

西安电子科技大学

B 级达标测试 课程实验报告

实验名称 数据网络综合设计实验

计算机科学与技术学院 1903051/011 班

姓名 徐静 学号 19030500043

同作者 梁婷婷 19030100094

实验日期 2022 年 5 月 10 日

成 绩

指导教师评语：

指导教师：

_____年____月____日

实验报告内容基本要求及参考格式

- 一、实验目的
- 二、实验所用仪器（或实验环境）
- 三、实验基本原理及步骤（或方案设计及理论计算）
- 四、实验数据记录（或仿真及软件设计）
- 五、实验结果分析及回答问题（或测试环境及测试结果）

一、实验背景

通用路由封装协议 (Generic Routing Encapsulation, GRE), 是对某些网络层协议 (如 IP 和 IPX) 的数据进行封装, 使这些被封装的数据报能在另一个网络层协议 (如 IP) 中传输。该协议最早是由思科提出的, 目前它已经成为了 1 种标准, 简单来说, GRE 是 VP (Virtual Private Network) 的第 3 层隧道协议, 即在协议层之间采用了 1 种被称之为 Tunnel (隧道) 的技术, GRE 就是利用隧道来从 1 个网络向另 1 个网络传输数据包。将通过隧道的报文用 1 个新的报文头 GRE 报文头) 进行封装, 然后带着隧道终点地址放入隧道中。当报文到达隧道的终点时, GRE 报文头被剥离, 再用原始报文的目的地址进行寻址。GRE 隧道通常是点到点的。

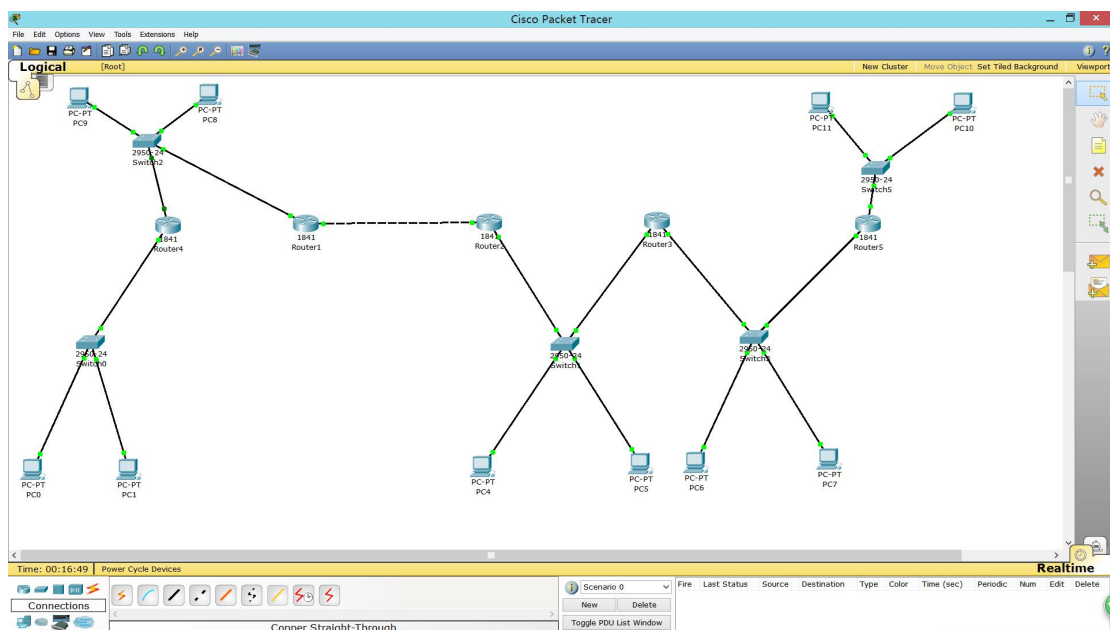
网络地址转换 (Network Address Translation, NAT) 技术提供了一种完全将内部网络和 Internet 网隔离的方法, 让内部网络中的计算机通过少数几个甚至一个合法 IP 地址 (已申请的一个公网 IP) 访问 Internet 资源, 从而节省了 IP 地址, 并得到广泛的应用。NAT 常见的三种类型: 静态转换, 动态转换, 端口多路复用。

动态主机设置协议 (Dynamic Host Configuration Protocol, DHCP) 是一个局域网的网络协议, 前身是 BOOTP 协议, 使用 UDP 协议工作, 常用的 2 个端口: 67 (DHCP server), 68 (DHCP client)。DHCP 通常被用于局域网环境, 主要作用是集中的管理、分配 IP 地址, 使 client 动态的获得 IP 地址、Gateway 地址、DNS 服务器地址等信息, 并能够提升地址的使用率。简单来说, DHCP 就是一个不需要账号密码登录的、自动给内网机器分配 IP 地址等信息的协议。

Packet Tracer 是 Cisco 公司为思科网络技术学院开发的一款模拟软件, 可以用来模拟 CCNA 的实验。Packet Tracer 模拟器的使用者可以在软件的图形用户界面上直接使用拖曳物件建立网络拓扑, 并可提供数据包在网络中行进の詳細处理过程, 观察网络实时运行情况。该软件以其方便性和真实性被广泛接受。

二、实验目的

利用仿真器 Packet Tracer, 实现如下图所示网络。



同时实现以下要求：

- 1、二层交换机的每个端口在不同 Vlan。
- 2、利用静态路由来配置路由。
- 3、要求用到 DHCP 技术来分配 IP 地址。
- 4、要求用到 NAT 技术完成地址变换。
- 5、要求用到单臂路由来实现互联。
- 6、要求用到 GRE 隧道技术。
- 7、要求用到访问控制列表技术。
- 8、任意两个节点之间能在规则下互相访问。

三、 实验环境

Cisco Packet Tracer 8.1.1

四、 实验步骤及结果

1. 配置二层交换机的每个端口在不同 Vlan

VLAN(Virtual Local Area Network,即“虚拟局域网”),VLAN 是一种将局域网(LAN)设备从逻辑上划分(不是从物理上划分)成一个个网段(或者说是更小的局域网 LAN),从而实现虚拟工作组(单元)的数据交换技术。

