基础上，通过软件技术将一个物理网络划分为多个逻辑子网的技术。
- **主要目的**：
  - 提高网络的安全性、管理性和灵活性。通过VLAN，网络管理员可以根据部门、功能或其他标准，将不同的设备分配到不同的广播域中，从而减少广播风暴、提升网络性能，并实现更精细的访问控制。

4.2 VLAN与子网的关系

4.2.1 同一VLAN内的主机配置

- **同一子网要求**：
  - 同一VLAN内的主机必须配置在同一个子网中，才能实现直接通信。例如，将两台PC配置为IP地址1.1.1.1/24和1.1.1.2/24，它们能够在同一VLAN内直接互通。

4.2.2 不同VLAN或子网的主机配置

- **不同子网的隔离**：
  - 不同VLAN或不同子网中的主机必须配置在不同的网段，才能通过路由器实现互通。例如，将两台PC配置为IP地址1.1.1.1/24和1.1.2.1/24，它们位于不同的子网中，必须通过路由器才能实现通信。
- **逻辑隔离与路由**：
  - 这种配置要求确保了网络的逻辑隔离，同时通过路由器实现跨子网通信，提升了网络的安全性和管理性。

4.3 STP在VLAN下避免交换机间环路的机制

- **动态调整阻塞链路**：
  - 在配置了VLAN后，STP不仅仅简单地阻塞一条链路，而是根据VLAN的流量情况动态调整，确保同一VLAN内的流量不会通过阻塞链路传输，从而避免环路。
- **VLAN感知的STP**：
  - 现代交换机通常支持VLAN感知的STP（如Per-VLAN Spanning Tree, PVST），能够为每个VLAN独立维护生成树，提供更高的网络容错性和优化路径。

4.4 VLAN互通的实现方式

4.4.1 Trunk方式

- **优势**：
  - 能够同时承载多个VLAN的流量，只需一条物理链路即可实现所有VLAN之间的互通，节省了物理端口资源，提升了网络的灵活性和带宽利用率。
- **推荐应用场景**：
  - 交换机之间的互联，或交换机与支持VLAN的设备（如路由器）之间的连接。
- **VLAN标签**：
  - 通过Trunk端口传输的数据帧会带有VLAN标签（通常为802.1Q），以标识所属的VLAN。

4.4.2 Access方式

- **劣势**：
  - 只能承载单一VLAN的流量，当需要多个VLAN互通时，需要配置多条物理链路，资源浪费且配置复杂。
- **应用限制**：
  - 不适用于需要多VLAN互通的场景，主要用于连接单一VLAN的终端设备。

4.5 Trunk端口允许所有VLAN通过的解释

- **允许通过的VLAN数据帧**：
  - 只有交换机上已经创建的VLAN的数据帧才能通过Trunk端口传输。未在交换机上创建的VLAN的数据帧将无法通过。
- **数据帧的封装**：
  - 通过Trunk端口传输的数据帧会带有VLAN标签（通常为802.1Q），以标识所属的VLAN，确保不同VLAN流量的正确区分和处理。

4.6 同一VLAN中不同子网主机的互通可能性

4.6.1 配置情况

- **单一VLAN内不同子网**：
  - 将两台PC配置为IP地址1.1.0.1/8和1.1.1.1/16，尽管它们位于不同的子网中（网络号分别为1.0.0.0和1.1.0.0），但由于它们处于同一VLAN内，且交换机无需路由，PC1能够直接与PC2通信。

4.6.2 互通原理

- **相同VLAN内的直接通信**：
  - 交换机在同一VLAN内处理所有流量，不需要经过路由器进行跨子网通信。即使子网掩码不同，只要在同一VLAN内，数据帧可以直接在交换机内转发，实现互通。
- **子网判定**：
  - 由于PC只知道目的设备的IP而不知道对方的子网掩码，所以它是以自己的掩码来判断是否在同一子网内，因此导致虽然子网不同但能够通信的情况。

4.6.3 使用路由器实现不同VLAN通信的心得

4.7 使用路由器实现不同VLAN通信的心得

4.7.1 不同VLAN之间通信的基本原理

- **VLAN隔离**：

- VLAN通过逻辑方式将一个物理网络划分为多个独立的广播域，不同VLAN之间的设备默认情况下无法直接通信，这种隔离机制提升了网络的安全性和管理性。

- **跨VLAN通信需求**：
  - 在实际应用中，常常需要不同VLAN之间进行通信，这就需要借助路由器或三层交换机来实现Inter-VLAN路由。

4.7.2 路由器接入多个VLAN的两种方式

1. **多台路由器分别接入各个VLAN**
   - **优点**：
     - 每个VLAN都有独立的路由器接口，配置简单。
   - **缺点**：
     - 需要多台路由器，增加了硬件成本和复杂性，不利于资源的集中管理。
2. **单臂路由（Router-on-a-Stick）**
   - **定义**：
     - 通过在路由器的单个物理接口上配置多个子接口，每个子接口对应一个VLAN，实现不同VLAN之间的通信。
   - **优点**：
     - 节省硬件资源，仅需一台路由器即可实现多VLAN通信，简化了网络架构。
   - **缺点**：
     - 单一链路可能成为瓶颈，影响整体网络性能；依赖单一设备，存在单点故障风险。

