

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



## LAB REPORT

on

## COMPUTER NETWORKS

*Submitted by*

**Gaurav Ramachandra (1BM22CS100)**

*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 **Gaurav Ramachandra (1BM22CS100)**, 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.

**Dr. Latha N.R.**

Associate 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 No.</b>
1	25/09/2024	Create a topology involving multiple hubs and a switch connecting them to simulate with simple PDU.	1
2	9/10/2024	Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply	4
3	16/10/2024	Configure default route, static route to the router	13
4	23/10/2024	Configure DHCP within a LAN and outside LAN.	26
5	13/11/2024	Configure RIP routing Protocol in Routers	41
6	20/11/2024	Configure OSPF Routing Protocol	54
7	27/11/2024	Demonstrate the TTL/ Life of a Packet	41
8	18/12/2024	Configure Web Server & DNS	60
9	18/12/2024	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)	64
10	18/12/2024	To understand the operation of TELNET by accessing the router in the server room from a PC in the IT office.	68
11	18/12/2024	To construct a VLAN and make the PC's communicate among a VLAN	73
12	18/12/2024	To construct a WLAN and make the nodes communicate wirelessly	78

# **INDEX**

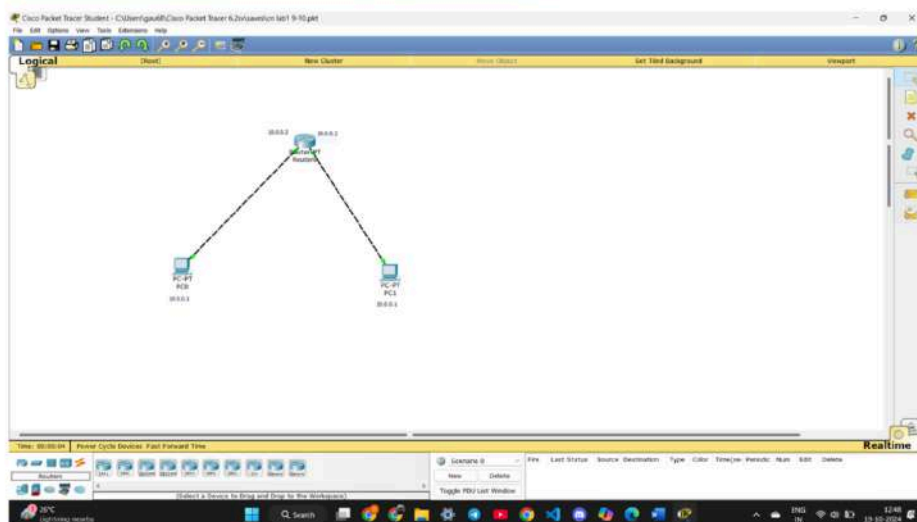
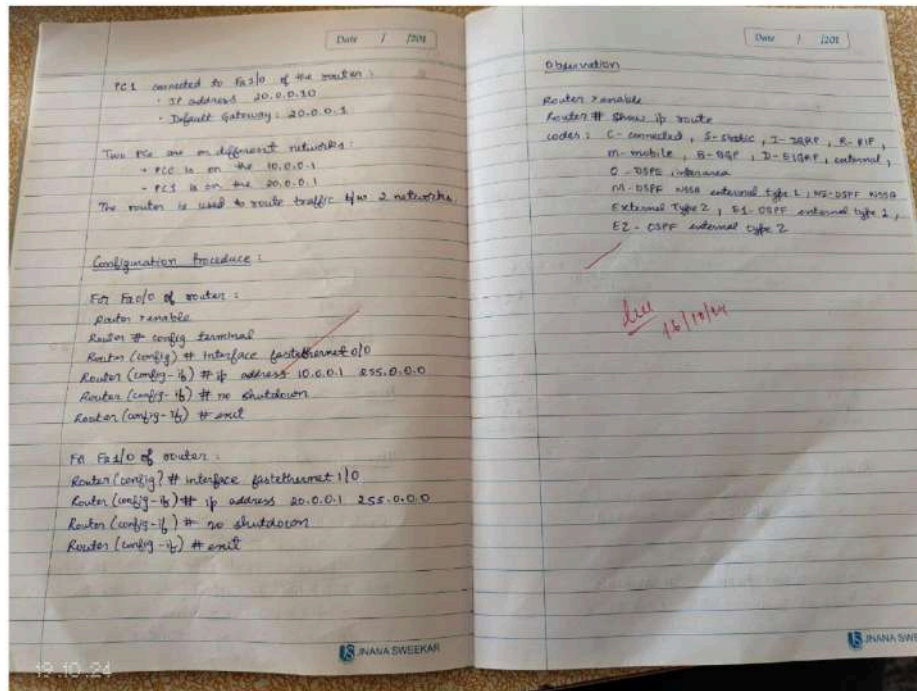
## **CYCLE 2**

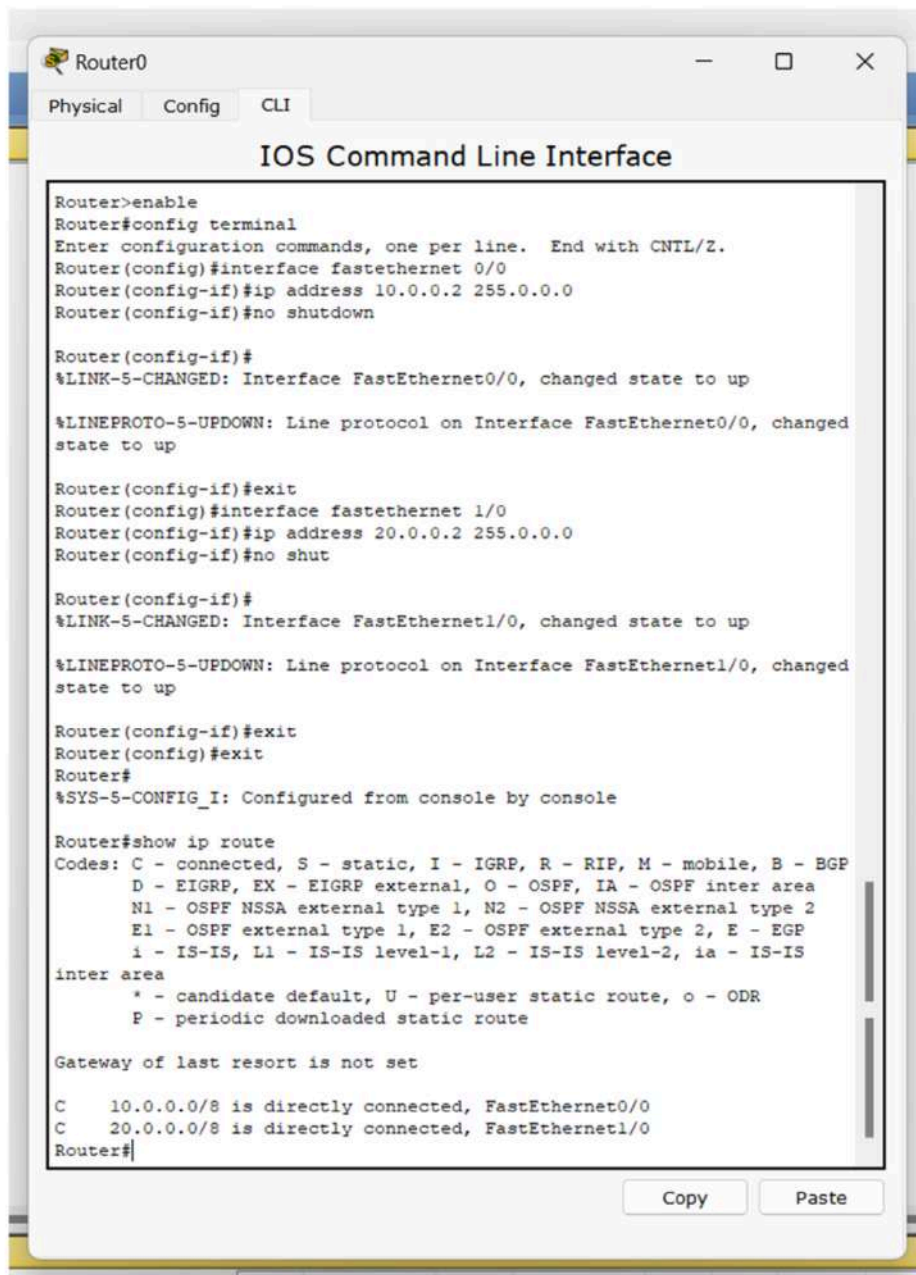
<b>Sl. No.</b>	<b>Date</b>	<b>Experiment Title</b>	<b>Page No.</b>
1	21/12/2024	Write a program for error detecting code using CRC-CCITT (16-bits).	81
2	21/12/2024	Write a program for congestion control using Leaky bucket algorithm	84
3	21/12/2024	Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.	86
4	21/12/2024	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.	90
5	21/12/2024	Tool Exploration - Wireshark	91

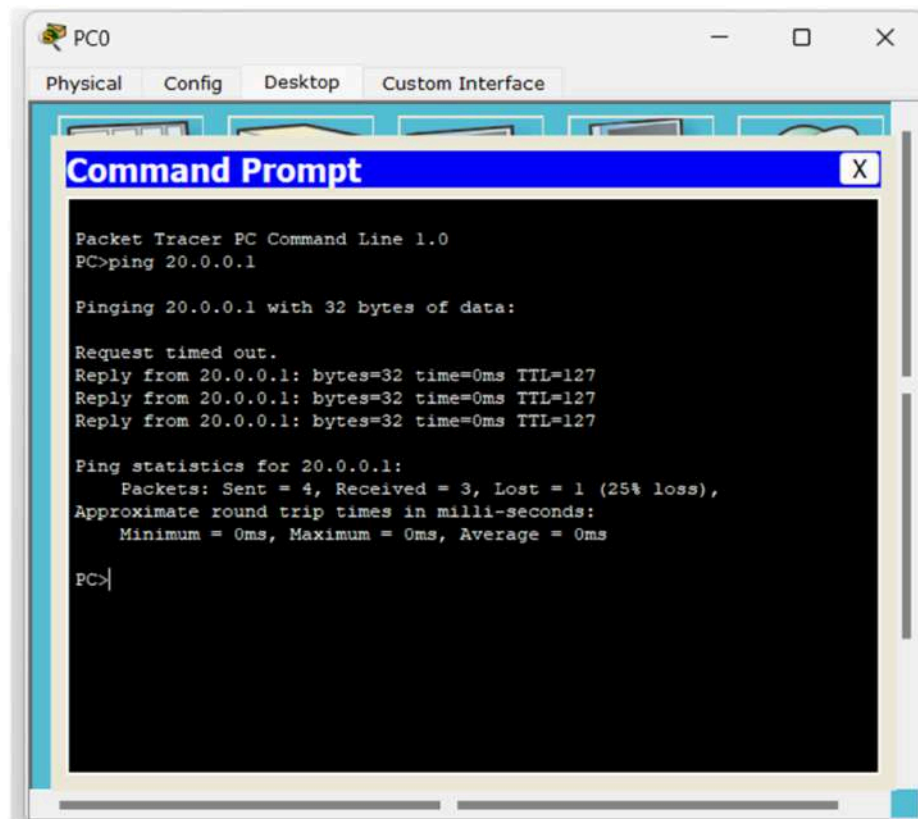
Github Link: [https://github.com/Gaurav-Ramachandra/Sem5-CN\\_Lab](https://github.com/Gaurav-Ramachandra/Sem5-CN_Lab)

## Program 1

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



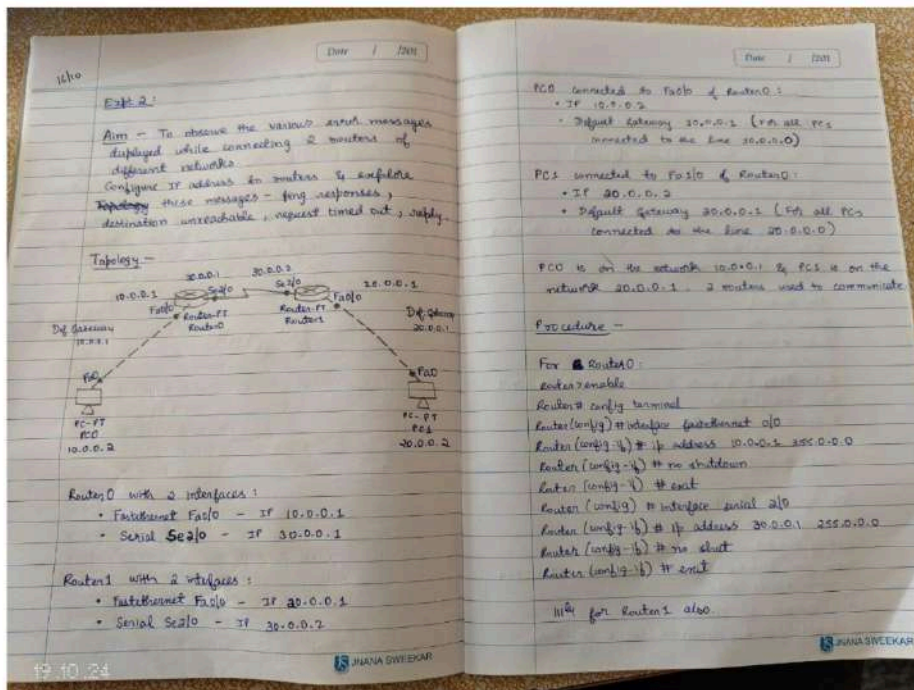




## **Program 2**

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 -

~~PC~~ For PC0 : In Command Prompt

PC > ping 20.0.0.2

Destination host unreachable. [Error]

For Router 0 :-

Router > enable

Router # config terminal

Router (config) # ip route 20.0.0.0 255.0.0.0 30.0.0.2

Router (config) # exit

Router # show ip route

For Router 1 :-

Router > enable

Router # show ip route

Router # config terminal

Router (config) # ip route 10.0.0.0 255.0.0.0 30.0.0.1

Router (config) # exit

Router # show ip route

S 10.0.0.0/8 [1/0] via 30.0.0.1

Router # exit

PC0 command Prompt :-

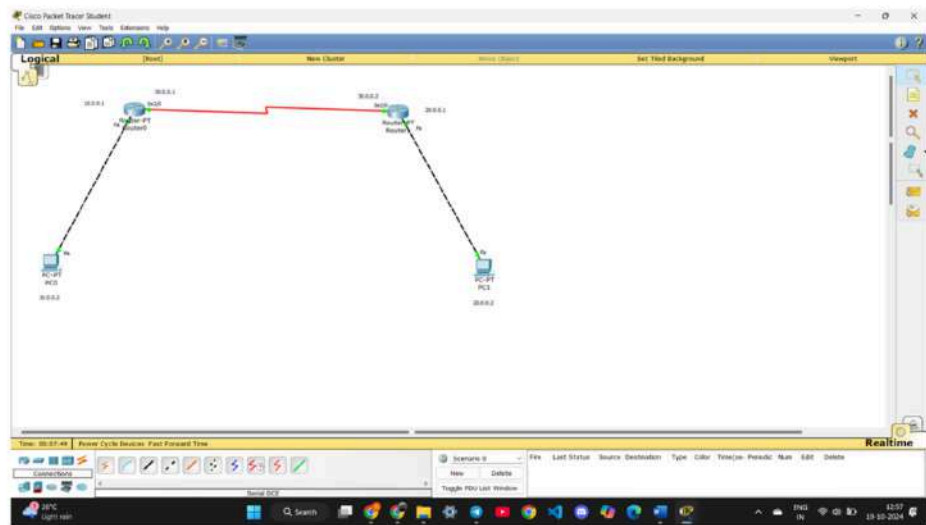
PC > ping 30.0.0.2

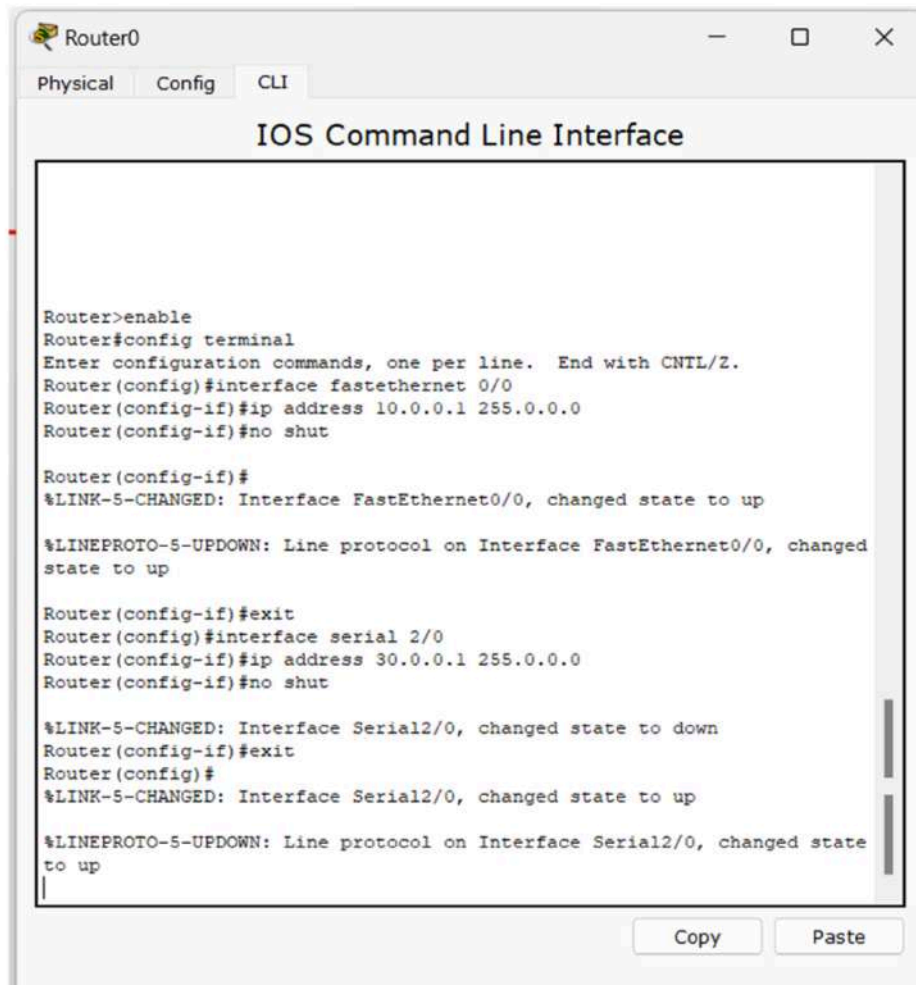
Pinging 30.0.0.2 with 32 bytes of data:

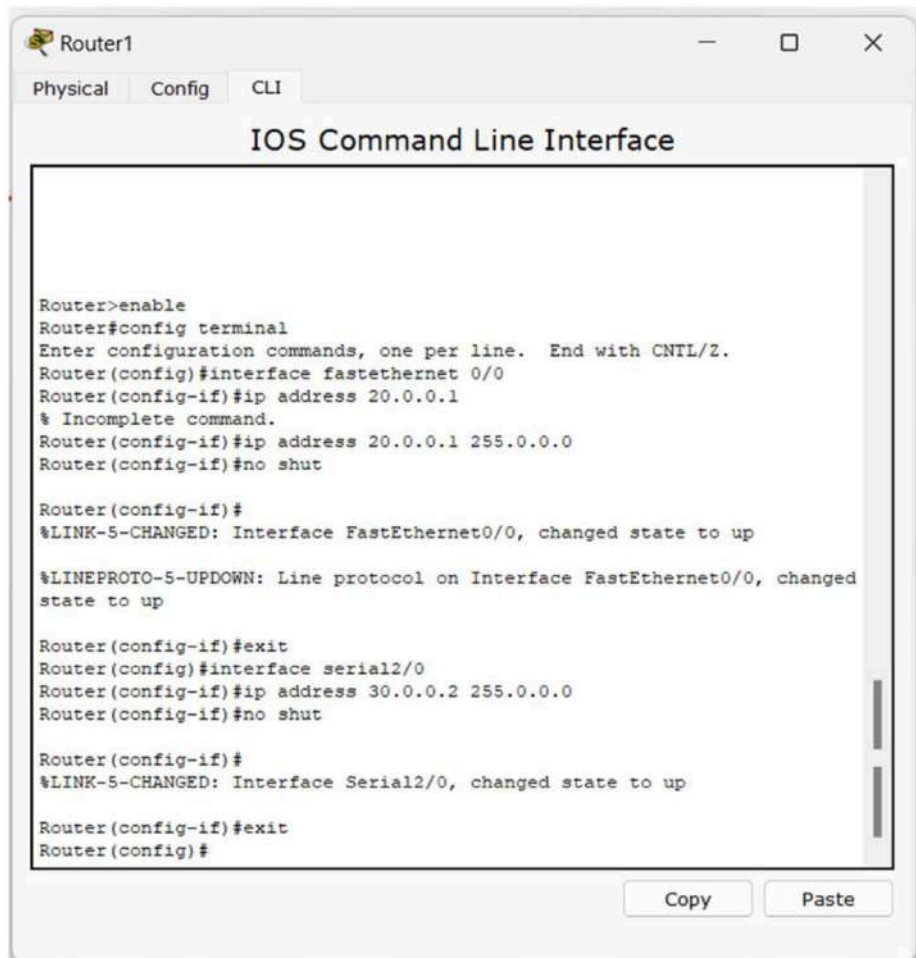
Reply from 30.0.0.2: bytes=32 time=1ms TTL=254  
4 times total

