Lab 07: Configuration of ACLs, NAT, and DHCP

7.1 Objectives:

- Describe the concept of Access Control List (ACL)
- Describe the concept of Network Address Translation (NAT)
- Describe the concept of Dynamic Host Configuration Protocol (DHCP)
- Implement ACL, NAT, and DHCP for given topologies

7.2 Theory:

As with other labs, this lab will also build up on the concepts and techniques of previous labs. So, make sure you have properly understood the previous lab contents.

This section explores key networking concepts, including Access Control Lists (ACLs), Network Address Translation (NAT), and Dynamic Host Configuration Protocol (DHCP). ACLs are introduced as mechanisms for regulating access to network resources, ensuring that only authorized users or devices can connect to specific services. NAT addresses the limitations of IPv4 address exhaustion by allowing internal devices to use private IP addresses while a gateway router translates these to a smaller number of globally unique addresses for internet communication. DHCP is a protocol that automates the assignment of IP addresses and network configuration, facilitating efficient network management through a server that allocates addresses from a maintained pool.

7.2.1 Access Control List (ACL)

Defining who can/can't access what is basically the gist of ACL. In our day-to-day lives, we apply the ACL concept in many areas. A simple example could be that you need to show your ID card to enter an office. There is a list of employees, and your ID is checked against that list to grant access. Similar access controls are in effect virtually everywhere, especially in places where security is critical. In the digital world, this access control is needed so that only the allowed ones can access a certain digital resource. For example, only admins would be allowed access to the backend of a web server, or only database admins would be allowed to access a database server, etc.
In networked devices, ACLs allow only authorized persons/devices to access a certain resource. For example, you can define that only a certain host device can access your webserver. You can also define ACLs so that hosts belonging to a particular network can not communicate with hosts of certain other networks. More scenarios can be defined depending on the needs of an administrator.
In this part of the lab, we will learn about Cisco IP ACL, i.e., filtering network traffic based on IP address. Several ACL types can be configured on a Cisco device. However, we will only focus on Numbered Standard IPv4 ACL. There are two steps to implement an ACL. First, **define the rule**. Second, **apply the rule to an interface.**

The command format for defining a numbered standard IP ACL is:

```
Router(config)# access-list access_list_number {permit|deny}
                {(source_address source_wildcard)|any}
```

You can either permit or deny a packet based on the source IP of the packet in numbered standard IP ACL. Like the OSPF configuration, you must specify a wildcard mask to permit/deny a range of source IP addresses based on the given pattern. You should remember that whenever you apply an ACL to an interface, all the traffic that does not match any ACL rule will be discarded by default. For example, you have defined an ACL to deny a certain source IP. Whenever you apply that rule to an interface, all packets other than the denied source will also be discarded because there is no matching rule for those packets. So, you must allow other traffic explicitly by defining another ACL. The *any* keyword is handy in this case. To permit (or deny) any packet other than the previously specified rules, you can add the keyword *any* in place of the source_address and source_wildcard like the following:

```
Router(config)# access-list 1 permit any
```

Another thing is you can only use numbers from 1 to 99 to specify the access list number. Other numbers are used for extended numbered ACLs. After defining the ACL rule, we must apply it to an interface. Remember that the ACL has no effect until you apply it. The command format for applying an ACL to an interface is:

```
Router(config-if)# ip access-group access_list_number {in|out}
```

The ACL is applied either for inbound or outbound traffic of an interface, and you need to specify the corresponding keyword, i.e., *in* or *out* for that. One best practice before applying an ACL to an interface is to verify the rule by using the following command:

```
Router# show access-lists
```

7.2.2 Network Address Translation (NAT)

You already know from your theory lectures that IPv6 was born partly due to the address space exhaustion of IPv4. One great technique that was the key to the survival of IPv4 is **NAT**. If not for NAT, IPv4 would be long gone by now. And that gave the world some time to adopt IPv6 on a mass scale. In this part of the lab, you will learn about this special technique called NAT.

Basically, the idea of NAT is that there will be a set of IP addresses for the hosts in the internal network, and to the outside world, those internal hosts will be exposed using a different set of IP addresses. You know that each host is recognized through its IP address on the internet. To conform with this, each host connected to the internet must have a unique IP, which would readily become nearly impossible considering billions of connected devices.

NAT allows you to assign arbitrary IP addresses from the **Private IP range** to your internal hosts where these addresses are only locally significant, i.e., locally unique. Then, in the edge or gateway router of the network, you will have one or a set of IP addresses that are globally unique. That edge or gateway router will convert/translate from a globally unique address to a locally unique one or vice versa. The outside world will not know the actual IP addresses of internal hosts. Moreover, your organization can buy only a handful of global IP addresses from the ISP but can use those with much greater Private IP addresses for the internal hosts through NAT. The following figure summarizes what we just talked about.

