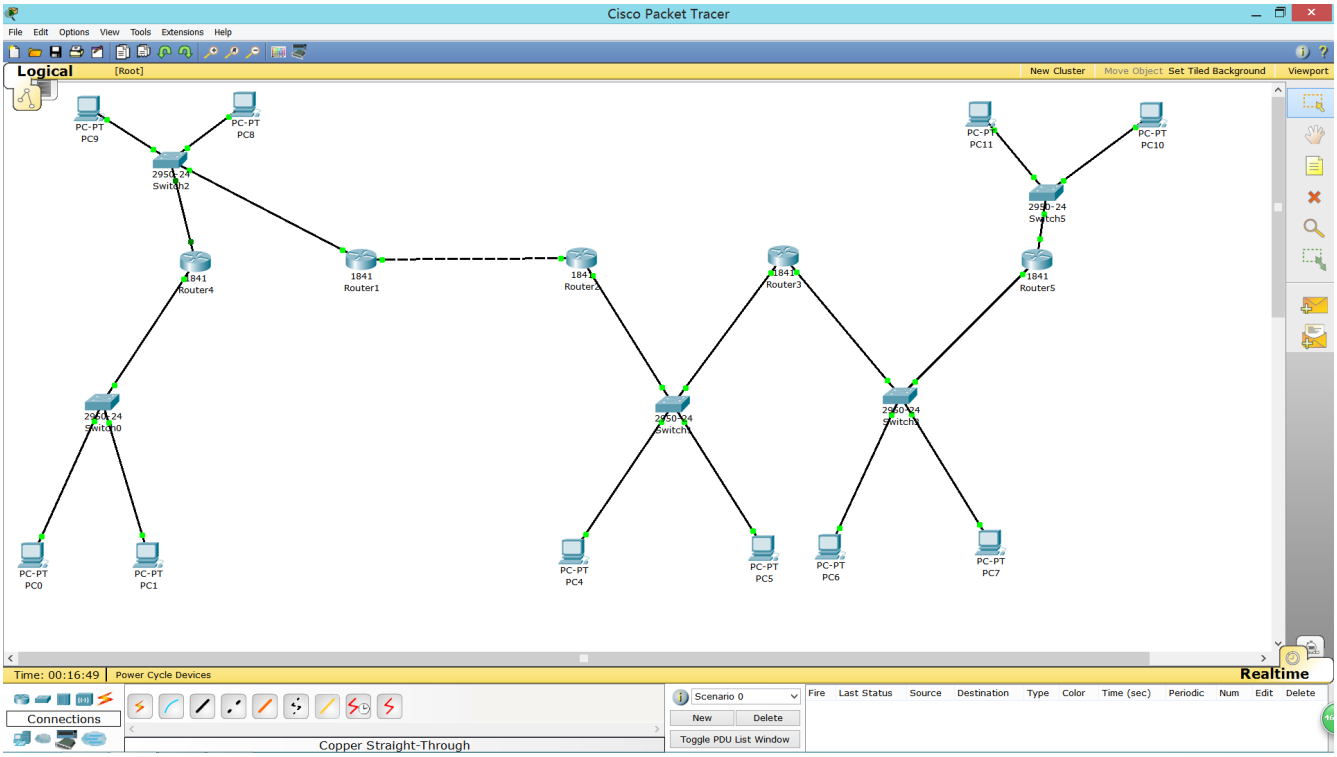
一，实验背景

- 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轮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,CRE 是 VPN（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（CRE 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。CRE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（英语：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的详细处理过程,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，实验要求

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，实验具体实现

1，划分Vlan和配置路由

- **以图中交换机0和交换机2之间的互连为例：交换机0的F0/1口与路由器4连接，F0/23口和F0/24口分别与两台PC相连。交换机2的F0/1口语路由器4连接，F0/23口和F0/24口分别与另外两台PC相连。**
- **对交换机0配置如下：**

（1）启用ip routing

（2）将F0/23口划入vlan2，F0/24口划入vlan3

命令：switch access vlan1

```
Switch(config)#vlan 2
Switch(config-vlan)#exit
Switch(config)#vlan 3
Switch(config-vlan)#exit

Switch(config)#interface fa 0/23
Switch(config-if)#switch access vlan 2
Switch(config-if)#exit

Switch(config)#interface fa 0/24
Switch(config-if)#switch access vlan 3
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 2,3    # 配置turnk口允许vlan2 和vlan3通过，除了vlan2和vlan3，其他所有的vlan默认拒绝所有

Switch#show vlan
```

