



# 计算机网络实验报告

## 警示

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. 在规定时间内未上交实验报告的，不得以其他方式补交，当次成绩按 0 分计。
4. 实验报告文件以 PDF 格式提交。

专业	软件工程	班 级	19 级软件工程	组长	冼子婷
学号	18338072	18346019	18322043		
学生	冼子婷	胡文浩	廖雨轩		
实验分工					
冼子婷	进行实验，截图，编写和分析实验报告		廖雨轩	进行实验，截图，编写和分析实验报告	
胡文浩	进行实验，截图，编写和分析实验报告				

## 【实验题目】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 里的计算机系统能跨交换机进行相互通信、而在不同 VLAN 里的计算机系统不能进行相互通信。

## 【实验内容】

- (1) 完成实验教材第 6 章实验 6-2 的实验(p172)。
- (2) 完成本章习题 6 的练习 9(p217)，用 Wireshark 进行抓包的时候注意截图，分析实验结果。
- (3) 跨交换机实现 VLAN 通信时，思考不用 Trunk 模式且也能进行跨交换机 VLAN 通信的替代方法，并进行实验验证。

## 【实验要求】

一些重要信息比如 VLAN 信息需给出截图，注意实验步骤的前后对比！

## 【实验记录】(如有实验拓扑，要求自行画出拓扑图，并表明 VLAN 以及相关接口。)

本次实验完成后，请根据组员在实验中的贡献，请实事求是，自评在实验中应得的分数。(按百分制)

## 一、实验 6-2：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 内的计算机系统能够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

### 【实验设备】

交换机 2 台，计算机 3 台

### 【实验拓扑】

本实验的拓扑结构如图所示：

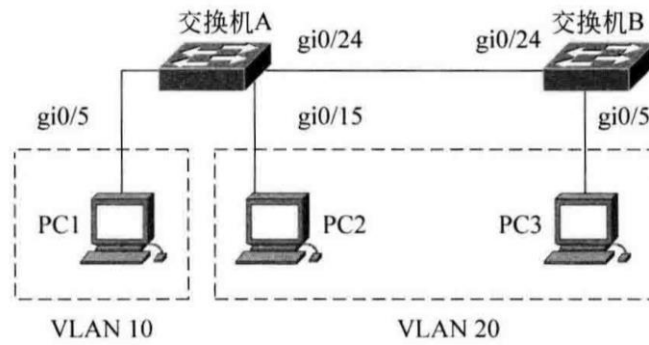


图 6-11 跨交换机实现 VLAN 实验拓扑

## 【实验步骤】

步骤 1: 在未划分 VLAN 前测试 3 台计算机的连通状态

(1) 验证 3 台主机是否可以两两互相 ping 通。

```
管理员: C:\Windows\system32\cmd.exe

C:\Users\Administrator>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=64
来自 192.168.10.20 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.10.20 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.10.20 的回复: 字节=32 时间<1ms TTL=64

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\Administrator>
```

```
管理员: 命令提示符

最短 = 0ms, 最长 = 0ms, 平均 = 0ms
Control-C
^C
C:\Windows\system32>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=64
来自 192.168.10.10 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.10.10 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.10.10 的回复: 字节=32 时间<1ms TTL=64

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 = 0ms, 平均 = 0ms

C:\Windows\system32>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=64
来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=64

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:\Windows\system32>
```

PC2: 按照拓扑图链接后, PC1、  
PC2、PC3之间互联互通



```
C:\Users\Administrator>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=64
来自 192.168.10.10 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.10.10 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.10.10 的回复: 字节=32 时间<1ms TTL=64

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 = 0ms, 平均 = 0ms

C:\Users\Administrator>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=64
来自 192.168.10.20 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.10.20 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.10.20 的回复: 字节=32 时间<1ms TTL=64

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
```

PC3

按照网络拓扑图连接计算机和交换机后，使用 netsh 设置每台计算机的 IP 和子网掩码之后，互相进行 ping 操作发现是互联互通的。

```
C:\Windows\system32>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=64
来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=64

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:\Windows\system32>netsh interface ip set address "实验网 2" static 192.168.10.20 255.255.255.0
```

(2) 记录交换机 A 和交换机 B 的 VLAN 信息

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

实验前交换机中仅有一个默认的 VLAN1，所有与交换机连接的设备都属于 VLAN1

步骤 2: 在交换机 A 上创建 VLAN10，并将端口 0/5 划分到 VLAN 10 中。

(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



```
SwitchA(config)#show vlan
```

VLAN Name	Status	Ports
1 VLAN0001	STATIC	Gi0/1, Gi0/2, Gi0/3, Gi0/4 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
10 sales	STATIC	Gi0/5
20 technical	STATIC	

```
SwitchA(config)#
```

(2) 检查 PC1、PC2、PC3 此时的连通情况

```
管理员: C:\Windows\system32\cmd.exe
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\Administrator>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\Administrator>
```

PC1

```
管理员: C:\Windows\system32\cmd.exe

C:\Users\Administrator>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=64
    来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=64
    来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=64
    来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=64

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\Administrator>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),

C:\Users\Administrator>
```

PC2

```
C:\Windows\system32>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=64
    来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=64
    来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=64
    来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=64

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:\Windows\system32>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:\Windows\system32>
```



```
C:\Users\Administrator>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),
    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\Administrator>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=64
来自 192.168.10.20 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.10.20 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.10.20 的回复: 字节=32 时间<1ms TTL=64

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
```

