PRACTICAL 1

AIM:-	Date:- / /

Simulation/Program:-

Step 1: Initially, create a topology as shown below.

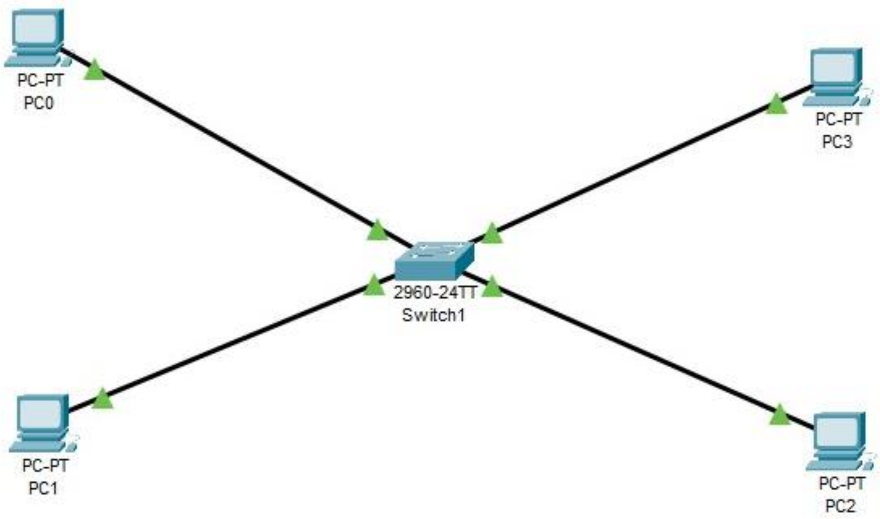


Figure 1.1: Topology Created

Step 2: Assign IP addresses to each pc.

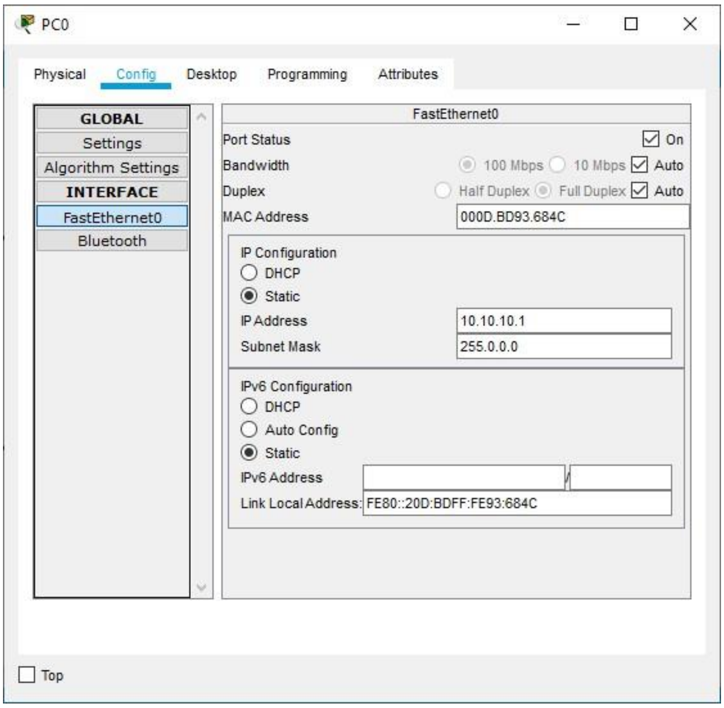


Figure 1.2: IP Address Assignment

Step 3: Create VLANs and enable showing Ethernet ports (Options – Preferences - Always Show Port Labels in Logical Workspace)

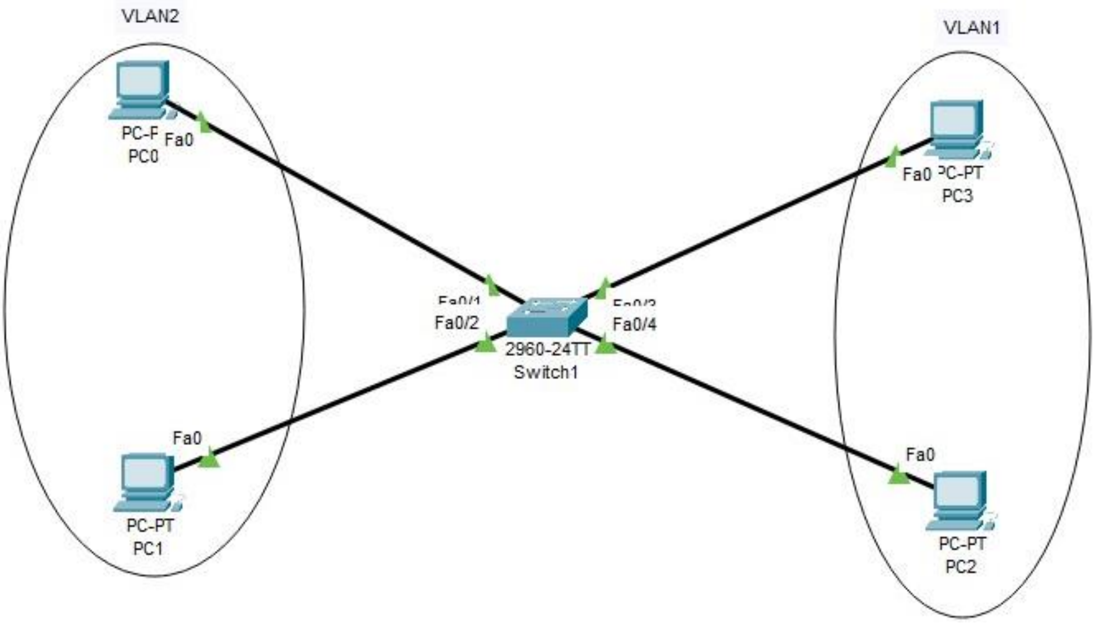


Figure 1.3: VLANs Creation

Step 4: Provide name to each VLAN and give access using Ethernet port number using commands shown below.

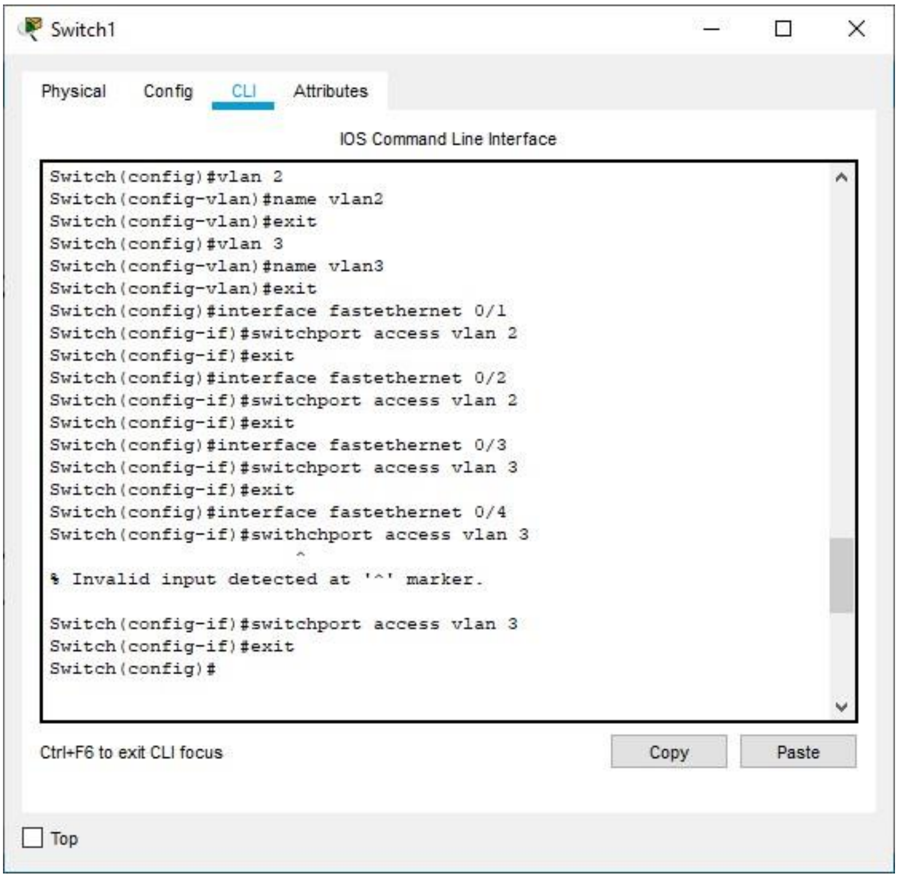


Figure 1.4: Providing Ethernet Access

Output:-

Finally, when we try to send packets from the pc-0 of vlan-2 to pc-2 of vlan-3, it will fail as it follows the VLAN approach now.

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit
	Failed	PC0	PC2	ICMP		0.000	N	0	(edit)

Figure 1.5: Resultant Image

Conclusion/Summary:-

Student Signature	Marks	Examiner Signature

PRACTICAL 2

AIM:-	**Date:- / /**

Theory:-

Router:

A router is a device that connects two or more packet-switched networks or subnetworks. It serves two primary functions: managing traffic between these networks by forwarding data packets to their intended IP addresses, and allowing multiple devices to use the same Internet connection.

There are several types of routers, but most routers pass data between LAN (Local Area Network) and WAN (Wide Area Network). A LAN is a group of connected devices restricted to a specific geographic area. A LAN usually requires a single router.

LAN:

A local area network (LAN) is a collection of devices connected together in one physical location, such as a building, office, or home. A LAN can be small or large, ranging from a home network with one user to an enterprise network with thousands of users and devices in an office or school.

Regardless of size, a LAN's single defining characteristic is that it connects devices that are in a single, limited area. In contrast, a Wide Area Network (WAN) or metropolitan area network (MAN) covers larger geographic areas. Some WANs and MANs connect many LANs together.

Gateway:

A gateway is a network node used in telecommunications that connects two networks with different transmission protocols together. Gateways serve as an entry and exit point for a network as all data must pass through or communicate with the gateway prior to being routed. In most IP-based networks, the only traffic that does not go through at least one gateway is traffic flowing among nodes on the same local area network (LAN) segment. The term default gateway or network gateway may also be used to describe the same concept.

The primary advantage of using a gateway in personal or enterprise scenarios is simplifying internet connectivity into one device. In the enterprise, a gateway node can also act as a proxy server and a firewall. Gateways can be purchased through popular technology retailers, such as Best Buy, or rented through an internet service provider.

Simulation/Program:-

Step 1: Initially, create a topology as shown below and assign IP addresses to each pc.

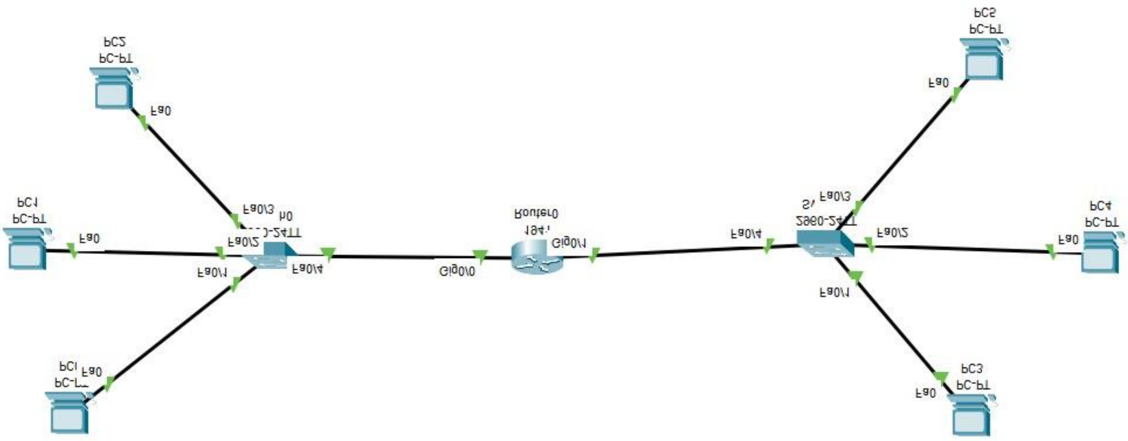


Figure 2.1: Topology Created

Step 3: Assign IP addresses 192.16.1.254 and 172.16.1.254 to the GigaEthernet0/0 and GigaEthernet0/1 respectively.

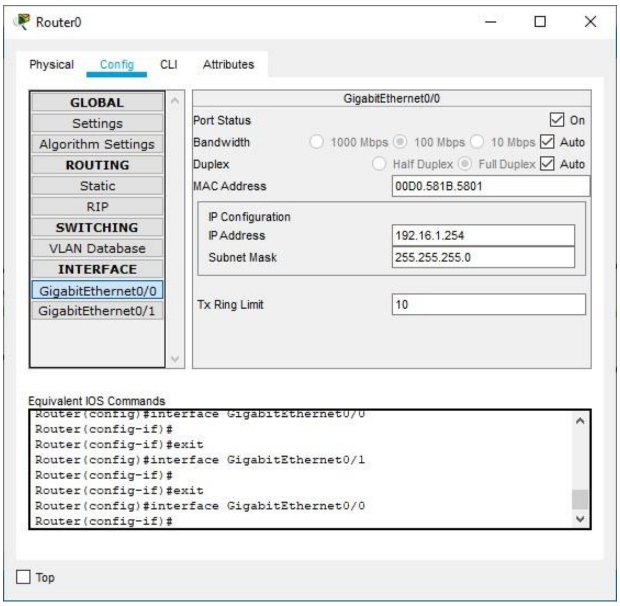


Figure 2.2: IP Address Assignment

Step 4: Turn ON port status for GigaEthernet0/0 and GigaEthernet0/1.

Figure 2.3: Port Status for GigaEthernet0/0

Figure 2.4: Port Status for GigaEthernet0/1

Step 5: Assign default gateway 192.16.1.254 to each pc of LAN 1 and 172.16.1.254 to each pc of LAN 2.

