

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



## LAB REPORT on COMPUTER NETWORKS

*Submitted by*

**Dipesh Sah**

*in partial fulfillment for the award of the degree of*  
**BACHELOR OF ENGINEERING**  
*in*  
**COMPUTER SCIENCE AND ENGINEERING**



**B.M.S. COLLEGE OF ENGINEERING**  
(Autonomous Institution under VTU)  
**BENGALURU-560019**  
**Sep 2024-Jan 2025**

**B. M. S. College of Engineering,  
Bull Temple Road, Bangalore 560019**  
(Affiliated To Visvesvaraya Technological University, Belgaum)  
**Department of Computer Science and Engineering**



**CERTIFICATE**

This is to certify that the Lab work entitled "**COMPUTER NETWORKS**" carried out by **Dipesh Sah (1BM22CS092)**, who is bonafide student of **B. M. S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2024-25. The Lab report has been approved as it satisfies the academic requirements in respect of **Computer Networks Lab - (23CS5PCCON)** work prescribed for the said degree.

**Sandhya A Kulkarni**

Assistant Professor,  
Department of CSE,  
BMSCE, Bengaluru

**Dr. Kavitha Sooda**

Professor and Head,  
Department of CSE  
BMSCE, Bengaluru

# **INDEX**

## **CYCLE 1**

<b>Sl. No.</b>	<b>Date</b>	<b>Experiment Title</b>	<b>Page</b>
1	25-09-2024	Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message.	1
2	9-10-2024	Single Router Configuration	10
3	16-10-2024	Dual Router Configuration	15
4	23-10-2024	Configure default route, static route to the Router	23
5	13-11-2024	Configure DHCP within a LAN and outside LAN	27
6	20-11-2024	Configure RIP routing Protocol in Routers	32
7	20-11-2024	Demonstrate the TTL/ Life of a Packet	35
8	27-11-2024	Configure OSPF routing protocol	38
9	18-12-2024	Configure Web Server, DNS within a LAN	44
10	18-12-2024	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)	46
11	18-12-2024	To understand the operation of TELNET by accessing the router in server room from a PC in IT office.	49
12	18-12-2024	To construct a VLAN and make the PC communicate among a VLAN	53
13	18-12-2024	To construct a WLAN and make the nodes communicate wirelessly	59

# **INDEX**

## **CYCLE 2**

<b>Sl. No.</b>	<b>Date</b>	<b>Experiment Title</b>	<b>Page</b>
1	1-1-2025	Write a program for congestion control using Leaky bucket algorithm.	62
2	1-1-2025	Write a program for error detecting code using CRC-CCITT (16-bits).	64
3	1-1-2025	Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.	68
4	1-1-2025	Using UDP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.	71
5	2-1-2025	Tool Exploration –Wireshark	74

# Computer Networks: Lab Record

## Week: 1

### Experiment 1: Hubs and Switches

Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message.

#### Observation:

Bafna Gold  
Date: \_\_\_\_\_  
Page: 3

9. Device - specific Selection Box : This box is where you choose specifically which devices you want to put in your network & which connections to make.

10. User Created Packet Window : This window manages the packets you put in the network during simulation scenarios.

#### # Experiment - 1

1. PC to Server.

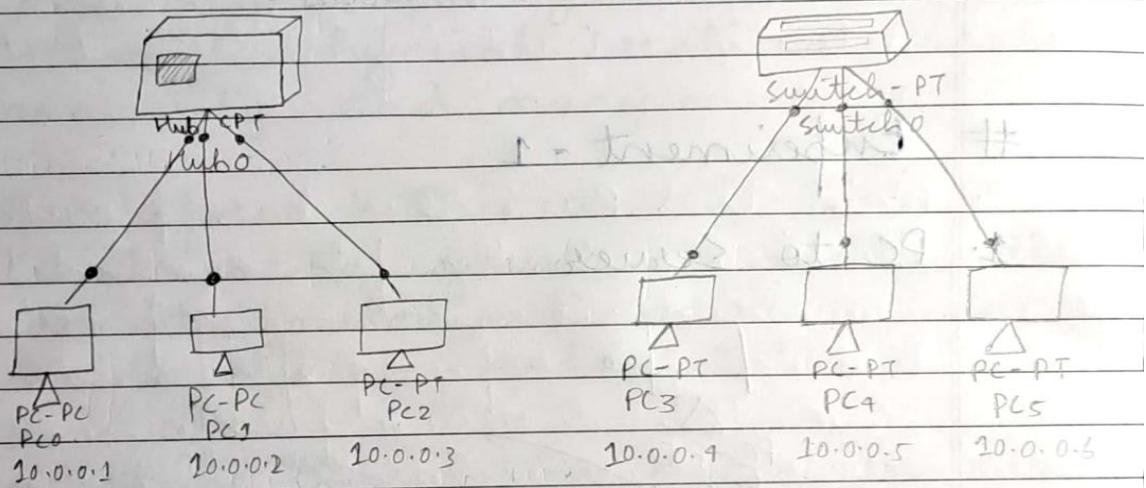
PC-PT  
PC0  
10.0.0.1

Server-PT  
Server 0  
10.0.0.2

- Aim : To set up a point - to - point network between a PC & a server, facilitating direct communication to observe data exchange.
- Topology : A PC (PC0) is connected to a server (server 0) using a cross-over ethernet cable.  
IP address of PC0 : 10.0.0.1  
IP address of server 0 : 10.0.0.2

- Observation  $\rightarrow$  The direct connection allows PC0 to communicate with server 0, which is typical in small networks for tasks such as file sharing, service requests or testing server responses to client queries.

## 2. Hub & switch



- Aim: To create a simple network that consists of three PCs connected to a central hub and another network with three PCs connected to a switch. This configuration will help observe the behaviour of data transmission using hub and switch devices.

- Topology:

- Hub Network: Three PCs (PC0, PC1, PC2) are connected to a hub (Hubo) using straight through Ethernet cables.

## IP Addresses :

 $PC_0 = 10 \cdot 0 \cdot 0 \cdot 1, PC_1 = 10 \cdot 0 \cdot 0 \cdot 2, PC_2 = 10 \cdot 0 \cdot 0 \cdot 2$ 

- ii) Switch Network : Three PCs ( $PC_3, PC_4, PC_5$ ) are connected to a switch (switch 0) using straight-through ethernet cables.

## IP Addresses :

 $PC_3 = 10 \cdot 0 \cdot 0 \cdot 4, PC_4 = 10 \cdot 0 \cdot 0 \cdot 5, PC_5 = 10 \cdot 0 \cdot 0 \cdot 6$ 

## Procedure :

- 1) Add 1 hub, 1 switch and 6 PCs ( $PC_0, PC_1, PC_2$ ) ~~for the hub~~ for the hub : ( $PC_3, PC_4, PC_5$ ) for the switch to the CISCO Packet Tracer workspace.
- 2) Use copper straight-through cables to connect  $PC_0, PC_1, PC_2$  to hub 0. Similarly connect  $PC_3, PC_4, PC_5$  to switch 0 using same type of cables.
- 3) Assign IP addresses to each PC and obtain subnet mask.
- 4) switch to simulation mode to observe data traffic behaviour when packets are sent between the devices.
- 5) In the hub network, observe how the hub broadcasts packets to all the devices, causing potential traffic overload.

In the switch network, observe how the switch forwards packets only to the intended recipient, reducing unnecessary traffic.

- 6. The hub broadcasts data to all the connected devices leading to more network congestion, while the switch efficiently sends data only to the correct device, optimizing performance.

#### # Observation :

1. The hub broadcasts packets to all devices, which may cause unnecessary traffic.
2. The switch forwards packets only to the appropriate device by learning MAC addresses, making it more efficient in reducing traffic.

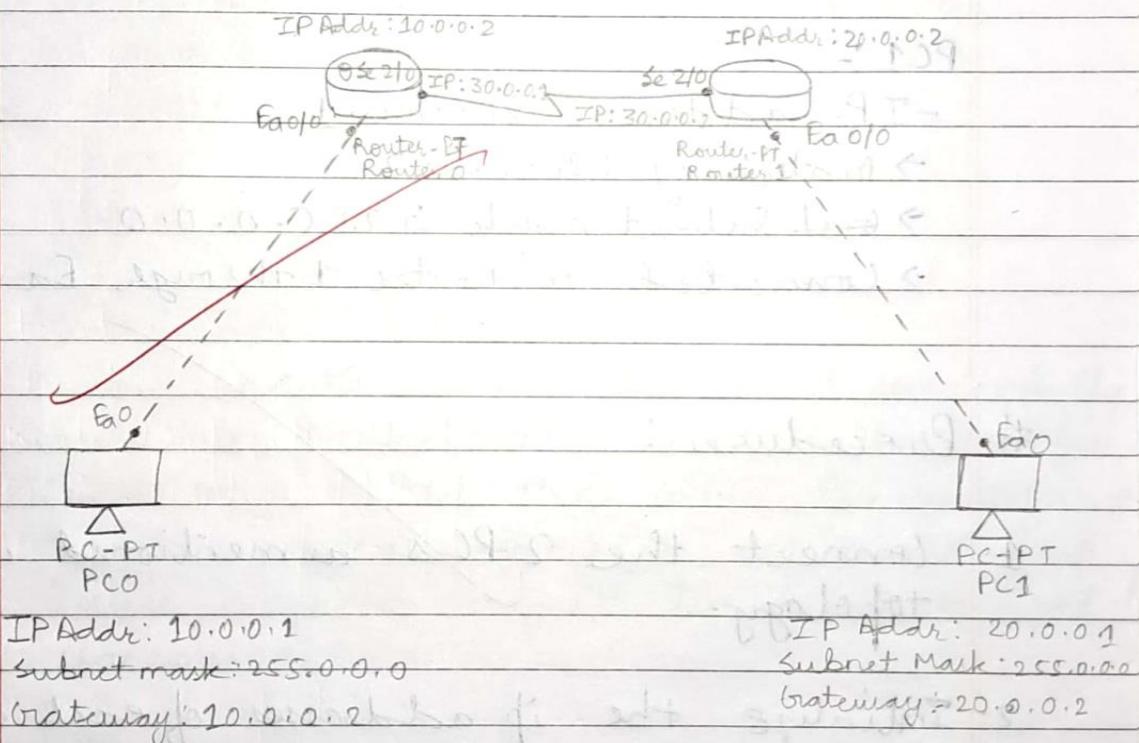
## Lab : 32(a)

# Aim : Configure IP Address to routers in Cisco Packet Tracer.

Explore the following messages :

- ① ping responses .
- ② destination unreachable
- ③ request timed out & reply .

# Topology :



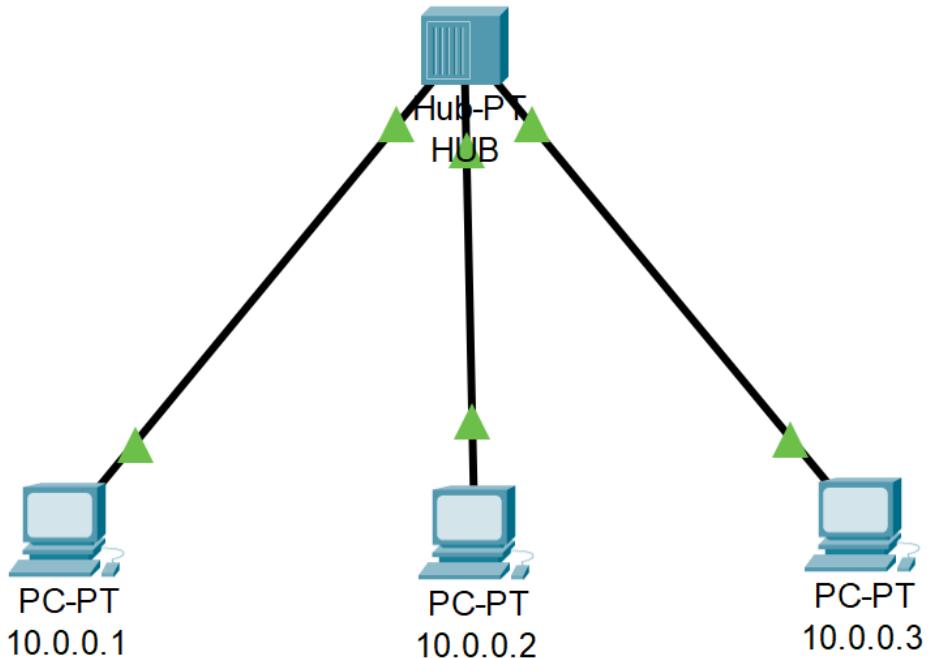
# Routers :

Router 0: Interface Fa2/0 connected to  
PC 0 : 10.0.0.2

Interface se 2/0 connected to  
Router 1: 30.0.0.1

connected to Router 1 & PC0 .

## Screenshots:



10.0.0.1

Physical Config Desktop Programming Attributes

Command Prompt

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time=20ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 20ms, Average = 5ms
```

```
C:\>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time=1ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

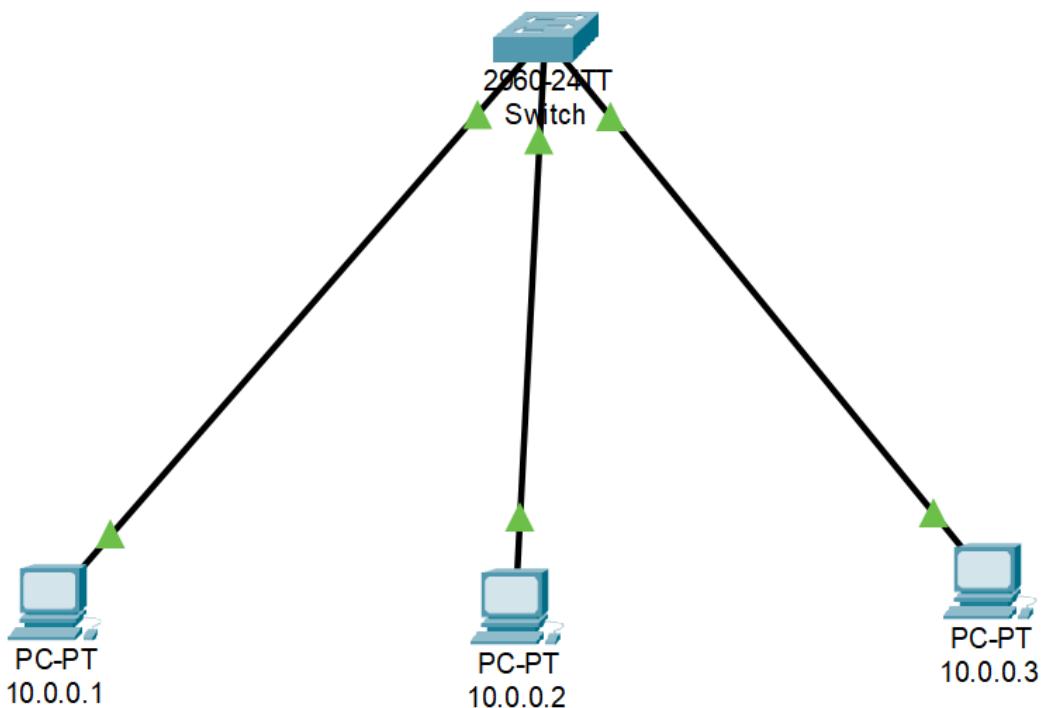
The screenshot shows the Cisco Packet Tracer interface. At the top, there is a search bar with the text "10.0.0.2". Below the search bar, there are five tabs: "Physical", "Config", "Desktop", "Programming", and "Attributes". The "Desktop" tab is currently selected and highlighted in blue. In the main area, there is a window titled "Command Prompt". Inside this window, the following command and its output are displayed:

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```



10.0.0.1

Physical    Config    Desktop    Programming    Attributes

Command Prompt

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
C:\>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time=1ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time=1ms TTL=128

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

10.0.0.2

Physical Config Desktop Programming Attributes

Command Prompt

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1: bytes=32 time<1ms TTL=128
Reply from 10.0.0.1: bytes=32 time=1ms TTL=128
Reply from 10.0.0.1: bytes=32 time=1ms TTL=128
Reply from 10.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

# Computer Networks: Lab Record

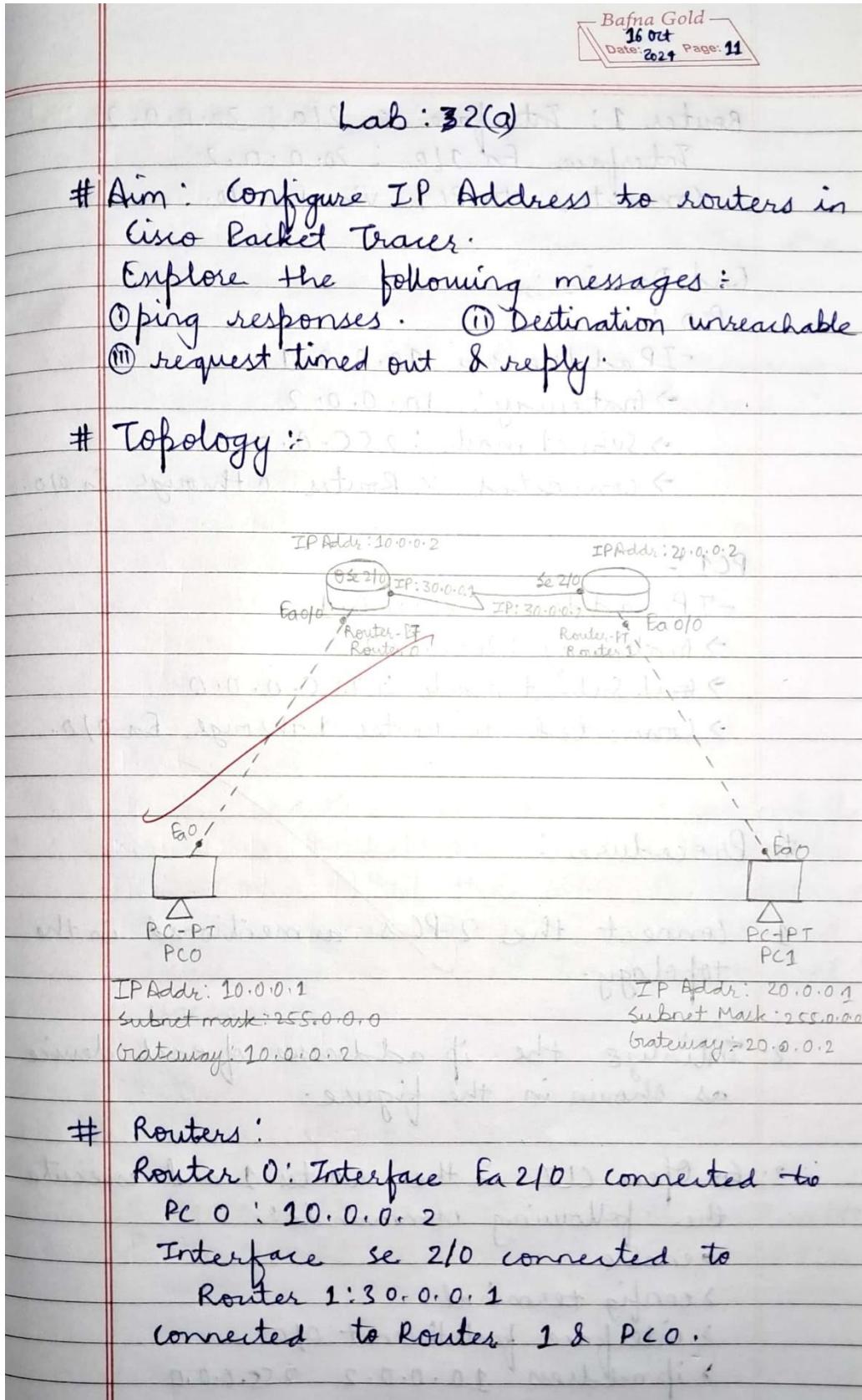
## Week: 3

### Experiment 3: Two Routers Configuration and Messages

Configure IP address to routers in packet tracer.