以图中交换机 0 和交换机 1 之间的互连为例:交换机 0 的 F0/0 口与路由器 0 连接,F0/23 口和 F0/24 口分别与两台 PC 相连。交换机 1 的 F0/1 口与路由器 0 连接, F0/23 口和 F0/24 口分别与另外两台 PC 相连。交换机与路由器相连的接口配置为 trunk 模式。trunk 是用来在不同的交换机之间进行连接,以保证在跨越多个交换机上建立的同一个 vlan 的成员能够相互通讯。

对交换机 0 配置如下:

1. 启用 ip routing
2. 将 F0/23 口划入 vlan43, F0/24 口划入 vlan94

交换机 0 的配置:

```
Switch(config)#vlan 43
Switch(config-vlan)#exit
Switch(config)#vlan 94
Switch(config-vlan)#exit

Switch(config)#interface fa 0/23
Switch(config-if)#switch access vlan 43
Switch(config-if)#exit

Switch(config)#interface fa 0/24
Switch(config-if)#switch access vlan 94
Switch(config-if)#exit
```

```
Switch(config)#interface fastEthernet 0/1
Switch(config-if)#switchport mode trunk    # 配置接口为trunk口, trunk口允许多个VLAN通过。
Switch(config-if)#switchport trunk allowed vlan 43,94    # 配置turnk口允许vlan43 和vlan94通过。

Switch#show vlan
```

对交换机 1 配置同理:

```
Switch(config)#vlan 43
Switch(config-vlan)#exit
Switch(config)#vlan 94
Switch(config-vlan)#exit

Switch(config)#interface fa 0/23
Switch(config-if)#switch access vlan 43
Switch(config-if)#exit

Switch(config)#interface fa 0/24
Switch(config-if)#switch access vlan 94
Switch(config-if)#exit


Switch(config)#interface fa 0/22
Switch(config-if)#switch access vlan 66
Switch(config-if)#exit

Switch(config)#interface fastEthernet 0/1
Switch(config-if)#switchport mode trunk    # 配置接口为trunk口, trunk口允许多个VLAN通过。
Switch(config-if)#switchport trunk allowed vlan 43,94    # 配置turnk口允许vlan43 和vlan94通过。

Switch#show vlan
```

完成了上面的命令后即在交换机 0 的两个端口下分别配置了两个 vlan。

实验结果:

 Switch0

| Physical | | | Config | CLI | Attributes |
|----------|--------------------|--------|---|-----|------------|
| VLAN | Name | Status | Ports | | |
| 1 | default | active | Fa0/2, Fa0/3, Fa0/5, Fa0/6 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 | | |
| 43 | VLAN0043 | active | Fa0/23 | | |
| 94 | VLAN0094 | active | Fa0/24 | | |
| 1002 | fddi-default | active | | | |
| 1003 | token-ring-default | active | | | |
| 1004 | fddinet-default | active | | | |
| 1005 | trnet-default | active | | | |

2. 配置单臂路由实现互联:

单臂路由 (router-on-a-stick) 是指在路由器的一个接口上通过配置子接口 (或 “逻辑接口”, 并不存在真正物理接口) 的方式, 实现原来相互隔离的不同 vlan 之间的互联互通。路由器的物理接口可以被划分成多个逻辑接口, 这些被划分后的逻辑接口被称为子接口。

在第一步的实验中, 我们已经把交换机的两个接口分配在了两个不同的 vlan 间。两个 vlan 的网络要通信, 必须通过路由器, 如果接入路由器的只有一个物理端口, 则必须有两个子接口分别与两个 vlan 对应。

对路由器 0 配置如下: R1 的接口 F0/0 上创建两个子接口, 分别是 F0/0.43 对应的 vlan43、F0/0.94 对应的 vlan94, 每个子接口必须封装 dot1Q 协议, 并且标记相应的 vlan id 号, dot1Q 协议主要是标记 vlan 的 id 号, 每个子接口必须配置 ip 地址, 而且该接口的 ip 地址必须和相应的 vlan 的在同一个网段。

路由器 0 的配置命令:

```
Router(config)#interface fastEthernet0/0.43
Router(config-subif)#encapsulation dot1Q 43
Router(config-subif)#ip address 192.168.43.13 255.255.255.224
Router(config-subif)#exit

Router(config)#interface fastEthernet0/0.94
Router(config-subif)#encapsulation dot1Q 94
Router(config-subif)#ip address 192.168.94.13 255.255.255.224
Router(config-subif)#exit

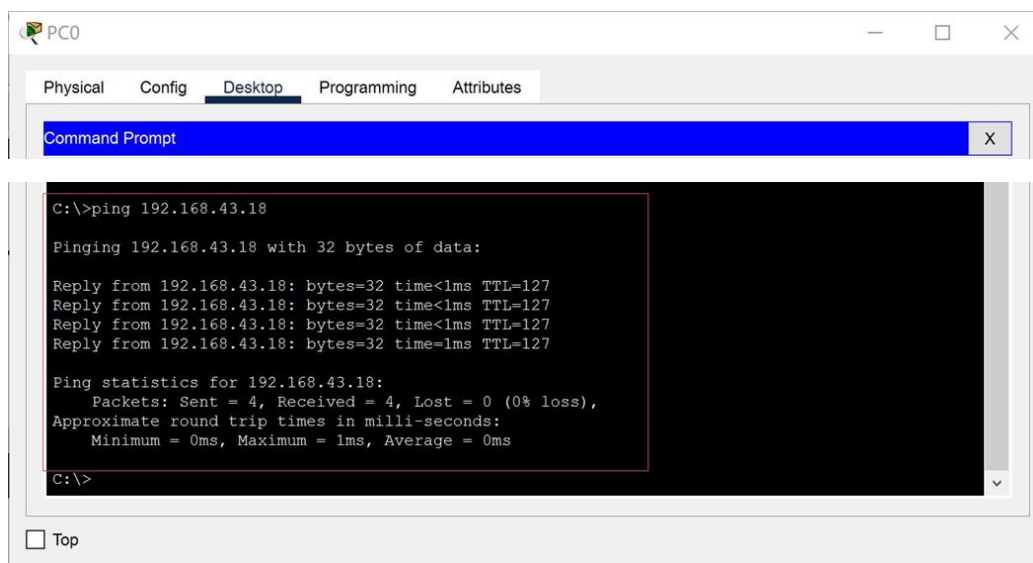
Router(config)#interface fastEthernet0/1.43
Router(config-subif)#encapsulation dot1Q 43
Router(config-subif)#ip address 192.168.43.60 255.255.255.224
Router(config-subif)#exit

Router(config)#interface fastEthernet0/1.94
Router(config-subif)#encapsulation dot1Q 94
Router(config-subif)#ip address 192.168.94.60 255.255.255.224
Router(config-subif)#exit
```

