VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB RECORD

Computer Network Lab (23CS5PCCON)

Submitted by

Harsha B(1BM23CS107)

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 Academic Year 2024-25 (odd)

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 Network (23CS5PCCON)" carried out by **Harsha B (1BM23CS107),** who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements of the above-mentioned subject and the work prescribed for the said degree.

Rashmi H Assistant Professor Department of CSE, BMSCE	Dr. Selva Kumar Professor & HOD Department of CSE, BMSCE

Index

14	29/10/25	Error detecting code using CRC-CCITT	48-51
15	03/11/25	TCP File Request–Response Using Client–Server Socket Program	52-53
16	03/11/25	UDP File Request–Response Using Client–Server Socket Program	54-55

Github Link:
https://github.com/Harsha6B/CN-Lab

CYCLE 1:

Program 1

Aim of the program:

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

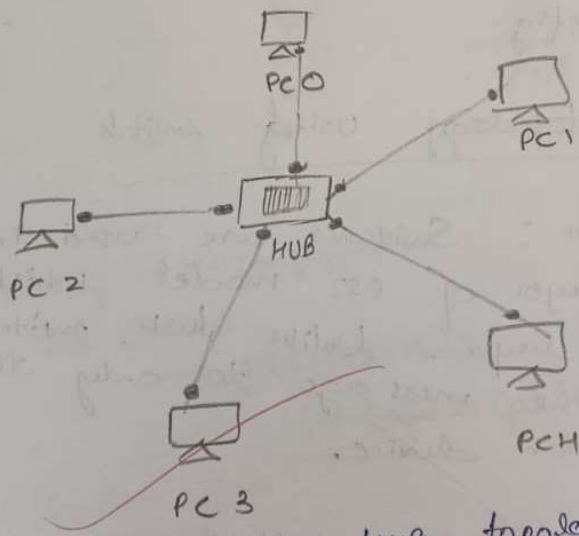
Procedure and topology:

20/08/2025

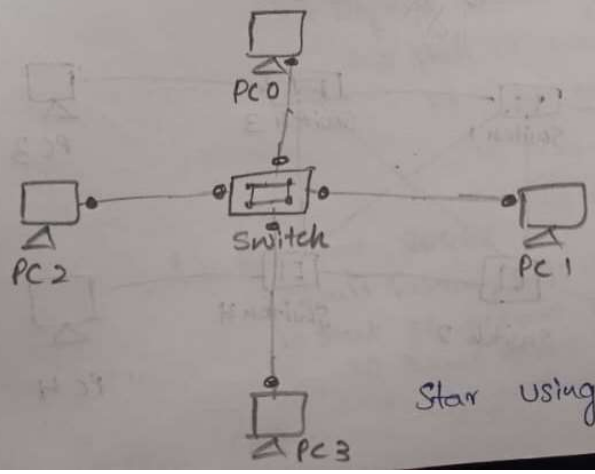
LAB 1

Create the topology and simulate simple POU from source to destination using HUB and switch as connecting and demonstrate the ping massage.

- Creating Star topology using HUB and Switch.



Star using HUB topology.



Star using Switch topolo

Star using HUB Topology.

Observation :- The pinged message is transfered to all the devices but acknowledged by only the distination

To ping a message
> device > desktop > Command prompt
> ping < ipaddress of distination >

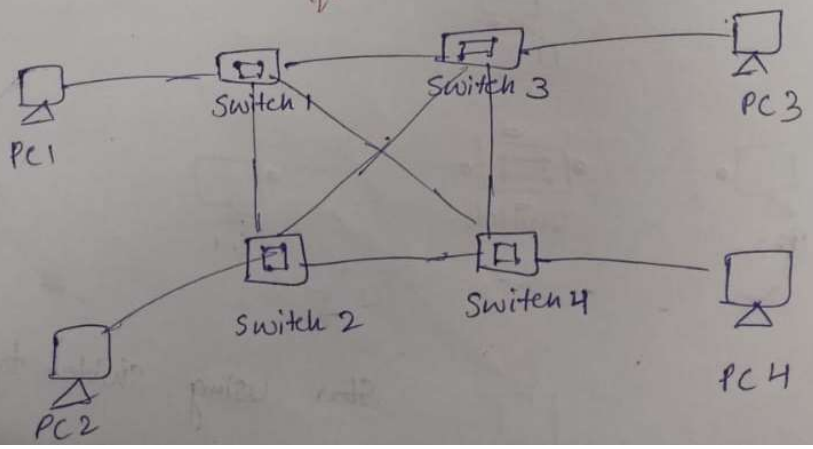
To know about the ipaddress of our device
> device > desktop > command prompt
> ipconfig.

Star topology using switch

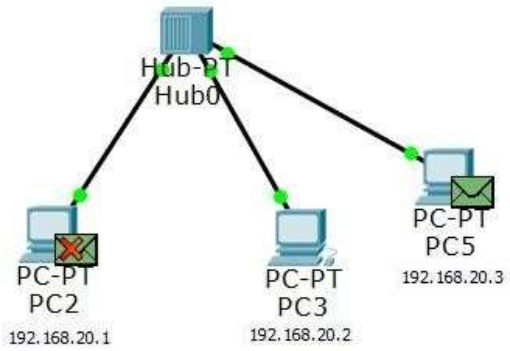
Observation : Switches are used in third layer of OSI model which is Network layer. Unlike hub, switch passes the message to only the distination device.

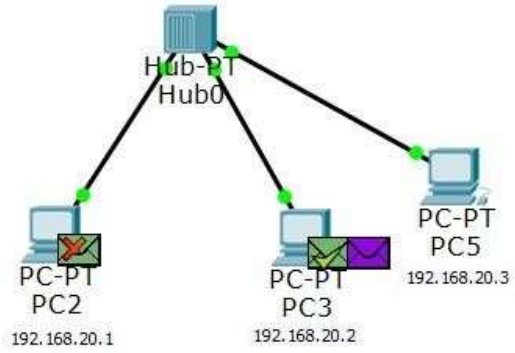
Mesh topology using switches :-

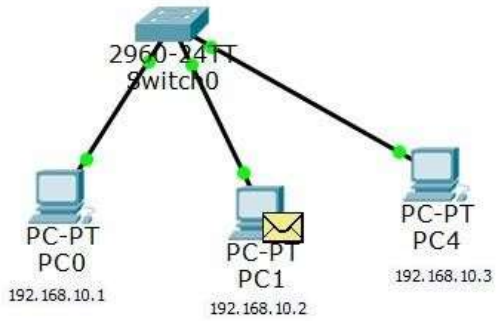
20/8/2025

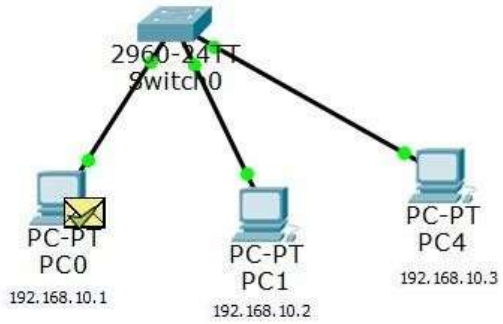


Screenshots/ Output:

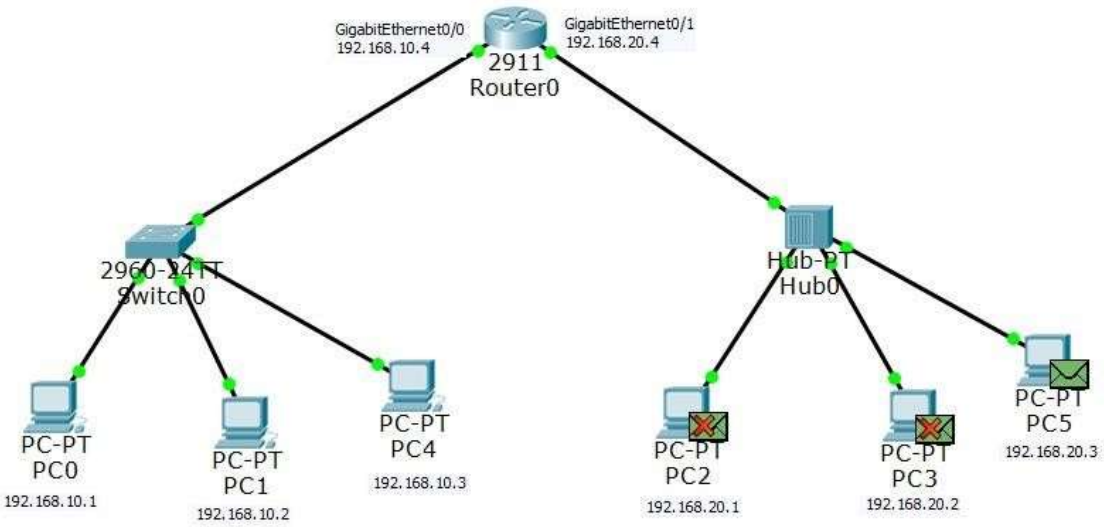


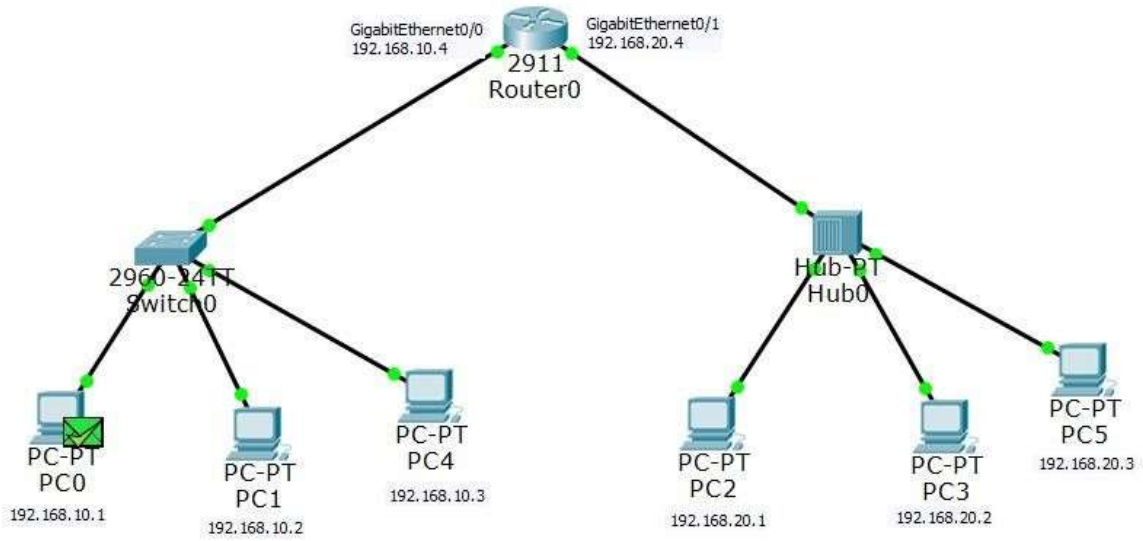






Updated topology





Observation:

- In the hub-based topology, the PDU was broadcast to all ports, while the switch forwarded the PDU only to the destination MAC after learning addresses from incoming frames.
- Successful ICMP echo and echo-reply messages confirm that both devices enabled connectivity, with the switch demonstrating selective unicast forwarding and reduced unnecessary traffic.

Program 2

Aim of the program:

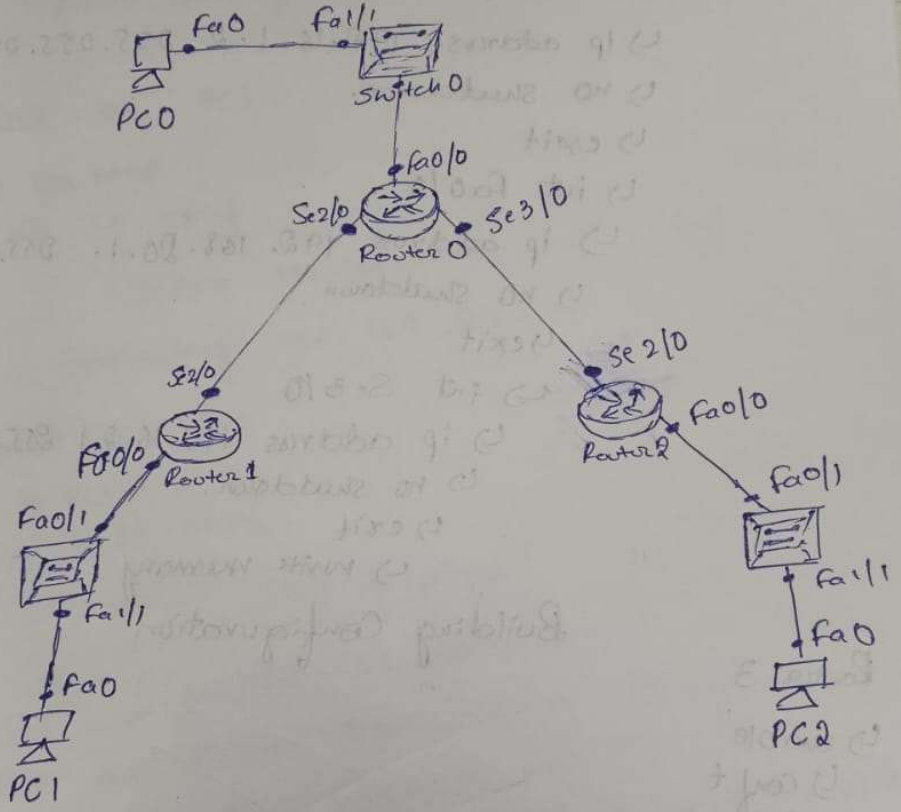
Configure default route, static route to the Router

Procedure and topology:

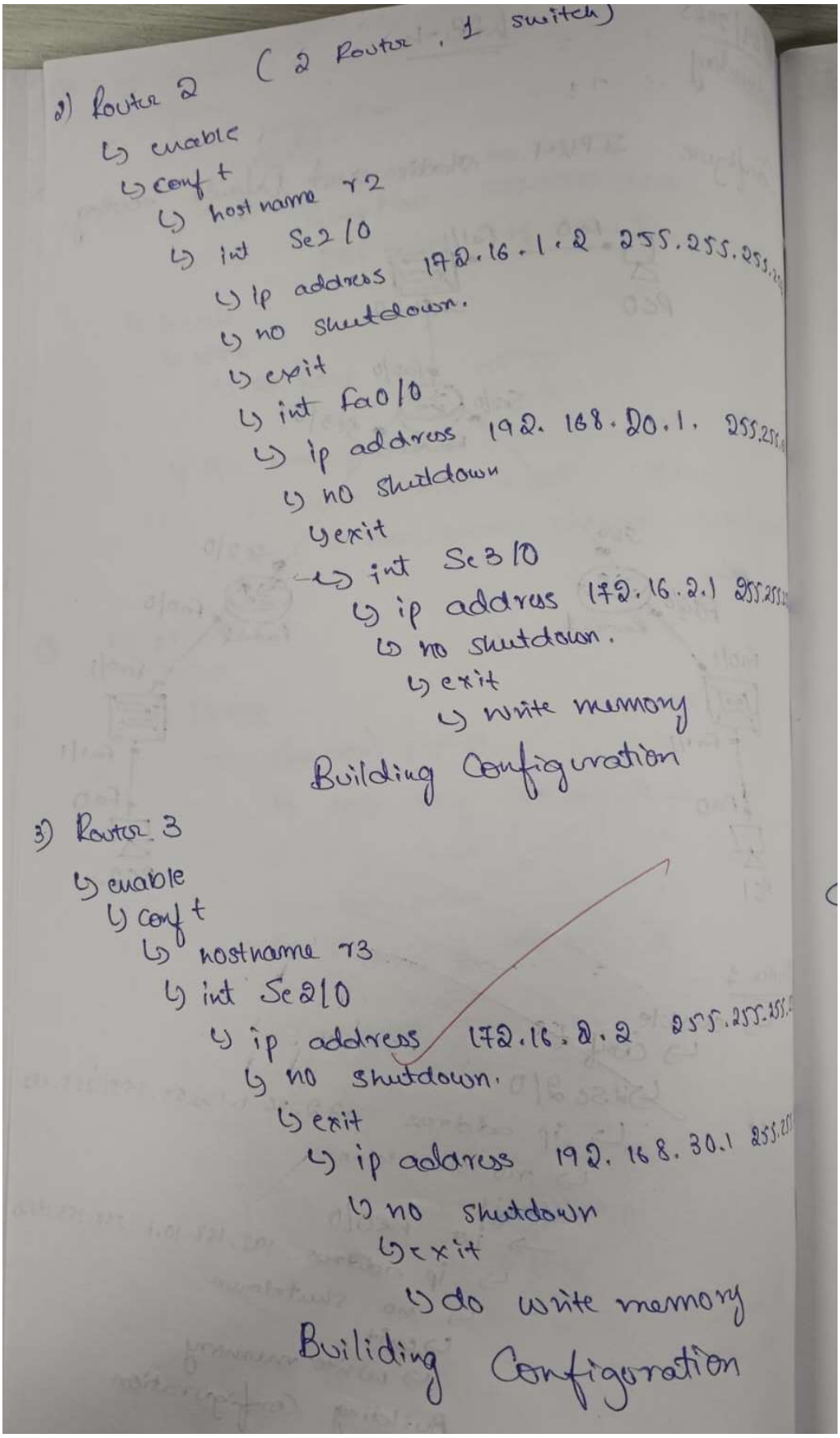
7/09/2025
Wednesday

LAB-4

Configure IPV4 static and Default routing.



