20201017_8_LOKNATH SAHA

1. Enter global configuration mode.

```
Router> enable
Router# configure terminal
```

2. Configure hostname, enable password and enable secret.

```
Router(config)#hostname GAD
GAD(config)#enable secret class
GAD(config)#enable password cisco
```

3. Configure the interfaces.

```
GAD(config)#int fa0/0
GAD (config-if)#ip address 192.168.14.1 255.255.255.0
GAD(config-if)#no shutdown
GAD(config)#int s 0/0/0
GAD(config-if)#ip address 192.168.15.1 255.255.255.0
GAD(config-if)#clock rate 64000
GAD(config-if)#no shutdown
```

4. Configure the routing protocol and table.

```
GAD(config)#router rip
GAD(config-router)#network 192.168.14.0
GAD(config-router)#network 192.168.15.0
GAD(config-router)#exit
```

5. Configure the line console and line vty 0 4 (the telnet login).

```
Router(config)#line console 0
Router(config-line)#login
Router(config-line)#password cisco
Router(config-line)#exit
Router(config)#
Router(config)#line vty 0 4
Router(config-line)#login
Router(config-line)#password Dhaka
Router(config-line)#exit
Router(config)#
```

6. Save the running configuration to NVRAM.

```
GAD#copy run start
```

7. configuring static route

GAD(config)# **ip route 192.168.16.0 255.255.255.0 192.168.15.2**

8. interface description

GAD(config)# interface e 0
GAD(config-if)# description BRAC University lan building

9. message of the day

GAD(config)# banner motd # THIS IS A RESTRICTER ROUTER #

Remote Access through SSH & Troubleshooting

For checking basic infos - ipconfig /all or ipv6config /all
ping ip address
tracert ip address

R1# show ip or ipv6 interface brief (To check router ports ip addresses)

R1# show ip or ipv6 route (To see which route is connected with which network)

### Remote access with SSH – First go to any pc connected with the router.

PC1-> ssh -l <username> <ip address of the router port with which the PC is connected or default gateway>

Then it will as for password:
<type the password>

Configuring DHCPv4 Using Cisco IOS

1. First I need to check how many **router** networks are connected with the router.

2. Set how many ip addresses need to be excluded 10/12/20 whatever ?

R2 -> ip dhcp excluded-address <starting ip address > <ending ip address >

3. Then need to set the pool and network name for each networks

R2 -> ip dhcp pool <u>Rn-LAN</u>

R2 -> network <network ip address> <network subnet mask>

R2 -> default-router <Router port ip address of <u>Rn</u> with which the network is connected>

R2 -> dns-server <dns-server ip address>

R2 -> domain-name example.com

exit

Note : Rn = R1, R2………

4. Then start the working on Rn routers to set up **<u>relays</u>**

R1(config)# int g0/0 [Router port ip address of <u>Rn</u> with which the network is connected
]
R1(config)# ip helper-address <ip address of s0/0/0 or the port of the R2 route>

exit

5. Last step to configure the <u>R2 or dhcp</u> router port which connected with the isp or internet

```
R2(config)# int g0/0
R2(config)# ip address dhcp
R2(config)# no shutdown
R2# show ip dhcp binding
```

Implementing Static and Dynamic NAT

Dynamic use -> ACL pool names R2ACL -> permit the hots -> create NAT Pool named R2NAT -> integrate (R2NAT + R2ACL)

```
R2 ->en
R2 ->ip access ?
R2 ->ip access-list ?
R2 ->ip access-list standard R2NAT
R2 ->permit 192.168.10.0 ?
R2 ->permit 192.168.10.0 0.0.0.255
R2 ->permit 192.168.20.0 0.0.0.255
R2 ->permit 192.168.30.0 0.0.0.255
R2 ->exit
R2 -> ip nat pool R2POOl ?
R2 -> ip nat pool R2POOl 209.165.202.129 ?
R2 -> ip nat pool R2POOl 209.165.202.129 209.165.202.129 ?
R2 -> ip nat pool R2POOL 209.165.202.129 209.165.202.129 netmask
255.255.255.252
R2 -> ip nat inside source list R2NAT pool R2POOL overload
R2 ->interface s0/1/0
R2 ->ip nat outside
R2 ->exit
R2 ->interface s0/0/1
R2 ->ip nat inside
R2 ->interface s0/0/0
R2 ->ip nat inside
R2 ->interface f0/0
R2 ->ip nat inside
R2 ->exit
```

Static -> manually

```
R2 -> ip nat inside source static 192.168.20.254 ?
R2 -> ip nat inside source static 192.168.20.254 209.165.202.130 [Global ip]
R2 ->exit
R2 -> show ip nat translations
```

In the context of server communication or processing, the terms "listening stage" and "accepting stage" are often associated with server sockets and their lifecycle. Let me explain the differences between these two stages:

1. Listening Stage:

Purpose:The listening stage is the initial phase of a server socket, where it waits and listens for incoming connection requests from clients.

Functionality: During this stage, the server socket is set to a state where it actively monitors for incoming connection attempts. It essentially opens up a communication channel and is ready to accept incoming connections from clients.

Socket State: The server socket is in a "listening" state, indicating that it is prepared to accept connection requests.

2. Accepting Stage:

Purpose: The accepting stage occurs when a connection request is received by the server socket during the listening stage.

Functionality: When a client attempts to connect to the server, the server socket accepts the connection request, creating a new socket specifically for communication with that particular client. This new socket is used for data exchange between the server and the connected client.

Socket State: The server socket transitions from the listening state to an "accepted" state, and a new communication socket is established for further data transmission.

In summary, the listening stage involves preparing the server socket to accept incoming connections, while the

a. Value of X:
   - The value of `X` is the number of echo clients that will be simulated in the network. However, the specific value of `X` is not provided in the pseudocode. You would need additional information from the context or configuration to determine the actual value of `X`.

b. Number of Packets a Client Can Send at a Time:
   - The attribute "MaxPackets" is set to 5 for the echo client: `echo_client.SetAttribute("MaxPackets", ns.core.UintegerValue(5))`. This means that each client will send a maximum of 5 packets to the server.

c. Interval Between Successive Packets:
   - The attribute "Interval" is set to 2.0 seconds for the echo client: `echo_client.SetAttribute("Interval", ns.core.TimeValue(ns.core.Seconds(2.0)))`. This means that the client will send packets with a time interval of 2.0 seconds between successive packets.

In summary:
- b. The client can send a maximum of 5 packets to the server.
- c. The client sends successive packets with a 2.0-second interval between them.

The listening stage and accepting stage are crucial phases in server communication. During the listening stage, the server initializes a passive socket to await incoming connection requests. This sets up the groundwork for potential interactions, allowing the server to be receptive to external requests. In contrast, the accepting stage occurs when a connection request is received, leading to the establishment of a connection with the client. Here, the server actively acknowledges and processes the incoming request, facilitating data exchange. The listening stage prepares the server to accept multiple connections, while the accepting stage handles the actual connection establishment and subsequent communication. Together, these stages enable servers to efficiently manage and respond to client requests in a networked environment, forming a fundamental aspect of robust server-client interactions.