对 PC 进行手动配置: PC 的地址与路由器端口地址掩码保持一致, 网关设置为路由器的地址。完成以上配置后, 该网段内的 vlan43 和 vlan94 可以实现相互通信。

实验结果:

ping 测试通过:



3. 配置静态路由

对路由器的路由表进行设置，当收到的 ip 数据包的主机不在当前路由器的网络范围内时，向路由表中下一跳地址转发，完成路由。

配置设置以 Router3 为例，局域网 3 和局域网 4 都是 Router3 的直连网络，所以在 Router3 的路由配置中，这两个网络的数据包不需要转发。如果 Router3 收到发往局域网 1 和局域网 2 的数据包，将下一跳地址设置为 Router2，让 Router2 去寻找局域网 1 和局域网 2 下对应的主机；如果 Router3 收到发往局域网 5 的数据包，则转由 Router4 处理。

分析完网络分布情况以及如何转发的情况下，下面即可 Router3 进行配置。对于静态路由的配置，我们可以选择在路由器的 Config 界面图形化的配置，也可以继续在命令行中配置。

下面的 图片是在图形化界面中对 Router3 配置静态路由：

Static Routes

| | |
|----------|--|
| Network | |
| Mask | |
| Next Hop | |

| Network Address |
|--------------------------------------|
| 192.168.43.128/27 via 192.168.43.125 |
| 192.168.94.128/27 via 192.168.94.125 |
| 192.168.43.0/27 via 192.168.43.94 |
| 192.168.43.32/27 via 192.168.43.94 |
| 192.168.94.32/27 via 192.168.94.94 |
| 192.168.94.0/27 via 192.168.94.94 |

在图形化界面的三个方框中添加信息后, 点击 Add 按钮即可将对应的静态路由添加至路由表中。除了这种配置方式, 我们也可以继续在命令行中进行配置, 命令如下图所示, ip route 后面跟着的三个参数顺序与上图一致, 分别是目标网络号、目标网络子网掩码以及下一跳地址。

配置完成后, 可以通过在特权模式下通过命令 show ip route 查看路由信息:

实验结果:

```
sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

    192.168.43.0/27 is subnetted, 5 subnets
S       192.168.43.0 [1/0] via 192.168.43.94
S       192.168.43.32 [1/0] via 192.168.43.94
C       192.168.43.64 is directly connected, FastEthernet0/0.43
C       192.168.43.96 is directly connected, FastEthernet0/1.43
S       192.168.43.128 [1/0] via 192.168.43.125
    192.168.94.0/27 is subnetted, 5 subnets
S       192.168.94.0 [1/0] via 192.168.94.94
S       192.168.94.32 [1/0] via 192.168.94.94
C       192.168.94.64 is directly connected, FastEthernet0/0.94
C       192.168.94.96 is directly connected, FastEthernet0/1.94
S       192.168.94.128 [1/0] via 192.168.94.125
--More--
```

4. 配置 DHCP 来分配 IP 地址

配置 DHCP 的功能通过在路由器上创建 ip 地址池来实现。

DHCP 的配置步骤:

1. 启用 DHCP 服务 (此软件默认为打开状态)。
2. 建立地址池, 其标识符为自己喜欢的名字 (如 token)。下面的命令将对其设置。
3. 设置 DHCP 地址池 token 的网络号和掩码。分配地址时从中选择一个未用地址分配。
4. 设置客户端的默认网关。
5. 设置域名服务器。
6. 退出 net172 地址池的设置状态。

以 Router4 配置为例

```
Router(config)#ip dhcp pool dhcp1
Router(dhcp-config)#network 192.168.43.128 255.255.255.224
Router(dhcp-config)#default-router 192.168.43.158
Router(dhcp-config)#exit
Router(config)#ip dhcp pool dhcp2
Router(dhcp-config)#network 192.168.94.128 255.255.255.224
Router(dhcp-config)#default-router 192.168.94.158
Router(dhcp-config)#exit
```

router3 同理

```
Router(config)#ip dhcp pool dhcp1
Router(dhcp-config)#network 192.168.43.96 255.255.255.224
Router(dhcp-config)#default-router 192.168.43.126
Router(dhcp-config)#exit
Router(config)#ip dhcp pool dhcp2
Router(dhcp-config)#network 192.168.96.96 255.255.255.224
Router(dhcp-config)#network 192.168.94.96 255.255.255.224
Router(dhcp-config)#default-router 192.168.94.126
Router(dhcp-config)#exit
```

router2 同理

```
Router(config)#ip dhcp pool dhcp1
Router(dhcp-config)#network 192.168.43.64 255.255.255.224
Router(dhcp-config)#default-router 192.168.43.94
Router(dhcp-config)#exit
Router(config)#ip dhcp pool dhcp2
Router(dhcp-config)#network 192.168.94.64 255.255.255.224
Router(dhcp-config)#default-router 192.168.94.94
Router(dhcp-config)#exit
```

router0 同理

```
Router(config)#ip dhcp pool dhcp1
Router(dhcp-config)#network 192.168.43.32 255.255.255.224
Router(dhcp-config)#default-router 192.168.43.60
Router(dhcp-config)#exit
Router(config)#ip dhcp pool dhcp2
Router(dhcp-config)#network 192.168.94.32 255.255.255.224
Router(dhcp-config)#default-router 192.168.94.60
Router(dhcp-config)#exit
```


