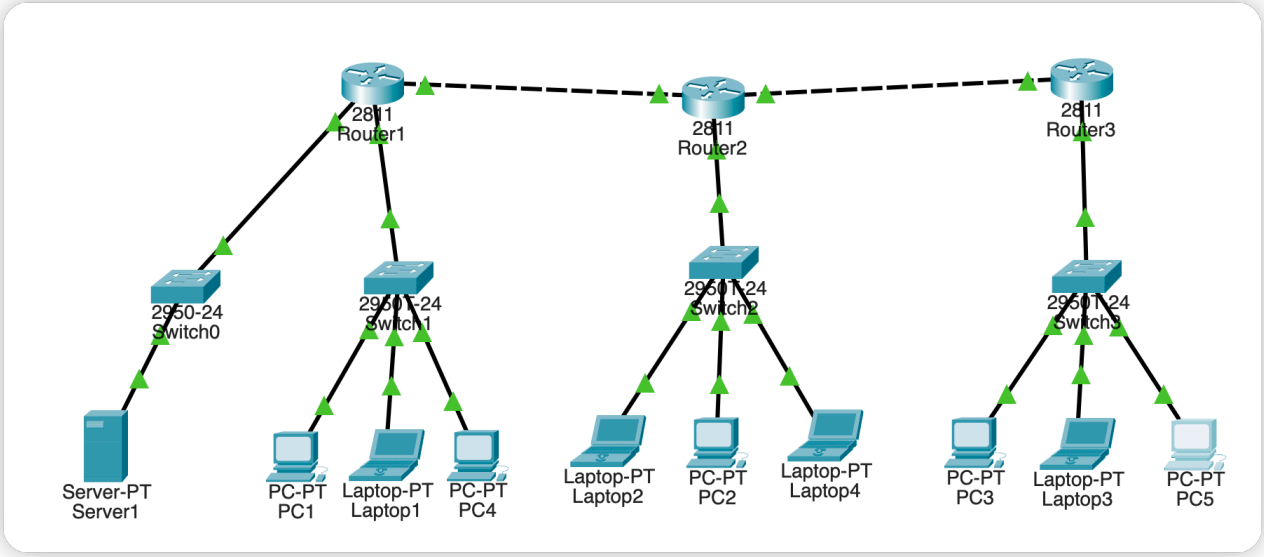
计14 汪隽立 2021012957

HW2

6.“三权”间的权限控制

新的网络拓扑图如下所示：



各终端设备的IP地址及权限级别如下表所示：

设备名称	部门	备注	IP	Gateway
PC1	元老院	领导人	192.168.1.2	192.168.1.1
Laptop1	元老院	联络人	192.168.1.4	192.168.1.1
PC4	元老院	/	192.168.1.5	192.168.1.1
Server1	元老院	机密管理	192.168.4.2	192.168.4.1
Laptop2	执政官首府	领导人	192.168.2.3	192.168.2.1
PC2	执政官首府	联络人	192.168.2.2	192.168.2.1
Laptop4	执政官首府	/	192.168.2.4	192.168.2.1
PC3	部族会议所	领导人	192.168.3.2	192.168.3.1
Laptop3	部族会议所	联络人	192.168.3.3	192.168.3.1
PC5	部族会议所	/	192.168.3.4	192.168.3.1

需要满足的访问权限为：

1. 各权力机构内部所有成员可以相互通信
2. 权力机构之间的相互通信只能通过联络人完成
3. 领导人之间可以相互通信
4. 只有PC1可以访问Server1

采用如下的思路实现：

- 对于1，各权力机构内部的成员通过交换机进行通信
- 对于2，在每个路由器，增加它对于它的子网的**Outbound ACL**：
  - **permit"src为另外两个子网的联络人"的情况**
  - **permit"dst为当前子网的联络人"的情况**
- 对于2，在每个路由器，增加它对于它的子网的**Inbound ACL**：
  - **permit"src为当前子网的联络人"的情况**
  - **permit"dst为另外两个子网的联络人"的情况**
- 对于3，在每个路由器，增加它对于它的子网的**Outbound ACL**
  - **permit"src为另外两个子网的领导人，dst为当前子网的领导人"的情况**

- 对于3，在每个路由器，增加它对于它的子网的**Inbound ACL**：
  - **permit"src为当前子网的领导人，dst为另外两个领导人"的情况**
- 对于4，对router1进行特殊处理：
  - 对于192.168.1.0子网的端口，增加Outbound ACL，permit"src为192.168.4.2，dst为192.168.1.2"的情况
  - 对于192.168.4.0子网的端口，增加Outbound ACL，permit "src为192.168.1.2，dst为192.168.4.2"的情况

采取“先特殊后一般”的思想，路由器设置的ACL（以Router1为例）为：

```
(config)# access-list 101 permit ip host 192.168.2.3 host 192.168.1.2
(config)# access-list 101 permit ip host 192.168.3.2 host 192.168.1.2
(config)# access-list 101 permit ip host 192.168.2.2 192.168.1.0 0.0.0.255
(config)# access-list 101 permit ip host 192.168.3.3 192.168.1.0 0.0.0.255
(config)# access-list 101 permit ip any host 192.168.1.4
(config)# access-list 101 permit ip host 192.168.4.2 host 192.168.1.2

(config-if)# ip access-group 101 out

(config)# access-list 103 permit ip host 192.168.1.2 host 192.168.2.3
(config)# access-list 103 permit ip host 192.168.1.2 host 192.168.3.2
(config)# access-list 103 permit ip host 192.168.1.4 any
(config)# access-list 103 permit ip 192.168.1.0 0.0.0.255 host 192.168.2.2
(config)# access-list 103 permit ip 192.168.1.0 0.0.0.255 host 192.168.3.3
(config)# access-list 103 permit ip host 192.168.1.2 host 192.168.4.2

(config-if)# ip access-group 103 in

```

```
19  (config)# access-list 102 permit ip host 192.168.1.2 host
    192.168.4.2
20
21  (config-if)# ip access-group 102 out
```

