

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB RECORD

Computer Network Lab (23CS5PCCON)

Submitted by

Sagar Bangari (1BM23CS231)

in partial fulfilment 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 **Sagar Bangari (1BM23CS231)**, 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.

Ramya K M Assistant Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE
--	--

Index-Cycle-I

Sl. No.	Date	Experiment Title	Page No.
1	04/10/2024	Create a topology involving multiple hubs and a switch connecting them to simulate with simple PDU.	1
2	18/10/2024	Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply	10
3	25/10/2024	Configure default route, static route to the router	17
4	08/11/2024	Configure DHCP within a LAN and outside LAN.	26
5	22/11/2024	Configure RIP routing Protocol in Routers	36
6	22/11/2024	Configure OSPF routing protocol	43
7	22/11/2024	Demonstrate the TTL/ Life of a Packet	53
8	08/11/2024	Configure Web Server, DNS within a LAN.	57
9	20/12/2024	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)	61
10	20/12/2024	To understand the operation of TELNET by accessing the router in server room from a PC in IT office.	66
11	20/12/2024	To construct a VLAN and make the PC's communicate among a VLAN	70
12	20/12/2024	To construct a WLAN and make the nodes communicate wirelessly	75

Github Link:

<https://github.com/Sagar-bangari/CN>

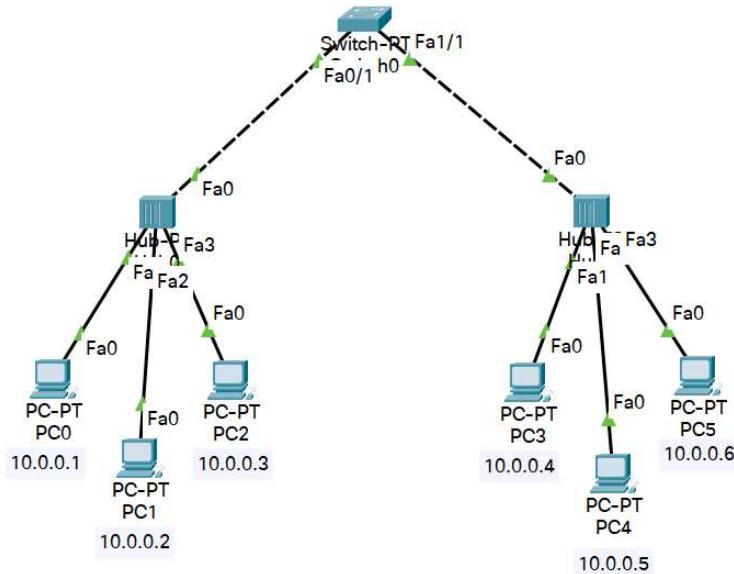
Index-Cycle-II

Sl. No.	Date	Experiment Title	Page No.
1	15/11/2024	Write a program for error detecting code using CRC-CCITT (16-bits).	80
2	15/11/2024	Write a program for congestion control using Leaky bucket algorithm	84
3	20/12/2024	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.	87
4	20/12/2024	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.	91

Cycle-I

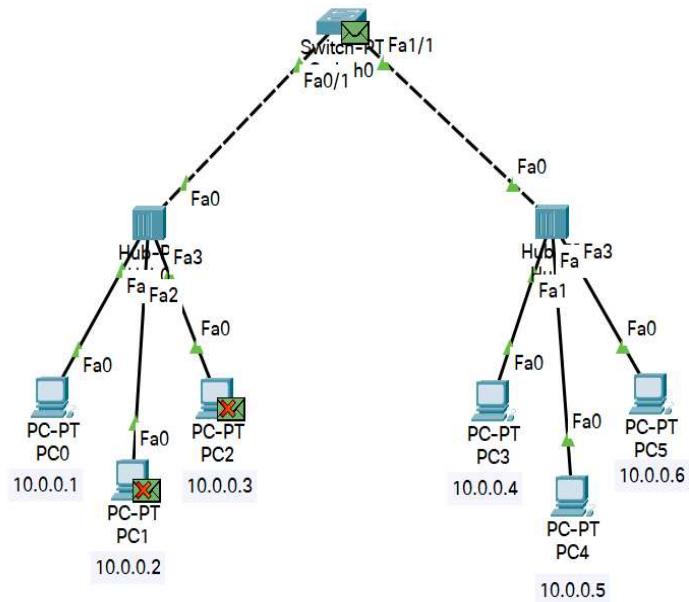
Program 1

- i. Create a topology involving multiple hubs and a switch connecting them to simulate with simple PDU.
- ii. Procedure along with the topology

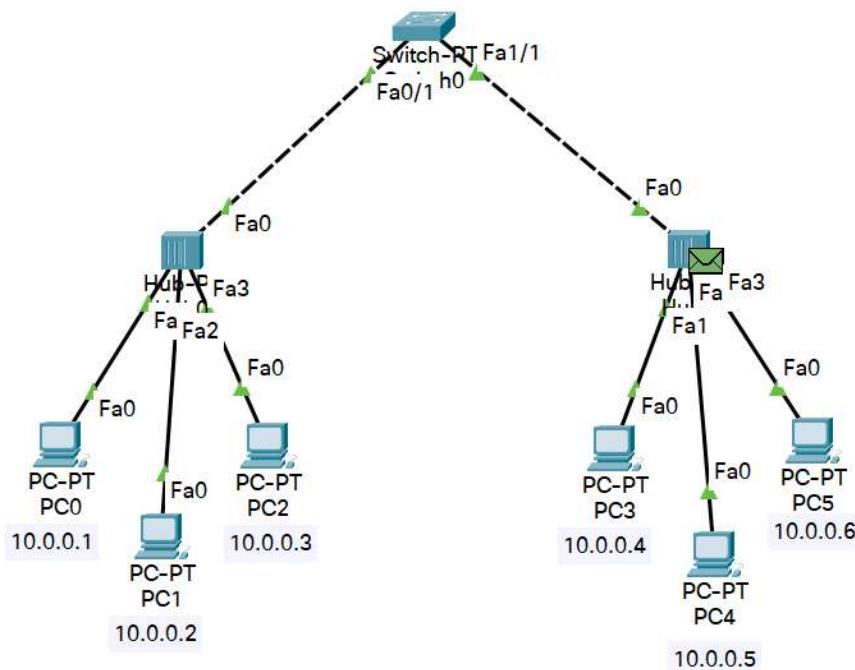


- iii. Screen shots/ output

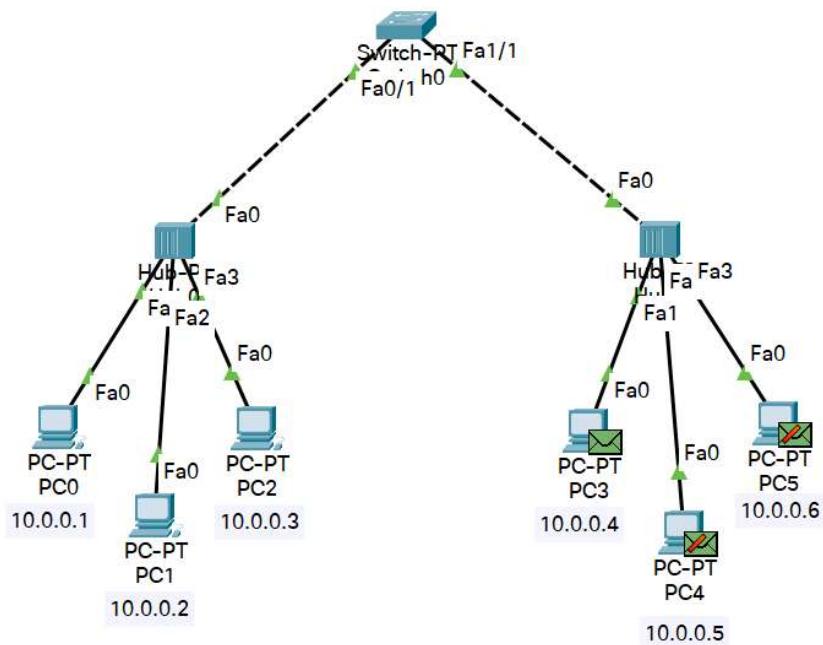
Hub behaviour at sending end



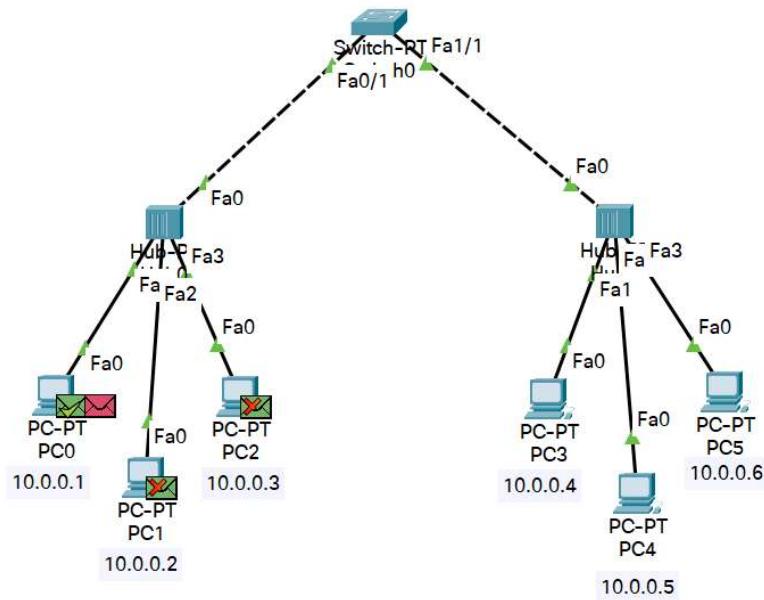
Switch behaviour



Hub behaviour at receiving end



Hub behaviour when back to sender



Ping command to connectivity

```

Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:

Reply from 10.0.0.4: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>

```

iv. Observation

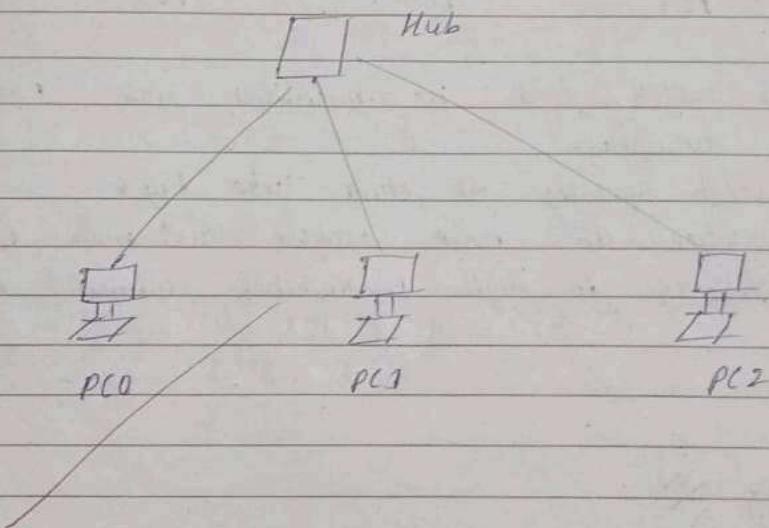
LAB - 1

(Creating topology and simulate simple PDU from source to destination using hub and switch as connecting devices)

Observations:-

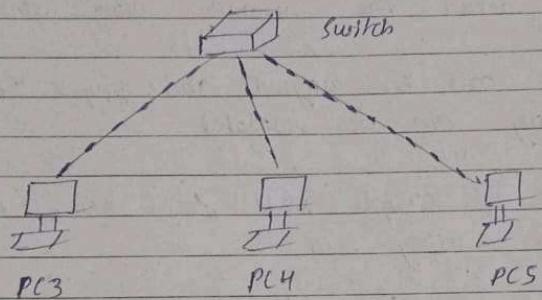
- We can easily visualize complex network topologies with different devices like routers, switches, PCs and servers connected through various interfaces
- Packet tracer allows us to observe the flow of packets across the network using simulation mode
- Subnets can be assigned, and proper IP addressing can be validated

Simulation of simple PDU from source to destination using hub



- In hub data is transmitted to all devices connecting to hub from source
- Hub operates at physical layer of OSI model
- Hubs are simpler and cheaper
- But efficiency is low in hub compared to switch

④ Simulation of simple PDU from source to destination using switch

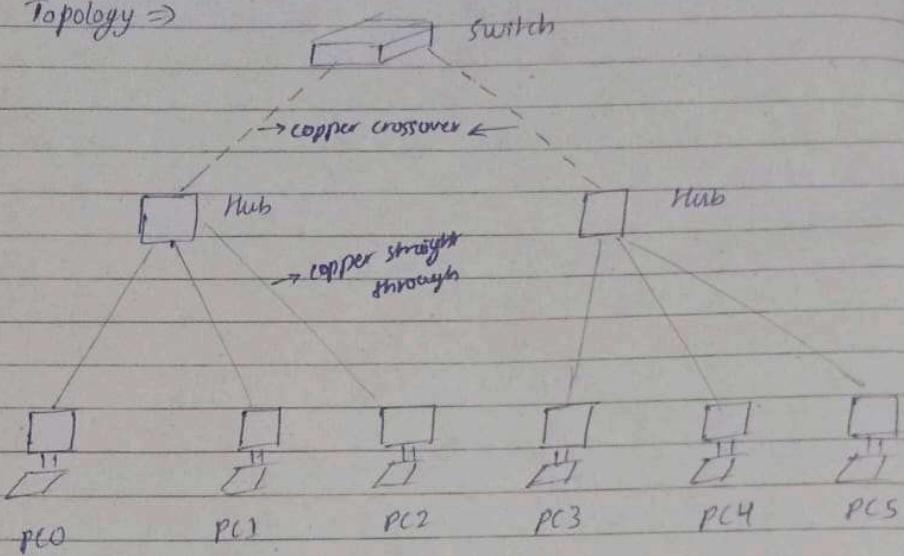


- In switch data is transmitted from source to destination
- switch operates at data link layer
- switches are more complex and more expensive
- Efficiency is high in switches compared to hub

~~✓ 27/0~~

Create a topology using multiple hub and switch connecting them to a simple PDU

Topology \Rightarrow



Aim \Rightarrow To obtain topology using hub and switch

Steps \Rightarrow

1) Drag and drop PCs, and hubs and switches and establish ~~above~~ as show above

2) Configure each PC

\rightarrow click on PC

\rightarrow Go to config

\rightarrow Go to fast ethernet and give unique IP address

3) Select simulation

\rightarrow click message

\rightarrow later, click source and destination

10.0.0.0

1) Running the ping command

- click on PC
- go to desktop
- go to command prompt
- Run command ping 10.0.0.3

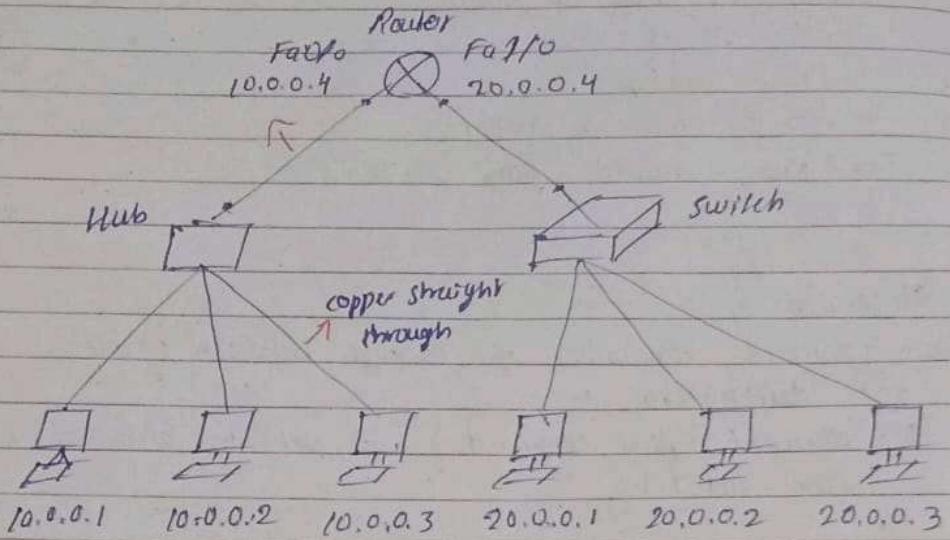
Observation:-

- observed simulation of message between source and destination e
- Executed ping command and get to know about bytes and TTL

Reply Bytes

0410

- pc > ping 10.0.0.4
- pinging 10.0.0.4 with 32 bytes of data
- request timed out
- Reply from 10.0.0.4: bytes=32 time=22ms



Configure routers

Router > enable

Router # config t

Router > interface fastethernet 0/0

Router > ip address 10.0.0.1 255.0.0.0

Router > no shutdown

Router > exit

→ Configure Hub connected PC's with IP address of 10.0.0.1 and so on.

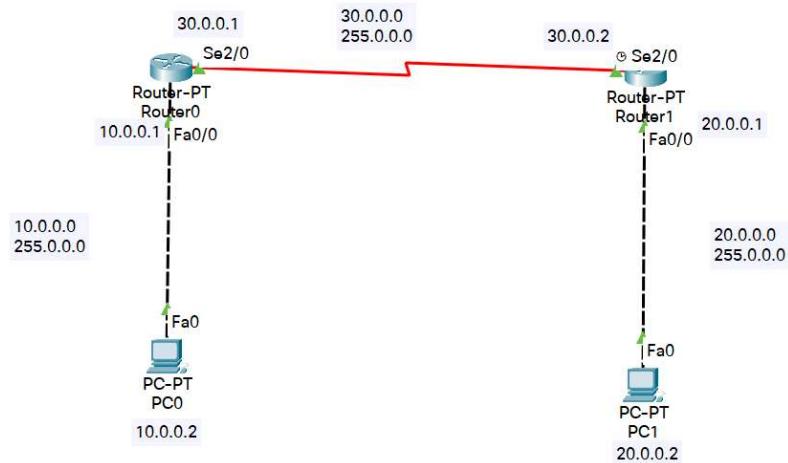
→ Configure switch connected PC's with IP address of 20.0.0.2 and so on

→ Set the gateway for PC's on hub network as 10.0.0.1

→ Set the gateway for PC's on switch network as 20.0.0.4

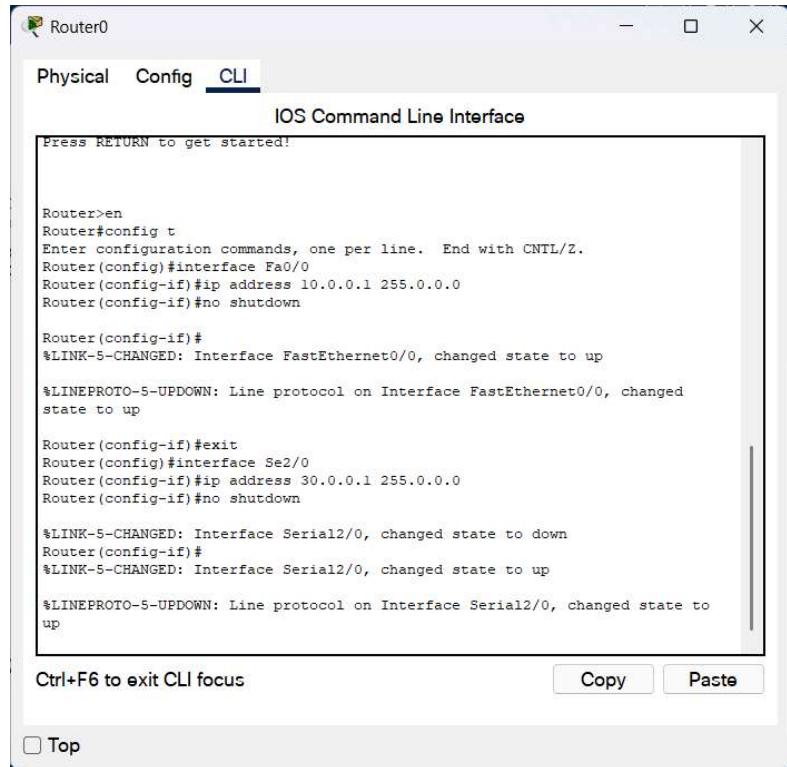
Program 2

- i. Create a topology involving multiple hubs and a switch connecting them to simulate with simple PDU.
- ii. Procedure along with the topology



- iii. Screen shots/ output

Router0 configuration



The screenshot shows the Cisco IOS Command Line Interface (CLI) for Router0. The window title is "Router0". The tab bar at the top has three tabs: "Physical", "Config", and "CLI", with "CLI" being the active tab. Below the tabs is a banner that says "IOS Command Line Interface" and "Press RETURN to get started!". The main text area displays the configuration commands entered by the user:

```
Router>en
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface Fa0/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shutdown

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

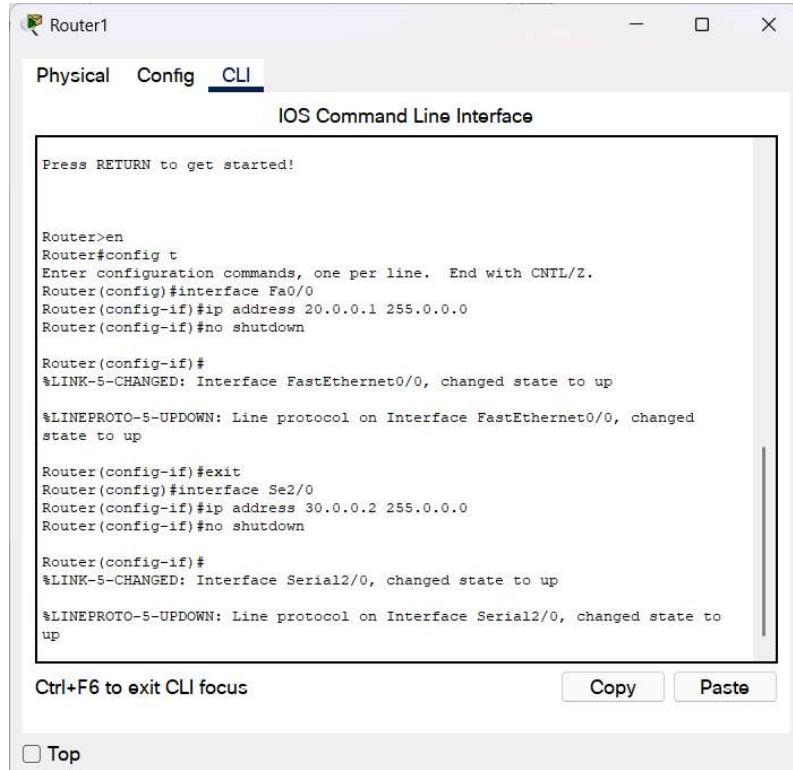
Router(config-if)#exit
Router(config)#interface Se2/0
Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial2/0, changed state to down
Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

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

At the bottom of the text area, there is a message "Ctrl+F6 to exit CLI focus" and two buttons: "Copy" and "Paste". There is also a checkbox labeled "Top".

Router1 configuration



The screenshot shows the Cisco IOS Command Line Interface (CLI) for Router1. The window title is "Router1". The tab bar at the top has three tabs: "Physical", "Config", and "CLI", with "CLI" being the active tab. Below the tabs is a banner that says "IOS Command Line Interface" and "Press RETURN to get started!". The main text area displays the configuration commands entered by the user:

```
Router>en
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface Fa0/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#no shutdown

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

Router(config-if)#exit
Router(config)#interface Se2/0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

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

At the bottom of the text area, there is a message "Ctrl+F6 to exit CLI focus" and two buttons: "Copy" and "Paste". There is also a checkbox labeled "Top".

