EXERCISE 5: DEMONSTRATION OF STATIC AND DEFAULT ROUTING

Exercise 5. a: Demonstration of Static Routing

Objective: To demonstrate the configuration of IP Addressing with Subnetting in WAN Configuration

Pre-requisite: IP Address, Range of IP Address, Classes of IP Address, Subnetting

Components:

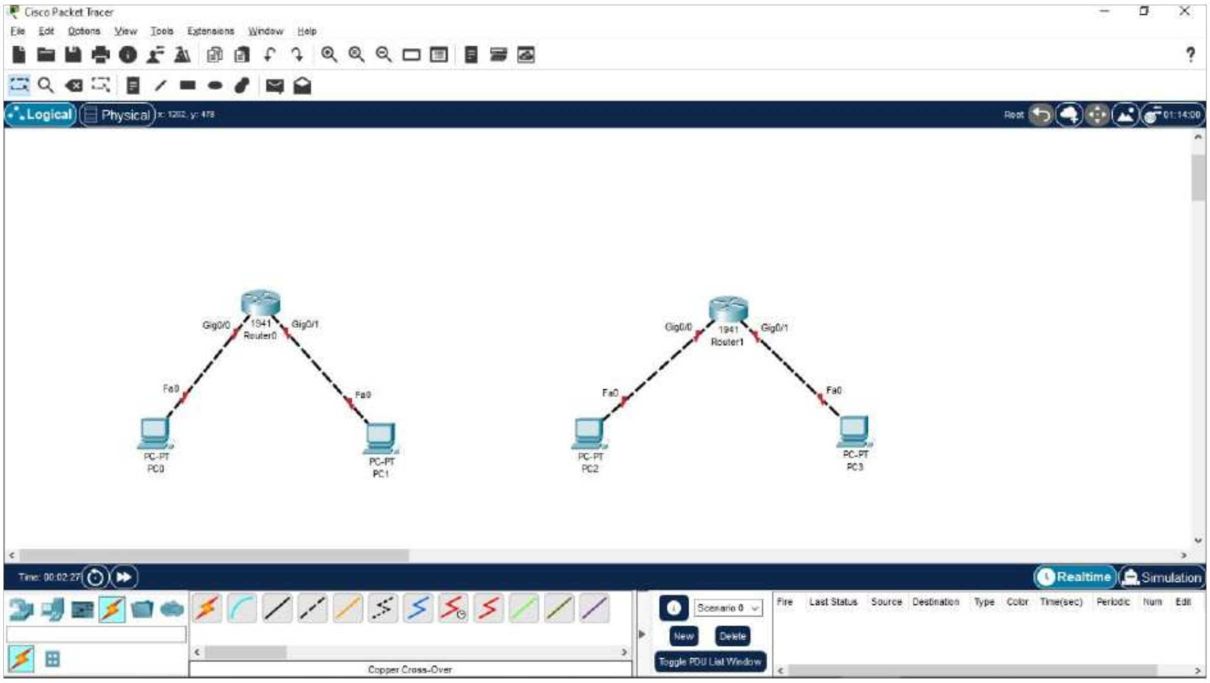
Devices	Required Nos
PCs	4
Copper cross-over Cables	4
Routers	2
Serial DCE	1

Addressing Table:

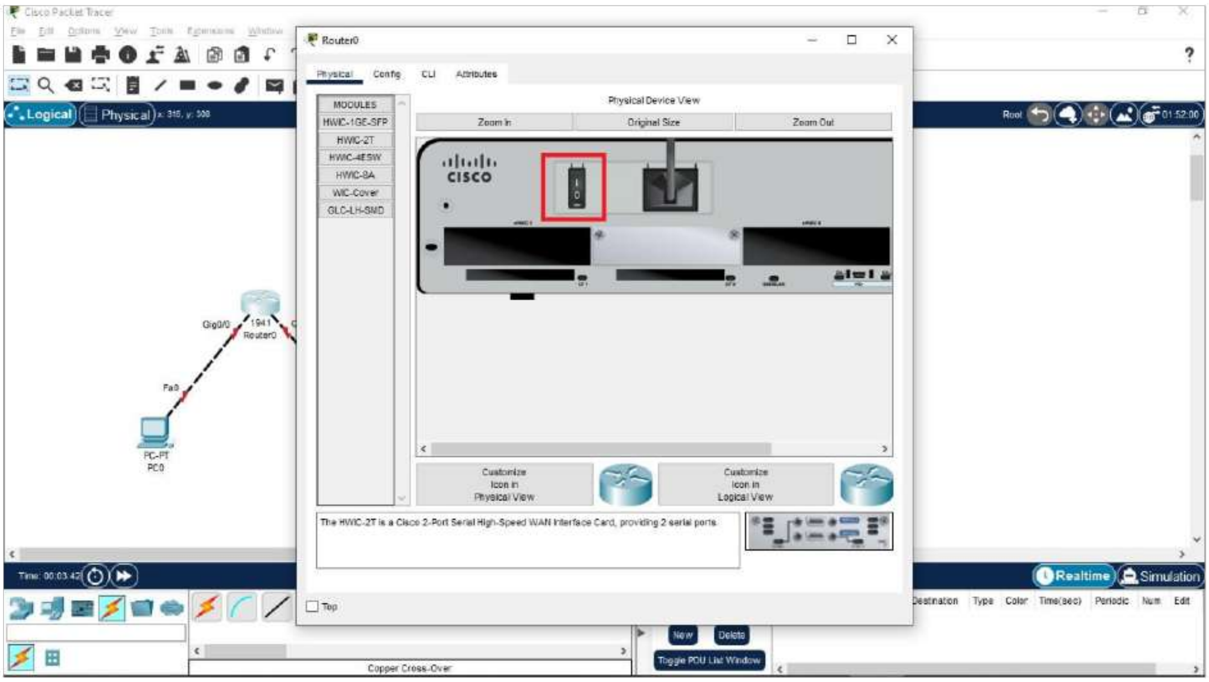
Device	Interface	IP Address	Subnet Mask	Gateway
PC0	Fa0/0	192.168.10.2	255.255.255.224	192.168.10.1
PC1	Fa0/0	192.168.10.34	255.255.255.224	192.168.10.33
PC2	Fa0/0	192.168.10.98	255.255.255.224	192.168.10.97
PC3	Fa0/0	192.168.10.130	255.255.255.224	192.168.10.129
Router0	Gigabit 0/0	192.168.10.1	255.255.255.224	-
Router0	Gigabit 0/1	192.168.10.33	255.255.255.224	-
Router0	Se0/1/0	192.168.10.65	255.255.255.224	-
Router1	Gigabit 0/0	192.168.10.97	255.255.255.224	-
Router1	Gigabit 0/1	192.168.10.129	255.255.255.224	-
Router1	Se0/1/0	192.168.10.66	255.255.255.224	-

Procedure:

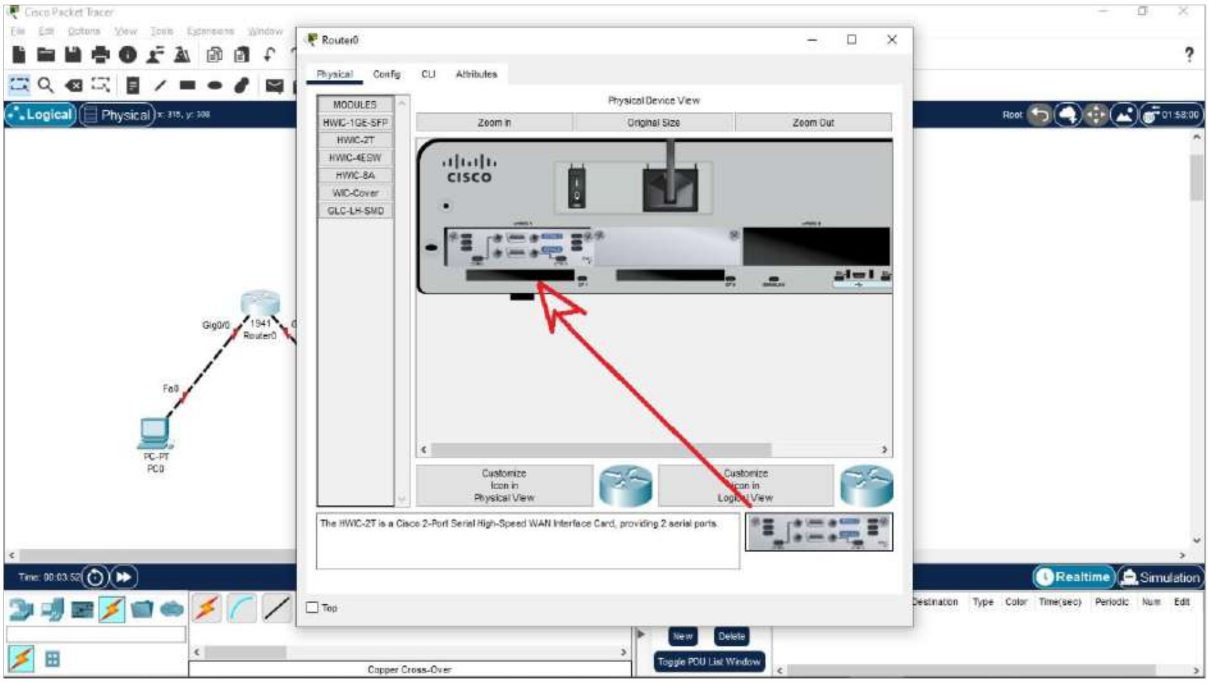
Step 1: Drag 4 PCs and 2 routers in the console area as shown in figure



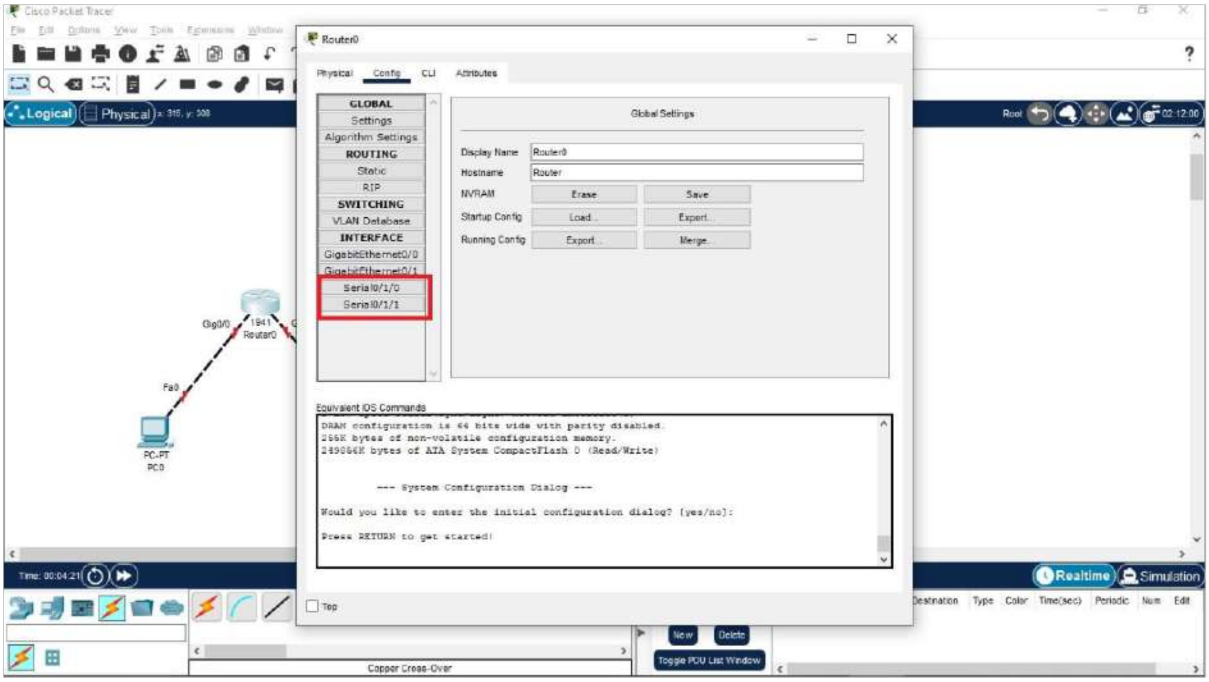
Step 2: Click on Router0 and go to Physical tab. Click HWIC2T in the left pane and Click on zoom in. Switch off the Hardware



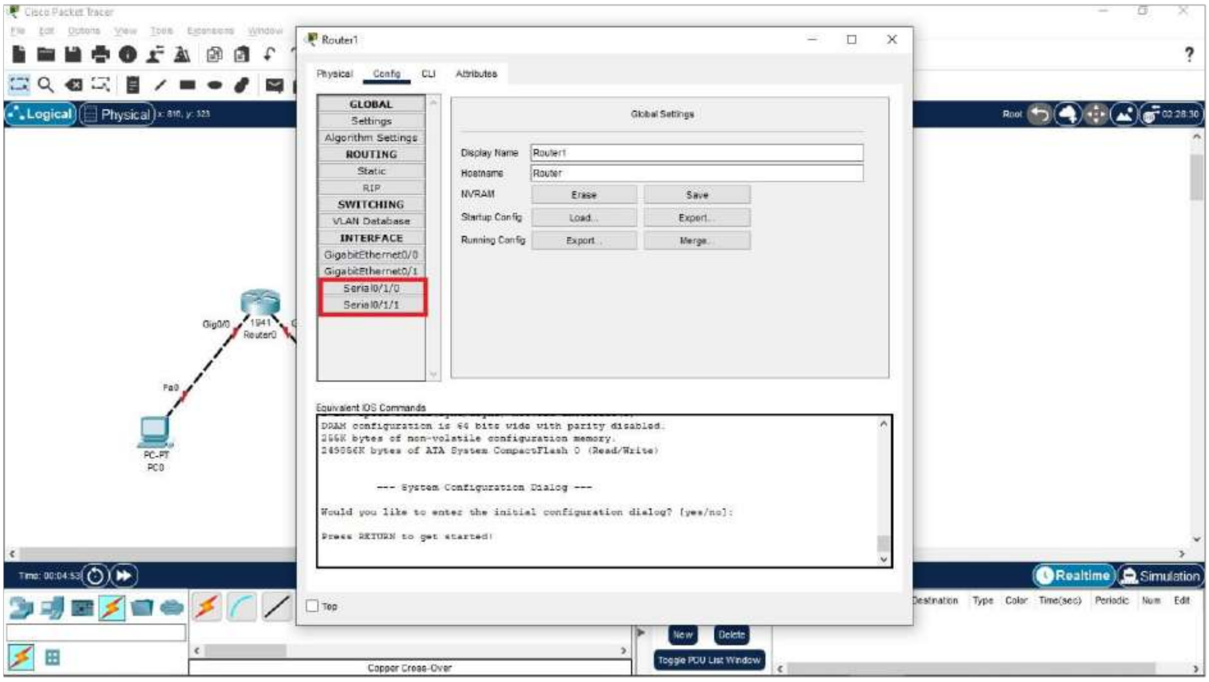
Step 3: Find the console from lower right corner. Drag and drop the console in the empty area as shown in the figure.



Step 4: Now again switch on the hardware and check in config tab for 2 serial ports added.

