

# **VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“JnanaSangama”, Belgaum -590014, Karnataka.**



## **LAB RECORD**

### **Computer Network Lab (23CS5PCCON)**

*Submitted by*

**Sohan A R(1BM22CS285)**

*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 **Sohan A R (1BM22CS285)**, 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.

Sneha P Assistant Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE
--	--

## Index-Cycle-I

<b>Sl. No.</b>	<b>Date</b>	<b>Experiment Title</b>	<b>Page No.</b>
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	5
3	25/10/2024	Configure default route, static route to the router	10
4	08/11/2024	Configure DHCP within a LAN and outside LAN.	16
5	22/11/2024	Configure RIP routing Protocol in Routers	21
6	22/11/2024	Configure OSPF routing protocol	24
7	22/11/2024	Demonstrate the TTL/ Life of a Packet	30
8	08/11/2024	Configure Web Server, DNS within a LAN.	32
9	20/12/2024	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)	36
10	20/12/2024	To understand the operation of TELNET by accessing the router in server room from a PC in IT office.	39
11	20/12/2024	To construct a VLAN and make the PC's communicate among a VLAN	42
12	20/12/2024	To construct a WLAN and make the nodes communicate wirelessly	46

Github Link:

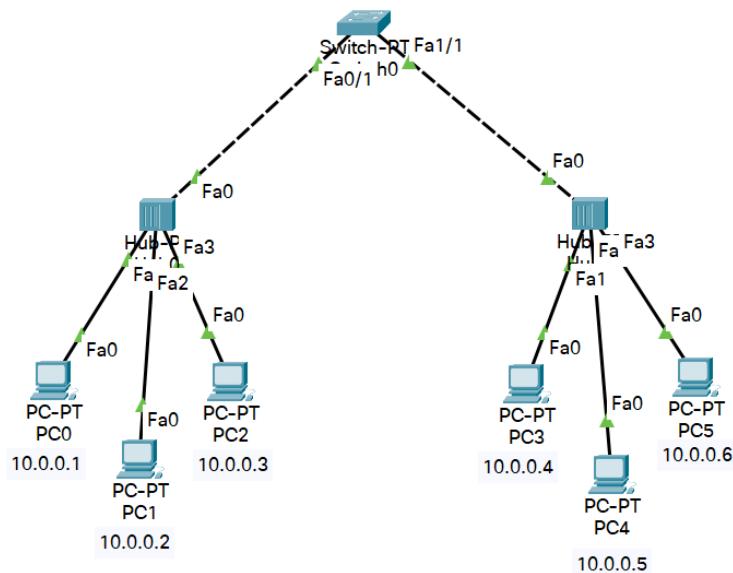
## **Index-Cycle-II**

<b>Sl. No.</b>	<b>Date</b>	<b>Experiment Title</b>	<b>Page No.</b>
1	15/11/2024	Write a program for error detecting code using CRC-CCITT (16-bits).	50
2	15/11/2024	Write a program for congestion control using Leaky bucket algorithm	52
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.	54
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.	57

# Cycle-I

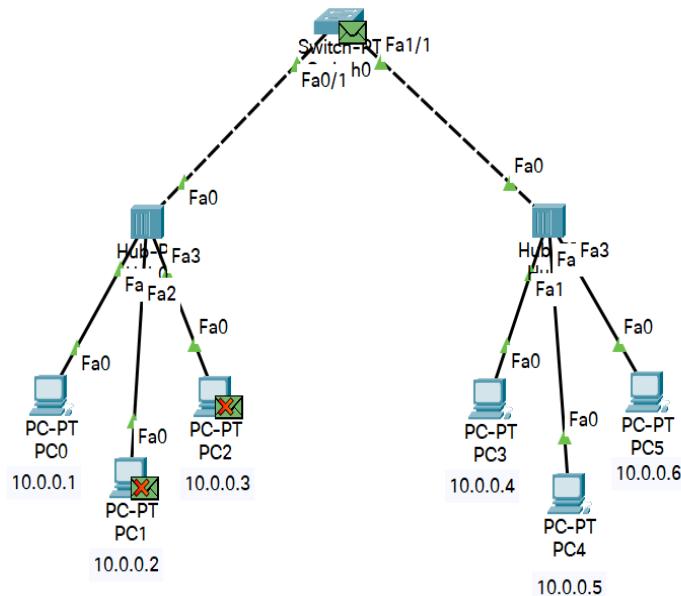
## Program 1

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

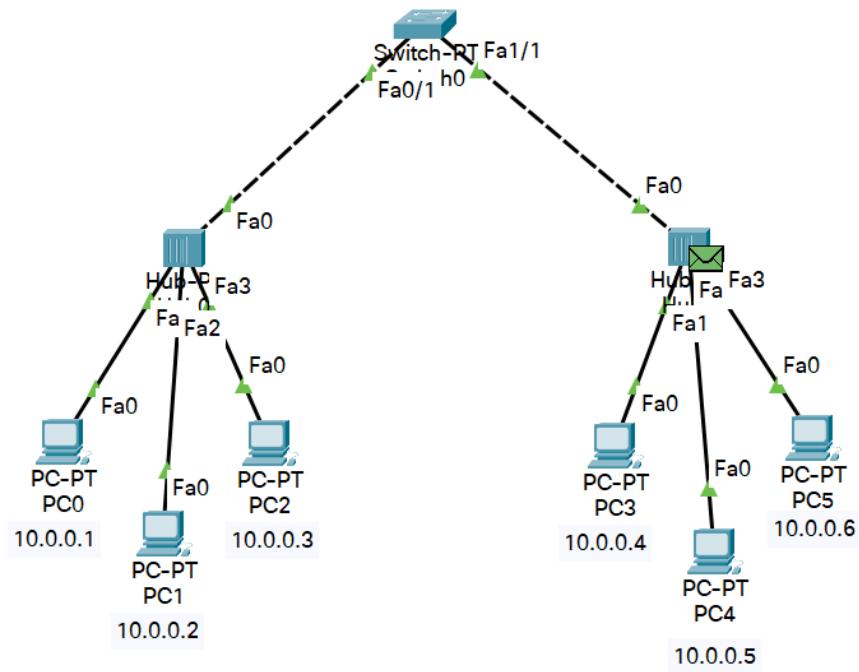


- iii. Screen shots/ output

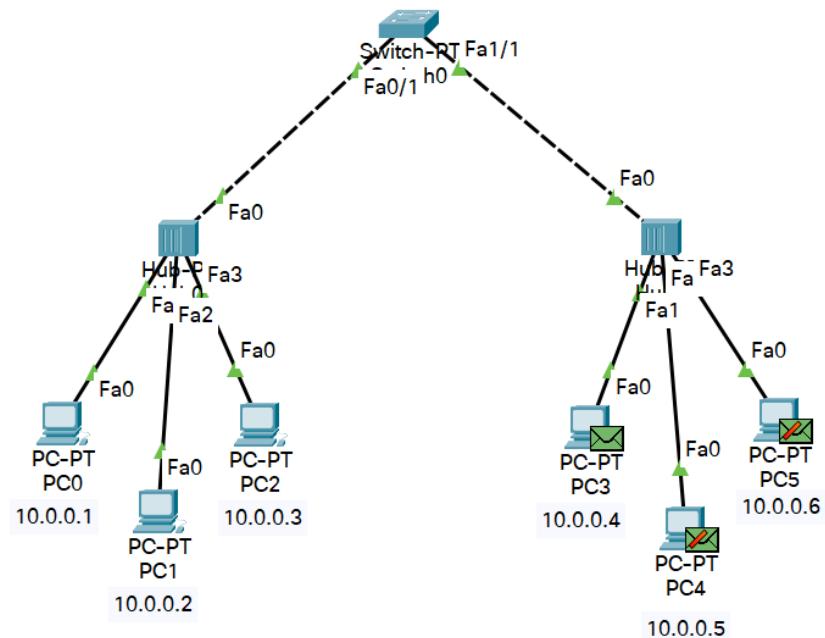
Hub behaviour at sending end



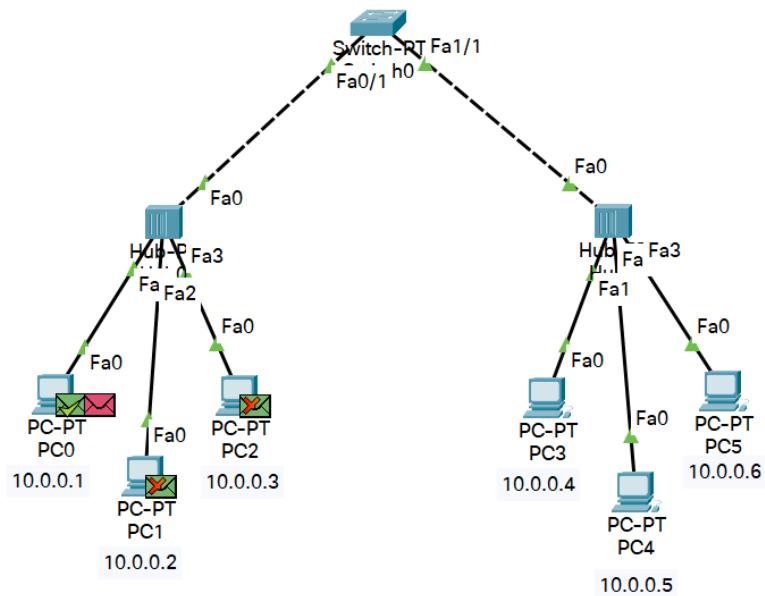
## Switch behaviour



## Hub behaviour at receiving end



## Hub behaviour when back to sender



Ping command to connectivity

A screenshot of a Windows Command Prompt window titled "PC0". The window shows the output of a ping command from PC0 to PC4. The output is as follows:

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

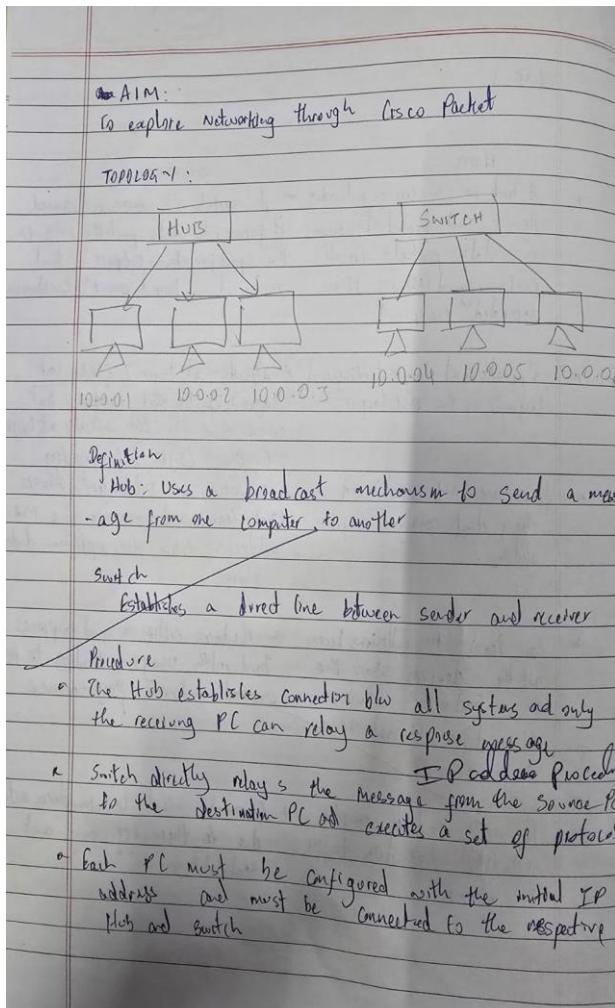
Pinging 10.0.0.4 with 32 bytes of data:

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



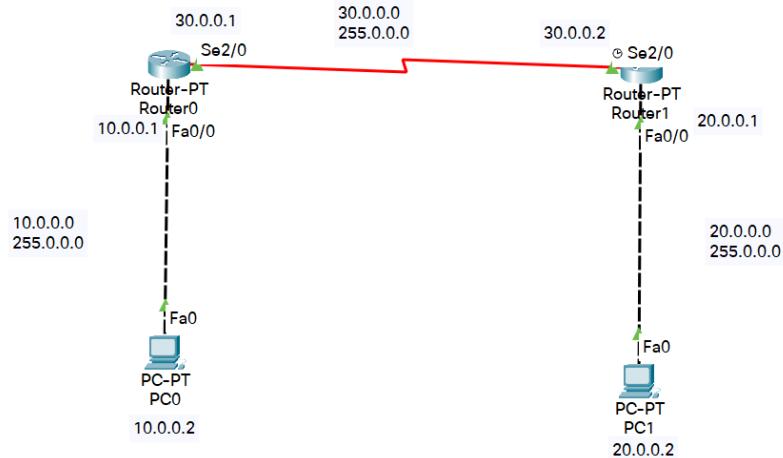
**Bafna Gold**  
Date: \_\_\_\_\_ Page: \_\_\_\_\_

**Observation :**  
The difference between Hub and switch is that the Hub relays the senders message to all the external devices connected to it and only the receiving device accepts the packet whereas a switch establishes a direct connection between the sending and receiving devices

