به نام خدا

آزمایشگاه شبکه‌های کامپیوتری

گزارش آزمایش ششم

استاد:

دکتر بردیا صفائی

نویسندگان:

محمدهومان کشوری

هیربد بهنام

علی نظری

شماره دانشجویی:

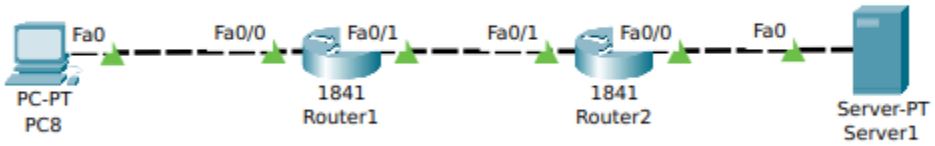
99105667

99171333

99102401

قسمت Static

ابتدا برای دستگرمی شروع این قسمت یک زیر شبکه کوچک شامل دو راوتر، یک کامپیوتر و یک سرور درست می‌کنیم.



برای Router1 :

```
interface FastEthernet0/0
 ip address 192.168.10.1 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 192.168.20.1 255.255.255.0
 duplex auto
 speed auto
!
```

برای Router2 :

```
interface FastEthernet0/0
 ip address 192.168.30.1 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 192.168.20.2 255.255.255.0
 duplex auto
 speed auto
```

حال قبل از کانفیگ کردن NAT مهم است که **مسیریابی** در راوترها را درست کنیم.

برای Router2 :

```
R2(config-if)#ip route 0.0.0.0 0.0.0.0 192.168.20.1
```

برای Router1 :

```
C    192.168.10.0/24 is directly connected, FastEthernet0/0
C    192.168.20.0/24 is directly connected, FastEthernet0/1
S*   0.0.0.0/0 [1/0] via 192.168.20.2
```

حال در router1 تنظیمات NAT را انجام می‌دهیم.

```
R1(config)#inter f 0/0
R1(config-if)#ip nat in
R1(config-if)#ip nat inside
R1(config-if)#inter f 0/1
R1(config-if)#ip nat out
R1(config-if)#ip nat outside
```

```
R1(config)#ip nat inside source static 192.168.10.10 100.100.100.100
```

حال می‌بینیم که آیا NAT وظیفه خود را به درستی انجام می‌دهد یا خیر :

```
R1#debug ip nat
IP NAT debugging is on
```

در صورتی که از کامپیوتر درون NAT ای پی سرور را پینگ کنیم :

```
C:\>ping 192.168.30.10

Pinging 192.168.30.10 with 32 bytes of data:

Reply from 192.168.30.10: bytes=32 time<1ms TTL=126
Reply from 192.168.30.10: bytes=32 time<1ms TTL=126
Reply from 192.168.30.10: bytes=32 time<1ms TTL=126
Reply from 192.168.30.10: bytes=32 time<1ms TTL=126

Ping statistics for 192.168.30.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
NAT: s=192.168.10.10->100.100.100.100, d=192.168.30.10 [9]

NAT*: s=192.168.30.10, d=100.100.100.100->192.168.10.10 [4]

NAT: s=192.168.10.10->100.100.100.100, d=192.168.30.10 [10]

NAT*: s=192.168.30.10, d=100.100.100.100->192.168.10.10 [5]

NAT: s=192.168.10.10->100.100.100.100, d=192.168.30.10 [11]

NAT*: s=192.168.30.10, d=100.100.100.100->192.168.10.10 [6]

NAT: s=192.168.10.10->100.100.100.100, d=192.168.30.10 [12]

NAT*: s=192.168.30.10, d=100.100.100.100->192.168.10.10 [7]
```

که این نشان می‌دهد، در فضای بیرون از NAT، کامپیوتر ما به صورت 100.100.100.100 شناخته می‌شود و NAT ما درست عمل کرده است.

از سرور هم یک پینگ برای 100.100.100.100 می‌گیریم.