Explore the following messages: ping responses, destination unreachable, request timed out, reply.

**Observation:**



Router 1: Interface Se 2/0 : 30.0.0.2

Interface Fa 1/0 : 20.0.0.2

Connected to PC1 via Fa 0/0.

End Devices :-

PC0 :

- IP address : 10.0.0.1

→ Gateway : 20.0.0.2

→ Subnet mask : 255.0.0.0

→ Connected to Router 0 through Fa 0/0.

PC1 :-

- IP address : 20.0.0.1

→ Gateway : 20.0.0.2

→ Subnet mask : 255.0.0.0

→ Connected to Router 1 through Fa 0/0.

# Procedure :

1. Connect the 2 PCs as mentioned in the topology.

2. Initialize the ip addresses of each device as shown in the figure.

3. Open CLI in the Router 1 and execute the following commands :-

> enable

> config terminal

> interface fastethernet 0/0

> ip address 10.0.0.2 255.0.0.0

> no shut  
> exit

3. Repeat the steps to connect to the other PC to Router.

4. Open CLI again. Run the following commands to connect the routers together:

> interface serial 2/0  
> ip address 30.0.0.1 255.0.0.0  
> no shut  
> exit

5. Repeat the same for others

#### # Observation:

Two routers are connected successfully but the packets are unable to transfer from one PC to the other PC of different router. When pinged, we encountered issues saying request timed out or host unreachable.

#### # Output:

> show ip route

- C 10.0.0.0/8 is directly connected, FastEthernet0/0
- C 30.0.0.0/8 is directly connected, serial 2/0.

Experiment 2(b) continued from 2(a) :

Configure default route, static route to the router.

# Aim: successful transmission of packets from one PC to other of different routers.

Topology : Two PCs are connected to two different routers using copper cross over wires and those two routers are connected to each other ~~are~~ using serial DCE.

# Procedure:

1. After successful configuration of the two routers. Open CLI of one router and follow the commands to configure static routing.

> ip route 20.0.0.0 255.0.0.0 30.0.0.2

2. Open the CLI of other router.

> ip route 10.0.0.0 255.0.0.0 30.0.0.1

### # Observation :-

After configuration of static routing, the two PCs of two different router networks are able to transfer / transmit the packets using the ping command.

### # Output :-

> show ip route;

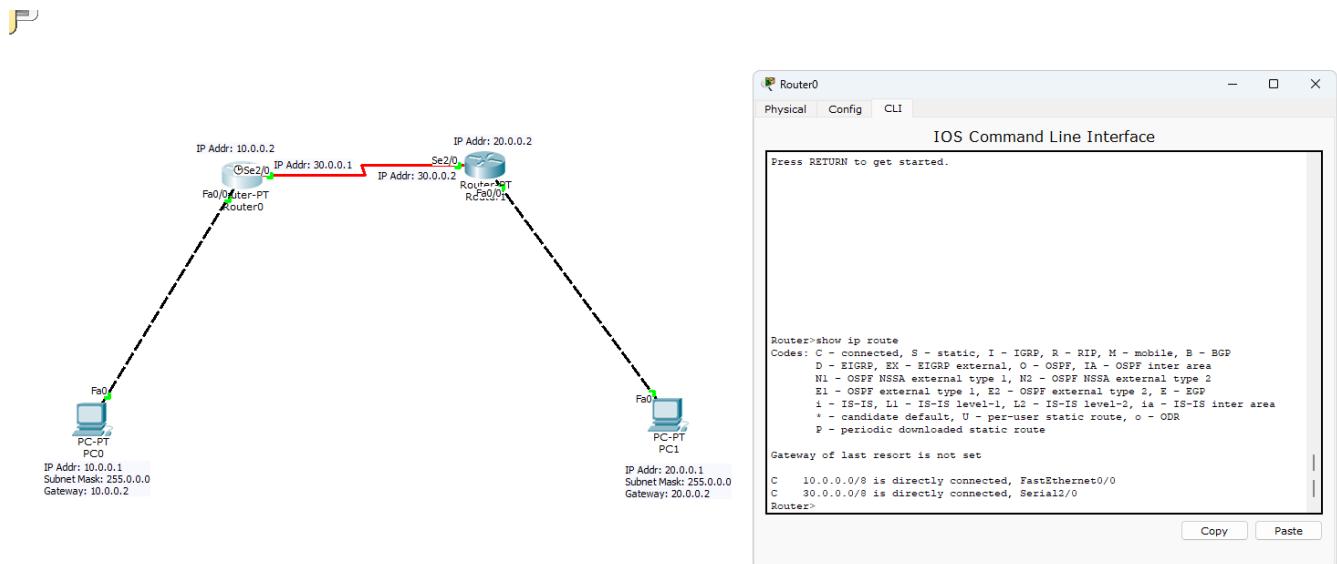
C 10.0.0.0/8 is directly connected, East Ethernet 0/0.

S 20.0.0.0/8 [1/0] via 30.0.0.2

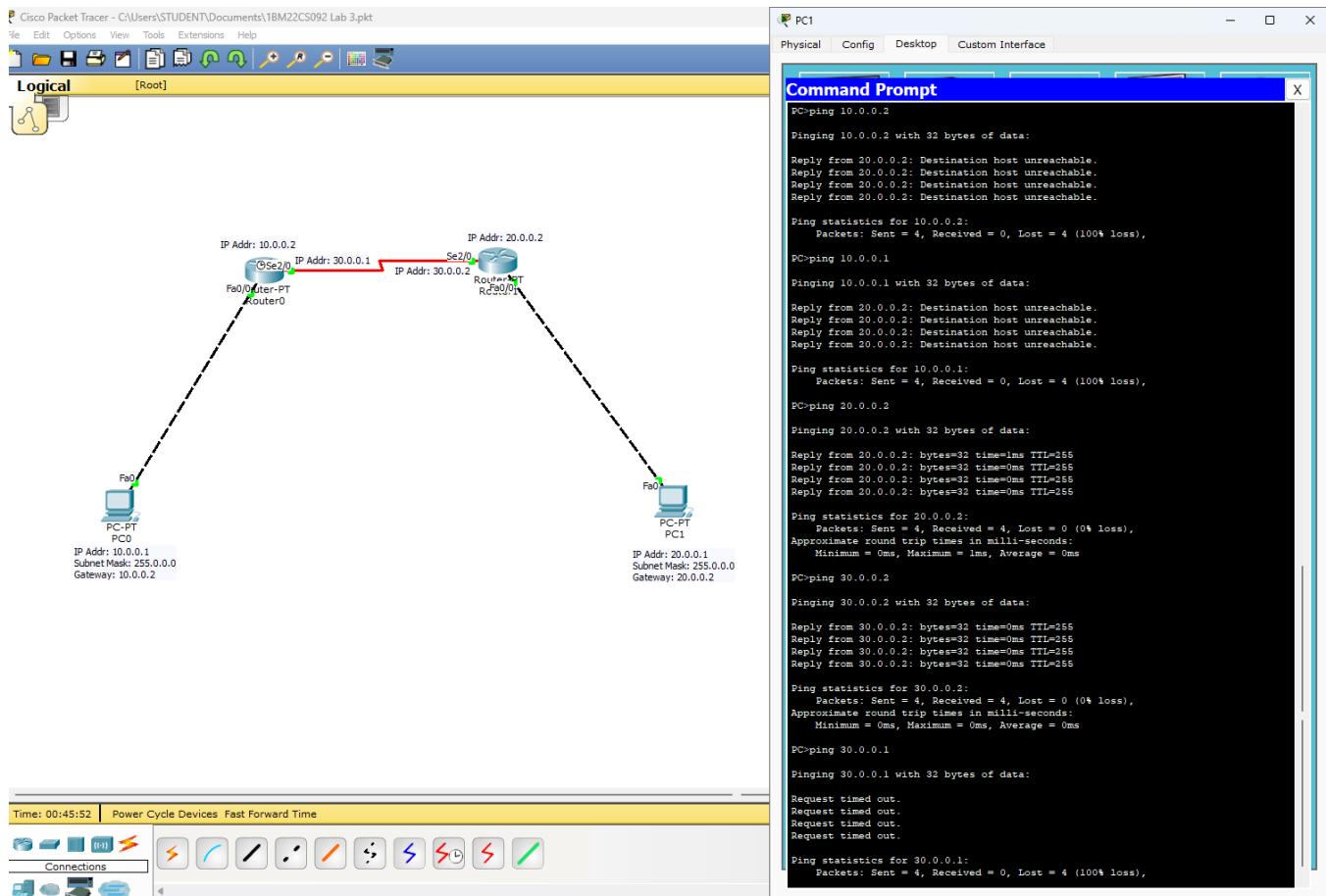
C 30.0.0.0/8 is directly connected, Serial 2/0.