- **对交换机2配置同理：**

```
Switch(config)#vlan 2
Switch(config-vlan)#exit
Switch(config)#vlan 3
Switch(config-vlan)#exit

Switch(config)#interface fa 0/23
Switch(config-if)#switch access vlan 2
Switch(config-if)#exit

Switch(config)#interface fa 0/24
Switch(config-if)#switch access vlan 3
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 2,3    # 配置turnk口允许vlan2 和vlan3通过，除了vlan2和vlan3，其他所有的vlan默认拒绝所有

Switch#show vlan
```

- **对路由器4配置如下：R1的接口F0/0上创建两个子接口，分别是F0/0.1对应的vlan2、F0/0.2对应的vlan 3，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**

```
Switch>     # 进入到用户模式
Switch>enable    # 进入到特权模式，该模式下的权限只允许查看命令
Swtich#config terminal    # 进入到全局模式，
# 该模式下的权限可以配置任何命令，如果想要执行查询命令，只需要在命令行首部加上 do命令，后面接需要查询的命令
Switch(config)#hostname R4    # 更改路由器的为R1
R4(config)#interface FastEthernet 0/0    # 进入到接口0/0
R4(config-if)#no shutdown    # 启动接口，路由器的接口状态默认是shutdown
R4(config-if)#exit    # 退出该接口
R4(config)#interface fastEthernet0/0.1
# 进入到F0/0.1接口
R4(config-subif)#encapsulation dot1Q 2
# 将vlan2封装在F0/0.1接口
R4(config-subif)#ip address 192.168.2.1 255.255.255.0
# 配置接口的ip地址，该ip地址作为vlan 2内的电脑的网关
R4(config-subif)#exit
# 退出F0/0.1接口
R4(config)#interface fastEthernet 0/0.2
# 进入到F0/0.1接口
R4(config-subif)#encapsulation dot1Q 3
# 将vlan3封装在F0/0.2接口
R4(config-subif)#ip address 192.168.3.1 255.255.255.0
# 配置ip地址，该ip地址作为vlan3内的电脑的网关
```

- **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**

2，用DHCP分配IP地址

- **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) 设置已分配地址的过期时间为3天。（此软件不支持此操作）

(7) 退出net172地址池的设置状态。

```
int f0/0
ip addr 192.168.1.1 255.255.255.0

int f0/1
ip addr 192.168.2.1 255.255.255.0

ip dhcp pool gpltoken1
network 192.168.1.0 255.255.255.0
```

```
default-router 192.168.1.1
dns-server 192.168.1.100
exit

ip dhcp pool gpltoken2
network 192.168.2.0 255.255.255.0
default-router 192.168.2.1
dns-server 192.168.2.100
exit
```

3，用到NAT技术完成地址变换

- **NAT（Network Address Translation ）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。**
- **静态地址转换：**

常用命令及步骤：

设置静态IP地址转换，需完成下列步骤：

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。

- **R1配制如下：**

```
Router>en
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int f0/0 //进入连接内网的接口
Router(config-if)#ip addr 172.16.1.254 255.255.255.0
Router(config-if)#no shut
Router(config-if)#int f0/1
Router(config-if)#ip addr 200.10.10.13 255.255.255.252
Router(config-if)#no shut
Router(config-if)#exit
Router(config)#
Router(config)#ip route 0.0.0.0 0.0.0.0 200.10.10.14 //配置默认路由
Router(config)#
Router (config)#ip nat inside source static 172.16.1.1 191.1.1.33
Router (config)#ip nat inside source static 172.16.1.2 191.1.1.34
Router (config)#int f0/0
```

```
Router (config-if)#ip nat inside
Router (config-if)#int f0/1
Router (config-if)#ip nat outside
```

- **R2配制如下：**

```
Router>en
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int f0/0
Router(config-if)#ip addr 200.10.10.14 255.255.255.252
Router(config-if)#no shut
Router(config-if)#int f0/1
Router(config-if)#ip addr 211.82.14.254 255.255.255.0
Router(config-if)#no shut
Router(config-if)#exit
Router(config)#ip route 191.1.1.32 255.255.255.240 200.10.10.13
```