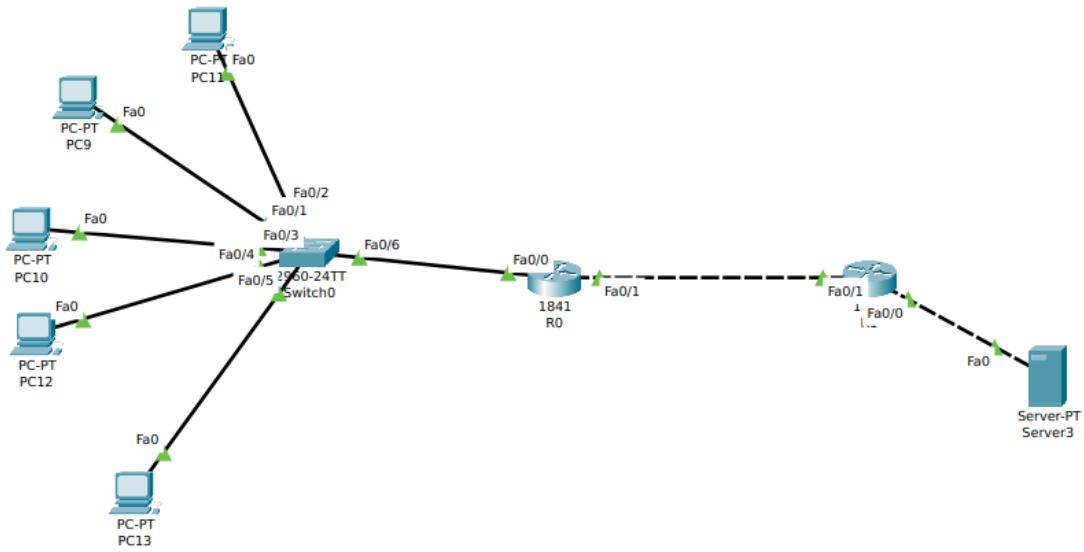
```
C:\>ping 100.100.100.100

Pinging 100.100.100.100 with 32 bytes of data:

Reply from 100.100.100.100: bytes=32 time<1ms TTL=126
Reply from 100.100.100.100: bytes=32 time=1ms TTL=126
Reply from 100.100.100.100: bytes=32 time<1ms TTL=126
Reply from 100.100.100.100: bytes=32 time<1ms TTL=126

Ping statistics for 100.100.100.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

حال شبکه را کمی پیچیده‌تر می‌کنیم.



و باز هم پس از پینگ کردن 100.100.100.100 توسط مثلا PC10 :

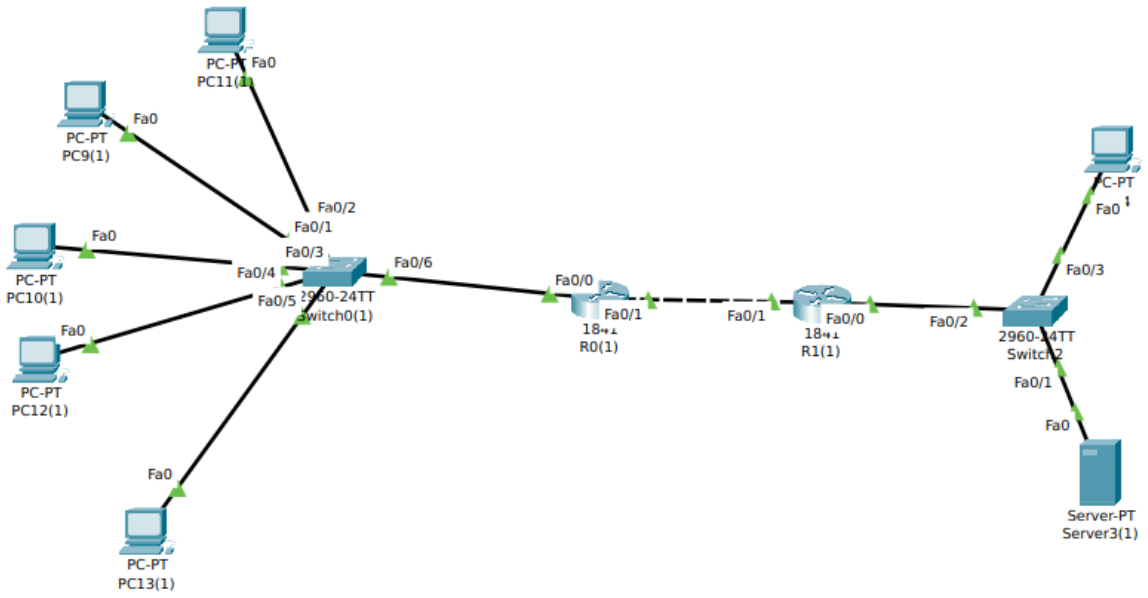
```
C:\>ping 100.100.100.100