27/2/24

## **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

```
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

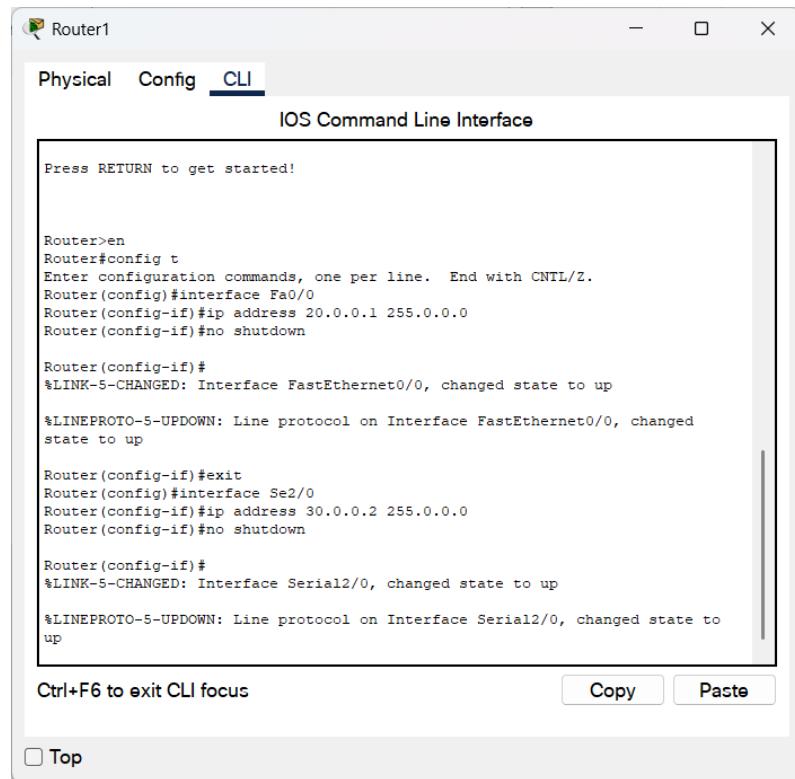
Router(config-if)#exit
Router(config)#interface Se2/0
Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#no shutdown

%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
```

The screenshot shows the Cisco IOS Command Line Interface (CLI) for Router0. The user is in configuration mode, entering commands to configure the FastEthernet0/0 and Serial2/0 interfaces. The FastEthernet0/0 interface is assigned the IP address 10.0.0.1 with a subnet mask of 255.0.0.0 and is set to 'no shutdown'. The Serial2/0 interface is assigned the IP address 30.0.0.1 with a subnet mask of 255.0.0.0 and is also set to 'no shutdown'. Log messages indicate the state transitions of these interfaces, showing them going from down to up. The interface names are highlighted in blue.

## Router1 configuration



The image shows a window titled "Router1" with a tab bar containing "Physical", "Config", and "CLI". The "CLI" tab is selected. Below the tabs is the title "IOS Command Line Interface". A message box says "Press RETURN to get started!". The main area contains the following configuration commands:

```
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 left is the text "Ctrl+F6 to exit CLI focus". On the right are "Copy" and "Paste" buttons. At the very bottom is 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)

PC1

Physical Config Desktop Programming

Command Prompt X

```
Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 20.0.0.1: Destination host unreachable.

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

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**

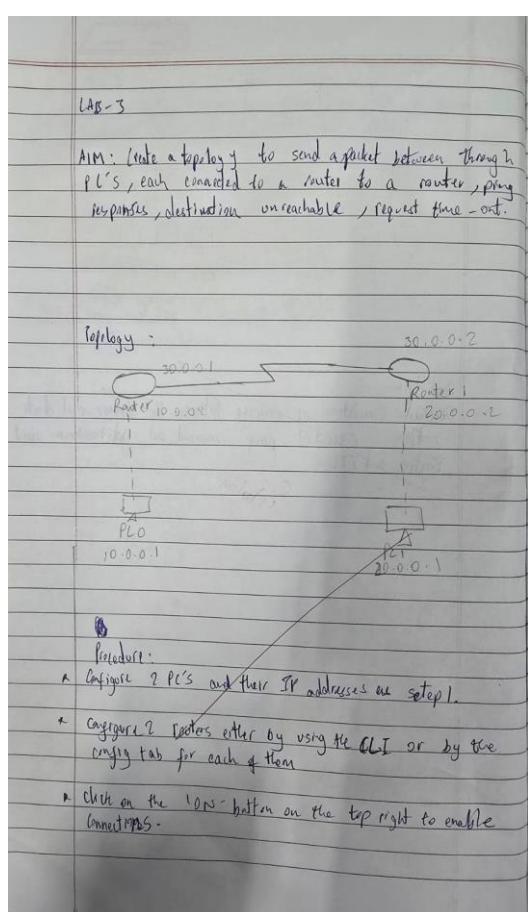
```
Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.2

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



Router 1 configuration:

Output	Source	Dest	Type	Color	Time (sec)	Rank
Successful	PC0	PC1	ICMP	Green	0.000	N
Failed	PC0	PC1	ICMP	Red	0.000	N

*(Handwritten notes: Router 1 configuration, successful and failed outputs, and a signature at the bottom right.)*

**LAB-2**  
 Create a topology involving multiple hubs and a switch connecting them to simulate a simple PDN.

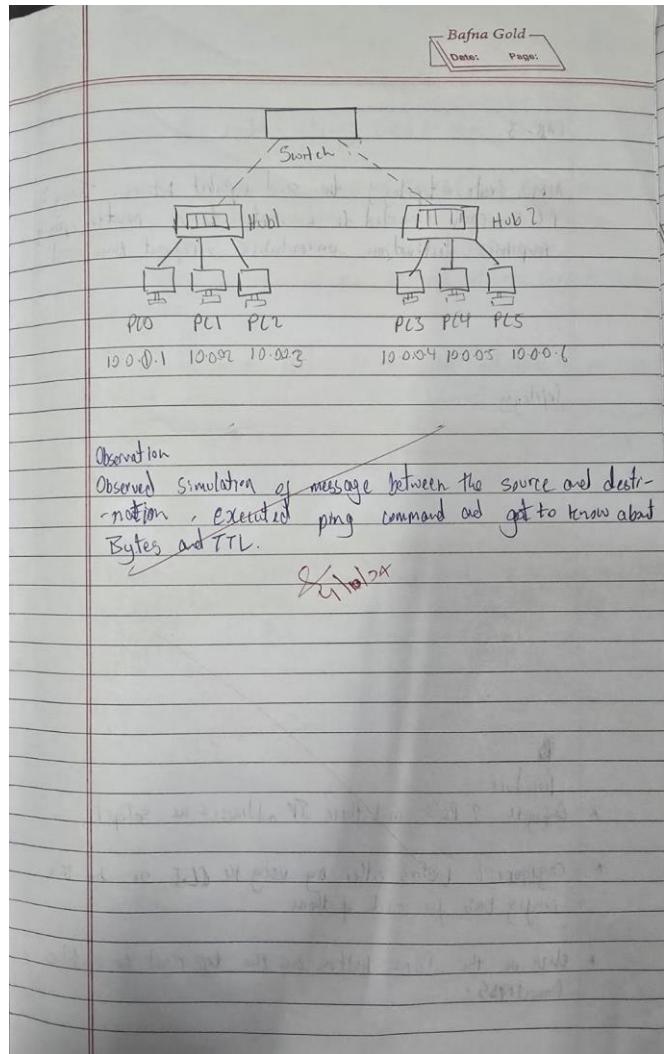
**AIM:**  
 To obtain the topology making the hubs and a switch

**REQUIREMENTS:**

**STEPS**

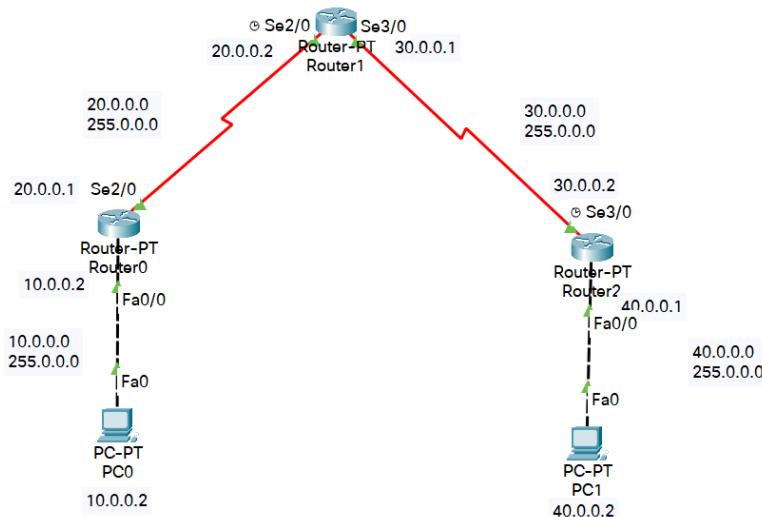
- ① Drag and drop PC's - ad hubs and switches and establish the shown below
- ② Configure each PC
  - Click on PC
  - Go to config
  - Go to fast ethernet and give a IP address ex: 10.0.0.1
- ③ Select simulation
  - Click message
  - Later click source and destination
- ④ Running the Ping command
  - Click on PC
  - Go to desktop
  - Go to Command prompt
  - Run command ping 10.0.0.3

**Topology:**



### **Program 3**

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



- iii. Screen shots/ output

Router0 configuration

```
Router0
Physical Config CLI
IOS Command Line Interface
Would you like to enter the initial configuration dialog? [yes/no]: n
Press RETURN to get started!

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

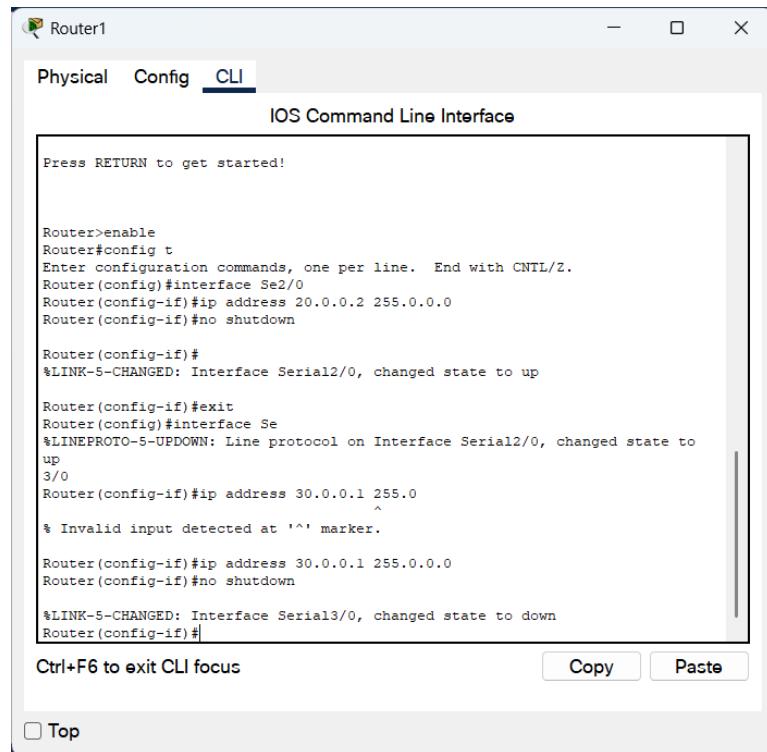
Router(config-if)#
$LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

$LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed
state to up
$IP-4-DUPLICADDR: Duplicate address 10.0.0.2 on FastEthernet0/0, sourced by
00C.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)#
Ctrl+F6 to exit CLI focus
Copy Paste
Top
```

## Router1 configuration



The screenshot shows a Windows-style application window titled "Router1". The tab bar at the top has three tabs: "Physical", "Config", and "CLI", with "CLI" being the active tab. The main area is labeled "IOS Command Line Interface". It displays the following configuration commands:

```
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 30.0.0.1 255.0.0.0
          ^
% Invalid input detected at '^' marker.