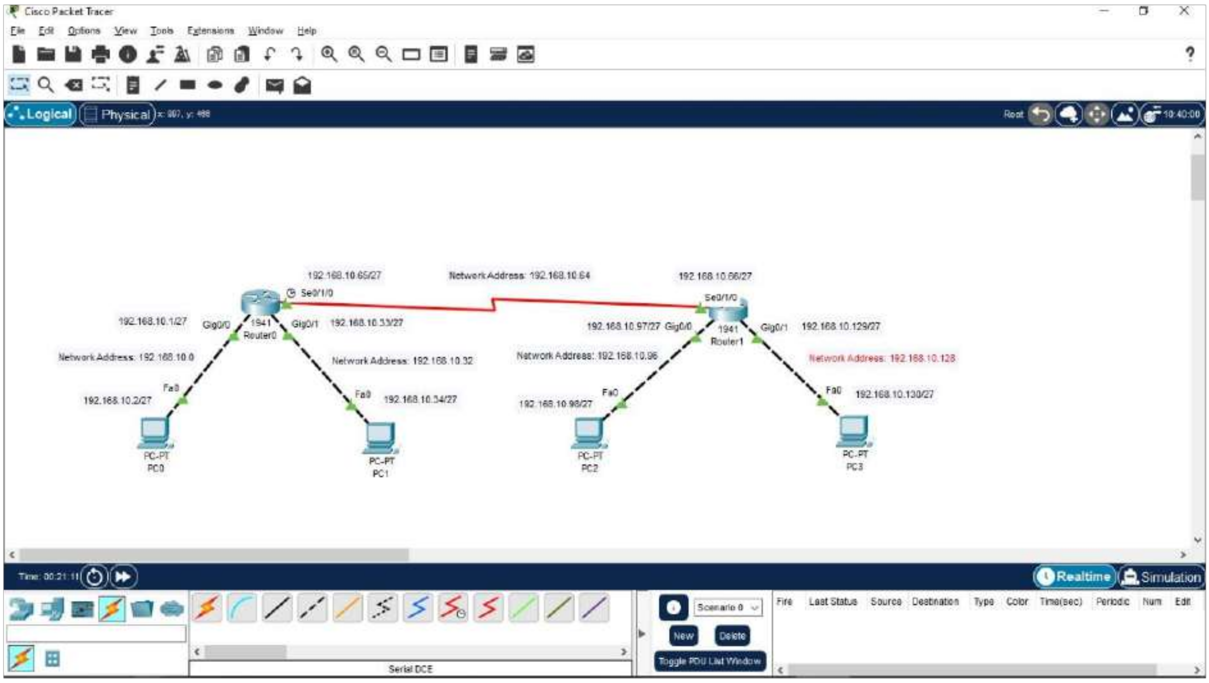


Step 5: Repeat the same procedure (Step 3 and Step 4) for Router1



Step 6: Now the PCs are physically connected through Router. To establish logical connectivity, assign IP addresses for 4 PCs (each 1 interface and corresponding router interface as gateway) and 2 Routers (each 3 ip addresses for 3 interfaces) as shown in the following table.

Device	Interface	IP Address	Subnet Mask	Gateway
PC0	Fa0/0	192.168.10.2	255.255.255.224	192.168.10.1
PC1	Fa0/0	192.168.10.34	255.255.255.224	192.168.10.33
PC2	Fa0/0	192.168.10.98	255.255.255.224	192.168.10.97
PC3	Fa0/0	192.168.10.130	255.255.255.224	192.168.10.129
Router0	Gigabit 0/0	192.168.10.1	255.255.255.224	-
Router0	Gigabit 0/1	192.168.10.33	255.255.255.224	-
Router0	Se0/1/0	192.168.10.65	255.255.255.224	-
Router1	Gigabit 0/0	192.168.10.97	255.255.255.224	-
Router1	Gigabit 0/1	192.168.10.129	255.255.255.224	-
Router1	Se0/1/0	192.168.10.66	255.255.255.224	-

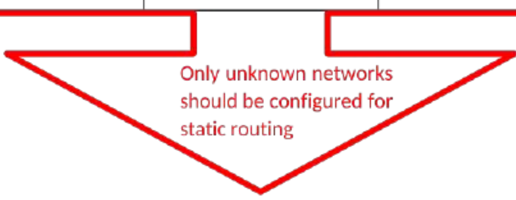


Scenario with Network Address for each link

Step 7:

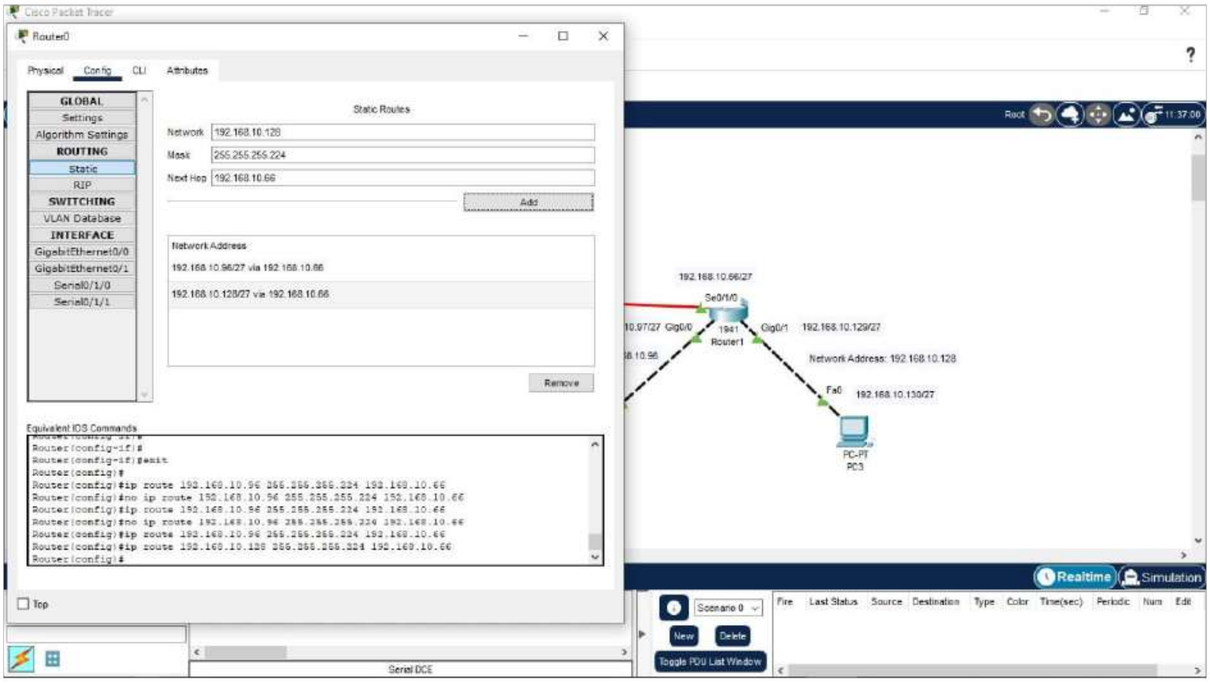
- **To enable packet transmission among the devices in the scenario, Static Routing has to be configured.**
- **To configure static routing, unknown networks and next-hop IP address to reach the unknown network for Router0 and Router1 has to be determined.**
- **Note: While specifying devices we should use IP-Address and while specifying network we should use Network Address.**
- **Unknown networks for the routers are derived in the following table**

Device	Known Networks	Subnet Mask	Unknown Networks	Subnet Mask	Next-hop Address
Router0	192.168.10.0	255.255.255.224	192.168.10.96	255.255.255.224	192.168.10.66
Router0	192.168.10.32	255.255.255.224	192.168.10.128	255.255.255.224	192.168.10.66
Router0	192.168.10.64	255.255.255.224	-	-	-
Router1	192.168.10.96	255.255.255.224	192.168.10.0	255.255.255.224	192.168.10.65
Router1	192.168.10.128	255.255.255.224	192.168.10.32	255.255.255.224	192.168.10.65
Router1	192.168.10.64	255.255.255.224	-	-	-

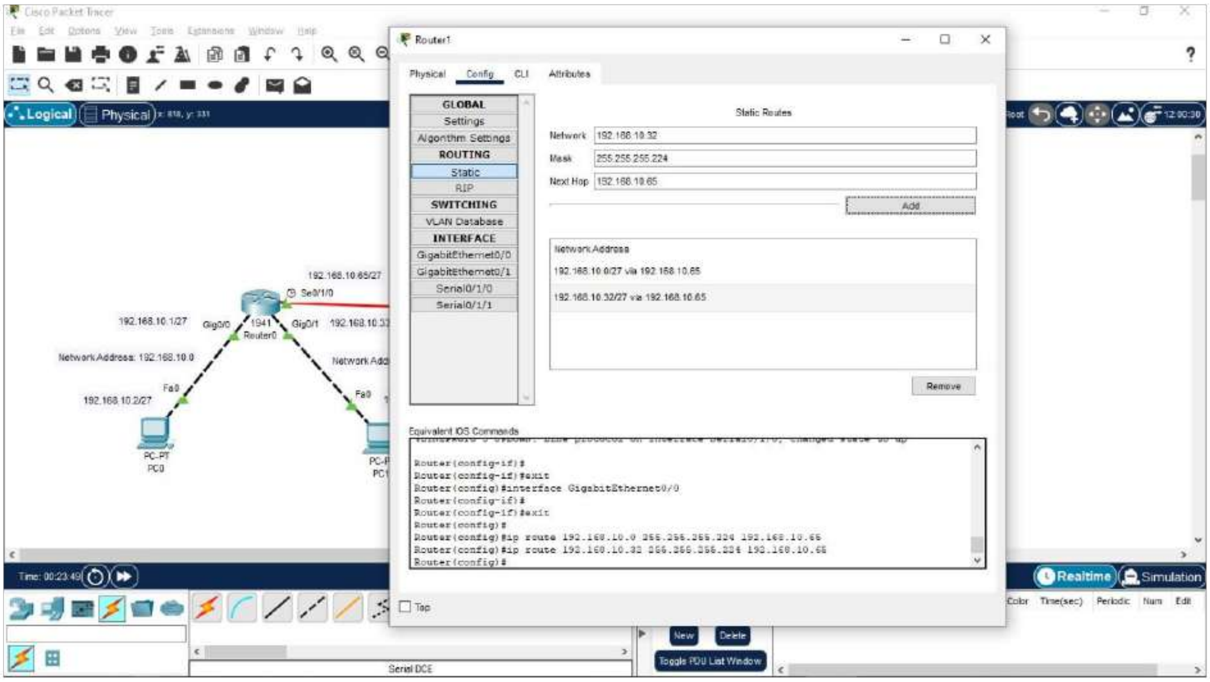


Step 8:

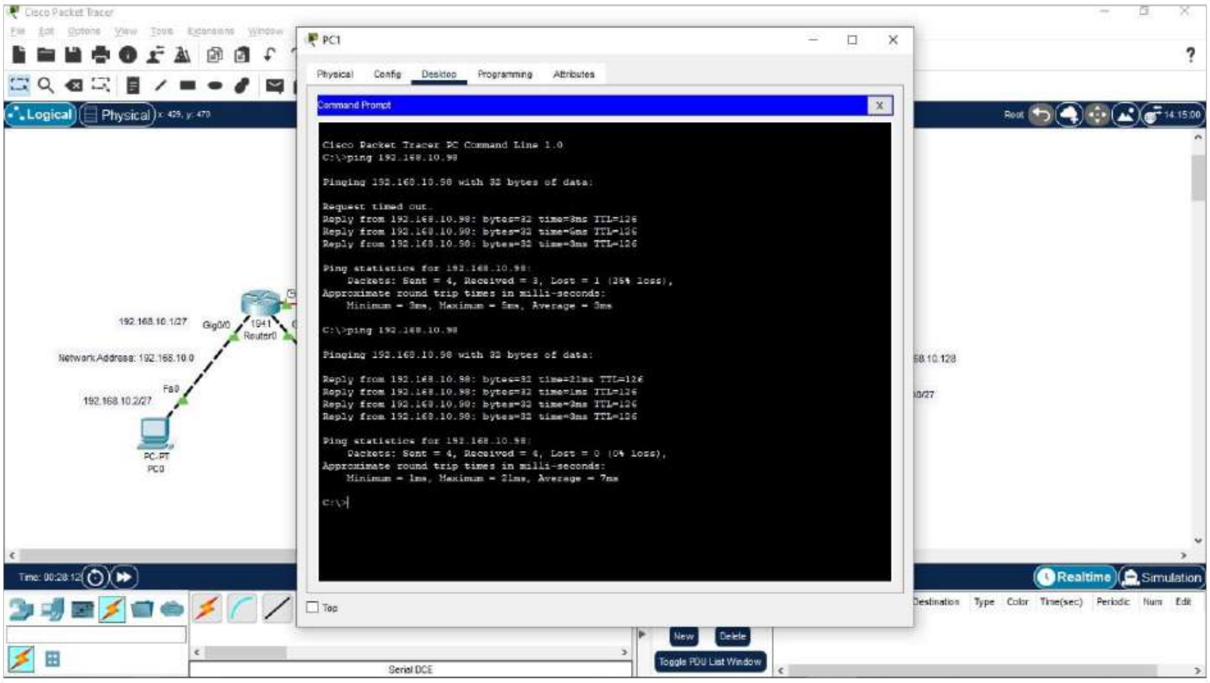
- To configure static routing, click on
  - Router0 => Config tab => Static =>Enter network address (refer unknown network for router0) => Subnet Mask for network address => Next-hop address => Add network
- Repeat the same to add the next network.
- Two networks should be added for the given scenario



Step 9: Repeat the same procedure to configure for Router1



Step 10: After configuration of Static routing in both Routers, Check the connectivity among any two devices using Ping Command



Step 11:

- To check routing table, go to CLI tab in Router and press enter to get the router prompt.
    - Router>
- Now type enable or en and press enter
    - Router>en
    - Router#