1) Router 1
↳ enable
↳ conf t
↳ int se 2/0
↳ ip address 172.16.1.1 255.255.255.252
↳ no shutdown
↳ exit
→ int fa0/0
↳ ip address 192.168.10.1 255.255.255.0
↳ no shutdown
↳ exit
↳ write memory
Building Configuration



2) Router 2 (2 Router, 1 switch)

↳ enable
↳ conf t
↳ host name r2
↳ int Se2/0
↳ ip address 172.16.1.2 255.255.255.
↳ no shutdown.
↳ exit
↳ int fa0/0
↳ ip address 192.168.20.1 255.255.
↳ no shutdown
↳ exit
↳ int Se3/0
↳ ip address 172.16.2.1 255.255.
↳ no shutdown.
↳ exit
↳ write memory

Building Configuration

3) Router 3

↳ enable
↳ conf t
↳ hostname r3
↳ int Se2/0
↳ ip address 172.16.2.2 255.255.255.
↳ no shutdown.
↳ exit
↳ ip address 192.168.30.1 255.255
↳ no shutdown
↳ exit
↳ do write memory

Building Configuration

Click on PC0

↳ Desktop

↳ ip configuration

IPv4 ↳ 192.168.10.10

Default Gateway 192.168.10.1

Click on PC1

↳ Desktop

↳ IP conf

↳ IPv4 192.168.20.10

Gateway 192.168.20.1

Click on PC3

↳ Desk

↳ IP conf

↳ IPv4 192.168.30.10

Gateway 192.168.30.1

Click on Router R1

R1 # conf t

R1(Conf) # ip route 192.168.20.0 255.255.255.0 172.16.1.2

ip route 172.16.2.0 255.255.255.252 172.16.1.2

ip route 192.168.30.0 255.255.255.0 172.16.1.2

exit

wr

Click on Router R2

Conf t

ip route 192.168.10.0 255.255.255.0 172.16.1.1

ip route 192.168.30.0 255.255.255.0 192.16.1.2

exit

wr

Click on R3

R3# conf t

R3 (conf) # ip route 0.0.0.0 0.0.0.0 S0/0/1

exit

wr

To see Routing table

click on R1

R1 # Show IP route

click on R2

R2# Show IP route

click on R3

R3# Show IP route

click on PC0

↳ Desktop

↳ Command Prompt

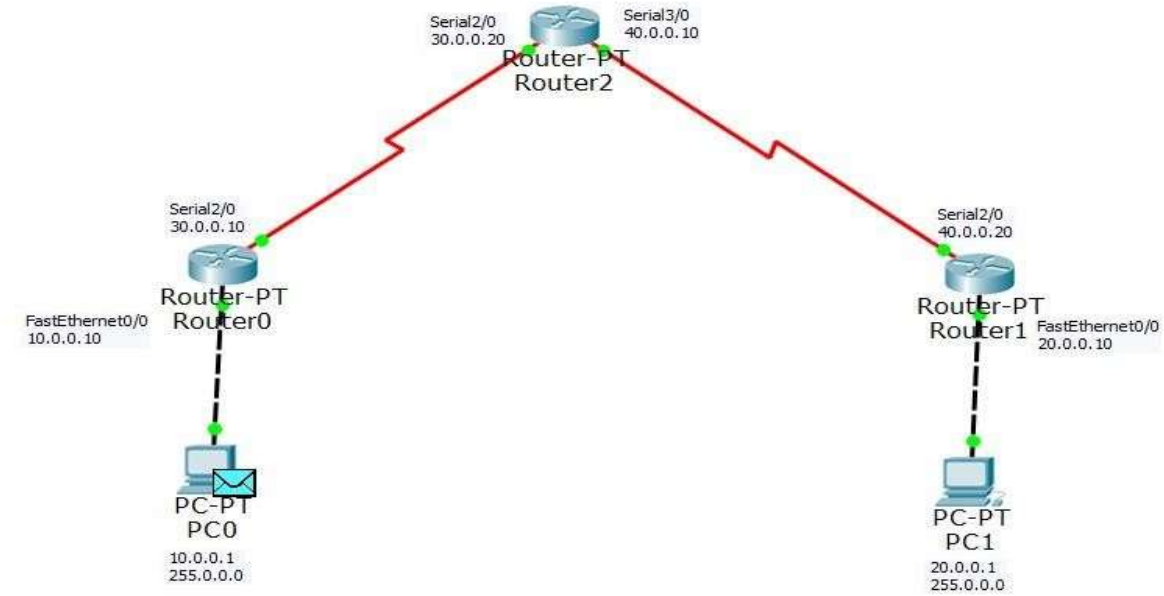
↳ Ping 192.168.10.1 ↵

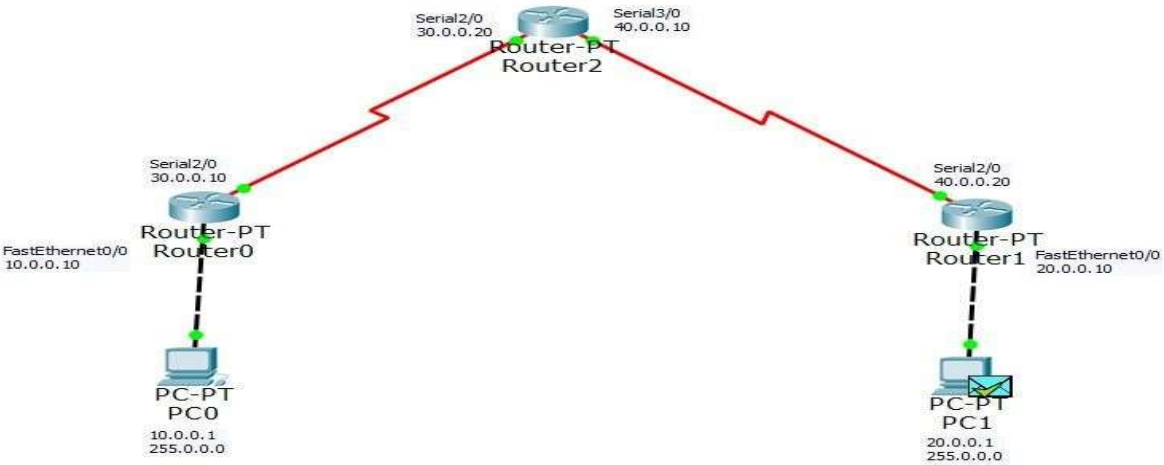
↳ Ping 192.168.20.1 ↵

Ping 192.168.30.1 ↵

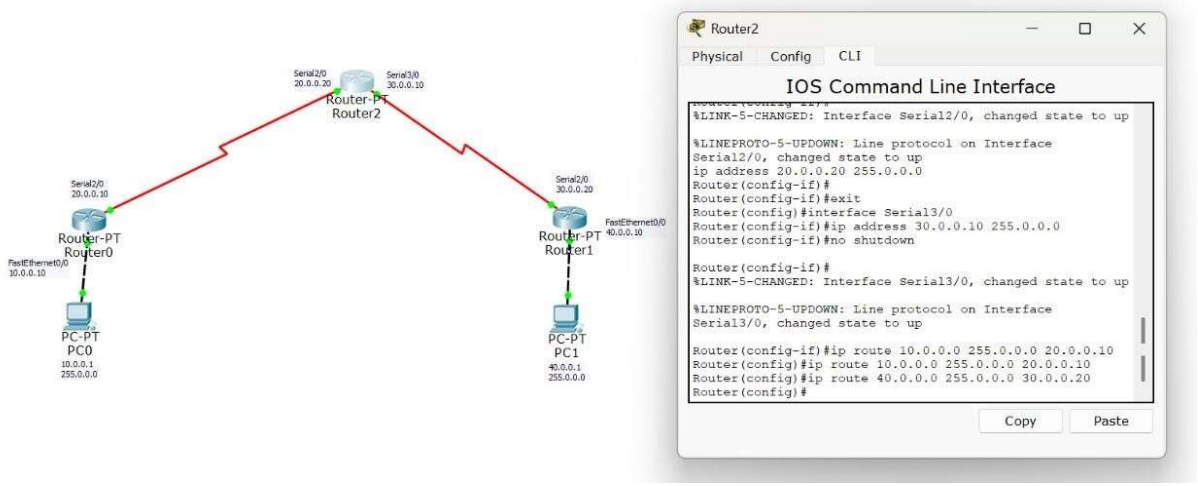
Once ping is Successful then send PDU from PC0 to PC1

Screenshots/ Output:

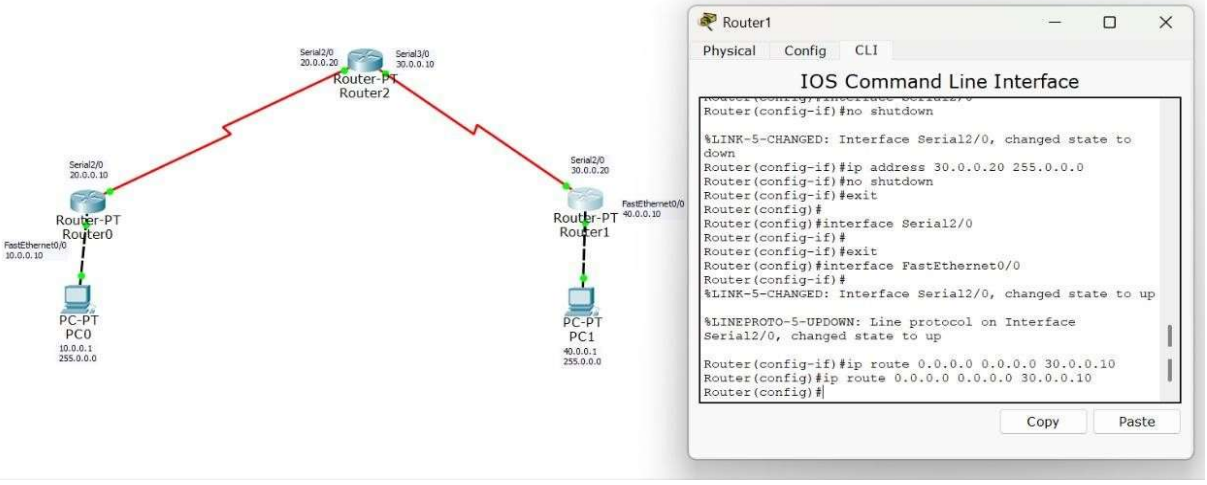




Static routing CLI commands:



Default routing CLI commands:



Observation:

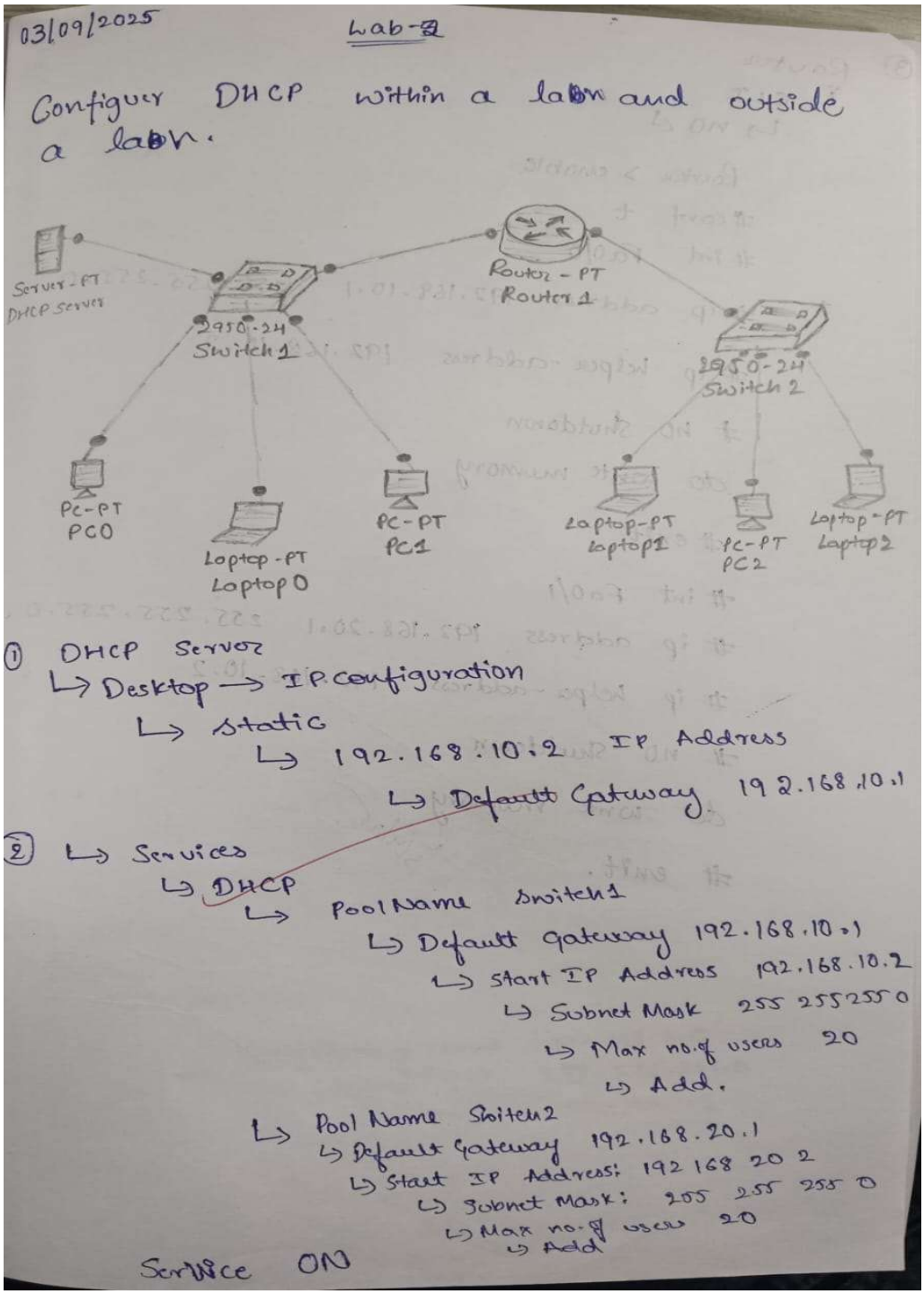
- The configured static and default routes correctly updated the router's routing table, enabling deterministic next-hop selection for remote networks.
- Successful ping tests verified that traffic was forwarded according to the static/default route entries, ensuring end-to-end reachability across different network segments.

Program 3

Aim of the program:

Configure DHCP within a LAN and outside LAN.

