Redes de Computadores (RCOMP) – 2017/2018 Laboratory Class Script – PL20	
• Network Address Translation (NAT). • Static and dynamic NAT in Cisco IOS. • Network Address and Port Translation (NAPT).	• Cisco Packet Tracer

1. Network Address Translation

NAT is a technique by which packets source and destination addresses are automatically changed to achieve several different objectives, most of them related to the use of private addresses. NAT is usually implemented in intermediate nodes like routers and it must be invisible to end-nodes.

Changing packets source addresses is called SNAT (Source Network Address Translation), changing packets destination addresses is called DNAT (Destination Network Address Translation). NAT is invisible to end nodes because to a SNAT application in one direction matches a DNAT application in the opposite direction.

Two things are required to apply NAT:

- Addresses equivalency – This defines which address in one side is equivalent to another address on the other side, meaning one will be automatically replaced by the other. NAT addresses equivalencies are usually presented as a **NAT table**.
- We must define the translation direction, to do this, sides need to be identified. We could call them side A and side B, but in Cisco routers, they are called **inside** and **outside**, though this names take no special meaning. Once sides are identified, then we can settle NAT application in either two ways:
    - We can instruct the router to apply SNAT when packets travel from inside to outside, this automatically implies DNAT will be applied when packets travel from outside to inside.
    - We can instruct the router to apply DNAT when packets travel from inside to outside, this automatically implies SNAT will be applied when packets travel from outside to inside.

1.1. Cisco IOS NAT commands

One thing we must do is set sides. In a router sides are related to network interfaces, so we must first enter the interface configuration level and then use commands **ip nat inside** or **ip nat outside** to settle to which side does the interface belongs. NAT is applied only to traffic being transferred from an inside interface and an outside interface and vice versa. **Never between same side interfaces**.

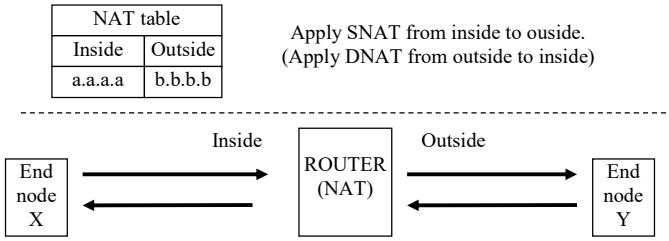
Once sides are settled, we can enforce NAT by using (in general configuration level) one of the following commands:

ip nat inside source ...	Apply SNAT to packets traveling from inside to outside. (Implies DNAT will be applied to packets traveling from outside to inside)
ip nat outside source ...	Apply SNAT to packets traveling from outside to inside (Implies DNAT will be applied to packets traveling from inside to outside)

1.2. Static NAT

Static NAT means the NAT table is static and manually defined by the administrator, there is a static equivalency between the inside address and the outside address. Both the inside and outside addresses are permanently allocated for this purpose.

Imagine the following setup:



We have a NAT table and sides identified, we also defined the SNAT application direction, and consequently DNAT in the opposite direction.

- Now imagine end-node X has address a.a.a.a and sends a packet to end-node Y, when the packet travels through the router (inside to outside) the packet's source address is translated from a.a.a.a to b.b.b.b, thus, end-node Y is going to receive a packet with source address b.b.b.b.

- Now imaging end node Y sends a packet to address b.b.b.b (may be a reply to the previous packet), when the packet travels through the router (outside to inside) the packet destination address is translated from b.b.b.b to a.a.a.a, thus, it's going to be delivered to node X.

From this setup we can see some NAT features. In this configuration, inside nodes have their source address hidden from outside nodes. Also, indirect access to inside nodes can be provided if outside nodes send requests to the outside address.

These features turn to be very useful if inside is a private network and outside a public network (internet). Inside nodes using private addresses will be able to access internet services because, due to SNAT, when requests reach the internet servers they will be coming from a public address (b.b.b.b on the previous scenario). Server replies, of course, are sent to the public address, but then, DNAT ensures they will reach the real source node with the inside private address.

