VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB RECORD

Computer Network Lab (23CS5PCCON)

Submitted by

Jayashree Tarai (1BM24CS407)

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 **Jayashree Tarai(1BM24CS407),** 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

Github Link https://github.com/Jayashreecse/CNLAB

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:

20/8/25

Experiment -01

Create a topology & simulate Sending a Simple PDU from source to Destination Using hub & switch
Demonstrate the Ping message

Star topology Using Switch

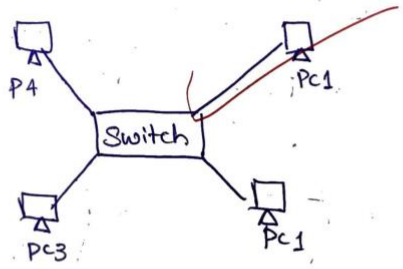
Components - switch, 4 devices

Implementation

Step1: Take a Switch and 4 wired end devices (PC)

Step2: Add the Connection wire (Automatic)

Step3: Ping the message by going to the Source computer in that open command prompt and ping the destination IP address and get the details whether the packets are received or not



Disadvantages

1) If even one hub goes down, everything goes down

If network device fails, all connected devices fail

→ Mesh topology (here every network node is connected to every other device)

Components : 4 switches, 4 PC

Implementation

Step1 : Take 4 Switches & 4 wired end device (PC)

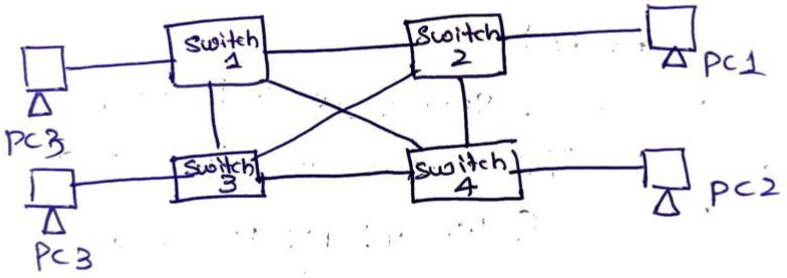
Step2 : Connect end devices with switch through cable.

Step3 : Assign IP address to each PC

Step4 : Test the network configuration using pinging

ping 192.168.30.2 & get the

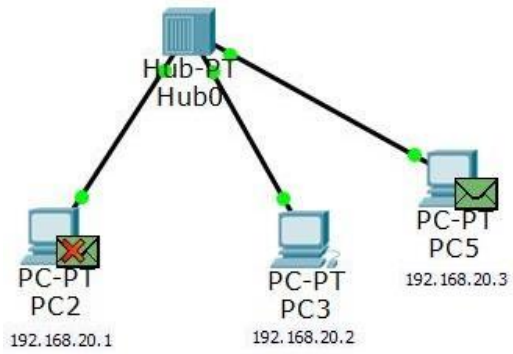
Details whether the messages (packets) har received or not

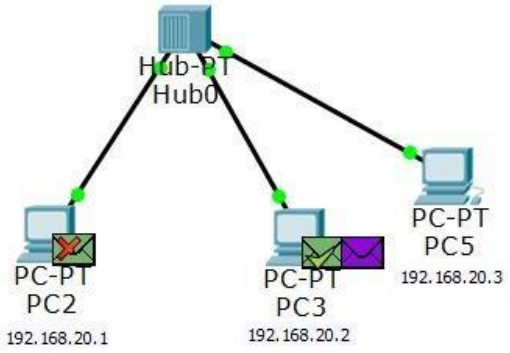


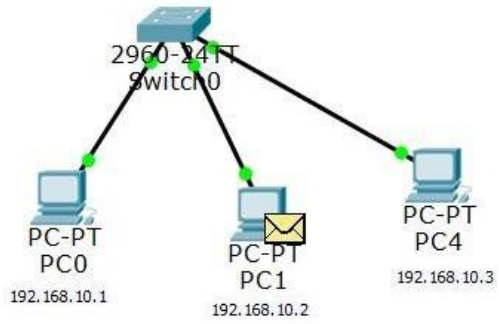
Advantages

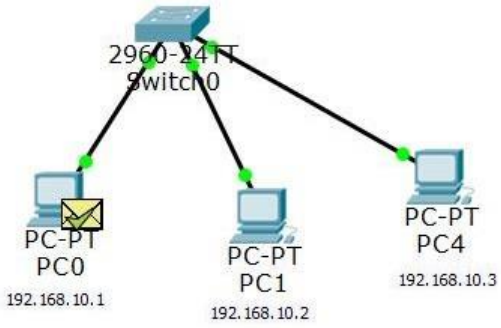
→ No traffic problem due to one to one connection with each device

Screenshots/ Output:

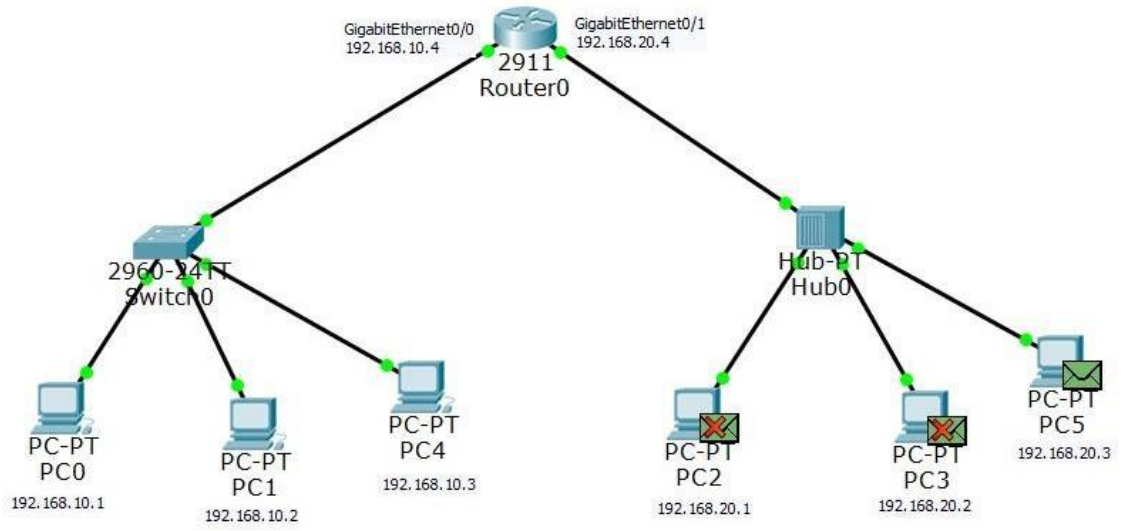


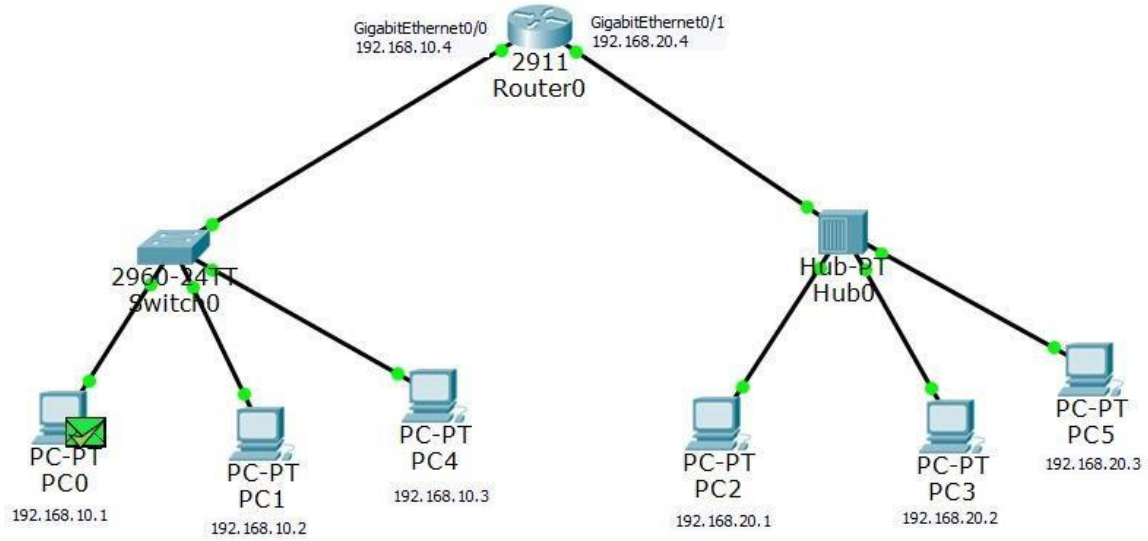






Updated topology





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

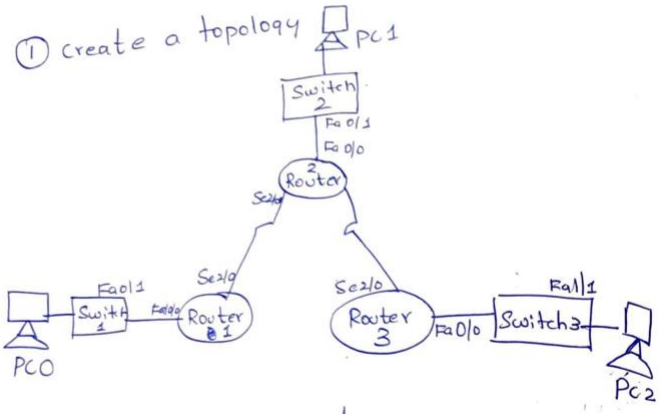
Procedure and topology:

Wed 17 sept

Lab 4

Configure IPV4 static and Default routing

① create a topology



② Do the Router Configuration (two configuration → one for switch & Router)

click on Router 1 → Go to → CLI

→ no
→ enable
→ conf t
→ # int se2/0 (Interface towards Switch / Router)
→ ip address 172.16.1.1 255.255.255.252 enter
→ no shutdown
→ exit → wr (Now towards Router is done) see Building configur. Successfull

Now do towards Switch

→ int Fa0/0
→ Ip address 192.168.10.1 255.255.255.0
→ no shutdown
→ exit
→ Do write memory or type wr
→ See the Succesful msg of Building configuration

Now next is Router 2 Configuration
We have 3 connection to Router 2 of 2 Router and one switch

Open CLI
→ NO → Enable
→ conf t → hostname R2 (change to R2)
→ int se2/0 (towards R1)
→ Ip address 172.16.1.2 255.255.255.252
→ no shutdown
→ exit
→ See the Building Configuration msg.

Next towards Switch

→ int Fa 0/0
→ ip address 192.168.20.1 255.255.255.0
→ no shutdown
→ exit → wr
→ See the msg Building Configuration

Next towards Router3

→ int se3/0
→ Ip address 172.16.2.1 255.255.255.252
→ no shutdown → exit
→ Do write memory
→ see the Building config msg.

Now next is Router 3 Configuration (2 connectio

CLI
→ no → Enable
→ conf t
→ # hostname r3
→ # int se2/0
→ ipaddress 172.16.2.2 255.255.255.252
→ no → shutdown → exit → Do write memory
Building Config

→ ip address 192.169.30.1 255.255.255.0
→ no shutdown
→ exit → wr → Building Config

Click on PC0
↳ Desk
↳ IP
↓
IPV4 192.168.10.10
Gateway 192.168.10.1

Click on PC1
↳ Desk
↳ IP
↓
IPV4
192.168.20.10
Gateway
192.168.20.1

click on PC3
↳ Desk
↳ IP
↓
IPV4 192.168.30.10
Gateway 192.168.30.1

Click on Router R1
R1 # conf t
→ 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 192.168.30.0 255.255.255.0 172.16.3.2
→ Exit → wr

Click on R3
conf t
→ ip route 0.0.0.0 0.0.0.0 s0/0/1
Exit → wr

To see Routing table
click on R1
show IP route

Check the same way for others

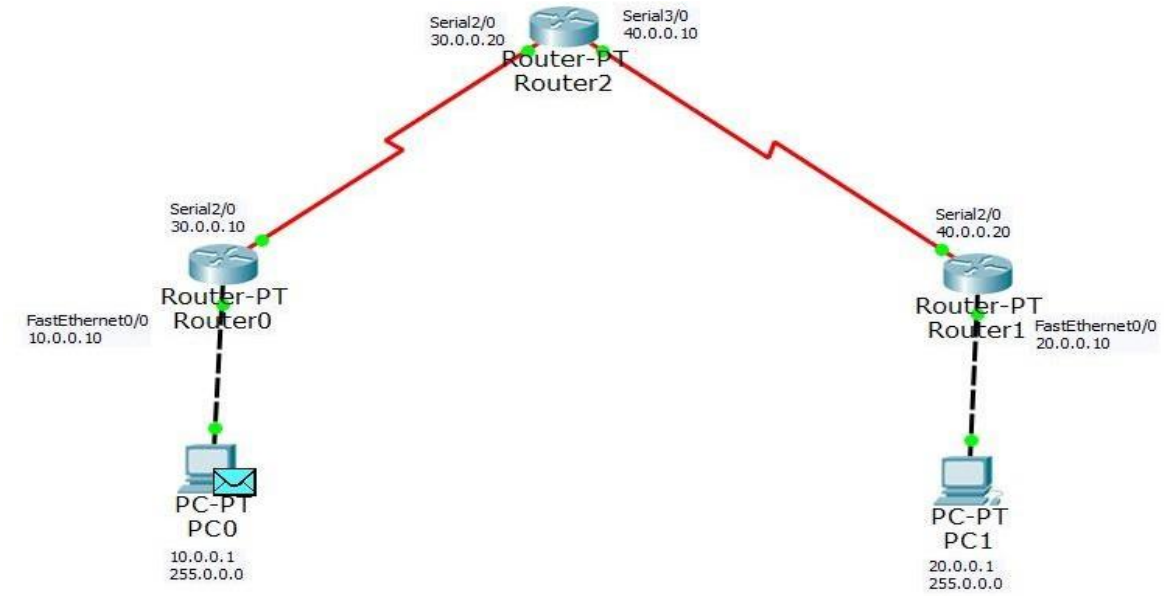
click on PC0
↳ Desktop
↳ command prompt
↳ ping 192.168.10.1
Ping 192.168.20.1
Ping 192.168.30.1