Router2，Router3的设置类似，只不过不需要处理有关192.168.4.0的内容。

- Router2（可以先忽略ICMP的部分）

```
Extended IP access list 101
    10 permit ip host 192.168.1.2 host 192.168.2.3 (4 match(es))
    20 permit ip host 192.168.3.2 host 192.168.2.3
    30 permit ip host 192.168.1.4 192.168.2.0 0.0.0.255 (8 match(es))
    40 permit ip host 192.168.3.3 192.168.2.0 0.0.0.255
    50 permit ip any host 192.168.2.2 (12 match(es))
    60 permit icmp host 192.168.1.2 any (4 match(es))
Extended IP access list 102
    10 permit ip host 192.168.2.3 host 192.168.1.2
    20 permit ip host 192.168.2.3 host 192.168.3.2
    30 permit ip host 192.168.2.2 any
    40 permit ip 192.168.2.0 0.0.0.255 host 192.168.1.4 (4 match(es))
    50 permit ip 192.168.2.0 0.0.0.255 host 192.168.3.3
```

- Router3:

```
Extended IP access list 101
    10 permit ip host 192.168.1.2 host 192.168.3.2
    20 permit ip host 192.168.2.3 host 192.168.3.2
    30 permit ip host 192.168.1.4 192.168.3.0 0.0.0.255
    40 permit ip host 192.168.2.2 192.168.3.0 0.0.0.255
    50 permit ip any host 192.168.3.3
    60 permit icmp host 192.168.1.2 any (8 match(es))
Extended IP access list 102
    10 permit ip host 192.168.3.2 host 192.168.1.2
    20 permit ip host 192.168.3.2 host 192.168.2.3
    30 permit ip host 192.168.3.3 any
    40 permit ip 192.168.3.0 0.0.0.255 host 192.168.1.4
    50 permit ip 192.168.3.0 0.0.0.255 host 192.168.2.2
```

下面是结果展示：

- 192.168.1.2（凯撒）：

```
Ping statistics for 192.168.4.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 192.168.1.4

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

C:\>ping 192.168.2.3

Pinging 192.168.2.3 with 32 bytes of data:

Reply from 192.168.2.3: bytes=32 time<1ms TTL=126
Reply from 192.168.2.3: bytes=32 time=1ms TTL=126
Reply from 192.168.2.3: bytes=32 time<1ms TTL=126
Reply from 192.168.2.3: bytes=32 time<1ms TTL=126

Ping statistics for 192.168.2.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>ping 192.168.2.4

Pinging 192.168.2.4 with 32 bytes of data:

Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.

Ping statistics for 192.168.2.4:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>
```

ping Server

ping 子网内其他设备

ping 其他首领

ping 其他普通设备（unreachable）

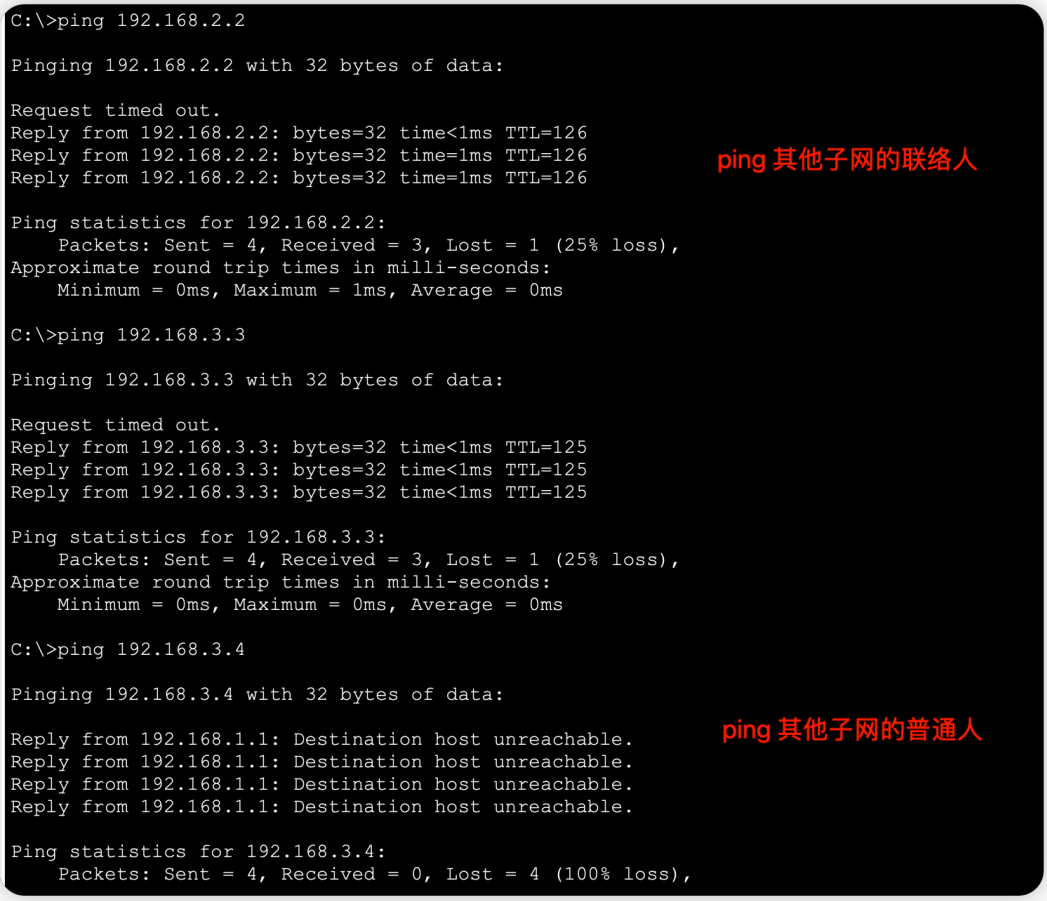
- 192.168.1.4（子网1的联络人）：

```
C:\>ping 192.168.4.2

Pinging 192.168.4.2 with 32 bytes of data:

Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.

Ping statistics for 192.168.4.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>ping 192.168.2.4

Pinging 192.168.2.4 with 32 bytes of data:

Reply from 192.168.2.4: bytes=32 time<1ms TTL=126
Reply from 192.168.2.4: bytes=32 time<1ms TTL=126
Reply from 192.168.2.4: bytes=32 time<1ms TTL=126
Reply from 192.168.2.4: bytes=32 time=1ms TTL=126

Ping statistics for 192.168.2.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>
```

ping 服务器（unreachable）

ping 其他子网的普通设备

- 192.168.1.5（子网1的普通人）：



7. 凯撒给予的最高权限

只有PC1可以随意地ping其他设备，我们在已有的ACL上进行修改：

- Router1:

```
1  (config)# access-list 103 permit icmp host 192.168.1.2 any
```