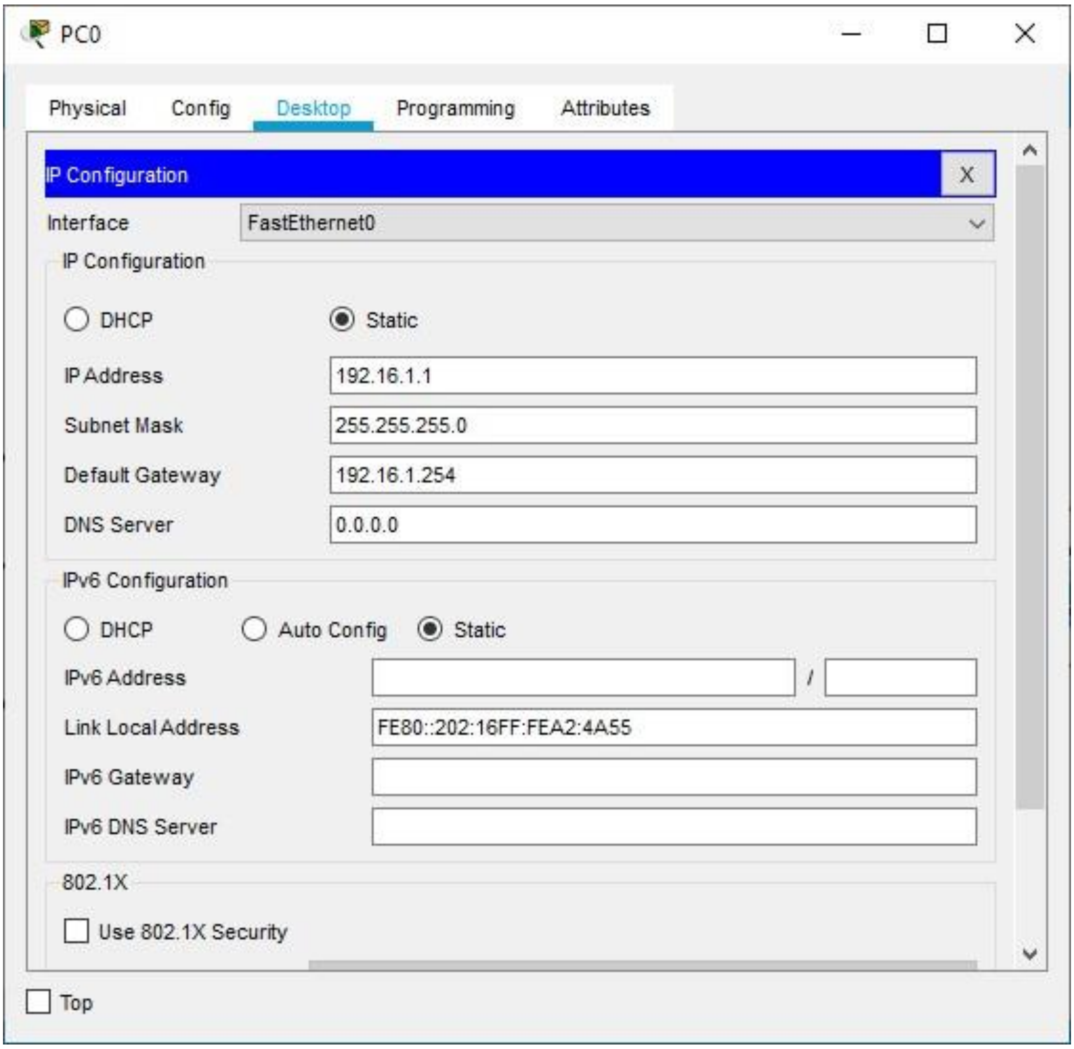


Figure 2.5: Default Gateway Assignment

Output:-

Finally, when we try to send packets from the pc-0 of LAN-1 to pc-3 of LAN-2, it will fail as it checks for path.

Now, for the second time when we try to send packets with the same route, it will be successful as shown below.

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num
	Failed	PC0	PC3	ICMP		0.000	N	0
	Successful	PC0	PC3	ICMP		0.000	N	1

Figure 2.6: Result

Conclusion/Summary:-

Student Signature	Marks	Examiner Signature

PRACTICAL 3

AIM:-	Date:- / /

Theory:-

Step 1: Set connection between router and switch.

We have taken three Router and connected each router with switch. Each switch is connected with two PCs. We provide the IP address to each PC. When we try to connect router with another Router 2 with Router 3 error occurs as all ports of Router 2 are busy.

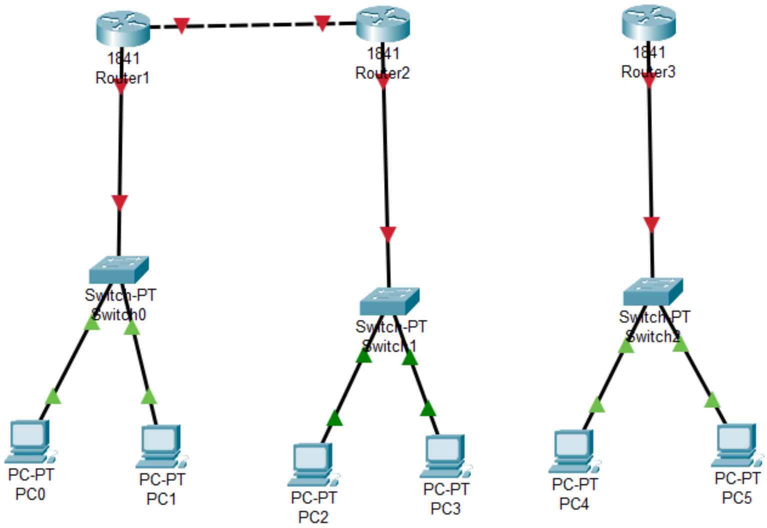


Figure 3.1: Router connection

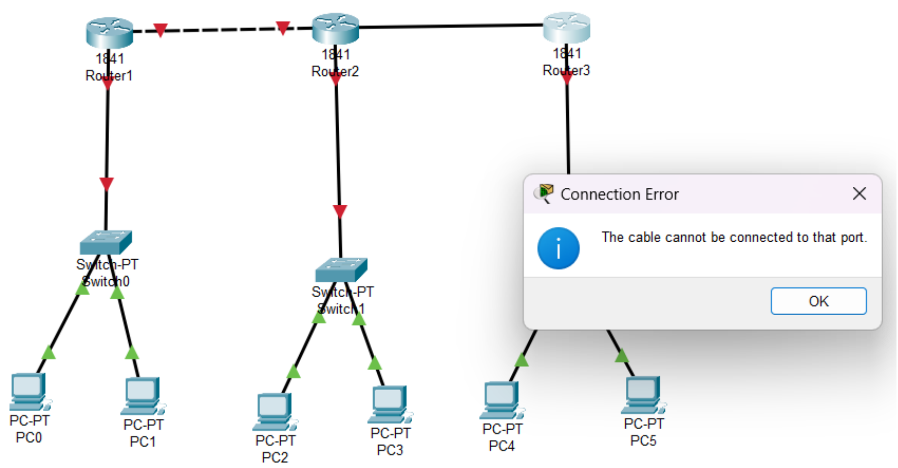


Figure 3.2: Router connection error

Step 2: Apply Port connection

As the connection between Router 2 and Router 3 are not happening so we try to add port in each Router. To do this we have to click on Router then click on physical then in WIC-1T we have to add port.

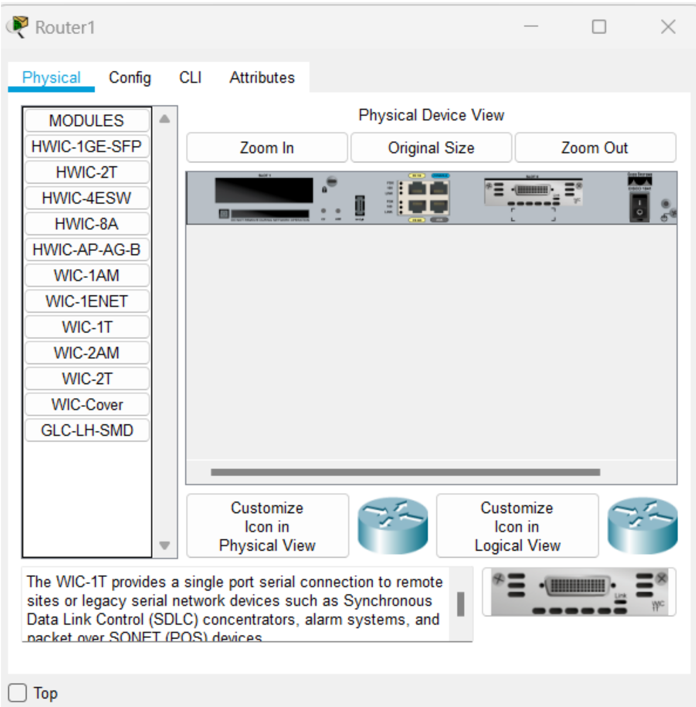


Figure 3.3: Port connection

After that we have to open port status and add IP address to each Router.

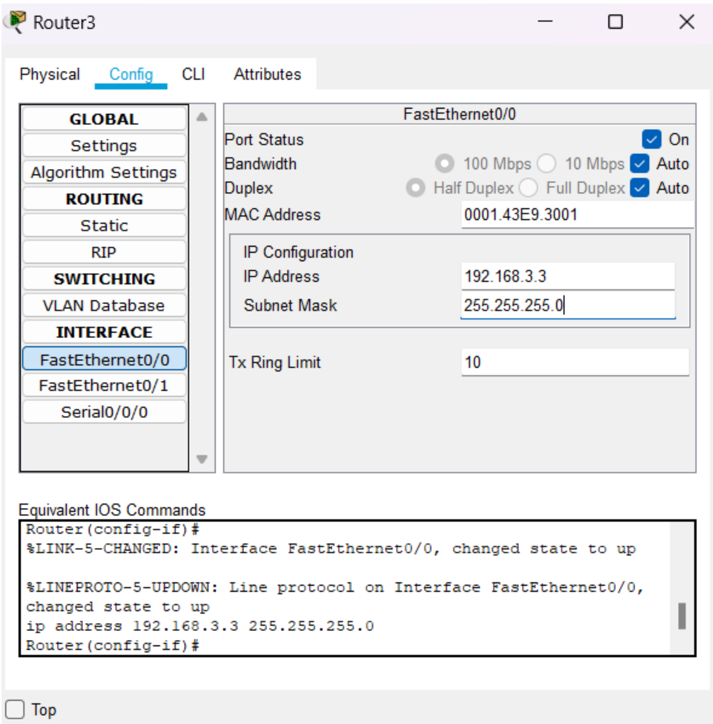


Figure 3.4: Opening port status & setting IP

Step 3: Inter Router connection

After setting the port in Router connection can be establish. Now we can choose serial cable to connect Routers.

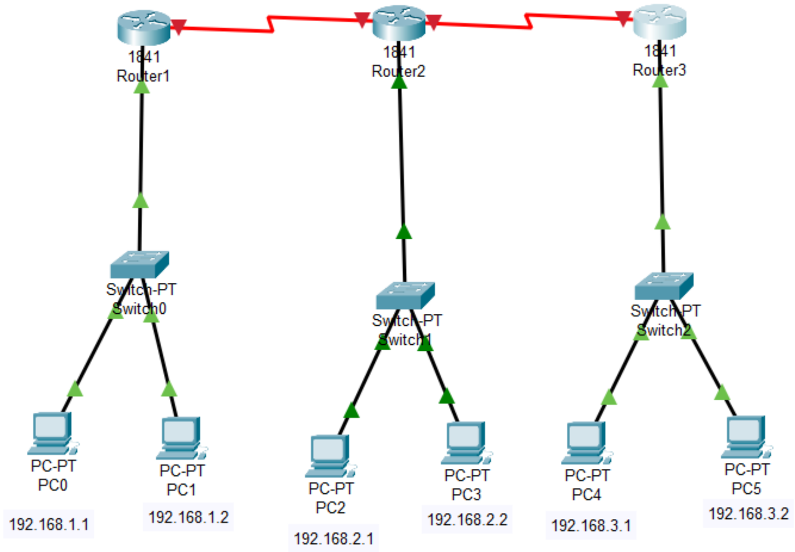


Figure 3.5: Router connection

Step 4: Set IP address in serial port connection of Router.

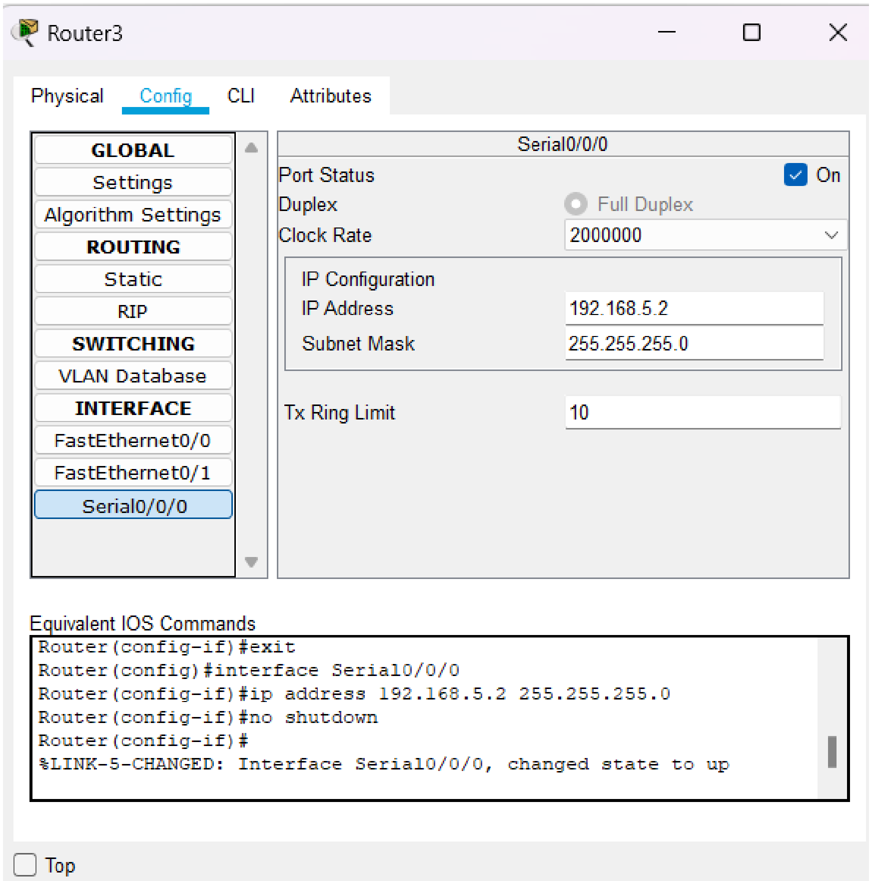


Figure 3.6: Setting IP address

Now connection will show green signal between Routers.

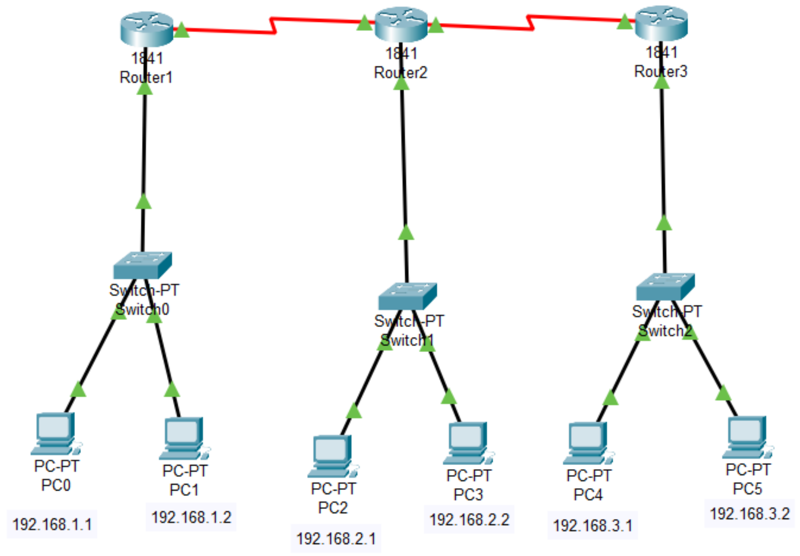


Figure 3.7: Router connection

Step 5: Try to ping message from one connection to another.

Initially we try to send message inside the connection from PC-0 to PC-1, PC-2 to PC-3, PC-4 to PC-5. This will show successful message transfer status.

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	PC1	ICMP		0.000	N	0	(edit)	(delete)
	Successful	PC2	PC3	ICMP		0.000	N	1	(edit)	(delete)
	Successful	PC4	PC5	ICMP		0.000	N	2	(edit)	(delete)

Figure 3.8: Successful status of message transfer

Now we try to send message from PC-0 to PC-2, PC-3, PC-4, PC-5. It will show status as failed because the connection between two different Router has not been established and default gateway is not yet set.