PC3

将 SwitchA 的 0/5 端口划分进入到 VLAN 10。由于 0/5 端口进入的计算机是 PC1，因此 PC1 被单独划分到一个虚拟网络之中。此时可以看见 PC1 与 PC2、PC3 不能 ping 通，而 PC2 和 PC3 则正常 ping 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。

(1) 在交换机 A 上通过命令 show vlan id 20 验证是否已创建 VLAN 10，查看端口 0/15 是否已划分到 VLAN 20 中

```
SwitchA(config)#show vlan
VLAN Name        Status    Ports
-----
 1 VLAN0001      STATIC    Gi0/1, Gi0/2, Gi0/3, Gi0/4
                               Gi0/6, Gi0/7, Gi0/8, Gi0/9
                               Gi0/10, Gi0/11, Gi0/12, Gi0/13
                               Gi0/14, 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
10 sales          STATIC    Gi0/5
20 technical      STATIC    Gi0/15
SwitchA(config)#
```

(2) 检查 PC1、PC2、PC3 此时的连通情况



```
C:\Users\Administrator>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

管理员: C:\Windows\system32\cmd.exe

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\Administrator>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),

C:\Users\Administrator>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\Administrator>

C:\Windows\system32>ping 192.168.10.10
正在 Ping 192.168.10.10 具有 32 字节的数据: PC2
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:\Windows\system32>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 192.168.10.20 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),

C:\Windows\system32>_

C:\Users\Administrator>ping 192.168.10.10
正在 Ping 192.168.10.10 具有 32 字节的数据: PC3
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\Administrator>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),
```

在 SwitchA 交换机中已经创建 VLAN10 和 VLAN20，并将端口 0/5 分到 VLAN10，将端口 0/15 分到了 VLAN20，于是 PC1 相当于单独隔离，不论是 pingPC2 还是 PC3 都无法 ping 通。



# 计算机网络实验报告

而 PC2 由于被划分到了 VLAN20，虽然与 PC3 在同一个 VLAN 中，但并未实现跨交换机通信，所以仍然无法与 PC3 互通。PC3 同样也无法和 PC2、PC1ping 通。

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

验证测试：端口 0/24 已被设置为 trunk 模式

```
SwitchA(config)#show interfaces gigabitEthernet 0/24 sw
SwitchA(config)#show interfaces gigabitEthernet 0/24 switchport
Interface                               Switchport Mode      Access Native Protected VLAN lists
-----
GigabitEthernet 0/24                   enabled      TRUNK    1      1      Disabled ALL
SwitchA(config)#
```

(1) 检查 PC1、PC2、PC3 此时的连通情况。

```
管理工具: C:\Windows\system32\cmd.exe
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\Administrator>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\Administrator>
```

PC1

```
管理工具: C:\Windows\system32\cmd.exe
C:\Users\Administrator>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\Administrator>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\Administrator>
```

```
C:\Windows\system32>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:\Windows\system32>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\Administrator>ping 192.168.10.10

正在 Ping 192.168.10.10 具有 32 字节的数据:
来自 192.168.10.30 的回复: 无法访问目标主机。
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 = 1, 丢失 = 3 (75% 丢失),

C:\Users\Administrator>ping 192.168.10.20

正在 Ping 192.168.10.20 具有 32 字节的数据:
来自 192.168.10.30 的回复: 无法访问目标主机。
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 = 1, 丢失 = 3 (75% 丢失),
```

PC3

开启交换机 A 的 trunk 模式，此时 PC1 处于 VLAN 10，PC2 处于 VLAN 20，而 PC3 处于 VLAN 1，三台计算机仍然处于不同的虚拟网络之中，相互之间无法 ping 通。

步骤 5: 在交换机 B 上创建 VLAN 20，并将端口 0/5 划分到 VLAN20 中。

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

```
SwitchB(config)#show vlan
VLAN Name                Status    Ports
-----
 1  VLAN0001                STATIC    Gi0/1, Gi0/2, Gi0/3, Gi0/4
                                     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
20 technical              STATIC    Gi0/5
```

(3) 检查 PC1、PC2、PC3 此时的连通情况。





```
管理员: C:\Windows\system32\cmd.exe
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\Administrator>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\Administrator>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\Administrator>

管理员: C:\Windows\system32\cmd.exe
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\Administrator>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\Administrator>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\Administrator>
```

PC1

```
C:\Windows\system32>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:\Windows\system32>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:\Windows\system32>
```

PC2

```
C:\Users\Administrator>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\Administrator>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),
```

PC3



# 计算机网络实验报告

将 SwitchB 的 0/5 端口划分进入到 VLAN 20, 此时 PC2 和 PC3 同处于 VLAN 20 的虚拟网络之中。但由于 SwitchB 并没有开启 trunk 模式, 两台交换机的同一个 ID 的 VLAN 局域网中, 但此时 VLAN 端口为 Access 端口, 只属于一个 VLAN 无法向其他 VLAN 发送信息, 无法进行跨交换机通信, 因此此时 PC2 与 PC3 仍然不能够相互 ping 通, 而 PC1 也因为不在同一虚拟局域网而无法 ping 通, 因此三台计算机互相不能通信。

步骤 6: 将交换机 B 与交换机 A 相连的端口 (假设为端口 0/24) 定义为 Tag VLAN 模式

```
SwitchB(config)#show interfaces gigabitEthernet 0/24 switchport
Interface                                Switchport Mode      Access Native Protected VLAN lists
-----
GigabitEthernet 0/24                    enabled      TRUNK    1       1       Disabled ALL
```