## Screenshots:

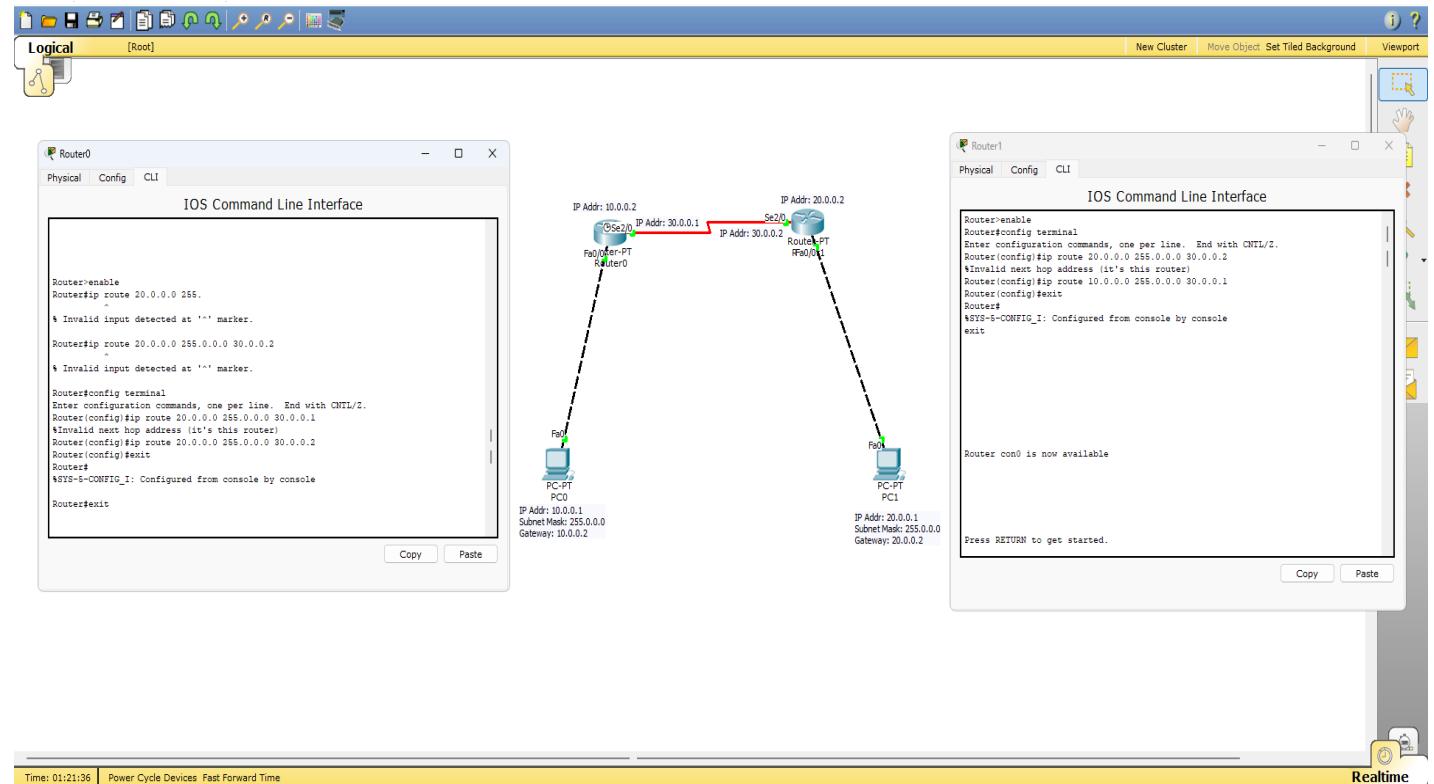
### Initial Setup



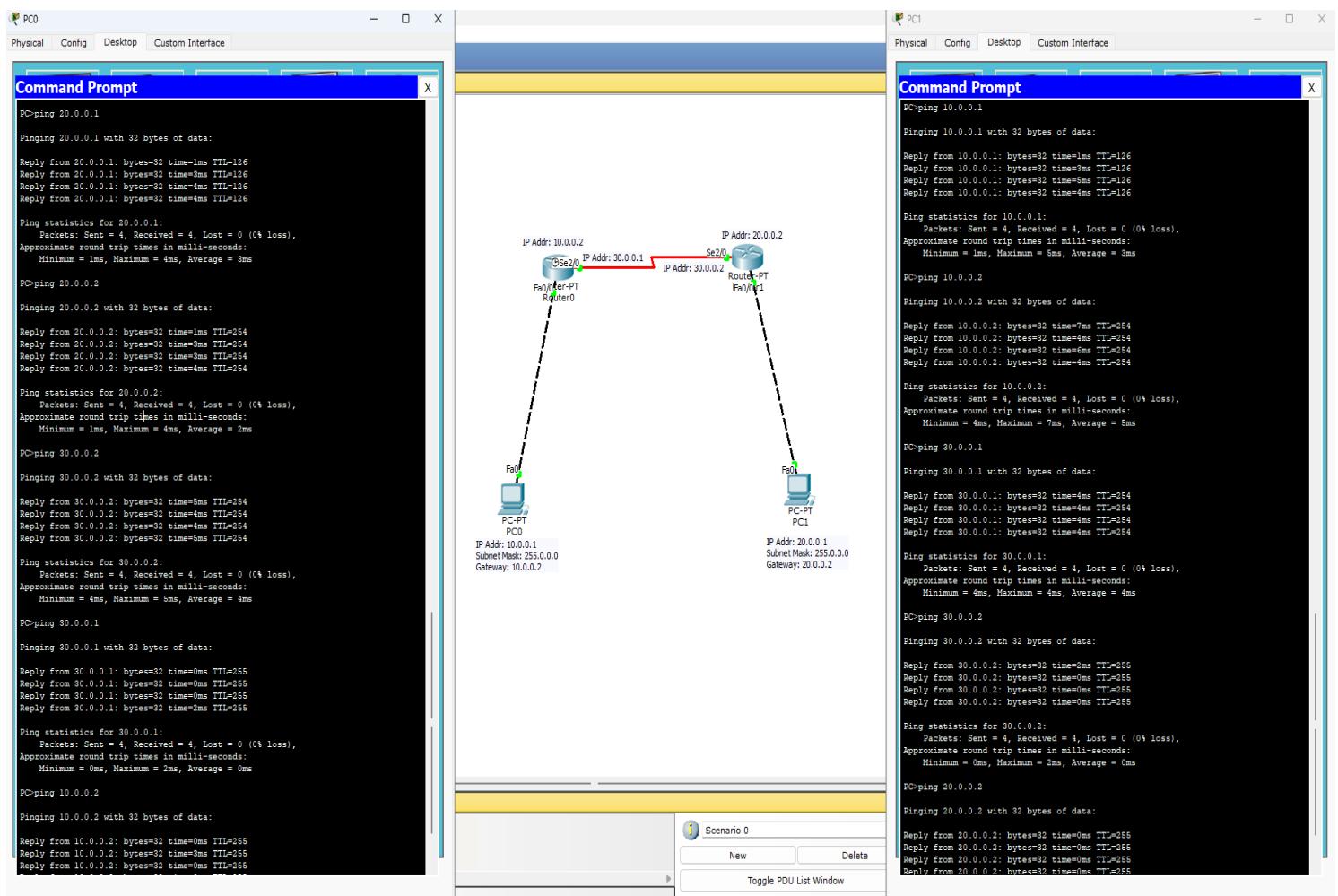
### Setup Done but no interconnectivity



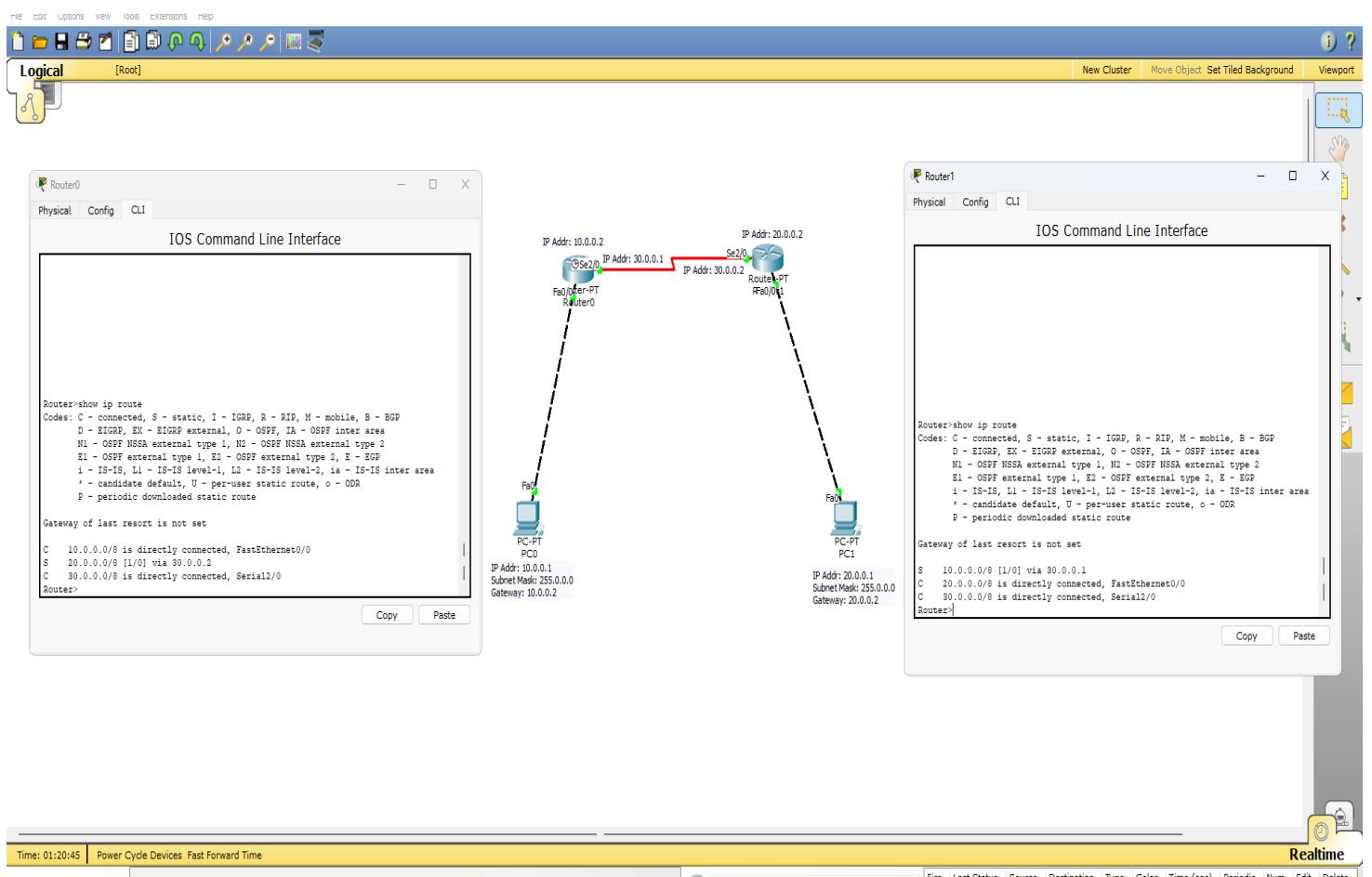
## Extra Configuration for Interconnectivity



### Successful Interconnectivity with 0% Loss while pinging



## Output for show IP Route



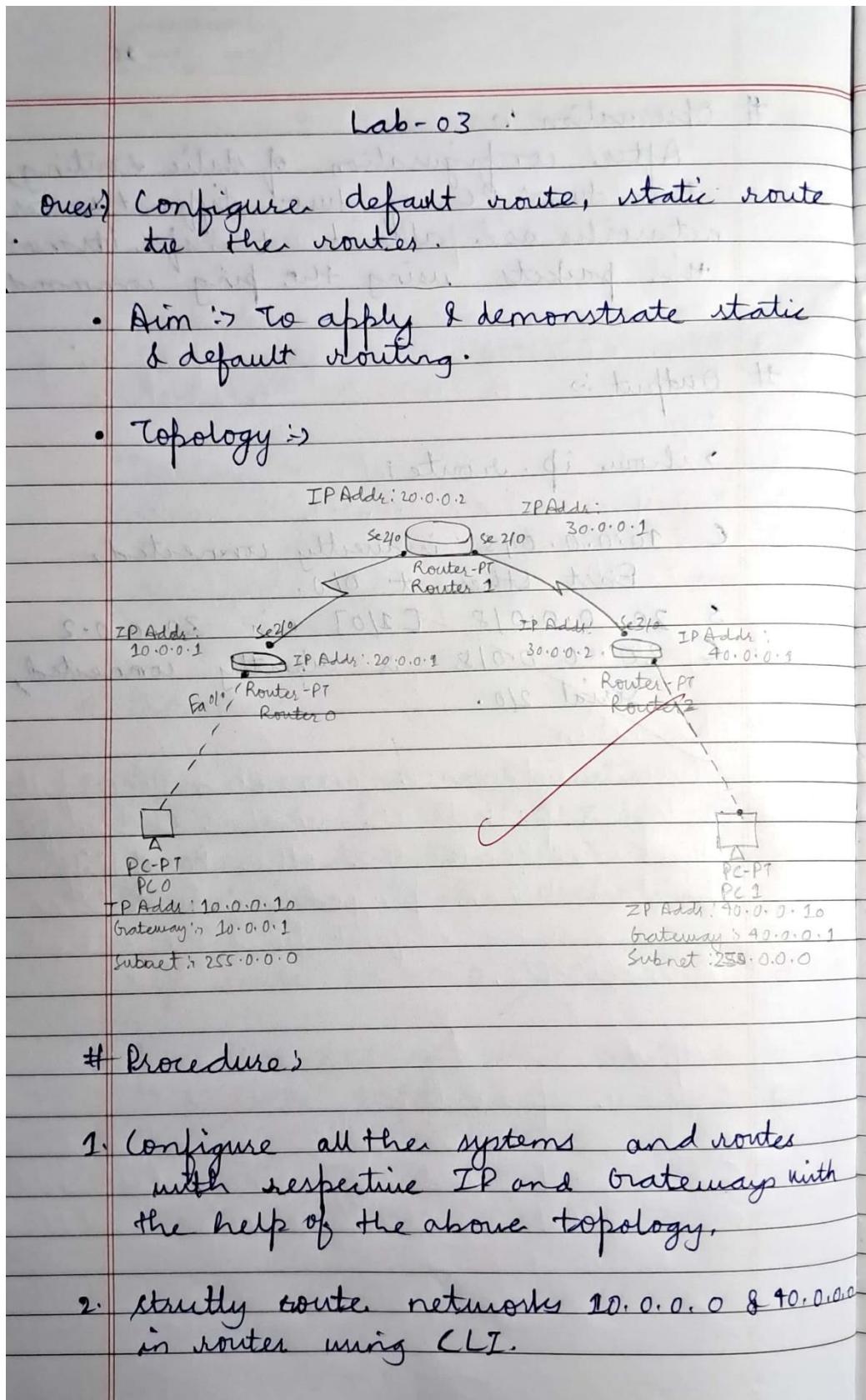
# Computer Networks: Lab Record

## Week: 4

### Experiment 4: Static Routing and Default Routing

Configure default route, static route to the Router to demonstrate Static and Default Routing.

#### Observation:



> ip route 10.0.0.0 255.0.0.0 20.0.0.1  
 > ip route 40.0.0.0 255.0.0.0 30.0.0.2

3 Execute commands on router 0 & router 2 as such to implement default routing.

~~(R0)~~ ip route 0.0.0.0 0.0.0.0 20.0.0.2  
~~(R2)~~ ip route 0.0.0.0 0.0.0.0 30.0.0.1

### # Observation :

Upon executing "show ip route" following observations are made with successful pinging on end devices with no connection errors.

R0 :

~~(R0)~~  
 C 10.0.0.0/8 is directly connected, FastEthernet0/0  
 C 20.0.0.0/8 is directly connected, serial 2/0  
 2) S\* 0.0.0.0 [1/0] via 20.0.0.2

R1 :

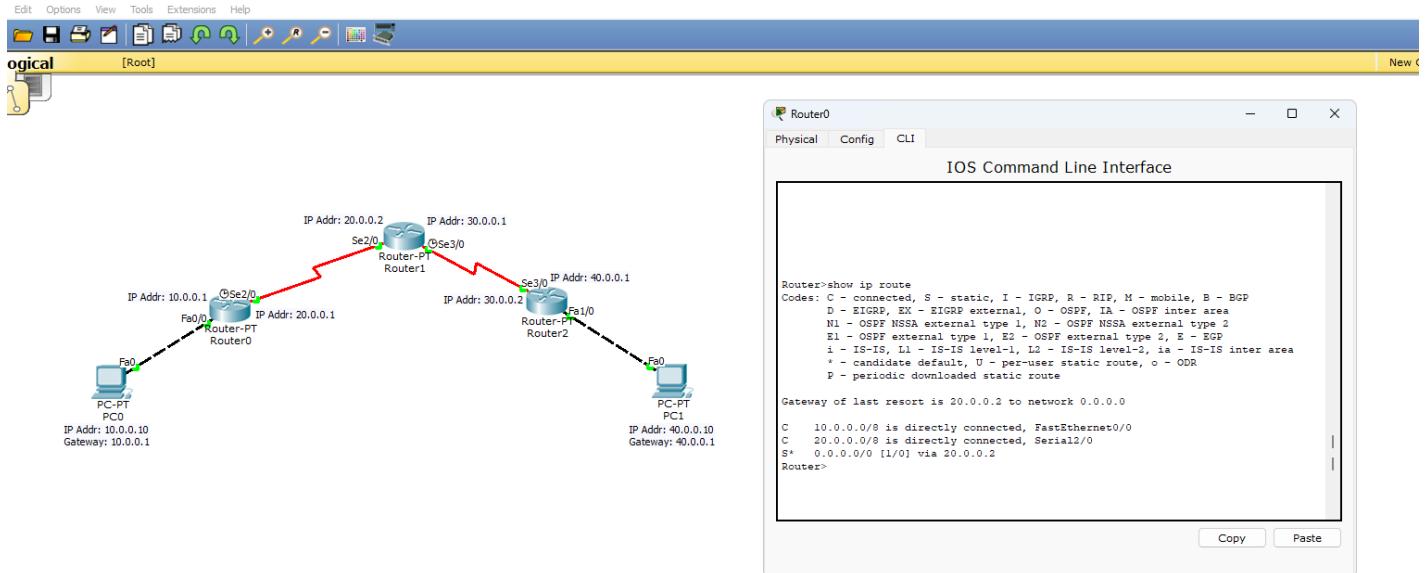
S 10.0.0.0/8 [1/0] via 20.0.0.1  
 C 20.0.0.0/8 is directly connected, serial 2/0  
 C 30.0.0.0/8 is directly connected, serial 3/0  
 S 40.0.0.0/8 [1/0] via 30.0.0.2

R2 :

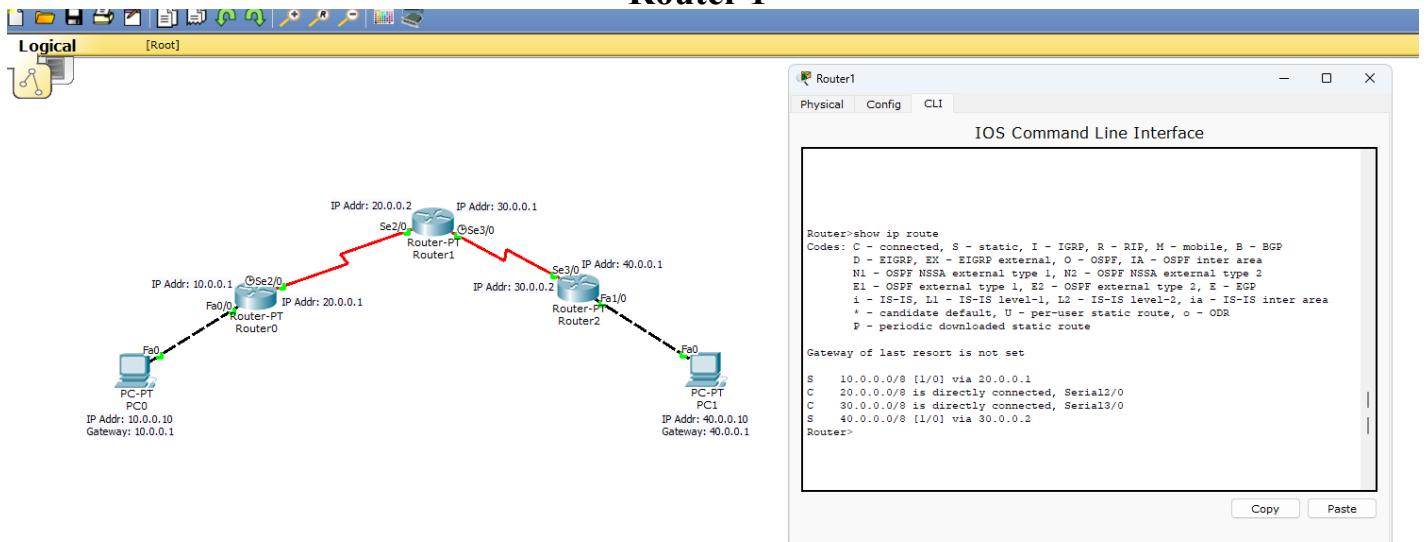
C 30.0.0.0/8 is directly connected, serial 3/0  
 C 40.0.0.0/8 is directly connected, FastEthernet0/0  
 S\* 0.0.0.0/8 [1/0] via 30.0.0.1