We can also have servers on the inside private network, then outside internet clients must send requests to the public address (b.b.b.b on the above scenario), DNAT will redirect them to the inside private server address and the server reply will have SNAT applied to hide the private source address.

We can deploy static NAT in a Cisco router by using the command:

```
ip nat inside source static IP-ADDRESS-1 IP-ADDRESS-2
```

This means packets with source address IP-ADDRESS-1 traveling from an inside interface to an outside interface will have the source address changed to IP-ADDRESS-2, also, packets with destination address IP-ADDRESS-2 traveling from an outside interface to an inside interface will have the destination address changed to IP-ADDRESS-1.

The command **ip nat outside source static IP-ADDRESS-2 IP-ADDRESS-1** is therefore equivalent to the previous command.

If there are two different commands to do the same thing, you may ask why they are required. Remember one interface may be inside or outside, not both. We may want to apply SNAT to some traffic in one direction and SNAT to other traffic in the opposite direction, then the two different commands will be required:

```
ip nat inside source static IP-ADDRESS-1 IP-ADDRESS-2
ip nat outside source static IP-ADDRESS-3 IP-ADDRESS-4
```

1.3. Dynamic NAT

Dynamic NAT means letting the router automatically create the NAT table entries as requests come. However, because we don't want to expose private nodes to external access, dynamic NAT entries store information about protocols and remote nodes being accessed:

Dynamic NAT table		
Inside	**Outside**	**Remote**
192.168.10.5 – ICMP:2234	190.10.7.8 – ICMP:10226	120.1.72.235 – ICMP:10226
192.168.10.200 – UDP:2288	190.10.7.9 – UDP:45611	182.67.20.1 – UDP:34
172.17.0.235 – TCP:45002	190.10.7.6 – TCP:25516	120.50.7.3 – TCP:80

Picking, for instance, the **second line**, it exists because inside node 192.168.10.200 has sent a UDP packet to port number 34 of node address 182.67.20.1. The line will allow that node to reply back (to 190.10.7.9), nevertheless, only a UDP packet from node 182.67.20.1, port 34 sent to address 190.10.7.9 port 45611 will be allowed. This avoids exposing inside node 192.168.10.200 to other external traffic, something static NAT could not prevent.

Dynamic NAT configuration in Cisco routers requires some additional items:

- **An address pool**. This is a set of outside IP addresses that are made available for dynamic NAT management and are to be assign to inside nodes as needed.
- **An ACL** to match (permit) packets to which we want NAT applied.

Defining an address pool:

```
ip nat pool POOL-NAME IP-ADDRESS-1 IP-ADDRESS-2 netmask NETWORK-MASK
```

This defines a pool named POOL-NAME with addresses starting at IP-ADDRESS-1 up to IP-ADDRESS-2 and the given network mask.

Dynamic NAT can then be deployed by:

```
ip nat inside source list ACL-IDENTIFIER pool POOL-NAME
```

Out of the configuration level, we can see the current NAT translations with the following command:

```
Router#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
icmp 12.0.0.101:2      192.168.0.3:2      170.0.0.3:2        170.0.0.3:2
---  12.0.0.5          192.168.0.4        ---                ---
tcp 12.0.0.100:1025    192.168.0.2:1025   170.0.0.2:80       170.0.0.2:80

Router#
```

In the sample output, the second line is from static NAT, others are from dynamic NAT. You may be puzzled why are there four columns of addresses.

Local means addresses to be used inside, **Global** refers to addresses to be used outside. If local is the same as global, then there is no translation on that side.

For instance, the first line in the above sample has one **Inside global** and a different **Inside local**, this means there is a translation for the inside address, however, **Outside local** and **Outside global** are the same, this means there is no translation for the outside address.

