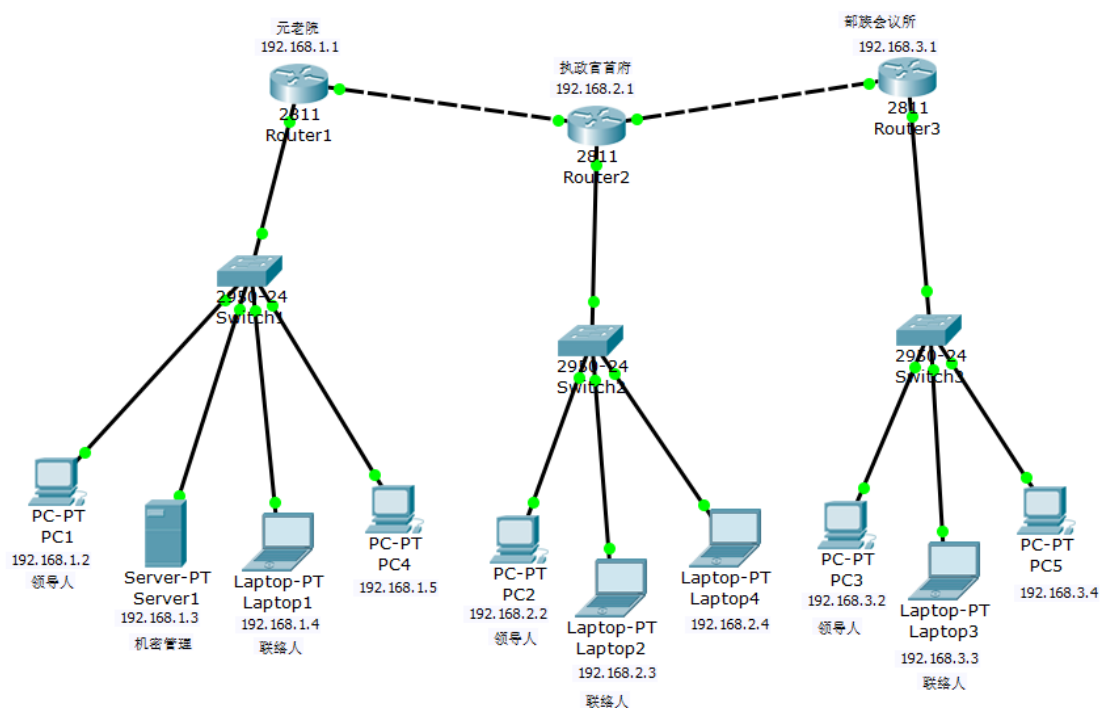


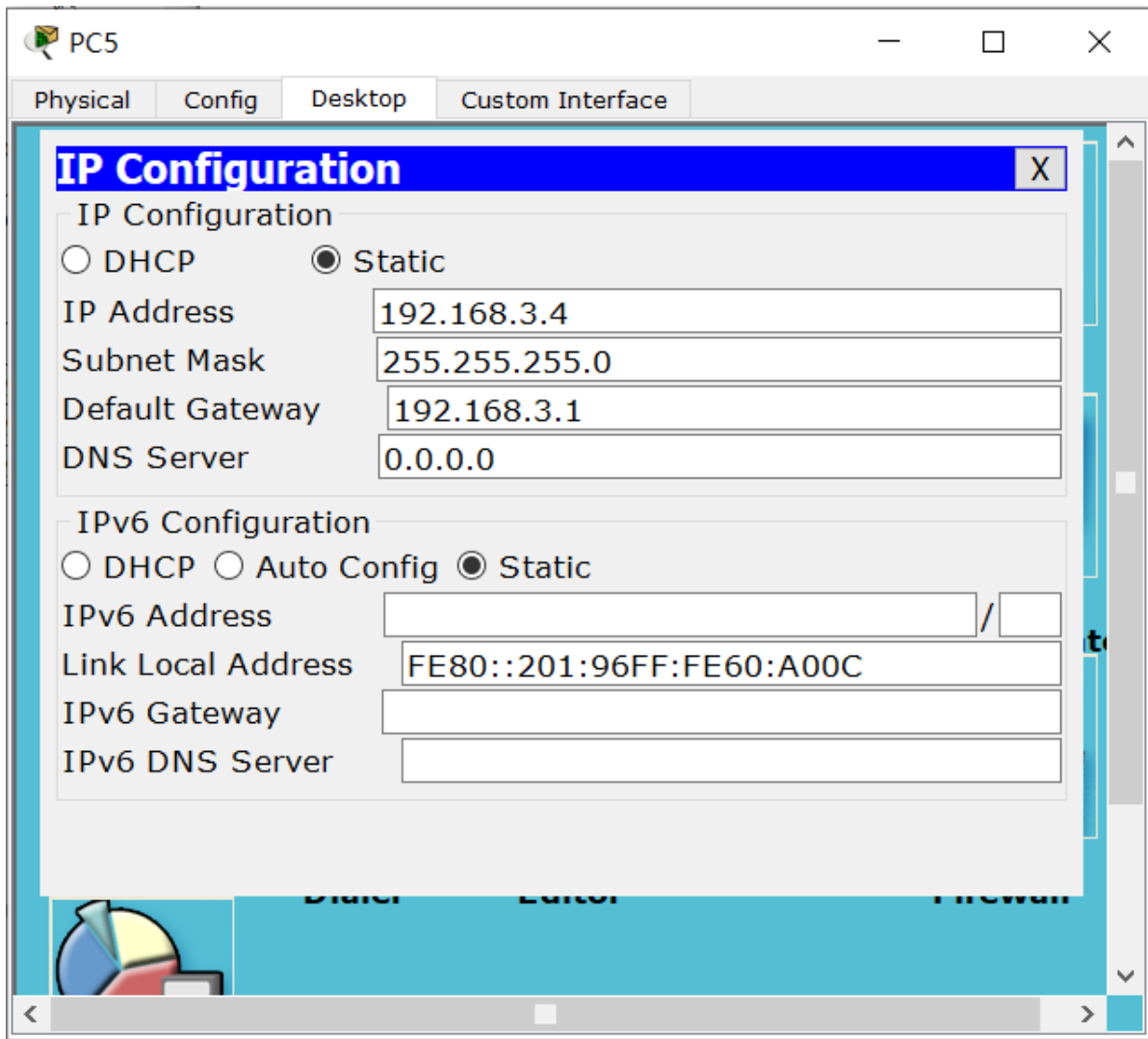
计算机网络安全技术：实验二

刘泓尊 2018011446 计84

初始的网络拓扑如下：



需要进行 IP 配置，增加了 PC4，Laptop4 和 PC5，以 PC5 为例进行 IP 配置。



任务六

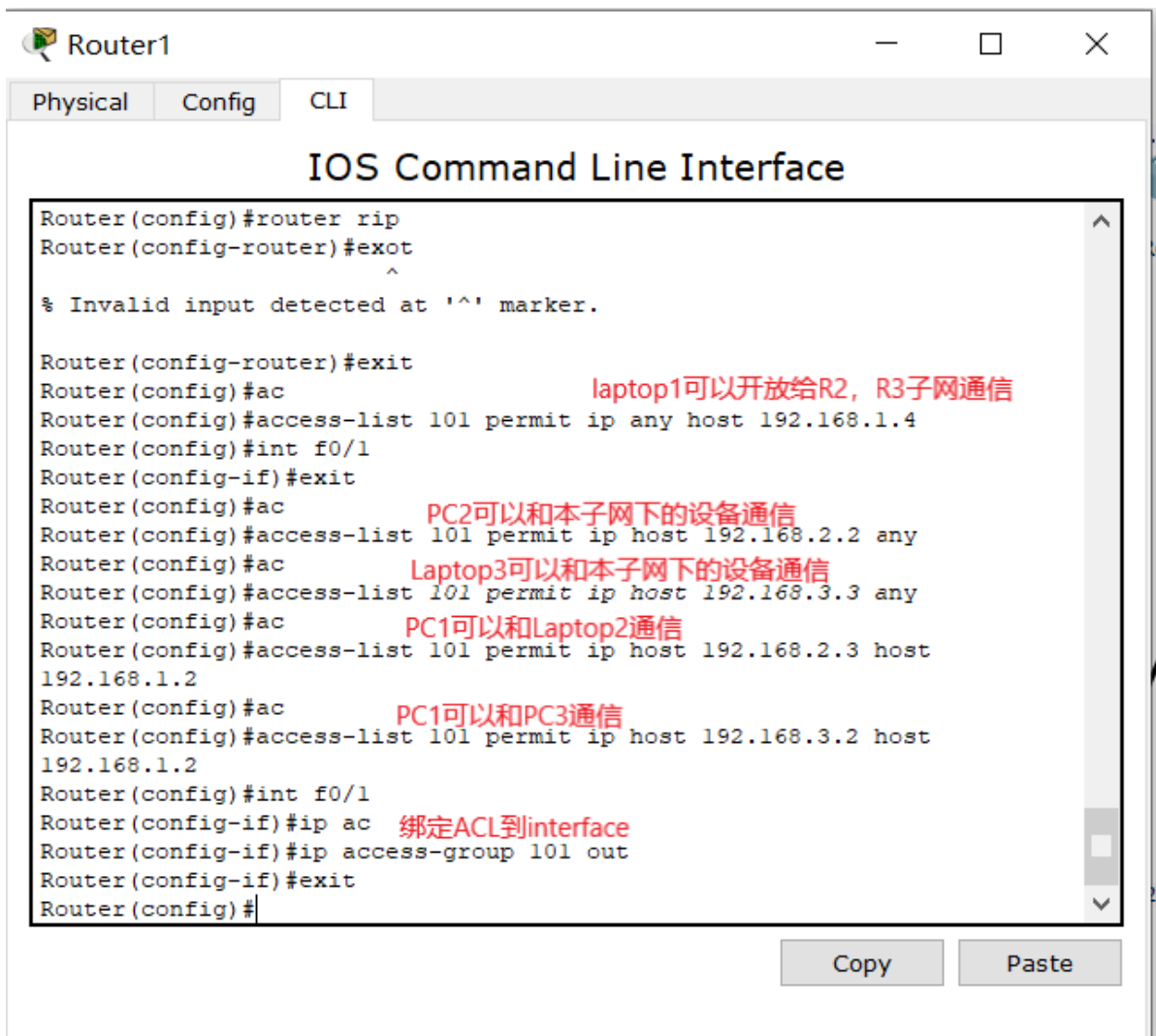
可以相互通信的机器总结如下：

0. 一般子网内计算机可以相互通信
1. Laptop1可以和 R2, R3子网下的所有计算机通信
2. PC2可以和R1, R3子网下的所有计算机通信
3. Laptop3可以和R1,R2子网下的所有计算机通信
4. PC1, Laptop2, PC3之间可以相互通信
5. 只有PC1可以和Server1通信。其他都不能和Server1通信

其中第0条配置完网络拓扑后即可实现。1-4条通过扩展ACL实现，第5条通过给server0配置防火墙实现。

ACL配置

配置Router1的ACL，主要步骤在图中已经标出。所有ACL都绑定到路由器对子网接口的out方向。



同理，配置 R2 和 R3 的 ACL，截图如下：

IOS Command Line Interface

```
Router#con
Router#conf ter
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ac
Router(config)#access-list 102 permit ip host any host 192.168.2.2
                                     ^
% Invalid input detected at '^' marker.
PC2可以和外网上的设备通信
Router(config)#access-list 102 permit ip any host 192.168.2.2
Router(config)#ac Laptop1可以访问本子网下的所有设备
Router(config)#access-list 102 permit ip host 192.168.1.4 any
Router(config)#ac
Router(config)#access-list 102 pe Laptop3可以和本子网下所有设备通信
Router(config)#access-list 102 permit ip host 192.168.3.3 any
Router(config)#ac
Router(config)#access-list 102 permit ip host 192.168.1.2 host
192.168.2.3
Laptop2可以和PC1通信
Router(config)#ac
Laptop2可以和PC3通信
Router(config)#access-list 102 permit ip host 192.168.3.2 host
192.168.2.3
Router(config)#int f1/0
Router(config-if)#ip ac 绑定ACL到interface的out端口
Router(config-if)#ip access-group 102 out
Router(config-if)#exit
Router(config)#
```