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Failed	PC0	PC3	ICMP		0.000	N	4	(edit)	(delete)
	Failed	PC0	PC4	ICMP		0.000	N	5	(edit)	(delete)
	Failed	PC0	PC5	ICMP		0.000	N	6	(edit)	(delete)

Figure 3.9: Failed status of message transfer

Step 6: Static Routes has to be added

This step is done by clicking on Router then in config section select static. Now set Network, Mask and Next Hop information in each Router.

In Router-1, 2, 3 static routes are added as shown below,

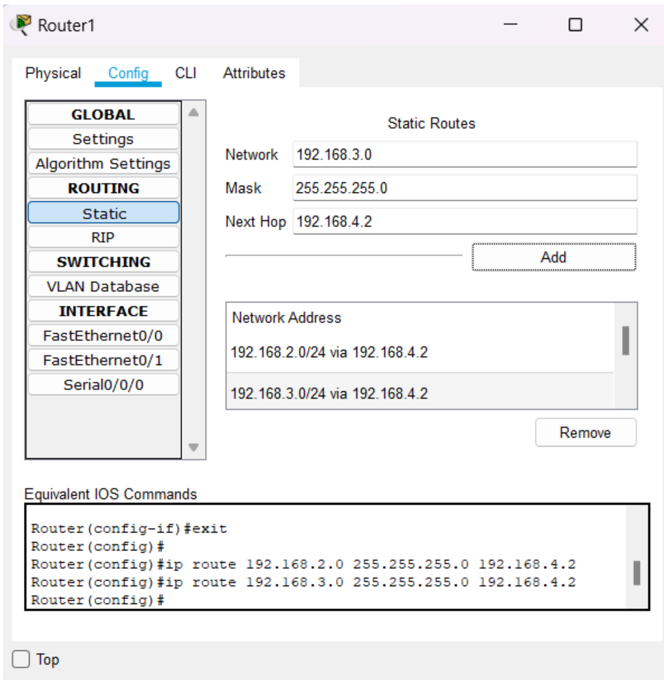


Figure 3.10: Adding Static Routes for Router-1

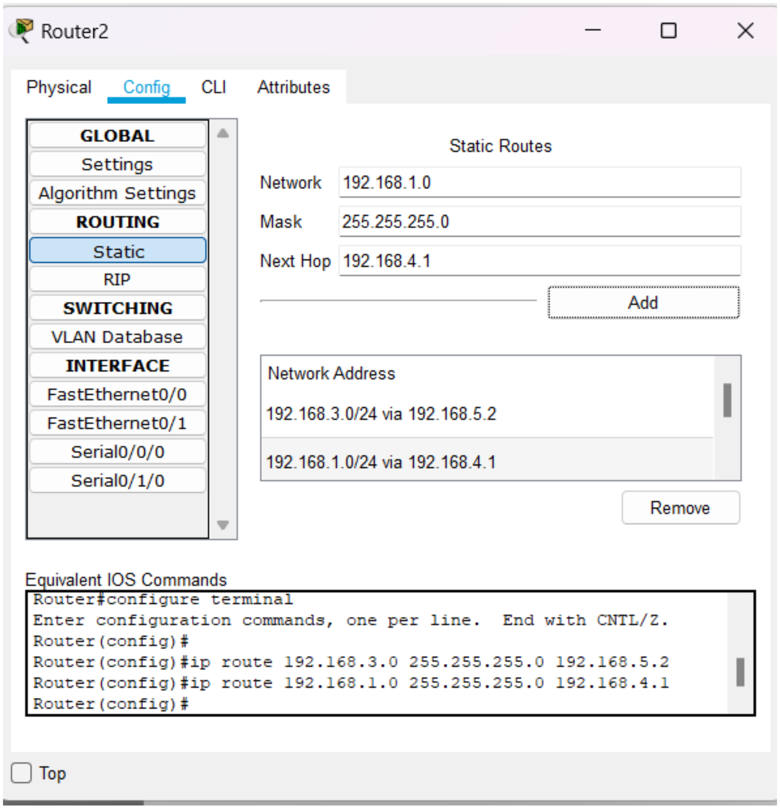


Figure 3.11: Adding static Routes for Router-2

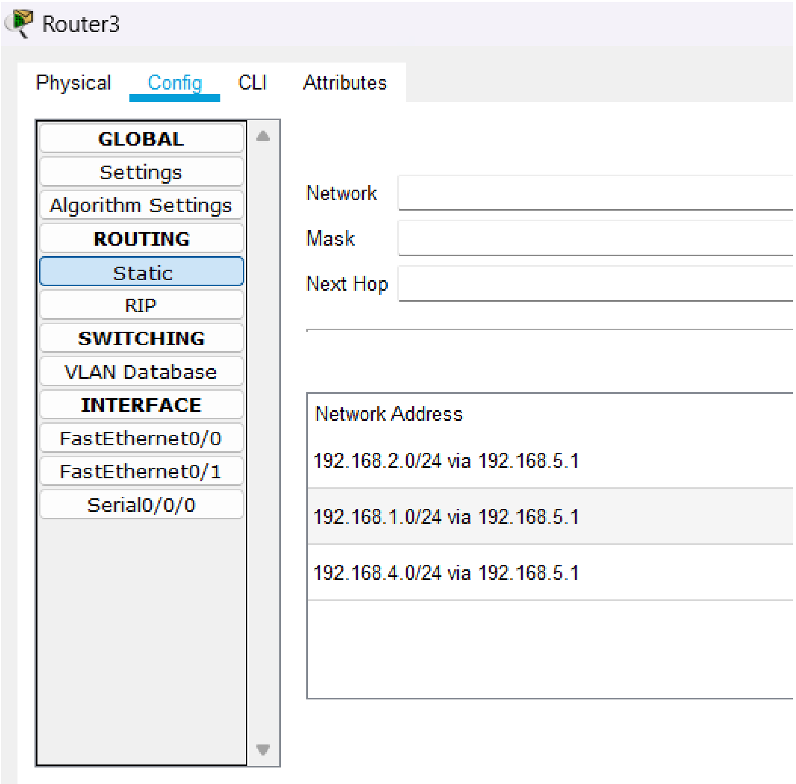


Figure 3.12: Adding static Routes for Router-3

Now we are adding Default gateway in each PCs.

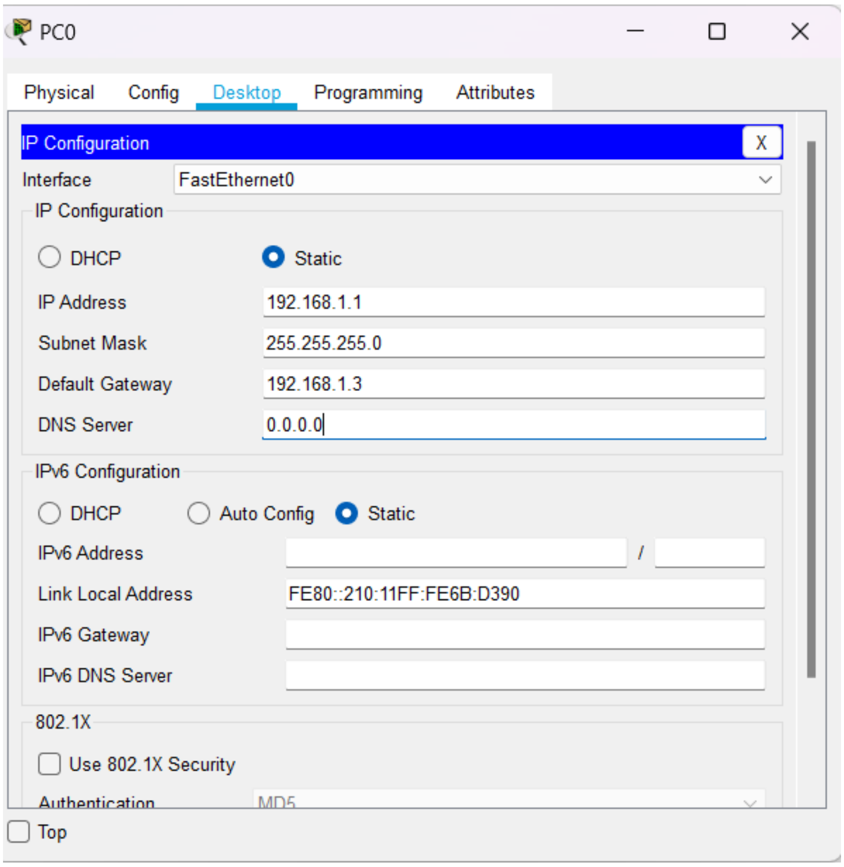


Figure 3.13: Default Gateway setting

Step 7: Try to ping message again

Now if we again try to Ping the message from one connection to another at first it will show an error due to checking the path and when we again try to ping it will show successful message transfer Status.

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	PC2	ICMP		0.000	N	9	(edit)	(delete)
	Successful	PC0	PC4	ICMP		0.000	N	10	(edit)	(delete)

Figure 3.14: Successful status of interconnection message ping

Conclusion/Summary:-

Student Signature	Marks	Examiner Signature

PRACTICAL 4

AIM:-	Date:- / /

Theory:-

Solution:

RIP (Routing Information Protocol):

It is a distance vector routing protocol. It uses distance routing algorithm to decide which path to put a packet on to get to its destination. You can use RIP to configure the hosts as part of a RIP network. It works on Bellman-ford algorithm to cover path. It permits maximum of 15 hops.

OSPF (Open Shortest Path First):

It is a link state routing protocol that was developed for IP networks and is based on the Shortest Path First algorithm which is Dijkstra algorithm. OSPF is an Interior Gateway Protocol (IGP). It has no limit on hops. This covers the best shortest path.

<u>Configuration Using RIP Protocol</u>

Step 1: Make connection between Routers.

First take two Router 1841 and connect each Router with PC. As shown in below figure PC-0 is connected with Router-0 and PC-1 connected with Router-1. Now give each PC IP address and similarly give IP address to Routers. For connecting the routers we have to select WIC-1T port and connect to make connection between two Routers successful.

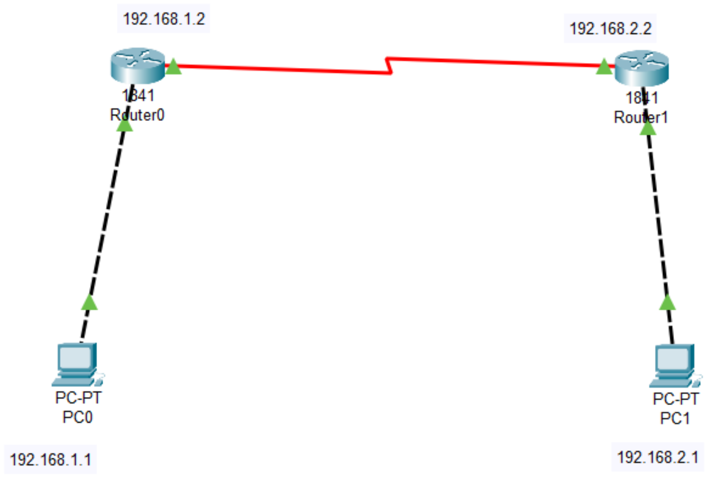


Figure 4.1: Connections between Routers

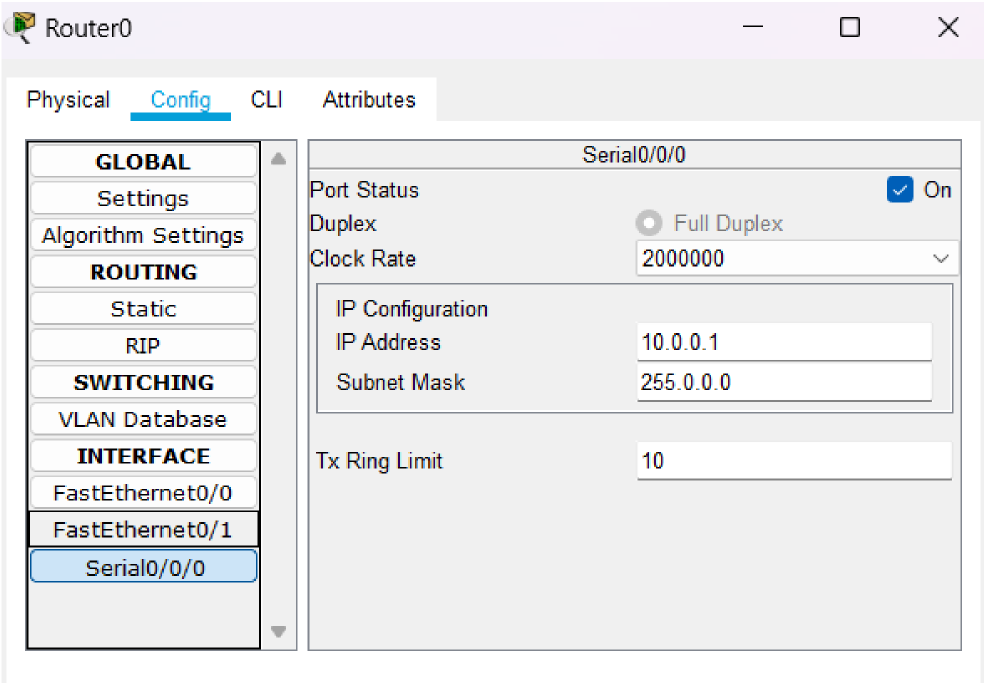


Figure 4.2: Router IP address setting

This IP address is set for connection between routers using serial cable 0/0/0. For fast-ethernet connection between device and router we have to enter related IP address.

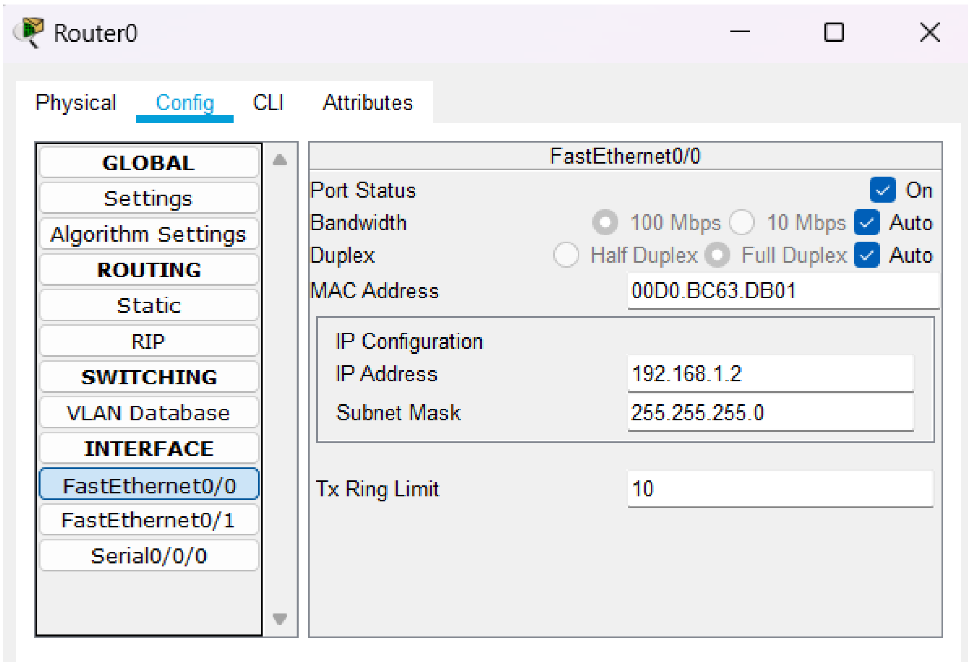


Figure 4.3: IP address in Router

Step 2: Enter Default Gateway in Devices.