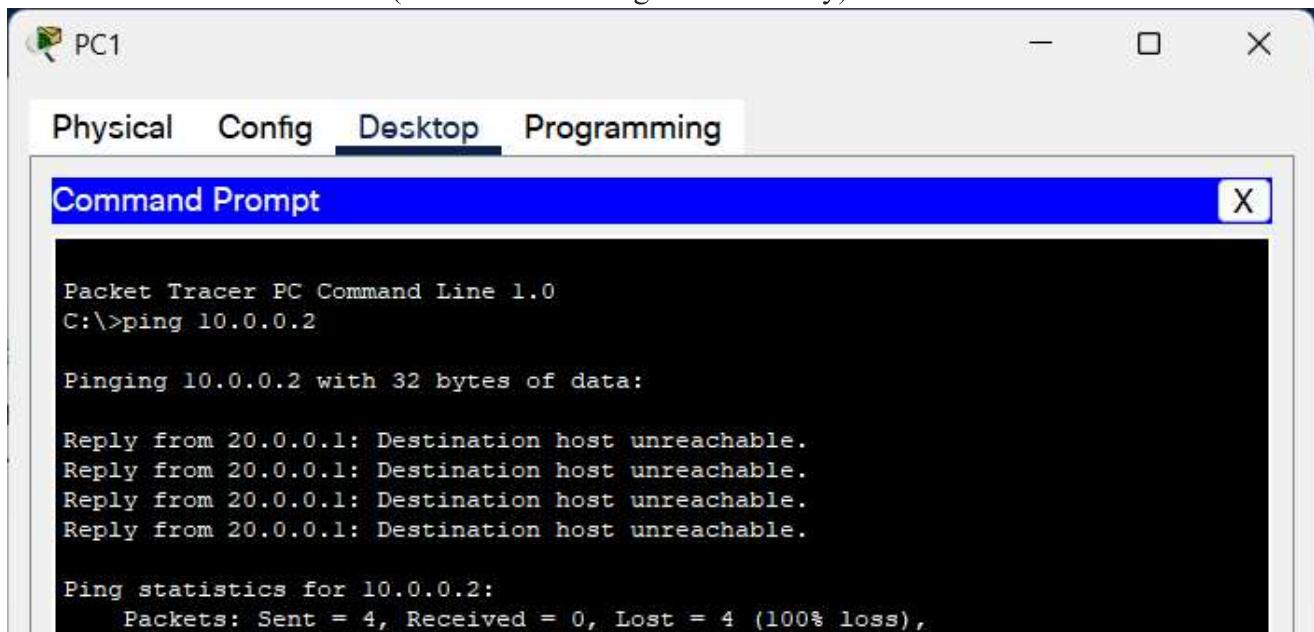
Ip route command in Router0

```
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 20.0.0.0 255.0.0.0 30.0.0.2
Router(config)#+
```

Ip route command in Router1

```
Router#config t
Enter configuration commands, one per line. End with CNTL/Z
Router(config)#ip route 10.0.0.0 255.0.0.0 30.0.0.1
Router(config)#+
```

Destination host Unreachable (Before establishing network Fully)



Request Timed Out

PC2

Physical Config Desktop Programming

Command Prompt X

```
Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.2

Pinging 20.0.0.2 with 32 bytes of data:

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

Ping statistics for 20.0.0.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Reply from Destination

PC0

Physical Config Desktop Programming

Command Prompt X

```
Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.2

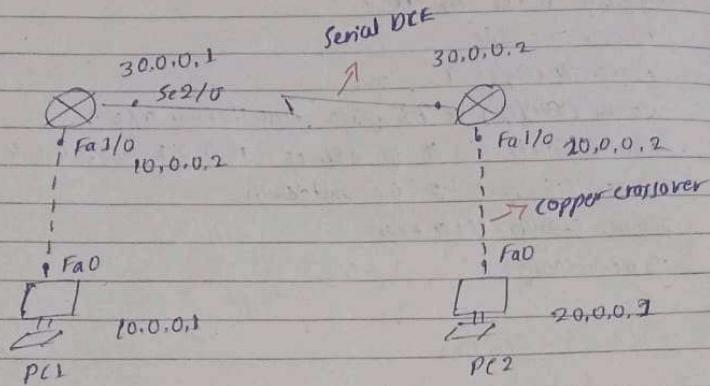
Pinging 20.0.0.2 with 32 bytes of data:

Reply from 20.0.0.2: bytes=32 time=1ms TTL=126
Reply from 20.0.0.2: bytes=32 time=18ms TTL=126
Reply from 20.0.0.2: bytes=32 time=1ms TTL=126
Reply from 20.0.0.2: bytes=32 time=1ms TTL=126

Ping statistics for 20.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 18ms, Average = 5ms
```

iv. Observation

LAB-3 Create topology to connect two three networks using two routers



- 1) Assign ip addresses for end devices and routers
- 2) connect router to pc with fastethernet 0/0 or fastethernet 1/0. which
- 3) Two routers are connected through serial 2/0 or serial 3/0
- 4) We have to give gateway to both pc
- 5) ~~We have~~ Enable both routers by enabling setting port status to ON mode or by providing commands on CLI.
- 6) view the routers with command ip show route
- 7) Ping the destination ip address from source PC.

Example of connecting two routers

Router > enable

Router # config t

Router (config) # interface Serial 2/0

Router (config-if) # ip address 30.0.0.1 255.0.0.0

Router (config-if) # no shutdown

Router (config-if) # exit

Router (config) # ip route 0.0.0.0 255.0.0.0 30.0.0.2

Ping command for not providing gateway

PC > ping 20.0.0.0

pinging 20.0.0.0 with 32 bytes of data

Request timed out.

Ping command for not providing destination ip address

PC > ping 20.0.0.1

pinging 20.0.0.0 with 32 bytes of data

Reply from 20.0.0.0 Destination not reachable

~~Ping command for correct configuration~~

PC > ping 20.0.0.1

pinging 20.0.0.1 with 32 bytes of data

Reply from 20.0.0.1 bytes=32 time=11ms TTL=126

Reply from 20.0.0.1 bytes=32 time=10ms TTL=126

Configure ip address to routers and packet tracer
explore the following test

i) ping responses

- i) Request timed out ✓
- ii) destination not reachable ✓
- iii) Reply

ip route

8/25/10

i) Request timed out ⇒

⇒ ip route is configured but destination ip
address is incorrect

ii) Destination not reachable ⇒

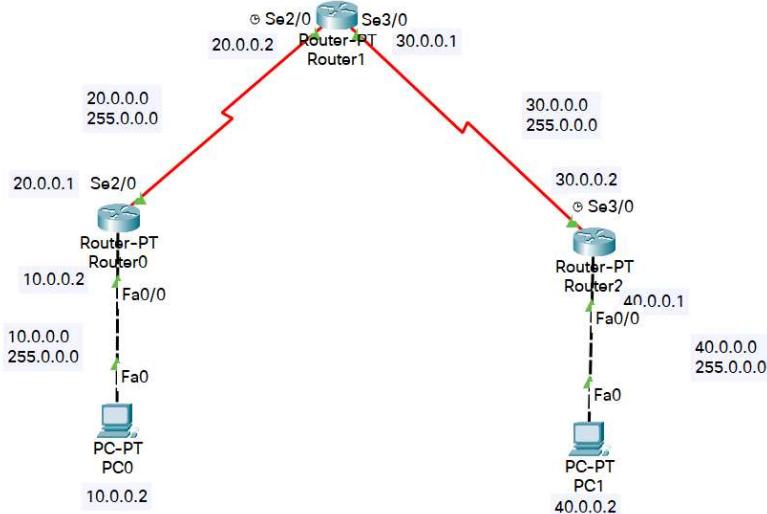
⇒ ip route is not configured

iii) Reply ⇒

⇒ ip route is configured and correct ip address
of destination is provided

Program 3

- i. Configure default route, static route to the router
- ii. Procedure along with the topology



- iii. Screen shots/ output

Router0 configuration

```
Router>en
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface Fa0/0
Router(config-if)#ip address 10.0.0.2 255.0.0.0
Router(config-if)#no shutdown

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
%IP-4-DUPADDR: Duplicate address 10.0.0.2 on FastEthernet0/0, sourced by 000C.CFC2.65B0

Router(config-if)#exit
Router(config)#interface Se2/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial2/0, changed state to down
Router(config-if)#

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

Press RETURN to get started!
```

Ctrl+F6 to exit CLI focus

Top

Router1 configuration

The screenshot shows the Router1 CLI interface. The title bar says "Router1". Below it, there are three tabs: "Physical", "Config", and "CLI", with "CLI" being the active tab. The main window is titled "IOS Command Line Interface" and contains the following text:

```
Press RETURN to get started!

Router>enable
Router>config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface Se2/0
Router(config-if)#ip address 20.0.0.2 255.0.0.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

Router(config-if)#exit
Router(config)#interface Se
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to
up
3/0
Router(config-if)#ip address 30.0.0.1 255.0
^
% Invalid input detected at '^' marker.

Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial3/0, changed state to down
Router(config-if)#

```

At the bottom of the window, there are buttons for "Copy" and "Paste". Below the window, there is a status bar with the text "Ctrl+F6 to exit CLI focus" and a checkbox labeled "Top".

Router2 configuration

Router2

Physical Config **CLI**

IOS Command Line Interface

```

Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface Se3/0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface Serial3/0, changed state to up

Router(config-if)#exit
Router(config)#inter
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to
up
face Fa0/0
Router(config-if)#ip address 40.0.0.1 255.0.0.0
Router(config-if)#no dhutdoe
          ^
% Invalid input detected at '^' marker.

Router(config-if)#no shutdown

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

```

Ctrl+F6 to exit CLI focus **Copy** **Paste**

Top

Static Routing:

Router0

Router0

Physical Config **CLI**

IOS Command Line Interface

```

Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 30.0.0.0 255.0.0.0 20.0.0.2
Router(config)#ip route 40.0.0.0 255.0.0.0 20.0.0.2
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

C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    20.0.0.0/8 is directly connected, Serial2/0
S    30.0.0.0/8 [1/0] via 20.0.0.2
S    40.0.0.0/8 [1/0] via 20.0.0.2

Router#

```

Ctrl+F6 to exit CLI focus **Copy** **Paste**

Top

Router1

Router1

Physical Config **CLI**

IOS Command Line Interface

```

Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#config t
Enter configuration commands, one per line. End with CNTL/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#

```

Ctrl+F6 to exit CLI focus Top

Router2

Router2

Physical Config **CLI**

IOS Command Line Interface

```

Router#enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 10.0.0.0 255.0.0.0 30.0.0.1
Router(config)#ip route 20.0.0.0 255.0.0.0 30.0.0.1
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 30.0.0.1
S  20.0.0.0/8 [1/0] via 30.0.0.1
C  30.0.0.0/8 is directly connected, Serial3/0
C  40.0.0.0/8 is directly connected, FastEthernet0/0

Router#

```

Ctrl+F6 to exit CLI focus Top

Dynamic Routing:

Route0

Router0

Physical Config CLI

IOS Command Line Interface

```

S 30.0.0.0/8 [1/0] via 20.0.0.2
S 40.0.0.0/8 [1/0] via 20.0.0.2

Router#config t
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-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 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, Serial2/0
S 30.0.0.0/8 [1/0] via 20.0.0.2
S 40.0.0.0/8 [1/0] via 20.0.0.2
S* 0.0.0.0/0 [1/0] via 20.0.0.2

Router#

```

Ctrl+F6 to exit CLI focus

Top

Router2

Router2

Physical Config CLI

IOS Command Line Interface

```

C 30.0.0.0/8 is directly connected, Serial3/0
C 40.0.0.0/8 is directly connected, FastEthernet0/0

Router#config t
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

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

S 10.0.0.0/8 [1/0] via 30.0.0.1
S 20.0.0.0/8 [1/0] via 30.0.0.1
C 30.0.0.0/8 is directly connected, Serial3/0
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#

```

Ctrl+F6 to exit CLI focus

Top

Pinging:

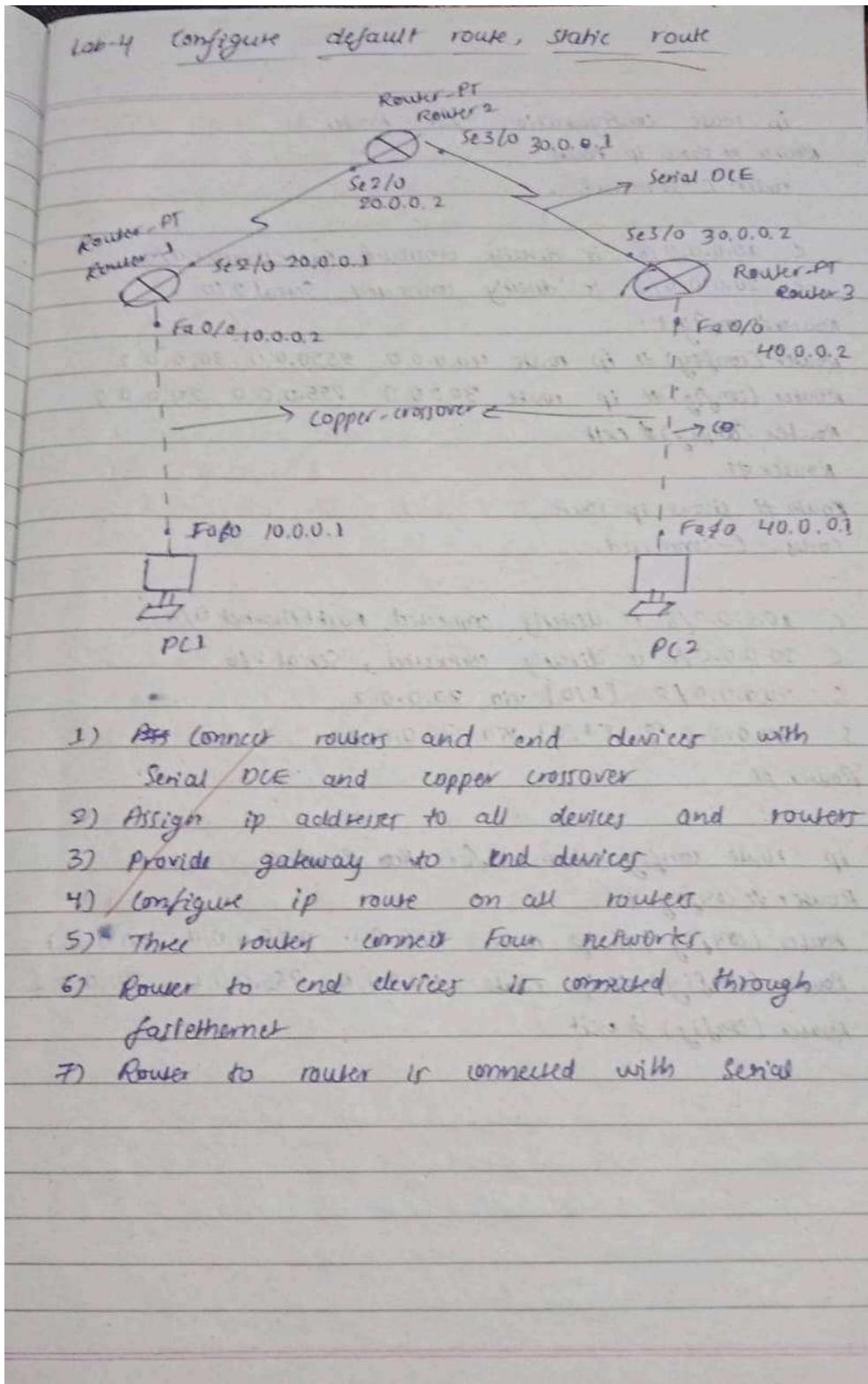
```
C:\>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

Reply from 40.0.0.2: bytes=32 time=21ms TTL=125
Reply from 40.0.0.2: bytes=32 time=17ms TTL=125
Reply from 40.0.0.2: bytes=32 time=25ms TTL=125
Reply from 40.0.0.2: bytes=32 time=2ms TTL=125

Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 25ms, Average = 16ms
```

iv. Observation



ip route configuration for Router 1

Router # show ip route

Codes: C - connected

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

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

Router# config t

Router (config) # ip route 40.0.0.0 255.0.0.0 30.0.0.2

Router (config) # ip route 30.0.0.0 255.0.0.0 20.0.0.2

Router (config) # exit

Router #

Router # show ip route

Codes: C - connected

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 30.0.0.0/8 [2/0] via 20.0.0.2

S 40.0.0.0/8 [3/0] via 30.0.0.2

Router #

ip route configuration for Router 2

Router # config t

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

ip route configuration for Router 3:

Router # config t

Router (config) # ip route 10.0.0.0 255.0.0.0 20.0.0.1

Router (config) # ip route 20.0.0.0 255.0.0.0 30.0.0.1

Default IP

o/p

For default route

ip route configuration for Router 1

Router # config t

Router (config) # ip route 0.0.0.0 0.0.0.0 20.0.0.2

Router (config) # exit

ip route configuration for Router 3

Router # config t

Router (config) # ip route 0.0.0.0 0.0.0.0 30.0.0.1

Router (config) # exit

ip route configuration for Router 2

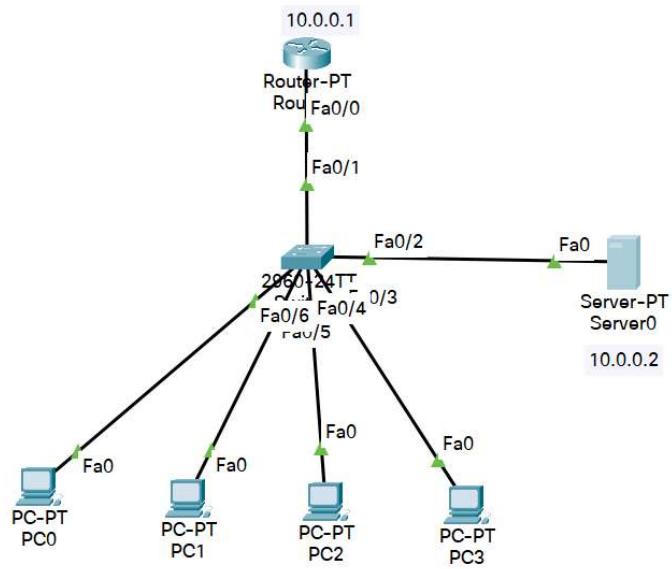
Router # config t

Router (config) # ip route 10.0.0.0 255.0.0.0 20.0.0.1

8/11

Program 4

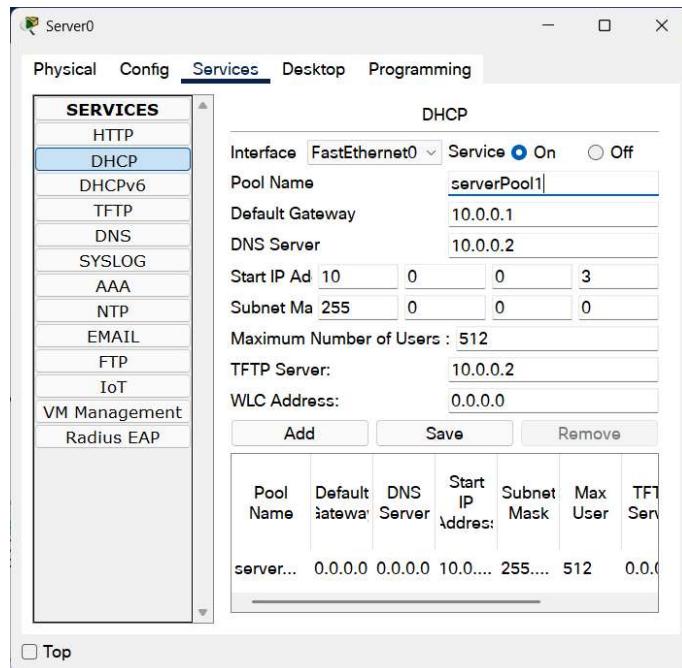
- i. Configure DHCP within a LAN and outside LAN.
- ii. Procedure along with the topology



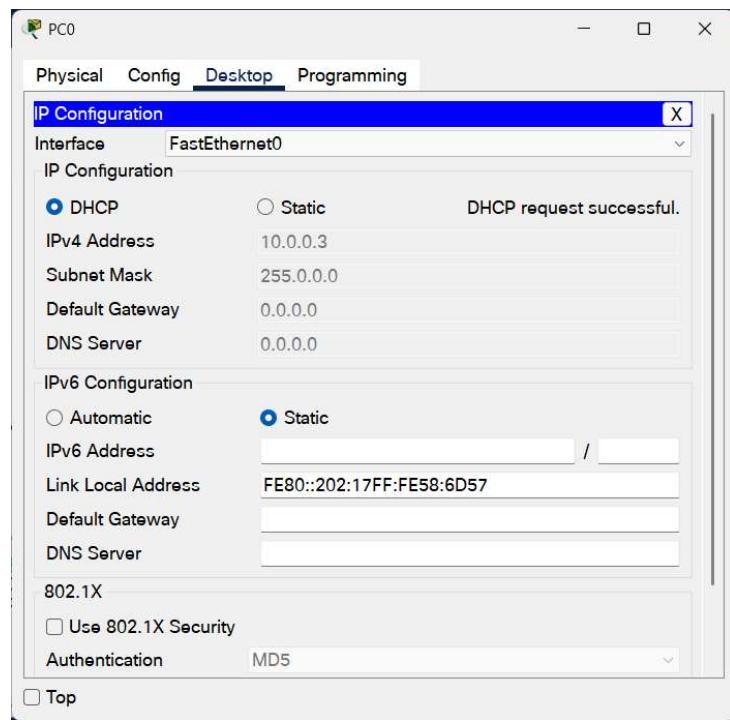
iii. Screen shots/ output

DHCP Within LAN

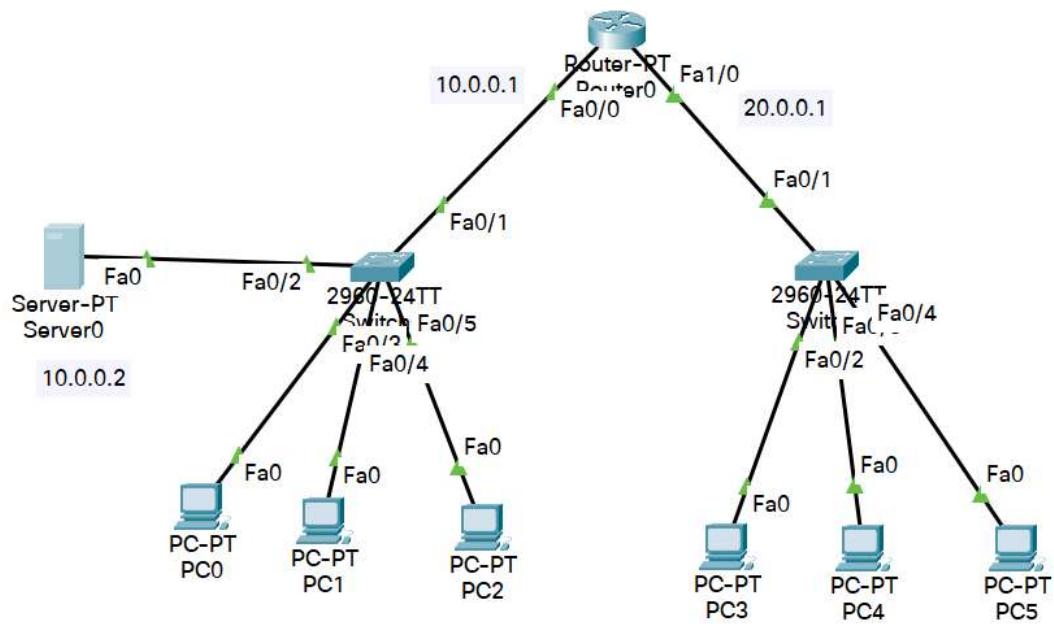
DHCP Configuration



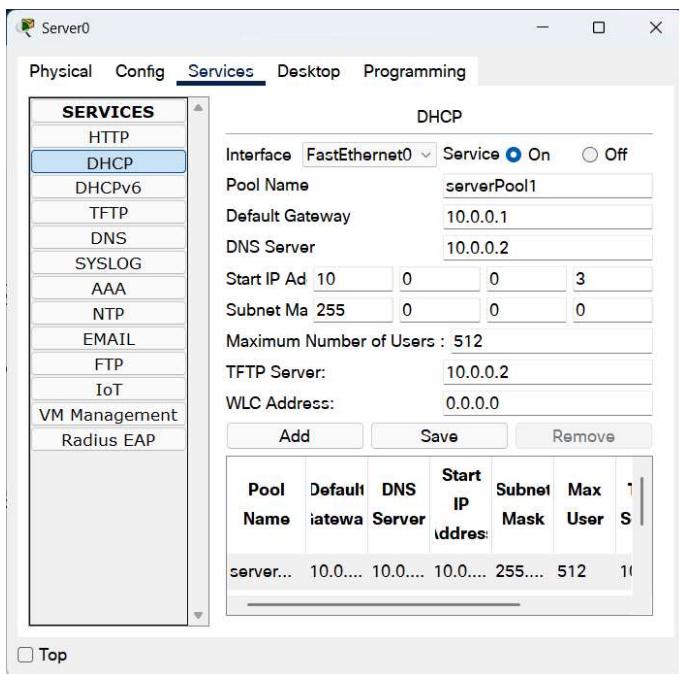
PC settings



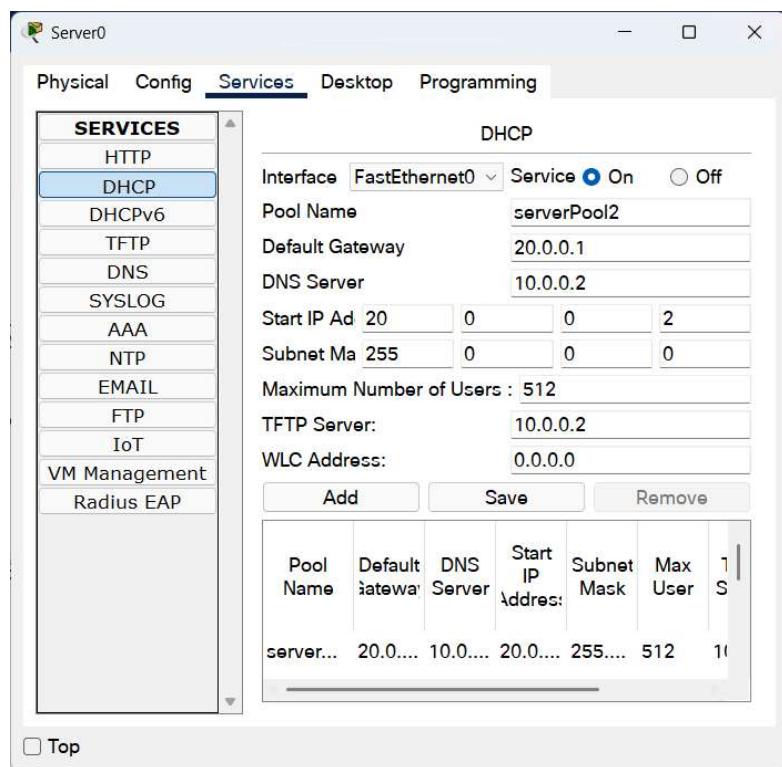
DHCP outside LAN:



DHCP configuration for inside LAN



DHCP configuration for outside LAN

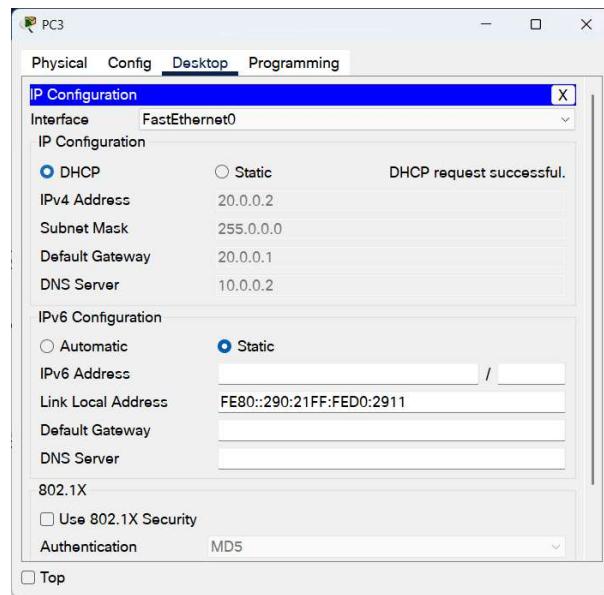


Ip helper command in Router

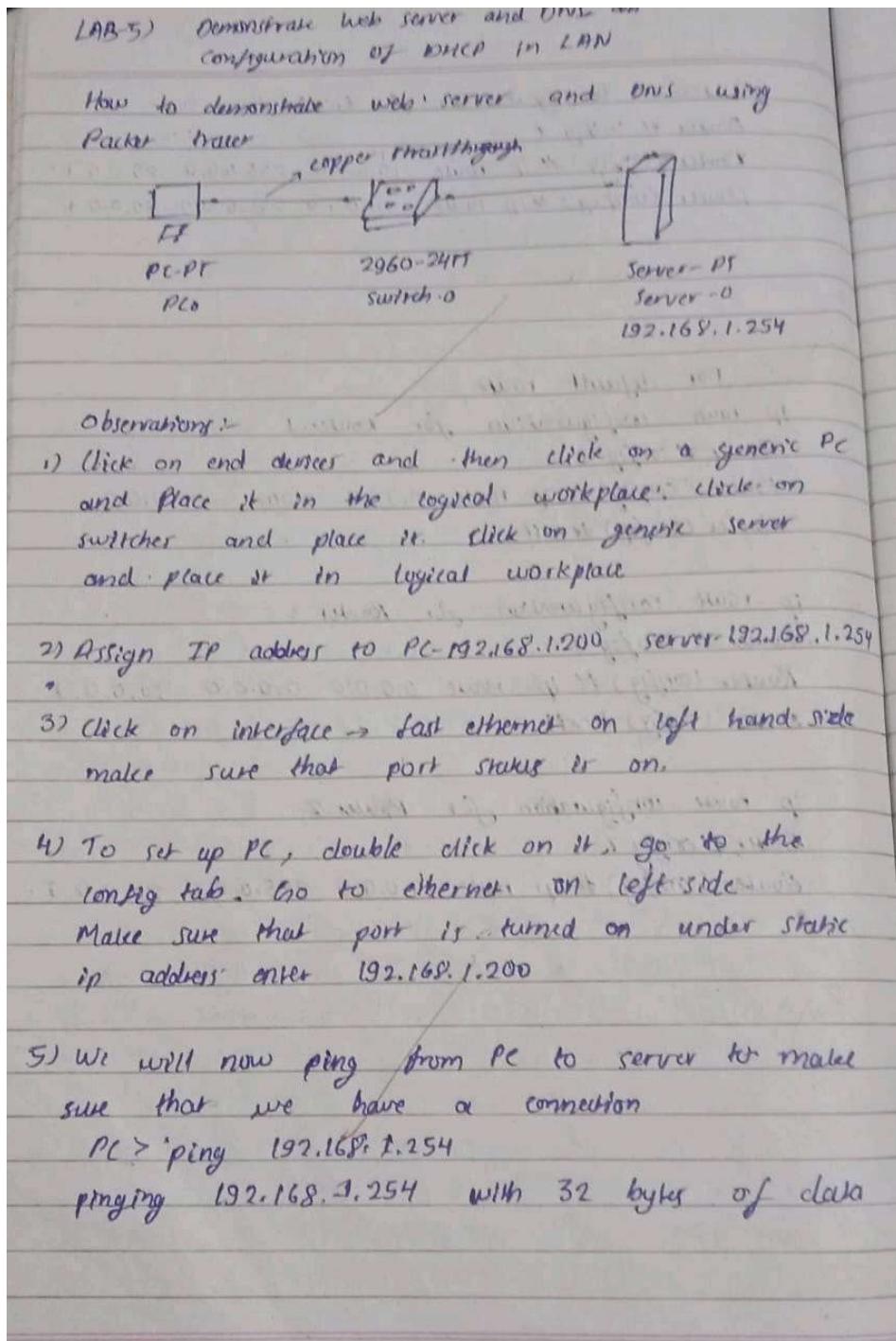
```
Router(config-if)#exit
Router(config)#interface Fa1/0
Router(config-if)#ip helper-address 10.0.0.2
Router(config-if)#

```

PC setting in another network



iv. Observation



Reply from 192.168.1.254: bytes=32 time=1.78ms TLE=128

Reply from 192.168.1.254: bytes=32 time=4ms TLE=128

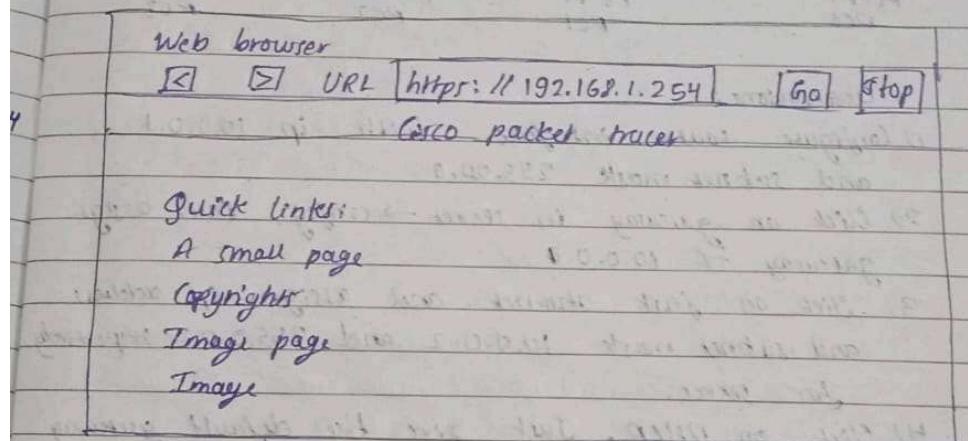
Reply from 192.168.1.254: bytes=32 time=4ms TLE=128

Reply from 192.168.1.254: bytes=32 time=4ms TLE=128

6) Double click on the screen, go to config tab

7) In the virtual browser that appears type in IP address of server and click go

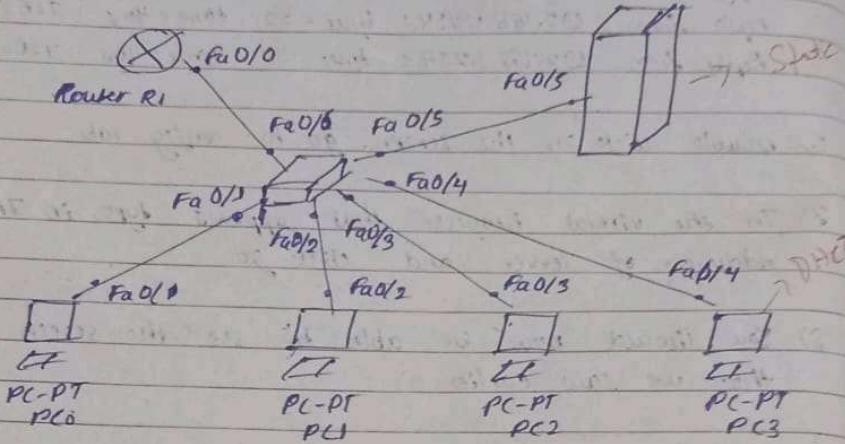
8) You should now be able to see the screen that we saw earlier



You are now connected to FTP server

Record

Q) How to configure DHCP within a LAN in a packet tracer



Observations:

- 1) Configure router interface with ip 10.0.0.1 and subnet mask 255.0.0.0
- 2) Click on gateway in server → config, then assign gateway of 10.0.0.1
- 3) Click on fast ethernet and assign ip address and subnet mask 10.0.0.2 and 255.0.0.0 respectively for server.
- 4) Click on DHCP. Just give the default gateway as 10.0.0.1.
- 5) DNS server, give our server ip address as 10.0.0.2
- 6) Then just edit start ip address, give as 10.0.0.10 and subnet mask 255.0.0.0.
- 7) In maximum number of users just give how many ip address you want in this pool (500)
- 8) Assign IP TFTP server ip address, just give our server ip address 10.0.0.2

g) And click save

10) Now click on any of PC - then click on Desktop → IP configuration and choose DHCP wait for some time if your DHCP request failed then try few more times. This is how you should get

Result

PC0

[Physical] [Config] [Desktop]

IP configuration

① DHCP DHCP request successful

② Static

IP address 10.0.0.14

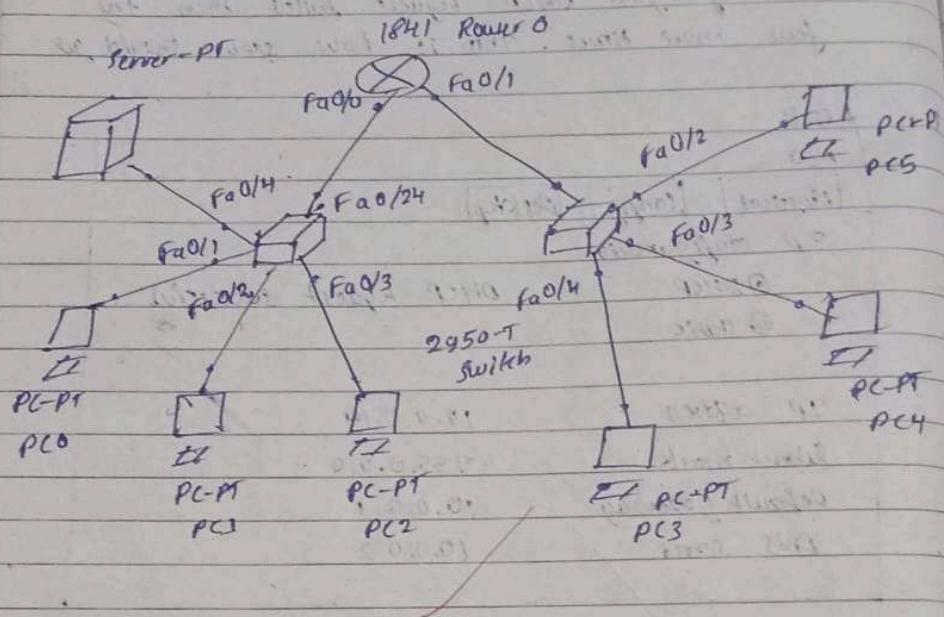
Subnet mask 255.0.0.0

Default gateway 10.0.0.1

DNS server 10.0.0.2

Q) How to get IP from DHCP that is in some other network using IP helper address

Topology:-



Observations:-

- 1) Drag 2 switches, 1 router, 1 server and 4 end devices in logical workplace
- 2) Configure router interface fastethernet 0/0 and fastethernet 0/1 with ip address
- 3) Click on server → Then give the gateway ip address 10.0.0.1
- 4) Then click on fast ethernet assign ip address 10.0.0.2 and subnet mask 255.0.0.0. DHCP will automatically assign IP network for default pool. We don't have to create pool for 10 network again. Just we need to give IP for DNS, gateway and TFTP

then we may configure starting ip address or leave it and save

5) Now click on PC

Xample

5) Now click on PC in a LAN with server and check whether DHCP working fine in this network
click on any pc → Desktop → IP configuration → choose DHCP then you will get ip from DHCP server for this PC

6) Default gateway → 20.0.0.1

DNS server → 20.0.0.2

Start ip address → 20.0.0.10.

Subnet mask → 255.0.0.0

max. number of users → 500

TFTP server → 10.0.0.2

Click on add and save

7) Router (config) # interface fastethernet 0/1

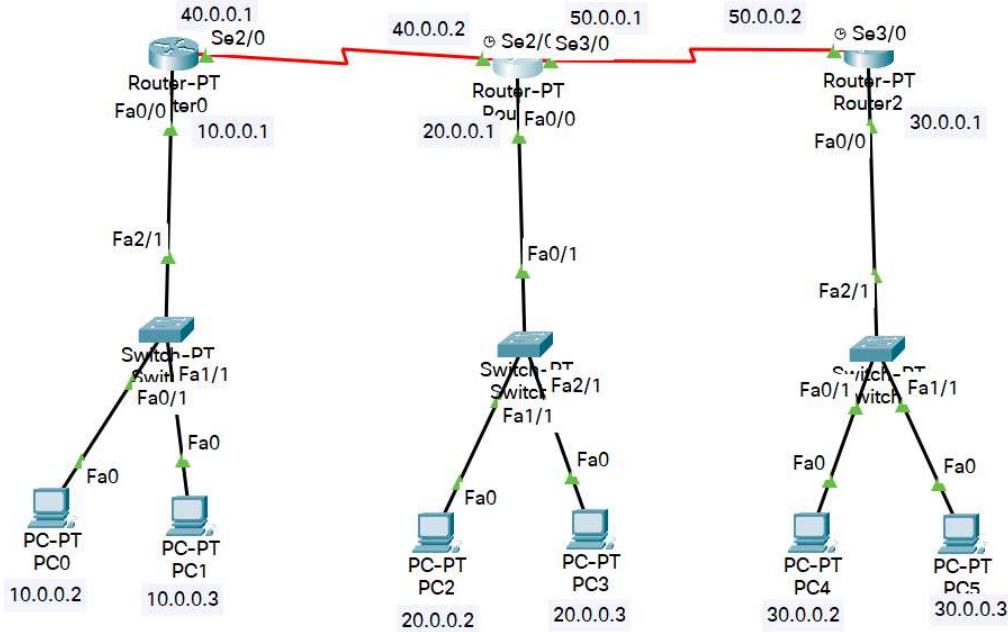
Router (config) # ip help-address 10.0.0.2

Router (config) # exit

8) Click on any pc → Desktop → IP configuration → choose DHCP. Now we get ip address from DHCP server.

Program 5

- i. Configure RIP routing Protocol in Routers
- ii. Procedure along with the topology



- iii. Screen shots/ output

Router0