实验结果:

Router4

Physical Config **CLI** Attributes

IOS Command Line Interface

```
Router#
Router#show vlan
```

| VLAN | Name | Status | Ports |
|------|--------------------|--------|-------|
| 1 | default | active | |
| 1002 | fddi-default | active | |
| 1003 | token-ring-default | active | |
| 1004 | fddinet-default | active | |
| 1005 | trnet-default | active | |

| VLAN | Type | SAID | MTU | Parent | RingNo | BridgeNo | Stp | BrdgMode | Trans1 | Trans2 |
|------|-------|--------|------|--------|--------|----------|------|----------|--------|--------|
| 1 | enet | 100001 | 1500 | - | - | - | - | - | 0 | 0 |
| 1002 | fddi | 101002 | 1500 | - | - | - | - | - | 0 | 0 |
| 1003 | tr | 101003 | 1500 | - | - | - | - | - | 0 | 0 |
| 1004 | fdnet | 101004 | 1500 | - | - | - | ieee | - | 0 | 0 |
| 1005 | trnet | 101005 | 1500 | - | - | - | ibm | - | 0 | 0 |

| VLAN | Type | SAID | MTU | Parent | RingNo | BridgeNo | Stp | BrdgMode | Trans1 | Trans2 |
|------|------|------|-----|--------|--------|----------|-----|----------|--------|--------|
|------|------|------|-----|--------|--------|----------|-----|----------|--------|--------|

Remote SPAN VLANs

| Primary | Secondary | Type | Ports |
|---------|-----------|------|-------|
|---------|-----------|------|-------|

Router4

Physical Config **CLI** Attributes

IOS Command Line Interface

```
S 192.168.94.0 [1/0] via 192.168.94.126
S 192.168.94.32 [1/0] via 192.168.94.126
S 192.168.94.64 [1/0] via 192.168.94.126
C 192.168.94.96 is directly connected, FastEthernet0/0.94

Router#show ip dhcp pool
```

Pool dhcp1 :

Utilization mark (high/low) : 100 / 0
Subnet size (first/next) : 0 / 0
Total addresses : 30
Leased addresses : 1
Excluded addresses : 0
Pending event : none

1 subnet is currently in the pool

| Current index | IP address range | Leased/Excluded/Total |
|----------------|---------------------------------|-----------------------|
| 192.168.43.129 | 192.168.43.129 - 192.168.43.158 | 1 / 0 / 30 |

Pool dhcp2 :

Utilization mark (high/low) : 100 / 0
Subnet size (first/next) : 0 / 0
Total addresses : 30
Leased addresses : 1
Excluded addresses : 0
Pending event : none

1 subnet is currently in the pool

| Current index | IP address range | Leased/Excluded/Total |
|----------------|---------------------------------|-----------------------|
| 192.168.94.129 | 192.168.94.129 - 192.168.94.158 | 1 / 0 / 30 |

5. 运用 NAT 技术完成地址变换

NAT 就是在局域网内部网络中使用内部地址，而当内部节点要与外部网络进行通讯时，在网关处将内部地址替换成公用地址，从而在外部公网上正常使用。通过该技术，让内部网络中的计算机通过少数几个甚至一个合法 IP 地址（已申请的一个公网 IP）访问 Internet 资源，从而节省 IP 地址。

静态地址转换配置步骤：

1. 在路由器上配置 IP 地址和 IP 路由；
2. 配置静态地址转换。全局配置模式下，使用如下格式命令：“ip nat inside source static 内部专用地址 内部合法地址”。其中，内部专用地址为内部网络的私有地址，内部合法地址为向因特网管理机构申请到的全球合法地址。
3. 进入接口配置模式，启用 NAT。命令格式为：“ip nat inside/outside”。其中，内网接口使用 inside，外部接口使用 outside。

下图是在 Router4 上的配置过程：

```
Router(config)#int fa0/1.43
Router(config-subif)#ip nat inside
Router(config-subif)#exit
Router(config)#int fa0/0.43
Router(config-subif)#ip nat outside
Router(config-subif)#exit
Router(config)#ip nat inside source static 192.168.43.129 192.168.43.100
```

