

# 计算机网络课程设计

学 院： 计算机科学与工程学院

专 业： 计算机科学与技术

科 目： 计算机网络

班 级： 计算机 191

姓 名： 吴世杰

指导教师： 李军怀

西安理工大学

2021 年 秋 学期

# 目录

<b>I 单交换机实现基于端口的 VLAN 设计</b>	<b>1</b>
一、设计目的 .....	1
二、设计原理 .....	1
三、设计内容 (方案及拓扑结构) .....	1
四、实验步骤 .....	2
4.1 配置主机 .....	2
4.1.1 配置 IP 地址和端口 .....	2
4.1.2 使用 PING 测试网络 .....	2
4.2 配置交换机 .....	5
4.2.1 创建 VLAN .....	5
4.2.2 划分 VLAN .....	5
4.3 测试 VLAN2 .....	6
4.3.1 测试 VLAN2 主机间的通讯 .....	6
4.3.2 测试 VLAN2 与 VLAN3 的通讯 .....	7
4.4 VLAN 广播 .....	7
五、实验结论 .....	8
 <b>II 跨交换机实现 VLAN 设计</b>	 <b>9</b>
一、设计目的 .....	9
二、设计原理 .....	9
三、设计内容 .....	10
四、实验步骤 .....	11
4.1 配置第二台交换机与主机 .....	11
4.1.1 测试网络是否通畅 .....	11
4.1.2 将交换机端口划分 VLAN .....	12
4.2 端口 Fa0/7 为 Access 时 .....	13
4.2.1 测试 VLAN2 中主机的通讯 .....	13
4.2.2 测试 VLAN3 中主机的通讯 .....	14
4.2.3 测试 VLAN2 与 VLAN3 间主机的通讯 .....	14

4.2.4 观察 VLAN 中的广播 . . . . .	15
4.3 端口 Fa0/7 为 Trunk 时 . . . . .	16
4.3.1 测试 VLAN2 中主机的通讯 . . . . .	16
4.3.2 测试 VLAN3 中主机的通讯 . . . . .	16
4.3.3 测试 VLAN2 与 VLAN3 间主机的通讯 . . . . .	17
4.3.4 观察 VLAN 中的广播 . . . . .	18
五、实验结论 . . . . .	19
<b>III 静态路由设计</b>	<b>20</b>
一、设计目的 . . . . .	20
二、设计原理 . . . . .	20
三、设计内容 . . . . .	21
四、实验步骤 . . . . .	22
4.1 配置路由器和主机 . . . . .	22
4.2 测试子网内主机的通讯 . . . . .	23
4.3 测试不同子网内主机的通讯 . . . . .	24
4.3.1 测试子网 1 与子网 2 的通讯 . . . . .	24
4.3.2 测试子网 1 与子网 4 的通讯 . . . . .	25
4.4 配置静态路由 . . . . .	26
五、实验总结 . . . . .	28
<b>IV 动态路由 (RIP 协议) 设计</b>	<b>29</b>
一、设计目的 . . . . .	29
二、设计原理 . . . . .	29
三、设计内容 . . . . .	30
四、实验步骤 . . . . .	31
4.1 配置主机 . . . . .	31
4.2 配置路由器 . . . . .	31
4.3 测试主机间通信 . . . . .	32
4.4 配置 RIP . . . . .	32

4.5 观察 RIP 协议运行过程 . . . . .	33
4.6 测试主机间的通信 . . . . .	34
五、实验结论 . . . . .	36
<b>V 动态路由 (OSPF 协议) 设计</b>	<b>37</b>
一、设计目的 . . . . .	37
二、设计原理 . . . . .	37
三、设计内容 . . . . .	37
四、实验步骤 . . . . .	38
4.1 配置主机 . . . . .	38
4.2 配置路由器 . . . . .	38
4.3 测试主机间通信 . . . . .	39
4.4 启动 OSPF 协议 . . . . .	40
4.5 测试主机间的通信 . . . . .	40
4.6 观察通信过程 . . . . .	42
五、实验总结 . . . . .	42
<b>VI DNS 服务器设计</b>	<b>43</b>
一、设计目的 . . . . .	43
二、设计原理 . . . . .	43
三、设计内容 . . . . .	43
四、实验步骤 . . . . .	44
4.1 配置主机 . . . . .	44
4.2 配置 DNS 服务器 . . . . .	45
4.3 测试配置是否成功 . . . . .	45
<b>VII WWW 服务器设计</b>	<b>47</b>
一、设计目的 . . . . .	47
二、设计原理 . . . . .	47

三、设计内容 .....	47
四、实验步骤 .....	48
4.1 配置 WEB 服务器 .....	48
4.2 测试配置是否成功 .....	49
 <b>VIII FTP 服务器设计</b>	<b>50</b>
一、设计目的 .....	50
二、设计原理 .....	50
三、设计内容 .....	50
四、实验步骤 .....	51
4.1 配置 FTP 服务器 .....	51
4.2 测试配置是否成功 .....	51
五、实验总结 .....	52
 <b>IX 综合实验</b>	<b>53</b>
一、设计任务 .....	53
二、网络拓扑设计 .....	54
2.1 配置主机 .....	54
2.2 配置多层交换机 .....	55
三、使用 OSPF 使主机间 ping 通 .....	56
3.1 配置网关静态路由 .....	56
3.2 配置 OSPF 协议 .....	57
3.3 尝试主机间的通信 .....	57
3.3.1 工厂内通信 .....	57
3.3.2 与家属区的通信 .....	58
四、构建服务器 .....	60
4.1 WWW 服务器配置 .....	60
4.2 DNS 服务器的配置 .....	61
4.3 构建 FTP 服务器 .....	62

4.4 尝试通过域名访问 . . . . .	63
<b>五、访问控制 FTP . . . . .</b>	<b>64</b>
5.1 配置 ACL . . . . .	64
5.2 测试 ACL 配置是否成功 . . . . .	64
<b>六、路由器 NAT 配置 . . . . .</b>	<b>65</b>
6.1 NAT 简介 . . . . .	65
6.2 配置路由器 . . . . .	65
6.3 NAT 过程分析 . . . . .	66
<b>七、实验总结 . . . . .</b>	<b>66</b>

## Part I

# 单交换机实现基于端口的 VLAN 设计

## 一、设计目的

虚拟局域网（VLAN——Virtual Local Area Network）是指在交换局域网的基础上，采用网络管理软件构建的可跨越不同网段、不同网络的端到端的逻辑网络。一个 VLAN 允许处于不同地理位置的网络用户加入同一个逻辑子网中，共享一个广播域，而不同 VLAN 之间广播信息是相互隔离的。通过对 VLAN 的创建可以控制广播风暴的产生，从而提高交换式网络的整体性能和安全性。

## 二、设计原理

VLAN 划分的方法主要有基于端口划分、基于协议划分和基于 MAC 地址划分。

1. 基于端口划分（Port Based VLAN）是把一个或多个交换机上的几个端口划分一个逻辑组，这是最简单、最有效的划分方法。该方法只需网络管理员对网络设备的交换端口进行重新分配即可，不用考虑该端口所连接的设备。
2. 基于协议划分（Protocol Based VLAN）是指使用相同网络协议（即基于 IP 和 IPX 协议的转发）的设备将处于同一个 VLAN，该方式允许一个 VLAN 跨越多个交换机，或一个端口位于多个 VLAN 中。
3. □ 基于 MAC 地址划分（MAC Based VLAN）即按网卡的标识符（唯一的）把一些站点划分为一个逻辑子网，这样 PC 机无论从哪个交换机端口接入网络，均不会改变它所处的 VLAN。

本实验中使用的是基于端口划分 VLAN 的方法。

## 三、设计内容 (方案及拓扑结构)

设置 6 台 PC 机依次编号为 PC0~PC5，依次配置 IP 地址从 192.168.0.1~192.168.0.6，依次 PING 通每一台主机，保证网络畅通。

设置一台交换机，将 PC0~PC5 机连接在交换机 0 上，将交换机基于端口划分为两个 VLAN，端口 Fa0/1-3 划分为 VLAN2，端口 Fa0/4-6 划分为 VLAN3，PC0~PC2 依次连接到端口 Fa0/1-3，PC3~PC5 依次连接到端口 Fa0/4-6，即把 PC0~PC2 划分到 VLAN2，将 PC3~PC5 划分到 VLAN3。

采用 show vlan 命令查看 VLAN 的配置，确认配置无误后

1. 在 VLAN2 中选择一台主机去 ping 通 VLAN2 中的另一台主机，查看是否能通讯？

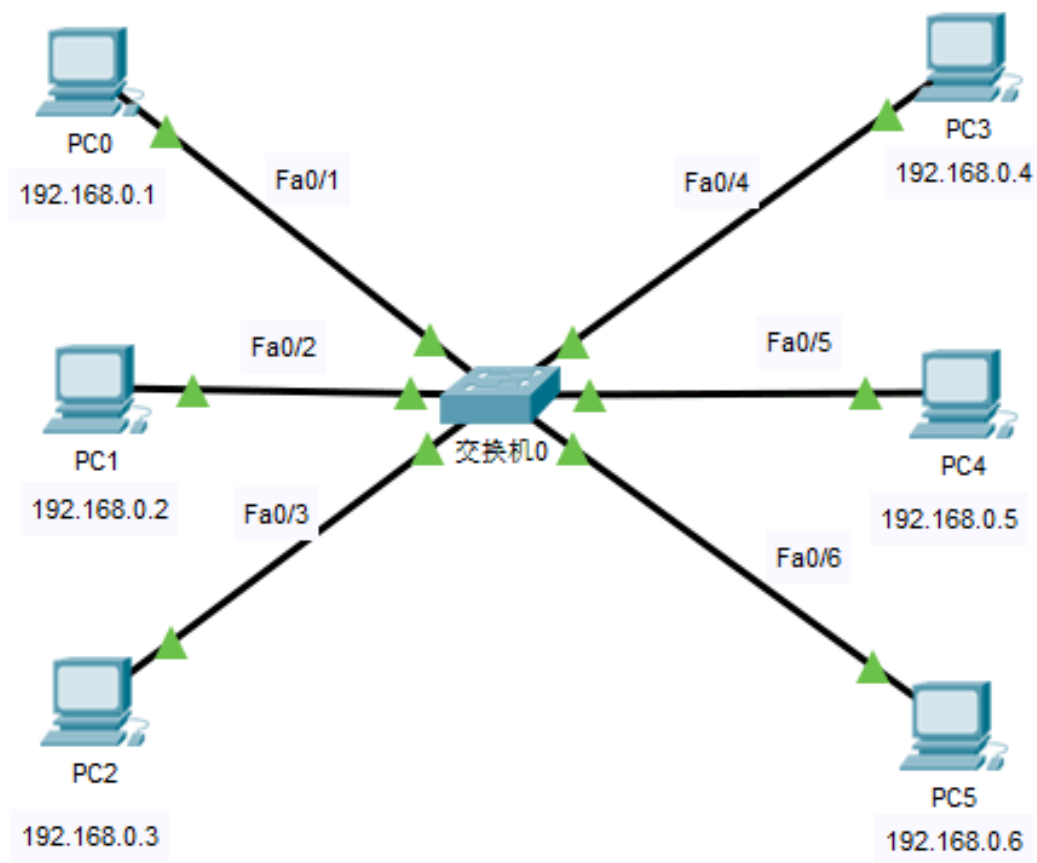


图 1 拓扑结构

2. 在 VLAN3 中选择一台主机去 ping 通 VLAN3 中的另一台主机, 查看是否能通讯?
3. 在 VLAN2 中选择一台主机去 ping 通 VLAN3 中的另一台主机, 查看是否能通讯?

最后通过创建一个广播请求来测试 VLAN 内的广播.

## 四、实验步骤

### 4.1 配置主机

#### 4.1.1 配置 IP 地址和端口

采用星型拓扑结构, 对于 PC 地址的配置和端口的配置如图 (1) 所示, 直通线上绿色箭头代表可以正常通信.

配置 PC 的 IP 地址点击桌面选择 IP 配置, 会出现图 (2) 的配置界面进行配置.

可以看到端口 FastEthernet0/1~FastEthernet0/6 的链接已经开启表示配置成功可以正常使用.

#### 4.1.2 使用 PING 测试网络



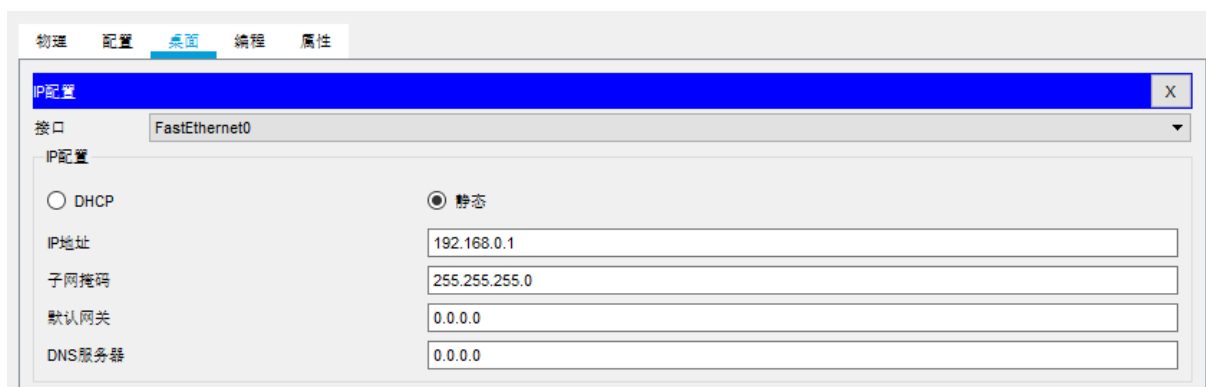


图 2 IP 地址配置

端口	链接	VLAN	IP地址	MAC地址
FastEthernet0/1	开启	1	--	0050.0F57.D701
FastEthernet0/2	开启	1	--	0050.0F57.D702
FastEthernet0/3	开启	1	--	0050.0F57.D703
FastEthernet0/4	开启	1	--	0050.0F57.D704
FastEthernet0/5	开启	1	--	0050.0F57.D705
FastEthernet0/6	开启	1	--	0050.0F57.D706
FastEthernet0/7	关闭	1	--	0050.0F57.D707
FastEthernet0/8	关闭	1	--	0050.0F57.D708
FastEthernet0/9	关闭	1	--	0050.0F57.D709
FastEthernet0/10	关闭	1	--	0050.0F57.D70A
FastEthernet0/11	关闭	1	--	0050.0F57.D70B
FastEthernet0/12	关闭	1	--	0050.0F57.D70C
FastEthernet0/13	关闭	1	--	0050.0F57.D70D
FastEthernet0/14	关闭	1	--	0050.0F57.D70E
FastEthernet0/15	关闭	1	--	0050.0F57.D70F
FastEthernet0/16	关闭	1	--	0050.0F57.D710
FastEthernet0/17	关闭	1	--	0050.0F57.D711
FastEthernet0/18	关闭	1	--	0050.0F57.D712
FastEthernet0/19	关闭	1	--	0050.0F57.D713
FastEthernet0/20	关闭	1	--	0050.0F57.D714
FastEthernet0/21	关闭	1	--	0050.0F57.D715
FastEthernet0/22	关闭	1	--	0050.0F57.D716
FastEthernet0/23	关闭	1	--	0050.0F57.D717
FastEthernet0/24	关闭	1	--	0050.0F57.D718
GigabitEthernet0/1	关闭	1	--	0050.0F57.D719
GigabitEthernet0/2	关闭	1	--	0050.0F57.D71A
Vlan1	关闭	1	<未设置>	00E0.A357.5E8C
主机名: Switch				
物理位置: 城际, 家园城市, 企业办公室, 主布线室, 机架				

图 3 交换机端口使用情况

```
1      Packet Tracer PC Command Line 1.0
2      C:\>ping 192.168.0.4
3
4      Pinging 192.168.0.4 with 32 bytes of data:
5
6      Reply from 192.168.0.4: bytes=32 time=1ms TTL=128
7      Reply from 192.168.0.4: bytes=32 time<1ms TTL=128
8      Reply from 192.168.0.4: bytes=32 time=1ms TTL=128
9      Reply from 192.168.0.4: bytes=32 time=1ms TTL=128
10
11     Ping statistics for 192.168.0.4:
12         Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
13     Approximate round trip times in milli-seconds:
14         Minimum = 0ms, Maximum = 1ms, Average = 0ms
15
16     C:\>ping 192.168.0.5
17
18     Pinging 192.168.0.5 with 32 bytes of data:
19
20     Reply from 192.168.0.5: bytes=32 time=1ms TTL=128
21     Reply from 192.168.0.5: bytes=32 time=1ms TTL=128
22     Reply from 192.168.0.5: bytes=32 time=1ms TTL=128
23     Reply from 192.168.0.5: bytes=32 time<1ms TTL=128
24
25     Ping statistics for 192.168.0.5:
26         Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
27     Approximate round trip times in milli-seconds:
28         Minimum = 0ms, Maximum = 1ms, Average = 0ms
29
30     C:\>ping 192.168.0.6
31
32     Pinging 192.168.0.6 with 32 bytes of data:
33
34     Reply from 192.168.0.6: bytes=32 time=1ms TTL=128
35     Reply from 192.168.0.6: bytes=32 time<1ms TTL=128
36     Reply from 192.168.0.6: bytes=32 time=1ms TTL=128
37     Reply from 192.168.0.6: bytes=32 time=4ms TTL=128
38
39     Ping statistics for 192.168.0.6:
40         Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
41     Approximate round trip times in milli-seconds:
42         Minimum = 0ms, Maximum = 4ms, Average = 1ms
43
44     C:\>ping 192.168.0.2
45
46     Pinging 192.168.0.2 with 32 bytes of data:
47
48     Reply from 192.168.0.2: bytes=32 time=1ms TTL=128
49     Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
50     Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
51     Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
52
53     Ping statistics for 192.168.0.2:
54         Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
55     Approximate round trip times in milli-seconds:
```

```

56         Minimum = 0ms, Maximum = 1ms, Average = 0ms
57
58     C:\>ping 192.168.0.3
59
60     Pinging 192.168.0.3 with 32 bytes of data:
61
62     Reply from 192.168.0.3: bytes=32 time=1ms TTL=128
63     Reply from 192.168.0.3: bytes=32 time<1ms TTL=128
64     Reply from 192.168.0.3: bytes=32 time<1ms TTL=128
65     Reply from 192.168.0.3: bytes=32 time<1ms TTL=128
66
67     Ping statistics for 192.168.0.3:
68         Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
69     Approximate round trip times in milli-seconds:
70         Minimum = 0ms, Maximum = 1ms, Average = 0ms

```

可以看到 ping 命令全都成功执行, 然后将 PC1~PC5 去 ping 其他主机进行测试, 结果为全部 ping 通.