- Router2:

```
1  (config)# access-list 101 permit icmp host 192.168.1.2 any
```

- Router3:

```
1  (config)# access-list 101 permit icmp host 192.168.1.2 any
```

同时，我们设置CBAC如下：

- Router1的Fa0/1（连接他自己的子网），由于对于PC1来说，ICMP报文是进入Fa0/1，所以最后我们要用in

```
Router(config)#ip inspect name caesar icmp
Router(config)#int fa0/1
Router(config-if)#ip inspect caesar in
Router(config-if)#
```

- 对其他两个路由器则是out

```
Router(config)#ip inspect name caesar icmp
Router(config)#int fa0/1
Router(config-if)#ip inspect caesar out
Router(config-if)#
```

- 实验结果：

192.168.1.2现在能ping任何地方

```
C:\>ping 192.168.3.4

Pinging 192.168.3.4 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.3.4:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>ping 192.168.3.4

Pinging 192.168.3.4 with 32 bytes of data:

Reply from 192.168.3.4: bytes=32 time=1ms TTL=125
Reply from 192.168.3.4: bytes=32 time<1ms TTL=125
Reply from 192.168.3.4: bytes=32 time=1ms TTL=125
Reply from 192.168.3.4: bytes=32 time<1ms TTL=125

Ping statistics for 192.168.3.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>
```

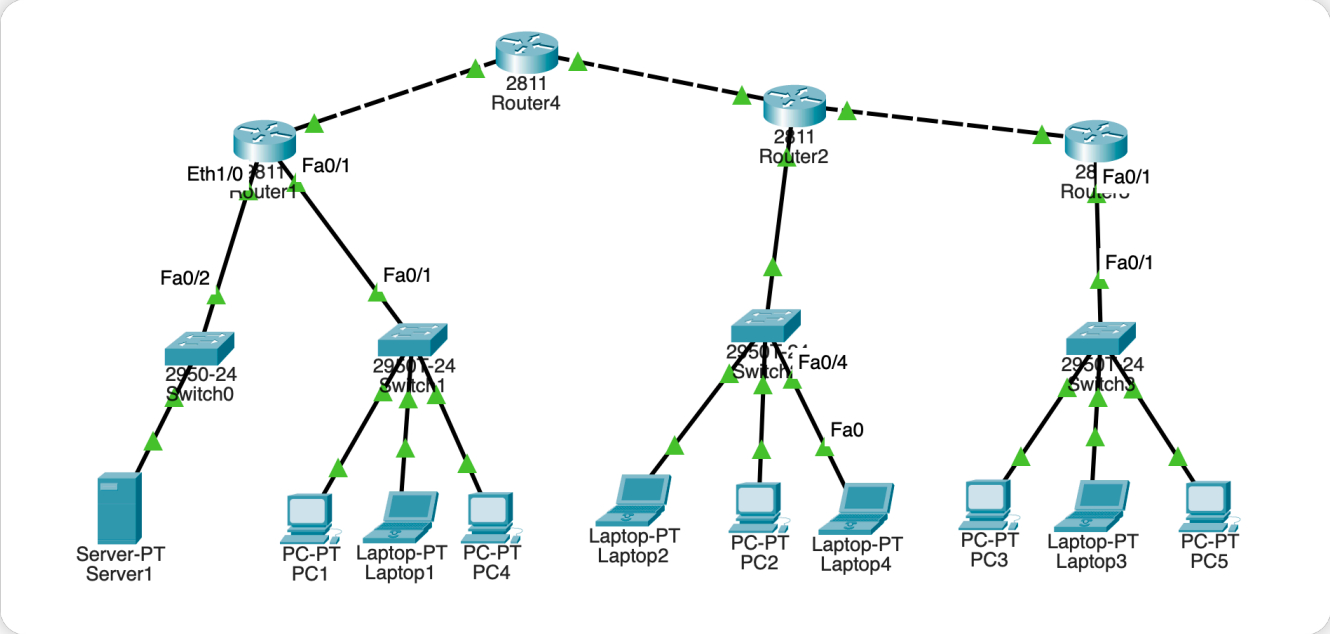
但别人不一定能ping到它

```
C:\>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.3.1: Destination host unreachable.
Reply from 192.168.3.1: Destination host unreachable.
Reply from 192.168.3.1: Destination host unreachable.
Reply from 192.168.3.1: Destination host unreachable.

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>
```