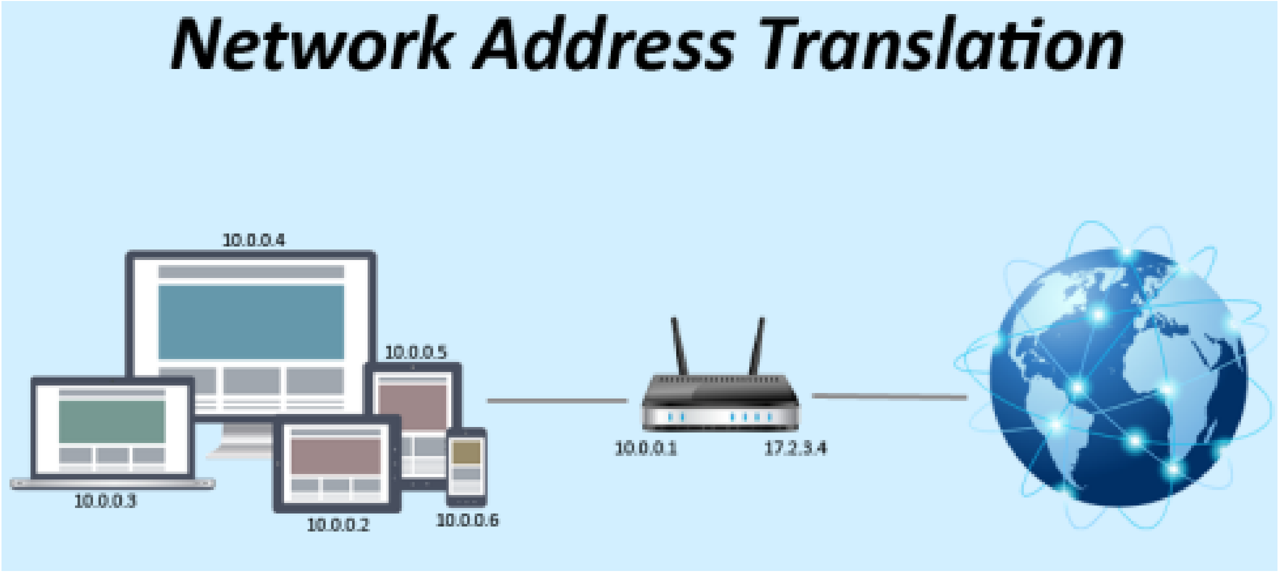

Figure 7.1: Network Address Translation from internal Private IP addresses to the globally unique IP address.

Now that we know the basics of NAT let us get technical. There are three types of NAT that you can define in Cisco devices: Static, Dynamic, and Overloaded or Port Address Translation (PAT).

Static NAT

It allows one-to-one mapping between local and global addresses. You will have to configure one global IP address for each internal host you 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 you have specified the translation, you must do two things — first, **you need to specify the inside interface**, and second, **you need to specify the outside interface**. The inside interface denotes that the hosts connected to it will have their IPs translated to the global one. The outside interface denotes that the translated packets will go out to the world through it. There can be more than one inside or outside interface. After you specify these interfaces, NAT will start the specified translations. You also need to specify these inside and outside interfaces for the other two NAT types. 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

This type of NAT establishes a mapping between a local address and a pool of global addresses. A global IP address will be selected dynamically from the pool for a single local address. When not in use, the assigned global IP will be released after a certain time-out period so other hosts can reuse it. This is more convenient than the static one as you do not need to manually configure every mapping. To configure dynamic NAT, you must create an access list that permits the local addresses to be translated. The command format for defining a numbered standard IP ACL is:

```
Router(config)# access-list access_list_number permit source_address
                source_wildcard
```

Then, you have to specify the pool of global IP 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, you must establish the relation between the earlier defined access list and the nat pool through the following command:

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

After that, you must specify the inside and outside interfaces, such as the static NAT.

Port Address Translation (PAT)
In the worst case, you would need as many global IP addresses as the internal hosts for dynamic NAT. This is not plausible in most circumstances where you have limited global IP and hundreds of local hosts. It is where PAT comes in. PAT establishes a many-to-one mapping between local hosts and a 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 significantly conserving the global address pool. The configuration of PAT is almost similar to dynamic NAT, except you have to add the overload keyword at the very end while specifying the relation between the access list and the nat pool. The command format for the configuration of PAT is as follows:

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

You can also configure PAT using the IP address of an interface rather than a NAT pool. In this case, the local IPs will be translated to the interface's IP address.

```
Router(config)# ip nat inside source list access_list_number interface
                Interface_NAME_Connected_to_Outside_Networks overload
```

7.2.3 Dynamic Host Configuration Protocol (DHCP)

DHCP is a network management protocol used to automate the process of configuring devices on IP networks, thus allowing them to use network services such as DNS, NTP, and any communication protocol based on UDP or TCP. A DHCP server dynamically assigns an IP address and other network configuration parameters to each device on a network so it can communicate with other IP networks.

The DHCP server maintains a database of available IP addresses and configuration information. When it receives a request from a client, the DHCP server determines the network to which the DHCP client is connected, then allocates an IP address or prefix appropriate for the client from that network, and sends configuration information appropriate for that client.

The DHCP server and DHCP client must be connected to the same network link. In larger networks, each network link contains one or more DHCP relay agents. These DHCP relay agents receive messages from DHCP clients and forward them to DHCP servers. DHCP servers send responses back to the relay agent, and the relay agent then sends these responses to the DHCP client on the local network link.

DHCP servers typically grant IP addresses to clients for a limited interval called a lease. DHCP clients are responsible for renewing their IP address before that interval has expired and must stop using it once it has expired if they have not been able to renew it.

7.3 Configuration of ACL, NAT, and DHCP with Cisco Devices

7.3.1 Configure ACL

A router of device model 2911, three switches of device model 2960, and three PCs have been used in the sample topology shown in Figure 7.2 for the ACL configuration.