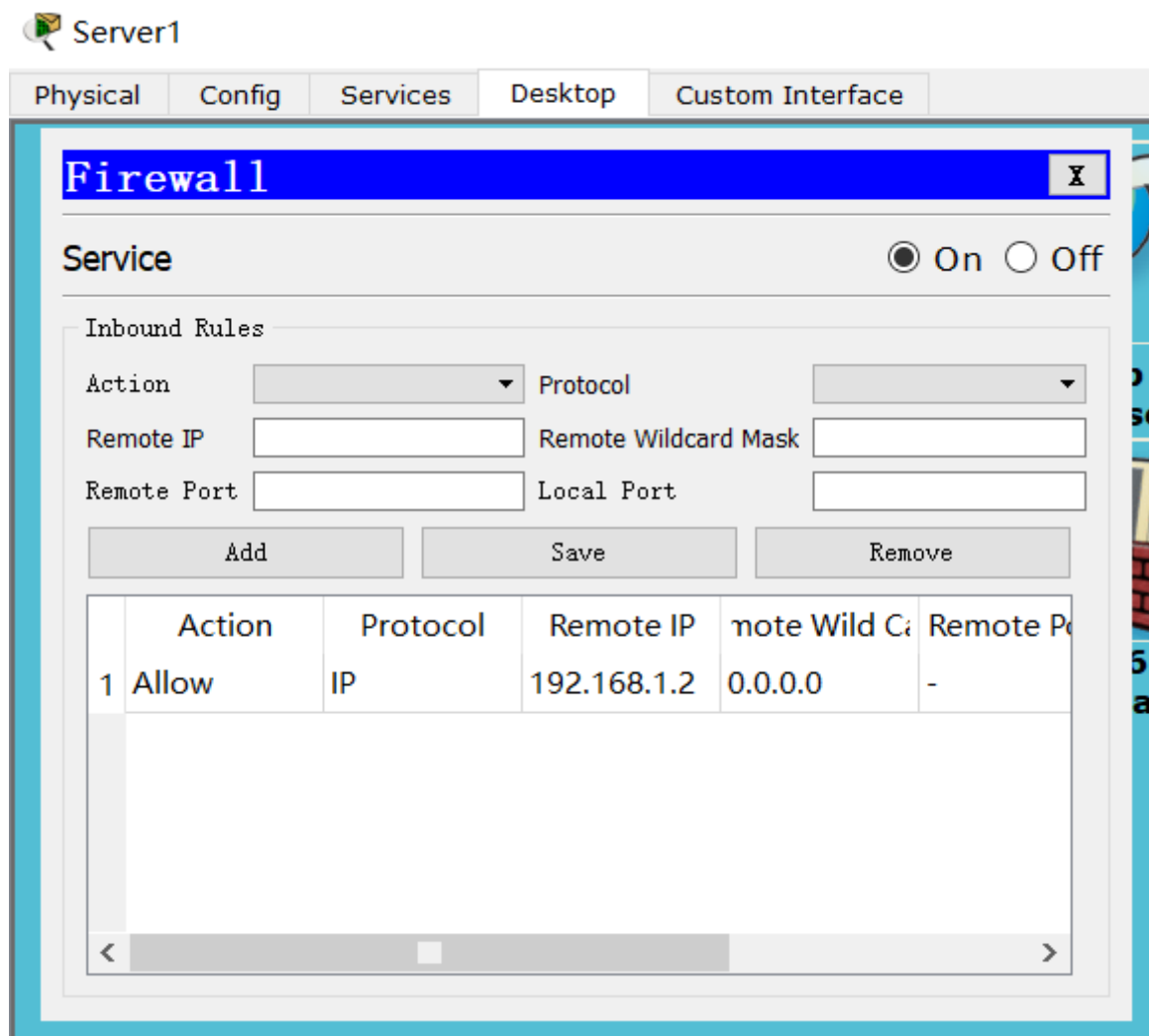
Copy

Paste

IOS Command Line Interface

```
Router>en
Router#conf ter
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ac
Router(config)#access-list 103 permit ip any host 192.168.3.3
Laptop3可以和外网上的所有设备通信
Router(config)#ac
Laptop1可以和本子网下所有设备通信
Router(config)#access-list 103 permit ip host 192.168.1.4 any
Router(config)#ac
PC2可以和本子网下所有设备通信
Router(config)#access-list 103 permit ip host 192.168.2.2 any
Router(config)#ac
PC3可以和PC1通信
Router(config)#access-list 103 permit ip host 192.168.1.2 host
192.168.3.2
Router(config)#ac
PC3可以和Laptop2通信
Router(config)#access-list 103 permit ip host 192.168.2.3 host
192.168.3.2
Router(config)#int f0/1
Router(config-if)#ip ac 绑定ACL到interface的out方向
Router(config-if)#ip access-group 103 out
Router(config-if)#exit
Router(config)#
```