（另外，在这次试验中我发现Ethernet接口是没有CBAC功能的。我不得不换了一个模块（NM-2FE2W））。

8. 新的远征

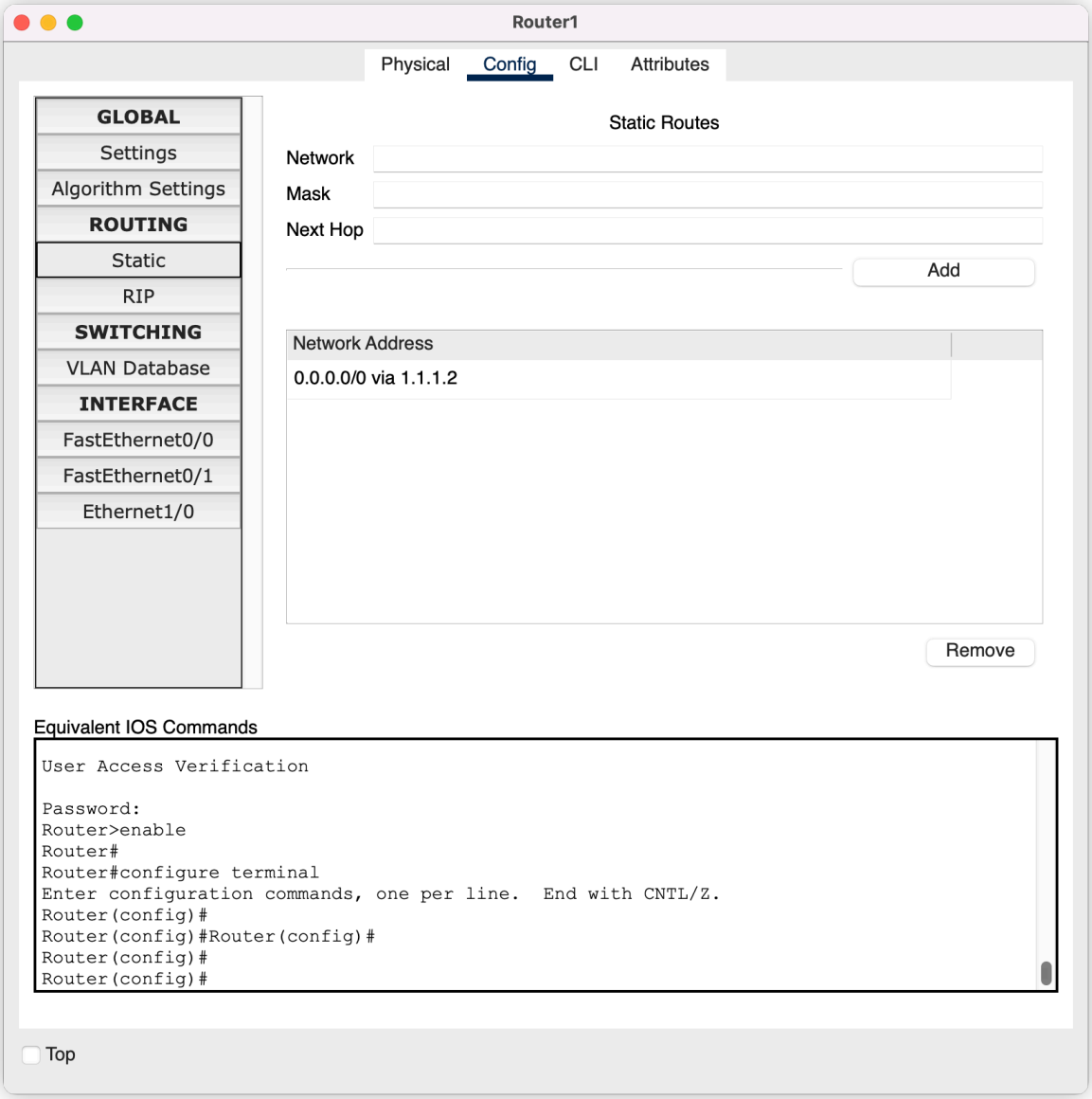
新的网络拓扑图如下（所有之前的ACL都已经被删除）：



对于router1，我们将其连接外网（router4）的端口设置为1.1.1.1，router4连接router1的端口设置为1.1.1.2

对称地，我们将router2连接外网的端口设置为1.1.2.1，router4连接router2的端口设置为1.1.2.2

然后我们给边界路由添加静态路由转发（以router1为例）：



接着我们进行ISAKMP的配置。以Router1为例，这经历了以下五个步骤：

1. 配置ISAKMP

```
Router(config)#cry
Router(config)#crypto is
Router(config)#crypto isakmp poli
Router(config)#crypto isakmp policy 1
Router(config-isakmp)#enc
Router(config-isakmp)#encryption 3des
Router(config-isakmp)#hash md5
Router(config-isakmp)#auth
Router(config-isakmp)#authentication pre
Router(config-isakmp)#authentication pre-share
Router(config-isakmp)#group 5
Router(config-isakmp)#exit
Router(config)#cry
Router(config)#crypto isa
Router(config)#crypto isakmp key 2023fall ad
Router(config)#crypto isakmp key 2023fall address 1.1.2.1
```

2. 设置ACL

（这里偷了个懒，直接保护了192.168.0.0到192.168.0.0的所有流量）

```
Router(config)#access-list 101 permit ip 192.168.0.0 0.0.255.255 192.168.0.0 0.0.255.255
```

3. 设置transform-set

```
Router(config)#crypto ipsec transform-set 2023set esp-3des esp-md5-hmac
```

4. 创建MAP映射表

```
Router(config)#crypto map 2023map 1 ipse
Router(config)#crypto map 2023map 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 peer 1.1.2.1
Router(config-crypto-map)#match add
Router(config-crypto-map)#match address 101
Router(config-crypto-map)#set trans
Router(config-crypto-map)#set transform-set 2023set
```

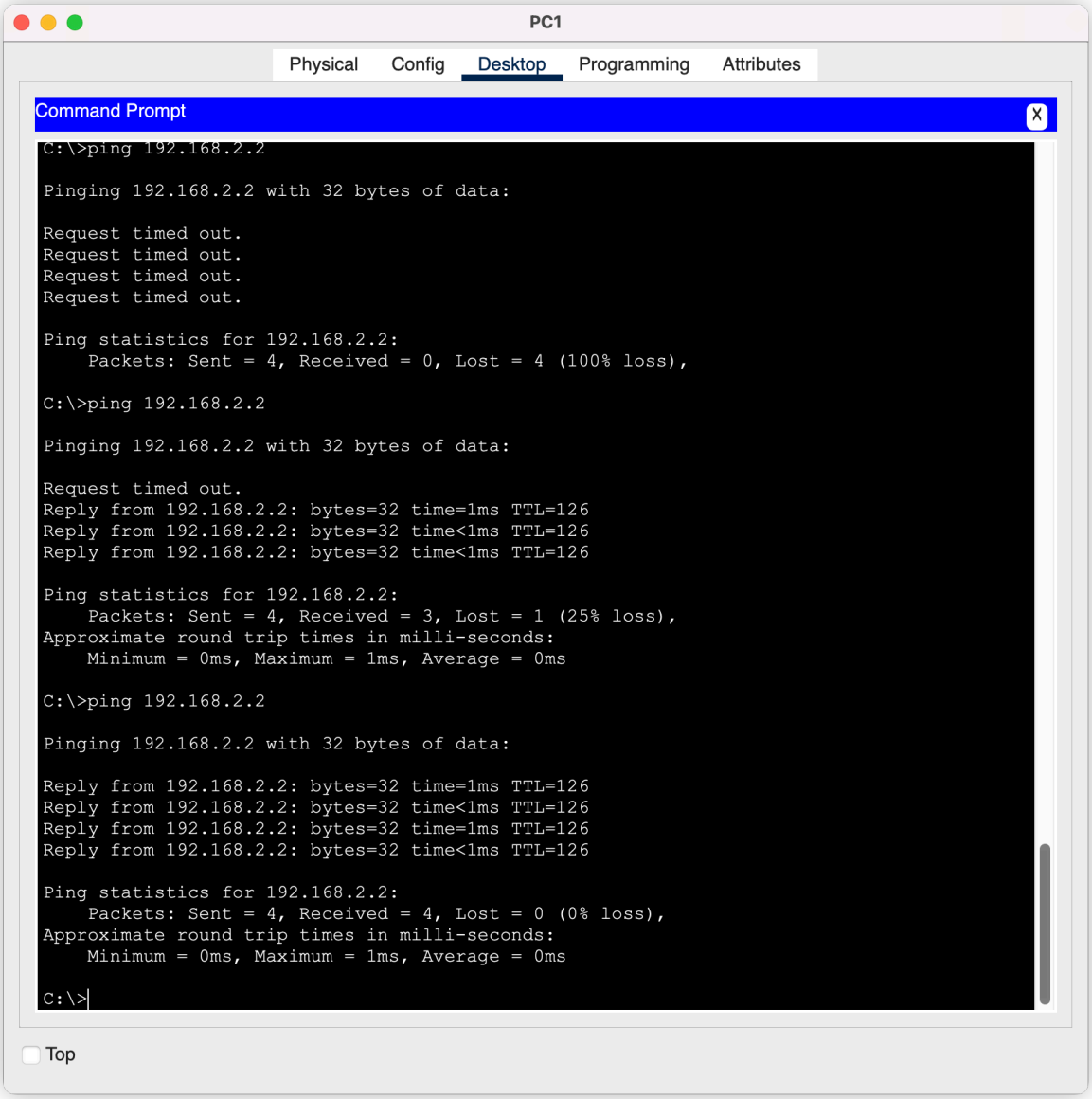
5. 绑定端口

```
Router(config)#int fa0/0
Router(config-if)#cy
Router(config-if)#cycr
Router(config-if)#crypto map 2023map
*Jan  3 07:16:26.785: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
```