## 4.2 配置交换机

### 4.2.1 创建 VLAN

```

1     Switch>en
2     Switch#conf t
3     Enter configuration commands, one per line.  End with CNTL/Z.
4
5     Switch(config)#vlan 2
6     Switch(config-vlan)#name VLAN2
7     Switch(config-vlan)#exit
8
9     Switch(config)#vlan 3
10    Switch(config-vlan)#name VLAN3
11    Switch(config-vlan)#exit
12    Switch(config)#exit

```

1. 使用 en 进入特权模式
2. 使用 conf t 进入全局配置模式
3. 创建 vlan 2 并且命名为 VLAN2
4. 创建 vlan 3 并且命名为 VLAN3

### 4.2.2 划分 VLAN

```

1     Switch#config t
2     Enter configuration commands, one per line.  End with CNTL/Z.
3     Switch(config)#interface range fastEthernet 0/1 - 3

```

```

4 Switch(config-if-range)#switch access vlan 2
5 Switch(config-if-range)#exit
6 Switch(config)#interface range fastEthernet 0/4 - 6
7 Switch(config-if-range)#switch access vlan 3
8 Switch(config-if-range)#exit
9 Switch(config)#exit

```

将接口 fastEthernet 0/1 - 3 划分到 vlan 2, 接口 fastEthernet 0/4 - 6 划分到 vlan 3.  
使用 show vlan 命令查看配置结果

VLAN Name	Status	Ports
1 default	active	Fa0/7, Fa0/8, Fa0/9, Fa0/10 Fa0/11, Fa0/12, Fa0/13, Fa0/14 Fa0/15, Fa0/16, Fa0/17, Fa0/18 Fa0/19, Fa0/20, Fa0/21, Fa0/22 Fa0/23, Fa0/24, Gig0/1, Gig0/2
2 VLAN2	active	Fa0/1, Fa0/2, Fa0/3
3 VLAN3	active	Fa0/4, Fa0/5, Fa0/6
1002 fddi - default	active	
1003 token-ring - default	active	
1004 fddinet - default	active	
1005 trnet - default	active	

可以看到 VLAN2 成功分配 Fa0/1, Fa0/2, Fa0/3 端口, VLAN3 成功分配 Fa0/4, Fa0/5, Fa0/6 端口。

以太网端口有三种链路类型: Access、Hybrid 和 Trunk。

1. Access 类型的端口只能属于 1 个 VLAN, 一般用于连接计算机的端口;
2. Trunk 类型的端口可以允许多个 VLAN 通过, 可以接收和发送多个 VLAN 的报文, 一般用于交换机之间连接的端口;
3. Hybrid 类型的端口可以允许多个 VLAN 通过, 可以接收和发送多个 VLAN 的报文, 可以用于交换机之间连接, 也可以用于连接用户的计算机。

## 4.3 测试 VLAN2

### 4.3.1 测试 VLAN2 主机间的通讯

```

1 C:\>ping 192.168.0.2
2
3 Pinging 192.168.0.2 with 32 bytes of data:
4
5 Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
6 Reply from 192.168.0.2: bytes=32 time=1ms TTL=128
7 Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
8 Reply from 192.168.0.2: bytes=32 time<1ms TTL=128

```

```

9
10 Ping statistics for 192.168.0.2:
11     Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
12     Approximate round trip times in milli-seconds:
13         Minimum = 0ms, Maximum = 1ms, Average = 0ms

```

使用 PC0 去 ping 主机 PC1 可以看到成功 ping 通

### 4.3.2 测试 VLAN2 与 VLAN3 的通讯

```

1 C:\>ping 192.168.0.6
2
3 Pinging 192.168.0.6 with 32 bytes of data:
4
5 Request timed out.
6 Request timed out.
7 Request timed out.
8 Request timed out.
9
10 Ping statistics for 192.168.0.6:
11     Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

```

使用主机 PC0 去 ping 在 VLAN3 的主机 PC6 测试发现连接超时, 说明不能 ping 通

## 4.4 VLAN 广播

**源设置**

源设备: PC0

输出端口: FastEthernet0 ☒ 自动选择端口

**PDU 设置**

选择应用程序: PING

目的IP地址: 255.255.255.255

源IP地址: 192.168.0.1

TTL: 32

TOS: 0

序号:

数据包大小: 0

**仿真设置**

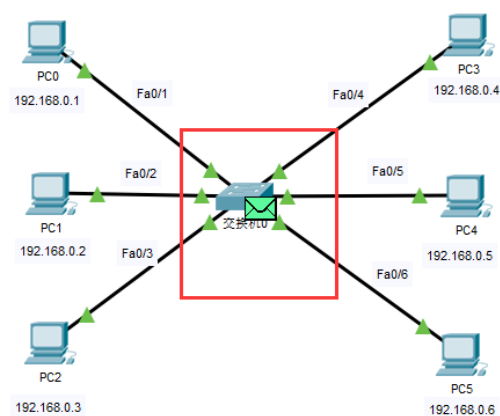
☒ 单次 时间: 1 秒

☐ 周期性 间隔: 1 秒

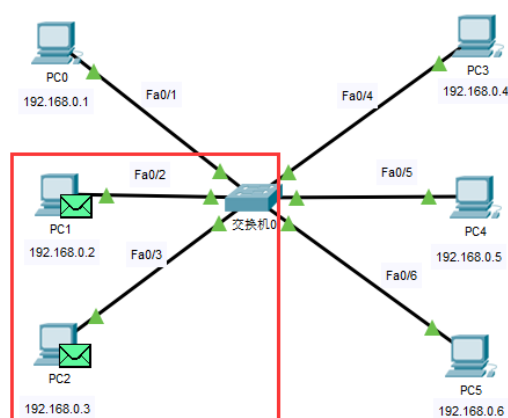
创建PDU

图 4 创建复杂 PDU

首先创建一个复杂 PDU, 将目的 IP 地址设置为 255.255.255.255 进行广播, 源地址为 192.168.0.1 代表是主机 PC1 发出的。



(a) 广播发向交换机



(b) 交换机广播至主机

可以看到 PC1 先将广播发向交换机, 然后交换机将广播发送至 VLAN 中的主机 PC1, PC2.

## 五、实验结论

实验说明同一交换机的 VLAN 之间可以通信, 同一交换机的不同 VLAN 之间不能通信。另外 VLAN 限制广播只能在 VLAN 中进行可以解决当网络规模很大时, 网上的广播信息增多, 使网络性能恶化, 形成广播风暴, 引起网络堵塞的问题。

## Part II

# 跨交换机实现 VLAN 设计

## 一、设计目的

理解 VLAN 如何跨交换机实现及应用环境

## 二、设计原理

跨交换机实现 VLAN 是通过 Trunk 端口实现的。

VLAN 中继 (VLAN Trunk) 也称为 VLAN 主干, 是指在交换机与交换机或交换机与路由器之间连接的情况下, 在互相连接的端口上配置中继模式, 使得属于不同 VLAN 的数据帧都可以通过这条中继链路进行传输。Trunk 可以在不同的交换机之间连接多个 VLAN; 由于 Trunk 实时平衡各个交换机端口和服务器的流量, 一旦某个端口出现故障, 它会自动把故障端口从 Trunk 组中撤销, 进而重新分配各个 Trunk 端口的流量, 从而实现系统容错。

VLAN 是指在一个物理网段内。进行逻辑的划分, 划分成若干个虚拟局域网, VLAN 的特性是不受物理位置的限制, 可以进行灵活的划分。VLAN 具备了一个物理网段所具备的特性。相同 VLAN 内的主机可以相互直接通信, 不同 VLAN 间的主机之间互相访问必须经路由设备进行转发, 广播数据包只可以在本 VLAN 内进行广播, 不能传输到其他 VLAN 中。

Tag VLAN 是基于交换机端口的另一种类型, 主要用于是交换机的相同 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 IEEE802.1Q 协议的标准, 在使用配置了 Tag VLAN 的端口进行数据传输时, 需要在数据帧内添加 4 个字节的 802.1Q 标签信息, 用于标示该数据帧属于哪个 VLAN, 便于对端交换机接收到数据帧后进行准确的过滤。

### 三、设计内容

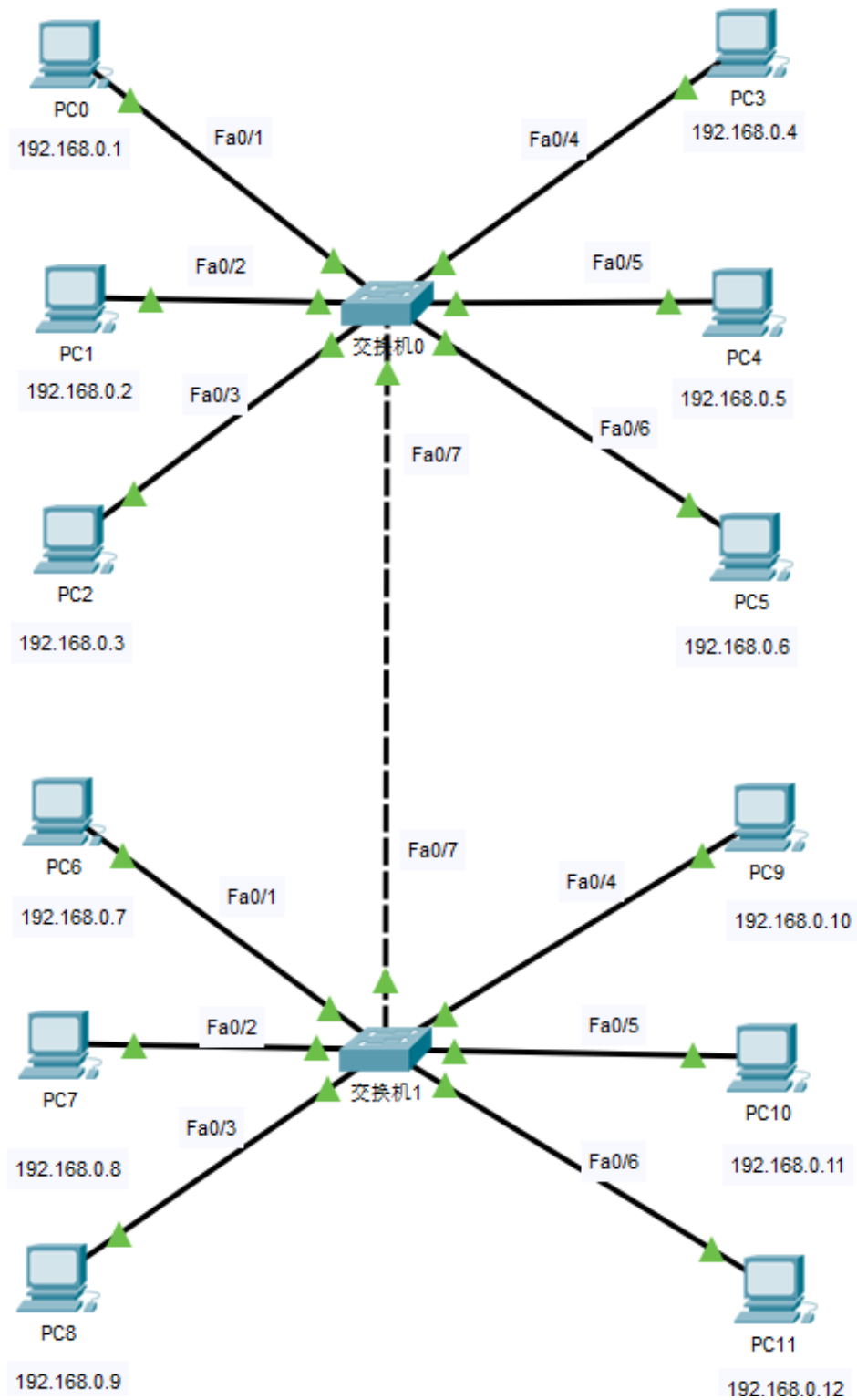


图 1

配置两台交换机, 每台交换机链接六台主机, 交换机 0 连接主机 PC0~PC5, 主机依次连接端口 Fa0/1-6, 主机 IP 地址从 192.168.0.1~192.168.0.6, 交换机 1 连接主机 PC6~PC11,



主机依次连接端口 Fa0/1-6, 主机 IP 地址从 192.168.0.7~192.168.0.11. 两台交换机连接端口号都设置为 Fa0/7.

将交换机 0 和交换机 1 的 Fa0/1-3 划分到 VLAN2, 将交换机 0 和交换机 1 的 Fa0/4-6 划分到 VLAN3, 即将图 (1) 左边的六台主机 (PC0~PC2 和 PC6~PC8) 划分到 VLAN2 中, 将图 (1) 右边的六台主机 (PC3~PC5 和 PC9~PC11) 划分到 VLAN3 中.

对于以下两种情况用 ping 命令测试两台主机之间的通讯和利用复杂 PDU 观察 VLAN 中的广播。

1. 交换机 0 和交换机 1 端口 Fa0/7 链路类型设置为 Access
2. 交换机 0 和交换机 1 端口 Fa0/7 链路类型设置为 Trunk

## 四、实验步骤

### 4.1 配置第二台交换机与主机

#### 4.1.1 测试网络是否通畅

```
1      C:\>ping 192.168.0.8
2
3      Pinging 192.168.0.8 with 32 bytes of data:
4
5      Reply from 192.168.0.8: bytes=32 time<1ms TTL=128
6      Reply from 192.168.0.8: bytes=32 time<1ms TTL=128
7      Reply from 192.168.0.8: bytes=32 time<1ms TTL=128
8      Reply from 192.168.0.8: bytes=32 time<1ms TTL=128
9
10     Ping statistics for 192.168.0.8:
11         Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
12     Approximate round trip times in milli-seconds:
13         Minimum = 0ms, Maximum = 0ms, Average = 0ms
14
15     C:\>ping 192.168.0.9
16
17     Pinging 192.168.0.9 with 32 bytes of data:
18
19     Reply from 192.168.0.9: bytes=32 time=1ms TTL=128
20     Reply from 192.168.0.9: bytes=32 time<1ms TTL=128
21     Reply from 192.168.0.9: bytes=32 time<1ms TTL=128
22     Reply from 192.168.0.9: bytes=32 time<1ms TTL=128
23
24     Ping statistics for 192.168.0.9:
25         Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
26     Approximate round trip times in milli-seconds:
27         Minimum = 0ms, Maximum = 1ms, Average = 0ms
28
29     C:\>ping 192.168.0.10
30
```

```

31      Pinging 192.168.0.10 with 32 bytes of data:
32
33      Reply from 192.168.0.10: bytes=32 time=1ms TTL=128
34      Reply from 192.168.0.10: bytes=32 time=1ms TTL=128
35      Reply from 192.168.0.10: bytes=32 time=1ms TTL=128
36      Reply from 192.168.0.10: bytes=32 time<1ms TTL=128
37
38      Ping statistics for 192.168.0.10:
39          Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
40      Approximate round trip times in milli-seconds:
41          Minimum = 0ms, Maximum = 1ms, Average = 0ms
42
43      C:\>ping 192.168.0.11
44
45      Pinging 192.168.0.11 with 32 bytes of data:
46
47      Reply from 192.168.0.11: bytes=32 time=1ms TTL=128
48      Reply from 192.168.0.11: bytes=32 time<1ms TTL=128
49      Reply from 192.168.0.11: bytes=32 time=1ms TTL=128
50      Reply from 192.168.0.11: bytes=32 time<1ms TTL=128
51
52      Ping statistics for 192.168.0.11:
53          Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
54      Approximate round trip times in milli-seconds:
55          Minimum = 0ms, Maximum = 1ms, Average = 0ms
56
57      C:\>ping 192.168.0.12
58
59      Pinging 192.168.0.12 with 32 bytes of data:
60
61      Reply from 192.168.0.12: bytes=32 time=1ms TTL=128
62      Reply from 192.168.0.12: bytes=32 time<1ms TTL=128
63      Reply from 192.168.0.12: bytes=32 time=1ms TTL=128
64      Reply from 192.168.0.12: bytes=32 time<1ms TTL=128
65
66      Ping statistics for 192.168.0.12:
67          Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
68      Approximate round trip times in milli-seconds:
69          Minimum = 0ms, Maximum = 1ms, Average = 0ms

```

#### 4.1.2 将交换机端口划分 VLAN

在交换机 1 中重复在交换机 0 中的操作即可, 使用 `show vlan brief` 命名查看 VLAN 划分结果

```

1      Switch#show vlan brief
2
3      VLAN Name                Status    Ports
4      ----  -
5      1      default                active    Fa0/7, Fa0/8, Fa0/9, Fa0/10
6                                          Fa0/11, Fa0/12, Fa0/13, Fa0/14
7                                          Fa0/15, Fa0/16, Fa0/17, Fa0/18

```

8				Fa0/19, Fa0/20, Fa0/21, Fa0/22
9				Fa0/23, Fa0/24, Gig0/1, Gig0/2
10	2	VLAN2	active	Fa0/1, Fa0/2, Fa0/3
11	3	VLAN3	active	Fa0/4, Fa0/5, Fa0/6
12	1002	fddi - default	active	
13	1003	token-ring - default	active	
14	1004	fddinet - default	active	
15	1005	trnet - default	active	

可以看到端口 Fa0/1~Fa0/3 成功被分配到 VLAN2 中, 端口 Fa0/4~Fa0/6 成功被分配到 VLAN3 中, 说明配置成功.