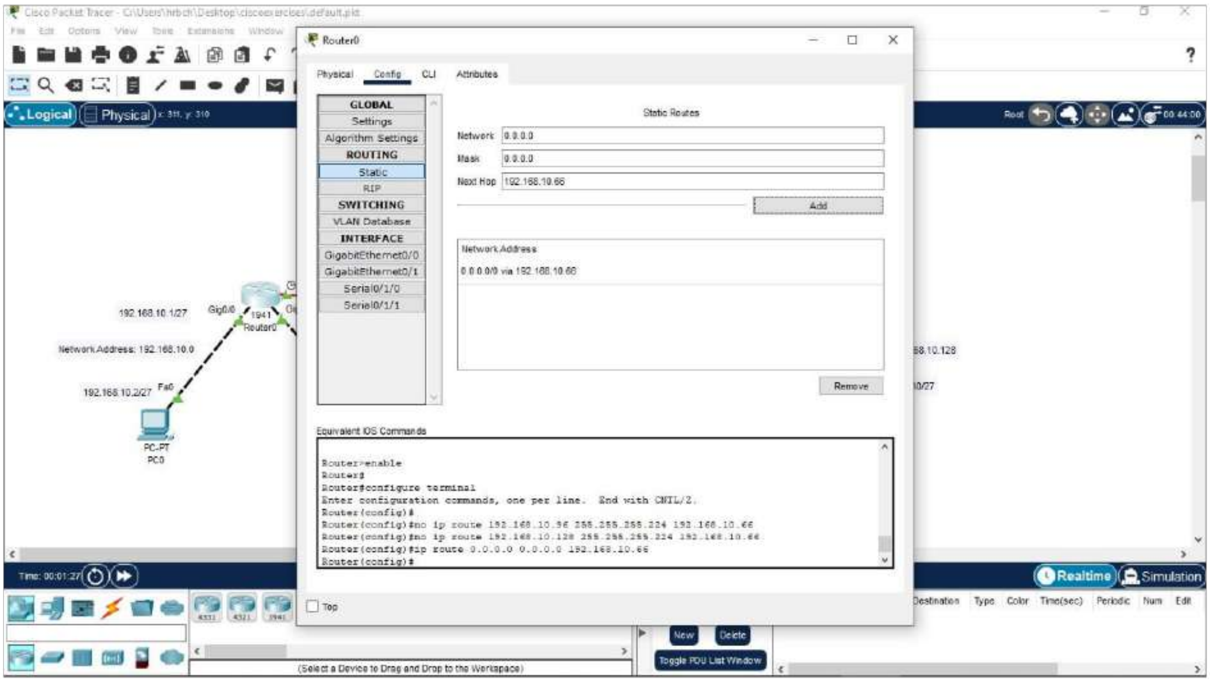
Follow the command “show ip route” to get the routing table of a router

```
Router>en
Router#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route

Gateway of last resort is not set

192.168.10.0/24 is variably subnetted, 8 subnets, 2 masks
C 192.168.10.0/27 is directly connected, GigabitEthernet0/0
L 192.168.10.1/32 is directly connected, GigabitEthernet0/0
C 192.168.10.32/27 is directly connected, GigabitEthernet0/1
L 192.168.10.33/32 is directly connected, GigabitEthernet0/1
C 192.168.10.64/27 is directly connected, Serial0/1/0
L 192.168.10.65/32 is directly connected, Serial0/1/0
S 192.168.10.96/27 [1/0] via 192.168.10.66
S 192.168.10.128/27 [1/0] via 192.168.10.66
```

5 b: For Default Routing replace Step 8 and Step 9 with Step 12 and Step 13

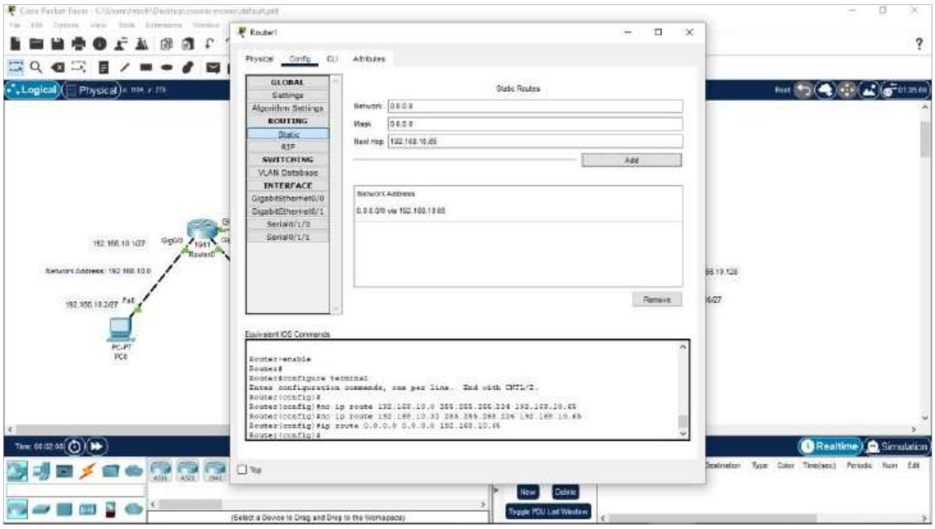
Step 12:

- To configure static routing, click on
    - Router0 => Config tab => Static =>Enter network address (as 0.0.0.0) => Subnet Mask (as 0.0.0.0) => Next-hop address (192.168.10.66) => Add network
- One entry should be added for the given scenario

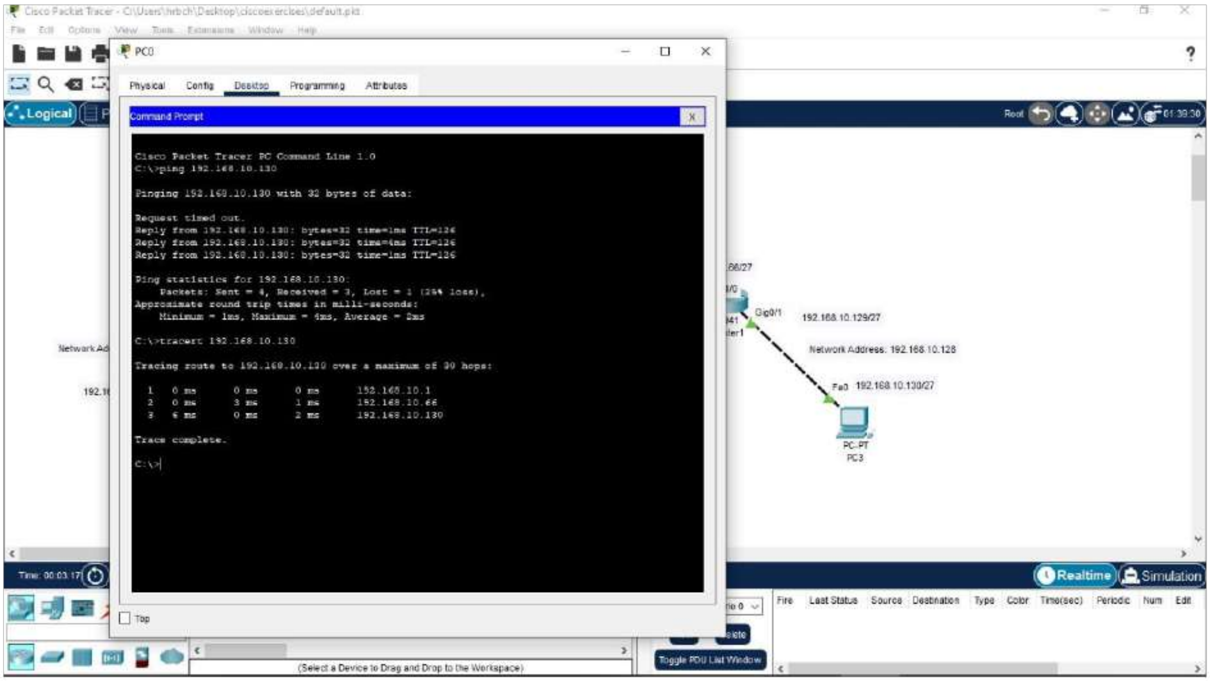


Step 13: Repeat the same procedure to configure for Router1.

Router1 => Config tab => Static =>Enter network address (as 0.0.0.0) => Subnet Mask (as 0.0.0.0) => Next-hop address (192.168.10.65) => Add network



Output for Default Routing: Click on PC0 => Desktop => Command Prompt and check **"tracert 192.168.10.130"** command



EXERCISE 6: NAT CONFIGURATION

AIM:

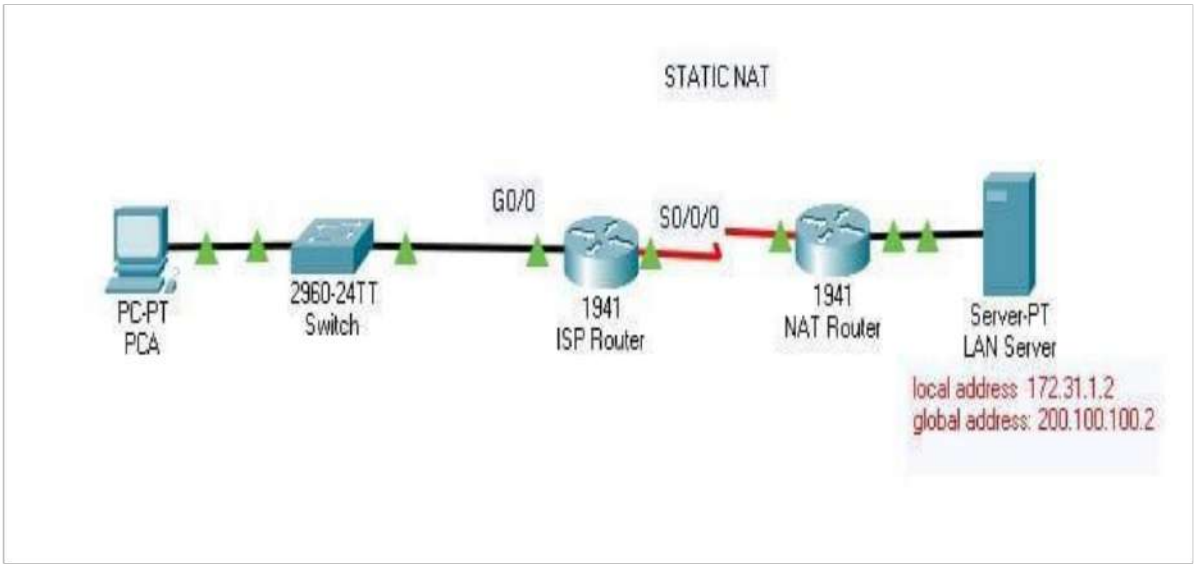
To configure the Network Address Translation (NAT) on a Cisco router.

a. Static NAT configuration

Steps to configure static NAT
Static NAT can be configured using the following two steps.
They are:
1. Creating a mapping between the private internal address and public global address using the `ip nat inside source static [private-address] [public-address]` global configuration command.

2. After the mapping is made, the interfaces taking part in the NAT translation are configured as either `inside` or `outside` with respect to NAT.

The router interface associated within the LAN is assigned the `inside` interface using the `ip nat inside` interface mode command.
Similarly, the router interface associated with the internet is assigned the outside interface using the `ip nat inside` interface mode command.



Static NAT topology

In the figure above, the Gigabit 0/0 `(g0/0)` interface is the `inside` interface because it is connected to the LAN. In contrast, the `S0/0/0` interface is configured as the `outside` interface because it is connected to the internet.

Configuring static NAT

To configure a static NAT between the private address `172.31.1.2` and public address `200.100.100.2`:

- Map the server's private address `172.31.1.2` to the public routable address `200.100.100.2` using the command `ip nat inside source static 172.31.1.2 200.100.100.2`.
- Enter the "interface serial `s0/0/0/`" command and identify the interface as the outside interface using the command `ip nat outside`.
- Enter the "interface gigabitethernet `g0/0`" command and identify it as the inside interface relative to NAT using the `ip nat inside` command.

```
Router>
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip nat inside source static 172.31.1.2 200.100.100.2
Router(config)#interface s0/0/0
Router(config-if)#ip nat outside
Router(config-if)#interface g0/0
Router(config-if)#ip nat inside
Router(config-if)#
Router(config-if)#
```

Static NAT configuration

b. Dynamic NAT configuration

Steps to configure dynamic NAT

Dynamic NAT still requires that both the inside and the outside interfaces be configured.

For allocation, it uses an access control list (ACL) to specify which private addresses are subject to translation and a NAT pool of registered IP addresses.

1. Create an ACL using the `access-list 1 permit address wildcard mask` command.

2. Create a NAT pool using the `ip nat pool [name] [first-address] [last-address] [netmask] [subnet mask]` global configuration command.

This pool will contain the public addresses for the translation. Because, ISP assigns the public addresses contiguous to the organizations.

The `first address` is the least in the given address range. And, the `last address` is the highest address of that range.

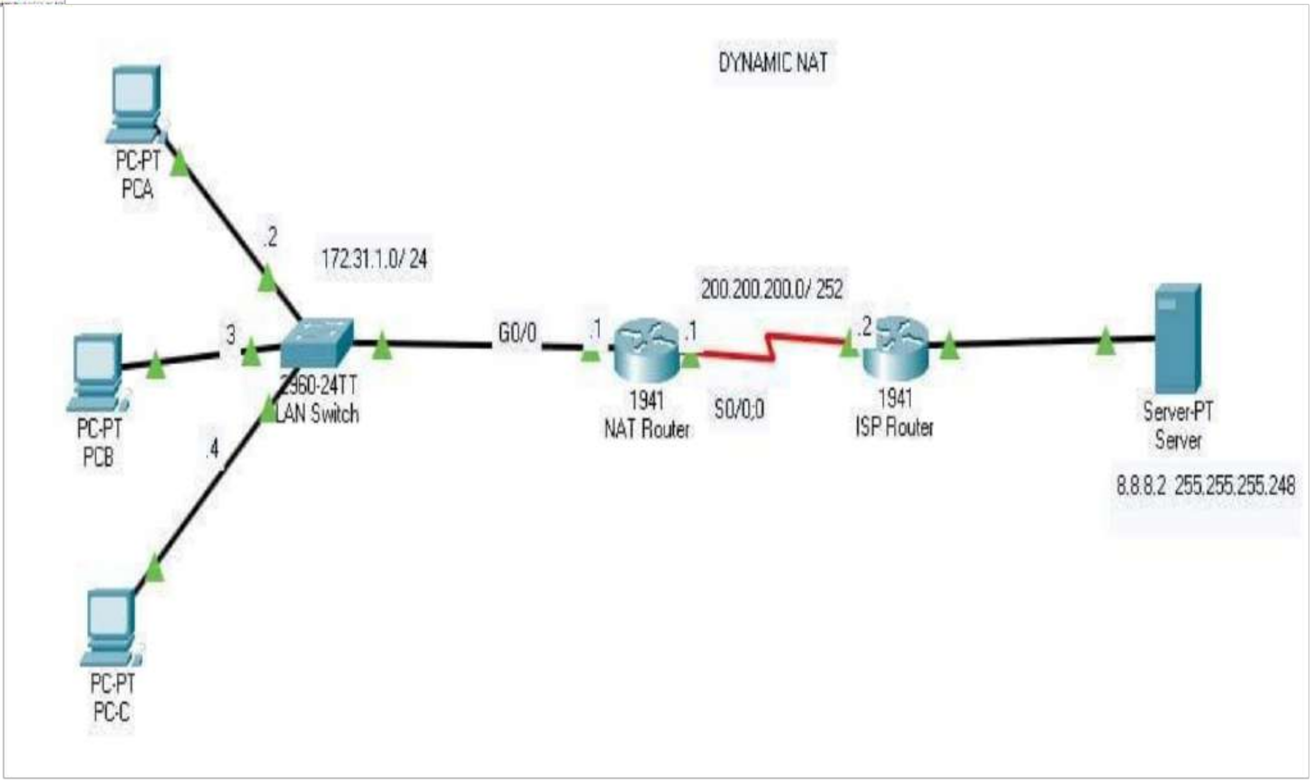
The netmask identifies the network to which of these addresses belong to, using the `ip nat inside source list [access-list] [number] pool [name]` command to bind the ACL and the NAT Pool created.
In this case, the ACL number is `1`, and the NAT POOL is `LAN`.

NOTE: Different ACL numbers and pool names can be created and used, but ACL `1` and pool name `LAN` will be used throughout this tutorial for simplicity.
1. Use the `ip nat inside` interface command to enable the `inside` interface for NAT translation

2. Use the `ip nat outside` interface command to enable the `outside` interface for NAT translation.

Configuring dynamic NAT

An organization is assigned with two public addresses: `200.100.100.1` and `200.100.100.2`. It wants to allow its internal hosts, in the private network `172.31.1.0` and `255.255.255.0` to reach the internet using dynamic NAT.