Procedure and topology:



③ Router

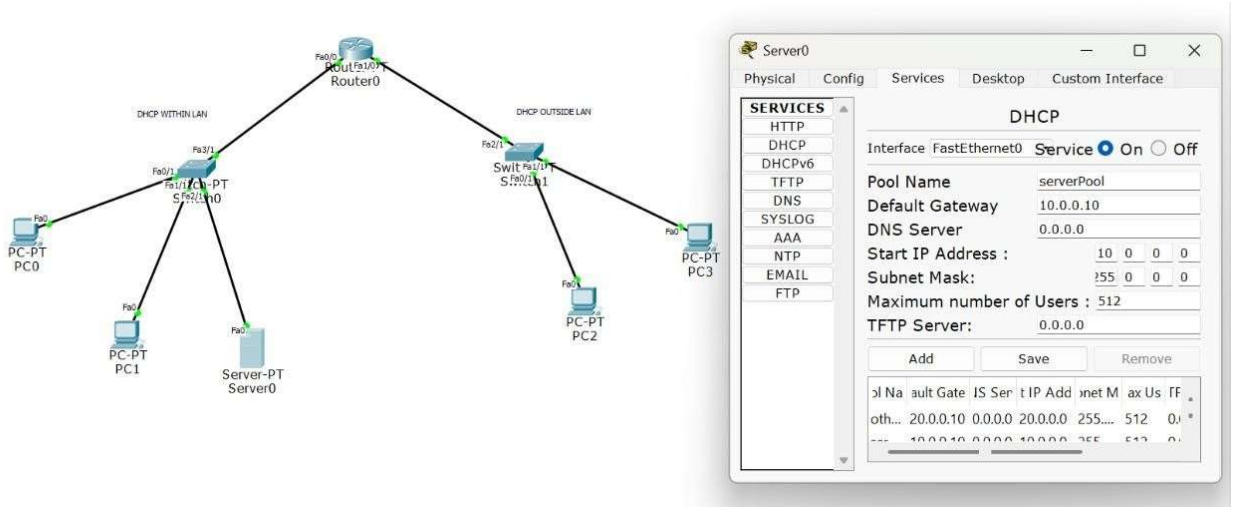
↳ CLI

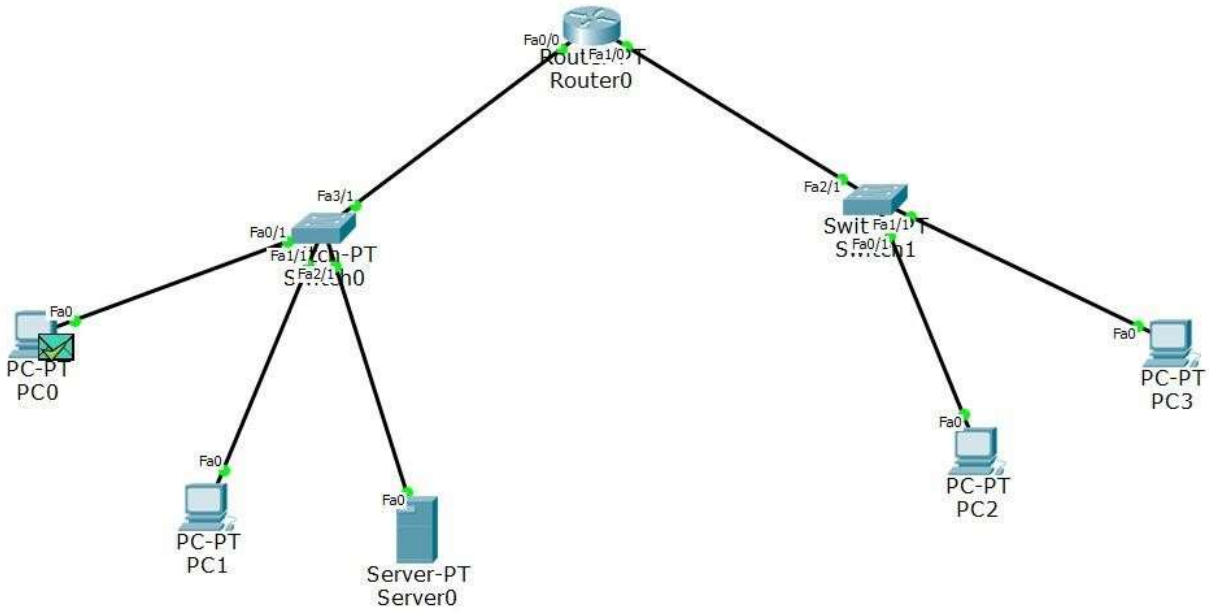
↳ no ↵

```
Router > enable
#conf t
#int Fa0/0
#ip address 192.168.10.1 255.255.255.0
#ip helper-address 192.168.10.2
#no shutdown
do write memory
#exit
#int Fa0/1
#ip address 192.168.20.1 255.255.255.0
#ip helper-address 192.168.10.2
#no shutdown
do write memory
#exit.
```

3/9/25

Screenshots/ Output:





Observation:

- The DHCP server successfully allocated IP addresses to clients within the LAN, confirming proper scope configuration and automatic distribution of network parameters.
- DHCP relay (IP Helper) enabled clients outside the LAN to obtain leases from the central DHCP server, demonstrating correct inter-network forwarding of DHCP Discover and Offer messages.

Program 4

Aim of the program:

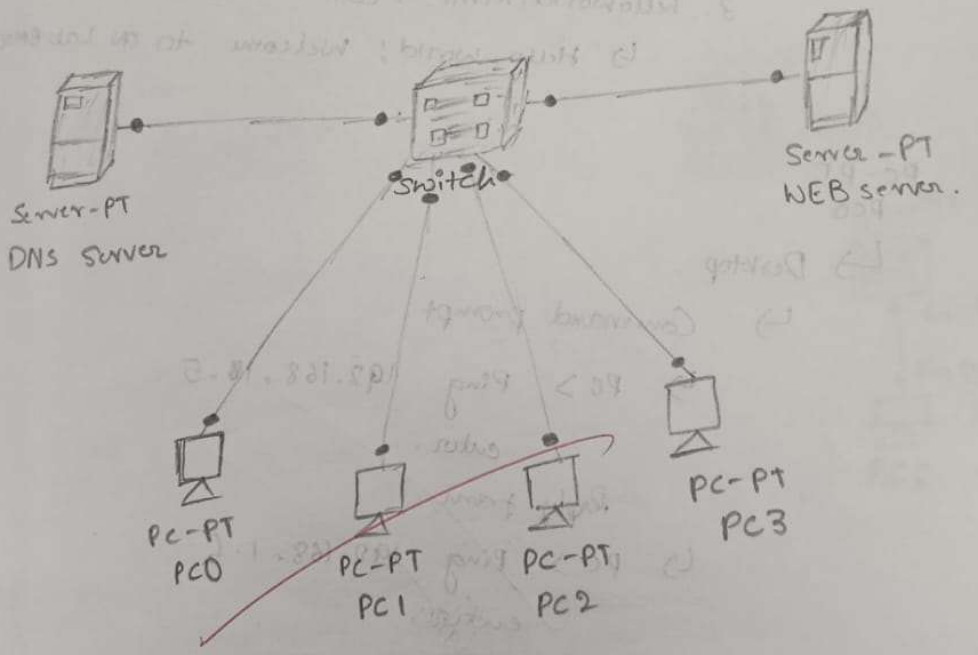
Configure Web Server, DNS within a LAN.

Procedure and topology:

10/09/2025
Wednesday

Lab-13

1) Configure web server, DNS within a LAN
2) Configure IP address to routes in packet tracer. Explore the following messages.
   i) Ping response
   ii) Destination unreachable
   iii) Request time out
   iv) Reply.



① DNS server
↳ Desktop
↳ IP address
↳ Static
↳ IP Address 192.168.1.5
Subnet Mask 255.255.255.0
DNS server 192.168.1.5

↳ Services
↳ DNS
↳ ON
↳ Name www.letslearn.com
↳ Type A Record
↳ 192.168.1.6 → Add

② WEB server.

↳ Desktop

↳ IP configuration

↳ static

↳ IP address 192.168.1.6

Subnet Mask 255.255.255.0

DNS server 192.168.1.5

↳ Services

↳ HTTP

↳ HTTP ON , HTTPS ON

↳ File Manager.

3. helloworld.html → edit Click

↳ Hello, world! Welcome to CN Lab BMSCE

③ PC-PT

PC0

↳ Desktop

↳ Command prompt

↳ PC > Ping 192.168.10.5

enter.

Reply from

↳ PC > Ping 192.168.1.6

enter

Reply from.

↳ Web Browser

↳ URL www.letslearn.com

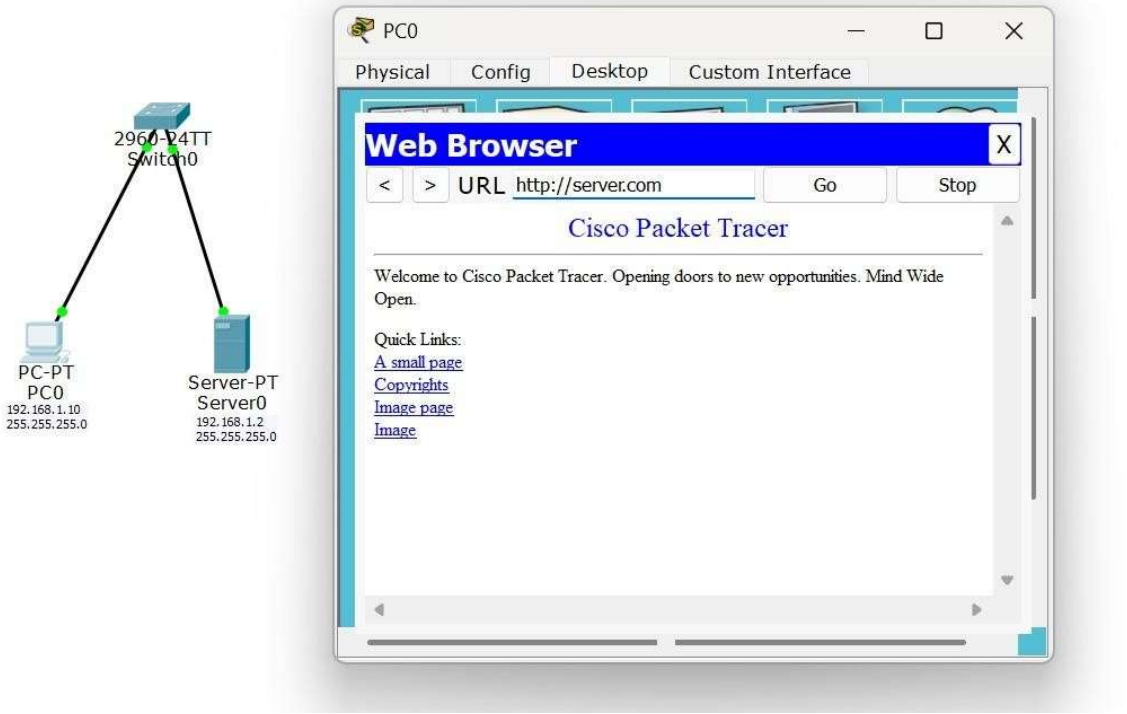
↳ enter (A0)

↳ A small page

Hello, world! Welcome to

CN Lab BMSCE

Screenshots/ Output:



Observation:

- The DNS server successfully resolved domain names to the corresponding web server's IP address, confirming proper hostname-to-IP mapping within the LAN.
- HTTP requests reached the web server using the DNS-resolved address, validating correct server configuration and internal LAN communication.

Program 5

Aim of the program:

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

Procedure and topology:

8/10/25 Lab-5

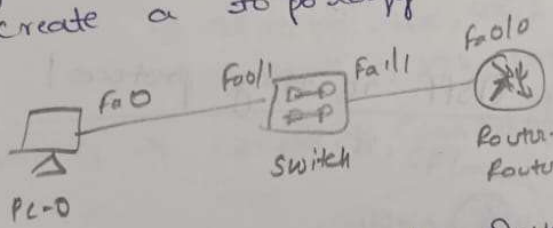
Configure telnet to access router remotely

Telnet is used to access remote server it is simple cmd line tool that runs on your comp and it allow you to send cmd remotely to a server and Admistration.

→ Telnet is also used to manage other devices like Router, switch to check if ports are open or closed on server.

fig

① create a topology

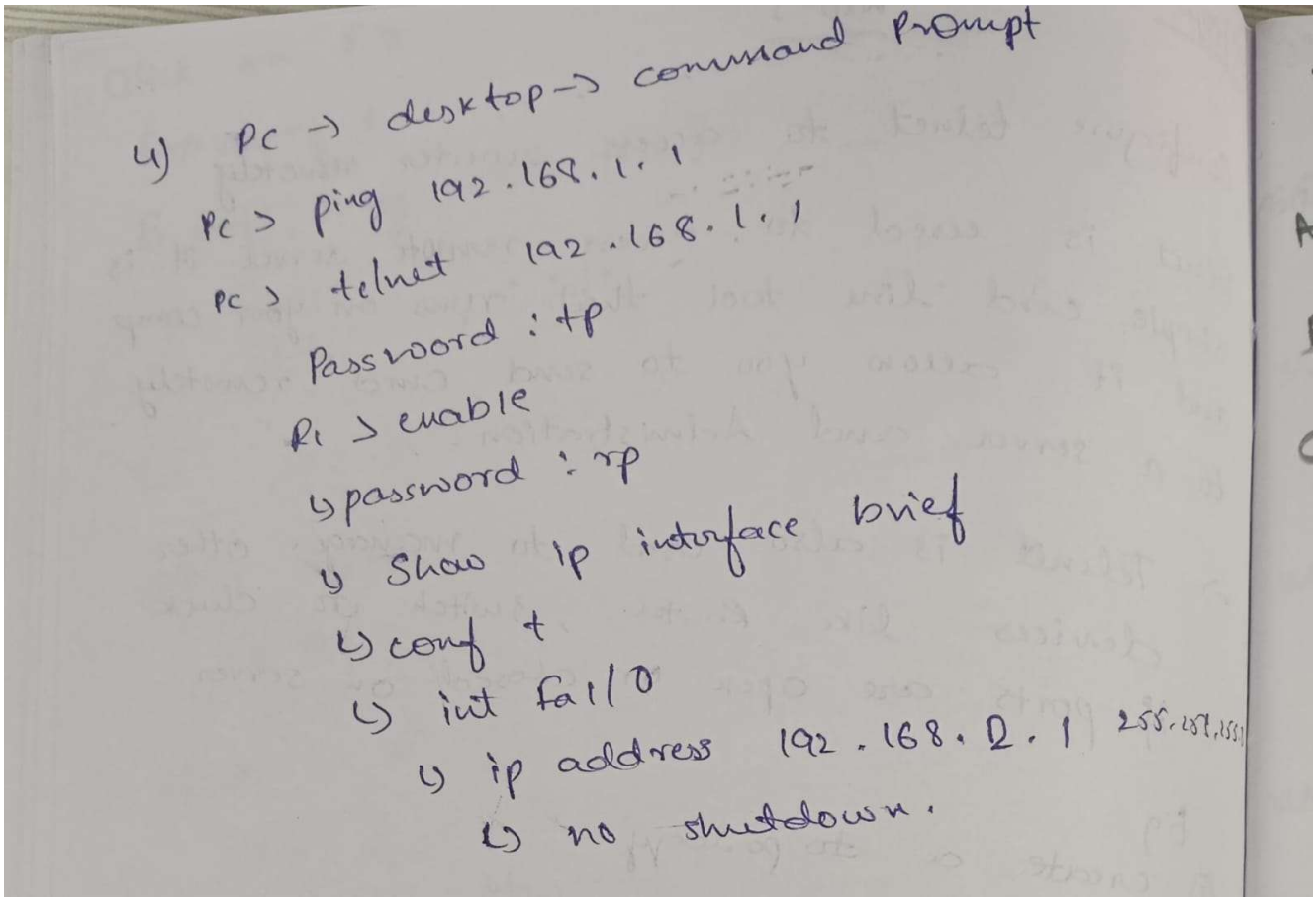


② In PC-0 → go to Desktop → IP configration

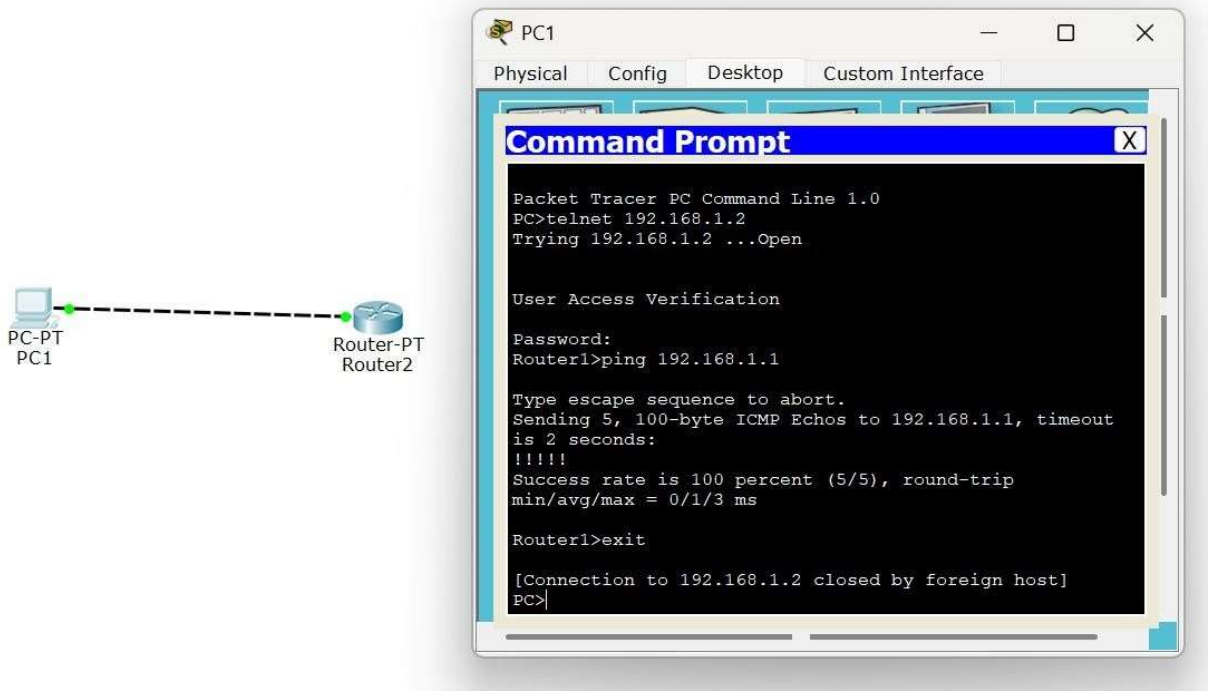
IP-address → 192.168.1.2
subnet mask → 255.255.255.0
Default gateway → 192.168.1.1 (Router)

③ Router Configuration

Router → enable
→ conf t
↳ int fa0/0
↳ ip address 192.168.1.1 255.255.255.0
↳ no shutdown
↳ line vty 0 5
↳ password tp
↳ exit
↳ wr
→ exit
→ show ip interface brief



Screenshots/ Output:



Observation:

- The Telnet session successfully established a remote CLI connection to the router, confirming proper VTY line configuration and IP reachability between the IT office PC and the server room router.
- Command execution over the Telnet session demonstrated reliable remote device management.