## Screenshots:

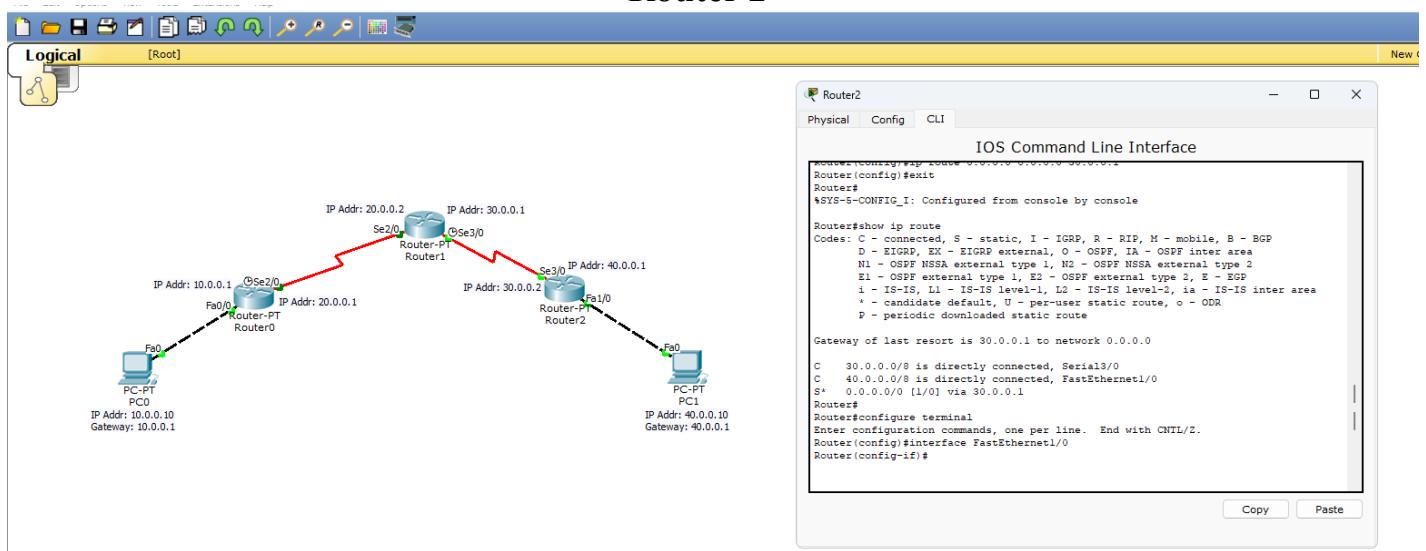
### Router 0



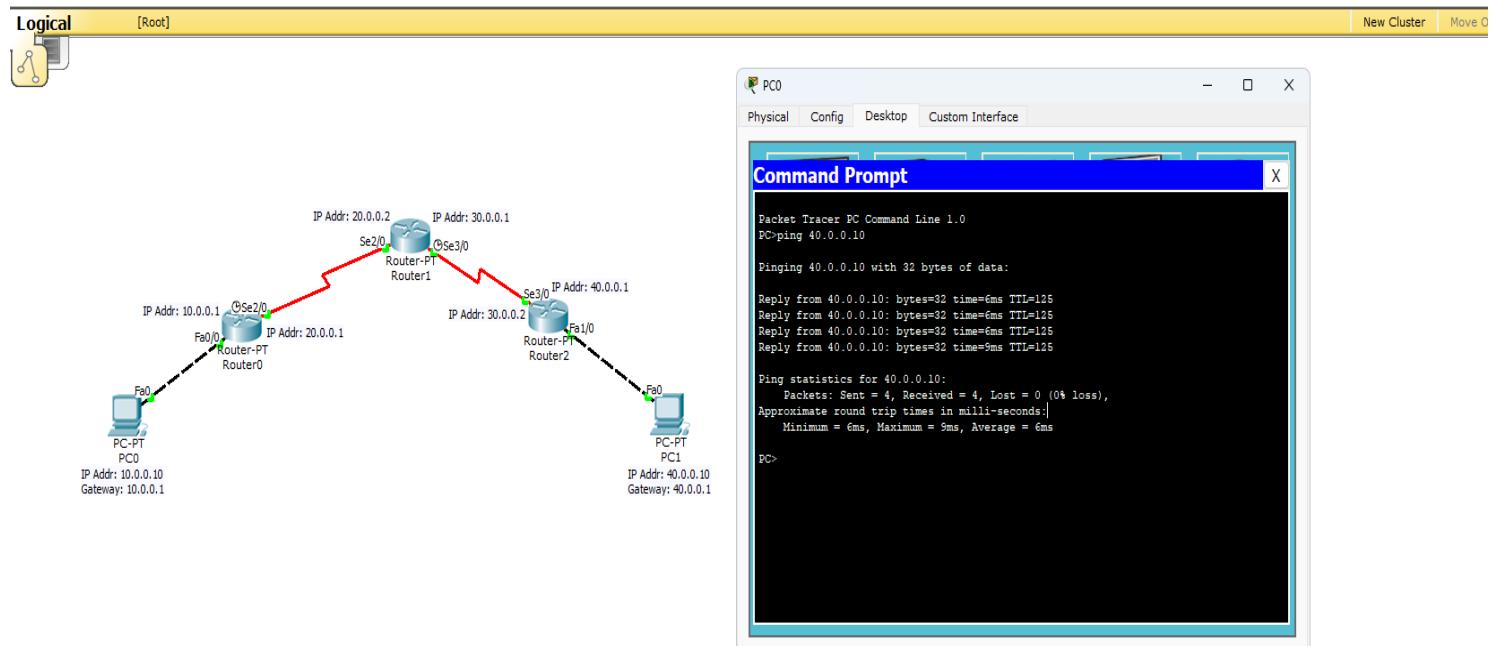
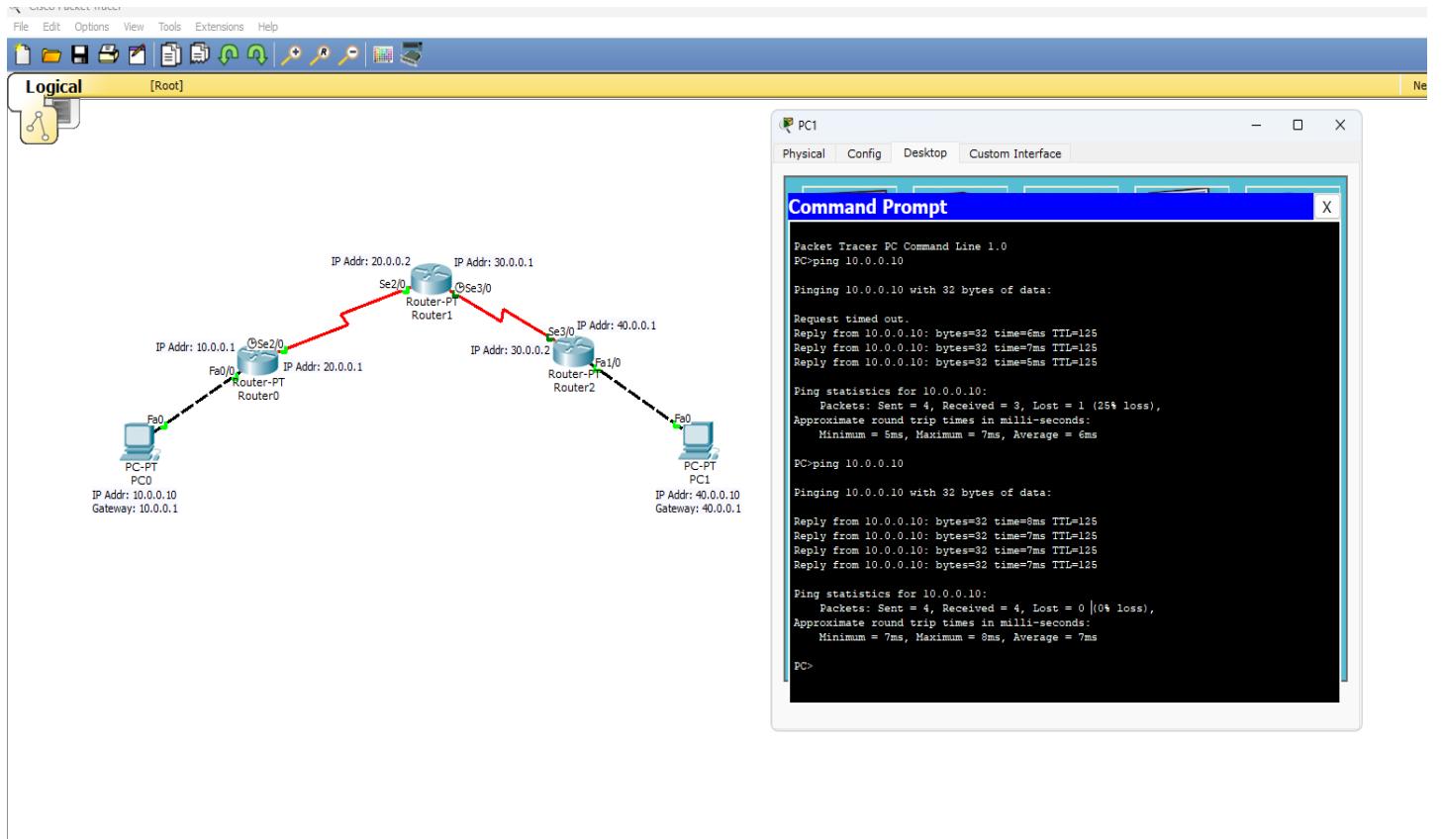
### Router 1



### Router 2



## After Successful Setup using Command Line Interface



## Program 5

Aim: Configure DHCP within a LAN and outside LAN.

Topology , Procedure and Observation:

Lab - 04

DHCP.

Aim :- Configure DHCP within a LAN and outside LAN.

① within the LAN.

- Topology

- Procedure:

1. choose a generic server, a switch and 2 PCs with a laptop and connect them to switch using auto cable.
2. Click on server  $\rightarrow$  Desktop  $\rightarrow$  IP config  $\rightarrow$  static . Set IP . Address as 10.0.0.1 and default gateway : 10.0.0.0
3. Again configure the server PT by : config  $\rightarrow$  Services  $\rightarrow$  DHCP service  $\rightarrow$  ON, pool name : switch1. Default gateway : 10.0.0.0 man . no . of users = 100. Start IP  $\rightarrow$  10.0.0.3 then click 'Add'

4) For each PC  $\rightarrow$  go to config  $\rightarrow$  IP config  
change static to DHCP.

# Observation:

- IP address was allocated dynamically.
- Data was sent successfully among PC's when pinged.

# Output:

PC> ping 10.0.0.4

pinging 10.0.0.4 with 32 bytes of data:

Reply from 10.0.0.4; byte = 32 time = 0ms  
TTL = 28

"

"

"

"

Ping statistics for 10.0.0.4

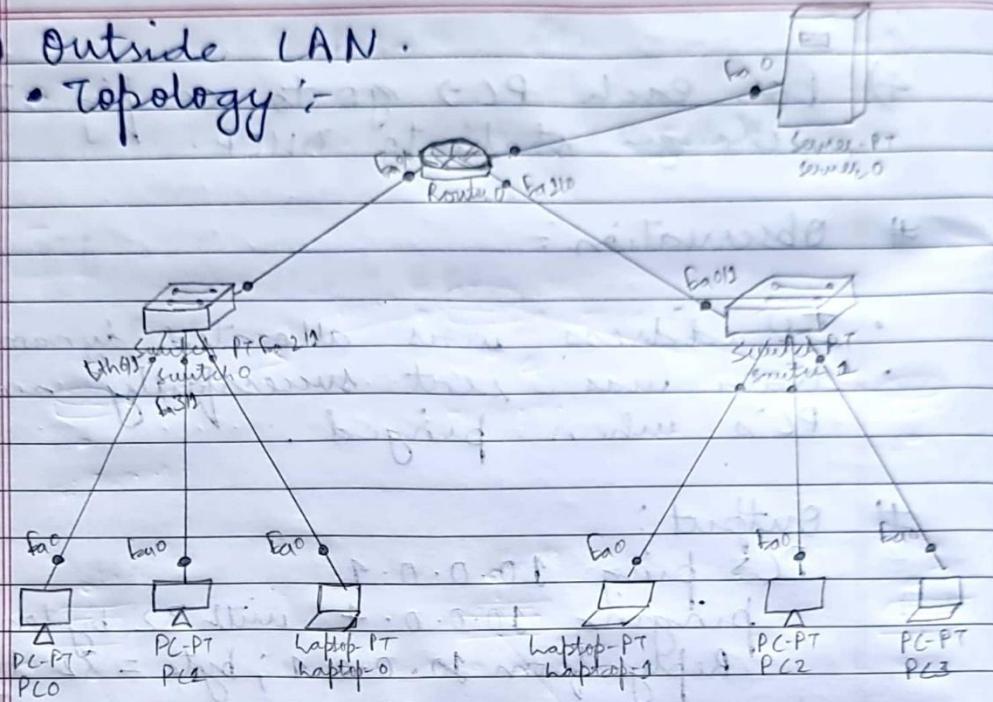
Packets sent = 4, received = 4, lost = 0  
(0% loss)

appropriate round trip time in  
milliseconds.

Minimum = 0ms, Maximum = 0ms,

## ⑪ Outside LAN.

- Topology:-



### # Procedure :

- ① For the existing server, switch & server, a laptop and 2 PCs, add a router and add second network switch.
2. In server IP config → static → IP addr: 10.0.0.2  
def gateway: 10.0.0.1
3. In server 2 → config → server → DHCP → modify  
the existing switch 1 → def gateway: 10.0.0.0  
in some: 10.0.0.0.
4. In server → DHCP → for switch 2,  
Poolname: switchtwo.  
def gateway: 20.0.0.1  
Start IP: 20.0.0.3

5. Do the router configuration :

enable ,

config terminal

interface fastethernet 4/0

ip address 10.0.0.1 255.0.0.0

ip helper address 10.0.0.2

no shut

exit

6. Now same for 2nd network ,

interface fastethernet 0/0

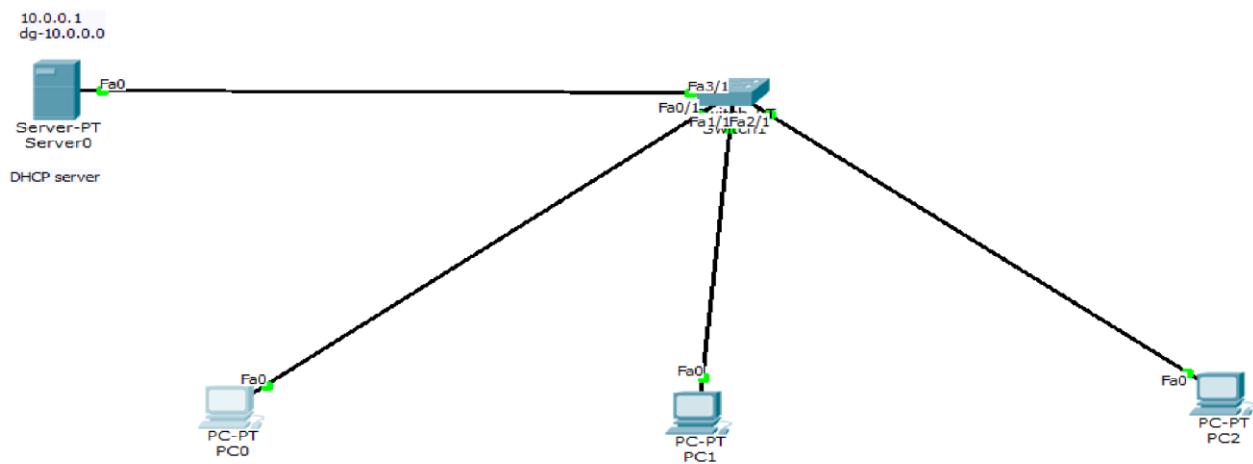
ip address 20.0.0.1 255.0.0.0

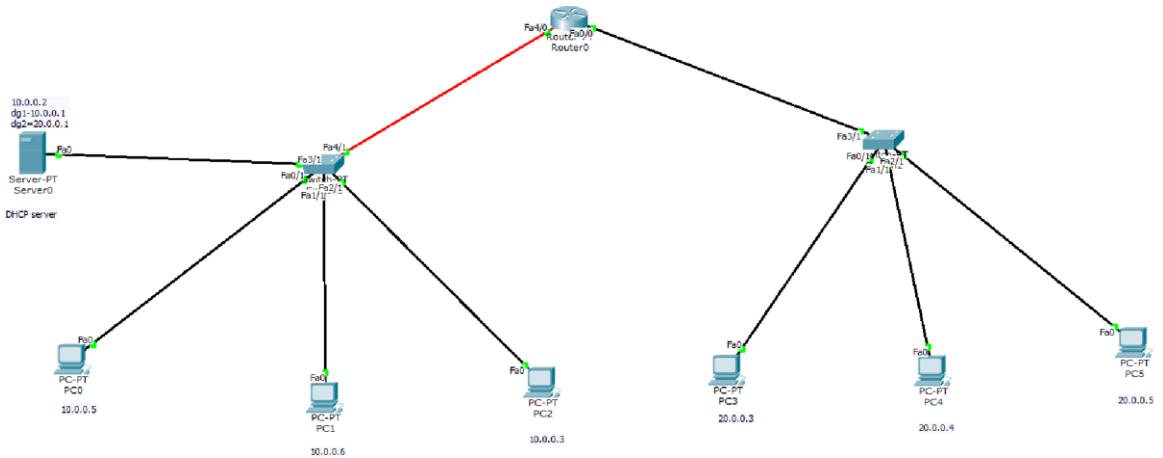
ip helper address 10.0.0.2

no shut

exit .

### Screen Shots:





PC0

Physical Config Desktop Custom Interface

## Command Prompt

```

Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:

Reply from 10.0.0.4: bytes=32 time=lms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128

Ping statistics for 10.0.0.4:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = lms, Average = 0ms

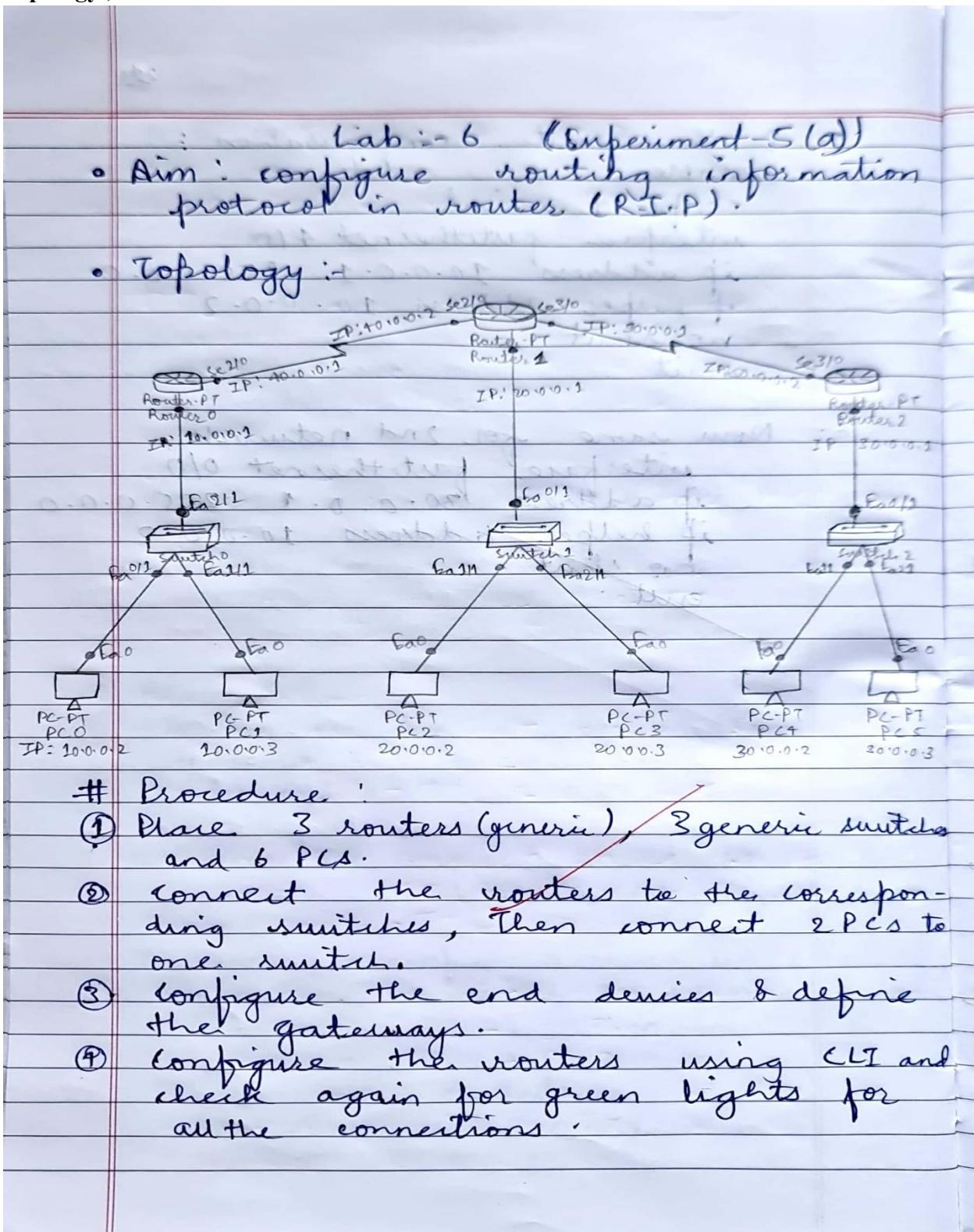
PC>

```

## Program 6

Aim: Configure RIP routing Protocol in Routers .