步骤 7: 验证 PC2 与 PC3 能互相通信, 但 PC1 与 PC3 不能互相通信。

```
管理工具: C:\Windows\system32\cmd.exe
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),
    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\Administrator>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\Administrator>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\Administrator>
    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\Administrator>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\Administrator>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\Administrator>
```

PC1

```
管理工具: 命令提示符
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:\Windows\system32>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:\Windows\system32>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=64
    来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=64
    来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=64
    来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=64
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:\Windows\system32>
```

PC2



```
C:\Users\Administrator>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=64
来自 192.168.10.20 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.10.20 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.10.20 的回复: 字节=32 时间<1ms TTL=64

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
```

PC3

```
C:\Users\Administrator>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。

192.168.10.10 的 Ping 统计信息:
    数据包: 已发送 = 3, 已接收 = 0, 丢失 = 3 (100% 丢失),
```

PC3

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) 主机之间能否相互通信?

如网络拓扑图中, PC1 单独处于 VLAN10 中, PC2 和 PC3 处于 VLAN20 中, 交换机 A 和交换机 B 在端口 0/24 开启 trunk 模式, 于是 PC2 和 PC3 可以进行跨交换机通信, PC2 和 PC3 成功互相 ping 通, 而 PC1 则因为处于不同的虚拟局域网中而被单独隔离。

(2) 能否监测到 PC1、PC2、PC3 的 ICMP 包?

801	592.741639	192.168.10.10	192.168.10.255	UDP	1482 58890 → 1689 Len=1440
802	593.041222	192.168.10.10	192.168.10.20	ICMP	74 Echo (ping) request id=0x0001, seq=90/23040, ttl=64
803	594.873885	192.168.10.10	192.168.10.30	ICMP	74 Echo (ping) request id=0x0001, seq=91/23296, ttl=64
804	597.893398	Shenzhen_0e:ab:71	Shenzhen_0e:c2:60	ARP	42 Who has 192.168.10.20? Tell 192.168.10.10
805	597.895555	192.168.10.10	192.168.10.20	ICMP	74 Echo (ping) request id=0x0001, seq=92/23552, ttl=64
806	598.892748	Shenzhen_0e:ab:71	Shenzhen_0e:c2:60	ARP	42 Who has 192.168.10.20? Tell 192.168.10.10
807	599.393206	Shenzhen_0e:ab:71	Shenzhen_0e:ab:7a	ARP	42 Who has 192.168.10.30? Tell 192.168.10.10
808	599.395291	192.168.10.10	192.168.10.30	ICMP	74 Echo (ping) request id=0x0001, seq=93/23808, ttl=64
809	599.893461	Shenzhen_0e:ab:71	Shenzhen_0e:c2:60	ARP	42 Who has 192.168.10.20? Tell 192.168.10.10
810	600.392992	Shenzhen_0e:ab:71	Shenzhen_0e:ab:7a	ARP	42 Who has 192.168.10.30? Tell 192.168.10.10

PC1

12	35.389714	192.168.10.20	192.168.10.30	ICMP	74 Echo (ping) request id=0x0001, seq=69/17664, ttl=64 (reply in 13)
13	35.389927	192.168.10.30	192.168.10.20	ICMP	74 Echo (ping) reply id=0x0001, seq=69/17664, ttl=64 (request in 12)
15	36.391777	192.168.10.20	192.168.10.30	ICMP	74 Echo (ping) request id=0x0001, seq=70/17920, ttl=64 (reply in 16)
16	36.391965	192.168.10.30	192.168.10.20	ICMP	74 Echo (ping) reply id=0x0001, seq=70/17920, ttl=64 (request in 15)
17	37.394430	192.168.10.20	192.168.10.30	ICMP	74 Echo (ping) request id=0x0001, seq=71/18176, ttl=64 (reply in 18)
18	37.394784	192.168.10.30	192.168.10.20	ICMP	74 Echo (ping) reply id=0x0001, seq=71/18176, ttl=64 (request in 17)
19	38.397084	192.168.10.20	192.168.10.30	ICMP	74 Echo (ping) request id=0x0001, seq=72/18432, ttl=64 (reply in 20)
20	38.397313	192.168.10.30	192.168.10.20	ICMP	74 Echo (ping) reply id=0x0001, seq=72/18432, ttl=64 (request in 19)
28	48.100498	192.168.10.30	192.168.10.20	ICMP	74 Echo (ping) request id=0x0001, seq=61/15616, ttl=64 (reply in 29)
29	48.100550	192.168.10.20	192.168.10.30	ICMP	74 Echo (ping) reply id=0x0001, seq=61/15616, ttl=64 (request in 28)
30	49.102218	192.168.10.30	192.168.10.20	ICMP	74 Echo (ping) request id=0x0001, seq=62/15872, ttl=64 (reply in 31)
31	49.102276	192.168.10.20	192.168.10.30	ICMP	74 Echo (ping) reply id=0x0001, seq=62/15872, ttl=64 (request in 30)
33	50.105316	192.168.10.30	192.168.10.20	ICMP	74 Echo (ping) request id=0x0001, seq=63/16128, ttl=64 (reply in 34)
34	50.105362	192.168.10.20	192.168.10.30	ICMP	74 Echo (ping) reply id=0x0001, seq=63/16128, ttl=64 (request in 33)
36	51.109186	192.168.10.30	192.168.10.20	ICMP	74 Echo (ping) request id=0x0001, seq=64/16384, ttl=64 (reply in 37)
37	51.109224	192.168.10.20	192.168.10.30	ICMP	74 Echo (ping) reply id=0x0001, seq=64/16384, ttl=64 (request in 36)