```

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

C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    40.0.0.0/8 is directly connected, Serial2/0

```

Router1

```

Router#config t
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 50.0.0.0
Router(config-router)#network 20.0.0.0
Router(config-router)#end
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

R    10.0.0.0/8 [120/1] via 40.0.0.1, 00:00:08, Serial2/0
C    20.0.0.0/8 is directly connected, FastEthernet0/0
R    30.0.0.0/8 [120/1] via 50.0.0.2, 00:00:10, Serial3/0
C    40.0.0.0/8 is directly connected, Serial2/0
C    50.0.0.0/8 is directly connected, Serial3/0

```

Router2

```

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

R    10.0.0.0/8 [120/2] via 50.0.0.1, 00:00:28, Serial3/0
R    20.0.0.0/8 [120/1] via 50.0.0.1, 00:00:28, 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, 00:00:28, Serial3/0
C    50.0.0.0/8 is directly connected, Serial3/0

```

Pinging:

```

Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.2

Pinging 20.0.0.2 with 32 bytes of data:

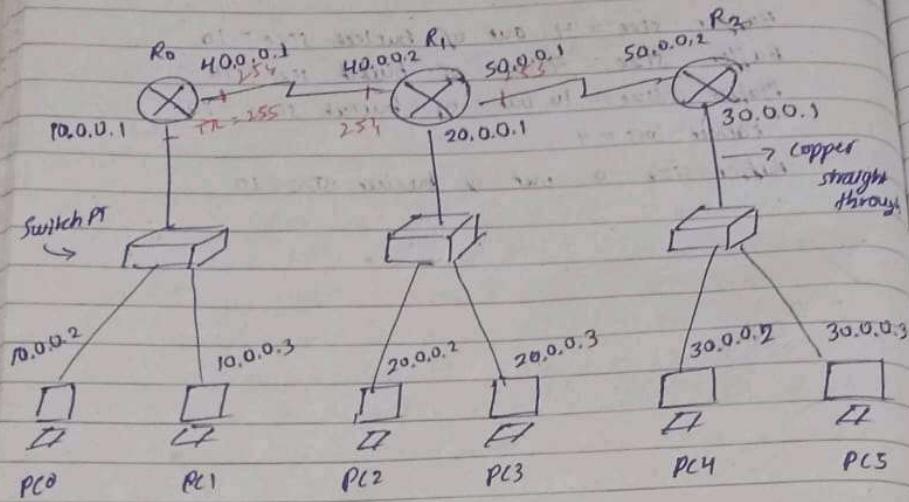
Request timed out.
Reply from 20.0.0.2: bytes=32 time=9ms TTL=126
Reply from 20.0.0.2: bytes=32 time=1ms TTL=126
Reply from 20.0.0.2: bytes=32 time=9ms TTL=126

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

```

iv. Observation

LAB-7 Routing Information Protocol



Observations :-

- 1) Pick up three routers, 3 switches and 6 PCs and drag them to connect as shown above
- 2) Give ip address and gateway to all PCs configure ip addresses for all three routers by first turning on port status to know which is the connection.
- 3) Router to router is connected with Serial DCE
Router to switch is connected with copper straight through.

Click router and go to CLI enable the router
and configure the terminal

For Router R0

Router > enable

Router # config t

Router (config) # router rip

Router (config-router) # network 20.0.0.0

Router (config-router) # network 40.0.0.0

Router (config-router) # exit

Router (config) # exit

Router # show ip route

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

c 40.0.0.0/8 is directly connected, serial 2/0

For Router R1

Router > enable

Router # config t

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) # exit

For Router R2

Router > enable

Router # config t

Router (config) # Router rip

Router (config-router) # network 50.0.0.0

Router (config-router) # network 30.0.0.0

Router (config-router) # exit

Click router and go to CLI enable the router
and configure the terminal

For Router R₀

Router > enable

Router # config t

Router (config) # router rip

Router (config-router) # network 10.0.0.0

Router (config-router) # network 40.0.0.0

Router (config-router) # exit

Router (config) # exit

Router # show ip routes

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

C 40.0.0.0/8 is directly connected, Serial 2/0

For Router R₁

Router > enable

Router # config t

Router (config) # Router IP. 10.0.1.1 255.0.0.0

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) # exit

For Router R₂

Router > enable

Router # config t

Router (config) # Router rip

Router (config-router) # network 50.0.0.0

Router (config-router) # network 30.0.0.0

Router (config-router) # exit

Ringing PC5 from PC0

PC > ping 30.0.0.3

Pinging 30.0.0.3 with 32 bytes of data:

Reply from 30.0.0.3: bytes=32 time=3ms TTL=125

Reply from 30.0.0.3: bytes=32 time=2ms TTL=125

Reply from 30.0.0.3: bytes=32 time=9ms 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 = 4, lost = 0 (0% loss).

Approximate round trip times in milliseconds:

minimum = 2ms, Maximum = 9ms Average = 4ms

For Router R0

Router# show ip route

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

R 20.0.0.0/8 [120/1] via 40.0.0.1, 00:00:07, Serial 2/0

R 30.0.0.0/8 [120/2] via 40.0.0.2, 00:00:07, Serial 2/0

c 40.0.0.0/8 is directly connected, Serial 2/0

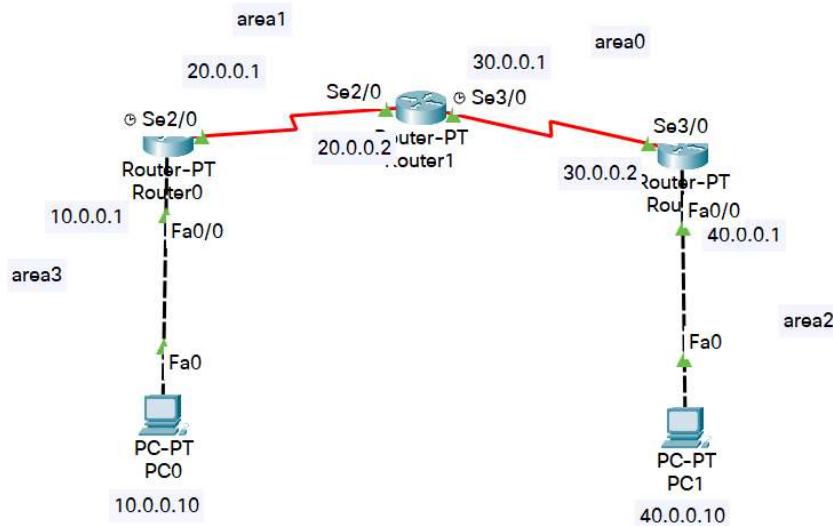
R 50.0.0.0/8 [120/1] via 40.0.0.2, 00:00:07, Serial 2/0

Router#

✓ 22/11

Program 6

- i. Configure OSPF routing protocol
- ii. Procedure along with the topology



- iii. Screen shots/ output

Encapsulation:

Router0

```
Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface Fa0/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shutdown

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

Router(config-if)#exit
Router(config)#interface Se2/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#clock rate 64000
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial2/0, changed state to down
Router(config-if)#exit
Router(config)#

```

Router1

```
Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface Se2/0
Router(config-if)#ip address 20.0.0.2 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

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

Router(config-if)#exit
Router(config)#interface Se3/0
Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#clock rate 64000
Router(config-if)#no shutdown
^
% Invalid input detected at '^' marker.

Router(config-if)#no shutdown

```

Router2

```

Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface Fa0/0
Router(config-if)#ip address 40.0.0.1 255.0.0.0
Router(config-if)#no shutdown

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
%IP-4-DUPADDR: Duplicate address 40.0.0.1 on FastEthernet0/0, sourced by
000D.BDDA.0123

Router(config-if)#exit
Router(config)#interface Se3/0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface Serial3/0, changed state to up

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

```

OSPF Routing Protocol

Router0

```

Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#router-id 1.1.1.1
Router(config-router)#network 10.0.0.0 0.255.255.255 area 3
Router(config-router)#network 20.0.0.0 0.255.255.255 area 1
Router(config-router)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#sho
00:27:19: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Serial2/0 from LOADING to FULL, Loading Done
w 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
      20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        20.0.0.0/8 is directly connected, Serial2/0
C        20.0.0.2/32 is directly connected, Serial2/0
O  IA 30.0.0.0/8 [110/128] via 20.0.0.2, 00:00:02, Serial2/0
O  IA 40.0.0.0/8 [110/129] via 20.0.0.2, 00:00:02, Serial2/0

```

Router1

```
Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#router-id 2.2.2.2
Router(config-router)#network 20.0.0.0 0.255.255.255 area 1
Router(config-router)#network 30.0.0.0 0.255.255.255 area 0
Router(config-router)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

00:26:21: %OSPF-5-ADJCHG: Process 1, Nbr 3.3.3.3 on Serial3/0 from LOADING to FULL, Loading Done
00:27:18: %OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on Serial2/0 from LOADING to FULL, Loading Done

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

 20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       20.0.0.0/8 is directly connected, Serial2/0
C       20.0.0.1/32 is directly connected, Serial2/0
      30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       30.0.0.0/8 is directly connected, Serial3/0
C       30.0.0.2/32 is directly connected, Serial3/0
O IA 40.0.0.0/8 [110/65] via 30.0.0.2, 00:02:00, Serial3/0
```

Router2

```
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#router-id 3.3.3.3
Router(config-router)#network 40.0.0.0 0.255.255.255 area 2
Router(config-router)#network 30.0.0.0 0.255.255.255 area 0
Router(config-router)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#
00:26:19: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Serial3/0 from LOADING to FULL, Loading Done

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

O IA 20.0.0.0/8 [110/128] via 30.0.0.1, 00:02:45, Serial3/0
      30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       30.0.0.0/8 is directly connected, Serial3/0
C       30.0.0.1/32 is directly connected, Serial3/0
C       40.0.0.0/8 is directly connected, FastEthernet0/0
```

Configure Loopback address

Router0

```
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface loopback 0

Router(config-if)#
%LINK-5-CHANGED: Interface Loopback0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to
up

Router(config-if)#ip address 172.16.1.252 255.255.0.0
Router(config-if)#no shutdown
```

Router1

```
Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface loopback 0

Router(config-if)#
%LINK-5-CHANGED: Interface Loopback0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to
up

Router(config-if)#ip address 172.16.1.253 255.255.0.0
Router(config-if)#no shutdown
Router(config-if)#

```

Router2

```
Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface loopback 0

Router(config-if)#
%LINK-5-CHANGED: Interface Loopback0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to
up

Router(config-if)#ip address 172.16.1.254 255.255.0.0
Router(config-if)#no shutdown
Router(config-if)#

```

Create Virtual Link

Router0

```
Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#area 1 virtual-link 2.2.2.2
Router(config-router)#

```

Router1

```

Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
01:11:01: %OSPF-4-ERRRCV: Received invalid packet: mismatch area ID, from
backbone area must be virtual-link but not found from 20.0.0.2, Serial2/0

01:11:11: %OSPF-4-ERRRCV: Received invalid packet: mismatch area ID, from
backbone area must be virtual-link but not found from 20.0.0.2, Serial2/0

Router(config)#route
01:11:21: %OSPF-4-ERRRCV: Received invalid packet: mismatch area ID, from
backbone area must be virtual-link but not found from 20.0.0.2, Serial2/0
r ospf 1
Router(config-router)#
01:11:31: %OSPF-4-ERRRCV: Received invalid packet: mismatch area ID, from
backbone area must be virtual-link but not found from 20.0.0.2, Serial2/0

Router(config-router)#area 1 v
01:11:41: %OSPF-4-ERRRCV: Received invalid packet: mismatch area ID, from
backbone area must be virtual-link but not found from 20.0.0.2, Serial2/0
irtual-link 1.1.1.1
Router(config-router)#
01:11:56: %OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on OSPF_VL0 from LOADING to
FULL, Loading Done

```

Pinging

```

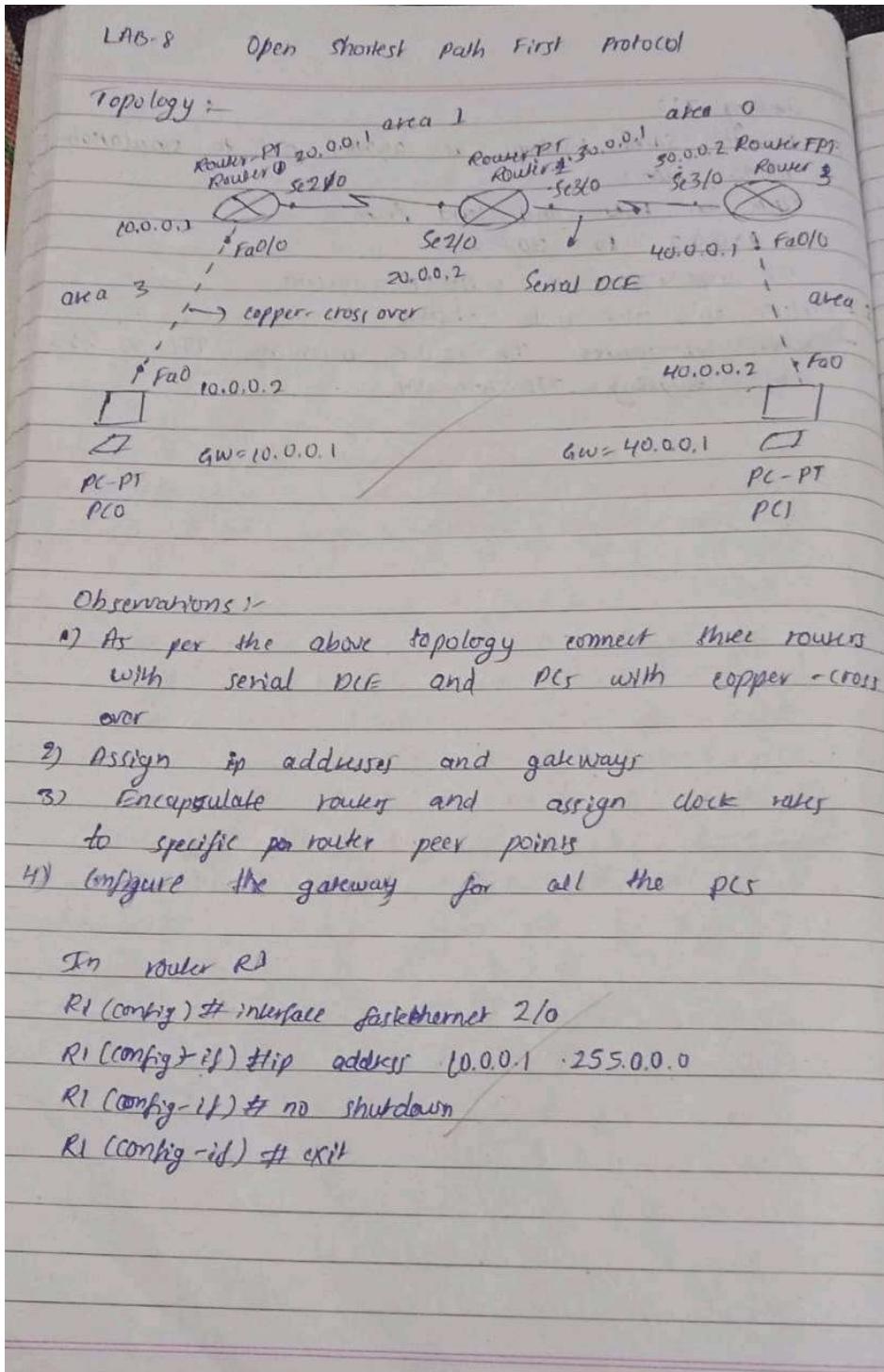
C:\>ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data:
Reply from 40.0.0.10: bytes=32 time=24ms TTL=125
Reply from 40.0.0.10: bytes=32 time=18ms TTL=125
Reply from 40.0.0.10: bytes=32 time=18ms TTL=125
Reply from 40.0.0.10: bytes=32 time=20ms TTL=125

