



警告

1. 实验报告如有雷同，雷同各方当次实验成绩均以 0 分计。
2. 当次小组成员成绩只计学号、姓名登录在下表中的。
3. 规定时间内未上交实验报告，不得以其他方式补交，当次成绩按 0 分计。
4. 根据组员在实验中的贡献，实事求是自评实验分数。（按百分制）

院系	计算机学院	班 级	计算机科学与技术 1 班
学号	21307035		
学生	邓栩瀛		

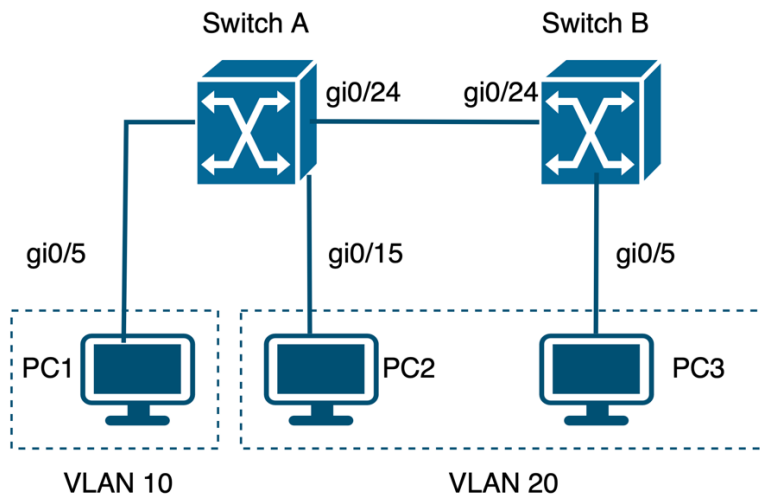
跨交换机实现 VLAN

实验目的: 理解跨交换机之间 VLAN 的特点。使在同一 VLAN 内的计算机系统能跨交换机进行相互通信，而在不同 VLAN 的计算机系统不能进行相互通信。

技术原理: Tag Vlan 是基于交换机端口的一种类型，主要用于实现跨交换机的相同 VLAN 内的主机之间可以直接访问，同时对不同 VLAN 的主机进行隔离。Tag Vlan 遵循 IEEE 802.1q 协议标准。在利用配置了 Tag Vlan 的端口进行数据传输时，需要在数据帧内添加 4B 的 802.1q 标签信息，用于表示该数据帧属于哪个 VLAN，以便在段交换机接收到数据帧后进行准确的过滤。

实验设备: 交换机 2 台，计算机 3 台

实验拓扑:



实验步骤:

步骤 1: 实验前的测试

(1) 实验开始时，用 netsh 命令将 PC1、PC2、PC3 的网卡分别配置如下 IP、掩码:

```
PC1 192.168.10.10 255.255.255.0
PC2 192.168.10.20 255.255.255.0
PC3 192.168.10.30 255.255.255.0
```

验证 3 台主机是否可以两两 ping 通



PC1

PC2

```
C:\Users\D502>ping 192.168.10.30

正在 Ping 192.168.10.30 具有 32 字节的数据:
来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=128

192.168.10.30 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms, 最长 = 0ms, 平均 = 0ms

C:\Users\D502>ping 192.168.10.20

正在 Ping 192.168.10.20 具有 32 字节的数据:
来自 192.168.10.20 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.10.20 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.10.20 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.10.20 的回复: 字节=32 时间<1ms TTL=128

192.168.10.20 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms, 最长 = 0ms, 平均 = 0ms
```

```
C:\Users\D502>ping 192.168.10.10

正在 Ping 192.168.10.10 具有 32 字节的数据:
来自 192.168.10.10 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.10.10 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.10.10 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.10.10 的回复: 字节=32 时间<1ms TTL=128

192.168.10.10 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms, 最长 = 1ms, 平均 = 0ms

C:\Users\D502>ping 192.168.10.30

正在 Ping 192.168.10.30 具有 32 字节的数据:
来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=128

192.168.10.30 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms, 最长 = 0ms, 平均 = 0ms
```