可以看出，我们ISAKMP的key为**2023fall**，index为1，transform-set的名称为**2023set**，MAP映射表的名字为**2023map**，index为1

对于router2，用完全对称的方式配置即可，只不过address和peer要改为1.1.1.1

这时从192.168.1.2可以ping通192.168.2.2，而我们没有对router4进行任何设置



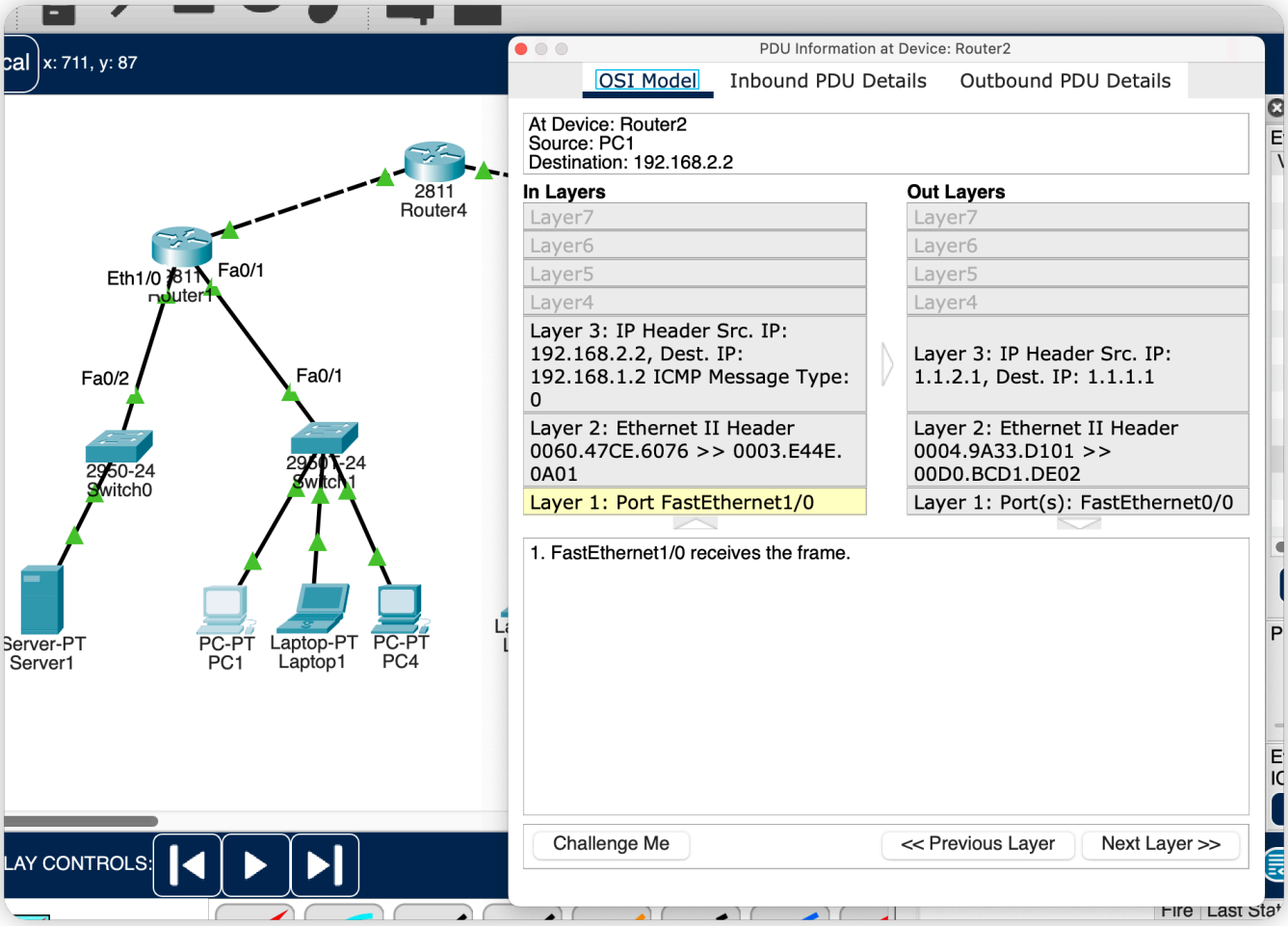
可以看到router1的isakmp状态也发生改变：

```
Router#show cry
Router#show crypto is
Router#show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id slot status
1.1.2.1         1.1.1.1         QM_IDLE           1083    0 ACTIVE


IPv6 Crypto ISAKMP SA


Router#
```