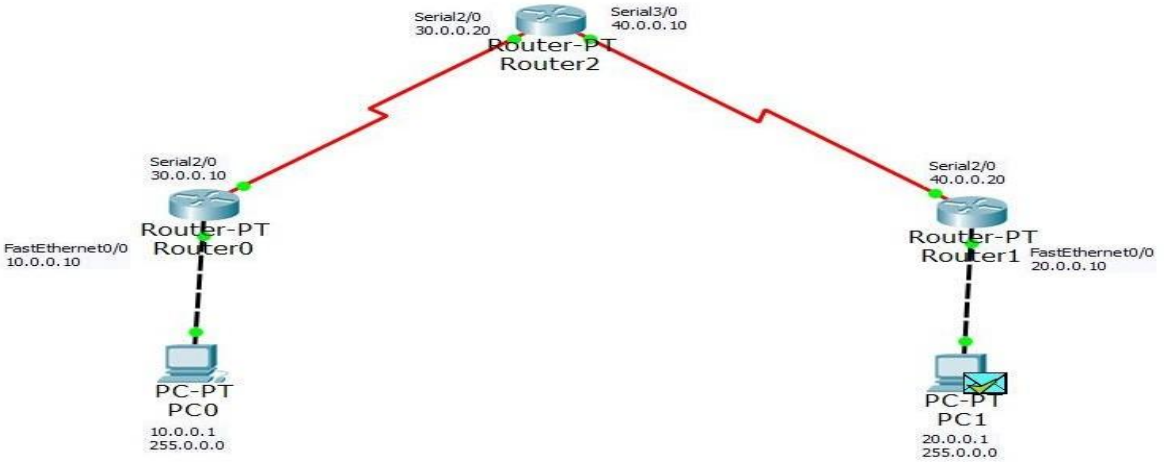
then send the PDU from PC0 to PC1
PC2 to PC3

See the message

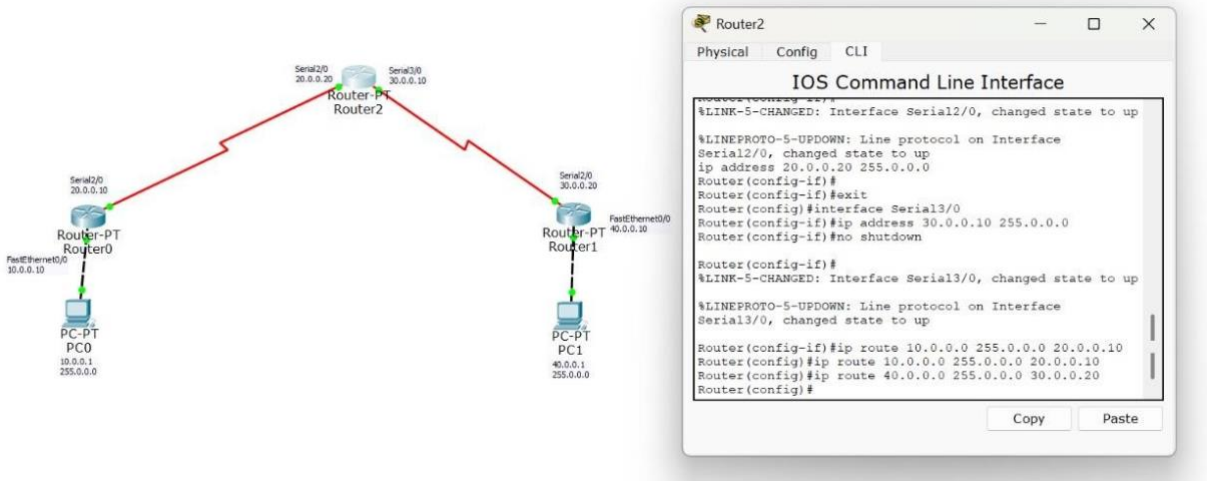
17/9/2022

Screenshots/ Output:

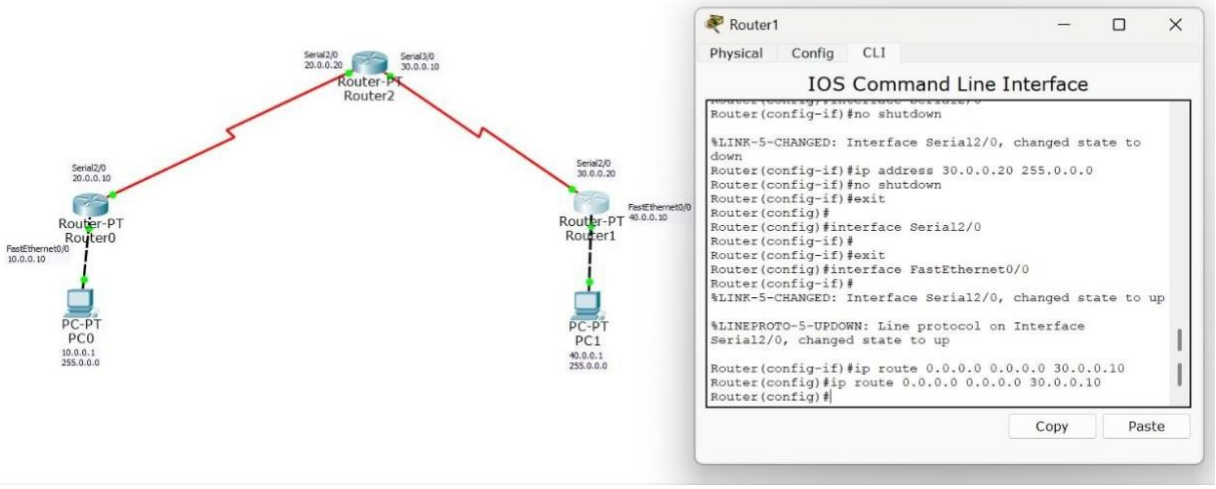




Static routing CLI commands:



Default routing CLI commands:



Observation:

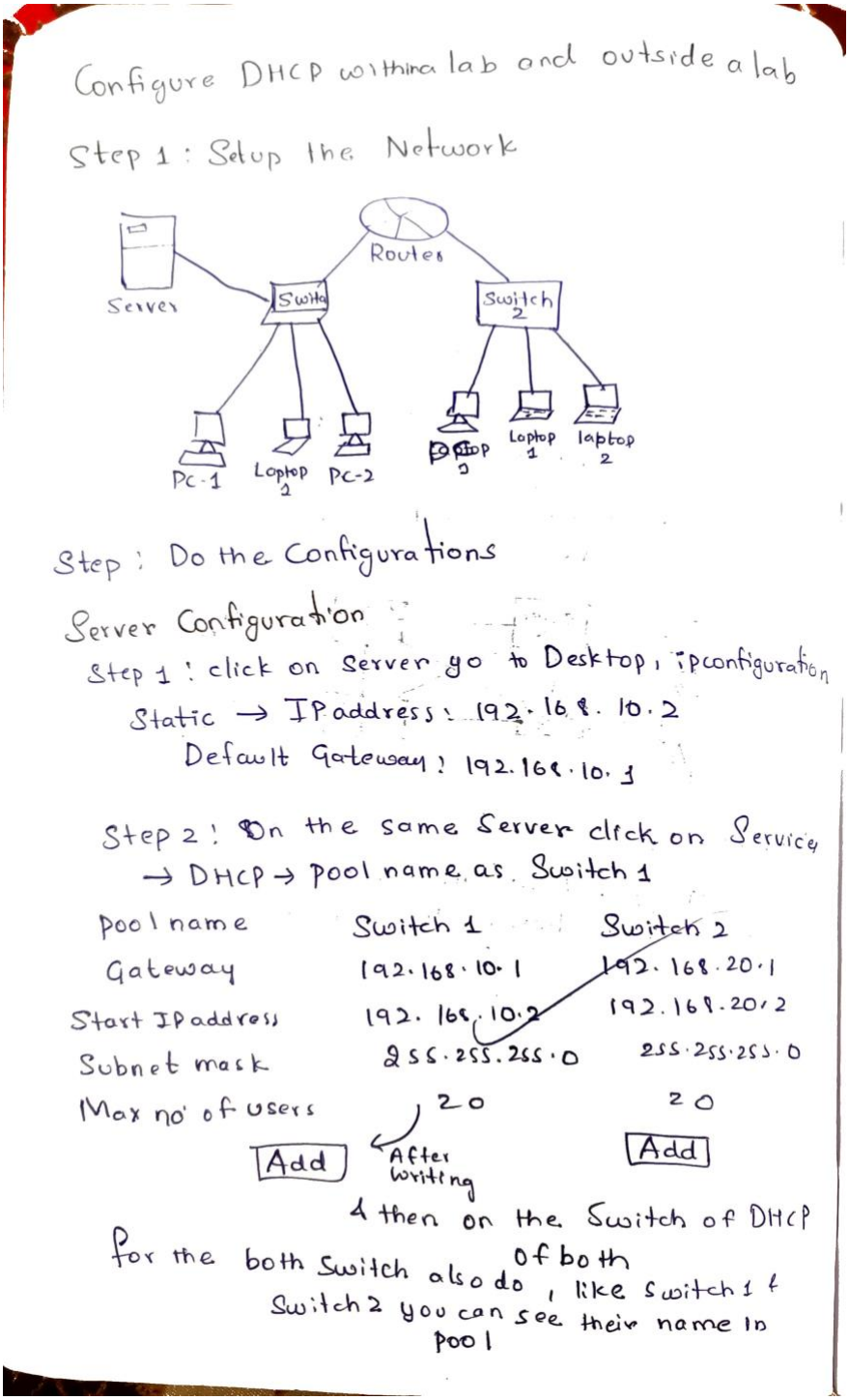
- The configured static and default routes correctly updated the router's routing table, enabling deterministic next-hop selection for remote networks.
- Successful ping tests verified that traffic was forwarded according to the static/default route entries, ensuring end-to-end reachability across different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

Program 3

Aim of the program:

Configure DHCP within a LAN and outside LAN.

Procedure and topology:



Router Configuration

click on CLI

Type NO

Then you can see Router>

step2: type the enable Next

Conf t

int Fa 0/0 (for this once check with the Router green dots then you can see) the eth 0/1 1/0

→ # ip address 192.168.10.1 255.255.255.0

→ # ip helper-address 192.168.10.2

→ # no shutdown

→ do write memory

→ # Exit

Repeat the same

int Fa0/1 or Fa 1/0

ip address 192.168.20.1 255.255.255.0

ip helper-address 192.168.10.2

no shutdown

do write memory

exit

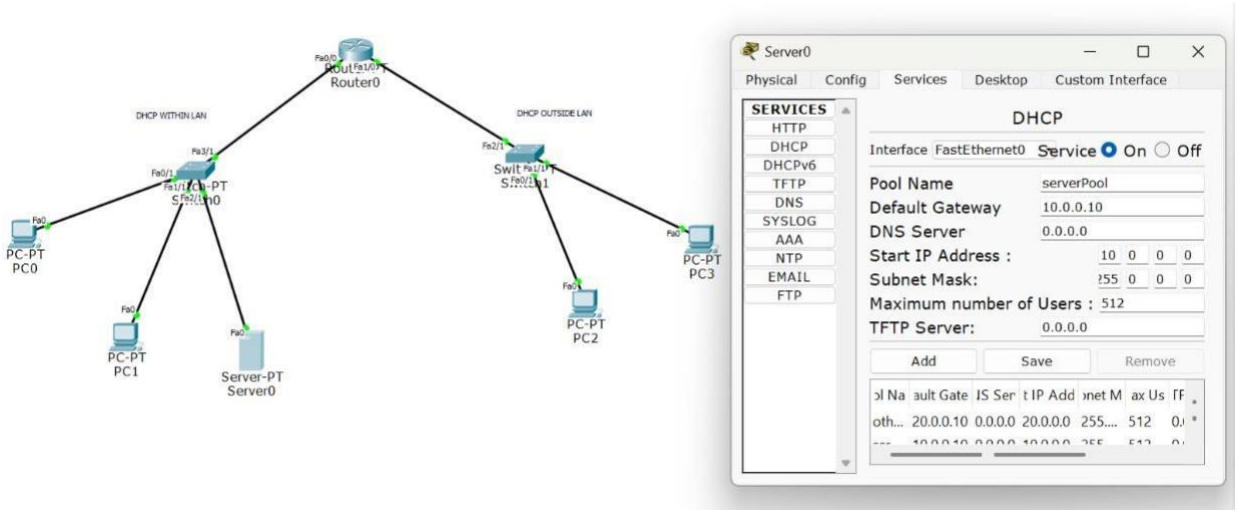
Then to check it it is working or not

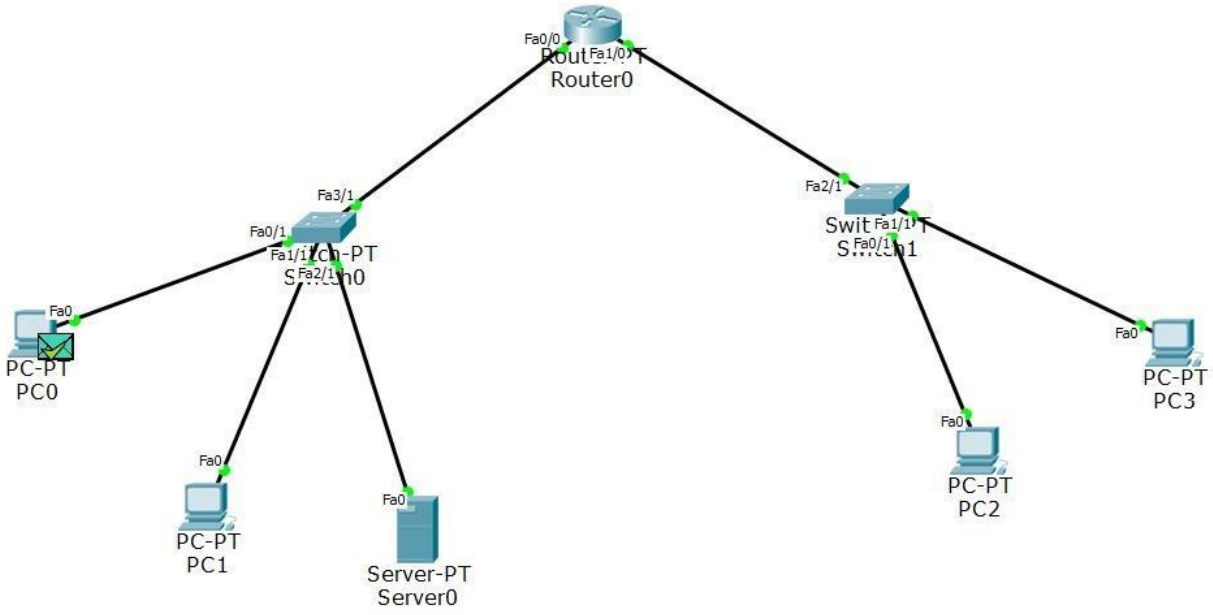
go to any of laptop or ~~screenshot~~ PC

PC→ Desktop → IP configuration → DHCP

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:

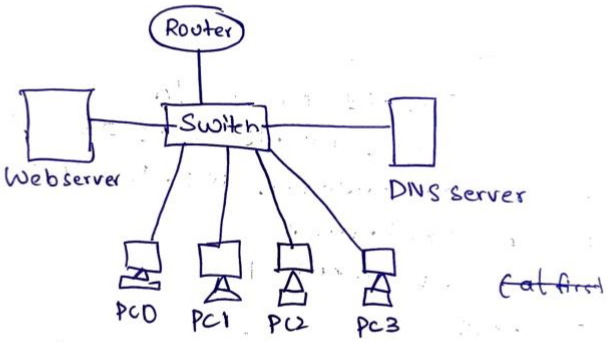
Configure Web Server, DNS within a LAN.