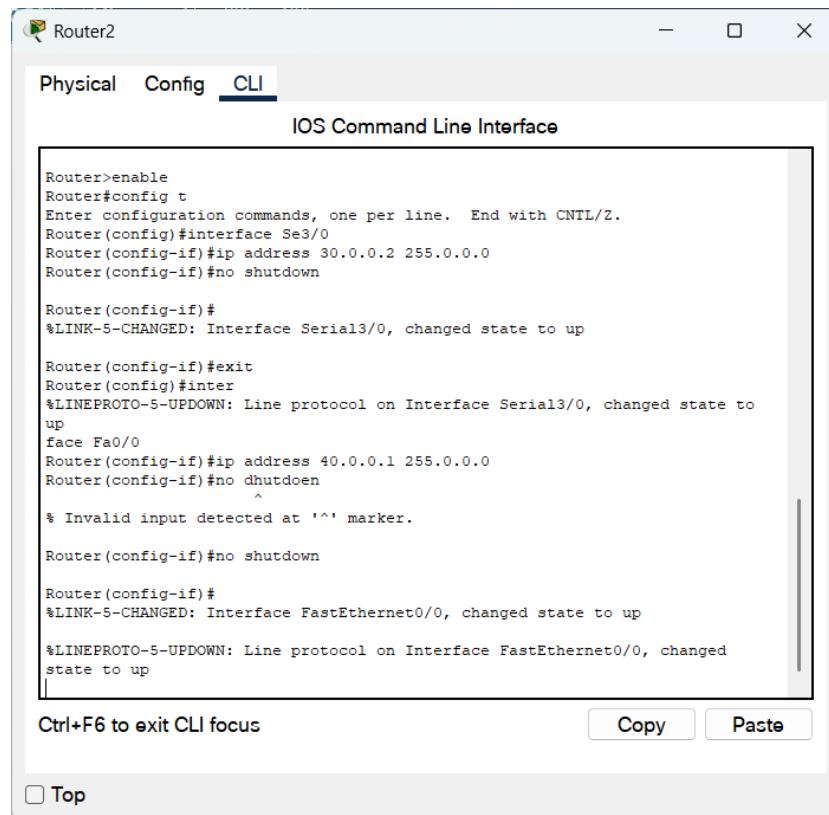
Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#no shutdown

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

Ctrl+F6 to exit CLI focus
```

At the bottom of the window, there are "Copy" and "Paste" buttons, and a checkbox labeled "Top".

## Router2 configuration



The screenshot shows a Windows-style application window titled "Router2". The tab bar at the top has three tabs: "Physical", "Config", and "CLI", with "CLI" being the active tab. The main area is labeled "IOS Command Line Interface". It displays the following configuration commands:

```
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 dhtddo
          ^
% 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
```

At the bottom of the window, there are "Copy" and "Paste" buttons, and a checkbox labeled "Top".

## Static Routing:

Router0

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     

Top

Router1

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

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

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

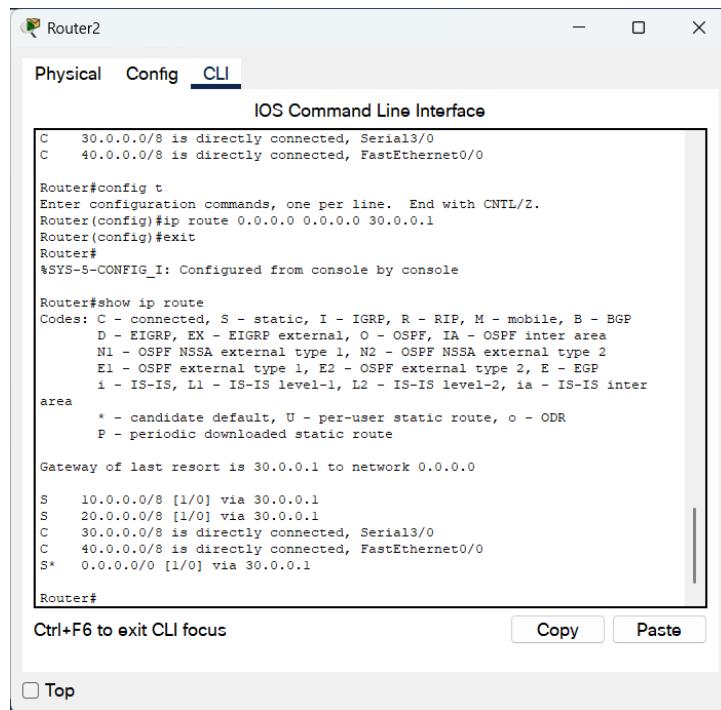
C  10.0.0.0/8 is directly connected, FastEthernet0/0
C  20.0.0.0/8 is directly connected, Serial2/0
S  30.0.0.0/8 [1/0] via 20.0.0.2
S  40.0.0.0/8 [1/0] via 20.0.0.2
S* 0.0.0.0/0 [1/0] via 20.0.0.2

Router#
```

Ctrl+F6 to exit CLI focus     

Top

## Router2



The screenshot shows the Router2 CLI interface. The title bar says "Router2". The tabs at the top are "Physical", "Config", and "CLI", with "CLI" being the active tab. The main window is titled "IOS Command Line Interface". It displays the following configuration and route information:

```
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#
```

At the bottom of the window, there are "Copy" and "Paste" buttons, and a "Top" link.

## Pinging:

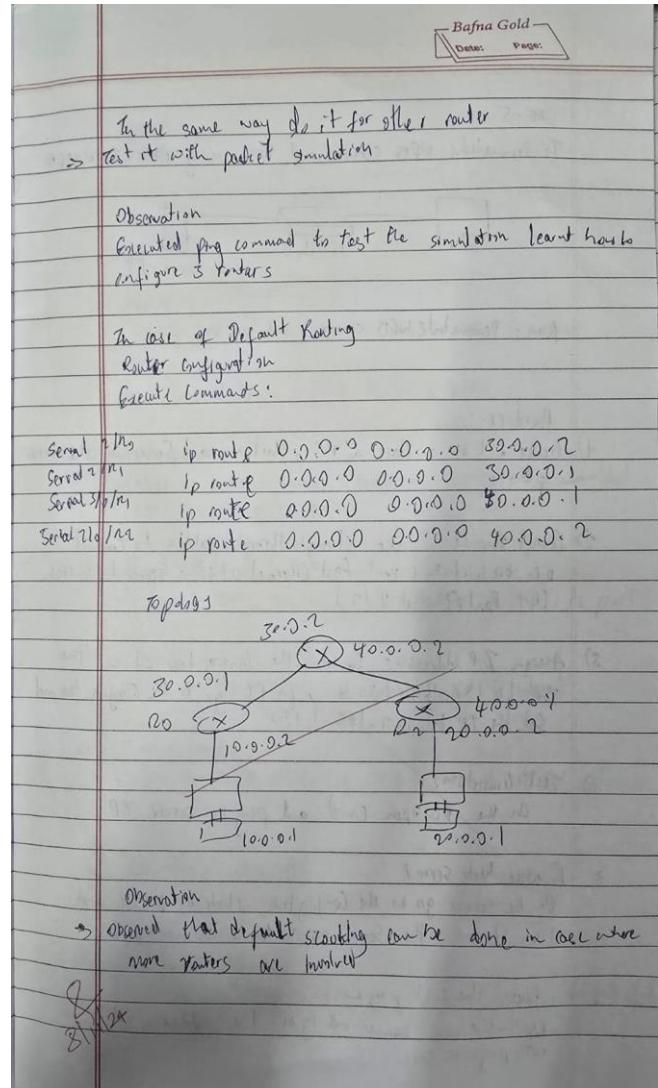
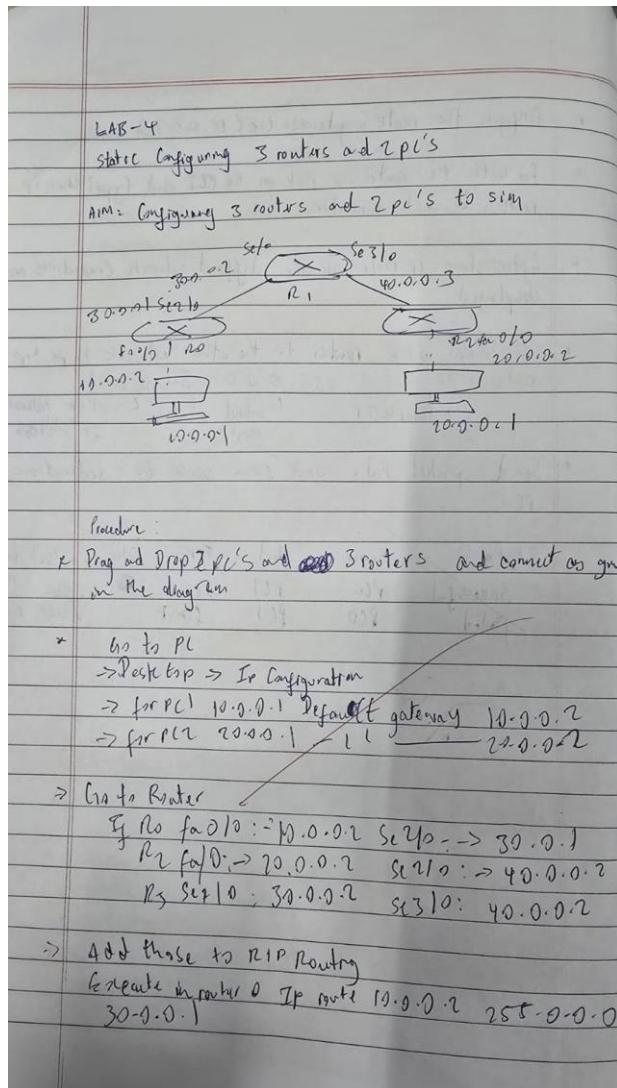
```
C:\>ping 40.0.0.2

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

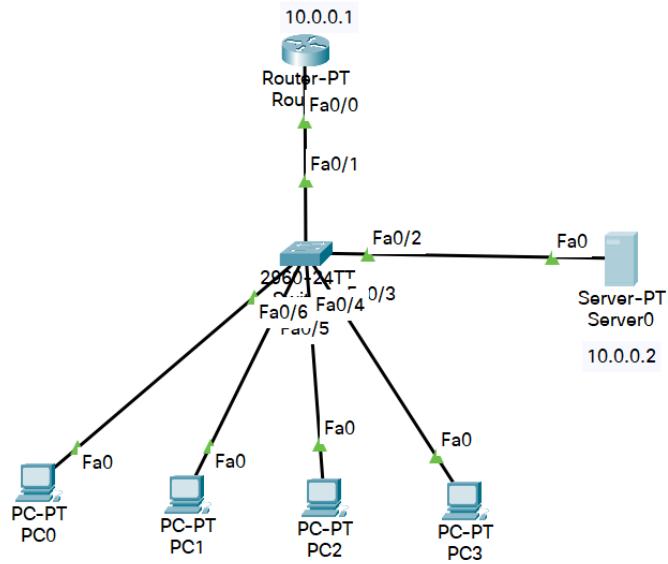
Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 25ms, Average = 16ms
```

#### iv. Observation



## **Program 4**

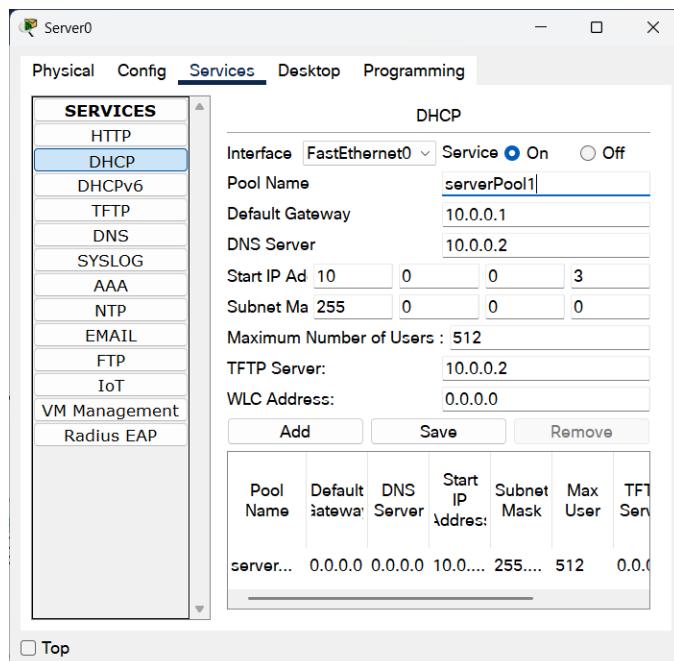
- i. Configure DHCP within a LAN and outside LAN.
- ii. Procedure along with the topology