PC2

4	5.281230	192.168.10.30	192.168.10.20	ICMP	74 Echo (ping) request id=0x0001, seq=65/16640, ttl=64 (reply in 5)
5	5.281342	192.168.10.20	192.168.10.30	ICMP	74 Echo (ping) reply id=0x0001, seq=65/16640, ttl=64 (request in 4)
6	6.048330	192.168.10.20	192.168.10.255	NBNS	96 Name query NB DESKTOP-BVAQLT3<1c>
7	6.284020	192.168.10.30	192.168.10.20	ICMP	74 Echo (ping) request id=0x0001, seq=66/16896, ttl=64 (reply in 8)
8	6.284147	192.168.10.20	192.168.10.30	ICMP	74 Echo (ping) reply id=0x0001, seq=66/16896, ttl=64 (request in 7)
9	6.799254	192.168.10.20	192.168.10.255	NBNS	96 Name query NB DESKTOP-BVAQLT3<1c>
10	7.287864	192.168.10.30	192.168.10.20	ICMP	74 Echo (ping) request id=0x0001, seq=67/17152, ttl=64 (reply in 11)
11	7.288026	192.168.10.20	192.168.10.30	ICMP	74 Echo (ping) reply id=0x0001, seq=67/17152, ttl=64 (request in 10)
12	7.550172	192.168.10.20	192.168.10.255	NBNS	96 Name query NB DESKTOP-BVAQLT3<1c>
13	8.291729	192.168.10.30	192.168.10.20	ICMP	74 Echo (ping) request id=0x0001, seq=68/17408, ttl=64 (reply in 14)
14	8.291838	192.168.10.20	192.168.10.30	ICMP	74 Echo (ping) reply id=0x0001, seq=68/17408, ttl=64 (request in 13)

PC3

三台计算机的 Wireshark 都能捕捉 ICMP 数据包。但是由于 PC1 被单独隔离, PC2 与 PC3 互通, 因此 PC1 中只能捕获到由 PC1 发出的 ICMP 数据包, 而不能收到从 PC2 和 PC3 的 ICMP 数据包; 而在互通的 PC2 和 PC3 中我们能够成功捕获两台计算机相互发送与接收的 ICMP 数据包



(3) 能够捕获到 Trunk 链路上的 VLAN ID? 请讨论原因

No.	Time	Source	Destination	Protocol	Length	Info
1160	1996.840170	fe80::84c4:8b53:a11...	ff02::1:2	DHCPv6	157	Solicit XID: 0x294513 CID: 000100012723eb7880c16ee
1161	1998.841002	fe80::84c4:8b53:a11...	ff02::1:2	DHCPv6	157	Solicit XID: 0x294513 CID: 000100012723eb7880c16ee
1162	2000.399418	192.168.10.10	192.168.10.255	UDP	1482	58890 → 1689 Len=1440
1163	2002.841839	fe80::84c4:8b53:a11...	ff02::1:2	DHCPv6	157	Solicit XID: 0x294513 CID: 000100012723eb7880c16ee
1164	2008.932101	192.168.10.10	192.168.10.255	UDP	1482	58890 → 1689 Len=1440
1165	2010.842817	fe80::84c4:8b53:a11...	ff02::1:2	DHCPv6	157	Solicit XID: 0x294513 CID: 000100012723eb7880c16ee
1166	2017.464657	192.168.10.10	192.168.10.255	UDP	1482	58890 → 1689 Len=1440
1167	2018.560584	RuijieMe_15:55:12	LLDP_Multicast	LLDP	388	MA/58:69:6c:15:55:12 121 Sys
1168	2025.990834	192.168.10.10	192.168.10.255	UDP	1482	58890 → 1689 Len=1440
1169	2026.843399	fe80::84c4:8b53:a11...	ff02::1:2	DHCPv6	157	Solicit XID: 0x294513 CID: 000100012723eb7880c16ee
1170	2034.520616	192.168.10.10	192.168.10.255	UDP	1482	58890 → 1689 Len=1440
1171	2043.055813	192.168.10.10	192.168.10.255	UDP	1482	58890 → 1689 Len=1440
1172	2048.561394	RuijieMe_15:55:12	LLDP_Multicast	LLDP	388	MA/58:69:6c:15:55:12 121 Sys
1173	2051.588210	192.168.10.10	192.168.10.255	UDP	1482	58890 → 1689 Len=1440

1111 111. .... = TLV Type: Organization Specific (127)
.... 0000 0110 = TLV Length: 6
Organization Unique Code: 00:80:c2 (IEEE)
IEEE 802.1 Subtype: Port VLAN ID (0x01)
Port VLAN Identifier: 10 (0x000a)
IEEE - Port and Protocol VLAN ID
1111 111. .... = TLV Type: Organization Specific (127)
.... 0000 0111 = TLV Length: 7
Organization Unique Code: 00:80:c2 (IEEE)
IEEE 802.1 Subtype: Port and Protocol VLAN ID (0x02)