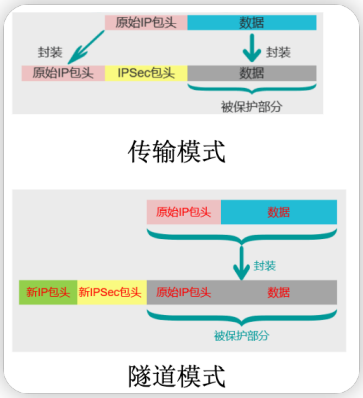
在搬迁之后，使用配置静态路由的方法将无法让各个权力机构正常通信，请简述原因。

- 首先，我们只能改变边界路由器的设置，而不能改变公网路由器的设置，也就是说公网路由器拿到来自（例如）192.168.1.0子网的数据，并不知道它的下一跳是哪里。
- 其次，私网地址段（可能主要为）192.168.0.0-192.168.255.255。若内网的计算机想要与外网通信，其必须将ip经过NAT转换。

通过仿真抓包分析，如上配置的IPSec VPN使用了传输模式还是隧道模式，为什么？



使用了隧道模式。原因如下：



可以看到，传输模式是不改变原始IP包头的。然而这里，经过Router转发以后，报文的IP header中的src和dst都发生了变化，这说明ip包头被改变了。说明只能使用了隧道模式，附加了一个新IP包头上去。

凯撒的馈赠：NAT转换

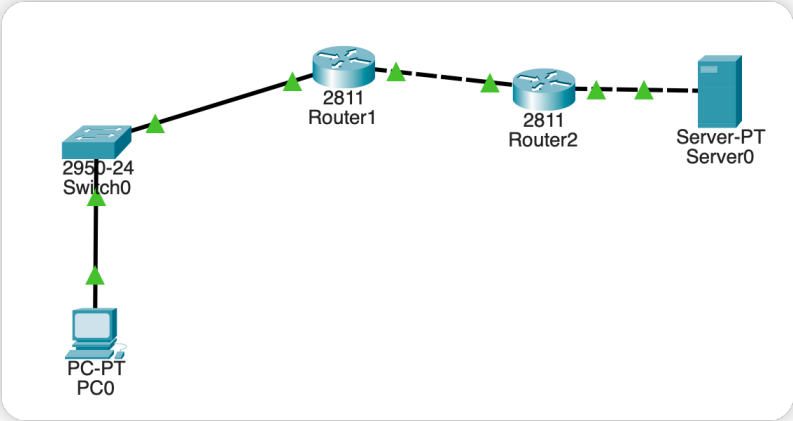
NAT简介

NAT (Network Address Translation，网络地址转换)，允许将**私有的IP地址**（10.0.0.0-10.255.255.255；172.16.0.0-172.31.255.255；192.168.0.0-192.168.255.255）映射到**合法的Internet IP地址（公网IP地址）**。