4.7.3 逻辑子接口与物理接口的关系

- **逻辑子接口（Sub-interface）**：
  - 是路由器物理接口上的虚拟接口，用于支持多VLAN的流量。每个子接口对应一个特定的VLAN，通过802.1Q（dot1q）封装协议在数据帧中插入VLAN标签，实现不同VLAN流量的识别和分离。
- **关系解析**：
  - **物理接口**：路由器的实际网络接口，如e1/1，用于连接交换机的Trunk端口。
  - **逻辑子接口**：在物理接口上划分的多个虚拟接口，如e1/1.1、e1/1.2，对应VLAN 2和VLAN 3。通过配置逻辑子接口，单一物理接口能够承载多个VLAN的流量，实现不同VLAN之间的路由功能。

4.7.4 单臂路由的完整传输流程

1. **VLAN划分与配置**

- **交换机端口配置**：
  - 将连接PC的端口配置为Access模式，分别加入VLAN 2和VLAN 3。
  - 将连接路由器的端口配置为Trunk模式，允许VLAN 2和VLAN 3的流量通过。
- **路由器子接口配置**：
  - 在路由器的物理接口上创建子接口，分别对应VLAN 2和VLAN 3，并为每个子接口配置相应的IP地址。

2. **数据帧的生成与封装**
   - **发送数据**：
     - PC1（VLAN 2）向PC2（VLAN 3）发送数据包。
   - **交换机处理**：
     - 数据包首先在PC1所属的VLAN 2中被发送到交换机的Access端口。
     - 交换机识别到数据包属于VLAN 2，通过Trunk链路将带有VLAN标签的数据帧发送到路由器的Trunk端口。
3. **路由器处理与转发**
   - **子接口识别**：
     - 路由器接收到带有VLAN 2标签的数据帧，通过子接口e1/1.1识别该流量，并根据配置的IP路由表将数据包转发到子接口e1/1.2（对应VLAN 3）。
   - **数据包转发**：
     - 数据包经过VLAN 3的子接口被发送回交换机，通过Trunk链路发送到PC2所属的Access端口。
4. **PC2接收数据包**
   - **实现通信**：
     - PC2接收到来自PC1的数据包，实现不同VLAN之间的通信。

五、VLAN、物理网络及IP网段的关系

5.1 基本概念

5.1.1 物理网络（Physical Network）

- **定义**：
  - 物理网络指由实际的物理设备（如交换机、路由器、网线、光纤等）构成的网络基础设施。包括网络拓扑结构、设备连接方式和物理介质。
- **功能**：
  - 提供网络设备之间的物理连接，确保数据在设备间传输。

5.1.2 VLAN（虚拟局域网）

- **定义**：
  - VLAN是一种通过逻辑方式将一个物理网络划分为多个独立的广播域的技术。每个VLAN被视为一个独立的网络，即使它们共享相同的物理网络基础设施。
- **功能**：
  - **广播域隔离**：不同VLAN之间的广播流量互相隔离，减少广播风暴的风险。
  - **安全性增强**：通过逻辑隔离，限制不同部门或功能组之间的直接通信，提升网络安全性。
  - **管理灵活性**：基于功能、部门或其他标准划分VLAN，而不受物理位置限制，简化网络管理和配置。

5.1.3 IP网段（子网，Subnet）

- **定义**：
  - IP网段是基于IP地址和子网掩码划分的网络范围。一个子网包含一组连续的IP地址，通常用于定义网络中的逻辑分组。
- **功能**：
  - **地址管理**：有效地分配和管理IP地址，避免地址冲突。
  - **网络分段**：将大型网络分割为多个较小的子网，优化网络性能和安全性。
  - **路由优化**：通过子网划分，优化路由器的路由表，提升数据包转发效率。

5.2 VLAN、物理网络与IP网段的关系

5.2.1 VLAN与物理网络的关系

- **逻辑划分**：
  - VLAN通过交换机的配置，将物理网络中的端口划分到不同的VLAN中。尽管这些端口连接到相同的物理交换机或不同交换机，逻辑上它们属于不同的广播域。
- **物理共享**：
  - 多个VLAN可以共享同一套物理网络基础设施，如交换机、光纤链路等，但彼此之间的流量是隔离的，除非通过路由器或三层交换机进行路由。
- **灵活性**：
  - VLAN的存在使得网络设计更加灵活，管理员可以根据需要动态调整VLAN的成员，而无需重新布线或更换物理设备。