Dynamic NAT topology

To configure the dynamic NAT for the network topology above:

- Create an access list that will specify the private addresses that are allowed to be translated using the `access-list 1 permit 172.31.1.0 0.0.0.255`.
- Creates a pool that will contain the public addresses to be utilized for translation using the `ip nat pool LAN 200.100.100.1 200.100.100.1 netmask 255.255.255.0`.
- Bind the access list and the pool together using the `ip inside source list 1 pool LAN`. This allows for the dynamic translation of the private addresses and the public addresses in a NAT pool named `LAN`.
- Enter the `interface serial 0/0/0/` command and identify it as an outside interface using the `ip nat outside` command.
- Enter the `interface gigabitethernet g0/0` command and identify it as the `inside` interface using the `ip nat inside` command.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#access-list 1 permit 172.31.1.0 0.0.0.255
Router(config)#ip nat pool LAN 200.100.100.1 200.100.100.2 netmask 255.255.255.0
Router(config)#ip nat inside source list 1 pool LAN
Router(config)#interface s0/0/0
Router(config-if)#ip nat inside
Router(config-if)#interface g0/0
Router(config-if)#ip nat inside
Router(config-if)#
```

Result:

Thus, we have configured the Network Address Translation on a cisco router.

EXERCISE 7. A: DEMONSTRATION OF RIP V1

[Note: RIP v1 does not support classless addressing mode and RIP v2 must be used in Subnet enabled Scenario]

Objective: To demonstrate the configuration of IP Addressing with Subnetting in WAN Configuration

Pre-requisite: IP Address, Range of IP Address, Classes of IP Address, Subnetting

Components:

Devices	Required Nos
PCs	4
Copper cross-over Cables	4
Routers [1941]	2
Serial DCE	1

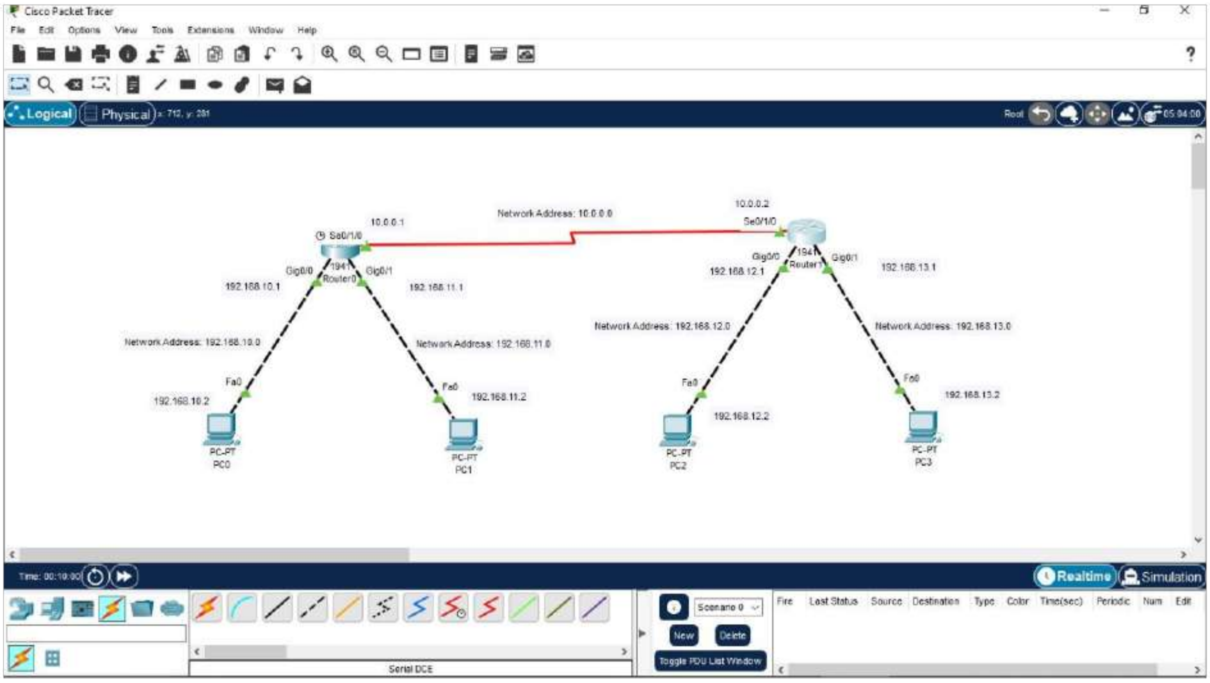
Addressing Table:

Device	Interface	IP Address	Subnet Mask	Gateway
PC0	Fa0/0	192.168.10.2	255.255.255.0	192.168.10.1
PC1	Fa0/0	192.168.11.2	255.255.255.0	192.168.11.1
PC2	Fa0/0	192.168.12.2	255.255.255.0	192.168.12.1
PC3	Fa0/0	192.168.13.2	255.255.255.0	192.168.13.1
Router0	Gigabit 0/0	192.168.10.1	255.255.255.0	-
Router0	Gigabit 0/1	192.168.11.1	255.255.255.0	-
Router0	Se0/1/0	10.0.0.1	255.0.0.0	-
Router1	Gigabit 0/0	192.168.12.1	255.255.255.0	-
Router1	Gigabit 0/1	192.168.13.1	255.255.255.0	-
Router1	Se0/1/0	10.0.0.2	255.0.0.0	-

Procedure:

Step 1:

- Drag 4 PCs and 2 routers in the console area as shown in the figure.
- Follow the procedure for connecting **Serial DCE cable from Exercise 4.**
- Assign IP addresses for 4 PCs (each 1 interface and corresponding router interface as gateway) and 2 Routers (each 3 ip addresses for 3 interfaces) as shown in the Addressing Table



Step 2:

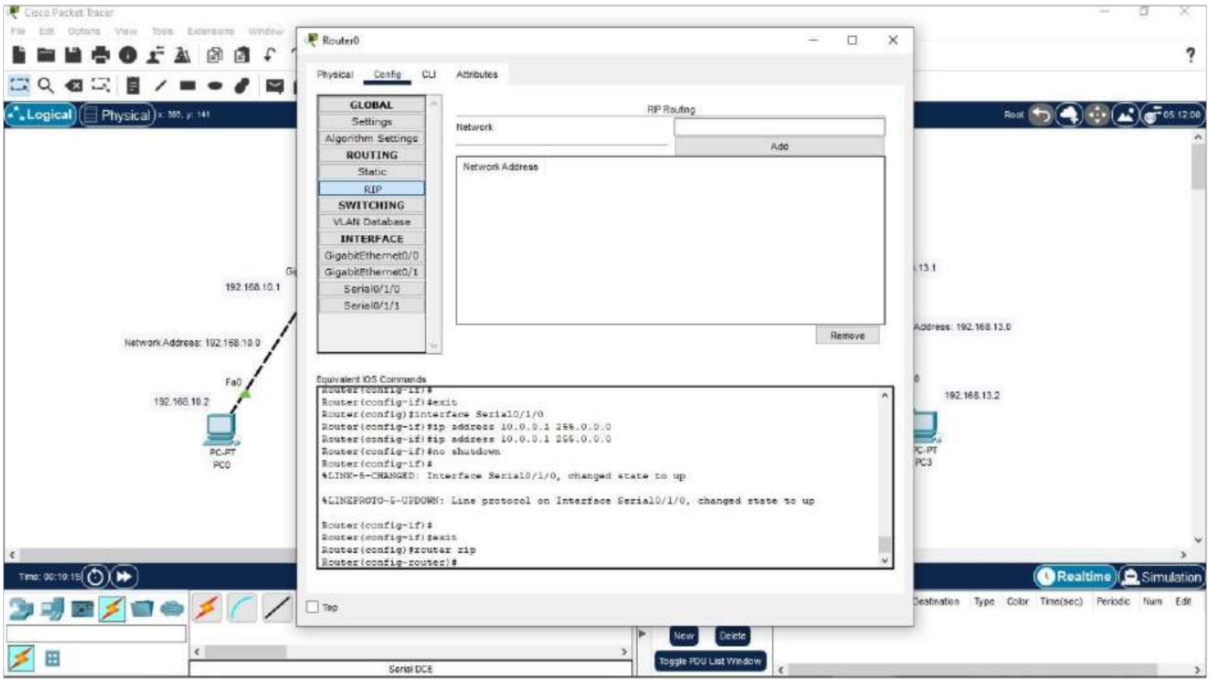
- To enable packet transmission among the devices in the scenario, RIP Routing has to be configured.
- To configure RIP routing, **known networks** for Router0 and Router1 has to be determined.
- Note: While specifying devices we should use IP-Address and while specifying network we should use Network Address.
- The **known networks** for the routers are derived in the following table

Device	Known Networks	Subnet Mask
Router0	192.168.10.0	255.255.255.0
Router0	192.168.11.0	255.255.255.0
Router0	10.0.0.0	255.0.0.0
Router1	192.168.12.0	255.255.255.0
Router1	192.168.13.0	255.255.255.0
Router1	10.0.0.0	255.0.0.0

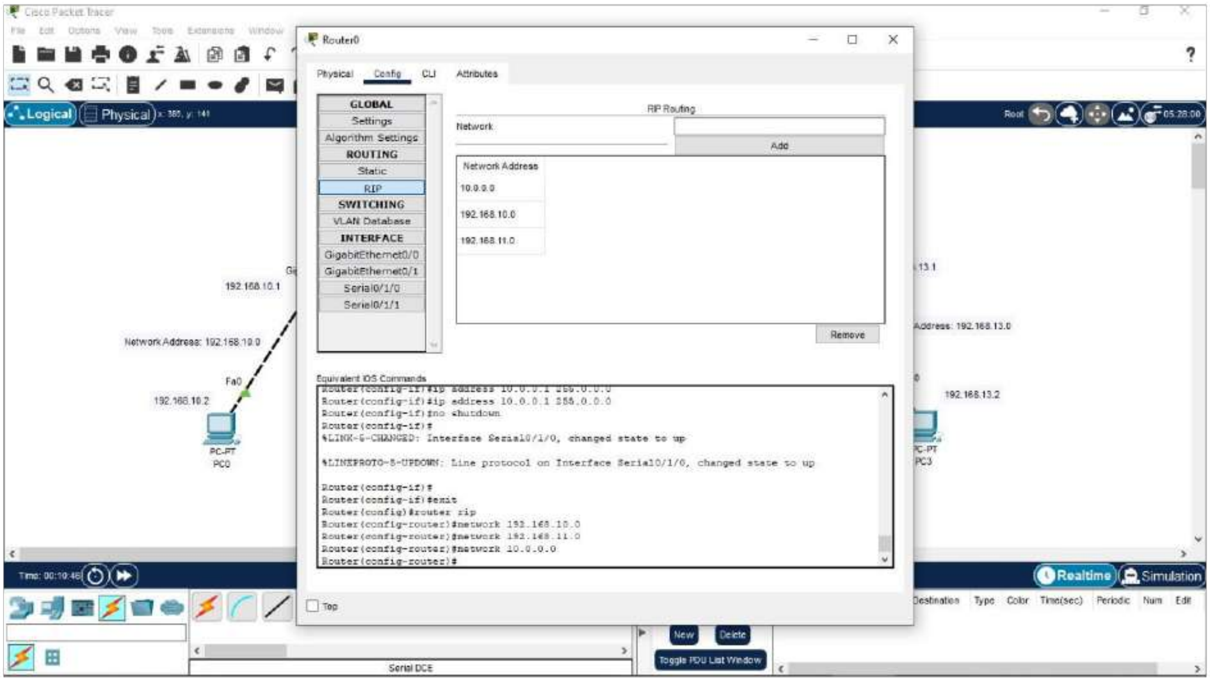
Only known networks should be configured for RIP routing

Step 3:

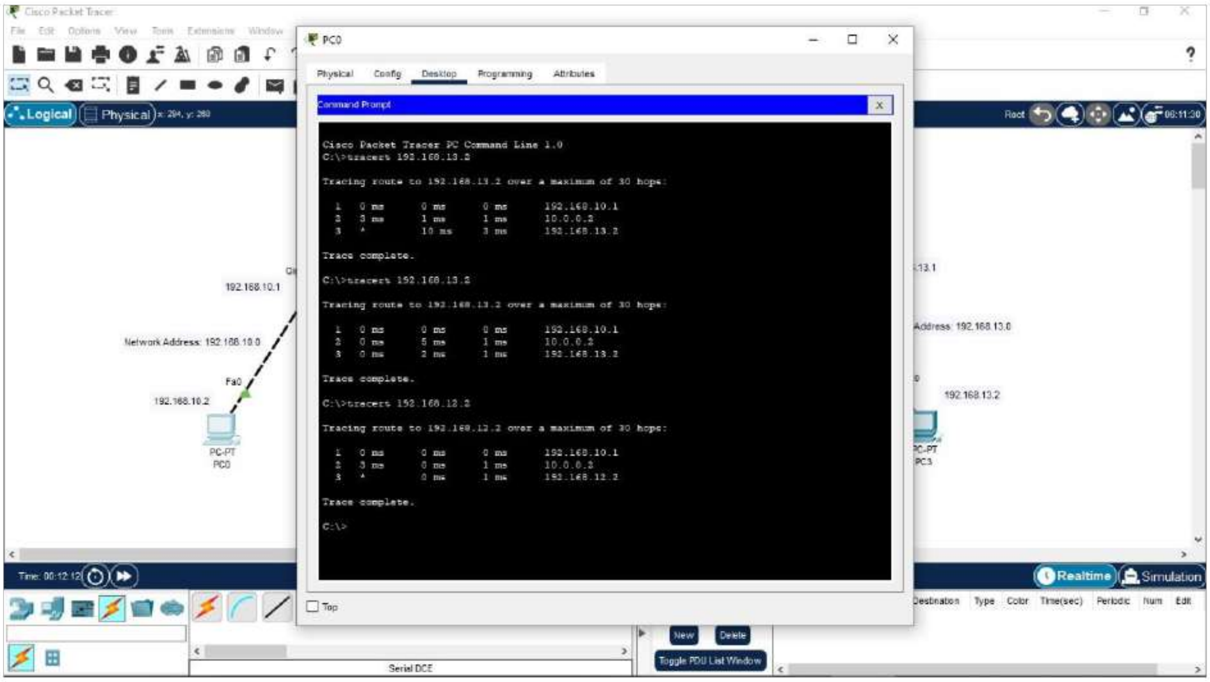
- To configure RIP routing, click on
    - Router0 => Config tab => RIP =>Enter network address (refer table) => Add network
- Repeat the same to add the next network.
- Three networks should be added for the given scenario



Step 4: Repeat the same procedure to configure for Router1

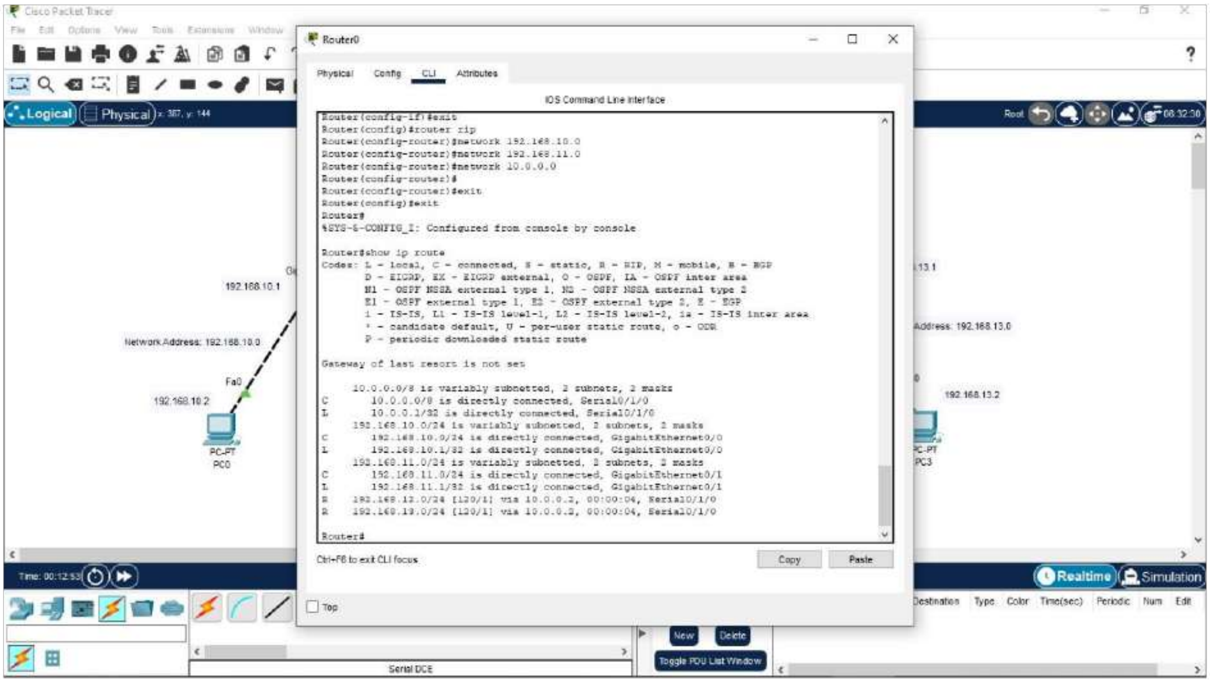


Step 5: After configuration of RIP routing in both Routers, Check the connectivity among any two devices using Ping Command or tracertcommand



Step 6:

- To check routing table, go to CLI tab in Router and press enter to get the router prompt.
  - Router>
- Now type enable or en and press enter
  - Router>en
  - Router# show ip route



EXERCISE 7. B: DEMONSTRATION OF RIP V2

Objective: To demonstrate the configuration of RIP v2

Pre-requisite: IP Address, Range of IP Address, Classes of IP Address, Subnetting

Components:

Devices	Required Nos
PCs	4
Copper cross-over Cables	4
Routers	2
Serial DCE	1

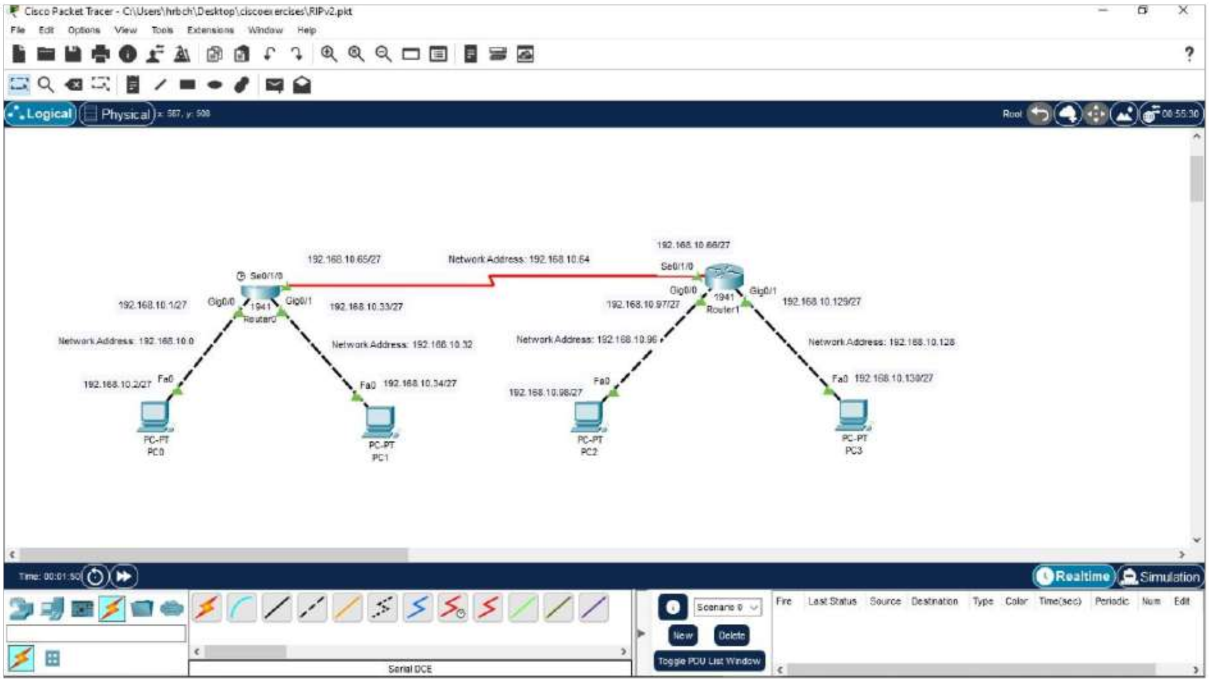
Addressing Table:

Device	Interface	IP Address	Subnet Mask	Gateway
PC0	Fa0/0	192.168.10.2	255.255.255.224	192.168.10.1
PC1	Fa0/0	192.168.10.34	255.255.255.224	192.168.10.33
PC2	Fa0/0	192.168.10.98	255.255.255.224	192.168.10.97
PC3	Fa0/0	192.168.10.130	255.255.255.224	192.168.10.129
Router0	Gigabit 0/0	192.168.10.1	255.255.255.224	-
Router0	Gigabit 0/1	192.168.10.33	255.255.255.224	-
Router0	Se0/1/0	192.168.10.65	255.255.255.224	-
Router1	Gigabit 0/0	192.168.10.97	255.255.255.224	-
Router1	Gigabit 0/1	192.168.10.129	255.255.255.224	-
Router1	Se0/1/0	192.168.10.66	255.255.255.224	-

Procedure:

Step 1:

- Drag 4 PCs and 2 routers in the console area as shown in the figure.
- Follow the procedure for connecting **Serial DCE cable from Exercise 4.**
- Assign IP addresses for 4 PCs (each 1 interface and corresponding router interface as gateway) and 2 Routers (each 3 ip addresses for 3 interfaces) as shown in the Addressing Table



Step 2:

- To enable packet transmission among the devices in the scenario, RIP v2 Routing has to be configured.
- To configure RIP v2 routing, **known networks** for Router0 and Router1 has to be determined.
- Note: While specifying devices we should use IP-Address and while specifying network we should use Network Address.
- The **known networks** for the routers are derived in the following table

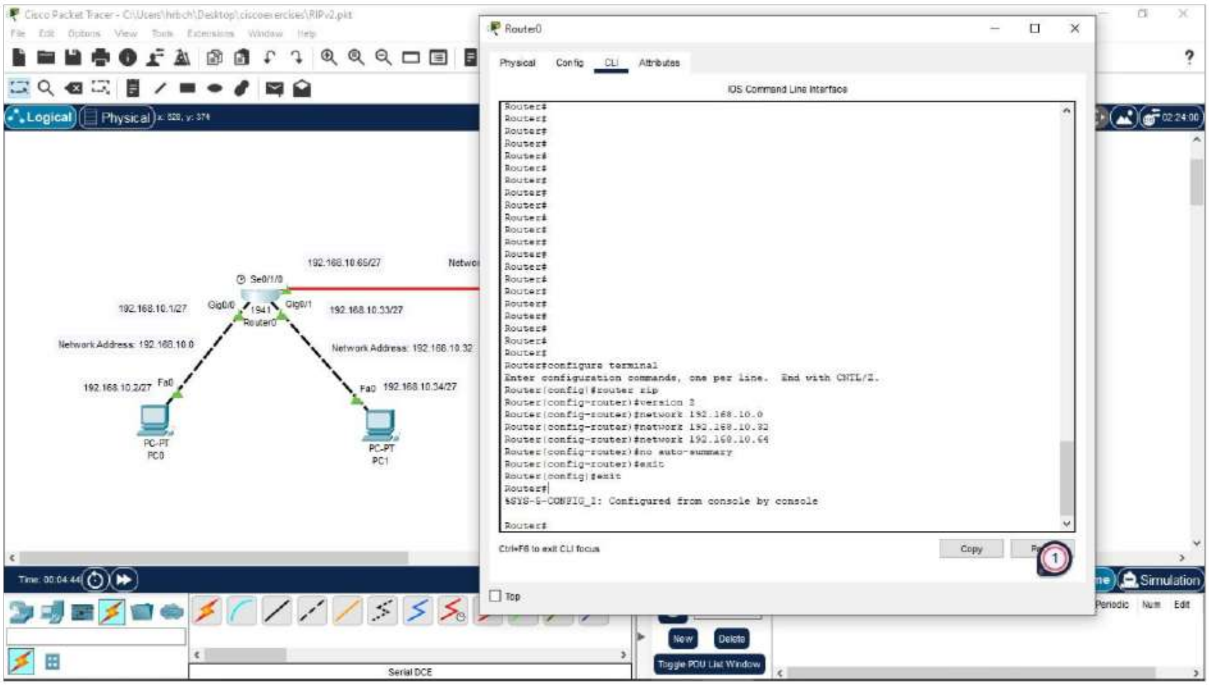
Device	Known Networks	Subnet Mask	Unknown Networks	Subnet Mask	Next-hop Address
Router0	192.168.10.0	255.255.255.224	192.168.10.96	255.255.255.224	192.168.10.66
Router0	192.168.10.32	255.255.255.224	192.168.10.128	255.255.255.224	192.168.10.66
Router0	192.168.10.64	255.255.255.224	-	-	-
Router1	192.168.10.96	255.255.255.224	192.168.10.0	255.255.255.224	192.168.10.65
Router1	192.168.10.128	255.255.255.224	192.168.10.32	255.255.255.224	192.168.10.65
Router1	192.168.10.64	255.255.255.224	-	-	-

Only unknown networks should be configured for static routing

Step 3:

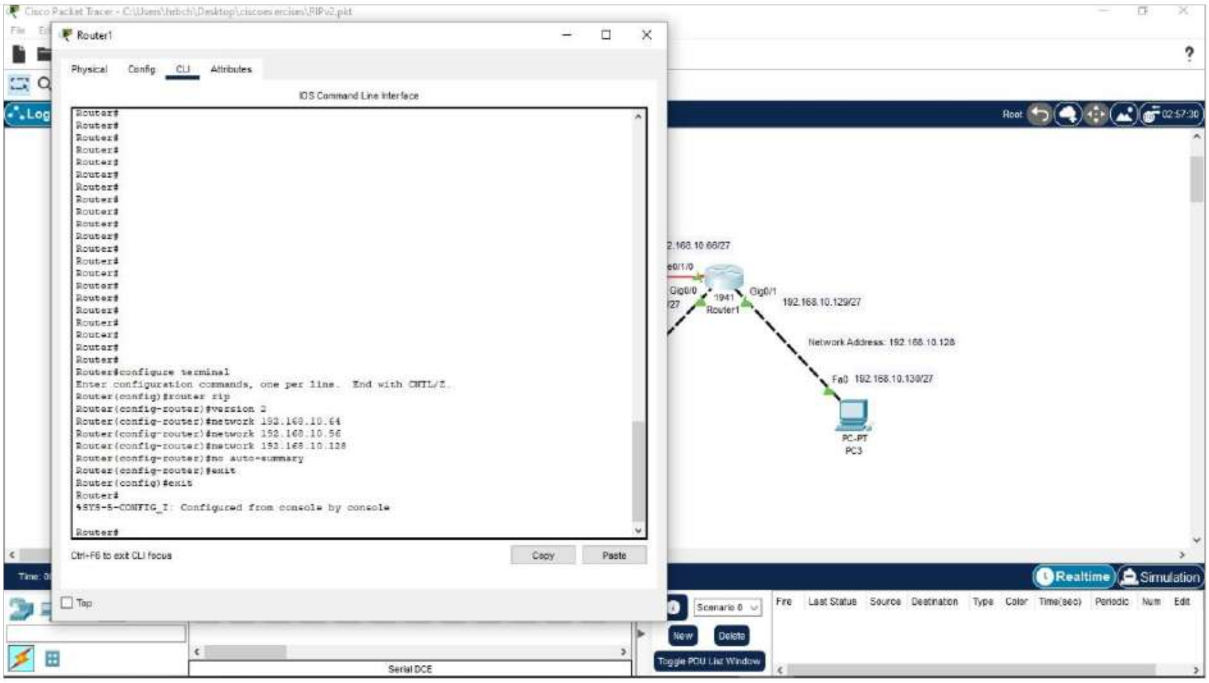
- To configure static routing, click on
  - Router0 => Config tab => CLI
- Type "enable" in "Router>" prompt and type the following commands to configure RIP v2 in Router0