4，用单臂路由实现互联

- **单臂路由：是为实现VLAN间通信的三层网络设备路由器，它只需要一个以太网，通过创建子接口可以承担所有VLAN的网关，而在不同的VLAN间转发数据。**
- **当交换机设置两个Vlan时，逻辑上已经成为两个网络，广播被隔离了。两个Vlan的网络要通信，必须通过路由器，如果接入路由器的一个物理端口，则必须有两个子接口分别与两个Vlan对应，同时还要求与路由器相连的交换机的端口fa 0/1要设置为trunk，因为这个接口要通过两个Vlan的数据包。检查设置情况，应该能够正确的看到Vlan和Trunk信息。计算机的网关分别指向路由器的子接口。配置子接口，开启路由器物理接口。默认封装dot1q协议。配置路由器子接口IP地址。**
- **交换机配制如下：**

```
Switch>en
Switch#conf t
Switch(config)#vlan 2
Switch(config-vlan)#exit
Switch(config)#fvlan 3
Switch(config-vlan)exit
Switch(config)interface fastEthernet 0/2  //进入交换机0模块第2端口
Switch(config-if)switchport access vlan 2  //加入vlan 2
Switch(config-if)exit
Switch(config)inter fa 0/3  //进入交换机0模块第3端口
Switch(config-if)switchport access vlan 3  //加入vlan 3
Switch(config-if)exit
Switch(config)inter fa 0/1    //进入交换机0模块第1端口
Switch(config-if)switchport mode trunk     //设置端口的工作模式为trunk
Switch(config-if)end
```

- **路由器配置如下：**

```
Router>en
Router#config t
Router(config)#inter fa 0/0  //进入路由器0模块第0端口
Router(config-if)#no shutdown  //开启该端口(此时路由连接交换机的端口 从block转到forward)
Router(config-if)#exit
Router(config)#interface fast 0/0.1  //进入路由器0模块第0端口第1子接口
Router(config-subif)#encapsulation dot1Q 2  //封装协议设置为dot1q 允许通过的vlan 为2
Router(config-subif)#ip address 192.168.45.66 255.255.255.0  //该子接口配置IP地址为
192.168.45.66
Router(config-subif)#exit
Router(config)#inter fa 0/0.2    //进入路由器0模块第0端口第2子接口
Router(config-subif)#encapsulation dot1q 3 //封装协议设置为dot1q 允许通过的vlan 为3
Router(config-subif)#ip address 192.168.45.65 255.255.255.0 //该子接口配置IP地址为
192.168.45.65
Router(config-subif)#end
Router#show ip route
```

5，要求用到GRE隧道技术。

- **GRE是一种最传统的隧道协议，其根本功能就是要实现隧道功能，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，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。

- **例如，当R2将数据包IP地址封装为192.168.1.4发往R4时，GRE操作过程如下：**

1．假设R1与R3的GRE虚拟直连链路（隧道）已经建立，隧道链路两端的地址分别为1.1.1.1和2.2.2.2，隧道两端的起源和终点分别为100.1.1.1和200.1.1.1。

2．R1收到目标IP为192.168.1.4的数据包后，将原始数据包当作数据包封装进GRE协议中，并且添加GRE包头，包头中源IP为隧道本端地址1.1.1.1，包头中目标IP为隧道对端地址1.1.1.2，从而完成GRE数据包的封装。

3．在封装了GRE隧道地址的数据包外面封装GRE隧道起源IP地址，该IP地址为公网地址，即源IP为100.1.1.1，目标IP为隧道终点200.1.1.1，最后将数据包发出去。

数据包被发到Internet之后，所有路由器只根据数据包最外面的公网IP进行转发，也就是只根据公网目标IP地址200.1.1.1来转发，直到数据包到达公网IP的真正目的地后，即到达R3（IP：200.1.1.1）之后，公网IP包头才会被剥开，当R3剥开数据包的公网IP包头后，发现GRE包头，发现目标IP为1.1.1.2，从而得知自己就是GRE隧道的终点，所以继续将GRE包头剥开，最后发现目标IP地址为192.168.1.4，然后将数据包发往192.168.1.4（路由器R4）。通过以上GRE过程，R2直接通过私有IP地址192.168.1.4，最终成功与R4通信。

- **隧道源端配置如下：**

```
PE1(config)#intface f0/0
PE1(config-if)#ip add 192.168.1.1 255.255.255.0
PE1(config)#intface f0/1
PE1(config-if)#ip add 12.1.1.1 255.255.255.0
PE1(config)#int tunnel 0
PE1(config-if)#tunnel source fa0/0
PE1(config-if)#tunnel destination 23.1.1.2
PE1(config-if)#ip add 10.0.0.1 255.255.255.0      用于tunnel隧道接口之间通信，必须在一个网段
PE1(config)#ip route 172.16.1.0 255.255.255.0 tunnel 0    告诉去往172.16.1.0/24的vpn数据包下一跳出接口走隧道
```

- **隧道目地端配置如下：**

```
PE2(config)#intface f0/1
PE2(config-if)#ip add 172.16.1.1 255.255.255.0
PE2(config)#int tunnel 0
PE2(config-if)#tunnel source fa0/0
PE2(config-if)#tunnel destination 12.1.1.1
PE2(config-if)#ip add 10.0.0.2 255.255.255.0
PE2(config)#ip route 192.168.1.0 255.255.255.0 tunnel 0
```

6，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:—个用于过滤进入（inbound)端口的数据流，另一个用于过滤流出（outboimd)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。