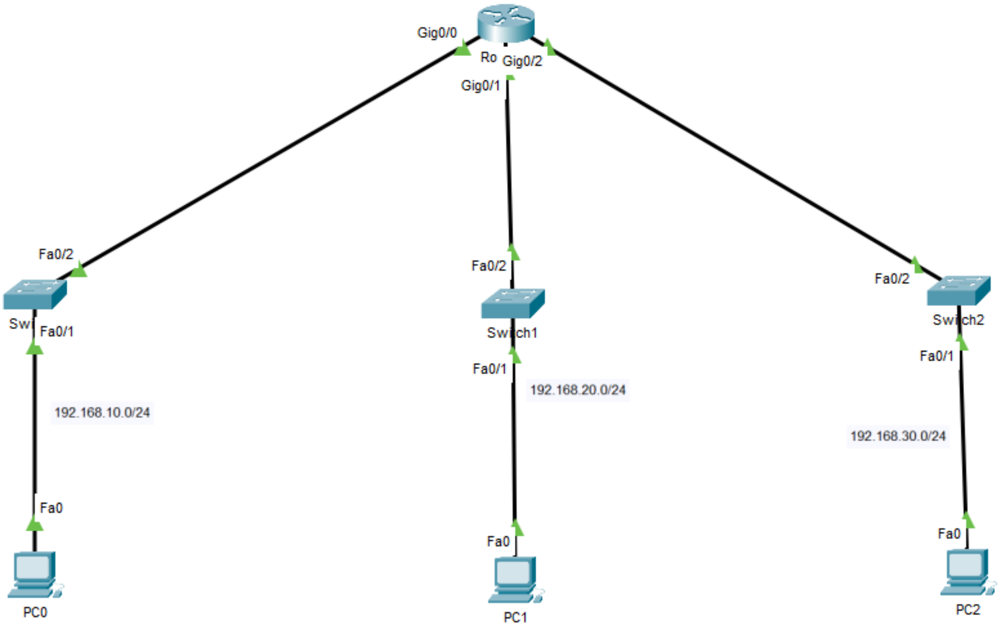

Figure 7.2: A sample network topology for the configuration of ACL.

a. **Configure R1 Interfaces**

```
Router(config)# int g0/0
Router(config-if)# ip address 192.168.10.1 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# int g0/1
Router(config-if)# ip address 192.168.20.1 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# int g0/2
Router(config-if)# ip address 192.168.30.1 255.255.255.0
Router(config-if)# no shutdown

Router(config-if)# end
Router# copy running-config startup-config
```

b. **Configure PC0**

```
IP: 192.168.10.5
Mask: 255.255.255.0
Gateway: 192.168.10.1
```

c. **Configure PC1**

```
IP: 192.168.20.5
Mask: 255.255.255.0
Gateway: 192.168.20.1
```

d. **Configure PC1**

```
IP: 192.168.30.5
Mask: 255.255.255.0
Gateway: 192.168.30.1
```

e. **Define ACL**

```
Router(config)# access-list 1 deny 192.168.10.0 0.0.0.255
Router(config)# access-list 1 permit any
```

f. **Verify ACL**

```
Router# show access-lists
```

g. **Apply ACL**

```
Router(config)# interface g0/2
Router(config-if)# ip access-group 1 out
```

h. **Verify if ACL has been applied to the router interface**

   i. Go to the simulation mode.

   ii. From the event list, allow only ICMP-type messages.

   iii. Ping a message from PC0 to PC2.

   iv. The message will not be sent to the PC1 from the router. Instead, it will send back a notification message to PC0.

7.3.2 Configure Static NAT

Two routers of device model 2811, one switch of device model 2960, two PCs, and one server have been used in the sample topology shown in Figure 7.3 for the NAT configuration.

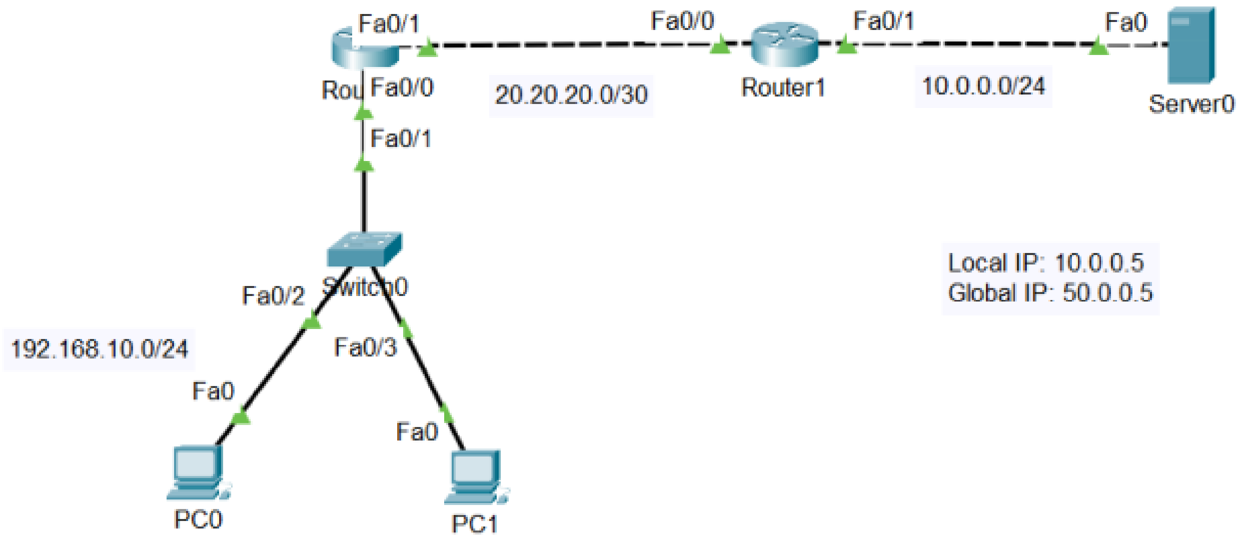

Figure 7.3: A sample network topology for the configuration of NAT.