## 4.2 端口 Fa0/7 为 Access 时

### 4.2.1 测试 VLAN2 中主机的通讯

```

1      C:\>ping 192.168.0.2
2
3      Pinging 192.168.0.2 with 32 bytes of data:
4
5      Reply from 192.168.0.2: bytes=32 time=4ms TTL=128
6      Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
7      Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
8      Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
9
10     Ping statistics for 192.168.0.2:
11         Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
12     Approximate round trip times in milli-seconds:
13         Minimum = 0ms, Maximum = 4ms, Average = 1ms
14
15     C:\>ping 192.168.0.7
16
17     Pinging 192.168.0.7 with 32 bytes of data:
18
19     Request timed out.
20     Request timed out.
21     Request timed out.
22     Request timed out.
23
24     Ping statistics for 192.168.0.7:
25         Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

```

首先使用 VLAN2 中的主机 PC0 去 Ping 同处于 VLAN2 且连接同一交换机的主机 PC1, 发现能够 ping 通.

然后使用 VLAN2 中的主机 PC0 去 Ping 同处于 VLAN2 但并没有连接同一交换机的主机 PC6, 发现 Request timed out, 说明在这两台主机间并不能通讯.

### 4.2.2 测试 VLAN3 中主机的通讯

```
1 C:\>ping 192.168.0.5
2
3 Pinging 192.168.0.5 with 32 bytes of data:
4
5 Reply from 192.168.0.5: bytes=32 time=1ms TTL=128
6 Reply from 192.168.0.5: bytes=32 time<1ms TTL=128
7 Reply from 192.168.0.5: bytes=32 time<1ms TTL=128
8 Reply from 192.168.0.5: bytes=32 time=1ms TTL=128
9
10 Ping statistics for 192.168.0.5:
11     Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
12     Approximate round trip times in milli-seconds:
13         Minimum = 0ms, Maximum = 1ms, Average = 0ms
14
15 C:\>ping 192.168.0.10
16
17 Pinging 192.168.0.10 with 32 bytes of data:
18
19 Request timed out.
20 Request timed out.
21 Request timed out.
22 Request timed out.
23
24 Ping statistics for 192.168.0.10:
25     Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

首先使用 VLAN3 中的主机 PC3 去 Ping 同处于 VLAN3 且连接同一交换机的主机 PC4, 发现能够 ping 通.

然后使用 VLAN3 中的主机 PC3 去 Ping 同处于 VLAN3 但并没有连接同一交换机的主机 PC9, 发现 Request timed out, 说明在这两台主机间并不能通讯.

### 4.2.3 测试 VLAN2 与 VLAN3 间主机的通讯

```
1 C:\>ping 192.168.0.4
2
3 Pinging 192.168.0.4 with 32 bytes of data:
4
5 Request timed out.
6 Request timed out.
7 Request timed out.
8 Request timed out.
9
10 Ping statistics for 192.168.0.4:
11     Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
12
13 C:\>ping 192.168.0.10
14
15 Pinging 192.168.0.10 with 32 bytes of data:
```

```

14
15     Request timed out.
16     Request timed out.
17     Request timed out.
18     Request timed out.
19
20     Ping statistics for 192.168.0.10:
21         Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

```

首先使用 VLAN2 中的主机 PC1 去 Ping 处于 VLAN3 且连接同一交换机的主机 PC3, 结果为 Request timed out, 说明在这两台主机间并不能通讯。

然后使用 VLAN2 中的主机 PC1 去 Ping 处于 VLAN3 但并没有连接同一交换机的主机 PC9, 结果为 Request timed out, 说明在这两台主机间并不能通讯。

#### 4.2.4 观察 VLAN 中的广播

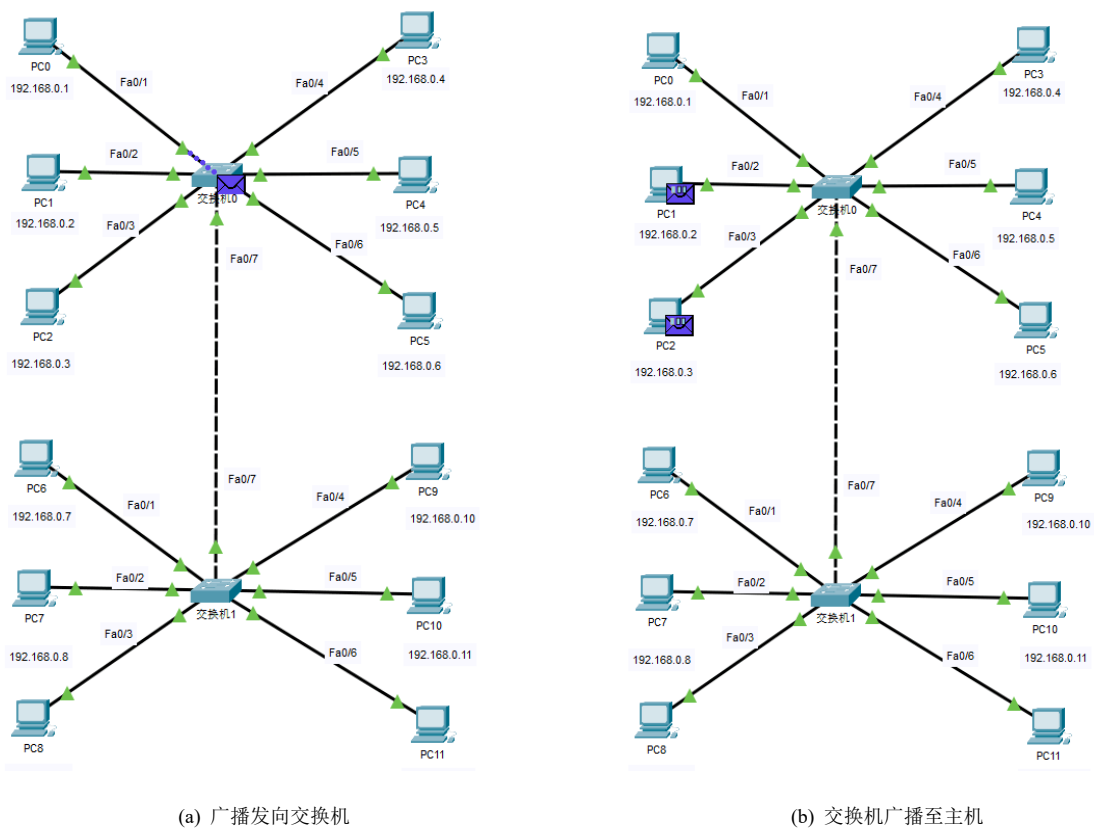


图 2 Access 端口类型广播

从图 (3) 可以看出, 如果交换机端口 Fa0/7 的链路类型为 Access 时, 同一 VLAN 的 PC 连接不同交换机时并不能直接通讯。

## 4.3 端口 Fa0/7 为 Trunk 时

### 4.3.1 测试 VLAN2 中主机的通讯

```
1 C:\>ping 192.168.0.2
2 Pinging 192.168.0.2 with 32 bytes of data:
3
4 Reply from 192.168.0.2: bytes=32 time=1ms TTL=128
5 Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
6 Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
7 Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
8
9 Ping statistics for 192.168.0.2:
10     Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
11     Approximate round trip times in milli-seconds:
12         Minimum = 0ms, Maximum = 1ms, Average = 0ms
13
14 C:\>ping 192.168.0.7
15 Pinging 192.168.0.7 with 32 bytes of data:
16
17 Reply from 192.168.0.7: bytes=32 time=1ms TTL=128
18 Reply from 192.168.0.7: bytes=32 time=1ms TTL=128
19 Reply from 192.168.0.7: bytes=32 time=1ms TTL=128
20 Reply from 192.168.0.7: bytes=32 time<1ms TTL=128
21
22 Ping statistics for 192.168.0.7:
23     Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
24     Approximate round trip times in milli-seconds:
25         Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

首先使用 VLAN2 中的主机 PC1 去 Ping 同处于 VLAN2 且连接同一交换机的主机 PC2, 发现能够 ping 通.

然后使用 VLAN2 中的主机 PC1 去 Ping 同处于 VLAN2 但并没有连接同一交换机的主机 PC6, 发现可以 ping 通, 说明在这两台主机间可以通讯.

### 4.3.2 测试 VLAN3 中主机的通讯

```
1 C:\>ping 192.168.0.5
2 Pinging 192.168.0.5 with 32 bytes of data:
3
4 Reply from 192.168.0.5: bytes=32 time=1ms TTL=128
5 Reply from 192.168.0.5: bytes=32 time=1ms TTL=128
6 Reply from 192.168.0.5: bytes=32 time=3ms TTL=128
7 Reply from 192.168.0.5: bytes=32 time<1ms TTL=128
8
9 Ping statistics for 192.168.0.5:
10     Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
11     Approximate round trip times in milli-seconds:
```

```

12         Minimum = 0ms, Maximum = 3ms, Average = 1ms
13
14     C:\>ping 192.168.0.10
15     Pinging 192.168.0.10 with 32 bytes of data:
16
17     Reply from 192.168.0.10: bytes=32 time=11ms TTL=128
18     Reply from 192.168.0.10: bytes=32 time<1ms TTL=128
19     Reply from 192.168.0.10: bytes=32 time=1ms TTL=128
20     Reply from 192.168.0.10: bytes=32 time<1ms TTL=128
21
22     Ping statistics for 192.168.0.10:
23         Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
24     Approximate round trip times in milli-seconds:
25         Minimum = 0ms, Maximum = 11ms, Average = 3ms

```

首先使用 VLAN3 中的主机 PC3 去 Ping 同处于 VLAN3 且连接同一交换机的主机 PC4, 发现能够 ping 通.

然后使用 VLAN3 中的主机 PC3 去 Ping 同处于 VLAN3 但并没有连接同一交换机的主机 PC9, 发现可以 ping 通, 说明在这两台主机间可以通讯.

### 4.3.3 测试 VLAN2 与 VLAN3 间主机的通讯

```

1     C:\>ping 192.168.0.1
2     Pinging 192.168.0.1 with 32 bytes of data:
3
4     Request timed out.
5     Request timed out.
6     Request timed out.
7     Request timed out.
8
9     Ping statistics for 192.168.0.1:
10         Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
11
12     C:\>ping 192.168.0.7
13     Pinging 192.168.0.7 with 32 bytes of data:
14
15     Request timed out.
16     Request timed out.
17     Request timed out.
18     Request timed out.
19
20     Ping statistics for 192.168.0.7:
21         Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

```

首先使用 VLAN2 中的主机 PC1 去 Ping 处于 VLAN3 且连接同一交换机的主机 PC3, 结果为 Request timed out, 说明在这两台主机间并不能通讯.

然后使用 VLAN2 中的主机 PC1 去 Ping 处于 VLAN3 但并没有连接同一交换机的主机 PC9, 结果为 Request timed out, 说明在这两台主机间并不能通讯.

#### 4.3.4 观察 VLAN 中的广播



图 3 Trunk 端口类型广播



## 五、实验结论

Access 类型的端口在转发数据帧的时候会将数据帧首部中的 VLAN 号提取出来和自己所处的 VLAN 的 VLAN 号做比对, 相等的话就转发, 不相等的话就不转发.

Trunk 类型端口可以转发所有 VLAN 上的数据帧, 分为两种情况进行转发。

1. 第一种情况是数据帧的 VLANID 和 trunk 端口的 VLANID 相等, 这种情况 Trunk 会将数据帧首部的 VLANID 去掉 (去标签), 然后再进行转发。
2. 第二种情况是数据帧首部的 VLANID 与 Trunk 端口的本征 VLANID 不等, 这种情况端口之间将数据帧发出去而不去掉标签。

观察本次实验过程可以得出同一 VLAN 跨交换机通讯需要使用 Trunk 端口.

## Part III

# 静态路由设计

## 一、设计目的

1. 掌握静态路由的配置方法和技巧
2. 掌握通过静态路由方式实现网络的连通性
3. 熟悉广域网线缆的连接方式

## 二、设计原理

1. 路由器属于网络层设备，能够根据 IP 包头的信息，选择一条最佳路径，将数据包转发出去。实现不同网段的主机之间的互相访问。路由器是根据路由表进行选路和转发的。而路由表里就是由一条条路由信息组成
2. 生成路由表主要有两种方法：手工配置和动态配置，即静态路由协议配置和动态路由协议配置。
3. 静态路由是指有网络管理员手工配置的路由信息。
4. 静态路由除了具有简单、高效、可靠的优点外，它的另一个好处是网络安全保密性高。
5. 缺省路由可以看做是静态路由的一种特殊情况。当数据在查找路由表时，没有找到和目标相匹配的路由表项时，为数据指定路由

### 三、设计内容

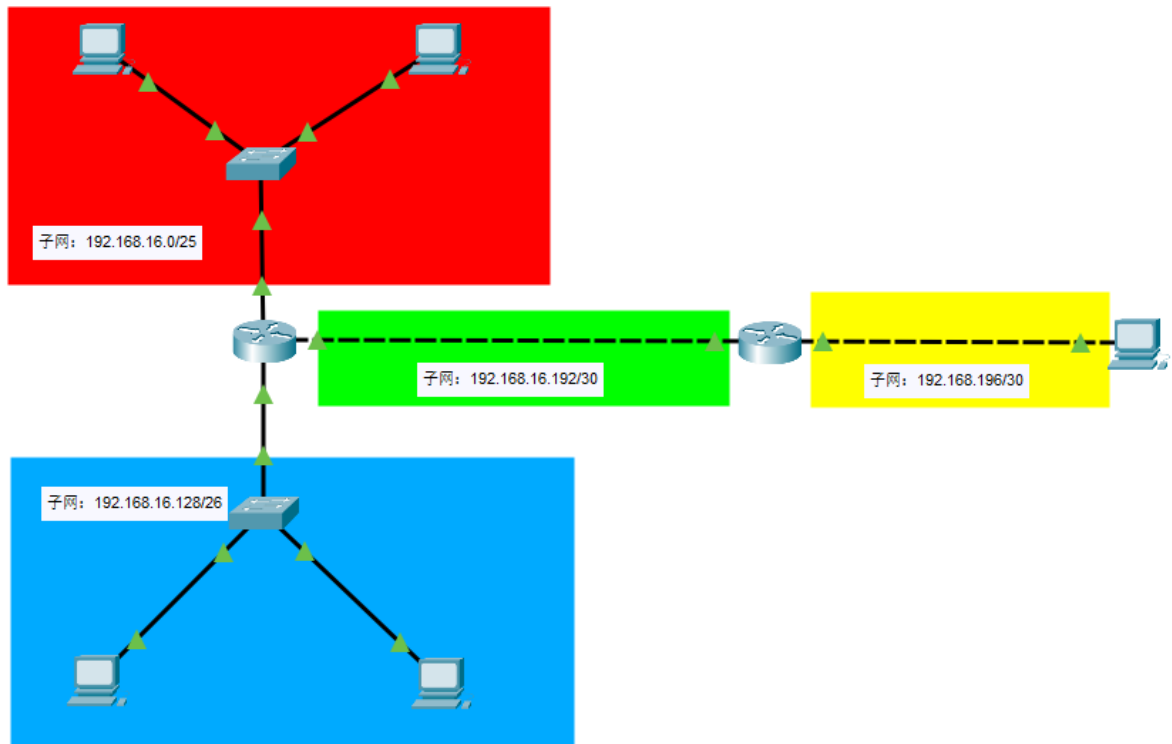


图 1 子网划分

拓扑结构图 (2) 所示, 将各个主机和路由器划分成了四个子网

1. 给各主机分配一个 IP 地址并设置子网掩码
2. 给各路由器的各接口分配一个 IP 地址并设置子网掩码
3. 给各主机指定默认网关 (默认路由器)
4. 尝试各个子网间的通讯
5. 给各路由器配置静态路由 (进行路由聚合)
6. 尝试各个子网间的通讯

## 四、实验步骤

### 4.1 配置路由器和主机

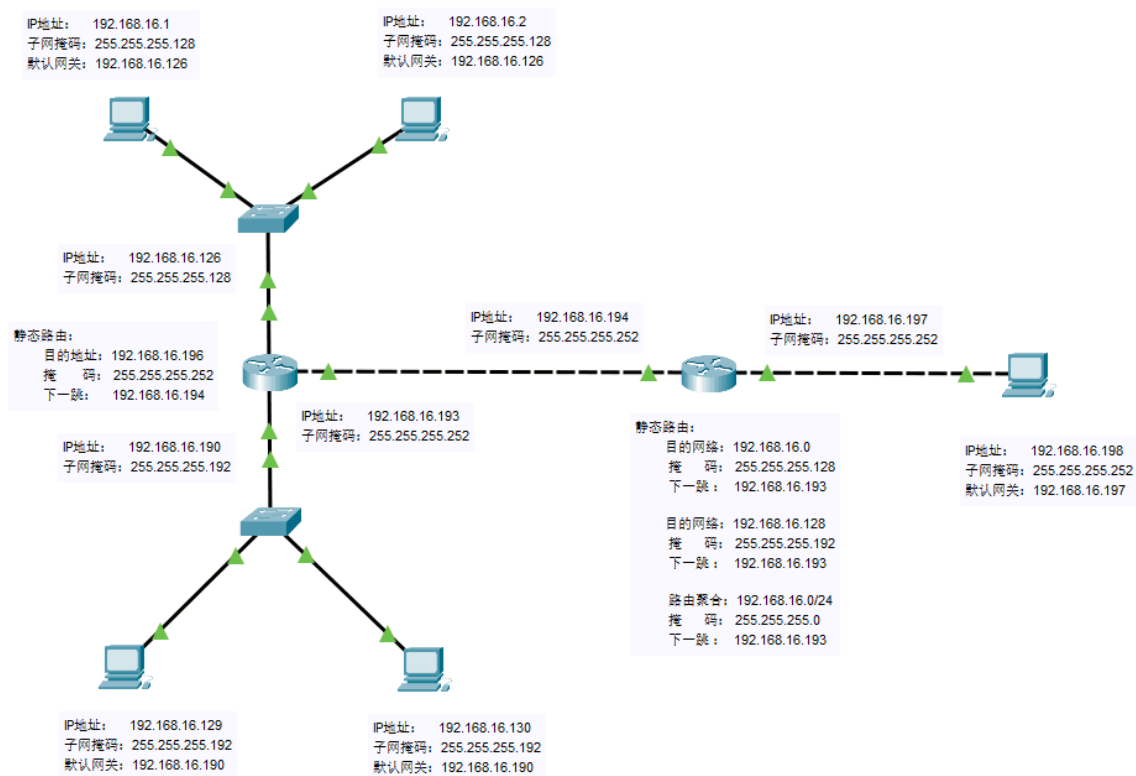


图 2 网络拓扑

将路由器端口和主机 IP 地址和子网掩码按照图 (2) 进行配置。  
配置路由器代码如下:

```
1 Router>en
2 Router#conf t
3 Enter configuration commands, one per line. End with CNTL/Z.
4 Router(config)#interface GigabitEthernet0/0
5 Router(config-if)#no shut
6 Router(config-if)#
7 %LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up
8 Router(config-if)#exit
9 Router(config)#interface GigabitEthernet0/0
10 Router(config-if)#ip address 192.168.16.126 255.255.255.128
11 Router(config-if)#exit
```