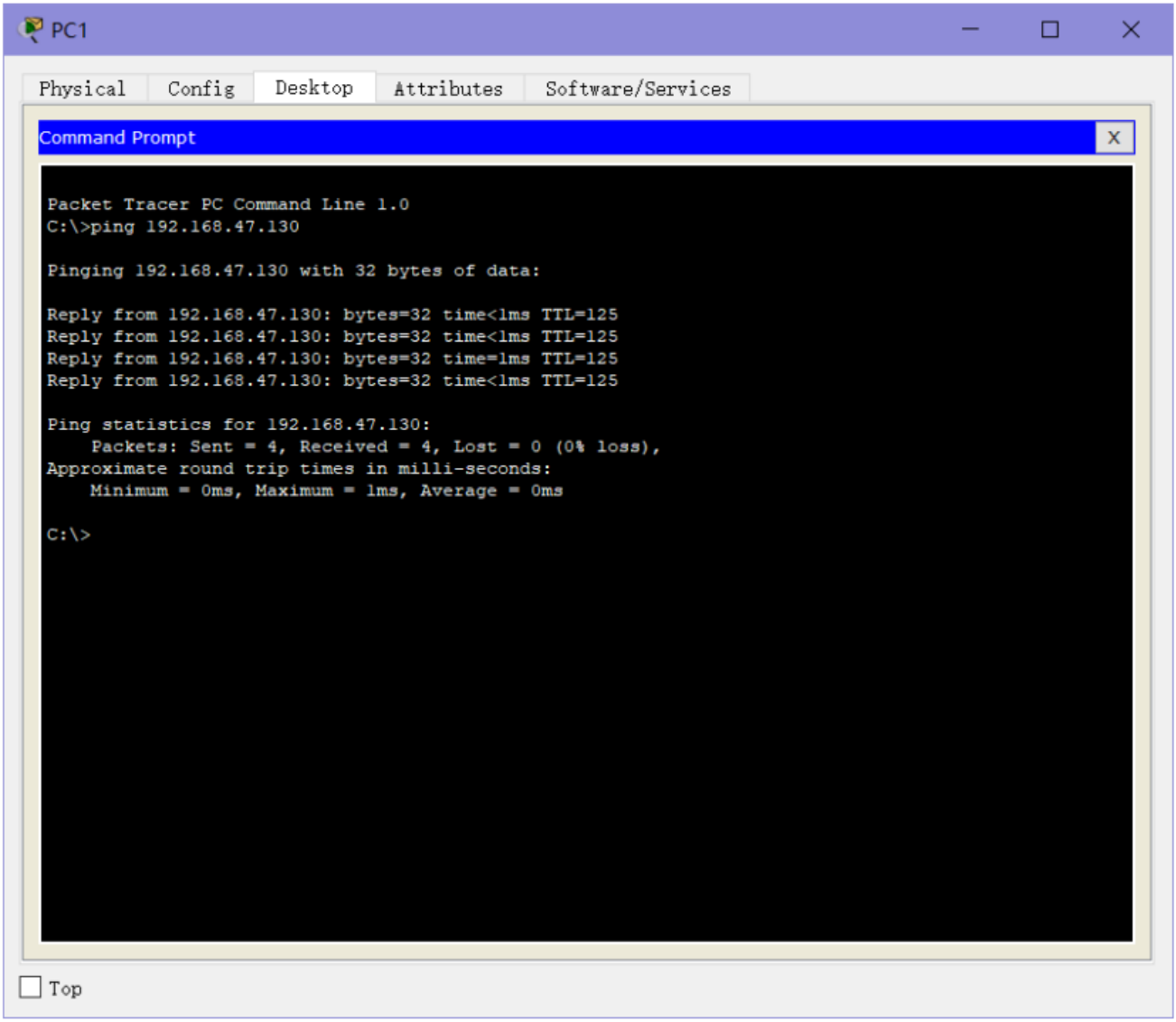
ACL配置如下：

```
Router>en
Router#conf t
Router(config)#access-list 101 deny icmp 192.168.45.34 0.0.0.0 192.168.47.0 0.0.0.255
Router(config)#access-list 101 deny icmp 192.168.45.50 0.0.0.0 192.168.47.0 0.0.0.255
Router(config)#access-list 101 permit ip any any
//启用ACL
Router(config)#int fa0/0.1
Router(config-if)#ip access-group 101 in
Router(config)#int fa0/0.2
Router(config-if)#ip access-group 101 in
```