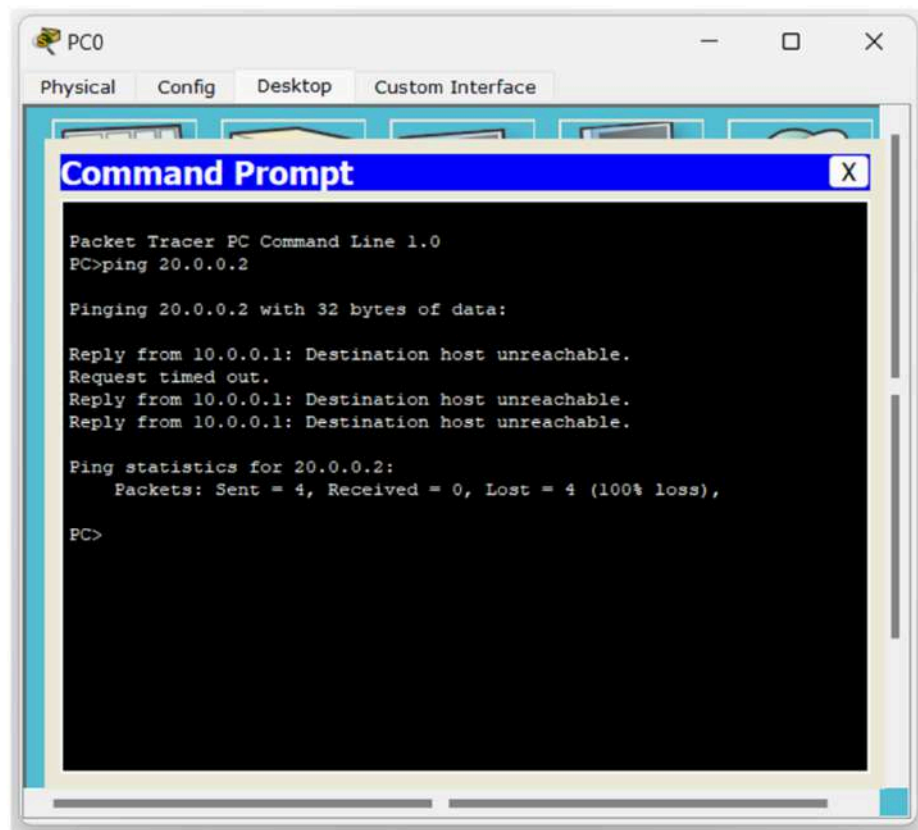
Ping statistics for 30.0.0.2 :

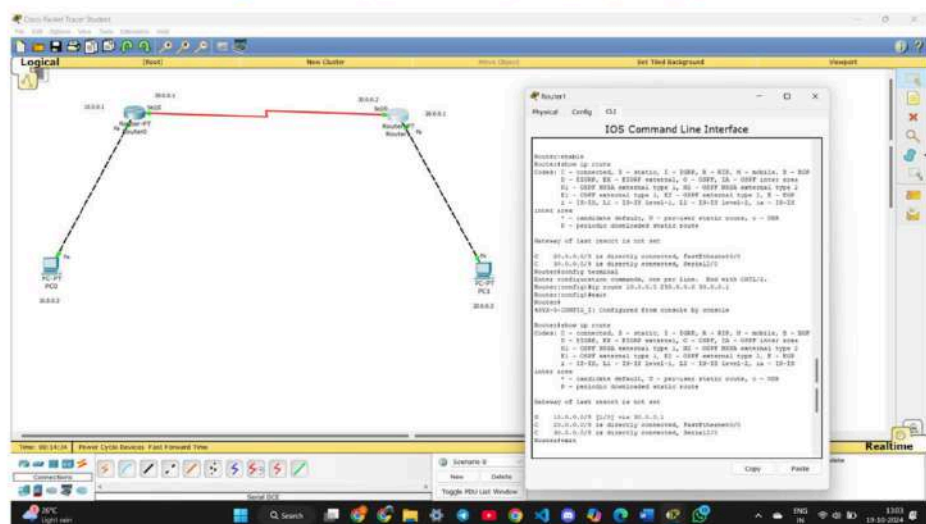
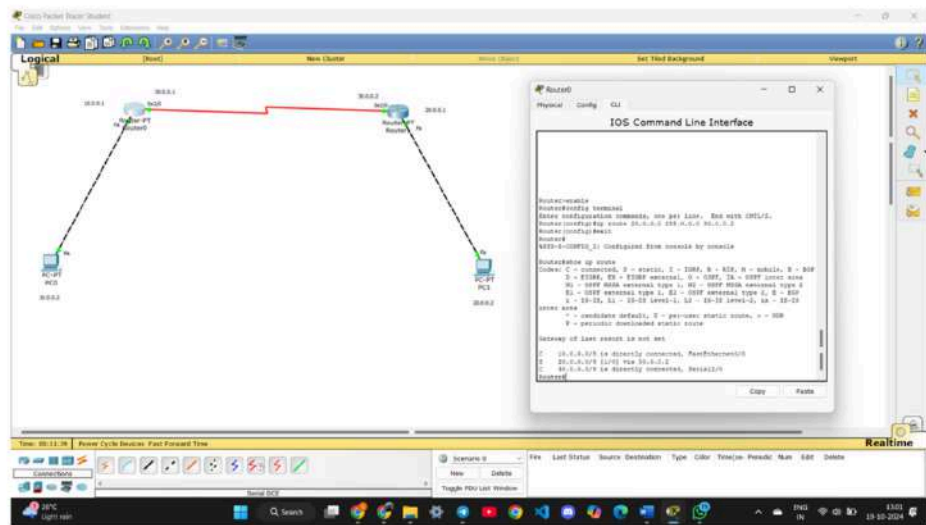
Packets: Sent=4, Received=4, Lost=0 (0% loss)

















### **Program 3**

Configure default route, static route to the Router.

23/10

Expt 3:

Aim - Configure Default Route, Static Route to the Router.

Topology - 3 routers with PCs at their ends

PC0: FastEthernet 0/0  
IP - 10.0.0.2, Def. Gateway - 10.0.0.1

PC1: FastEthernet 0/0  
IP - 40.0.0.2, Def. Gateway - 40.0.0.1

PC2: FastEthernet 0/0  
IP - 30.0.0.2, Def. Gateway - 30.0.0.1

23/10

Procedure:

1. Configure 2 PCs properly
2. Configure 3 routers according to their interface - FastEthernet & Serial.
3. Default route R1:  
# ip route 10.0.0.0 255.0.0.0 20.0.0.1  
# ip route 40.0.0.0 255.0.0.0 30.0.0.1
4. Static route R0:  
# ip route 0.0.0.0 0.0.0.0 20.0.0.1
5. Static route R2:  
# ip route 0.0.0.0 0.0.0.0 30.0.0.1

Date / /201

### Observation :

R0 :

# show ip route

C 10.0.0.0/8 is directly connected, FastEthernet 0/0

C 20.0.0.0/8 is directly connected, Serial 2/0

S\* 0.0.0.0/0 ~~is directly connected~~ [1/0] via 20.0.0.2

R2 :

# show ip route

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

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

S\* 0.0.0.0/0 [1/0] via 30.0.0.1

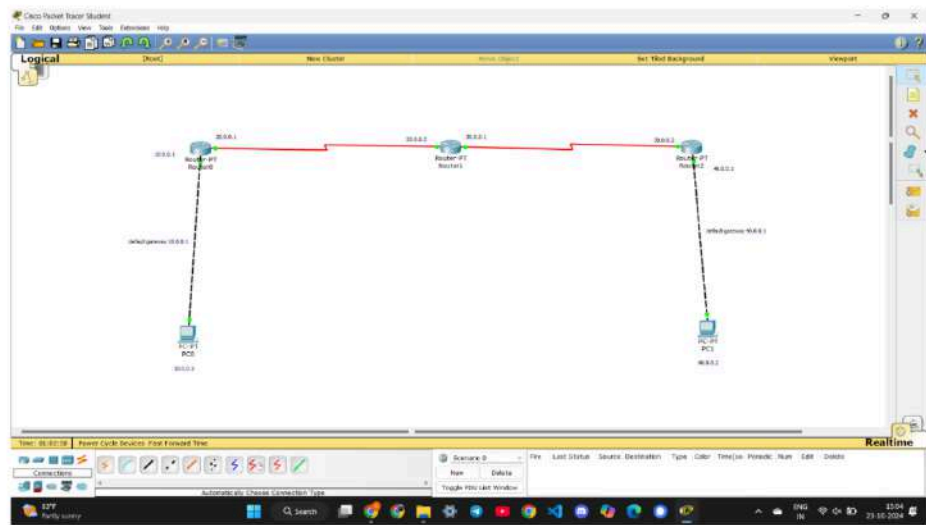
~~PC0 :~~

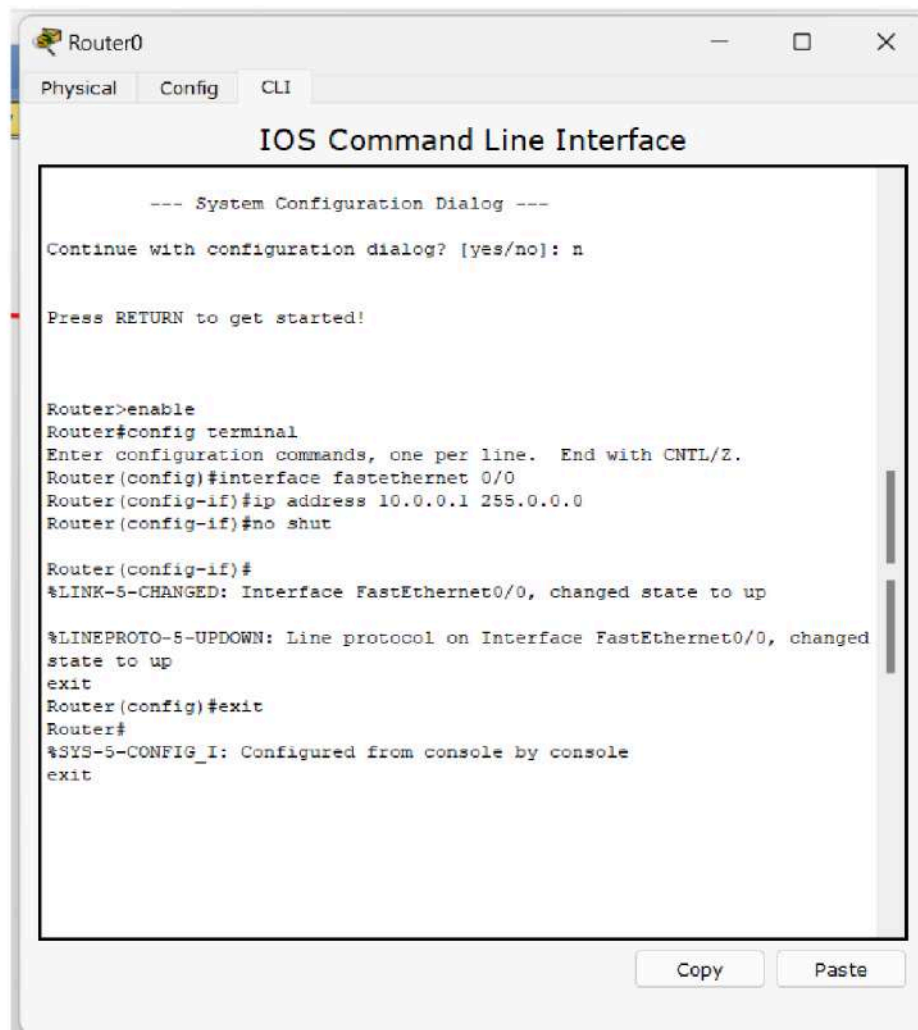
PC > ping 40.0.0.2

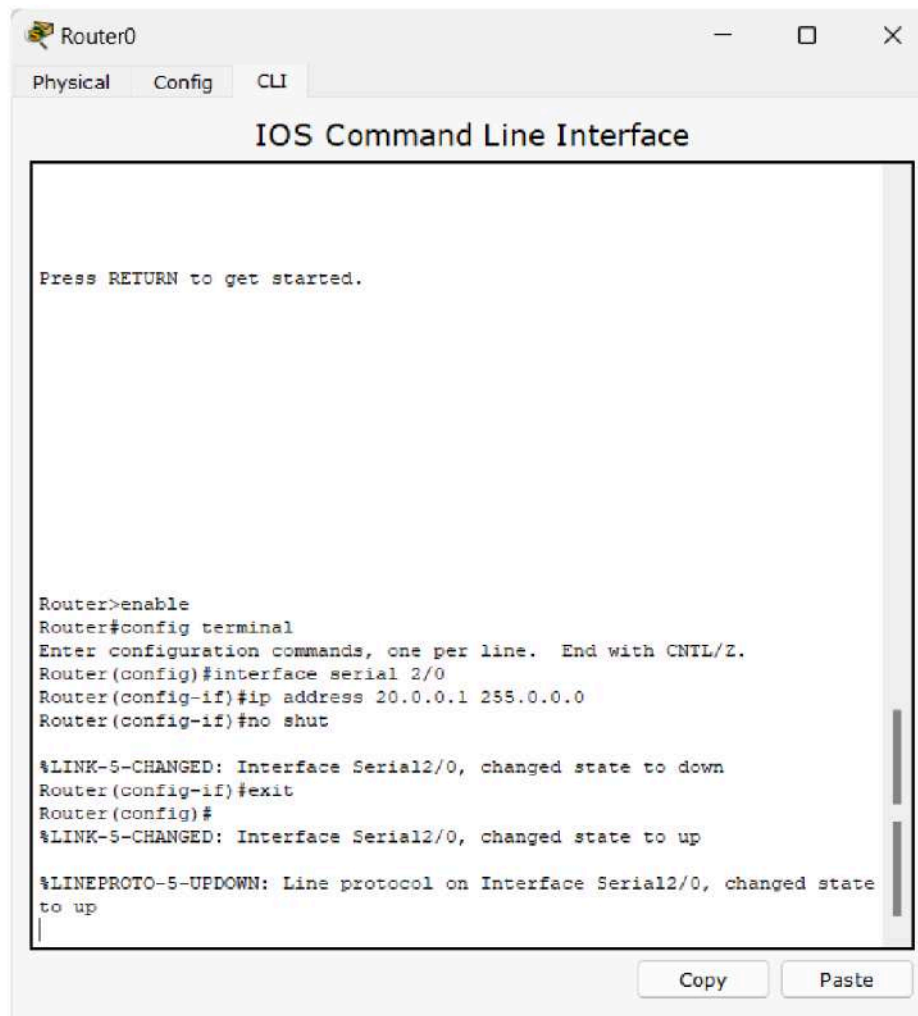
Packets : Sent = 4, Received = 4, Lost = 0 (0% loss)

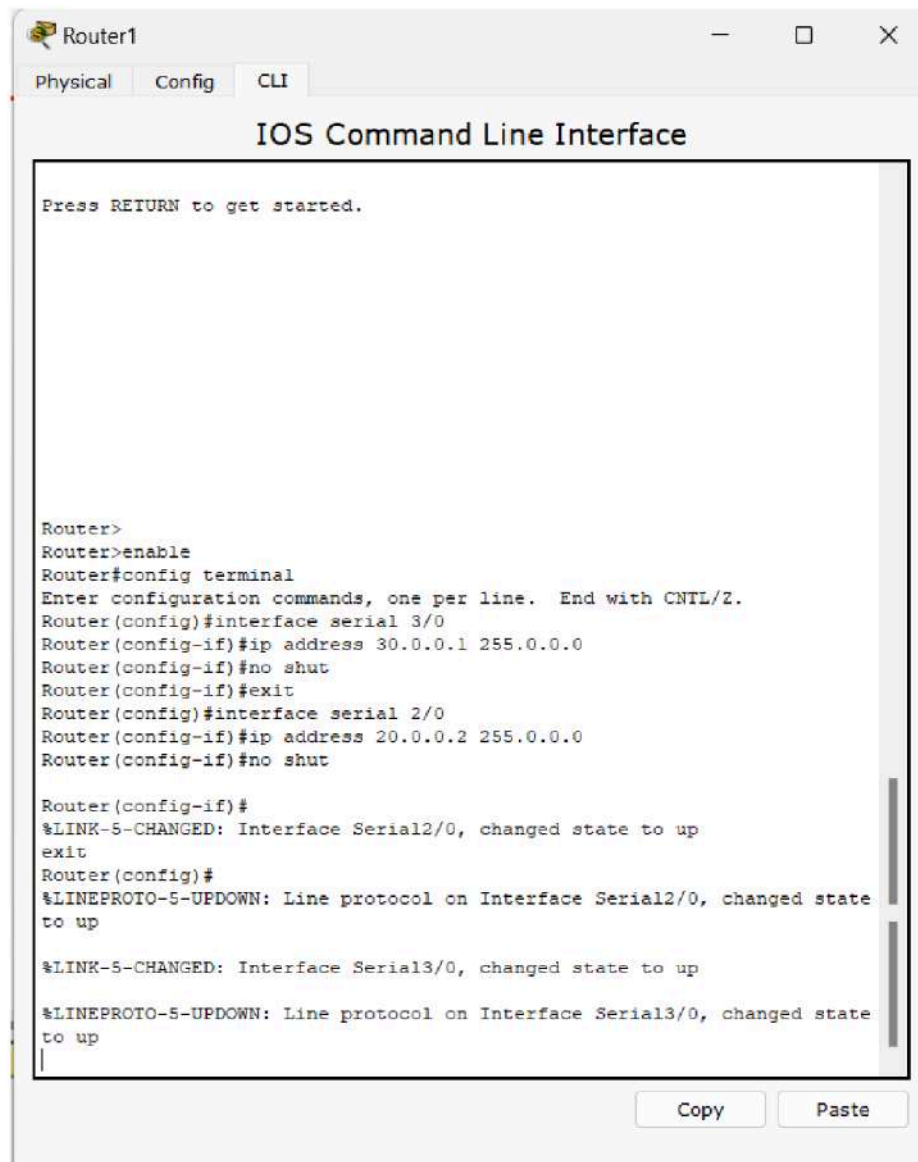
live  
23/10/24

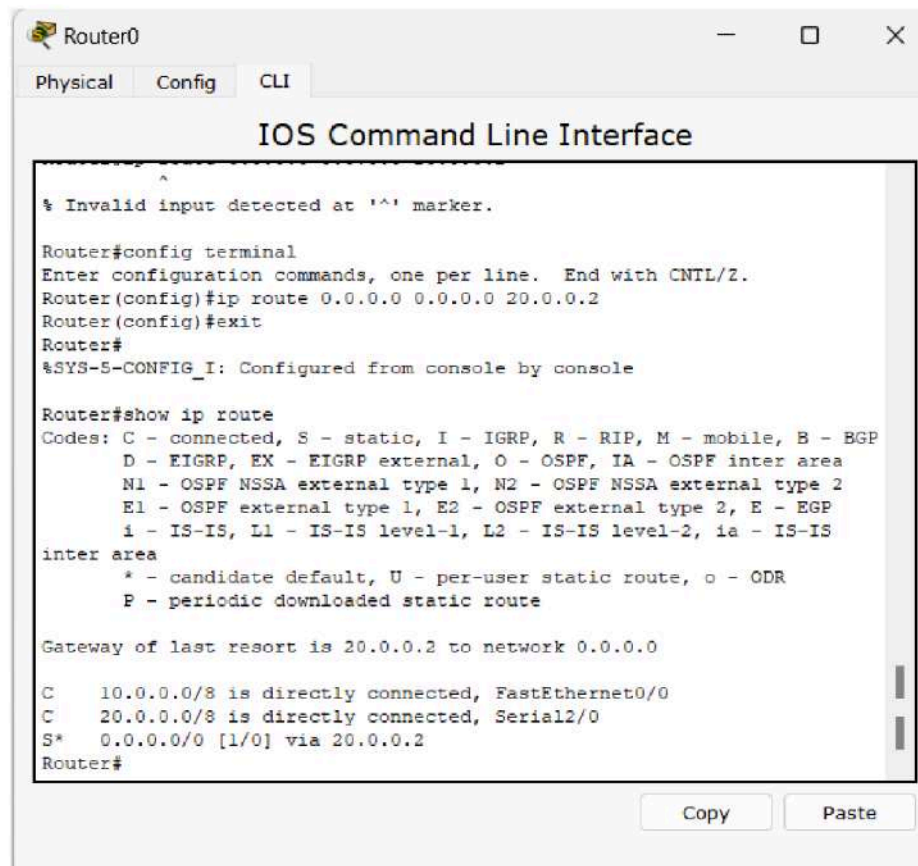
23.10.24











The screenshot shows a window titled "Router0" with three tabs: "Physical", "Config", and "CLI". The "CLI" tab is active, displaying the "IOS Command Line Interface". The terminal text is as follows:

```
% Invalid input detected at '^' marker.

Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 0.0.0.0 0.0.0.0 20.0.0.2
Router(config)#exit
Router#
%SYS-S-CONFIG_I: Configured from console by console

Router#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 20.0.0.2 to network 0.0.0.0

C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    20.0.0.0/8 is directly connected, Serial12/0
S*   0.0.0.0/0 [1/0] via 20.0.0.2
Router#
```

At the bottom right of the CLI window, there are two buttons: "Copy" and "Paste".