之后给 server1 配置**防火墙**，只允许 PC1 访问。



下面进行 Ping 测试：

1.一般情况下子网内部可以相互通信：

PC1 ping Laptop1 成功

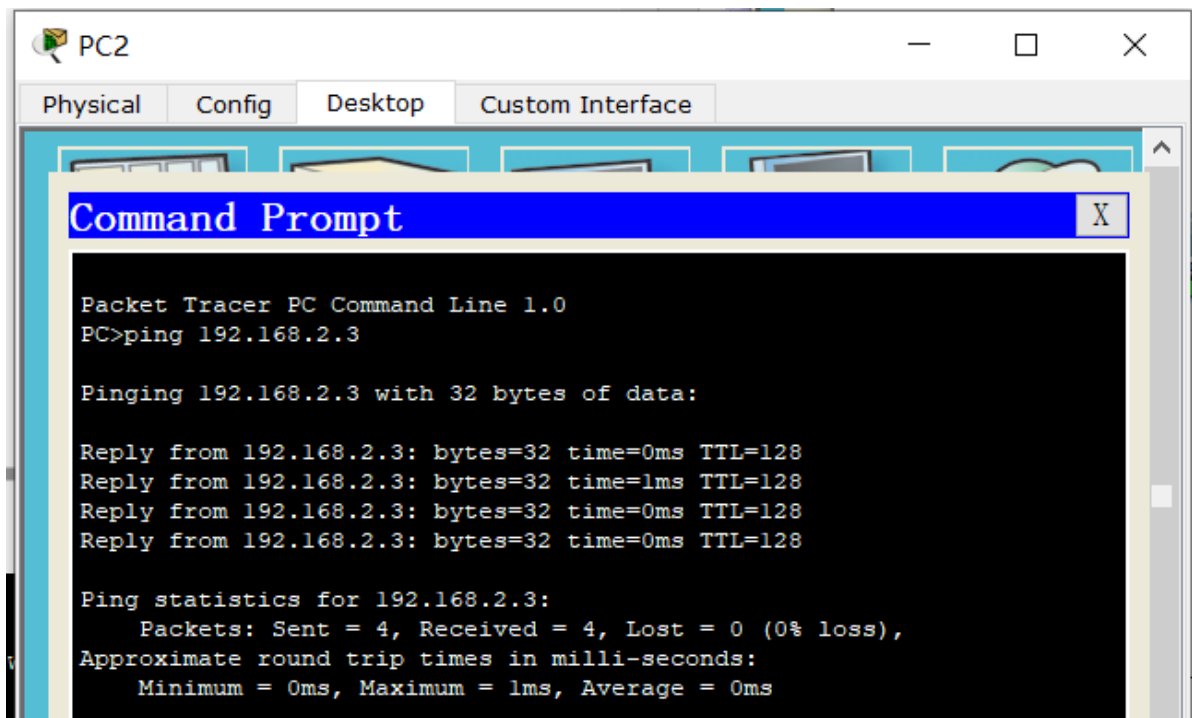
```
PC>ping 192.168.1.4

Pinging 192.168.1.4 with 32 bytes of data:

Reply from 192.168.1.4: bytes=32 time=1ms TTL=128
Reply from 192.168.1.4: bytes=32 time=0ms TTL=128
Reply from 192.168.1.4: bytes=32 time=0ms TTL=128
Reply from 192.168.1.4: bytes=32 time=0ms TTL=128

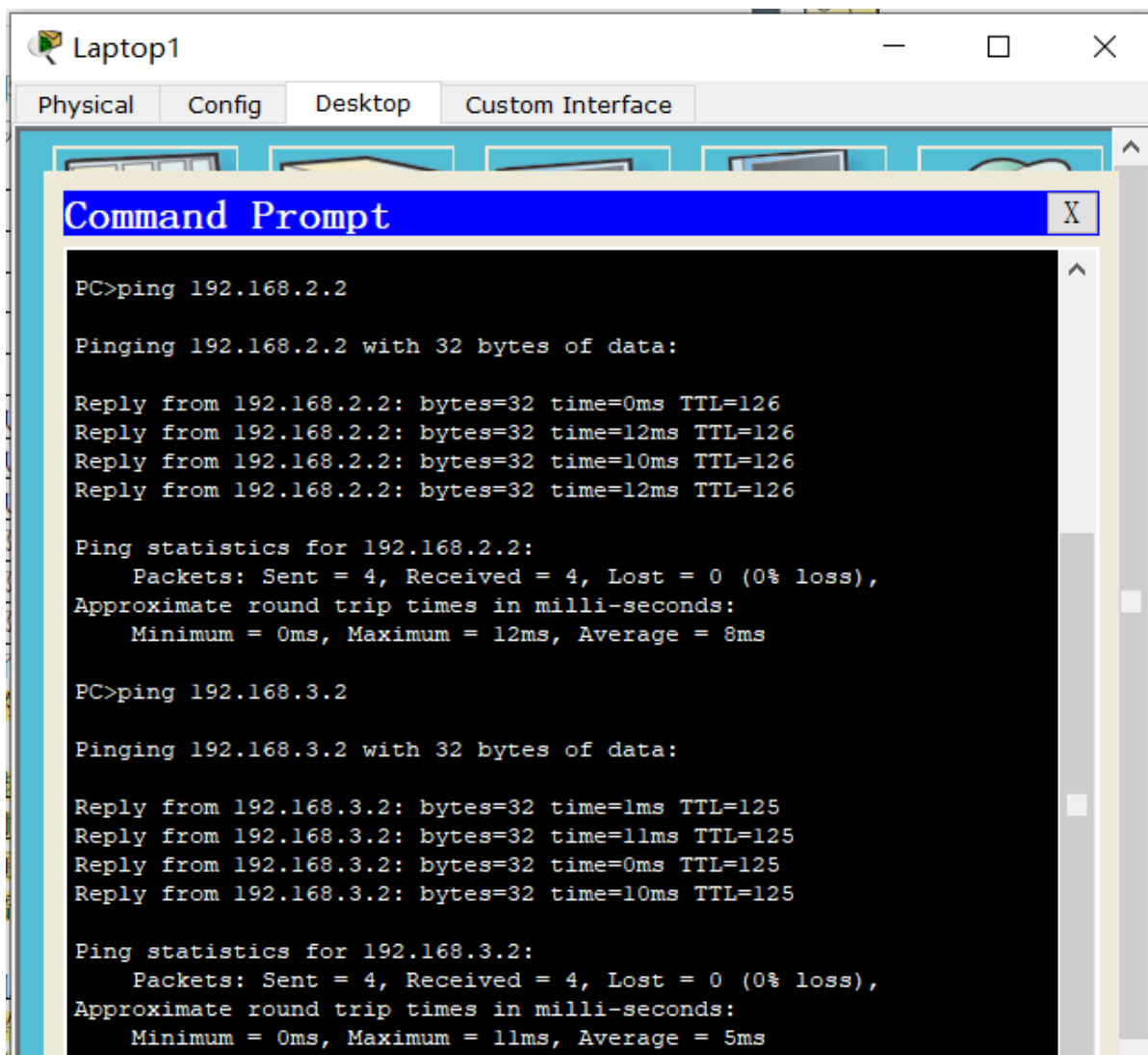
Ping statistics for 192.168.1.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

PC2 ping Laptop2 成功

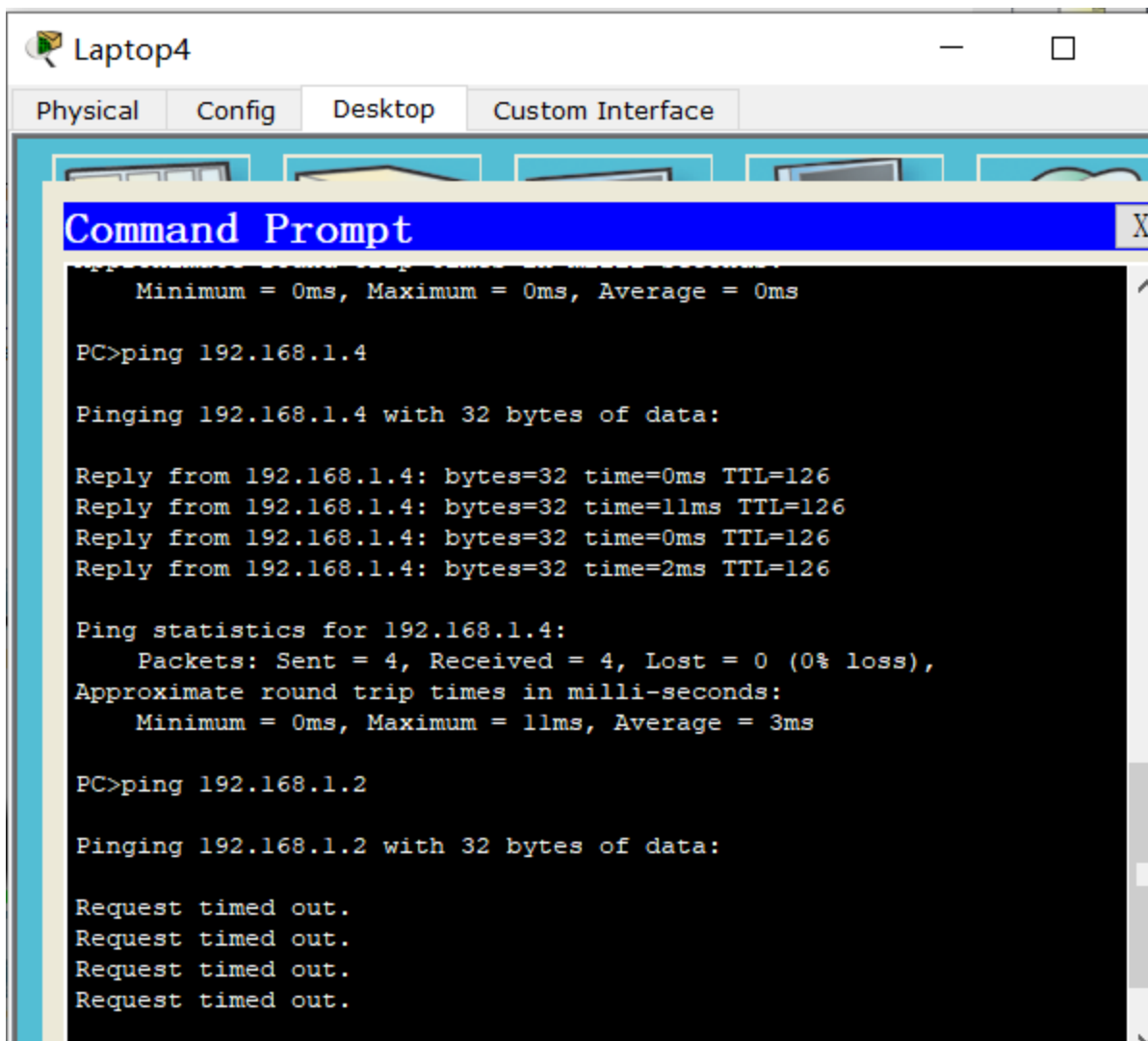


2.子网之间的用户通过联络人通信

Laptop1 ping PC2 成功, Laptop1 ping PC3 成功

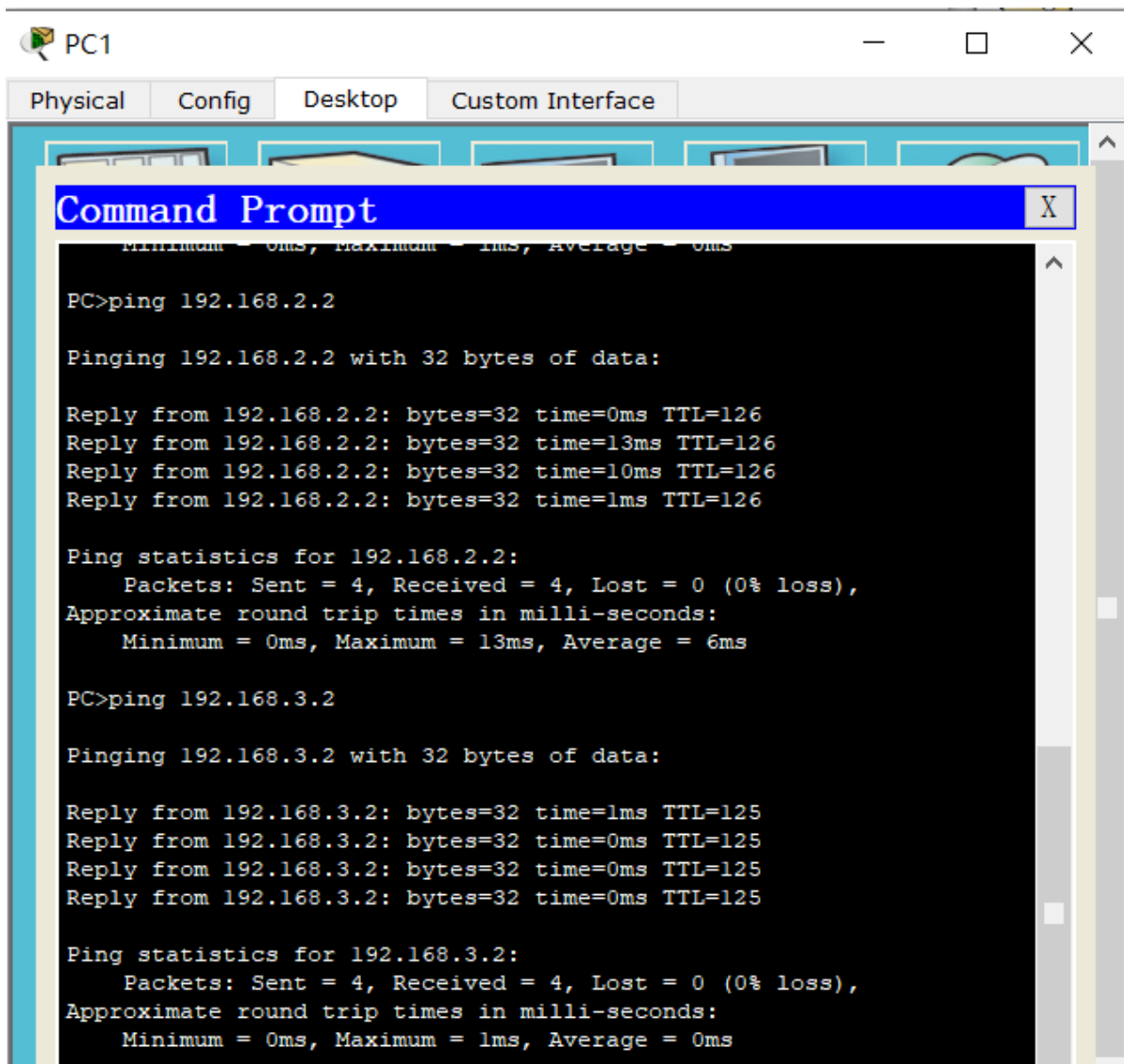


Laptop4 ping Laptop1 成功, Laptop4 ping PC1 失败

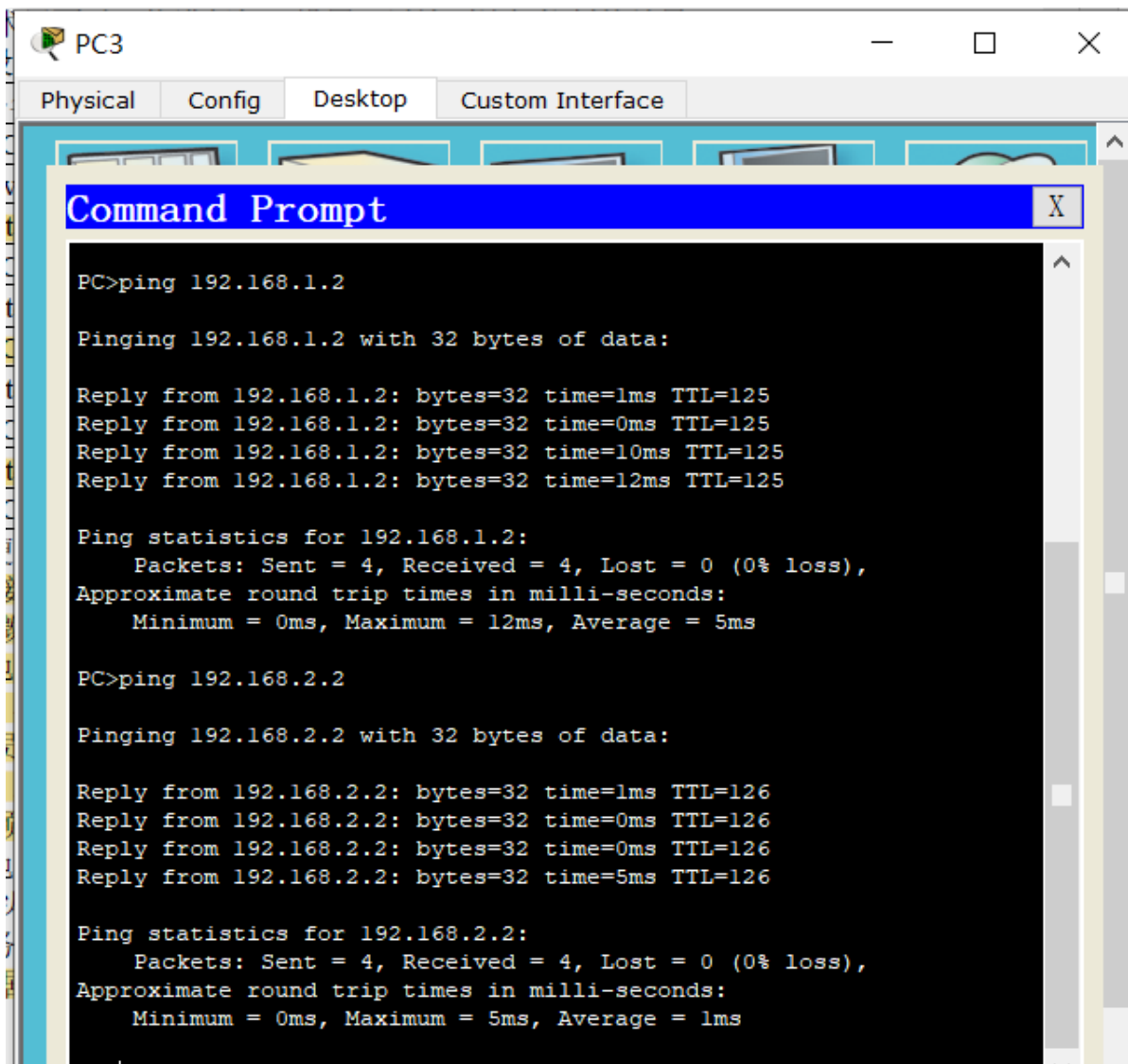


3.领导人之间可以相互通信

PC1 ping PC2 成功, PC1 ping PC3 成功

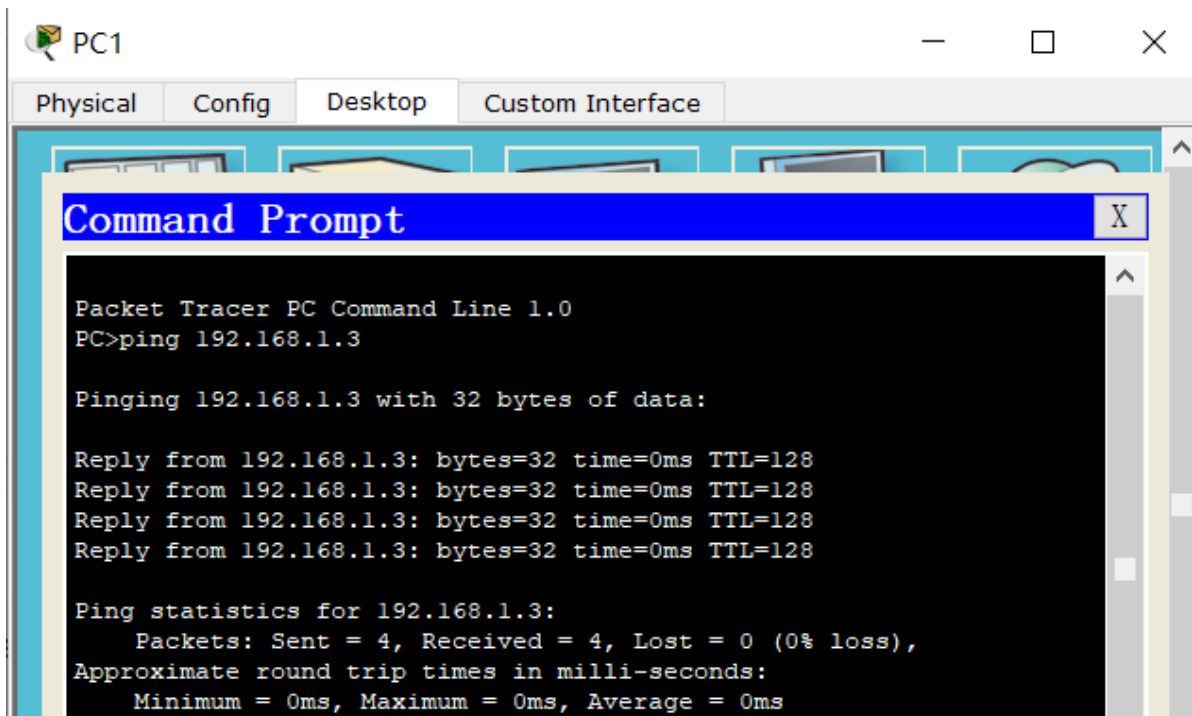


PC3 ping PC1成功, PC3 ping PC2成功

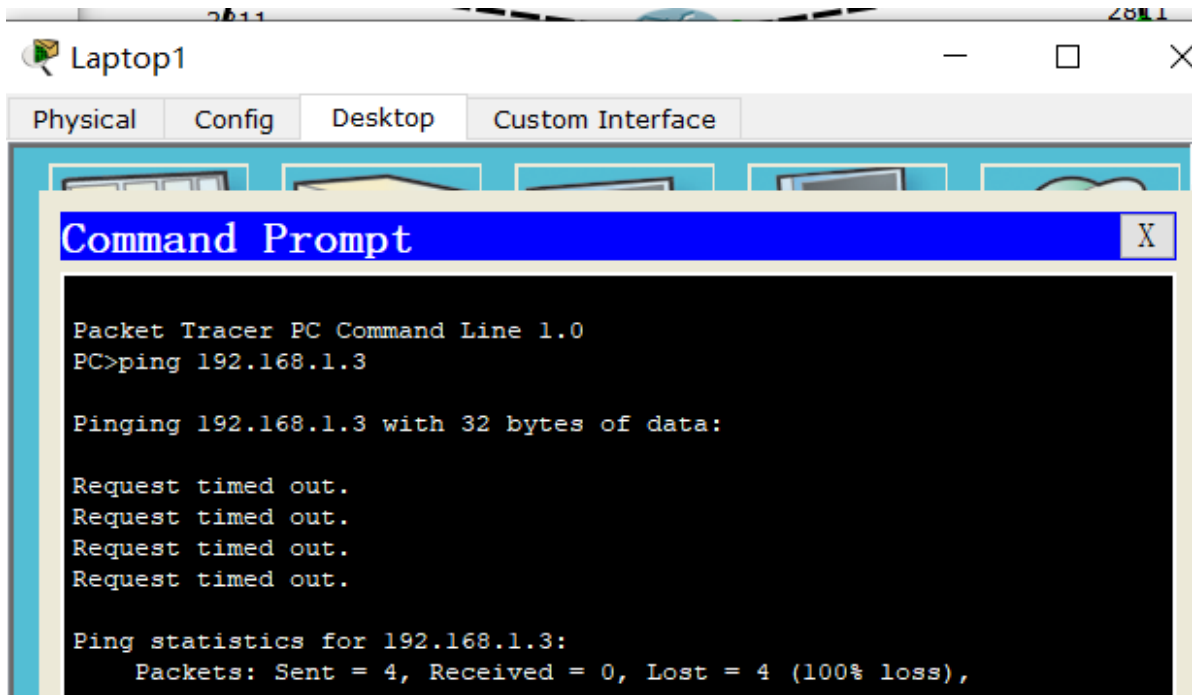


4.只有 PC1 可以访问 Server1

PC1 ping Server1 成功



Laptop1 ping Server1 失败

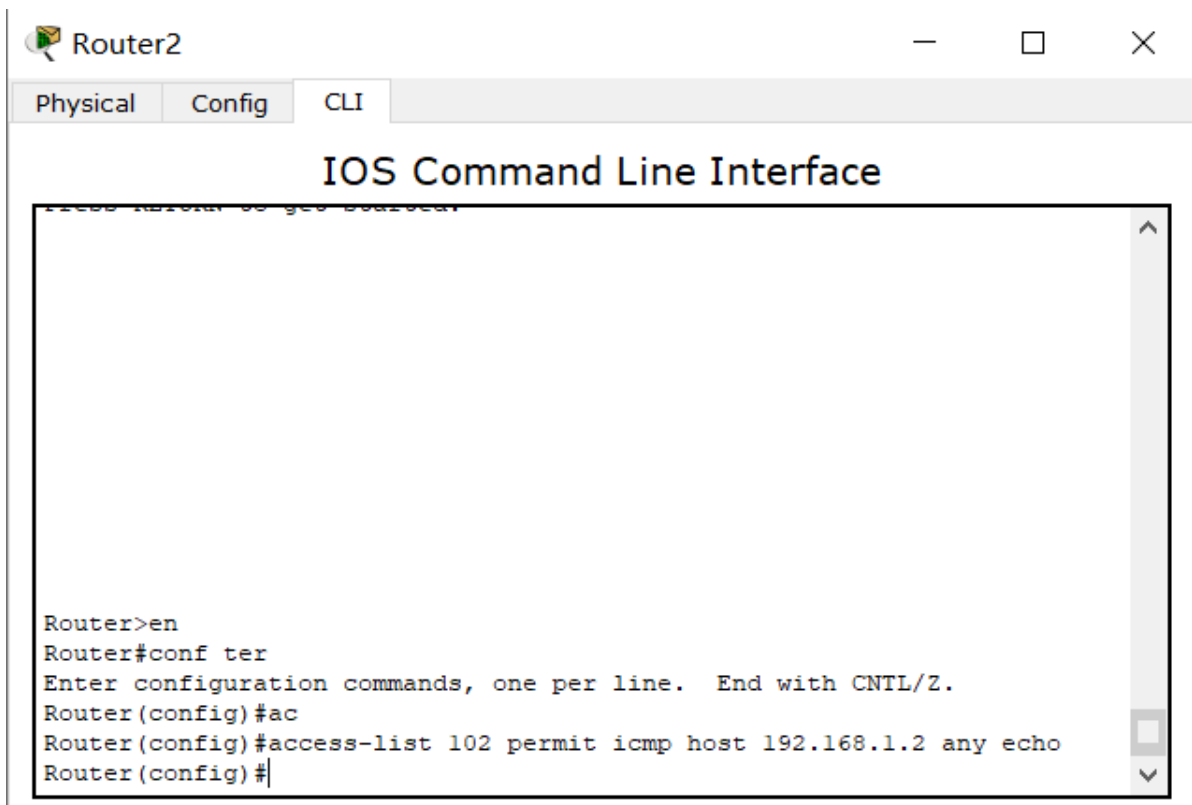


任务七

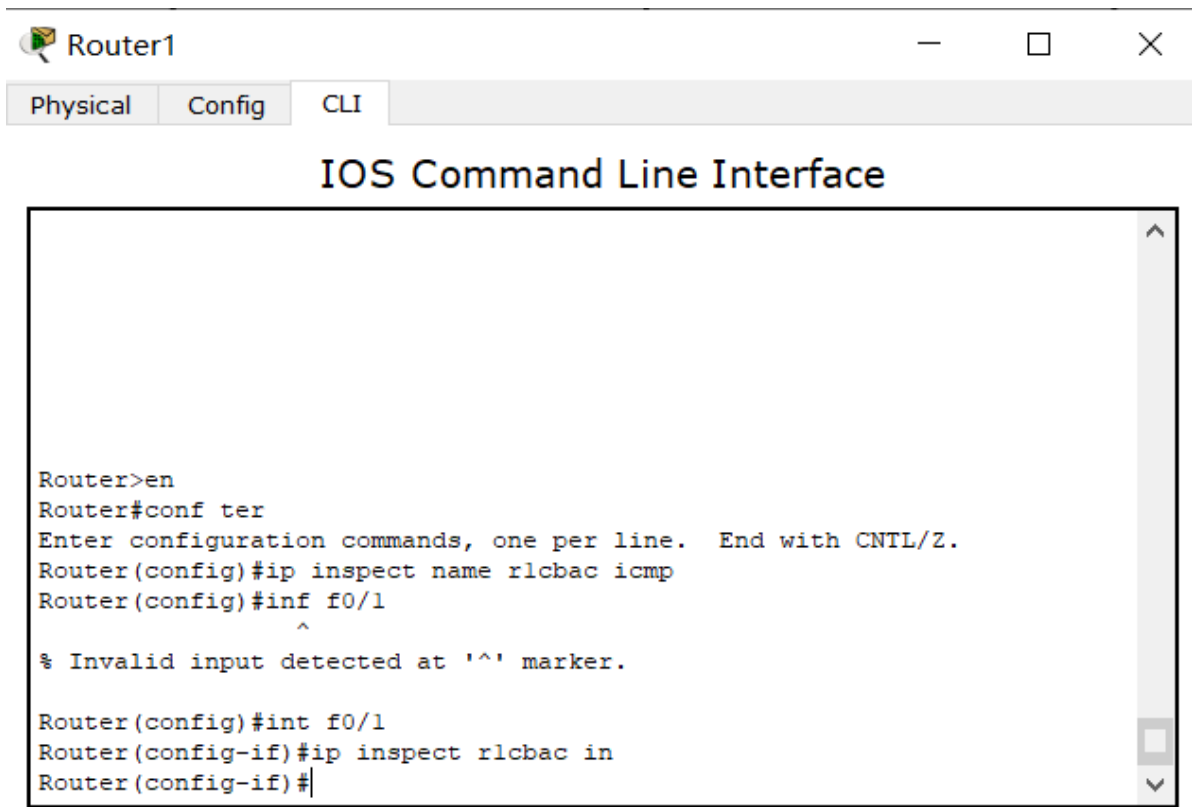
PC1 需要能够 ping 到网络中的所有中断，所以我们需要在 R2 和 R3 中加入 ACL 控制，允许 PC1 发送 ICMP 报文。同时在 R1 的 in 方向上配置 CBAC，允许 ICMP 报文流量。

ACL 和 CBAC 配置

R2 和 R3 允许 ICMP 流量的 ACL 配置如下，以 R2 的截图为例：



R1 上配置 CBAC 在 in 方向上，允许 ICMP 流量

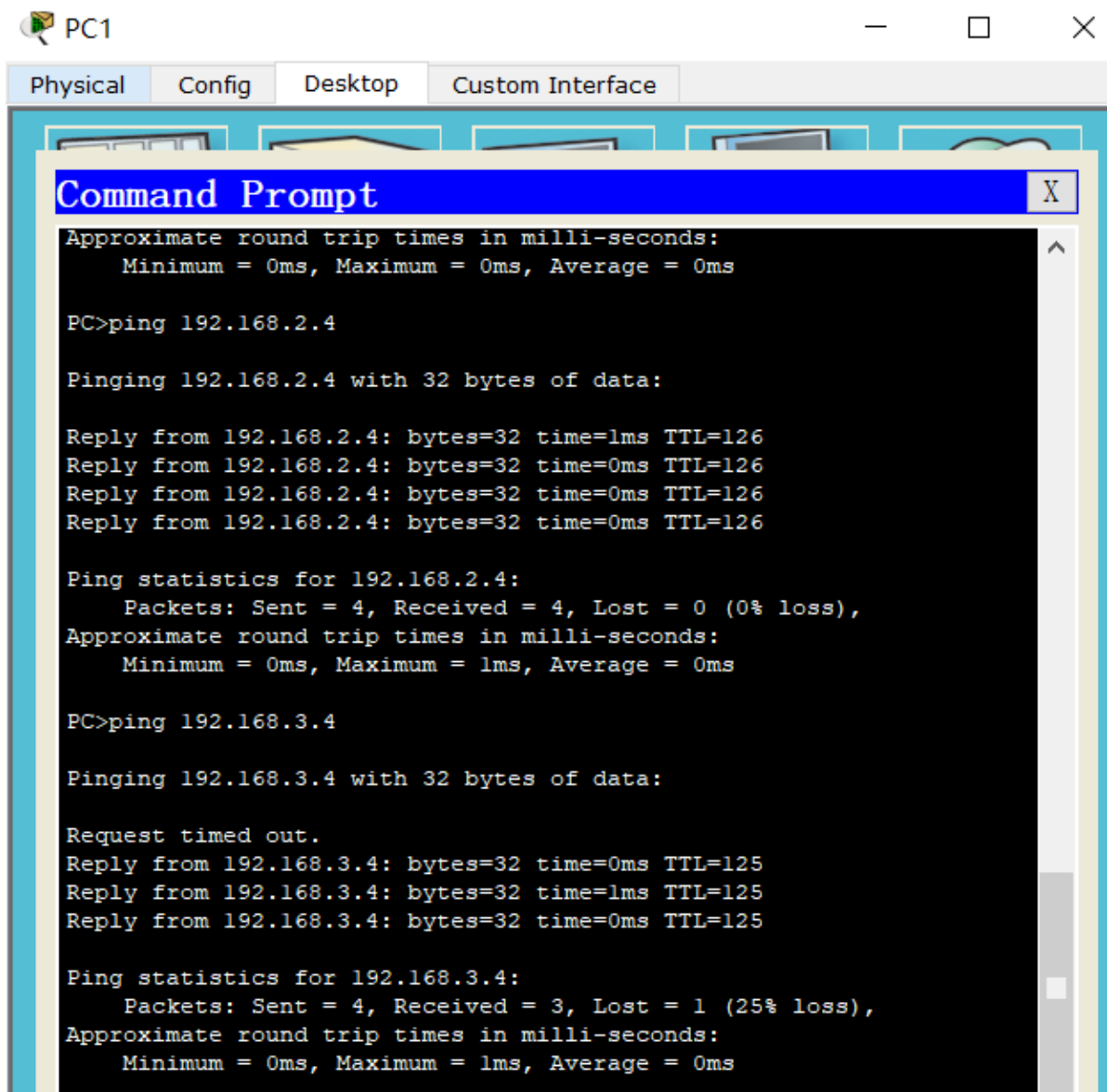


配置完成后，使用 `show access-lists` 命令查看 r1,r2,r3 上的 ACL 列表，下面依次为 R1 和 R3 的 ACL：