四，实验结果展示

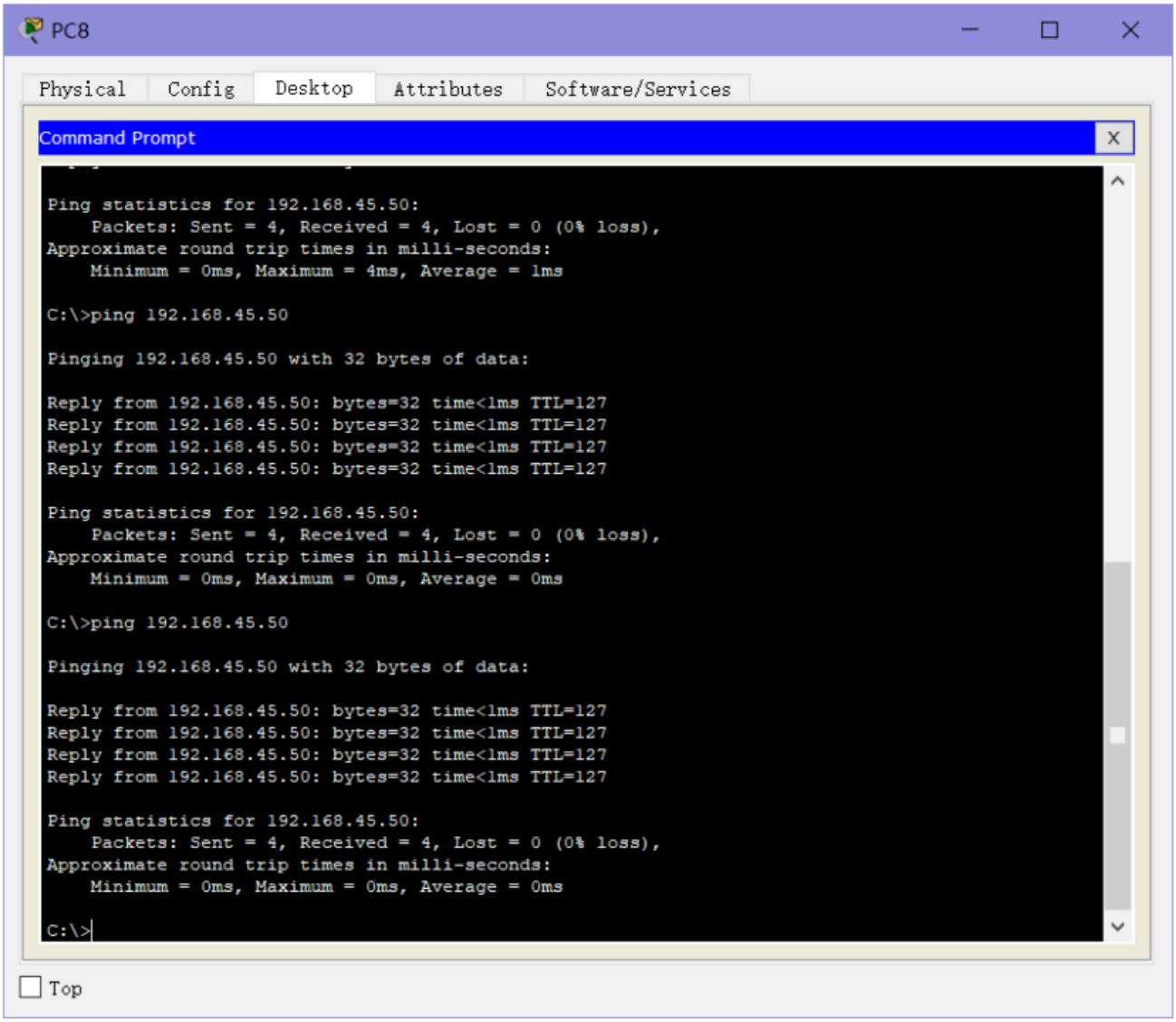
- **NAT互通：**

NAT：


PC1
Physical
Config
Desktop
Attributes
Software/Services
Command Prompt
Packet Tracer PC Command Line 1.0
C:\>ping 192.168.47.130
Pinging 192.168.47.130 with 32 bytes of data:
Reply from 192.168.47.130: bytes=32 time<1ms TTL=125
Reply from 192.168.47.130: bytes=32 time<1ms TTL=125
Reply from 192.168.47.130: bytes=32 time=1ms TTL=125
Reply from 192.168.47.130: bytes=32 time<1ms TTL=125
Ping statistics for 192.168.47.130:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
C:\>
Top