Router1

Physical Config CLI

### IOS Command Line Interface

Press RETURN to get started.

```
Router>enable
Router#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    20.0.0.0/8 is directly connected, Serial2/0
C    30.0.0.0/8 is directly connected, Serial3/0
Router#config terminal
Enter configuration commands, one per line. End with CNTRL/Z.
Router(config)#ip route 10.0.0.0 255.0.0.0 20.0.0.1
Router(config)#ip route 40.0.0.0 255.0.0.0 30.0.0.2
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#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

S    10.0.0.0/8 [1/0] via 20.0.0.1
C    20.0.0.0/8 is directly connected, Serial2/0
C    30.0.0.0/8 is directly connected, Serial3/0
S    40.0.0.0/8 [1/0] via 30.0.0.2
Router#
```

Copy Paste

Router2

PhysicalConfigCLI

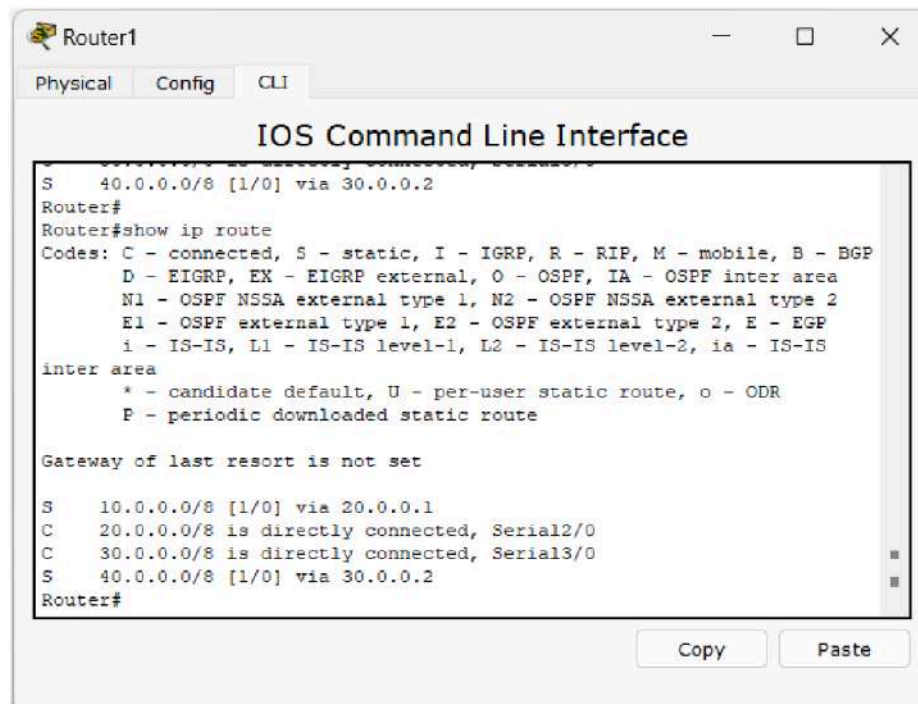
IOS Command Line Interface

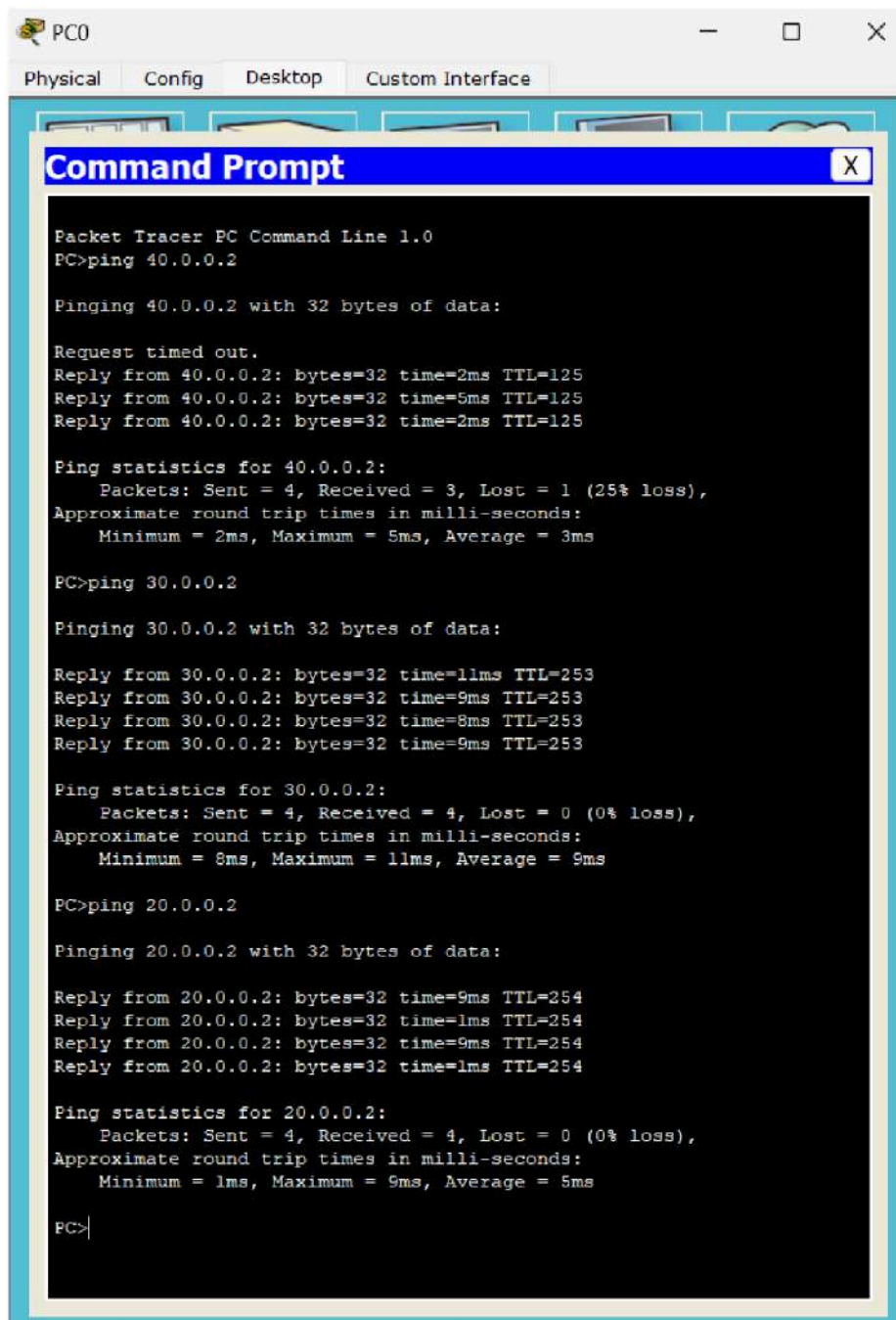
```
Router#enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 0.0.0.0 0.0.0.0 30.0.0.1
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
s
% Ambiguous command: "s"
Router#show ip route
Codes: C - connected, 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

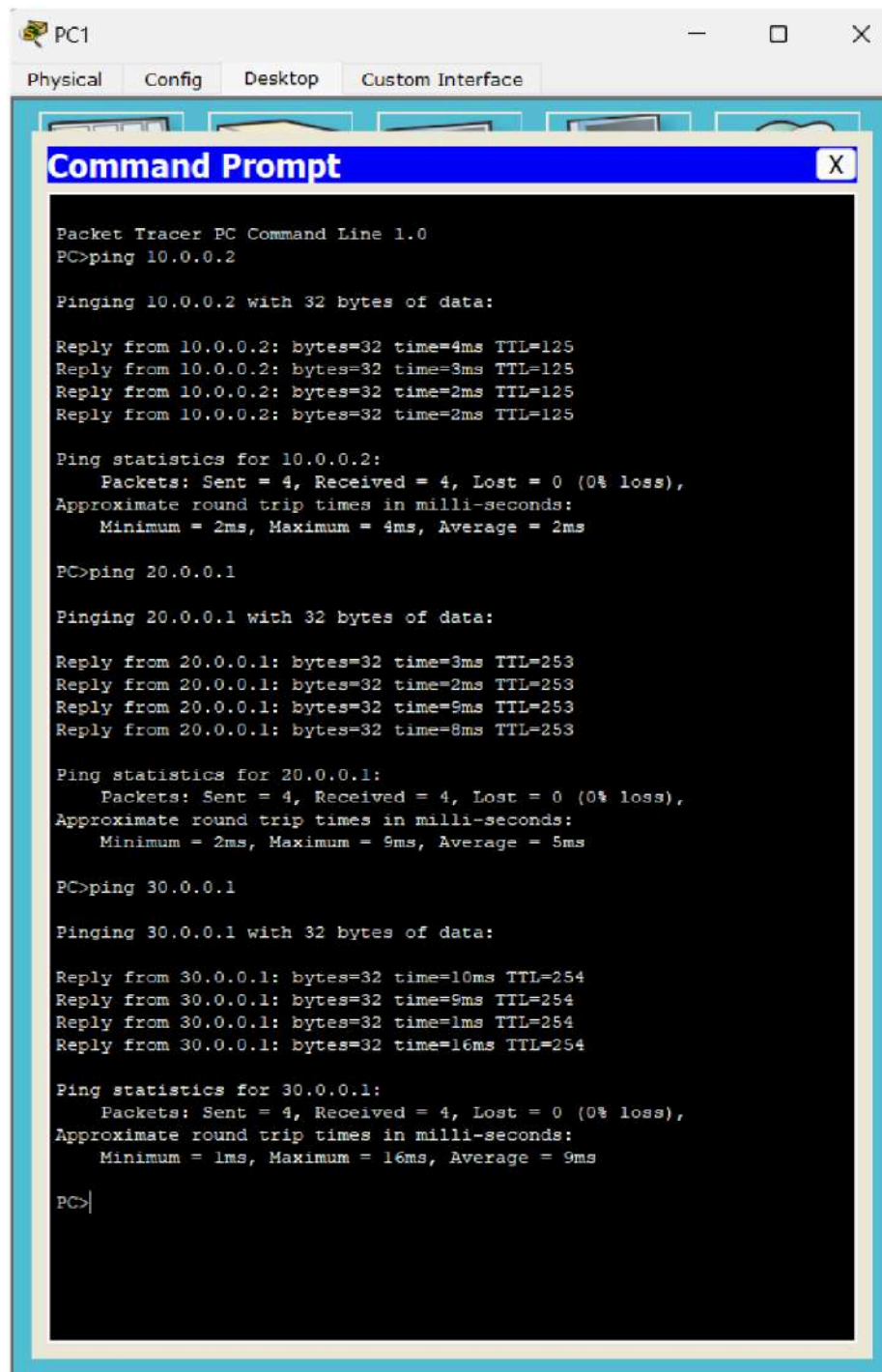
Gateway of last resort is 30.0.0.1 to network 0.0.0.0

C    30.0.0.0/8 is directly connected, Serial2/0
C    40.0.0.0/8 is directly connected, FastEthernet0/0
S*   0.0.0.0/0 [1/0] via 30.0.0.1
Router#
```

CopyPaste







#### **Program 4**

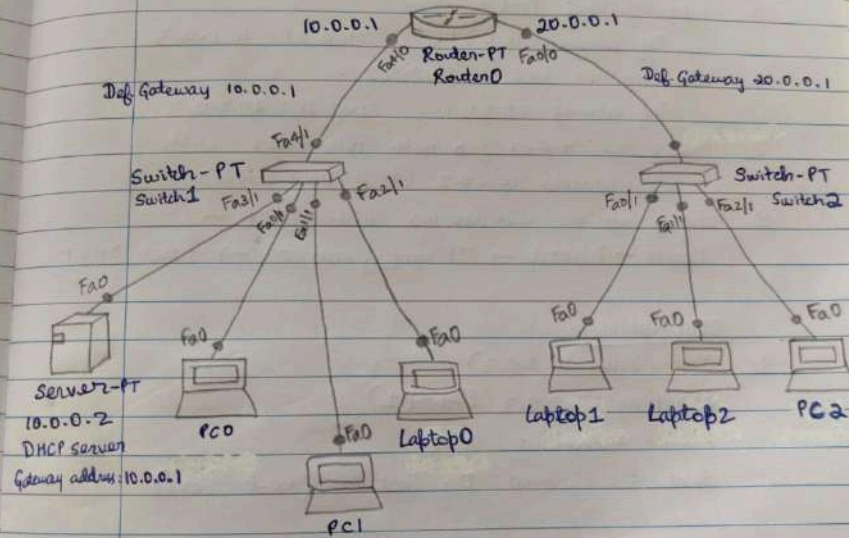
Configure DHCP within a LAN and outside LAN.

13/11

Date / /201

Expt. 4 :

Aim - To configure DHCP <sup>server</sup> within a LAN & outside LAN. ~~using~~

Topology - Part 2



Exam / Day

Procedure:

1. Setup the first network (Part 1)
2. Add another network with a switch & 3 end devices
3. Add a switch to connect the 2 networks
4. Edit the IP config of DHCP Server -  
Change IP to 10.0.0.2
5. Edit Switch1 Port -  
Def Gateway: 10.0.0.1, Static IP: 10.0.0.3
6. Add another Port - Switch2 -  
Def Gateway: 10.0.0.1, Static IP: 10.0.0.3
7. Link the switch for both networks with  
helper address 10.0.0.2 (IP of DHCP server)
8. Setup all the end devices with DHCP -  
Device → Desktop → IP configuration → Select DHCP

Observations:

PC0 (with 10.0.0.3):  
PC0 ping 10.0.0.3 (up & 1)  
Request timed out X 4  
Sent=4, Received=0, lost=4 (100% loss)

PC0 ping 10.0.0.4 (PC1)  
Reply from 10.0.0.4: bytes=32 time=1ms TTL=128 X 4  
Sent=4, Received=4, lost=0 (0% loss)

PC0 ping 10.0.0.5 (up & 0)  
Reply from 10.0.0.5: bytes=32 time=0ms TTL=128 X 4  
Sent=4, Received=4, lost=0 (0% loss)

Exam / Day

Topology - Part 2

Procedure:

1. Connect the topology as above
2. Server → Desktop → IP config. → IP address: 10.0.0.1  
Def Gateway: 10.0.0.0
3. Config → all → Server → Config → DHCP →  
Switch1, Def Gateway: 10.0.0.1, Static IP: 10.0.0.3
4. End Devices → Desktop → IP config → DHCP

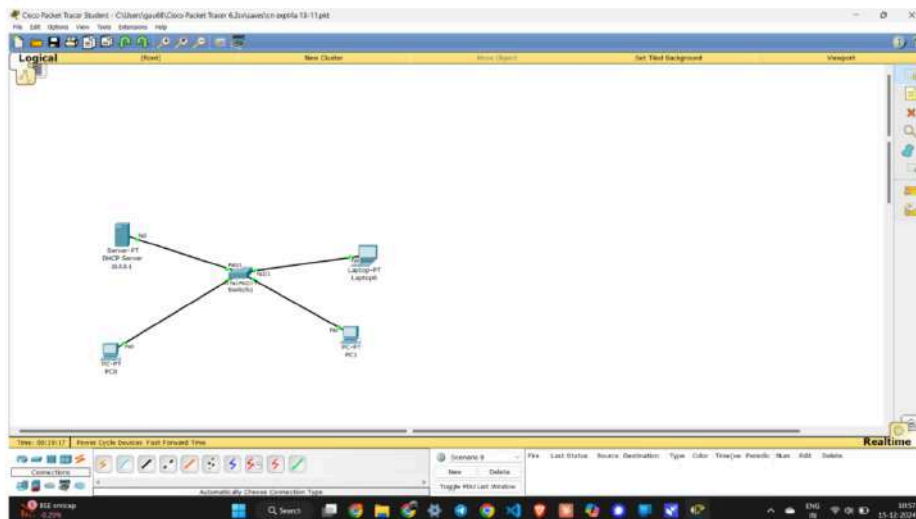
Observations:

5. IP address is allocated dynamically to end devices

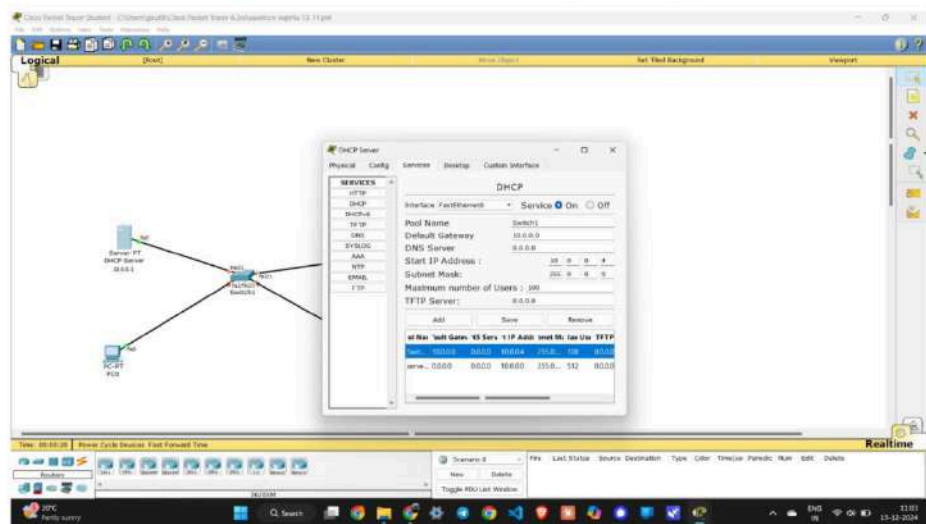
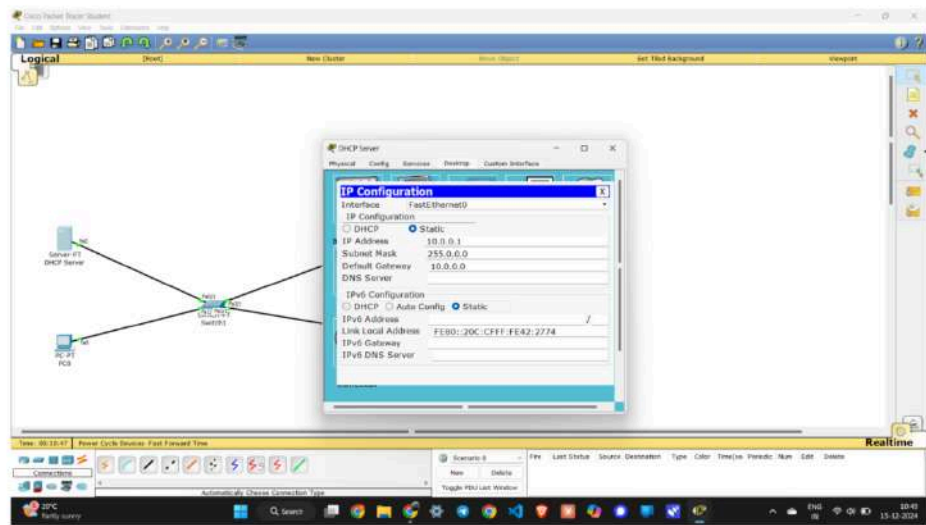
6. Ping from 3 end devices to another was successful.

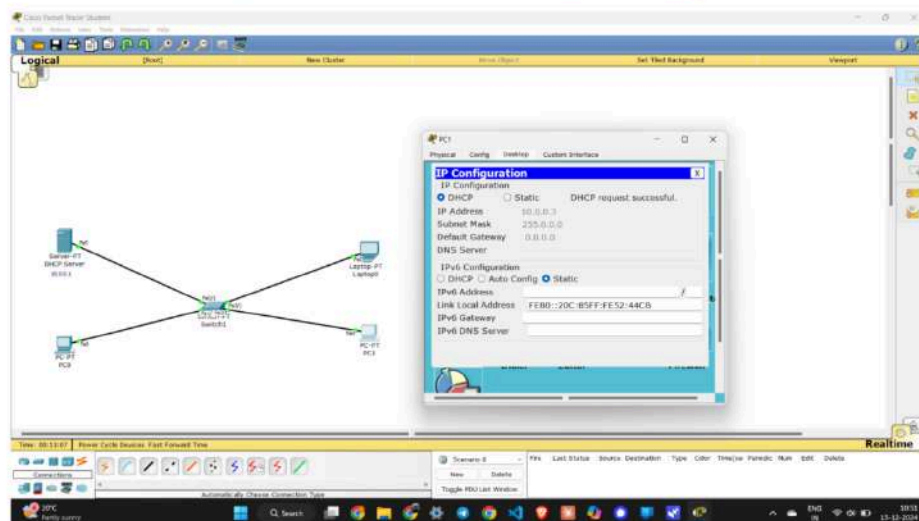
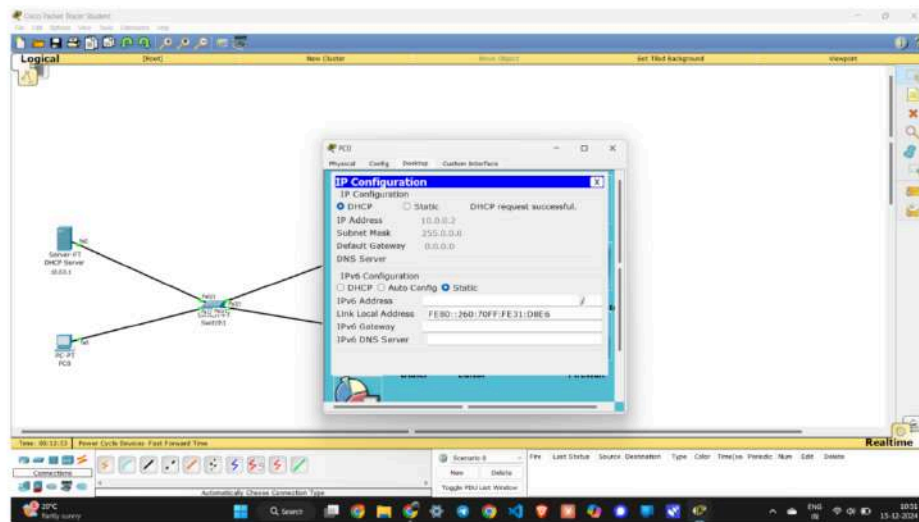
PC0 (with 10.0.0.3):  
PC0 ping 10.0.0.3 (PC3)  
Reply from 10.0.0.3: bytes=32 time=1ms TTL=128 X 4  
Sent=4, Received=4, lost=0 (0% loss)

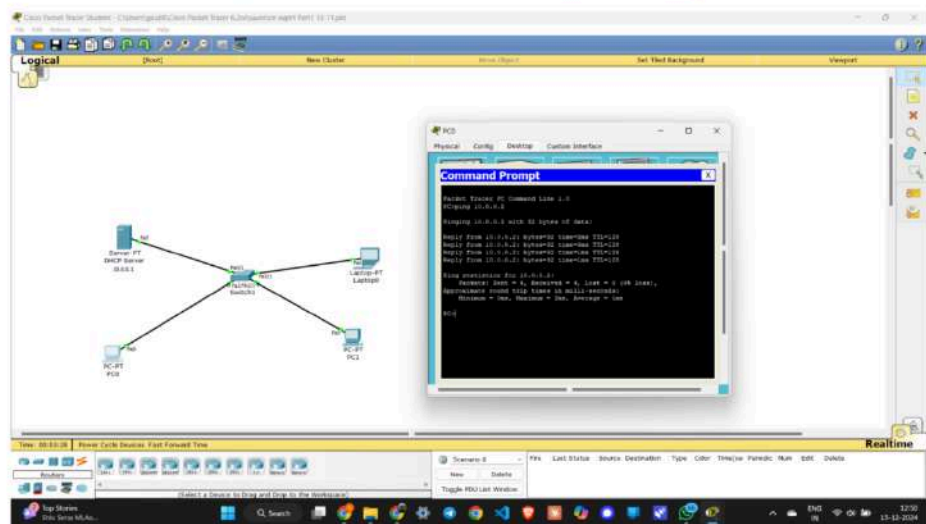
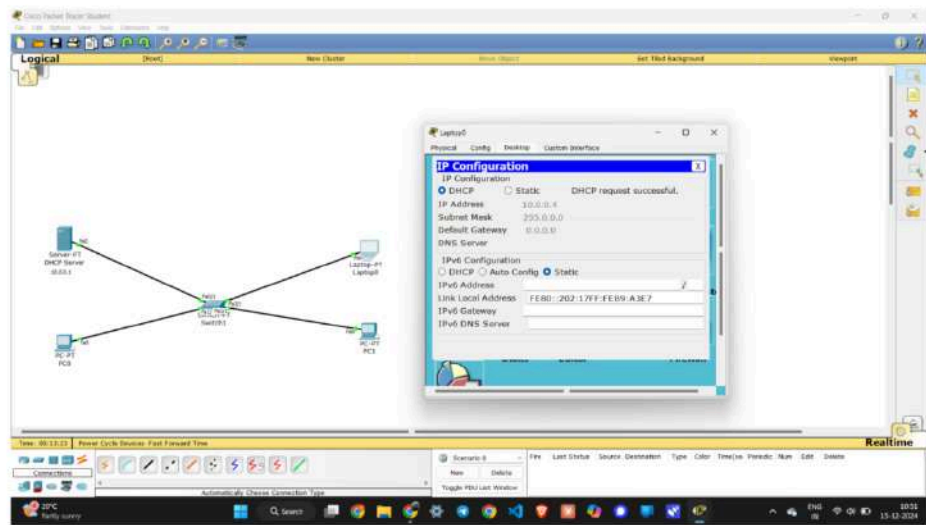
## Part1:



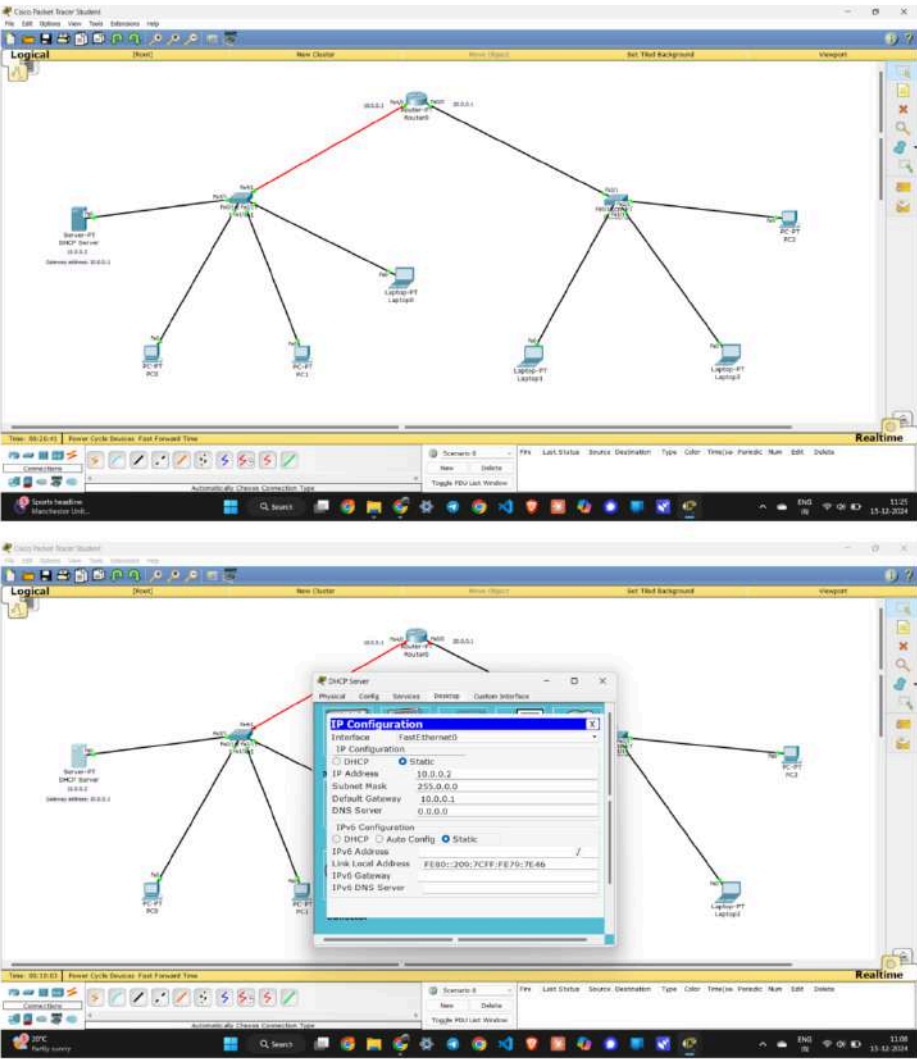


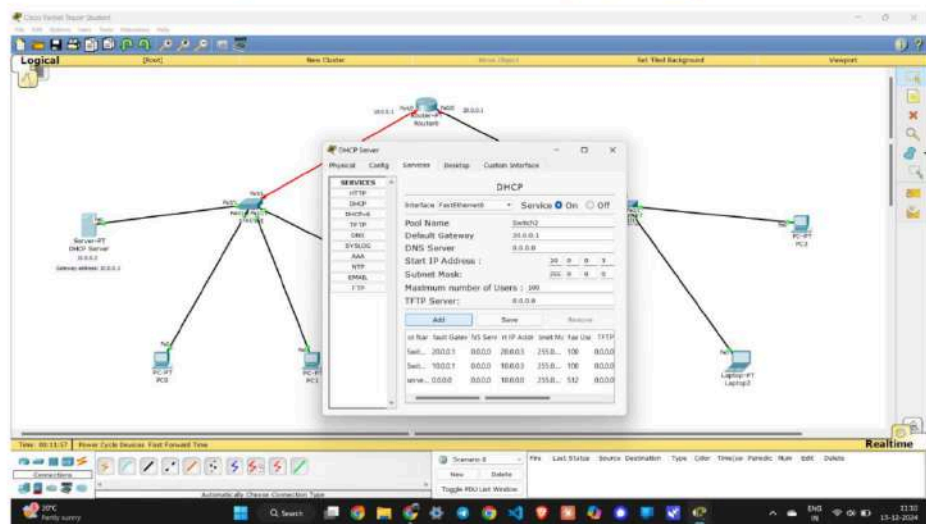
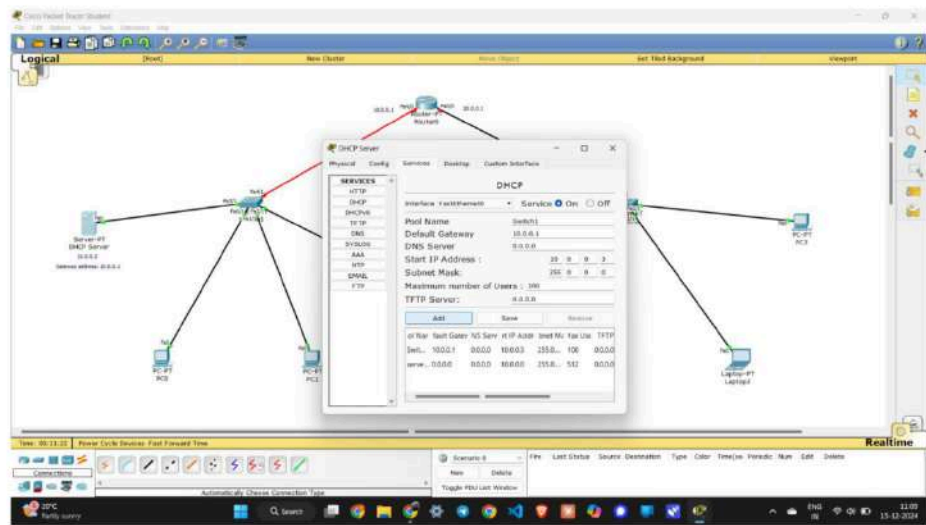


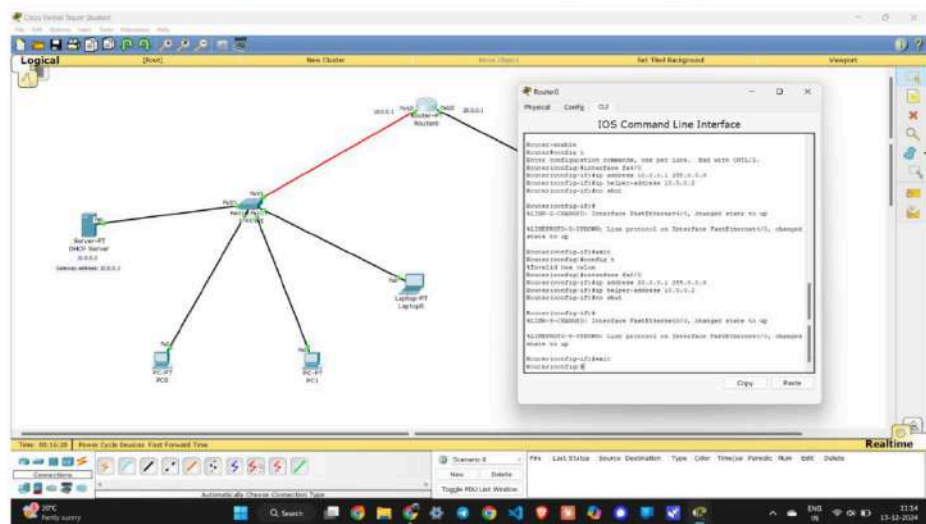
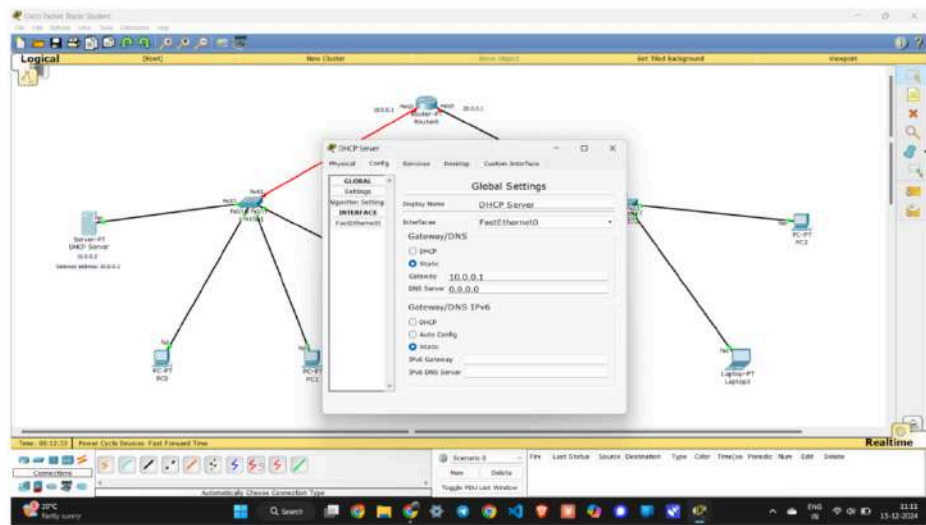


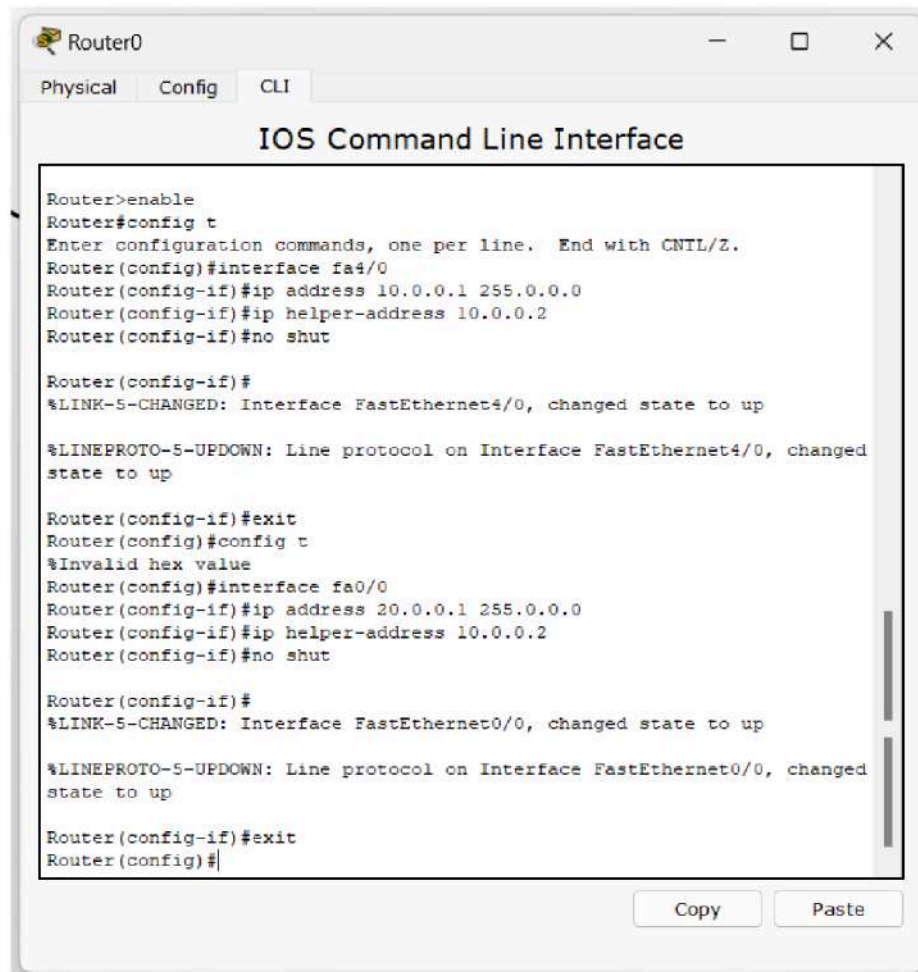


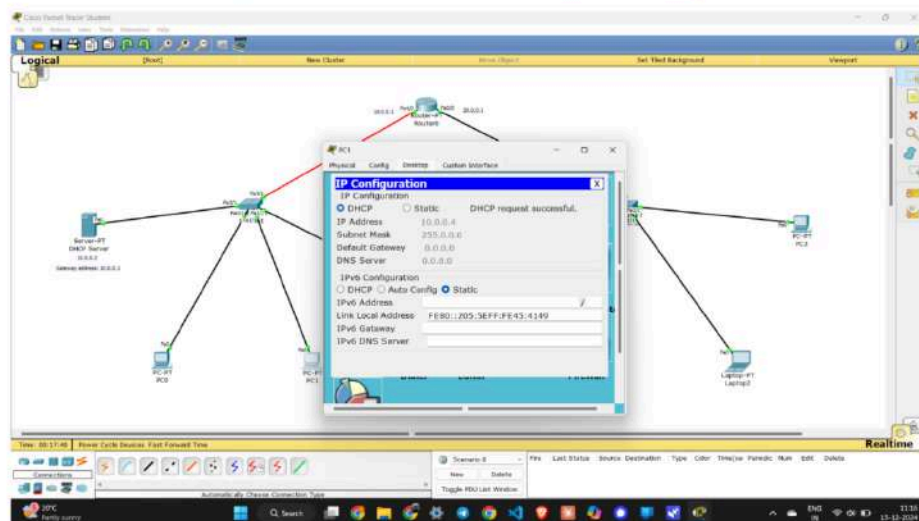
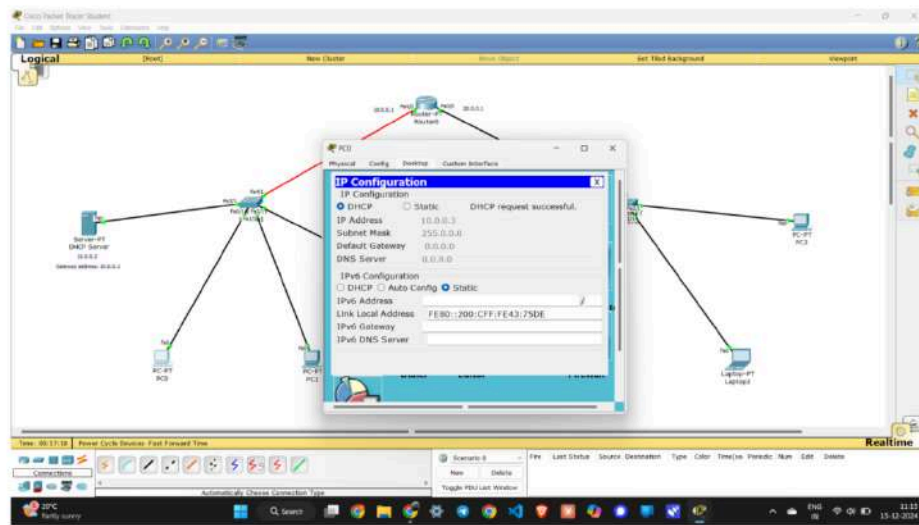
Part2:



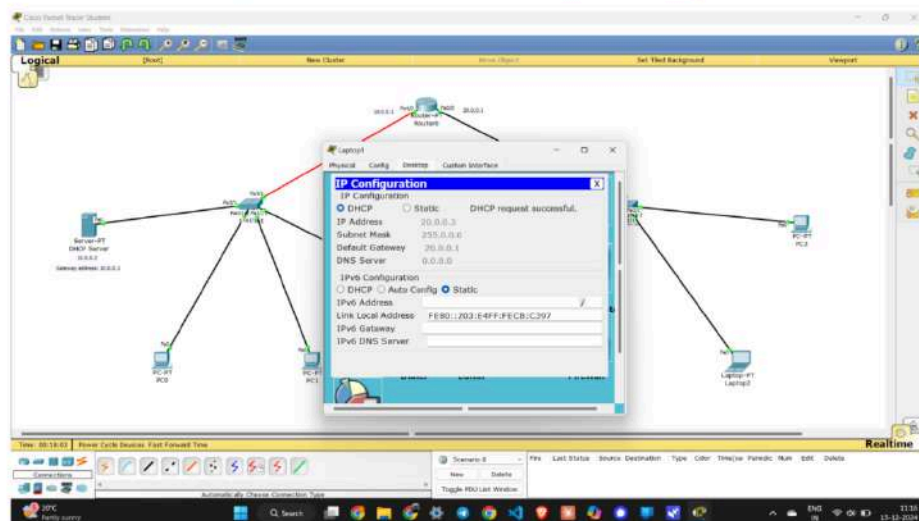
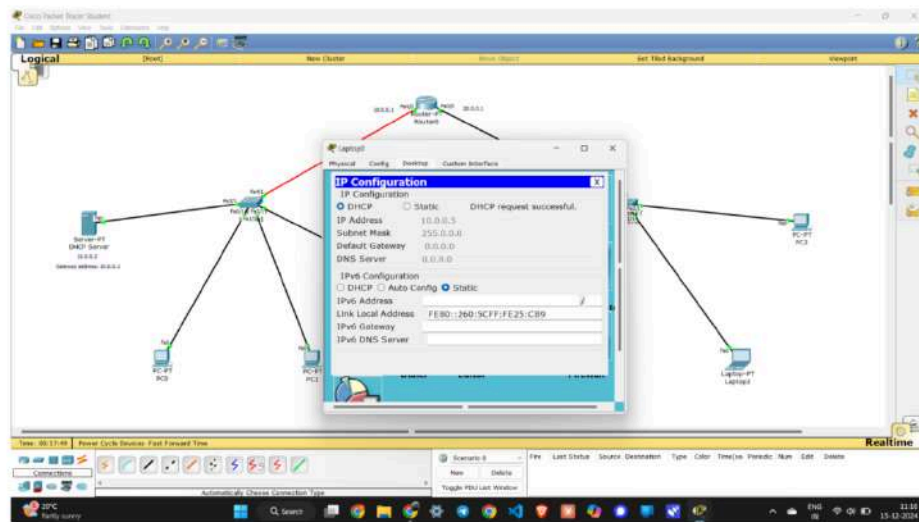


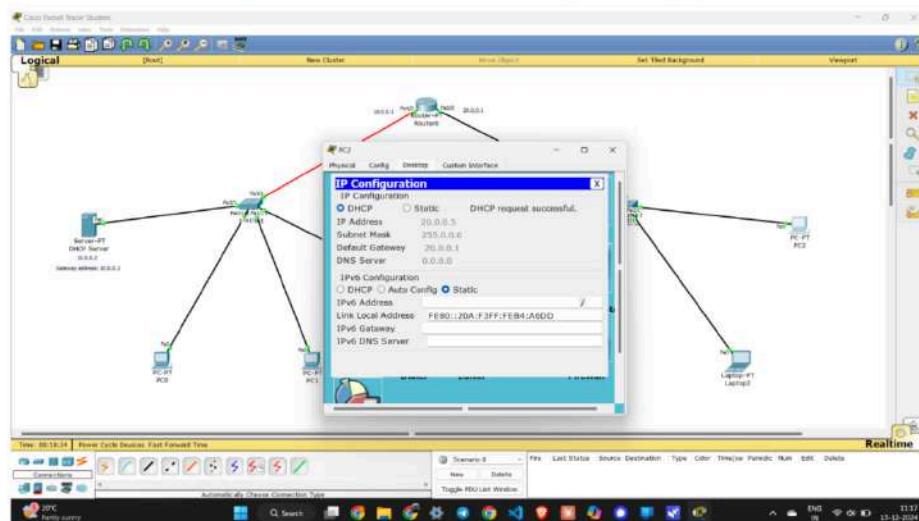
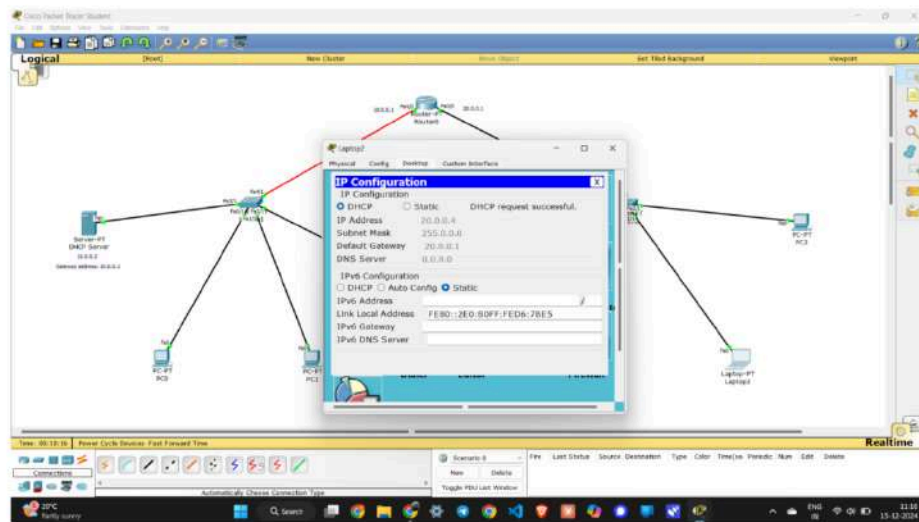