No.	Time	Source	Destination	Protocol	Length	Info
1167	2018.560584	RuijieMe_15:55:12	LLDP_Multicast	LLDP	388	MA/58:69:6c:15:55:12 121 Sys
1168	2025.990834	192.168.10.10	192.168.10.255	UDP	1482	58890 → 1689 Len=1440
1169	2026.843399	fe80::84c4:8b53:a11...	ff02::1:2	DHCPv6	157	Solicit XID: 0x294513 CID: 000100012723eb7880c16ee
1170	2034.520616	192.168.10.10	192.168.10.255	UDP	1482	58890 → 1689 Len=1440
1171	2043.055813	192.168.10.10	192.168.10.255	UDP	1482	58890 → 1689 Len=1440
1172	2048.561394	RuijieMe_15:55:12	LLDP_Multicast	LLDP	388	MA/58:69:6c:15:55:12 121 Sys
1173	2051.588210	192.168.10.10	192.168.10.255	UDP	1482	58890 → 1689 Len=1440

1111 111. .... = TLV Type: Organization Specific (127)
.... 0000 0111 = TLV Length: 7
Organization Unique Code: 00:80:c2 (IEEE)
IEEE 802.1 Subtype: Port and Protocol VLAN ID (0x02)

从 ping 命令所使用的 ICMP 协议数据包中并没有找到 VLAN ID。因为 ICMP 是网络层的协议，而 VLAN 协议是属于数据链路层的协议，比网络层低一层。但从处于数据链路层的 LLDP 协议中，可以找到 VLAN ID

(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 命令现实的 MAC 地址与在命令提示符下通过 ipconfig/all 命令显示的 MAC 地址是否相同？

VLAN Name	Status	Ports
1 VLAN0001	STATIC	Gi0/1, Gi0/2, Gi0/3, Gi0/4 Gi0/6, Gi0/7, Gi0/8, Gi0/9 Gi0/10, Gi0/11, Gi0/12, Gi0/13 Gi0/14, 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
10 sales	STATIC	Gi0/5
20 technical	STATIC	Gi0/15

```
SwitchA(config)#show mac-ad
SwitchA(config)#show mac-address-t
SwitchA(config)#show mac-address-table
Vlan      MAC Address      Type      Interface
-----
1         4433.4c0e.ab7a   DYNAMIC  GigabitEthernet 0/24
1         5869.6c15.5518   DYNAMIC  GigabitEthernet 0/24
10        4433.4c0e.ab71   DYNAMIC  GigabitEthernet 0/5
20        4433.4c0e.ab7a   DYNAMIC  GigabitEthernet 0/24
20        4433.4c0e.c260   DYNAMIC  GigabitEthernet 0/15
```

以太网通配器 实验网 2:	Realtek Common Ethernet Controllers
物理地址:	44-33-4C-0E-AB-71
DHCP 已启用:	是
自动配置已启用:	是
本地地址 IPv6 地址:	fe80::84c4:8b53:a11b:6d56(首选)
子网掩码:	192.168.10.10(首选)
默认网关:	255.255.255.0
DHCPv6 IAID:	105132876
DHCPv6 客户端 DUID:	00-01-00-01-27-23-E8-78-80-C1-6E-E3-CA-42
DNS 服务器:	fec0:0:0:ffff::1
TCPIP 上的 NetBIOS:	已启用

无线局域网适配器 WLAN:	媒体状态: 媒体已断开连接
连接特定的 DNS 后缀:	连接特定的 DNS 后缀
物理地址:	Realtek RT61 Turbo Wireless LAN Card
DHCP 已启用:	是
自动配置已启用:	是
本地地址 IPv6 地址:	00-0D-0A-4B-0B-46
子网掩码:	是
默认网关:	是

以太网通配器 校园网:	Realtek PCIe GBE Family Controller
物理地址:	18-60-24-88-AF-F7
DHCP 已启用:	是
自动配置已启用:	是
本地地址 IPv6 地址:	2001:250:3002:4b98:7119:4a8:e5e9:7a0d(首选)
子网掩码:	2001:250:3002:4b98:3bf1:4607:4c5e:e894(首选)
默认网关:	fe80::7119:4a8:e5e9:7a0d(首选)
本地地址 IPv6 地址:	172.16.11.3(首选)
子网掩码:	255.255.0.0
默认网关:	255.255.0.0





```
连接特定的 DNS 后缀 . . . . . : 
描述 . . . . . : Realtek Common Ethernet Controllers
物理地址 . . . . . : 44-33-4C-0E-C2-60
DHCP 已启用 . . . . . : 否
自动配置已启用 . . . . . : 是
本地连接 IPv6 地址 . . . . . : fe80::a06d:7e12:78be:9d17%6(首选)
IPv4 地址 . . . . . : 192.168.10.20(首选)
子网掩码 . . . . . : 255.255.255.0
默认网关 . . . . . : 
DHCPv6 IALD . . . . . : 105132876
DHCPv6 客户端 DUID . . . . . : 00-01-00-01-27-23-EB-78-80-C1-6E-E3-CA-42
DNS 服务器 . . . . . : fec0:0:0:ffff::1%1
                          fec0:0:0:ffff::2%1
                          fec0:0:0:ffff::3%1
TCP/IP 上的 NetBIOS . . . . . : 已启用
```

交换机 A 进行地址学习，记录连接到交换机的各个机器的 MAC 地址，show mac-address-table 命令现实的 MAC 地址与在命令提示符下通过 ipconfig/all 命令显示的 MAC 地址是相同的。