完成如上 NAT 地址转换配置后，PC7 的内部地址 192.168.43.129 与外部地址 192.168.43.100 形成映射关系。

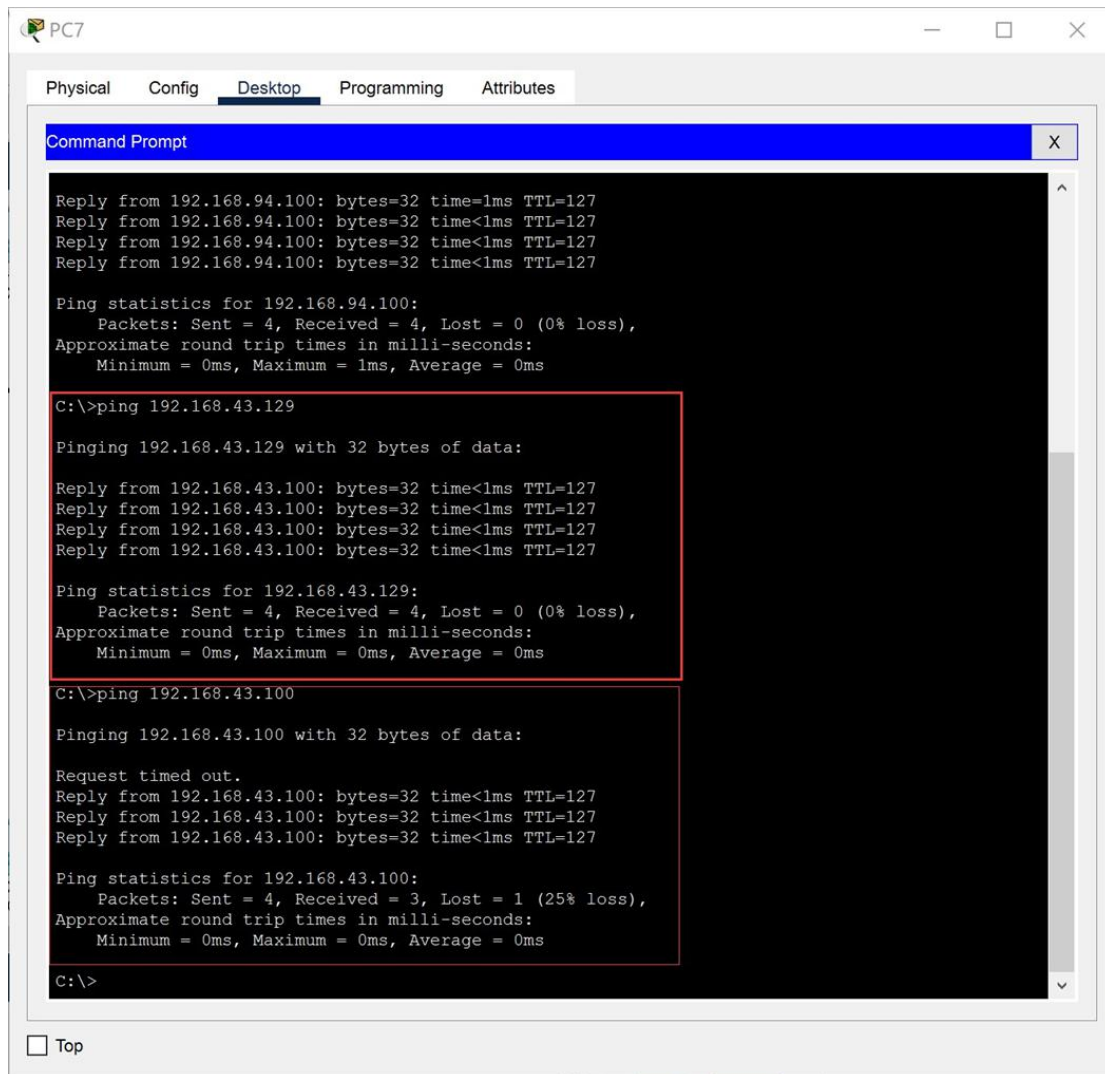
实验结果：



The screenshot shows the CLI of Router4. The top part displays the routing table with entries for 192.168.94.0/24 and 192.168.94.32/24. The bottom part shows the output of the command 'show ip nat translations', which displays a mapping between the internal address 192.168.43.100 and the external address 192.168.43.129. Below this, the output of 'show ip access-lists' is shown, indicating that the access list is empty.

```
Router#show ip nat translations
Pro Inside global Inside local Outside local Outside global
--- 192.168.43.100 192.168.43.129 ---
--- 192.168.94.100 192.168.94.129 ---

Router#show ip access-lists
Router#
```



The screenshot shows a PC7 virtual machine window with tabs for Physical, Config, Desktop, Programming, and Attributes. The Desktop tab is active, displaying a Command Prompt window. The Command Prompt shows the results of several ping commands. The first set of pings is to 192.168.94.100, showing successful results with 0% loss. The second set of pings is to 192.168.43.129, also showing successful results with 0% loss. The third set of pings is to 192.168.43.100, showing a 25% loss (1 packet lost) and a request timed out. The Command Prompt window has a red box highlighting the ping results for 192.168.43.129 and 192.168.43.100.

```
PC7
Physical Config Desktop Programming Attributes
Command Prompt
Reply from 192.168.94.100: bytes=32 time<1ms TTL=127
Reply from 192.168.94.100: bytes=32 time<1ms TTL=127
Reply from 192.168.94.100: bytes=32 time<1ms TTL=127
Reply from 192.168.94.100: bytes=32 time<1ms TTL=127
Ping statistics for 192.168.94.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
C:\>ping 192.168.43.129
Pinging 192.168.43.129 with 32 bytes of data:
Reply from 192.168.43.100: bytes=32 time<1ms TTL=127
Reply from 192.168.43.100: bytes=32 time<1ms TTL=127
Reply from 192.168.43.100: bytes=32 time<1ms TTL=127
Reply from 192.168.43.100: bytes=32 time<1ms TTL=127
Ping statistics for 192.168.43.129:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>ping 192.168.43.100
Pinging 192.168.43.100 with 32 bytes of data:
Request timed out.
Reply from 192.168.43.100: bytes=32 time<1ms TTL=127
Reply from 192.168.43.100: bytes=32 time<1ms TTL=127
Reply from 192.168.43.100: bytes=32 time<1ms TTL=127
Ping statistics for 192.168.43.100:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>
```

6. 运用 GRE 隧道技术

通用路由封装协议 (Generic Routing Encapsulation, GRE), 其根本功能就是要实现隧道功能, 对某些网络层协议 (如 IP 和 IPX) 的数据进行封装, 使这些被封装的数据包能够在另一个网络层协议 (如 IP) 中传输。通过隧道连接的两个远程网络就如同直连, GRE 在两个远程网络之间模拟出直连链路, 从而使网络间达到直连的效果。

该技术将通过隧道的报文用 1 个新的报文头 (GRE 报文头) 进行封装, 然后带着隧道终点地址放入隧道中。当报文到达隧道的终点时, GRE 报文头被剥离, 再用原始报文的目的地地址进行寻址。

隧道传递数据包的过程分为 3 步:

1. 接收原始 IP 数据包当作乘客协议, 原始 IP 数据包包头的 IP 地址为私有 IP 地址。
2. 将原始 IP 数据包封装进 GRE 协议, GRE 协议称为封装协议 (Encapsulation Protocol), 封装的包头 IP 地址为虚拟直连链路两端的 IP 地址。
3. 将整个 GRE 数据包当作数据, 在外层封装公网 IP 包头, 也就是隧道的起源和终点, 从而路由到隧道终点。

实验中我们在 Router2 和 Router4 之间搭建一条隧道, 过程如下:

以 Router2 中的命令为例：

1. interface tunnel 0 命令创建了 tunnel 接口，编号为 0；
2. ip address 192.168.3.1 255.255.255.0 命令设置隧道接口上的 ip 地址，创建隧道后，可以把隧道看成一条专线，显然该地址应该是私网地址；
3. tunnel source fa0/1.43 命令指定了 tunnel 的源接口，路由器将以此接口的地址作为源地址重新封装数据包，也可以直接输入接口的地址；
4. tunnel destination 192.168.43.125 命令是 tunnel 的目的地址，路由器将以此地址作为目的地址重新封装数据包。

隧道配置如下：

Router4：

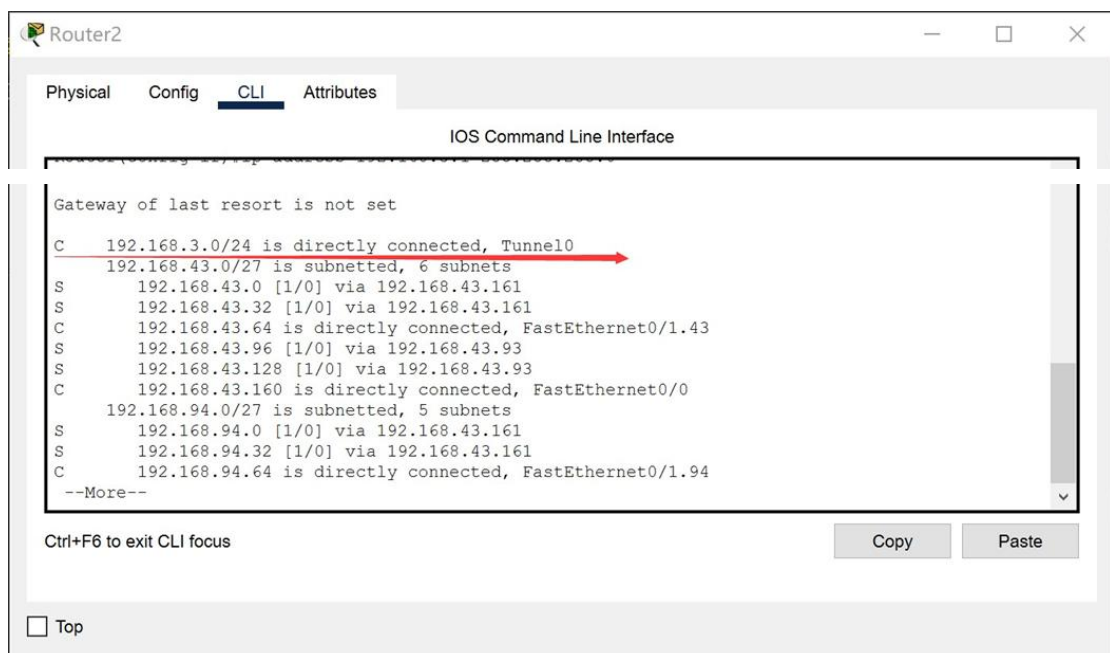
```
Router(config)#interface tunnel 0
Router(config-if)#ip add 192.168.3.2 255.255.255.0
Router(config-if)#tunnel source fa0/0.43
Router(config-if)#tunnel destination 192.168.43.94
```

Router2：

```
Router(config)#interface tunnel 0
Router(config-if)#ip address 192.168.3.1 255.255.255.0
Router(config-if)#tunnel source fa0/1.43
Router(config-if)#tunnel destination 192.168.43.125
```

配置完成后，查看 Router2 的路由表检验配置结果。

实验结果：



7. 访问控制列表技术

ACL (Access Control List, 访问控制列表) 是一系列运用到路由器接口的指令列表。这些指令告诉路由器接收哪些数据包、拒绝哪些数据包, 接收或者拒绝根据一定的规则进行, 如源地址、目标地址、端口号等。ACL 使得用户能够管理数据流, 检测特定的数据包。

路由器将根据 ACL 中指定的条件, 对经过路由器端口的数据包进行检查。ACL 可以基于所有的 Routed Protocols (被路由协议, 如 IP、IPX 等) 对经过路由器的数据包进行过滤。ACL 在路由器的端口过滤数据流, 决定是否转发或者阻止数据包。ACL 应该根据路由器的端口所允许的每个协议来制定, 如果需要控制流经某个端口的所有数据流, 就需要为该端口允许的每一个协议分别创建 ACL。例如, 如果端口被配置为允许 IP、AppleTalk 和 IPX 协议的数据流, 那么就需要创建至少 3 个 ACL, 本文中仅讨论 IP 的访问控制列表。针对 IP 协议, 在路由器的每一个端口, 可以创建两个 ACL: 一个用于过滤进入 (inbound) 端口的数据流, 另一个用于过滤流出 (outbound) 端口的数据流。

ACL 作用:

- 限制网络流量, 提高网络性能。

- 提供数据流控制。

- 为网络访问提供基本的安全层。

配置标准 ACL 需要两步, 一是创建访问控制列表, 二是将列表绑定到特定端口。

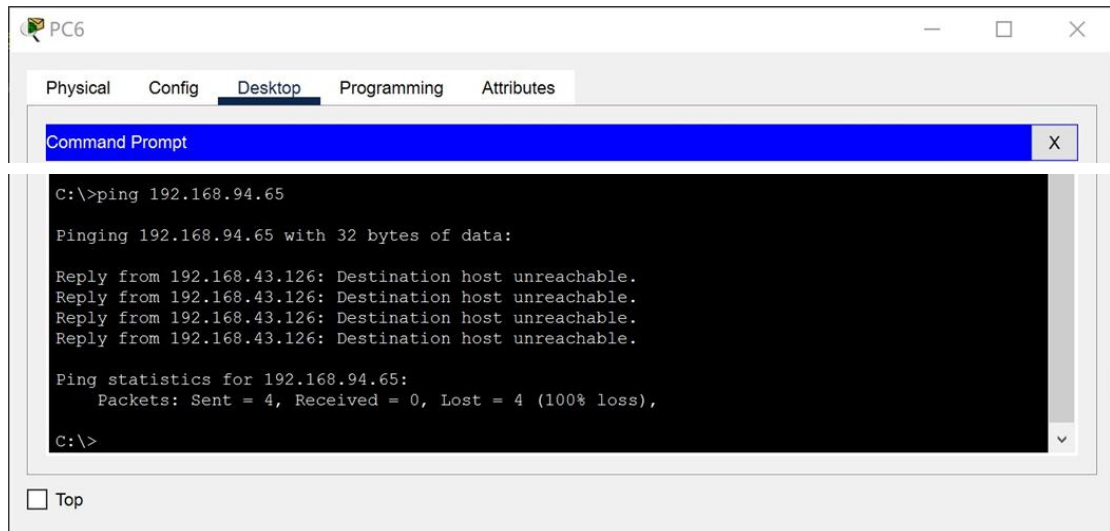
ACL 配置如下:

```
Router(config)#access-list 113 deny icmp 192.168.43.96 0.0.0.31 192.168.94.0 0.0.0.255
Router(config)#access-list 113 deny icmp 192.168.43.96 0.0.0.31 192.168.43.0 0.0.0.31
Router(config)#access-list 113 deny icmp 192.168.43.96 0.0.0.31 192.168.43.32 0.0.0.31
Router(config)#access-list 113 deny icmp 192.168.43.96 0.0.0.31 192.168.43.64 0.0.0.31
Router(config)#access-list 113 deny icmp 192.168.43.96 0.0.0.31 192.168.43.128 0.0.0.128
Router(config)#access-list 113 permit icmp 192.168.43.96 0.0.0.31 192.168.43.0 0.0.0.31 echo-reply
Router(config)#access-list 113 permit icmp 192.168.43.96 0.0.0.31 192.168.43.32 0.0.0.31 echo-reply
Router(config)#access-list 113 permit icmp 192.168.43.96 0.0.0.31 192.168.43.64 0.0.0.31 echo-reply
Router(config)#access-list 113 permit icmp 192.168.43.96 0.0.0.31 192.168.43.128 0.0.0.31 echo-reply
Router(config)#int fa0/1.43
Router(config-subif)#ip access-group 113 in
```

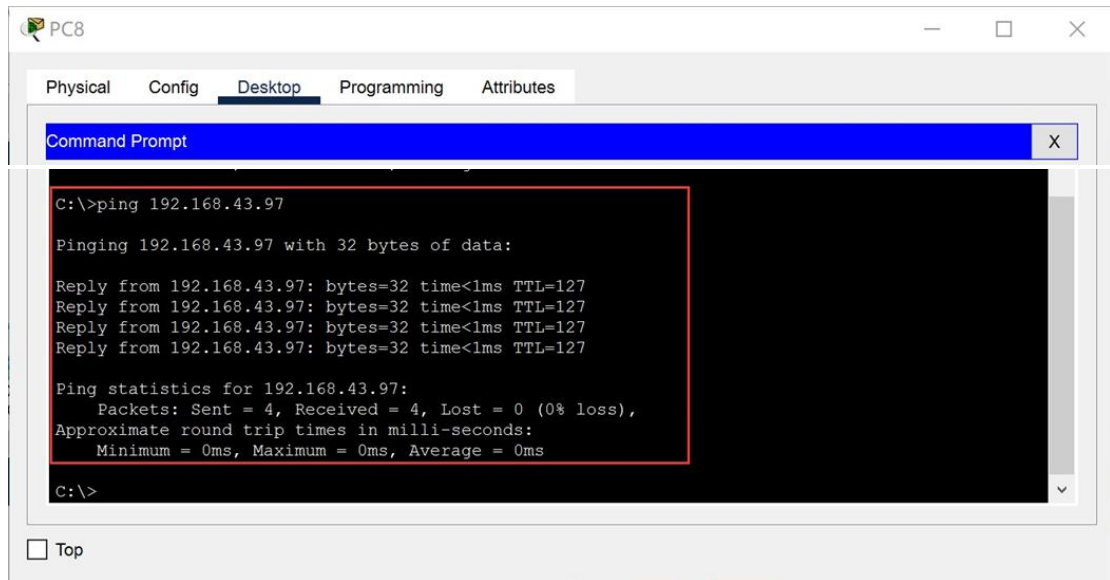
至此, 访问控制的配置已经完成, 我们可以在 PC6 上向其他局域网内的主机用 ping 命令测试连通性:

实验结果:

```
Router>en
Router#show ip access-lists
Extended IP access list 113
 10 deny icmp 192.168.43.96 0.0.0.31 192.168.94.0 0.0.0.255 (28 match(es))
 20 deny icmp 192.168.43.96 0.0.0.31 192.168.43.0 0.0.0.31 (4 match(es))
 30 deny icmp 192.168.43.96 0.0.0.31 192.168.43.32 0.0.0.31 (16 match(es))
 40 deny icmp 192.168.43.96 0.0.0.31 192.168.43.64 0.0.0.31 (12 match(es))
 50 deny icmp 192.168.43.96 0.0.0.31 192.168.43.0 0.0.0.128
 60 permit icmp 192.168.43.96 0.0.0.31 192.168.43.0 0.0.0.31 echo-reply
 70 permit icmp 192.168.43.96 0.0.0.31 192.168.43.32 0.0.0.31 echo-reply
 80 permit icmp 192.168.43.96 0.0.0.31 192.168.43.64 0.0.0.31 echo-reply
 90 permit icmp 192.168.43.96 0.0.0.31 192.168.43.128 0.0.0.31 echo-reply
100 permit ip any any (5 match(es))
```

