Department of Computer Science and Engineering

Islamic University of Technology (IUT)

A subsidiary organ of OIC

Laboratory Report

CSE 4512: Computer Networks Lab

Name: Md Farhan Ishmam

Student ID: 180041120

Section: CSE-1

Semester: Fifth

Academic Year: 2021

Date of Submission: 03-11-21

Title: Configuring Network Address Translation (NAT) in Cisco Devices

Objective:

1. Describe the concept of Network Address Translation (NAT)
2. Explain different types of NAT configuration
3. Implement NAT in a given topology

Devices/ software Used:

1. Device: Windows PC
2. Software: Cisco Packet Tracer 7.3.0

Theory:

Static NAT: Static NAT allows one-to-one mapping between the local and global addresses. We have to configure one global IP address for each of the internal hosts that we want NAT to translate. The command to enable the static translation is as follows:

```
Router(config)# ip nat inside source static local_ip global_ip
```

After we've specified the translation, we need to - specify the inside interface and the outer interface. The inside interface denotes that the hosts connected to this interface will have their IPs translated to the global one. Outside interface denotes that through this interface the translated packets will go out to the world. There can be more than one inside or outside interface. After specifying these interfaces, NAT will start the specified translations. The inside and outside interfaces need to be specified for the other two NAT types also. Following are the commands to specify the inside and outside interfaces:

```
Router(config-if)# ip nat inside
Router(config-if)# ip nat outside
```

Dynamic NAT: Dynamic NAT establishes a mapping between a local address and a pool of global addresses. For a single local address, a global IP address will be selected from the pool of addresses dynamically. When not in use, the assigned global IP address will be released after a certain time-out period, so that the other hosts can reuse that global address. Since we do not need to manually configure every mapping, dynamic NAT is more convenient to use.

In order to configure Dynamic NAT, we need to first create an access list permitting the local addresses to get translated. Then, we have to specify the pool of global addresses from where the IPs will be allocated. The pool is a range of IP addresses in a given network where the subnet mask will specify the corresponding network portion. The command to specify the pool is as follows:

```
Router(config)# ip nat pool POOL_NAME start_ip end_ip netmask
subnet_mask
```

Then we need to establish the relation between the earlier defined access list and nat pool through the following command:

```
Router(config)# ip nat inside source list access_list_number pool
POOL_NAME
```

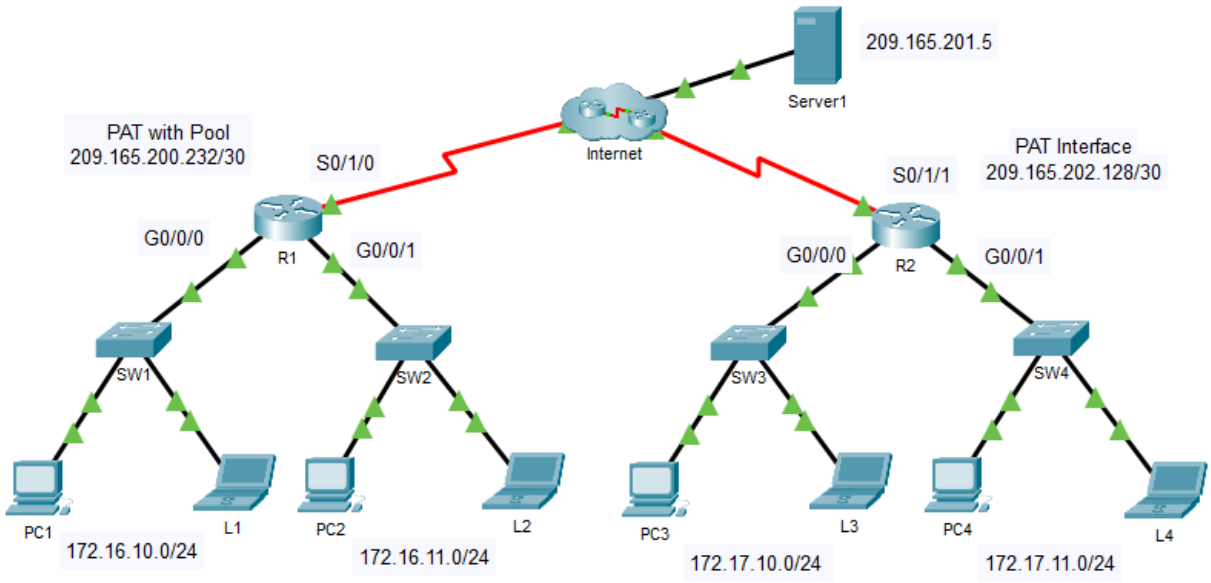
Afterward, similar to static NAT, the inside and outside interfaces need to be specified.