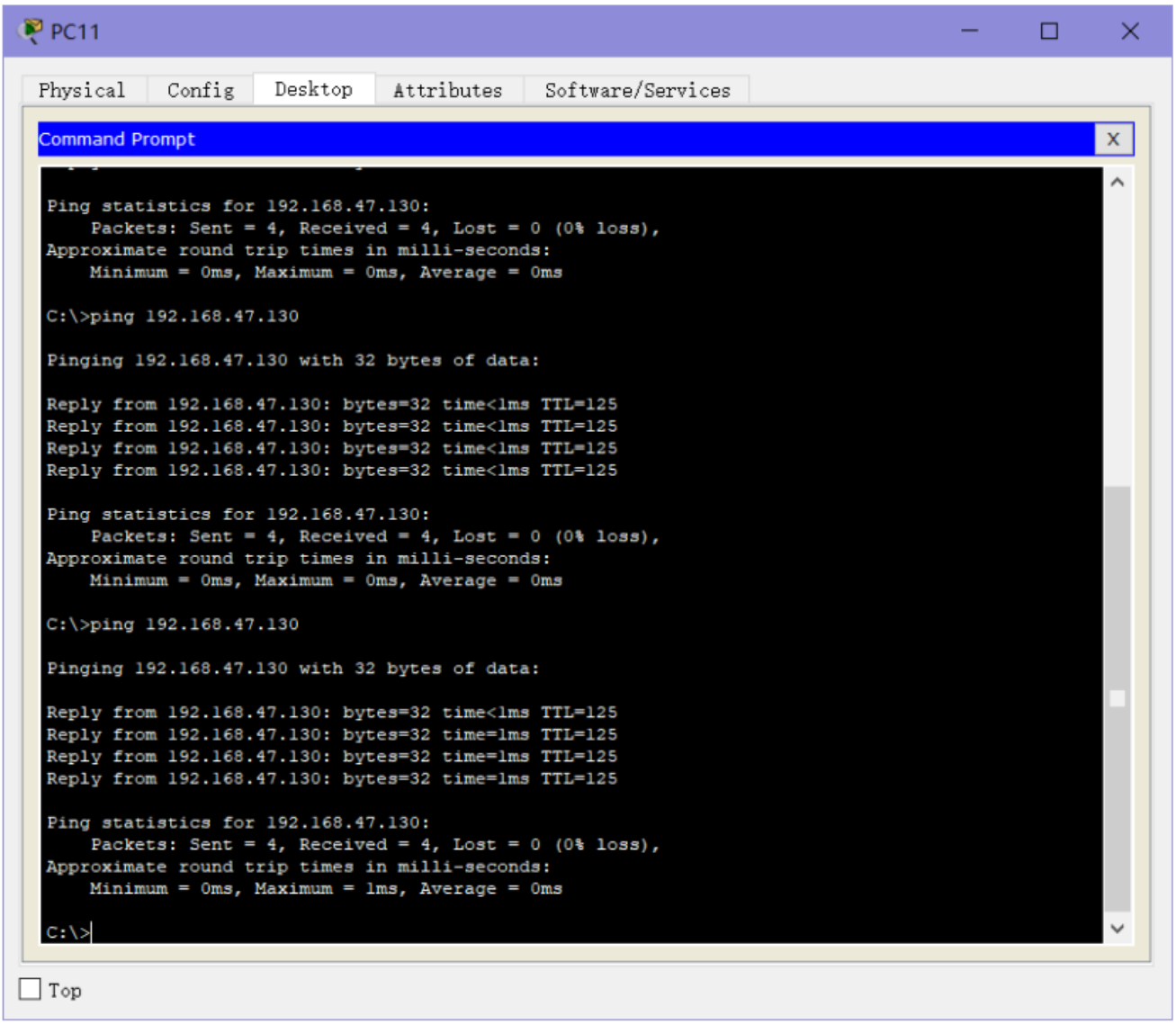
- ACL互通：

ACL：


PC8
Physical
Config
Desktop
Attributes
Software/Services
Command Prompt
Ping statistics for 192.168.45.50:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 4ms, Average = 1ms
C:\>ping 192.168.45.50
Pinging 192.168.45.50 with 32 bytes of data:
Reply from 192.168.45.50: bytes=32 time<1ms TTL=127
Reply from 192.168.45.50: bytes=32 time<1ms TTL=127
Reply from 192.168.45.50: bytes=32 time<1ms TTL=127
Reply from 192.168.45.50: bytes=32 time<1ms TTL=127
Ping statistics for 192.168.45.50:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>ping 192.168.45.50
Pinging 192.168.45.50 with 32 bytes of data:
Reply from 192.168.45.50: bytes=32 time<1ms TTL=127
Reply from 192.168.45.50: bytes=32 time<1ms TTL=127
Reply from 192.168.45.50: bytes=32 time<1ms TTL=127
Reply from 192.168.45.50: bytes=32 time<1ms TTL=127
Ping statistics for 192.168.45.50:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>
Top