Program 6

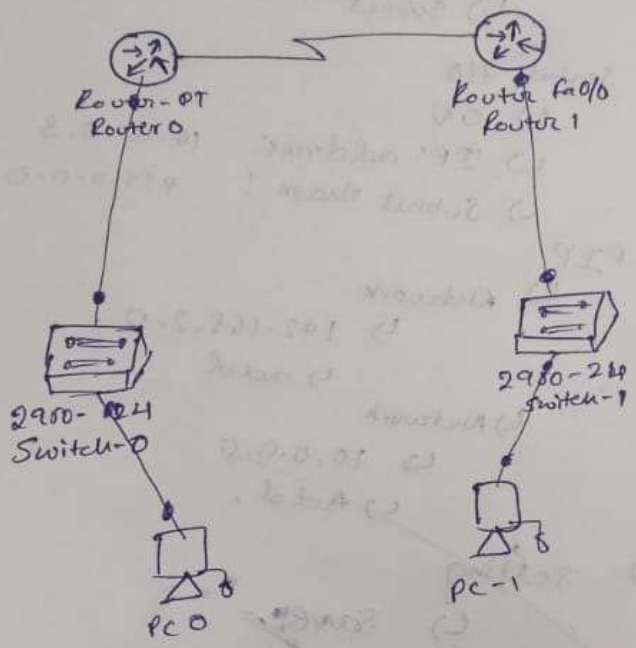
Aim of the program:

Configure RIP routing Protocol in Routers

Procedure and topology:

RIP

- Configure RIP routing protocol in routers to transfer packet from nod-k to nod-b.



PC0
↳ Desktop
↳ IP address : 192.168.1.2
Subnet Mask : 255.255.255.0
Default Gateway : 192.168.1.1

PC-1
↳ Desktop
↳ IP address : 192.168.2.2
Subnet Mask : 255.255.255.0
Default gateway : 192.168.2.1

Router 0
↳ Config
↳ FastEthernet 0/0
↳ on
↳ IP address : 192.168.1.1
↳ Subnet Mask : 255.255.255.0
↳ Serial 2/0
↳ on
↳ IP address : 10.10.0.2
↳ Subnet Mask : 255.0.0.0
↳ RIP
↳ Network → 192.168.1.0 ↳ add.
↳ Network → 10.0.0.0 → add
↳ setting
↓
save

Router-1

↳ config

↳ Fast Ethernet 0/0

↳ ON

↳ IP configuration 192.168.2.1

↳ Subnet Mask 255.255.255.0

↳ Serial 2/0

↳ ON

↳ IP address: 10.10.0.3

↳ Subnet Mask: 255.0.0.0

↳ RIP

↳ Network

↳ 192.168.2.0

↳ add

↳ Network

↳ 10.0.0.0

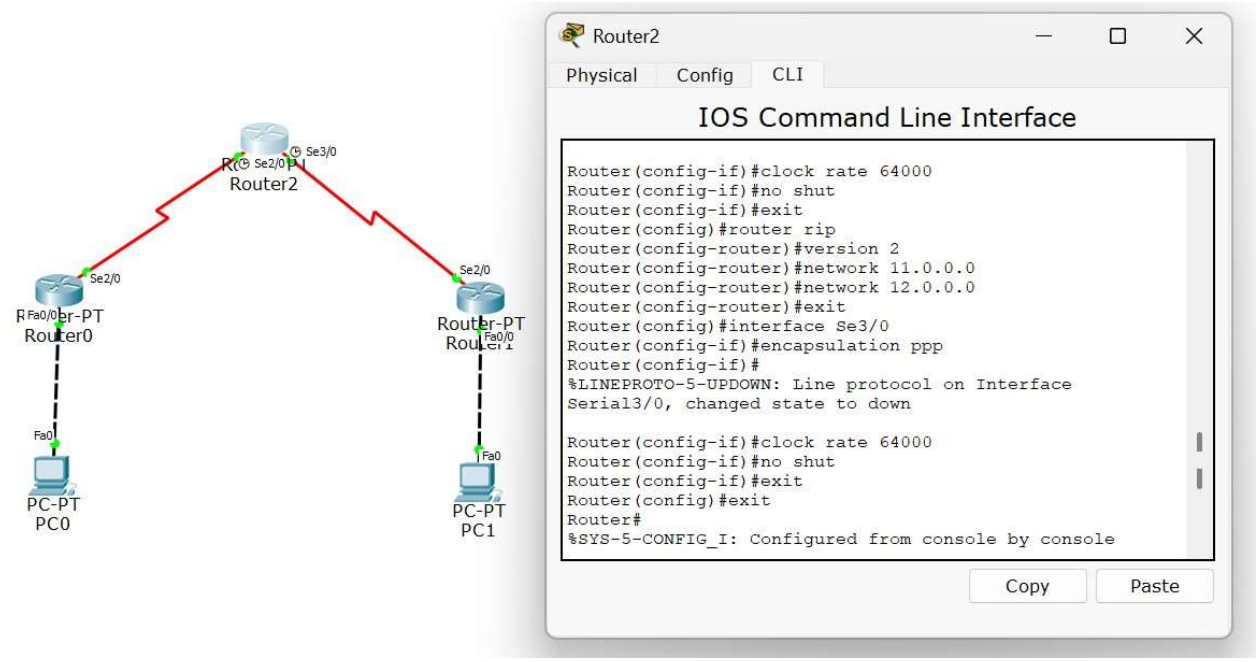
↳ Add.

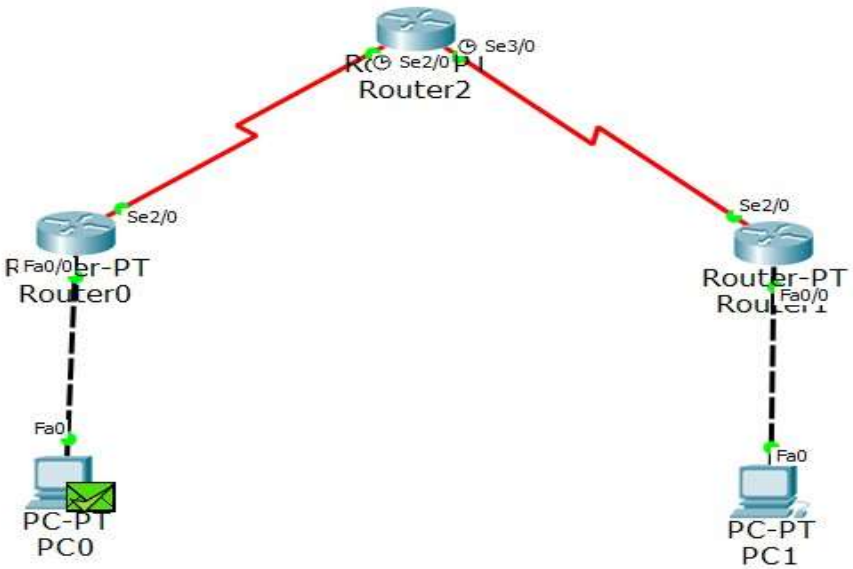
↳ setting

↳ save

19/11/25

Screenshots/ Output:





Observation:

- RIP routing updates were successfully exchanged between routers, allowing each router to dynamically learn remote network routes through hop-count–based distance vector advertisements.

- The routing tables converged correctly, and successful ping tests confirmed end-to-end connectivity maintained by periodic RIP updates and route propagation.

Program 7

Aim of the program:

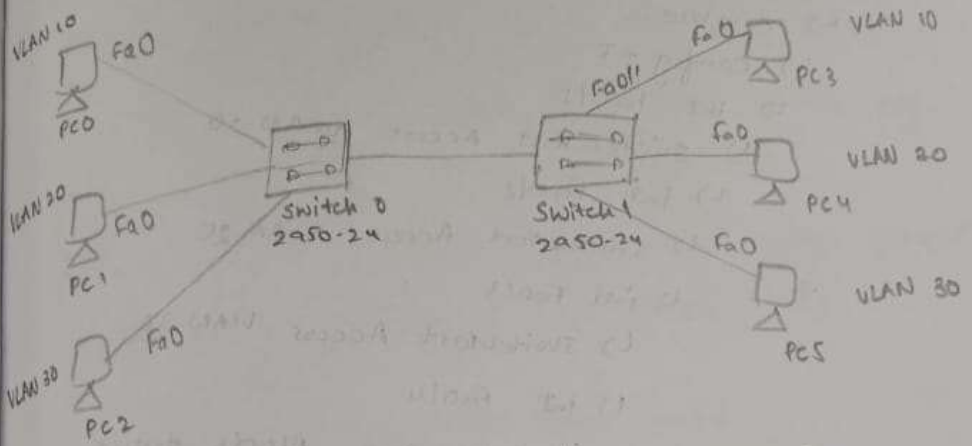
To construct a VLAN and make the PCs communicate among a VLAN

Procedure and topology:

15/10/25

Lab - 6

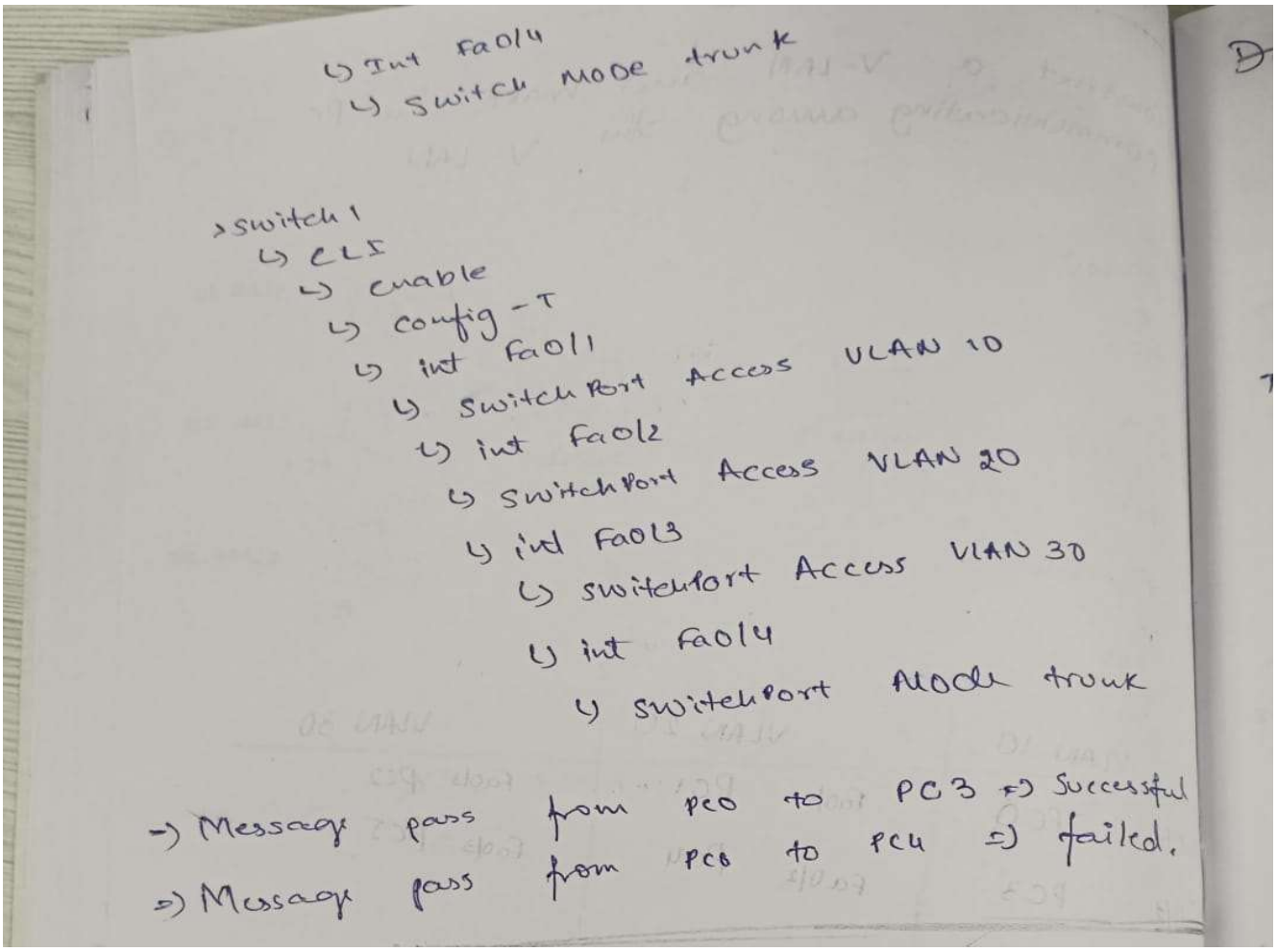
(OR) Construct a V-LAN and make the PC communicating among the V-LAN.



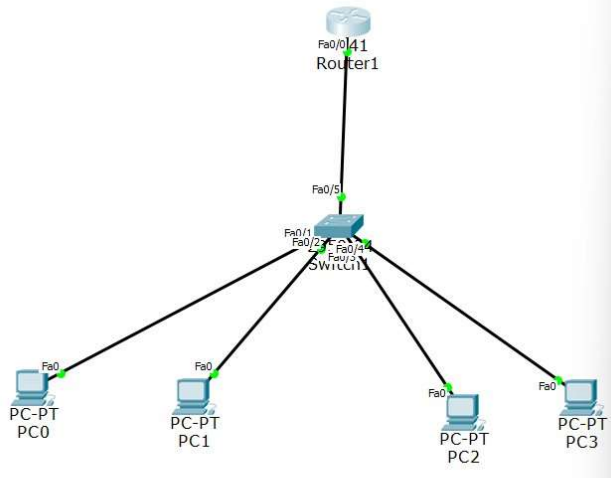
VLAN 10		VLAN 20		VLAN 30	
Fa0/1	PC0	Fa0/2	PC1	Fa0/3	PC2
Fa0/2	PC3	Fa0/2	PC4	Fa0/3	PC5

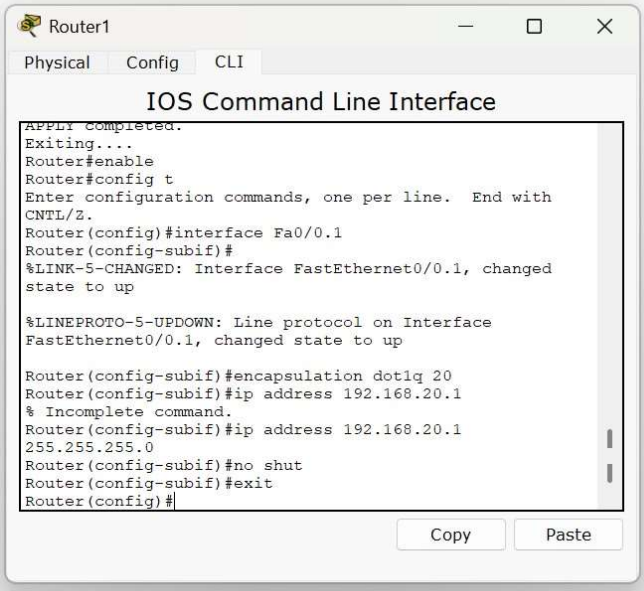
PC0	192.168.1.2
PC1	192.168.1.3
PC2	192.168.1.4
PC3	192.168.1.5
PC4	192.168.1.6
PC5	192.168.1.7