```
Router#show access-lists
Extended IP access list 101
 10 permit ip any host 192.168.1.4 (16 match(es))
 20 permit ip host 192.168.2.2 any (4 match(es))
 30 permit ip host 192.168.3.3 any
 40 permit ip host 192.168.2.3 host 192.168.1.2
 50 permit ip host 192.168.3.2 host 192.168.1.2 (8 match(es))
Router#
Router#show access-lists
Extended IP access list 103
 10 permit ip any host 192.168.3.3
 20 permit ip host 192.168.1.4 any (4 match(es))
 30 permit ip host 192.168.2.2 any (4 match(es))
 40 permit ip host 192.168.1.2 host 192.168.3.2 (8 match(es))
 50 permit ip host 192.168.2.3 host 192.168.3.2
 60 permit icmp host 192.168.1.2 any echo (4 match(es))
Router#
```

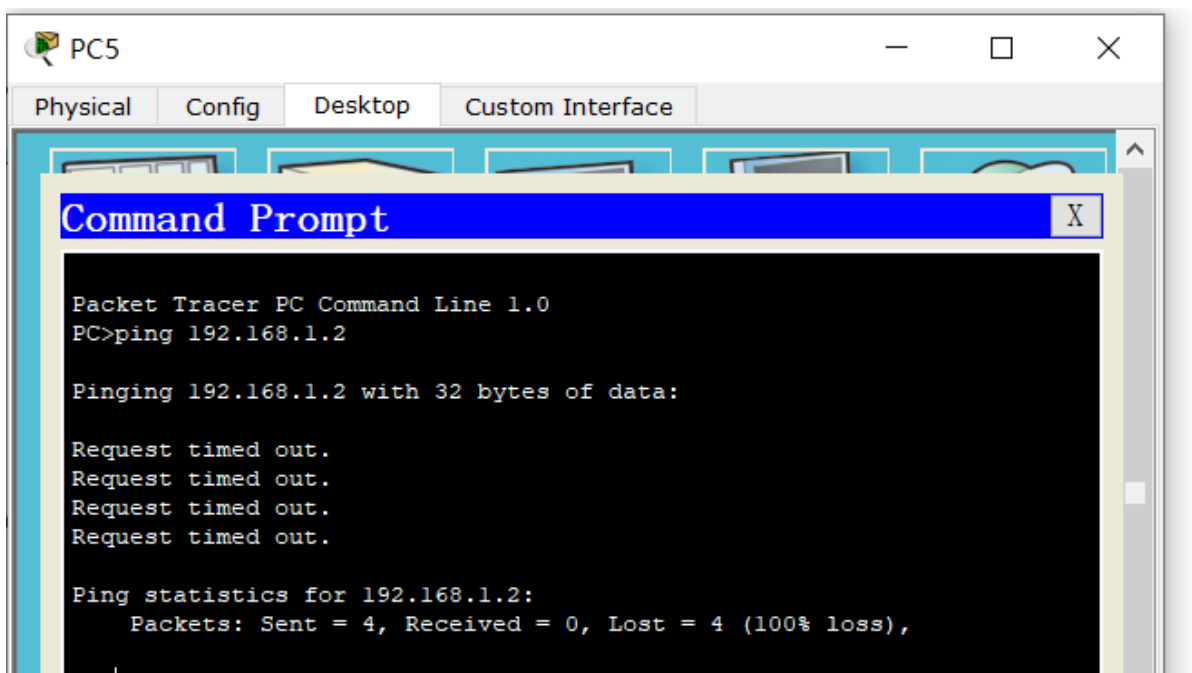
Ping 测试

PC1 ping laptop4 成功， PC1 ping PC5 成功



但是反过来，因为任务六访问控制的存在，其他子网下的设备依然无法访问 PC1。保证了访问权限。

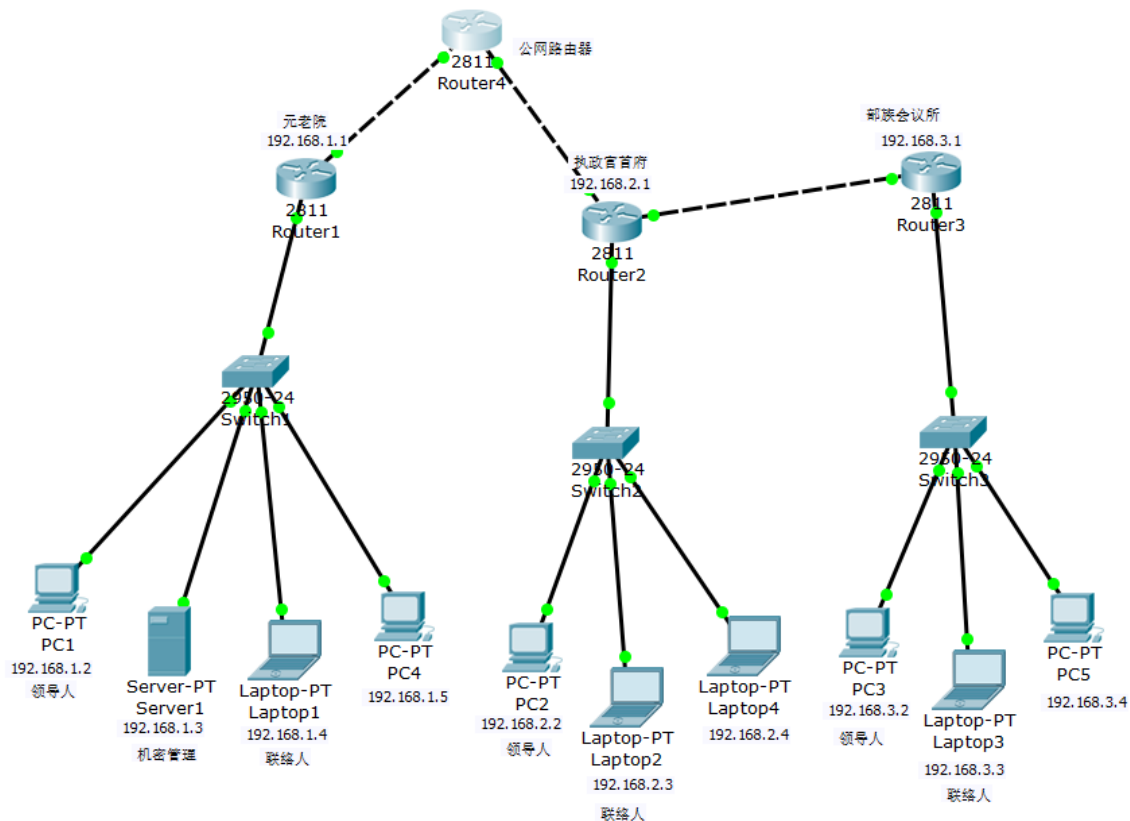
如 PC5 ping PC1 失败:



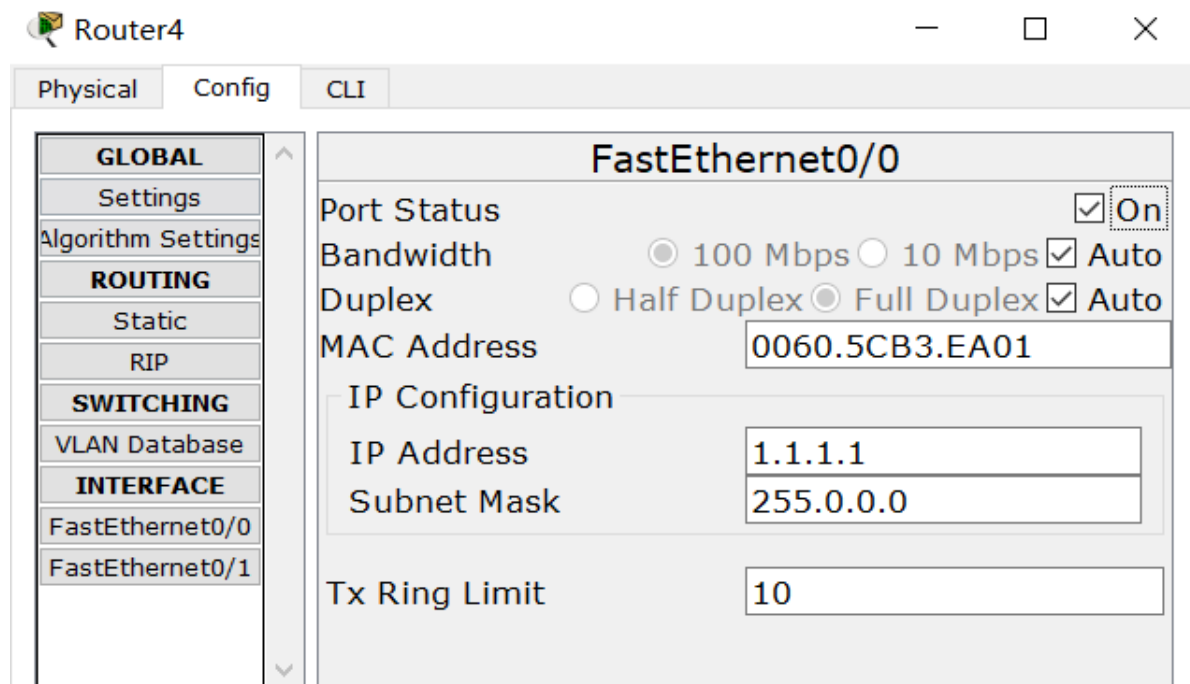
任务八

拓扑改变与IP配置

更改后的网络拓扑如下：



为 R4 的两个端口分配 1.1.1.1 和 2.2.2.1 IP。以 0/0 接口为例：



之后将 R1, R2 和 R4 相连的端口IP置为 1.1.1.2 和 2.2.2.2, 以 R1 为例：

Router1

Physical

Config

CLI

GLOBAL

Settings

Algorithm Settings

ROUTING

Static

RIP

SWITCHING

VLAN Database

INTERFACE

FastEthernet0/0

FastEthernet0/1

FastEthernet0/0

Port Status ☒ On

Bandwidth ☒ 100 Mbps ☐ 10 Mbps ☒ Auto

Duplex ☐ Half Duplex ☒ Full Duplex ☒ Auto

MAC Address 0010.116B.9B01

IP Configuration

IP Address 1.1.1.2

Subnet Mask 255.0.0.0

Tx Ring Limit 10

之后取消接口上的 ACL 规则。

思考题：在搬迁之后，使用配置静态路由的方法将无让各个在搬迁之后，使用配置静态路由的方法将无让各个权力机构正常通信，请简述原因。

一方面，在VPN协商过程中，IP地址可能不是这3个子网里存在的IP地址(192.168.x.x)，所以可能无法协商，导致无法通信。

另一方面，在大规模网络中，不同的子网可能用了相同的IP地址。配置静态路由会让路由器不知道转发到有相同IP的哪个子网那里，因此配置静态路由无法让各个部门正常通信。

配置IPSec VPN

以 Router1 为例

1.首先配置 ISAKMP，加密算法使用 3des，哈希算法使用 md5，密钥协商使用 DH5，并设置共享密码为 liuhz18。VPN SET 名称为 liuhz18_vpnset，MAP名称为 liuhz18_vpnmap。



Router1

Physical Config CLI

IOS Command Line Interface