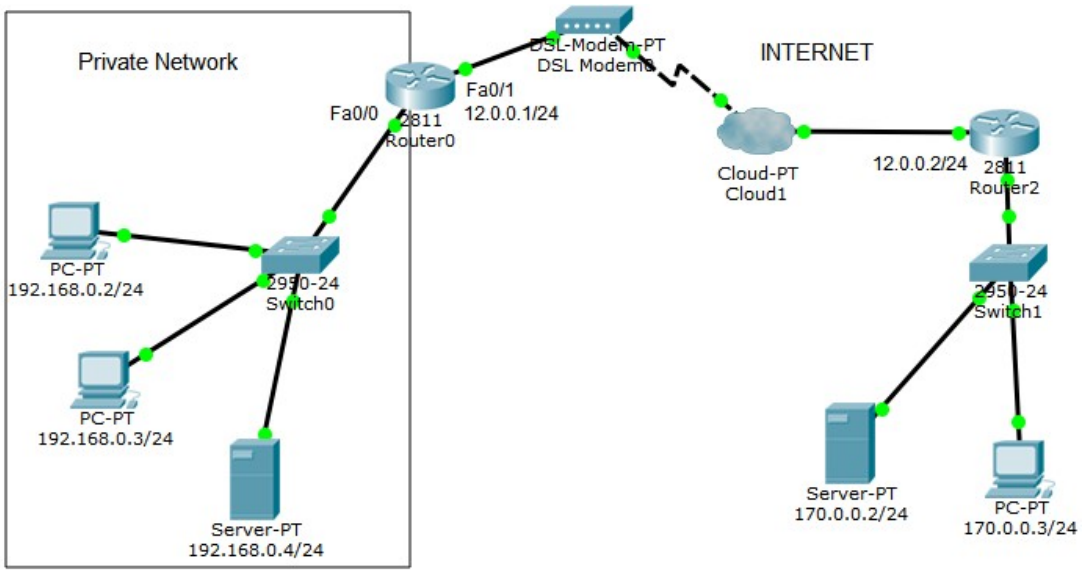
Yet, we can have translation for both inside addresses and outside addresses at the same time:

```
Router#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
icmp 12.0.0.101:1      192.168.0.3:1      170.0.0.2:1        170.0.0.2:1
icmp 12.0.0.101:2      192.168.0.3:2      170.0.0.2:2        170.0.0.2:2
---  12.0.0.5          192.168.0.4        ---                ---
tcp 12.0.0.100:1025    192.168.0.2:1025   170.0.0.2:80       170.0.0.2:80
---  ---               ---                192.168.0.5        12.0.0.2

Router#
```

2. Practical exercise

Create the following diagram on Packet Tracer.
(This diagram is available for download with already settled IP addresses and routing information)



The left side network is private, therefore, is not known to the internet, namely Router 2.

a) If not done already, setup IP addresses and routing.

Default gateway for Router 0 is 12.0.0.2, however, Router 2 will not have a default gateway, it knowns Router 0 (12.0.0.1) but it's not aware there is a network behind it, because it's a private network.

Please, before carrying on, save this configuration to a separate file for later use.

b) Check IPv4 connectivity with ICMP echo requests.

The left side network fails to communicate with right side network and vice versa. This is because the left side network is unknown and, therefore, unreachable. In simulation mode we can see requests from left side (private) are reaching the right side (public), however, replies are not reaching the private network.

c) Apply static NAT on Router 0 to allow access from the internet to server 192.168.0.4

This is what static NAT is good for. First we must set NAT sides:

```
(config)#interface Fa0/0
(config-if)#ip nat inside
(config)#interface Fa0/1
(config-if)#ip nat outside
```

Now we can set static NAT:

```
(config)#ip nat inside source static 192.168.0.4 12.0.0.10
```

d) Check if it works.

Go to PC 170.0.0.3 and open the Web Browser, to access server 192.168.0.4 we must now enter URL http://12.0.0.10

This static NAT configuration has another effect, now server 192.168.0.4 can access to the internet. Check that by sending ICMP echo requests form it to internet nodes.

e) Apply dynamic NAT on Router 0 to allow private nodes access to the internet.