809	599.893461	Shenzhen_0e:ab:71	Shenzhen_0e:c2:60	ARP	42 Who has 192.168.10.20? Tell 192.168.10.10
810	600.392992	Shenzhen_0e:ab:71	Shenzhen_0e:ab:7a	ARP	42 Who has 192.168.10.30? Tell 192.168.10.10
811	601.268944	192.168.10.10	192.168.10.255	UDP	1482 58890 → 1689 Len=1440
812	601.393188	Shenzhen_0e:ab:71	Shenzhen_0e:ab:7a	ARP	42 Who has 192.168.10.30? Tell 192.168.10.10
813	602.895603	Shenzhen_0e:ab:71	Broadcast	ARP	42 Who has 192.168.10.20? Tell 192.168.10.10
814	603.893190	Shenzhen_0e:ab:71	Broadcast	ARP	42 Who has 192.168.10.20? Tell 192.168.10.10

如果当交换机不能正确学习 MAC 地址，则会导致数据包丢失及泛洪现象。交换机会向接收端口外的所有端口广播该数据帧。

(5) 判断实验是否达到预期目标。

实验达到了预期目标：

1. 在没有划分 VLAN 之前，三台计算机之间是互通的
2. 在划分 VLAN 之后如果没有开启 trunk 模式三台计算机将相互隔离
3. 开启 trunk 模式之后 PC1 单独隔离，PC2 和 PC3 处于同一个 VLAN，但可以进行跨交换机通信

【实验思考】

(1) 实验时，要注意两台交换机之间相连的端口应该设置为 Tag VLAN 模式。配置时要注意区别每个操作模式下可执行的命令种类。交换机不可以跨模式执行命令，返回上级模式一般用 exit 命令。交换机端口在默认情况下是开启的（up 表示开启状态，down 表示关闭状态）。一般配置好 IP 地址后要用 no shutdown 开启端口，这样才能使物理设备端口正常通信。

(2) 为什么不同的 VLAN 之间不能直接互相通信？

VLAN（虚拟局域网，Virtual Local Area Network）是一种通过将局域网内的设备逻辑地划分成一个一个网段，从而实现虚拟工作组的技术。VLAN 是为了解决以太网的广播问题 and 安全性而提出的一种协议，它在以太网帧的基础上增加了 VLAN 首部，用 VLAN ID 把用户划分为更小的工作组，限制不同工作组间的二层互访，每个工作组就形成一个虚拟局域网。各域中的广播帧只在各自的域中广播，互不干扰。

(3) 说明 VLAN 技术中的 Trunk 模式端口的用途和特点。

Trunk 端口通常用于交换机之间（或者交换机和其他网络设备之间）的连接，以保证在跨越多台交换机上建立的同一个 VLAN 的成员能够相互通信。其中交换机之间互连用的端口即为 Trunk 端口，Trunk 端口同时可以承载带 VLAN 和不带 VLAN 的报文。交换机的 Trunk 端口不属于某个 VLAN，而是可以承载所有 VLAN 的帧。

Trunk 端口会转发交换机上存在的所有 VLAN 数据，传输多个 VLAN 信息，实现同一 VLAN 跨越不同的交换机。

(4) 如何查看 Trunk 端口允许哪些 VLAN 通过？

Trunk 端口默认可以传输本交换机支持的所有 VLAN，也可以通过设置端口的许可 VLAN 列表限制某些 VLAN 的流量不能通过此 Trunk 端口，其中命令为：switchport trunk allowed vlan {all | [add



| remove | except]]} vlan-list。

vlan-list 可以是一个 VLAN，也可以是一系列 VLAN，以小的 VLAN ID 开头，大的 VLAN ID 结尾，用-连接。

(5) 实验前要先确定 3 台主机处于同一个网段内，为什么要这样限定？

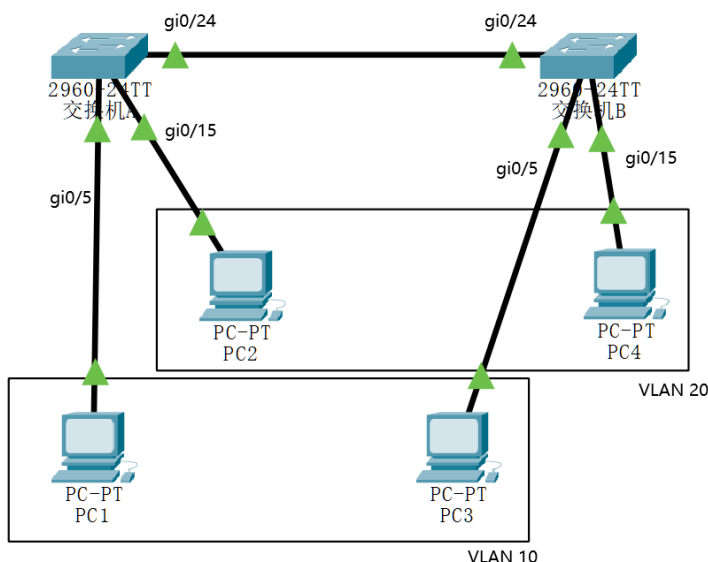
因为 VLAN 是通过将局域网内的设备逻辑划分成一个个虚拟局域网的技术，所以实验前需要确保 3 台主机本是同一个局域网内的设备，也即同一个网段内。