a. **Configure R0 Interfaces**

```
Router(config)# int fa0/0
Router(config-if)# ip address 192.168.10.1 255.255.255.0
Router(config-if)# no shutdown
Router(config)# int fa0/1
Router(config-if)# ip address 20.20.20.1 255.255.255.252
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# ip route 10.0.0.0 255.255.255.0 20.20.20.2
Router(config)# exit
Router# copy running-config startup-config
```

b. **Configure R1 Interfaces**

```
Router(config)# int fa0/1
Router(config-if)# ip address 10.0.0.1 255.255.255.0
Router(config-if)# no shutdown
Router(config)# int fa0/0
Router(config-if)# ip address 20.20.20.2 255.255.255.252
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# ip route 192.168.10.0 255.255.255.0 20.20.20.1
Router(config)# exit
Router# copy running-config startup-config
```

c. **Configure PC0**

```
IP: 192.168.10.5
Mask: 255.255.255.0
Gateway: 192.168.10.1
```

d. **Configure PC1**

```
IP: 192.168.10.10
Mask: 255.255.255.0
Gateway: 192.168.10.1
```

e. **Configure Server0**

```
IP: 10.0.0.5
Mask: 255.255.255.0
Gateway: 10.0.0.1
```

f. **Enable static NAT insider Router1**

```
Router(config)# ip nat inside source static 10.0.0.5 50.0.0.5
Router(config)# int fa0/1
Router(config-if)# ip nat inside
Router(config)# int fa0/0
Router(config-if)# ip nat outside
Router(config-if)# end
Router# copy running-config startup-config
```

g. **Verify NAT**

```
Router# show ip nat translations
Router# show ip nat statistics
```

7.3.3 Configure Dynamic NAT

The exact network topology with the same device models, as shown in Figure 7.3, has been used for this configuration.

The parts **Configure R0 Interfaces**, **Configure R1 Interfaces**, **Configure all the PCs and Server0**, and **Verify NAT** are the same as the section 7.3.2. The only difference is in the configuration of NAT inside Router1.

a. **Enable static NAT insider Router1**
Create an ACL that permits all the IP addresses from the network 192.168.10.0/24.

```
Router(config)# access-list 1 permit 192.168.10.0 0.0.0.255
```

Create a NAT Pool for the Global IP Addresses from the network 50.0.0.0/30

```
Router(config)# ip nat pool NAT_Pool 50.0.0.1 50.0.0.3 netmask 255.255.255.252
```

Associate the ACL to the NAT pool and configure the interfaces with appropriate **inside** and **outside** NAT commands.

```
Router(config)#ip nat inside source list 1 pool NAT_Pool
Router(config)# int fa0/1
Router(config-if)# ip nat inside
Router(config)# int fa0/0
Router(config-if)# ip nat outside
Router(config-if)# end
Router# copy running-config startup-config
```

7.3.4 Configure DHCP (using server)

A router of device model 2811, two switches of device model 2960, two PCs, two laptops, and one server have been used in the sample topology shown in Figure 7.4 for the DHCP configuration.

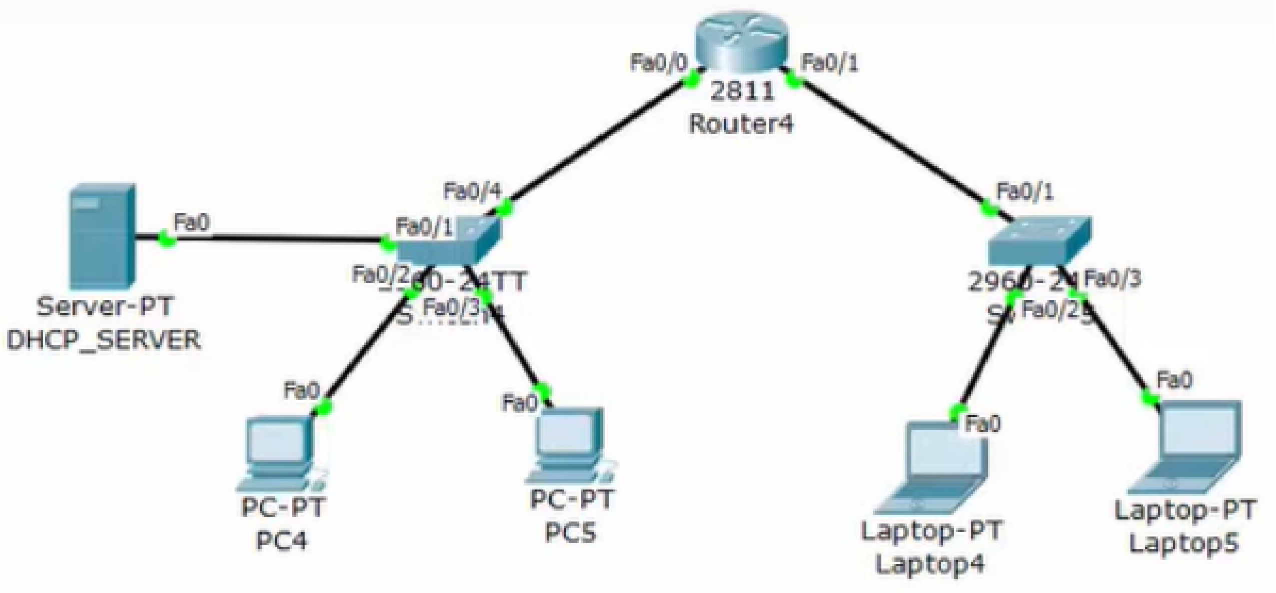

Figure 7.4: A sample network topology for the configuration of DHCP.