Pinging 100.100.100.100 with 32 bytes of data:

Reply from 100.100.100.100: bytes=32 time<1ms TTL=126
Reply from 100.100.100.100: bytes=32 time<1ms TTL=126
Reply from 100.100.100.100: bytes=32 time<1ms TTL=126
Reply from 100.100.100.100: bytes=32 time<1ms TTL=126

Ping statistics for 100.100.100.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

قسمت Dynamic



حال تنظیمات را عوض می‌کنیم و به صورت dynamic nat استفاده می‌کنیم.

باری این کار ابتدا باید static nat را غیر فعال کنیم.

```
R1(config)#no ip nat inside source static 192.168.10.10 100.100.100.100
```

سپس ورودی خروجی نت و نیز access-list را مشخص می‌کنیم.

```
R1(config)#inter f 0/0
R1(config-if)#ip nat inside
R1(config-if)#inter f 0/1
R1(config-if)#ip nat outside
R1(config-if)#acc
R1(config-if)#access-list 10 permit 192.168.10.0 0.0.0.255
```

در اخر باید گروه‌های خارجی و داخلی را مشخص کنیم که از pool برای این کار استفاده می‌کنیم.

```
R1(config)#ip nat pool CCNP 200.200.200.1 200.200.200.200 netmask 255.255.255.0
R1(config)#ip nat inside so
R1(config)#ip nat inside source li
R1(config)#ip nat inside source list 10 pool CCNP
R1(config)#ip nat inside source list 10 pool CCNP overload
```