二、本章习题练习 9，并用 Wireshark 截图，分析实验结果

【9】假设某企业的网络中，计算机 PC1 和 PC3 属于营销部门，PC2 和 PC4 属于技术部门，PC1 和 PC2 连接在交换机 A 上，PC3 和 PC4 连接在交换机 B 上，而 2 个部门要求互相隔离。本实验的目的是实现跨 2 台交换机将不同端口划分到不同的 VLAN。

【要求】

(1) 画出拓扑图，并标明 VLAN 以及相关端口。



(2) 在实验设备上完成“跨交换机实现 VLAN”实验并测试实验网连通性

正在 Ping 192.168.10.40 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):

最小 = 0ms, 最长 = 0ms, 平均 = 0ms

C:\Windows\system32>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。

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:\Windows\system32>ping 192.168.10.40

正在 Ping 192.168.10.40 具有 32 字节的数据:

来自 192.168.10.40 的回复: 字节=32 时间<1ms TTL=64

来自 192.168.10.40 的回复: 字节=32 时间<1ms TTL=64

来自 192.168.10.40 的回复: 字节=32 时间<1ms TTL=64

来自 192.168.10.40 的回复: 字节=32 时间<1ms TTL=64

192.168.10.40 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):

最小 = 0ms, 最长 = 0ms, 平均 = 0ms

C:\Windows\system32>

正在捕获 实验网

文件(F) 编辑(E) 视图(V) 跟踪(T) 捕获(C) 分析(A) 统计(S) 电话(W) 无线(W) 工具(T) 帮助(H)

应用显示过滤器: <Ctrl>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.800000	192.168.10.40	192.168.10.255	UDP	1482	55692 → 1689 Len=1440
2	2.741320	192.168.10.20	192.168.10.40	ICMP	74	Echo (ping) request id=0x0001
3	2.741655	192.168.10.40	192.168.10.20	ICMP	74	Echo (ping) reply id=0x0001
4	3.743361	192.168.10.20	192.168.10.40	ICMP	74	Echo (ping) request id=0x0001
5	3.743749	192.168.10.40	192.168.10.20	ICMP	74	Echo (ping) reply id=0x0001
6	4.746123	192.168.10.20	192.168.10.40	ICMP	74	Echo (ping) request id=0x0001
7	4.746473	192.168.10.20	192.168.10.40	ICMP	74	Echo (ping) reply id=0x0001
8	5.750908	192.168.10.20	192.168.10.40	ICMP	74	Echo (ping) request id=0x0001
9	5.750904	192.168.10.40	192.168.10.20	ICMP	74	Echo (ping) reply id=0x0001
10	7.246129	Shenzhen_0e:b7:06	00:88:99:00:13:4a	ARP	60	Who has 192.168.10.20? Tell 19;
11	7.246136	00:88:99:00:13:4a	Shenzhen_0e:b7:06	ARP	42	192.168.10.20 is at 00:88:99:0
12	7.594146	00:88:99:00:13:4a	Shenzhen_0e:b7:06	ARP	42	Who has 192.168.10.40? Tell 19;

Frame 1: 1482 bytes on wire (11856 bits), 1482 bytes captured (11856 bits) on interface \Device\NPF\_{EBC28BE2...}

Ethernet II, Src: Shenzhen\_0e:b7:06 (44:33:4c:0e:b7:06), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

Internet Protocol Version 4, Src: 192.168.10.40, Dst: 192.168.10.255

User Datagram Protocol, Src Port: 55692, Dst Port: 1689

Data (1440 bytes)

PC2与PC4进行  
同一VLAN下跨交换机通信



PC1与PC3进行  
同一VLAN下跨  
交换机通信

(3) PC1 ping PC3, PC2 ping PC4, 在交换机 A 的端口抓包查看报文。捕获到的报文有 VLAN ID 吗？如果没有，讨论能捕获到的方法。

当 PC1 ping PC3、PC2 ping PC4 时，用 wireshark 捕获报文，发现其使用的是 ICMP 协议，其中并不会会有 VLAN ID。如果需要捕获 VLAN ID 需要使用 LLDP 链路层发现协议。

No.	Time	Source	Destination	Protocol	Length	Info
1160	1996.840170	fe80::84c4:8b53:a11...	ff02::1:2	DHCPv6	157	Solicit XID: 0x294513 CID: 000100012723eb7880c16ee
1161	1998.841002	fe80::84c4:8b53:a11...	ff02::1:2	DHCPv6	157	Solicit XID: 0x294513 CID: 000100012723eb7880c16ee
1162	2000.399418	192.168.10.10	192.168.10.255	UDP	1482	58890 → 1689 Len=1440
1163	2002.841839	fe80::84c4:8b53:a11...	ff02::1:2	DHCPv6	157	Solicit XID: 0x294513 CID: 000100012723eb7880c16ee
1164	2008.932101	192.168.10.10	192.168.10.255	UDP	1482	58890 → 1689 Len=1440
1165	2010.842817	fe80::84c4:8b53:a11...	ff02::1:2	DHCPv6	157	Solicit XID: 0x294513 CID: 000100012723eb7880c16ee
1166	2017.464657	192.168.10.10	192.168.10.255	UDP	1482	58890 → 1689 Len=1440
1167	2018.560584	RuijieNe_15:55:12	LLDP_Multicast	LLDP	388	MA/58:69:6c:15:55:12 MA/58:69:6c:15:55:12 121 SysN
1168	2025.990834	192.168.10.10	192.168.10.255	UDP	1482	58890 → 1689 Len=1440
1169	2026.843399	fe80::84c4:8b53:a11...	ff02::1:2	DHCPv6	157	Solicit XID: 0x294513 CID: 000100012723eb7880c16ee
1170	2034.520616	192.168.10.10	192.168.10.255	UDP	1482	58890 → 1689 Len=1440
1171	2043.055813	192.168.10.10	192.168.10.255	UDP	1482	58890 → 1689 Len=1440
1172	2048.561394	RuijieNe_15:55:12	LLDP_Multicast	LLDP	388	MA/58:69:6c:15:55:12 MA/58:69:6c:15:55:12 121 SysN
1173	2051.588210	192.168.10.10	192.168.10.255	UDP	1482	58890 → 1689 Len=1440