PAT: For dynamic NAT, in the worst case, we'd need as many global IP addresses as the internal hosts. This is not plausible in most circumstances where we have limited global IP and hundreds of local hosts. It's where PAT comes in. PAT establishes a many-to-one mapping between multiple local hosts and a single global IP address. It uses the Port (TCP/UDP port) information to distinguish between different internal hosts and assign a single global IP to all those addresses thus greatly conserving the global address pool.

The configuration of PAT is almost similar to dynamic NAT except you just have to add the overload keyword at the very end while specifying the relation between the access list and nat pool. The command format is below:

```
Router(config)# ip nat inside source list access_list_number pool
POOL_NAME overload
```

Diagram of the experiment(s):



Working Procedure:

Part 1: Configure Dynamic NAT with Overload

Step 1: Configure traffic that will be permitted

On R1, configure one statement for ACL 1 to permit any address belonging to 172.16.0.0/16.

```
R1(config)# access-list 1 permit 172.16.0.0 0.0.255.255
```

The screenshot of the command is given below:

```
R1>en
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#access-list 1 permit 172.16.0.0 0.0.255.255
```

Step 2: Configure a pool of addresses for NAT

Configure R1 with a NAT pool that uses the two useable addresses in the 209.165.200.232/30 address space. The following command is used:

```
R1(config)# ip nat pool pool_1 209.165.200.233 209.165.200.234
netmask 255.255.255.252
```

The screenshot of the command executed is given below:

```
R1(config)#ip nat pool pool_1 209.165.200.233 209.165.200.234
netmask 255.255.255.252
R1(config)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1/0,
changed state to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1/0,
changed state to up
```

Step 3: Associate ACL 1 with the NAT pool and allow addresses to be reused

The name of the pool was given as pool_1. The following command was executed:

```
R1(config)# ip nat inside source list 1 pool pool_1 overload
```

The screenshot below shows the execution of the command:

```
R1(config)#ip nat inside source list 1 pool pool_1 overload
R1(config)#
```

Ctrl+F6 to exit CLI focus Copy Paste

Step 4: Configure the NAT interfaces

Configure R1 interfaces with the appropriate inside and outside NAT commands. The following commands were executed:

```
R1(config)# interface s0/1/0
R1(config-if)# ip nat outside
R1(config-if)# interface g0/0/0
R1(config-if)# ip nat inside
R1(config-if)# interface g0/0/1
R1(config-if)# ip nat inside
```

The screenshot of the commands executed is given below:

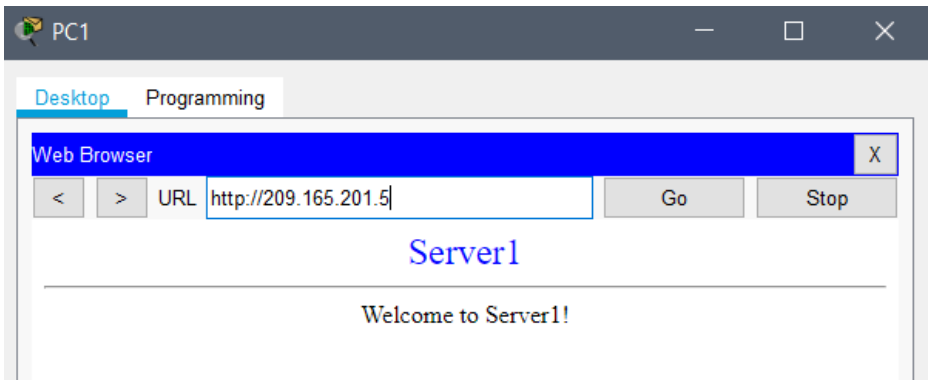
```
R1(config)#int s0/1/0
R1(config-if)#ip nat outside
R1(config-if)#int g0/0/0
R1(config-if)#ip nat inside
R1(config-if)#int g0/0/1
R1(config-if)#ip nat inside
```