PC3

```
C:\Users\D502>ping 192.168.10.10

正在 Ping 192.168.10.10 具有 32 字节的数据:
来自 192.168.10.10 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.10.10 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.10.10 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.10.10 的回复: 字节=32 时间=1ms TTL=128

192.168.10.10 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms, 最长 = 1ms, 平均 = 0ms

C:\Users\D502>ping 192.168.10.20

正在 Ping 192.168.10.20 具有 32 字节的数据:
来自 192.168.10.20 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.10.20 的回复: 字节=32 时间=1ms TTL=128
来自 192.168.10.20 的回复: 字节=32 时间=1ms TTL=128
来自 192.168.10.20 的回复: 字节=32 时间<1ms TTL=128

192.168.10.20 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms, 最长 = 1ms, 平均 = 0ms
```

(2) 记录交换机 A 和交换机 B 的 VLAN 信息

Switch A

```
11-S5750-1#show vlan
VLAN Name                Status    Ports
-----
1 VLAN0001                STATIC    Gi0/1, Gi0/2, Gi0/3, Gi0/4
                                   Gi0/5, Gi0/6, Gi0/7, Gi0/8
                                   Gi0/9, Gi0/10, Gi0/11, Gi0/12
                                   Gi0/13, Gi0/14, Gi0/15, Gi0/16
                                   Gi0/17, Gi0/18, Gi0/19, Gi0/20
                                   Gi0/21, Gi0/22, Gi0/23, Gi0/24
                                   Gi0/25, Gi0/26, Gi0/27, Gi0/28
```

Switch B

```
11-S5750-2#show vlan
VLAN Name                Status    Ports
-----
1 VLAN0001                STATIC    Gi0/1, Gi0/2, Gi0/3, Gi0/4
                                   Gi0/5, Gi0/6, Gi0/7, Gi0/8
                                   Gi0/9, Gi0/10, Gi0/11, Gi0/12
                                   Gi0/13, Gi0/14, Gi0/15, Gi0/16
                                   Gi0/17, Gi0/18, Gi0/19, Gi0/20
                                   Gi0/21, Gi0/22, Gi0/23, Gi0/24
                                   Gi0/25, Gi0/26, Gi0/27, Gi0/28
```

步骤 2: 在交换机 A 上创建 VLAN10, 并将端口 0/5 划分到 VLAN10 中



```
11-S5750-1>enable 14

Password:
11-S5750-1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
11-S5750-1(config)#vlan 10
11-S5750-1(config-vlan)#name sales
11-S5750-1(config-vlan)#exit
11-S5750-1(config)#interface gigabitethernet 0/5
11-S5750-1(config-if-GigabitEthernet 0/5)#switchport access vlan 10
11-S5750-1(config-if-GigabitEthernet 0/5)#show vlan id 10
VLAN Name                Status    Ports
-----
10 sales                  STATIC    Gi0/5
```

验证测试:

(1) 在交换机 A 上通过命令 show vlan id 10 验证是否已创建 VLAN 10, 查看端口 0/5 是否已划分到 VLAN 10 中。

```
11-S5750-1(config-if-GigabitEthernet 0/5)#show vlan id 10
VLAN Name                Status    Ports
-----
10 sales                  STATIC    Gi0/5
```

端口 0/5 已划分到 VLAN 10 中

(2) 检查 PC1、PC2、PC3 此时的连通情况

PC1

```
C:\Users\D502>ping 192.168.10.20

正在 Ping 192.168.10.20 具有 32 字节的数据:
请求超时。
请求超时。
来自 192.168.10.10 的回复: 无法访问目标主机。
请求超时。

192.168.10.20 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 1, 丢失 = 3 (75% 丢失),

C:\Users\D502>ping 192.168.10.30

正在 Ping 192.168.10.30 具有 32 字节的数据:
请求超时。
请求超时。
来自 192.168.10.10 的回复: 无法访问目标主机。
请求超时。

192.168.10.30 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 1, 丢失 = 3 (75% 丢失),
```