حال تست می‌کنیم و همزمان با دو دستگاه پینگ می‌گیریم.

```
C:\>ping 192.168.30.20

Pinging 192.168.30.20 with 32 bytes of data:

Reply from 192.168.30.20: bytes=32 time=25ms TTL=126
Reply from 192.168.30.20: bytes=32 time<1ms TTL=126
Reply from 192.168.30.20: bytes=32 time<1ms TTL=126
Reply from 192.168.30.20: bytes=32 time=1ms TTL=126

Ping statistics for 192.168.30.20:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 25ms, Average = 6ms

C:\>
```

```
C:\>ping 192.168.30.20

Pinging 192.168.30.20 with 32 bytes of data:

Reply from 192.168.30.20: bytes=32 time<1ms TTL=126
Reply from 192.168.30.20: bytes=32 time<1ms TTL=126
Reply from 192.168.30.20: bytes=32 time<1ms TTL=126
Reply from 192.168.30.20: bytes=32 time<1ms TTL=126

Ping statistics for 192.168.30.20:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

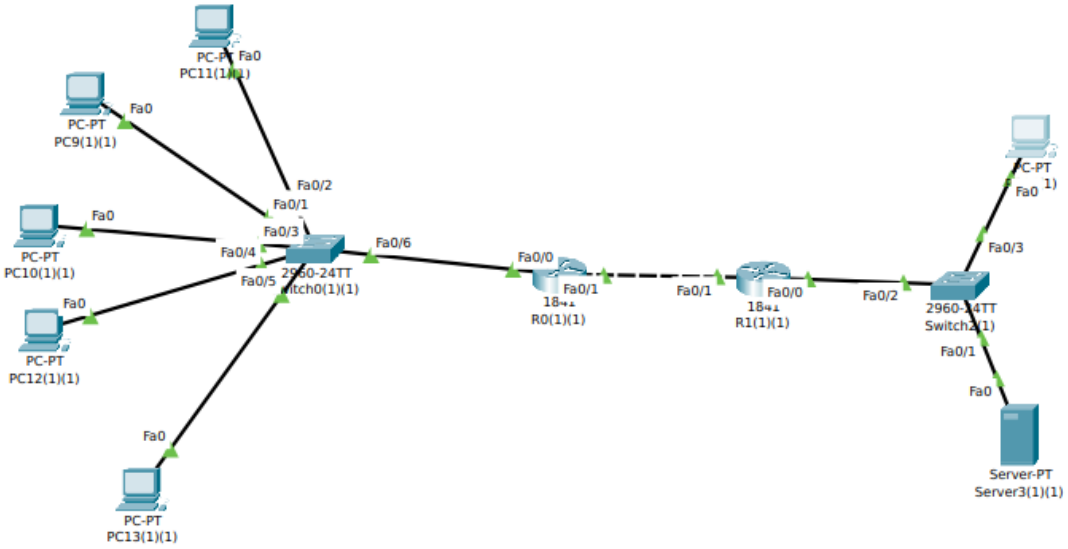
حال در Router1، اتفاقات زیر را مشاهده می‌کنیم.

```
NAT: s=192.168.10.10->200.200.200.1, d=192.168.30.20 [5]
NAT*: s=192.168.30.20, d=200.200.200.1->192.168.10.10 [4]
NAT: s=192.168.10.20->200.200.200.1, d=192.168.30.20 [13]
NAT*: s=192.168.30.20, d=200.200.200.1->192.168.10.20 [5]
NAT: s=192.168.10.10->200.200.200.1, d=192.168.30.20 [6]
NAT*: s=192.168.30.20, d=200.200.200.1->192.168.10.10 [6]
NAT: s=192.168.10.20->200.200.200.1, d=192.168.30.20 [14]
NAT*: s=192.168.30.20, d=200.200.200.1->192.168.10.20 [7]
NAT: s=192.168.10.10->200.200.200.1, d=192.168.30.20 [7]
NAT*: s=192.168.30.20, d=200.200.200.1->192.168.10.10 [8]
NAT: s=192.168.10.20->200.200.200.1, d=192.168.30.20 [15]
NAT*: s=192.168.30.20, d=200.200.200.1->192.168.10.20 [9]
NAT: s=192.168.10.10->200.200.200.1, d=192.168.30.20 [8]
NAT*: s=192.168.30.20, d=200.200.200.1->192.168.10.10 [10]
NAT: s=192.168.10.20->200.200.200.1, d=192.168.30.20 [16]
NAT*: s=192.168.30.20, d=200.200.200.1->192.168.10.20 [11]
```

می‌بینیم که در NAT ما دو ای پی پشت نت به یک ای پی بیرون nat مپ می‌شوند.
این کار به علت overload است و NAT در هر درخواست خارجی منتظر می‌ماند تا جواب را به ip داخلی درست بازگرداند برای همین به ترتیب درخواست‌ها به **صف منتظر می‌مانند**.
حال می‌توان این فرایند را برای 5 کامپیوتر پشت NAT با ۴ آی پی نیز به همین صورت انجام داد به این صورت که به جای 200.200.200.1 تا 200.200.200.200 از 200.200.200.1 تا 200.200.200.4 استفاده می‌کنیم.

قسمت PAT

از توپولوژی قسمت قبل استفاده می‌کنیم با این تفاوت که این دفعه در راوتر ۱ یک PAT تنظیم می‌کنیم.