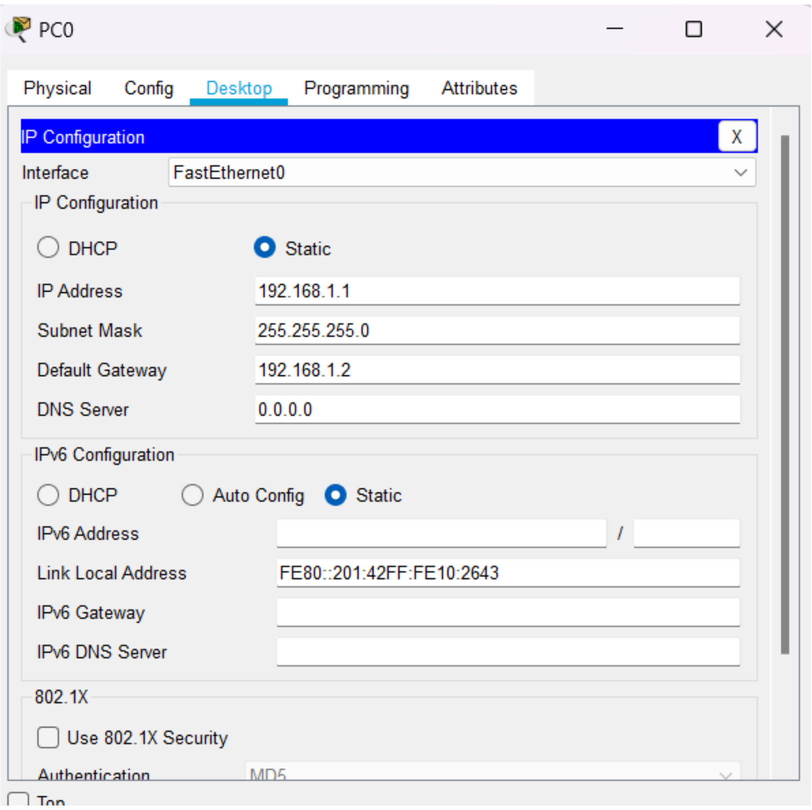


Figure 4.4: IP address and Gateway setting

Step 3: Add Network Address in RIP Routing in Router-0

We have to add two IP address of two connection in our configuration in Router-0. Click on add to add the network address.

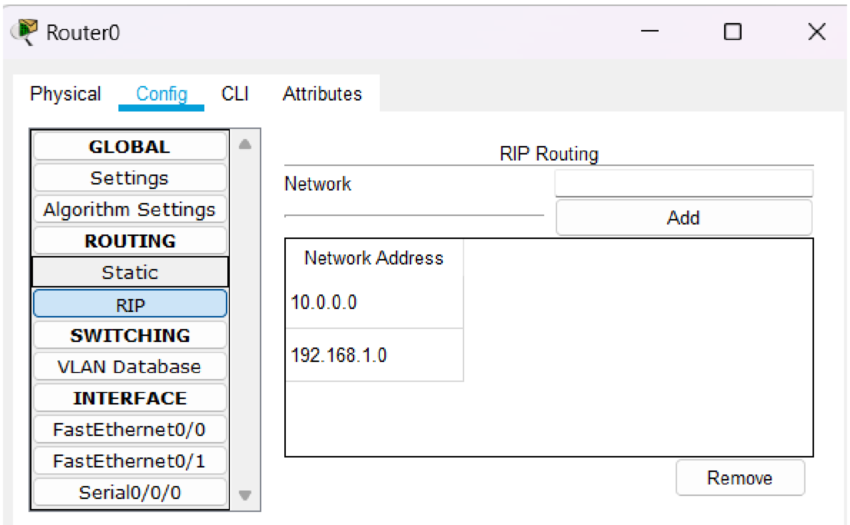


Figure 4.5: RIP Routing

Step 4: Enter network address in CLI of Router-1

Now we have to add Network address of connection in CLI of Router-2. To do this enter ‘network ip address’ command in CLI.

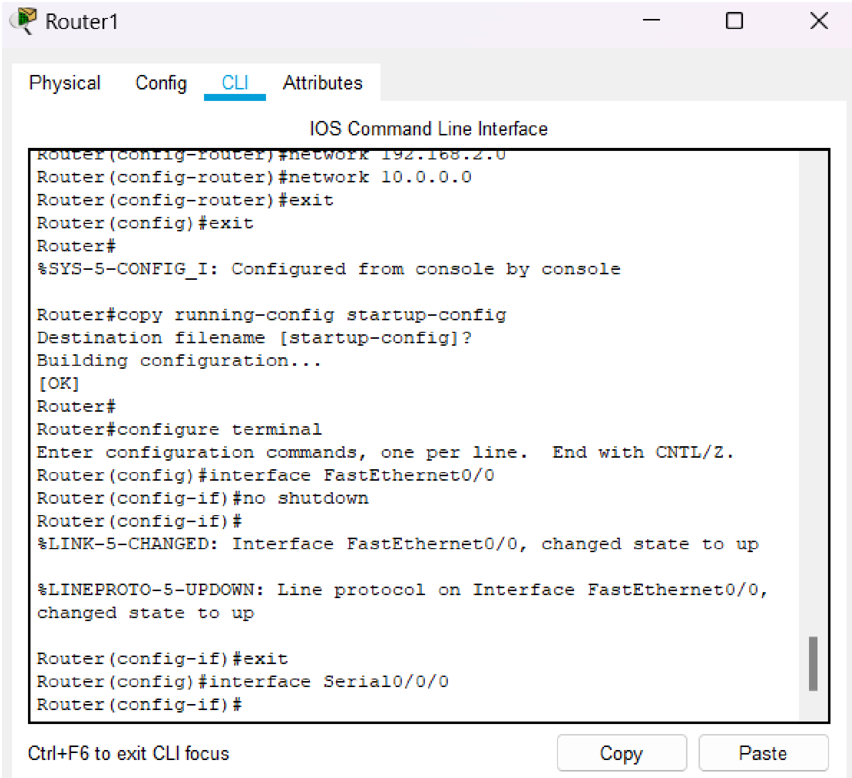
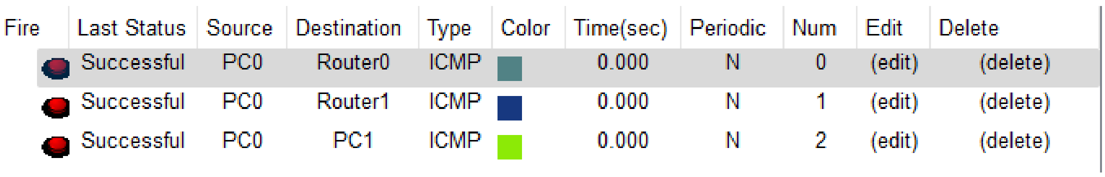


Figure 4.6: Network address in CLI

Step 5: Ping message

Now try to ping the message from PC-0 to Router-0, Router-1 and PC-1. This will show successful status.



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	Router0	ICMP		0.000	N	0	(edit)	(delete)
	Successful	PC0	Router1	ICMP		0.000	N	1	(edit)	(delete)
	Successful	PC0	PC1	ICMP		0.000	N	2	(edit)	(delete)

Figure 4.7: Status of Successful

To see it in simulator, click on edit filter then uncheck all the option other than RIP.

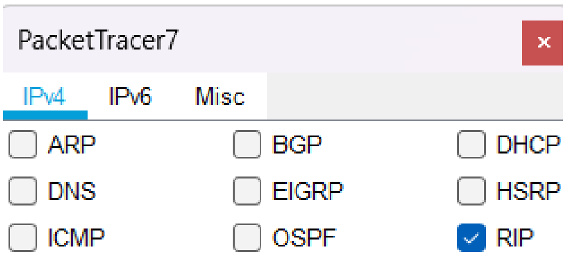


Figure 4.8: Unchecking other han RIP option

Now start the simulator and see the event and time in pinging message.

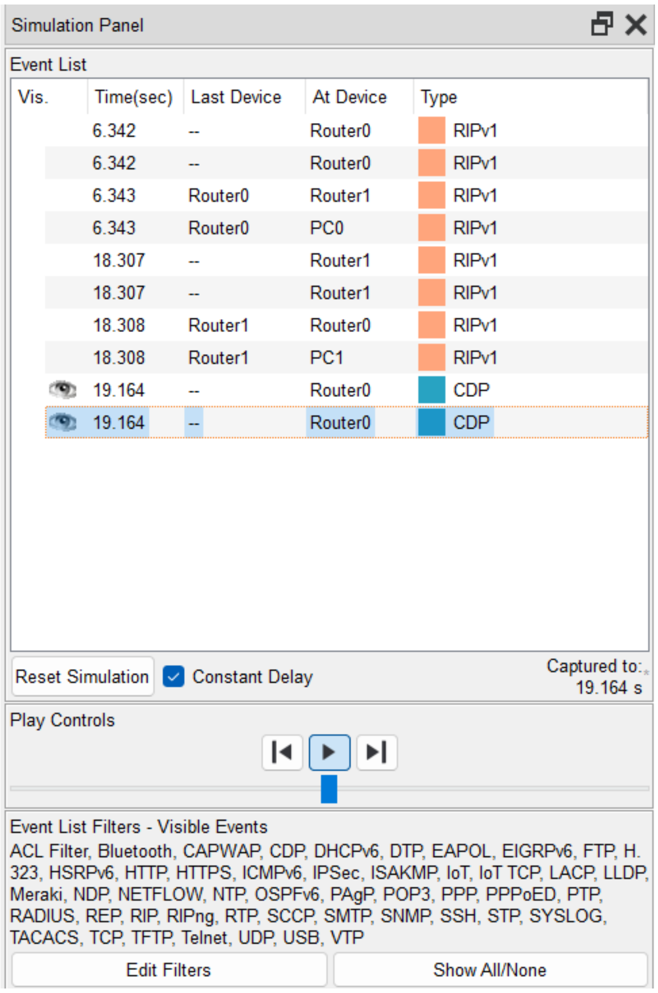


Figure 4.9: Simulator

Configuration Using OSPF Protocol

Step 1: Make connection between Routers

Take three Routers and connect them as shown in figure below. Connect PC-0 to Router-0 and connect PC-1 to Router-2. Give each device IP address. As we have previously done add WIC-1T port and make connection between three Routers. Add IP address in serial port according to connected Router.

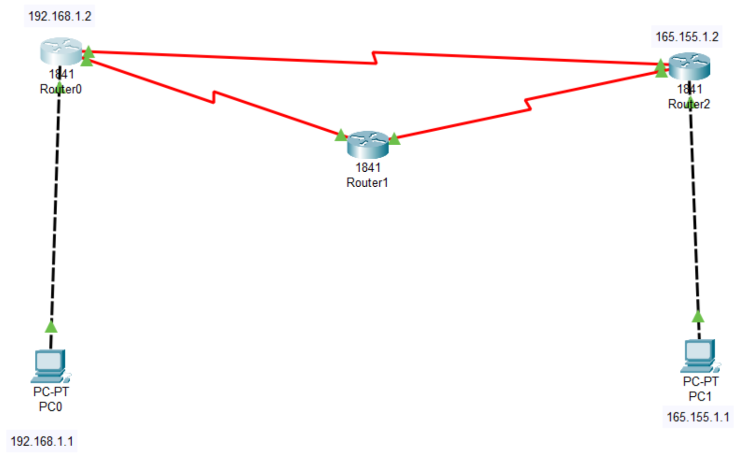


Figure 4.10: Router configuration

Step 2: Add Router configuration from CLI.

To add this, you have to write command ‘network_IP-address_Complement-of-Subnet Mask_area 0’.

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#ip address 10.10.0.2 255.0.0.0
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial2/0, changed state to down
Router(config-if)#clock rate 64000
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial3/0, changed state to down
Router(config-if)#clock rate 64000
Router(config-if)#ip address 12.12.0.2 255.0.0.0
Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up

%LINK-5-CHANGED: Interface Serial3/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up

Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#
Router(config-if)#exit
Router(config)#router ospf 1
Router(config-router)#network 192.168.1.0
```

Figure 4.11: Router configuration

Now try to ping message from PC-0 to PC-1 and see the status of this process. At first it will show failed as it finds the path and second time it will show status successful. As we have chosen OSPF connection the best and shortest path in network connection will be taken from passing the message.

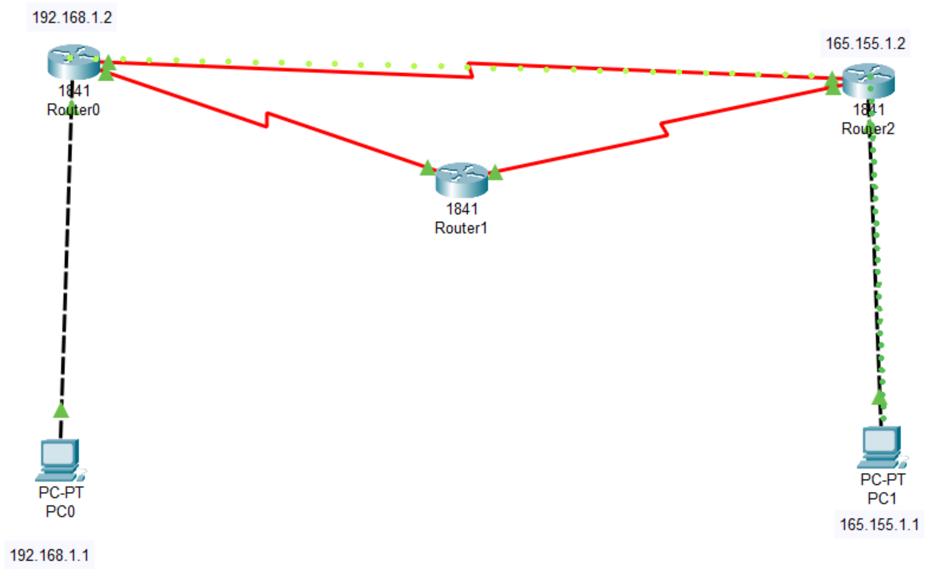
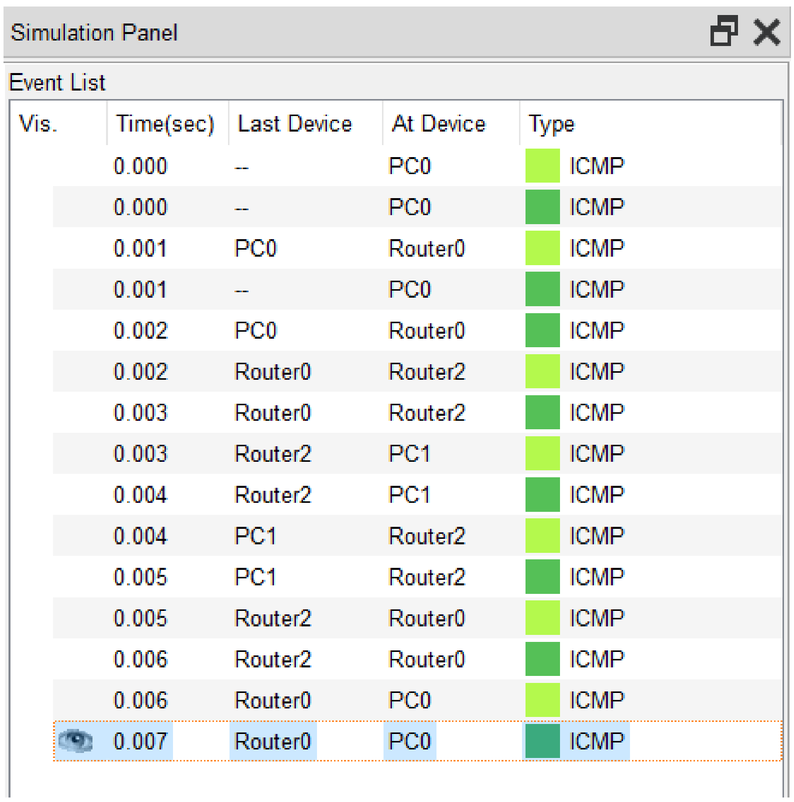


Figure 4.12: Path of message passing

Here green dotted line shows the path of passing the message. Below we have shown the image of Simulation Panel to show route and time.


Simulation Panel

Event List

Vis.	Time(sec)	Last Device	At Device	Type
	0.000	--	PC0	ICMP
	0.000	--	PC0	ICMP
	0.001	PC0	Router0	ICMP
	0.001	--	PC0	ICMP
	0.002	PC0	Router0	ICMP
	0.002	Router0	Router2	ICMP
	0.003	Router0	Router2	ICMP
	0.003	Router2	PC1	ICMP
	0.004	Router2	PC1	ICMP
	0.004	PC1	Router2	ICMP
	0.005	PC1	Router2	ICMP
	0.005	Router2	Router0	ICMP
	0.006	Router2	Router0	ICMP
	0.006	Router0	PC0	ICMP
	0.007	Router0	PC0	ICMP

Figure 4.13: Simulation Panel

Conclusion/Summary:-

Student Signature	Marks	Examiner Signature

PRACTICAL 5

AIM:-	Date:- / /

Theory:-

EIGRP Protocol:

Enhanced Interior Gateway Routing Protocol (EIGRP) is an advanced distance-vector routing protocol that is used on a computer network for automating routing decisions and configuration. The protocol was designed by Cisco Systems as a proprietary protocol, available only on Cisco routers.

BGP protocol:

BGP (Border Gateway Protocol) is the protocol underlying the global routing system of the internet. It manages how packets get routed from network to network through the exchange of routing and reachability information among edge routers.

Configuration using EIGRP Protocol:

Step 1: Do Basic connections

Take four 1841 Router and take two PCs. Connect two PCs each with Routers. Connect these four Routers to each other.

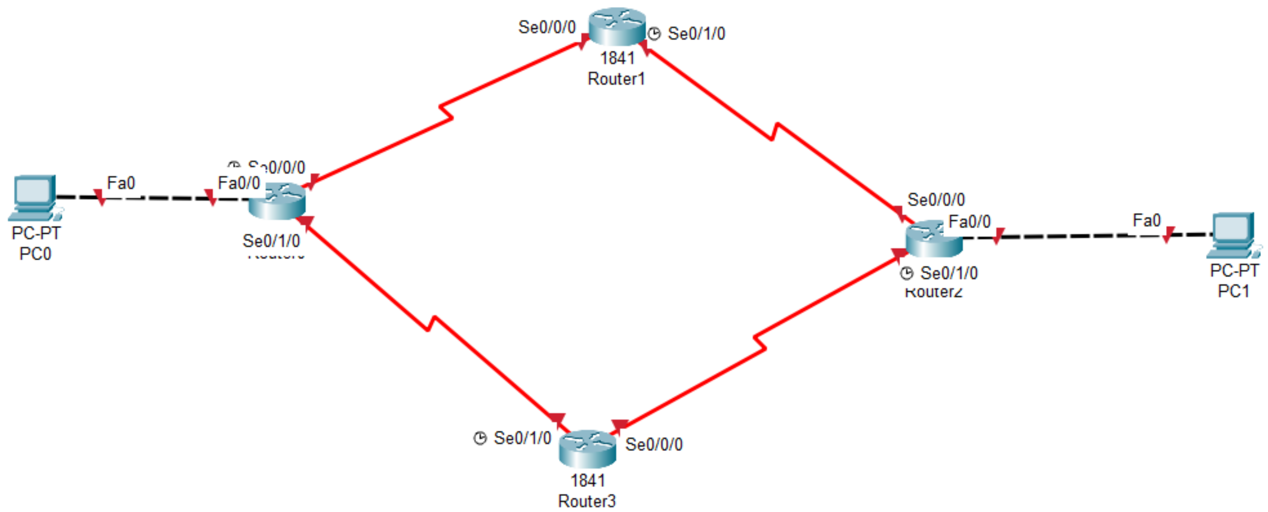


Figure 5.1: Basic Connection

Step 2: Apply IP address to each PC and Router to related networks. Also apply default gateway.

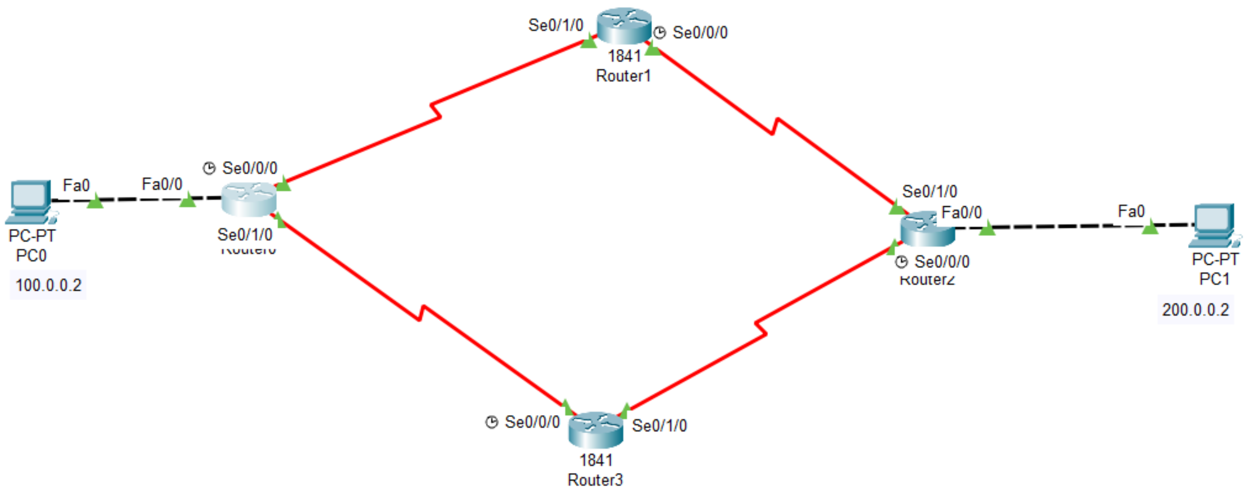


Figure 5.2: Set IP Address

Step 3: Apply commands in CLI of each Router.

Open CLI and apply commands which are shown in below image this will configure the whole connection and implement EIGRP Protocol.

We also have to set clock rate and add IPs of related Virtual Connection. After doing this we have to ping the message and check process in simulation panel also check status of message passing.

Router0

Physical Config CLI Attributes

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface serial0/0/0
Router(config-if)#exit
Router(config)#interface serial0/0/0
Router(config-if)#clock rate 64000
Router(config-if)#exit
Router(config)#interface serial0/1/0
Router(config-if)#no clock rate
This command applies only to DCE interfaces
Router(config-if)#interface serial0/0/0
Router(config-if)#exit
Router(config)#interface serial0/1/0
Router(config-if)#no clock rate
This command applies only to DCE interfaces
Router(config-if)#interface serial0/0/0
Router(config-if)#clock rate 64000
Router(config-if)#exit
Router(config)#interface serial0/1/0
Router(config-if)#no clock rate
This command applies only to DCE interfaces
Router(config-if)#interface serial0/1/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface serial0/1/0
Router(config-if)#exit
Router(config)#^Z
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface fastethernet0/0
Router(config-if)#exit
Router(config)#interface s0/0/0
Router(config-if)#ip add 10.0.0.1 255.0.0.0
Router(config-if)#clock rate 64000
Router(config-if)#bandwidth 72000
Router(config-if)#no shut
Router(config-if)#^Z
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface s0/1/0
Router(config-if)#ip add 40.0.0.2 255.0.0.0
Router(config-if)#no shut
Router(config-if)#^Z
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface f0/0
Router(config-if)#ip add 100.0.0.1 255.0.0.0
Router(config-if)#no shut
Router(config-if)#^Z
Router#
```

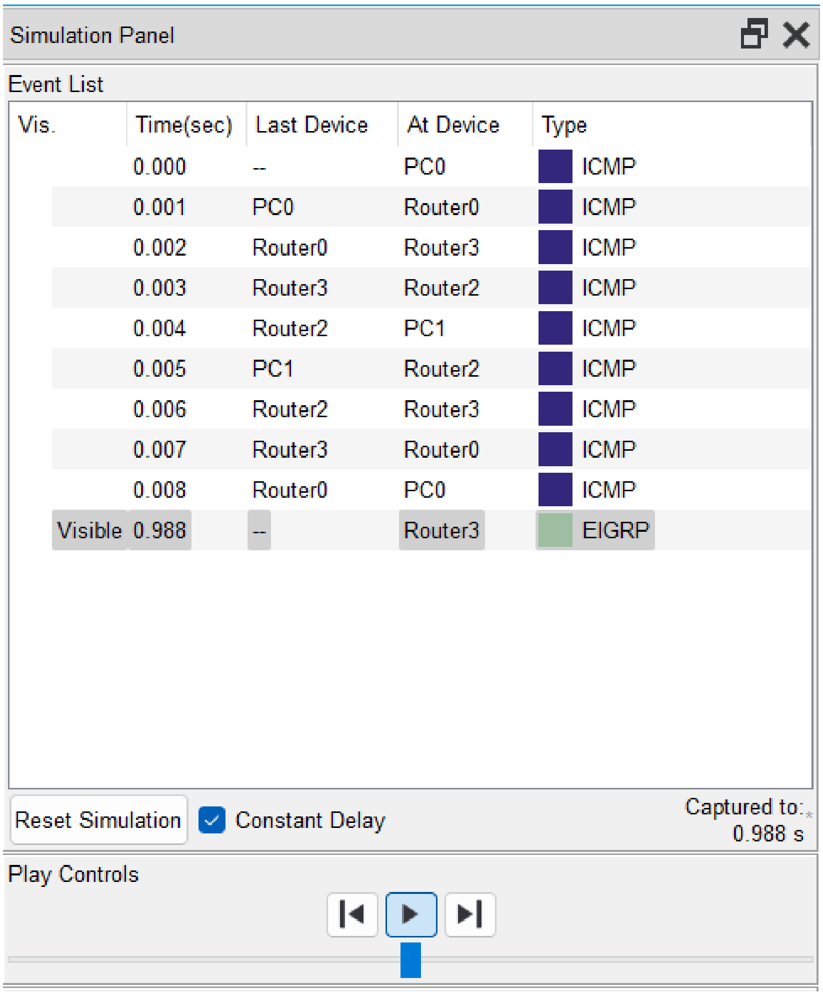
Ctrl+F6 to exit CLI focus

Figure 5.3: Router-0 Configuration

Step 4: Ping the message

Now ping the message from PC-0 to PC-1 and check the whole process in simulation also check the status of this message transfer.

We have shown below the image of simulation panel to track the message.



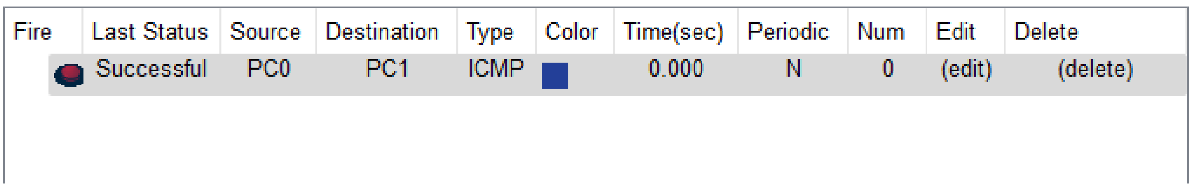
Simulation Panel

Event List

Vis.	Time(sec)	Last Device	At Device	Type
	0.000	--	PC0	ICMP
	0.001	PC0	Router0	ICMP
	0.002	Router0	Router3	ICMP
	0.003	Router3	Router2	ICMP
	0.004	Router2	PC1	ICMP
	0.005	PC1	Router2	ICMP
	0.006	Router2	Router3	ICMP
	0.007	Router3	Router0	ICMP
	0.008	Router0	PC0	ICMP
Visible	0.988	--	Router3	EIGRP

Figure 5.4: Simulation Panel

Status will be shown as successful.



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	PC1	ICMP		0.000	N	0	(edit)	(delete)

Figure 5.5: Successful Status

Configuration using BGP Protocol:

Step 1: Do Basic connection

Take two Routers and two PCs and connect this two PCs with Router. Give IP address to each PC and Router than apply default gateway.

While choosing Router we have to select Router-PT for connection.

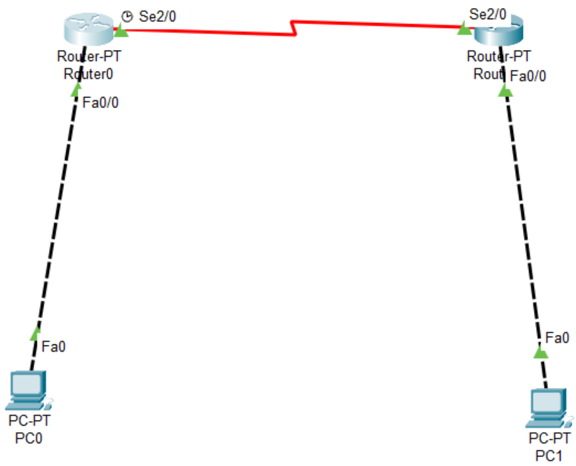


Figure 5.6: Basic Connection

Step 2: Apply commands in CLI

Now open CLI and apply command to implement connection using BGP Protocol.

While doing command we have to give IP address of neighbour's connection. For each Router we have to apply individual command to implement BGP Protocol.