上述代码为配置路由器端口 0/0, 将 IP 地址设置为 192.168.16.126, 子网掩码设置为 255.255.255.128, 命令 no shut 用来启用端口, 同理可以配置路由器的其他两个端口  
配置主机 IP 地址, 网关和子网掩码如图 (??):

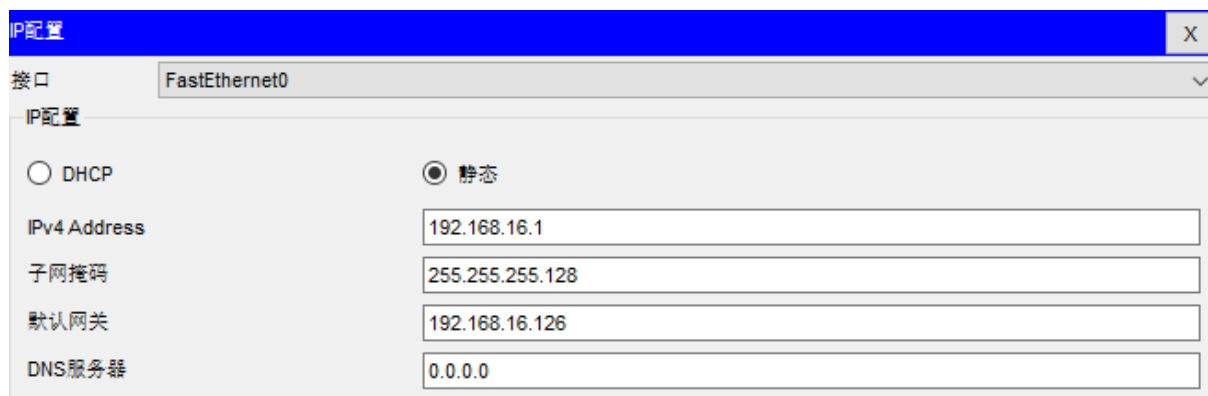


图 3 配置主机

## 4.2 测试子网内主机的通讯

1. 用子网 1 的主机 PC0(192.168.16.1) 去 ping 同处于子网 1 的主机 PC1(192.168.16.2), 结果如下, 说明子网 1 内主机通讯正常.

```
1 C:\>ping 192.168.16.2
2
3 Pinging 192.168.16.2 with 32 bytes of data:
4
5 Reply from 192.168.16.2: bytes=32 time<1ms TTL=128
6 Reply from 192.168.16.2: bytes=32 time<1ms TTL=128
7 Reply from 192.168.16.2: bytes=32 time<1ms TTL=128
8 Reply from 192.168.16.2: bytes=32 time<1ms TTL=128
9
10 Ping statistics for 192.168.16.2:
11     Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
12     Approximate round trip times in milli-seconds:
13         Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

2. 用子网 2 的主机 PC2(192.168.16.129) 的主机去 ping 同处于子网 2 的主机 PC3(192.168.16.130) 的主机, 结果如下, 说明子网 2 内主机通讯正常.

```
1 C:\>ping 192.168.16.130
2
3 Pinging 192.168.16.130 with 32 bytes of data:
4
5 Reply from 192.168.16.130: bytes=32 time<1ms TTL=128
6 Reply from 192.168.16.130: bytes=32 time<1ms TTL=128
7 Reply from 192.168.16.130: bytes=32 time=21ms TTL=128
8 Reply from 192.168.16.130: bytes=32 time<1ms TTL=128
9
10 Ping statistics for 192.168.16.130:
11     Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
12     Approximate round trip times in milli-seconds:
13         Minimum = 0ms, Maximum = 21ms, Average = 5ms
```

## 4.3 测试不同子网内主机的通讯

### 4.3.1 测试子网 1 与子网 2 的通讯

用子网 1 的地址为 PC0(192.168.16.1) 的主机去 ping 处于子网 2 地址为 PC2(192.168.16.129) 的主机, 观察结果并进行简单分析

```
1      C:\>ping 192.168.16.129
2
3      Pinging 192.168.16.129 with 32 bytes of data:
4
5      Request timed out.
6      Request timed out.
7      Reply from 192.168.16.129: bytes=32 time<1ms TTL=127
8      Reply from 192.168.16.129: bytes=32 time<1ms TTL=127
9
10     Ping statistics for 192.168.16.129:
11         Packets: Sent = 4, Received = 2, Lost = 2 (50% loss),
12     Approximate round trip times in milli-seconds:
13         Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

可以发现前两次请求超时, 超时的原因是因为, 主机判断目的地址和自己并不在一个网络内所以需要交付给路由器进行处理, 于是发送一个广播请求询问路由器的 MAC 地址. 路由器会先查找 ARP 表发现并没有对应的记录于是路由器也会发一个广播请求询问目的地址的 MAC 地址, 目的主机收到后, 就会给路由器发送一个单播的响应, 告诉路由器自己的 MAC 地址, 路由器将 MAC 地址缓存进 ARP 表中, 所以才会导致超时. 但是在后续的请求中就不需要再发送广播请求寻址主机, 可以直接查询 ARP 表, 所以后面两次正常通讯.

可以看到路由器 ARP 缓存表中增加了 192.168.16.129 的缓存.

ARP表 来自 路由器1			
IP 地址	硬件 地址	接口	
192.168.16.1	0060.7055.54DB	GigabitEthernet0/0	
192.168.16.126	0002.4ABD.2001	GigabitEthernet0/0	
192.168.16.129	0030.A37C.D0BD	GigabitEthernet0/1	
192.168.16.190	0002.4ABD.2002	GigabitEthernet0/1	
192.168.16.193	0002.4ABD.2003	GigabitEthernet0/2	

图 4 ARP 表

#### 4.3.2 测试子网 1 与子网 4 的通讯

```

1      C:\>ping 192.168.16.198
2
3      Pinging 192.168.16.198 with 32 bytes of data:
4
5      Reply from 192.168.16.126: Destination host unreachable.
6      Reply from 192.168.16.126: Destination host unreachable.
7      Request timed out.
8      Reply from 192.168.16.126: Destination host unreachable.
9
10     Ping statistics for 192.168.16.198:
11         Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

```

用处于子网 1 的 PC0 去 ping 处于子网 4 的 PC6, 发现收到的响应来自 192.168.16.126 即主机的网关地址. 且显示目的主机不可达

其原因是因为主机判断自己和目的地址不在一个网段内所以将数据包转发给路由器, 由路由器对数据包进行转发. 但是路由器不知道怎么转发, 所以回复 Destination host unreachable(目的主机不可达).

路由器找不到目的地址的原因是再路由表上并没有对应的网络信息, 相当于路由器不知道网络的存在, 所以需要静态配置路由, 来告诉路由器网络的存在

路由表 for 路由器1					
类型	网络	端口	下一跳 IP	度量	
C	192.168.16.0/25	GigabitEthernet0/0	---	0/0	
L	192.168.16.126/32	GigabitEthernet0/0	---	0/0	
C	192.168.16.128/26	GigabitEthernet0/1	---	0/0	
L	192.168.16.190/32	GigabitEthernet0/1	---	0/0	
C	192.168.16.192/30	GigabitEthernet0/2	---	0/0	
L	192.168.16.193/32	GigabitEthernet0/2	---	0/0	

图 5 路由表

4.4 配置静态路由

配置静态路由由两种方法

1. 使用命令行

```
1 Router(config)#ip route 192.168.16.0 255.255.255.128 192.168.16.193
```

2. 采用图形化窗口配置



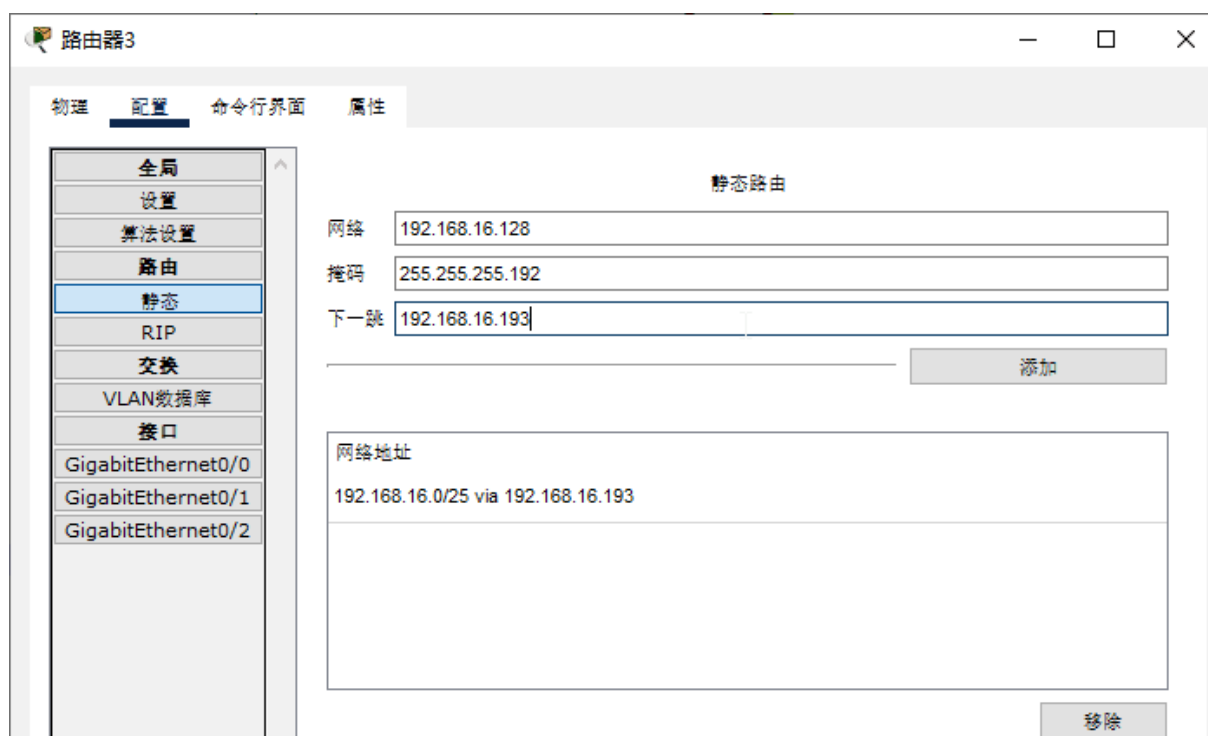


图 6 配置静态路由

配置好后查看路由表, 其中类型 S 代表配置的静态路由

类型	网络	端口	下一跳 IP	度量
C	192.168.16.0/25	GigabitEthernet0/0	---	0/0
L	192.168.16.126/32	GigabitEthernet0/0	---	0/0
C	192.168.16.128/26	GigabitEthernet0/1	---	0/0
L	192.168.16.190/32	GigabitEthernet0/1	---	0/0
C	192.168.16.192/30	GigabitEthernet0/2	---	0/0
L	192.168.16.193/32	GigabitEthernet0/2	---	0/0
S	192.168.16.196/30	---	192.168.16.194	1/0

图 7 路由表

最后分别测试子网 1 和子网 2 是否能和子网 4 通讯

```
1 C:\>ping 192.168.16.1
2
3 Pinging 192.168.16.1 with 32 bytes of data:
4
5 Reply from 192.168.16.1: bytes=32 time<1ms TTL=126
6 Reply from 192.168.16.1: bytes=32 time<1ms TTL=126
7 Reply from 192.168.16.1: bytes=32 time<1ms TTL=126
8 Reply from 192.168.16.1: bytes=32 time<1ms TTL=126
9
10 Ping statistics for 192.168.16.1:
11     Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
12     Approximate round trip times in milli-seconds:
13         Minimum = 0ms, Maximum = 0ms, Average = 0ms
14
15 C:\>ping 192.168.16.129
16
17 Pinging 192.168.16.129 with 32 bytes of data:
18
19 Reply from 192.168.16.129: bytes=32 time<1ms TTL=126
20 Reply from 192.168.16.129: bytes=32 time<1ms TTL=126
21 Reply from 192.168.16.129: bytes=32 time<1ms TTL=126
22 Reply from 192.168.16.129: bytes=32 time<1ms TTL=126
23
24 Ping statistics for 192.168.16.129:
25     Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
26     Approximate round trip times in milli-seconds:
27         Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

可以看到子网 1 可以和子网 4 正常通讯了。

## 五、实验总结

通过本实验，我对静态路由有了一定的了解，也基本掌握了静态路由的配置方法，熟悉了使用 Ping 命令查询路由信息的方法，对各个网络的互联也有一个深刻的理解。对于位于不同网段内的主机通信需要配置路由器

## Part IV

# 动态路由 (RIP 协议) 设计

## 一、设计目的

1. 掌握 RIP 协议的配置方法；
2. 掌握查看通过动态路由协议 RIP 学习产生的路由；
3. 熟悉广域网线缆的连接方式；

## 二、设计原理

1. RIP(Routing Information Protocols, 路由信息协议) 是应用较早、使用较普遍的 IGP 内部网管协议，使用于小型同类网络，是距离矢量协议；
2. RIP 协议跳数作为衡量路径开销的，RIP 协议里规定最大跳数为 15；
3. RIP 协议有两个版本：RIPv1 和 RIPv2，RIPv1 属于有类路由协议，不支持 VLSM，以广播形式进行路由信息的更新，更新周期为 30 秒；RIPv2 属于无类路由协议，支持 VLSM，以组播形式进行路由更细。

### 三、设计内容

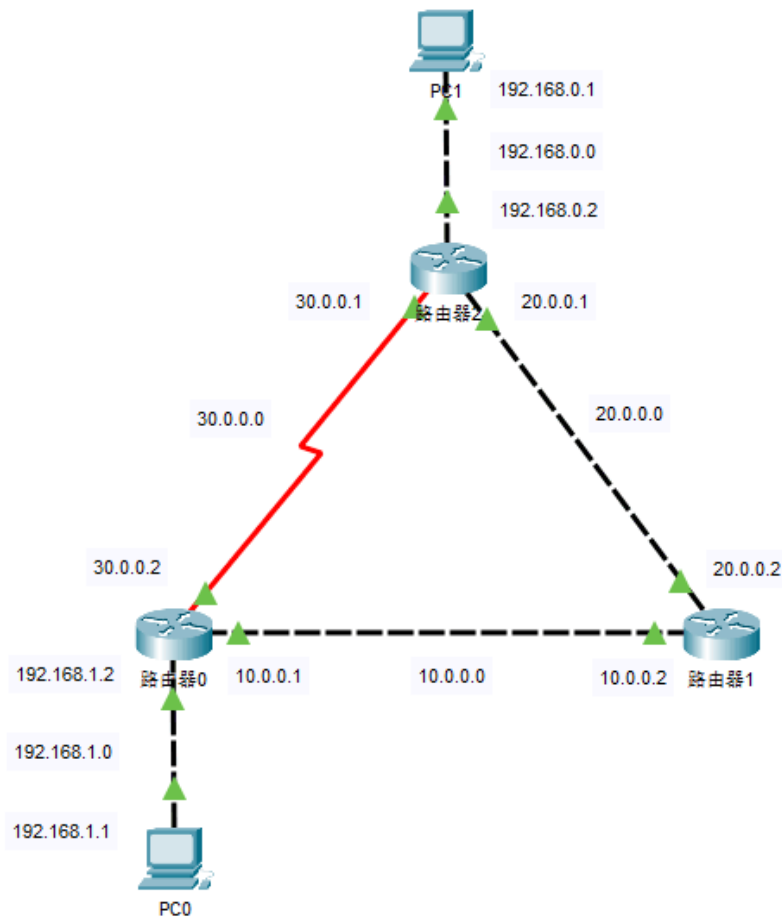


图 1 网络拓扑

网络拓扑如图 (1) 所示,PC0 与路由器 0 连接,PC1 与路由器 2 连接. 路由器 0 与路由器 1, 路由器 1 与路由器 2 都用高速链路连接, 而路由器 0 与路由器 2 通过低速链路连接.

对于主机 PC0 与主机 PC1 的通信路径有两条

1. PC0→ 路由器 0→ 路由器 2→PC1.
2. PC0→ 路由器 0→ 路由器 1→ 路由器 2→PC1.

由于由于 RIP 协议考虑的是最少的跳数, 而不考虑通信速度, 所以 RIP 协议得出主机 PC0 到 PC1 的路径应该为 PC0→ 路由器 0→ 路由器 2→PC1.

下面通过实验验证, 实验步骤如下:

1. 构建网络拓扑
2. 配置主机
3. 配置路由器
4. 使用 ping 命令测试主机间的通信
5. 配置 RIP

- 6. 观察 RIP 协议交换过程并分析 RIP 协议
- 7. 使用 ping 命令测试主机间的通信

四、实验步骤

4.1 配置主机



图 2 配置主机

按照网络拓扑结构所示配置主机的 IP 地址和网关.

4.2 配置路由器

由于路由器 0 和路由器 2 需要通过一条串行线连接, 所以需要增加一块网卡



图 3 配置网卡

其他路由器之间的连接选择交叉线即可, 不需要进行额外配置.

GigabitEthernet0/0	
接口状态	<input checked="" type="checkbox"/> 开
带宽	<input type="radio"/> 1000 Mbps <input checked="" type="radio"/> 100 Mbps <input type="radio"/> 10 Mbps <input checked="" type="checkbox"/> 自动
双工	<input type="radio"/> 半双工 <input checked="" type="radio"/> 全双工 <input checked="" type="checkbox"/> 自动
MAC地址	000B.BE02.4901
<div>IP配置</div> <div> IPv4 Address: 192.168.1.2  子网掩码: 255.255.255.0 </div>	

图 4 配置端口

对于所有路由器的各个端口按照网络拓扑图进行配置即可。

### 4.3 测试主机间通信

尝试使用主机 PC0 去 ping 主机 PC1