<

>

1111 111. .... = TLV Type: Organization Specific (127)  
.... 0000 0110 = TLV Length: 6  
Organization Unique Code: 00:80:c2 (IEEE)  
IEEE 802.1 Subtype: Port VLAN ID (0x01)  
Port VLAN Identifier: 10 (0x000a)  
IEEE - Port and Protocol VLAN ID  
1111 111. .... = TLV Type: Organization Specific (127)  
.... 0000 0111 = TLV Length: 7  
Organization Unique Code: 00:80:c2 (IEEE)  
IEEE 802.1 Subtype: Port and Protocol VLAN ID (0x02)

三、跨交换机实现 VLAN 通信时，思考不用 Trunk 模式且也能进行跨交换机 VLAN 通信的替代方法，并进行实验验证。





```
SwitchA(config-if-GigabitEthernet 0/24)#sw
SwitchA(config-if-GigabitEthernet 0/24)#switchport ac
SwitchA(config-if-GigabitEthernet 0/24)#switchport mode hy
SwitchA(config-if-GigabitEthernet 0/24)#switchport mode hybrid
SwitchA(config-if-GigabitEthernet 0/24)#exit
SwitchA(config)#show int
SwitchA(config)#show interfaces gig
SwitchA(config)#show interfaces gigabitEthernet 0/24 sw
SwitchA(config)#show interfaces gigabitEthernet 0/24 switchport
Interface          Switchport Mode      Access Native Protected VLAN lists
-----
GigabitEthernet 0/24  enabled  HYBRID    1        1        Disabled ALL
SwitchA(config)#interface gigabitEthernet 0/24
SwitchA(config-if-GigabitEthernet 0/24)#sw
SwitchA(config-if-GigabitEthernet 0/24)#
SwitchA(config-if-GigabitEthernet 0/24)#sw
SwitchA(config-if-GigabitEthernet 0/24)#switchport mode t
SwitchA(config-if-GigabitEthernet 0/24)#switchport mode trunk
SwitchA(config-if-GigabitEthernet 0/24)#exit
SwitchA(config)#show interfaces gigabitEthernet 0/24 switchport
Interface          Switchport Mode      Access Native Protected VLAN lists
-----
GigabitEthernet 0/24  enabled  TRUNK    1        1        Disabled ALL
SwitchA(config)#in
SwitchA(config)#interface giga
SwitchA(config)#interface gigabitEthernet 0/24
SwitchA(config-if-GigabitEthernet 0/24)#sw
SwitchA(config-if-GigabitEthernet 0/24)#switchport mode
SwitchA(config-if-GigabitEthernet 0/24)#switchport mode hy
SwitchA(config-if-GigabitEthernet 0/24)#switchport mode hybrid
SwitchA(config-if-GigabitEthernet 0/24)#exit
SwitchA(config)#show inte
SwitchA(config)#show interfaces gig
SwitchA(config)#show interfaces gigabitEthernet 0/24 sw
SwitchA(config)#show interfaces gigabitEthernet 0/24 switchport
Interface          Switchport Mode      Access Native Protected VLAN lists
-----
GigabitEthernet 0/24  enabled  HYBRID    1        1        Disabled ALL
SwitchA(config)#
```

```
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),
    请求超时。
C:\Users\Administrator>
C:\Users\Administrator>
C:\Users\Administrator>
C:\Users\Administrator>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=64
来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.10.30 的回复: 字节=32 时间<1ms TTL=64
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\Administrator>ping 192.168.10.40
正在 Ping 192.168.10.40 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。
192.168.10.40 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 1, 丢失 = 3 (75% 丢失),
    请求超时。
C:\Users\Administrator>
```

将端口 0/24 改成 Hybrid 即混合模式下，也可以实现同一 VLAN 下跨交换机之间的通信。

### 【交实验报告】

上传实验报告: <ftp://172.18.178.1/> 截止日期 (不迟于): 1 周之内

上传包括两个文件:

(1) 小组实验报告。上传文件名格式: 小组号\_Ftp 协议分析实验.pdf (由组长负责上传)

例如: 文件名 “10\_Ftp 协议分析实验.pdf” 表示第 10 组的 Ftp 协议分析实验报告

(2) 小组成员实验体会。每个同学单独交一份只填写了实验体会的实验报告。只需填写自己的学号和姓名。

文件名格式: 小组号\_学号\_姓名\_Ftp 协议分析实验.pdf (由组员自行上传)

例如: 文件名 “10\_05373092\_张三\_Ftp 协议分析实验.pdf” 表示第 10 组的 Ftp 协议分析实验报告。

学号	学生	自评分
18338072	冼子婷	98
18322043	廖雨轩	98
18346019	胡文浩	98