Procedure and topology:

Lab-3

1) Configure Web Server, DNS within a LAN
2) Configure IP Addresses to Router in Packet Tracer. Explore the following messages
   i) Ping Response (IP settings are correct, Devices are connected route exits
   ii) Destination Unreachable (Device does not know how to reach destination gateway/ Router is not set
   iii) Request Time out (device is there but not responding in time msg sent, no reply
   iv) Reply (connected showing Reply msg

1) Create a topology Using the Devices



2) click on web server & click on services
   → Select http & HTTPS should be on
   → Select anyone file.htm in the file manager
   Click on file & click on edit and add some new message & save the file
   → In that only Go to Desktop & Ipaddress (IP config)
   Ip address : 192.168.1.6
   Subnet mask : 255.255.255.0
   DNS server : 192.168.1.5

3) DNS Server

→ click on Services → select DNS → should be on

→ Give → Name : www.letslearn.com
Type : A Record
Address : 192.168.1.6 & click on Add

In that only Go go to Desktop → Ip config
IP Add : 192.168.1.5
Subnet : 255.255.255.0
DNS : 192.168.1.5

4) Go to PC 0 click on that select or click on IP configuration

Ip add : 192.168.1.101
Subnet : 255.255.255.0
DNS : 192.168.1.5

Same for PC 2 & PC 3

After that to check the Connection Established Succesfully

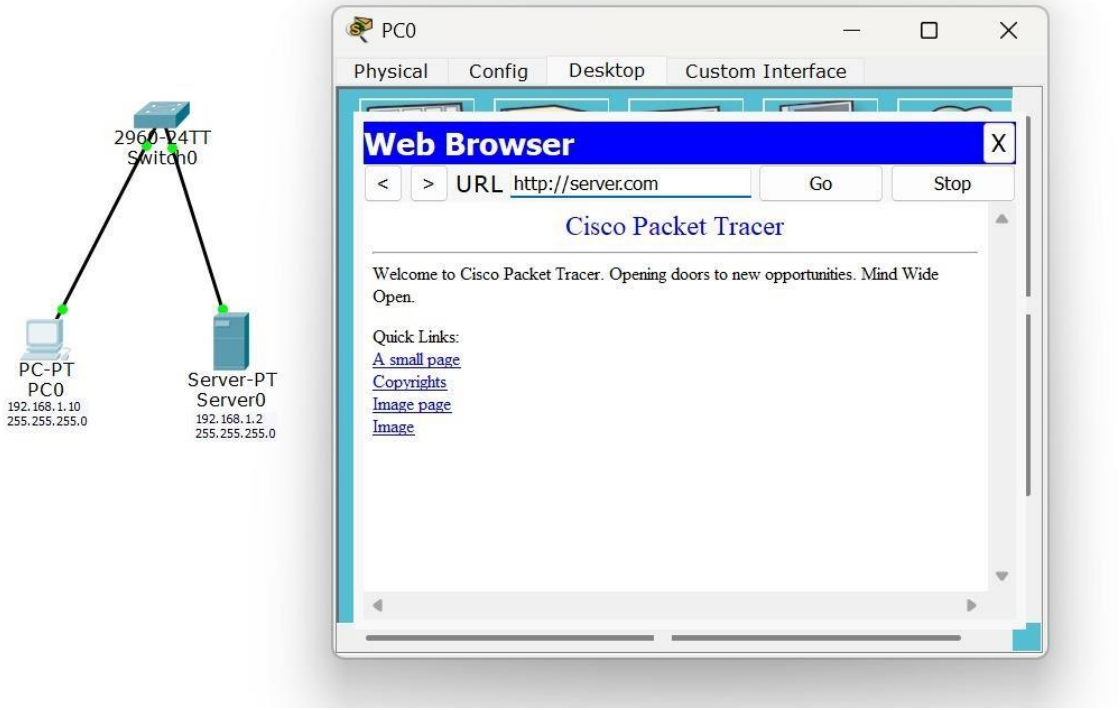
Go to PC 1 and click on cmd :

Ping 192.168.1.5
output : message sent Replaying....
Ping 192.168.1.6
Output Replaying....

Then 1090 to web Browser and type the website name www.letslearn.com then the page will come

click on → A small page
There you can Successful see the message

Screenshots/ Output:



Observation:

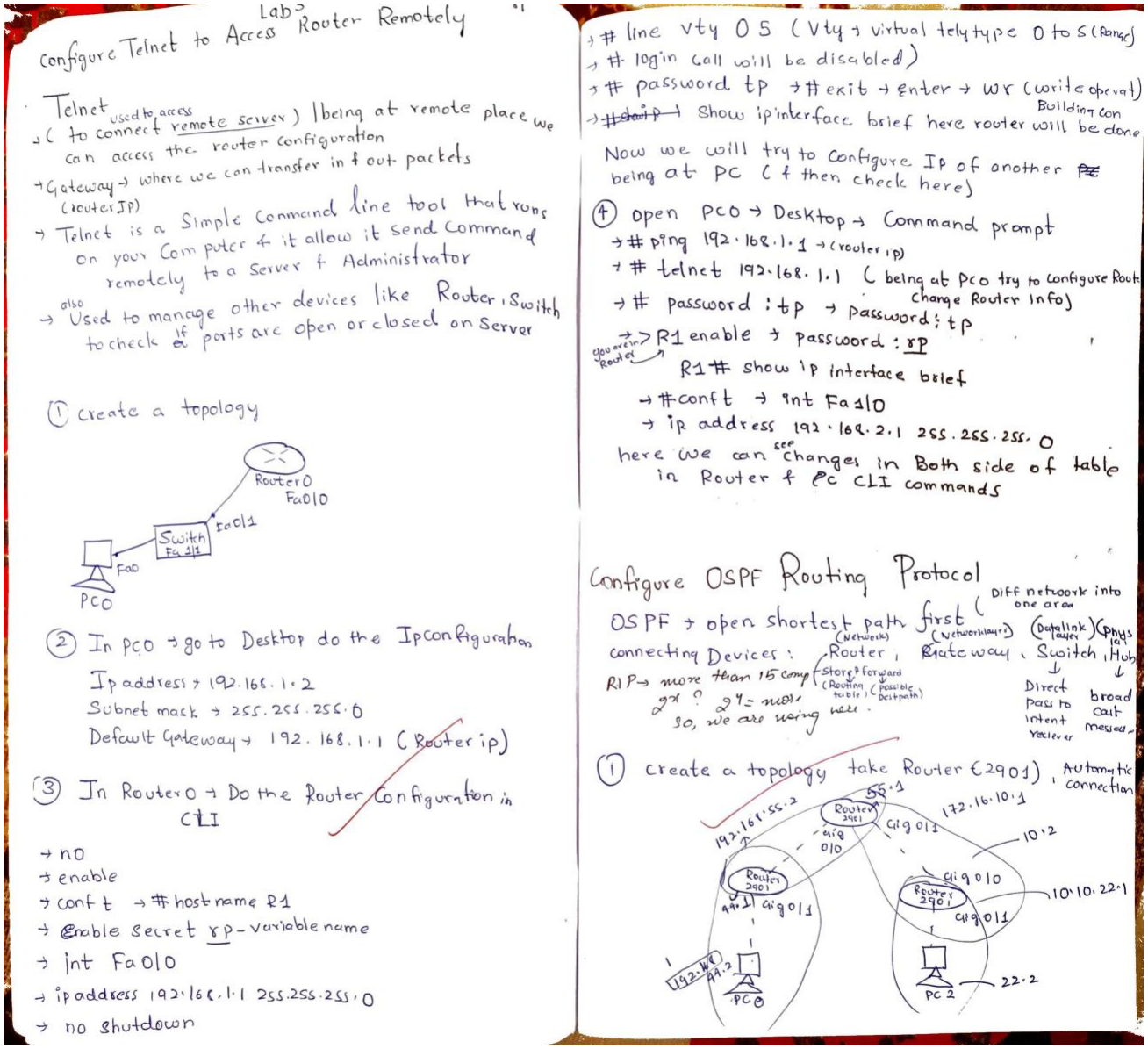
- The DNS server successfully resolved domain names to the corresponding web server's IP address, confirming proper hostname-to-IP mapping within the LAN.
- HTTP requests reached the web server using the DNS-resolved address, validating correct server configuration and internal LAN communication.

Program 5

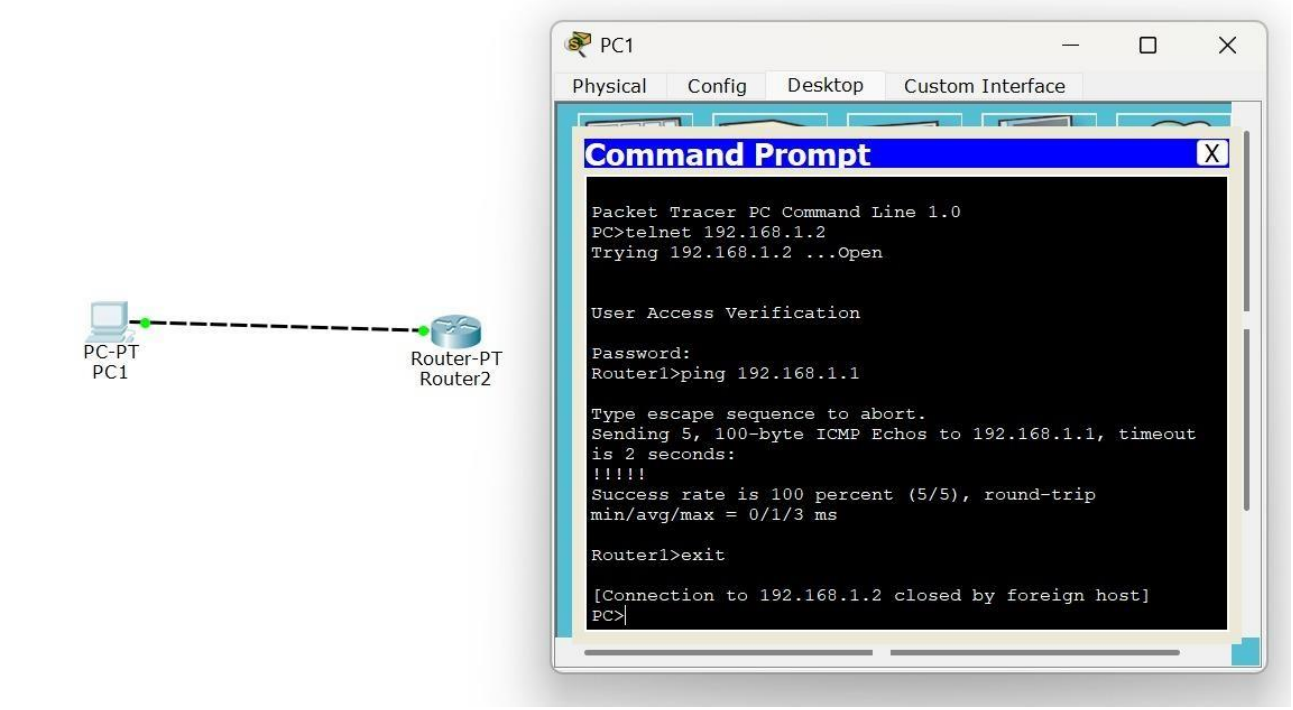
Aim of the program:

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

Procedure and topology:



Screenshots/ Output:



Observation:

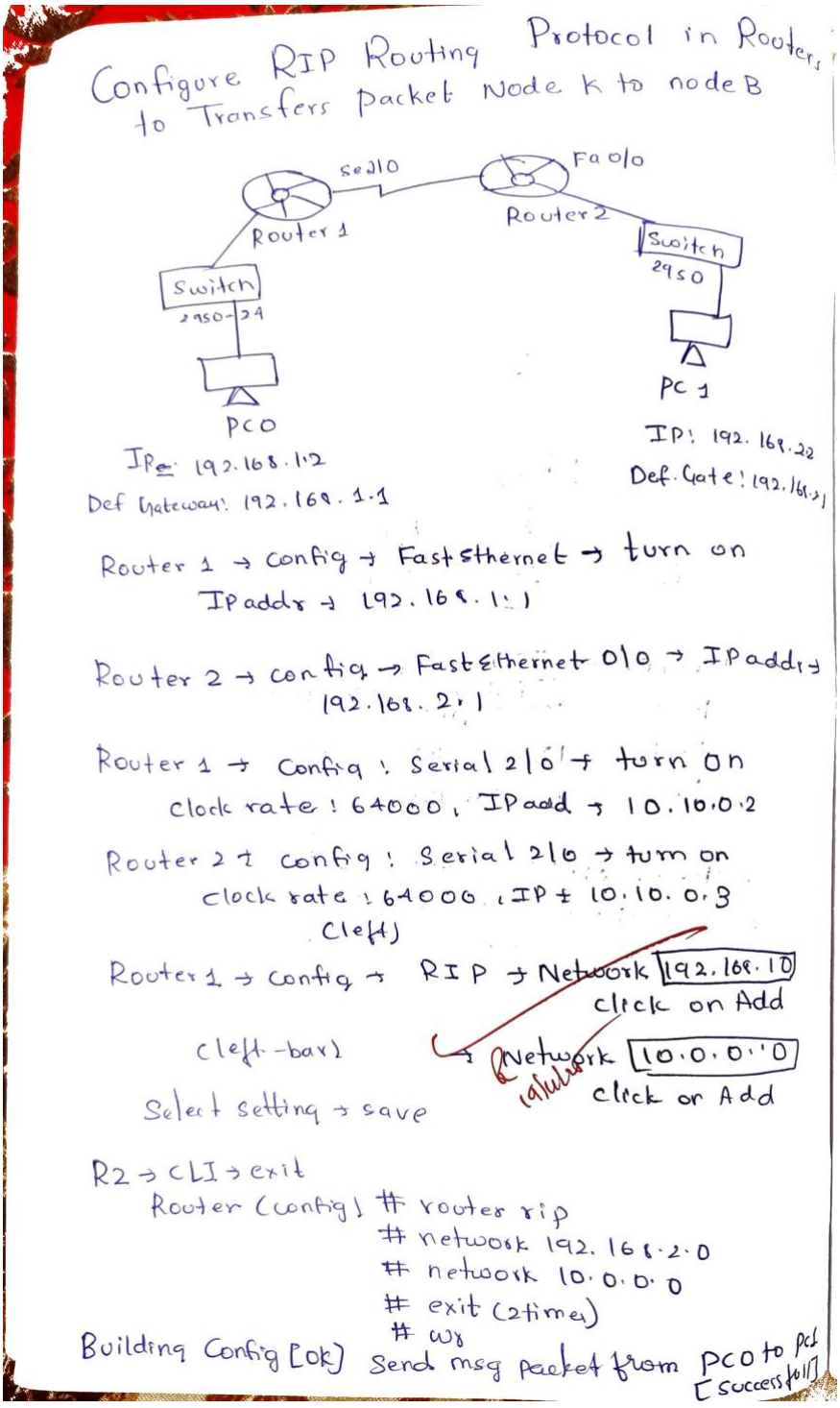
- The Telnet session successfully established a remote CLI connection to the router, confirming proper VTY line configuration and IP reachability between the IT office PC and the server room router.
- Command execution over the Telnet session demonstrated reliable remote device management.