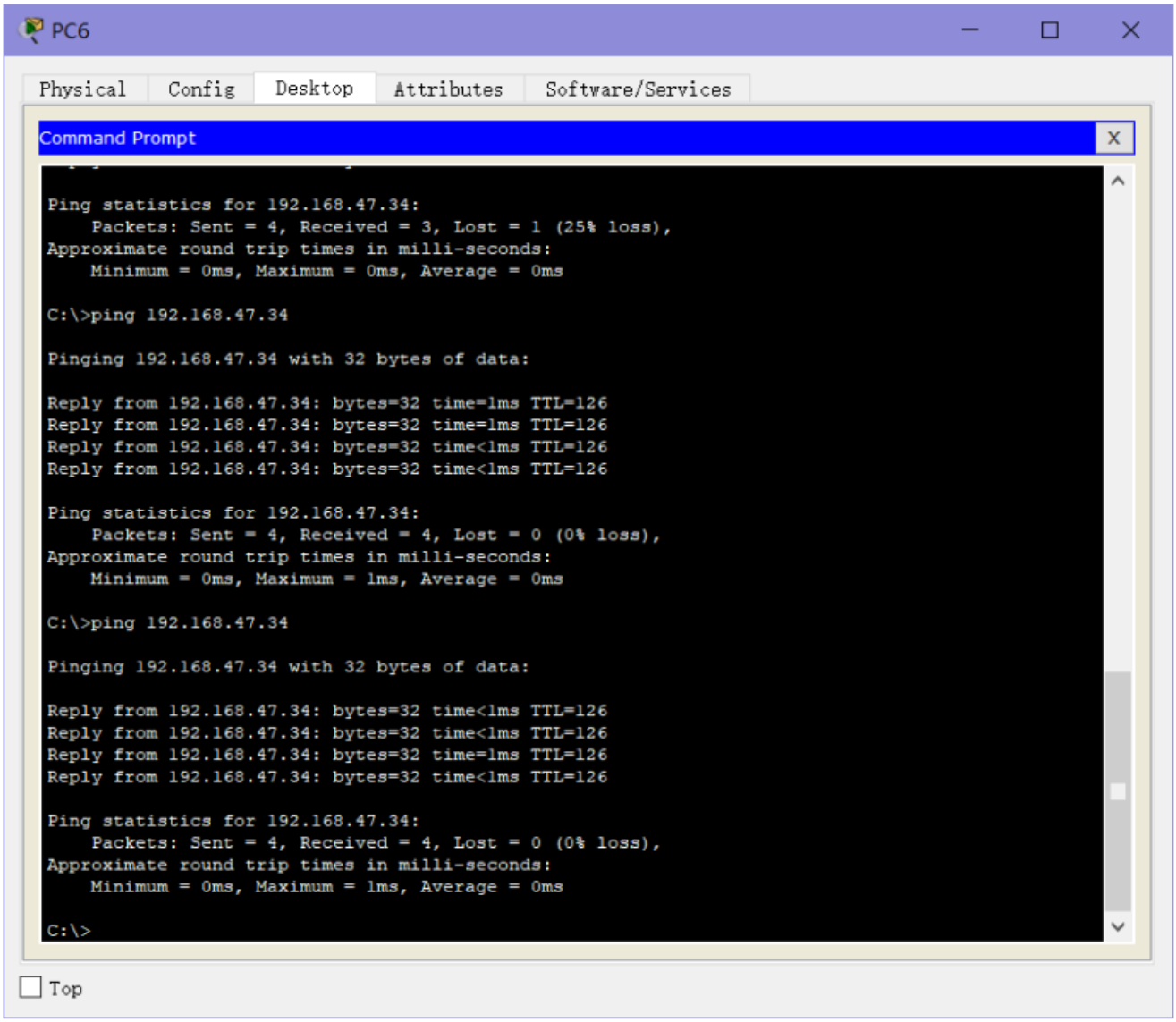
- DHCP互通：

DHCP：


PC11
Physical
Config
Desktop
Attributes
Software/Services
Command Prompt
Ping statistics for 192.168.47.130:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>ping 192.168.47.130
Pinging 192.168.47.130 with 32 bytes of data:
Reply from 192.168.47.130: bytes=32 time<1ms TTL=125
Reply from 192.168.47.130: bytes=32 time<1ms TTL=125
Reply from 192.168.47.130: bytes=32 time<1ms TTL=125
Reply from 192.168.47.130: bytes=32 time<1ms TTL=125
Ping statistics for 192.168.47.130:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>ping 192.168.47.130
Pinging 192.168.47.130 with 32 bytes of data:
Reply from 192.168.47.130: bytes=32 time<1ms TTL=125
Reply from 192.168.47.130: bytes=32 time=1ms TTL=125
Reply from 192.168.47.130: bytes=32 time=1ms TTL=125
Reply from 192.168.47.130: bytes=32 time=1ms TTL=125
Ping statistics for 192.168.47.130:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
C:\>
Top