```

1      C:\>ping 192.168.0.1
2
3      Pinging 192.168.0.1 with 32 bytes of data:
4
5      Reply from 192.168.1.2: Destination host unreachable.
6      Reply from 192.168.1.2: Destination host unreachable.
7      Reply from 192.168.1.2: Destination host unreachable.
8      Reply from 192.168.1.2: Destination host unreachable.
9
10     Ping statistics for 192.168.0.1:
11         Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

```

提示目的不可达, 其原因在静态路由色设计中分析过了, 是因为主机 PC0 判断目的地址不在本网络, 所以转发给网关路由器, 但是网关路由器的路由表中并没有到达目的网络的信息所以返回 Destination host unreachable.

### 4.4 配置 RIP

将与路由器互联的网段都配置进路由器的 RIP 协议, 以配置路由器 0 为例, 路由器 0 连接了 10.0.0.0 网段, 30.0.0.0 网段, 192.168.1.0 网段, 命令如下

```

1      Router(config)#router rip
2      Router(config-router)#network 10.0.0.0
3      Router(config-router)#network 30.0.0.0
4      Router(config-router)#network 192.168.1.0

```

如上所示分别配置路由器 1 和路由器 2.

#### 4.5 观察 RIP 协议运行过程

将 Packet Tracker 软件从实时模式切换到仿真模式, 观察 RIP 协议的运行过程如图 (5) 所示

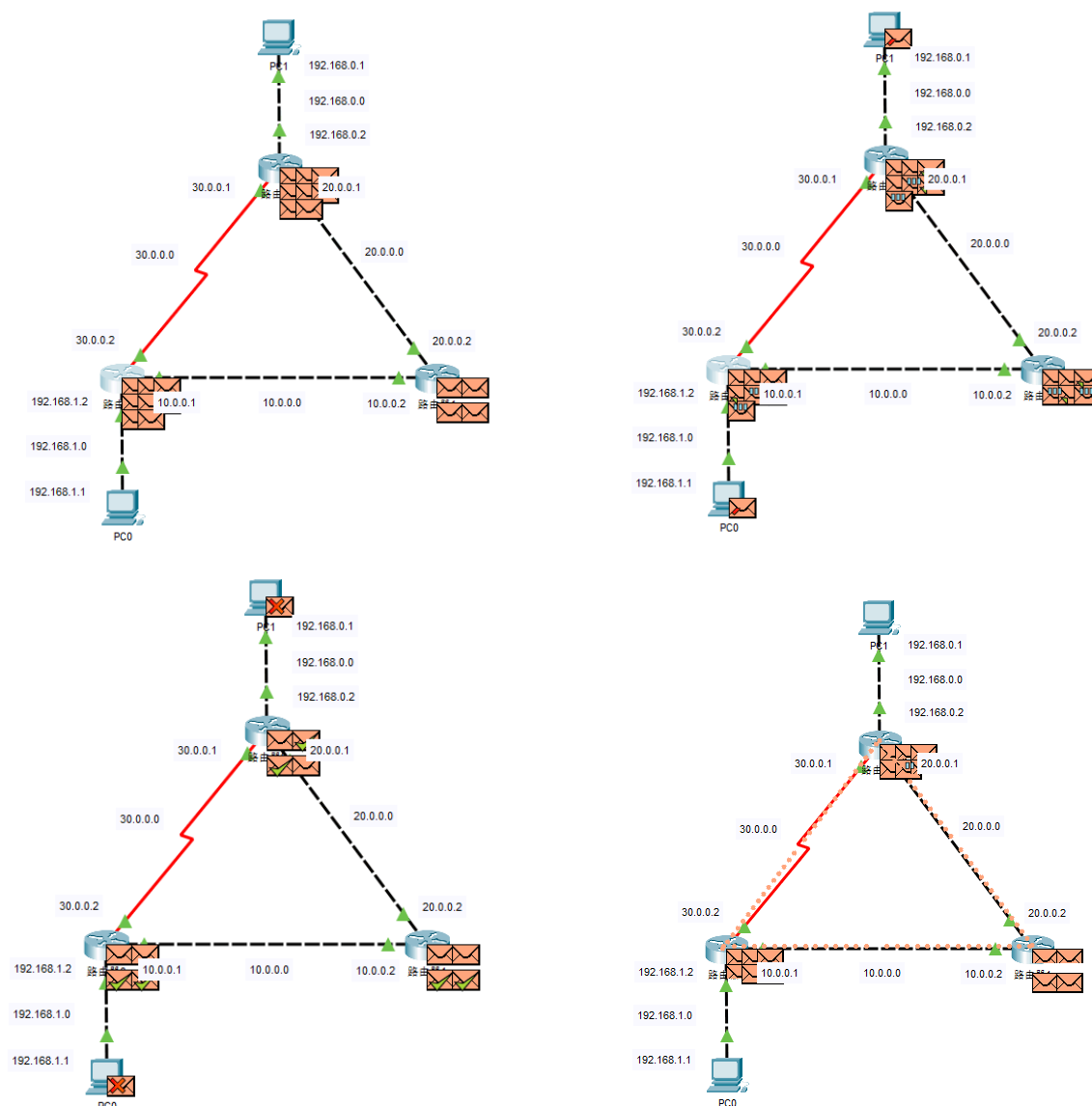


图 5 RIP 数据交换过程

可以看出 RIP 协议通过不断的发送广播来更新自己的路由信息, 以此找到最短路径. 捕获一个 RIP 协议的数据包如图 (6) 所示. 可以看出 RIP 协议是基于 UDP 的, 并且采用广播的方式来获取并更新信息.

当前在设备：路由器0  
源：路由器0  
目的：255.255.255.255

#### 逐层输入

第七层
第六层
第五层
第四层
第三层
第二层
第一层

#### 逐层输出

第七层：RIP 版本：1， 命令：2
第六层
第五层
第四层：UDP 源端口：520， 目的端口：520
第三层：IP 首部 源IP：10.0.0.1, 目的IP：255.255.255.255
第二层：以太网V2 首部 000B.BE02.4902 >> FFFF.FFFF.FFFF
第一层：端口(s)：

1. 设备构造一个周期性的RIP更新数据包并将其发送给GigabitEthernet0/1。
2. 设备添加一个更新路由30.0.0.0到RIP数据包。
3. 设备添加一个更新路由192.168.1.0到RIP数据包。



挑战我

<< 上一层

下一层 >>

图 6 RIP 数据包

## 4.6 测试主机间的通信

运行 RIP 协议一段时间后采用命令 `show ip route` 观察路由器 0 的路由表



路由表 for 路由器0				
类型	网络	端口	下一跳 IP	度量
C	10.0.0.0/8	GigabitEthernet0/1	---	0/0
L	10.0.0.1/32	GigabitEthernet0/1	---	0/0
R	20.0.0.0/8	GigabitEthernet0/1	10.0.0.2	120/1
R	20.0.0.0/8	Serial0/0/0	30.0.0.1	120/1
C	30.0.0.0/8	Serial0/0/0	---	0/0
L	30.0.0.2/32	Serial0/0/0	---	0/0
R	192.168.0.0/24	Serial0/0/0	30.0.0.1	120/1
C	192.168.1.0/24	GigabitEthernet0/0	---	0/0
L	192.168.1.2/32	GigabitEthernet0/0	---	0/0

图 7 路由表信息

主机 PC0 需要 ping 通 PC1 的话需要路由器知道怎么到达 PC1 所在的网络 192.168.0.0, 可以看到路由器 0 的路由表中已经增加了一条类型为 R 路由信息代表到达 192.168.0.0/24 这个网络下一跳的地址是 30.0.0.1, 且通过 Serial0/0/0 端口。

使用主机 PC0 去 ping 主机 PC1 结果如下:

```

1      C:\>ping 192.168.0.1
2
3      Pinging 192.168.0.1 with 32 bytes of data:
4
5      Request timed out.
6      Reply from 192.168.0.1: bytes=32 time=19ms TTL=126
7      Reply from 192.168.0.1: bytes=32 time=19ms TTL=126
8      Reply from 192.168.0.1: bytes=32 time=14ms TTL=126
9
10     Ping statistics for 192.168.0.1:
11         Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
12         Approximate round trip times in milli-seconds:
13             Minimum = 14ms, Maximum = 19ms, Average = 17ms

```

第一次出现 Request timed out. 的原因是因为主机 PC0 的 ARP 表中没有网关路由器的 MAC 地址, 需要通过广播请求路由器的 MAC 地址, 路由器将请求转发到路由器 2 后, 路由器 2 的 ARP 表中也没有主机 PC1 的 MAC 地址, 所以需要广播请求主机 PC1 的 MAC 地址. 在第一次超时后, 主机和路由器的 ARP 缓存表中都增加了对于的 MAC 记录

所以后面的请求均可以正常到达.

## 五、实验结论

观察路由器的路由表信息如图 (7) 可以验证使用 RIP 协议, 主机间的通信路由器选择的路径是 PC0→ 路由器 0→ 路由器 2→PC1. 进一步说明了 RIP 协议是以跳数作为度量的, 而不管路径长度和链路速度, 而 OSPF 协议则会考虑链路速度, 所以使用一样的拓扑图进行 OSPF 协议的实验进行验证.

## Part V

# 动态路由 (OSPF 协议) 设计

## 一、设计目的

1. 掌握 OSPF 协议的配置方法;
2. 掌握查看通过动态路由协议 OSPF 学习产生的路由;
3. 熟悉广域网线缆的连接方式;

## 二、设计原理

OSPF 开放式最短路径优先协议，是目前网路中应用最广泛的路由协议之一。属于内部网管路由协议，能够适应各种规模的网络环境，是典型的链路状态协议。OSPF 路由协议通过向全网扩散本设备的链路状态信息，使网络中每台设备最终同步一个具有全网链路状态的数据库，然后路由器采用 SPF 算法，以自己为根，计算到达其他网络的最短路径，最终形成全网路由信息。

## 三、设计内容

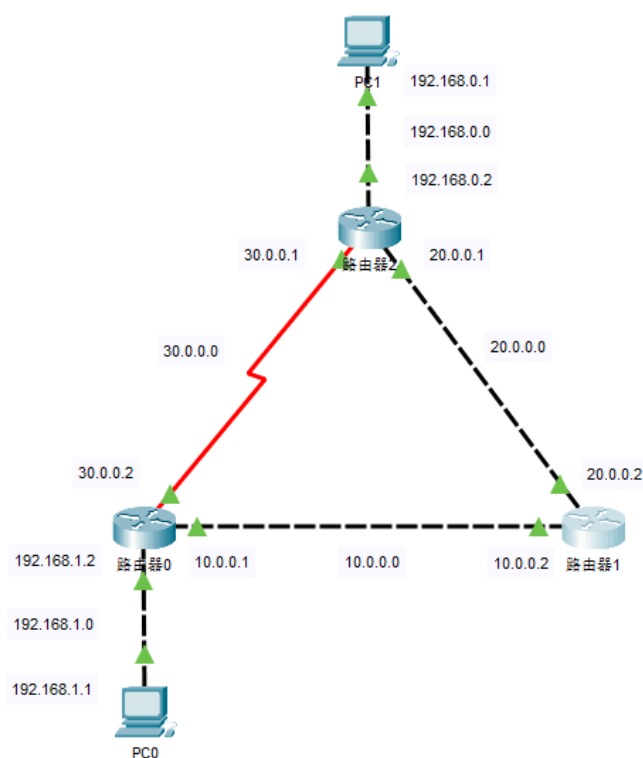


图 1 动态路由 (OSPF 协议) 设计

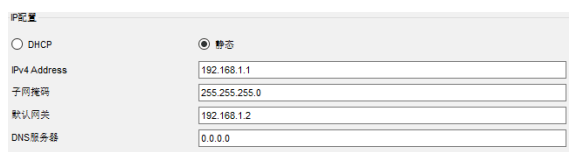
连接和动态路由 (RIP 协议) 一致的拓扑图, 配置 OSPF 协议观察路由器选择的路由.  
对于主机 PC0 与主机 PC1 的通信路径有两条

1. PC0→ 路由器 0→ 路由器 2→PC1.
2. PC0→ 路由器 0→ 路由器 1→ 路由器 2→PC1.

使用 RIP 协议会选择第一条, 因为 RIP 协议是用跳数来衡量的. 但是使用 OSPF 协议会选择第二条, 因为 OSPF 协议是基于链路状态的。接下来通过实验进行验证.

## 四、实验步骤

### 4.1 配置主机



IP配置

☐ DHCP ☒ 静态

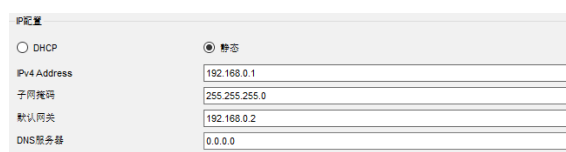
IPv4 Address: 192.168.1.1

子网掩码: 255.255.255.0

默认网关: 192.168.1.2

DNS服务器: 0.0.0.0

(a) 配置 PC0



IP配置

☐ DHCP ☒ 静态

IPv4 Address: 192.168.0.1

子网掩码: 255.255.255.0

默认网关: 192.168.0.2

DNS服务器: 0.0.0.0

(b) 配置 PC1

图 2 配置主机

按照网络拓扑结构所示配置主机的 IP 地址和网关.

### 4.2 配置路由器

由于路由器 0 和路由器 2 需要通过一条串行线连接, 所以需要增加一块网卡

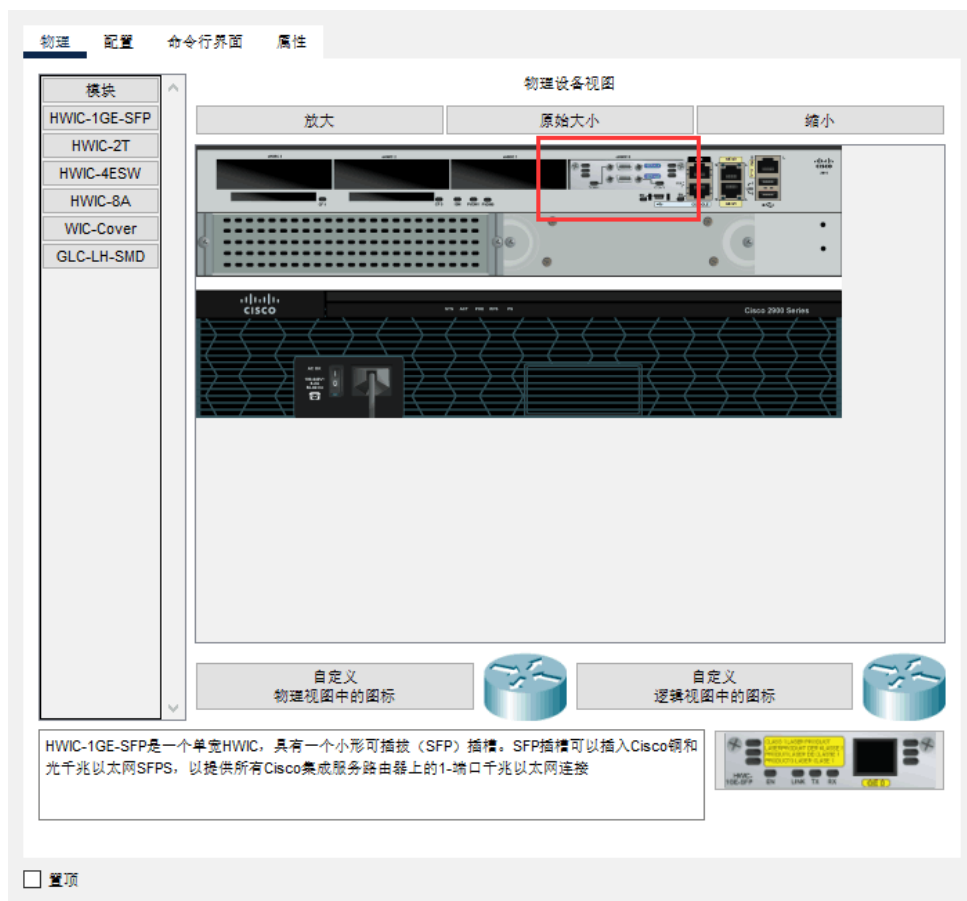


图 3 配置网卡

其他路由器之间的连接选择交叉线即可，不需要进行额外配置。

GigabitEthernet0/0	
接口状态	<input checked="" type="checkbox"/> 开
带宽	<input type="radio"/> 1000 Mbps <input checked="" type="radio"/> 100 Mbps <input type="radio"/> 10 Mbps <input checked="" type="checkbox"/> 自动
双工	<input type="radio"/> 半双工 <input checked="" type="radio"/> 全双工 <input checked="" type="checkbox"/> 自动
MAC地址	000B.BE02.4901
IP配置	
IPv4 Address	192.168.1.2
子网掩码	255.255.255.0

图 4 配置端口

对于所有路由器的各个端口按照网络拓扑图进行配置即可。

#### 4.3 测试主机间通信

尝试使用主机 PC0 去 ping 主机 PC1

```

1      C:\>ping 192.168.0.1
2
3      Pinging 192.168.0.1 with 32 bytes of data:
4
5      Reply from 192.168.1.2: Destination host unreachable.
6      Reply from 192.168.1.2: Destination host unreachable.
7      Reply from 192.168.1.2: Destination host unreachable.
8      Reply from 192.168.1.2: Destination host unreachable.
9
10     Ping statistics for 192.168.0.1:
11     Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

```

提示目的不可达, 其原因在静态路由色设计中分析过了, 是因为主机 PC0 判断目的地址不在本网络, 所以转发给网关路由器, 但是网关路由器的路由表中并没有到达目的网络的信息所以返回 Destination host

#### 4.4 启动 OSPF 协议

```

1      Router>en
2      Router#conf t
3      Enter configuration commands, one per line.  End with CNTL/Z.
4      Router(config)#router ospf 100
5      Router(config-router)#network 30.0.0.0 0.255.255.255 area 0
6      Router(config-router)#network 10.0.0.0 0.255.255.255 area 0
7      Router(config-router)#network 192.168.1.0 0.0.0.255 area 0

```

使用以上命令启动 OSPF 协议并且通告路由器网段, 注意 OSPF 协议填的不是子网掩码, 而是反子网掩码. 上述代码是启动路由器 0 的启动命令. 因为路由器 0 连接三个网段所以需要通告三个网段. 同理可以配置其他两个路由器.

#### 4.5 测试主机间的通信

使用主机 PC0 去 ping 主机 PC1 查看通信结果.