Part 2: Verify Dynamic NAT with Overload Implementation

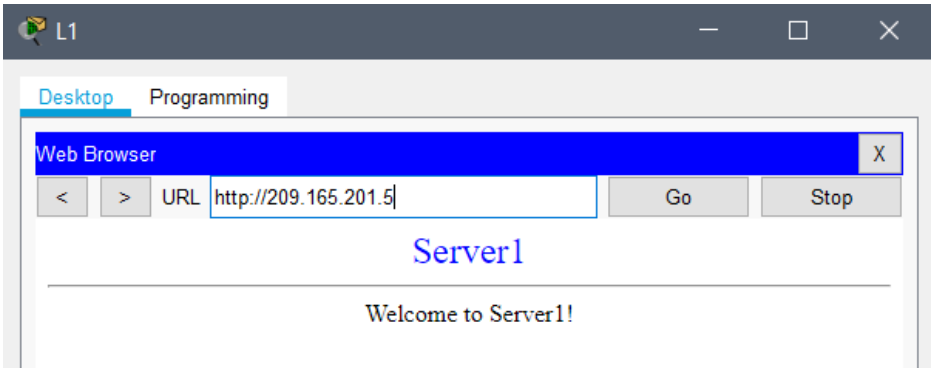
Step 1: Access services across the internet

From the web browser of each of the PCs that use R1 as their gateway (PC1, L1, PC2, and L2), access the web page for Server1.

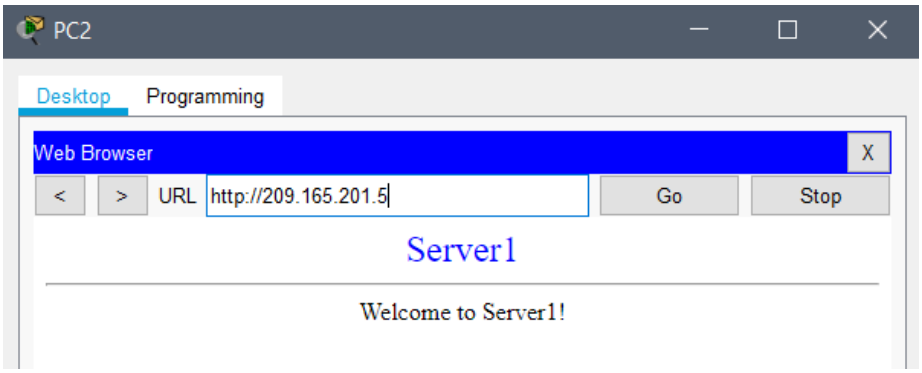
PC1:



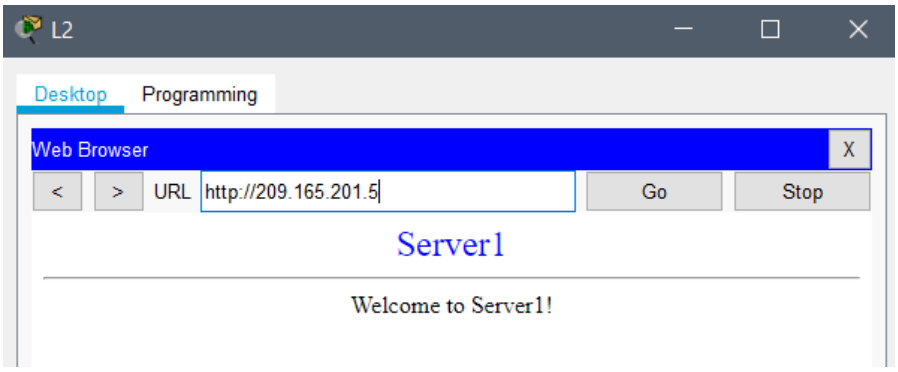
L1:



PC2:



L2:



Step 2: View NAT translations

View the NAT translations on R1. The following command was used:

```
R1# show ip nat translations
```

The execution of the command is given below:

```
R1#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
tcp 209.165.200.233:1024172.16.10.11:1025  209.165.201.5:80   209.165.201.5:80
tcp 209.165.200.233:1025172.16.10.10:1025  209.165.201.5:80   209.165.201.5:80
tcp 209.165.200.233:1026172.16.10.11:1026  209.165.201.5:80   209.165.201.5:80
tcp 209.165.200.233:1027172.16.11.10:1027  209.165.201.5:80   209.165.201.5:80
tcp 209.165.200.233:1028172.16.11.11:1026  209.165.201.5:80   209.165.201.5:80
```

Part 3: Configure PAT using an Interface

Step 1: Configure traffic that will be permitted

On R2, configure one statement for ACL 2 to permit any address belonging to 172.17.0.0/16. The following commands were executed:

```
R2>enable
R2#configure terminal
R2(config)#access-list 2 permit 172.17.0.0 0.0.255.255
```

The screenshot of the commands executed is given below:

```
R2>en
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#access-list 2 permit 172.17.0.0 0.0.255.255
```

Step 2: Associate ACL 2 with the NAT interface and allow addresses to be reused

Enter the R2 NAT statement to use the interface connected to the internet and provide translations for all internal devices. The following commands were used:

```
R2(config)# ip nat inside source list 2 interface s0/1/1
overload
```

The screenshot of the commands executed is given below:

```
R2(config)#ip nat inside source list 2 interface s0/1/1 overload
R2(config)#
```

Step 3: Configure the NAT interfaces

Configure R2 interfaces with the appropriate inside and outside NAT commands. The following commands were executed:

```
R2(config)#int s0/1/1
R2(config-if)#ip nat outside
R2(config-if)#int g0/0/0
R2(config-if)#ip nat inside
R2(config-if)#int g0/0/1
R2(config-if)#ip nat inside
R2(config-if)#exit
R2(config)#exit
```

The screenshot of the commands executed is given below:

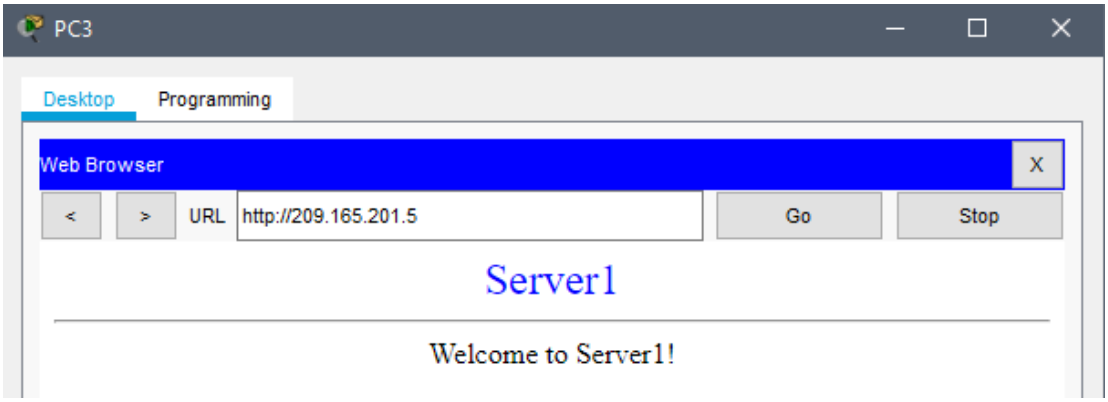
```
R2(config)#int s0/1/1
R2(config-if)#ip nat outside
R2(config-if)#int g0/0/0
R2(config-if)#ip nat inside
R2(config-if)#int g0/0/1
R2(config-if)#ip nat inside
R2(config-if)#exit
R2(config)#exit
R2#
%SYS-5-CONFIG_I: Configured from console by console
```