```
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router rip
Router(config-router)#version 2
Router(config-router)#network 192.168.10.0
Router(config-router)#network 192.168.10.32
Router(config-router)#network 192.168.10.64
Router(config-router)#no auto-summary
Router(config-router)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#
```

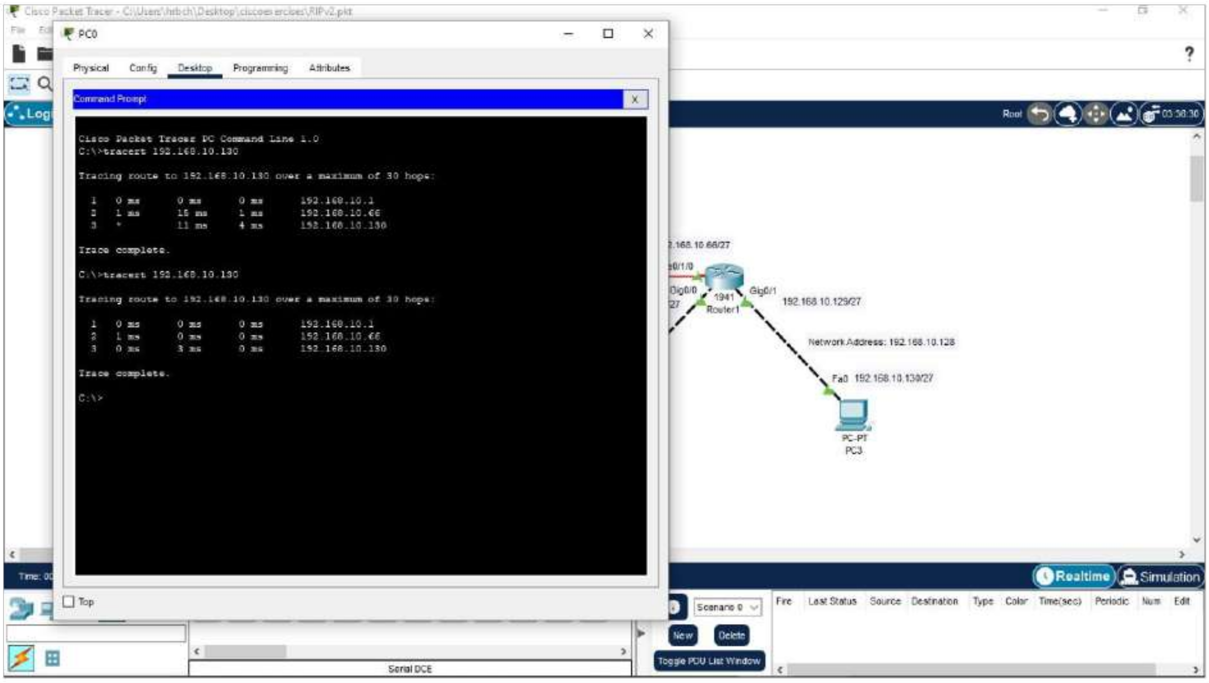


Step 4: Repeat the same procedure to configure for Router1