We need an address pool with outside addresses:

```
(config)#ip nat pool MYPOOL 12.0.0.100 12.0.0.150 netmask 255.255.255.0
```

We also need an ACL to match traffic to be translated:

```
(config)#no access-list 10
(config)#access-list 10 permit 192.168.0.0 0.0.0.255
```

Now we can settle dynamic NAT:

```
(config)#ip nat inside source list 10 pool MYPOOL
```

f) See dynamic NAT working

Use the Inspect tool to display Router 0 NAT table. Keep it visible.

Now, send ICMP echo requests from PCs 192.168.0.2 and 192.168.0.3 to internet nodes.

Open the Web Browsers in PCs 192.168.0.2 and 192.168.0.3 and access to URL http://170.0.0.2

3. Network Address and Port Translation

For protocols that use port numbers, NAPT can be used to handle port numbers and not only IP addresses. It work the same way as NAT.

For static NAT commands are:

ip nat inside source static udp IP-ADDRESS-1 PORT-1 IP-ADDRESS-2 PORT-2
This will apply SNAT to UDP packets coming from address IP-ADDRESS-1 with source port number PORT-1, traveling from inside to outside, changing the packet source address to IP-ADDRESS-2 and changing the source port number to PORT-2. Likewise it will apply DNAT to UDP packets sent to address IP-ADDRESS-2 with destination port number PORT-2, traveling from outside to inside, changing the packet destination address to IP-ADDRESS-1 and the destination port to PORT-1-

For the TCP protocols it's exactly the same:

ip nat inside source static tcp IP-ADDRESS-1 PORT-1 IP-ADDRESS-2 PORT-2

4. Dynamic NAT with overload

Overload means using the same inside global address for several different inside local addresses, this is possible because port numbers are used to distinguish traffic belonging to different inside local addresses.

If only one inside global address is to be used and it's the address already assigned to the outside interface, then we may skip the address pool definition and use the command:

```
ip nat inside source list ACL-IDENTIFIER interface INTERFACE-NAME overload
```

This will apply SNAT to packets traveling from inside to outside that match access list ACL-IDENTIFIER, changing the packet source address to the IP address of interface INTERFACE-NAME. Possibly source port (for UDP or TCP) or message ID (for ICMP) will also be changed so that later reply traffic can have DNAT applied correctly.

The following command output show a NAT table in this overload scenario, we can see there is only one inside global address being overloaded with several inside local addresses. If we pay a close attention to the last two lines we can see dynamic NAT was forced to change inside global source port numbers.