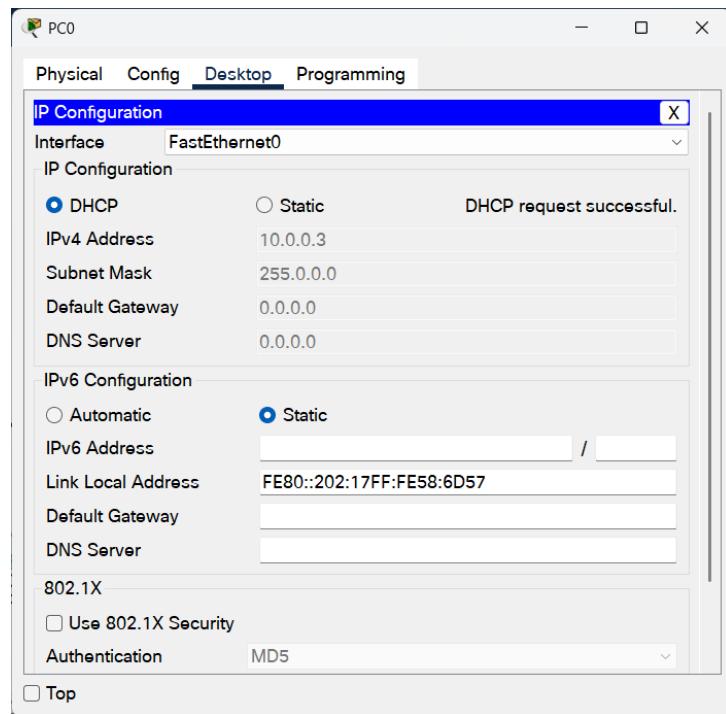
- iii. Screen shots/ output

DHCP Within LAN

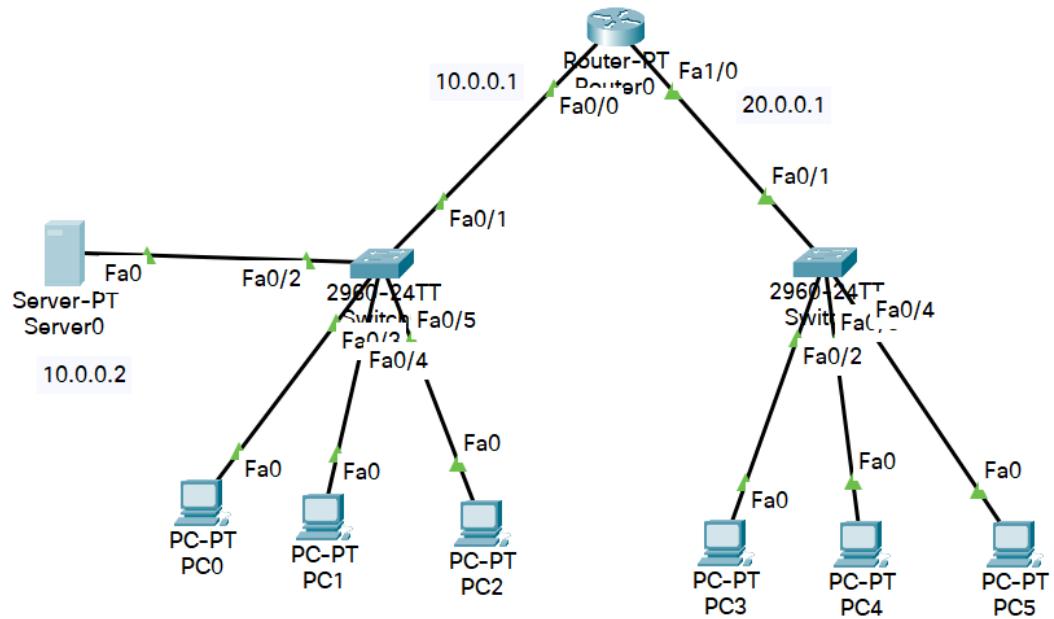
DHCP Configuration



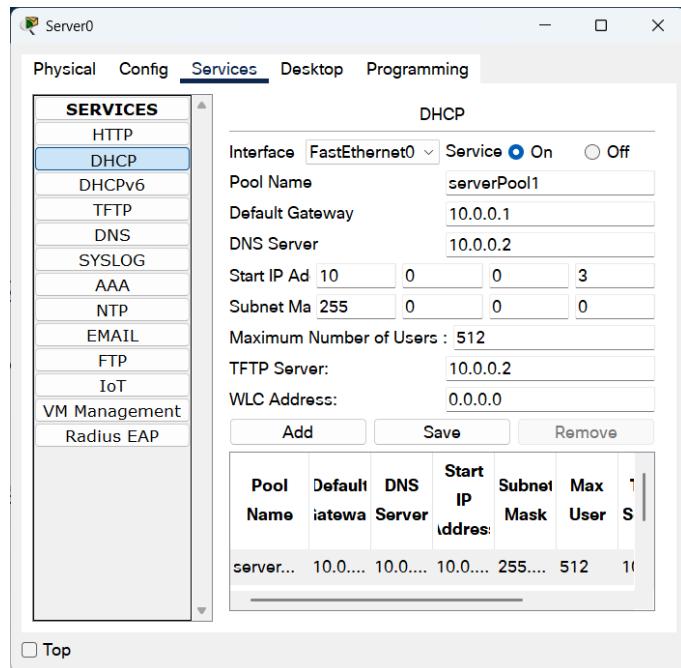
## PC settings



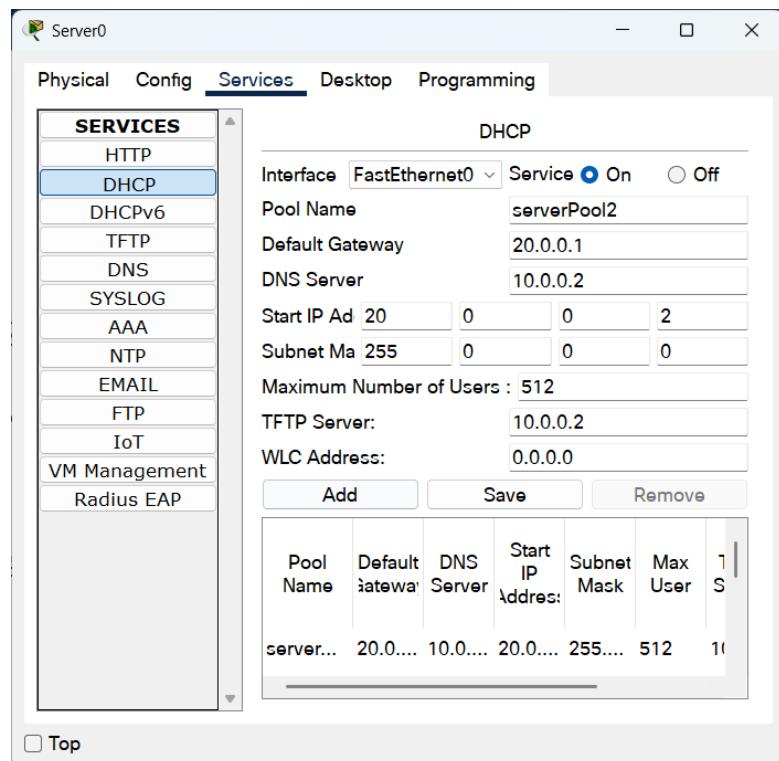
DHCP outside LAN:



DHCP configuration for inside LAN



DHCP configuration for outside LAN



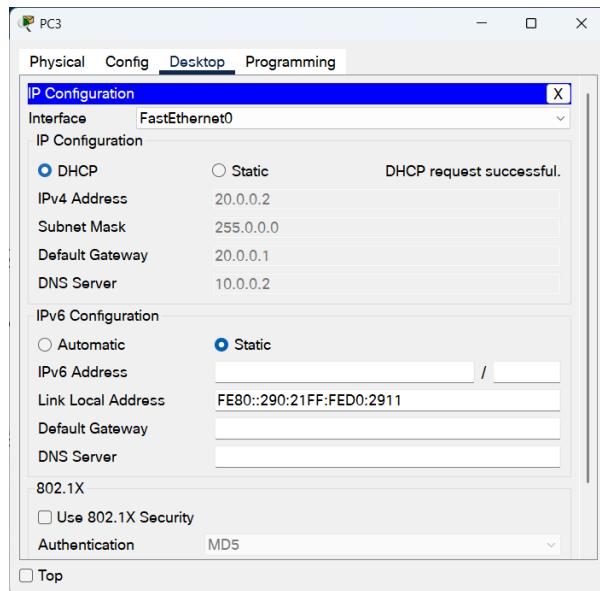
Ip helper command in Router

```

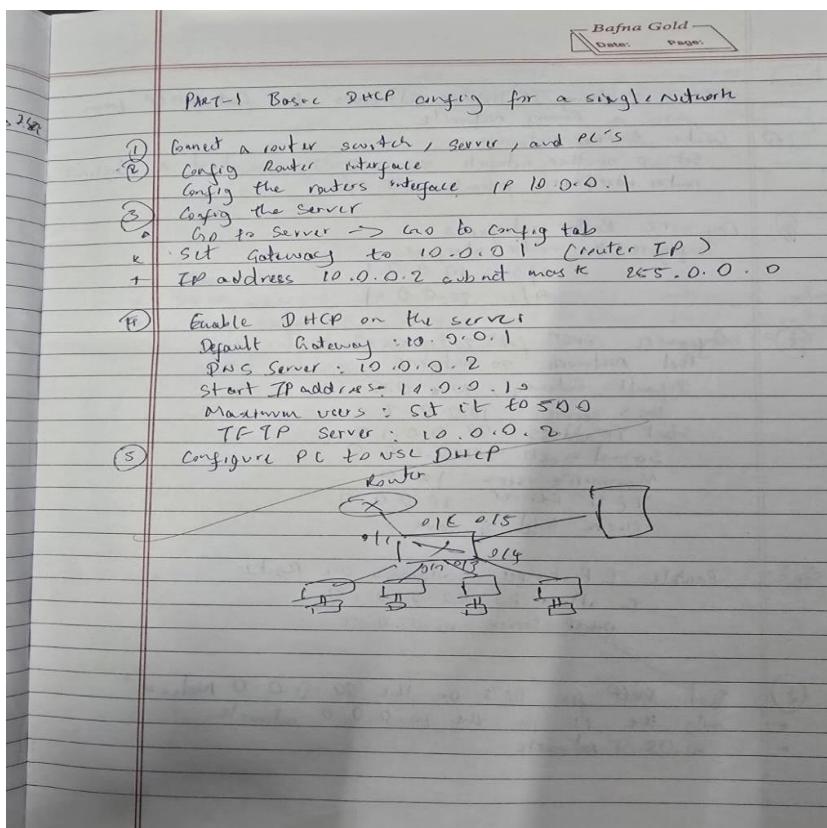
Router(config-if)#exit
Router(config)#interface Fa1/0
Router(config-if)#ip helper-address 10.0.0.2
Router(config-if)#

```

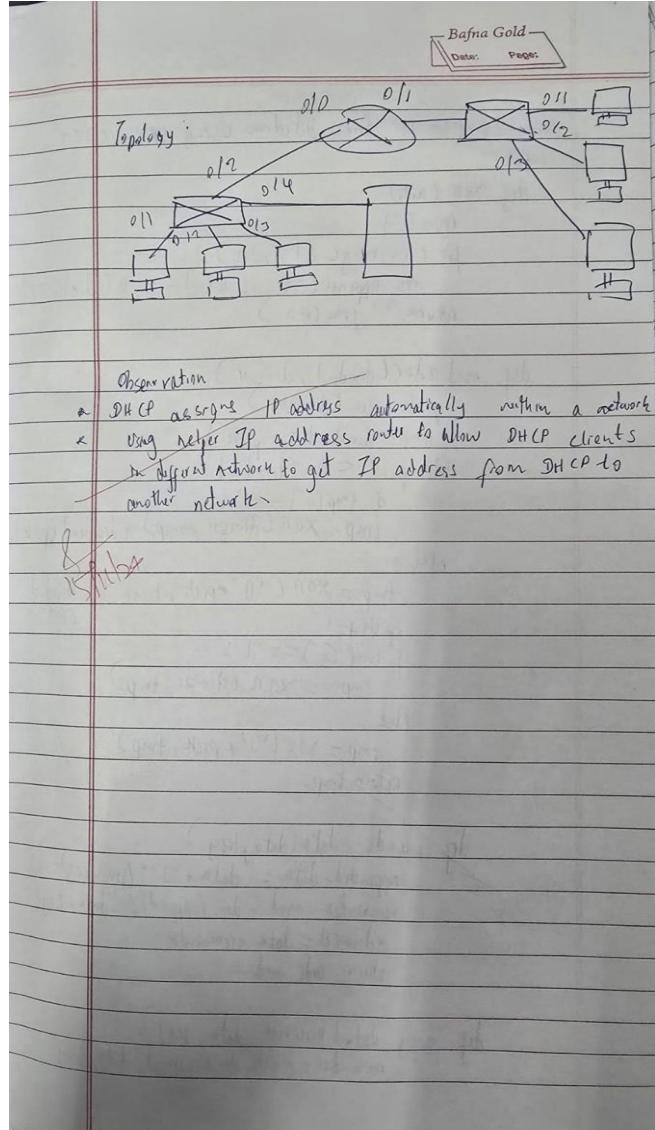
PC setting in another network



#### iv. Observation

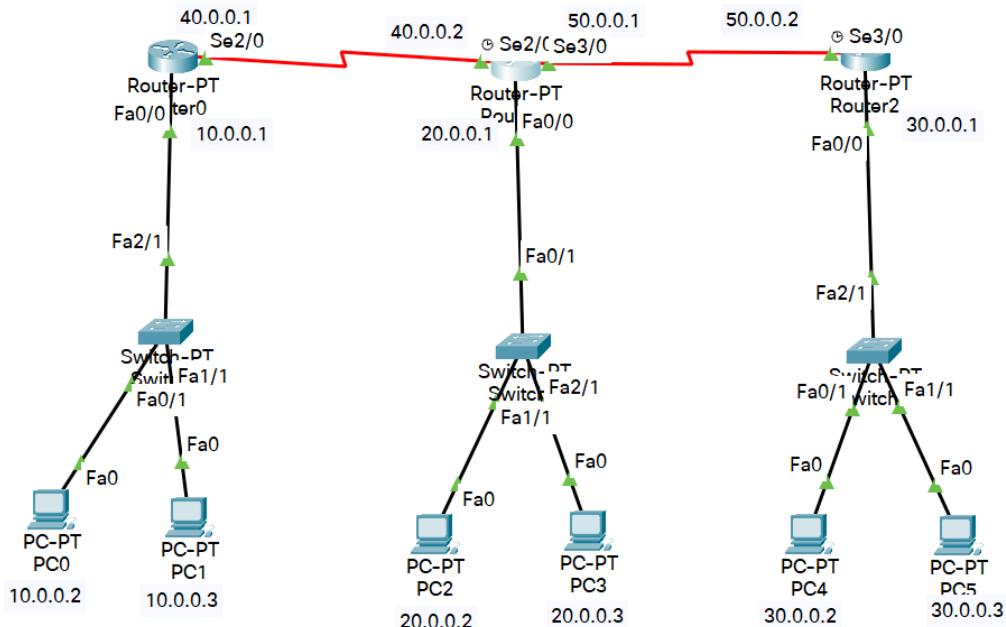


- Part-2 : Using IP Helper Addresses to Get DHCP from a Remote Network
- (1) Create Another Network  
Set up another network 20.0.0.0 connected a different router interface parenthesised 0/1
  - (2) Configure Router interfaces  
Configure the router interfaces
    - 0/0 : 10.0.0.1
    - 0/1 : 20.0.0.1
  - (3) Configure DHCP for 20.0.0.0  
Pool Network : 20 Network  
Default Gateway : 20.0.0.1  
DNS Server : 10.9.9.2  
Start IP address : 20.0.0.10  
Subnet mask : 255.0.0.0  
Maximum users : 100  
TFTP server : 10.0.0.2  
Click, hold & save
  - (4) Enable IP helper address on Router  
To allow to 20.0.0.0  
DHCP Server 10.0.0.2
  - (5) Test DHCP for PCs on the 20.0.0.0 Network  
only the PC in the 20.0.0.0 network  
10.0.0.0 network