当**本地内网没有公网ip地址**，或是**需要合并多个一样的内网时**，NAT可以发挥作用。

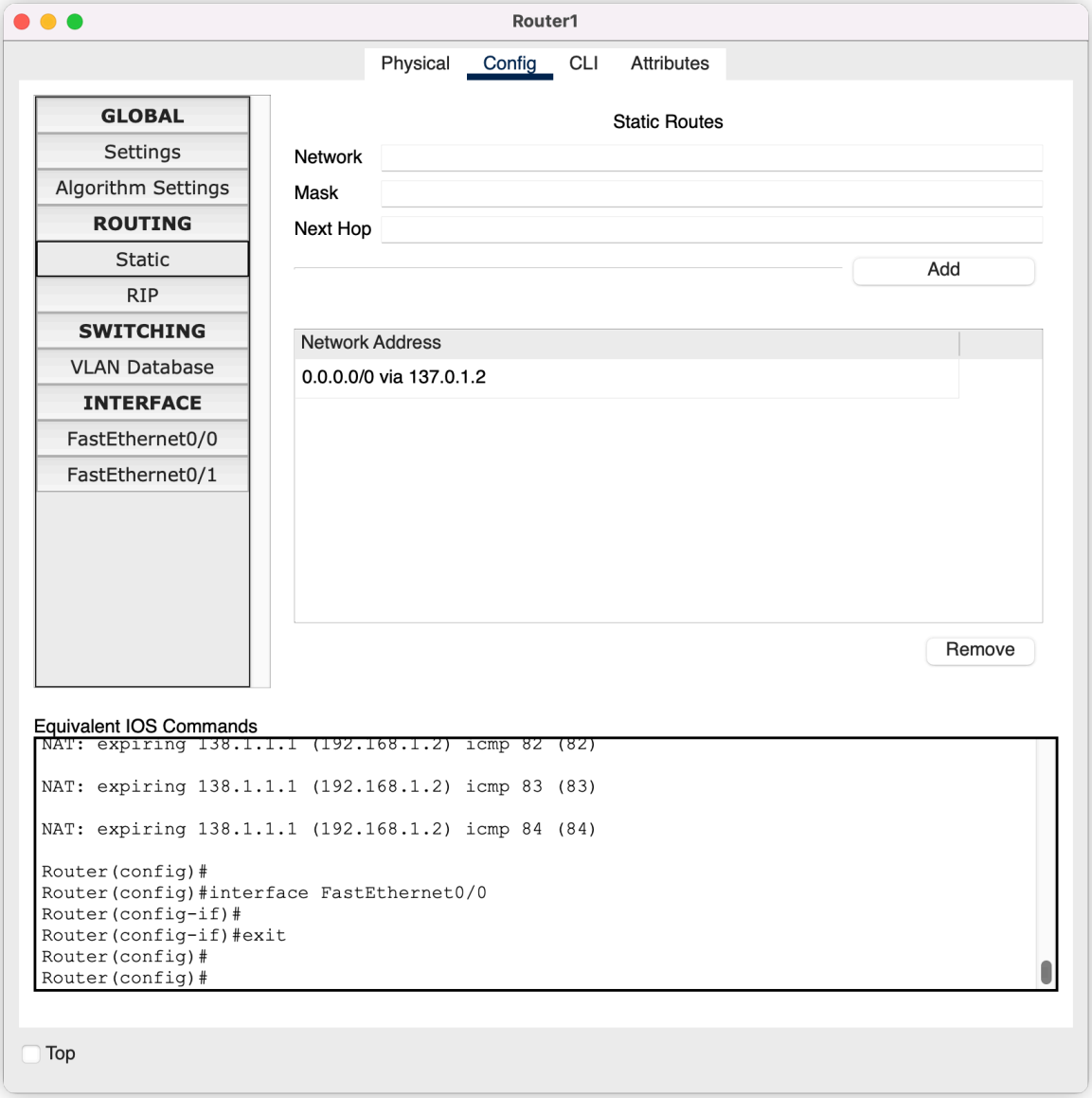
静态NAT

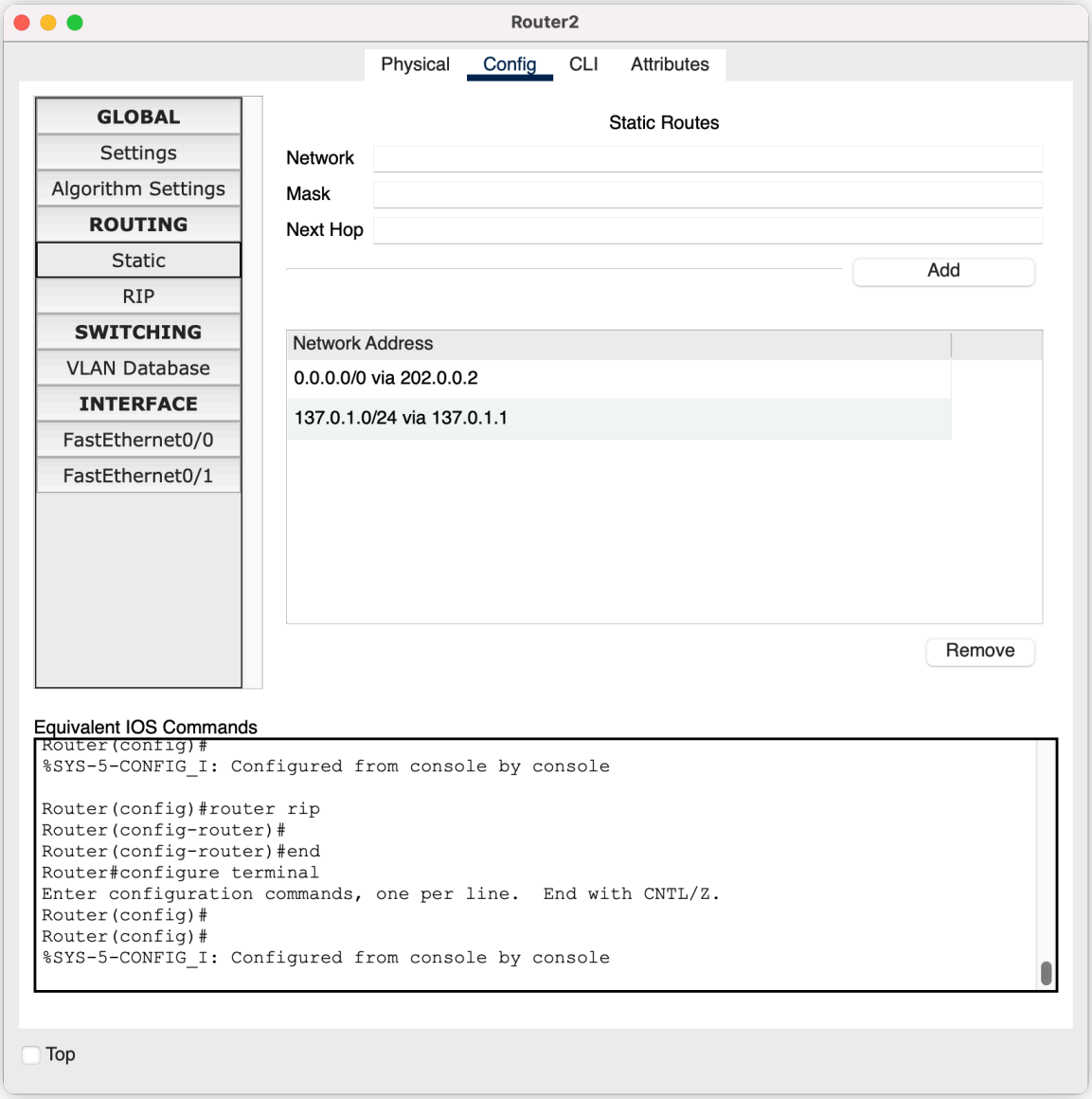
- 每台主机**得到一个真实的IP地址**，一一映射



以这样一个网络拓扑结构为例。假如router1左端口为192.168.1.1，PC0的地址为192.168.1.2。router1右端口为137.0.1.1，router2左端口为137.0.1.2。router2右端口为202.0.1.1，server0的地址为202.0.1.2。其中switch0，PC0为内网设备，Router1位边界路由器，Router2和Server0为公网设备。