```
Router#
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router rip
Router(config-router)#version 2
Router(config-router)#network 192.168.10.64
Router(config-router)#network 192.168.10.96
Router(config-router)#network 192.168.10.128
Router(config-router)#no auto-summary
Router(config-router)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#
```



Step 5: After configuration of RIP v2 routing in both Routers, Check the connectivity among any two devices using Ping Command/ tracert command



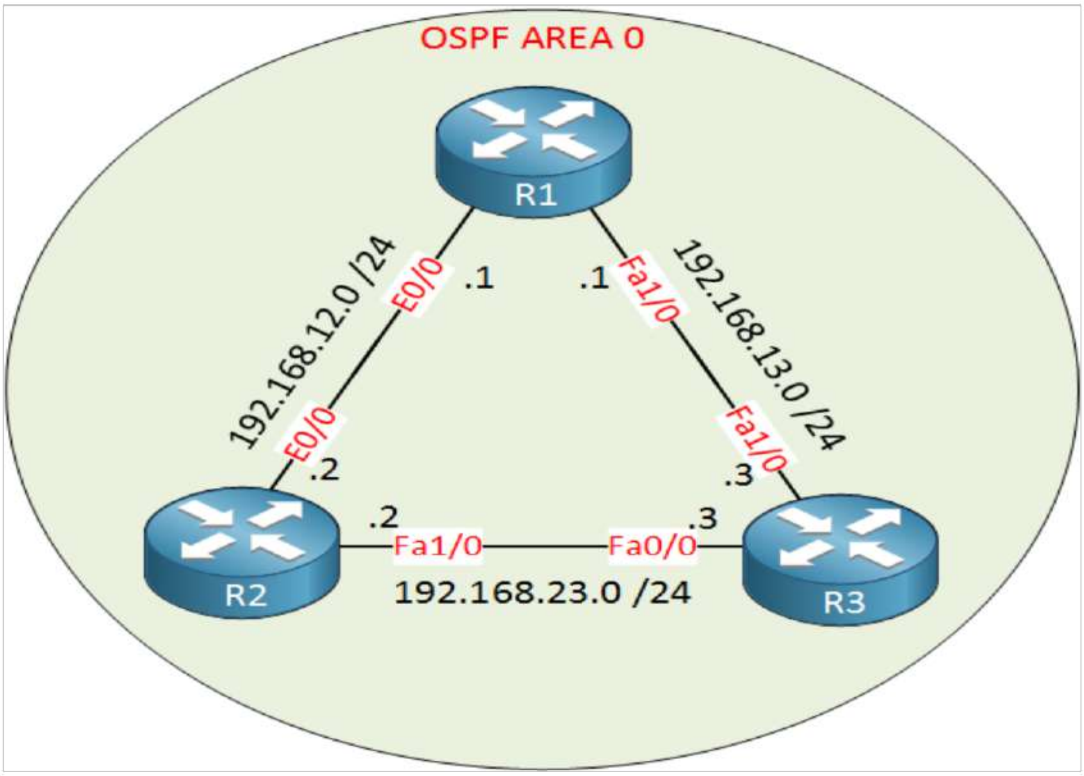
Result:

The demonstration of RIP v1 & RIP v2 is done successfully in the packet tracer.

EXERCISE 8: IMPLEMENTATION OF SINGLE AREA OSPF

Aim:

To configure the basic OSPF methodology using packet tracer.



This is the topology that we'll use. All routers are in OSPF Area 0. Note that the link between R2 and R1 is an Ethernet (10Mbit) link. All other links are FastEthernet (100Mbit) interfaces.

We'll start with the configuration between R2 and R3:

```
R2(config)#router ospf 1

R2(config-router)#network 192.168.23.0 0.0.0.255 area 0

R3(config)#router ospf 1

R3(config-router)#network 192.168.23.0 0.0.0.255 area 0
```

I need to use the router ospf command to get into the OSPF configuration. The number "1" is a process ID and you can choose any number you like.

The second step is to use the network command. It works similar to RIP but it is slightly different, let me break it down for you:

```
network 192.168.23.0 0.0.0.255
```

Just like RIP the network command does two things:

- Advertise the networks that fall within this range in OSPF.
- Activate OSPF on the interface(s) that fall within this range. This means that OSPF will send hello packets on the interface.

Behind 192.168.23.0 you can see it says 0.0.0.255. This is not a subnet mask but a **wildcard mask**. A wildcard mask is a **reverse subnet mask**. Let me give you an example:

Subnetmask	**255**	**255**	**255**	**0**
	11111111	11111111	11111111	00000000
Wildcardmask	**0**	**0**	**0**	**255**
	00000000	00000000	00000000	11111111

When I say reverse subnet mask I mean that the binary 1s and 0s of the wildcard mask are flipped compared to the subnet mask. A subnet mask of 255.255.255.0 is the same as wildcard mask 0.0.0.255. Don't worry about this too much for now as I'll explain wildcard masks to you when we talk about access-lists!

OSPF uses areas so you need to specify the area:

```
area 0
```

In our example we have configured single area OSPF. All routers belong to area 0.

After typing in my network command you'll see this message in the console:

```
R3# %OSPF-5-ADJCHG: Process 1, Nbr 192.168.23.2 on FastEthernet0/0 from
LOADING to FULL, Loading Done

R2# %OSPF-5-ADJCHG: Process 1, Nbr 192.168.23.3 on FastEthernet1/0 from
LOADING to FULL, Loading Done
```

Great! It seems that R3 and R2 have become neighbors. There's another command we can use to verify that we have become neighbors:

```
R3#show ip ospf neighbor
```

```
Neighbor ID Pri State Dead Time Address Interface

192.168.23.2 1 FULL/BDR 00:00:36 192.168.23.2 FastEthernet0/0

R2#show ip ospf neighbor



Neighbor ID Pri State Dead Time Address Interface

192.168.23.3 1 FULL/DR 00:00:32 192.168.23.3 FastEthernet1/0
```

Show ip ospf neighbor is a great command to see if your router has OSPF neighbors. When the state is full you know that the routers have successfully become neighbors.

Each OSPF router has a router ID and we check it with the show ip protocols command:

```
R2#show ip protocols

Routing Protocol is "ospf 1"
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Router ID 192.168.23.2
R3#show ip protocols
Routing Protocol is "ospf 1"
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Router ID 192.168.23.3
```

Above you see the router ID of R2 and R3. They used their highest active IP address as the router ID. Let's create a loopback on R2 to see if the router ID changes…

```
R2(config)#interface loopback 0

R2(config-if)#ip address 2.2.2.2 255.255.255.0
```

This is how you create a loopback interface. You can pick any number that you like it really doesn't matter.

```
R2#show ip protocols
Routing Protocol is "ospf 1"
```

```
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Router ID 192.168.23.2
```

The router ID still the same. We need to reset the OSPF process before the change will take effect, this is how you do it:

```
R2#clear ip ospf process

Reset ALL OSPF processes? [no]: yes
```

Use clear ip ospf process to reset OSPF. Let's see if there is a difference:

```
R2#show ip protocols
Routing Protocol is "ospf 1"
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Router ID 2.2.2.2
```

We can also change the router ID manually. Let me demonstrate this on R3:

```
R3#show ip protocols

Routing Protocol is "ospf 1"
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Router ID 192.168.23.3
```

Right now it's 192.168.23.3…

```
R3(config-router)#router-id 3.3.3.3
Reload or use "clear ip ospf process" command, for this to take effect
R3#clear ip ospf process
Reset ALL OSPF processes? [no]: yes
```

The router is friendly enough to warn me to reload or clear the OSPF process. Let's verify our configuration:

```
R3#show ip protocols
Routing Protocol is "ospf 1"
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Router ID 3.3.3.3
```

As you can see above the router ID is now 3.3.3.3.

Changing the router ID isn't something you would normally do. IP addresses on your router have to be unique so your OSPF router ID will also be unique. Understanding how OSPF selects a router ID is something you have to understand for the exam however.

Right now we have an OSPF neighbor adjacency between R2 and R3. Let's configure our routers so that R2/R1 and R1/R3 also become OSPF neighbors:

```
R2(config)#router ospf 1
R2(config-router)#network 192.168.12.0 0.0.0.255 area 0
R1(config)#router ospf 1
R1(config-router)#network 192.168.12.0 0.0.0.255 area 0

R1(config-router)#network 192.168.13.0 0.0.0.255 area 0

R3(config)#router ospf 1

R3(config-router)#network 192.168.13.0 0.0.0.255 area 0
```

I'll advertise all networks in OSPF. Before we check the routing table it's a good idea to see if our routers have become OSPF neighbors:

```
R2#show ip ospf neighbor

Neighbor ID    Pri   State      Dead Time    Address        Interface

192.168.13.1     1   FULL/BDR   00:00:31     192.168.12.1   Ethernet0/0

3.3.3.3          1   FULL/DR    00:00:38     192.168.23.3   FastEthernet1/0

R1#show ip ospf neighbor

Neighbor ID    Pri   State      Dead Time    Address        Interface

3.3.3.3          1   FULL/BDR   00:00:33     192.168.13.3   FastEthernet1/0

2.2.2.2          1   FULL/DR    00:00:30     192.168.12.2   Ethernet0/0
```

```
R3#show ip ospf neighbor

Neighbor ID    Pri   State      Dead Time    Address         Interface

192.168.13.1     1   FULL/DR    00:00:37     192.168.13.1    FastEthernet1/0

2.2.2.2          1   FULL/BDR   00:00:30     192.168.23.2    FastEthernet0/0
```

Excellent our routers have become OSPF neighbors and the state is full which means they are done exchanging information. Let's check the routing tables:

```
R2#show ip route ospf

O    192.168.13.0/24 [110/2] via 192.168.23.3, 00:09:45, FastEthernet1/0
```

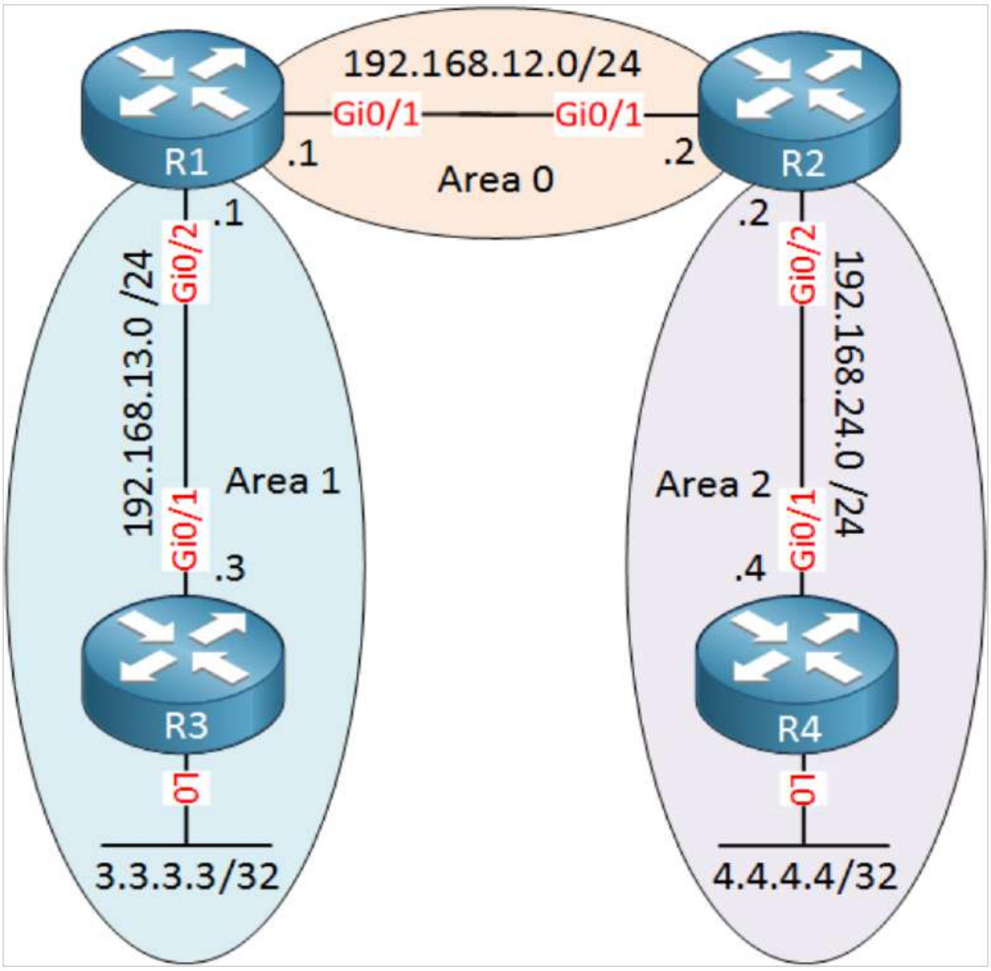
Result:

The implementation and verification of single area OSPF is done in packet tracer.

EXERCISE 9: IMPLEMENTATION OF MULTI AREA OSPF

Aim:

To configure the multi area OSPF topology using packet tracer.



Let's start with all network commands to get OSPF up and running. The network command defines to which area each interface will belong.First, we will configure R1 and R2 for the backbone area:

```
R1(config)#router ospf 1

R1(config-router)#network 192.168.12.0 0.0.0.255 area 0

R2(config)#router ospf 1

R2(config-router)#network 192.168.12.0 0.0.0.255 area 0
```

Let's configure R1 and R3 for area 1:

```
R1(config)#router ospf 1

R1(config-router)#network 192.168.13.0 0.0.0.255 area 1

R3(config)#router ospf 1
```

```
R3(config-router)#network 192.168.13.0 0.0.0.255 area 1

R3(config-router)#network 3.3.3.3 0.0.0.0 area 1
```

And last but not least, R2 and R4 for area 2:

```
R2(config)#router ospf 1

R2(config-router)#network 192.168.24.0 0.0.0.255 area 2

R4(config)#router ospf 1

R4(config-router)#network 192.168.24.0 0.0.0.255 area 2

R4(config-router)#network 4.4.4.4 0.0.0.0 area 2
```

Those are all the network commands we need.

Verification

Let's verify our work. First, let's make sure we have OSPF neighbors:

```
R1#show ip ospf neighbor



Neighbor ID     Pri   State           Dead Time   Address         Interface

192.168.24.2      1   FULL/DR         00:00:36    192.168.12.2
GigabitEthernet0/1

3.3.3.3           1   FULL/BDR        00:00:34    192.168.13.3
GigabitEthernet0/2
```

R1 has formed a neighbor adjacency with R2 and R3. Let's check R2:

```
R2#show ip ospf neighbor



Neighbor ID     Pri   State           Dead Time   Address         Interface
```

```
192.168.13.1      1   FULL/BDR        00:00:34    192.168.13.1
GigabitEthernet0/1

4.4.4.4           1   FULL/BDR        00:00:30    192.168.24.4
GigabitEthernet0/2
```

R2 has formed neighbor adjacencies with R1 and R4. The s**how ip** ospf **neighbor** command, however, doesn't tell me anything about the areas that are used. If you want to see this, you could add the **detail** parameter like this:

```
R2#show ip ospf neighbor detail

 Neighbor 192.168.13.1, interface address 192.168.12.1

    In the area 0 via interface GigabitEthernet0/1

    Neighbor priority is 1, State is FULL, 6 state changes

    DR is 192.168.12.2 BDR is 192.168.12.1

    Options is 0x12 in Hello (E-bit, L-bit)

    Options is 0x52 in DBD (E-bit, L-bit, O-bit)

    LLS Options is 0x1 (LR)

    Dead timer due in 00:00:33

    Neighbor is up for 00:17:30

    Index 1/1/1, retransmission queue length 0, number of retransmission 0

    First 0x0(0)/0x0(0)/0x0(0) Next 0x0(0)/0x0(0)/0x0(0)

    Last retransmission scan length is 0, maximum is 0

    Last retransmission scan time is 0 msec, maximum is 0 msec

 Neighbor 4.4.4.4, interface address 192.168.24.4

    In the area 2 via interface GigabitEthernet0/2

    Neighbor priority is 1, State is FULL, 6 state changes

    DR is 192.168.24.2 BDR is 192.168.24.4
```

```
    Options is 0x12 in Hello (E-bit, L-bit)

    Options is 0x52 in DBD (E-bit, L-bit, O-bit)

    LLS Options is 0x1 (LR)

    Dead timer due in 00:00:31

    Neighbor is up for 00:15:57

    Index 1/1/2, retransmission queue length 0, number of retransmission 0

    First 0x0(0)/0x0(0)/0x0(0) Next 0x0(0)/0x0(0)/0x0(0)

    Last retransmission scan length is 0, maximum is 0

    Last retransmission scan time is 0 msec, maximum is 0 msec
```

Above you can see that interface GigabitEthernet0/1 is in area 0 and interface GigabitEthernet0/2 is in area 2. Another good command to find area information is **show ip protocols**:

```
R2#show ip protocols

*** IP Routing is NSF aware ***



Routing Protocol is "application"

  Sending updates every 0 seconds

  Invalid after 0 seconds, hold down 0, flushed after 0

  Outgoing update filter list for all interfaces is not set

  Incoming update filter list for all interfaces is not set

  Maximum path: 32

  Routing for Networks:

  Routing Information Sources:

    Gateway         Distance      Last Update
```

```
  Distance: (default is 4)

Routing Protocol is "ospf 1"

  Outgoing update filter list for all interfaces is not set

  Incoming update filter list for all interfaces is not set

  Router ID 192.168.24.2

  It is an area border router

  Number of areas in this router is 2. 2 normal 0 stub 0 nssa

  Maximum path: 4

  Routing for Networks:

    192.168.12.0 0.0.0.255 area 0

    192.168.24.0 0.0.0.255 area 2

  Routing Information Sources:

    Gateway         Distance      Last Update

    4.4.4.4              110      00:16:04

    192.168.13.1         110      00:16:53

  Distance: (default is 110)
```

Above you can see which networks belong to which area:

- Network 192.168.12.0 in area 0.
- Network 192.168.24.0 in area 2.

Let's check our routing tables. Let's start with R1:

```
R1#show ip route ospf

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP

       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
```

```
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static
route
        o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
        a - application route
        + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set
      3.0.0.0/32 is subnetted, 1 subnets
O        3.3.3.3 [110/2] via 192.168.13.3, 00:01:47, GigabitEthernet0/2
      4.0.0.0/32 is subnetted, 1 subnets
O IA     4.4.4.4 [110/3] via 192.168.12.2, 00:00:54, GigabitEthernet0/1
O IA  192.168.24.0/24 [110/2] via 192.168.12.2, 00:01:44, GigabitEthernet0/1
```

Above we see three OSPF entries. The first one is for 3.3.3.3/32, the loopback interface of R3. It shows up with an O since this is an intra-area route. R1 has also learned about 4.4.4.4/32 and 192.168.24.0/24. These two entries show up as O IA since they are inter-area routes.

R2 has a similar output:

```
R2#show ip route ospf

      3.0.0.0/32 is subnetted, 1 subnets
O IA     3.3.3.3 [110/3] via 192.168.12.1, 00:02:19, GigabitEthernet0/1
      4.0.0.0/32 is subnetted, 1 subnets
O        4.4.4.4 [110/2] via 192.168.24.4, 00:01:29, GigabitEthernet0/2
```

```
O IA  192.168.13.0/24 [110/2] via 192.168.12.1, 00:02:24, GigabitEthernet0/1
```

Above we see that R2 has learned about 3.3.3.3/32 and 192.168.13.0/24 which area inter-area routes. 4.4.4.4/32 is an intra-area route.

Let's check R3:

```
R3#show ip route ospf

      4.0.0.0/32 is subnetted, 1 subnets

O IA     4.4.4.4 [110/4] via 192.168.13.1, 00:01:57, GigabitEthernet0/1

O IA  192.168.12.0/24 [110/2] via 192.168.13.1, 00:02:50, GigabitEthernet0/1

O IA  192.168.24.0/24 [110/3] via 192.168.13.1, 00:02:47, GigabitEthernet0/1
```

Everything that R3 has learned is from another area, that's why we only see inter-area routes here. The same thing applies to R4:

```
R4#show ip route ospf

      3.0.0.0/32 is subnetted, 1 subnets

O IA     3.3.3.3 [110/4] via 192.168.24.2, 00:02:13, GigabitEthernet0/1

O IA  192.168.12.0/24 [110/2] via 192.168.24.2, 00:02:13, GigabitEthernet0/1

O IA  192.168.13.0/24 [110/3] via 192.168.24.2, 00:02:13, GigabitEthernet0/1
```

Just to be sure, let's try a quick ping between R3 and R4 to prove that our multi-area OSPF configuration is working:

```
R3#ping 4.4.4.4 source 3.3.3.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:

Packet sent with a source address of 3.3.3.3

!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 9/11/13 ms
```

Our ping is successful. That will be all for now.

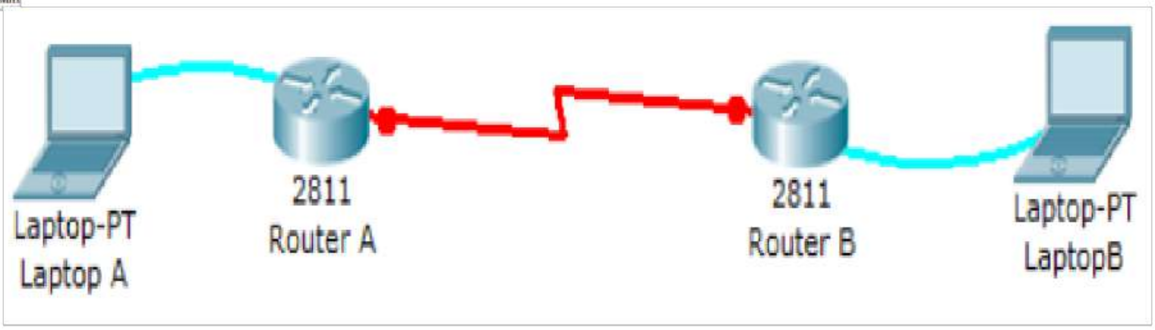
Result:

Thus, we have configured multiple OSPF areas and verified OSPF routes in the routing table that are from different areas.

EXERCISE 10-A: PPP CONFIGURATION

Objective:

To configure PPP on a serial link in Packet Tracer.



1. Use the connected laptops to find the DCE and DTE routers. You can connect to the routers using CLI.

2. Configure the routers with the following parameters :

- Clock : 250000
- PPP link between the routers
- DCE IP : 192.168.10.5/30

- DTE IP : 192.168.10.6/30

3. Check IP connectivity between the two routers using the ping command.

1. Use the connected laptops to find the DCE and DTE routers

The **show controllers <serial interface>** command is used to determine which side of the cable is the DCE side.

In this example, Router-A is the DTE side, and Router-B the DCE side (DCE V.35, clock rate set).

```
Router-A#show controllers serial 0/0/0
Interface Serial0/0/0
Hardware is PowerQUICC MPC860
DTE V.35 TX and RX clocks detected

Router-B#show controllers serial 0/0/0
Interface Serial0/0/0
Hardware is PowerQUICC MPC860
DCE V.35, clock rate 2000000
```

2. Configure the routers with the following parameters

Router-B beeing the DCE, clock rate has to be configured on Router-B serial 0/0/0 interface

```
Router-B(config)#interface serial 0/0/0
Router-B(config-if)#clock rate 250000
```

Then, configure PPP encapsulation and IP address on Router-B serial 0/0/0 interface.
The **encapsulation ppp** configures PPP protocol on the serial interface. PPP authenication can be oprtionnally configured using the following IOS commands which are not used in this lab :

- ppp authentication : Set PPP link authentication method
- ppp pap: Set PAP authentication parameters

Router-B beeing the DCE side of the serial link, the 192.168.10.5/30 IP address is configured on Router-B serial 0/0/0 interface. Don't forget to enable the interface with a **no shutdown** command.

```
Router-B(config)#interface serial 0/0/0
Router-B(config-if)#encapsulation ppp
Router-B(config-if)#ip address 192.168.10.5 255.255.255.252
Router-B(config-if)#no shutdown
```

The show interfaces serial 0/0/0 confirms that PPP encapsulation is enabled on the interface : *Encapsulation PPP, loopback not set, keepalive set (10 sec)*

```
Router-B#show interfaces serial 0/0/0
Serial0/0/0 is up, line protocol is up (connected)
 Hardware is HD64570
 Internet address is 192.168.10.5/30
 MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
   reliability 255/255, txload 1/255, rxload 1/255
 Encapsulation PPP, loopback not set, keepalive set (10 sec)
 Last input never, output never, output hang never
 [...]
```

Finally, configure PPP encapsulation and IP address on Router-A serial 0/0/0 interface. The link becomes up as both routers are correctly configured.

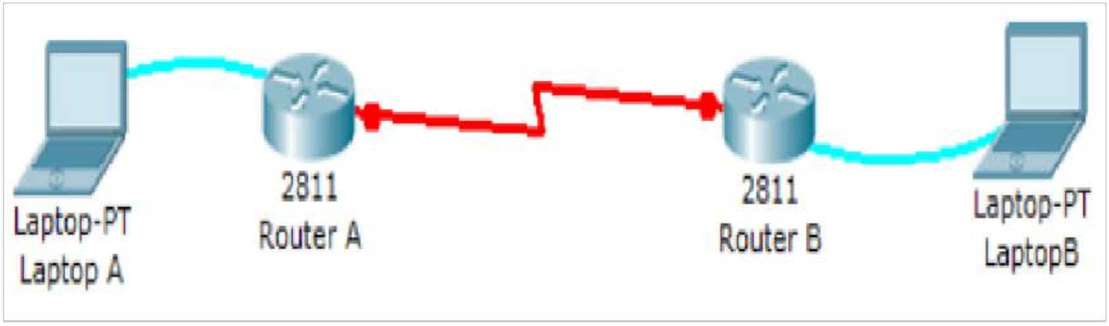
```
Router-A(config)#interface serial 0/0/0
Router-A(config-if)#encapsulation ppp
Router-A(config-if)#ip address 192.168.10.6 255.255.255.252
Router-A(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
```

3. Check IP connectivity between the two routers using the ping command.

Issue a ping from Router-A to Router-B to test network connectivity between the two routers.

```
Router-A# ping 192.168.10.5
```

EXERCISE 10-B: HDLC CONFIGURATION



1. Use the connected laptops to find the DCE and DTE routers

The show controllers <serial interface> command is used to determine which side of the cable is the DCE side.

In this example, Router-A is the DTE side, and Router-B the DCE side (DCE V.35, clock rate set).

```
Router-A#show controllers serial 0/0/0
Interface Serial0/0/0
Hardware is PowerQUICC MPC860
DTE V.35 TX and RX clocks detected

Router-B#show controllers serial 0/0/0
Interface Serial0/0/0
Hardware is PowerQUICC MPC860
DCE V.35, clock rate 2000000
```

2. Configure the routers with the following parameters

Router-B beeing the DCE, clock rate has to be configured on Router-B serial 0/0/0 interface

```
Router-B(config)#interface serial 0/0/0
Router-B(config-if)#clock rate 250000
```

Then, configure HDLC encapsulation and IP address on Router-B serial 0/0/0 interface.
The **encapsulation hdlc** configures HDLC protocol on the serial interface.

Router-B beeing the DCE side of the serial link, the 192.168.1.5/30 IP address is configured on Router-B serial 0/0/0 interface. Don't forget to enable the interface with a **no shutdown** command.

```
Router-B(config)#interface serial 0/0/0
Router-B(config-if)#encapsulation hdlc
Router-B(config-if)#ip address 192.168.10.5 255.255.255.252
Router-B(config-if)#no shutdown
```

The show interfaces serial 0/0/0 confirms that HDLC encapsulation is enabled on the interface : *Encapsulation HDLC, loopback not set, keepalive set (10 sec)*

```
Router-B#show interfaces serial 0/0/0
Serial0/0/0 is up, line protocol is up (connected)
```

```
 Hardware is HD64570
 Internet address is 192.168.10.5/30
 MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
   reliability 255/255, txload 1/255, rxload 1/255
 Encapsulation HDLC, loopback not set, keepalive set (10 sec)
 Last input never, output never, output hang never
 [...]
```

Finally, configure HDLC encapsulation and IP address on Router-A serial 0/0/0 interface. The link becomes up as both routers are correctly configured.

```
Router-A(config)#interface serial 0/0/0
Router-A(config-if)#encapsulation hdlc
Router-A(config-if)#ip address 192.168.10.6 255.255.255.252
Router-A(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up
```

3. Check IP connectivity between the two routers using the ping command.

Issue a ping from Router-A to Router-B to test network connectivity between the two routers.