```

1      C:\>ping 192.168.0.1
2
3      Pinging 192.168.0.1 with 32 bytes of data:
4
5      Reply from 192.168.0.1: bytes=32 time=1ms TTL=125
6      Reply from 192.168.0.1: bytes=32 time<1ms TTL=125
7      Reply from 192.168.0.1: bytes=32 time=12ms TTL=125
8      Reply from 192.168.0.1: bytes=32 time=10ms TTL=125
9
10     Ping statistics for 192.168.0.1:
11     Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
12     Approximate round trip times in milli-seconds:

```

可以看到主机间可以正常通信.

使用 `show ip route` 命令查看路由器的路由表.

```

1          10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
2      C      10.0.0.0/8 is directly connected, GigabitEthernet0/1
3      L      10.0.0.1/32 is directly connected, GigabitEthernet0/1
4      O      20.0.0.0/8 [110/2] via 10.0.0.2, 00:38:38, GigabitEthernet0/1
5          30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
6      C      30.0.0.0/8 is directly connected, Serial0/0/0
7      L      30.0.0.2/32 is directly connected, Serial0/0/0
8      O      192.168.0.0/24 [110/3] via 10.0.0.2, 00:38:38, GigabitEthernet0/1
9          192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
10     C      192.168.1.0/24 is directly connected, GigabitEthernet0/0
11     L      192.168.1.2/32 is directly connected, GigabitEthernet0/0

```

可以看到路由表中有到达目的网络 192.168.0.0 的信息, 下一跳为 10.0.0.2 就是路由器 1, 度量为 3

查看路由器 1 的路由表

```

1          10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
2      C      10.0.0.0/8 is directly connected, GigabitEthernet0/0
3      L      10.0.0.2/32 is directly connected, GigabitEthernet0/0
4          20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
5      C      20.0.0.0/8 is directly connected, GigabitEthernet0/1
6      L      20.0.0.2/32 is directly connected, GigabitEthernet0/1
7      O      30.0.0.0/8 [110/65] via 10.0.0.1, 00:39:57, GigabitEthernet0/0
8          [110/65] via 20.0.0.1, 00:39:57, GigabitEthernet0/1
9      O      192.168.0.0/24 [110/2] via 20.0.0.1, 00:39:57, GigabitEthernet0/1
10     O      192.168.1.0/24 [110/2] via 10.0.0.1, 00:40:07, GigabitEthernet0/0

```

路由器 1 的路由表中指明了到达网络 192.168.0.0 的下一跳为 20.0.0.1, 度量为 2.

其中路由器 1 的路由表有 O 30.0.0.0/8 [110/65] via 10.0.0.1, 00:39:57, GigabitEthernet0/0

说明到达 30.0.0.0/8 的网络的度量为 65. 这是因为 30.0.0.0 所在网络使用了串行低速链路, 这也是为什么 PC0 到 PC1 的通信不走 30.0.0.0 的原因.

## 4.6 观察通信过程

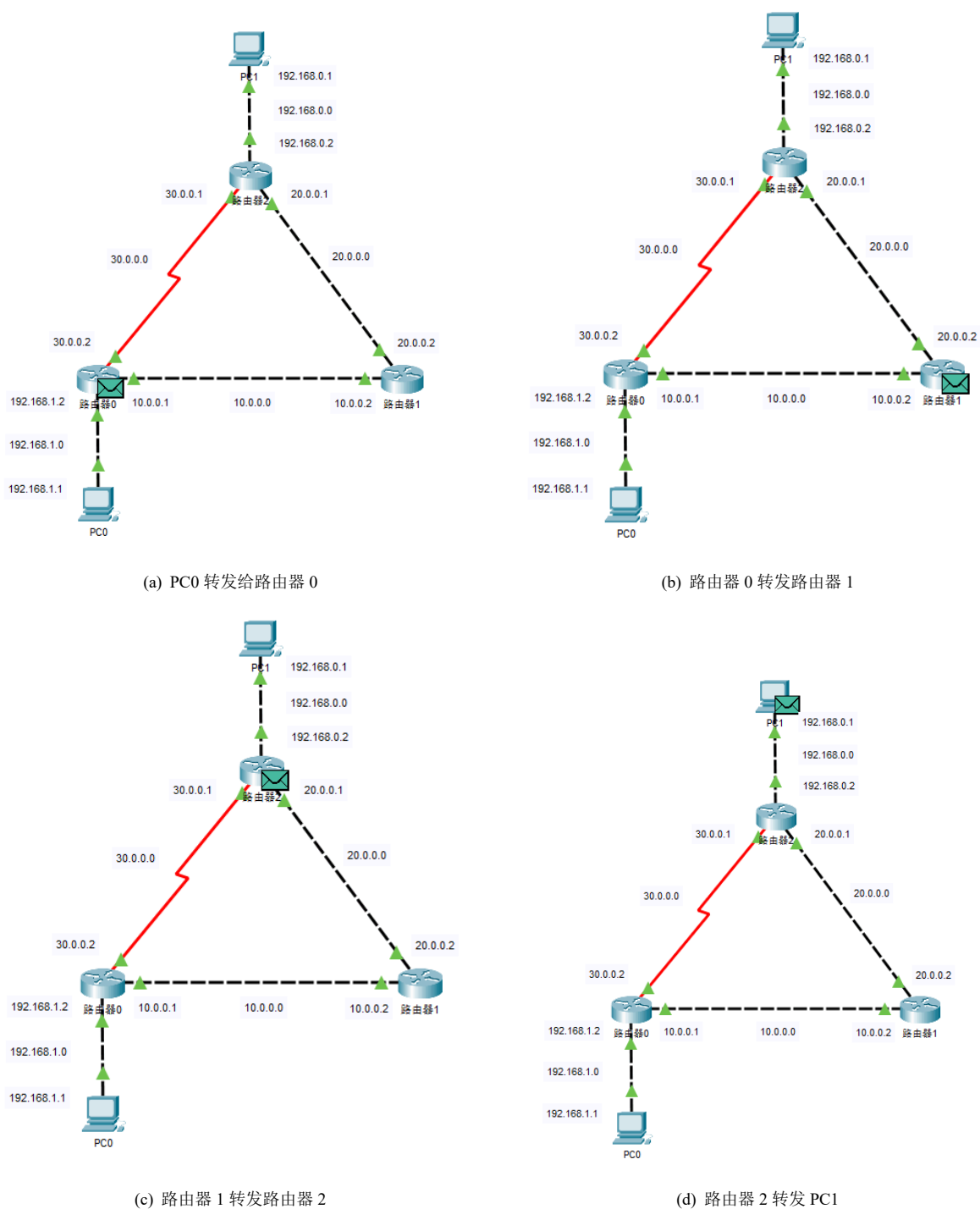


图 5 OSPF 链路选择

## 五、实验总结

这次实验验证了 OSPF 是基于链路状态的, 而 RIP 是基于跳数的. 对于同一幅拓扑图, OSPF 选择的路径和 RIP 选择的路径可能是不同的. 同时也掌握了 OSPF 协议的配置.



## Part VI

# DNS 服务器设计

## 一、设计目的

1. 理解 DNS 服务器原理
2. 掌握 DNS 服务器配置
3. 理解 DNS 服务器解析过程

## 二、设计原理

在 Internet 上，计算机之间的 TCP/IP 通信是通过 IP 地址来进行的。而 IP 地址通常采用点分十进制的数字表示，这给记忆 IP 地址带来了难度。在网络中每一台计算机（主机）都独立分配了 IP 地址，它们可以通过 IP 地址找到终端并与之通信。但是，当网络的规模较大时，使用 IP 地址就不太方便了，所以，便出现主机名（Host Name）与 IP 地址之间的一种对应解决方案，可以通过使用形象易记的主机名而非 IP 地址进行网络访问，这比单纯使用 IP 地址显然方便得多。DNS（Domain Name System）即域名系统，使网络中的客户端可以使用友好的名称来访问 Internet 或局域网中的计算机。其实，这种解决方案中使用了解析的概念和原理，单独通过主机名是无法建立网络连接的，只有通过 DNS 服务器的解析过程，在主机名与 IP 地址之间建立了映射关系后，才可以通过主机名间接地通过 IP 地址建立网络连接。DNS 的工作任务是域名与 IP 地址之间进行映射

## 三、设计内容



图 1 DNS 服务器配置

一台交换机一台主机一台服务器，做以下配置

- 1. IP 地址：192.168.6.1，192.168.6.7，192.168.6.8，192.168.6.9
- 2. 默认网关：192.168.6.1
- 3. 首选 DNS 服务器：192.168.6.1

将域名与 IP 做地址映射

- admin.abc.com → 192.168.6.8
- www.abc.com → 192.168.6.1
- ftp.abc.com → 192.168.6.9
- abc.com → 192.168.6.1
- www.bbc.com → 192.168.6.7

测试 ping www.abc.com

## 四、实验步骤

### 4.1 配置主机

IP配置

☐ DHCP

☒ 静态

IPv4 Address	192.168.6.10
子网掩码	255.255.255.0
默认网关	192.168.6.1
DNS服务器	192.168.6.1

图 2 配置主机信息

配置主机的 IP 地址和 DNS 服务器.

4.2 配置 DNS 服务器

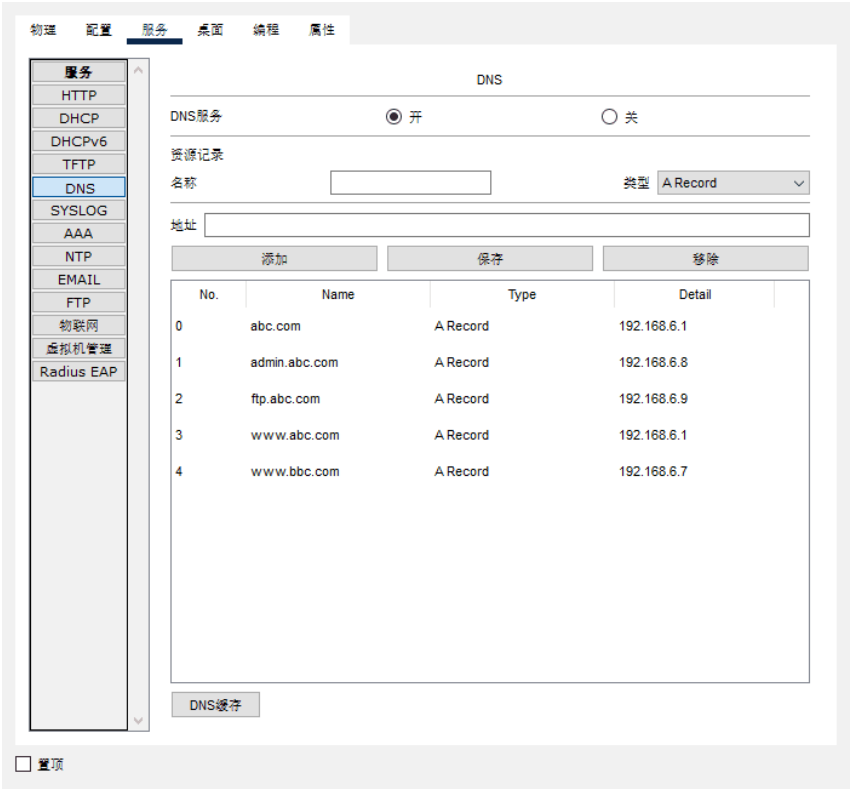


图 3 DNS 服务

按照实验内容配置 DNS 服务器对于的域名解析.

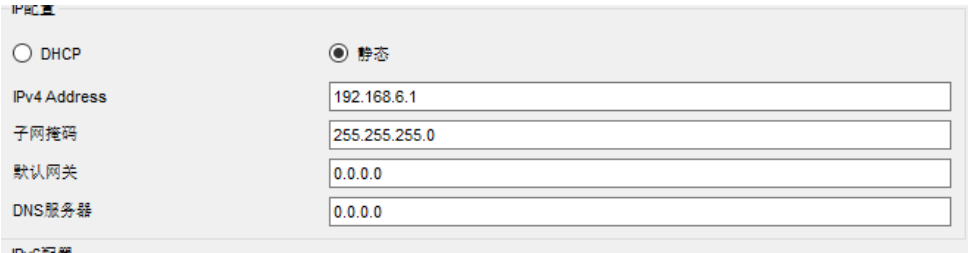


图 4 服务器 IP 配置

修改 DNS 服务器的 HTTP 协议放回的网页内容增加一行 <br> ABC IS OK, 如果在返回的网页中出现 ABC IS OK 代表返回正确.

4.3 测试配置是否成功

```
1 C:\>ping www.abc.com
2
3 Pinging 192.168.6.1 with 32 bytes of data:
4
```

```
5      Reply from 192.168.6.1: bytes=32 time<1ms TTL=128
6      Reply from 192.168.6.1: bytes=32 time<1ms TTL=128
7      Reply from 192.168.6.1: bytes=32 time<1ms TTL=128
8      Reply from 192.168.6.1: bytes=32 time<1ms TTL=128
9
10     Ping statistics for 192.168.6.1:
11         Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
12     Approximate round trip times in milli-seconds:
13         Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

通过 PING 命令尝试域名是否解析成功,发现可以 ping 通,通过浏览器访问 [www.abc.com](http://www.abc.com) 得到如下结果,图 (5) 中出现 ABC IS OK 说明能够正确访问。

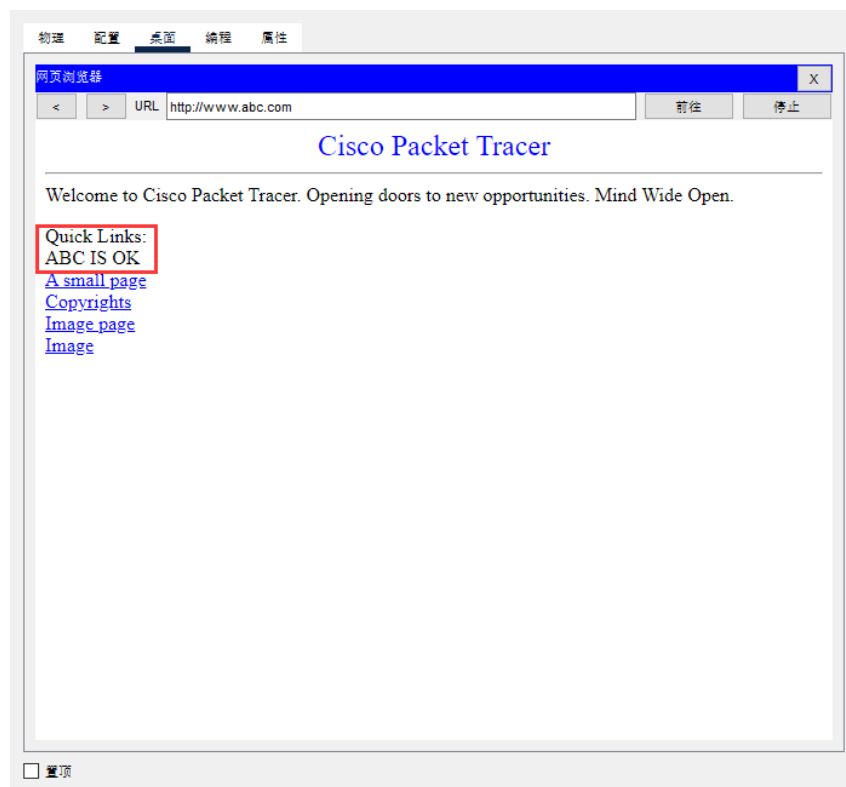


图 5 访问 [www.abc.com](http://www.abc.com)

## Part VII

# WWW 服务器设计

## 一、设计目的

1. 理解 www 服务原理
2. 掌握统一资源定位符 URL 的格式和使用
3. 理解超文本传送协议 HTTP 和超文本标记语言
4. 掌握 Web 站点的创建和配置

## 二、设计原理

1. 得到连接中的 URL 和端口号
2. 浏览器向 DNS 请求解析 www.edu.cn 的 IP 地址
3. DNS 将解析出的 IP 地址返回客户机
4. 客户机与服务器建立 TCP 连接 (80 端)
5. 客户机请求文档: GET .shtml(HTTP/1.0 指出 Web 浏览器使用的 HTTP 版本。)
6. 服务器给出响应, 将文档 index.shtml 发送给浏览器
7. 释放 TCP 连接
8. 显示 index.shtml 中的内容

## 三、设计内容

1. 将 www.abc.com 指向 192.168.6.1, 要求在浏览器中输入此域名就能调出 “abc” 服务器下的网页文件。
2. 将 www.bbc.com 指向 192.168.6.7, 要求在浏览器中输入此域名就能调出 “bbc” 服务器下的网页文件。

基于 DNS 服务器的配置基础上增加一个 Web 服务器, 地址为 192.168.6.7 并修改 HTTP 的网页内容.