Topology , Procedure and Observation:



5) Configure routing information protocol to 3 routers.

In Router - 0;

```
(config) # router rip  
(config-router) # network 10.0.0.0  
(config-router) # network 40.0.0.0
```

In Router - 1:

```
(config) # router rip  
(config-router) # network 40.0.0.0  
(config-router) # network 50.0.0.0  
(config-router) # network 20.0.0.0
```

In Router - 2:

```
(config) # router rip  
(config-router) # network 50.0.0.0  
(config-router) # network 30.0.0.0
```

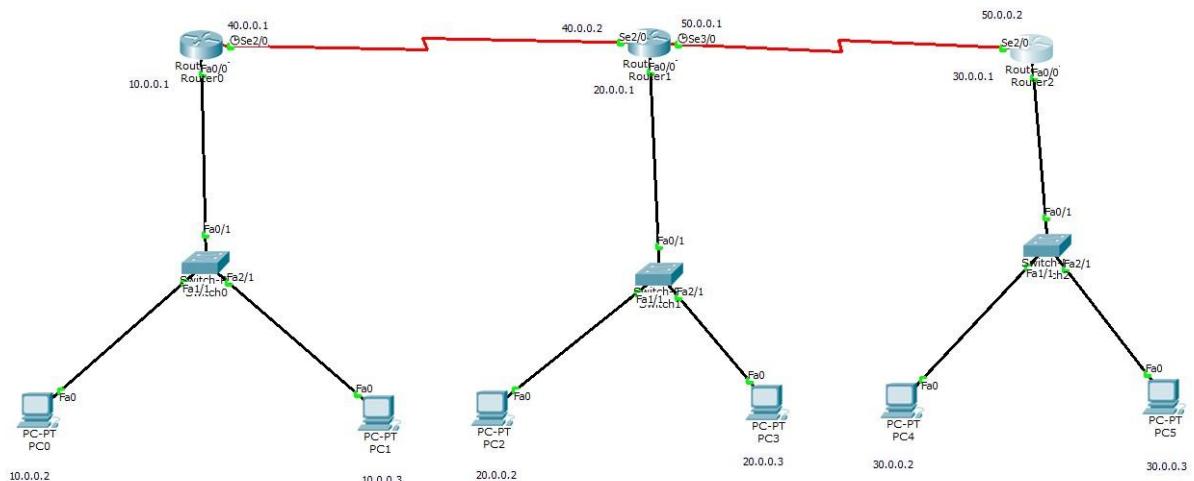
# Observations:

- Before doing RIP when pinging it showed timed out.
- After applying RIP when 30.0.0.3 was pinged from 10.0.0.1 was pinged successfully.

# show ip route for router 2:

```
R 10.0.0.0/8 [120/2] via 50.0.0.1, serial 2/0  
R 20.0.0.0/8 [120/1] via 50.0.0.1, serial 2/0  
R 50.0.0.0/8 & is directly connected, serial 2/0.
```

## Screen Shots:



PC0

Physical    Config    Desktop    Custom Interface

**Command Prompt**

```

Pinging 30.0.0.2 with 32 bytes of data:

Request timed out.
Reply from 30.0.0.2: bytes=32 time=7ms TTL=125
Reply from 30.0.0.2: bytes=32 time=6ms TTL=125
Reply from 30.0.0.2: bytes=32 time=7ms TTL=125

Ping statistics for 30.0.0.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 6ms, Maximum = 7ms, Average = 6ms

PC>ping 30.0.0.2

Pinging 30.0.0.2 with 32 bytes of data:

Reply from 30.0.0.2: bytes=32 time=4ms TTL=125
Reply from 30.0.0.2: bytes=32 time=7ms TTL=125
Reply from 30.0.0.2: bytes=32 time=7ms TTL=125
Reply from 30.0.0.2: bytes=32 time=7ms TTL=125

Ping statistics for 30.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 4ms, Maximum = 7ms, Average = 6ms

PC>

```

## Program 7

Aim: Demonstrate the TTL / Life of a Packet .

### Topology , Procedure and Observation:

Experiment : 6(b)

- Aim:- Demonstrate TTL / life of a packet .

TTL means 'time to leave' for a packet. It tells that for how many time units the packet will be there in the network.

- Procedure :-

- Send a simple PDU from PC1 to PC3.

- Click on Autos capture the event list then observe the TTF of each router in PDU information .

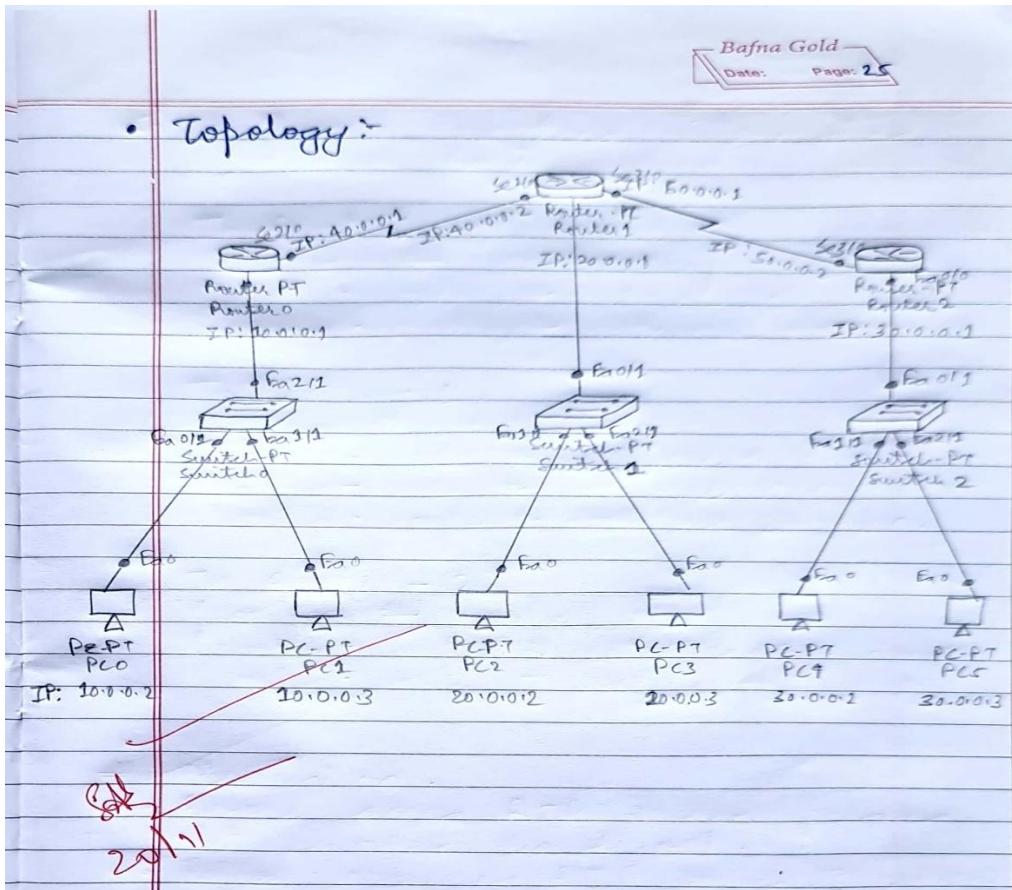
- Observations :-

- (i) When the packet passes Router 0,  
inbound - TTL = 255 ms  
outbound - TTL = 254 ms.

- (ii) When the packet passes to Router 1,  
inbound - TTL = 254 ms.  
outbound - TTL = 253 ms.

- (iii) When the packet passes across Router 2,  
inbound - TTL = 253 ms.  
outbound - TTL = 252 ms.

- Thus, we conclude that there will be a decrement in TTL for 1ms when it passes across a Router .



### Screen Shots:

PDU Information at Device: Router0

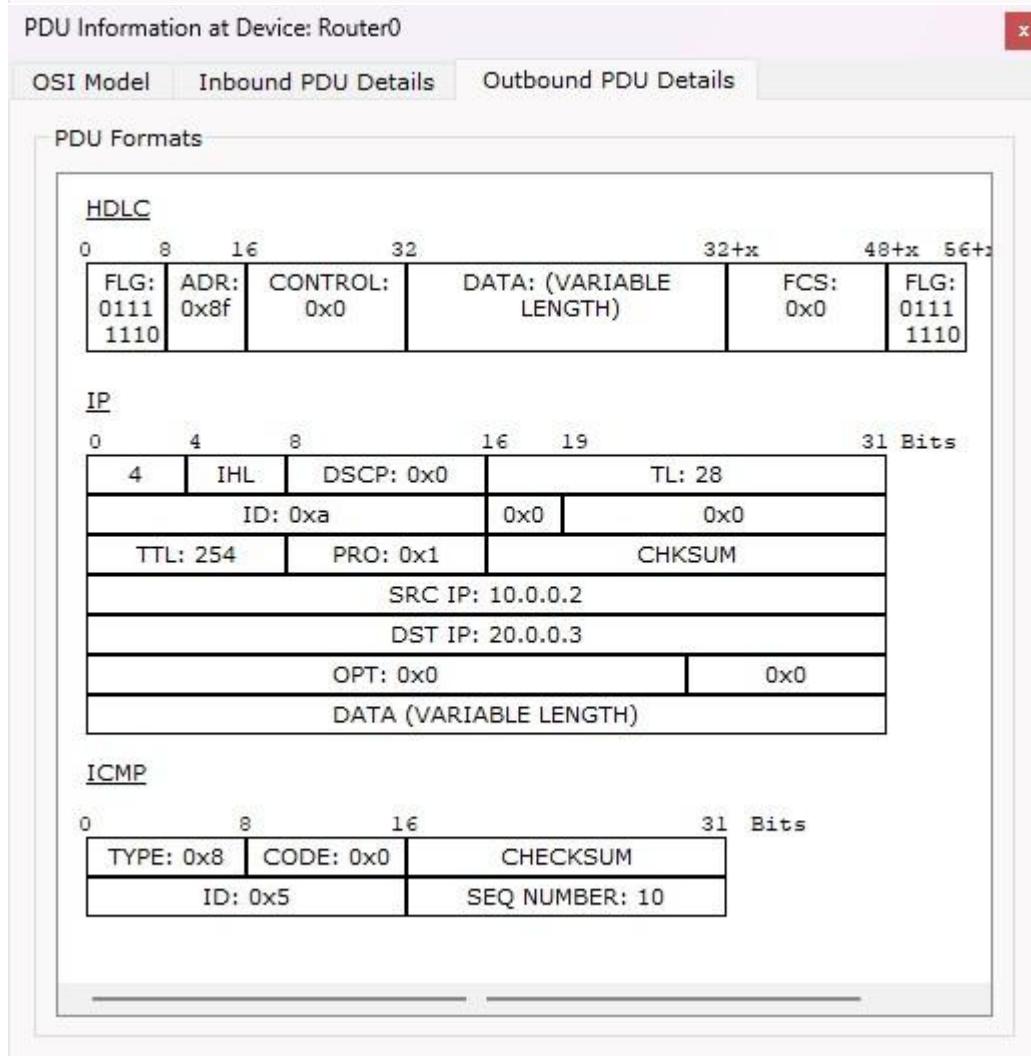
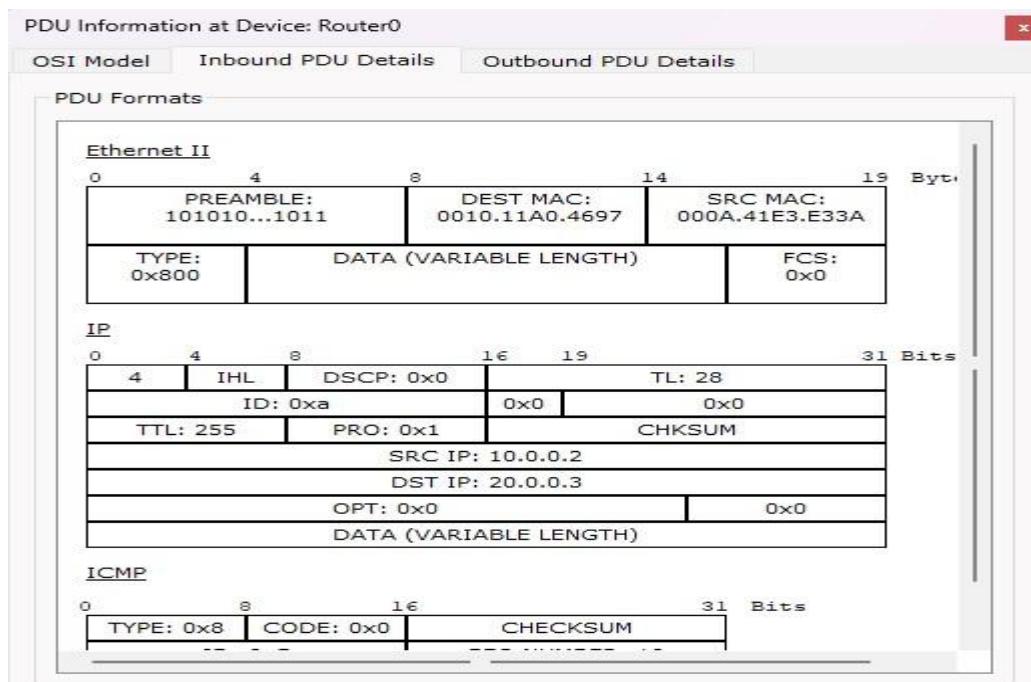
OSI Model   Inbound PDU Details   Outbound PDU Details

At Device: Router0  
Source: PC0  
Destination: PC3

In Layers	Out Layers
Layer7	Layer7
Layer6	Layer6
Layer5	Layer5
Layer4	Layer4
Layer 3: IP Header Src. IP: 10.0.0.2, Dest. IP: 20.0.0.3 ICMP Message Type: 8	Layer 3: IP Header Src. IP: 10.0.0.2, Dest. IP: 20.0.0.3 ICMP Message Type: 8
Layer 2: Ethernet II Header 000A.41E3.E33A >> 0010.11A0.4697	Layer 2: HDLC Frame HDLC
<b>Layer 1: Port FastEthernet0/0</b>	<b>Layer 1: Port(s): Serial2/0</b>

1. FastEthernet0/0 receives the frame.

Challenge Me   << Previous Layer   Next Layer >>



## Program 8

Aim: Configure OSPF routing protocol.

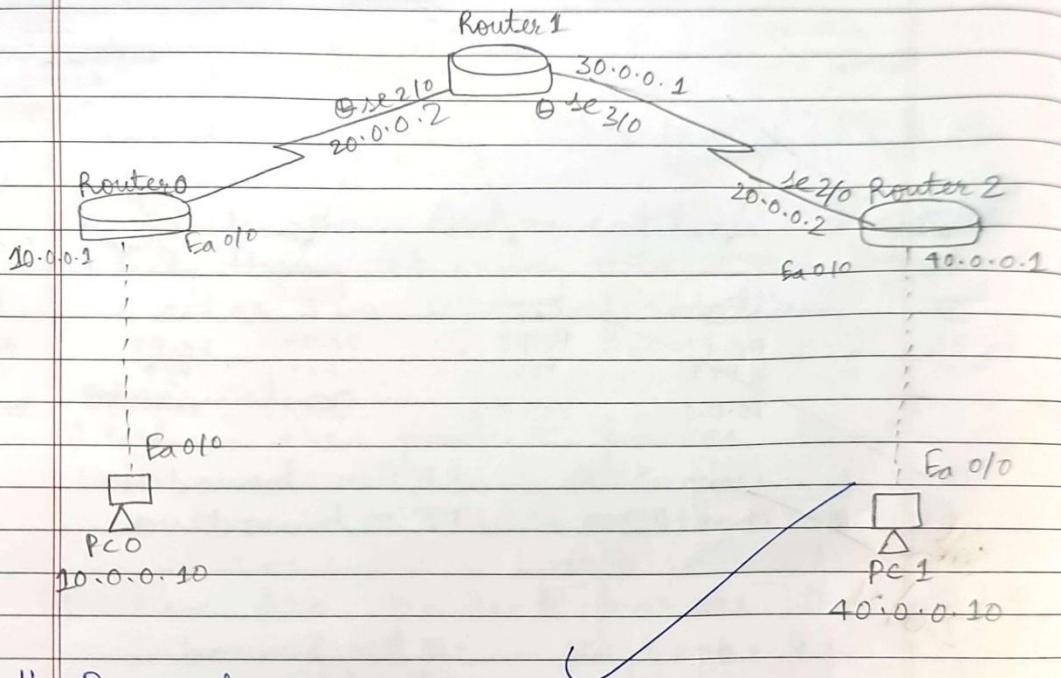
Topology , Procedure and Observation:

### Experiment - 7

# Question  $\Rightarrow$  OSPF routing protocol configuration.

# Aim  $\Rightarrow$  To configure OSPF routing protocol.

# Topology  $\Rightarrow$



# Procedure:

1) Connect the devices in the same manner as shown above.