PC2

```
C:\Users\D502>ping 192.168.10.10

正在 Ping 192.168.10.10 具有 32 字节的数据:
请求超时。
请求超时。
来自 192.168.10.20 的回复: 无法访问目标主机。
请求超时。

192.168.10.10 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 1, 丢失 = 3 (75% 丢失),

C:\Users\D502>ping 192.168.10.30

正在 Ping 192.168.10.30 具有 32 字节的数据:
来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=128

192.168.10.30 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms, 最长 = 0ms, 平均 = 0ms
```

PC3

```
C:\Users\D502>ping 192.168.10.10

正在 Ping 192.168.10.10 具有 32 字节的数据:
请求超时。
请求超时。
来自 192.168.10.30 的回复: 无法访问目标主机。
请求超时。

192.168.10.10 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 1, 丢失 = 3 (75% 丢失),

C:\Users\D502>ping 192.168.10.20

正在 Ping 192.168.10.20 具有 32 字节的数据:
来自 192.168.10.20 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.10.20 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.10.20 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.10.20 的回复: 字节=32 时间<1ms TTL=128

192.168.10.20 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms, 最长 = 1ms, 平均 = 0ms
```

PC1 与 PC2、PC1 与 PC3 之间不能连通, PC2 与 PC3 可以连通

步骤 3: 在交换机 A 上创建 VLAN 20, 并将端口 0/15 划分到 VLAN 20 中。



```
11-S5750-1(config-if-GigabitEthernet 0/5)#exit
11-S5750-1(config)#vlan 20
11-S5750-1(config-vlan)#name technical
11-S5750-1(config-vlan)#exit
11-S5750-1(config)#interface gigabitethernet 0/15
11-S5750-1(config-if-GigabitEthernet 0/15)#switchport access vlan 20
```

验证测试:

(1) 在交换机 A 上通过命令 show vlan id 20 验证是否已创建 VLAN 20, 查看端口 0/15 是否已划分到 VLAN 20 中。

```
11-S5750-1(config-if-GigabitEthernet 0/15)#show vlan id 20
```

VLAN Name	Status	Ports
20 technical	STATIC	Gi0/15

端口 0/15 已划分到 VLAN 20 中

(2) 检查 PC1、PC2、PC3 此时的连通情况

PC1

PC2

```
C:\Users\D502>ping 192.168.10.20
正在 Ping 192.168.10.20 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。
192.168.10.20 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),

C:\Users\D502>ping 192.168.10.30
正在 Ping 192.168.10.30 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。
192.168.10.30 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),

C:\Users\D502>ping 192.168.10.10
正在 Ping 192.168.10.10 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。
192.168.10.10 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),

C:\Users\D502>ping 192.168.10.20
正在 Ping 192.168.10.20 具有 32 字节的数据:
请求超时。
请求超时。
来自 192.168.10.20 的回复: 无法访问目标主机。
请求超时。
192.168.10.20 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 1, 丢失 = 3 (75% 丢失),
```

PC3

```
C:\Users\D502>ping 192.168.10.10
正在 Ping 192.168.10.10 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。
192.168.10.10 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),

C:\Users\D502>ping 192.168.10.20
正在 Ping 192.168.10.20 具有 32 字节的数据:
请求超时。
请求超时。
来自 192.168.10.30 的回复: 无法访问目标主机。
请求超时。
192.168.10.20 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 1, 丢失 = 3 (75% 丢失),
```

PC1 与 PC2、PC1 与 PC3、PC2 与 PC3 都不能连通

步骤 4: 将交换机 A 与交换机 B 相连的端口 0/24 定义为 Tag VLAN 模式



计算机网络实验报告

```
11-S5750-1(config-if-GigabitEthernet 0/15)#exit
11-S5750-1(config)#interface gigabitethernet 0/24
11-S5750-1(config-if-GigabitEthernet 0/24)#switchport mode trunk
11-S5750-1(config-if-GigabitEthernet 0/24)#exit
```

验证测试：端口 0/24 已被设置为 trunk 模式

```
11-S5750-1(config)#show interfaces gigabitethernet 0/24 switchport
Interface                               Switchport Mode      Access Native Protected VL
AN lists
```

```
-----
GigabitEthernet 0/24                  enabled    TRUNK          1          1          Disabled AL
L
```