Program 6

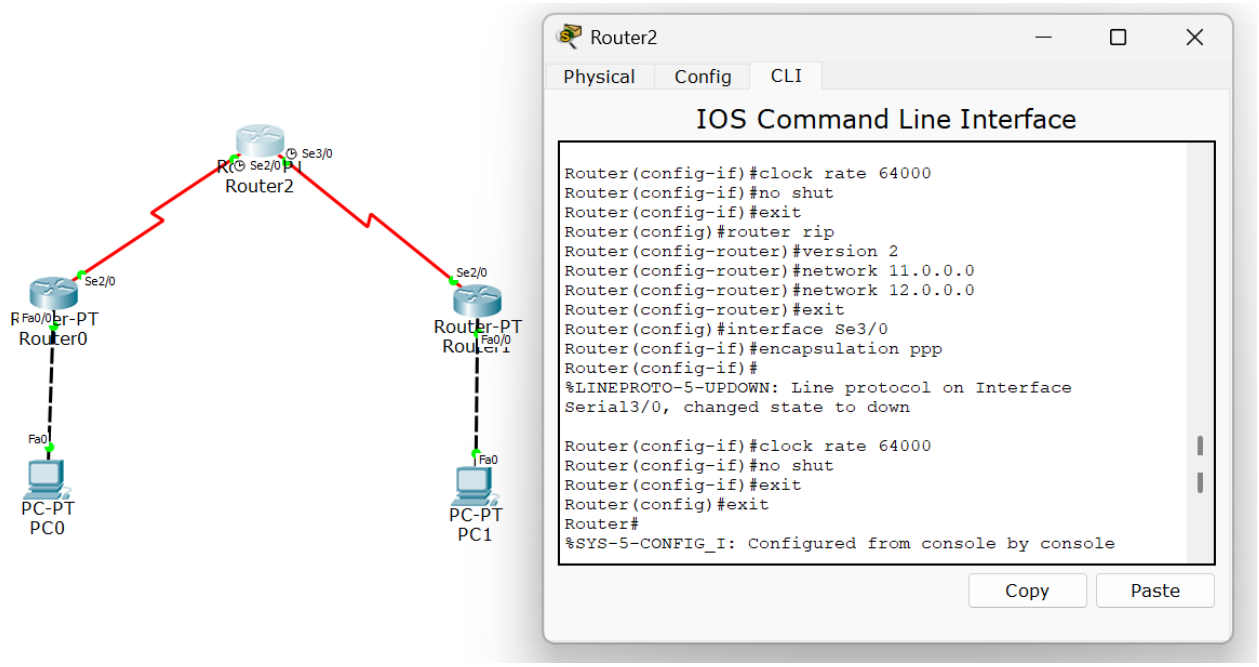
Aim of the program:

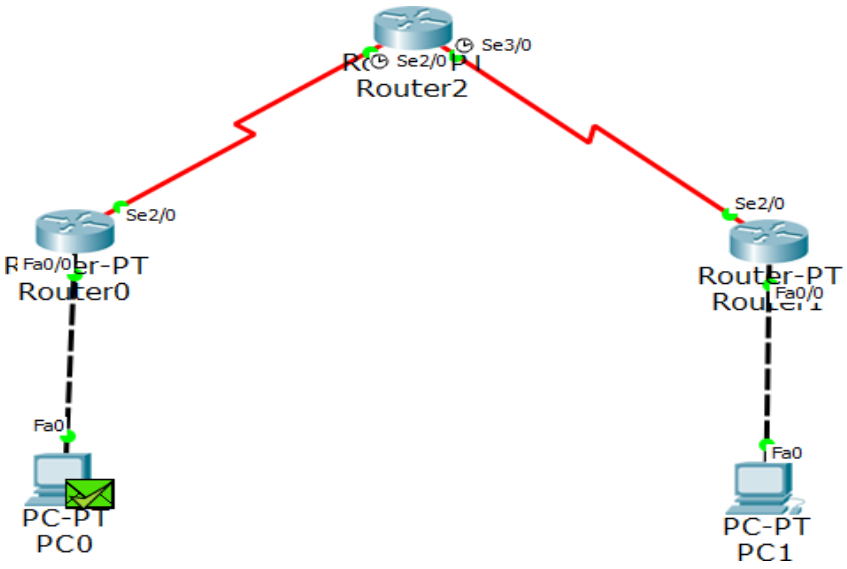
Configure RIP routing Protocol in Routers

Procedure and topology:



Screenshots/ Output:





Observation:

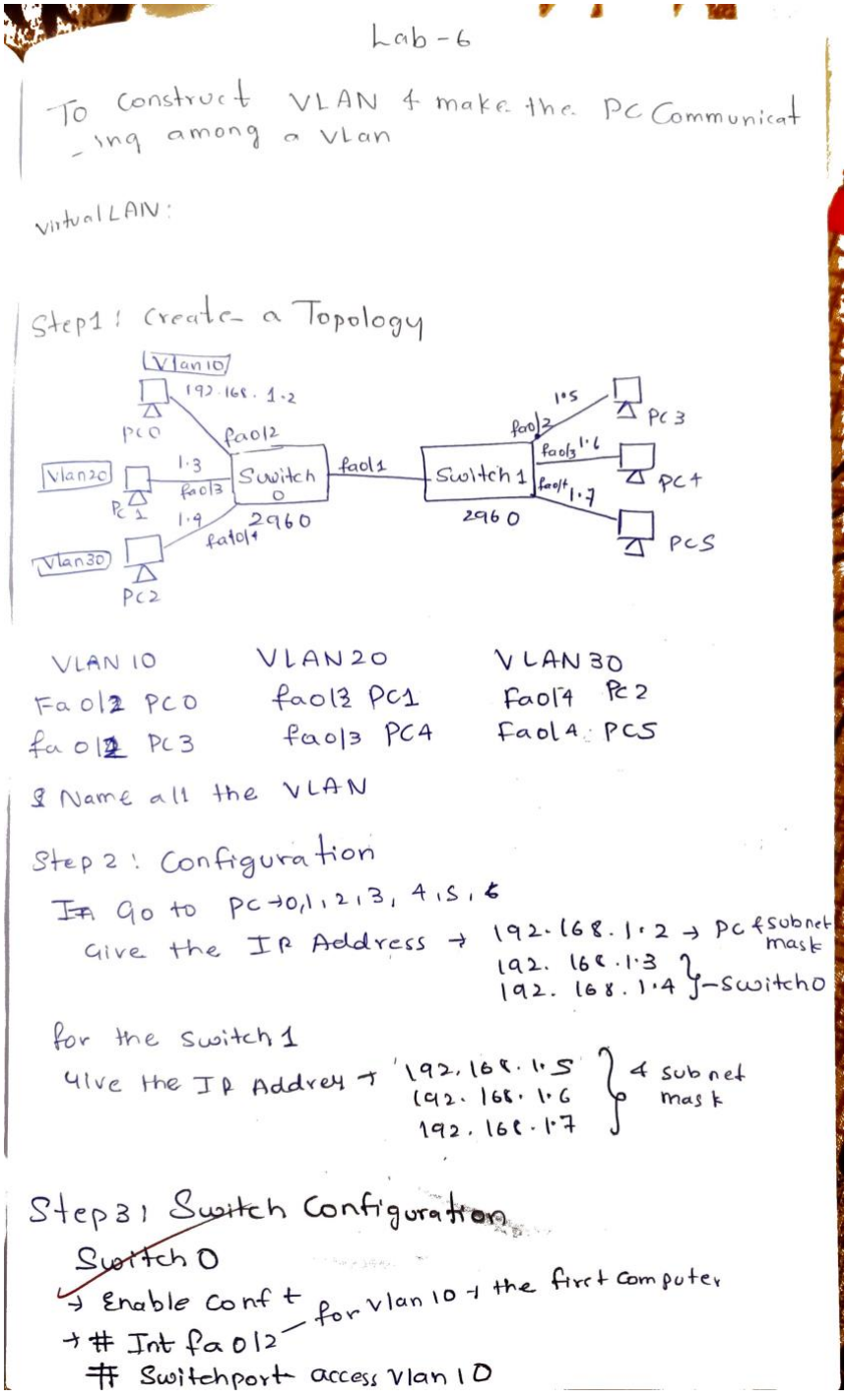
- RIP routing updates were successfully exchanged between routers, allowing each router to dynamically learn remote network routes through hop-count–based distance vector advertisements.
- The routing tables converged correctly, and successful ping tests confirmed end-to-end connectivity maintained by 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:



Lab-6

To construct VLAN & make the PC Communicating among a VLAN

Virtual LAN:

Step1: Create a Topology

VLAN 10	VLAN 20	VLAN 30
Fa0/2 PC0	fa0/3 PC1	Fa0/4 PC2
fa0/2 PC3	fa0/3 PC4	Fa0/4 PC5

& Name all the VLAN

Step2: Configuration

Go to PC→0,1,2,3,4,5,6

Give the IP Address → 192.168.1.2 → PC & subnet mask

192.168.1.3 } - Switch0

192.168.1.4 }

for the switch 1

Give the IP Address → 192.168.1.5 } & subnet mask

192.168.1.6

192.168.1.7

Step3: Switch Configuration

Switch 0

→ Enable conf t

→ # Int fa0/2 — for vlan 10 → the first computer

Switchport access vlan 10

Int Fa0/3 - 2nd com
Switchport access vlan 20
Int Fa0/4 - 3rd com
Switchport access vlan 30
Int Fa0/1 - b/w switches
Switchport mode Trunk

Same way for Switch 2 → CLI

→ Enable
→ Config t
→ Int Fa0/2
→ switchport access vlan 10
→ Int Fa0/3
→ Switchport access vlan 20
→ Int Fa0/4
→ switchport access vlan 30
→ Int Fa0/1
→ Switchport mode Trunk

Now 3. Send the msg / PDU packet from

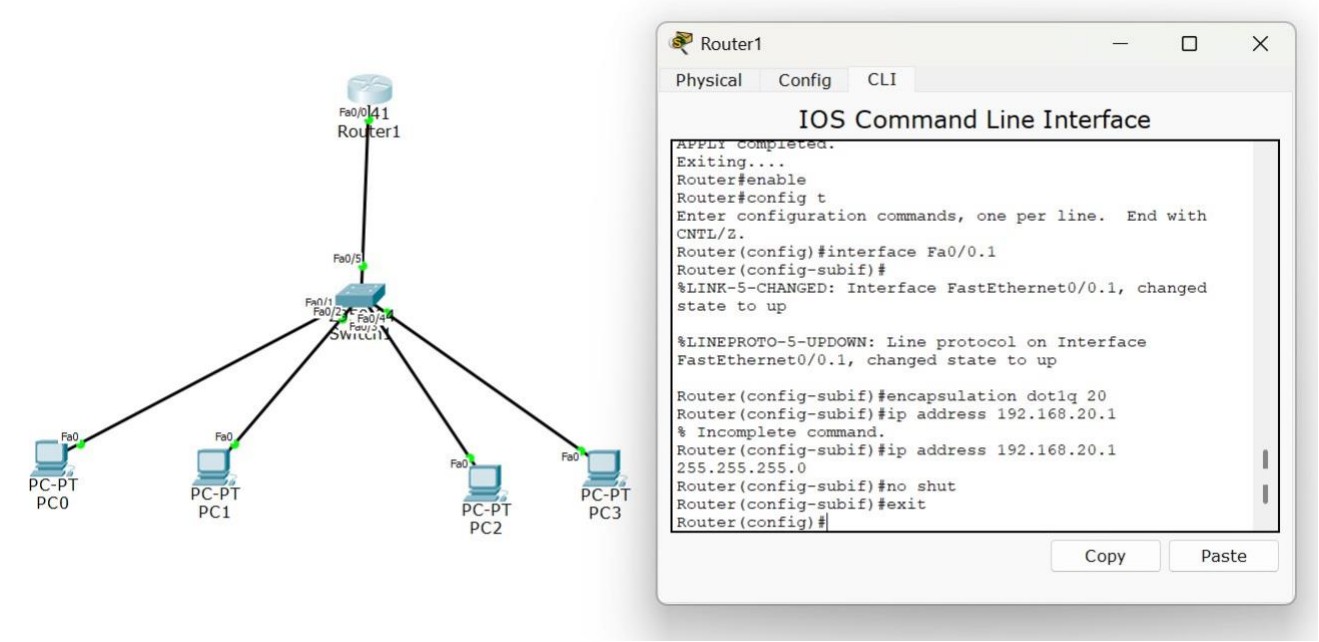
1) PC0 to PC3 (which is under the VLAN 1)