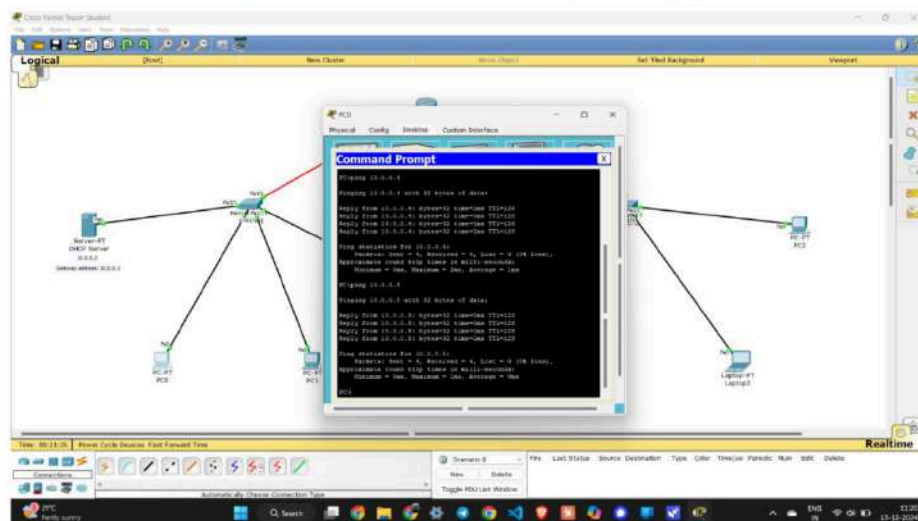








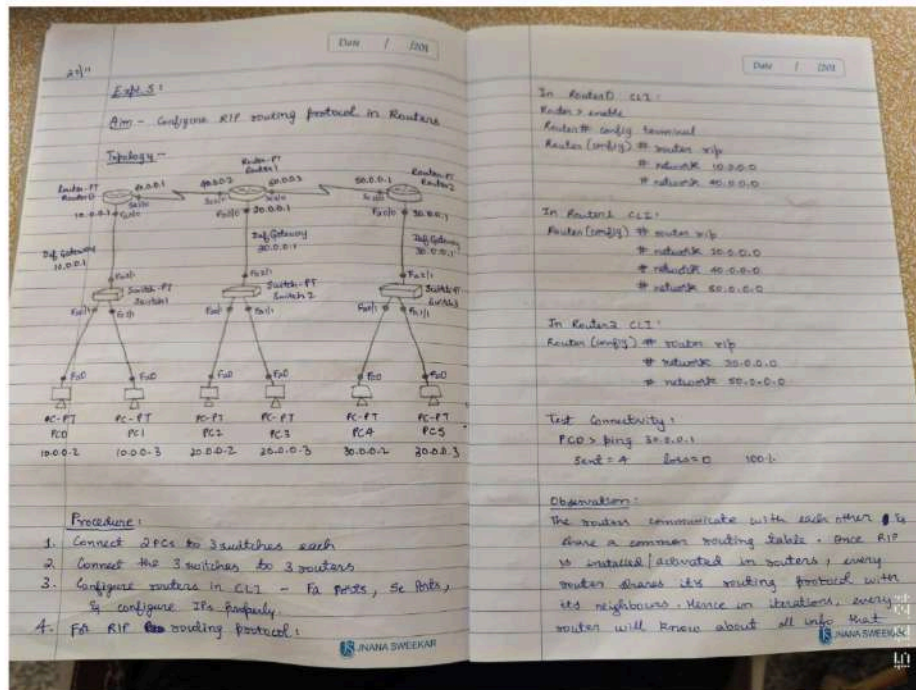


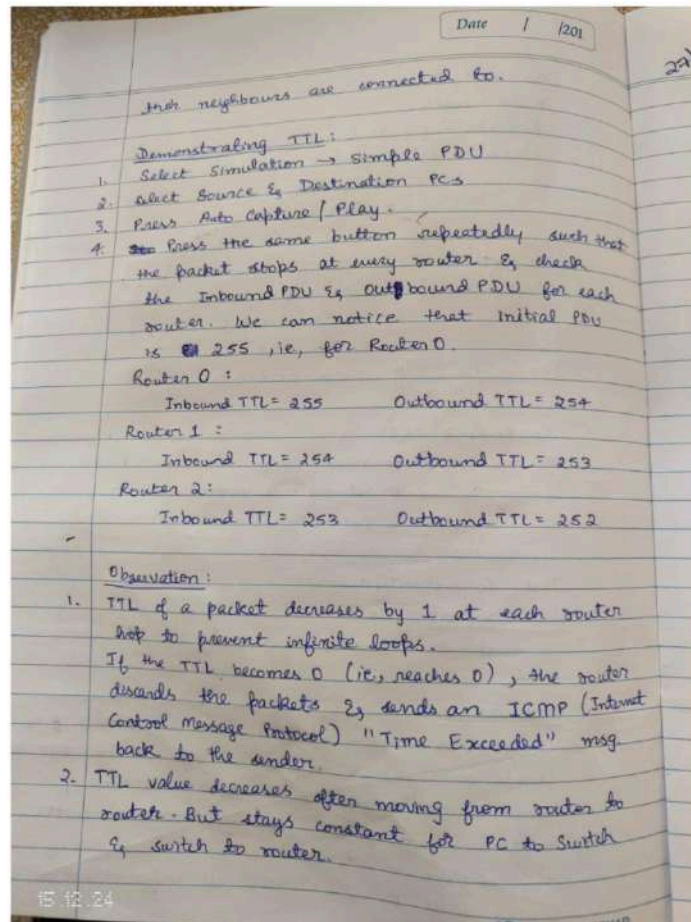




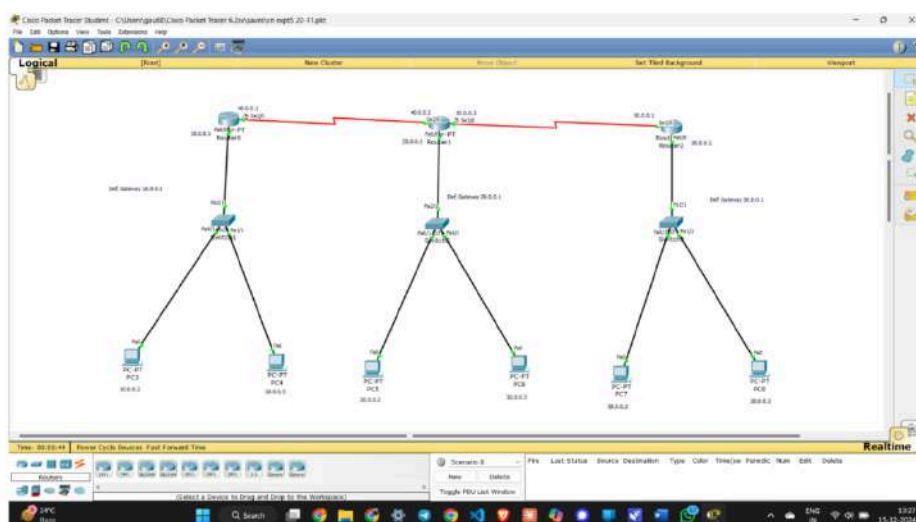
## Program 5,7

Configure RIP routing Protocol in Routers  
Demonstrate the TTL/ Life of a Packet

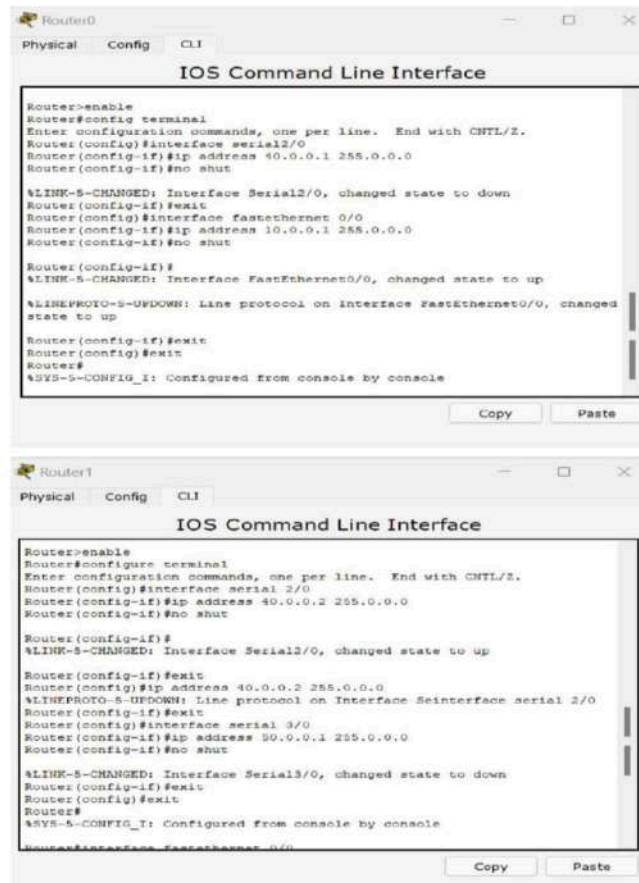




## Topology:



## Configure Network:



The image displays two screenshots of the Cisco IOS Command Line Interface (CLI) for two routers, Router0 and Router1. Both windows show the configuration process for interfaces Serial2/0 and Serial3/0, including IP address assignment and status changes.

**Router0 CLI Screenshot:**

```
Router0>enable
Router0#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router0(config)#interface serial2/0
Router0(config-if)#ip address 40.0.0.1 255.0.0.0
Router0(config-if)#no shut

%LINK-5-CHANGED: Interface Serial2/0, changed state to down
Router0(config-if)#exit
Router0(config)#interface fastethernet0/0
Router0(config-if)#ip address 10.0.0.1 255.0.0.0
Router0(config-if)#no shut

Router0(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed
state to up
Router0(config-if)#exit
Router0(config)#exit
Router0#
%SYS-5-CONFIG_I: Configured from console by console
```

**Router1 CLI Screenshot:**

```
Router1>enable
Router1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router1(config)#interface serial 2/0
Router1(config-if)#ip address 40.0.0.2 255.0.0.0
Router1(config-if)#no shut

Router1(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
Router1(config-if)#exit
Router1(config)#ip address 40.0.0.2 255.0.0.0
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0
Router1(config-if)#exit
Router1(config)#interface serial 3/0
Router1(config-if)#ip address 50.0.0.1 255.0.0.0
Router1(config-if)#no shut

%LINK-5-CHANGED: Interface Serial3/0, changed state to down
Router1(config-if)#exit
Router1(config)#exit
Router1#
%SYS-5-CONFIG_I: Configured from console by console
Router1#
```



Router1

Physical Config CLI

### IOS Command Line Interface

```

Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#interface fastethernet 0/0
^
% Invalid input detected at '^' marker.

Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface fastethernet 0/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed
state to up
exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
Router#

```

Copy Paste

Router2

Physical Config CLI

### IOS Command Line Interface

```

Router#enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface serial 3/0
Router(config-if)#ip address 50.0.0.2 255.0.0.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface Serial3/0, changed state to up
exit
Router(config)#inte
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state
to up
% Incomplete command.
Router(config)#interface fastethernet 0/0
^
% Invalid input detected at '^' marker.

Router(config)#interface fastethernet 0/0
Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

```

Copy Paste

## Configure Routing:

### Router 0:

```
Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 40.0.0.0
Router(config-router)#network 10.0.0.0
Router(config-router)#
```

### Router 1:

```
Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 40.0.0.0
Router(config-router)#network 20.0.0.0
Router(config-router)#network 50.0.0.0
Router(config-router)#network 10.0.0.0
Router(config-router)#exit
Router(config)#
```

### Router 2:

```
Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 50.0.0.0
Router(config-router)#network 30.0.0.0
Router(config-router)#
```

Router2

Physical Config CLI

### IOS Command Line Interface

```
Router>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

R    10.0.0.0/8 [120/2] via 50.0.0.1, 00:00:25, Serial3/0
R    20.0.0.0/8 [120/1] via 50.0.0.1, 00:00:25, Serial3/0
C    30.0.0.0/8 is directly connected, FastEthernet0/0
R    40.0.0.0/8 [120/1] via 50.0.0.1, 00:00:25, Serial3/0
C    50.0.0.0/8 is directly connected, Serial3/0
Router>
```

Copy Paste

## Test Connectivity:

From PC0, ping PC4 –

```
Command Prompt

Pinging 20.0.0.3 with 32 bytes of data:

Request timed out.
Reply from 20.0.0.3: bytes=32 time=12ms TTL=126
Reply from 20.0.0.3: bytes=32 time=6ms TTL=126
Reply from 20.0.0.3: bytes=32 time=3ms TTL=126

Ping statistics for 20.0.0.3:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 3ms, Maximum = 12ms, Average = 7ms

PC>ping 30.0.0.2

Pinging 30.0.0.2 with 32 bytes of data:

Request timed out.
Reply from 30.0.0.2: bytes=32 time=2ms TTL=125
Reply from 30.0.0.2: bytes=32 time=2ms TTL=125
Reply from 30.0.0.2: bytes=32 time=11ms 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 = 2ms, Maximum = 11ms, Average = 5ms

PC>ping 20.0.0.2

Pinging 20.0.0.2 with 32 bytes of data:

Request timed out.
Reply from 20.0.0.2: bytes=32 time=9ms TTL=126
Reply from 20.0.0.2: bytes=32 time=6ms TTL=126
Reply from 20.0.0.2: bytes=32 time=7ms TTL=126

Ping statistics for 20.0.0.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 6ms, Maximum = 9ms, Average = 7ms

PC>ping 30.0.0.3

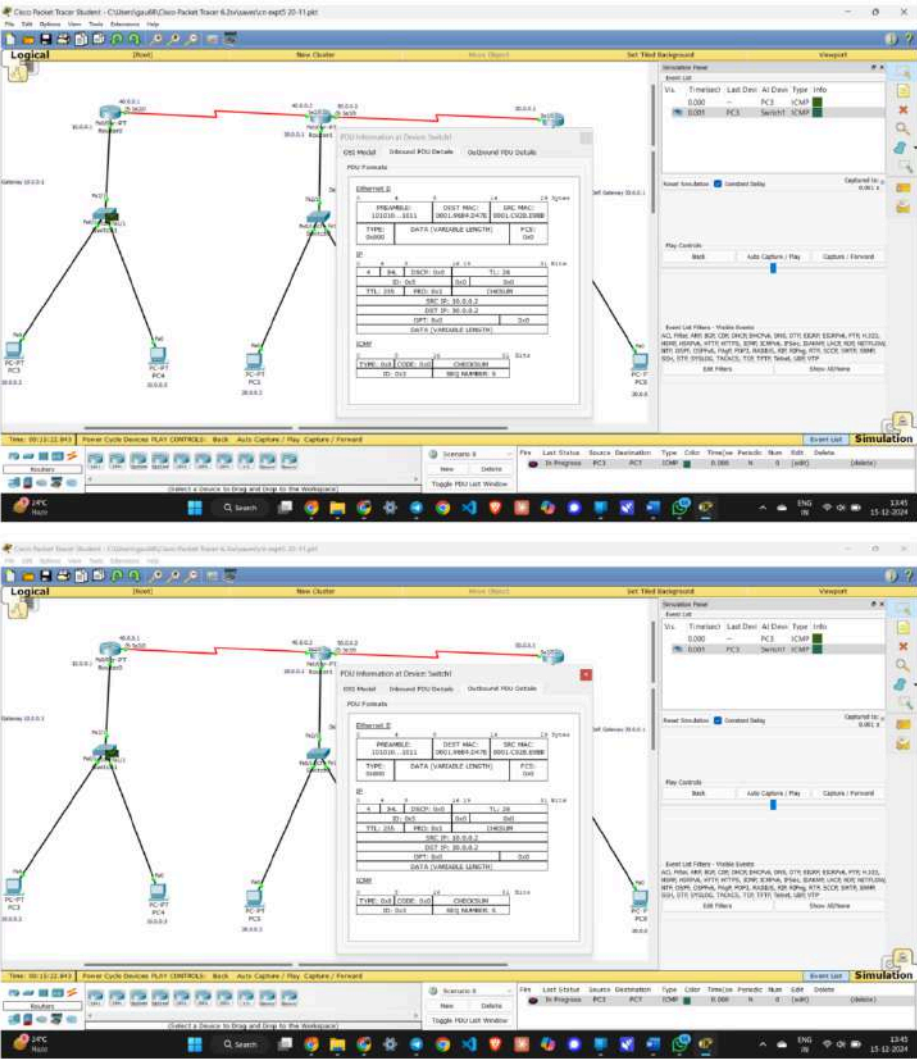
Pinging 30.0.0.3 with 32 bytes of data:

Request timed out.
Reply from 30.0.0.3: bytes=32 time=15ms TTL=125
Reply from 30.0.0.3: bytes=32 time=2ms TTL=125
Reply from 30.0.0.3: bytes=32 time=2ms TTL=125

Ping statistics for 30.0.0.3:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 15ms, Average = 5ms
```

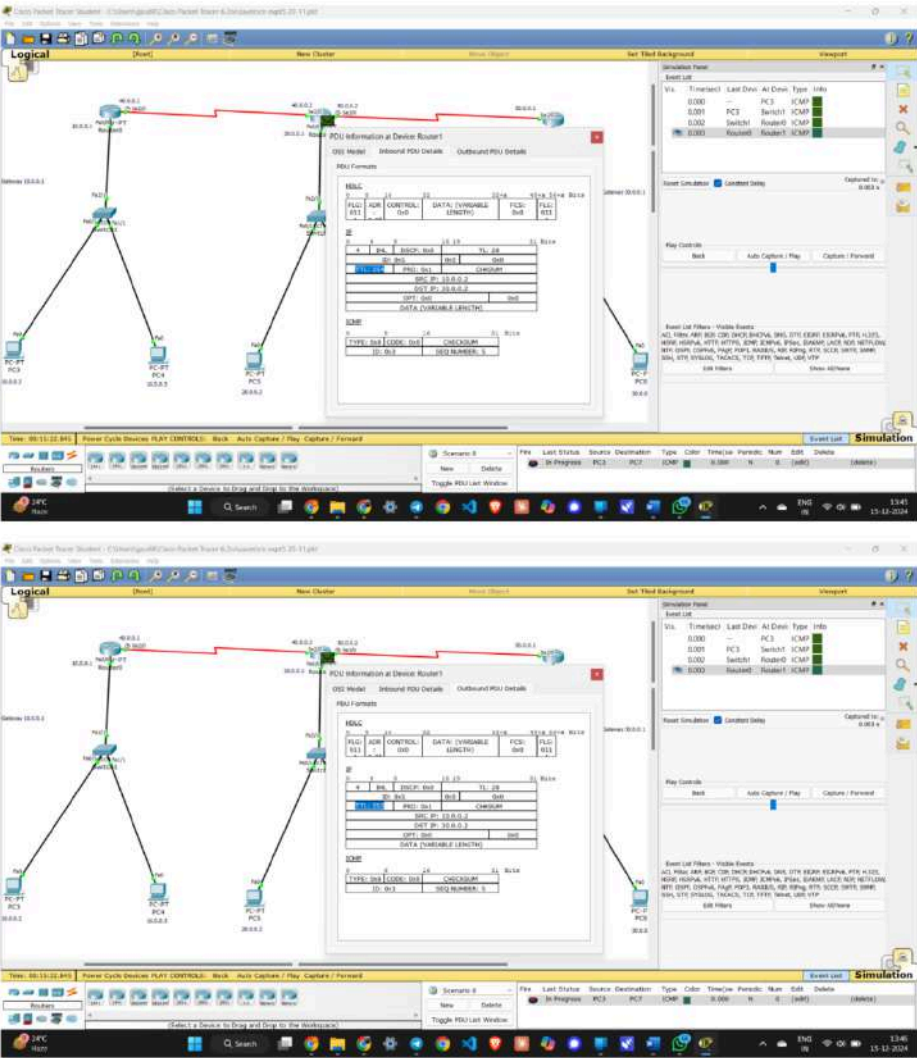
# Demonstrating TTL:

Switch1:

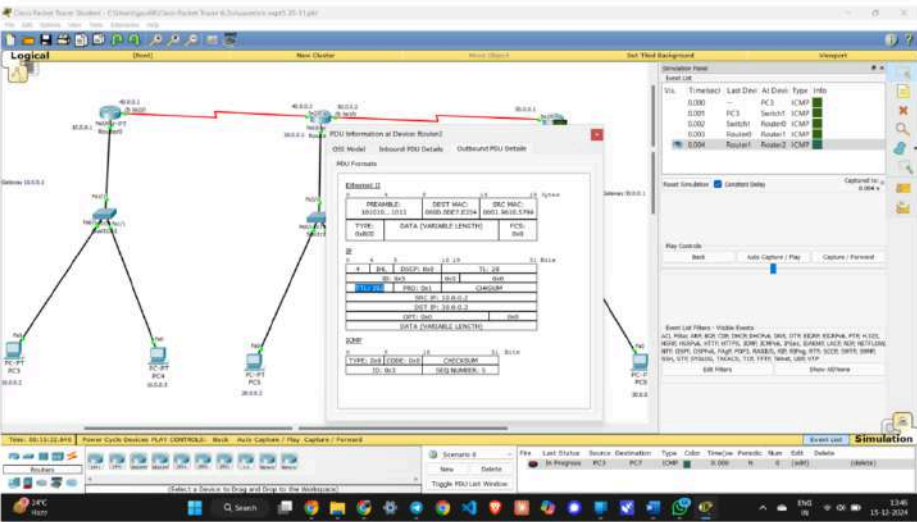
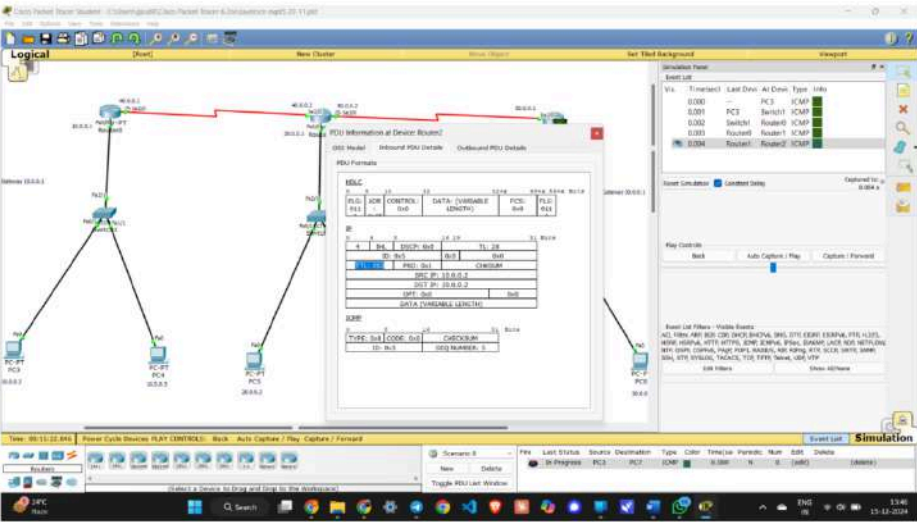


[illegible]

Router1:

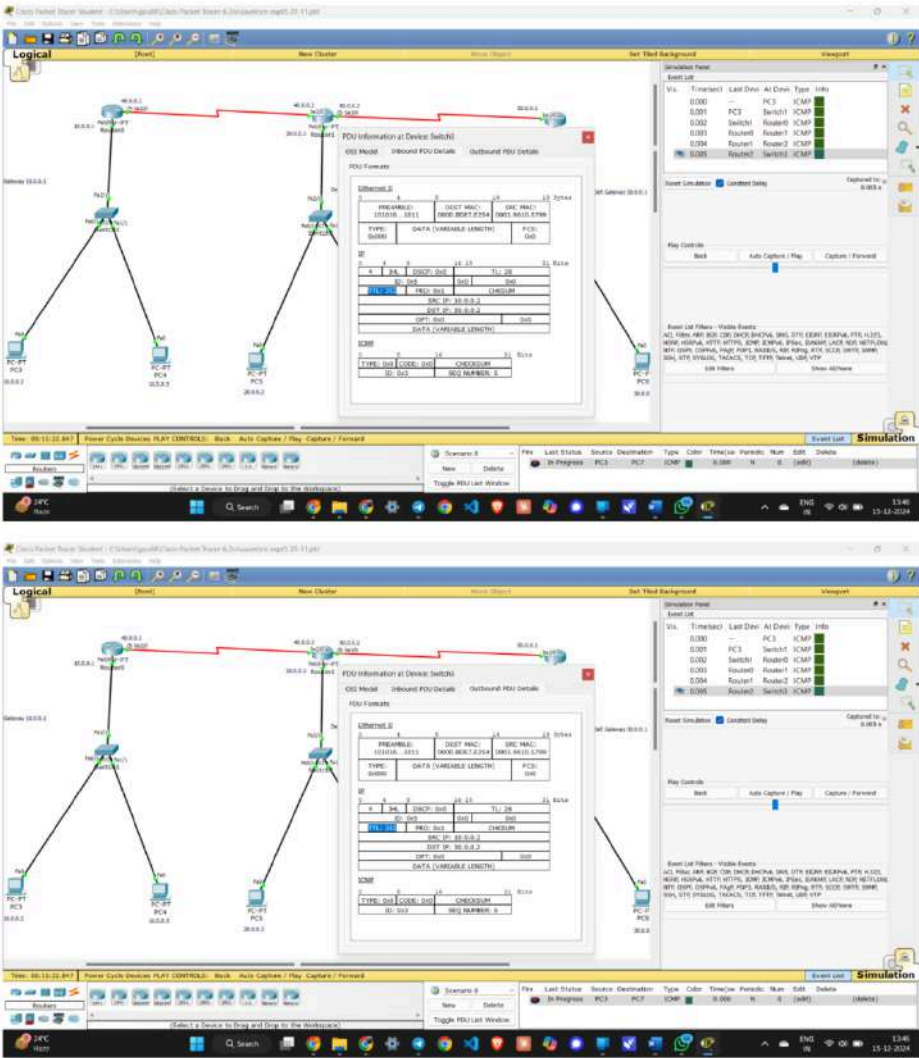


Router2:



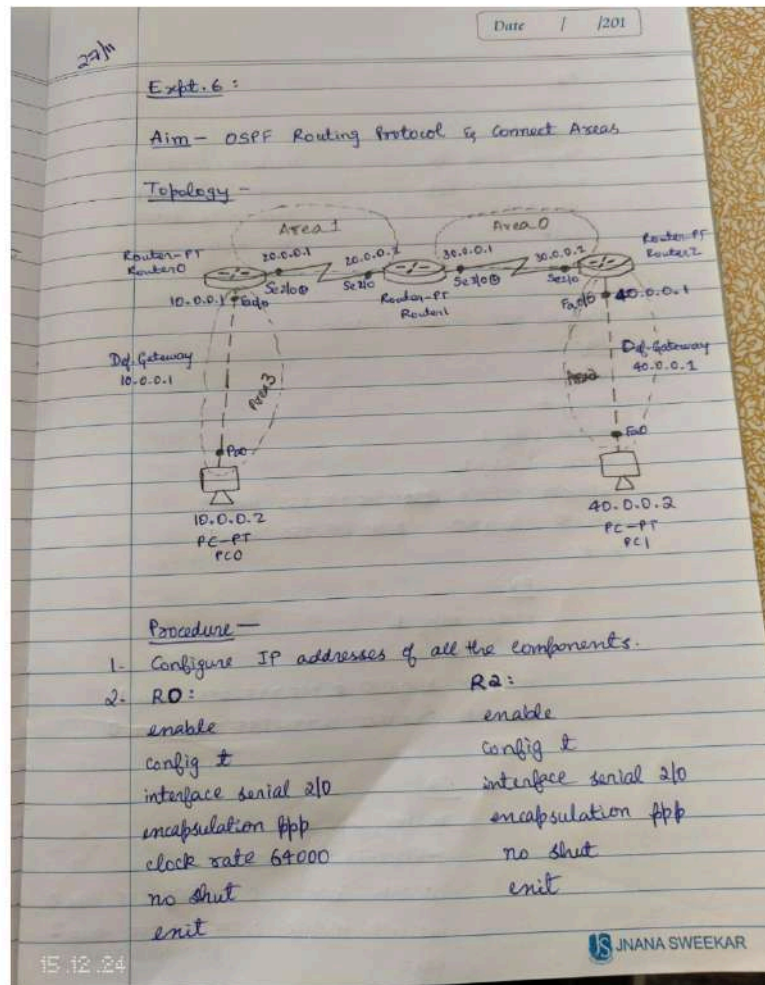


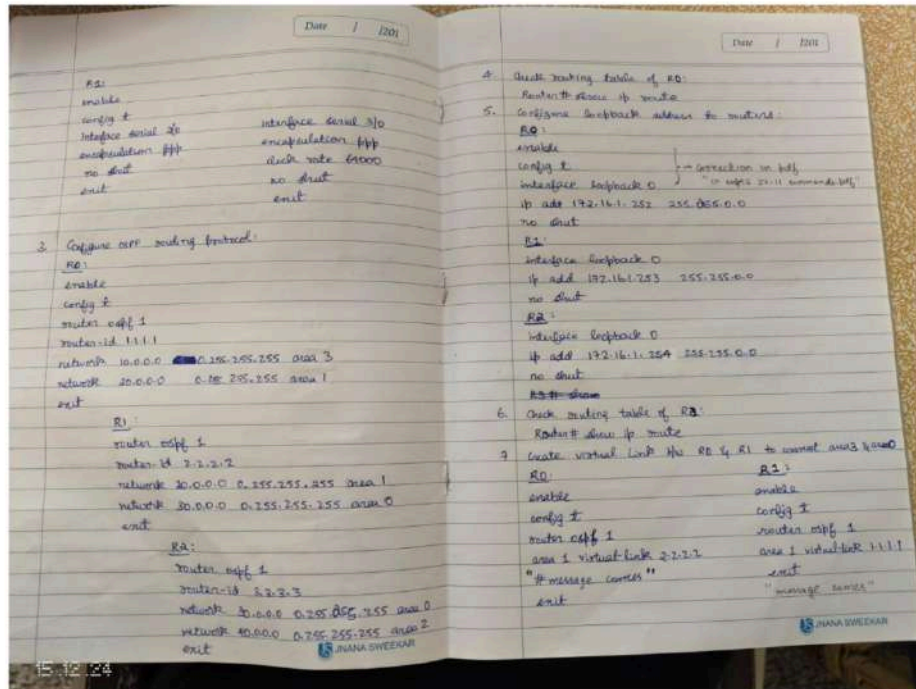
Switch3:



## Program 6

Configure OSPF routing protocol





- B. R1 & R2 get updates about area 3. check routing table of R2 now:  
 RA # shows the route  
 O IA 20.0.0.0/8 via 30.0.0.1, Serial 2/0  
 C 40.0.0.0/8 is directly connected, Fa 0/0  
 O IA 10.0.0.0/8 via 30.0.0.1, Serial 2/0  
 C 30.0.0.0/8 is directly connected, Serial 2/0

9. check connectivity b/w host 10.0.0.2 to 40.0.0.2

PC0 > ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

Reply from 40.0.0.2: bytes=32 time=2ms TTL=125

\_\_\_\_\_||\_\_\_\_\_

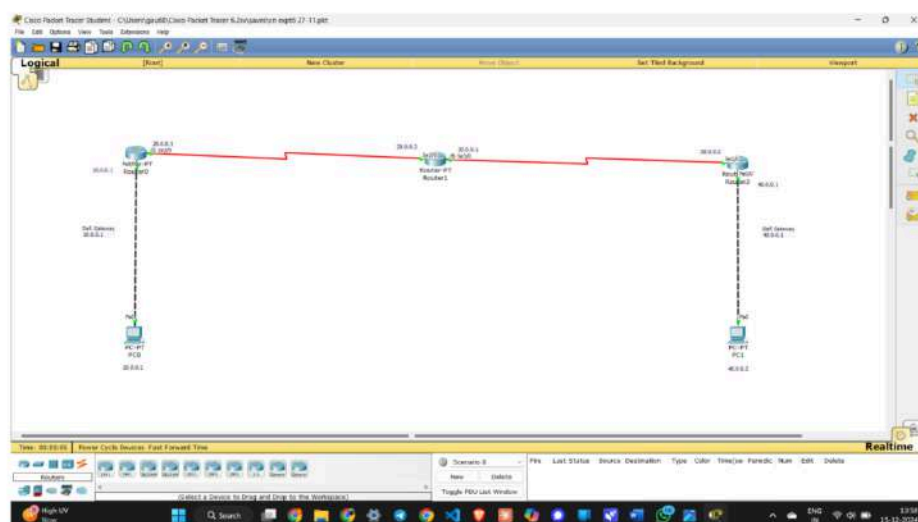
\_\_\_\_\_||\_\_\_\_\_

\_\_\_\_\_||\_\_\_\_\_

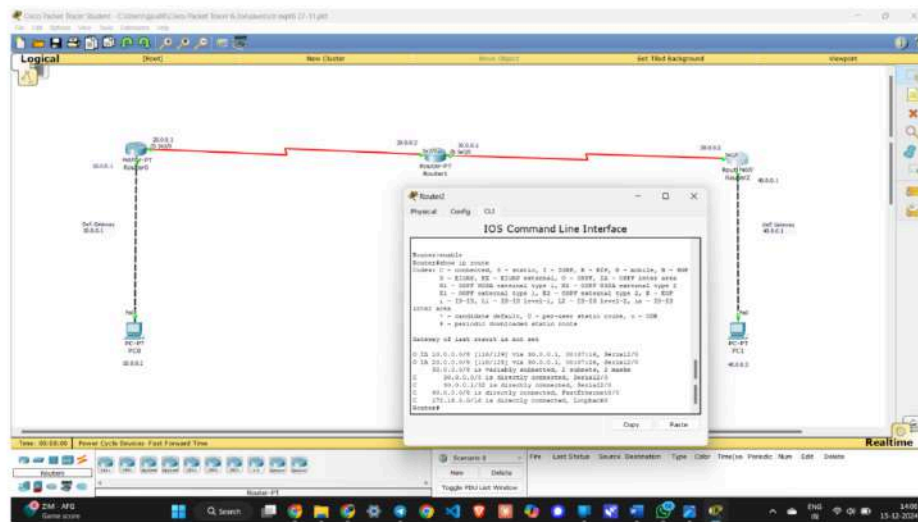
Ping statistics for 40.0.0.2:

Packets: Sent=4, Received=4, Loss=0 (0% loss)

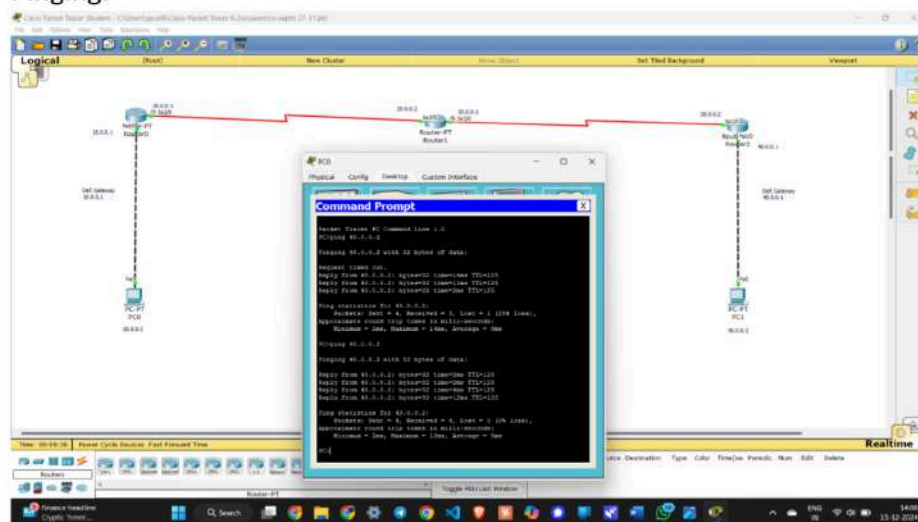
## Topology:







## Pinging:





## Program 8

Configure Web Server, DNS within a LAN.

18/12

Date / /201

Expt. 8 :

Aim : To ~~construct~~ configure Web Server, DNS within a LAN.

Topology :

```
graph TD; Switch[Switch] --- PC[PC 10.0.0.1]; Switch --- Server[Server 10.0.0.2];
```

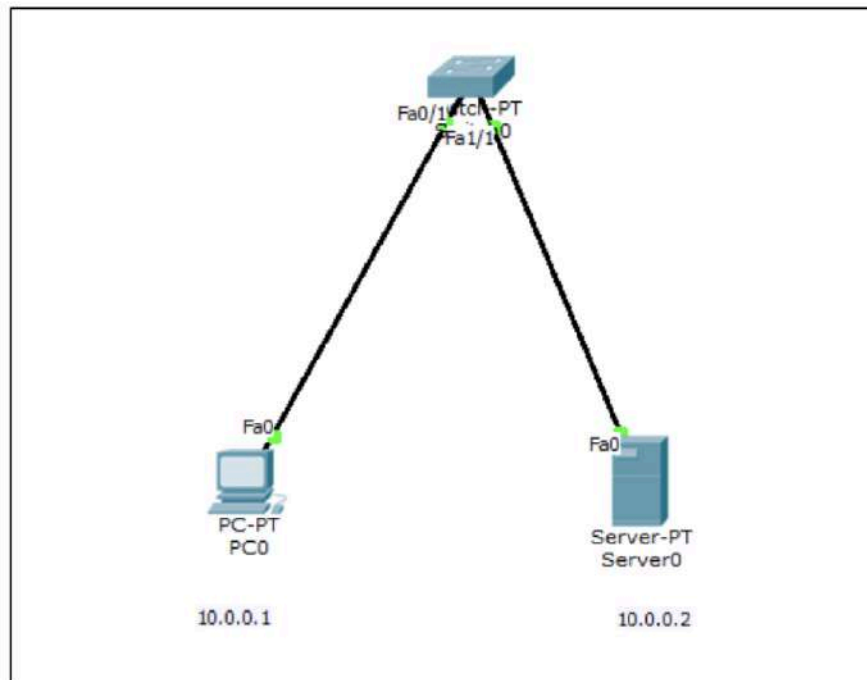
Procedure :

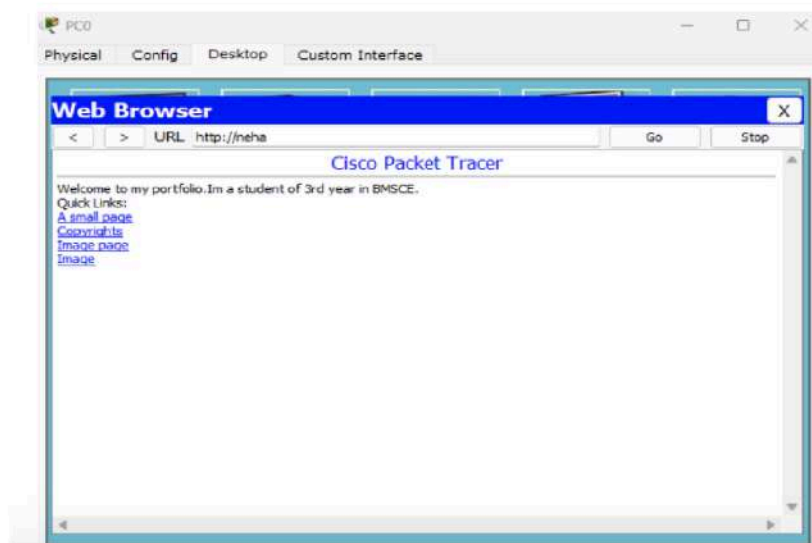
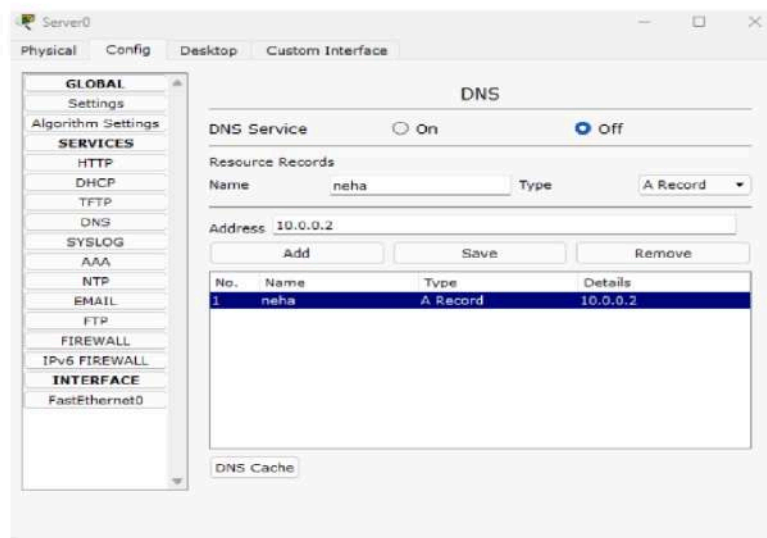
1. Set up the LAN as per the topology mentioned above & configure the devices.
2. Go to Server → Service → DNS :  
Name: bmsce (Domain name)  
Address: 10.0.0.2  
Add the mapping of domain name to address
3. Go to PC → Config → Global → Settings →  
DNS Server: 10.0.0.2  
[The server that provides the DNS mapping]
4. Go to PC → Desktop → Web Browser  
Type the URL: <http://bmsce>



Observation:

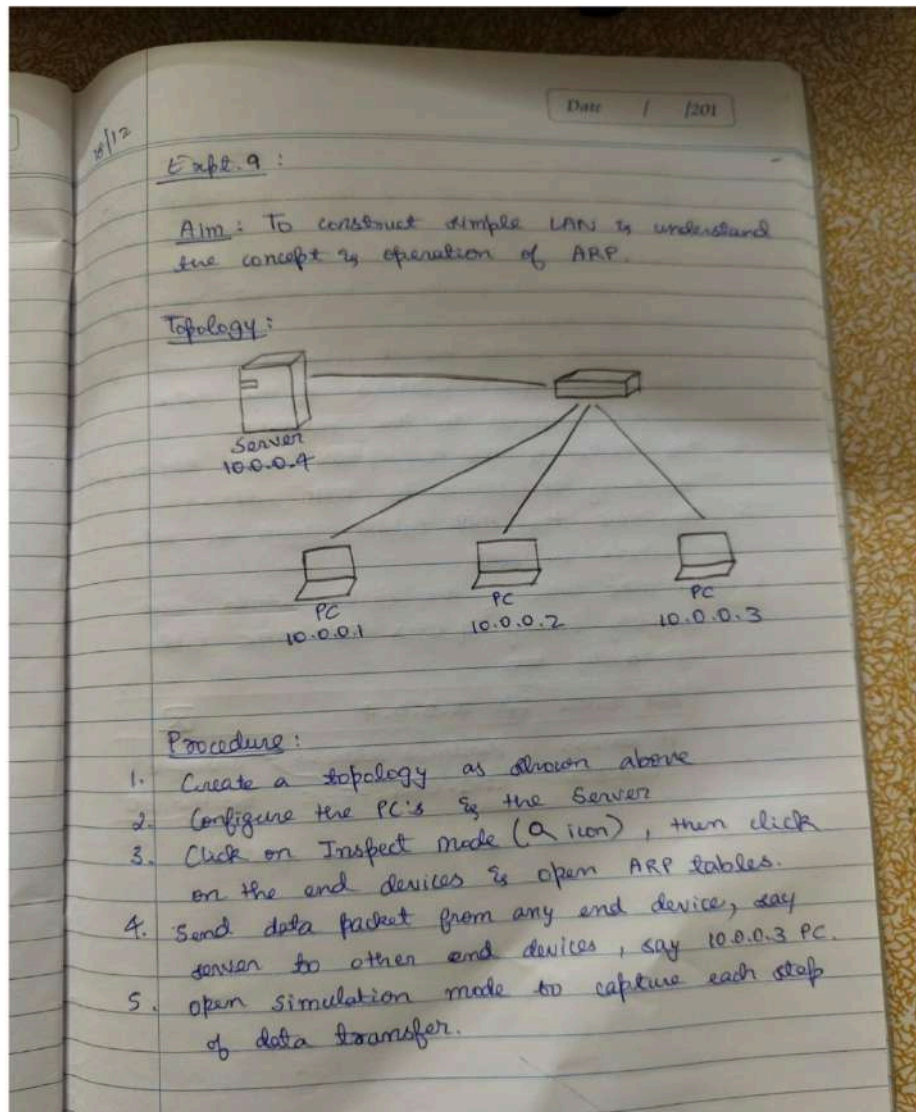
1. Webpage hosted by the server were visible on the browser.
2. The DNS was successful in mapping the domain name to the IP address.
3. DNS server is a server that contains a Domain name: IP address mapping to which the end devices send requests to map the Name to IP address.