信息显示：端口 0/24 已打开，模式为 trunk。

验证测试：检查 PC1、PC2、PC3 的连通情况

PC1

```
C:\Users\D502>ping 192.168.10.20
正在 Ping 192.168.10.20 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。
192.168.10.20 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),
C:\Users\D502>ping 192.168.10.30
正在 Ping 192.168.10.30 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。
192.168.10.30 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),
```

PC2

```
C:\Users\D502>ping 192.168.10.10
正在 Ping 192.168.10.10 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。
192.168.10.10 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),
C:\Users\D502>ping 192.168.10.30
正在 Ping 192.168.10.30 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。
192.168.10.30 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),
```

PC3

```
C:\Users\D502>ping 192.168.10.10
正在 Ping 192.168.10.10 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。
192.168.10.10 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),
C:\Users\D502>ping 192.168.10.20
正在 Ping 192.168.10.20 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。
192.168.10.20 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),
```

PC1 与 PC2、PC1 与 PC3、PC2 与 PC3 都不能连通

步骤 5：在交换机 B 上创建 VLAN 20，并将端口 0/5 划分到 VLAN 20 中

```
11-S5750-2(config)#vlan 20
11-S5750-2(config-vlan)#name technical
11-S5750-2(config-vlan)#exit
11-S5750-2(config)#interface gigabitethernet 0/5
11-S5750-2(config-if-GigabitEthernet 0/5)#switchport access vlan 20
```

验证测试：

(1) 验证已在交换机 B 上创建 VLAN 20，查看端口 0/5 的划分情况



计算机网络实验报告

```
11-S5750-2(config-if-GigabitEthernet 0/5)#show vlan id 20
```

VLAN Name	Status	Ports
20 technical	STATIC	Gi0/5

端口 0/5 已划分到 VLAN 20

(2) 检查 PC1、PC2、PC3 此时的连通情况

PC1

```
C:\Users\D502>ping 192.168.10.20
正在 Ping 192.168.10.20 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。
192.168.10.20 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),

C:\Users\D502>ping 192.168.10.30
正在 Ping 192.168.10.30 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。
192.168.10.30 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),
```

PC2

```
C:\Users\D502>ping 192.168.10.10
正在 Ping 192.168.10.10 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。
192.168.10.10 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),

C:\Users\D502>ping 192.168.10.30
正在 Ping 192.168.10.30 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。
192.168.10.30 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),
```

PC3

```
C:\Users\D502>ping 192.168.10.10
正在 Ping 192.168.10.10 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。
192.168.10.10 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),

C:\Users\D502>ping 192.168.10.20
正在 Ping 192.168.10.20 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。
192.168.10.20 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),
```

PC1 与 PC2、PC1 与 PC3、PC2 与 PC3 都不能连通

步骤 6: 将交换机 B 与交换机 A 相连的端口 0/24 定义为 Tag VLAN 模式。

```
11-S5750-2(config)#interface gigabitethernet 0/24
11-S5750-2(config-if-GigabitEthernet 0/24)#switchport mode trunk
```

步骤 7: 验证 PC2 与 PC3 能相互通信, 但 PC1 与 PC3 不能相互通信

启动监控软件 Wireshark, 用 ping 命令测试 3 台主机的连通性, 并进行以下观察:

(1) 主机直接能否相互通信?

PC1

PC2



```
C:\Users\D502>ping 192.168.10.20

正在 Ping 192.168.10.20 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。

192.168.10.20 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),

C:\Users\D502>ping 192.168.10.30

正在 Ping 192.168.10.30 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。

192.168.10.30 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),

C:\Users\D502>ping 192.168.10.10

正在 Ping 192.168.10.10 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。

192.168.10.10 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),
```