## **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:

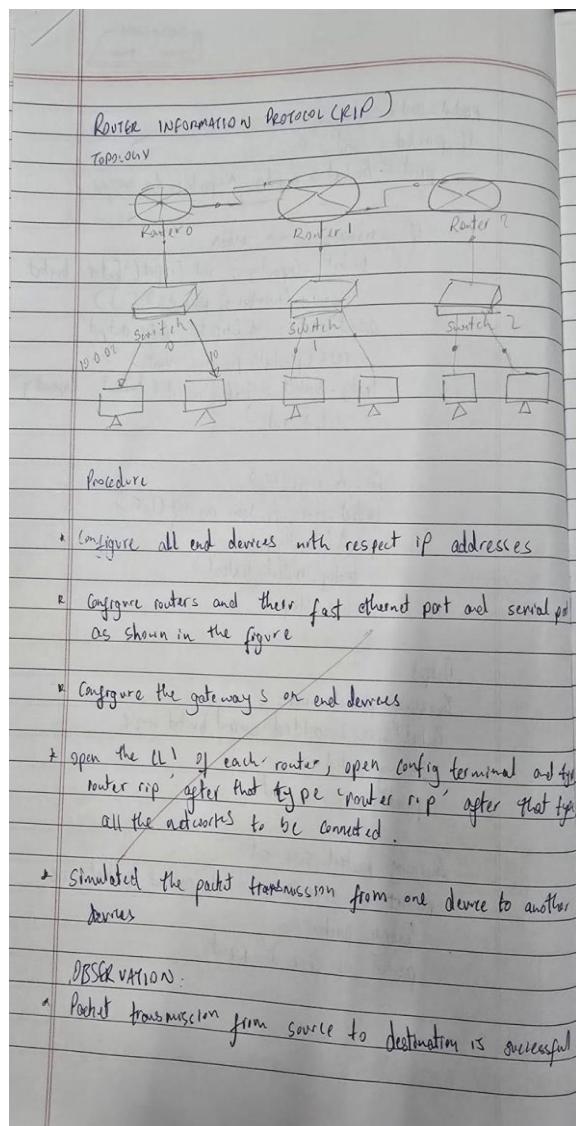
```
Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.2

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

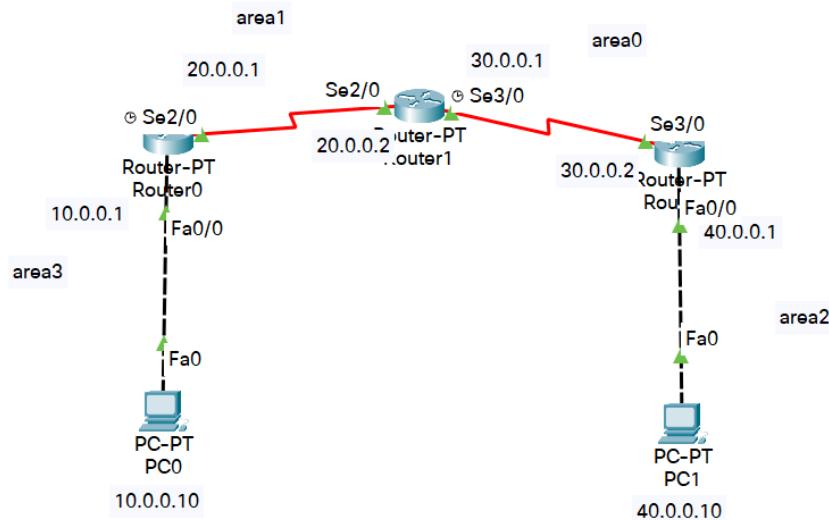
Ping statistics for 20.0.0.2:
  Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
  Minimum = 1ms, Maximum = 9ms, Average = 6ms
```

iv. Observation



## **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

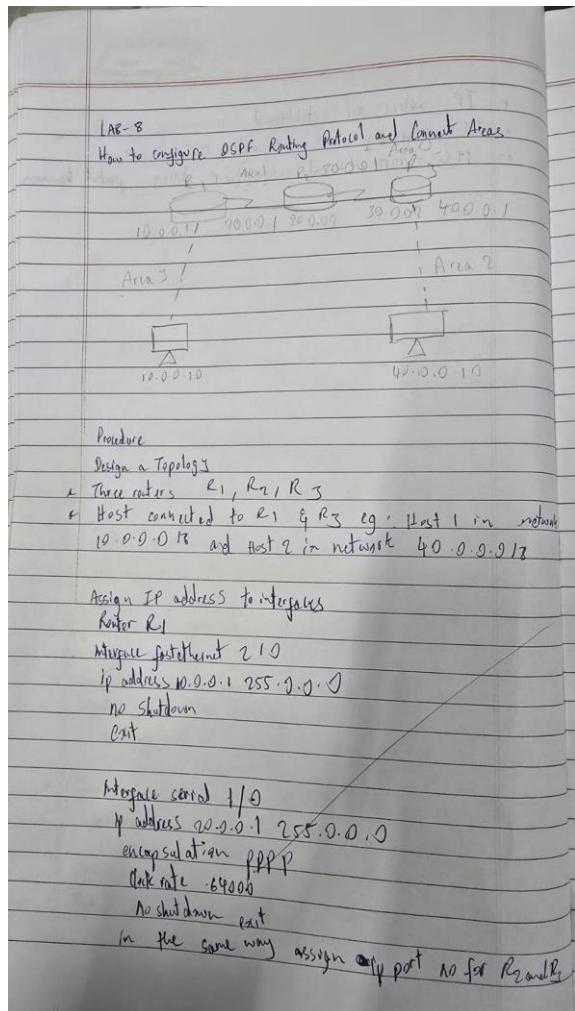
```
C:\>ping 40.0.0.10

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



Bafna C. I Date: \_\_\_\_\_

Step 3: Configure OSPF on all routers  $R_1$

```
# border ospf 1
# router-id 1.1.1.1
# network 10.0.0.0 0.255.255.255 area 0
# network 20.0.0.0 0.255.255.255 area 1
# exit
```

$R_2$

```
# router ospf 1
# router-id 2.2.2.2
# network 20.0.0.0 0.255.255.255 area 0
# network 30.0.0.0 0.255.255.255 area 1
# exit
```

$R_3$

```
# border ospf 1
# router-id 3.3.3.3
# network 30.0.0.0 0.255.255.255 area 0
# network 40.0.0.0 0.255.255.255 area 1
# exit
```

Step 4

Configure loopback interface  $R_1$

```
# interface loopback 0
# ip address 172.16.1.252 255.255.0.0
# no shutdown
```

$R_2$