a. **Configure DHCP Server**

```
IP Address: 192.168.1.2
Default: 192.168.1.1
```

b. **Make DHCP Pools**
Go to Services and then DHCP, and change the following fields.

```
Pool Name: dotONEnetwork
Default: 192.168.1.1
Start IP: 192.168.1.3
Max Number: 20
```

Select **Add** to add the Address Pool to the server. Select **Save** to save the modification to the server.
To add another pool to the server, change the information as follows and then hit **Add** and **Save** buttons again.

```
Pool Name: dotTWOnetwork
Default: 192.168.2.1
Start IP: 192.168.2.2
Max Number: 20
```

Do not forget to turn on the DHCP server.

c. **Configure R1 Interfaces**

```
R1(config)#int fa0/0
R1(config-if)# ip address 192.168.1.1 255.255.255.0
R1(config-if)# ip helper-address 192.168.1.2
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#int fa0/1
R1(config-if)# ip address 192.168.2.1 255.255.255.0
R1(config-if)# ip helper-address 192.168.1.2
R1(config-if)#no shutdown
R1(config-if)# end
R1# copy running-config startup-config
```

d. **Configure all the PCs**
Just click DHCP and the server will do the rest.

e. **Verify**

```
Ping PC1 from PC0
```

7.3.5 Configure DHCP (using router)

The exact network topology with the same device models, as shown in Figure 7.4, has been used for this configuration.

Configure the DHCP server in the router instead of a server.

```
R1(config)# ip dhcp pool dotONEnetwork
R1(dhcp-config)# default-router 192.168.1.1
R1(dhcp-config)# network 192.168.1.0 255.255.255.0
R1(dhcp-config)# exit

R1(config)# ip dhcp pool dotTWOnetwork
R1(dhcp-config)# default-router 192.168.2.1
R1(dhcp-config)# network 192.168.2.0 255.255.255.0
R1(dhcp-config)# exit
```

The rest of the configuration, i.e., Configure R1 Interfaces, Configure all the PCs, and Verify are same as the section 7.3.4.

Use **show ip dhcp binding** inside the router to see the status of the configured DHCP.

7.4 Tasks:

1. In this task, you will configure ACL with the given requirements for a given scenario. The task description is provided in 7.4.1. No .pka file is provided for this task.
2. You will configure Static NAT following the instructions given in the task. The task description for this task is provided in 7.4.2. You are provided with a .pka file for this task.
3. You will configure dynamic NAT following the instructions given in the task. The task description for this task is provided in 7.4.3. You are provided with a .pka file for this task.
4. You will configure PAT following the instructions given in the task. The task description for this task is provided in 7.4.4. You are provided with a .pka file for this task.
5. In this task, you configure DHCP servers. You will configure servers using a server and a router. The detailed description is given in 7.4.5. No .pka file is provided for this task.

7.4.1 Task 1 - Configure ACL

Scenario

IUT is given an IP address of 192.168.X.0, where X is the last two (2) digits of your Student ID. IUT Medical Center only wants to allow traffic from the IUT Administrative Building and block traffic from Academic Building -1 and Academic Building -2. Make appropriate connections with subnet masks to form the topology. (You can consider the academic buildings sharing a single switch) Then apply ACL to fulfill the criteria.

Objectives

- IUT Medical Center wants to allow traffic only from the IUT Administrative Building and block traffic from Academic Building-1 and Academic Building-2.
- You need to configure a network that ensures that:
  - The IUT Medical Center can communicate with the IUT Administrative Building, Academic Building-1, and Academic Building-2.
  - The Academic Building-1 and Academic Building-2 are blocked from accessing the IUT Medical Center.
- Implement appropriate subnets and apply an Access Control List (ACL) to restrict traffic.

Network Components and Design

i. **IP Addressing**

- Assign the network IP address 192.168.X.0/24, where X is the last two digits of your student ID. For example, if your student ID ends with 45, your network address will be 192.168.45.0/24.
- This address space will be used for assigning IP addresses to all IUT devices.

ii. **Building Segments**

- IUT Administrative Building: Devices in this building should be on the same network as the IUT Medical Center to ensure they can communicate with each other.
- Academic Building-1 and Academic Building-2: These buildings will have separate subnets, and traffic to the IUT Medical Center should be blocked using ACL.

iii. **Subnetting**

- You can use VLSM masks to divide the network into smaller subnets. For example:
  - The first subnet with a subnet prefix of 25 can be used for the IUT Medical Center and the IUT Administrative Building.
  - The second Subnet with a prefix of 26 is for Academic Building-1.
  - And the third Subnet with a prefix of 26 is for Academic Building-2.

iv. **Access Control List (ACL)**

- The IUT Medical Center should be accessible from the IUT Administrative Building but not from Academic Building-1 and Academic Building-2.