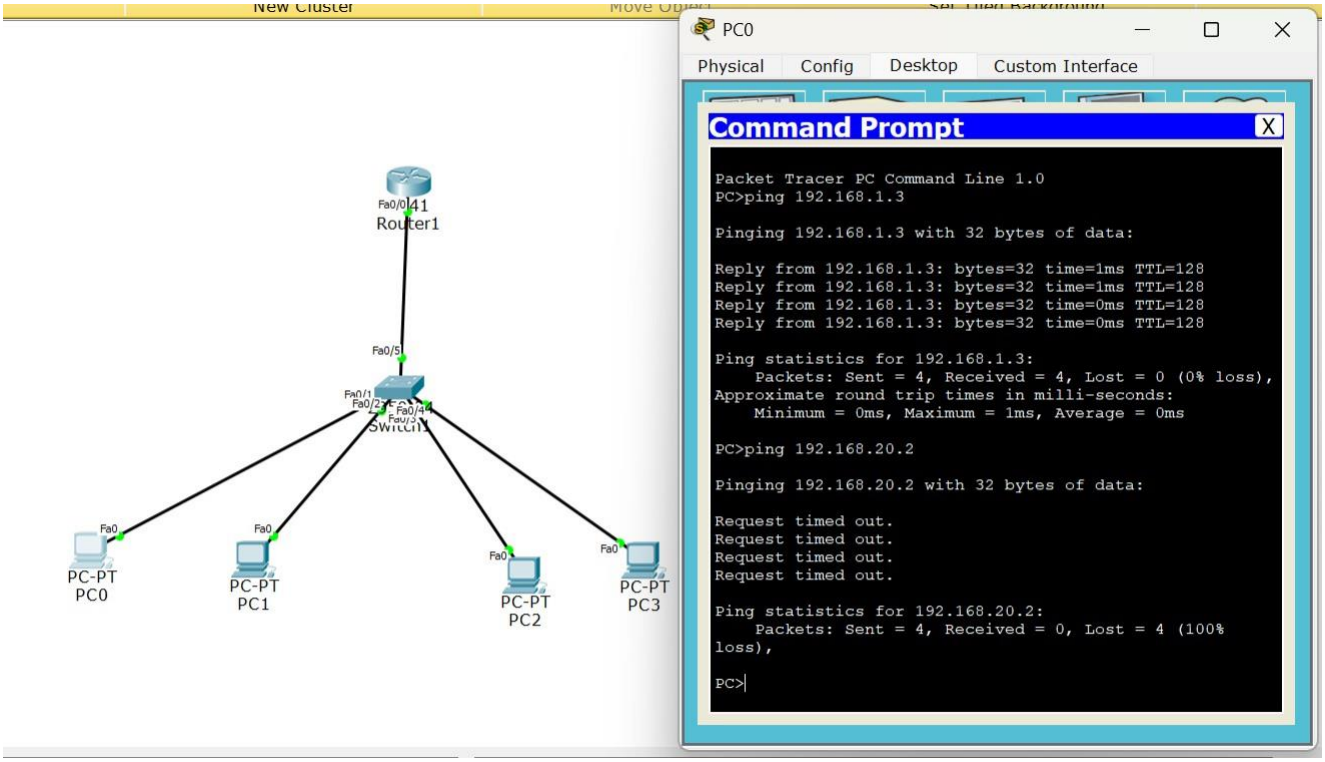
Output: message is sent Successfully

2) PC1 to PC3 (Not Same VLAN) from VLAN 2 to VLAN 3

Output: message is failed

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:

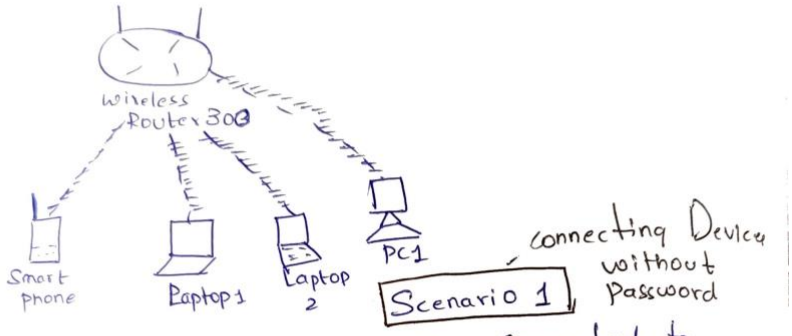
To construct a WLAN and make the nodes communicate wirelessly

Procedure and topology:

15/10/25

To Construct a WLAN & make the nodes Communicate wirelessly

Create the topology



1) here the Smartphone will be Connected to Router Directly (no need for connectivity

2) For the Laptop another device to be connected to WLAN
   - → click on Laptop
   - → open physical device / switch off then
   - → Remove the physical Adapter connected to it
   - → Add the wireless Adapter (Drag & Add)

   Now the laptop would be connected to WLAN

3) Same way Connecting to ~~w~~ PC
   - → click on Pc
   - → open its physical / switch off then
   - → Remove the physical Adapter connected to it
   - → then Add the wireless Adapter

   Now the PC will be Connected to WLAN

Scenario 2 → Adding Password to the WLAN, where the Device who knows password can be Connected to it

Step 1 : click on Wireless Router (WLAN)

click on Physical

↳ wireless

here the name of device → SSID will be Default

Change that to name

↳ SSID → BMSCE

then click on WPAS password prass

↳ Give password → bmsce1234

then close the tab

After this the Device Connected to it will all be Disconnected

Step 2 : Connect the Devices

Laptop / Smartphone

↳ physical click on

↳ ~~S~~Wireless

↳ Give the SSID as BMSCE

And give the password

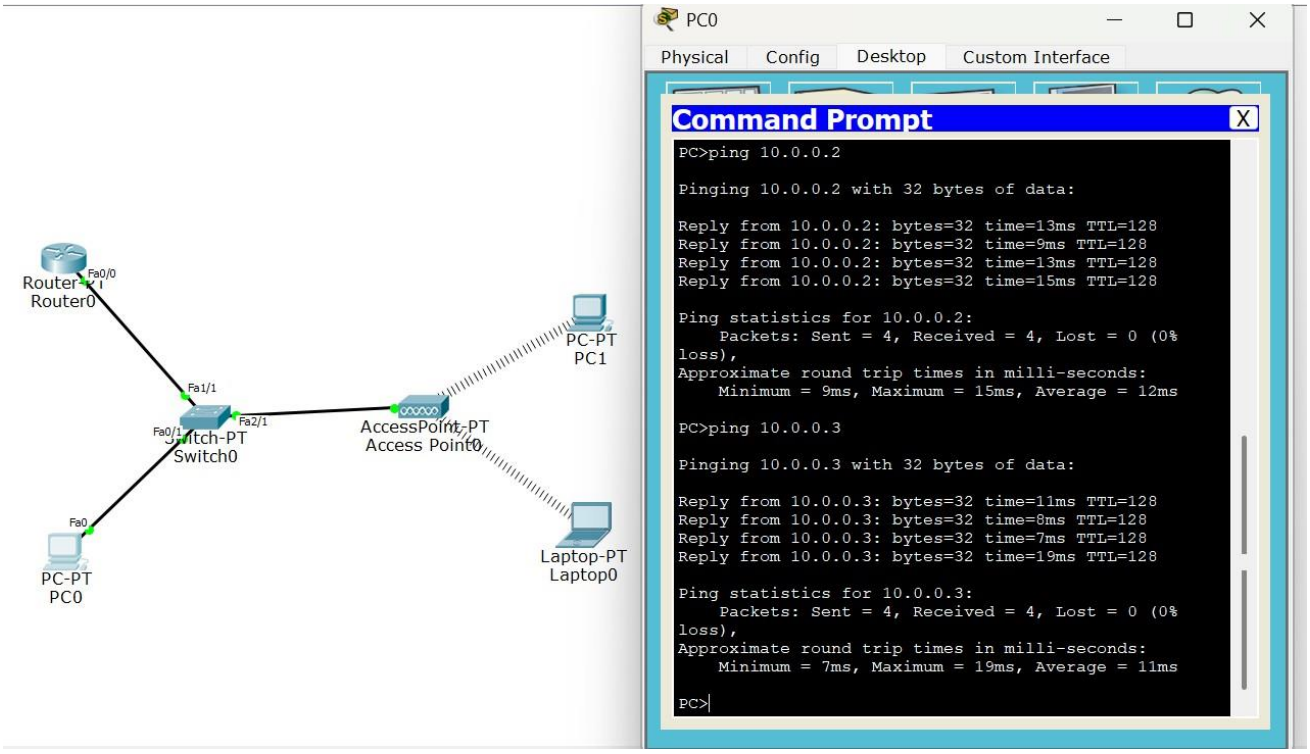
WPAS → bmsce

Note that the Encryption should be like AES

Now the Device will be Connected to WLAN

10/10/25

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:

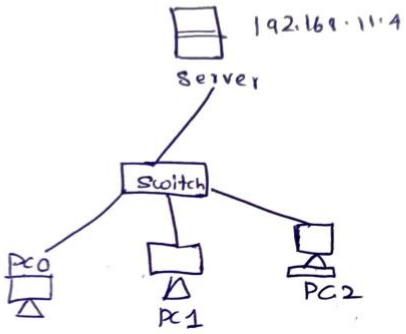
To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)

Procedure and topology:

ARP → Address Resolution Protocol

ARP is used to map an IP Address to a mac Address
ARP is used to get Datalink layer Address, MAC Address with the help of IP address

Step 1 create the Topology

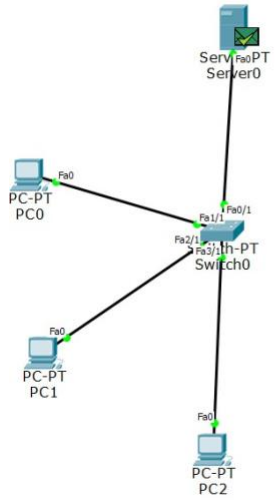


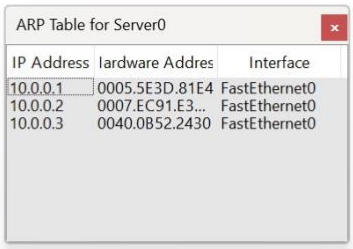
Step2 : Configure IP Address

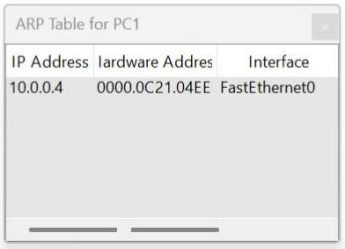
	IP Add	Subnetmask	DNS Server
PC0	192.168.11.1	255.255.255.0	192.168.11.4
PC1	192.168.11.2	255.255.255.0	192.168.11.4
PC2	192.168.11.3	255.255.255.0	192.168.11.4

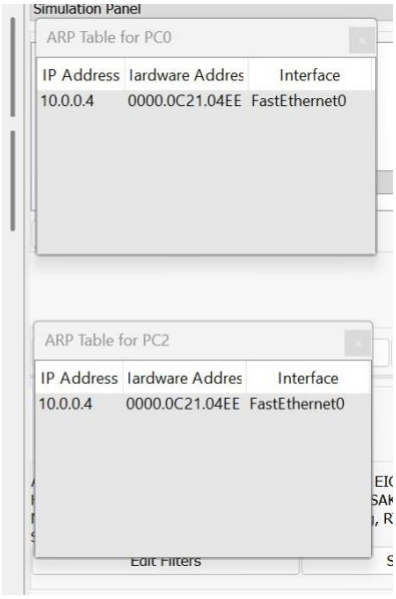
Simulation: After sending 1 PDU delete Scenario then resend other PDU then click on that to see MAC address C caputure |Forward

Screenshots/ Output:









Observation:

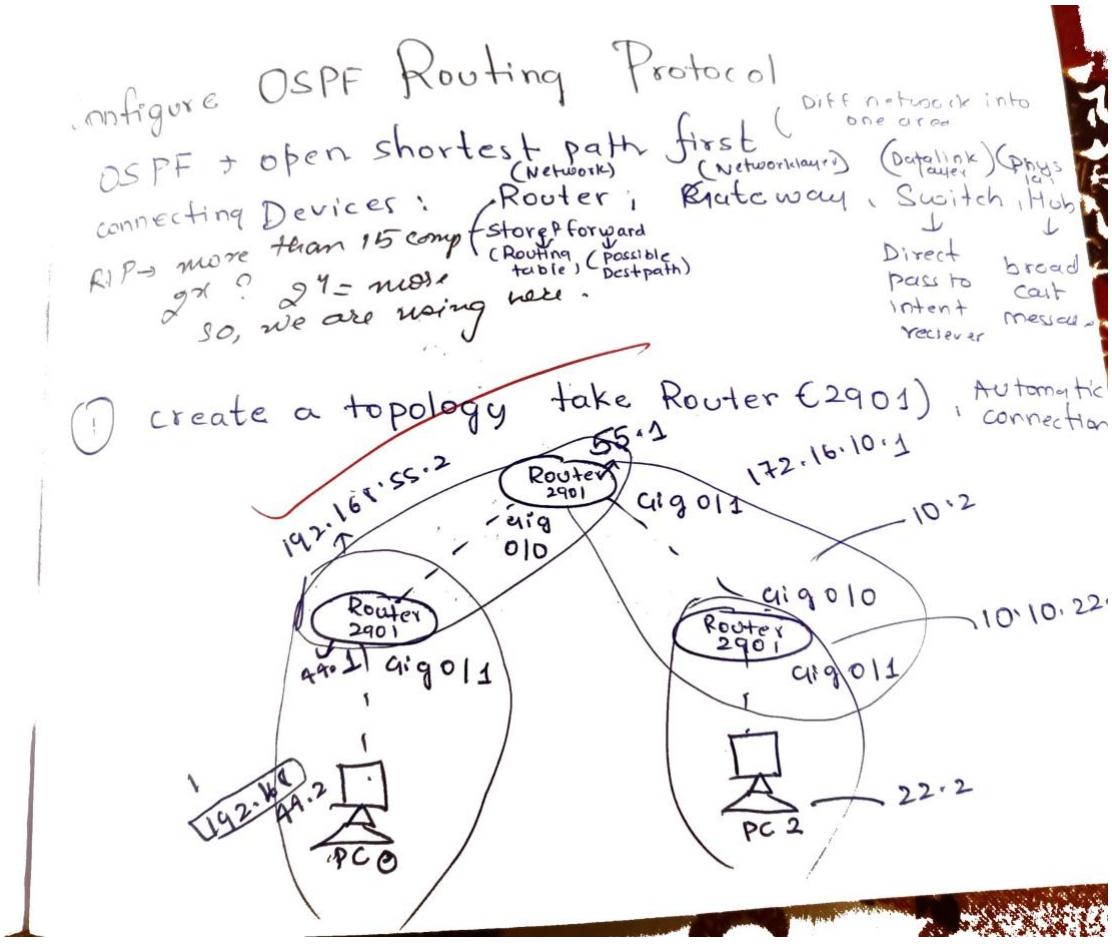
- ARP successfully resolved the destination IP address to its corresponding MAC address, as seen from ARP request and reply exchanges between LAN hosts.
- The populated ARP tables and successful ping communication confirmed correct layer-2 addressing, frame forwarding, and basic LAN operation.