```
Router>
Router>en
Router#conf ter
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#cr
Router(config)#crypto isa
Router(config)#crypto isakmp pol
Router(config)#crypto isakmp policy 1
Router(config-isakmp)#en
Router(config-isakmp)#encryption 3des 加密算法为3des
Router(config-isakmp)#ha
Router(config-isakmp)#hash md5 哈希算法为md5
Router(config-isakmp)#au
Router(config-isakmp)#authentication pre-share
Router(config-isakmp)#gr
Router(config-isakmp)#group 5 秘钥协商DH5
Router(config-isakmp)#exit
Router(config)#cr
Router(config)#crypto is 对等路由器预共享密码liuhz18
Router(config)#crypto isakmp key liuhz18 address 2.2.2.2
Router(config)#
Router(config)#ac
Router(config)#exit
```

2.之后配置 IPsec, 并且允许 192.168.2.x 的流量, 截图中有详细过程说明。



Router1

Physical Config CLI

IOS Command Line Interface

```
Router(config-if)#exit
Router(config)#ac 允许192.168.2.0的流量
Router(config)#access-list 101 permit ip 192.168.1.0 0.0.0.255
192.168.2.0 0.0.0.255
Router(config)#acr
Router(config)#cr
Router(config)#crypto ipsec tra IPsec转换集liuhz18_vpnset
Router(config)#crypto ipsec transform-set liuhz18_vpnset esp-3des esp-
md5-hmac
Router(config)#cr
Router(config)#crypto ip
Router(config)#crypto ipsec se 加密map为liuhz18_vpnmap
Router(config)#crypto map liuhz18_vpnmap 1 ipsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer
and a valid access list have been configured.
Router(config-crypto-map)#set
Router(config-crypto-map)#set peer 2.2.2.2 对等体为R2
Router(config-crypto-map)#match address 101 指定ACL 101的流量为VPN流量
Router(config-crypto-map)#set ra
Router(config-crypto-map)#set tra
Router(config-crypto-map)#set transform-set liuhz18_vpnset
Router(config-crypto-map)#exit 设置transform-set
```

之后给 R1 的接口配置VPN, 同时配置默认路由。

```

Router(config)#int f0/0
Router(config-if)#cr
Router(config-if)#crypto map liuhz18_vpnmap
*Jan  3 07:16:26.785: %CRYPTO-6-ISA_KMP_ON_OFF: ISAKMP is ON
Router(config-if)#exit
Router(config)#int f0/0|
Router(config-if)#ip ac
Router(config-if)#ip access-group 102 out
Router(config-if)#exit
Router(config)#ip route 0.0.0.0 0.0.0.0 1.1.1.1
Router(config)#

```

R2 的配置和 R1 类似，注意对应的VPN名称应保持一致。在配置路由时，为了允许 192.168.1.x 和 192.168.3.x 子网中的主机访问，需要配置这2条静态路由：

Router2

Physical Config CLI

GLOBAL

Settings

Algorithm Settings

ROUTING

Static

RIP

SWITCHING

VLAN Database

INTERFACE

FastEthernet0/0

FastEthernet0/1

FastEthernet1/0

Static Routes

Network

192.168.3.0

Mask

255.255.255.0

Next Hop

10.2.3.2

Add

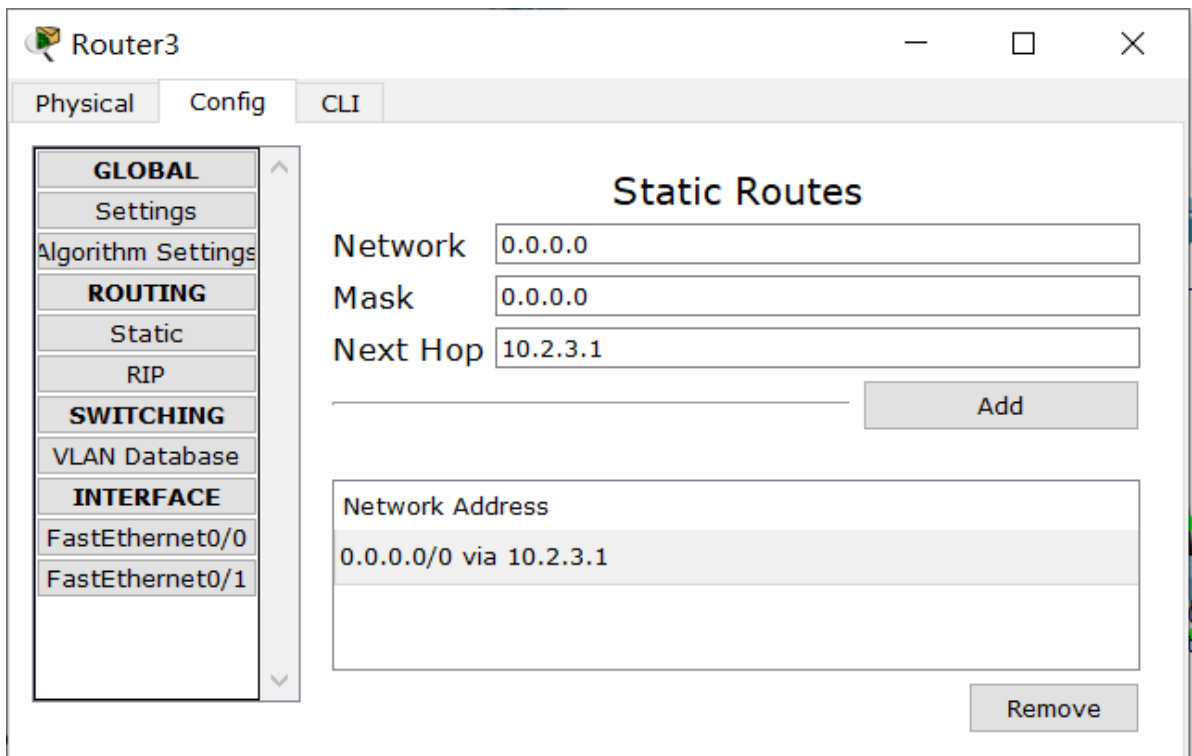
Network Address

0.0.0.0/0 via 2.2.2.1

192.168.3.0/24 via 10.2.3.2

Remove

R3 同理需要配置默认路由：



查看配置信息

下面使用 `show running-config` 命令检验是否配置成功，以 R2 为例：

R2 的 IPsec 配置信息如下：

```
!
crypto isakmp policy 1
  encr 3des
  hash md5
  authentication pre-share
  group 5
!
crypto isakmp key liuhz18 address 1.1.1.2
!
!
!
crypto ipsec transform-set liuhz18_vpnset esp-3des esp-md5-hmac
!
crypto map liuhz18_vpnmap 1 ipsec-isakmp
  set peer 1.1.1.2
  set transform-set liuhz18_vpnset
  match address 102
!
```

R2 的接口配置信息如下：