7.4.2 Task 2 - Configure Static NAT

Topology

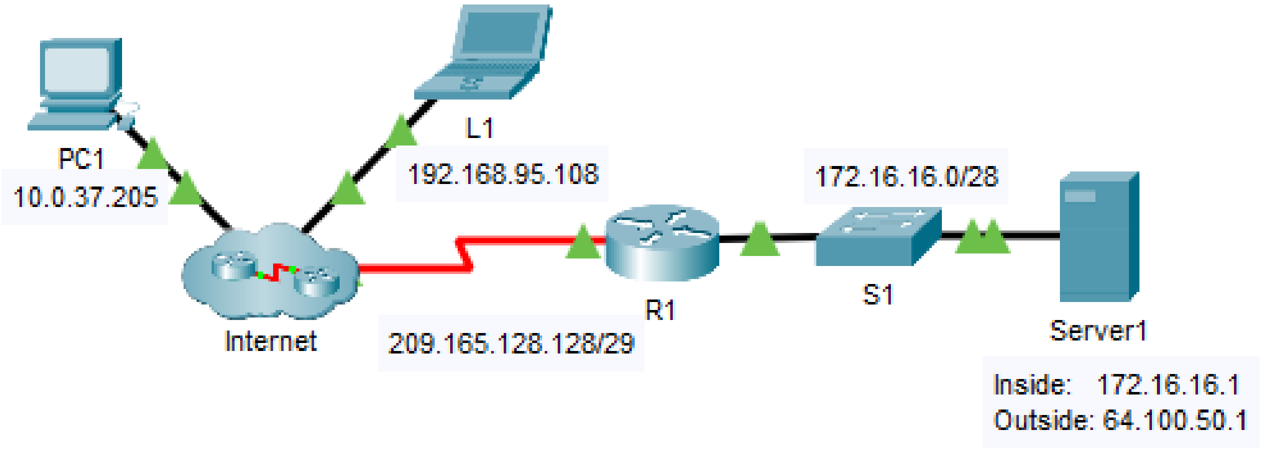

Figure 7.5: A sample network topology for the configuration of Static NAT.

Objectives

Part 1: Test Access without NAT

Part 2: Configure Static NAT

Part 3: Test Access with NAT

Scenario

In IPv4-configured networks, clients and servers use private addressing. Before packets with private addressing can cross then Internet, they need to be translated to public addressing. Servers that are accessed from outside the organization are usually assigned both a public and a private static IP address. In this activity, you will configure static NAT so that outside devices can access and inside server at its public address.

Instruction

Part 1: Test Access without NAT

Step 1: Attempt to connect to Server1 using Simulation Mode.

a. From **PC1** or **L1**, attempt to connect to the **Server1** web page at 172.16.16.1. Use the Web Browser to browse **Server1** at 172.16.16.1. The attempts should fail.

b. From PC1, ping the R1 S0/0/0 interface. The ping should succeed.

Step 2: View R1 routing table and running-config.

a. View the running configuration of **R1**. Note that there are no commands referring to NAT.

b. Verify that the routing table does not contain entries referring to the IP addresses used by **PC1** and **L1**.

c. Verify that NAT is not being used by **R1**.

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

Part 2: Configure Static NAT

Step 1: Configure static NAT statements.

Refer to the Topology shown in Figure 7.5. Create a static NAT translation to map the **Server1** inside address to its outside address.

Step 2: Configure interfaces.

Configure the correct inside and outside interfaces.

Part 3: Test Access with NAT

Step 1: Verify connectivity to the Server1 web page.

a. Open the command prompt on **PC1** or **L1**, attempt to ping to the public address for **Server1**. Pings should succeed.

b. Verify that both **PC1** and **L1** can now access the **Server1** web page.

Step 2: View NAT translations

Use the following commands to verify the static NAT configuration:

```
show running-config
show ip nat translations
show ip nat statistics
```

7.4.3 Task 3 - Configure Dynamic NAT

Topology

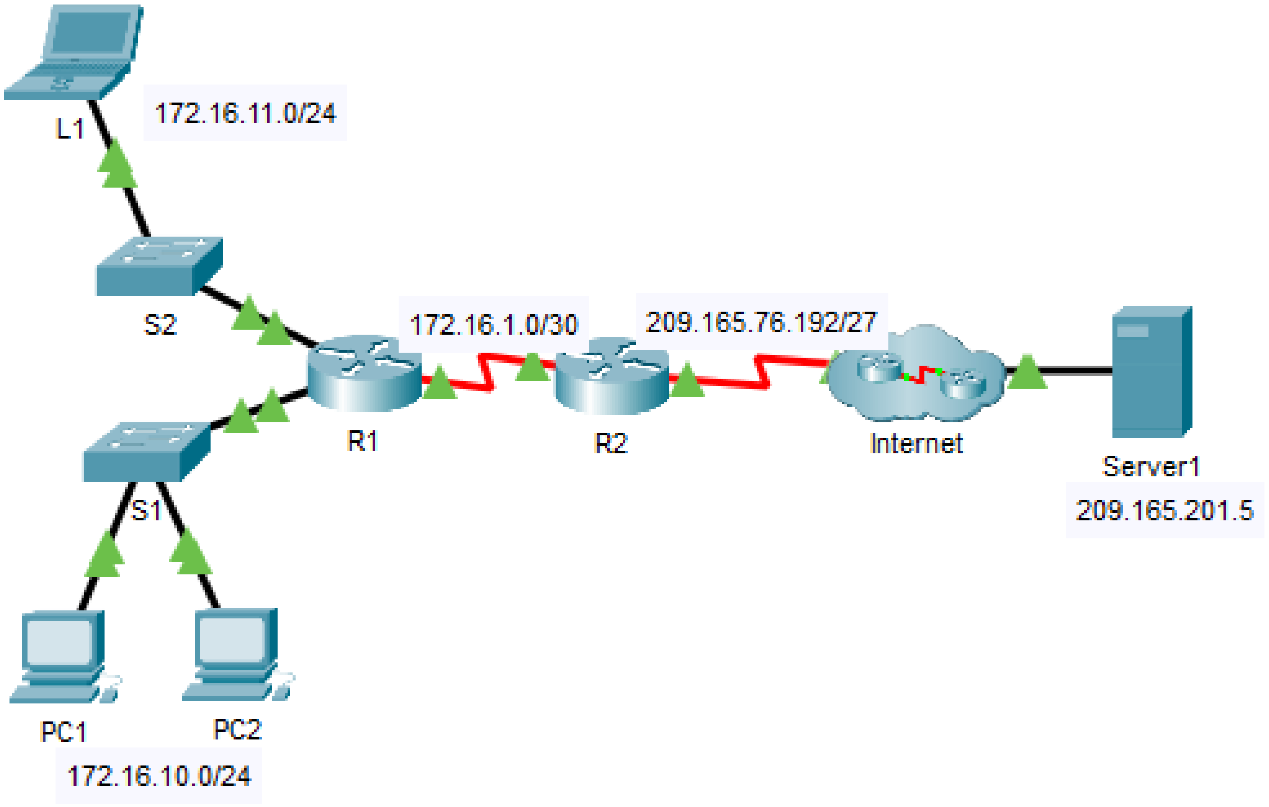

Figure 7.6: A sample network topology for the configuration of Dynamic NAT.