PC3

```
C:\Users\D502>ping 192.168.10.10

正在 Ping 192.168.10.10 具有 32 字节的数据:
请求超时。
请求超时。
请求超时。
请求超时。

192.168.10.10 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 0, 丢失 = 4 (100% 丢失),

C:\Users\D502>ping 192.168.10.20

正在 Ping 192.168.10.20 具有 32 字节的数据:
来自 192.168.10.20 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.10.20 的回复: 字节=32 时间=1ms TTL=128
来自 192.168.10.20 的回复: 字节=32 时间<1ms TTL=128
来自 192.168.10.20 的回复: 字节=32 时间=1ms TTL=128

192.168.10.20 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms, 最长 = 1ms, 平均 = 0ms
```

PC1 与 PC2、PC3 均不能通信，PC2 与 PC3 直接可以相互通信

(2) 能否监测到 PC1、PC2、PC3 的 ICMP 包？

PC2

19	44.208592	192.168.10.20	192.168.10.30	ICMP	74	Echo (ping) request	id=0x0001, seq=105/26880, ttl=128 (reply i
20	44.209259	192.168.10.30	192.168.10.20	ICMP	78	Echo (ping) reply	id=0x0001, seq=105/26880, ttl=128 (request
21	45.215322	192.168.10.20	192.168.10.30	ICMP	74	Echo (ping) request	id=0x0001, seq=106/27136, ttl=128 (reply i
22	45.216003	192.168.10.30	192.168.10.20	ICMP	78	Echo (ping) reply	id=0x0001, seq=106/27136, ttl=128 (request
23	45.930157	192.168.10.30	192.168.10.20	ICMP	78	Echo (ping) request	id=0x0001, seq=113/28928, ttl=128 (reply i
24	45.930286	192.168.10.20	192.168.10.30	ICMP	74	Echo (ping) reply	id=0x0001, seq=113/28928, ttl=128 (request
25	46.227415	192.168.10.20	192.168.10.30	ICMP	74	Echo (ping) request	id=0x0001, seq=107/27392, ttl=128 (reply i
26	46.227892	192.168.10.30	192.168.10.20	ICMP	78	Echo (ping) reply	id=0x0001, seq=107/27392, ttl=128 (request
27	46.948379	192.168.10.30	192.168.10.20	ICMP	78	Echo (ping) request	id=0x0001, seq=114/29184, ttl=128 (reply i
28	46.948521	192.168.10.20	192.168.10.30	ICMP	74	Echo (ping) reply	id=0x0001, seq=114/29184, ttl=128 (request
29	47.236648	192.168.10.20	192.168.10.30	ICMP	74	Echo (ping) request	id=0x0001, seq=108/27648, ttl=128 (reply i
30	47.237551	192.168.10.30	192.168.10.20	ICMP	78	Echo (ping) reply	id=0x0001, seq=108/27648, ttl=128 (request
31	47.963330	192.168.10.30	192.168.10.20	ICMP	78	Echo (ping) request	id=0x0001, seq=115/29440, ttl=128 (reply i
32	47.963461	192.168.10.20	192.168.10.30	ICMP	74	Echo (ping) reply	id=0x0001, seq=115/29440, ttl=128 (request
38	48.975919	192.168.10.30	192.168.10.20	ICMP	78	Echo (ping) request	id=0x0001, seq=116/29696, ttl=128 (reply i
39	48.976049	192.168.10.20	192.168.10.30	ICMP	74	Echo (ping) reply	id=0x0001, seq=116/29696, ttl=128 (request