```
interface FastEthernet0/0
  ip address 2.2.2.2 255.0.0.0
  duplex auto
  speed auto
  crypto map liuhz18_vpnmap
!
interface FastEthernet0/1
  ip address 10.2.3.1 255.255.255.0
  duplex auto
  speed auto
!
interface FastEthernet1/0
  ip address 192.168.2.1 255.255.255.0
  duplex auto
  speed auto
```

R1 的接口配置信息如下:

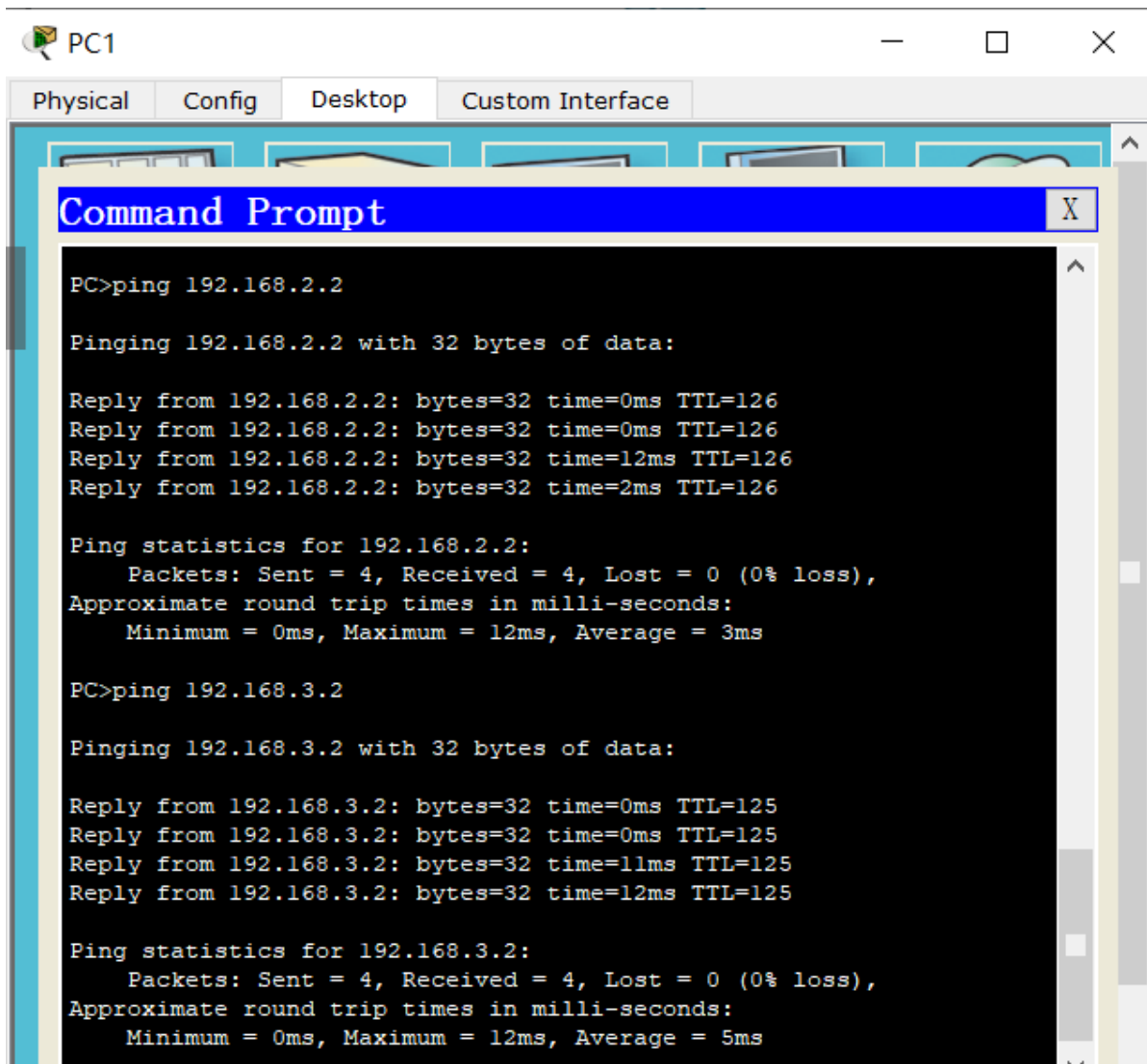
```
interface FastEthernet0/0
 ip address 1.1.1.2 255.0.0.0
 duplex auto
 speed auto
 crypto map liuhz18_vpnmap
!
interface FastEthernet0/1
 ip address 192.168.1.1 255.255.255.0
 duplex auto
 speed auto
!
```

R1 的 ACL 配置如下:

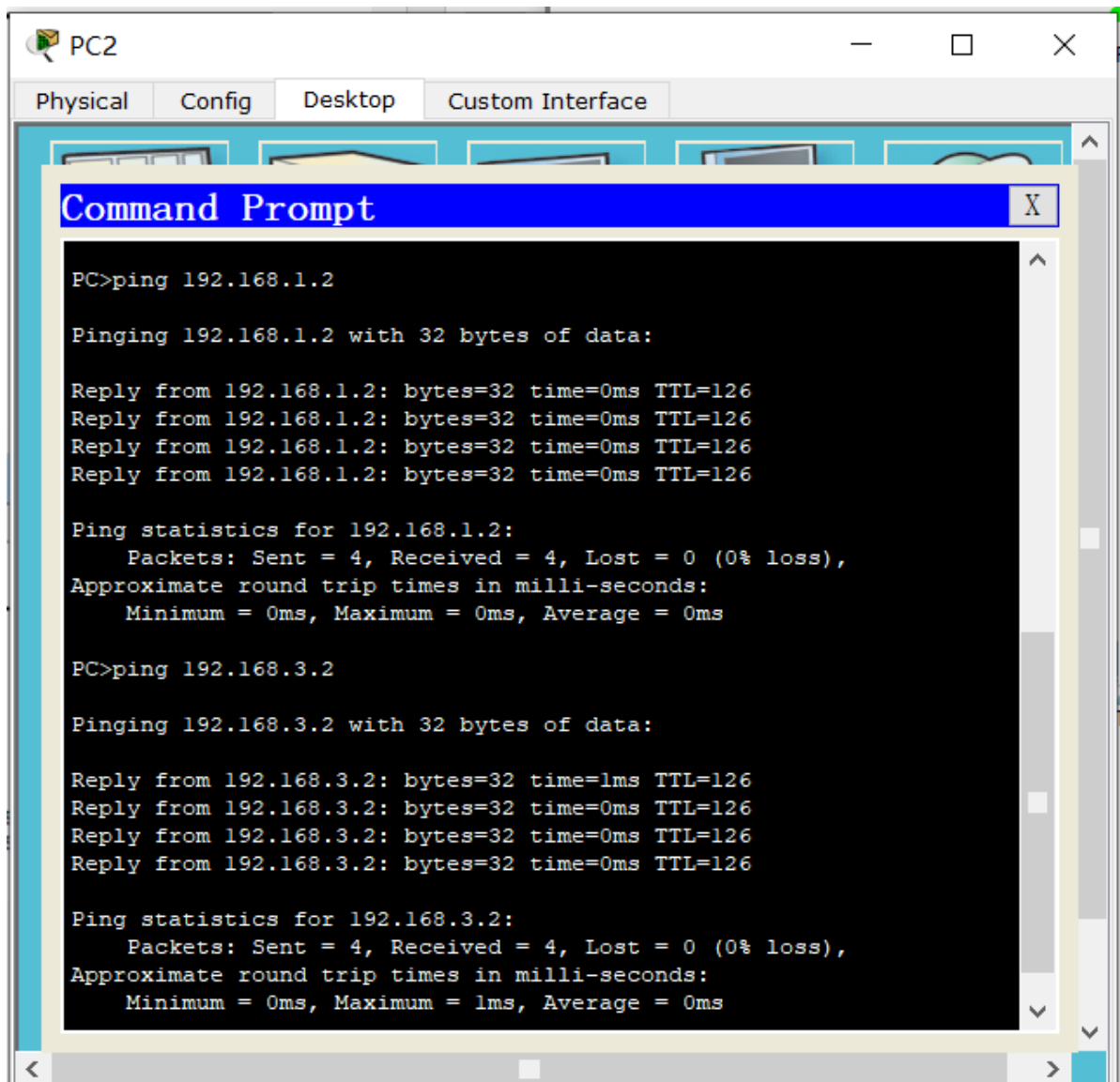
```
:
ip classless
ip route 0.0.0.0 0.0.0.0 1.1.1.1
!
ip flow-export version 9
!
!
access-list 101 permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
access-list 101 permit ip 192.168.1.0 0.0.0.255 192.168.3.0 0.0.0.255
!
:
```

进行 PING 测试

1. PC1 ping PC2, PC3 均成功



2. PC2 ping PC1, PC3 均成功



The screenshot shows a virtual PC2 window with a desktop background. A 'Command Prompt' window is open, displaying the results of two ping commands. The first command is 'ping 192.168.1.2', which shows four successful replies with 0ms round-trip times. The second command is 'ping 192.168.3.2', which shows four successful replies with 1ms round-trip times. Both tests show 0% packet loss.

```
PC2
Physical Config Desktop Custom Interface

Command Prompt

PC>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time=0ms TTL=126
Reply from 192.168.1.2: bytes=32 time=0ms TTL=126
Reply from 192.168.1.2: bytes=32 time=0ms TTL=126
Reply from 192.168.1.2: bytes=32 time=0ms TTL=126

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

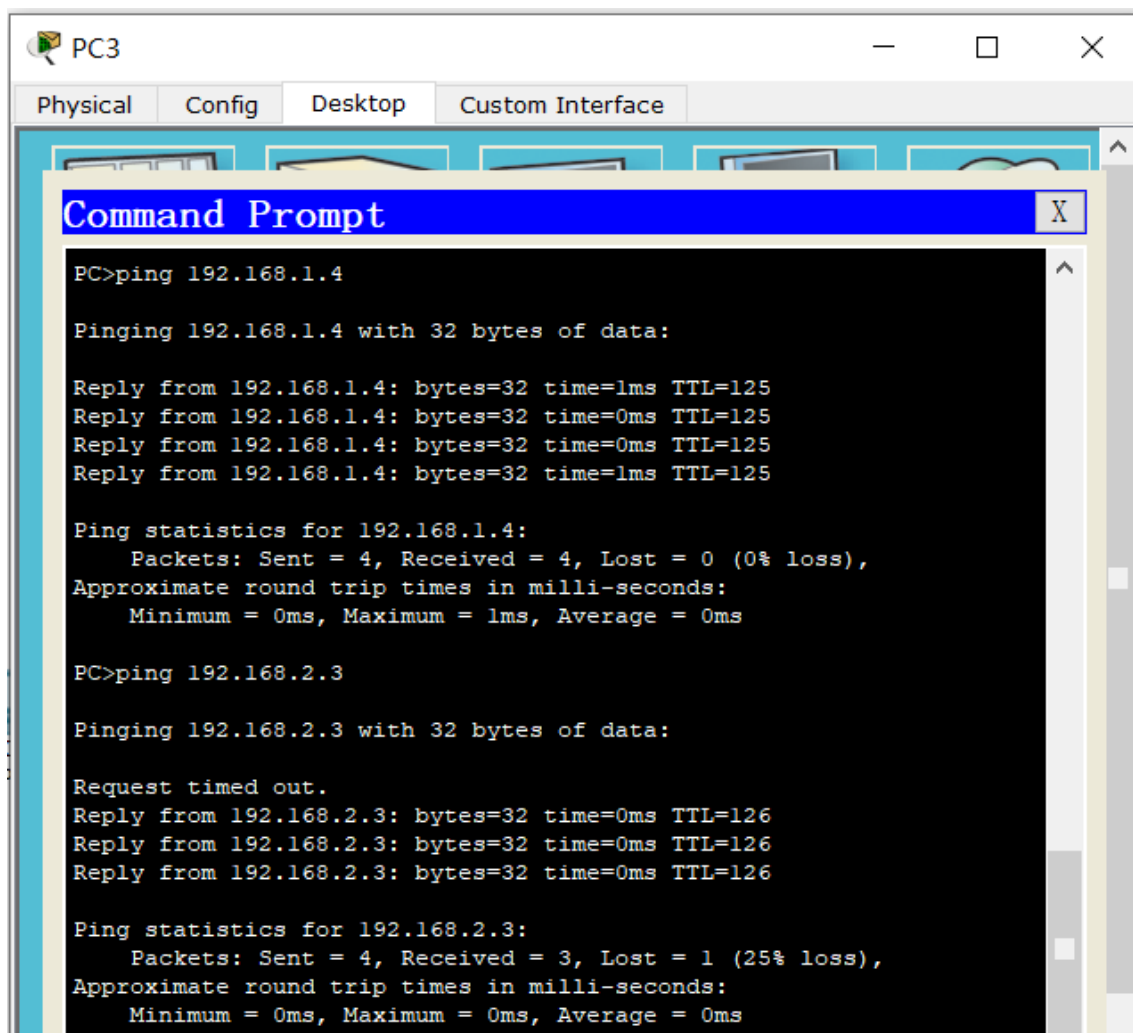
PC>ping 192.168.3.2

Pinging 192.168.3.2 with 32 bytes of data:

Reply from 192.168.3.2: bytes=32 time=1ms TTL=126
Reply from 192.168.3.2: bytes=32 time=0ms TTL=126
Reply from 192.168.3.2: bytes=32 time=0ms TTL=126
Reply from 192.168.3.2: bytes=32 time=0ms TTL=126

Ping statistics for 192.168.3.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

3. PC3 ping Laptop1, Laptop2 均成功



从上述测试可以看出，引入公网之后，三个子网之间的设备依然可以相互通信。

仿真抓包分析：隧道模式

判断得出上面的IPSec VPN使用了隧道。

执行 PC1 ping PC2，查看抓包结果。

下图是ICMP报文经过R1之前的结构

PDU Formats

Ethernet II

0	4	8	14	19	ytes
PREAMBLE: 101010...1011		DEST MAC:	SRC MAC: 00D0.FFE7		
TYPE: 0x800	DATA (VARIABLE LENGTH)			FCS: 0x0	

IP

0	4	8	16	19	31	Bits
4	IHL	DSCP: 0x0	TL: 128			
ID: 0x34			0x	0x0		
TTL: 128	PRO: 0x1	CHKSUM				
SRC IP: 192.168.1.2						
DST IP: 192.168.2.2						
OPT: 0x0				0x0		
DATA (VARIABLE LENGTH)						

ICMP

0	8	16	31	Bits
TYPE:	CODE:	CHECKSUM		
ID: 0xb		SEQ NUMBER: 37		

下图是在 R1 出端口上的结构，可以看到 R1 将原来的 IP 包封装了一层，加上了 ESP，并且产生了新的 IP 头，没有破坏原有 IP 头。

PDU Formats

Ethernet II

0	4	8	14	19 bytes
PREAMBLE:	DEST	SRC MAC:		
101010...1011	MAC:	0010.116B		
TYPE:	DATA (VARIABLE LENGTH)	FCS:		
0x800		0x0		