Objectives

Part 1: Configure Dynamic NAT

Part 2: Verify NAT Implementation

Instruction

Part 1: Configure Dynamic NAT

Step 1: Configure traffic that will be permitted.
On **R2**, configure one statement for ACL 1 to permit any address belonging to 172.16.0.0/16.

Step 2: Configure interfaces.
Configure **R2** with a NAT pool that uses all four addresses in the 209.165.76.196/30 address space.
Notice in the topology shown in Figure 7.6, there are 3 network ranges that would be translated based on the ACL created. What will happen if more than 2 devices attempt to access the Internet?

Step 3: Associate ACL1 with the NAT pool.

Step 4: Configure the NAT interfaces.
Configure **R2** interfaces with the appropriate inside and outside NAT commands.

Part 2: Verify NAT Implementation

Step 1: Access services across the Internet.

From the web browser of **L1**, **PC1** or **PC2**, access the web page for **Server1**.

Step 2: View NAT translations

View the NAT translations on R2.

```
show running-config
show ip nat translations
show ip nat statistics
```

7.4.4 Task 4 - Configure PAT

Topology

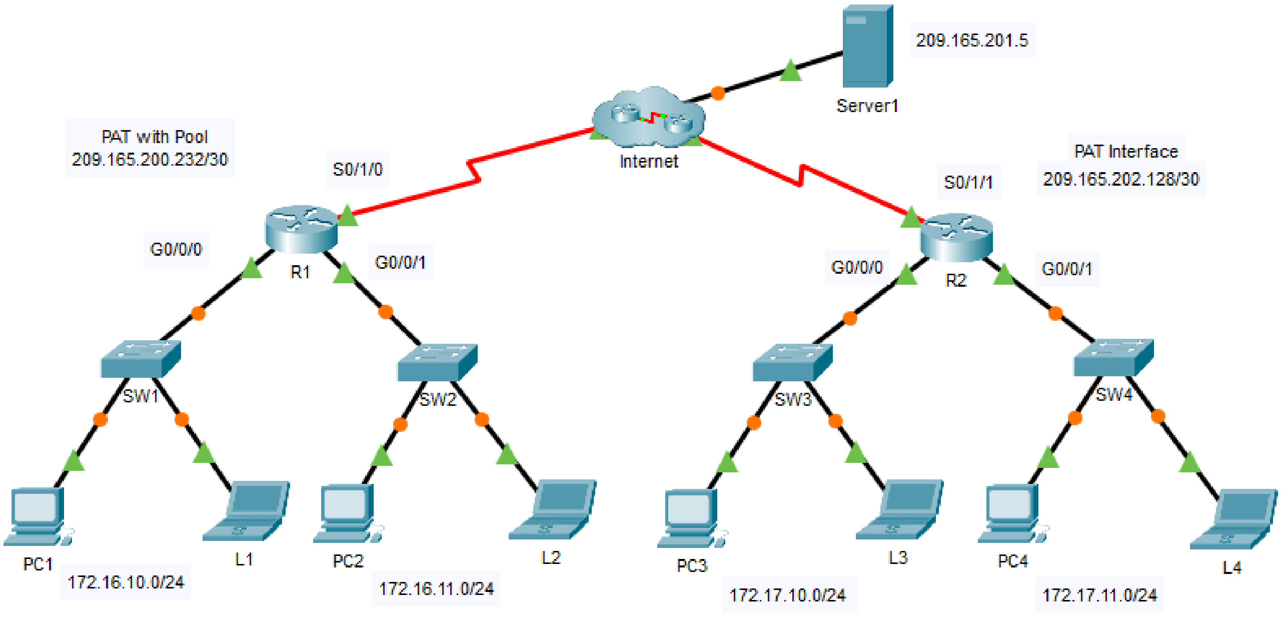

Figure 7.7: A sample network topology for the configuration of PAT.

Objectives

Part 1: Configure Dynamic NAT with Overload

Part 2: Verify Dynamic NAT with Overload Implementation

Part 3: Configure PAT using an Interface

Part 4: Verify PAT Interface Implementation

Instruction

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
```

Step 2: Configure a pool of address for NAT.
Configure **R1** with a NAT pool that uses the two useable addresses in the 209.165.200.232/30 address space.

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

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

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

Step 4: Configure the NAT interfaces.

Configure **R1** interfaces with the appropriate inside and outside NAT commands.