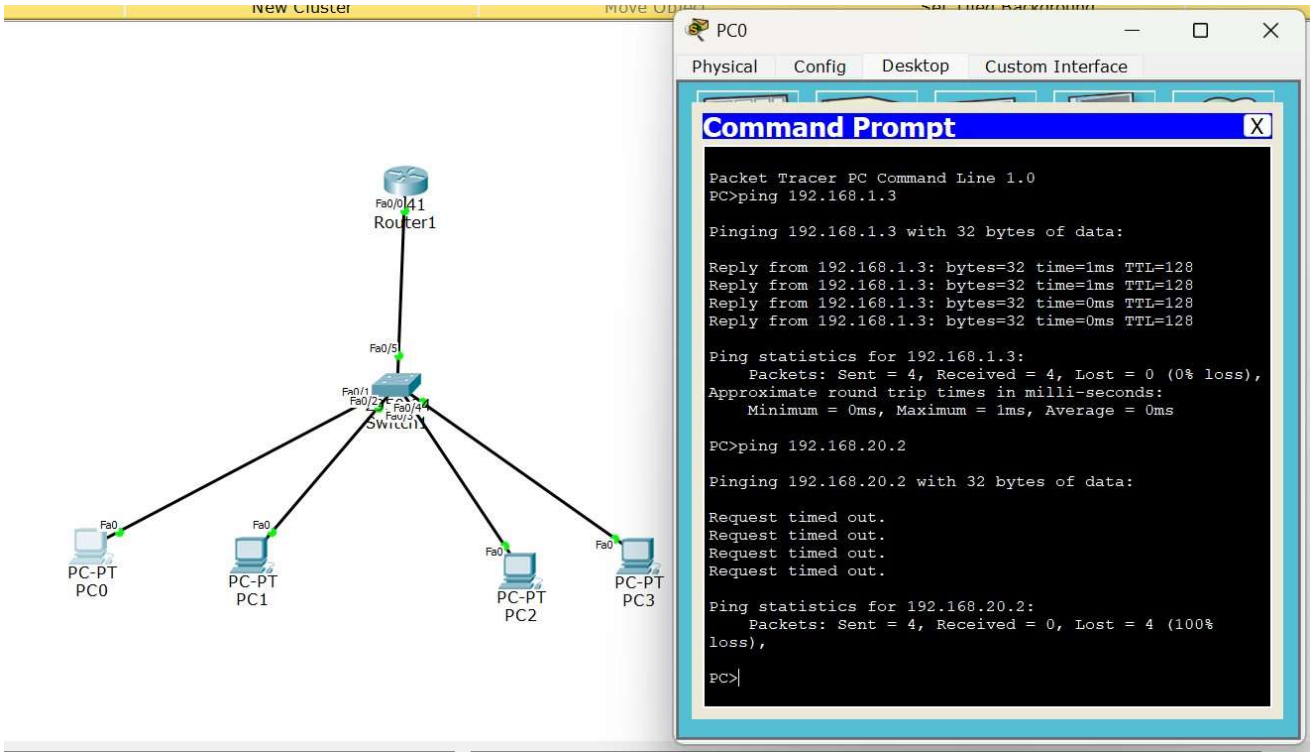
> Switch 0
↳ CLI
↳ enable
↳ Config T
↳ Int Fa0/1
↳ switchport Access VLAN 10
↳ Int Fa0/2
↳ switchport Access VLAN 20
↳ Int Fa0/3
↳ switchport Access VLAN 30



Screenshots/ Output:







Observation:

- VLAN segmentation successfully separated broadcast domains, and switch ports were correctly assigned to their respective VLAN IDs using access mode configuration.
- Inter-VLAN communication was achieved through the Layer-3 device, and successful ping tests confirmed proper VLAN membership, tagging, and routing functionality.

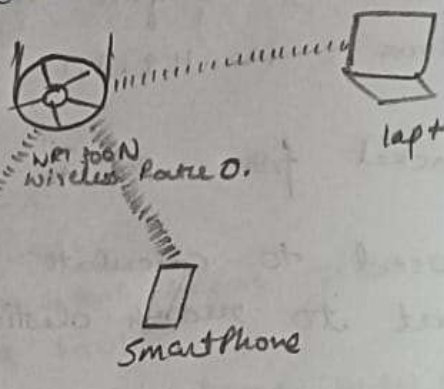
Program 8

Aim of the program:

To construct a WLAN and make the nodes communicate wirelessly

Procedure and topology:

6c) To construct a W-LAN and make the nodes to communicate wirelessly.



Connect SmartPhone

Click on PC → Physical → turn-off → drag and drop that LAN to modules, then to drop that empty space add new one then turn-on → Connection will be done automatically.

[To change that default wired connection to wireless]. [IP add of all PC will be done automatically].

Router → config → Wireless

SSID : bmsce

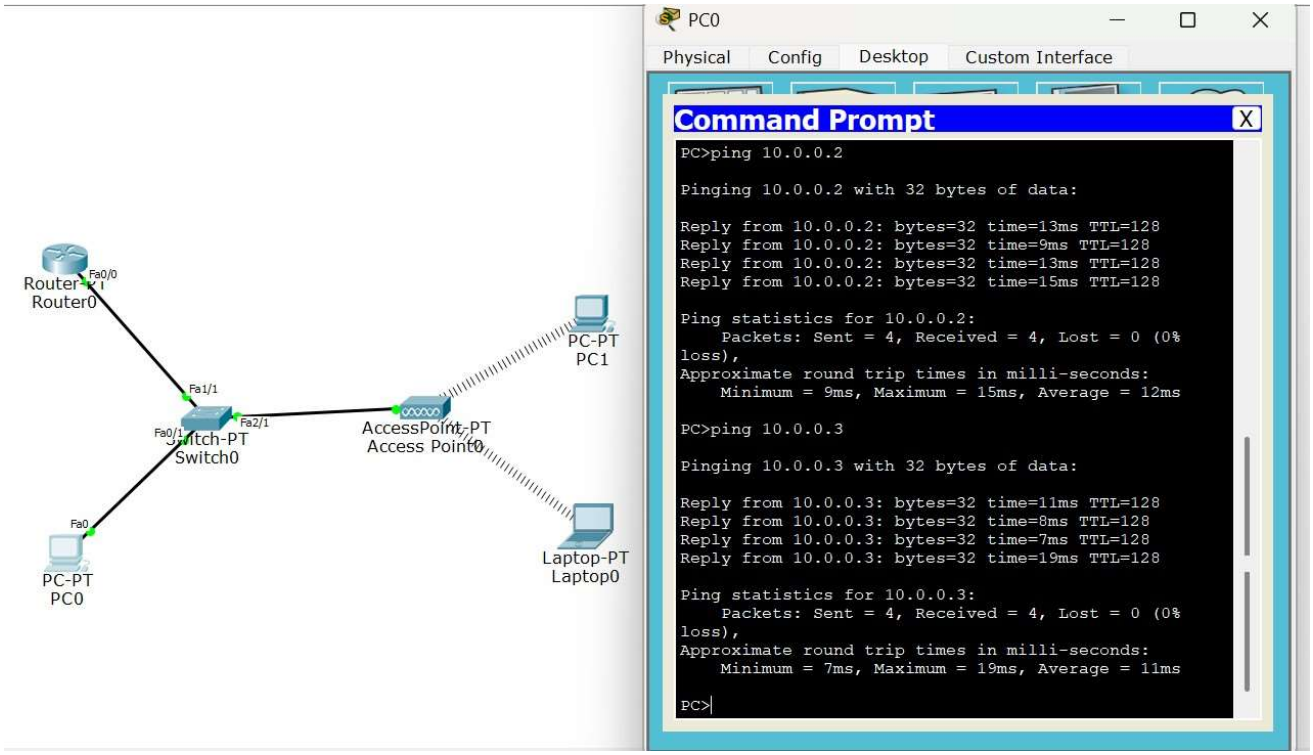
Authentication : WAPA2-PSK Phase : bmsce1234 [password]

[Now automatic connection to PC will be lost].

15/10/24

PC → click config → wireless → Add that SSID & connection will be done.

Screenshots/ Output:



Observation:

• The WLAN configuration enabled wireless nodes to associate with the access point using the configured SSID and security settings, confirming proper authentication and signal coverage. • Successful ping communication between wireless devices verified stable wireless Connectivity.

Program 9

Aim of the program:

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

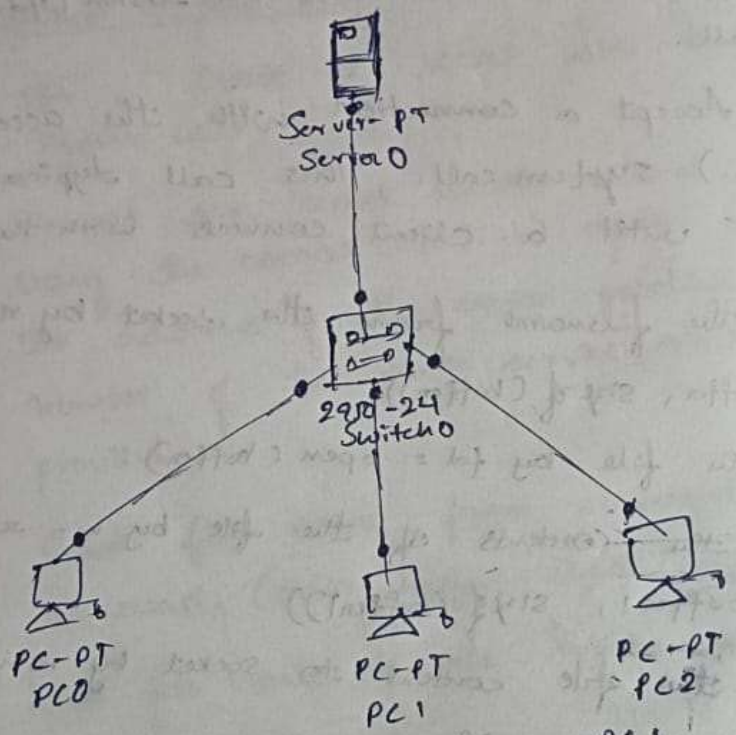
Procedure and topology:

19/11/2025

ARP

ARP → Address resolution Protocol

- ARP is used to map an IP address to a mac address.
- ARP is used to get data link layer address, mac address with the help of IP address



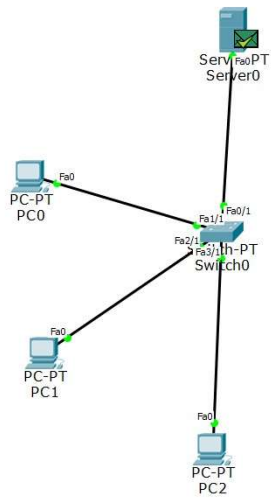
PC0
↳ Desktop
↳ IP configuration
↳ IP add : 192.168.11.1
↳ Sub Mask : 255.255.255.0
↳ DNS server : 192.168.11.4

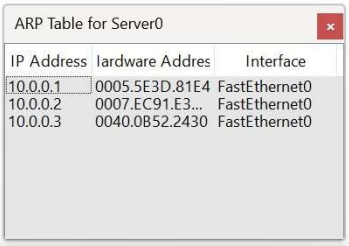
PC1
↳ Desktop
↳ IP conf
↳ IP add : 192.168.11.2
↳ Sub Mask : 255.255.255.0
↳ DNS server : 192.168.11.4

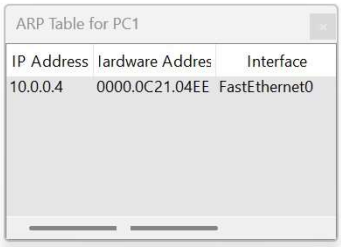
PC2
↳ Desktop
↳ IP config
↳ IP add : 192.168.11.3
↳ Sub Mask : 255.255.255.0
↳ DNS server : 192.168.11.4

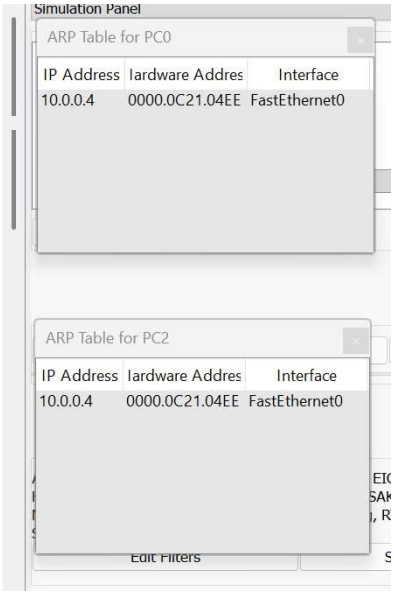
Server
↳ Desktop
↳ IP config
↳ IP add : 192.168.11.4
↳ Sub mask : 255.255.255.0

Screenshots/ Output:









Observation:

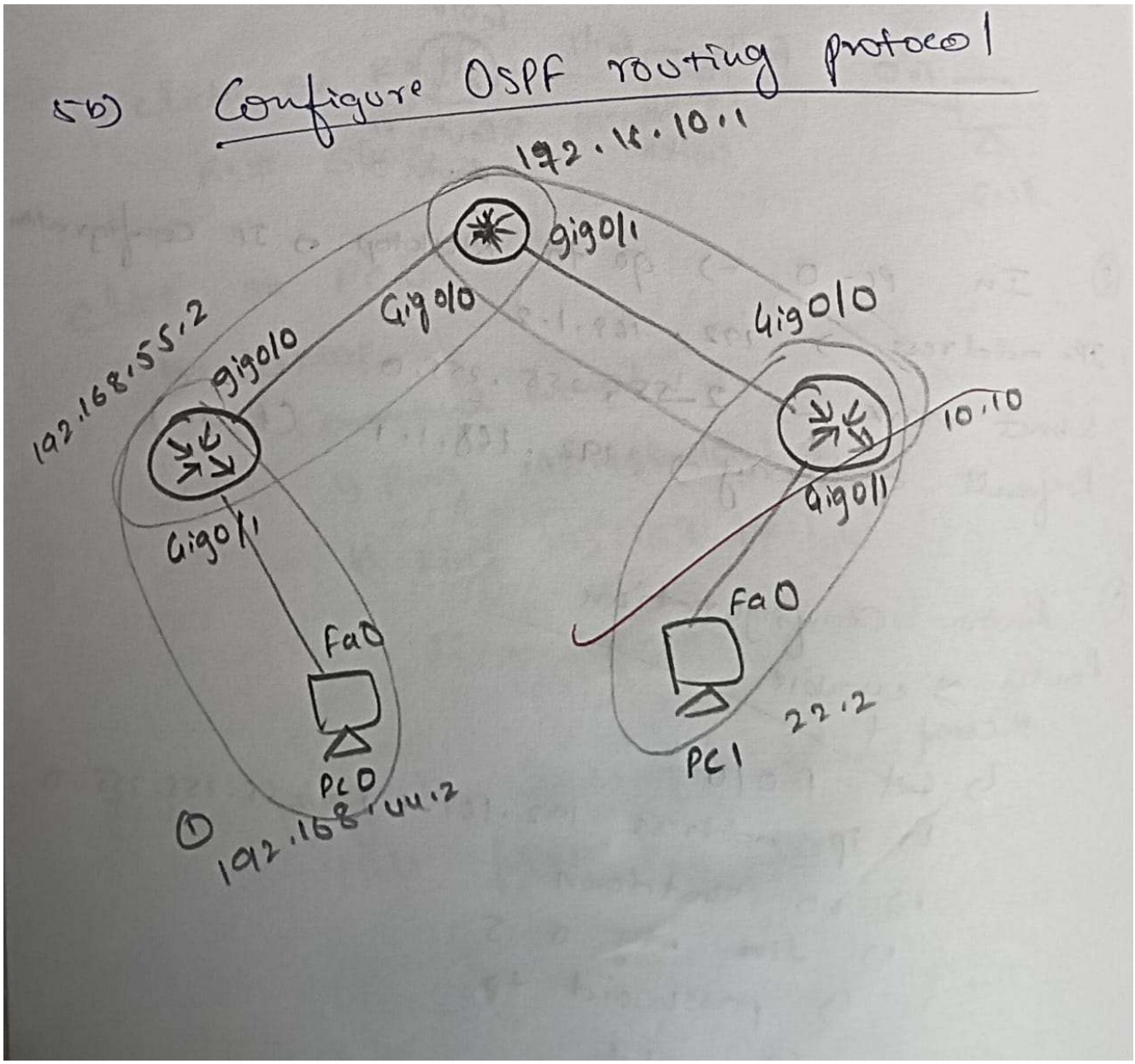
- ARP successfully resolved the destination IP address to its corresponding MAC address, as seen from ARP request and reply exchanges between LAN hosts.
- The populated ARP tables and successful ping communication confirmed correct layer-2 addressing, frame forwarding, and basic LAN operation.