我们再为Router1（边界路由器）和Router2（公网路由器）设置静态路由：





这时我们在私网的192.168.1.2，去ping服务器202.0.1.2，是ping不通的。因为公网路由器不知道怎么转发192.168.1.2这个私网字段的地址。

```
C:\>ping 202.0.1.2

Pinging 202.0.1.2 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 202.0.1.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

这时，我们在router1中为192.168.1.2手动指定公网地址：

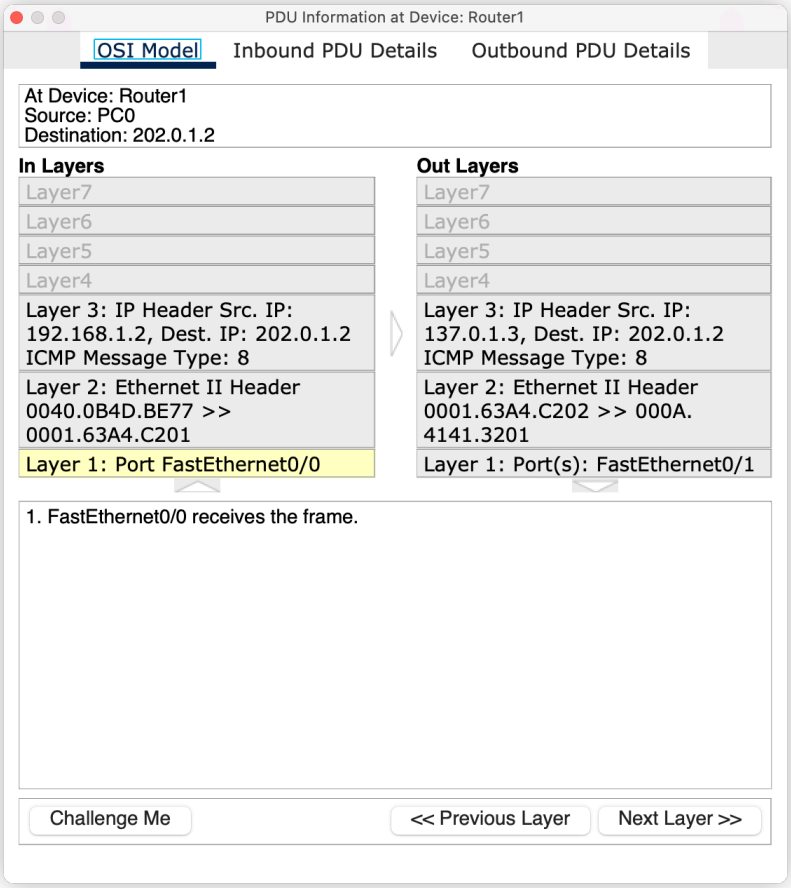
```
// 左端为内部端口
int fa0/0
ip nat inside
exit

// 右端为外部端口
int fa0/1
ip nat outside

ip nat inside source static 192.168.1.2 131.0.1.3
```

```
Router(config)#ip nat inside source static 192.168.1.2 137.0.1.3
Router(config)#ipnat_add_static_cfg: id 8, flag 6

id 8, flags 0, domain 0, lookup 0, from_addr C0A80102,

        from_mask FFFFFFFF, from_port 0, to_addr 89000103, to_port 0

        to_mask FFFFFFFF, proto 0
```

这时，我们再ping服务器：

```
C:\>ping 202.0.1.2

Pinging 202.0.1.2 with 32 bytes of data:

Request timed out.
Reply from 202.0.1.2: bytes=32 time<1ms TTL=126
Reply from 202.0.1.2: bytes=32 time<1ms TTL=126
Reply from 202.0.1.2: bytes=32 time=1ms TTL=126

Ping statistics for 202.0.1.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

可以ping通了！我们观察一下报文：



可以发现IP头的src被改变了。

动态NAT

静态NAT只是简单的一一映射，可能不能很好地解决公网ip不够用的情况。

我们先取消之前的静态NAT配置：

```
Router(config)#no ip nat inside source static 192.168.1.2 137.0.1.3
Router(config)#
ipnat_remove_static_cfg: id 8, flag A
```

然后配置动态NAT：

1. 设置ACL

```
Router(config)#access-list 101 permit ip 192.168.0.0 0.0.255.255 any
Router(config)#
```

2. 定义公网地址池

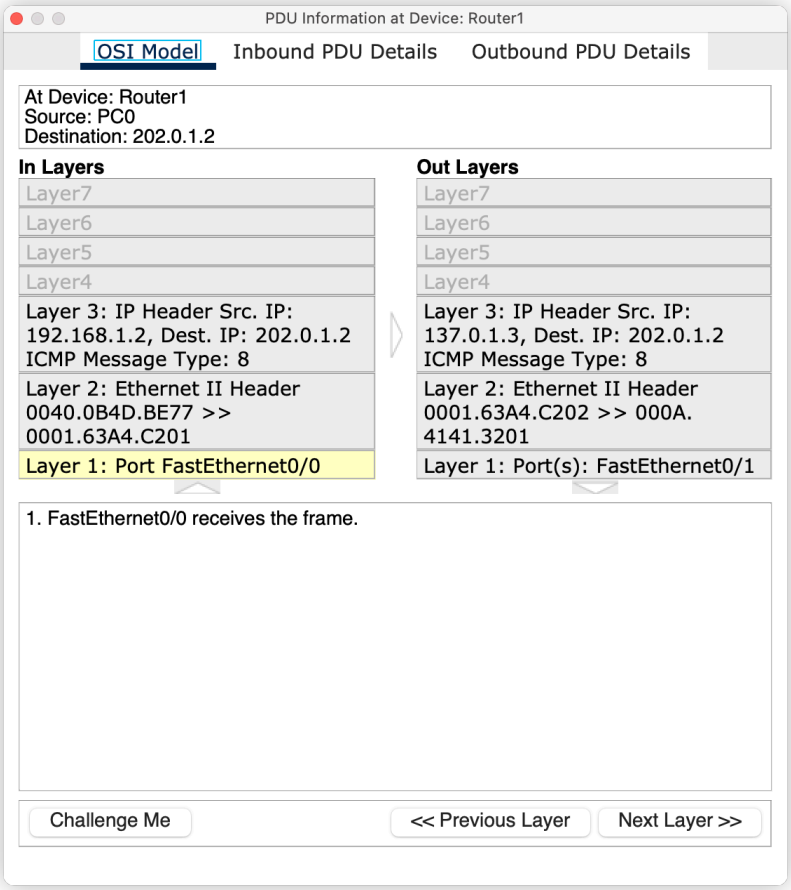
```
Router(config)#ip nat pool junli 137.0.1.3 137.0.1.255 netmask 255.255.255.0
```

3. 绑定

```
Router(config)#ip nat inside source list 101 pool junli
Router(config)#ipnat_add_dynamic_cfg: id 1, flag 5, range 0

poolstart 137.0.1.3 poolend 137.0.1.255

id 1, flags 0, domain 0, lookup 0, aclnum 101 ,

        aclname 101 , mapname idb 0
```

```
C:\>ping 202.0.1.2

Pinging 202.0.1.2 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Reply from 202.0.1.2: bytes=32 time<1ms TTL=126

Ping statistics for 202.0.1.2:
    Packets: Sent = 4, Received = 1, Lost = 3 (75% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

可以ping通了，看一下报文：



可以看到分配到的公网地址为137.0.1.3