```
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
```

Part 2: Verify Dynamic NAT with Overload Implementation

Step 1: Access services across the Internet.

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

Question: Were all connections successful?

Step 2: View NAT translations

View the NAT translations on **R1.**

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

Notice that all four devices were able to communicate, and they are using just one address out of the pool. PAT will continue to use the same address until it runs out of port numbers to associate with the translation. Once that occurs, the next address in the pool will be used. While the theoretical limit would be 65,536 since the port number field is a 16 bit number, the device would likely run out of memory before that limit would be reached.

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.

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

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

Step 4: Configure the NAT interfaces.

Configure **R2** interfaces with the appropriate inside and outside NAT commands.

Part 4: Verify PAT Interface Implementation

Step 1: Access services across the Internet.

From the web browser of each of the PCs **(PC3, L3, PC4, and L4)** that use **R2** as their gateway, access the web page for **Server1.**

Question: Were all connections successful?

Step 2: View NAT translations

View the NAT translations on **R2.**

Step 3: Compare NAT statistics on R1 and R2.

Compare the NAT statistics on the two devices.

Question: Why doesn't R2 list any dynamic mappings?

7.4.5 Task 5 - Configure DHCP servers

Topology

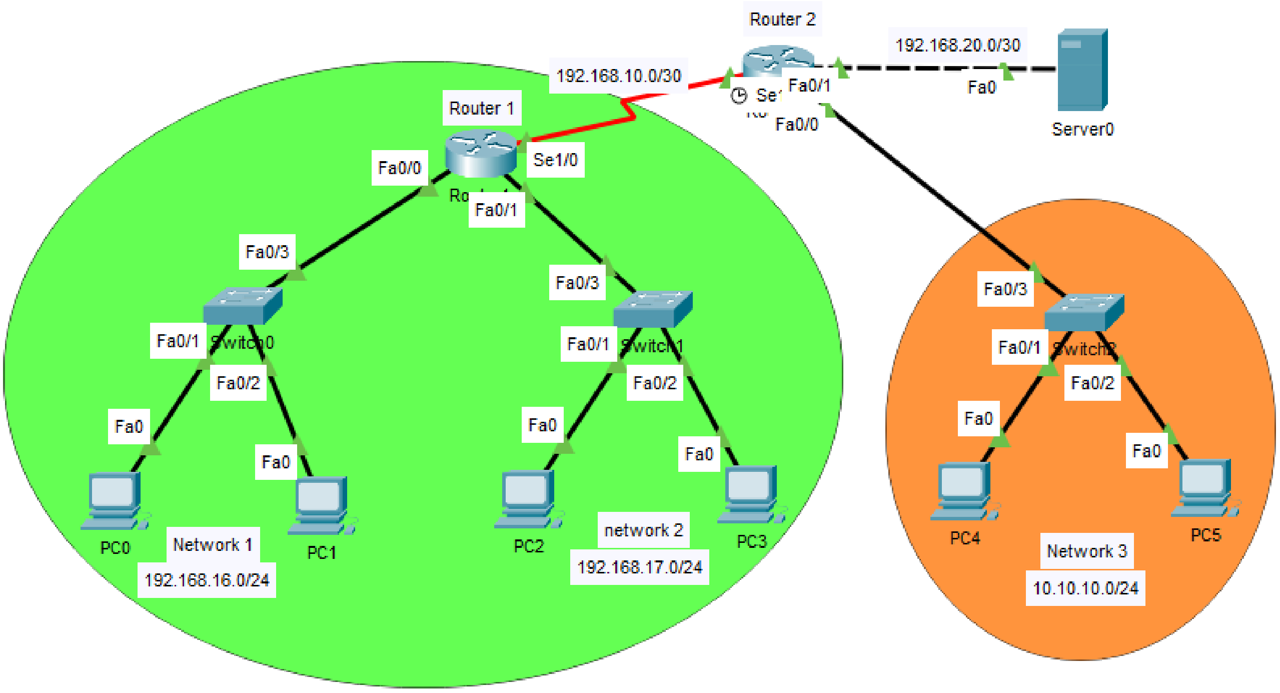

Figure 7.8: A sample network topology for the configuration of DHCP servers.

Scenario

You are tasked with configuring three networks connected to two routers. Devices in Network 1 and Network 2 should receive IP addresses dynamically from Router 1, while devices in Network 3 should receive IP addresses from Server 0. You are also required to configure appropriate routing protocols for inter-network communication.

Network Design Overview

i. **Network 1**

- Subnet: 192.168.16.0/24
- Devices in Network 1 will receive IP addresses dynamically from Router 1 using DHCP.
- Devices in this network need to communicate with Network 2 and Network 3.

ii. **Network 2**

- Subnet: 192.168.17.0/24
- Devices in Network 2 will receive IP addresses dynamically from Router 1 using DHCP.
- Devices in this network need to communicate with Network 1 and Network 3.

iii. **Network 3**

- Subnet: 10.10.10.0/24

- Devices in Network 3 will receive IP addresses dynamically from Server 0 (Which acts as a DHCP server for this network).
- Devices in this network need to communicate with Network 1 and Network 2.

DHCP Configuration

Router 1 (DHCP Server for Network 1 and Network 2)
Router 1 will act as the DHCP server for Network 1 and Network 2. Configure DHCP pools to assign IP addresses dynamically to devices in both networks.

a. **Configure DHCP for Network 1 on Router 1**

- Configure a DHCP Pool Name.
- Specify the address space for the DHCP Pool.
- Define the default Gateway for the Pool.
- Define the DNS Server for the Pool.

b. **Configure DHCP for Network 2 on Router 1**

- Configure a DHCP Pool Name.
- Specify the address space for the DHCP Pool.
- Define the default Gateway for the Pool.
- Define the DNS Server for the Pool.

Server 0 (DHCP Server for Network 3)

- Go to the Services tab.
- Go to DHCP and enable DHCP.
- Set the DHCP Range for the DHCP Pool.
- Set the Default Gateway to the appropriate IP address.
- (Optional) Set DNS Server to 8.8.8.8.

Assign appropriate IP addresses to the router interfaces and the server. Refer to the topology shown in the Figure 7.8 for the IP addressing.

Configure suitable routing protocol in the routers for successful communication.