IP

0	4	8	16	19	31 Bits
4	IHL	DSCP: 0x0	TL: 20		
ID: 0xa3	0x	0x0			
TTL: 255	PRO: 0x32	CHKSUM			
SRC IP: 1.1.1.2					
DST IP: 2.2.2.2					
OPT: 0x0	0x0				
DATA (VARIABLE LENGTH)					

ENCAPSULATING SECURITY PAYLOAD

0	8	16	31 Bits
ESP SPI: 862136785			
ESP SEQUENCE: 92			
ESP DATA ENCRYPTED WITH 3DES			
ESP DATA AUTHENTICATED WITH MD5			

IP

0	4	8	16	19	31 Bits
4	IHL	DSCP: 0x0	TL: 128		
ID: 0x34	0x	0x0			
TTL: 127	PRO: 0x1	CHKSUM			
SRC IP: 192.168.1.2					
DST IP: 192.168.2.2					
OPT: 0x0	0x0				
DATA (VARIABLE LENGTH)					

ICMP

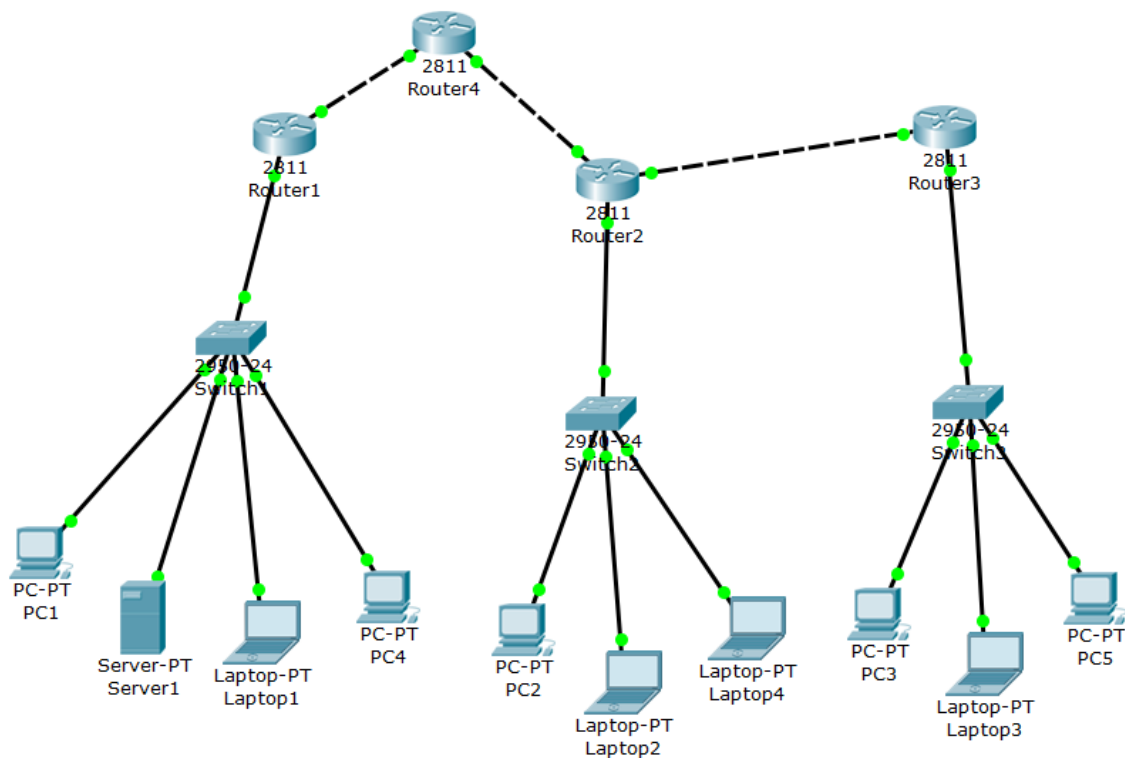
0	8	16	31 Bits
TYPE:	CODE:	CHECKSUM	
ID: 0xb	SEQ NUMBER: 37		

实际上从 R1 的配置信息中也可以知道使用了隧道模式，在 R1 上执行 `crypto ipsec transform-set` 命令结果如下，可以看到使用了隧道模式：

```
Router#show crypto ipsec transform-set
Transform set liuhzl8_vpnsset: {    { esp-3des esp-sha-hmac  }
will negotiate = { Tunnel,  },
```

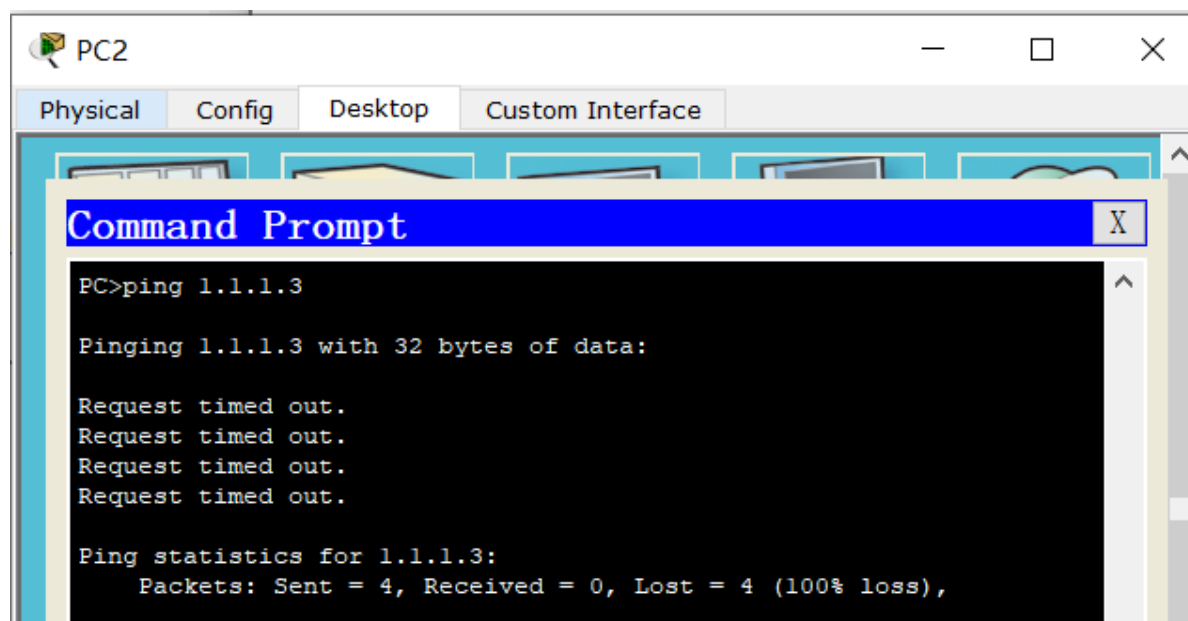
Bonus: 网络地址转换(NAT)探究

根据网络原理课上的知识, NAT用于将私有IP地址映射到公网IP地址,以此来减缓IP地址空间的消耗。主要应用于需要连接Internet但是主机没有公网IP的情况, 或者合并2个具有相同网络地址的内网。一般设置在边界路由器上。分为**静态NAT**、**动态NAT**和**PAT**。因为之前的实验已经有了边界路由器, 所以我直接使用本实验的拓扑进行配置和测试。



下面我将使用3种不同的NAT方式将 192.168.1.x 和 192.168.2.x 两个子网分别映射到 1.1.1.x 和 2.2.2.x 两个公网。边界路由为 R1, R2。

在**配置之前**, 用 PC2 ping 1.1.1.3 是无法成功的, 截图如下:



静态NAT

一对一映射，每个主机对应一个公网IP地址。地址映射如下

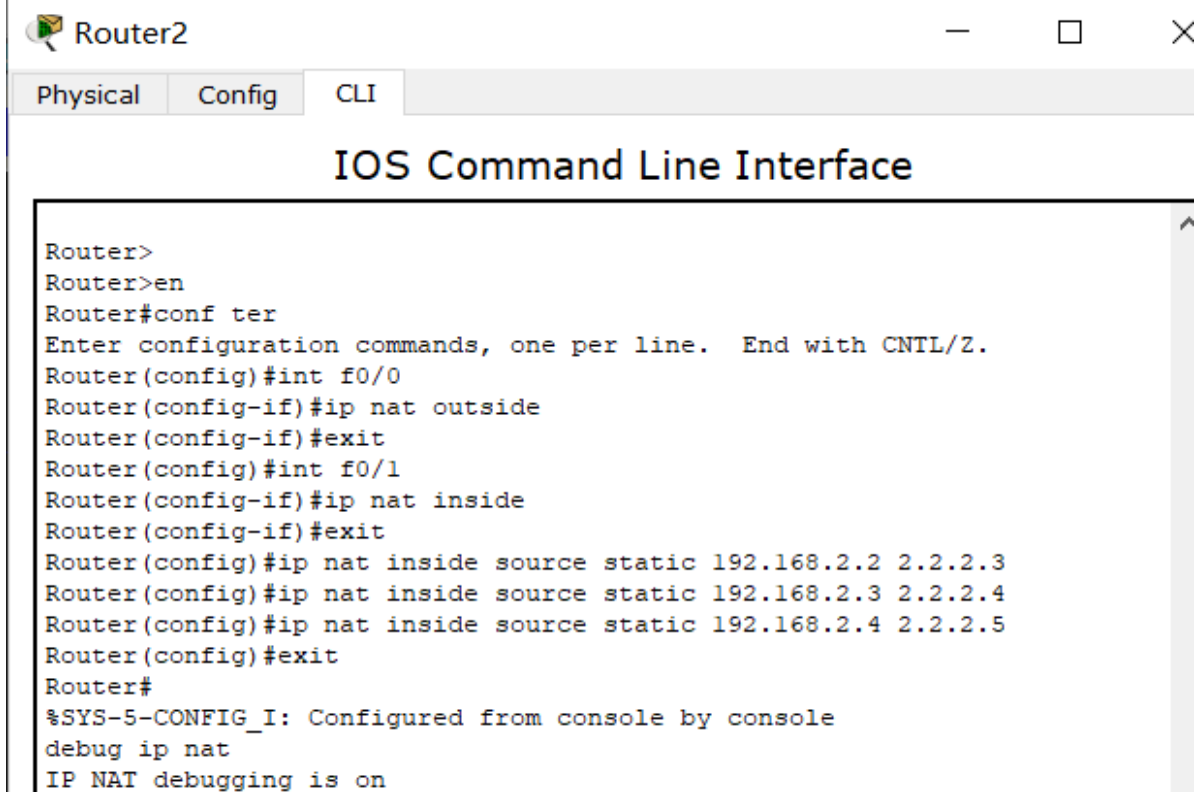
设备	私网IP	公网IP
pc1	192.168.1.2	1.1.1.3
server1	192.168.1.3	1.1.1.4
laptop1	192.168.1.4	1.1.1.5
pc4	192.168.1.5	1.1.1.6
pc2	192.168.2.2	2.2.2.3
laptop2	192.168.2.3	2.2.2.4
laptop4	192.168.2.4	2.2.2.5
r3	10.2.3.2	2.2.2.6

对**边界路由 R1** 进行静态NAT配置，分别设置内部端口和外部端口，以及IP地址映射。

```
Router(config)#int f0/0
Router(config-if)#ip nat outside
Router(config-if)#exit
Router(config)#int f0/1
Router(config-if)#ip nat inside
Router(config-if)#exit
Router(config)#ip nat source static 192.168.1.2 1.1.1.3
                                     ^
% Invalid input detected at '^' marker.

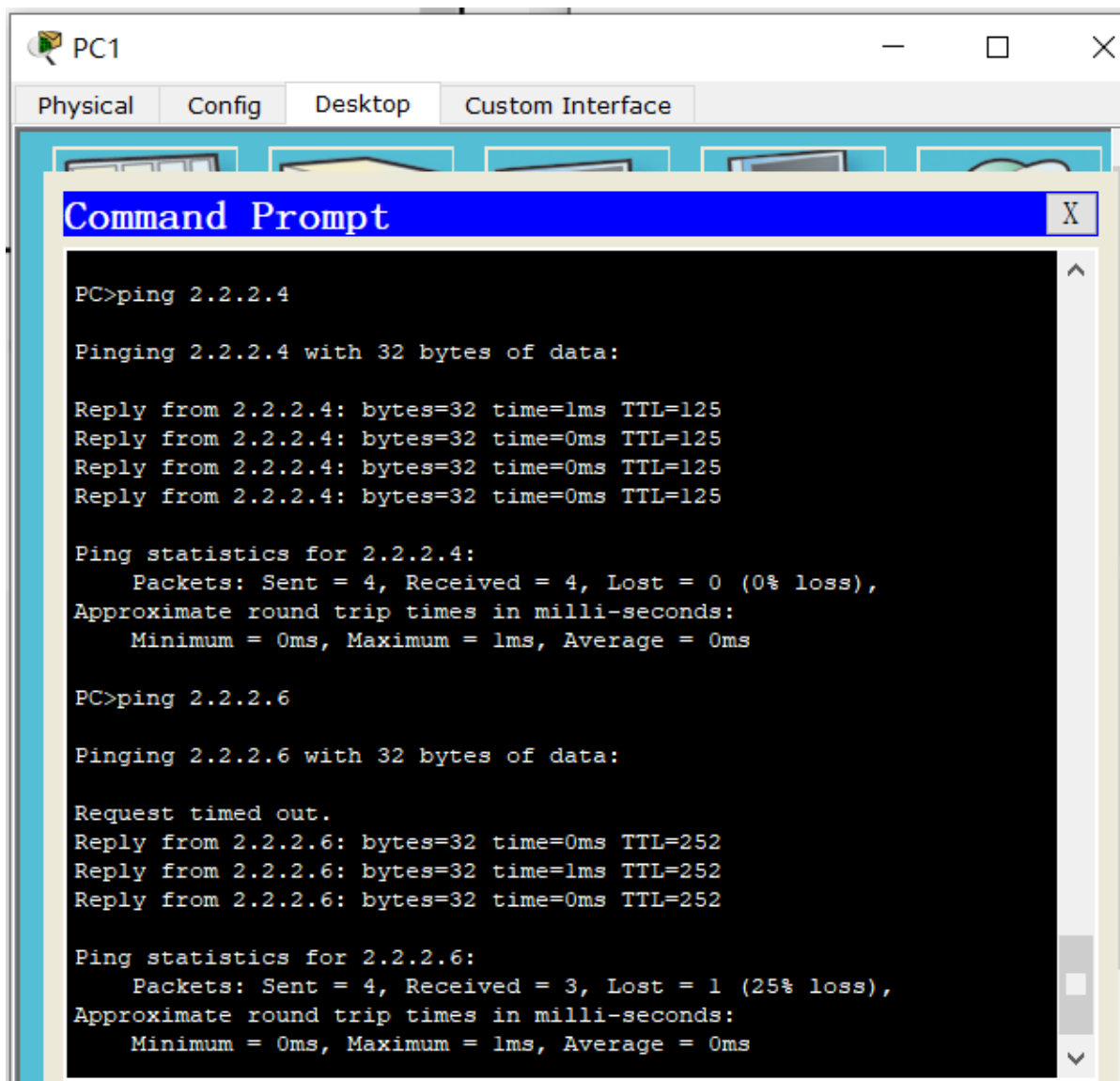
Router(config)#ip nat inside source static 192.168.1.2 1.1.1.3
Router(config)#ip nat inside source static 192.168.1.3 1.1.1.4
Router(config)#ip nat inside source static 192.168.1.4 1.1.1.5
Router(config)#ip nat inside source static 192.168.1.5 1.1.1.6
Router(config)#exit
```