```
# interface loopback 0
# ip address 172.16.1.253 255.255.255.0.0
# no shutdown
```

R3

- # interface loopback 0
- # ip address 172.16.1.254 255.255.0.0
- # no shutdown

Step 5 : test  
show ip route

Step 6  
Configure virtual link

R1

- # router ospf 1
- # area 1 virtual link 9-2-2-2

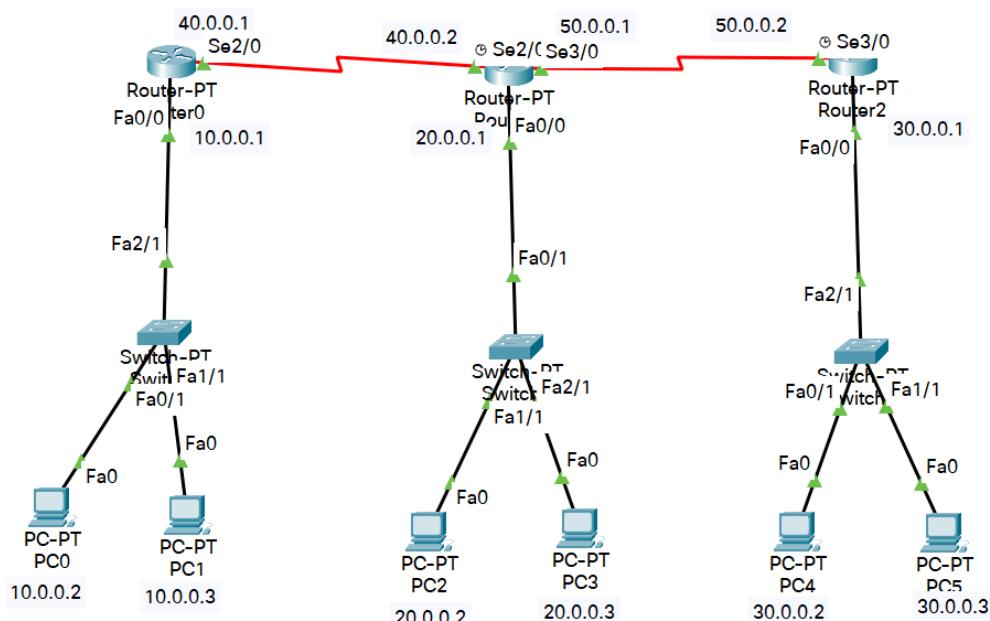
R2

- # router ospf 1
- # area 1 virtual link 1.1.1.1

Step 7  
Test connectivity  
in PC1 ping 40.0.0.10

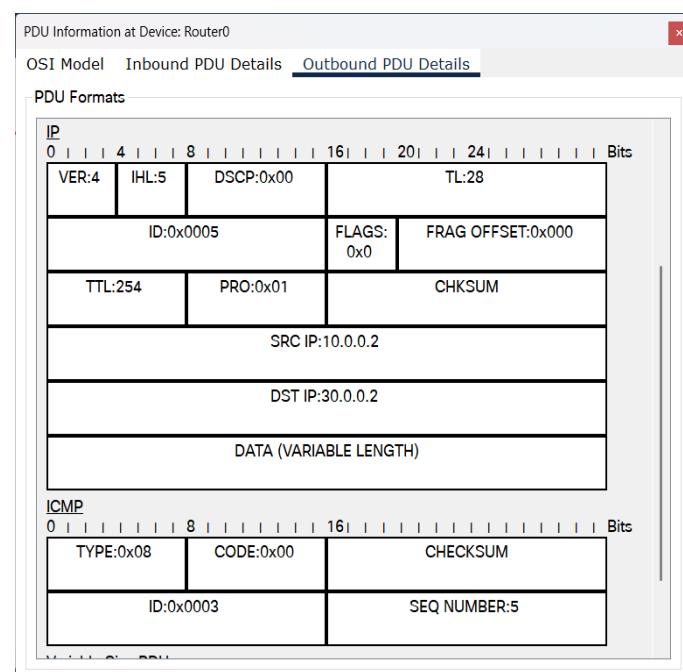
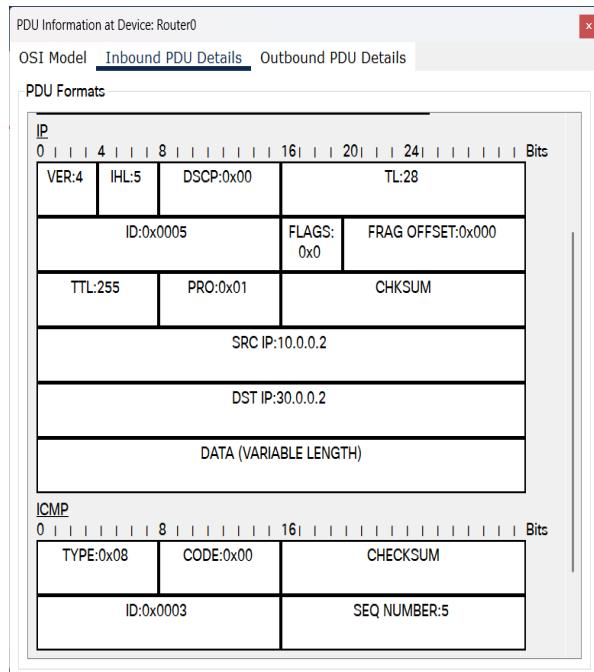
## Program 7

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

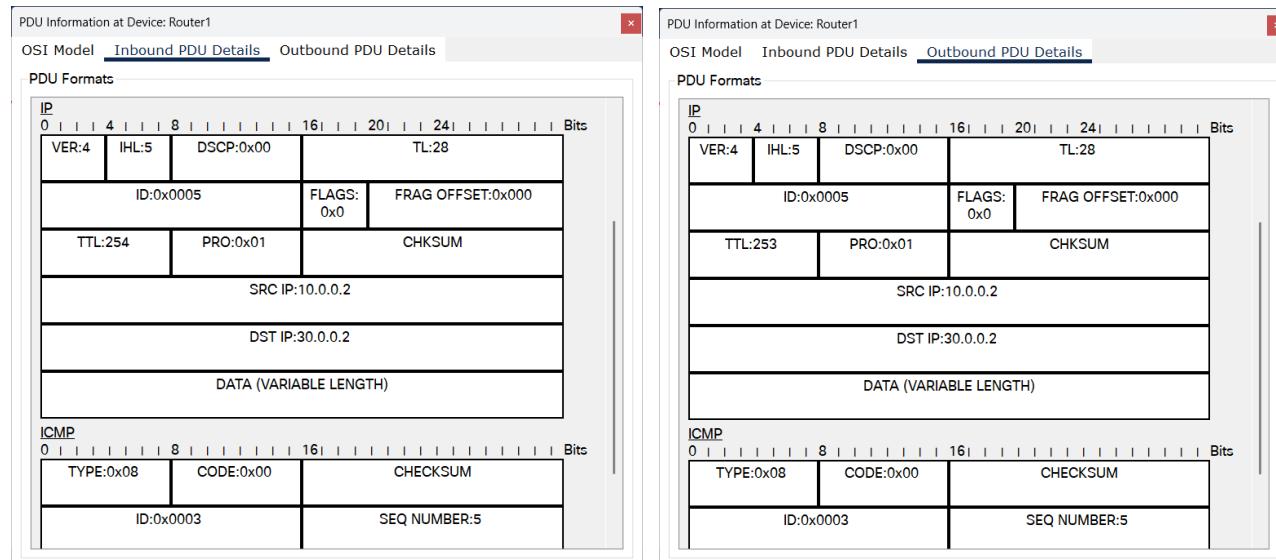


- ### iii. Screen shots/ output

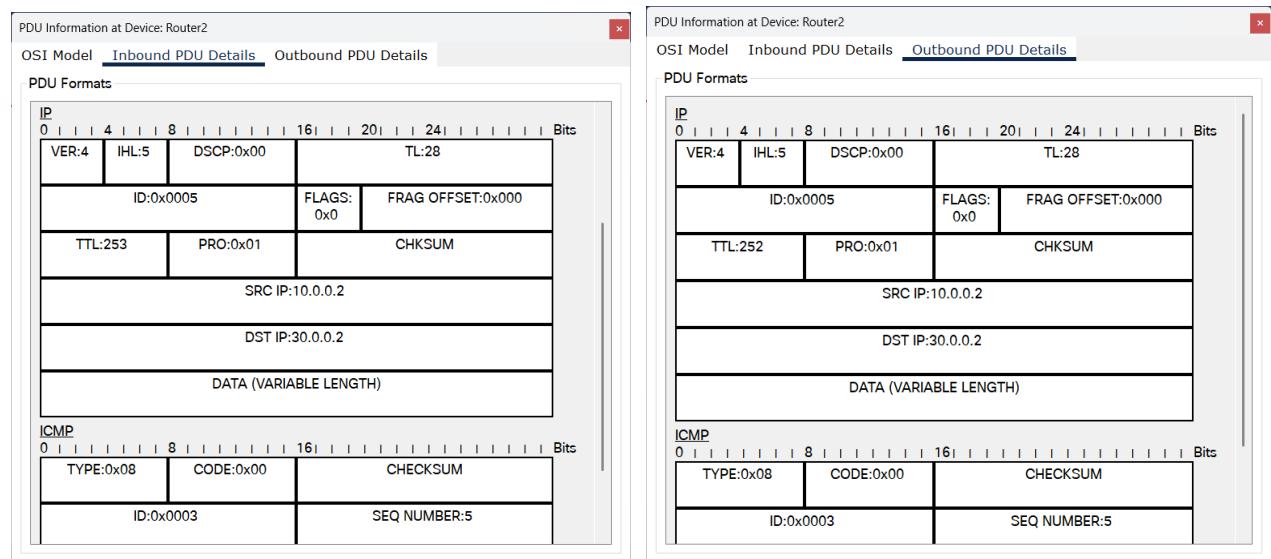
## Packet at Router0



## Packet at Router1

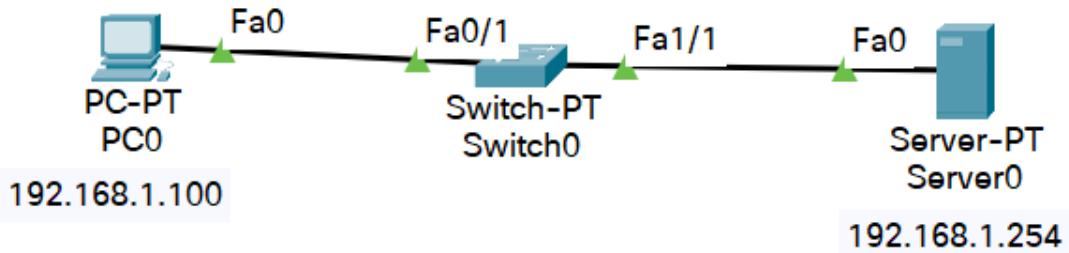


## Packet at Router2



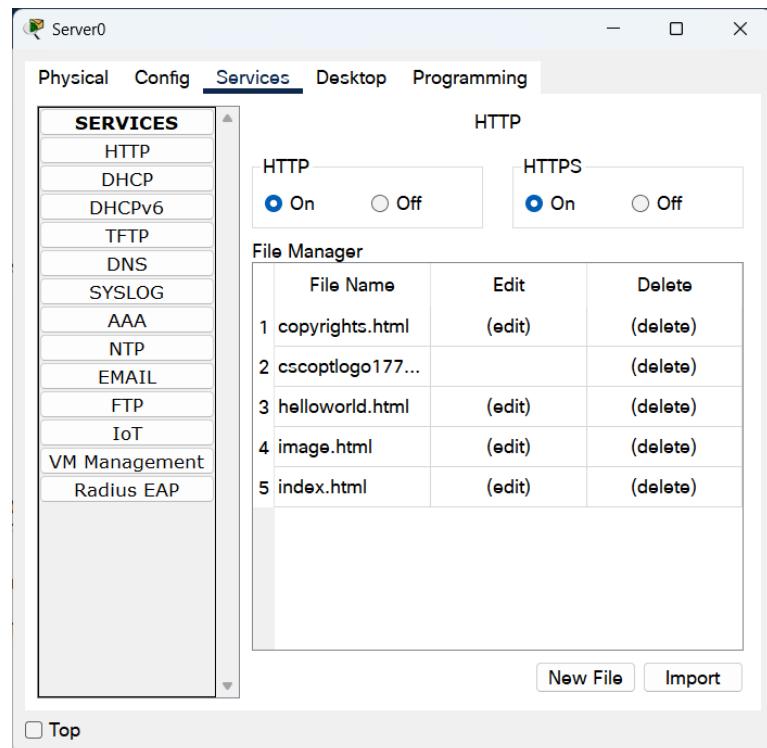
## Program 8

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

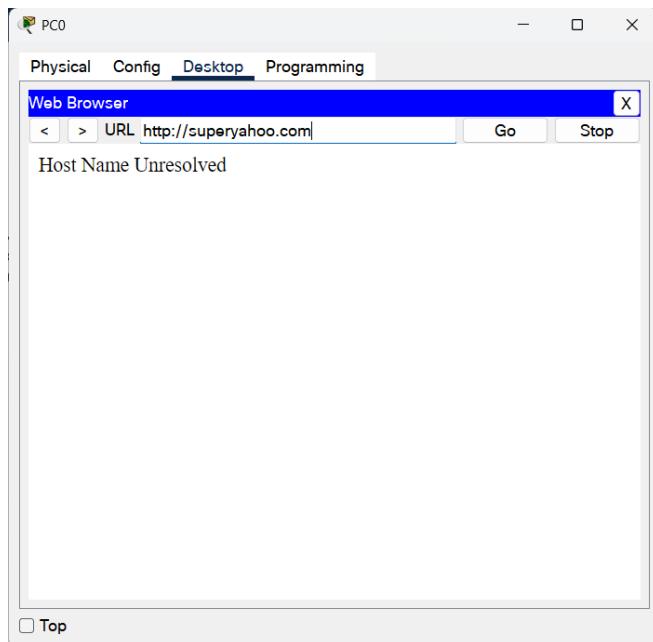
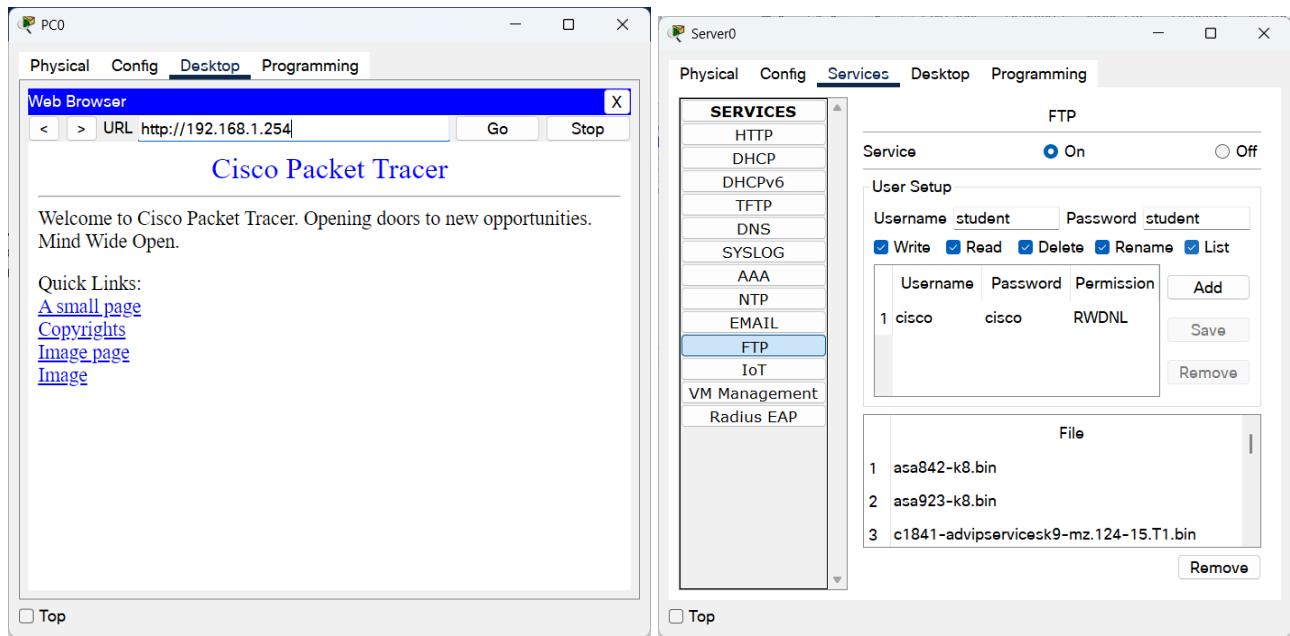


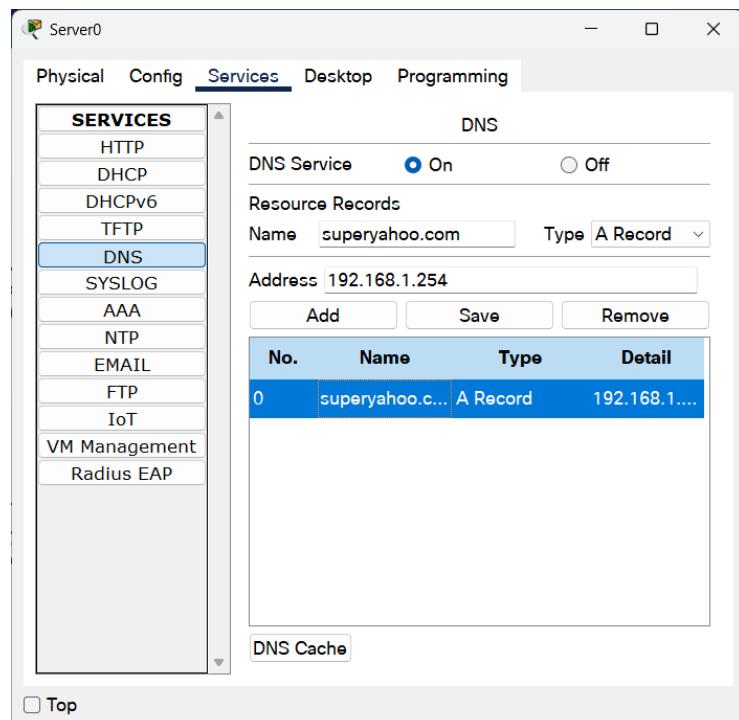