Program 10

Aim of the program:

Configure OSPF routing protocol

Procedure and topology:


Configure OSPF Routing Protocol
OSPF → open shortest path first (Diff network into one area)
(Network) (Network layer) (Datalink layer) (Physical)
connecting Devices : Router, Gateway, Switch, Hub
RIP → more than 15 comp
Router (store & forward) (Routing table) (Possible Destpath)
Direct pass to intent receiver
broad cast messa
create a topology take Router (2901), Automatic connection
192.168.55.2
55.1
Router 2901
172.16.10.1
gig 0/0
gig 0/1
10.2
Router 2901
gig 0/0
10.10.22
49.1
gig 0/1
Router 2901
gig 0/1
192.168
49.2
PC 0
PC 2
22.2

Ip V4 addess 32 bit

class A → first 8 bits (octacle) IP & Remaing host name

Class B → 2octale (Network ID, Network ID)

class C → 3octale

To reach each network applying Subnetmask

(1) Configure all the devices

PC 0 → 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 0/1 → on → 192.168.44.1, 255.255.255.0

(3) router 2 → config → Gigath 0/0 → on → 192.168.55.1, 255.255.255.0
Giga 0/1 → on → 172.16.10.1, 255.255.255.0

(4) router 1 → config → Gigath0/0 → on → 172.16.10.2, 255.255.0.0
Giga 0/1 → on → 10.10.22.1

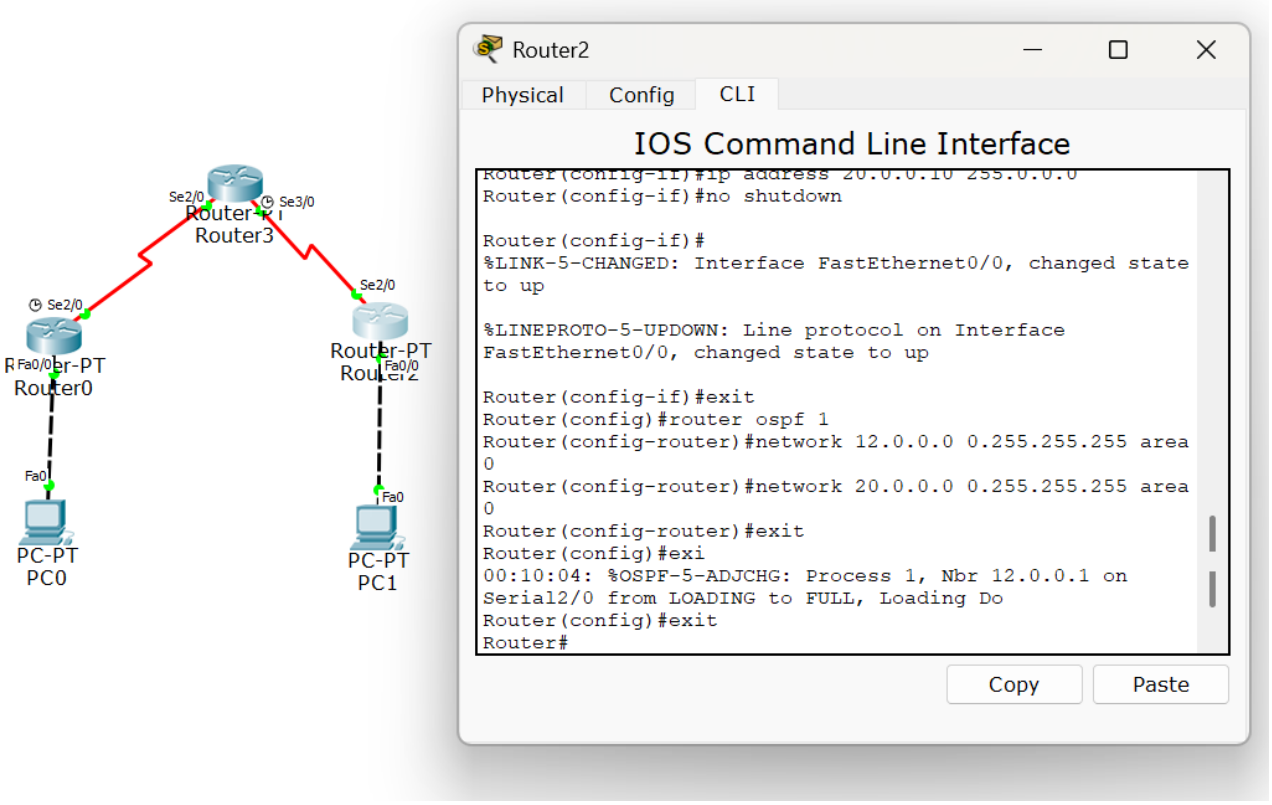
Rg10/25

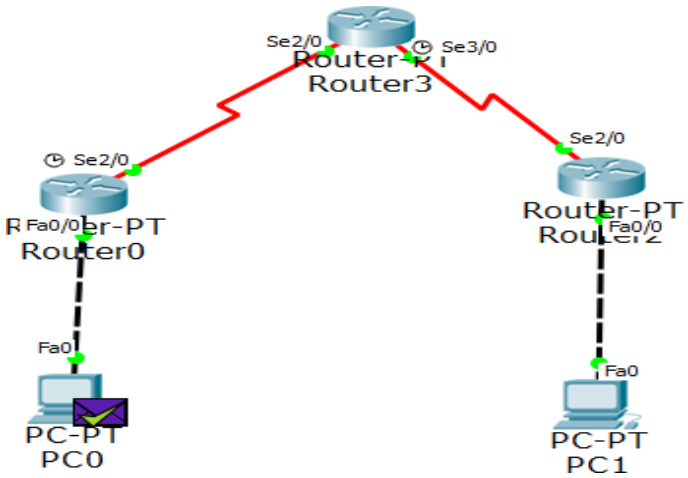
(5) PC 1 → Desktop
↳ IP config
↳ IP → 10.10.22.2, 255.0.0.0
Gateway → 10.10.22.1

Now send the Packet and if it is Successfully completed

Output
Status → Successful

Screenshots/ Output:





Observation:

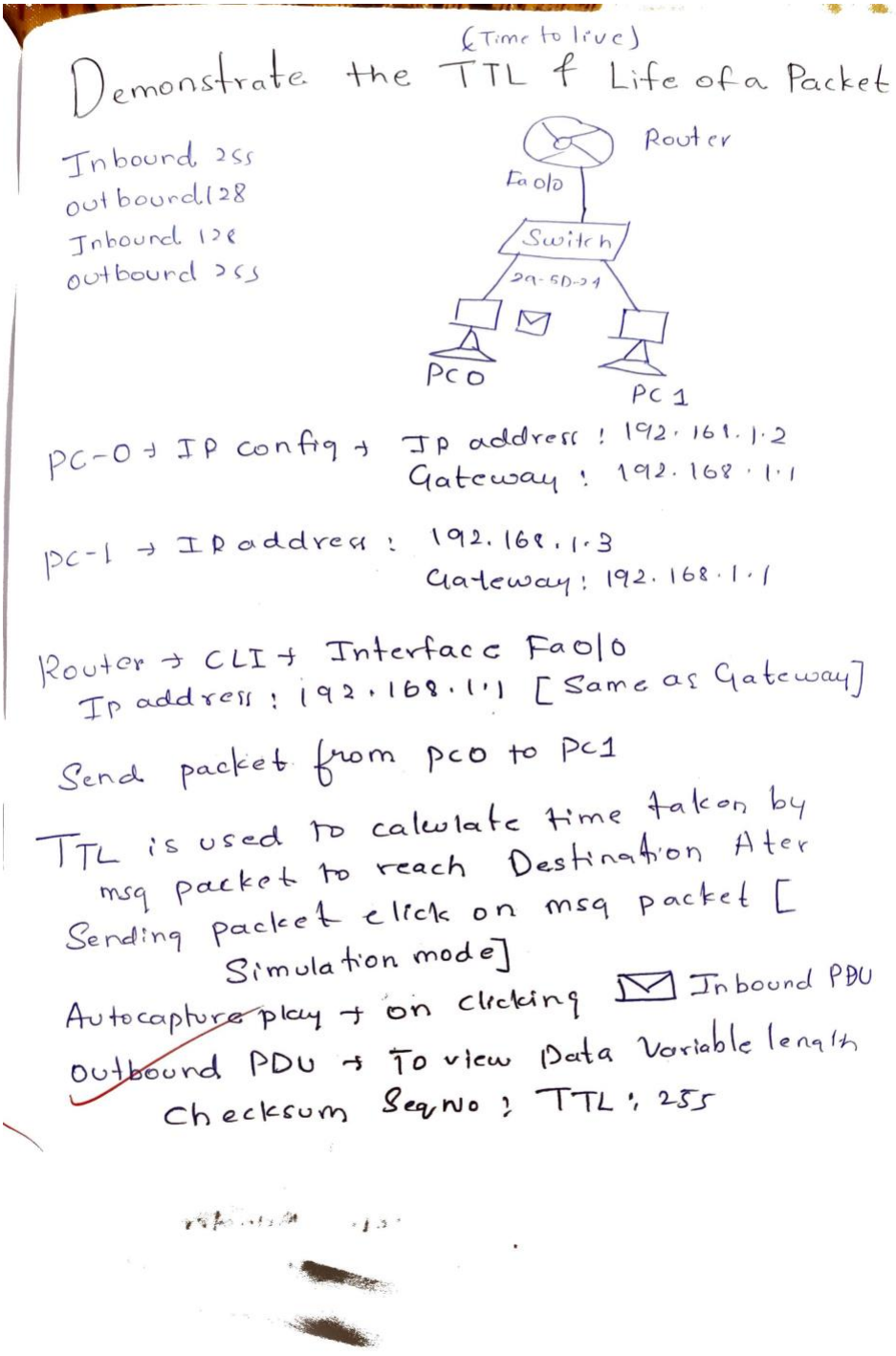
- OSPF successfully established neighbour adjacencies and exchanged LSAs, allowing routers to build a synchronized link-state database across the OSPF area.
- The routing tables converged using SPF calculations, and successful pings confirmed efficient path selection and dynamic route learning through OSPF.

Program 11

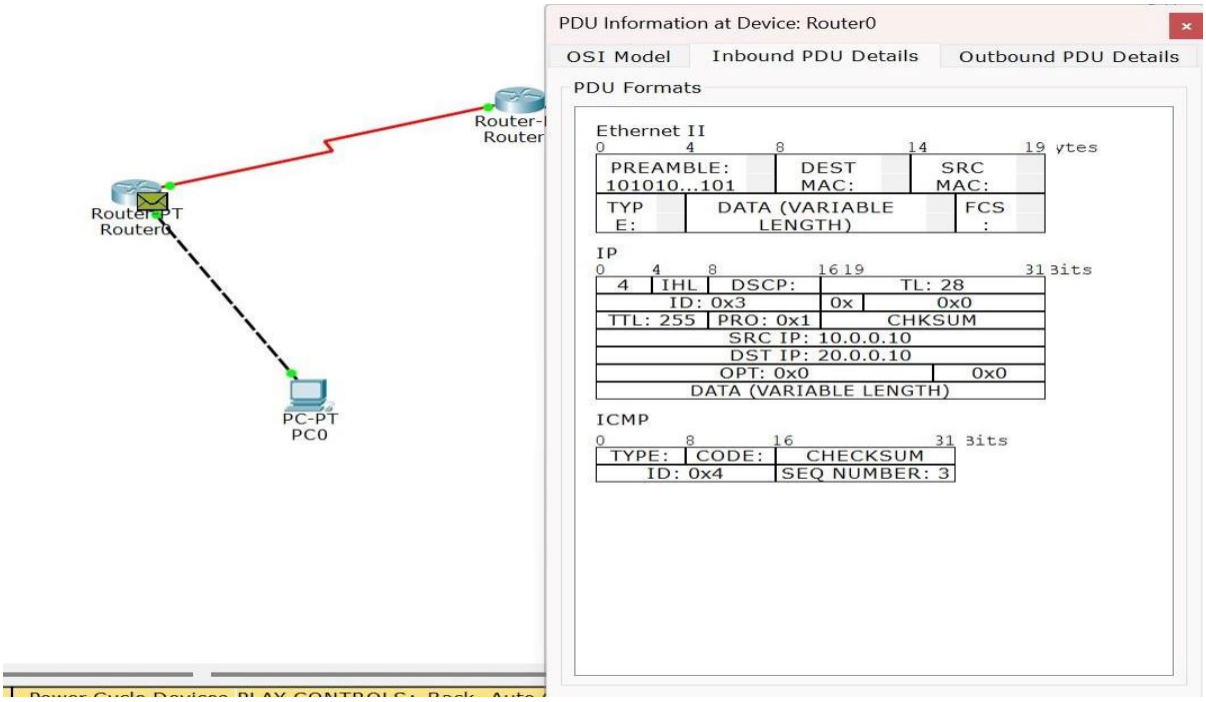
Aim of the program:

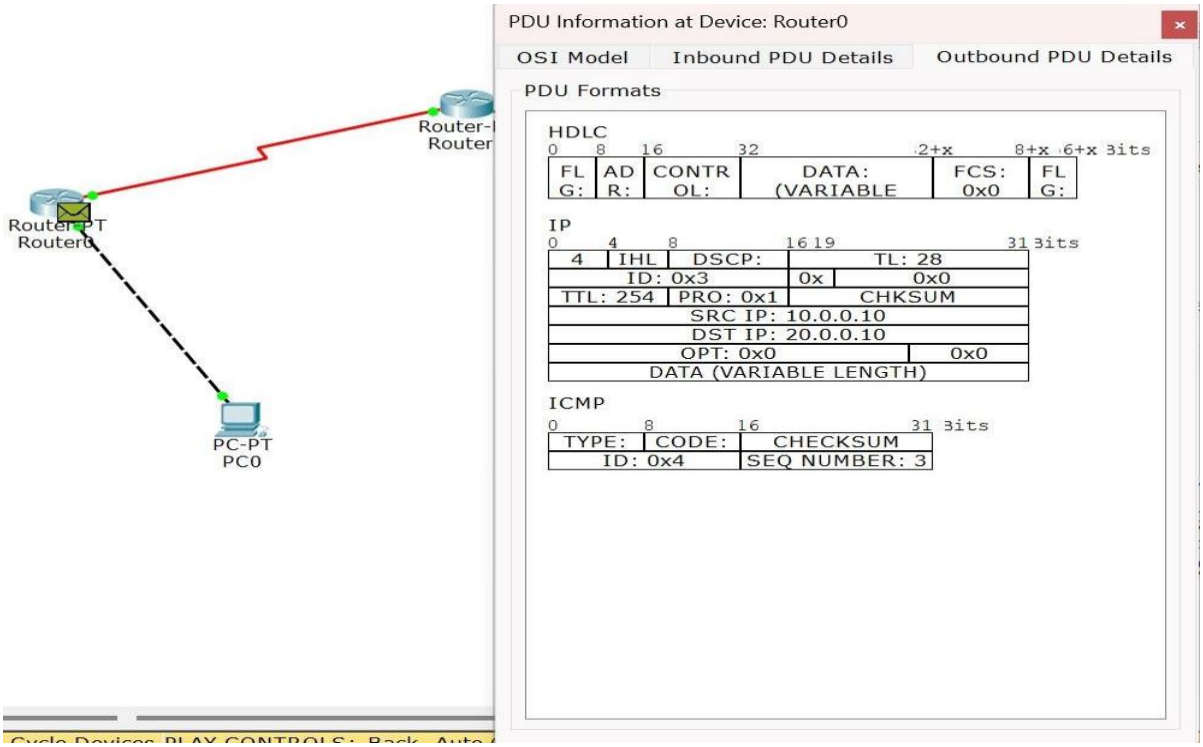
Demonstrate the TTL/ Life of a Packet

Procedure and topology:



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

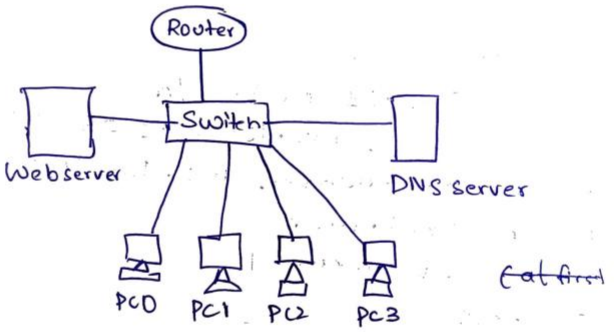
Procedure and topology:

Lab-3 10/9/25

1) Configure Web Server, DNS within a LAN
2) Configure IP Addresses to Router in Packet Tracer. Explore the following messages
   i) Ping Response (IP settings are correct, Devices are connected route exits)
   ii) Destination Unreachable (Device does not know how to reach destination gateway, Router is not set)
   iii) Request Time out (device is there but not responding in time msg sent, no reply)
   iv) Reply (connected showing Reply msg

(1)

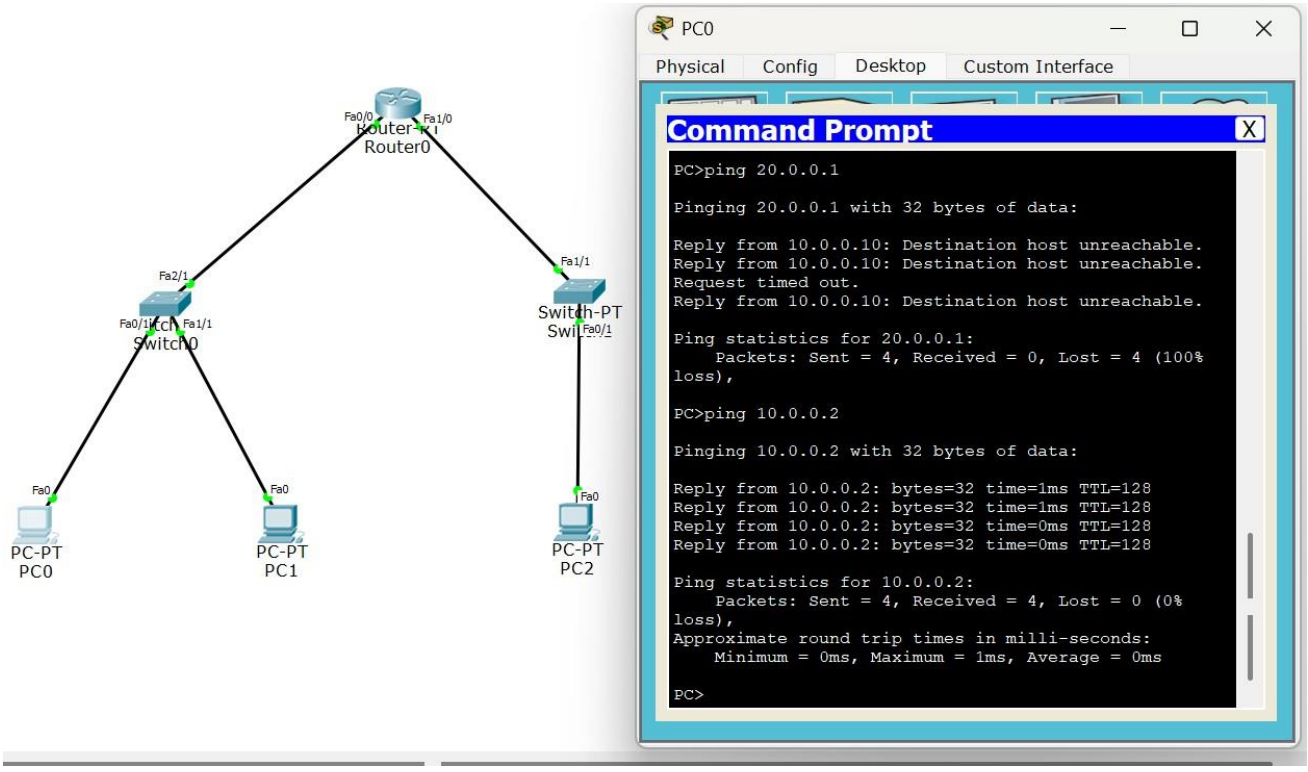
1) Create a topology Using the Devices



2) click on web server & click on Services
   → Select http & HTTPS should be on
   → Select anyone file.htm in the file manager
      Click on file & click on edit and add some new message & save the file
   → In that only Go to Desktop & Ipaddress (IP config)
      IP address : 192.168.1.6
      Subnet mask : 255.255.255.0
      DNS server : 192.168.1.5

Router Configuration
click on CLI
Type NO
Then you can see Router>
step2: type the enable Next
#Conf t
int Fa 0/0 (For this once Check with the Router green dots then you can see) ~~Fastethernet~~ 0/1 1/0
→ # ip address 192.168.10.1 255.255.255.0
→ # ip helper-address 192.168.10.2
→ # no shutdown
→ do write memory
→ #Exit
Repeat the same
int Fa0/1 or Fa 1/0
ip address 192.168.20.1 255.255.255.0
iphelper-address 192.168.10.2
no shutdown
do write memory
exit
Then to check it it is working or not
go ~~do~~ any of laptop or ~~screenshot~~ PC
PC→ Desktop→ IPconfiguration → DHCP

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 non-responsive hosts.

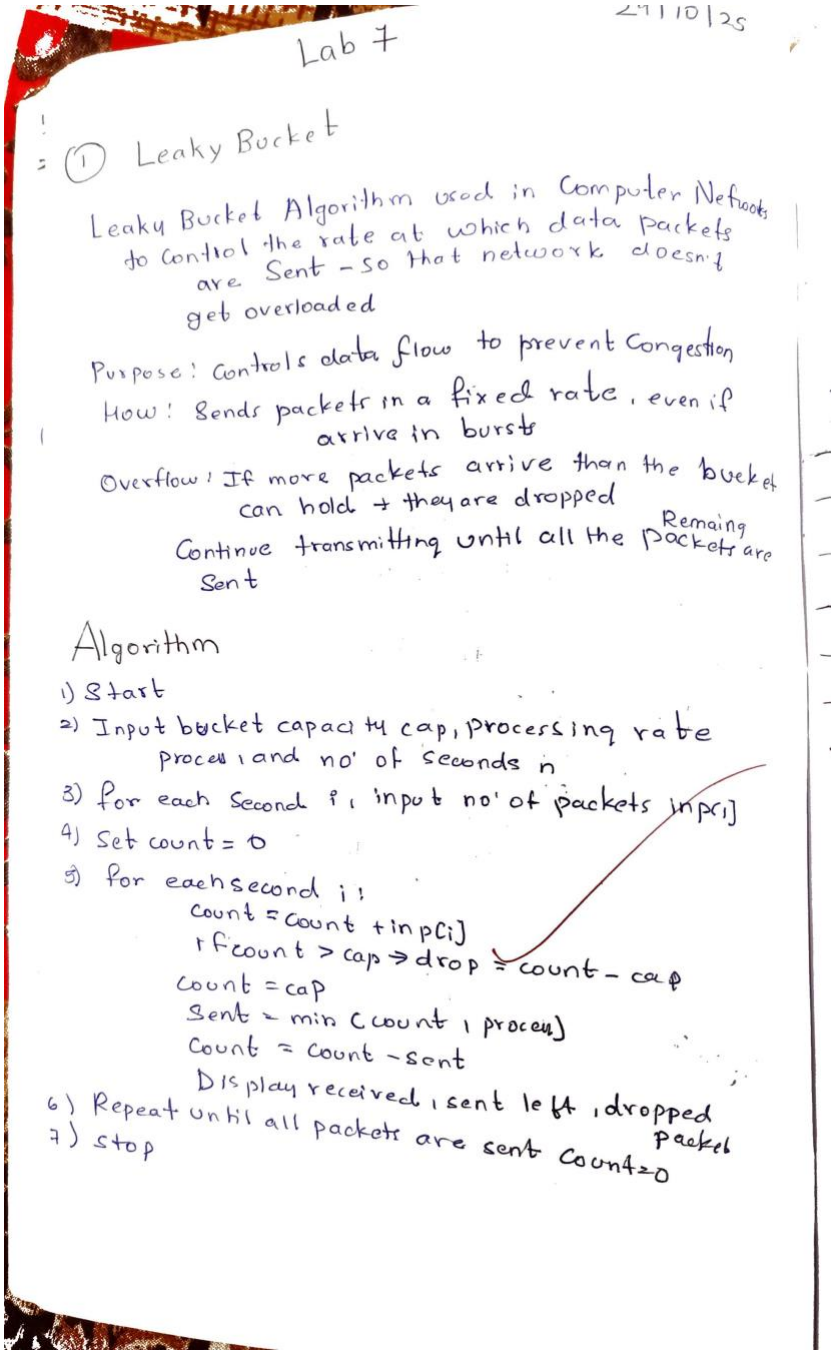
CYCLE 2:

Program 13

Aim of the program:

Write a program for congestion control using Leaky bucket algorithm

Procedure and topology:



Lab 7

2/11/25

① Leaky Bucket

Leaky Bucket Algorithm used in Computer Networks to control the rate at which data packets are sent - so that network doesn't get overloaded

Purpose: Controls data flow to prevent Congestion

How: Sends packets in a fixed rate, even if arrive in bursts

Overflow: If more packets arrive than the bucket can hold → they are dropped

Continue transmitting until all the Remaing Packets are Sent

Algorithm

1) Start
2) Input bucket capacity cap, processing rate proces, and no' of seconds n
3) for each second i, input no' of packets inp[i]
4) Set count = 0
5) for each second i:
   count = count + inp[i]
   if count > cap → drop = count - cap
   count = cap
   Sent = min (count, proces)
   count = count - sent
   Display received, sent left, dropped Packets
6) Repeat until all packets are sent count=0
7) stop

Output:

Enter the bucket size: 10

Enter the processing rate: 4

Enter the no of seconds you want to simulate: 4

Enter the size of packet entering at 1sec: 6

Enter the size of packet entering at 2sec: 8

" " 3sec: 2

" " 4sec: 10

Seconds	Packet Received	Before Sending	Sent	Left	Dropped
1	6	6	4	2	0
2	8	10 (fits in bucket)	4	6	0
3	2	8	4	4	0
4	10	14 (exceeds capacity)	4	6	4 dropped
5	0	6	4	2	0
6	0	2	2	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:

② Error Detecting code Using CRC

CCITT ((6 bit)

(a) F = 1101011011 $f(x) = x^4 + x + 1$

(a) G = 10011

5 zero − 1 = 4 zero

EOR → add in Quotient
→ differ → 1
Same → 0

```
           110000101
 10011 ) 1101011011 0000
         10011
         0 10011          (exclude starting 0's)
           10011
         00000 10110      (only take extra zeroes to match 5 digit (10011))
               10011
               00 10100
                  10011
                  00 1110 → Remainder
```

If Remainder is 0 then no error
Here It is not 0 So error is there

b) F = 1101011011
G = 10011

c) F = 100100
$G = x^3 + x^2 + 1$
G = 1101 (4−1 = 3 0's)

```
         11110
 1101 ) 100100 000
        1101
        0 1000
          1101
          0 1010
            1101
            0 1110
              1101
              00 1100
                 1101
                 0001
```

There is error in this

R 29/10/21

Screenshots/ Output:

```
Output

Enter message bits: 101100
Enter polynomial g(x): 1001

Padded data (Message + zeros): 101100000
CRC bits (remainder): 001
Transmitted message: 101100001

Enter received bits: 101100001

No Error detected. Message OK.


=== Code Execution Successful ===
```

Observation:

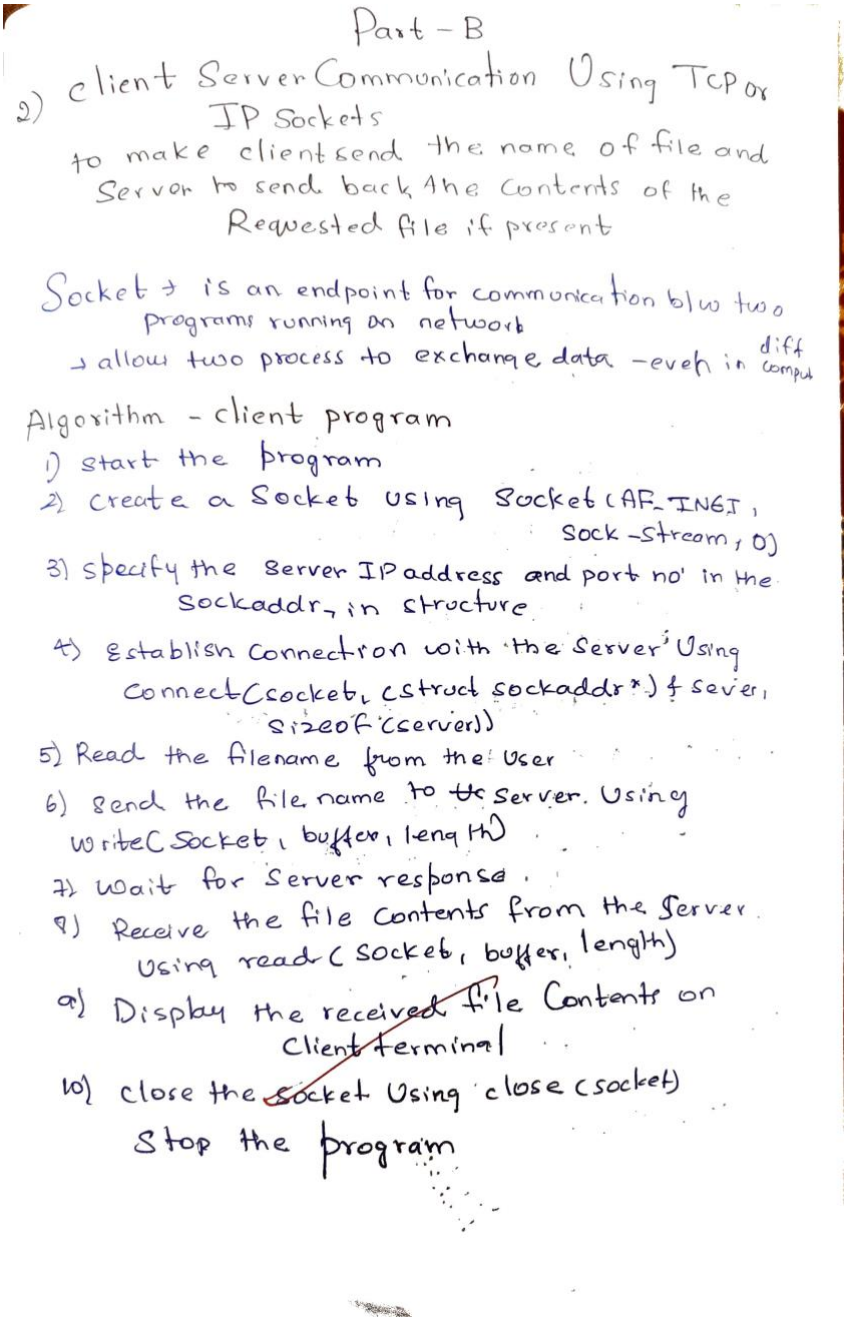
- The CRC-CCITT (16-bit) algorithm correctly generated a checksum for the transmitted data, ensuring reliable detection of single-bit and burst errors.
- Intentional error tests produced mismatched CRC values at the receiver, confirming accurate error detection through polynomial division.

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:



Part - B

2) Client Server Communication Using TCP or IP Sockets

to make client send the name of file and Server to send back the contents of the Requested file if present

Socket → is an endpoint for communication b/w two programs running on network

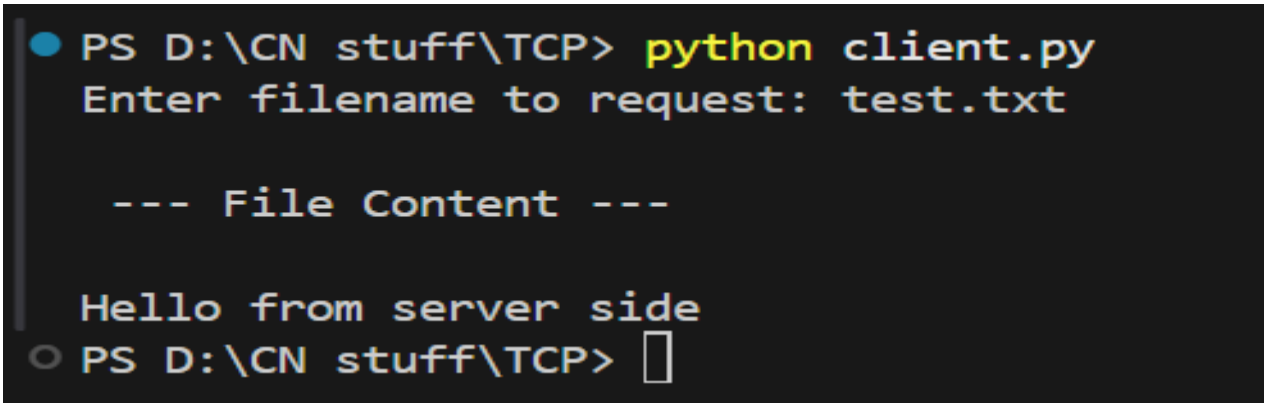
→ allows two process to exchange data - even in diff comput

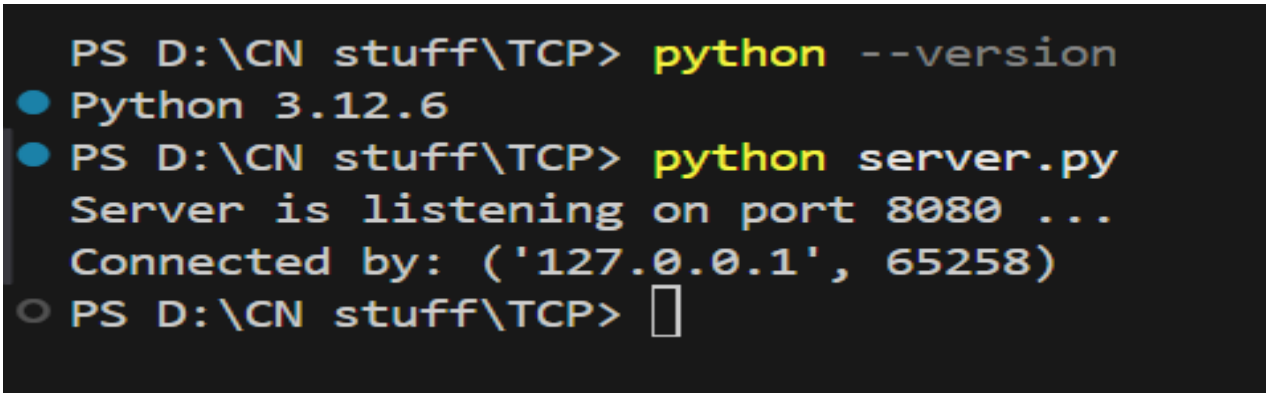
Algorithm - client program

1) start the program
2) create a Socket using Socket(AF_INET, Sock_Stream, 0)
3) specify the server IP address and port no' in the sockaddr_in structure
4) Establish Connection with the Server Using Connect(socket, (struct sockaddr*) & sever, sizeof(server))
5) Read the filename from the User
6) Send the file name to the server. Using write(Socket, buffer, length)
7) Wait for Server response.
8) Receive the file contents from the Server using read(socket, buffer, length)
9) Display the received file Contents on Client terminal
10) close the Socket Using close(socket)

Stop the program

Screenshots/ Output:





Observation:

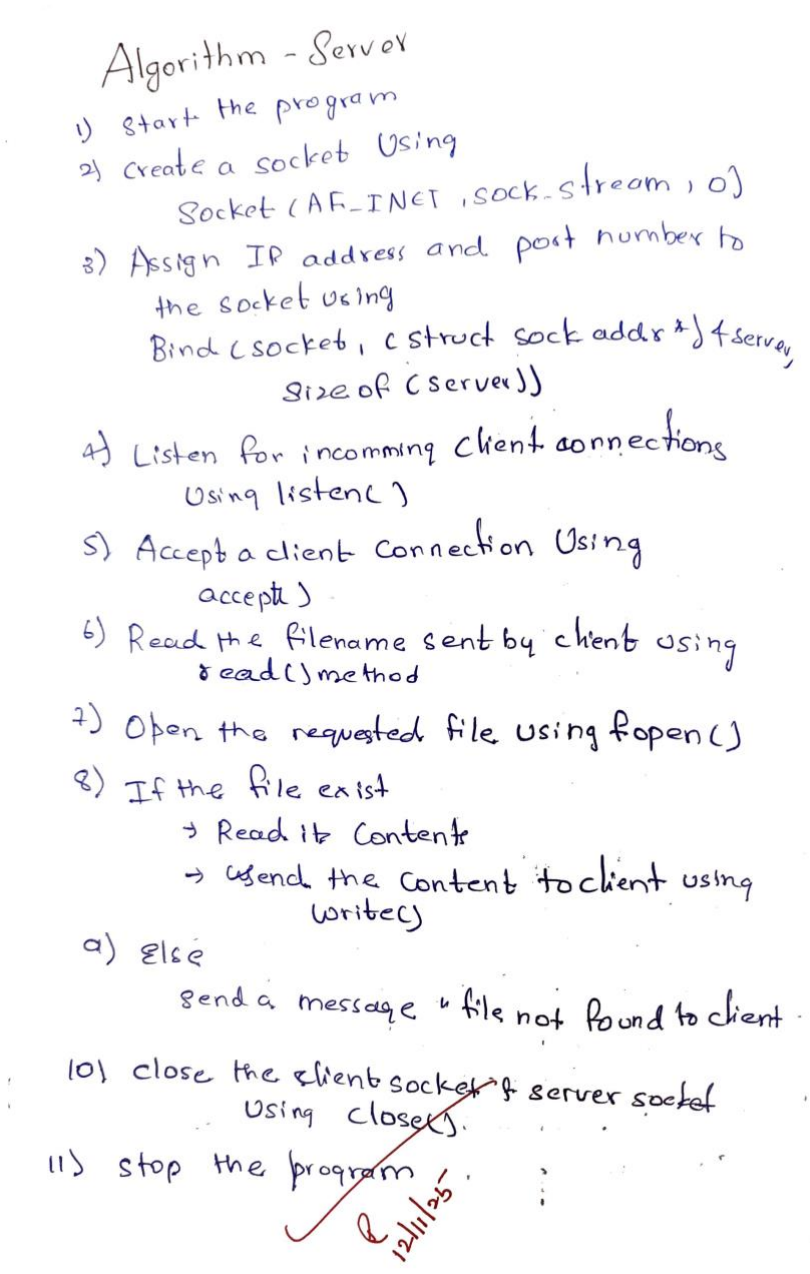
- The TCP client successfully established a reliable connection with the server and transmitted the requested filename using stream-oriented communication.
- The server correctly retrieved and returned the file contents over the same TCP session, demonstrating reliable data transfer, acknowledgment handling,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

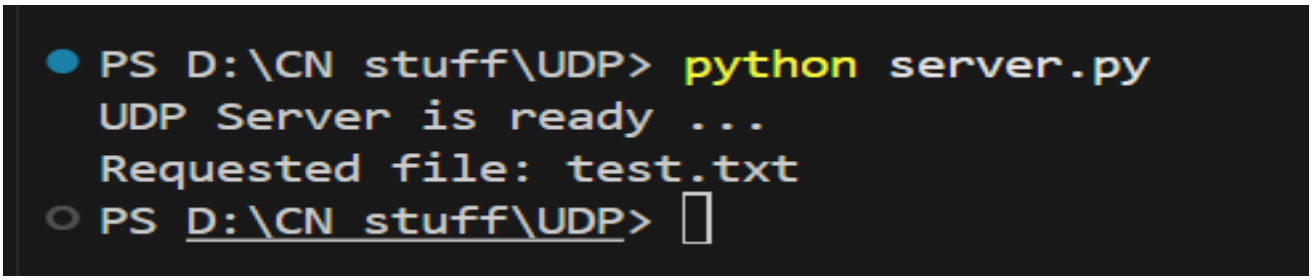


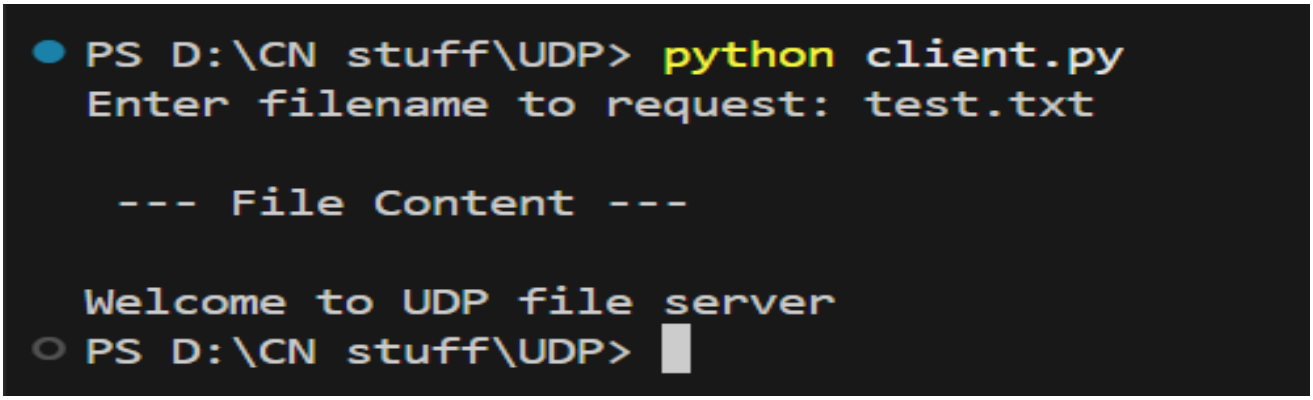
Algorithm - Server

1) Start the program
2) Create a socket Using
   Socket (AF_INET, sock_stream, 0)
3) Assign IP address and port number to the socket using
   Bind (socket, (struct sock addr *) &server, sizeof (server))
4) Listen for incomming client connections Using listen()
5) Accept a client connection Using accept()
6) Read the filename sent by client using read() method
7) Open the requested file using fopen()
8) If the file exist
   → Read it Contents
   → Send the content to client using write()
9) Else
   send a message "file not found to client
10) close the client socket & server socket Using close().
11) stop the program

12/11/25

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.