Part 4: Verify PAT Interface Implementation

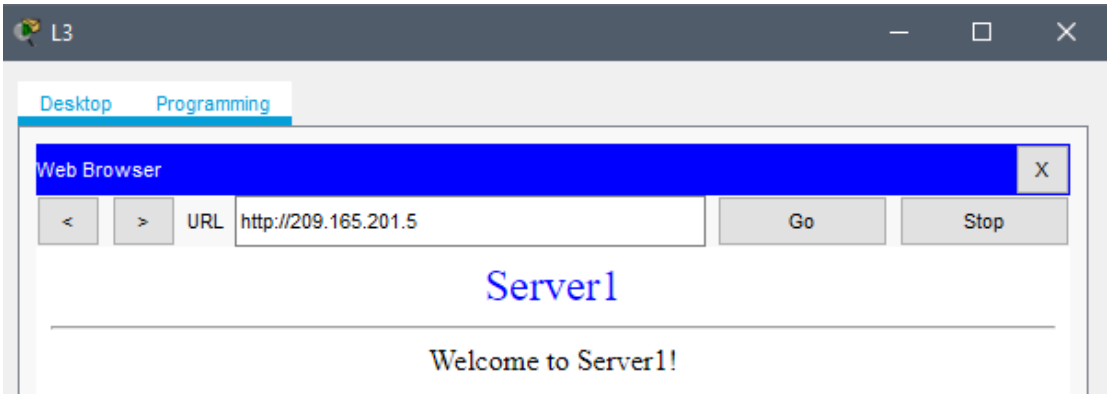
Step 1: Access services across the internet

From the web browser of each of the PCs that use R2 as their gateway (PC3, L3, PC4, and L4), access the web page for Server1. The screenshots are provided below:

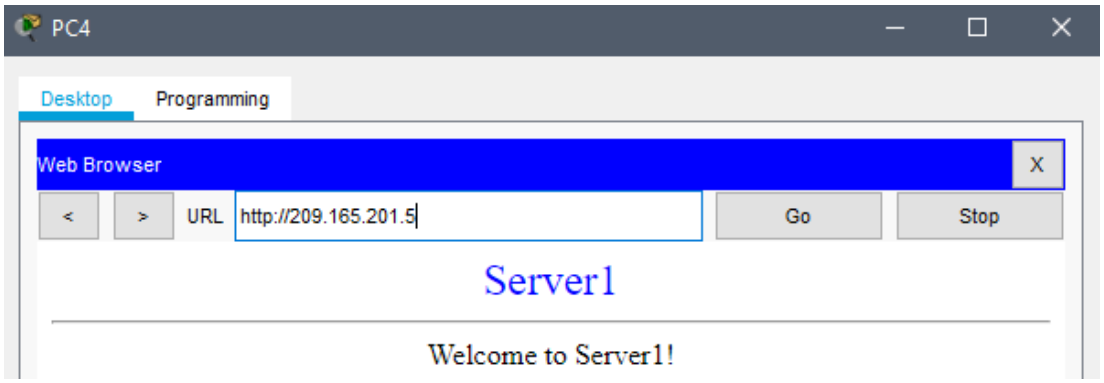
PC3:



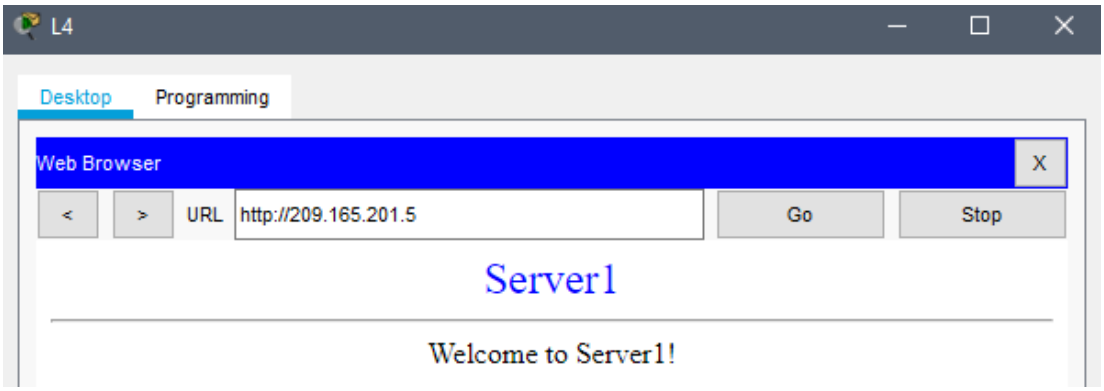
L3:



PC4:



L4:



Step 2: View NAT translations

View the NAT translations on R2. We used the show ip nat translations command to do so. The screenshot of the execution of the command is given below:

```
R2>en
R2#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
tcp 209.165.202.130:1024172.17.11.10:1025  209.165.201.5:80   209.165.201.5:80
tcp 209.165.202.130:1025172.17.10.11:1025  209.165.201.5:80   209.165.201.5:80
tcp 209.165.202.130:1026172.17.11.11:1025  209.165.201.5:80   209.165.201.5:80
tcp 209.165.202.130:1027172.17.10.10:1027  209.165.201.5:80   209.165.201.5:80
```

Step 3: Compare NAT statistics on R1 and R2

Compare the NAT statistics on the two devices. Weuse the show ip nat statistics in R1 and R2. The screenshot of the command in **R1** is given below:

```
R1>en
R1#show ip nat statistics
Total translations: 0 (0 static, 0 dynamic, 0 extended)
Outside Interfaces: Serial0/1/0
Inside Interfaces: GigabitEthernet0/0/0 , GigabitEthernet0/0/1
Hits: 0  Misses: 0
Expired translations: 0
Dynamic mappings:
-- Inside Source
access-list 1 pool pool_1 refCount 0
 pool pool_1: netmask 255.255.255.252
       start 209.165.200.233 end 209.165.200.234
       type generic, total addresses 2 , allocated 0 (0%), misses 0
```

The screenshot of the command in **R2** is given below:

```
R2>en
R2#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
tcp 209.165.202.130:1024172.17.11.10:1025  209.165.201.5:80   209.165.201.5:80
tcp 209.165.202.130:1025172.17.10.11:1025  209.165.201.5:80   209.165.201.5:80
tcp 209.165.202.130:1026172.17.11.11:1025  209.165.201.5:80   209.165.201.5:80
tcp 209.165.202.130:1027172.17.10.10:1027  209.165.201.5:80   209.165.201.5:80

R2#show ip nat statistics
Total translations: 4 (0 static, 4 dynamic, 4 extended)
Outside Interfaces: Serial0/1/0 , Serial0/1/1
Inside Interfaces: GigabitEthernet0/0/0 , GigabitEthernet0/0/1
Hits: 29  Misses: 4
Expired translations: 0
Dynamic mappings:
```

Questions:

Task # 01:

1. From the web browser of each of the PCs that use R1 as their gateway (PC1, L1, PC2, and L2), access the web page for Server1.
   Question: Were all connections successful?
   Ans: Yes, all the connections were successful.

2. From the web browser of each of the PCs that use R2 as their gateway (PC3, L3, PC4, and L4), access the web page for Server1.
   Question: Were all connections successful?
   Ans: Yes, all the connections were successful.

3. Compare the NAT statistics on the two devices.
   Question: Why doesn't R2 list any dynamic mappings?
   Ans: A pool of global addresses wasn't used in R2, and so, there was no need for dynamic mapping.

Challenges (if any):

- I faced no major challenges while doing this task