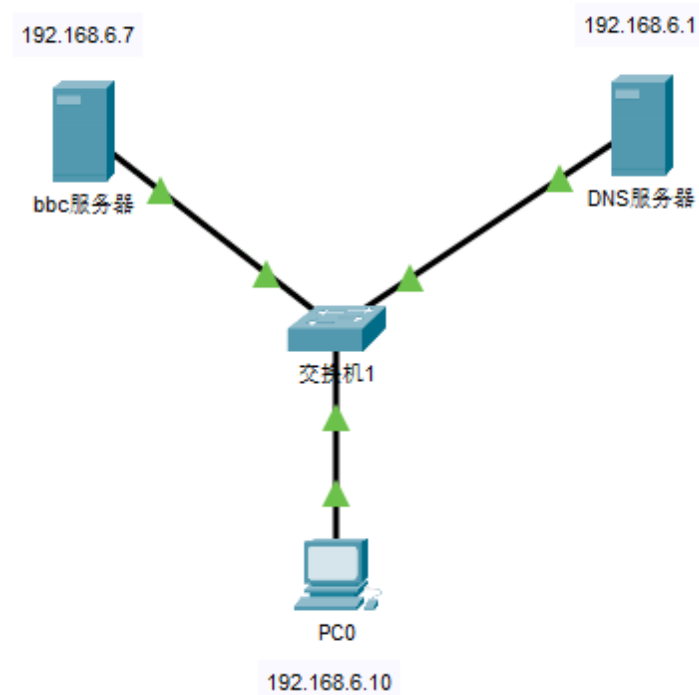


图 1 WEB 服务器配置

## 四、实验步骤

### 4.1 配置 WEB 服务器

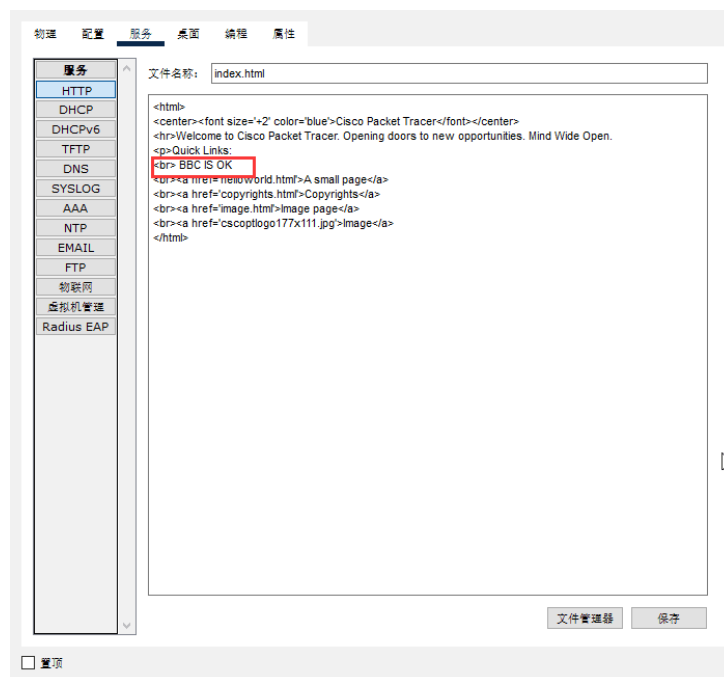


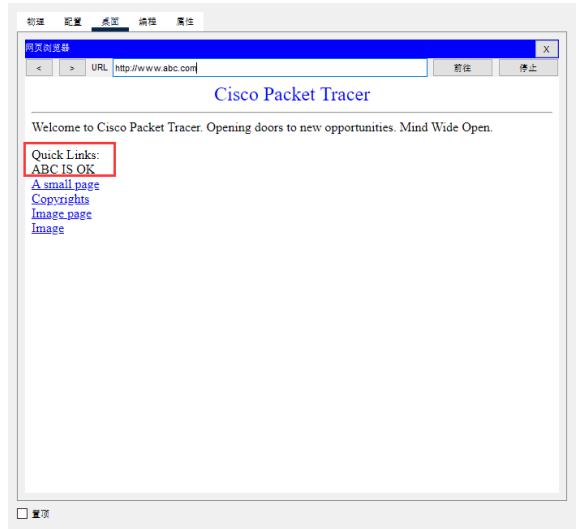
图 2 HTTP 配置

将 HTTP 配置的 index.html 页面增加一行 BBC IS OK 用来区分不同服务器返回的页面。

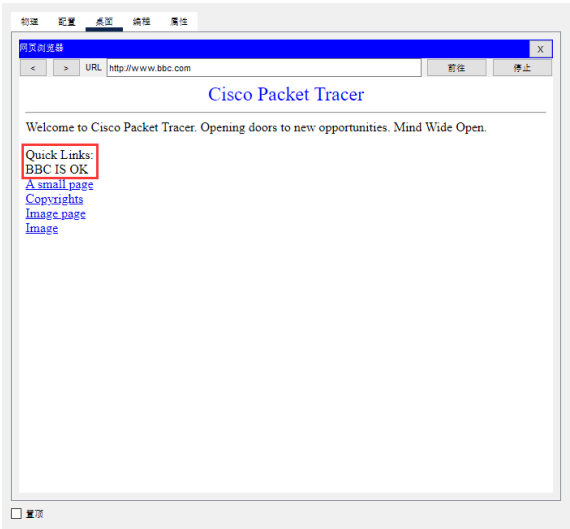
IPv4 Address	192.168.6.7
子网掩码	255.255.255.0
默认网关	0.0.0.0
DNS服务器	0.0.0.0

图 3 服务器 IP 配置

4.2 测试配置是否成功



(a) www.abc.com



(b) www.bbc.com

# FTP 服务器设计

## 一、设计目的

1. 掌握 FTP 的原理
2. 掌握配置 FTP 服务器
3. 掌握 FTP 命令行的使用

## 二、设计原理

FTP 是一种文件传输协议，它支持两种模式，一种方式叫做 Standard (也就是 Active, 主动方式)，一种是 Passive (也就是 PASV, 被动方式)。Standard 模式 FTP 的客户端发送 PORT 命令到 FTP server。Passive 模式 FTP 的客户端发送 PASV 命令到 FTP Server。

### 1. Standard 模式

FTP 客户端首先和 FTP Server 的 TCP 21 端口建立连接，通过这个通道发送命令，客户端需要接收数据的时候在这个通道上发送 PORT 命令。PORT 命令包含了客户端用什么端口接收数据。在传送数据的时候，服务器端通过自己的 TCP 20 端口发送数据。FTP server 必须和客户端建立一个新的连接用来传送数据。

### 2. Passive 模式

在建立控制通道的时候和 Standard 模式类似，当客户端通过这个通道发送 PASV 命令的时候，FTP server 打开一个位于 1024 和 5000 之间的随机端口并且通知客户端在这个端口上传送数据的请求，然后 FTP server 将通过这个端口进行数据的传送，这个时候 FTP server 不再需要建立一个新的和客户端之间的连接。

## 三、设计内容

1. 将 ftp.abc.com 指向 192.168.6.9，要求在浏览器中输入此域名就可登录到 FTP，使用 FTP 相关服务。
2. 使用不同的 TCP 值或不同的 IP 地址建立 FTP 站点。
3. 设置虚拟目录，规划不同用户的目录权限。



IPv4 Address	192.168.6.9
子网掩码	255.255.255.0
默认网关	0.0.0.0
DNS服务器	0.0.0.0

图 2 FTP 服务器地址

四、实验步骤

4.1 配置 FTP 服务器

FTP

服务 ☒ 开 ☐ 关

用户设置

用户名

密码

☐ 写 ☐ 读 ☐ 删除 ☐ 重命名 ☐ 列表

	用户名	密码	权限
1	admin	1	RWDNL
2	cisco	cisco	RWDNL
3	down	1	R
4	up	1	W

添加

保存

移除

图 1 配置 FTP 协议

增加 UP 用户给写的权限，增加 Down 用户给读的权限, 增加 admin 用户给所有权限.

给 FTP 服务器配置 ftp.abc.com 对于的 IP 地址 192.168.6.9.

4.2 测试配置是否成功

```
1 C:\>ftp ftp.abc.com
2 Trying to connect...ftp.abc.com
3 Connected to ftp.abc.com
4 220- Welcome to PT Ftp server
5 Username:up
```

```
6      331- Username ok, need password
7      Password:
8      230- Logged in
9      (passive mode On)
10     ftp>?
11         ?
12         cd
13         delete
14         dir
15         get
16         help
17         passive
18         put
19         pwd
20         quit
21         rename
22     ftp>
```

可以看到成功登录 ftp 服务器

## 五、实验总结

通过配置 DNS 服务器可以使得主机通过域名范围 IP 地址, 通过配置 WWW 服务器可以使得资源保存以便于访问, 通过配置 FTP 服务器可以使得用户下载和上传资源.

## Part IX

# 综合实验

### 一、设计任务

某工厂园区网有：2 个分厂（分别是：零件分厂、总装分厂）+1 个总厂网络中心 + 1 个总厂会议室；

1. 每个分厂有自己的路由器，均各有：1 个楼宇 + 分厂网络中心
  - (a) 每个楼宇均包含：20 台工作计算机（具体图中可以画 2 台计算机示意就可，可采用 DHCP 自动获取 IP 地址）
  - (b) 每个分厂网络中心均有 1 台服务器，上面启动了：WWW 服务、FTP 服务
2. 总厂网络中心有自己的路由器，有：WWW 服务器 1 台、DNS 服务器 1 台
3. 总厂会议室有 2 台计算机，用 WIFI 无线接入总厂网络中心

**请在模拟器中完成**

1. 使用静态路由和动态路由分别实现工厂内任意计算机之间的通信（使用 PING 测试）
2. 构建所有服务器，使用 IP 和域名对服务器进行访问
3. 使用访问控制规则，实现每个分厂内部的 WWW 服务可以被其他分厂访问，但是分厂内部的 FTP 服务不能被其他分厂访问
4. 若增加一个家属区网络，由于网络地址不够用，请使用 NAT 的方式配置路由器，使家属区网络可以正常访问工厂园区网的服务器



## 2.2 配置多层交换机

```
1 SW1(config)#vlan 10
2 SW1(config-vlan)#name VLAN10
3 SW1(config-vlan)#exit
4 SW1(config)#vlan 20
5 SW1(config-vlan)#name VLAN20
6 SW1(config-vlan)#exit
7 SW1(config)#vlan 30
8 SW1(config-vlan)#name VLAN30
9 SW1(config-vlan)#exit
10 SW1(config)#int fa0/1
11 SW1(config-if)#switch access vlan 10
12 SW1(config-if)#exit
13 SW1(config)#int fa0/2
14 SW1(config-if)#switch access vlan 20
15 SW1(config-if)#exit
16 SW1(config)#int fa0/3
17 SW1(config-if)#switch access vlan 30
18 SW1(config-if)#exit
```

划分 VLAN10,VLAN20,VLAN30, 将端口 fa0/1 绑定到 VLAN10, 将端口 fa0/2 绑定到 VLAN20, 将端口 fa0/3 绑定到 VLAN30.

```
1 SW1(config)#int vlan 10
2 %LINK-5-CHANGED: Interface Vlan10, changed state to up
3 SW1(config-if)#ip add 192.168.0.253 255.255.255.0
4 SW1(config-if)#standby ip 192.168.0.250
5 SW1(config-if)#st
6 SW1(config-if)#standby preempt
```

为 vlan 配置 IP 地址和虚拟路由器端口,preempt 代表开启抢占模式, 其中虚拟路由器端口用来作为零件分厂的网关地址. 按照如上操作配置 vlan20 和 vlan30, 配置完成后用命令 `do show run` 来查看配置是否成功

```
1 interface Vlan10
2 mac-address 0060.70a4.3201
3 ip address 192.168.0.253 255.255.255.0
4 standby 0 ip 192.168.0.250
5 standby 0 preempt
6 !
7 interface Vlan20
8 mac-address 0060.70a4.3202
9 no ip address
10 standby 0 ip 192.168.1.250
11 standby 0 preempt
12 !
13 interface Vlan30
```

```
14      mac-address 0060.70a4.3203
15      ip address 192.168.2.253 255.255.255.0
16      standby 0 ip 192.168.2.250
17      !
```

可以看到已经配置成功.

接下来配置 Core\_sw,Core\_sw 交换机作为核心层连接外网同时也连接汇集层的两台交换机. 通 no switch 命令配置接口的 IP 地址.

```
1      Switch(config)#interface FastEthernet0/1
2      Switch(config-if)#no switch
3      Switch(config-if)#ip address 10.10.0.2 255.255.255.0
4      Switch(config-if)#exit
5      Switch(config)#interface FastEthernet0/2
6      Switch(config-if)#no switch
7      Switch(config-if)#ip address 10.10.1.2 255.255.255.0
8      Switch(config-if)#int fa0/3
9      Switch(config-if)#ip address 10.0.0.1 255.255.255.252
10     Switch(config-if)#exit
11     Switch(config)#interface FastEthernet0/3
12     Switch(config-if)#no switch
13     Switch(config-if)#ip address 10.0.0.1 255.255.255.252
14     Switch(config-if)#exit
```

交换机配置完成, 对于多层交换机 SW2 按照 SW1 的配置重复一遍即可.

```
1      vlan 10
2      int vlan 10  进入vlan1的逻辑接口（不是物理接口，用来给vlan做路由用）
3      ip add 192.168.0.253 255.255.255.0 //配置IP地址和子网掩码
4      standby ip 192.168.0.250 %虚拟路由器端口为192.168.0.250
5      standby preempt //preempt是用来开启抢占模式
6      int f0/1
7      port li
8      switchport access vlan 10
9
10
11     do show run
```

## 三、使用 OSPF 使主机间 ping 通

### 3.1 配置网关静态路由

家属区想要访问工厂内的网络需要对路由器进行配置告诉路由器到达目的网路的下一跳应该怎么走,对于工厂内的三个网段零件工厂:192.168.0.0/24,总装分厂:192.168.1.0/24,总厂分厂: 192.168.2.0/24,可以通过路由聚合简化为 192.168.0.0/16 对网关路由进行配置.

**静态路由**

网络	<input type="text" value="192.168.0.0"/>
掩码	<input type="text" value="255.255.0.0"/>
下一跳	<input type="text" value="10.0.0.1"/>
<input type="button" value="添加"/>	

图 3 静态路由配置

## 3.2 配置 OSPF 协议

```
1 SW1(config)#int fa0/4
2 SW1(config-if)#router ospf 1
3 SW1(config-router)net 10.10.0.0 0.0.0.255 area 0
4 SW1(config-router)#net 192.168.0.0 0.0.0.255 area 0
5 SW1(config-router)#net 192.168.1.0 0.0.0.255 area 0
6 SW1(config-router)#net 192.168.2.0 0.0.0.255 area 0
7 SW1(config-router)#passive-interface vlan 10
8 SW1(config-router)#passive-interface vlan 20
9 SW1(config-router)#passive-interface vlan 30
```

将连接的四个网络通告给交换机, 采用 `do show run` 命令查看配置

```
1 !
2 router ospf 1
3 log-adjacency-changes
4 passive-interface Vlan10
5 passive-interface Vlan20
6 passive-interface Vlan30
7 network 10.10.0.0 0.0.0.255 area 0
8 network 192.168.1.0 0.0.0.255 area 0
9 network 192.168.0.0 0.0.0.255 area 0
10 network 192.168.2.0 0.0.0.255 area 0
11 !
```

可以看到配置成功.

## 3.3 尝试主机间的通信

### 3.3.1 工厂内通信

用位于零件工厂的主机 1 去尝试与位于三个不同网络 (零件工厂, 总装工厂, 总长工厂) 进行通信, 查看结果如下, 发现再第一次 ping 其他网络主机时, 第一次会超时, 原因是请求到达目的主机所在主机时, 交换机的 ARP 表中并没有对应的 MAC 地址, 所以需要

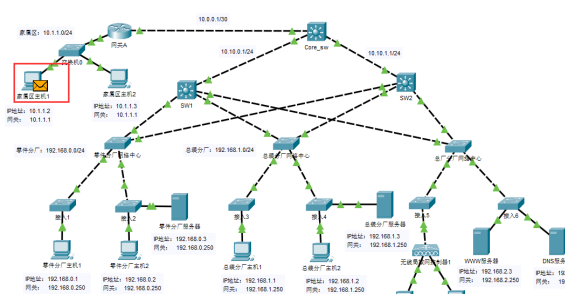
进行广播请求询问目的主机的 MAC 地址,再由目的主机进行单播告诉交换机 MAC 地址,这一过程会导致超时,但是再交换机缓存了主机的 MAC 地址后,后面的请求都会正常到达.

```
1 C:\>ping 192.168.0.2
2
3 Pinging 192.168.0.2 with 32 bytes of data:
4
5 Reply from 192.168.0.2: bytes=32 time=10ms TTL=128
6 Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
7 Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
8 Reply from 192.168.0.2: bytes=32 time=3ms TTL=128
9
10 Ping statistics for 192.168.0.2:
11     Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
12     Approximate round trip times in milli-seconds:
13         Minimum = 0ms, Maximum = 10ms, Average = 3ms
14
15 C:\>ping 192.168.1.1
16
17 Pinging 192.168.1.1 with 32 bytes of data:
18
19 Request timed out.
20 Reply from 192.168.1.1: bytes=32 time=21ms TTL=127
21 Reply from 192.168.1.1: bytes=32 time=10ms TTL=127
22 Reply from 192.168.1.1: bytes=32 time=10ms TTL=127
23
24 Ping statistics for 192.168.1.1:
25     Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
26     Approximate round trip times in milli-seconds:
27         Minimum = 10ms, Maximum = 21ms, Average = 13ms
28
29 C:\>ping 192.168.2.1
30
31 Pinging 192.168.2.1 with 32 bytes of data:
32
33 Request timed out.
34 Reply from 192.168.2.1: bytes=32 time=1ms TTL=127
35 Reply from 192.168.2.1: bytes=32 time=12ms TTL=127
36 Reply from 192.168.2.1: bytes=32 time=11ms TTL=127
37
38 Ping statistics for 192.168.2.1:
39     Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
40     Approximate round trip times in milli-seconds:
41         Minimum = 1ms, Maximum = 12ms, Average = 8ms
```

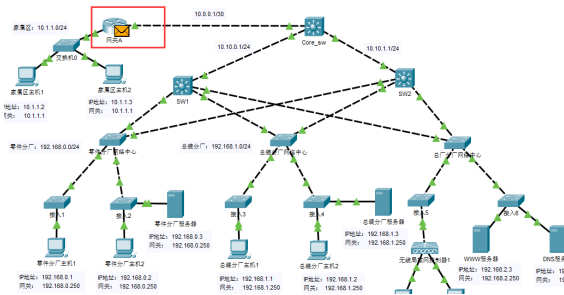
### 3.3.2 与家属区的通信

由于多层交换机直接与交换机相连接,所以在主机间的通信并不需要用到配置的 OSPF 协议,但是当主机与家属区进行通信的时候需要用到静态路由和动态路由.