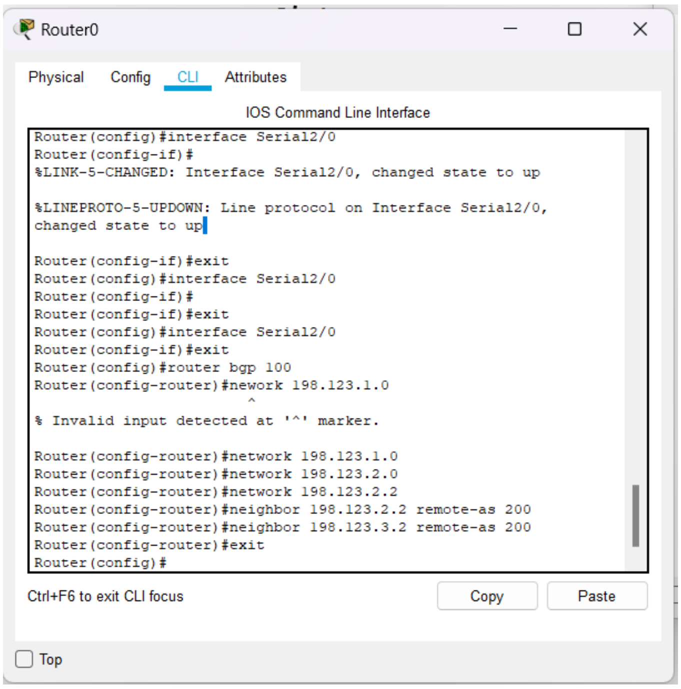


Figure 5.7: Applying commands in CLI

Do the same for Router-2

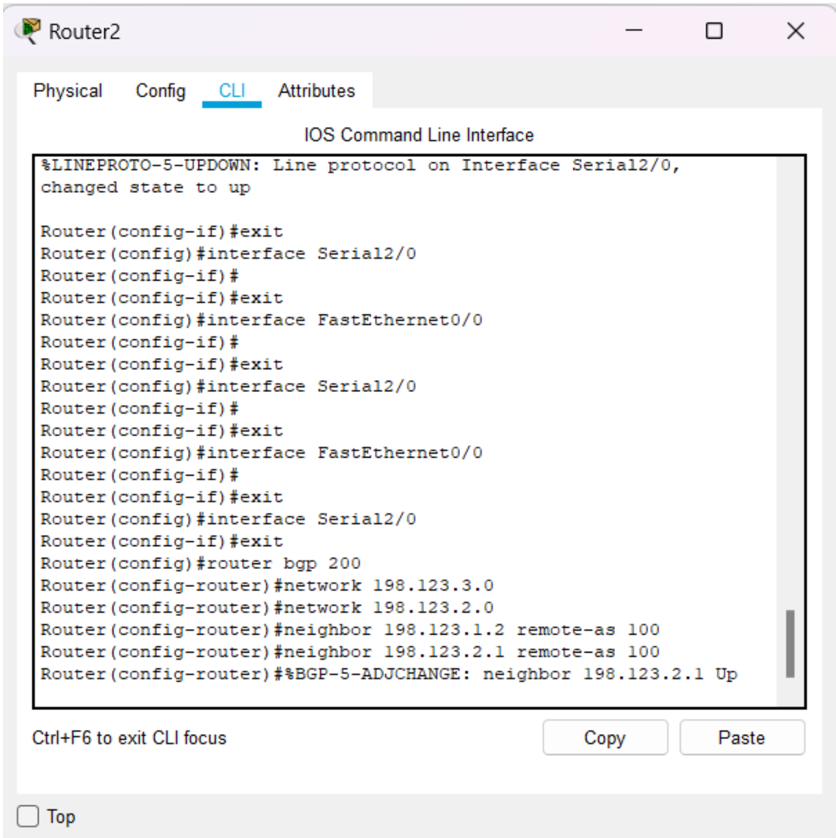
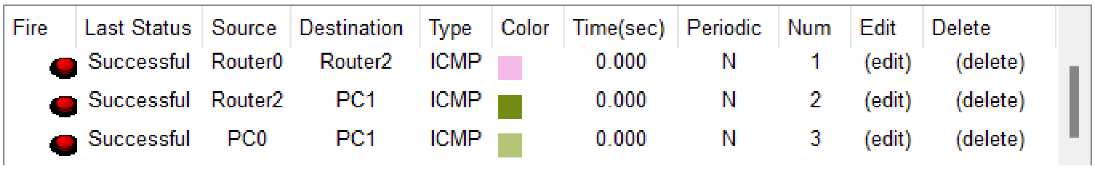


Figure 5.8: CLI commands for Router-2

Step 3: Ping the message

Now Ping the message from PC-0 to PC-1. This will show successful in Status panel.



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	Router0	Router2	ICMP		0.000	N	1	(edit)	(delete)
	Successful	Router2	PC1	ICMP		0.000	N	2	(edit)	(delete)
	Successful	PC0	PC1	ICMP		0.000	N	3	(edit)	(delete)

Figure 5.9: Successful Status

Conclusion/Summary:-

Student Signature	Marks	Examiner Signature

PRACTICAL 6

AIM:-	**Date:- / /**

Theory:-

This module describes how to configure Network Address Translation (NAT) for IP address conservation and how to configure inside and outside source addresses. This module also provides information about the benefits of configuring NAT for IP address conservation.

NAT enables private IP internetworks that use nonregistered IP addresses to connect to the Internet.

NAT operates on a device, usually connecting two networks. Before packets are forwarded onto another network, NAT translates the private (not globally unique) addresses in the internal network into legal addresses.

NAT can be configured to advertise to the outside world only one address for the entire network. This ability provides more security by effectively hiding the entire internal network behind that one address.

Simulation/Program:-

Step 1: Connect routers and PC's as shown below.

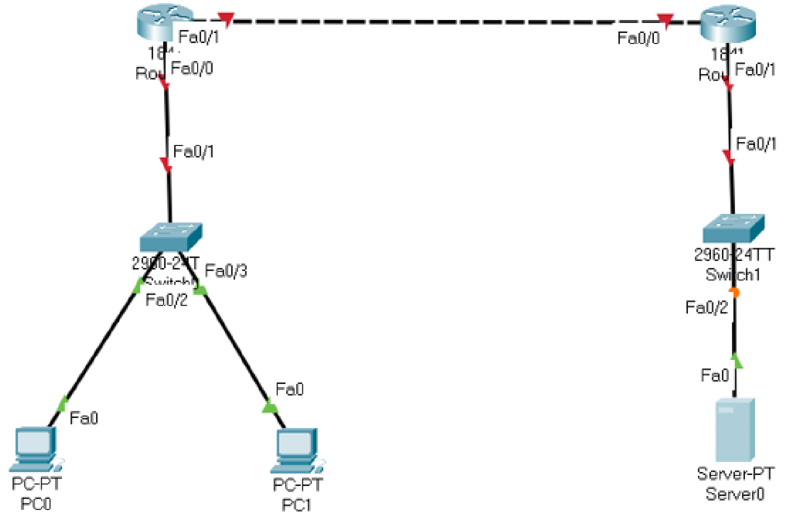


Figure 6.1: Connection

Step 2: In order to connect both the routers, we provide serial port. Attach the serial port in WIC-1T.

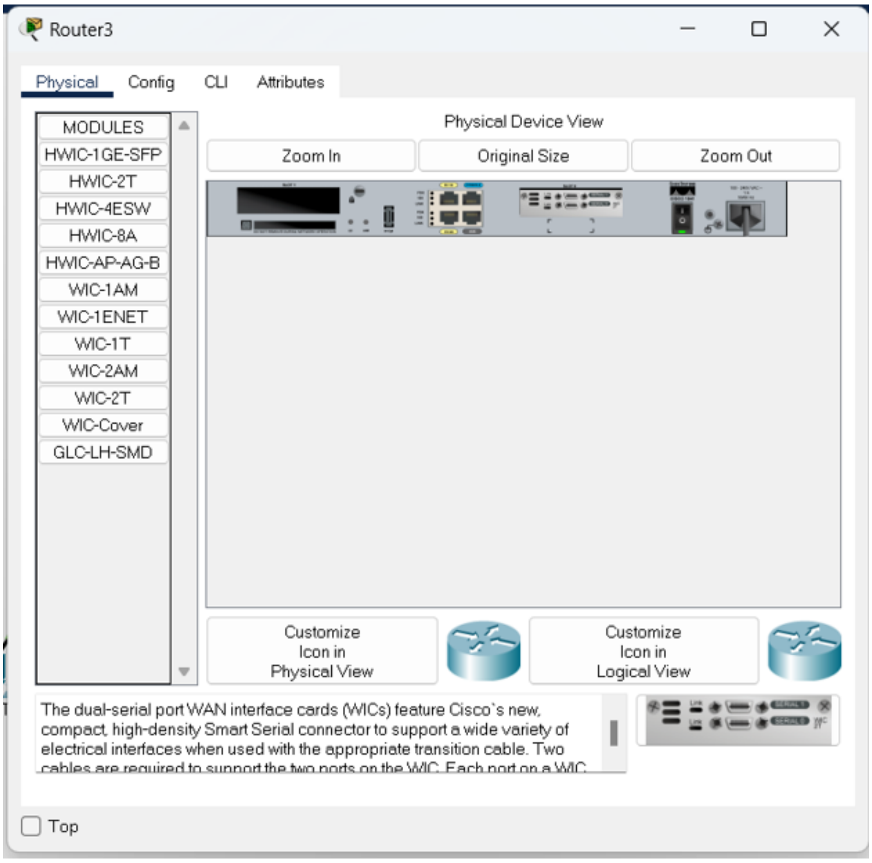


Figure 6.2: Serial Port configuration

Step 3: Provide IP address to the PCs with routers IP address as default gateway.

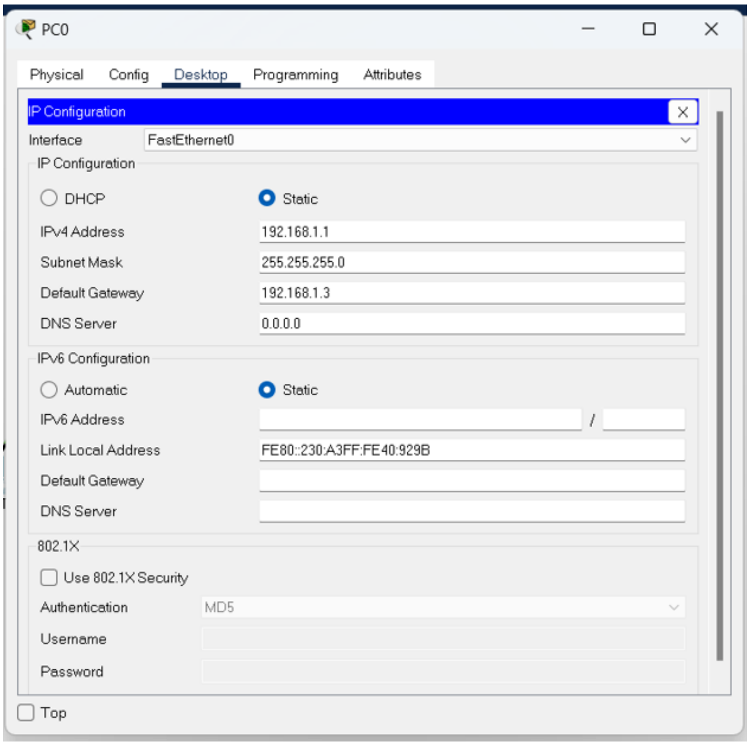


Figure 6.3. PC's IP address

Step 4: Router 0 configuration.

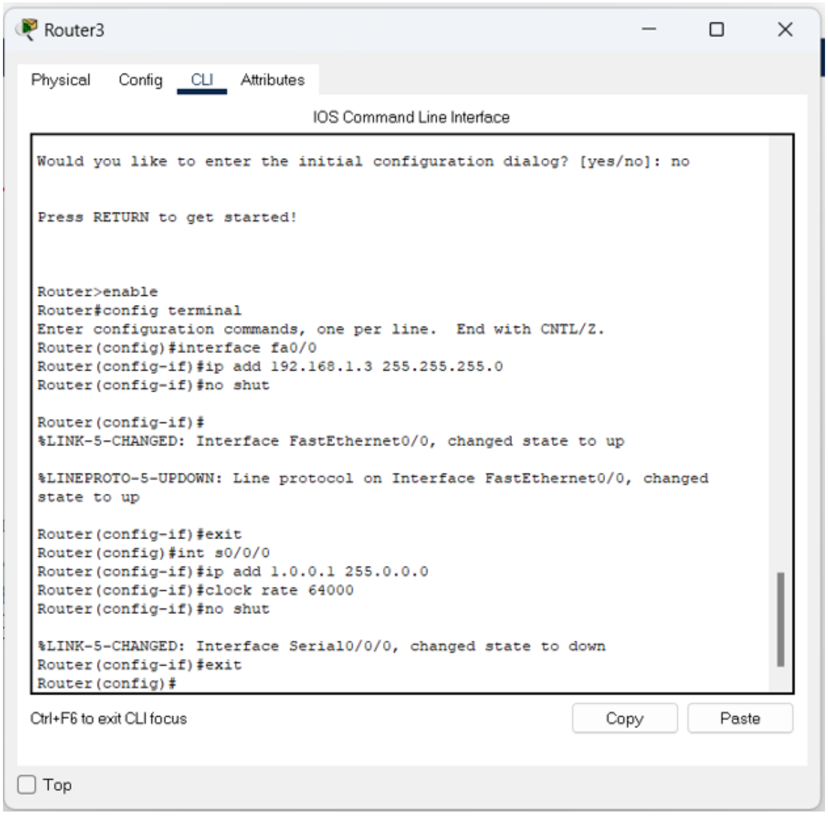
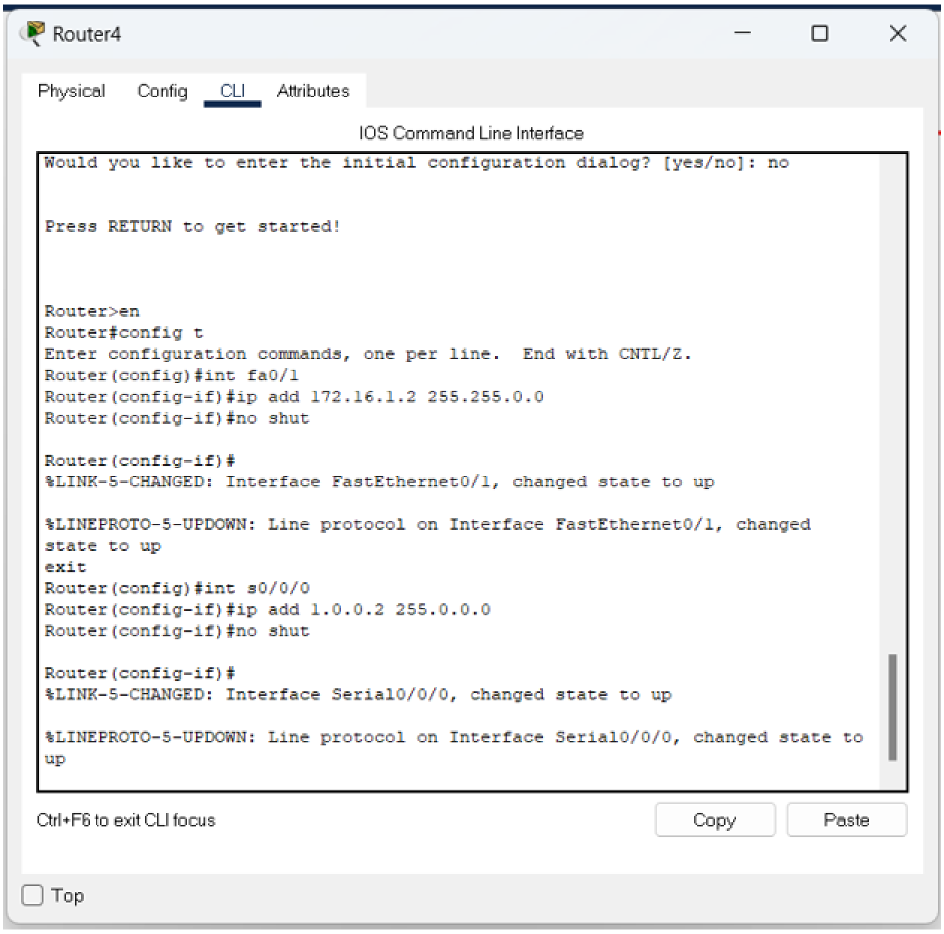