同样，对**边界路由 R2** 进行静态NAT配置



下面进行ping测试举例

PC1 ping 2.2.2.4(laptop2) 和 2.2.2.6(r2)



可以看到可以成功ping到，并且Reply的地址变成了公网IP。说明静态NAT配置成功。

从R2的NAT debug log可以看出来，确实进行了地址转换

```
debug ip nat
IP NAT debugging is on
Router#
NAT: s=1.1.1.3, d=2.2.2.3->192.168.2.2 [57]

NAT: s=1.1.1.3, d=2.2.2.3->192.168.2.2 [58]

NAT: s=1.1.1.3, d=2.2.2.3->192.168.2.2 [59]

NAT: s=1.1.1.3, d=2.2.2.3->192.168.2.2 [60]

NAT: s=1.1.1.3, d=2.2.2.3->192.168.2.2 [61]

NAT: s=1.1.1.3, d=2.2.2.3->192.168.2.2 [62]

NAT: s=1.1.1.3, d=2.2.2.3->192.168.2.2 [63]
```

动态NAT

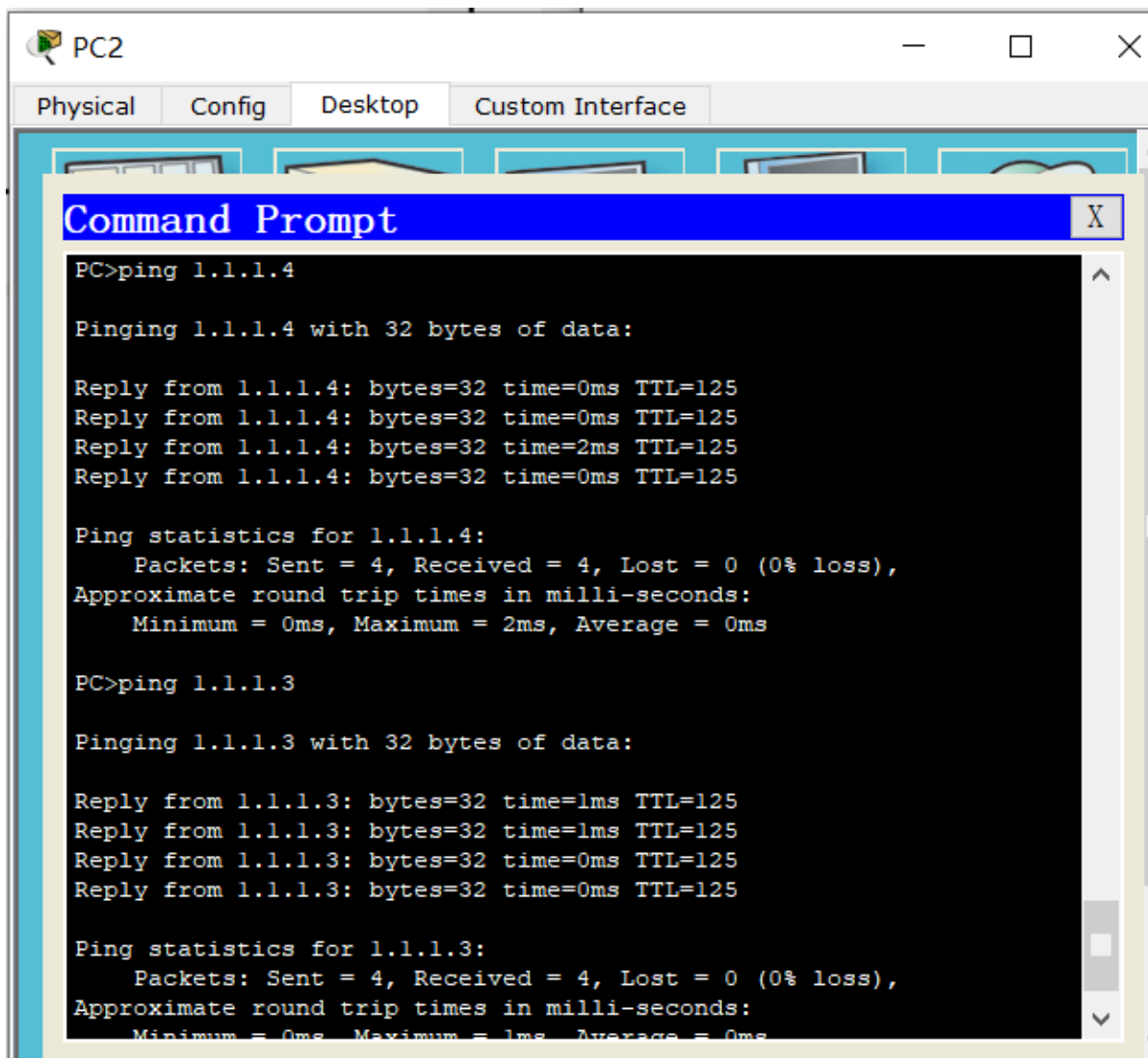
动态NAT使用IP池机制，可以实现将一个未注册IP地址映射到IP池中的一个地址。但是依然需要保证IP池中有足够的公网IP。如果IP池大小为N，那么子网下只有N台设备可以访问外网。

首先删除R1和R2上的静态NAT。并给连接子网的接口配置访问控制列表，编号为10，允许子网流量。之后定义公网IP池，名字为myippool。之后将10号ACL和公网IP池myippool关联即可。

下面以R2为例说明配置过程，R1同理即可。

```
Router#conf ter
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#no ip nat inside source static 192.168.2.2 2.2.2.3
Router(config)#no ip nat inside source static 192.168.2.3 2.2.2.4
Router(config)#no ip nat inside source static 192.168.2.4 2.2.2.5
Router(config)#ac
Router(config)#access-list 10 permit 192.168.2.0 0.0.0.255
Router(config)#access-list 10 permit 192.168.1.0 0.0.0.255
Router(config)#ip nat pool myippool 2.2.2.3 2.2.2.5 netmask
255.255.255.0
Router(config)#ip nat inside source list 10 pool myippool
Router(config)#exit
```

进行Ping测试，以PC2 ping 1.1.1.3 和 1.1.1.4为例



可以看到 ping 测试成功，并且Reply地址为映射后的公网地址。

抓包可以看到，PC2 的私有IP在经过 R2 时，**通过NAT进行了转换**

OSI Model

Inbound PDU Details

Outbound PDU Details

At Device: Router2
Source: PC2
Destination: 1.1.1.3

In Layers

Layer7
Layer6
Layer5
Layer4
Layer 3: IP Header Src. IP: <u>192.168.2.2</u> , Dest. IP: 1.1.1.3 ICMP Message Type: 8
Layer 2: Ethernet II Header 0060.3EC9.4581 >> 000C.CFBB.828B
Layer 1: Port FastEthernet1/0

Out Layers

Layer7
Layer6
Layer5
Layer4
Layer 3: IP Header Src. IP: <u>2.2.2.3</u> , Dest. IP: 1.1.1.3 ICMP Message Type: 8
Layer 2: Ethernet II Header 0001.97D3.2A01 >> 00D0.584D.A502
Layer 1: Port(s): FastEthernet0/0

1. FastEthernet1/0 receives the frame.

查看一下 R2 上的 NAT Table, 有相应的表项, 配置成功。

```
Router#show ip nat translations
Pro  Inside global      Inside local          Outside local          Outside
-----
icmp 2.2.2.3:86         192.168.2.2:86       1.1.1.3:86            1.1.1.3:86
icmp 2.2.2.3:87         192.168.2.2:87       1.1.1.3:87            1.1.1.3:87
icmp 2.2.2.3:88         192.168.2.2:88       1.1.1.3:88            1.1.1.3:88
icmp 2.2.2.3:89         192.168.2.2:89       1.1.1.3:89            1.1.1.3:89
---  2.2.2.6            10.2.3.2              ---                    ---
```

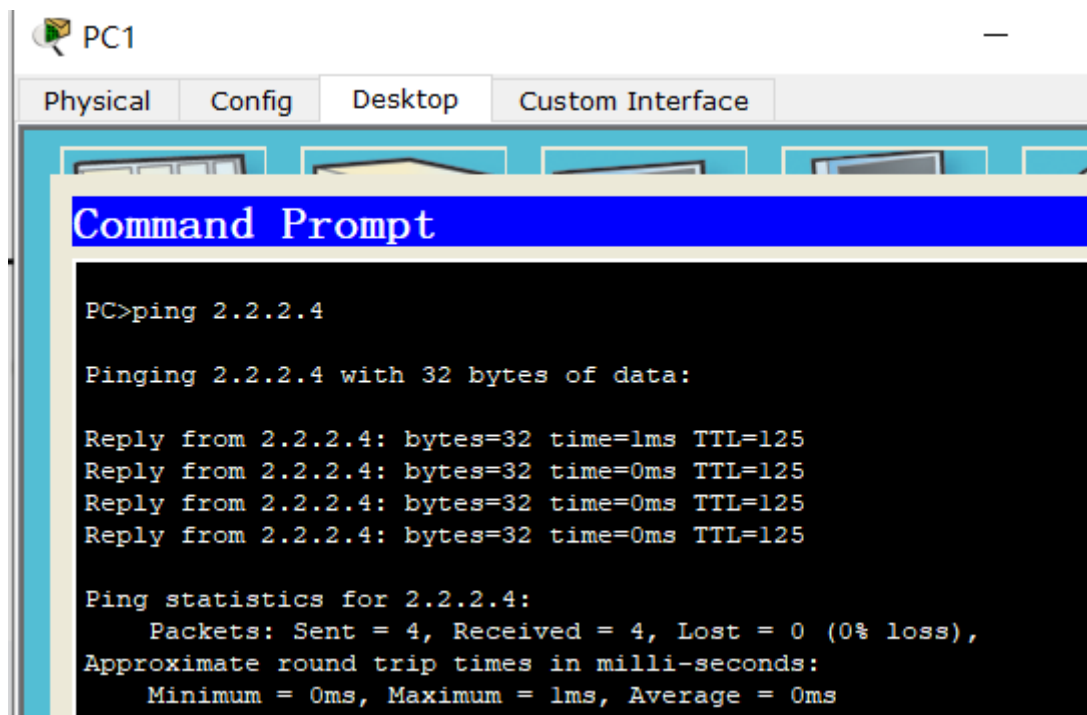
PAT

称为**端口地址转换**, 是一种动态地址转换。通过多个端口映射到多个私有IP来实现一个公有IP即可管理大量终端的功能。

PAT 的配置十分简单, 只需要在动态 NAT 的配置上加一个 `overload` 选项。以 R1 的配置为例:

```
Router#conf ter
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#no ip nat inside source list 10 pool myippool
Router(config)#ip nat inside source list 10 pool myippool overload
Router(config)#exit
```

再次进行Ping测试, PC1 ping 2.2.2.4(laptop2)



可以看到依然是成功的。

观察 R2 上的 NAT Table, 可以看到 NAT 表项中的记录

```

Router#show ip nat translations
Pro  Inside global      Inside local          Outside local          Outside global
icmp 2.2.2.3:134         192.168.2.2:134      1.1.1.3:134           1.1.1.3:134
icmp 2.2.2.3:135         192.168.2.2:135      1.1.1.3:135           1.1.1.3:135
icmp 2.2.2.3:136         192.168.2.2:136      1.1.1.3:136           1.1.1.3:136
icmp 2.2.2.3:137         192.168.2.2:137      1.1.1.3:137           1.1.1.3:137
---  2.2.2.6             10.2.3.2             ---                    ---
  
```

在 R4 上抓包, 可以看到地址都被正确映射到公网 IP。

PDU Information at Device: Router4

OSI Model	Inbound PDU Details	Outbound PDU Details
At Device: Router4		
Source: PC1		
Destination: 2.2.2.4		
In Layers		Out Layers
Layer7		Layer7
Layer6		Layer6
Layer5		Layer5
Layer4		Layer4
Layer 3: IP Header Src. IP: 1.1.1.3, Dest. IP: 2.2.2.4 ICMP Message Type: 8		Layer 3: IP Header Src. IP: 1.1.1.3, Dest. IP: 2.2.2.4 ICMP Message Type: 8
Layer 2: Ethernet II Header 0010.116B.9B01 >> 00D0.584D.A501		Layer 2: Ethernet II Header 00D0.584D.A502 >> 0001.97D3.2A01
Layer 1: Port FastEthernet0/0		Layer 1: Port(s): FastEthernet0/1

1. FastEthernet0/0 receives the frame.

实际上，内网地址访问外网地址时，**PAT对外的地址只有一个**。尽管我使用了动态NAT中配置的地址池，但 R1 只会将所有私网IP转换为同一个公网IP，在本实验中为 1.1.1.3。