Ping statistics for 40.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 18ms, Maximum = 24ms, Average = 20ms

```

iv. Observation



R1(config)# interface serial 1/0
R1(config-if)# ip address 20.0.0.1 255.0.0.0
R1(config-if)# encapsulation ppp
R1(config-if)# clock rate 64000
R1(config-if)# no shutdown
R1(config-if)# exit

Router 2 : configure serial Se2/0 → ip 20.0.0.2 & Se3/0
Router 3 : configure Fastethernet 0/0 → ip 40.0.0.1 30.0.0.1

Configuring OSPF routing protocol

Router R1

R1(config)# router ospf 1
R1(config-router)# router-id 1.1.1.1
R1(config-router)# network 10.0.0.0 0.255.255.255 area 0
R1(config-router)# network 20.0.0.0 0.255.255.255 area 1
R1(config-router)# exit

Router R2

Router(config)# router ospf 2
Router(config-router)# router-id 2.2.2.2
Router(config-router)# network 20.0.0.0 0.255.255.255 area 1
Router(config-router)# network 30.0.0.0 0.255.255.255 area 2

Router R3

Router(config)# router ospf 3
Router(config-router)# router-id 3.3.3.3
Router(config-router)# network 30.0.0.0 0.255.255.255 area 0
Router(config-router)# network 40.0.0.0 0.255.255.255 area 2

Routing Table R1:

Router # show ip route

Gateway at last resort is not set

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

20.0.0.0/8 is ~~remotely~~ subnetted, 2 subnets, 2 masks

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

C 20.0.0.0/32 is directly connected, Serial 2/0

O IA 30.0.0.0/8 [110/128] via 20.0.0.2, 00:01:51, Serial 2/0

O IA 40.0.0.0/8 [110/128] via 20.0.0.2, 00:00:04, Serial 2/0

Similarly for R2 and R3

Configure loopback address

R1 =>

Router (config)# interface loopback 0

Router (config-if) # ip add 172.16.1.252 255.255.0.0

Router (config-if) # no shutdown

R2 =>

Router (config)# interface loopback 0

Router (config-if) # ip add 172.16.1.253 255.255.0.0

Router (config-if) # no shutdown

R3 =>

Router (config)# interface loopback 0

Router (config-if) # ip add 172.16.1.254 255.255.0.0

Router (config-if) # no shutdown

Router # show ip route

- C 10.0.0.1/8 is directly connected, FastEthernet 0/0
- C 20.0.0.0/8 is variably connected, 2 subnets, 2 masks
- C 20.0.0.1/15 is directly connected, serial 2/0
- C 20.0.0.182/24 is directly connected, serial 2/0
- S 0 30.0.0.0/8 [110/128] via 20.0.0.2, 00:11:11 Serial 2/0
- S 0 40.0.0.0/8 [110/128] via 20.0.0.2, 00:00:04 Serial 2/0
- C 172.16.0.0/16 is directly connected, loopback 0

Virtual link

R1 →

Router (config) # router ospf 1

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

R2 →

Router (config) # router ospf 2

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

Ping from 10.0.0.10 to 40.0.0.10

PC > ping 40.0.0.10

Pinging 40.0.0.1 with 32 bytes of data
Request timed out.

Reply from 40.0.0.1 : bytes=32 time=16ms TTL=128

Reply from 40.0.0.11 : bytes=32 time=2ms TTL=128

Reply from 40.0.0.1 : bytes=32 time=2ms TTL=128

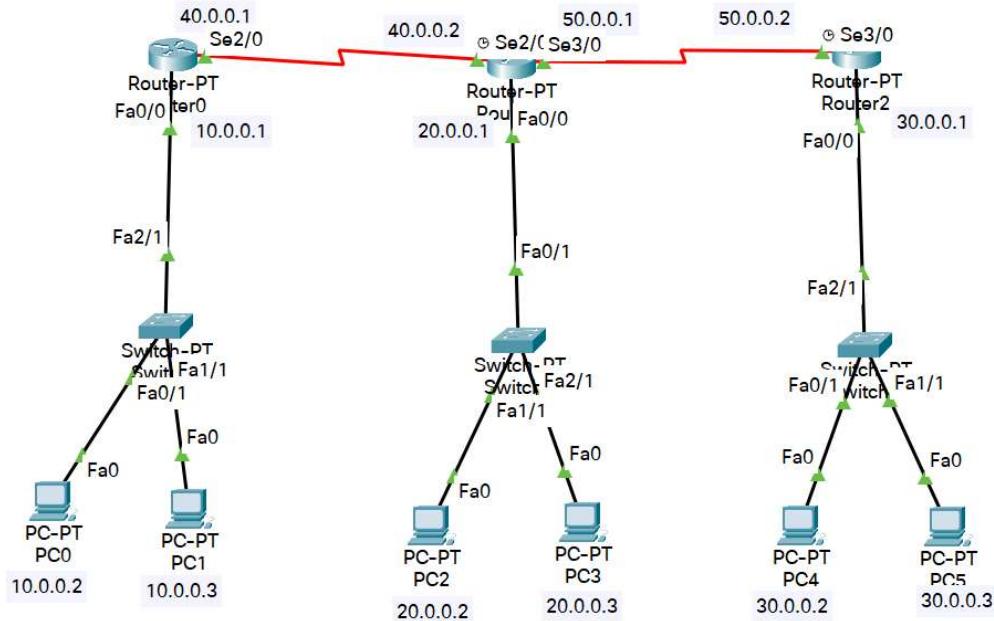
Ping statistics:-

Packet Sent = 4 Received = 3 Lost = 1 (25% loss)

Minimum = 2ms Maximum = 16ms Average = 6ms

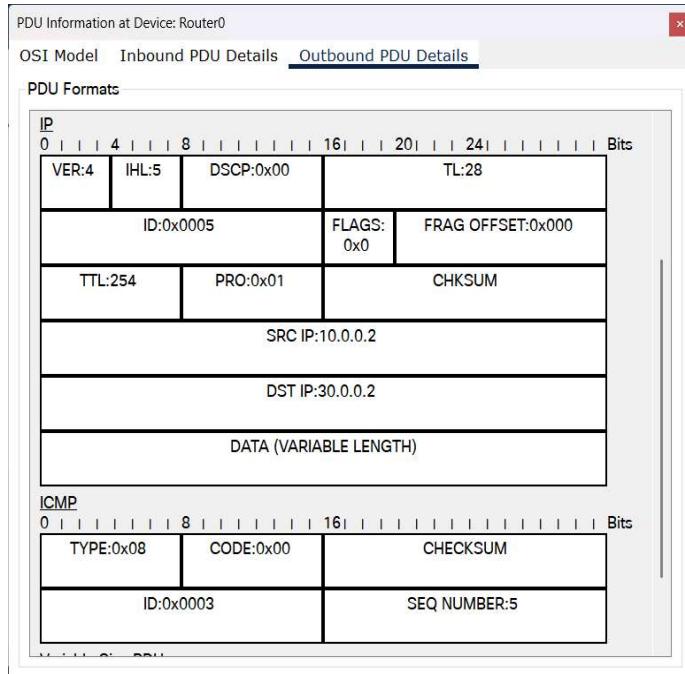
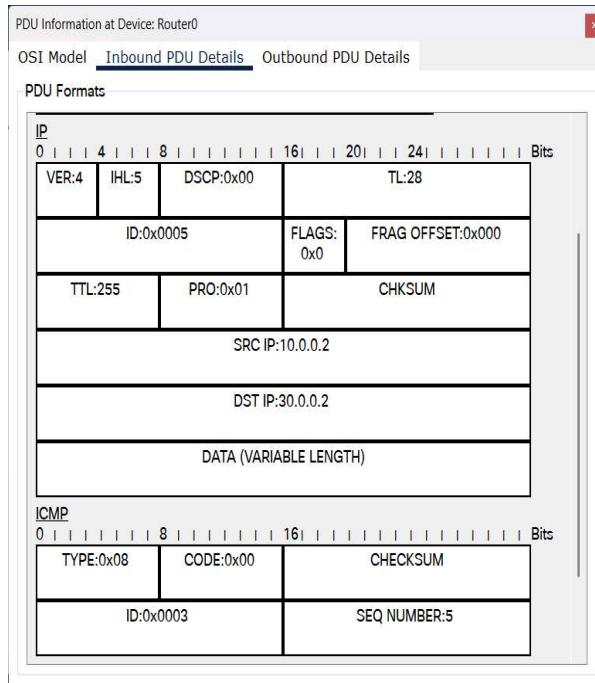
Program 7

- i. Demonstrate the TTL/ Life of a Packet
- ii. Procedure along with the topology

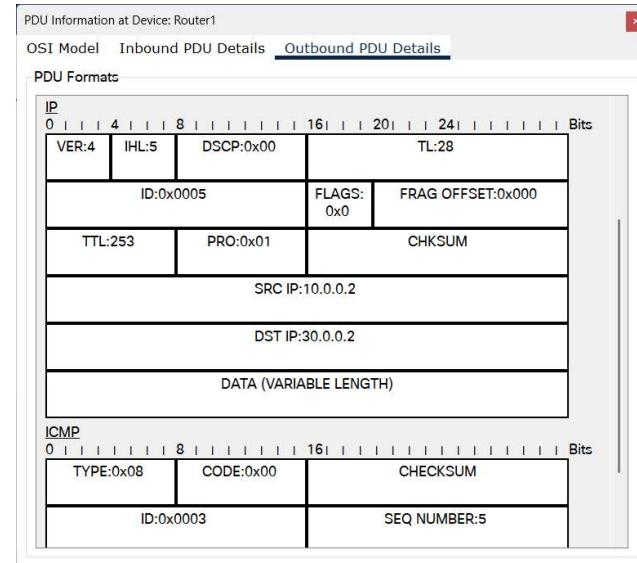
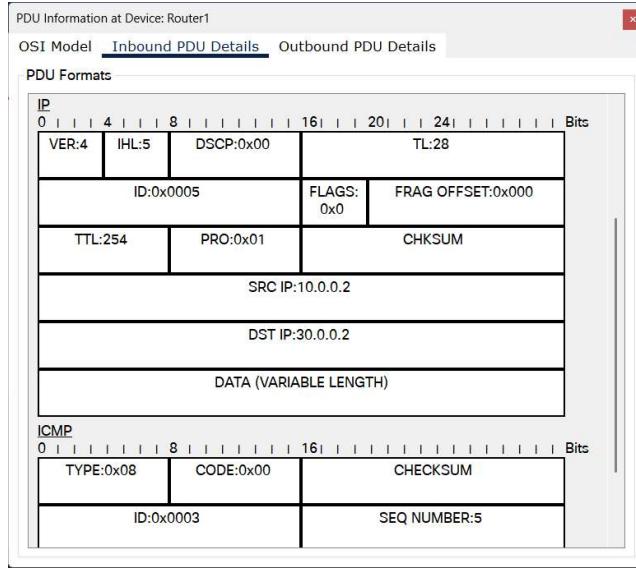


- iii. Screen shots/ output

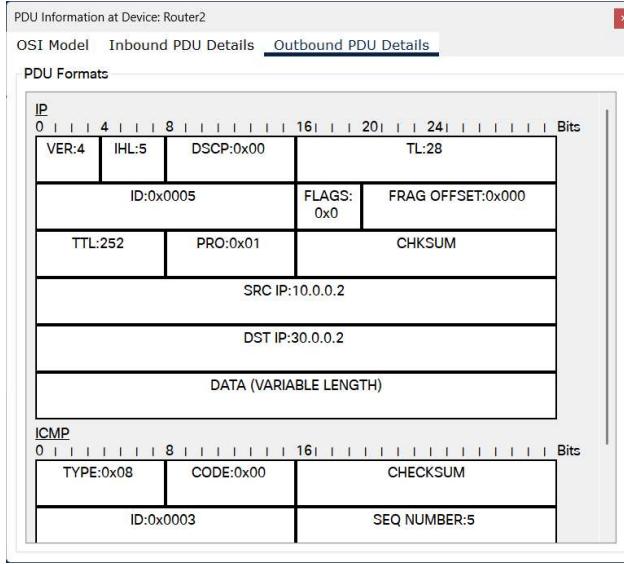
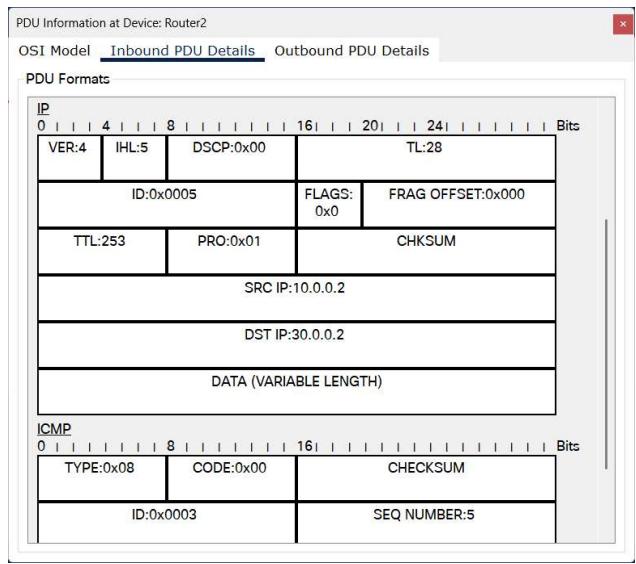
Packet at Router0



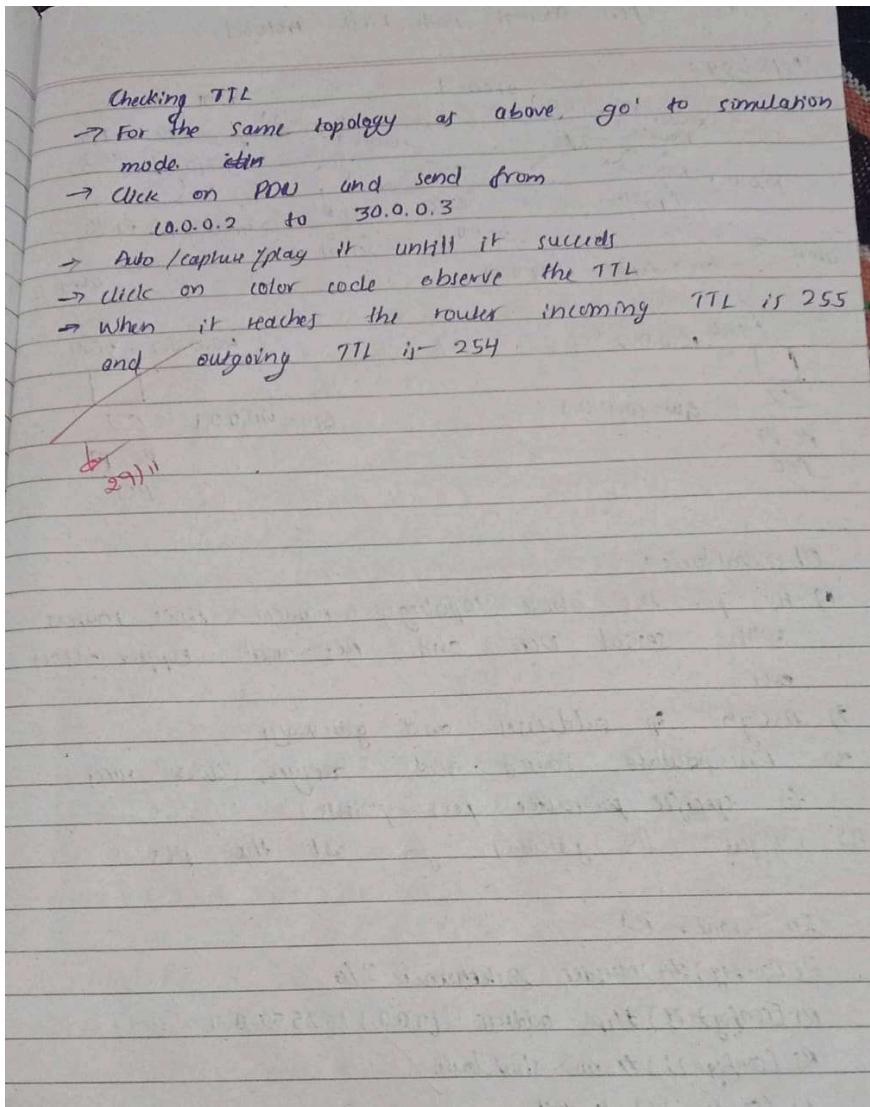
Packet at Router1



Packet at Router2

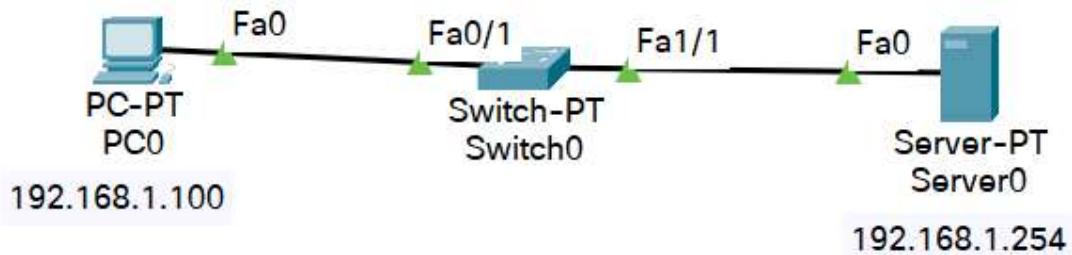


iv. Observation



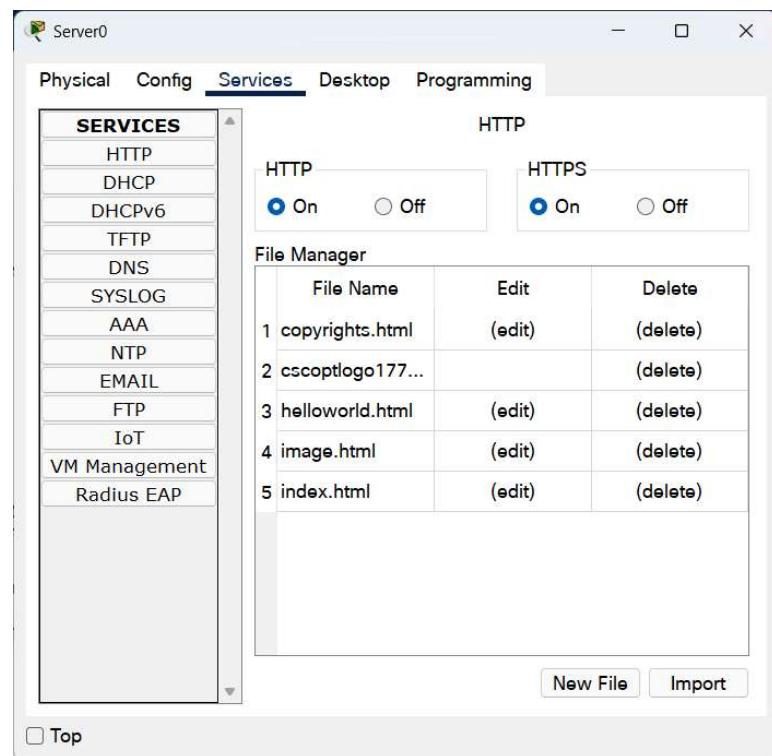
Program 8

- i. Configure Web Server, DNS within a LAN.
- ii. Procedure along with the topology



- iii. Screen shots/ output

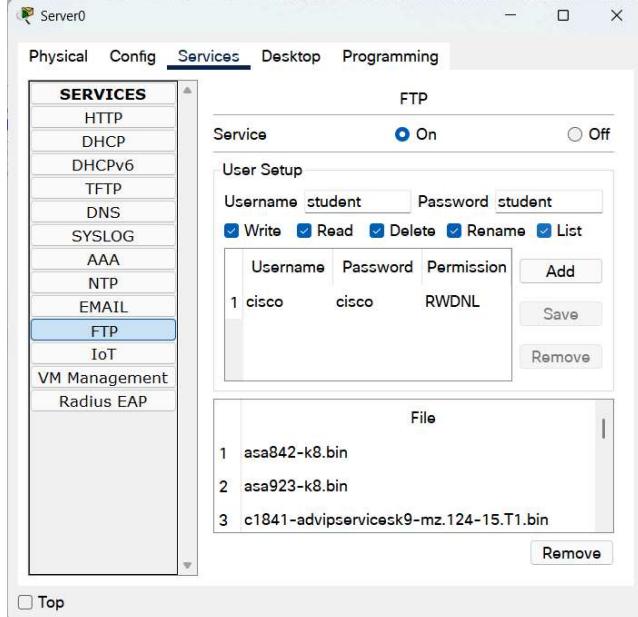
Server's services



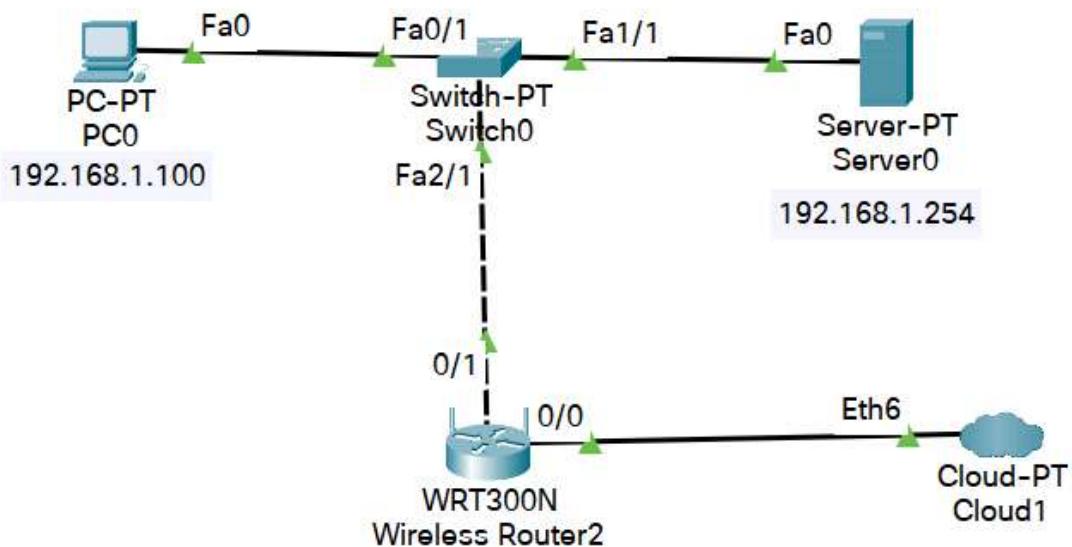
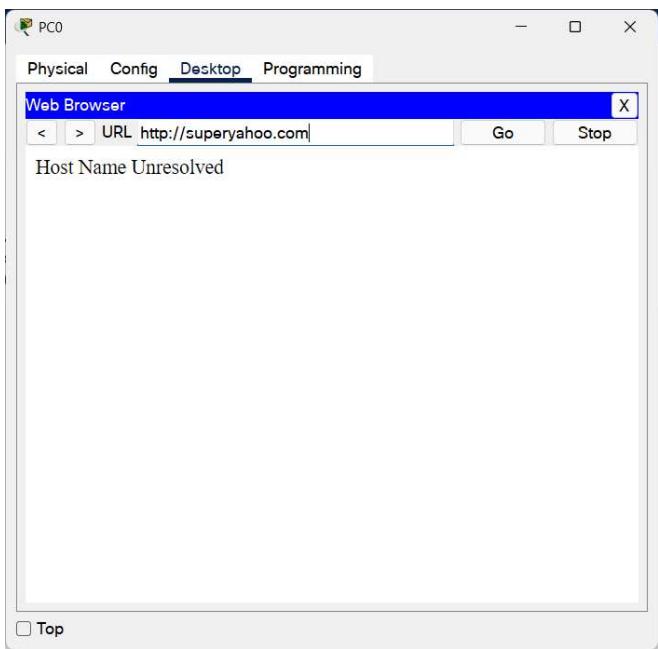
PC's Web Browser

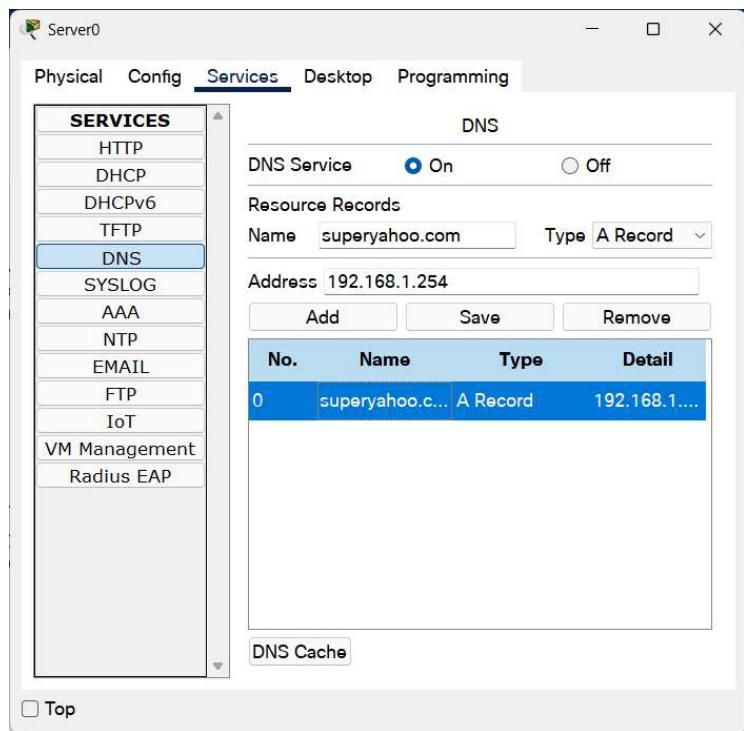


Top



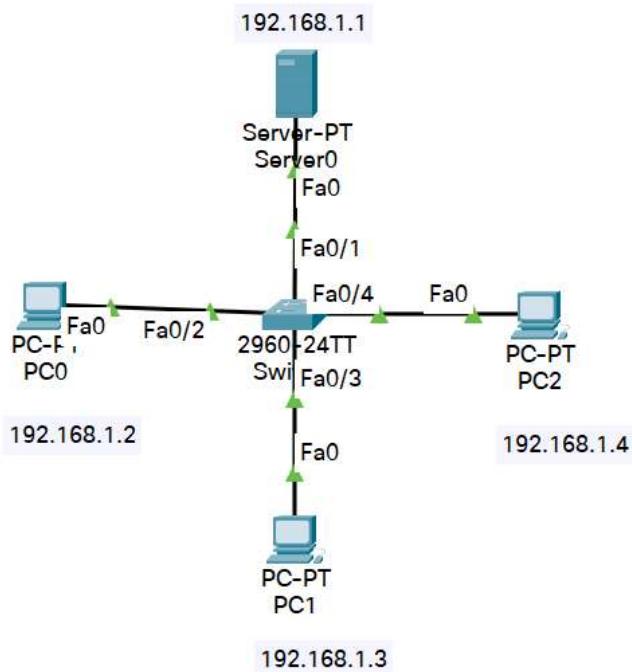
Top





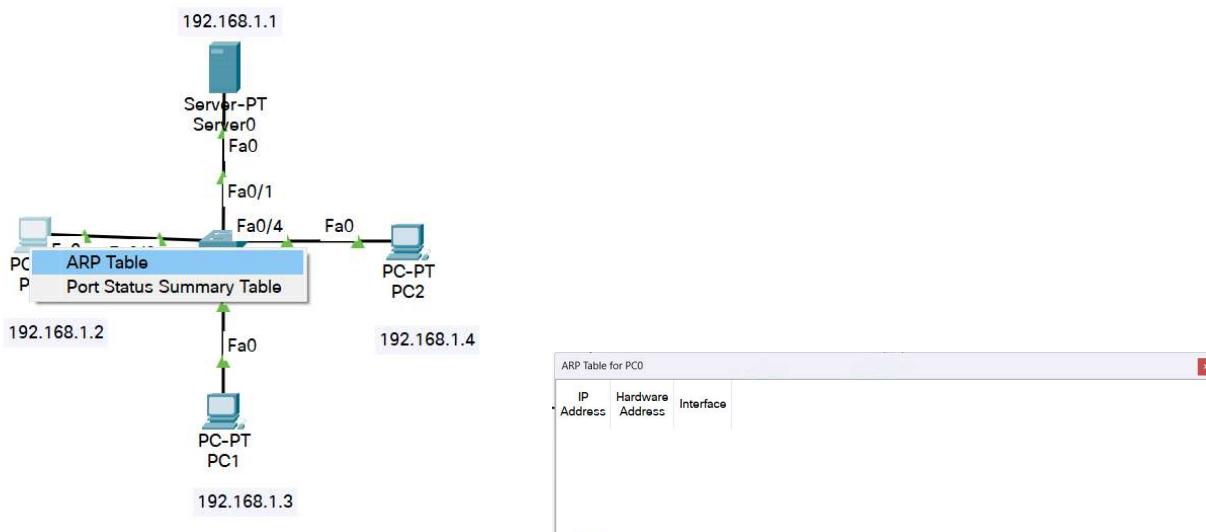
Program 9

- i. To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)
- ii. Procedure along with the topology

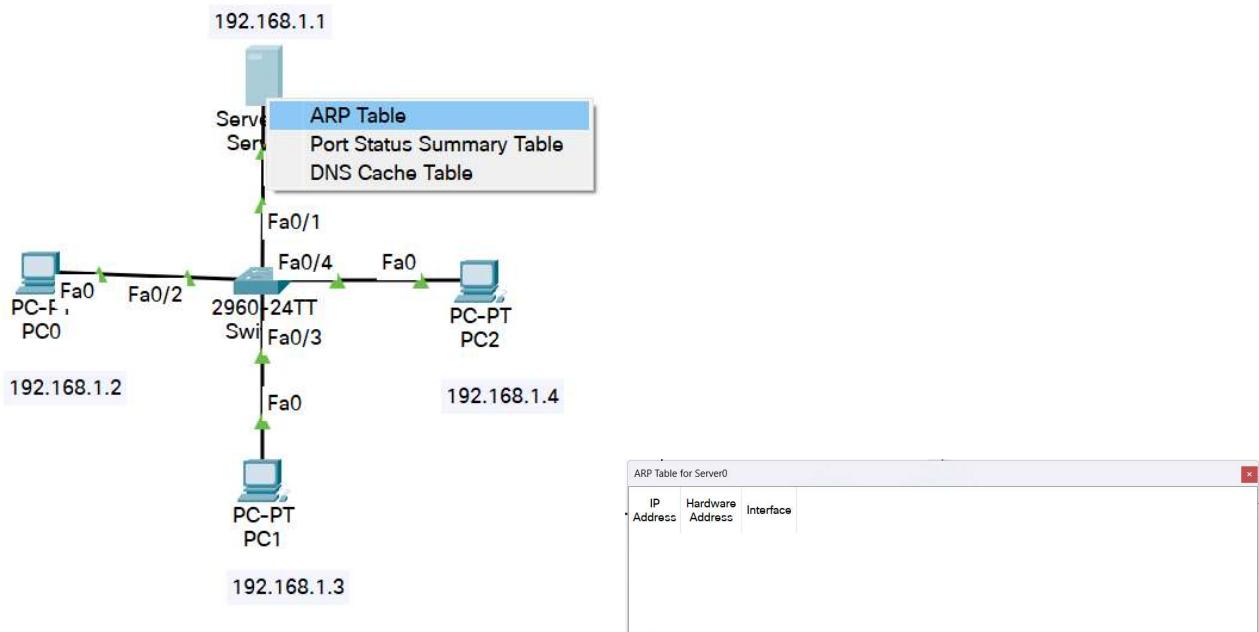


- iii. Screen shots/ output

ARP Table of PC



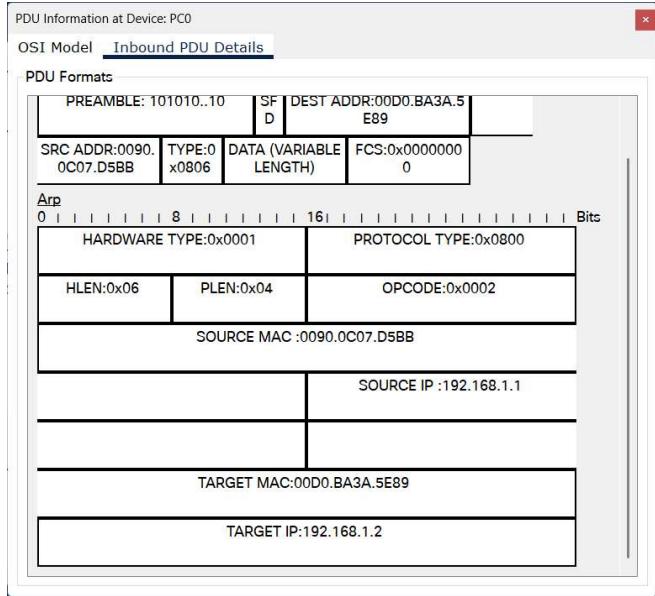
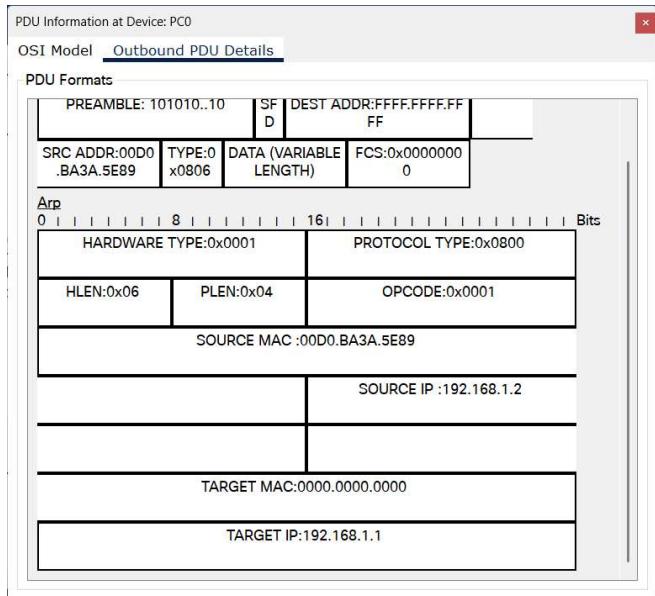
ARP Table of Server



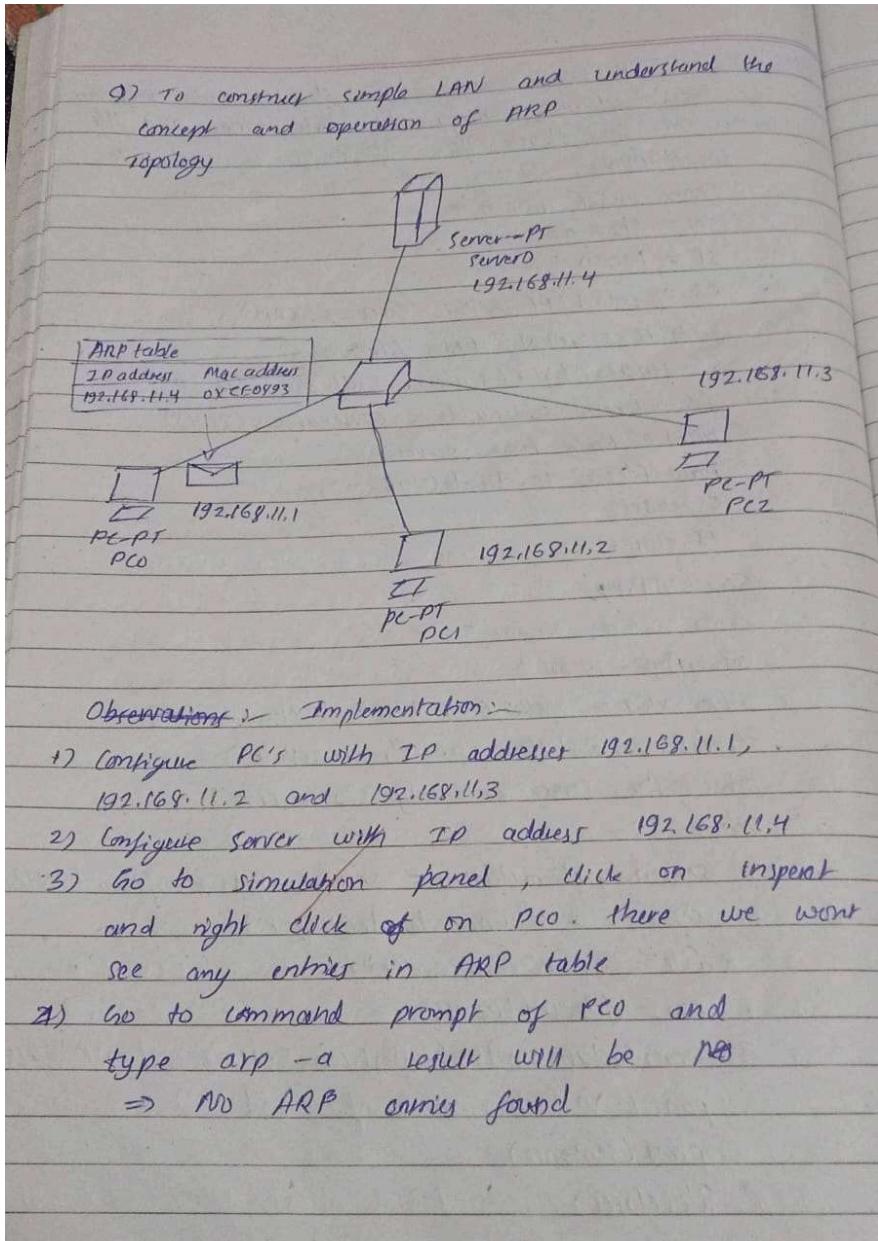
Command at PC