PC3

31	70.941244	192.168.10.20	192.168.10.30	ICMP	78	Echo (ping) request	id=0x0001, seq=105/26880, ttl=128 (reply in 32
32	70.941446	192.168.10.30	192.168.10.20	ICMP	74	Echo (ping) reply	id=0x0001, seq=105/26880, ttl=128 (request in
33	71.947999	192.168.10.20	192.168.10.30	ICMP	78	Echo (ping) request	id=0x0001, seq=106/27136, ttl=128 (reply in 34
34	71.948078	192.168.10.30	192.168.10.20	ICMP	74	Echo (ping) reply	id=0x0001, seq=106/27136, ttl=128 (request in
35	72.661996	192.168.10.30	192.168.10.20	ICMP	74	Echo (ping) request	id=0x0001, seq=113/28928, ttl=128 (reply in 36
36	72.662568	192.168.10.20	192.168.10.30	ICMP	78	Echo (ping) reply	id=0x0001, seq=113/28928, ttl=128 (request in
38	72.959764	192.168.10.20	192.168.10.30	ICMP	78	Echo (ping) request	id=0x0001, seq=107/27392, ttl=128 (reply in 39
39	72.959961	192.168.10.30	192.168.10.20	ICMP	74	Echo (ping) reply	id=0x0001, seq=107/27392, ttl=128 (request in
40	73.680200	192.168.10.30	192.168.10.20	ICMP	74	Echo (ping) request	id=0x0001, seq=114/29184, ttl=128 (reply in 41
41	73.681133	192.168.10.20	192.168.10.30	ICMP	78	Echo (ping) reply	id=0x0001, seq=114/29184, ttl=128 (request in
42	73.969268	192.168.10.20	192.168.10.30	ICMP	78	Echo (ping) request	id=0x0001, seq=108/27648, ttl=128 (reply in 43
43	73.969445	192.168.10.30	192.168.10.20	ICMP	74	Echo (ping) reply	id=0x0001, seq=108/27648, ttl=128 (request in
44	74.695298	192.168.10.30	192.168.10.20	ICMP	74	Echo (ping) request	id=0x0001, seq=115/29440, ttl=128 (reply in 45
45	74.695901	192.168.10.20	192.168.10.30	ICMP	78	Echo (ping) reply	id=0x0001, seq=115/29440, ttl=128 (request in
46	75.038056	192.168.10.30	192.168.10.255	UDP	1482	49949 → 1689 Len=1440	
51	75.707759	192.168.10.30	192.168.10.20	ICMP	74	Echo (ping) request	id=0x0001, seq=116/29696, ttl=128 (reply in 52
52	75.708573	192.168.10.20	192.168.10.30	ICMP	78	Echo (ping) reply	id=0x0001, seq=116/29696, ttl=128 (request in

PC2 和 PC3 可以监测到 ICMP 包，而 PC1 不能监测到 ICMP 包

(3) 能否捕获到 Trunk 链路上的 VLAN ID？请讨论原因

Wireshark 只能在 Trunk 链路的两端捕获到完整的 VLAN 信息，而在链路中间的设备（如交换机）通



常会剥离 VLAN 标签，并根据 VLAN ID 进行处理。因此，如果 Wireshark 在链路中间的设备上进行捕获，可能无法看到 VLAN ID。而如果 Wireshark 在链路的源端或目的端进行捕获，能够获取到完整的 VLAN 信息。

(4) 查看交换机的地址表。清除地址表，适当更改、增加网线接口，然后观察与分析地址表的形成与变化过程（配合 Wireshark 分析洪泛现象）。Show mac-address-table 命令显示的 MAC 地址与在命令提示符下通过 ipconfig/all 显示的 MAC 地址是否相同？

Show mac-address-table

```
11-S5750-2(config)#show mac-address-table
```

Vlan	MAC Address	Type	Interface
1	4433.4c0e.c260	DYNAMIC	GigabitEthernet 0/24
1	5869.6c15.5512	DYNAMIC	GigabitEthernet 0/24
20	4433.4c0e.ab71	DYNAMIC	GigabitEthernet 0/5
20	4433.4c0e.c260	DYNAMIC	GigabitEthernet 0/24

在 PC3 中，ipconfig/all

```
连接特定的 DNS 后缀 . . . . . :  
描述. . . . . : Realtek Common Ethernet Controllers  
物理地址. . . . . : 44-33-4C-0E-AB-71  
DHCP 已启用 . . . . . : 否  
自动配置已启用. . . . . : 是  
本地链接 IPv6 地址. . . . . : fe80::f7da:6c4f:af70:4d12%5(首选)  
IPv4 地址 . . . . . : 192.168.10.30(首选)  
子网掩码 . . . . . : 255.255.255.0
```

在 PC3 中，两个指令显示的 mac 地址都是 44-33-4C-0E-AB-71

(5) 判断实验是否达到预期目标

实验达到预期目标，在统一 VLAN 内的计算机系统能跨交换机进行相互通信，而在不同 VLAN 的计算机系统不能进行相互通信。