```
Router-A#ping 192.168.10.5
```

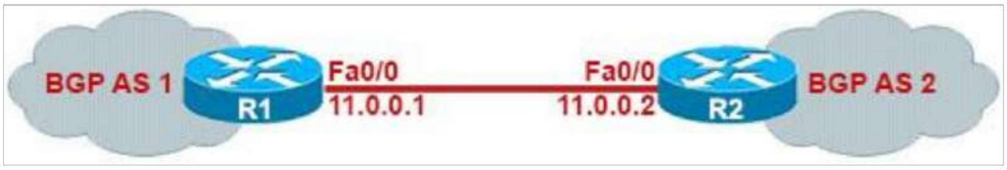
Result:

Thus, we have configured PPP and HDLC protocols and verified in the packet tracer.

EXERCISE 11-A: BASIC BGP CONFIGURATION

Aim:

To configure BGP routers with different Autonomous System numbers configuration between two routers.



First we need to configure some interfaces on two routers as follows:

R1(config)#interface fastethernet0/0 R1(config-if)#ip address 11.0.0.1 255.255.255.0 R1(config-if)#no shutdown R1(config-if)#interface loopback 0 R1(config-if)#ip address 1.1.1.1 255.255.255.0	R2(config)#interface fastethernet0/0 R2(config-if)#ip address 11.0.0.2 255.255.255.0 R2(config-if)#no shutdown R2(config-if)#interface loopback 0 R2(config-if)#ip address 2.2.2.2 255.255.255.0

So we have just configured interface fa0/0 and loopback0 on both routers. Next we will configure the BGP configuration part on R1:

R1(config)#**router bgp 1**
R1(config-router)#**neighbor 11.0.0.2 remote-as 2**

The configuration is very simple with only two lines on R1. In the first line, BGP configuration begins with a familiar type of command: the **router bgp** command, where **AS number** is the BGP AS number used by that router (same as EIGRP, OSPF configuration).

The next command defines the IP address of the neighbor. Unlike OSPF or EIGRP, BGP cannot discover its neighbors automatically so we have to explicitly declare them. We also have to know and declare the neighbor's BGP AS number as well. In this case R1 wants to establish BGP neighbor relationship with R2 (in BGP AS 2) so it choose an interface on R2 (Fa0/0: **11.0.0.2**) and specify R2 is in **BGP AS 2** via the command "neighbor **11.0.0.2** remote-as **2**". At the other end R2 will do the same thing for R1 to set up BGP neighbor relationship.

R2(config)#**router bgp 2**
R2(config-router)#**neighbor 11.0.0.1 remote-as 1**

After a moment we should see a message (on each router) similar to the following, letting us know that an adjacency has been formed:

On R1:

```
*Aug 17 00:09:38.453: %BGP-5-ADJCHANGE: neighbor 11.0.0.2 Up
```

On R2:

```
*Aug 17 00:09:38.453: %BGP-5-ADJCHANGE: neighbor 11.0.0.1 Up
```

So after forming BGP neighbor relationship we can verify by using the "show ip bgp summary" command on both routers:

```
R1#show ip bgp summary
BGP router identifier 1.1.1.1, local AS number 1
BGP table version is 1, main routing table version 1

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
11.0.0.2        4    2      19      19        1    0    0 00:16:21        0
R2#show ip bgp summary
BGP router identifier 2.2.2.2, local AS number 2
BGP table version is 1, main routing table version 1

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
11.0.0.1        4    1      20      20        1    0    0 00:17:13        0
```

Let's try advertising the loopback 0 interface on R1 to R2:

```
R1(config-router)#network 1.1.1.0 mask 255.255.255.0
```

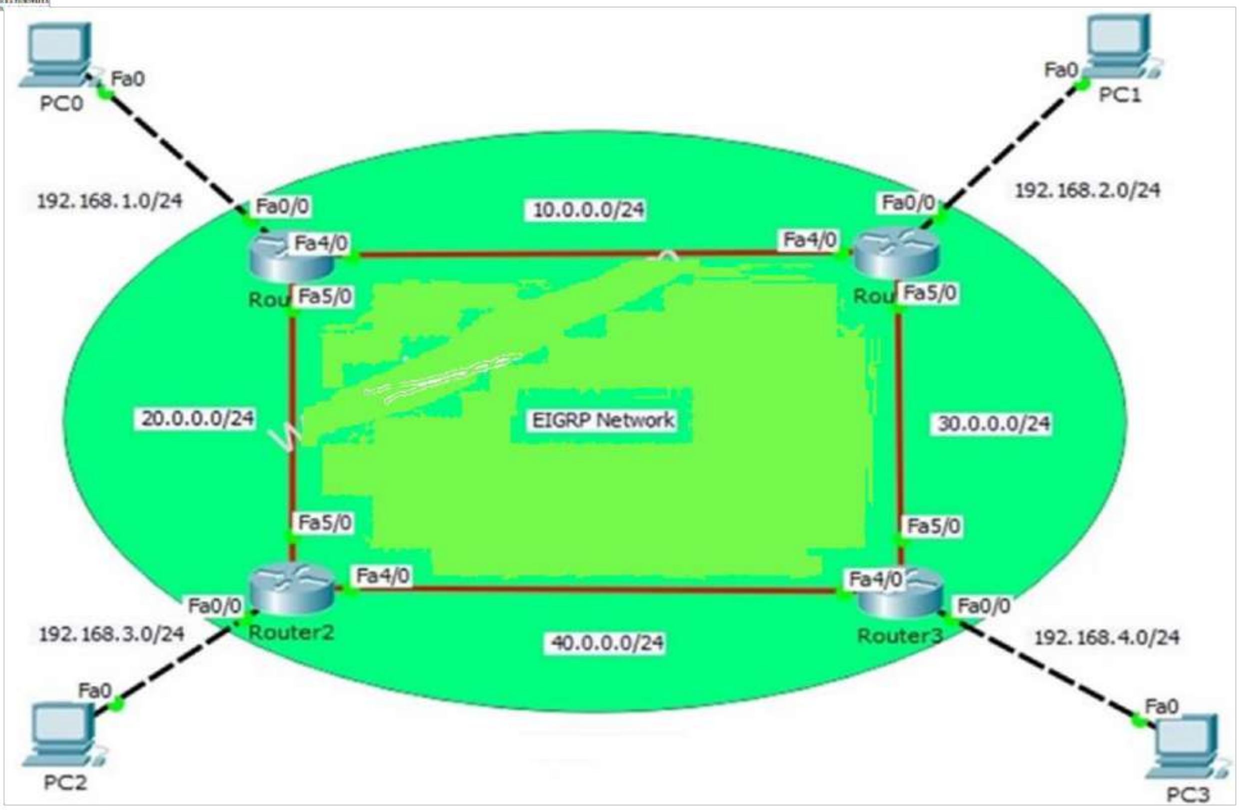
Now the BGP routing tables on these two routers contain this route:

```
R1#sh ip bgp
BGP table version is 4, local router ID is 11.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 1.1.1.0/24       0.0.0.0                  0         32768 i
```

EXERCISE 12: EIGRP CONFIGURATION

Objective:

To configure and verify the EIGRP routing in packet tracer.



First, Assign the IP addresses for PC's as given below.

PC0: 192.168.1.1 255.255.255.0 GW:192.168.1.2

PC1: 192.168.2.1 255.255.255.0 GW:192.168.2.2

PC2: 192.168.3.1 255.255.255.0 GW:192.168.3.2

PC3: 192.168.4.1 255.255.255.0 GW:192.168.4.2

EIGRP Configuration

```
Router0(config)# interface FastEthernet0/0

Router0(config-if)# ip address 192.168.1.2 255.255.255.0

Router0(config-if)# no shutdown

Router0(config-if)# exit

Router0(config)# interface FastEthernet4/0

Router0(config-if)# ip address 10.0.0.1 255.255.255.0

Router0(config-if)# no shutdown

Router0(config-if)# exit

Router0(config)# interface FastEthernet5/0
```

```
Router0(config-if)# ip address 20.0.0.1 255.255.255.0

Router0(config-if)# no shutdown

Router0(config-if)# end

Router0# copy running-config startup-config
```

```
Router1(config)# interface FastEthernet0/0

Router1(config-if)# ip address 192.168.2.2 255.255.255.0

Router1(config-if)# no shutdown

Router1(config-if)# exit

Router1(config)# interface FastEthernet4/0

Router1(config-if)# ip address 10.0.0.2 255.255.255.0

Router1(config-if)# no shutdown

Router1(config-if)# exit

Router1(config)# interface FastEthernet5/0

Router1(config-if)# ip address 30.0.0.1 255.255.255.0

Router1(config-if)# no shutdown

Router1(config-if)# end

Router1# copy running-config startup-config
```

```
Router2(config)# interface FastEthernet0/0

Router2(config-if)# ip address 192.168.3.2 255.255.255.0

Router2(config-if)# no shutdown

Router2(config-if)# exit

Router2(config)# interface FastEthernet4/0

Router2(config-if)# ip address 40.0.0.1 255.255.255.0

Router2(config-if)# no shutdown
```

```
Router2(config-if)# exit

Router2(config)# interface FastEthernet5/0

Router2(config-if)# ip address 20.0.0.2 255.255.255.0

Router2(config-if)# no shutdown

Router2(config-if)# end

Router2# copy running-config startup-config
```

```
Router3(config)# interface FastEthernet0/0

Router3(config-if)# ip address 192.168.4.2 255.255.255.0

Router3(config-if)# no shutdown

Router3(config-if)# exit

Router3(config)# interface FastEthernet4/0

Router3(config-if)# ip address 40.0.0.2 255.255.255.0

Router3(config-if)# no shutdown

Router3(config-if)# exit

Router3(config)# interface FastEthernet5/0

Router3(config-if)# ip address 30.0.0.2 255.255.255.0

Router3(config-if)# no shutdown

Router3(config-if)# end

Router3# copy running-config startup-config
```

Here, our EIGRP Autonomous number will be **100**. And for Router0, **192.168.1.0, 10.0.0.0, 20.0.0.0** network will be added under this EIGRP process. These are the directly connected networks to Router0.

```
Router0(config)# router eigrp 100

Router0(config-router)# network 192.168.1.0

Router0(config-router)# network 10.0.0.0

Router0(config-router)# network 20.0.0.0
```

```
Router0(config-router)# no auto-summary

Router0(config-router)# end

Router0# copy running-config startup-config
```

The similar configuration will be done on Router1 also. Here, only the networks will be changed. The direclty connected networks to Router1 will be added.

```
Router1(config)# router eigrp 100

Router1(config-router)# network 192.168.2.0

Router1(config-router)# network 10.0.0.0

Router1(config-router)# network 30.0.0.0

Router1(config-router)# no auto-summary

Router1(config-router)# end

Router1# copy running-config startup-config
```

For Router2 and Router3 the similar **Cisco EIGRP Configuration** will be done. Only one added networks will be changed. Because each router has different directly connected networks.

```
Router2(config)# router eigrp 100

Router2(config-router)# network 192.168.3.0

Router2(config-router)# network 20.0.0.0

Router2(config-router)# network 40.0.0.0

Router2(config-router)# no auto-summary

Router2(config-router)# end

Router2# copy running-config startup-config
```

```
Router3(config)# router eigrp 100

Router3(config-router)# network 192.168.4.0

Router3(config-router)# network 30.0.0.0
```

```
Router3(config-router)# network 40.0.0.0

Router3(config-router)# no auto-summary

Router3(config-router)# end

Router3# copy running-config startup-config
```

Configuration Verification

```
Router0# show ip eigrp neighbors
IP-EIGRP neighbors for process 100
H Address Interface Hold Uptime SRTT RTO Q Seq
(sec) (ms) Cnt Num
0 10.0.0.2 Fa4/0 14 00:14:53 40 1000 0 12
1 20.0.0.2 Fa5/0 12 00:14:53 40 1000 0 11
```

```
Router1# show ip eigrp interfaces
IP-EIGRP interfaces for process 100
Xmit Queue Mean Pacing Time Multicast Pending
Interface Peers Un/Reliable SRTT Un/Reliable Flow Timer Routes
Fa4/0 1 0/0 1236 0/10 0 0
Fa5/0 1 0/0 1236 0/10 0 0
Fa0/0 0 0/0 1236 0/10 0 0
```

```
Router1# show ip eigrp topology

IP-EIGRP Topology Table for AS 100/ID(192.168.2.2)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
r - Reply status
```

Result:

Thus, the configuration and verification of EIGRP is successfully done.

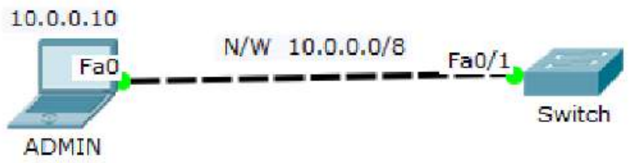
Exercise 13: Configuring Telnet on a switch and a router in Packet Tracer

Objective:

To configure Telnet on a switch and a router in Packet Tracer.

A. Telnet configuration on a switch

1. Create the network topology below in Packet Tracer.



Assign the laptop a static IP address of 10.0.0.10.

The topology above consists of an **ADMIN** laptop and a remote switch. We'll configure Telnet on the switch so that as the admin, you'll be able to access and manage the switch remotely.

2. Configure **enable password** or **enable secret password** on the switch. If you fail to do this,you won't get past the executive mode of the switch even after you establish a telnet connection to the switch.

```
Switch>enable

Switch#config terminal

Switch(config)#enable password admin
```

3.**Configure a VLAN interface on the switch**

We assign an IP address to the **VLAN** interface of the switch so that we can Telnet the switch from the laptop using this address.

```
Switch(config)#int VLAN 1

Switch(config-if)#ip address 10.0.0.20   255.0.0.0

Switch(config-if)#no shut

Switch(config-if)#exit
```

4.Configure a **Telnet password** for remote access.

This password is configured on **VTY** lines. VTY means Virtual Terminal. Before you can manage the switch remotely via Telnet, you'll have to provide this password.

```
Switch(config)#line vty 0 15

Switch(config-line)#password cisco

Switch(config-line)#login
```

Telnet access to the switch is allowed through VTY lines. We can establish up to 16 telnet connections to to the switch at the same time. Tha'ts what '0 15' means.

Next,

5. Test Telnet connectivity.

Go to command prompt of the laptop and type *telnet 10.0.0.20*

Hope you remember that 10.0.0.20 is the VLAN address of the switch through which we can access it remotely.

6. Now provide the Telnet password that you set in step 3. Mine is **cisco.** Notice that password characters **won't show up(no echo)** on the screen as you type them, but just type, then hit ENTER.

After you're authenticated, you will see the the CLI of the remote switch appear.

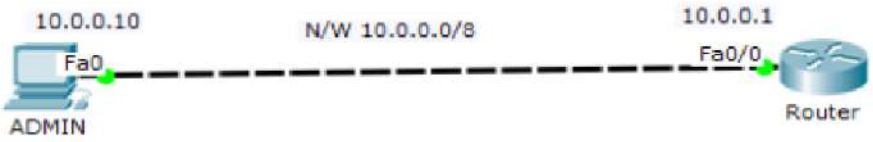
Now provide the enable password **admin** (or yours which you set in step 1) to enter the **privileged executive mode** of the switch. You can then continue to configure your switch the way you desire(but now, **remotely**)

Note that Telnet and enable passwords are different. **Enable password** authenticates you into privileged executive mode of the terminal device(switch, for example), but you're using **Telnet Password** to allow you access the interface of the remote device after connecting to it. You can see that we used telnet password to access the CLI of the remote switch.

B. Telnet Configuration on Router

For a router, Telnet configuration is almost same as that of the switch.

1. Build the network topology below



2. Configure **enable password** or **enable secret password** on the router

Router>en

Router#config term

Router(config)#enable password admin

3. Configure IP addresses on the admin PC and interface fa0/0 of the router

Router

Router(config)#int fa0/0

Router(config-if)#ip address 10.0.0.1 255.0.0.0

Router(config-if)#no shut

Admin PC

IP address 10.0.0.10 **Subnet mask** 255.0.0.0 **Default gateway** 10.0.0.1

4. Configure VLAN interface on the router. This interface allows for remote access on a switch or router via protocols such as Telnet or [Secure Shell(SSH)]

Router(config)#int VLAN 1

Router(config-if)#no shutdown

As you can see, we've not configured the VLAN interface with an IP address. We could do this but it is unnecessary. We already have an interface fa0/0 of the router with an IP address through which we can Telnet the router from the PC.

4. Configure **Telnet password** on **VTY lines** and configure remote login.

```
Router(config)#

Router(config)#line vty 0 15

Router(config-line)#password cisco

Router(config-line)#login
```

5. We can now telnet the router using the IP address of fa0/0 interface. So, in the command prompt of the admin PC type ***telnet 10.0.0.1*** then hit enter key.

6. Provide Telnet Password (that you set in step 4), then hit enter. Correct password allows you access the CLI of the router .

7. Now provide the enable password (that you set in step 2) to be allowed into **privileged executive mode** of the router.

You can now do configurations on the router from the PC remotely.

Result:

Thus, the configuration and verification of Telnet protocol is done successfully.