```
Packet Tracer PC Command Line 1.0
C:\>arp -a
No ARP Entries Found
C:\>
```

Pinging in Simulation Mode



iv. Observation



Observation:-

97 Ping from PC0 to server

Ping 192.168.11.4

→ Two packets are created ICMP and ARP

→ Hover over the packets to check the type of packets

→ Click on ARP packets

→ Now we can see outbound flow details

→ Click on capture to start the simulation

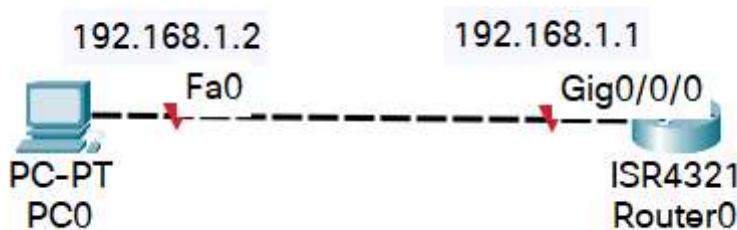
→ Ping request from PC0 to switch will happen

→ Again click on capture, request will be sent to server from switch

→ we can see the ICMP packet movement

Program 10

- i. To understand the operation of TELNET by accessing the router in server room from a PC in IT office.
- ii. Procedure along with the topology



- iii. Screen shots/ output

Router

```
Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname R1
R1(config)#enable secret hello
R1(config)#interface g0/0/0
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#no shutdown

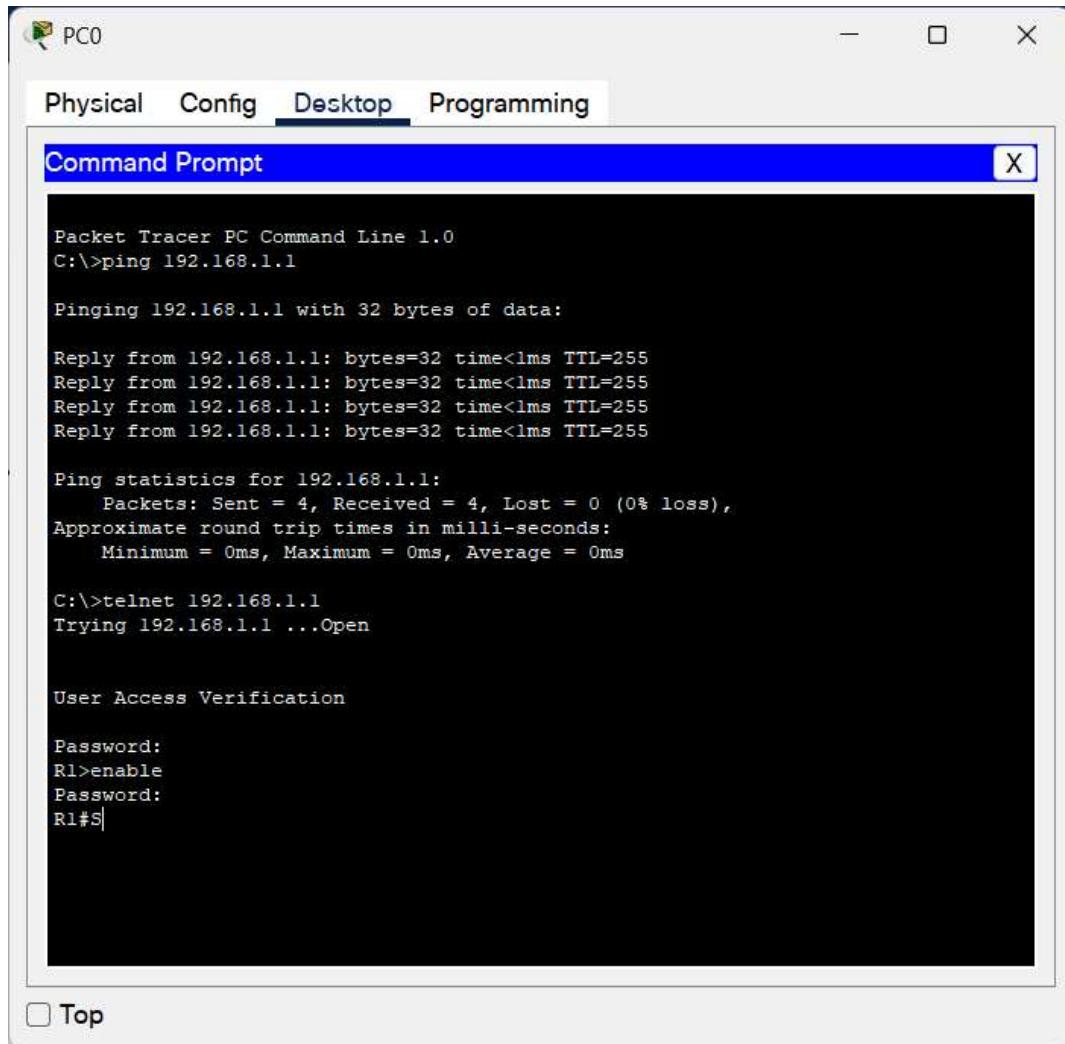
R1(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/0, changed state to up

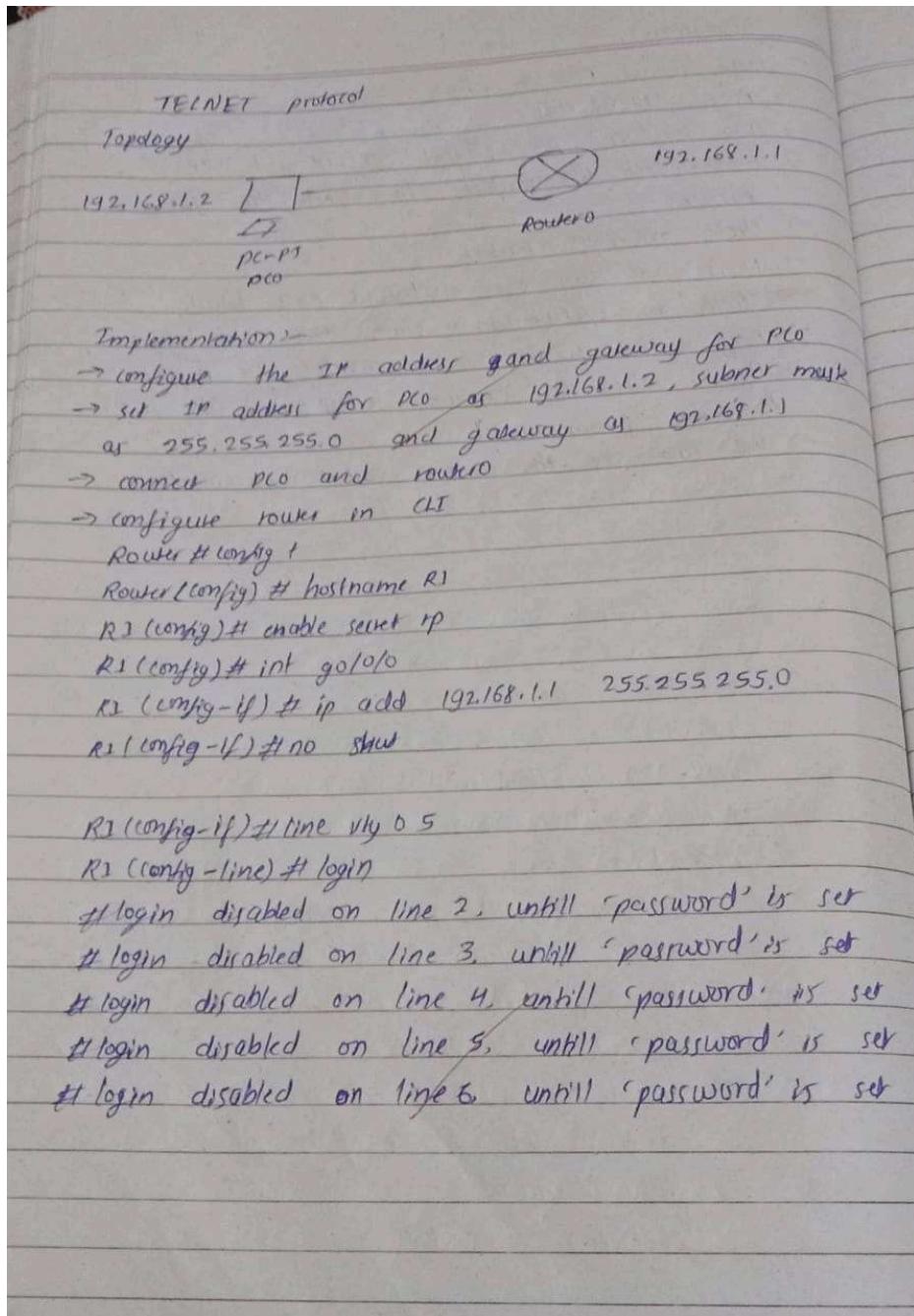
R1(config-if)#line vty 0 5
R1(config-line)#login
% Login disabled on line 2, until 'password' is set
% Login disabled on line 3, until 'password' is set
% Login disabled on line 4, until 'password' is set
% Login disabled on line 5, until 'password' is set
% Login disabled on line 6, until 'password' is set
% Login disabled on line 7, until 'password' is set
R1(config-line)#password pass
R1(config-line)#end
R1#
%SYS-5-CONFIG_I: Configured from console by console