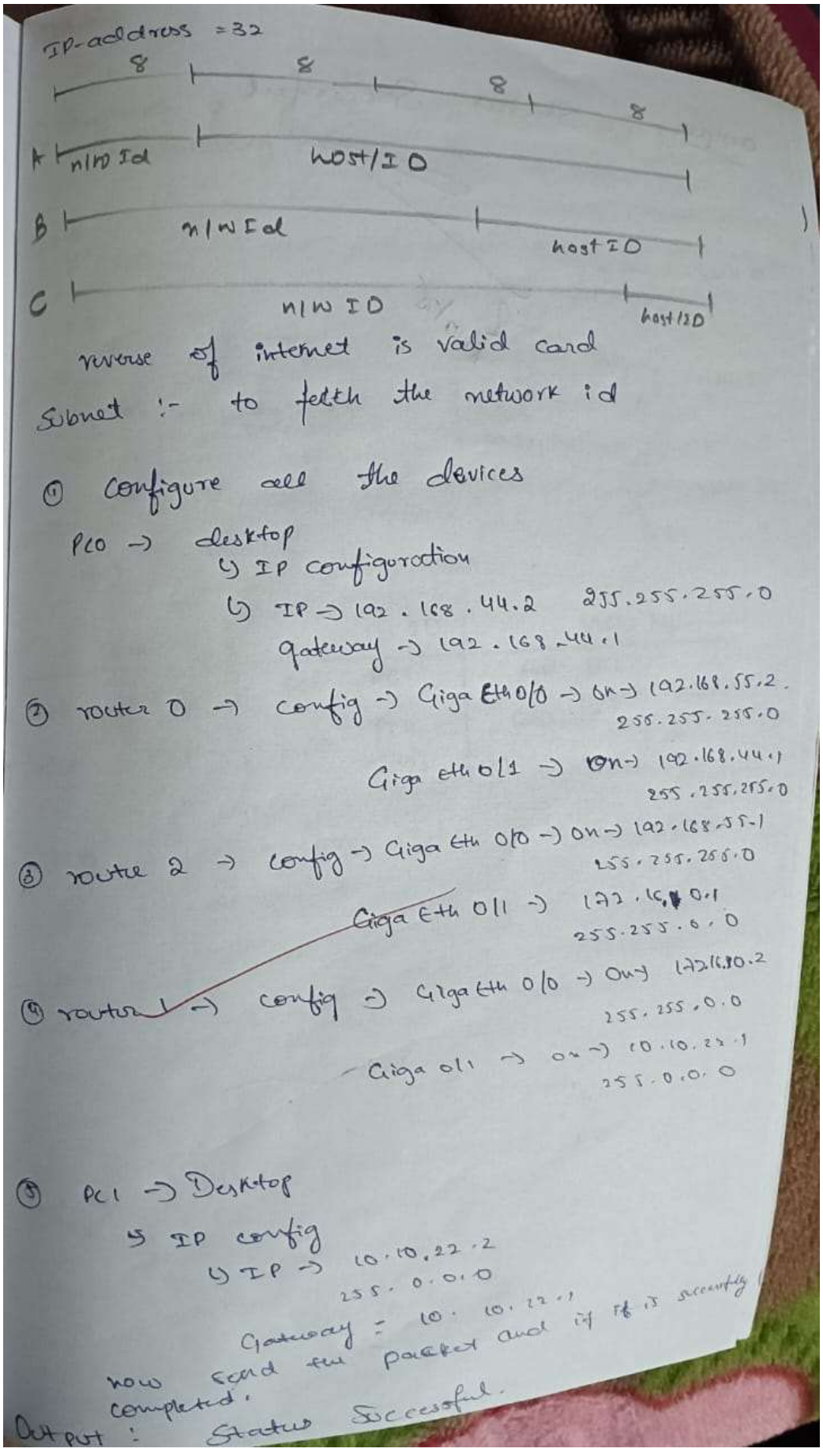
Program 10

Aim of the program:

Configure OSPF routing protocol

Procedure and topology:


5b) Configure OSPF routing protocol
172.16.10.1
gig0/1
Gig0/0
192.168.55.2
gig0/0
Gig0/1
FaO
PC0
192.168.44.2
Gig0/0
10.10
Gig0/1
FaO
PC1
22.2



reverse of internet is valid card

Subnet :- to fetch the network id

① configure all the devices

PC0 → desktop
↳ IP configuration
↳ IP → 192.168.44.2 255.255.255.0
gateway → 192.168.44.1

② router 0 → config → Giga Eth 0/0 → On → 192.168.55.2
255.255.255.0

Giga eth 0/1 → On → 192.168.44.1
255.255.255.0

③ router 2 → config → Giga Eth 0/0 → On → 192.168.55.1
255.255.255.0

Giga Eth 0/1 → 172.16.0.1
255.255.0.0

④ router 1 → config → Giga Eth 0/0 → On → 172.16.0.2
255.255.0.0

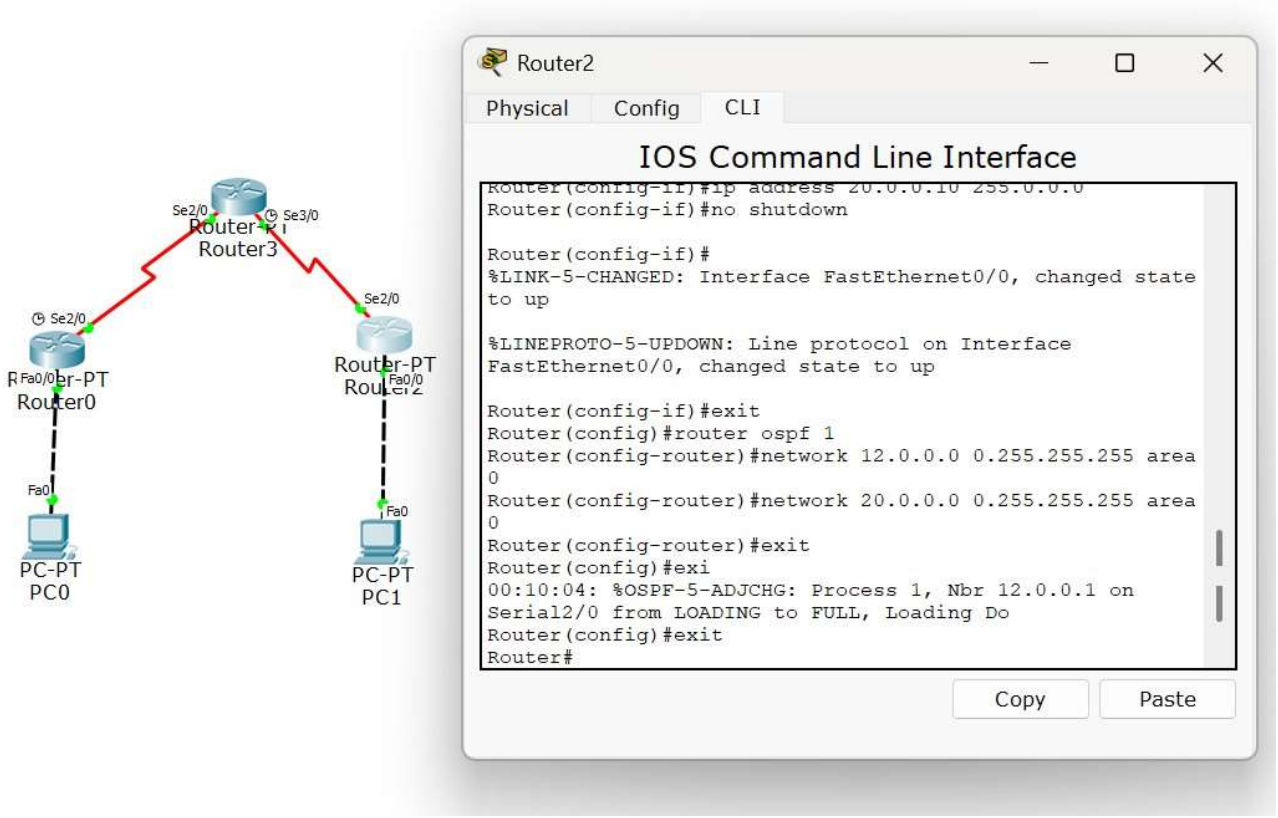
Giga 0/1 → On → 10.10.22.1
255.0.0.0

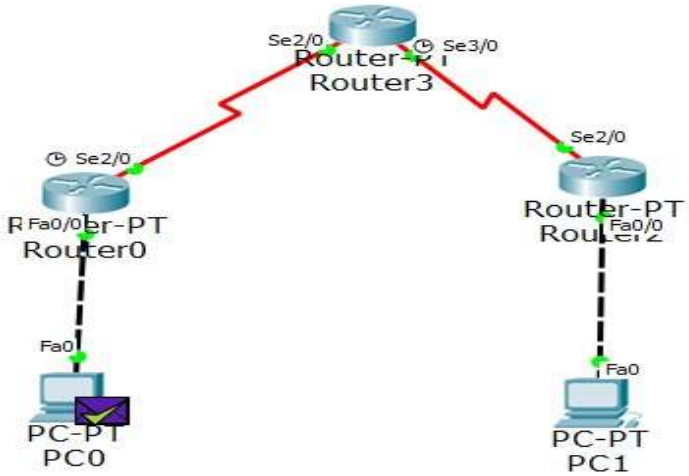
⑤ PC1 → Desktop
↳ IP config
↳ IP → 10.10.22.2
255.0.0.0
Gateway = 10.10.22.1

now send the packet and if it successfully completed.

Output : Status Successful.

Screenshots/ Output:





Observation:

- OSPF successfully established neighbour adjacencies and exchanged LSAs, allowing routers to build a synchronized link-state database across the OSPF area.

- The routing tables converged using SPF calculations, and successful pings confirmed efficient path selection and dynamic route learning through OSPF.

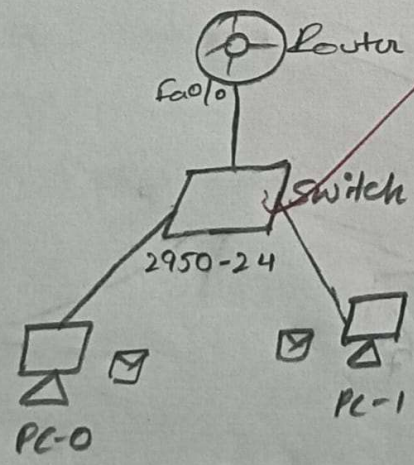
Program 11

Aim of the program:

Demonstrate the TTL/ Life of a Packet

Procedure and topology:

6(a) Demonstrate the TTL / Life of a packet.



PC-0 → IP Config → IP address : 192.168.1.2
Gateway : 192.168.1.1

PC-1 → IP address : 192.168.1.3
Gateway : 192.168.1.1

Router -> CLI -> interface fa0/0

IP address : 192.168.1.1 (same as Gateway)

Send packet from PC0 to PC-1

TTL is used to calculate time taken by msg packet to reach destination.

After sending packet click on msg packet

Auto-capture play => on clicking ✉

Inbound PDU

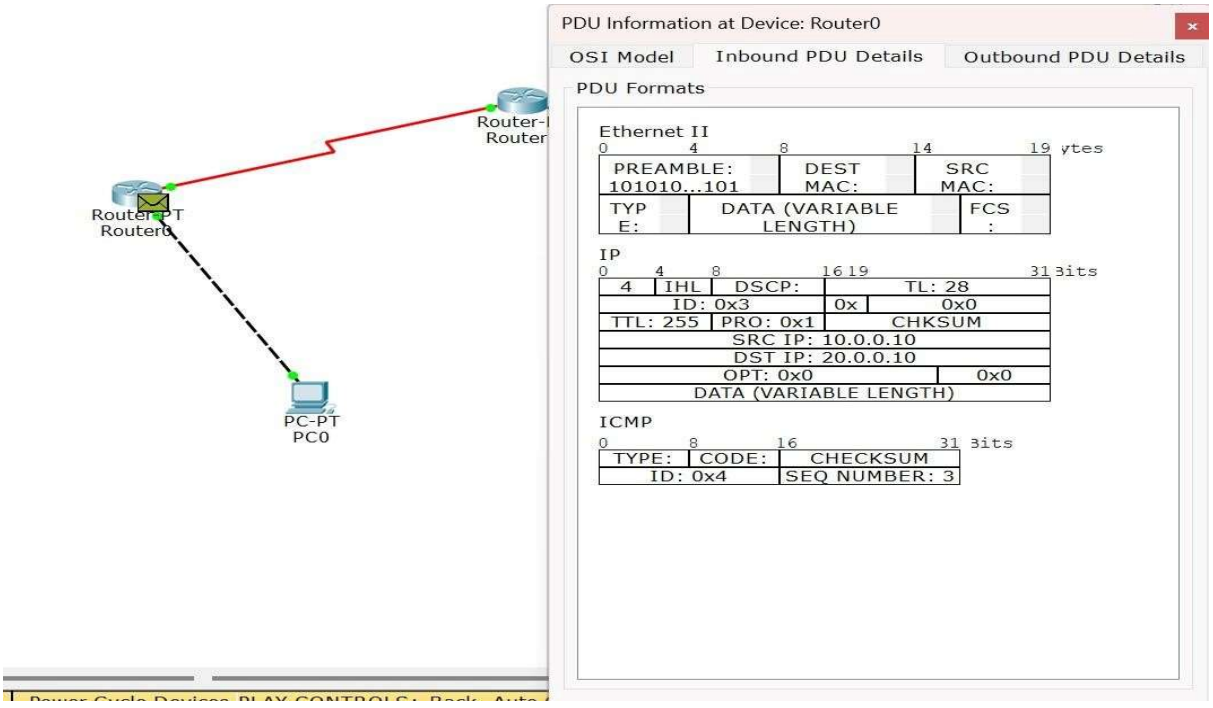
outbound PDU => To view Data

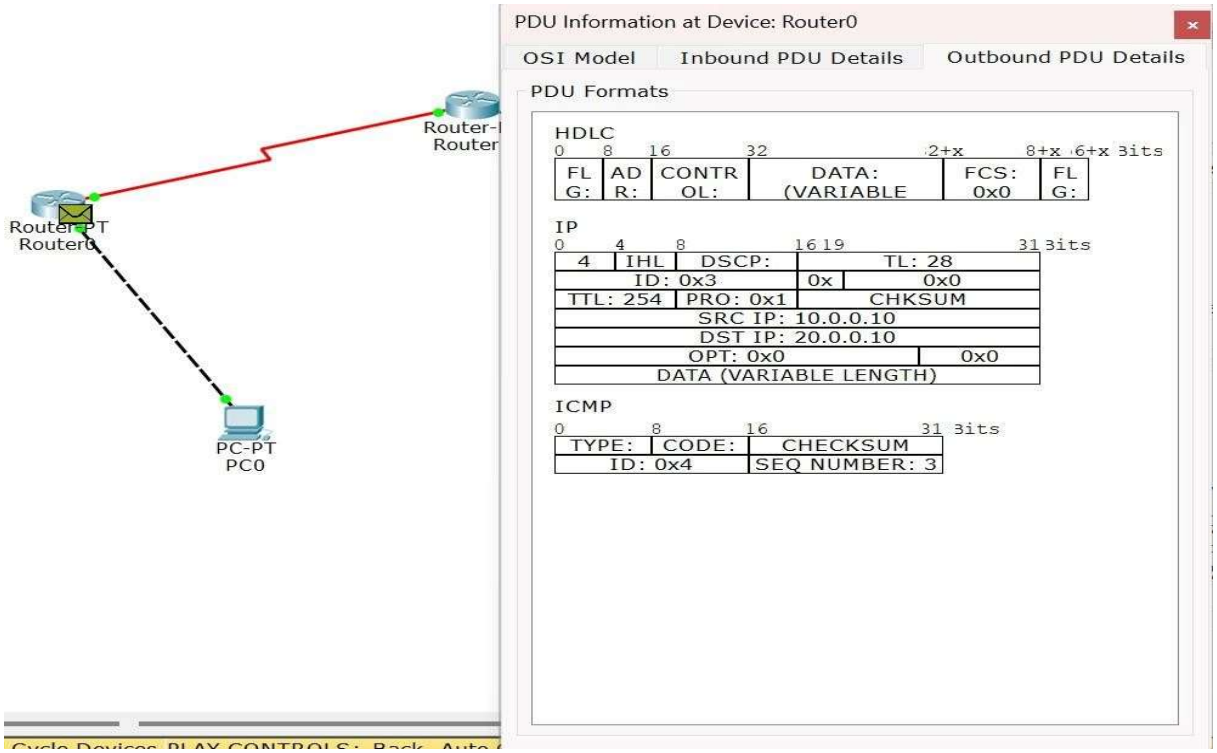
Variable length, checksum seq No :-

TTL : 255.