Click on end devices  $\rightarrow$  config  $\rightarrow$  settings  $\rightarrow$  set the default gateway (IP address of its router)  $\rightarrow$  then click on fast ethernet ()  $\rightarrow$  set the IP address of the end device & subnet mask.

Click on Router:

for Router 0 → CLI

(setting up fast ethernet)

R0 (config)# interface fastethernet 0/0

R0 (config-if)# ip address 10.0.0.1 255.0.0.0

R0 (config-if)# no shutdown.

R0 (config-if)# exit

(setting up serial connection)

R0 (config)# interface serial 2/0

R0 (config)# ip address 20.0.0.1 255.0.0.0

R0 (config-if)# encapsulation PPP

R0 (config-if)# clock rate 64000

R0 (config-if)# no shutdown

R0 (config-if)# exit

Similarly, we set up the IP's of R1 and R2 while the setup of fast ethernet remains the same, the setting up of serial connections has 2 extra lines (encapsulation PPP, clock rate 64000).

Clock rate 64000 must only be written if the serially connected port shows a  symbol.

Thus, we write the clock rate command for R0 serial 2/0, R1 serial 3/0.

After this step, all the connections must have been turned to green.

2.) To enable IP routing by configuring OSPF routing protocol in all routers.

Router R0 → CLI

R0(config)# router ospf 1

R0(config-router)# router-id 1.1.1.1

R0(config-router)# network 10.0.0.0

R0(config-router)# network 20.0.0.0

R0(config-router)# exit .

Similarly do the same for R1 and R2 and specify the area numbers and ip addresses with subnet masks in CLI.

3.) Once the setting up of networking area is done , we configure loopback address to the routers :

R0(config-if)# interface loopback 0

R0(config-if)# ip add 172.16.1.252 255.255.0.0

R0(config-if)# no shutdown

R1(config-if)# interface loopback 0

R1(config-if)# ip add 172.16.1.253 255.255.0.0

R1(config-if)# no shutdown

R2(config-if)# interface loopback 0

R2(config-if)# ip add 172.16.1.254 255.255.0.0

R2(config-if)# no shutdown .

4) On checking routing table of R2 using show ip route , we can see that R2 doesn't know about area 3.

Gateway of last resort is not set.

O IA 20.0.0.0/8 [110/128] via 30.0.0.1 serial 2/0 .

C 40.0.0.0/8 is directly connected, fastethernet 0/0

C 30.0.0.0/8 is directly connected, serial 2/0 .

Since, R2 doesn't know about area 3, we have to create a virtual link between R0 and R1.

5. Creating virtual link between R1, R0

In Router R0 ,

R0(config)# router ospf 1

R0(config-router)# area 1 virtual link  
2.2.2.2

R0(config-router)# exit.

In Router R1

R1(config)# router ospf 1

R1(config-router)# area 1 virtual link  
1.1.1.1

R1(config-router)# exit .

6) Now, check routing table of R2 . Once all these steps are completed , the messages can be pinged across the devices.

## # Observations :

In Router R2 :

Router # show ip route

O IA 20.0.0.0/8 [110/128] via 30.0.0.1,  
00:57:25, serial 2/0

C 40.0.0.0/8 is directly connected,  
Fast Ethernet 0/0

O IA 10.0.0.0/8 [110/129] via 30.0.0.1  
00:57:25, serial 2/0

C 30.0.0.0/8 is directly connected,  
serial 2/0

C 172.16.0.0/16 is directly connected,  
loopback0.

Similarly the output is shown for  
Router 0 and 1.

Ping output :

(from PC0 to PC1)

PC0 → command prompt

C:\> ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data

Request timed out

Reply from 40.0.0.10: bytes=32, time=21ms TTL=125

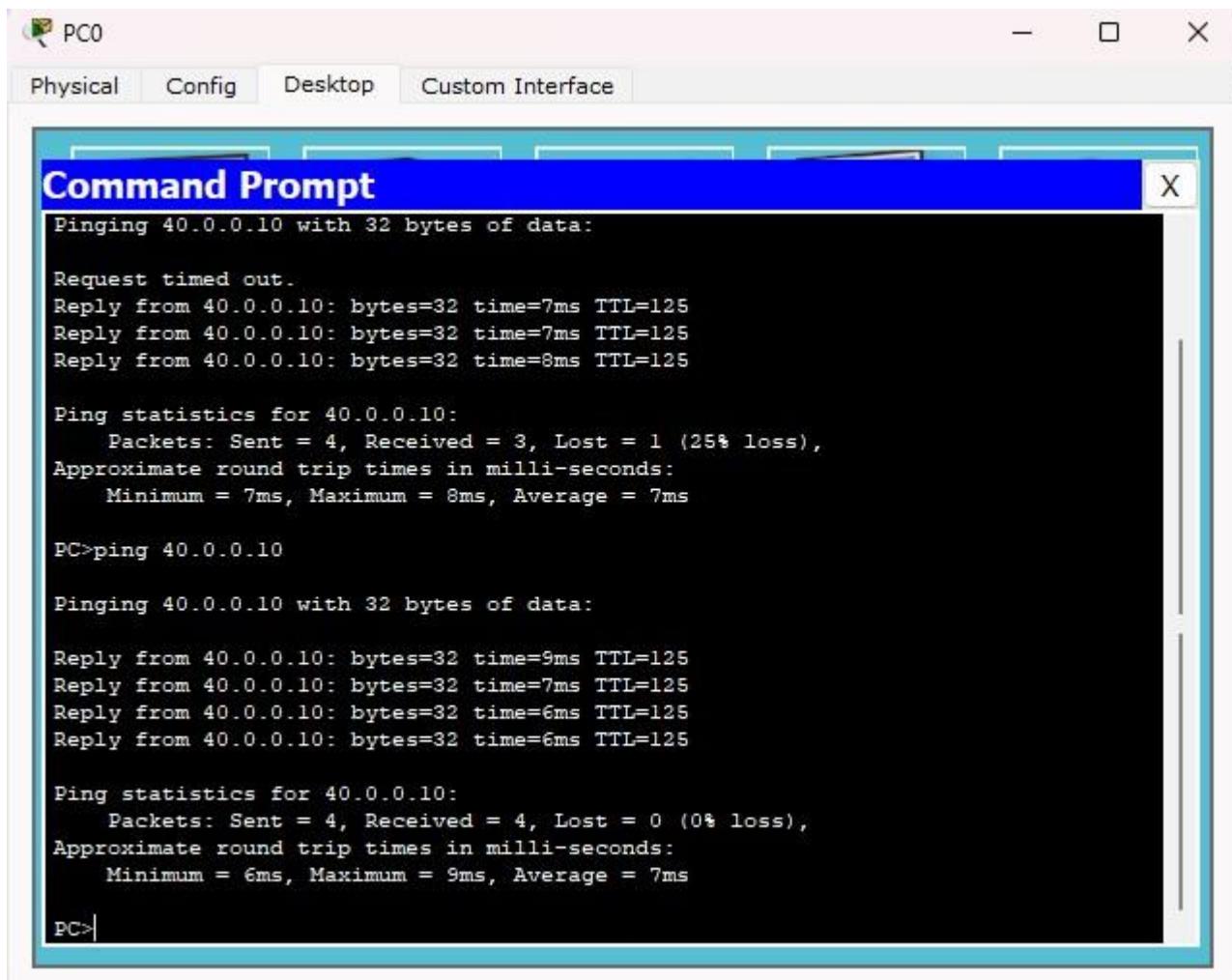
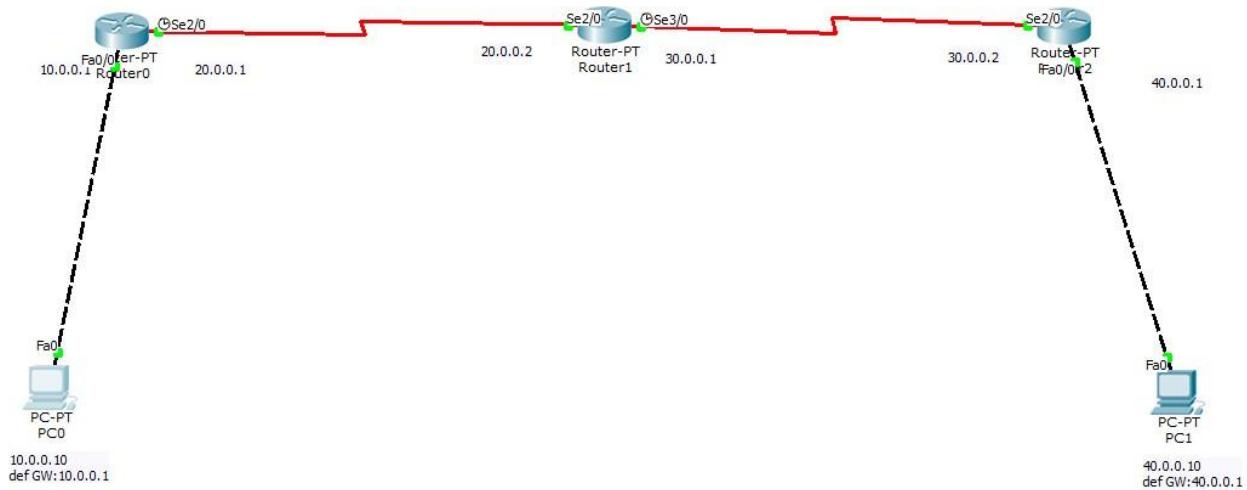
Reply from 40.0.0.10: bytes=32, time=2ms TTL=125

Reply from 40.0.0.10: bytes=32, time=28ms TTL=128

Pinging statistics for 40.0.0.10

Packets: sent=4, received=3, loss=1 (25% loss)

## Screen Shots:



## Program 9

Aim: Configure Web Server, DNS within a LAN.

### **Topology , Procedure and Observation:**

Batch 18  
Date: Dec Page 32

Experiment - 10  
DNS

# Aim : Configure Web server, DNS within a LAN.

# Topology :

# Procedure :-

1.) ~~set up the LAN as per the topology mentioned above and configure the devices accordingly.~~

2.) Go to Server  $\Rightarrow$  Services  $\Rightarrow$  DNS:  
Name : bmsce [Domain name]  
Address : 10.0.0.2  
Add the mapping of domain name to the address.

3.) Go to PC  $\Rightarrow$  config  $\Rightarrow$  Global  $\Rightarrow$  setting  
 $\rightarrow$  DNS Server : 10.0.0.2

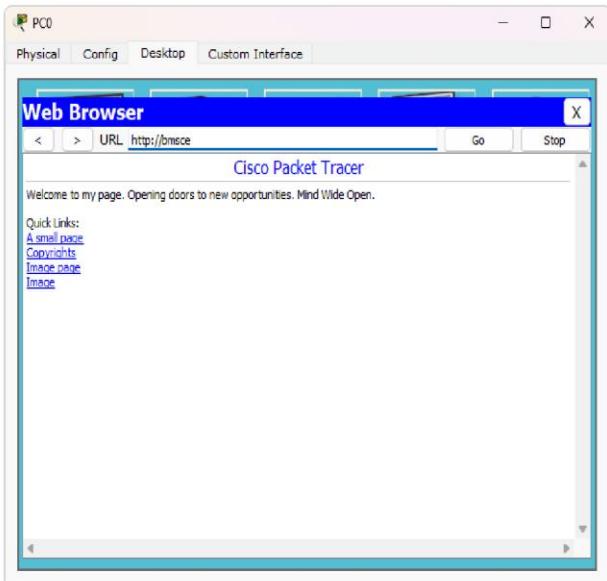
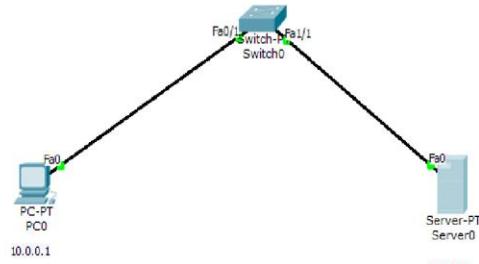
[the server that provides the DNS mapping].

- 4) Go to PC  $\rightarrow$  Desktop  $\rightarrow$  Web Browser  
Type the URL: `http://bmsce`

# Observations :-

- 1) The webpages hosted by the server were visible on the browser.
- 2) The DNS was successful in mapping the domain name to the IP address.
- 3) DNS Server is a server that contains a Domain Name : IP address mapping to which the end devices send requests to map the Name to IP addresses.

### Screen Shots:



## **Program 10**

**Aim:** To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)

### **Topology , Procedure and Observation:**

Bafra Gold -  
Date \_\_\_\_\_ Page: 35

Experiment - 9  
ARP

# Aim :- To construct a simple LAN & understand the concept and operation of Address Resolution Protocol (ARP).

# Topology :-

# Procedure :

- 1.) Create the topology as shown above.
- 2.) Configure the PCs and the server.
- 3.) Click on the Inspect mode (Q), then click on the end devices and open ARP tables.
- 4.) Send a data packet from any end device say server to other end device say 10.0.0.3 .PC.

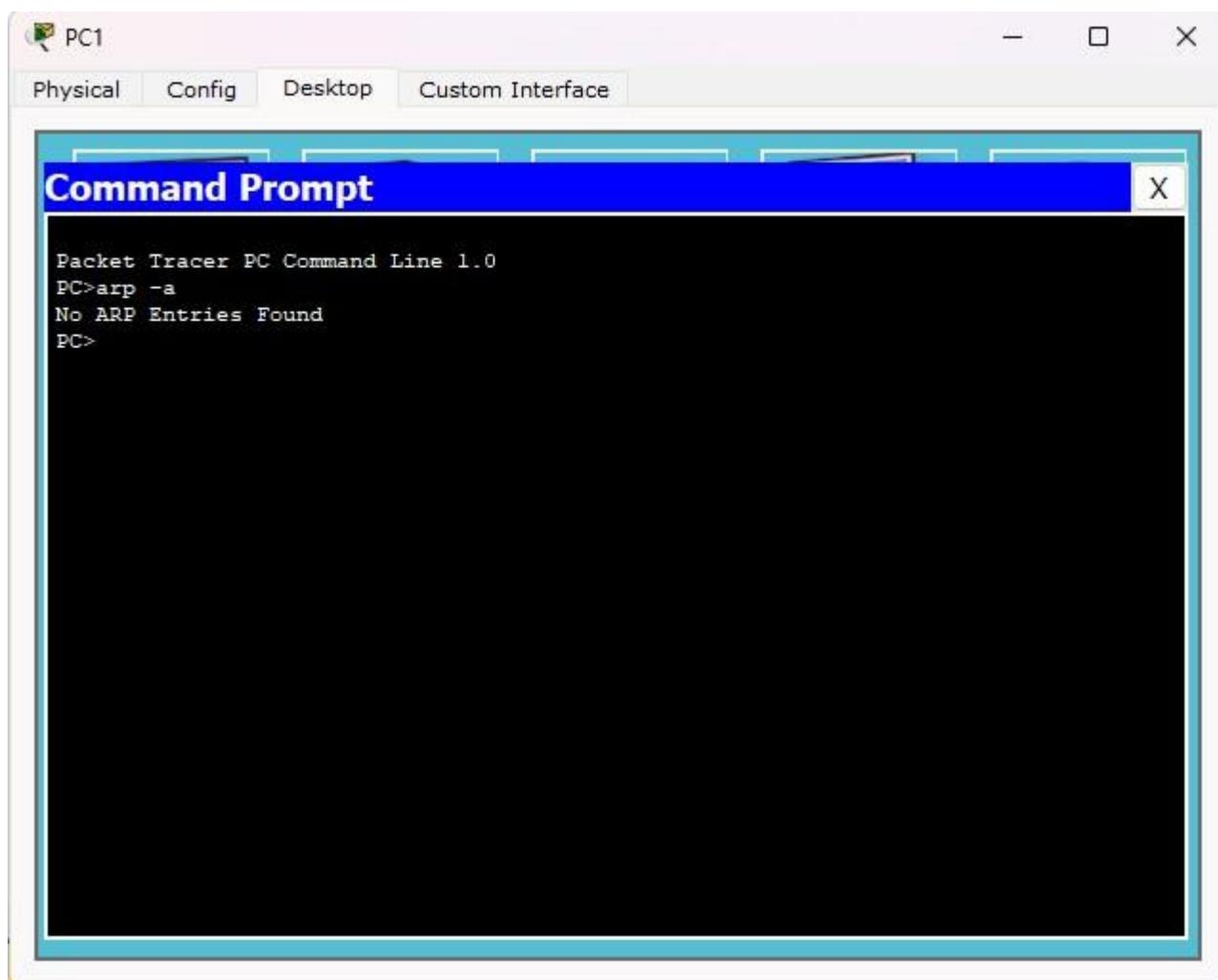
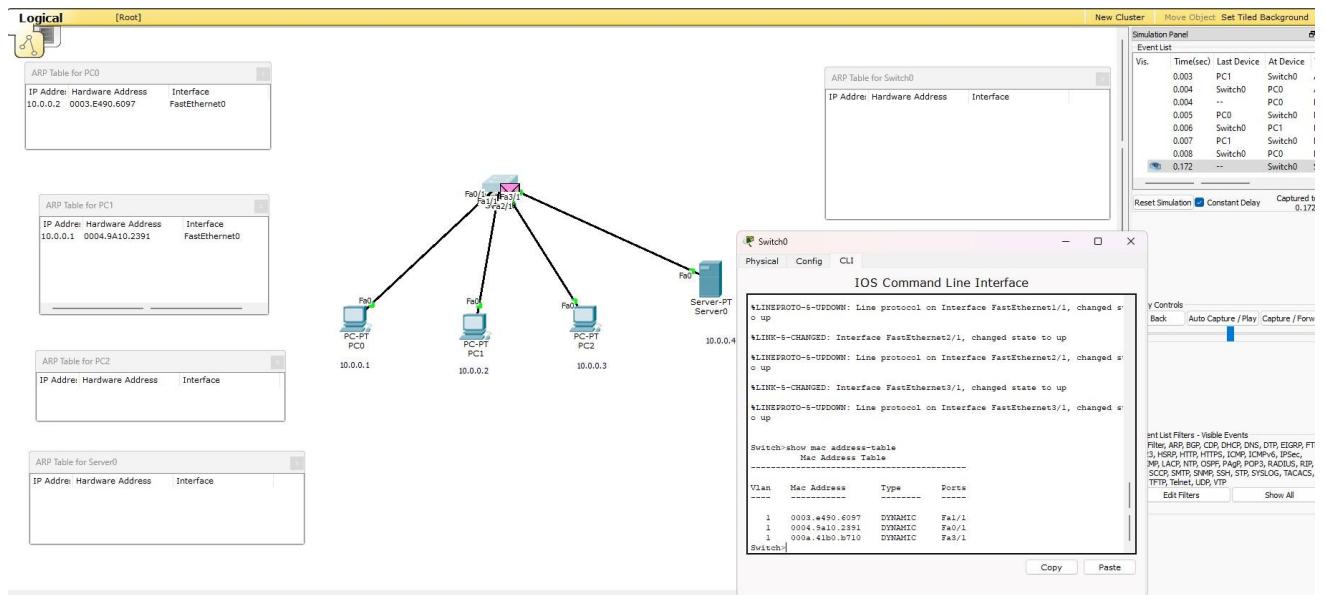
- 5.) Open simulation mode to capture each step of data transfer.