R1#wr
Building configuration...
[OK]
R1#
```

PC



iv. Observation



R1 (config-line) # password rp

R1 (config-line) # exit

R1 (config) # exit

R1 # wr

Building

[OK]

Observation:-

→ Go to PCD - Go to command prompt

ping 192.168.1.1

pinging from 192.168.1.1 with 32 bytes of data

Reply from 192.168.1.1 : bytes=32 time <1ms TTL=255

Ping statistics for 192.168.1.1

packets sent = 4 Received = 4 lost = 0 (0% loss)

→ C:\> telnet 192.168.1.1

Trying 192.168.1.1 port 23

User Access Verification

Password: rp

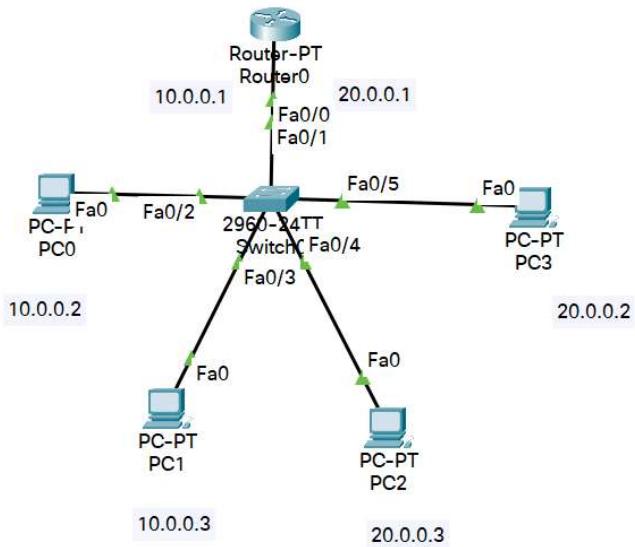
R1 > en

password: rp

R1#

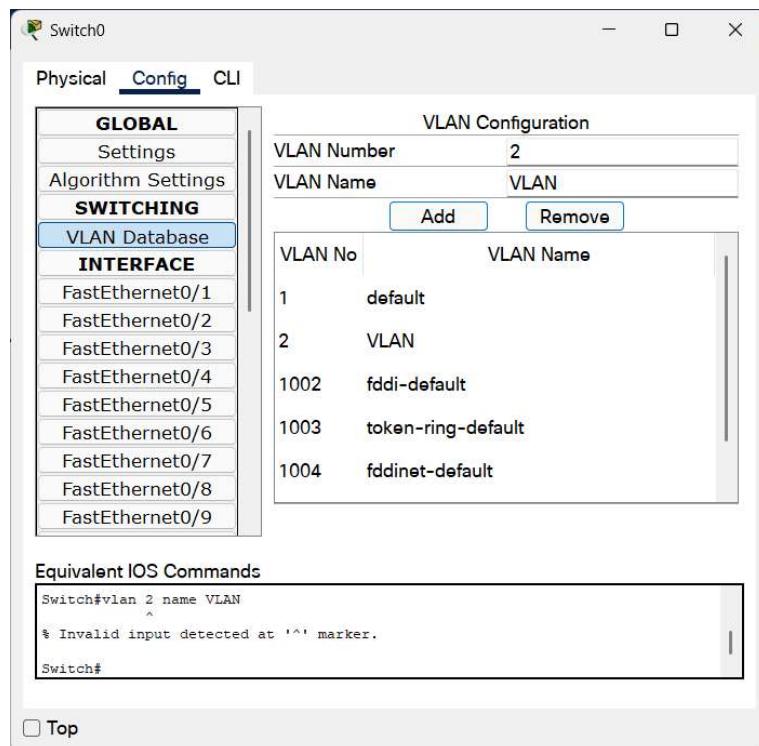
Program 11

- i. To construct a VLAN and make the PC's communicate among a VLAN
- ii. Procedure along with the topology

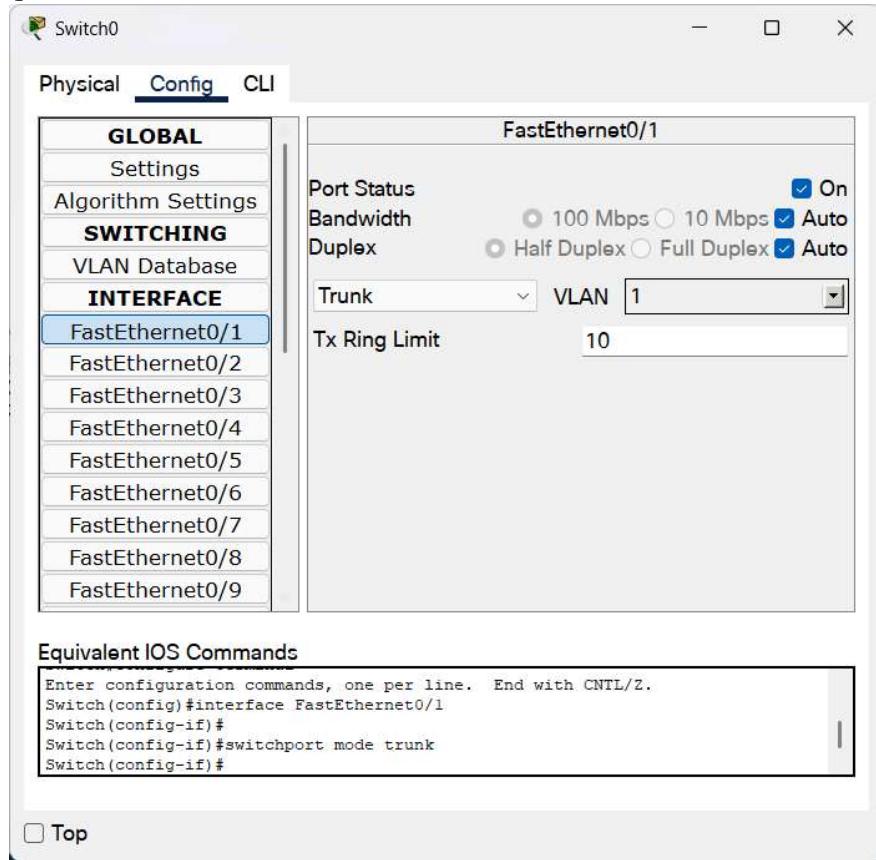


- iii. Screen shots/ output

Switch Configuration



Configuration of port connected to router



Configuration of ports connected to other networks

This screenshot shows the configuration of two ports connected to other networks. The left sidebar lists the same configuration sections as the previous screenshot. Under INTERFACE, 'FastEthernet0/4' and 'FastEthernet0/5' are selected sequentially. The main panel shows the configuration for 'FastEthernet0/4' and 'FastEthernet0/5'. Both ports are set to On (checked). Bandwidth is set to Auto (checked). Duplex is set to Auto (checked). For 'FastEthernet0/4', the Access setting is selected and assigned to VLAN 2. The Tx Ring Limit is set to 10. For 'FastEthernet0/5', the Access setting is selected and assigned to VLAN 2. The Tx Ring Limit is set to 10. Below each port's configuration, there is an 'Equivalent IOS Commands' section:

For FastEthernet0/4:

```

Switch(config)#interface FastEthernet0/4
Switch(config-if)#
Switch(config-if)#
Switch(config-if)#switchport access vlan 2
Switch(config-if)#

```

For FastEthernet0/5:

```

Switch(config)#interface FastEthernet0/5
Switch(config-if)#
Switch(config-if)#
Switch(config-if)#switchport access vlan 2
Switch(config-if)#

```

Configuartion of Router

Router0

Physical Config **CLI**

IOS Command Line Interface

```
Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface Fa0/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shutdown

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

Router(config-if)#exit
Router(config)#interface Fa0/0.1
Router(config-subif)#
%LINK-5-CHANGED: Interface FastEthernet0/0.1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0.1, changed
state to up

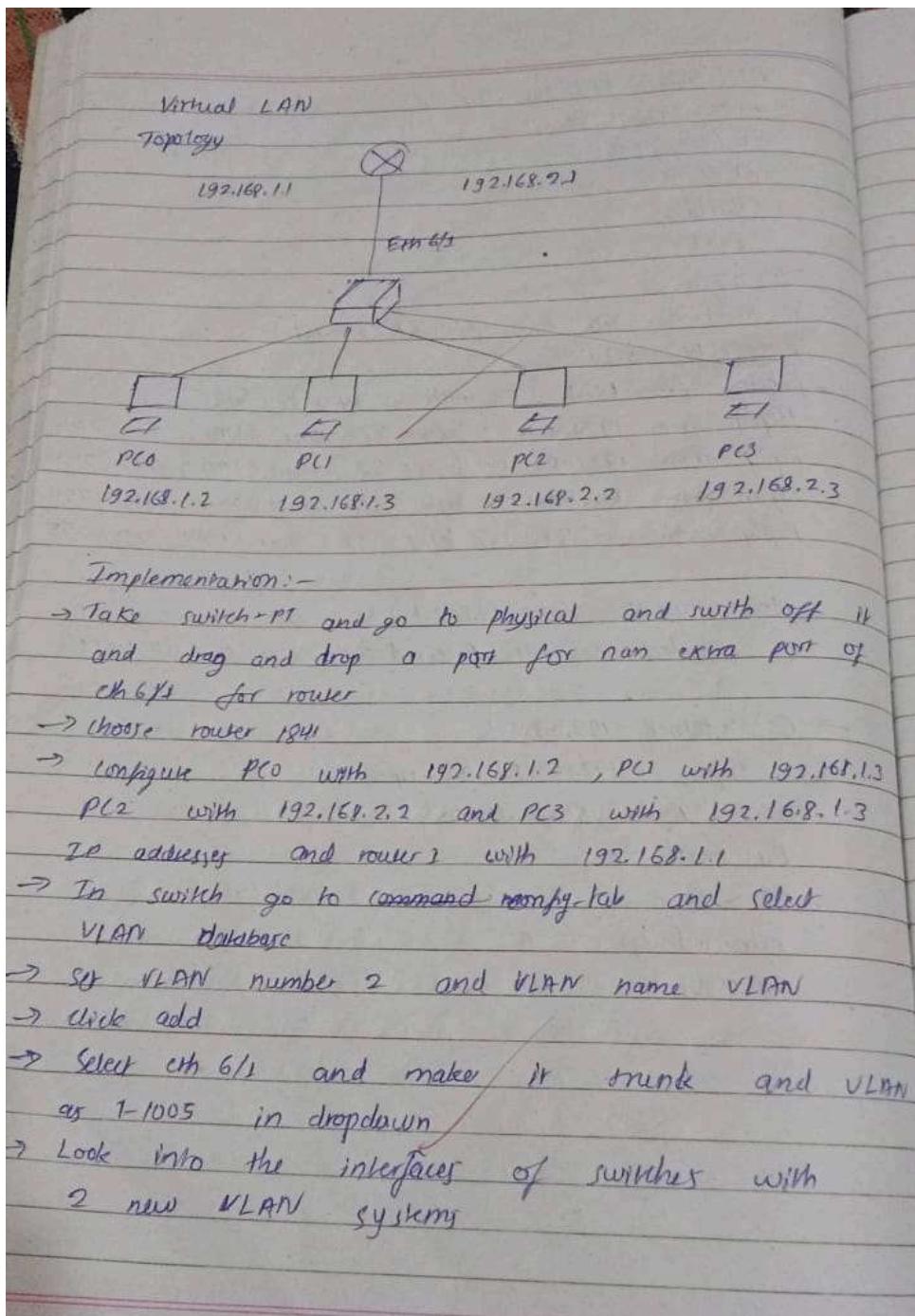
Router(config-subif)#encapsulation dot1q 2
Router(config-subif)#ip address 20.0.0.1 255.0.0.0
Router(config-subif)#no shutdown
Router(config-subif)#exit
Router(config)#

Ctrl+F6 to exit CLI focus
```

Top

Copy **Paste**

iv. Observation



→ Next click fastethernet 0/1 and select VLAN and do the same for fastethernet 3/1
Observation

→ Go to router-config tab select VLAN database Enter the number and name of VLAN created

→ Go to clt
Router (vlan) #exit
Apply completed
Exiting...

Router # config #

Router (config) # interface fastethernet 0/0.1

Router (config-subif) # encapsulation dot1q 2

Router (config-subif) # ip address 192.168.2.1 255.255.255.0

Router (config-subif) # no shutdown

Router (config-subif) # exit

Observations -

→ Now two LANs are connected we can ping from PC0 to PC3

ping 192.168.2.3

Pinging 192.168.2.3 with 32 bytes of data

Reply from 192.168.2.3 bytes=32 time=11ms TTL=125

Reply from 192.168.2.3 bytes=32 time=9ms TTL=125

Reply from 192.168.2.3 bytes=32 time=6ms TTL=125

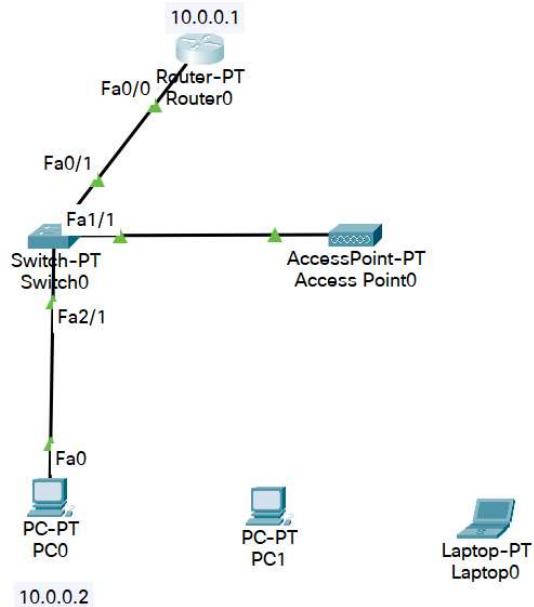
Reply from 192.168.2.3 bytes=32 time=8ms TTL=125

Ping Statistics for 192.168.2.3

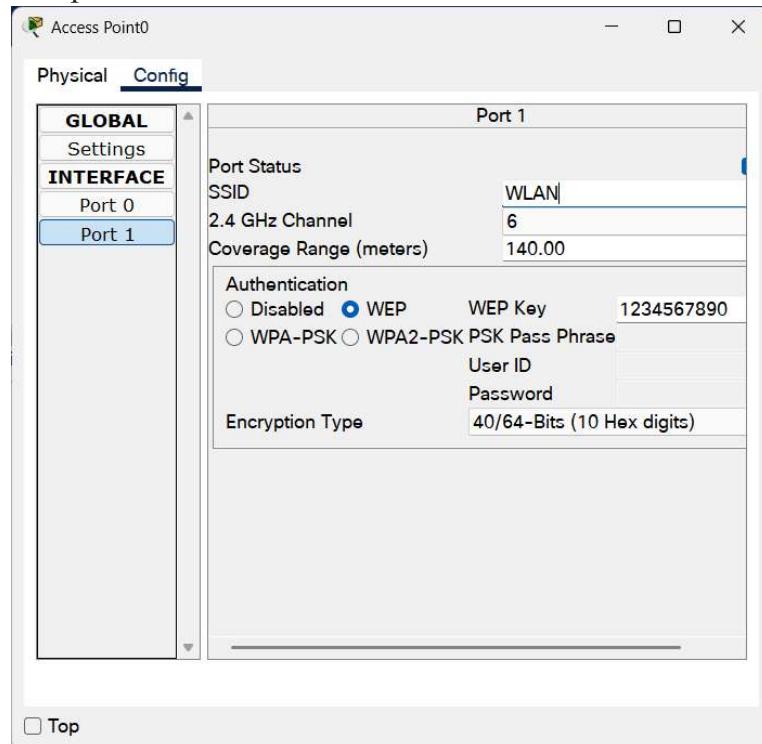
Packets sent = 4 Received = 4 Lost = 0 (0% loss)

Program 12

- i. To construct a WLAN and make the nodes communicate wirelessly.
- ii. Procedure along with the topology



- iii. Screen shots/ output



PC1

Physical Config Desktop Programming

MODULES	
WMP300N	
PT-HOST-NM-1AM	
PT-HOST-NM-1CE	
PT-HOST-NM-1CFE	
PT-HOST-NM-1CGE	
PT-HOST-NM-1FFE	
PT-HOST-NM-1FGE	
PT-HOST-NM-1W	
PT-HOST-NM-1W-A	
PT-HOST-NM-1W-AC	
PT-HOST-NM-3G/4G	
PT-HOST-NM-COVER	
PT-HEADPHONE	
PT-MICROPHONE	

The WMP300N module provides one 2.4GHz wireless interface suitable for connection to wireless networks. The module supports protocols that use Ethernet for...

Top

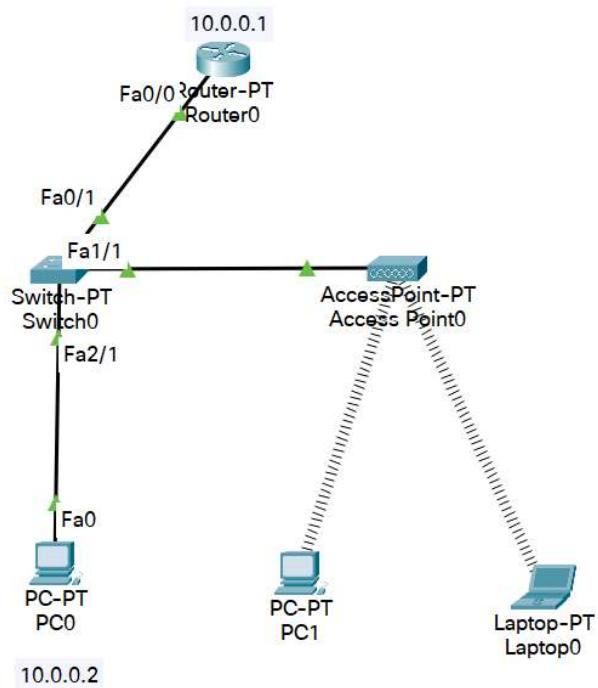
Laptop0

Physical Config Desktop Programming

MODULES	
WPC300N	
PT-LAPTOP-NM-1AM	
PT-LAPTOP-NM-1CE	
PT-LAPTOP-NM-1CFE	
PT-LAPTOP-NM-1CGE	
PT-LAPTOP-NM-1FFE	
PT-LAPTOP-NM-1FGE	
PT-LAPTOP-NM-1W	
PT-LAPTOP-NM-1W-A	
PT-LAPTOP-NM-1W-AC	
PT-LAPTOP-NM-3G/4G	
PT-HEADPHONE	
PT-MICROPHONE	

The Linksys-WPC300N module provides one 2.4GHz wireless interface suitable for connection to wireless...

Top



Ping:

Laptop0

Physical Config Desktop Programming

Command Prompt

```

Packet Tracer PC Command Line 1.0
C:\>

C:\>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time=40ms TTL=128
Reply from 10.0.0.3: bytes=32 time=25ms TTL=128
Reply from 10.0.0.3: bytes=32 time=26ms TTL=128
Reply from 10.0.0.3: bytes=32 time=24ms TTL=128

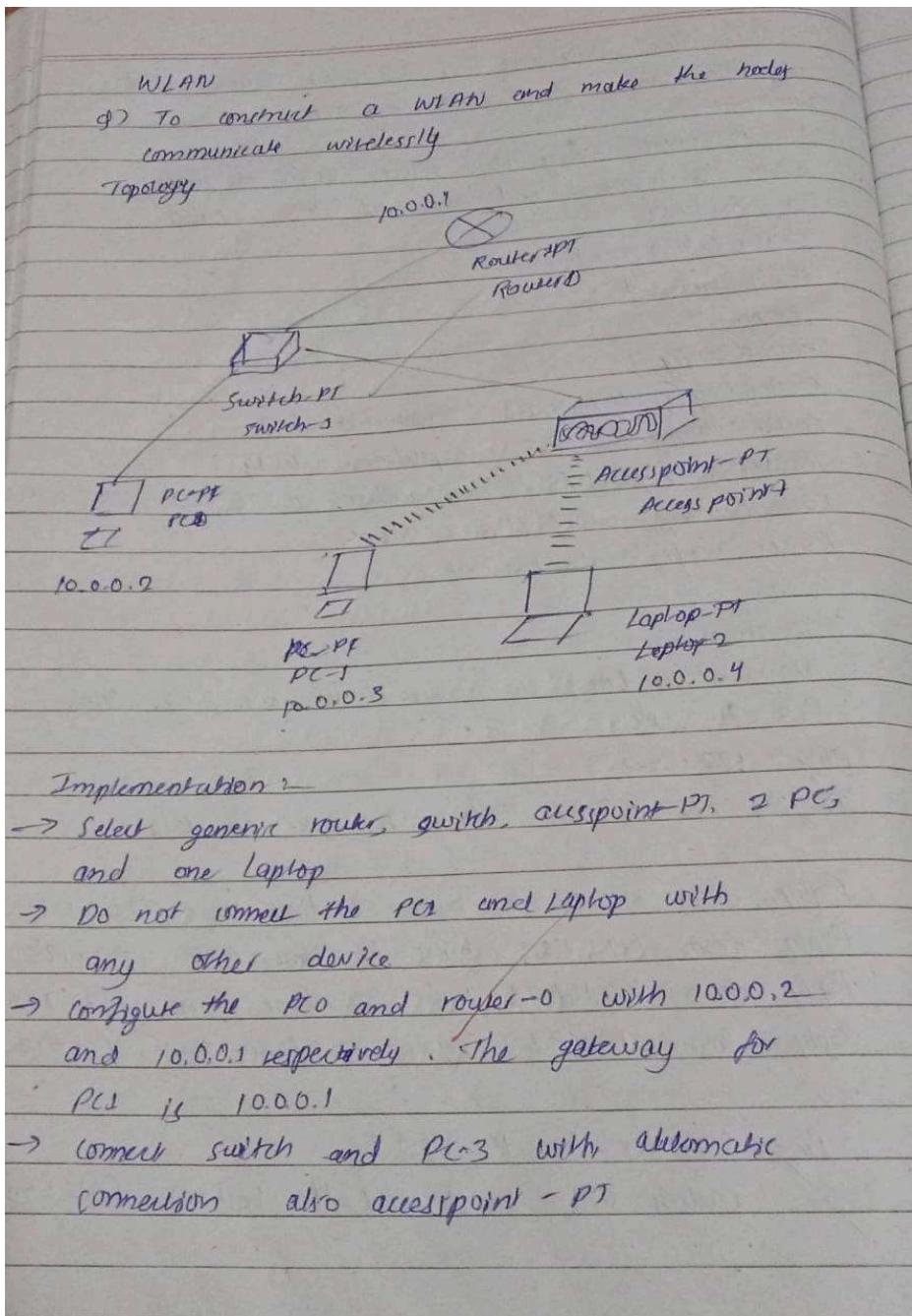
Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 24ms, Maximum = 40ms, Average = 28ms

C:\>

```

Top

iv. Observation



- configure access point → port 1 → SSID name → any name (WLAN here)
- Select WEP and give any 10 digit hex key
- 1234567890
- configure PC4 and laptop with wireless standards
- switch off the device - Drag the existing PI-HOST-NNA to the component listed in the L45. - 7AM
- Drag WMP300N wireless interface to empty slot
- switch on the device
- In the config tab a new wireless interface would have been added. Now configure SSID, WEP key, IP address and gateway to the device

Observations:-

- We see that laptop and PC1 have been connected to the network through access point
- We ping laptop from PC0
ping 10.0.0.4

pinging 10.0.0.4 with 32 bytes of data

Reply from 10.0.0.4 : bytes = 32 time = 3 ms TTL = 125

Reply from 10.0.0.4 : bytes = 32 time = 2 ms TTL = 125

Reply from 10.0.0.4 : bytes = 32 time = 9 ms TTL = 125

Reply from 10.0.0.4 : bytes = 32 time = 9 ms TTL = 125

Ping statistics for 10.0.0.3

Packets: sent = 4 Received = 4 Lost = 0 (0% loss)

26/12

Cycle-II

Program 1

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