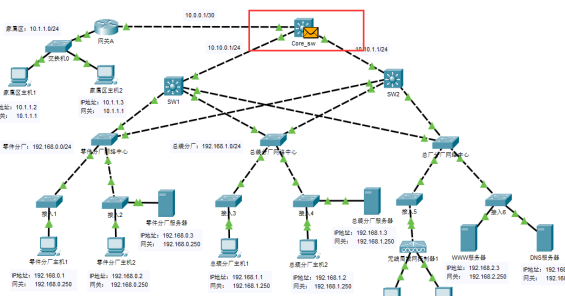




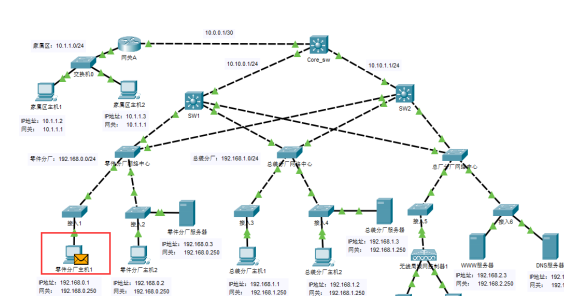
(a) 家属区主机 1 创建 PDU



(b) 家属区主机转发给网关



(c) 网关根据静态路由转发



(d) Core\_sw 根据 OSPF 转发

图 4 OSPF 链路选择

如图 (4) 所示, 首先创建了一个 PDU 在家属区主机 1 上, 家属区主机 1 判定目的地址与自己不在一个网络内于是转发到网关. PDU 到达网关后, 根据目的地址和自己的静态路由配置知道了下一跳的地址并转发到下一跳交换机 Core\_sw, 下一跳交换机 Core\_sw 根据路由表进行转发, 转发到交换机 SW1 或交换机 SW2. 到达 SW1 或者 SW2 后就可以更具 ARP 缓存到达目的主机.

类型	网络	端口	下一跳 IP	度量
S	0.0.0.0/0	---	10.0.0.2	1/0
C	1.1.1.1/32	Loopback0	---	0/0
C	10.0.0.0/30	FastEthernet0/3	---	0/0
C	10.10.0.0/24	FastEthernet0/1	---	0/0
C	10.10.1.0/24	FastEthernet0/2	---	0/0
O	192.168.0.0/24	FastEthernet0/1	10.10.0.1	110/2
O	192.168.0.0/24	FastEthernet0/2	10.10.1.1	110/2
O	192.168.1.0/24	FastEthernet0/1	10.10.0.1	110/2
O	192.168.1.0/24	FastEthernet0/2	10.10.1.1	110/2
O	192.168.2.0/24	FastEthernet0/1	10.10.0.1	110/2
O	192.168.2.0/24	FastEthernet0/2	10.10.1.1	110/2

图 5 Core\_sw 路由表

## 四、构建服务器

### 4.1 WWW 服务器配置

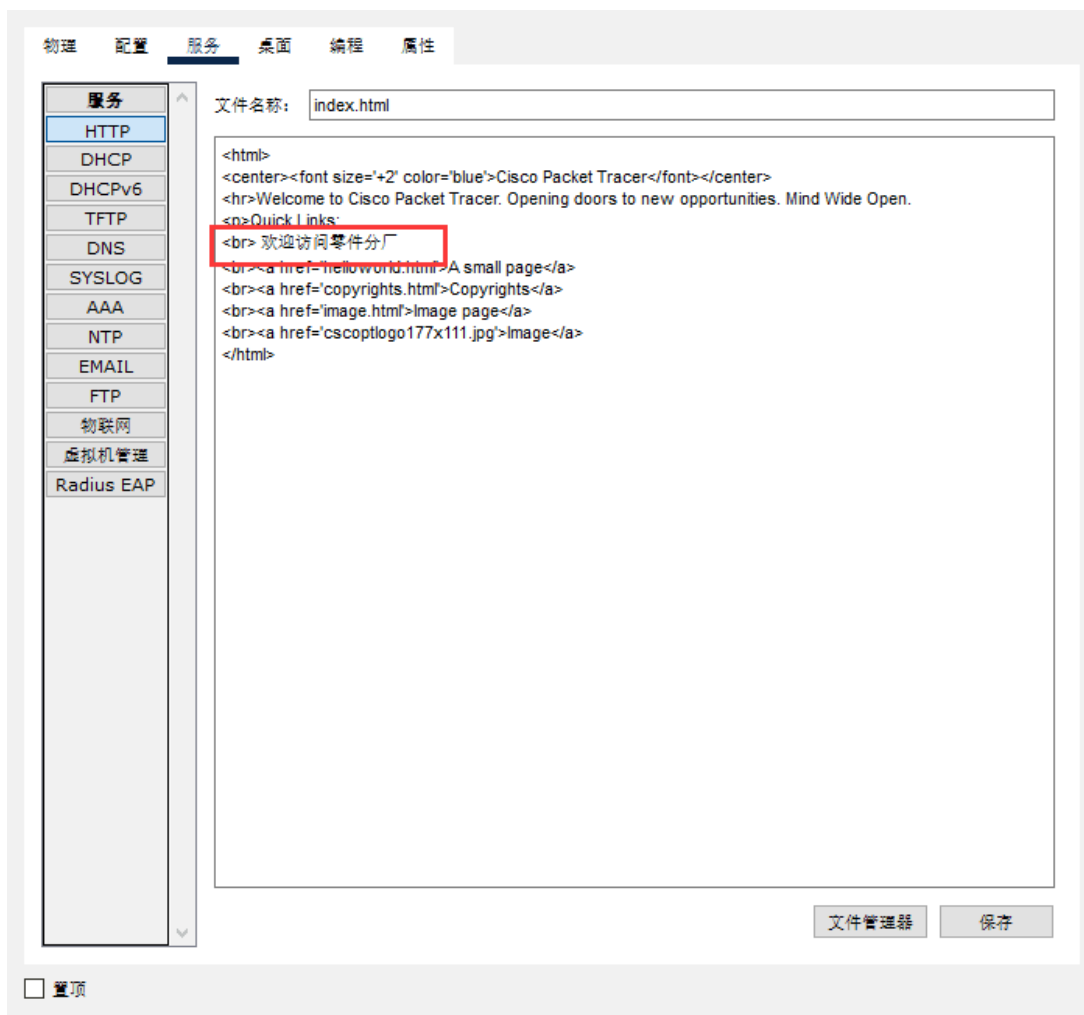


图 6 修改 HTML 内容

对于零件分厂,总装分厂,总厂分厂的服务器的 HTML 进行修改,对于零件工厂服务器如图 (6) 所示,增加一行欢迎访问零件工厂.对于总装分厂和总厂分厂服务器的 HTTP 也进行同样的配置.增加一行欢迎访问 XX 工厂.用来区分不同服务器的服务.

4.2 DNS 服务器的配置



图 7 DNS 服务器的配置

将零件分厂, 总装分厂, 总厂分厂的服务器配置 DNS 协议, 使得主机可以通过域名访问目的地址. 配置如上三条 DNS 记录.

4.3 构建 FTP 服务器



图 8 FTP 服务器

配置 FTP 服务器, 为 FTP 服务器增加两个用户 up 和 down, 同时还有自带的一个 Cisco 的超级用户.

## 4.4 尝试通过域名访问

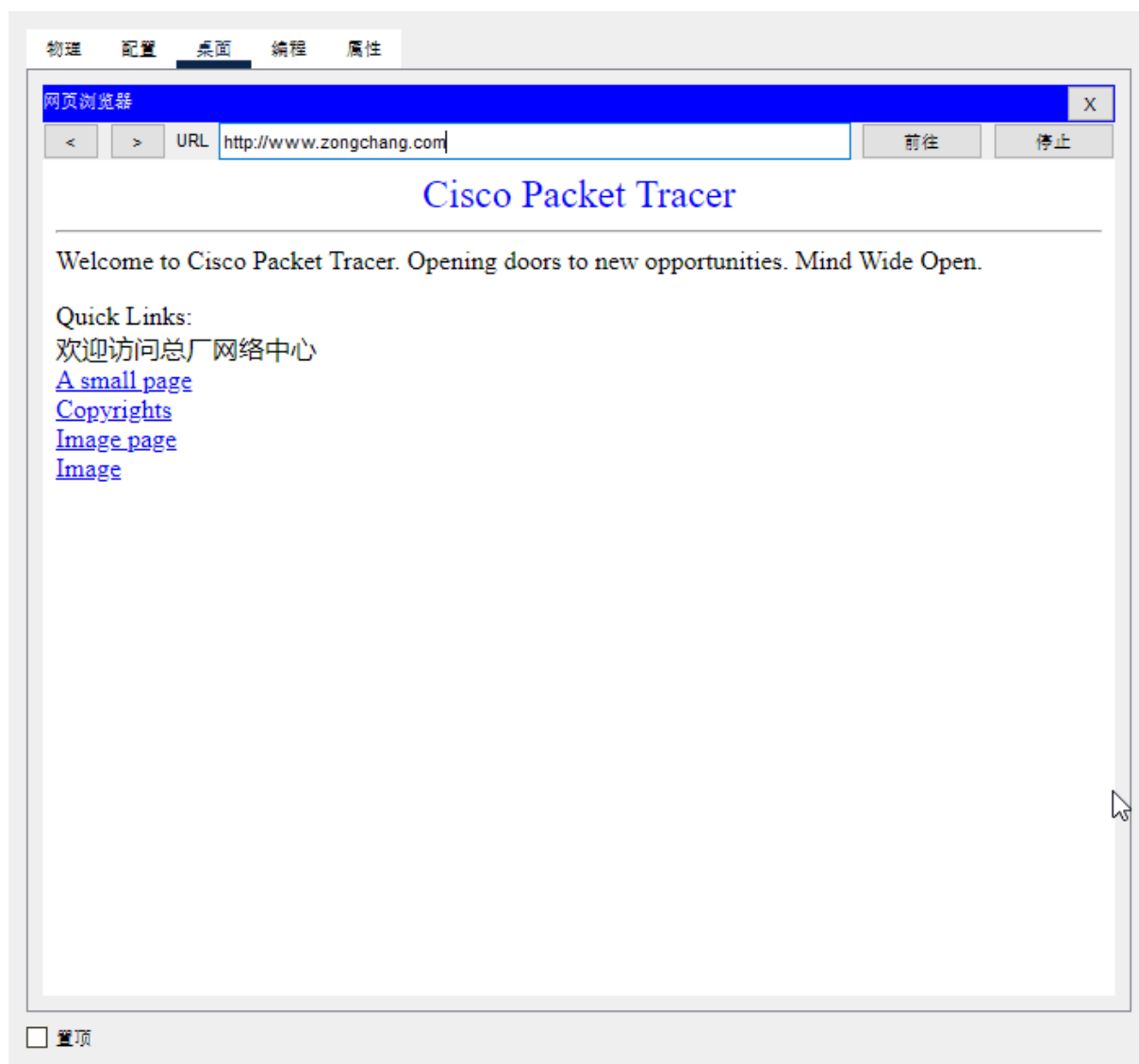


图 9 浏览器访问域名

通过在主机浏览器上输入 URL 来访问 WWW 服务器的 HTML 资源, 如图 (9) 所示, 访问 `www.zongchang.com` 域名时, 成功显示欢迎访问总厂分厂.

```
1 C:\>ftp www.lingjian.com
2 Trying to connect... www.lingjian.com
3 Connected to www.lingjian.com
4 220- Welcome to PT Ftp server
5 Username:up
6 331- Username ok, need password
7 Password:
8 230- Logged in
9 (passive mode On)
10 ftp>
```

尝试通过域名来使用 FTP 协议,发现配置成功,可以成功登录.

## 五、访问控制 FTP

通过配置 ACL 来禁止其他工厂主机访问本工厂服务器的 FTP 协议

### 5.1 配置 ACL

```
1 SW1(config)#access-list 100 deny tcp host 192.168.1.1 host 192.168.0.3 eq 21
2 SW1(config)#access-list 100 permit ip any any
3 SW1(config)#int vlan 10
4 SW1(config)#ip access 100 out
```

FTP 协议使用的是 21 端口,所以禁止其他主机访问服务器的 21 端口即可,并且允许其他的协议运行. 分别对交换机 SW1 和 SW2 的三个端口配置相应的 ACL 配置. 配置完成后通过 do show acc 命令来查看配置的 ACL 列表如下.

```
1 SW1(config)#do show acc
2 Extended IP access list 100
3     10 deny tcp any host 192.168.0.3 eq ftp
4     20 permit ip any any (8 match(es))
5 Extended IP access list 101
6     10 deny tcp any host 192.168.1.3 eq ftp (15 match(es))
7     20 permit ip any any (4 match(es))
```

通过上述两条访问控制列表可以禁止其他分厂的主机访问零件分厂和总装分厂服务器的 FTP 协议.

### 5.2 测试 ACL 配置是否成功

使用零件分厂主机 1 去访问总装分厂的服务器的 FTP 服务,结果如下,说明通过 ACL 控制已经禁止了零件分厂访问总装分厂服务器的 FTP 协议

```
1 C:\>ftp 192.168.1.3
2 Trying to connect...192.168.1.3
3
4 %Error opening ftp://192.168.1.3/ (Timed out)
5
6 (Disconnecting from ftp server)
```

使用总厂分厂会议室的主机去访问零件分厂服务器的 FTP 和总装分厂服务器的 FTP 结果如下,代表 ACL 配置成功.

```
1      Cisco Packet Tracer PC Command Line 1.0
2      C:\>ftp 192.168.0.3
3      Trying to connect ...192.168.0.3
4
5      %Error opening ftp://192.168.0.3/ (Timed out)
6
7      (Disconnecting from ftp server)
8
9      C:\>ftp 192.168.1.3
10     Trying to connect ...192.168.1.3
11
12     %Error opening ftp://192.168.1.3/ (Timed out)
13
14     (Disconnecting from ftp server)
```

## 六、路由器 NAT 配置

### 6.1 NAT 简介

NAT 将网络划分为内部网络和外部网络两部分，局域网主机利用 NAT 访问网络时，是将局域网内部的本地地址转换为全局地址（互联网合法的 IP 地址）后转发数据包；内部网络的所有主机均可共享一个合法外部 IP 地址实现对 Internet 的访问，从而可以最大限度地节约 IP 地址资源。同时，又可隐藏网络内部的所有主机，有效避免来自 Internet 的攻击。因此，目前网络中应用最多的就是端口多路复用方式。

### 6.2 配置路由器

```
1      Router(config)#interface GigabitEthernet0/0
2      Router(config-if)#ip address 10.0.0.2 255.255.255.252
3      Router(config-if)#ip nat outside
4      Router(config-if)#interface GigabitEthernet0/1
5      Router(config-if)#ip address 10.1.1.1 255.255.255.0
6      Router(config-if)#ip nat inside
7      Router(config-if)#ip nat inside source list 1 interface GigabitEthernet0/0 overload
8      Router(config)#ip route 192.168.0.0 255.255.255.0 10.0.0.1
9      Router(config)#ip route 192.168.1.0 255.255.255.0 10.0.0.1
10     Router(config)#ip route 192.168.2.0 255.255.255.0 10.0.0.1
11     Router(config)#access-list 1 permit 10.1.1.0 0.0.0.255
```

给端口 fa0/0 配置 IP 地址并且定义 fa0/0 为外部端口，给端口 fa0/1 配置 IP 地址并且定义 fa0/1 为内部端口。然后定义需要访问 Internet 的本地网络的访问列表，使用 NAT 需要绑定 ACL，但是 ACL 可以不用做任何访问控制。

### 6.3 NAT 过程分析

```
1    NAT: s=10.1.1.2->10.0.0.2 , d=192.168.0.1 [18]
2
3    NAT*: s=192.168.0.1 , d=10.0.0.2->10.1.1.2 [79]
4
5    NAT: s=10.1.1.2->10.0.0.2 , d=192.168.0.1 [19]
6
7    NAT*: s=192.168.0.1 , d=10.0.0.2->10.1.1.2 [80]
8
9    NAT: s=10.1.1.2->10.0.0.2 , d=192.168.0.1 [20]
10
11   NAT*: s=192.168.0.1 , d=10.0.0.2->10.1.1.2 [81]
12
13   NAT: s=10.1.1.2->10.0.0.2 , d=192.168.0.1 [21]
14
15   NAT*: s=192.168.0.1 , d=10.0.0.2->10.1.1.2 [82]
```

打开路由器和家属区主机 1, 使用家属区主机去 ping 地址为 192.168.0.1 的主机, 一次 ping 操作会发送四个 ICMP 报文, 目的主机还会对每个 ICMP 报文做出回答, 所以在这个过程中 NAT 一共转换了八次, 在家属区主机发送 ICMP 报文到目的地址的过程中, NAT 将源主机的 10.1.1.2 地址转化为 10.0.0.2 的外部网络地址, 在目的主机给源主机发送回答报文的时候 NAT 将 10.0.0.2 转化为源主机 10.1.1.2 的地址.

## 七、实验总结

这次综合实验遇到的最大问题是对三层交换机配置命令不太熟悉, 很多命令需要上网查找, 在配置访问控制列表的时候开始将访问控制列表绑定到端口上, 但是这样配置并没有使得访问控制生效, 在网上查找后, 发现由于划分了 VLAN 所以需要将访问控制列表绑定到 VLAN 上, 最后 NAT 的划分也是比较简单的, 虽然对于 NAT 的原理并没有很深刻的了解但是配置起来并不复杂, 通过 NAT 可以节约地址的使用.