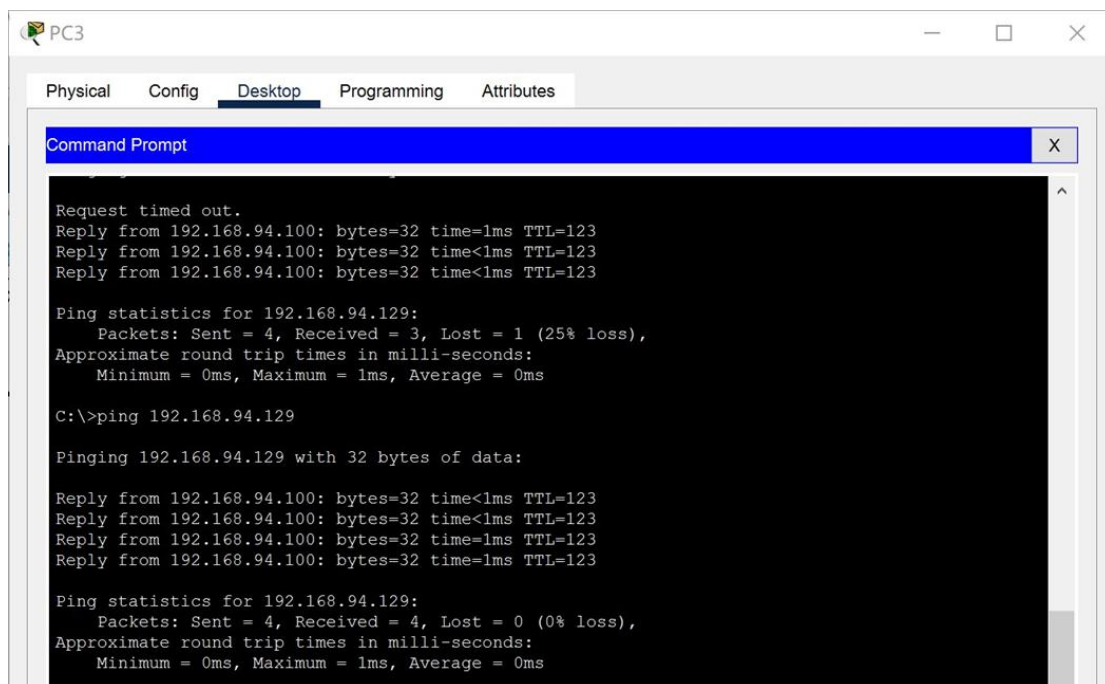
而外网的其他主机能够 ping 通主机 PC6，下图表示 PC8 可以 ping 通 PC6:



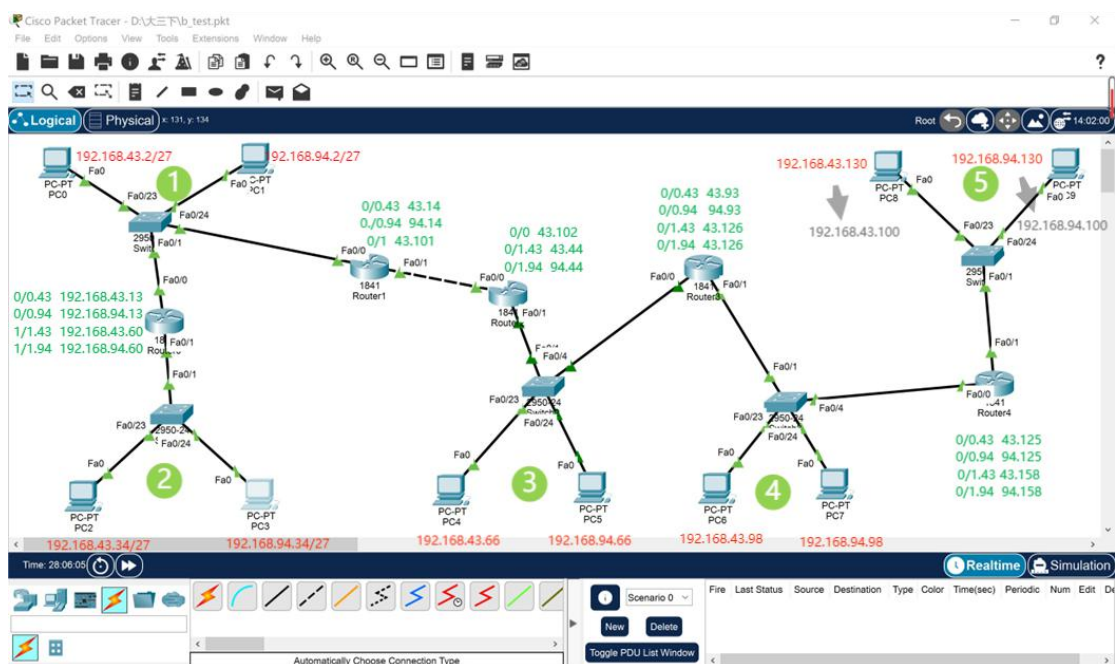
8. 任意两个节点之间能在规则下互相访问

在我们的规则中，除了主机 PC6 无法 ping 通任何其他主机外，其余的任何主机都是可以互相访问的。在这里的连通性因为主机很多，这里只展示出 PC3 和 PC9 之间的连通性：

实验结果：



9. 实验结果



五、实验总结

做实验前，对实验原理和软件使用都缺乏足够的认识，以前做过对路由器和交换机不同 Vlan 下的配置，难度较小，此次实验涉及对 GRE、ACL、DHCP 等协议的配置和实现，难度较大。起初是尝试性的做，一边搜资料学习，一边实践配置，多次反复测试。

实验期间遇到各种问题，由于一开始对此次 B 测的要求说明缺乏重视及理解不透彻，导致对网段、VLAN 的分配比较混乱，且由于对所涉及到的协议和技术的不熟悉、理解不清晰，导致配置时缺乏整体规划，测试时屡次出错，无法达到预期效果，之后明确了实验要求，并通过查找资料学习和理解，对相关协议和技术有了较深刻的认识，对整体实验有了明确规划，认真设计各部分 IP 地址和网关及网段，使各器件间连接清晰合理。通过网络学习了 Packet Tracer 软件的使用，学习了各指令及其作用，然后按照实验要求，按部就班对实验进行了重新配置，对实验中用到的协议、技术以及 Packet Tracer 软件的使用都有了深刻的理解和掌握。

实验中也学到了 IP 地址同掩码之间的配合，来进行不同网段的划分，二者间良好的划分和配合是此次实验得以顺利进行乃至成功的关键所在。在理解了不同网段配置之后，对实验的整体结构有了清晰的认识，像路由器的路由表等等。后面的各种配置要求也进行的比较顺利。最后，按照要求完成了实验的所有配置，得到了正确的实验结果，收获了新的知识和技能。