```
def crc_ccitt_16_bitstream(bitstream: str, poly: int = 0x1021, init_crc: int = 0xFFFF) -> int:  
    crc = init_crc  
    for bit in bitstream:  
        crc ^= int(bit) << 15 # Align the bit with CRC's uppermost bit  
        for _ in range(1): # Process the single bit  
            if crc & 0x8000: # Check if the leftmost bit is set  
                crc = (crc << 1) ^ poly  
            else:  
                crc <<= 1  
            crc &= 0xFFFF # Ensure CRC remains 16-bit  
    return crc
```

```
def append_crc_to_bitstream(bitstream: str) -> str:  
    crc = crc_ccitt_16_bitstream(bitstream)  
    crc_bits = f'{crc:016b}' # Convert CRC to a 16-bit binary string  
    return bitstream + crc_bits
```

```
def verify_crc_bitstream(bitstream_with_crc: str) -> bool:  
    if len(bitstream_with_crc) < 16:  
        return False # Not enough bits to contain CRC  
    data, received_crc = bitstream_with_crc[:-16], bitstream_with_crc[-16:]  
    calculated_crc = crc_ccitt_16_bitstream(data)  
    return calculated_crc == int(received_crc, 2)
```

```
# Example usage:  
if __name__ == "__main__":  
    # User input for original bitstream  
    message_bits = input("Enter the original bitstream (e.g., 11010011101100): ")  
  
    # Calculate and append CRC  
    bitstream_with_crc = append_crc_to_bitstream(message_bits)  
    print(f'Bitstream with CRC: {bitstream_with_crc}')  
  
    # User input for verification  
    user_bitstream = input("Enter the received bitstream for verification (e.g., 1101001110110011011011000011): ")
```

```
)  
  
# Verify CRC  
is_valid = verify_crc_bitstream(user_bitstream)  
print(f"CRC valid: {is_valid}")
```

iii. Screen shots/ output

```
In [1]: runcell(0, 'E:/python_files/untitled2.py')  
  
Enter the original bitstream (e.g., 11010011101100): 11111  
Bitstream with CRC: 111111111111111100000  
  
Enter the received bitstream for verification (e.g., 11010011101100110110110111000011): 111111111111111100000  
CRC valid: True  
  
In [2]: runcell(0, 'E:/python_files/untitled2.py')  
  
Enter the original bitstream (e.g., 11010011101100): 11111  
Bitstream with CRC: 111111111111111100000  
  
Enter the received bitstream for verification (e.g., 11010011101100110110110111000011): 111111111111111100001  
CRC valid: False
```

iv. Observation

LAB - 6

Program for Error detection in CRC
and leaky bucket algorithm

Q) Write a program for Error detection using CRC-CCITT
(16 bits)

```
#include <iostream>
#include <string.h>
using namespace std;
int s
int src(char *ip, char *op, char *poly, int mode)
{
    strcpy(op, ip);
    if(mode)
        for(int i=1; i<strlen(poly); i++)
            { strcat(op, "0"); }
    for(int i=0; i<strlen(ip); i++)
        { if(op[i] == '1')
            for(int j=0; j<strlen(poly); j++)
                { if(op[i+j] == poly[j])
                    op[i+j] = '0';
                else
                    op[i+j] = '1';
            }
        }
    for(int i=0; i<strlen(poly); i++)
        { if(op[i] == '1')
            return 0;
        }
    return 1;
}
```

```

int main()
{
    char ip[50], op[50], recv[50];
    char poly[] = "10001000000100001";
    cout << "Enter the input message in binary" << endl;
    cin >> ip;
    crc(ip, op, poly);
    cout << "The transmitted message is:" << ip << op
        strlen(ip) << endl;

    cout << "Enter the received message in binary" << endl;
    cin >> recv;
    if (crc(recv, op, poly, 0))
    {
        cout << "No error in data" << endl;
    }
    else
    {
        cout << "Error in data transmission has occurred" << endl;
    }
    return 0;
}

```

Output:-

Enter the input message in binary: 10010
 The transmitted message is : 10010001100100110011
 Enter the received message in binary: 10011
 Error in data transmission

Program 2

i. Write a program for congestion control using Leaky bucket algorithm

ii. Procedure

```
def main():
    # Initial packets in the bucket
    storage = 0

    # Total number of times bucket content is checked
    no_of_queries = 4

    # Total number of packets that can be accommodated in the bucket
    bucket_size = 10

    # Number of packets that enter the bucket at a time
    input_pkt_size = 4

    # Number of packets that exit the bucket at a time
    output_pkt_size = 1

    for _ in range(no_of_queries):
        # Space left in the bucket
        size_left = bucket_size - storage

        if input_pkt_size <= size_left:
            # Update storage
            storage += input_pkt_size
        else:
            print(f'Packet loss = {input_pkt_size}')

        print(f'Buffer size = {storage} out of bucket size = {bucket_size}')

        # Remove packets from storage
        storage -= output_pkt_size

if __name__ == "__main__":
    main()
```

iii. Screen shots/ output

```
In [3]: runcell(0, 'E:/Engineering/5Sem/CN/Experiments/untitled3.py')
Buffer size = 4 out of bucket size = 10
Buffer size = 7 out of bucket size = 10
Buffer size = 10 out of bucket size = 10
Packet loss = 4
Buffer size = 9 out of bucket size = 10
```

iv. Observation

Q) write a program for leaky bucket algorithm

```
#include <iostream.h>
using namespace std;
```

```
int main() {
    int no_of_queries, storage, output_pkt_size;
    int input_pkt_size, bucket_size, size_left;

    Storage = 0;
    no_of_queries = 4;
    bucket_size = 10;
    input_pkt_size = 4;
    output_pkt_size = 2;

    for (int i=0; i<no_of_queries; i++) {
        size_left = bucket_size - storage;
        if (input_pkt_size <= size_left)
            storage += input_pkt_size;
        else {
            cout << "Packet loss = " << input_pkt_size;
            cout << endl;
            cout << "Buffer size" << storage << "out of
            bucket size" << bucket_size << endl;
        }
        storage -= output_pkt_size;
    }
    return 0;
}
```

output:

Buffer size = 4 out of bucket size = 10

Buffer size = 7 out of bucket size = 10

Buffer size = 10 out of bucket size = 10

Packet loss = 4

Buffer size = 9 out of bucket size = 10

✓
ISPN

224

224

224

224

224

Program 3

- i. 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.

- ii. Procedure

```
clientTCP.py
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input("\nEnter file name: ")

clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print ('\nFrom Server:\n')
print(filecontents)
clientSocket.close()
```

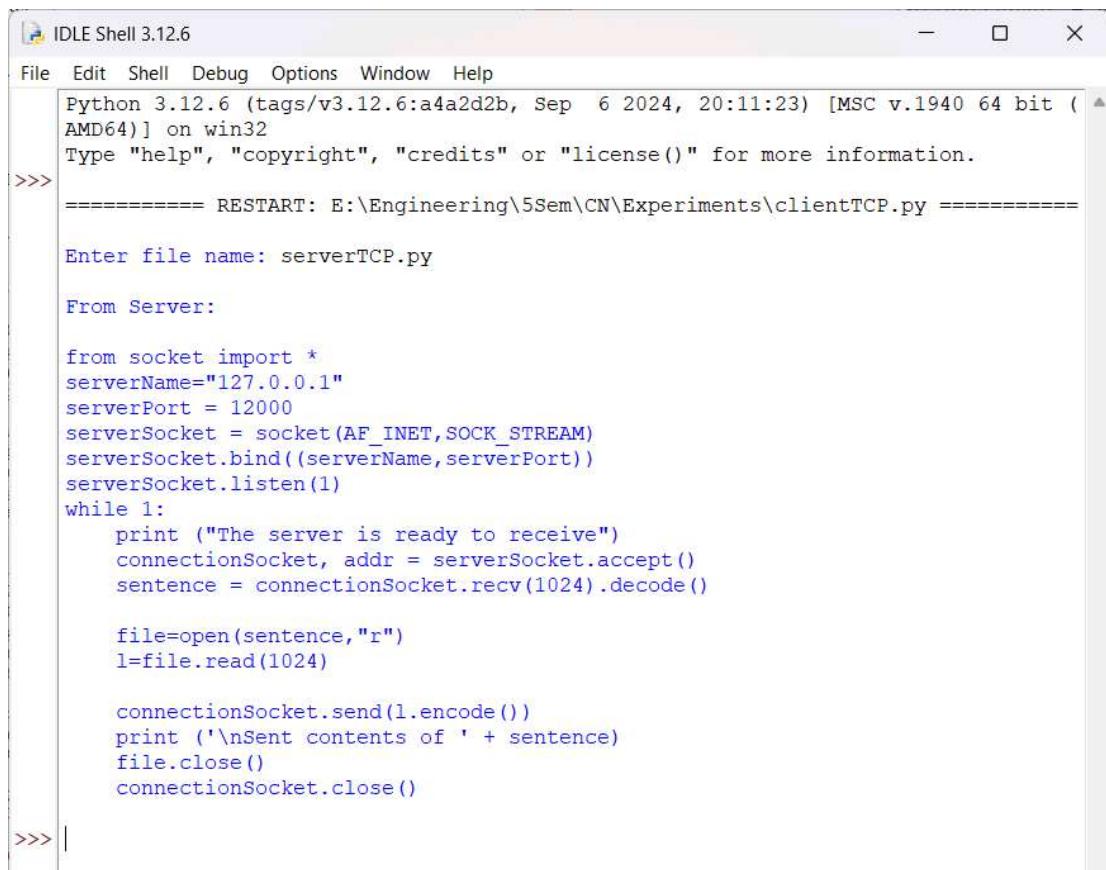
```
serverTCP.py
from socket import *
serverName="127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print ("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()

    file=open(sentence,"r")
    l=file.read(1024)

    connectionSocket.send(l.encode())
    print ('\nSent contents of ' + sentence)
    file.close()
    connectionSocket.close()
```

iii. Screen shots/ output

Client



```
IDLE Shell 3.12.6
File Edit Shell Debug Options Window Help
Python 3.12.6 (tags/v3.12.6:a4a2d2b, Sep 6 2024, 20:11:23) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> ===== RESTART: E:\Engineering\5Sem\CN\Experiments\clientTCP.py =====

Enter file name: serverTCP.py

From Server:

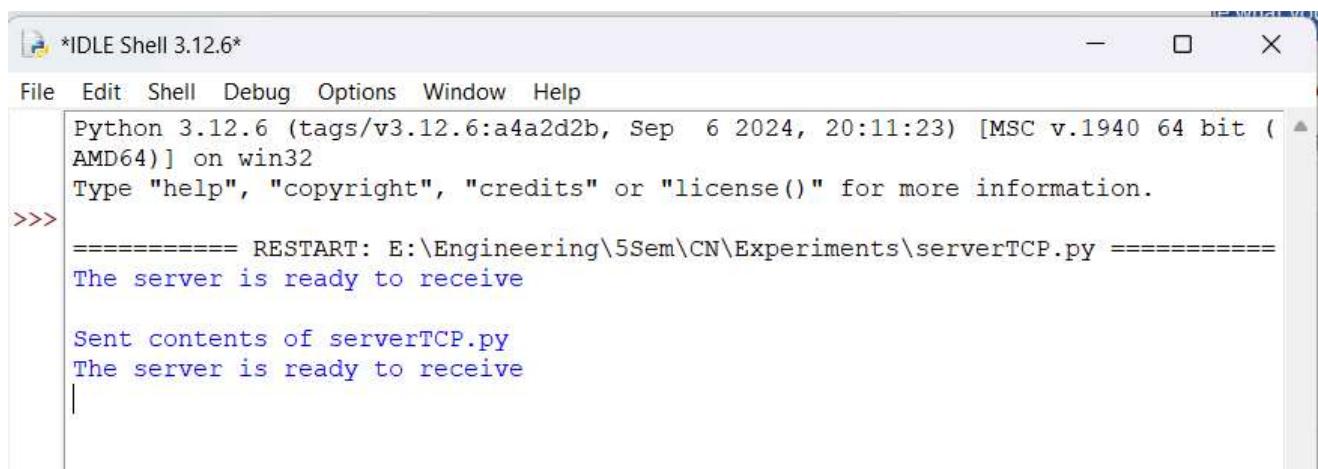
from socket import *
serverName="127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print ("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()

    file=open(sentence,"r")
    l=file.read(1024)

    connectionSocket.send(l.encode())
    print ('\nSent contents of ' + sentence)
    file.close()
    connectionSocket.close()

>>>
```

Server



```
*IDLE Shell 3.12.6*
File Edit Shell Debug Options Window Help
Python 3.12.6 (tags/v3.12.6:a4a2d2b, Sep 6 2024, 20:11:23) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> ===== RESTART: E:\Engineering\5Sem\CN\Experiments\serverTCP.py =====

The server is ready to receive

Sent contents of serverTCP.py
The server is ready to receive
```

iv. Observation

LAB-9

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

Client TCP.py

```
from socket import *
servername = '127.0.0.1'
serverport = 12000
clientsocket = socket(AF_INET, SOCK_STREAM)
clientsocket.connect((servername, serverport))
sentence = input("Enter file names")
filecontents = clientsocket.recv(1024).decode()
print("From server\n")
print(filecontents)
clientsocket.close()
```

Server TCP.py

```
from socket import *
servername = '127.0.0.1'
serverport = 12000
serversocket = socket(AF_INET, SOCK_STREAM)
serversocket.bind((servername, serverport))
serversocket.listen()
while 1:
    print("Server is ready")
    CS, addr = serversocket.accept()
    sen = CS.recv(1024).decode()
    file = open(sen, "r")
    l = file.read(1024)
    print("sent contents of " + sen)
    file.close()
    CS.close()
```

TCP output:

Server: Client:

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(2)

while 1:

print("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("\nSent contents of " + sentence)

file.close()

connectionSocket.close()

Server:

The server is ready to receive

Sent contents of serverTCP.py

The server is ready to receive

Program 4

- i. 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.

- ii. Procedure

```
clientUDP.py
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
```

```
sentence = input("\nEnter file name: ")
clientSocket.sendto(bytes(sentence,"utf-8"),(serverName, serverPort))
```

```
filecontents,serverAddress = clientSocket.recvfrom(2048)
print ('\nReply from Server:\n')
print (filecontents.decode("utf-8"))
# for i in filecontents:
#     print(str(i), end = '')
clientSocket.close()
clientSocket.close()
```

```
serverUDP.py
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file=open(sentence,"r")
    con=file.read(2048)
```

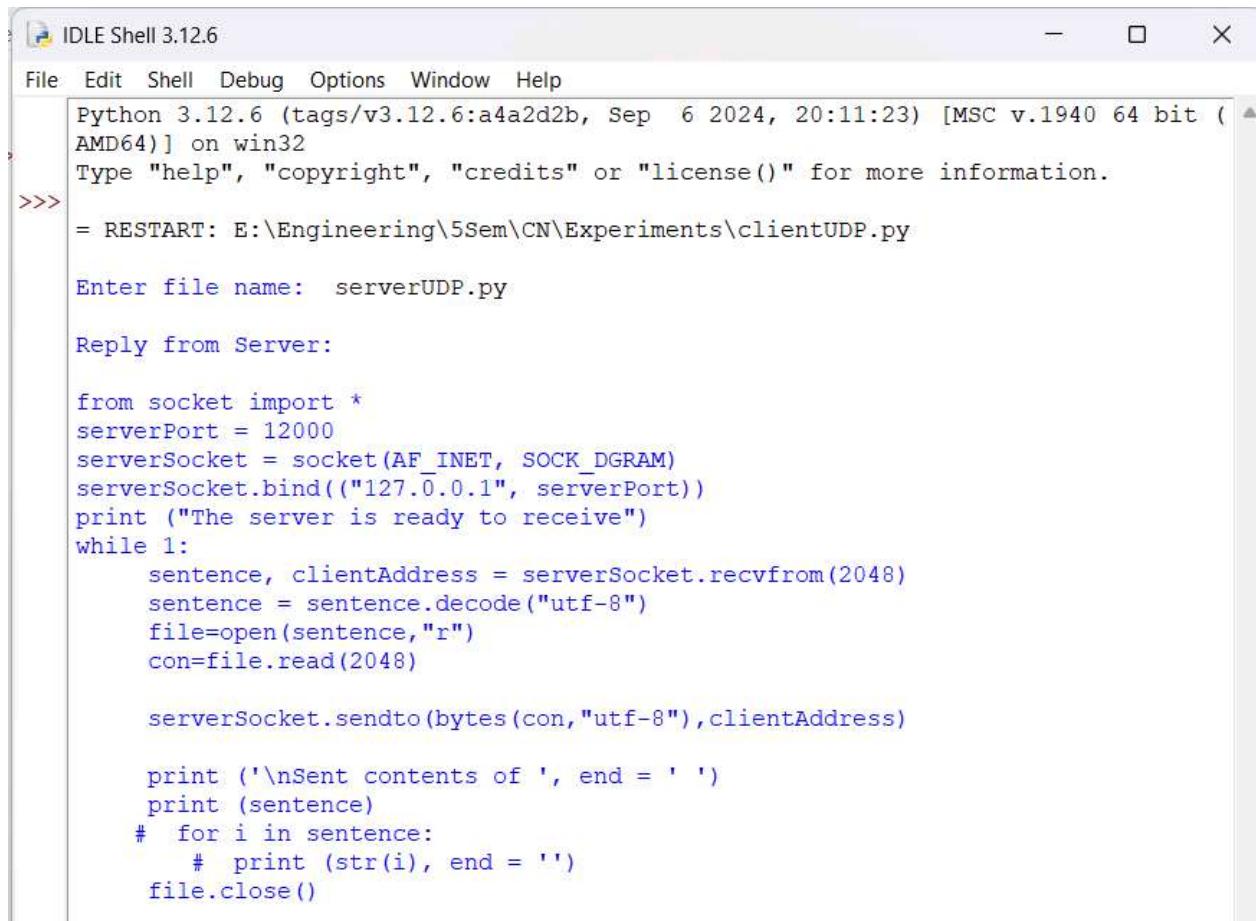
```
serverSocket.sendto(bytes(con,"utf-8"),clientAddress)
```

```
print ('\nSent contents of ', end = ' ')
print (sentence)
# for i in sentence:
```

```
# print(str(i), end = '')
file.close()
```

iii. Screen shots/ output

Client



The screenshot shows the Python IDLE Shell 3.12.6 interface. The window title is "IDLE Shell 3.12.6". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The shell area displays the following text:

```
Python 3.12.6 (tags/v3.12.6:a4a2d2b, Sep  6 2024, 20:11:23) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> = RESTART: E:\Engineering\5Sem\CN\Experiments\clientUDP.py

Enter file name: serverUDP.py

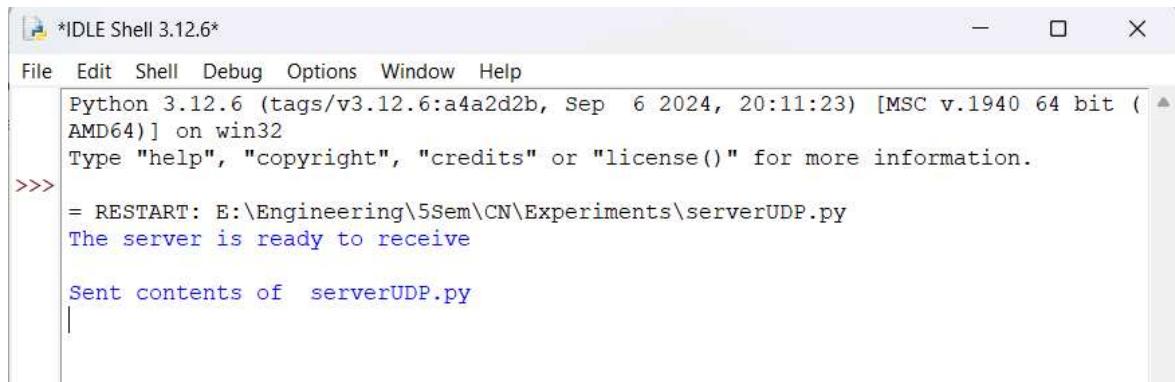
Reply from Server:

from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file=open(sentence,"r")
    con=file.read(2048)

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

    print ('\nSent contents of ', end = ' ')
    print (sentence)
#   for i in sentence:
#       #   print (str(i), end = '')
    file.close()
```

Server



The screenshot shows the Python IDLE Shell 3.12.6 interface. The window title is "*IDLE Shell 3.12.6*". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The shell area displays the following text:

```
Python 3.12.6 (tags/v3.12.6:a4a2d2b, Sep  6 2024, 20:11:23) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> = RESTART: E:\Engineering\5Sem\CN\Experiments\serverUDP.py
The server is ready to receive

Sent contents of  serverUDP.py
```

iv. Observation

D) Using UDP sockets write a client-server program to make client sending the file name and the server to send back the contents of requested file

Client UDP.py

```
from socket import *
SN = '127.0.0.1'
SP = 12000
CS = socket(AF_INET, SOCK_DGRAM)
sentence = input("Enter filename")
CS.sendto(sentence.encode("utf-8"), (SN, SP))
filecontents, serveraddr = CS.recvfrom(2048)
print("Reply from server\n")
print(filecontents.decode("utf-8"))
CS.close()
CS.close()
```

Server UDP.py

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print("The server is ready to receive")
while 1:
```

```
sentence, clientAddress = serverSocket.recvfrom(2048)
sentence = sentence.decode("utf-8")
file = open(sentence, "r")
con = file.read(2048)
serverSocket.sendto(con.encode("utf-8"), clientAddress)
print("\n Sent contents of", end=" ")
print(sentence)
file.close()
```

UDP output

client :-

Enter filename: server UDP.py

Reply from server:

```
from socket import *
serverport = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind((("127.0.0.1", serverport)))
print("The server is ready to receive")
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file = open(sentence, "r")
    cm = file.read(2048)
    serverSocket.sendto(bytes(cm, "utf-8"), clientAddress)
    print("\nSent contents of", end=" ")
    print(sentence)
    file.close()
```

Server

The server is ready to receive

Sent contents of server UDP.py