```
R1(config)#inter f 0/0
R1(config-if)#ip nat inside
R1(config-if)#inter f 0/1
R1(config-if)#ip nat outside
R1(config-if)#exit
R1(config)#
R1(config)#
R1(config)#
R1(config)#access-list 10 permit 192.168.10.0 0.0.0.255
R1(config)#
R1(config)#
R1(config)#
R1(config)#ip nat inside source list 10 interface f 0/1 overload
```

حال مشاهده می‌کنیم که در این قسمت دیگر به لیست خارجی کاری نداریم و صرفا برای ما interface ورودی آنها مهم است.

حال :

```
Pinging 192.168.30.10 with 32 bytes of data:

Request timed out.
Request timed out.
Reply from 192.168.30.10: bytes=32 time<1ms TTL=126
Reply from 192.168.30.10: bytes=32 time<1ms TTL=126

Ping statistics for 192.168.30.10:
    Packets: Sent = 4, Received = 2, Lost = 2 (50% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

بعد از debug ip nat :

```
R1#debug ip nat
IP NAT debugging is on
R1#
NAT: s=192.168.10.20->192.168.20.1, d=192.168.30.10 [1]

NAT: s=192.168.10.20->192.168.20.1, d=192.168.30.10 [2]

NAT: s=192.168.10.20->192.168.20.1, d=192.168.30.10 [3]

NAT*: s=192.168.30.10, d=192.168.20.1->192.168.10.20 [1]

NAT: s=192.168.10.20->192.168.20.1, d=192.168.30.10 [4]

NAT*: s=192.168.30.10, d=192.168.20.1->192.168.10.20 [2]
```

مشاهده می‌کنیم که به عنوان خروجی به صورت خودکار یک آی پی مشخص شده است.

سوالات:

۱.

```
R2(config)#ip nat ?
  inside   Inside address translation
  outside  Outside address translation
  pool     Define pool of addresses
R2(config)#ip nat
```

در صورت اجرای inside عملا اینترفیس به صورت NAT داخلی(آدرس داخلی) در می‌آید.
در صورت اجرای outside عملا اینترفیس به صورت NAT خارجی(آدرس خارجی) در می‌آید.
در صورت اجرای pool یک گروه خارجی درست می‌شود که یک ای پی رنج خاص دارد.

۲. به صورت کلی access listها برای کنترل کردن پکت‌های خروجی و ورودی به یک router و ip range استفاده می‌شود. دو نوع مدل permit و deny دارد که permit به packetها اجازه‌ی عبور می‌دهد و deny اجازه‌ی عبور نمی‌دهد. به عنوان مثال در این

آزمایش ما از deny استفاده کردیم. برای فیلتر کردن بسته‌های پورت ۸۰ می‌توانیم از دستور زیر استفاده کنیم: (منبع)

```
access-list 100 deny tcp any any eq 80
```

۳. اولا که بسیار تنظیمات سرراست‌تر و ساده‌تری نسبت به حالت dynamic , مخصوصا static داشت از این جهت که خود، ای پی خروجی را تنظیم می‌کرد و نیز پورت‌ها را خود مشخص می‌کرد.

۴. مشخص می‌کند عملا راوتر در پورت خروجی انتظار درخواست به چه ادرسی را داشته باشد (مثلا فردی که از خارج شبکه داخلی برای یک کامپیوتر داخلی پیام می‌فرستد) و داخلی نیز مشخص می‌کند که اینترفیس مربوطه درخواست‌ها را از کدام ای پی گرفته و آنرا ترجمه کند.

برای تعویض قسمت static دستور زیر را می‌زنیم :

```
ip nat inside source static 10.0.0.12 240.230.220.210
```

برای تعویض قسمت Dynamic :

```
Ip nat pool CCNP 220.220.220.2 220.220.220.6 netmask 255.255.255.0
```