Figure 6.4: Router0 configuration

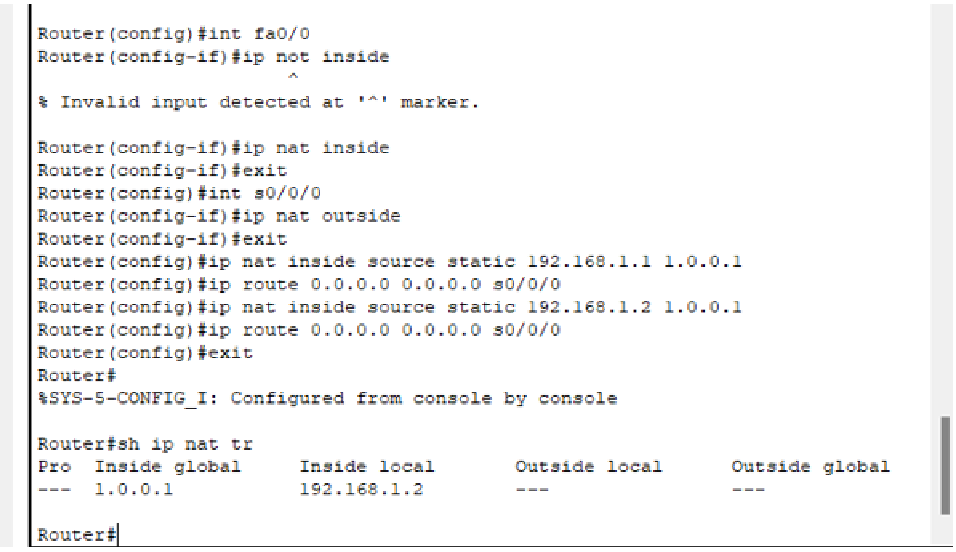
Step 5: Router 1 configuration.




```
Would you like to enter the initial configuration dialog? [yes/no]: no


Press RETURN to get started!



Router>en
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int fa0/1
Router(config-if)#ip add 172.16.1.2 255.255.0.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed
state to up
exit
Router(config)#int s0/0/0
Router(config-if)#ip add 1.0.0.2 255.0.0.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up

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

Figure 6.5: Router1 configuration

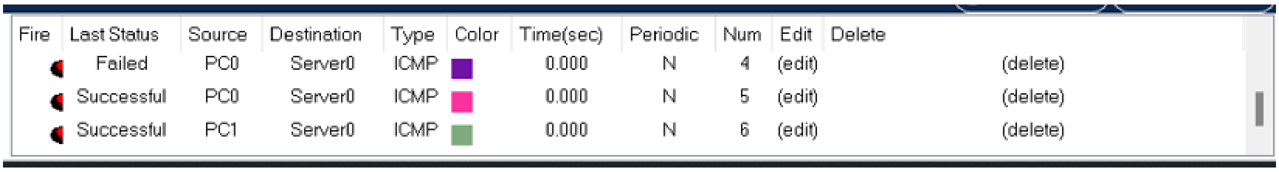
Step 6: Router's NAT command.



```
Router(config)#int fa0/0
Router(config-if)#ip not inside
                      ^
% Invalid input detected at '^' marker.

Router(config-if)#ip nat inside
Router(config-if)#exit
Router(config)#int s0/0/0
Router(config-if)#ip nat outside
Router(config-if)#exit
Router(config)#ip nat inside source static 192.168.1.1 1.0.0.1
Router(config)#ip route 0.0.0.0 0.0.0.0 s0/0/0
Router(config)#ip nat inside source static 192.168.1.2 1.0.0.1
Router(config)#ip route 0.0.0.0 0.0.0.0 s0/0/0
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#sh ip nat tr
Pro  Inside global     Inside local       Outside local      Outside global
---  1.0.0.1           192.168.1.2        ---                ---

Router#
```

Figure 6.6: NAT command

Step 7: Now transfer a packet from one PC to another it will get successfully transferred.



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Failed	PC0	Server0	ICMP		0.000	N	4	(edit)	(delete)
	Successful	PC0	Server0	ICMP		0.000	N	5	(edit)	(delete)
	Successful	PC1	Server0	ICMP		0.000	N	6	(edit)	(delete)

Figure 6.7: Successful transfer

Conclusion/Summary:-

Student Signature	Marks	Examiner Signature

PRACTICAL 7

AIM:-	Date:- / /

Theory:-

Different Cables and Connectors:

Sr No.	Components Name & Picture	Description of Component
1	Ethernet Cables 	An ethernet cable carries the broadband signals between modems, routers, computer, and other wired internet capable devices. It is used for high speed wired network between the two or more devices.
2	Clamping Tool 	Clamping tool is a good helper for crimping & terminating connectors. Helps in building a solid connection between connectors like RJ-45, RJ-11, etc. to the ethernet cables.
3	Console Cable 	Console cable connects network devices to terminals or PCs for the configuration process. It is also used to connect projectors to the PC.

4	Cable Tester 	Used to verify electrical connectivity of a cable. It has two components, namely:- • Master • Remote
5	Manageable Switch 	Allows ethernet devices to communicate with each other. It contains features to configure, manage and monitor traffic on a LAN . It is a smart device.
6	Normal Switch 	Allows devices to share and transfer data, enabling communication between each device on a network. It is an intelligent device.
7	Router 	A device that connects two or more packet switched networks or subnetworks. There are many types of routers based on its bandwidth capability such as 100Mbps, 300Mbps, 1Gbps.

8	Chromecast 	A streaming media adapter from google that allows users to play online content on a digital TV.
9	Passive Extender 	Passive extender is used to extend HDMI audio/video signal up to 100ft via single cat5e or cat6 cable.
10	Active Extender 	It is a device used to extend an ethernet or network segment beyond its inherit distance limitation. (approx. 100 meters)
11	Converter 	A networking device that connects two different media, like ethernet copper and ethernet fiber.

12	Normal ethernet extender 	Ethernet extender is a device used to extend an ethernet or network segment beyond its inherit distance limitation.
13	Passive Splitter 	A passive splitter is a device used to split the cable signal to two or more devices.
14	USB to Ethernet Cable 	A device that is capable of connecting a USB type-A to an ethernet cable.
15	POE Splitter 	POE splitter supplies power to non-POE-compatible devices by splitting power from data feeding it to a separate input.

16	NIC Card for optical fiber 	NIC card for optical fiber is usually a separate adapter card that can be inserted into one of the server's motherboard expansion slots.
17	NIC Card for wireless 	NIC card for wireless allows a device to connect a desktop to a wireless network and access a high-speed internet connection.
18	NIC Card for wireline 	NIC card for wireline allows one to establish communication between two devices using ethernet as a medium.
19	Faceplate Punch Tool 	This tool is used to insert individual wire into the cat5 OTP modules on wall-plates and patch panels.

20	Rack 	Rack is structure that holds computer servers or networking equipment usually by means of shelves or mounting plates.

Conclusion/Summary:-

Student Signature	Marks	Examiner Signature

PRACTICAL 8

AIM:-	Date:- / /

Theory:-

P2P (Peer to peer) TOPOLOGY:

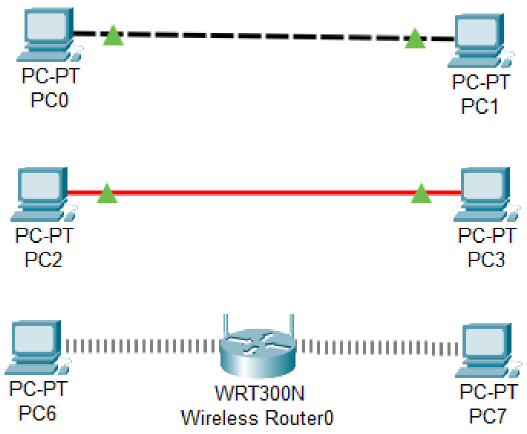


Figure 8.1: Peer to Peer Topology

Steps:

1. Place 2 end devices i.e. PCs, side by side and connect them via C.O. wire.
2. Now configure both the PCs ip address to 80.0.0.1 & 80.0.0.2 respectively by double clicking on each device -> config.
3. Now repeat the same process for optical fiber, but this time before configuring, change the physical component from CFE to FFE or FGE and connect both the PCs using optic fiber wire.
4. Do the same for wireless configuration using W300N and use a router (wireless) like WRT300N.
5. After configuring all these connections, a green triangle will appear on both sides which means the connections were successful.
6. Now double click on any PC go to Desktop then command prompt then type “ping 80.0.0.X “ and hit enter.

STAR TOPOLOGY:

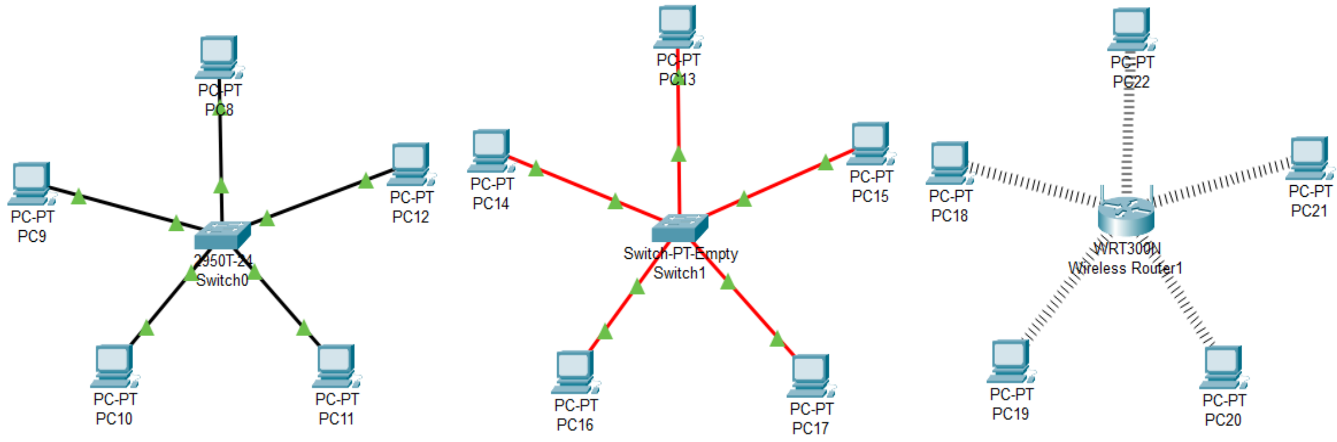


Figure 8.2: Star Topology

Steps:

1. Place 5 PCs in a star like formation and place a switch in the middle possibly, 2950T-24.
2. Now connect all the PCs to the switch via straight-through wire and configure all the PCs ip address.
3. Repeat the same process, but for optical fiber, alter the physical component of all the PCs i.e. CFE -> FGE. Connect all the PCs to the switch using optical fiber wires.
4. Do the same for wireless using a WRT300N wireless router. Configure all the PCs components by altering the CFE port to W300N port & configure all their IP addresses from DHCP to Static.
5. After all the connections are done, you will see green triangles between each connection.
6. Now ping any PC by double clicking on any PC then go to desktop then command prompt then type “ping 80.0.0.X” then hit enter.

RING TOPOLOGY:

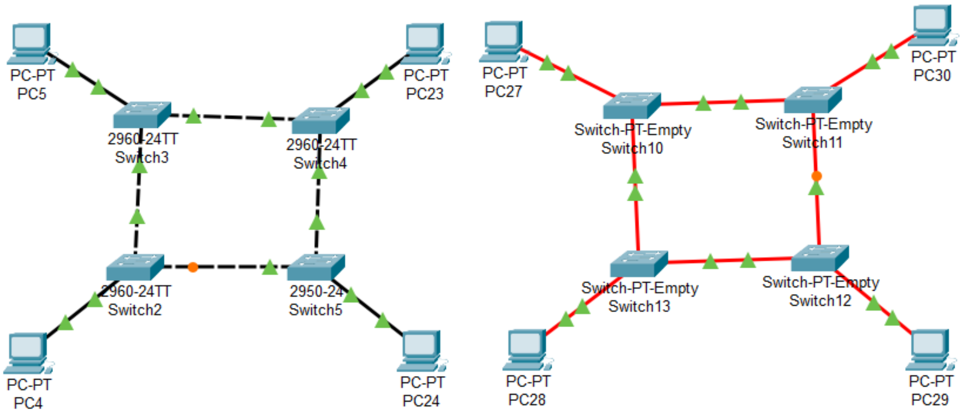


Figure 8.3: Ring Topology

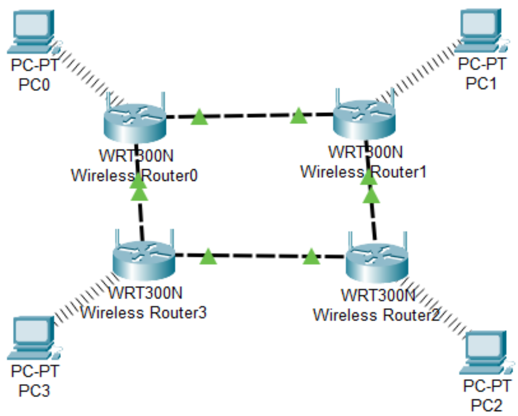


Figure 8.4: Ring Topology

Steps:

1. Place 4 PCs and 4 switches near each PC as sown in the figure and connect all the switches using cross-over wire and connect all the PCs to their corresponding switches via straight through cable.
2. Now configure all the PCs IP address.
3. Repeat the same steps for optical fiber but instead use switch-PT and insert 3 FQE ports in each side and replace the CPE port in all the PCs with FGE ports & connect them all using optic fiber cable.
4. In case of wireless config, place 4 WRT300N routers side by side as shown in the figure and connect all the routers in ring like structure via cross-over cable and configure the physical component of each PC with W300N and give them static IP address.
5. Now try to ping any PC by double click -> Desktop -> CMD -> “ping 80.0.0.X”.

MESH TOPOLOGY:

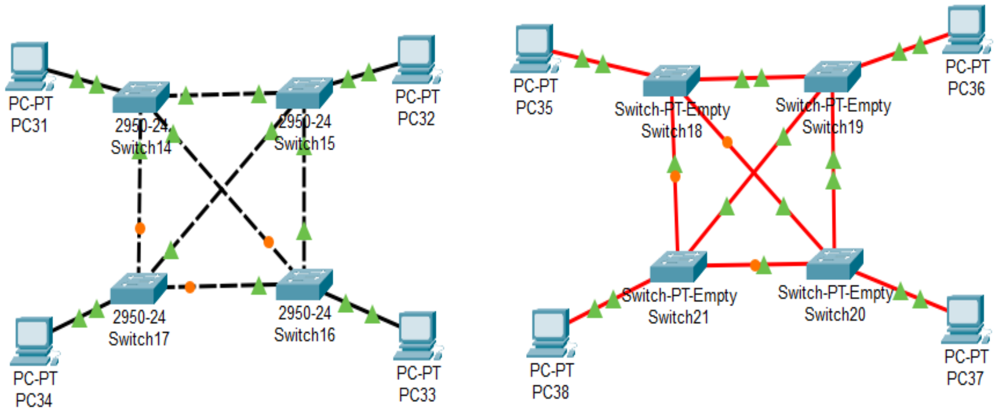


Figure 8.5: Mesh Topology

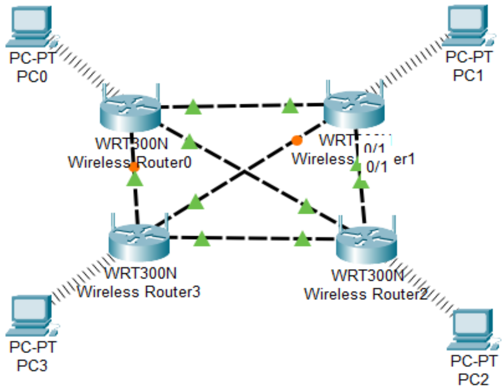


Figure 8.6: Mesh Topology

Steps:

1. Place 4 switches in a ring like structure, followed by placing 4 PCs near to each corresponding switch.
2. Now connect each switch to every other switch using cross-over cables. Now connect
   all the PCs to their corresponding switch
3. Now configure all the PCs ip configuration.
4. Repeat the same process for optical fiber, but this time use Switch PT-Empty and replace the CFE port manually add 4 FGE portsto all the switches and replace the CFE port in PCs to FGE port, and use opticalfiber cables.
5. For wireless, place 4 to routersside-by-side as shown in the figure, connect each router to every other router using cross-over cable then place 4 PCs, near each router place a PC
6. Alter the connections are made, green triangles will appear success. Now ping any PC by double click ->desktop -> cmd ->'ping 80.0.0.X" -> enter.

TREE TOPOLOGY:

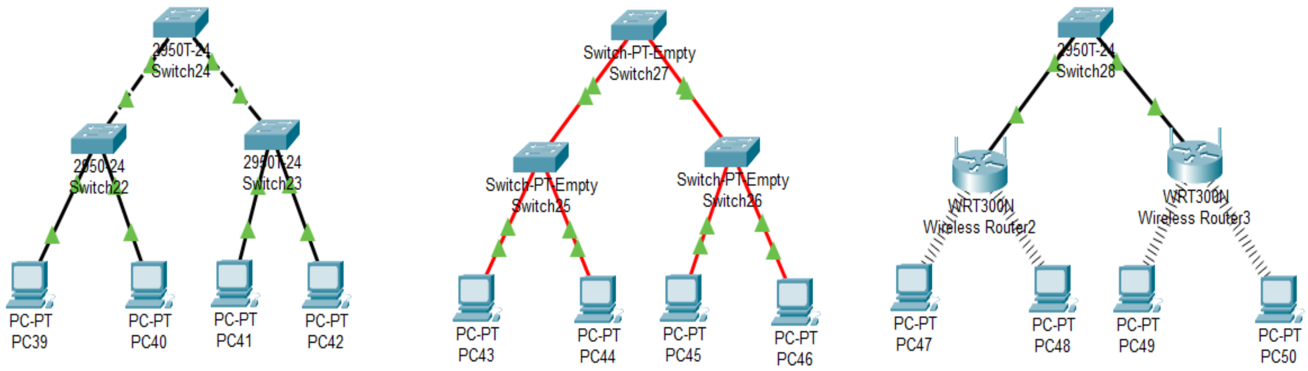


Figure 8.7: Tree Topology

Steps:

1. Place 3 switches in a triangle like shape and connect them as shown in the figure using cross over cable. Followed by place 4 PCs such that both the switches have two PCs never then as shown using straight-through cable.
2. Now configure all the PCs IP address.
3. Repeat the same for optical fiber but this time use switch-PT-empty and manually add 3 FGE ports in each switch and replace the CFE ports with FGE ports in all PCs use optical fiber cable for connections.
4. For wireless, place a switch from which 2 wireless routers are connected via straight-through cables, now place 2 PCs each near both wireless routers as shown.
5. Replace the CFE port with W300N port in all 4 PCs. They will get connected to their corresponding routers.
6. After all the connections, green triangle will appear -> success. Now ping other PCs.

Conclusion/Summary:-

Student Signature	Marks	Examiner Signature

PRACTICAL 9

AIM:-	Date:- / /

Theory:-

Exercise 1:

```
// Default Network Topology
//
//       80.1.1.0
// n0 -------------- n1
//    point-to-point
//

using namespace ns3;

NS_LOG_COMPONENT_DEFINE("FirstScriptExample");

int main(int argc, char* argv[])
{
    CommandLine cmd(__FILE__);
    cmd.Parse(argc, argv);
```

Prepare point-to-point topology in NS-3
Explain the Pseudo code for the given example code.

Pseudo Code:

The next two lines of code in our script will create the ns-3 Node objects that will represent the computers in the simulation.

```
NodeContainer nodes;
nodes.Create (2);
```

The next three lines in the script are,

```
PointToPointHelperpointToPoint;
pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));
```

The first line,

```
PointToPointHelperpointToPoint;
```

instantiates a PointToPointHelper object on the stack. From a high-level perspective the next line,

```
pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
```

The final line,

```
pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));
```

tells the PointToPointHelper to use the value “2ms” (two milliseconds) as the value of the transmission delay of every point-to-point channel it subsequently creates.

The next two lines of code,

```
   Ipv4AddressHelper address;
address.SetBase ("80.1.1.0", "255.255.255.0");
```

allocating IP addresses from the network 10.1.1.0 using the mask 255.255.255.0 to define the allocatable bits. By default, the addresses allocated will start at one and increase monotonically, so the first address allocated from this base will be 10.1.1.1, followed by 10.1.1.2, etc.

The following lines of code are used to set up a UDP echo server application on one of the nodes we have previously created.

```
UdpEchoServerHelperechoServer (9);

ApplicationContainerserverApps = echoServer.Install (nodes.Get (1));
serverApps.Start (Seconds (1.0));
serverApps.Stop (Seconds (10.0));
```

The echo client application is set up in a method substantially similar to that for the server. There is an underlying UdpEchoClientApplication that is managed by an UdpEchoClientHelper.

```
UdpEchoClientHelperechoClient (interfaces.GetAddress (1), 9);
echoClient.SetAttribute ("MaxPackets", UintegerValue (1));
echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.)));
echoClient.SetAttribute ("PacketSize", UintegerValue (1024));

ApplicationContainerclientApps = echoClient.Install (nodes.Get (0));
clientApps.Start (Seconds (2.0));
clientApps.Stop (Seconds (10.0));
```

Below four lines schedules server and client application,

```
serverApps.Start (Seconds (1.0));
serverApps.Stop (Seconds (10.0));
  ...
clientApps.Start (Seconds (2.0));
clientApps.Stop (Seconds (10.0));
```

We actually scheduled events in the simulator at 1.0 seconds, 2.0 seconds and two events at 10.0 seconds. When Simulator::Run is called, the system will begin looking through the list of scheduled events and executing them. First it will run the event at 1.0 seconds, which will enable the echo server application (this event may, in turn, schedule many other events). Then it will run the event scheduled for t=2.0 seconds which will start the echo client application. Again, this event may schedule many more events. The start event implementation in the echo client application will begin the data transfer phase of the simulation by sending a packet to the server.

Output of First.cc in NS 3:-

12 Mbps Data rate
1ms Delay

```
Scanning dependencies of target scratch_first
[  0%] Building CXX object scratch/CMakeFiles/scratch_first.dir/first.cc.o
[  0%] Linking CXX executable ../../build/scratch/ns3.37-first-default
At time +2s client sent 1024 bytes to 80.1.1.2 port 9
At time +2.0017s server received 1024 bytes from 80.1.1.1 port 49153
At time +2.0017s server sent 1024 bytes to 80.1.1.1 port 49153
At time +2.00341s client received 1024 bytes from 80.1.1.2 port 9
ritesh@ritesh-VirtualBox:~/Downloads/ns-allinone-3.37/ns-3.37$
```

5 Mbps Data Rate
2ms Delay

```
Scanning dependencies of target scratch_first
[  0%] Building CXX object scratch/CMakeFiles/scratch_first.dir/first.cc.o
[  0%] Linking CXX executable ../../build/scratch/ns3.37-first-default
At time +2s client sent 1024 bytes to 80.1.1.2 port 9
At time +2.00369s server received 1024 bytes from 80.1.1.1 port 49153
At time +2.00369s server sent 1024 bytes to 80.1.1.1 port 49153
At time +2.00737s client received 1024 bytes from 80.1.1.2 port 9
ritesh@ritesh-VirtualBox:~/Downloads/ns-allinone-3.37/ns-3.37$
```

Exercise 2:

```
// Default Network Topology
//
//       80.1.1.0
// n0 -------------- n1   n2   n3   n4
//    point-to-point  |    |    |    |
//                    ================
//                      LAN 80.1.2.0

```

Pseudo Code:

NodeContainer p2pNodes;

p2pNodes.Create(2);

PointToPointHelperpointToPoint;

pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));

pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));

NetDeviceContainer p2pDevices;

p2pDevices = pointToPoint.Install (p2pNodes);

CsmaHelpercsma;

csma.SetChannelAttribute ("DataRate", StringValue ("100Mbps"));

csma.SetChannelAttribute ("Delay", TimeValue (NanoSeconds (6560)));

Ipv4AddressHelper address;

address.SetBase ("80.1.1.0", "255.255.255.0");

Ipv4InterfaceContainer p2pInterfaces;

p2pInterfaces = address.Assign (p2pDevices);

address.SetBase ("80.1.2.0", "255.255.255.0");

Ipv4InterfaceContainer csmaInterfaces;

csmaInterfaces = address.Assign (csmaDevices);

UdpEchoClientHelperechoClient (csmaInterfaces.GetAddress (nCsma), 9);

echoClient.SetAttribute ("MaxPackets", UintegerValue (1));

echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.)));

echoClient.SetAttribute ("PacketSize", UintegerValue (1024));

Output:

12 Mbps Data Rate
1ms Delay

```
Scanning dependencies of target scratch_second
[  0%] Building CXX object scratch/CMakeFiles/scratch_second.dir/second.cc.o
[  0%] Linking CXX executable ../../build/scratch/ns3.37-second-default
At time +2s client sent 1024 bytes to 80.1.2.4 port 9
At time +2.01096s server received 1024 bytes from 80.1.1.1 port 49153
At time +2.01096s server sent 1024 bytes to 80.1.1.1 port 49153
At time +2.01992s client received 1024 bytes from 80.1.2.4 port 9
ritesh@ritesh-VirtualBox:~/Downloads/ns-allinone-3.37/ns-3.37$
```

5 Mbps Data Rate
2ms Delay

```
Scanning dependencies of target scratch_second
[  0%] Building CXX object scratch/CMakeFiles/scratch_second.dir/second.cc.o
[  0%] Linking CXX executable ../../build/scratch/ns3.37-second-default
At time +2s client sent 1024 bytes to 80.1.2.4 port 9
At time +2.0118s server received 1024 bytes from 80.1.1.1 port 49153
At time +2.0118s server sent 1024 bytes to 80.1.1.1 port 49153
At time +2.02161s client received 1024 bytes from 80.1.2.4 port 9
ritesh@ritesh-VirtualBox:~/Downloads/ns-allinone-3.37/ns-3.37$
```

Conclusion/Summary:-

Student Signature	Marks	Examiner Signature

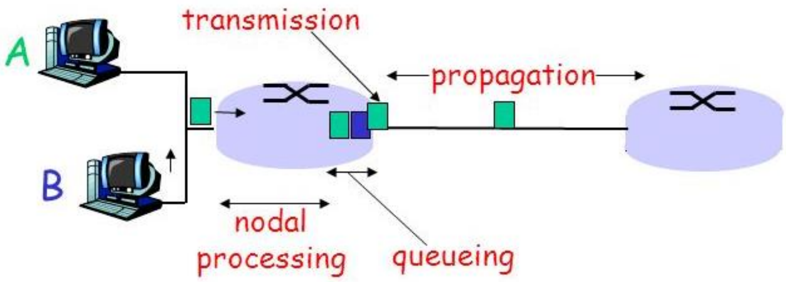
PRACTICAL 10

AIM:-	**Date:- / /**

Theory:-

Queueing Delay:

Let the packet is received by the destination, the packet will not be processed by the destination immediately. It has to wait in a queue in something called a buffer. So the amount of time it waits in queue before being processed is called queueing delay.



$$\boxed{d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}}$$

Propagation Delay:

After the packet is transmitted to the transmission medium, it has to go through the medium to reach the destination. Hence the time taken by the last bit of the packet to reach the destination is called propagation delay. (i.e. Distance/Speed)

```
======================================
 |      100kbps       |      50kbps       |
n0                   n1                  n2
```

Exercise 1: Prepare the above topology that we created in the cisco packet tracer, create the same in NS3.
Explain the Pseudo code for the given example code.

```
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
#include "ns3/netanim-module.h"
#include "ns3/flow-monitor-helper.h"
using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("myCode1");

int
main (int argc, char *argv[])
{
```

```
NodeContainerallNodes, nodes01, nodes12;
allNodes.Create(3);
nodes01.Add(allNodes.Get(0));
nodes01.Add(allNodes.Get(1));
nodes12.Add(allNodes.Get(1));
nodes12.Add(allNodes.Get(2));
```

```
PointToPointHelper p2pHelper01;
p2pHelper01.SetChannelAttribute ("Delay", StringValue ("0ms"));
p2pHelper01.SetQueue("ns3::DropTailQueue<Packet>", "MaxSize", QueueSizeValue
(QueueSize ("100p")));
p2pHelper01.SetDeviceAttribute ("DataRate", StringValue ("100kbps"));
```

```
PointToPointHelper p2pHelper12;
p2pHelper12.SetChannelAttribute ("Delay", StringValue ("0ms"));
p2pHelper12.SetQueue("ns3::DropTailQueue<Packet>", "MaxSize", QueueSizeValue (Q
ueueSize ("100p")));
p2pHelper12.SetDeviceAttribute ("DataRate", StringValue ("50kbps"));
```

```
NetDeviceContainer devices01;
devices01 = p2pHelper01.Install (nodes01);

NetDeviceContainer devices12;
devices12 = p2pHelper12.Install (nodes12);
```

```
InternetStackHelper stack;
stack.Install (allNodes);
Ipv4AddressHelper address;
```

```
address.SetBase ("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer interface01 = address.Assign (devices01);

address.SetBase ("10.1.2.0", "255.255.255.0");
Ipv4InterfaceContainer interface12 = address.Assign (devices12);
```

```
uint16_t port = 4000;
UdpServerHelper server (port);

ApplicationContainer apps = server.Install (nodes12.Get (1));
apps.Start (Seconds (1.0));
apps.Stop (Seconds (30.0));
```

```
uint32_t MaxPacketSize = 1000; // application packet size with out headers
Time interPacketInterval = Seconds (0.0);
uint32_t maxPacketCount = 200;

Address serverAddress = Address (interface12.GetAddress (1));
```

```
UdpClientHelper client (serverAddress, port);
client.SetAttribute ("MaxPackets", UintegerValue (maxPacketCount));
client.SetAttribute ("Interval", TimeValue (interPacketInterval));
client.SetAttribute ("PacketSize", UintegerValue (MaxPacketSize));
apps = client.Install (nodes01.Get (0));
apps.Start (Seconds (2.0));
apps.Stop (Seconds (20.0));
```

```
Ipv4GlobalRoutingHelper::PopulateRoutingTables ();


AnimationInterface::SetConstantPosition (nodes01.Get (0), 10.0, 20.0);
AnimationInterface::SetConstantPosition (nodes01.Get (1), 20.0, 20.0);
AnimationInterface::SetConstantPosition (nodes12.Get (1), 30.0, 20.0);
AnimationInterfaceanim ("myCode1_anim.xml");
```

```
Ptr<FlowMonitor>flowMonitor;
FlowMonitorHelperflowHelper;
flowMonitor = flowHelper.InstallAll();
p2pHelper01.EnablePcapAll ("myCode1");
```

```
Simulator::Stop(Seconds(40.0));
Simulator::Run();

// The following line should be put after Simulator::Run()
flowMonitor->SerializeToXmlFile("myCode1_flow.xml", true, true);

Simulator
```

Conclusion/Summary:-

Student Signature	Marks	Examiner Signature