5.2.2 VLAN与IP网段的关系

- **对应关系**：

- 通常，一个VLAN对应一个IP网段（子网），即同一VLAN内的设备配置在同一个IP子网中。这种对应关系有助于管理和路由不同VLAN之间的流量。

- **互通需求**：
  - 不同VLAN（不同IP网段）的设备默认情况下无法直接通信。要实现互通，需要通过路由器或三层交换机进行Inter-VLAN路由，将不同VLAN的流量进行路由转发。
- **广播隔离**：
  - 由于VLAN划分了广播域，同一IP网段内的设备能够互相通信，而跨VLAN的通信需要路由支持，进一步增强了网络的安全性和管理性。

5.2.3 物理网络与IP网段的关系

- **物理连接与逻辑分组**：
  - 物理网络提供设备之间的实际连接，而IP网段则在逻辑层面对这些设备进行分组。一个物理网络可以承载多个IP网段，通过路由器或三层交换机实现不同网段之间的通信。
- **子网划分**：
  - 在一个物理网络中，根据组织需求和网络规模，可以将IP地址空间划分为多个子网，每个子网对应不同的功能区域或部门，优化网络性能和管理。
- **路由配置**：
  - 路由器需要配置正确的路由表，以确保数据包能够在不同的IP网段之间正确转发，保证网络的连通性和高效性。

六、总结

MAC地址表：

- MAC地址表是交换机高效转发数据的基础，动态学习与更新机制确保了网络的灵活性和性能。

STP作用：

- 生存树协议在防止网络环路、维护网络稳定性方面不可或缺，理解其端口角色与状态有助于更好地配置和优化网络。

链路聚合：

- 通过链路聚合，可以有效提升网络带宽和冗余性，增强整体网络的可靠性和效率。

VLAN管理：

- VLAN通过逻辑隔离，实现了广播域的划分与管理的灵活性，同时提升了网络的安全性，是现代网络设计中的重要工具。

VLAN路由实现：

- 理解并掌握不同VLAN之间的通信实现方式，如单臂路由（Router-on-a-Stick），对构建复杂且高效的网络至关重要。

物理网络与逻辑分组协调：

- 物理网络提供基础连接，VLAN和IP网段的逻辑划分则赋予了网络更高的灵活性和管理性，二者的协调是实现高效网络设计的关键。

七、实验心得

在信息技术飞速发展的今天，网络的高效性与安全性成为企业运营的基石。作为一名计科专业的学生，我有幸参与了一次关于虚拟局域网（VLAN）配置与组网的实验。这次实验不仅加深了我对VLAN理论的理解，更通过让我体会到了网络设计与配置的复杂与乐趣。

在正式学习之前，我对VLAN的认识仅停留在课本上的定义：VLAN是一种通过逻辑方式将一个物理网络划分为多个独立广播域的技术。在老师的指导下，我们在GNS3上将交换机与路由器连接，并将几台PC分别接入交换机的不同端口。看似简单的步骤，却蕴含着网络拓扑设计的基本原则。确认无误后，我们开始了VLAN的配置。

在配置到路由器的时候，VLAN的隔离虽然提升了网络的安全性和管理性，但在实际应用中，不同VLAN之间的通信是不可避免的需求。这就需要通过路由器或三层交换机实现Inter-VLAN路由。我们选择了“Router-on-a-Stick”的配置方法，通过路由器的单一物理接口实现多VLAN的通信。

首先，在路由器上配置子接口，每个子接口对应一个VLAN，并分配相应的IP地址。接着，在交换机上配置Trunk端口，允许多个VLAN的流量通过，并连接到路由器的物理接口。配置完成后，我们在不同VLAN内的PC上分别配置了对应的IP地址，并再次进行了互ping测试。这一次，来自不同VLAN的PC成功实现了通信，验证了跨VLAN通信的实现。

这次VLAN组网实验，我们不仅成功配置了VLAN，实现了广播域的隔离，还通过Router-on-a-Stick的方法实现了不同VLAN之间的通信。实验成果也很显著：

1. **VLAN隔离的有效性**：不同VLAN内的PC无法直接通信，验证了VLAN的广播域隔离功能。
2. **跨VLAN通信的实现**：通过路由器配置，实现了不同VLAN间的互联，PC之间能够正常通信。
3. **带宽利用的优化**：Trunk端口成功承载了多个VLAN的流量，提高了物理链路的利用率。
4. **网络管理的简化**：VLAN配置使得不同部门或功能组的设备能够灵活划分到各自的VLAN中，简化了网络管理。

这次VLAN配置组网的实验，让我对CS有了更深入的理解和体会。未来，我将继续深入学习和实践，探索更复杂的网络配置与管理技术，如多层交换、动态路由协议、安全策略配置等，不断提升自己的技术水平。

网络技术的发展日新月异，只有不断学习与实践，才能在这个领域中游刃有余。通过这次实验，我不仅掌握了VLAN的配置方法，更培养了系统性思维和问题解决能力，这将为我未来的网络工程师之路奠定坚实的基础。