## Program 9

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

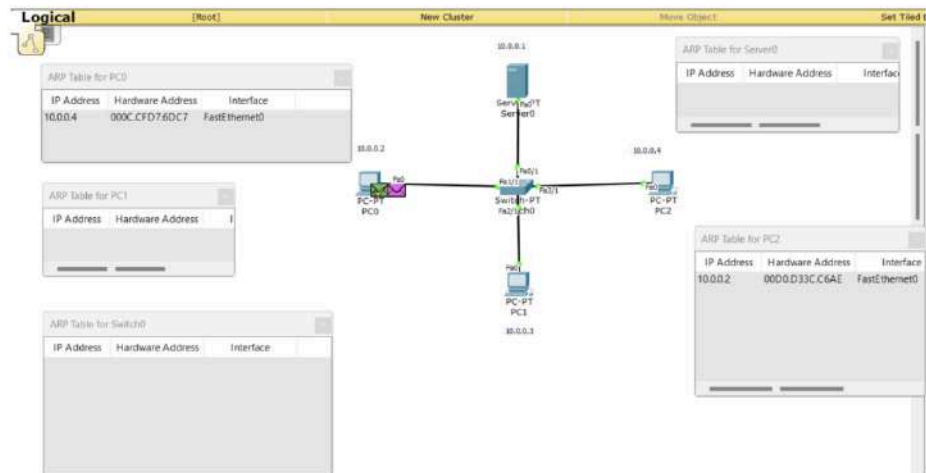


Observations:

1. The ARP tables of all end devices are initially empty.
2. When the data packet from Server arrives at the switch, since the Server 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 message.
4. The server & the PC update their ARP tables matching IP address to MAC address.
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 10.0.0.4			X
IP address	Hardware address	Interface	
10.0.0.3	0001-C726-47E5	FastEthernet0	

7. Similarly, other ARP tables are updated.



Simulation Panel

Event List

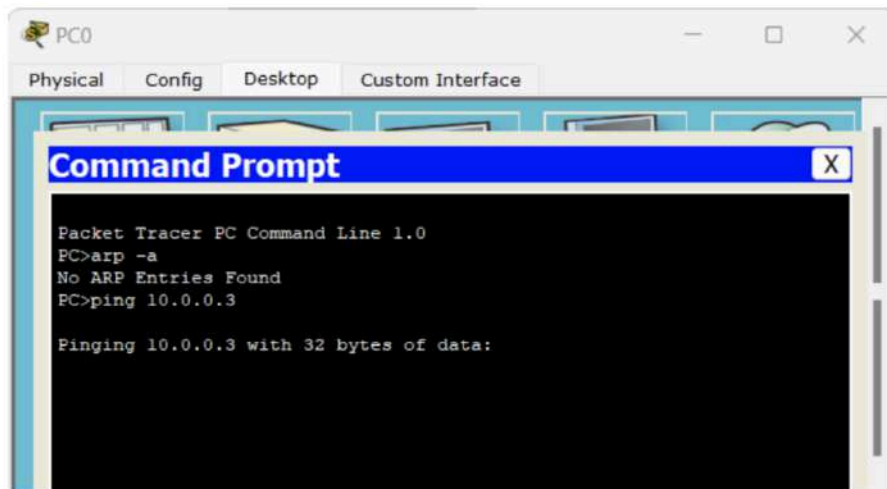
Vis.	Time(sec)	Last Devi	At Device	Type	Info
	0.001	PC0	Switch0	ARP	
	0.002	Switch0	Server0	ARP	
	0.002	Switch0	PC1	ARP	
	0.002	Switch0	PC2	ARP	
	0.003	PC2	Switch0	ARP	
	0.004	Switch0	PC0	ARP	
	0.004	--	PC0	ICMP	

Reset Simulation ☒ Constant Delay

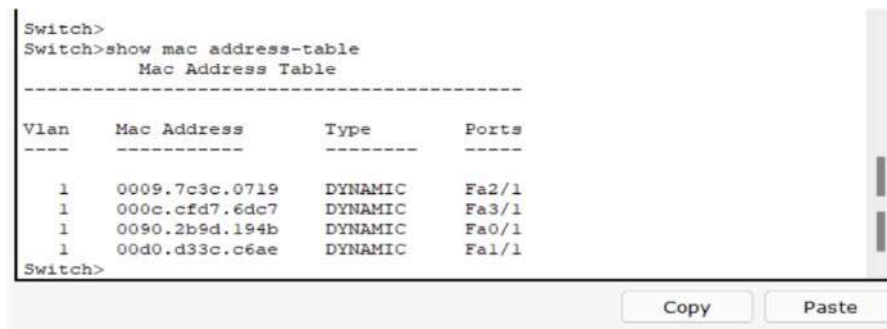
Captured to: \* 0.004 s

Play Controls

Back Auto Capture / Play Capture / Forward



**Switch:**





## Program 10

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

01

09/01/24

Date / / 201

Expt. 10:

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

Topology:

```
graph LR; PC[PC 10.0.0.1] -.- Router[Router 10.0.0.2]
```

Procedure:

1. Create a topology as given above & configure the devices.
2. Commands in Router:  
Router>enable  
Router# config terminal  
Router(config)# hostname R1  
Router(config)# enable secret 1234 → enable password  
Router(config)# interface fastethernet 0/0  
Router(config-if)# ip address 10.0.0.2 255.0.0.0  
Router(config-if)# no shut  
Router(config-if)# line vty 0 3  
Router(config-line)# login  
Router(config-line)# login disabled on line 194, until 'password' is set  
R1(config-line)# password 4321  
R1(config-line)# exit → user access verification password

09.01.25

ANANASWEEKAR



Date / / 201

D1 (config) # exit

D1 # nbr

Building configuration ...

[OK]

Note: why 0.0.0.0? First for virtual terminal  
then for telnet access

3] In PC: command Prompt

- First try ping to see if the devices are  
connected

PC > telnet 10.0.0.2

Trying 10.0.0.2... open

User Access Verification

Username: 4321

Password: 4321

D1 > enable

Password: 1234

D1 # show ip route

C 10.0.0.0/8 is directly connected, FastEthernet 0/0

R1 #

INANA SWEENAR

Date / / 201

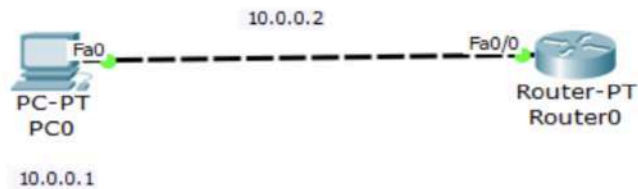
Observations:

1] The admin in PC is able to run commands on  
router, for router cli & see the results from  
PC

2] Telnet allows us to establish a remote  
session with another device like routers, even  
a Telnet emulator

3] Using Telnet, we can access and control other  
remote devices all as if you were physically  
connected to it

INANA SWEENAR



```
Router0
Physical Config CLI
IOS Command Line Interface
2 Low-speed serial(sync/async) network interface(s)
32K bytes of non-volatile configuration memory.
63488K bytes of ATA CompactFlash (Read/Write)

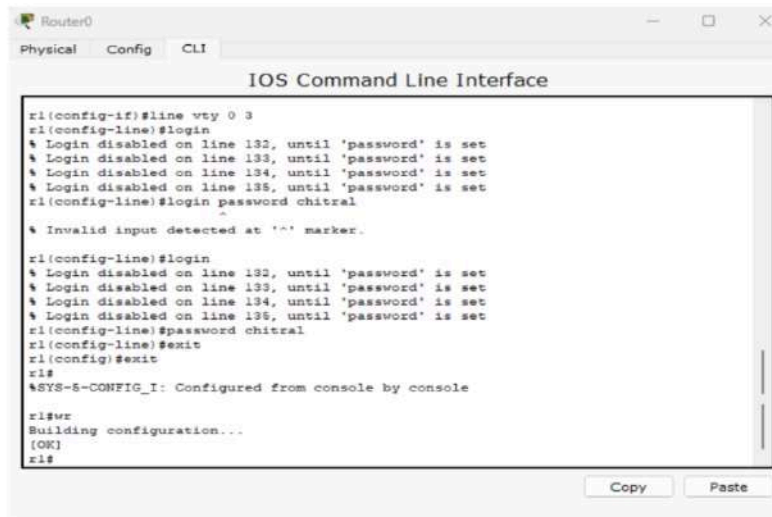
--- System Configuration Dialog ---
Continue with configuration dialog? [yes/no]: no

Press RETURN to get started!

Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname r1
r1(config)#enable secret neha
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)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
```

Copy Paste



Router0

Physical Config CLI

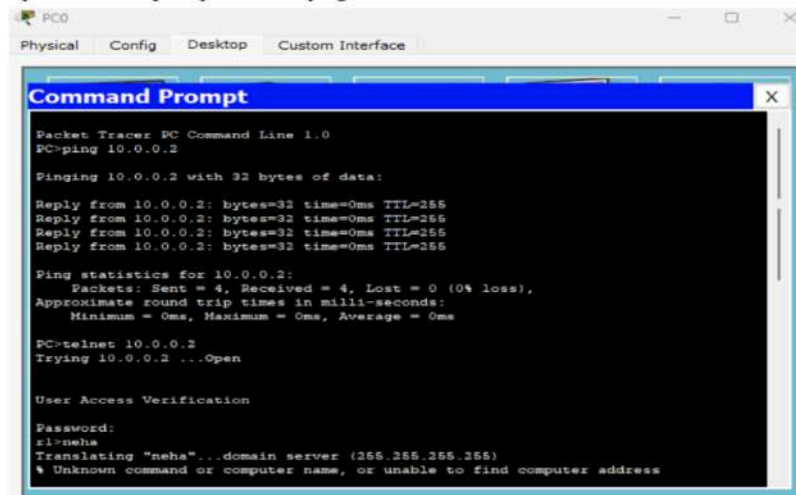
IOS Command Line Interface

```
rl(config-if)#line vty 0 3
rl(config-line)#login
% Login disabled on line 132, until 'password' is set
% Login disabled on line 133, until 'password' is set
% Login disabled on line 134, until 'password' is set
% Login disabled on line 135, until 'password' is set
rl(config-line)#login password chitral
rl(config-line)#login
% Invalid input detected at '^' marker.
rl(config-line)#login
% Login disabled on line 132, until 'password' is set
% Login disabled on line 133, until 'password' is set
% Login disabled on line 134, until 'password' is set
% Login disabled on line 135, until 'password' is set
rl(config-line)#password chitral
rl(config-line)#exit
rl(config)#exit
rl#
%SYS-5-CONFIG_I: Configured from console by console

rl#ur
Building configuration...
[OK]
rl#
```

Copy Paste

Open command prompt of PC0: ping 10.0.0.2



PC0

Physical Config Desktop Custom Interface

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
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
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:
rl>neha
Translating "neha"...domain server (255.255.255.255)
% Unknown command or computer name, or unable to find computer address
```

```
PC0
Physical Config Desktop Custom Interface

Command Prompt X
rl>
rl>exit

[Connection to 10.0.0.2 closed by foreign host]
PC>telnet 10.0.0.2
Trying 10.0.0.2 ...Open

User Access Verification

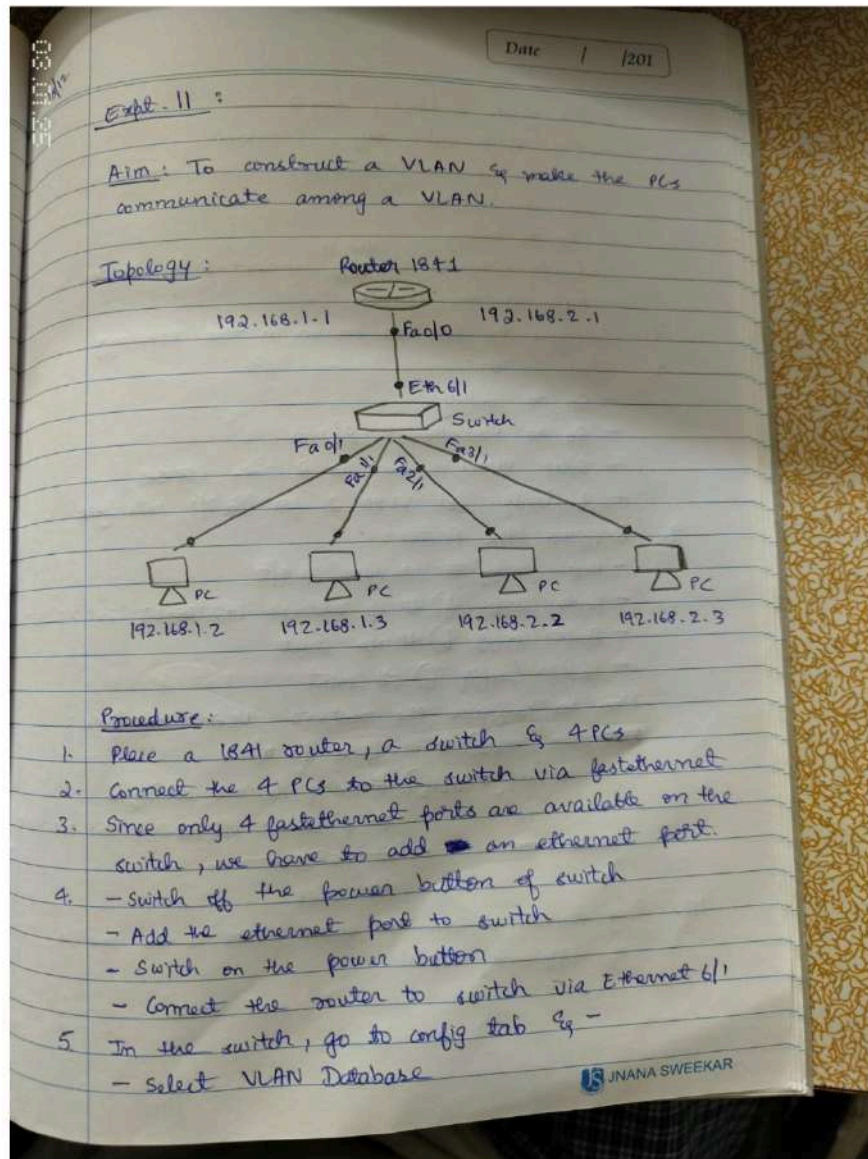
Password:
Password:
rl>enable
Password:
rl#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 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

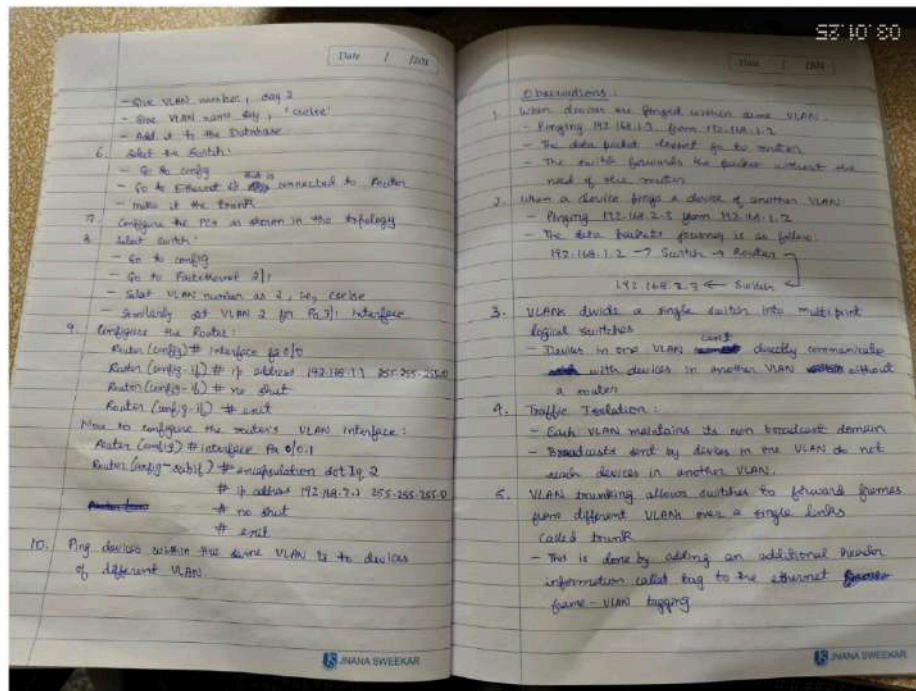
Gateway of last resort is not set

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

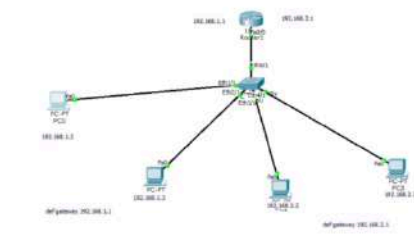
## Program 11

To construct a VLAN and make the PC's communicate among a VLAN









```

PC2
-----
Command Prompt

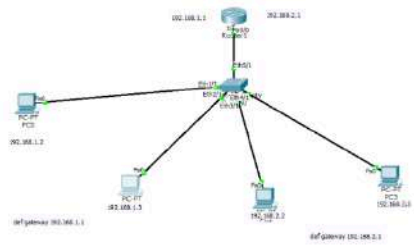
C:\>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.1.1:
    Packets: Sent = 5, Received = 0, Lost = 5 (100% loss),
    Approximate round trip times in milliseconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>

```



```

PC1
-----
Command Prompt

C:\>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.1.1:
    Packets: Sent = 5, Received = 0, Lost = 5 (100% loss),
    Approximate round trip times in milliseconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>

```

## Program 12

To construct a WLAN and make the nodes communicate wirelessly

10/12

Lab No. 12

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

Procedure:

1. Create the topology as given above & configure the devices.
2. Configure Access Point:
  - Access Point → Config → Port 1:
  - SSID: krose
  - Secret: @WEP
  - Set Key: 1234567890

3. Configure PC & Laptop with wireless interface:

- Switch on device
- Drag the existing FI-NET-100-200 to the configuration listed in the list of physical
- Drag wireless wireless interface to the empty slot
- 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 address & Gateway.

Topology after wireless configuration:

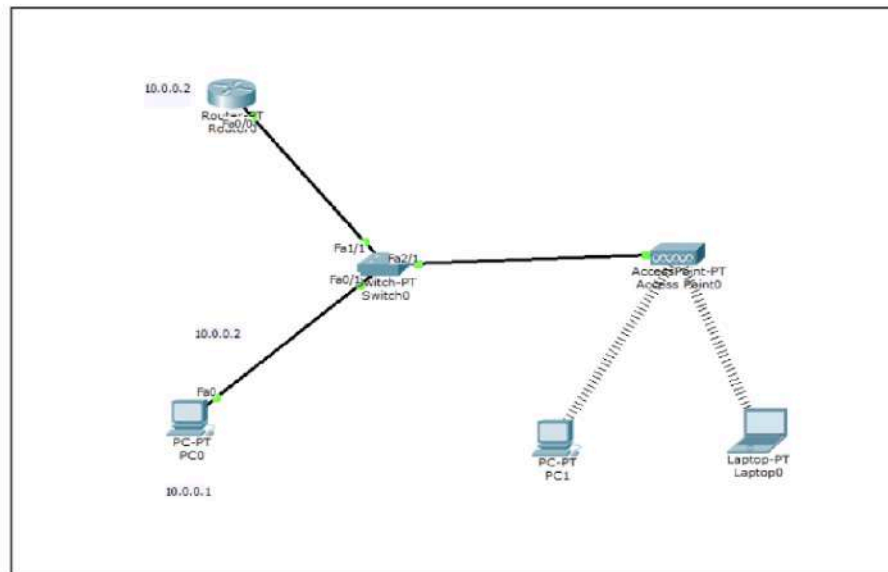
6. Ping from every device to every other device to check for connection.

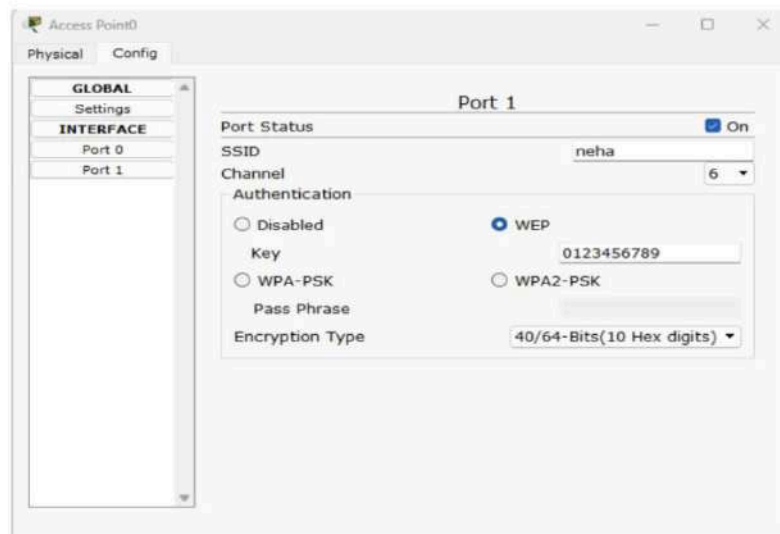
03.01.25

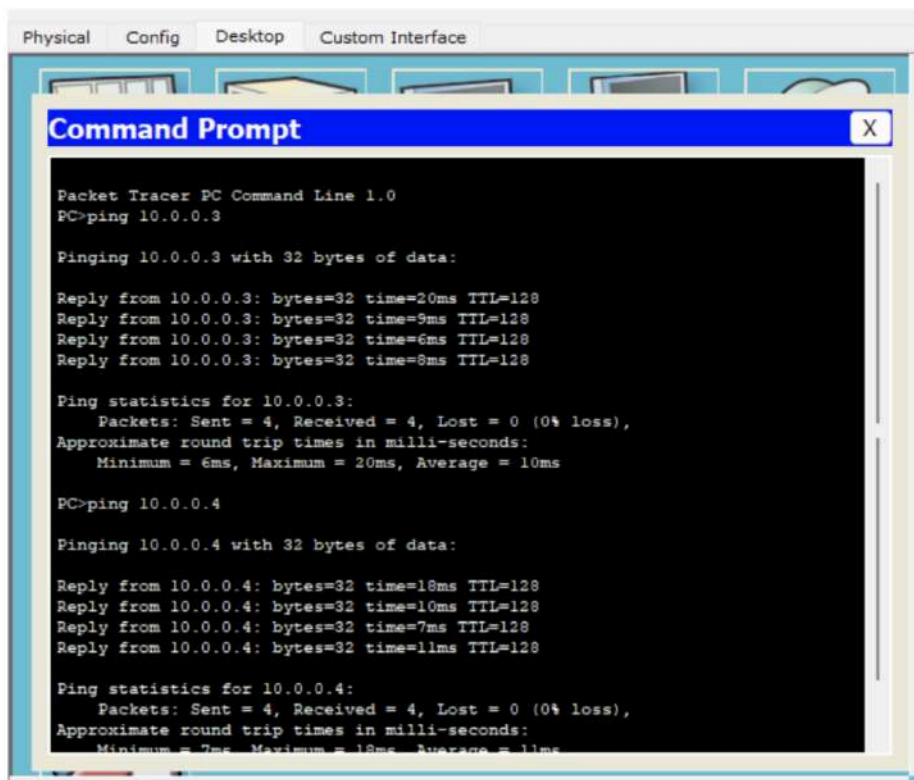
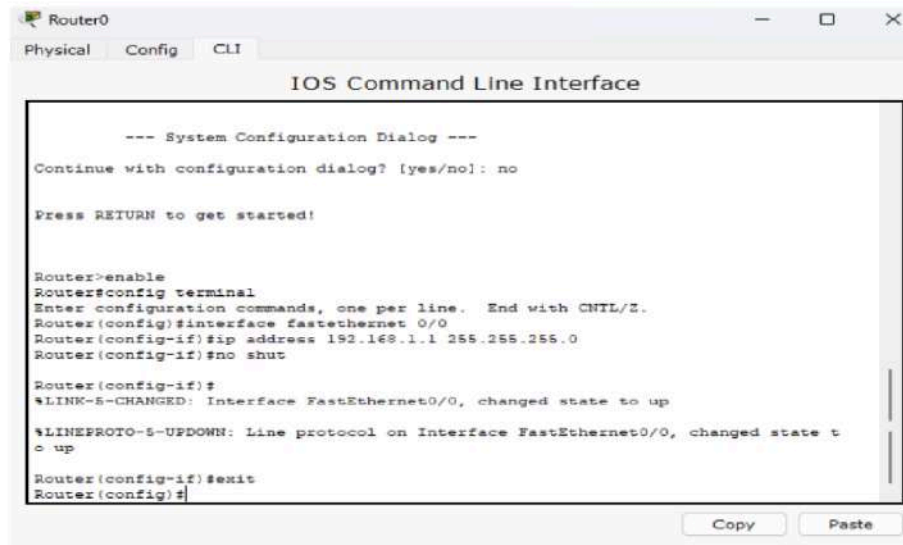