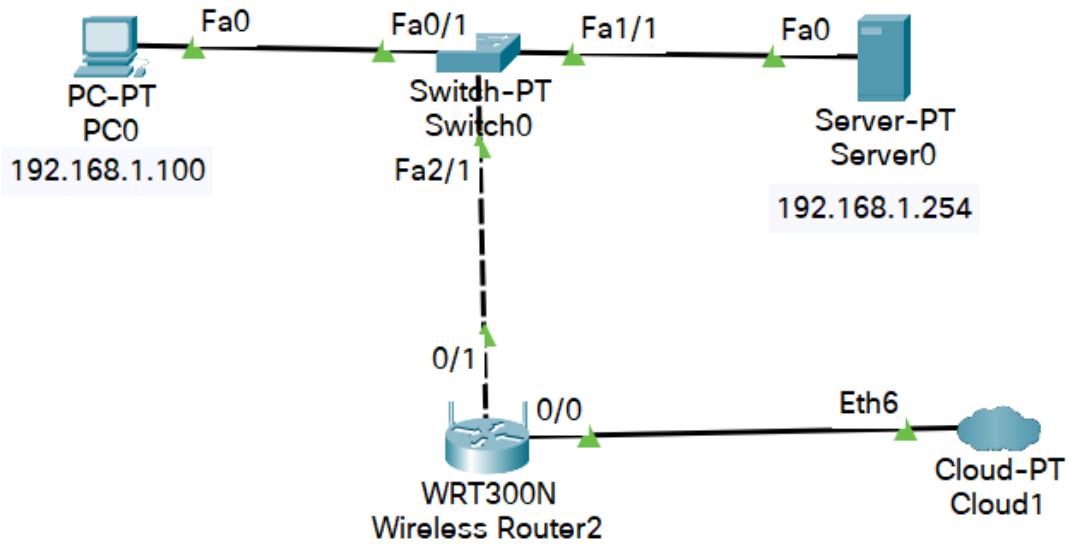
- iii. Screen shots/ output

Server's services

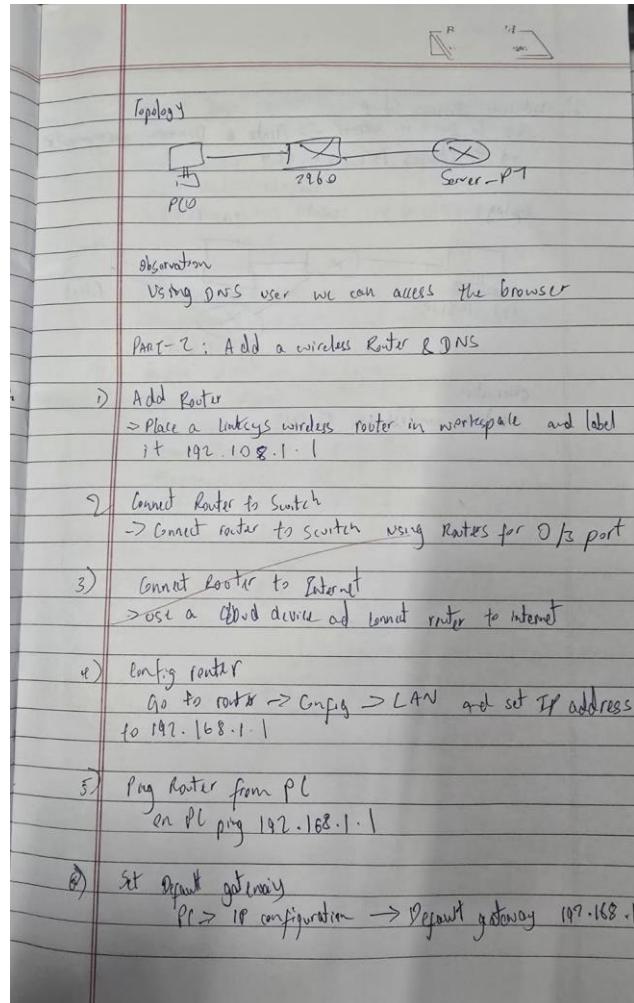
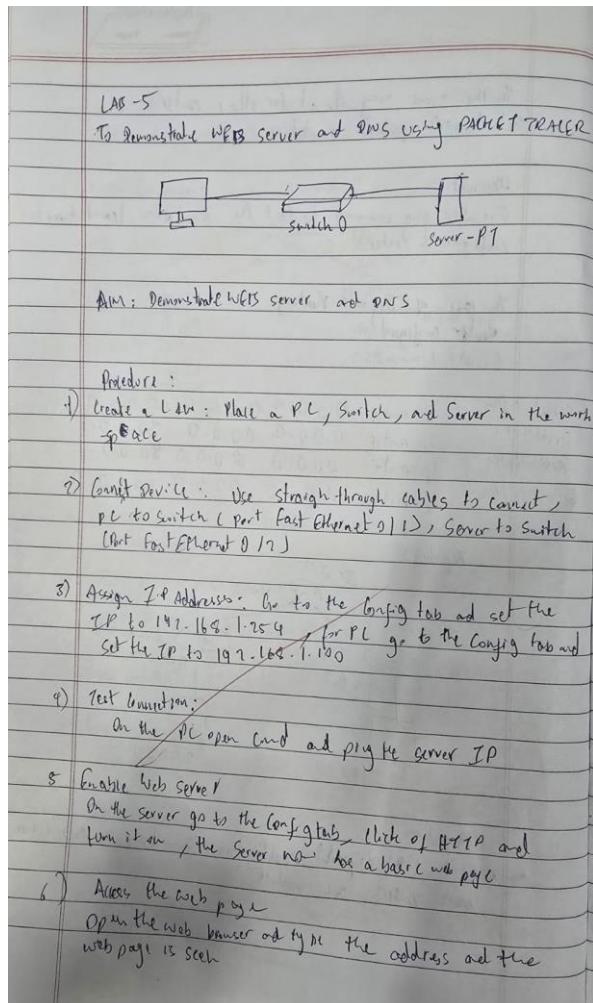


PC's Web Browser





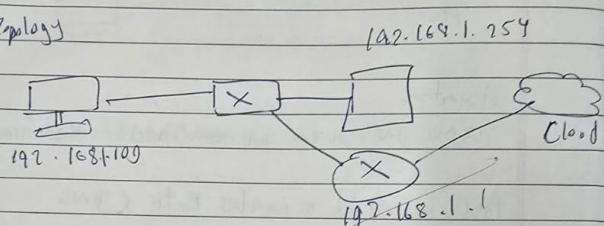
#### iv. Observation



7) Webserver Domain Set up

Go to DNS in server  $\Rightarrow$  Create a Domain superghost<sup>18</sup>  
and IP address 192.168.1.254

Topology

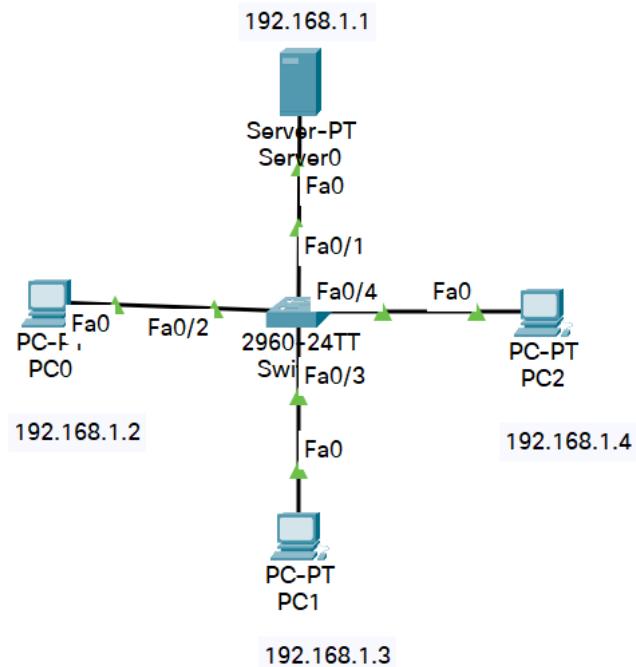


Observation

Used a wireless lan to DNS

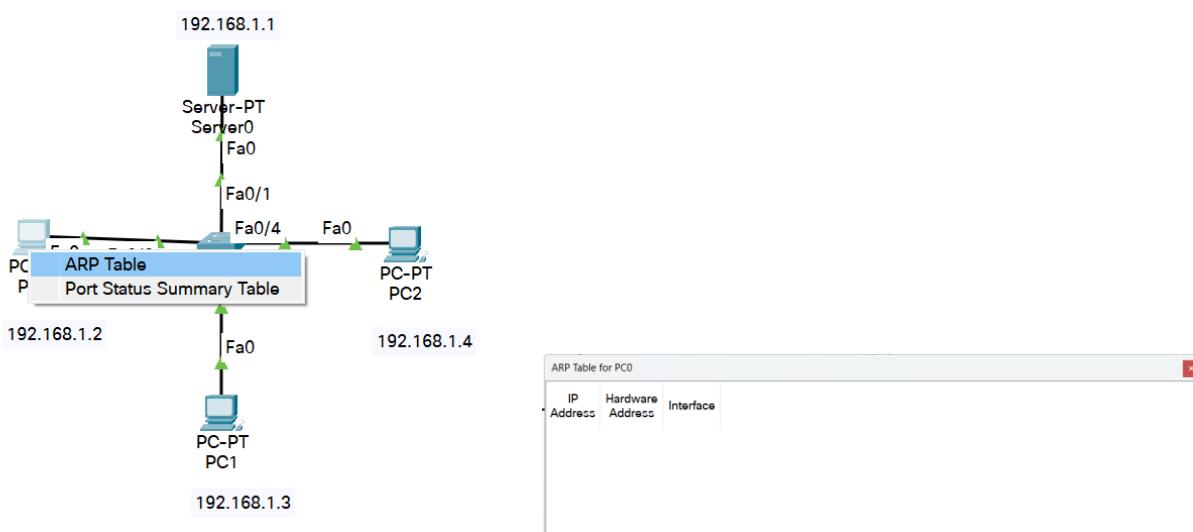
## Program 9

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

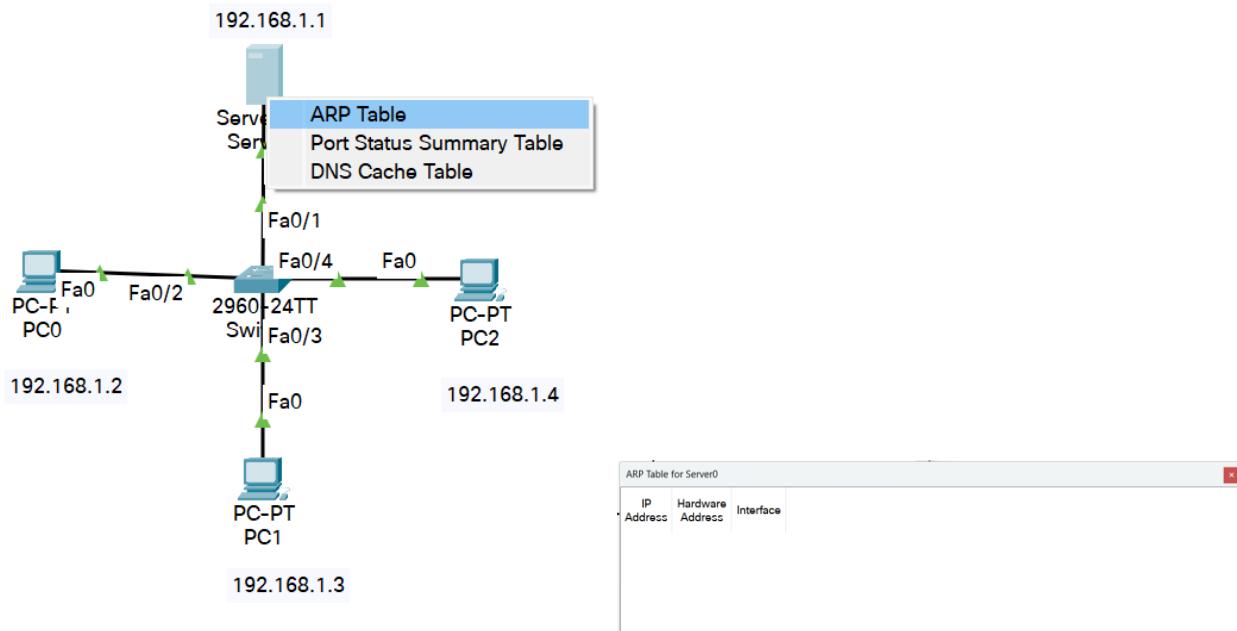


- iii. Screen shots/ output

ARP Table of PC



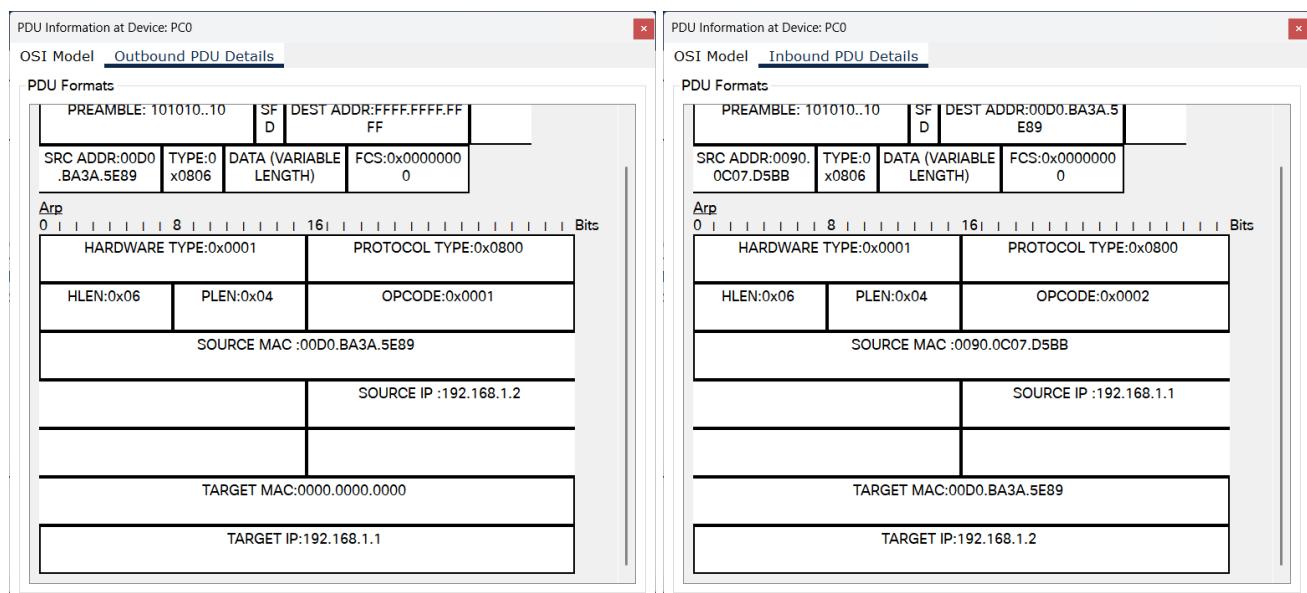
ARP Table of Server



### Command at PC

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

### Pinging in Simulation Mode



#### iv. Observation

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

**ARP Protocol Lab**

- To construct simple LAN and understand the concept & operation of ARP

**Topology :**