# Observations :

- 1.) The ARP tables of all end devices are initially empty.
  - 2.) When the data packet from server arrives at the switch, since the source MAC address is unknown, it sends a broadcast message to all devices.
  - 3.) The device with the IP address present in the destination address of the data packet responds to the messages.
  - 4.) The server and the PC updates their ARP tables matching the IP addresses to MAC addresses.
  - 5.) Over time, the ARP tables grow as data packets are sent.
  - 6.) The MAC table of the switch, which was initially empty updates its MAC table gradually too.

ARP Table for 10.0.0.9		
IP Address	Hardware Address	Interface
10.0.0.3	0001·(726.47E5)	East Ethernet0

## Screen Shots:



## Program 11

**Aim:** To understand the operation of TELNET by accessing the router in the server room from a PC in the IT office.

### **Topology , Procedure and Observation:**

Date: 08/01/2023  
Page No. 29

Experiment - 11  
TELNET

# Aim → To understand the operation of TELNET by accessing the router in server room from PC in IT office.

# Topology :-

PCO (10.0.0.1) ----- Router (10.0.0.2)

# Procedure :-

- 1.) Create the topology as given above and configure the devices accordingly
- 2.) Commands in Router :-  
Router > enable  
Router # config terminal  
Router (config) # hostname R1  
R1 (config) # enable secret 1234  
(enable password)  
R1 (config) # interface fastethernet 0/0  
R1 (config-if) # ip address 10.0.0.2  
255.0.0.0  
R1 (config-if) # no shut.  
R1 (config-if) # line vty 0..3  
R1 (config-line) # login.

% login disabled on line 194, until  
'password' is set.

R1(config-line)# password 4321

R1(config-line)# exit

<sup>user access</sup>  
↓  
verification process

R1(config)# exit

R1# wr → to perform write operation.  
Building configuration.

[OK]

Note: vty 0 3: First for virtual terminal lines for telnet access.

3.) In PC: command prompt:

- First try pinging to see if the devices are connected.

PC> telnet 10.0.0.2

Trying 10.0.0.2 open

User Access Verification:

Password : 4321

Password : 4321

R1> enable

Password : 1234

R1# show ip route.

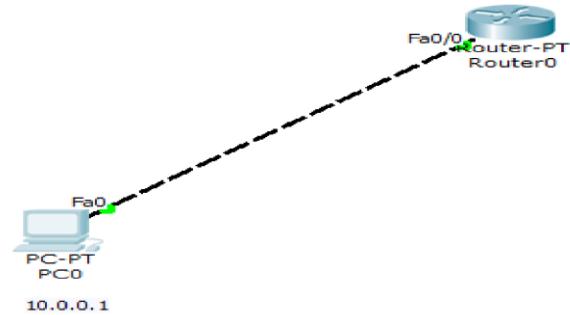
C: 10.0.0.0/8 is directly connected,  
East, Ethernet 0/0 .

R1#

## # Observations :

- 1.) The admin in PC is able to run commands as user in router CLI and see the results from PC.
- 2.) Telnet allows users to establish a remote session with another device like router, over a TCP/IP network.
- 3.) Using Telnet, we can access and control the remote device's CLI as if you were physically connected to it.

## Screen Shots:



## Command Prompt

```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time=0ms TTL=255

Ping statistics for 10.0.0.2:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>telnet 10.0.0.2
Trying 10.0.0.2 ...Open

User Access Verification

Password:
R1>enable
Password:
R1#show ip route
Codes: C - connected, S - static, I - IGRP, 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

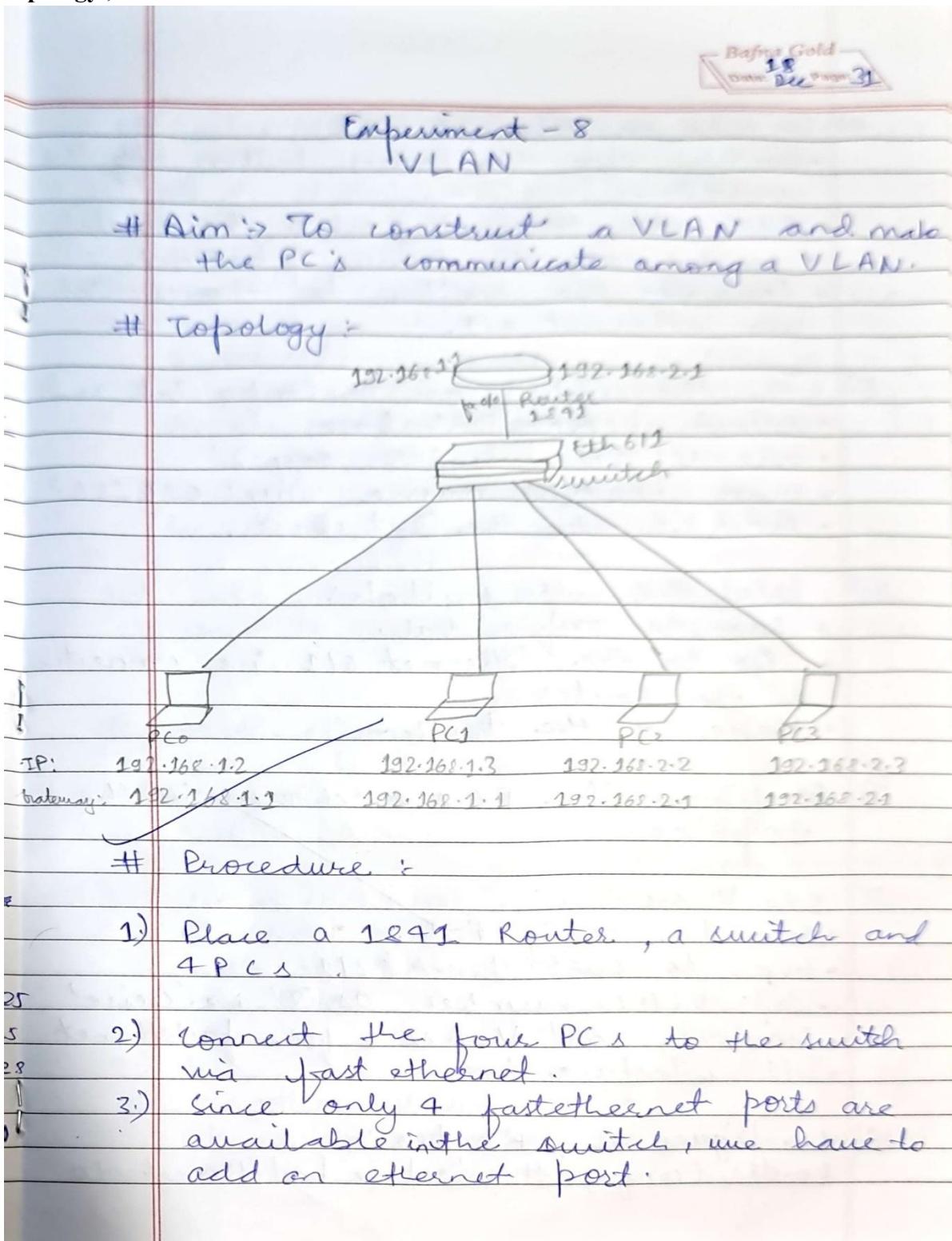
Gateway of last resort is not set

C   10.0.0.0/8 is directly connected, FastEthernet0/0
R1#
```

## Program 12

**Aim:** To construct a VLAN and make the PC's communicate among a VLAN .

**Topology , Procedure and Observation:**



4) To add an ethernet port:

- Switch off the Power button off the switch.
- Add the ethernet port to the switch
- Switch on the power & buttons
- Connect the router to the switch via ethernet 6/1.

5) In the switch, go to Config Tab and:

- Select VLAN Database
- Give VLAN number say '2'
- Give VLAN name say 'CSE110'.
- Add it to the Database.

6) Select the the switch:

- Go to config tab.
- Go to the Ethernet 6/1 i.e. connected to the Router.
- Make it the trunk.

7) Configure the PCs as shown in the topology.

8) Select switch:

- Go to config tab.
- Go to FastEthernet 2/1
- Set VLAN number as '2' i.e. 'cse110'
- Similarly set VLAN 2 for fastEthernet 3/1 interface too.

9) Configure the Router:

Router(config)# interface FastEthernet 0/0

Router (config-if)# ip address 192.168.1.1  
255.255.255.0

Router (config-if)# no shut

Router (config-if)# exit.

Now, to configure the router's VLAN interface:

Router (config)# interface fastethernet 0/0.1

Router (config-subif)# encapsulation dot1q 2

Router (config-subif)# ip address 192.168.1.1  
255.255.255.0

Router (config-subif)# no shut

Router (config-subif)# exit.

- 10) Ping devices within the same VLAN and to the devices of different VLAN.

### # Observations :

- 1) When devices are pinged within same VLAN:

- Pinging 192.168.1.3 from 192.168.1.2
  - the data packet doesn't go to the router.
  - the switch forwards the packet without the need of the router.

- 2) When a device pings a device of another VLAN:

- Pinging 192.168.2.3 from 192.168.1.2

- The data packet's journey is as follows:

192.168.1.2 → switch → Router

192.168.2.3 ← Switch

- VLANs divide a single switch into multiple logical switches.

- Devices in one VLAN cannot directly communicate with devices in another VLAN without a router.

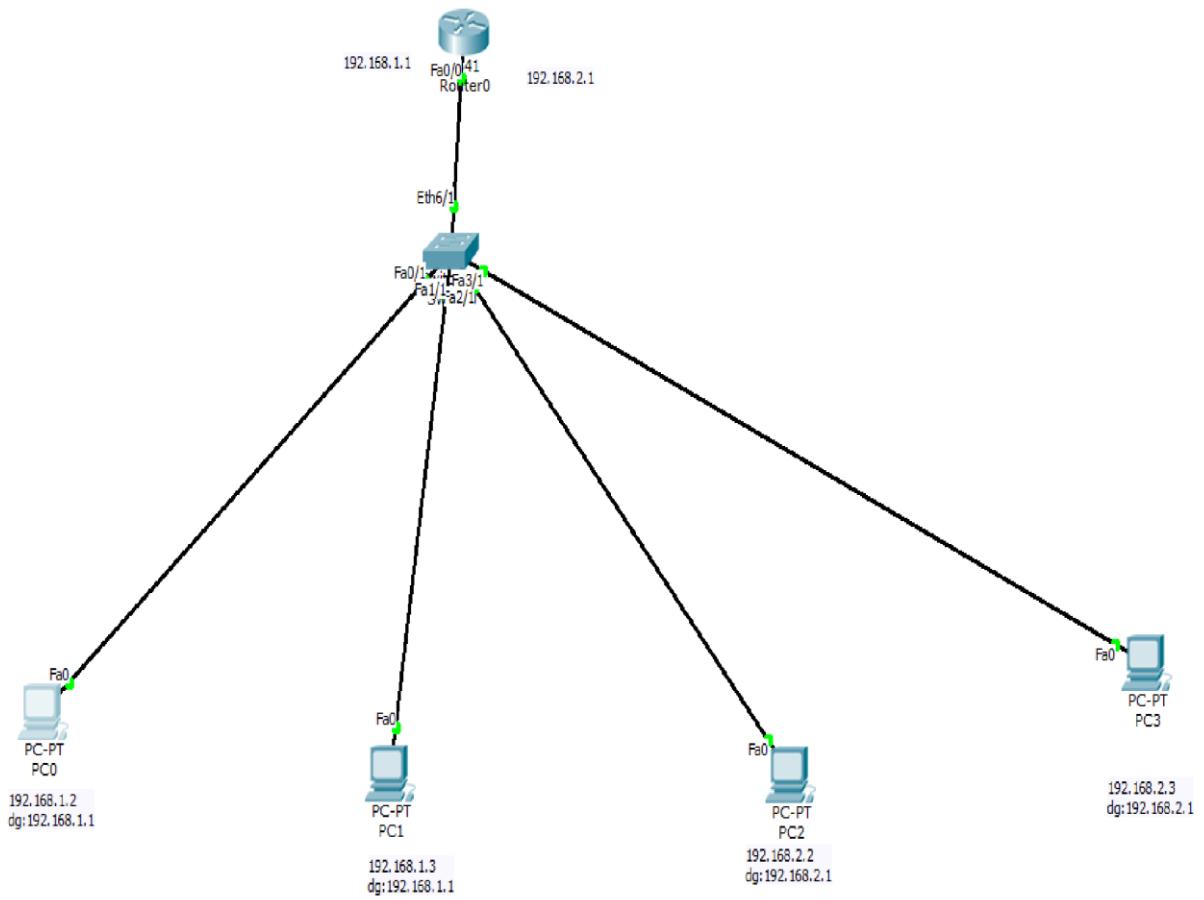
- Traffic Isolation:

- Each VLAN maintains its own broadcast domain.
- Broadcasts sent by devices in one VLAN do not reach devices in another VLAN.

- VLAN trunking allowed switches to forward frames from different VLANs over a switch link called the trunk.

- This is done by adding an additional header information called tag to the ethernet frame - VLAN tagging.

## Screen Shots:



## Command Prompt

```
Packet Tracer PC Command Line 1.0
PC>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Request timed out.
Reply from 192.168.2.2: bytes=32 time=0ms TTL=127
Reply from 192.168.2.2: bytes=32 time=0ms TTL=127
Reply from 192.168.2.2: bytes=32 time=4ms TTL=127

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 4ms, Average = 1ms

PC>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Reply from 192.168.2.2: bytes=32 time=0ms TTL=127
Reply from 192.168.2.2: bytes=32 time=0ms TTL=127
Reply from 192.168.2.2: bytes=32 time=2ms TTL=127
Reply from 192.168.2.2: bytes=32 time=0ms TTL=127

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms

PC>ping 192.168.2.3

Pinging 192.168.2.3 with 32 bytes of data:

Request timed out.
Reply from 192.168.2.3: bytes=32 time=3ms TTL=127
Reply from 192.168.2.3: bytes=32 time=2ms TTL=127
Reply from 192.168.2.3: bytes=32 time=1ms TTL=127

Ping statistics for 192.168.2.3:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 3ms, Average = 2ms

PC>ping 192.168.2.3

Pinging 192.168.2.3 with 32 bytes of data:

Reply from 192.168.2.3: bytes=32 time=0ms TTL=127
Reply from 192.168.2.3: bytes=32 time=0ms TTL=127
Reply from 192.168.2.3: bytes=32 time=2ms TTL=127
Reply from 192.168.2.3: bytes=32 time=0ms TTL=127

Ping statistics for 192.168.2.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms

PC>
```

### Program 13

**Aim:** To construct a WLAN and make the nodes communicate wirelessly.

#### **Topology , Procedure and Observation:**

18-Dec-2021

Experiment -12  
WLAN

# Aim : To construct a wireless LAN and make the nodes communicate wirelessly.

# Initial Topology :

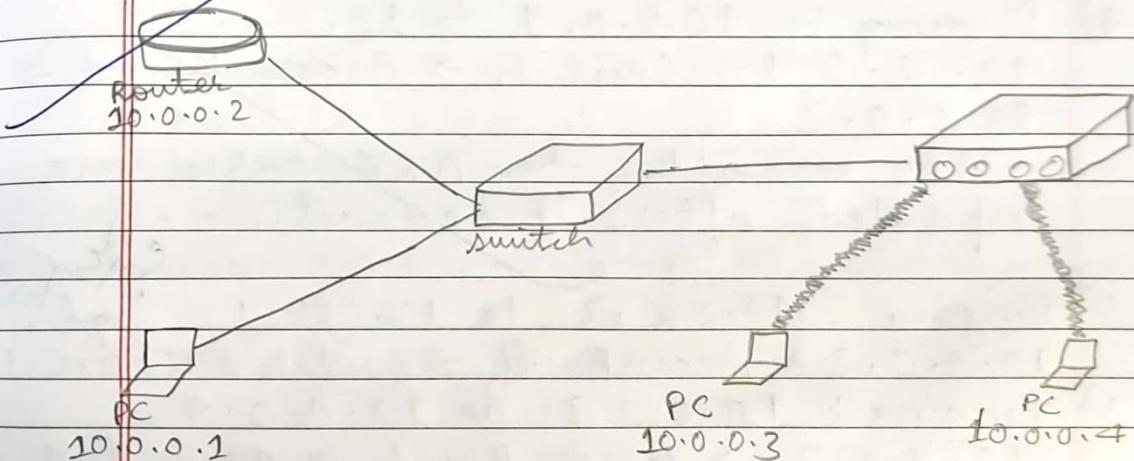
The diagram shows a network topology for a WLAN experiment. At the top left is a Router labeled "Router 10.0.0.2". A line connects it to a central switch. From the switch, four lines extend to four different devices: a PC labeled "PC 10.0.0.1" at the bottom left, another PC labeled "PC 10.0.0.3" at the bottom center, a third PC labeled "PC 10.0.0.2" at the bottom right, and a Laptop labeled "Laptop 10.0.0.4" at the top right. The connections are represented by simple lines.