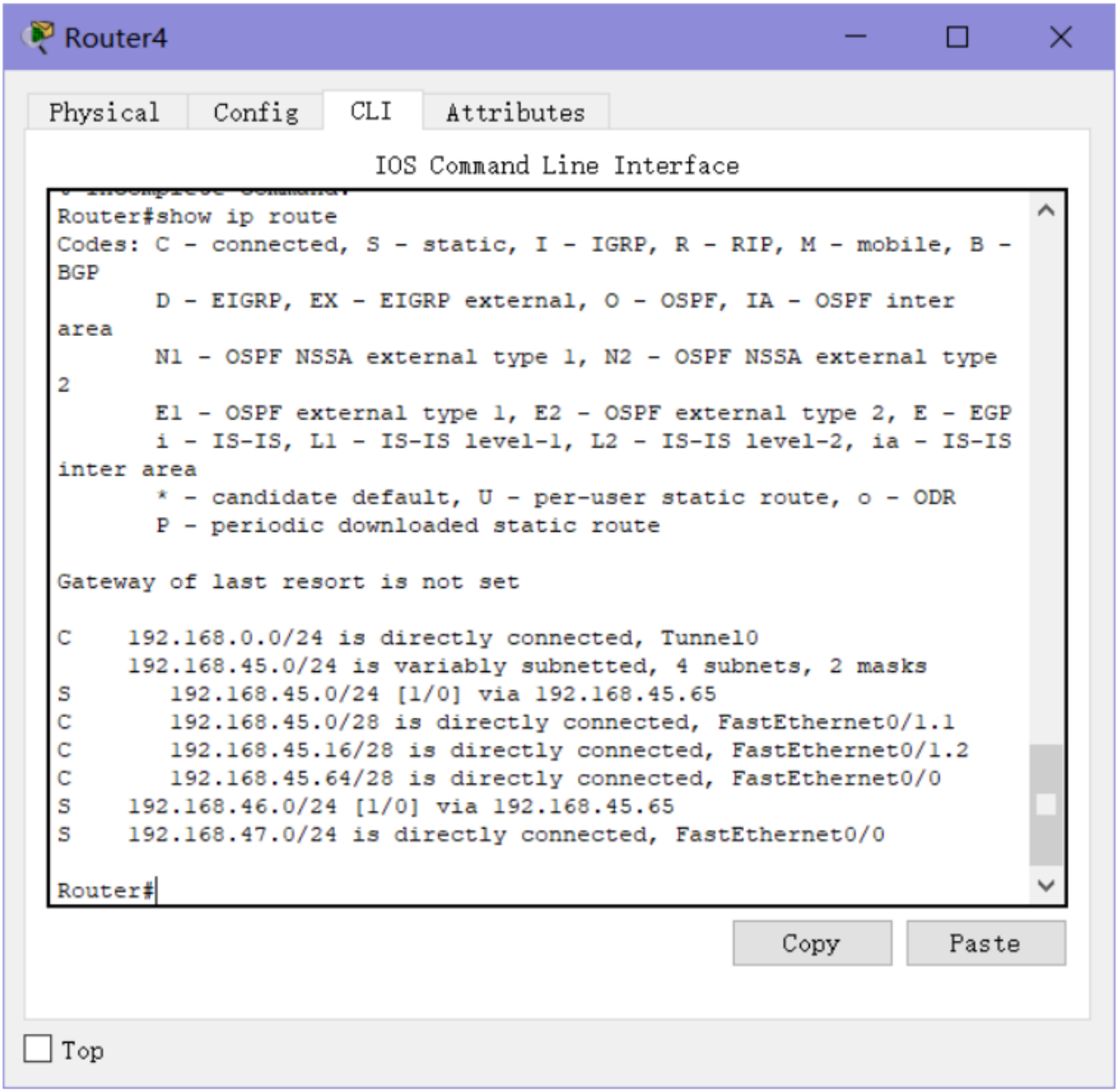
- **PC间互通：**

PC and PC：



- **GRE互通：**

GRE：



五，实验总结

做试验前，是有些懵的，因为之前只做过对路由器和交换机不同Vlan下的配置，对什么GRE、ACL、DHCP都没配置过，一开始只是尝试性的做，一边搜资料学习，一遍实践配置。

期间遇到很多问题，由于一开始没看到B测网站上的老师说明，自个儿按照说明上的要求配置，网段什么的都是乱分配的。后来看到要求之后，对实验进行了重新分配。

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（我觉得这是整个实验的关键部分）在理解了不同网段配置之后，好像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顺利的进行了下来。最后，完成了实验的所有配置。