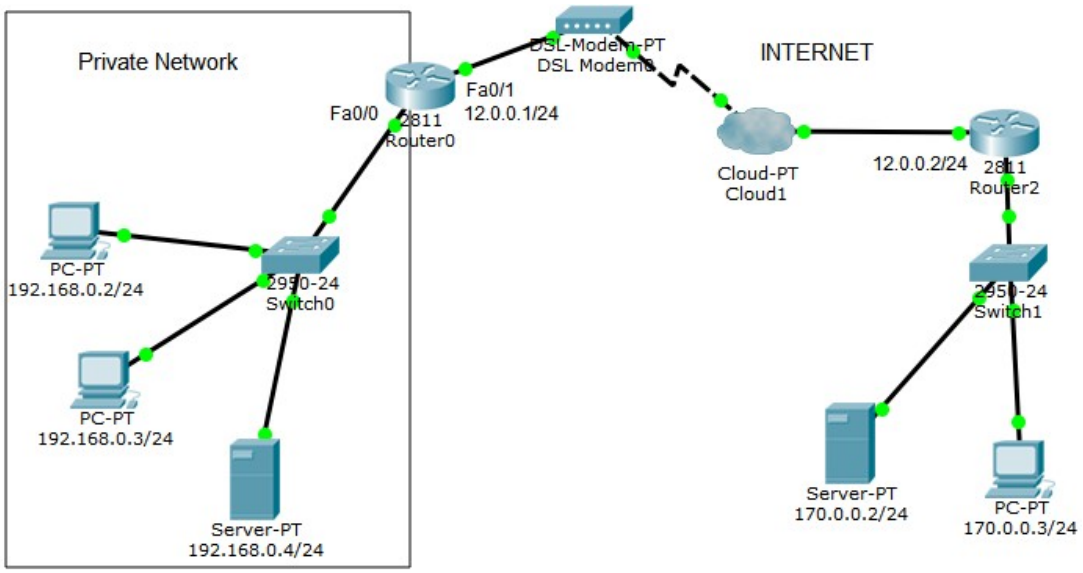
If it did not so, there would be lines with same values for Inside global, Outside local and Outside global. Then the router wouldn't be able to tell which DNAT translation it was supposed apply.

```
Router#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
icmp 12.0.0.1:1        192.168.0.3:1      170.0.0.3:1        170.0.0.3:1
icmp 12.0.0.1:2        192.168.0.3:2      170.0.0.2:2        170.0.0.2:2
tcp 12.0.0.1:1024      192.168.0.4:1025   170.0.0.2:80       170.0.0.2:80
tcp 12.0.0.1:1025      192.168.0.2:1025   170.0.0.2:80       170.0.0.2:80
tcp 12.0.0.1:1026      192.168.0.2:1026   170.0.0.2:80       170.0.0.2:80
tcp 12.0.0.1:1027      192.168.0.3:1025   170.0.0.2:80       170.0.0.2:80
tcp 12.0.0.1:1028      192.168.0.2:1027   170.0.0.2:80       170.0.0.2:80

Router#
```

5. Practical exercise

We will now redo the previous exercise, now, using NAPT a dynamic NAT with overload.
(Use the saved configuration or download it again)



a) Check IPv4 connectivity with ICMP echo requests.

Because left side network is private and right side public (internet), the left side network fails to communicate with right side network and vice versa.

b) Use static NAPT on Router 0 to allow internet access to the 192.168.0.4 HTTP server. The service must be available at address 12.0.0.1, port number 8080.

First we settle NAT sides:

```
(config)#interface Fa0/0
(config-if)#ip nat inside
(config)#interface Fa0/1
(config-if)#ip nat outside
```

Now we can set static NAPT (remember HTTP runs over TCP and, by default, uses port number 80):

```
(config)#ip nat inside source static tcp 192.168.0.4 80 12.0.0.1 8080
```

c) Check the created static NAPT configuration.

Go to PC 170.0.0.3 and open a Web Browser, then enter URL **http://12.0.0.1:8080**. This is Router 0 outside interface, the port number must be explicitly declared because it's not the standard port number for the HTTP service.

Notice that 192.168.0.4 server is not as exposed to external access as before, only TCP traffic to port 8080 of address 12.0.0.1 will go through. This also means that, unlike before, this server is no able to access internet addresses. But we will fix that next.

d) Apply dynamic NAT with overload to allow private nodes access to the internet

Creating and address pool is now pointless because we will use only one public address. Instead of declaring a pool we can directly refer the interface.

Though, an ACL to match traffic to be translated is required:

```
(config)#no access-list 10
(config)#access-list 10 permit 192.168.0.0 0.0.0.255
```

Now we can set dynamic NAT with overload:

```
(config)#ip nat inside source list 10 interface Fa0/1 overload
```

Because we chose not to create an address pool we use the interface parameter to identify the single address to be used and overloaded.

e) See dynamic NAT with overload working

Use the Inspect tool to display Router 0 NAT table. Keep it visible.

Now, send ICMP echo requests from PCs 192.168.0.2 and 192.168.0.3 to internet nodes.

Open the Web Browsers in PCs 192.168.0.2 and 192.168.0.3 and access to URL http://170.0.0.2