Observation:

1. We were able to ping from every device to every other device.
2. Access Point :  
Creates bridge b/w wired & wireless devices.  
- SSID Broadcasting : Announces the wireless network's name (SSID) to allow devices to connect using WEP, WPA or WPA2
3. WMP300N wireless interface :  
- Wireless network adapter that enables devices to communicate with access point using wireless signals.
4. Pinging: 10.0.0.1 to 10.0.0.3 :  
10.0.0.1 → Switch → Access Point → 10.0.0.3  
- This is after the ARP tables are updated after broadcasting.
5. Pinging: 10.0.0.3 to 10.0.0.1 :  
10.0.0.3 → Access Point → Switch → 10.0.0.1
6. Pinging: 10.0.0.3 to 10.0.0.4 :  
10.0.0.3 → Access Point → 10.0.0.4
7. Every device is now connected to every other device in the WLAN







### Program 13

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

```

CYCLE-2
Date / / 2001

Expt. 13:
Aim: Error detecting code using CRC-CCITT (16-bits)

def mod(a,b):
    auint = []
    for i in range(1, len(b)):
        if a[i] > b[i]:
            auint.append('0')
        else:
            auint.append('1')
    return ''.join(auint)

def moddiv(dividend, divisor):
    pdiv = bin(divisor)
    tmp = dividend[0:1+pdiv]
    while pdiv < len(dividend):
        if tmp[-1] == '1':
            tmp = not(divisor, tmp) +
                dividend[pdiv:]
        else:
            tmp = not('0' * pdiv, tmp) +
                dividend[pdiv:]
        pdiv += 1
    if tmp[-1] == '1':
        tmp = not(divisor, tmp)
    return dividend

def encode(data, key):
    Lkey = bin(key)
    appended_data = data + '0' * (16 - len(Lkey))
    remainder = moddiv(appended_data, key)
    codeword = data + remainder
    print('Encoded Data (Data + remainder):',
          codeword)

def decode(data, encoded_data, key):
    remainder = moddiv(encoded_data, key)
    print('Remainder after decoding:', remainder)
    if '1' not in remainder:
        print('No error detected in received data')
    else:
        print('Error detected in received data')
    data = '100100100100'
    key = '1101'
    encoded_data = encode(data, key)
    decoded_data = decode(data, encoded_data, key)

Output:
Remainder = 1)
encoded_data (Data + remainder) = 1001001001001
Remainder after decoding = 000
No error detected in received data.

```

Code:

```
#include <stdio.h>
#include <string.h>
#define N strlen(gen_poly)
char data[28], gen_poly[10], check[28];
int data_len, i, j;
void XOR() {
    for (j = 0; j < N; j++) {
        check[j] = (check[j] == gen_poly[j]) ? '0' : '1';
    }
}
void crc() {
    for (i = 0; i < N; i++) {
        check[i] = data[i];
    }
    do {
        if (check[0] == '1') {
            XOR();
        }
        for (j = 0; j < N - 1; j++) {
            check[j] = check[j + 1];
        }
        check[j] = data[i++];
    } while (i <= data_len + N - 1);
}
void receiver() {
    printf("\nData received: ");
    scanf("%s", data);
    crc();
    for (i = 0; i < N - 1; i++) {
        if (check[i] == '1') {
            break;
        }
    }
    if (i < N - 1) {
        printf("\nERROR!");
    } else {
        printf("\nNO ERROR!");
    }
}
5
1
int main() {
    printf("\nEnter data: ");
    scanf("%s", data);
    printf("\nEnter generator: ");
    scanf("%s", gen_poly);
    data_len = strlen(data);
    // Append N-1 zeros to the data
    for (i = data_len; i < data_len + N - 1; i++) {
        data[i] = '0';
    }
    data[data_len + N - 1] = '\0'; // Null-terminate the string
    printf("\nData with padded 0's: %s", data);
```

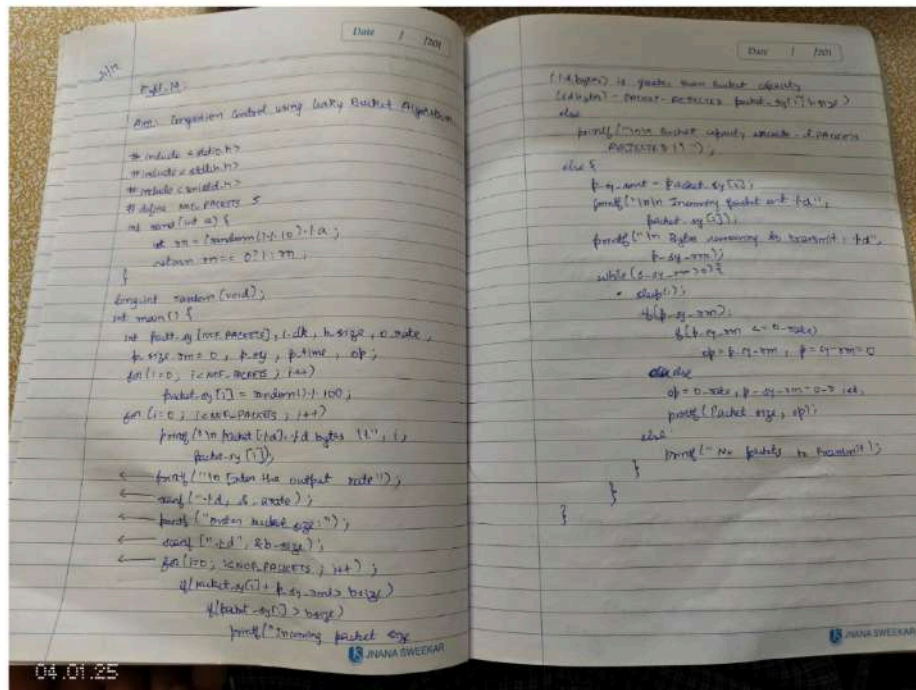
```

crc();
printf("\nCheck sum: ");
for (i = 0; i < N - 1; i++) {
printf("%c", check[i]);
}
// Append checksum to data
for (i = data_len; i < data_len + N - 1; i++) {
data[i] = check[i - data_len];
}
data[data_len + N - 1] = '\0'; // Null-terminate the string
printf("\nFinal data to be transmitted: %s", data);
receiver();
return 0;
}

```

### Program 14

Write a program for congestion control using Leaky bucket algorithm.





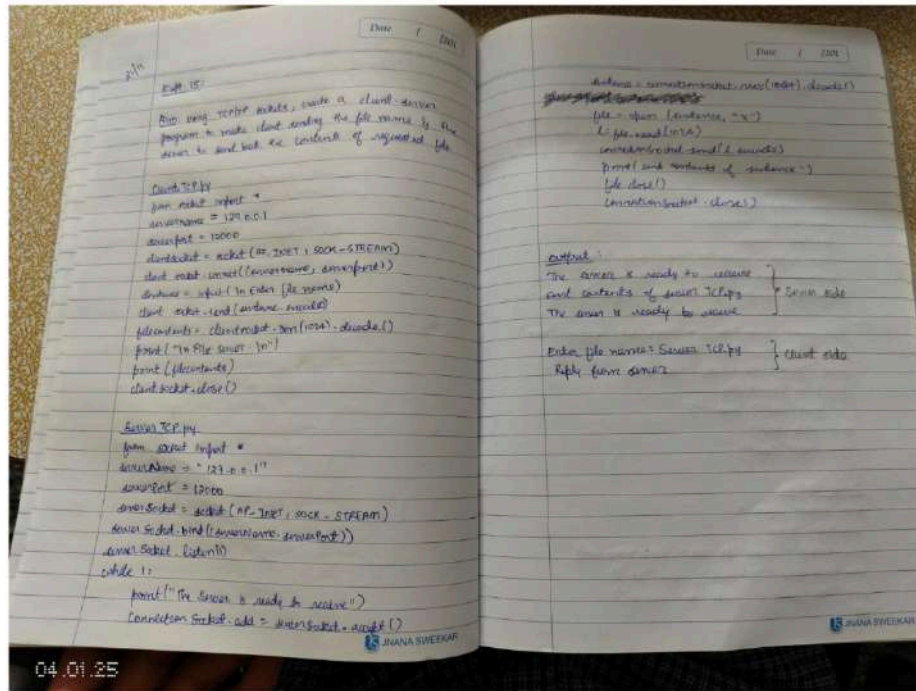
Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h> // for sleep function
#define NOF_PACKETS 5
// Function to simulate sending packets
void send_packet(int packet_size, int output_rate) {
    while (packet_size > 0) {
        int sent = (packet_size < output_rate) ? packet_size : output_rate;
        printf("Packet of size %d Transmitted---", sent);
        packet_size -= sent;
        printf("Bytes Remaining to Transmit: %d\n", packet_size);
        sleep(1); // Simulate time delay between packets
    }
}

int main() {
    int output_rate, bucket_size, incoming_packet_size;
    int i, packet_size[NOF_PACKETS];
    // Input number of packets and their sizes
    for(i = 0; i < NOF_PACKETS; i++) {
        packet_size[i] = rand() % 100; // Random packet size between 0 and 99
        printf("packet[%d]:%d bytes\n", i, packet_size[i]);
    }
    printf("Enter the Output rate:");
    scanf("%d", &output_rate);
    printf("Enter the Bucket Size:");
    scanf("%d", &bucket_size);
    for(i = 0; i < NOF_PACKETS; i++) {
        printf("\nIncoming Packet size: %d\n", packet_size[i]);
        if(packet_size[i] > bucket_size) {
            printf("Incoming packet size (%dbytes) is Greater than bucket capacity\n", packet_size[i], bucket_size);
            printf("55\n");
            continue;
        }
        printf("Bytes remaining to Transmit: %d\n", packet_size[i]);
        send_packet(packet_size[i], output_rate);
    }
    return 0;
}
```

## Program 15

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.



Code:

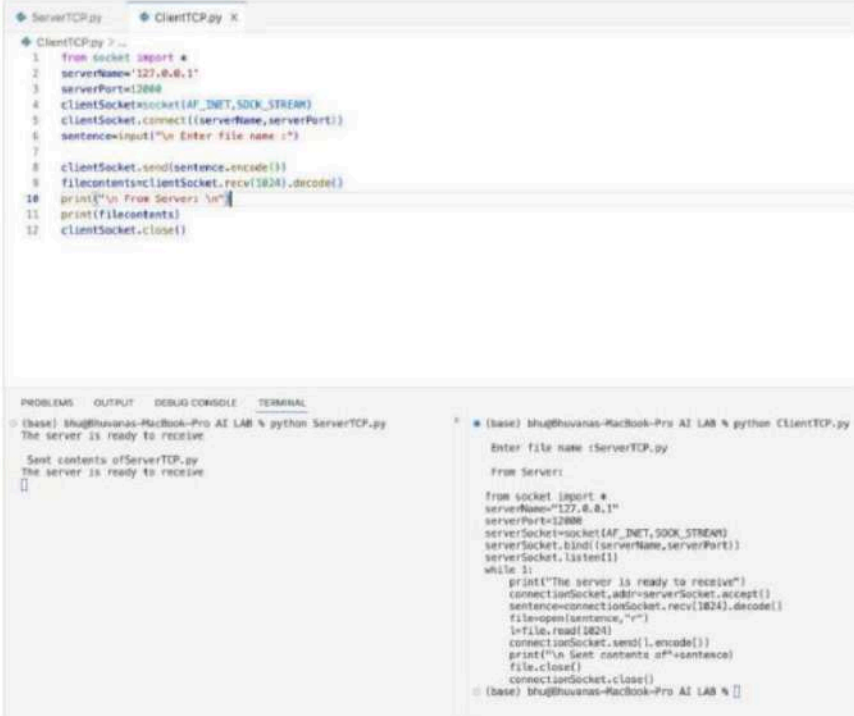
ServerTCP.py

```
from socket import *
serverName="127.0.0.1"
serverPort=12000
serverSocket=socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print("The server is ready to receive")
    connectionSocket,addr=serverSocket.accept()
    sentence=connectionSocket.recv(1024).decode()
    file=open(sentence,"r")
    l=file.read(1024)
    connectionSocket.send(l.encode())
    print("\n Sent contents of "+sentence)
    file.close()
    connectionSocket.close()
```

ClientTCP.py

```
from socket import *
serverName='127.0.0.1'
serverPort=12000
clientSocket=socket(AF_INET,SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence=input("\n Enter file name :")
clientSocket.send(sentence.encode())
filecontents=clientSocket.recv(1024).decode()
print("\n From Server: \n")
print(filecontents)
clientSocket.close()
```

## Output:



The screenshot shows an IDE with two tabs: `ServerTCP.py` and `ClientTCP.py`. The `ClientTCP.py` tab is active, displaying the following code:

```
1 from socket import *
2 serverName='127.0.0.1'
3 serverPort=12888
4 clientSocket=socket(AF_INET,SOCK_STREAM)
5 clientSocket.connect((serverName,serverPort))
6 sentence=input("Enter file name :")
7
8 clientSocket.send(sentence.encode())
9 fileContents=clientSocket.recv(1024).decode()
10 print("From Servers :")
11 print(fileContents)
12 clientSocket.close()
```

Below the code editor, the `TERMINAL` tab is active, showing the execution of both programs. On the left, the output of `ServerTCP.py` is shown:

```
(base) bhugbhuvan@MacBook-Pro AI LAB % python ServerTCP.py
The server is ready to receive
Sent contents ofServerTCP.py
The server is ready to receive
[]
```

On the right, the output of `ClientTCP.py` is shown:

```
(base) bhugbhuvan@MacBook-Pro AI LAB % python ClientTCP.py
Enter file name :ServerTCP.py
From Servers:
from socket import *
serverName="127.0.0.1"
serverPort=12888
serverSocket=socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print("The server is ready to receive")
    connectionSocket,addr=serverSocket.accept()
    sentence=connectionSocket.recv(1024).decode()
    file=open(sentence,"r")
    l=file.read(1024)
    connectionSocket.send(l.encode())
    print("\n Sent contents of"+sentence)
    file.close()
    connectionSocket.close()
(base) bhugbhuvan@MacBook-Pro AI LAB %
```

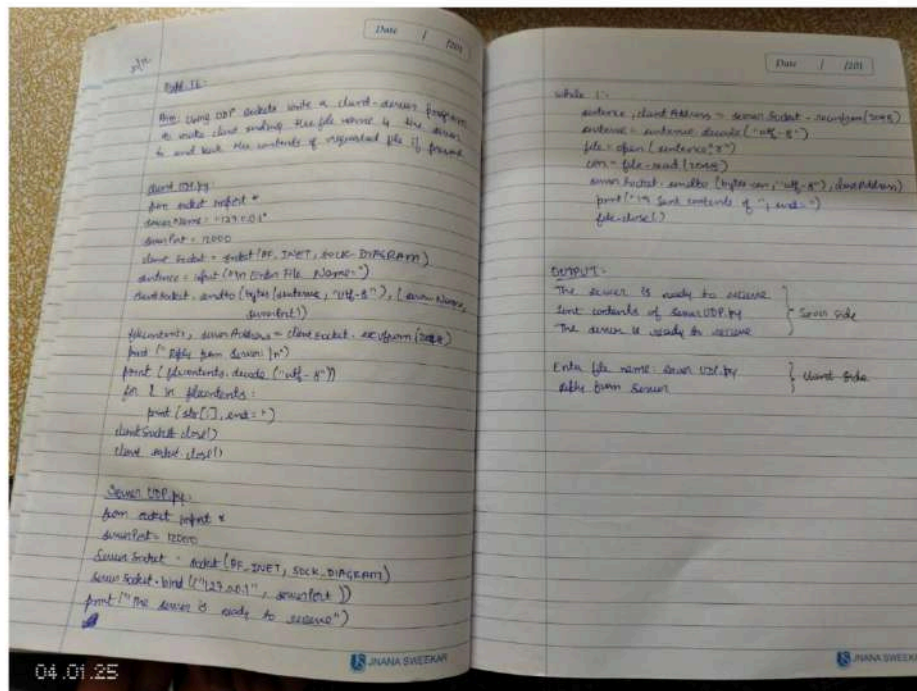
```
ServerTCP.py X ClientTCP.py
ServerTCP.py > ...
1 from socket import *
2 serverName="127.0.0.1"
3
4 (variable) serverSocket: socket (read)
5 serverSocket.bind((serverName,serverPort))
6 serverSocket.listen(1)
7 while 1:
8     print("The server is ready to receive")
9     connectionSocket,addr=serverSocket.accept()
10    sentence=connectionSocket.recv(1024).decode()
11    file=open(sentence,"r")
12    l=file.read(1024)
13    connectionSocket.send(l.encode())
14    print("\n Sent contents of"+sentence)
15    file.close()
16    connectionSocket.close()

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
(base) bhugbhuvas-MacBook-Pro AI LAB % python ServerTCP.py
The server is ready to receive
Sent contents ofServerTCP.py
The server is ready to receive
[]

(base) bhugbhuvas-MacBook-Pro AI LAB % python ClientTCP.py
Enter file name :ServerTCP.py
From Server:
from socket import *
serverName="127.0.0.1"
serverPort=12000
serverSocket=socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print("The server is ready to receive")
    connectionSocket,addr=serverSocket.accept()
    sentence=connectionSocket.recv(1024).decode()
    file=open(sentence,"r")
    l=file.read(1024)
    connectionSocket.send(l.encode())
    print("\n Sent contents of"+sentence)
    file.close()
    connectionSocket.close()
(base) bhugbhuvas-MacBook-Pro AI LAB %
```

## Program 16

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:

ServerUDP.py

```
from socket import *
serverPort=12000
serverSocket=socket(AF_INET,SOCK_DGRAM)
serverSocket.bind(("127.0.0.1",serverPort))
while 1:
    print("The server is ready to receive")
    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("\n Sent contents of "+sentence)
    file.close()
```

ClientUDP.py

```
from socket import *
serverName="127.0.0.1"
serverPort=12000
clientSocket=socket(AF_INET,SOCK_DGRAM)
sentence=input("\n Enter File Name:")
clientSocket.sendto(bytes(sentence,"utf-8"),(serverName,serverPort))
filecontents,serverAddress=clientSocket.recvfrom(2048)
print("\n Reply from server: \n")
print(filecontents.decode("utf-8"))
clientSocket.close()
```

## Output:

The image displays a code editor with two files: `ClientUDP.py` and `ServerUDP.py`. The `ServerUDP.py` file contains the following code:

```
1 serverPort=12000
2 serverSocket=socket(AF_INET,SOCK_DGRAM)
3 serverSocket.bind(("127.0.0.1",serverPort))
4
5 while 1:
6     print("The server is ready to receive")
7     sentence,clientAddress=serverSocket.recvfrom(2048)
8     sentence=sentence.decode("utf-8")
9     file=open(sentence,"r")
10    con=file.read(2048)
11    serverSocket.sendto(bytes(con,"utf-8"),clientAddress)
12    print("\n Sent contents of "+sentence)
13    file.close()
```

The `ClientUDP.py` file contains the following code:

```
1 from socket import *
2 serverName="127.0.0.1"
3 serverPort=12000
4 clientSocket=socket(AF_INET,SOCK_DGRAM)
5
6 sentence=input("\n Enter File Name:")
7
8 clientSocket.sendto(bytes(sentence,"utf-8"),(serverName,serverPort))
9
10 filecontents,serverAddress=clientSocket.recvfrom(2048)
11 print("\n Reply from server: \n")
12 print(filecontents.decode("utf-8"))
13 clientSocket.close()
```

The terminal window shows the execution of both programs. The server program is run first, and it prints "The server is ready to receive". The client program is then run, and it prompts the user to enter a file name. The user enters "ServerUDP.py", and the client sends the contents of that file to the server. The server then prints the received contents.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

(base) bhu@Bhuvanas-MacBook-Pro AI LAB % python ServerUDP.py

The server is ready to receive

Sent contents of ServerUDP.py

The server is ready to receive

(base) bhu@Bhuvanas-MacBook-Pro AI LAB % python ClientUDP.py

Enter File Name:ServerUDP.py

Reply from server:

from socket import \*

serverPort=12000

serverSocket=socket(AF\_INET,SOCK\_DGRAM)

serverSocket.bind(("127.0.0.1",serverPort))

while 1:

print("The server is ready to receive")

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("\n Sent contents of "+sentence)

file.close()

(base) bhu@Bhuvanas-MacBook-Pro AI LAB %

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

(base) bhu@Bhuvanas-MacBook-Pro AI LAB % python ServerUDP.py

The server is ready to receive

Sent contents of ServerUDP.py

The server is ready to receive

(base) bhu@Bhuvanas-MacBook-Pro AI LAB % python ClientUDP.py

Enter File Name:ServerUDP.py

Reply from server:

from socket import \*

serverPort=12000

serverSocket=socket(AF\_INET,SOCK\_DGRAM)

serverSocket.bind(("127.0.0.1",serverPort))

while 1:

print("The server is ready to receive")

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("\n Sent contents of "+sentence)

file.close()

(base) bhu@Bhuvanas-MacBook-Pro AI LAB %



## Program 17

### Tool Exploration - Wireshark