# Procedure :

- 1) Create the topology as given above and configure the devices.
- 2) Configure Access Point :  
click Access Point → config → Port 1  
SSID : bmsce  
Select ① WEP  
Set key : 1234567890

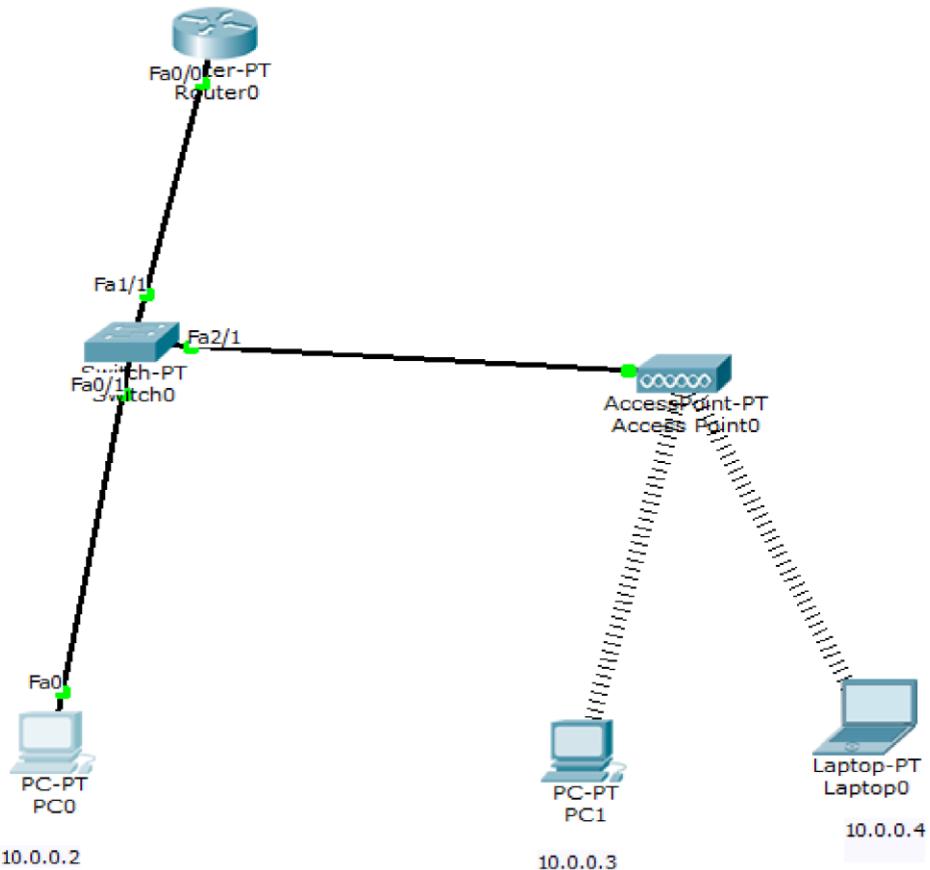
- 3.) Configure PC and laptops with wireless standards:
- Switch off device.
  - Drag the existing PT-HOST-NM-LAN to the component listed in the LHS of Physical.
  - Drag WMP300N wireless interface to the empty port.
  - Switch on the device.
- 4.) In the config tab, a new wireless interface was added.
- 5.) Configure the device by entering SSID, WEP, WEP key, IP addresses and gateway.

~~Topology after wireless configuration:~~



- 6.) Ping from every device to every other device to check for connection.

## Screen Shots:



PC0

Physical Config Desktop Custom Interface

Command Prompt

```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time=22ms TTL=128
Reply from 10.0.0.3: bytes=32 time=6ms TTL=128
Reply from 10.0.0.3: bytes=32 time=3ms TTL=128
Reply from 10.0.0.3: bytes=32 time=7ms TTL=128

Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 3ms, Maximum = 22ms, Average = 9ms

PC>ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:

Reply from 10.0.0.4: bytes=32 time=19ms TTL=128
Reply from 10.0.0.4: bytes=32 time=5ms TTL=128
Reply from 10.0.0.4: bytes=32 time=6ms TTL=128
Reply from 10.0.0.4: bytes=32 time=7ms TTL=128

Ping statistics for 10.0.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 5ms, Maximum = 19ms, Average = 9ms

PC>
```

## Cycle-B

### Program 1

Write a program for error detecting code using CRC-CCITT (16-bits).

### **Code and Output:**

01-Jan-2025

Experiment - 14

```
# AIM: Implementation of CRC -  
# Code :  
  
def XOR (a, b):  
    result = []  
    for i in range (0, len(b)):  
        if a[i] == b[i]:  
            result.append ('0');  
        else :  
            result.append ('1');  
  
    return ''.join (result)  
  
def moddiv (dividend, divisor):  
    pick = len (divisor)  
    temp = dividend [0: pick]  
    while pick < len (dividend):  
        if temp [0] == '1':  
            temp = XOR (divisor, temp) +  
            dividend [pick]  
        else :  
            temp = XOR ('0' * pick, temp)  
            + dividend [pick]  
        pick += 1  
  
        if temp [0] == '1':  
            temp = XOR (divisor, temp)  
        else :  
            temp = XOR ('0' * pick, temp)
```

checkword = temp  
return checkword

```
def encodeData(data, key):  
    l_key = len(key)  
    append_data = data + '0'*(l_key - 1)  
    remainder = mod2div(append_data, key)  
    codeword = data + remainder  
    print("Remainder", remainder)  
    print("Encode Data (Data + Remainder),  
        Codeword")
```

data = "100100"  
key = "1101"  
encodeData(data, key)

# Output :

Sender side : ---  
Remainder : 001

✓ Encode Data (Data + Remainder) : 100100001

Receiver side :  
correct message received.

(82)

#### Output Screenshot:

```
PS C:\Users\HP\OneDrive\Desktop\cn lab> python crc-ccitt.py  
Enter the original bitstream (e.g., 11010011101100): 11010011101100  
Transmitted bitstream with CRC: 110100111011001010011000100100  
Enter the received bitstream for verification: 1101000000000010100001  
Error detected! CRC invalid.
```

## Program 2

Write a program for congestion control using Leaky bucket algorithm.

### Code and Output:

Page No. 15  
Date 1-3-2025

Cycle - 2

Experiment - 13  
Leaky Bucket Algorithm.

In the network layer, before the network can make quality of service guarantees, it must know what traffic is being guaranteed. One of the main causes of congestion is that traffic is often

There are two types of traffic sharing :-

- 1.) Leaky Bucket
- 2.) Token Bucket.

For example, let  $n = 1000$   
packet = 200 700 500 450 400 200  
since  $n >$  size of the packet at the head of the queue i.e.  $n > 200$ .

Therefore,  $n = 1000 - 200 = 800$   
packet size of 200 is sent into the networks : 200 700 500 450 400

Now again,  $n >$  size of the packet of the head of the queue i.e.  $n > 400$   
Therefore,  $n = 800 - 400 = 400$

# Code :

```
#include <stdio.h>
int main () {
    int incoming, outgoing, bucket size,
        n, store = 0;
    printf ("Enter the bucket size,
            outgoing node, and the no. of
            stores : ");
    scanf ("%d, %d %d", &bucket-size,
           &outgoing, &n);
    while (n != 0) {
        printf ("Enter the incoming
                packet size : ");
        scanf ("%d", &incoming);
        printf ("Incoming packet size %d
                \n", incoming);
        if incoming > (bucket-size - store) {
            store += incoming;
            printf ("Bucket buffer size %d
                    out of %d \n", store, buck-size);
        } else {
            printf ("Dropped %d no of packets
                    in %d, incoming = %d - store");
            printf ("Bucket Buffer size %d
                    out of %d \n", store, buck-size);
            store = buck-size;
        }
        store = store - outgoing;
        printf ("After outgoing %d bytes
                left out of %d in buffer \n", store,
```

```
buck_size);  
m--;  
}
```

?

## # Output:

Enter bucket size, outgoing rate and no. of inputs : 100 20 3

Enter the incoming packet size : 30

Incoming packet size 30

Bucket suffix size 30 out of 100  
After outgoing 10 bytes left out of 100 in buffer.

Enter the incoming packet size : 50

Incoming packet size : 50

Bucket suffix size 60 out of 100

After outgoing 40 bytes left out of 100 in buffer.

Enter the incoming packet size : 80

Incoming packet size : 80

Dropped 20 no. of packets.

Bucket Buffer size 40 out of 100

After outgoing 80 bytes left out of 100 in buffer.

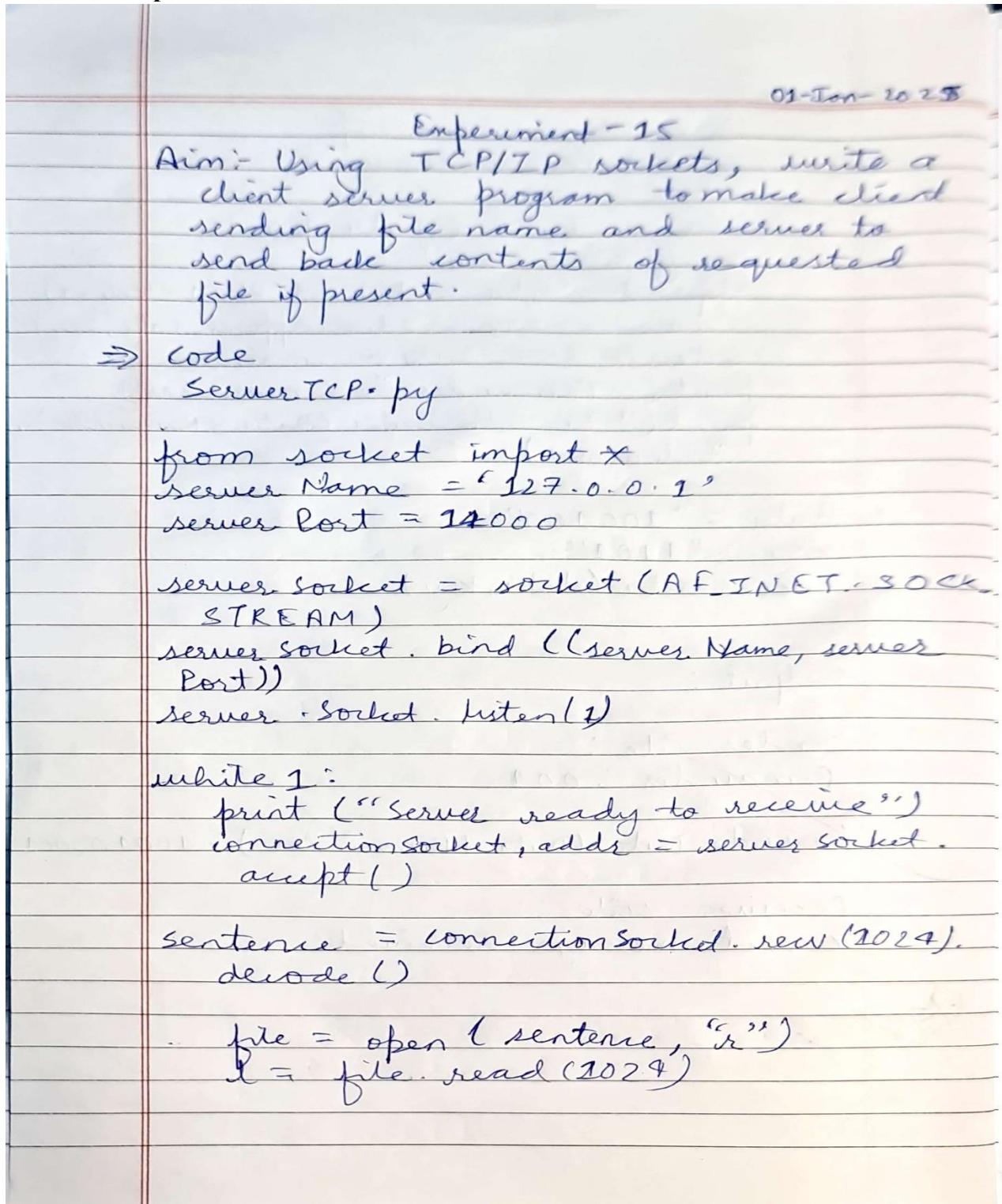
**Output Screenshot:**

```
PS C:\Users\DELL\OneDrive\Desktop\code> python leakybucketalgorithm.py
Enter no of queries:10
Enter bucket size:5
Enter input packet size:4
Enter output packet size:6
Bucket size=4out of bucket size=5
Bucket size=2out of bucket size=5
Bucket size=0out of bucket size=5
Bucket size=-2out of bucket size=5
Bucket size=-4out of bucket size=5
Bucket size=-6out of bucket size=5
Bucket size=-14out of bucket size=5
```

### **Program 3**

Using TCP/IP sockets, write a client-server program to make the client send the file name and the server to send back the contents of the requested file if present.

#### **Code and Output:**



```
connectionSocket.send(sentence.encode())
print('In sent contents of' + sentence)
file.close()
connectionSocket.close()
```

### # ClientTCP.py

```
from socket import *
serverName = '127.0.0.1'
serverPort = 14000
```

```
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
```

```
sentence = input("In Enter the file name: ")
```

```
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
```

```
print('In From Server: \n')
print(filecontents)
clientSocket.close()
```

⇒ Output:

The server is ready to receive..

Send contents of serverTCP.py.

Reply from server:

**Output Screenshots:**

```
PS C:\Users\DELL\OneDrive\Desktop\code> python tcpserver.py
The server is ready to receive
```

Sent contents of example.txt

The server is ready to receive

```
PS C:\Users\DELL\OneDrive\Desktop\code> python tcpclient.py
```

Enter file name: example.txt

From Server:

Hello, this is a sample file.

It is used for testing the TCP server.

#### Program 4

Using UDP sockets, write a client-server program to make the client send the file name and the server to send back the contents of the requested file if present.

#### Code and Output:

01-Jan-2025

Experiment - 16

# Aim: Using UDP sockets, write a client - server program to make client sending the file name and the server to send back the contents of the requested file if present.

⇒ Code:

Server UDP. py

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))

print ("The server is ready to receive")

while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file = open(sentence, "r")
    con = file.read(2048)

    serverSocket.sendto(bytes(con, "utf-8"), clientAddress)

    print ('In sent contents of', end=' ')
    print (sentence)

    file.close()
```

# clientVDP.py

```
from socket import *
```

```
serverName = "127.0.0.1")
```

```
serverPort = 12000
```

```
clientSocket = socket (AF_INET, SOCK_DGRAM)
```

```
sentence = input ("Enter file name : ")
```

```
clientSocket . sendto (bytes (sentence, "utf-8"),  
(serverName, serverPort))
```

```
filecontents, serverAddress = clientSocket.  
recvfrom (2048)
```

```
print ('Reply from server : \n')
```

```
print (filecontents . decode ("utf-8"))
```

```
clientSocket . close()
```

# Output :-

The server is ready to receive. <sup>7</sup> server  
sent contents of server VDP.py <sup>7</sup> side  
Server is ready to receive.

Enter file name : ServerVDP.py <sup>7</sup> client  
Reply from server:

**Output Screenshot:**

```
PS C:\Users\DELL\OneDrive\Desktop\code> python udpserver.py  
The server is ready to receive
```

```
Sent contents of example.txt
```

```
[REDACTED]
```

```
PS C:\Users\DELL\OneDrive\Desktop\code> python udpclient.py
```

```
Enter file name: example.txt
```

```
Reply from Server:
```

```
Hello, this is a sample file.  
It is used for testing the TCP server.
```

## Program 5

### Tool Exploration – Wireshark

02-Jan-2025

### Experiment - 17 Tool Exploration - Wireshark.

Wireshark is a powerful and widely used network protocol analyzer. It allows us to capture and inspect data packets travelling over networks in real-time, making it a crucial tool for studying computer networks, troubleshooting network issues and understanding the protocols.

#### Key Features:

- 1.) Packet capture : captures live network traffic from various interfaces (en1, ethernet, wi-fi)
- 2.) Protocol Analysis : supports hundreds of protocols (en: TCP, UDP, HTTP, FTP)
- 3.) Filtering : offers powerful filters to isolate specific packets or traffic types.
- 4.) Visualization : Displays packets details with hierarchical layers (Ethernet, IP, TCP / UDP).

## # use cases of Wireshark

### 1.) Network Troubleshooting :

Diagnosing slow network speeds

Identifying causes of a misconfiguration.

### 2.) Security Analysis :

Detecting malicious traffic / intrusions

### 3.) Protocol study.

Understanding packet structures and communication flows.

## # common filters:

1.) http : shows only HTTP traffic.

2.) tcp port == 80 : shows traffic on TCP port 80

3.) ip addr == 192.168.1.1 : shows packets to and from specific IP.

4.) UDP : shows only UDP traffic.