//.

Screenshots/ Output:





Observation:

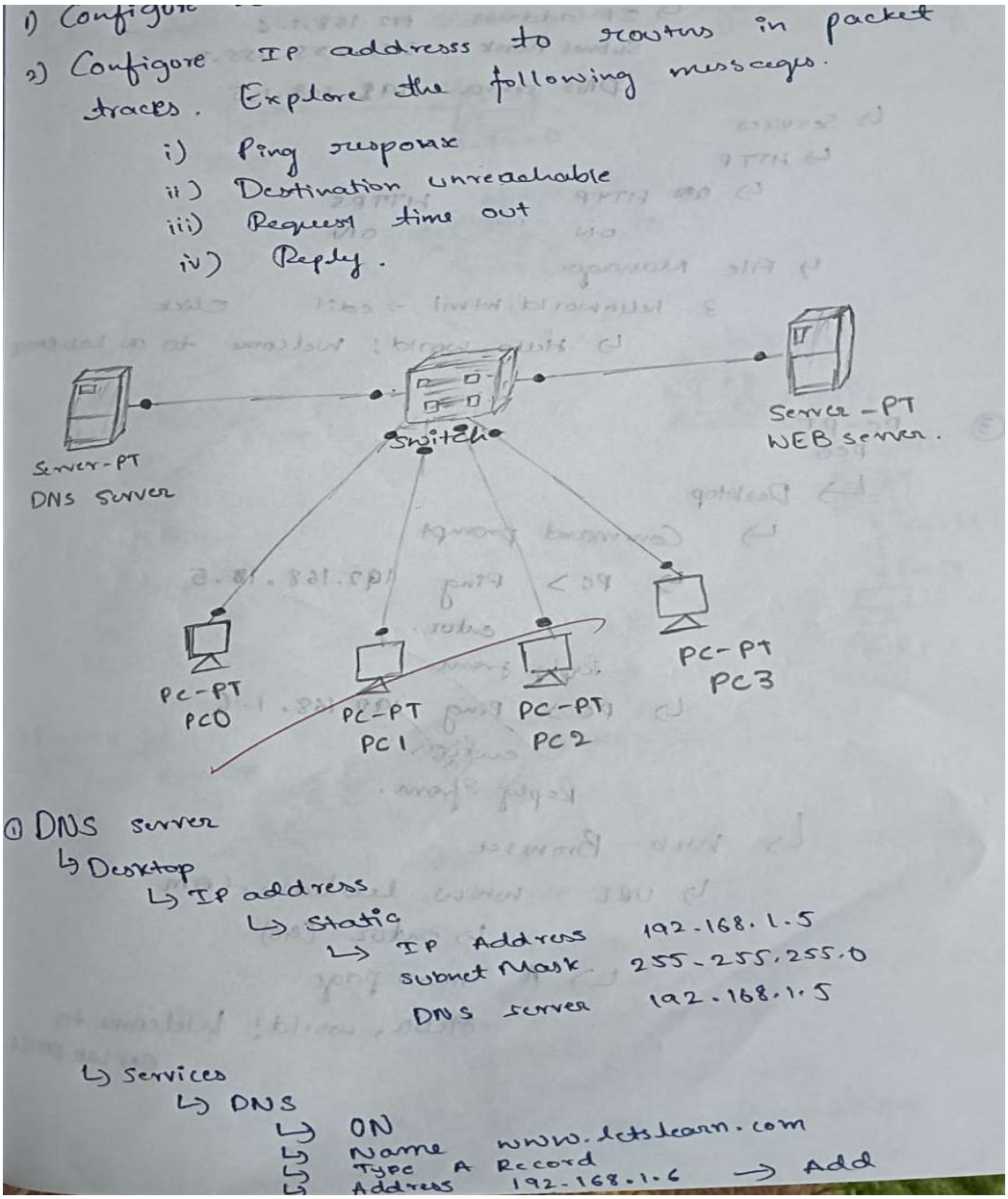
- The TTL field in the IP header decreased by one at each router hop, demonstrating its role in preventing packets from looping indefinitely in the network.

Program 12

Aim of the program:

Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply

Procedure and topology:



② WEB server.

↳ Desktop

↳ IP configuration

↳ static

↳ IP address 192.168.1.6

Subnet Mask 255.255.255.0

DNS server 192.168.1.5

↳ Services

↳ HTTP

↳ HTTP ON HTTPS ON

↳ File Manager.

3. helloworld.html → edit Click

↳ Hello, world! Welcome to CN Lab

③ PC-PT

PC0

↳ Desktop

↳ Command prompt

↳ PC > Ping 192.168.10.5

enter.

Reply from

↳ PC > Ping 192.168.1.6

enter

Reply from.

↳ Web Browser

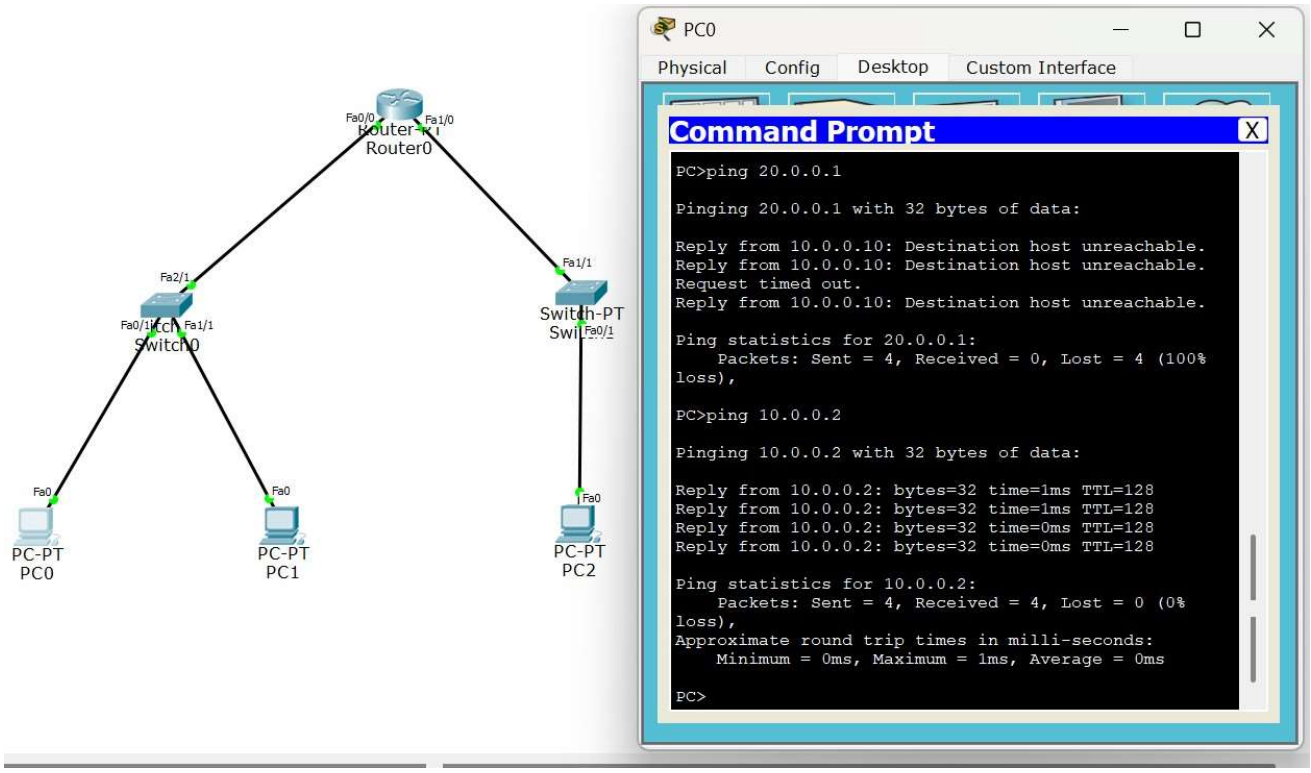
↳ URL www.letslearn.com

↳ enter (PC0)

↳ A small page

Hello, world! Welcome to CN Lab

Screenshots/ Output:



Observation:

- Proper IP addressing on router interfaces enabled successful ICMP echo and echo-reply communication, confirming correct Layer-3 configuration and reachability.
- “Destination Unreachable” and “Request Timed Out” responses occurred when routes or interfaces were misconfigured, demonstrating how routers handle missing paths and nonresponsive hosts.

CYCLE 2:

Program 13

Aim of the program:

Write a program for congestion control using Leaky bucket algorithm

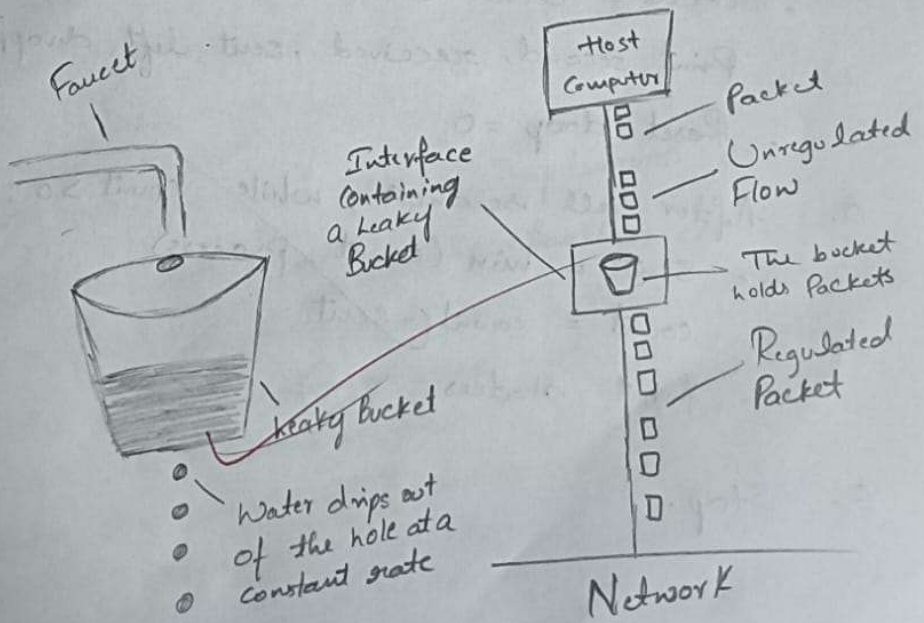
Procedure and topology:

29/10/2025

Lab-7

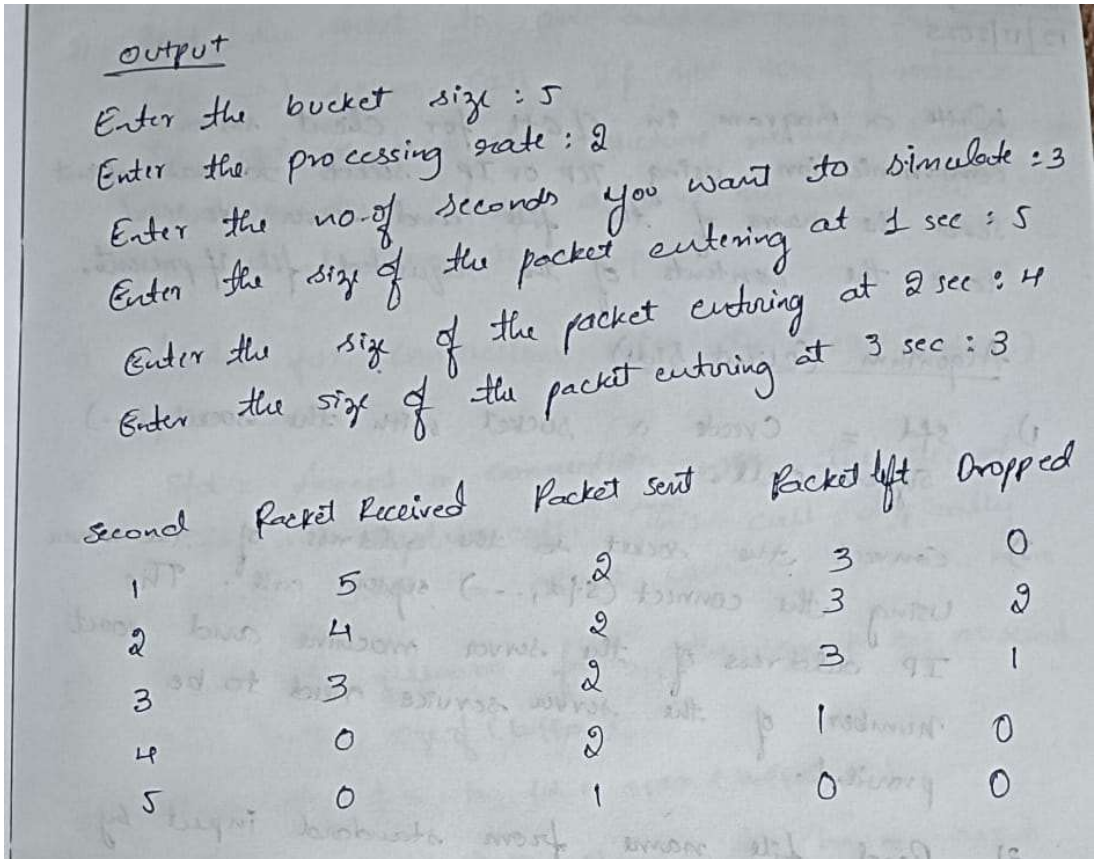
Write a Program in C/C++ for congestion control using leaky Bucket algorithm.

- The leaky Bucket algorithm is a traffic shaping technique used in computer networks to control data flow and prevent congestion. It allows packets to enter a bucket (buffer) and transmits them at a fixed, steady rate, regardless of how bursty the incoming traffic is.
- If the bucket becomes full, any extra packets are dropped. This ensures a smooth output rate and helps maintain network performance and reliability.



Algorithm

1. Start
2. Input
   - Bucket capacity (cap)
   - Output rate (process)
   - Number of seconds (nsec)
   - Packets arriving each second (inp[i])
3. Initialize count = 0, drop = 0
4. For each second:
   - count = count + inp[i]
   - If count > cap:
     - → drop = count - cap
     - → count = cap
   - Sent = min (count, process)
   - count = count - sent
   - Print second, received, sent, left, dropped
   - Reset drop = 0
5. After all seconds, while count > 0:
   - Sent = min (count, process)
   - count = count - sent
   - Print status
6. Stop.



output

Enter the bucket size : 5
Enter the processing rate : 2
Enter the no. of seconds you want to simulate : 3
Enter the size of the packet entering at 1 sec : 5
Enter the size of the packet entering at 2 sec : 4
Enter the size of the packet entering at 3 sec : 3

Second	Packet Received	Packet sent	Packet left	Dropped
1	5	2	3	0
2	4	2	3	2
3	3	2	3	1
4	0	2	1	0
5	0	1	0	0

Screenshots/ Output:

Output

```
Enter bucket size: 4
Enter outgoing size: 1
Enter number of inputs: 7
Enter the incoming packet size: 3
Bucket buffer size 3 out of 4
After outgoing 2 packets left out of 4 in buffer
Enter the incoming packet size: 2
Bucket buffer size 4 out of 4
After outgoing 3 packets left out of 4 in buffer
Enter the incoming packet size: 1
Bucket buffer size 4 out of 4
After outgoing 3 packets left out of 4 in buffer
Enter the incoming packet size: 4
Dropped 3 packets
Bucket buffer size 4 out of 4
After outgoing 3 packets left out of 4 in buffer
Enter the incoming packet size: 0
Bucket buffer size 3 out of 4
After outgoing 2 packets left out of 4 in buffer
Enter the incoming packet size: 0
Bucket buffer size 2 out of 4
After outgoing 1 packets left out of 4 in buffer
Enter the incoming packet size: 0
Bucket buffer size 1 out of 4
After outgoing 0 packets left out of 4 in buffer
```

Observation:

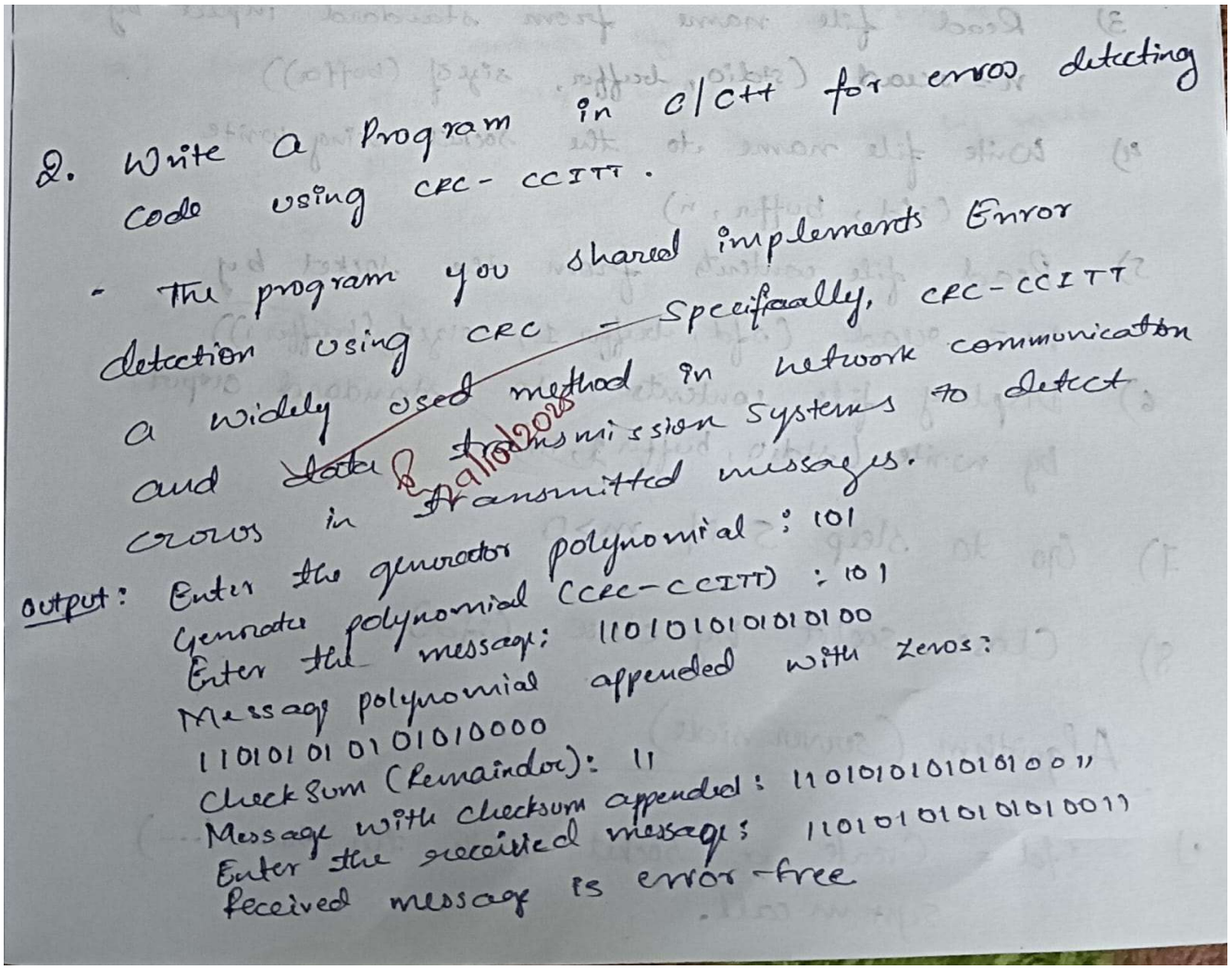
- The leaky bucket mechanism regulated the outgoing packet flow at a constant rate, preventing sudden traffic bursts from overwhelming the network.
- Packets exceeding the bucket capacity were dropped, demonstrating effective congestion control by smoothing traffic and enforcing rate-limiting.

Program 14

Aim of the program:

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

Procedure and topology:



Screenshots/ Output:

Observation:

- The CRC-CCITT (16-bit) algorithm correctly generated a checksum for the transmitted data, ensuring reliable detection of single-bit and burst errors.
- Intentional error tests produced mismatched CRC values at the receiver, confirming accurate error detection through polynomial division.

Program 15

Aim of the program:

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.

Procedure and topology:

Wednesday
12/11/2025

Write a Program in C/C++ for client server communication using TCP or IP sockets to make client send the name of the file and server to send back the contents of the requested file if present.

Algorithm (Client Side)

1) sfd = Create a socket with the socket(....) System call.
2) Connect the socket to the address of the server using the connect(sfd,---) system call. The IP address of the server machine and port number of the server service need to be provided.
3) Read file name from standard input by n = read (stdio, buffer, sizeof(buffer))
4) Write file name to the socket using write (sfd, buffer, n)
5) Read file contents from the socket by m = read (sfd, buffer1, sizeof(buffer1))
6) Display file contents to standard output by write (stdio, buffer1, m)
7) Go to sleep 5 if m>0
8) Close socket by close (sfd).

Algorithm (Server side)

1) sfd = Create a socket with the socket(....) System call.

2) Bind the socket to an address using the bind (sfd, ---) system call. If not sure of machine IP address, keep the structure member s_addr to INADDR_ANY. Assign a port number b/w 3000 and 5000 to sin_port.

3) listen for connections with the listen (sfd, ...) system call.

4) sfd = Accept a connection with the accept (sfd, ...) system call. This call typically blocks until a client connects with the server.

5) Read the filename from the socket by n = recvd (sfd, buffer, sizeof (buffer))

6) Open the file by fd = open (buffer)

7) Read the contents of the file by m = recvd (fd, buffer1, sizeof (buffer1))

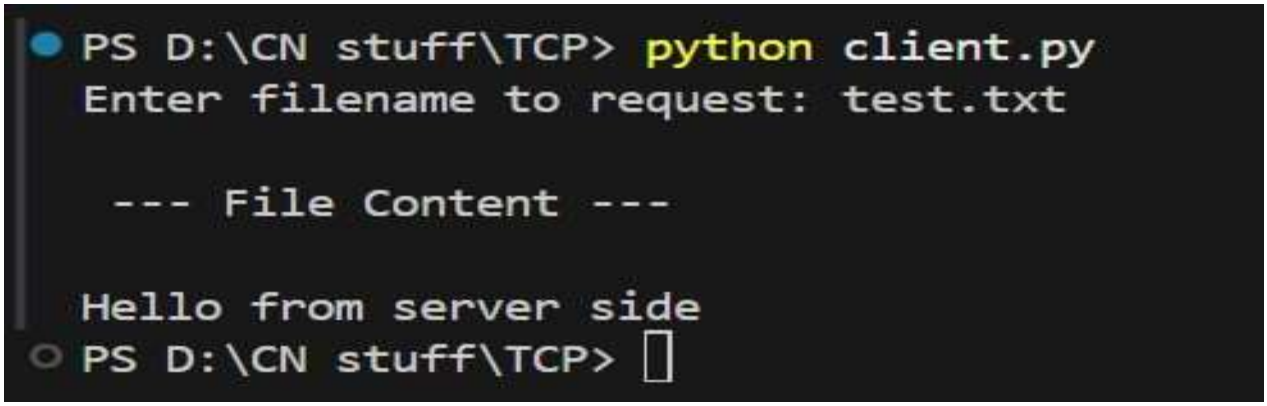
8) Write the file content to socket by write (sfd, buffer1, m))

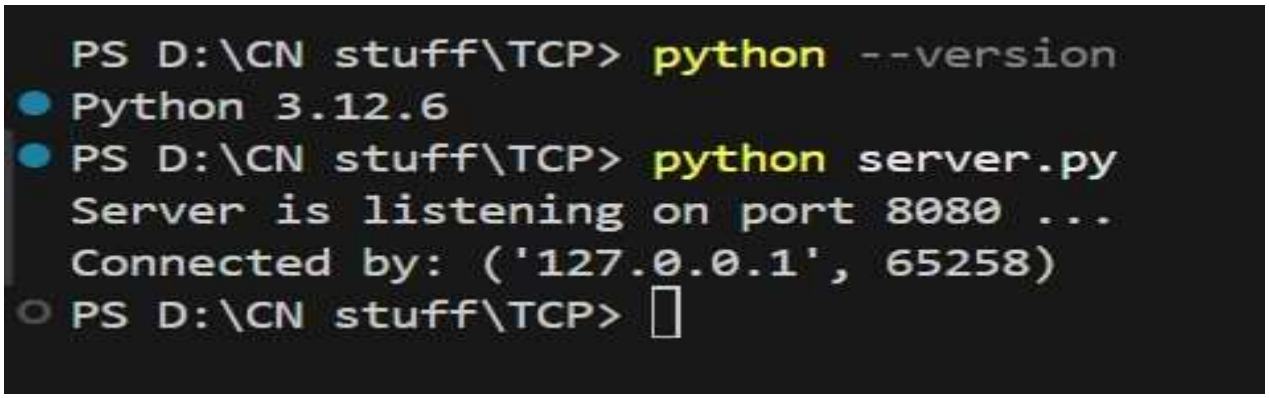
9) Go to step 7 if m > 0

10) close (sfd)

12/11/2025

Screenshots/ Output:





Observation:

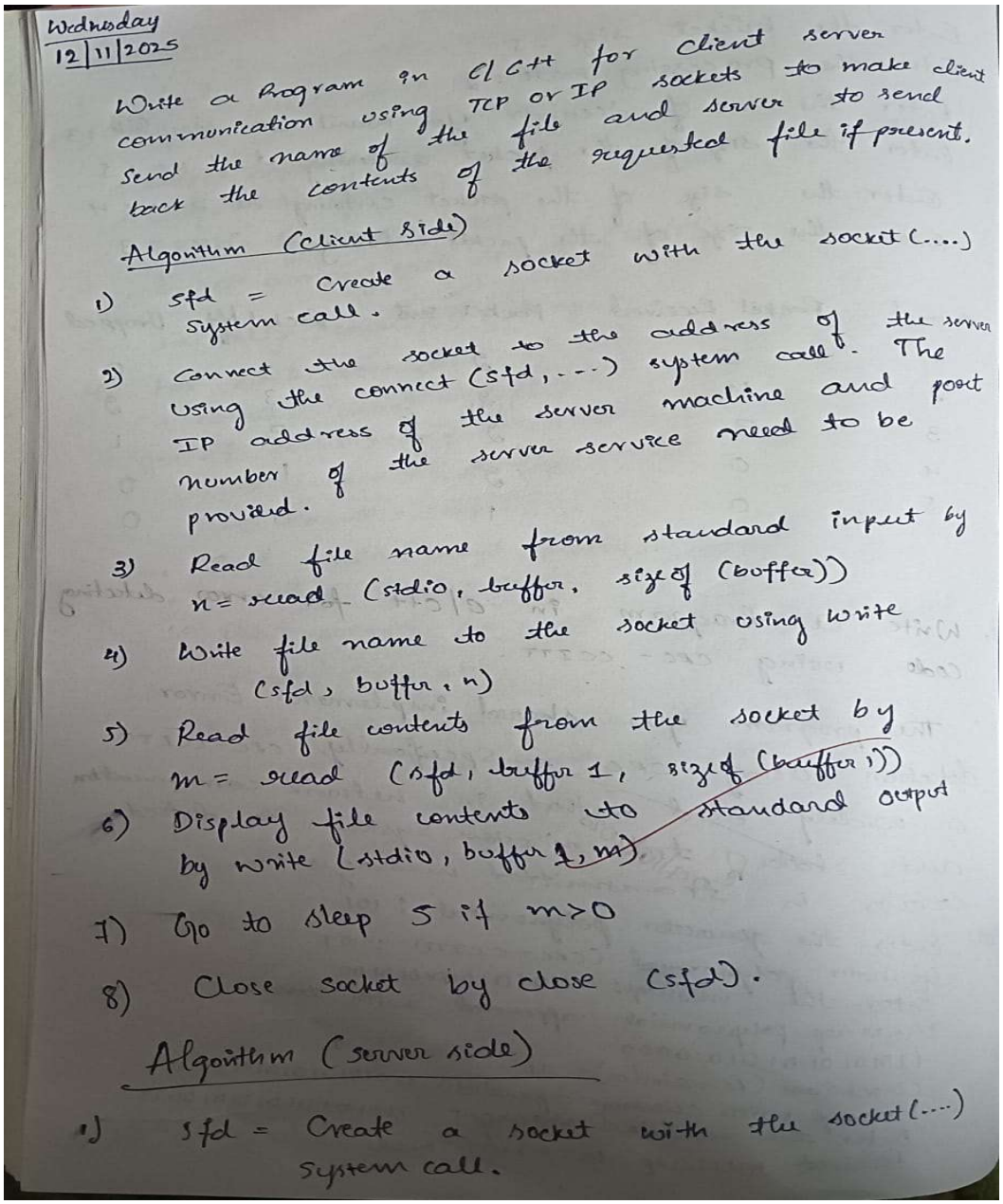
- The TCP client successfully established a reliable connection with the server and transmitted the requested filename using stream-oriented communication.
- The server correctly retrieved and returned the file contents over the same TCP session, demonstrating reliable data transfer, acknowledgment handling, and error-free delivery.

Program 16

Aim of the program:

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.

Procedure and topology:



Wednesday
12/11/2025

Write a Program in C/C++ for client server communication using TCP or IP sockets to make client send the name of the file and server to send back the contents of the requested file if present.

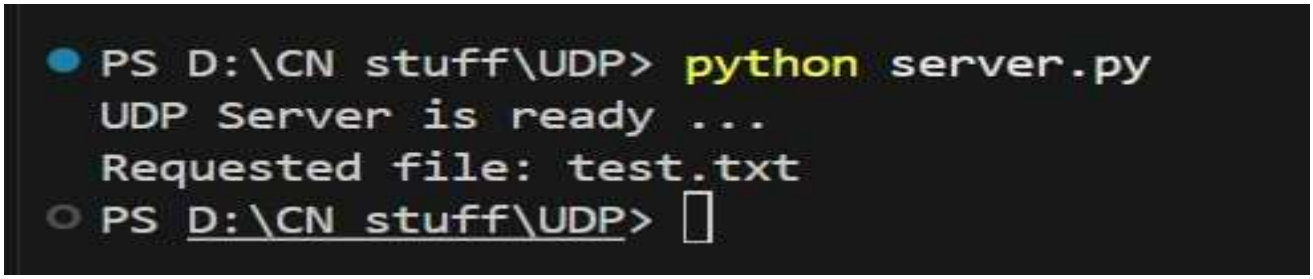
Algorithm (Client Side)

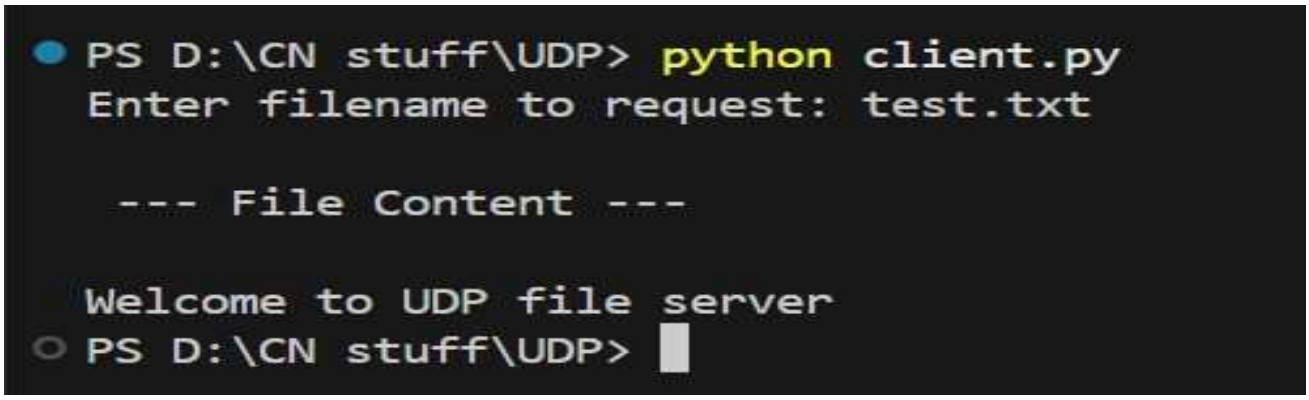
1) sfd = Create a socket with the socket(....) system call.
2) Connect the socket to the address of the server using the connect(sfd,...) system call. The IP address of the server machine and port number of the server service need to be provided.
3) Read file name from standard input by n = read (stdio, buffer, sizeof(buffer))
4) Write file name to the socket using write (sfd, buffer, n)
5) Read file contents from the socket by m = read (sfd, buffer 1, sizeof(buffer1))
6) Display file contents to standard output by write (stdio, buffer 1, m)
7) Go to sleep 5 if m>0
8) Close socket by close (sfd).

Algorithm (Server side)

1) sfd = Create a socket with the socket(....) system call.

Screenshots/ Output:





Observation:

- The UDP client successfully sent the filename as a connectionless datagram, demonstrating non-reliable, low-overhead message transfer without session establishment.
- The server responded with the file contents using UDP packets, and correct reception validated functional data exchange despite the absence of acknowledgment and retransmission mechanisms.