```

graph TD
    Server[Server - P1] --- P2[PC - P2]
    Server --- P3[PC - P3]
    Server --- P4[PC - P4]
    P2 --- P3
    P2 --- P4
    P3 --- P4
    style Server fill:#fff,stroke:#000
    style P2 fill:#fff,stroke:#000
    style P3 fill:#fff,stroke:#000
    style P4 fill:#fff,stroke:#000
  
```

**Procedure**

- Assign IP addresses to all PC's and server
- Go to simulation panel, click on inspect and right click on PC
- Notice that there are no entries in ARP Table
- Repeat the same for server
- Click on PC and go to Cmd  
Type arp -n
- Try pinging from PC to server
- Two packets are created ICMP & ARP
- Hover over the packets to check the type of packets
- click on ARP packets

(1) Click and capture button to start the simulation  
(2) Click on capture to set the ICMP packet mode  
- needs to know the MAC address of the server

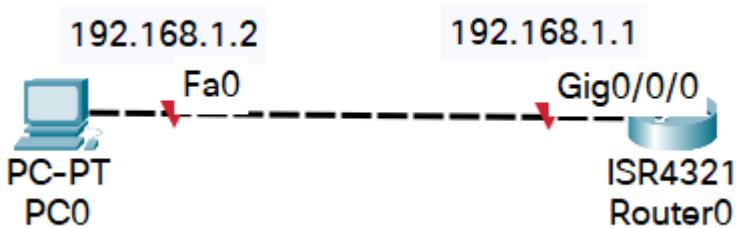
**Observation**

ARP is first used when PC0 attempts to ping the server, since it needs to know the MAC address of the server

X/2/2\*

## **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

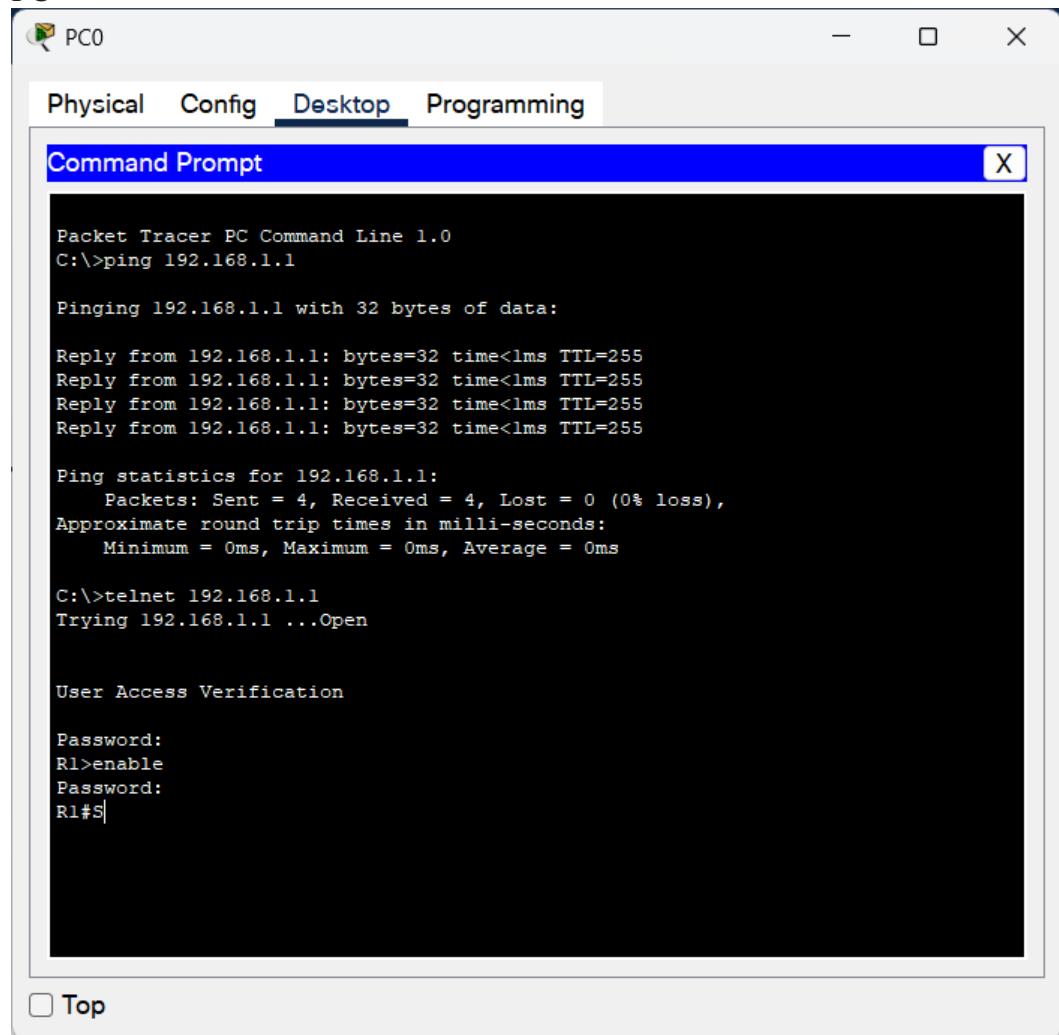
R1(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to up

%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



Top

#### iv. Observation

Bufna Gold  
Date: \_\_\_\_\_  
Page: \_\_\_\_\_

TELNET (CONTINUATION)

Topology

Procedure

- ① Create the topology
- ② Configure the IP address and gateway for PCO
- ③ Configure the router in CLI
- ④ Go to PC - Cmd

type Ping 192.168.1.1  
telnet 192.168.1.1

Observation

1. Ping

- + Successful pings indicate that the router is forwarding ICMP packets correctly between different subnets. ARP helps to map IP address

2. Telnet :-

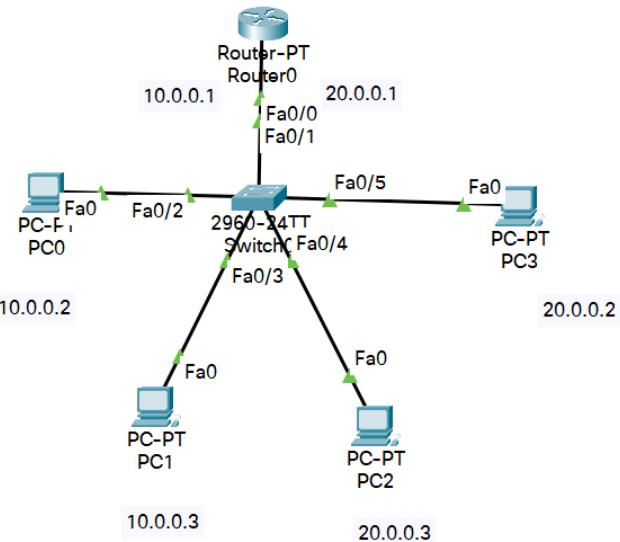
- + Successful Telnet connections show that PCO can establish a remote connection to PCT/Server through the router

MAC ADDRESS - ADDRESSES - MASK

ROUTING - FORWARDING THE NETWORK REQUESTS

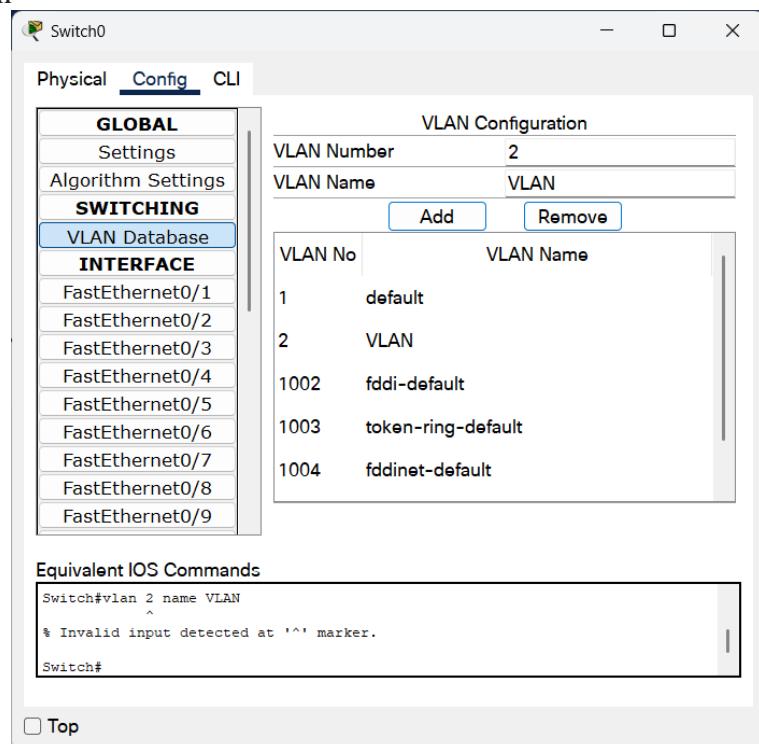
## Program 11

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

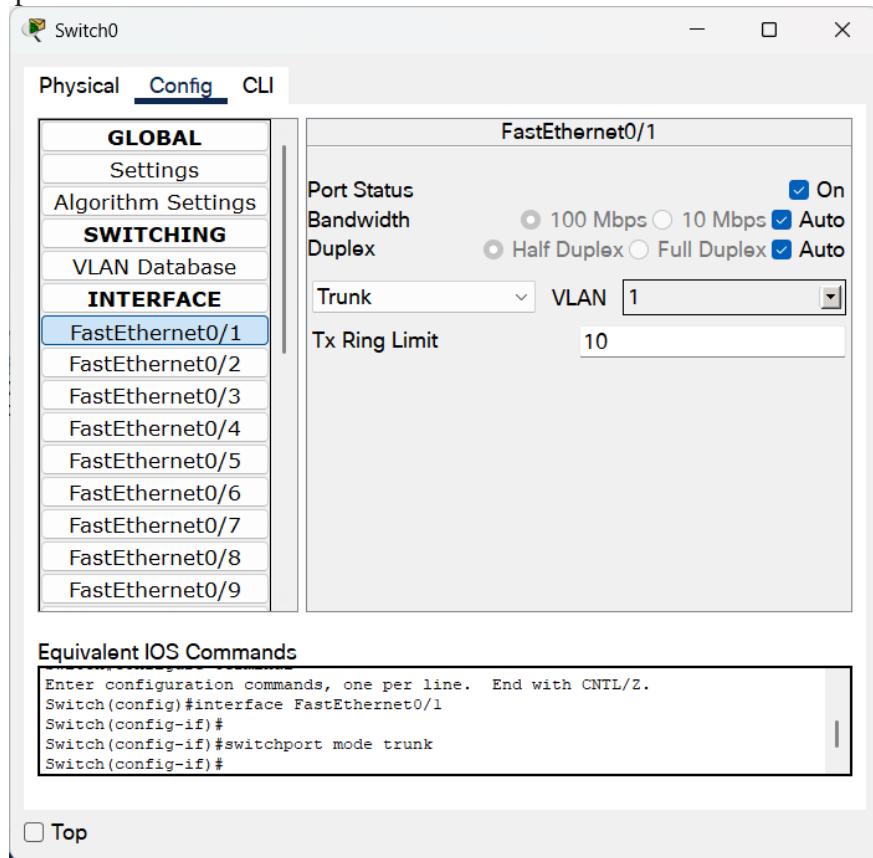


- iii. Screen shots/ output

Switch Configuration



## Configuration of port connected to router



## Configuration of ports connected to other networks

This screenshot shows two side-by-side configuration windows for "Switch0".

**Left Window (FastEthernet0/4 Configuration):**

- Port Status:** On (checked)
- Bandwidth:** 100 Mbps (radio button selected)
- Duplex:** Auto (checkbox checked)
- Access:** Trunk (dropdown selected)
- VLAN:** 2 (dropdown selected)
- Tx Ring Limit:** 10

**Equivalent IOS Commands:**

```

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

```

**Right Window (FastEthernet0/5 Configuration):**

- Port Status:** On (checked)
- Bandwidth:** 100 Mbps (radio button selected)
- Duplex:** Auto (checkbox checked)
- Access:** VLAN (dropdown selected)
- VLAN:** 2 (dropdown selected)
- Tx Ring Limit:** 10

**Equivalent IOS Commands:**

```

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

```

Both windows have a "Top" button at the bottom left.

## 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

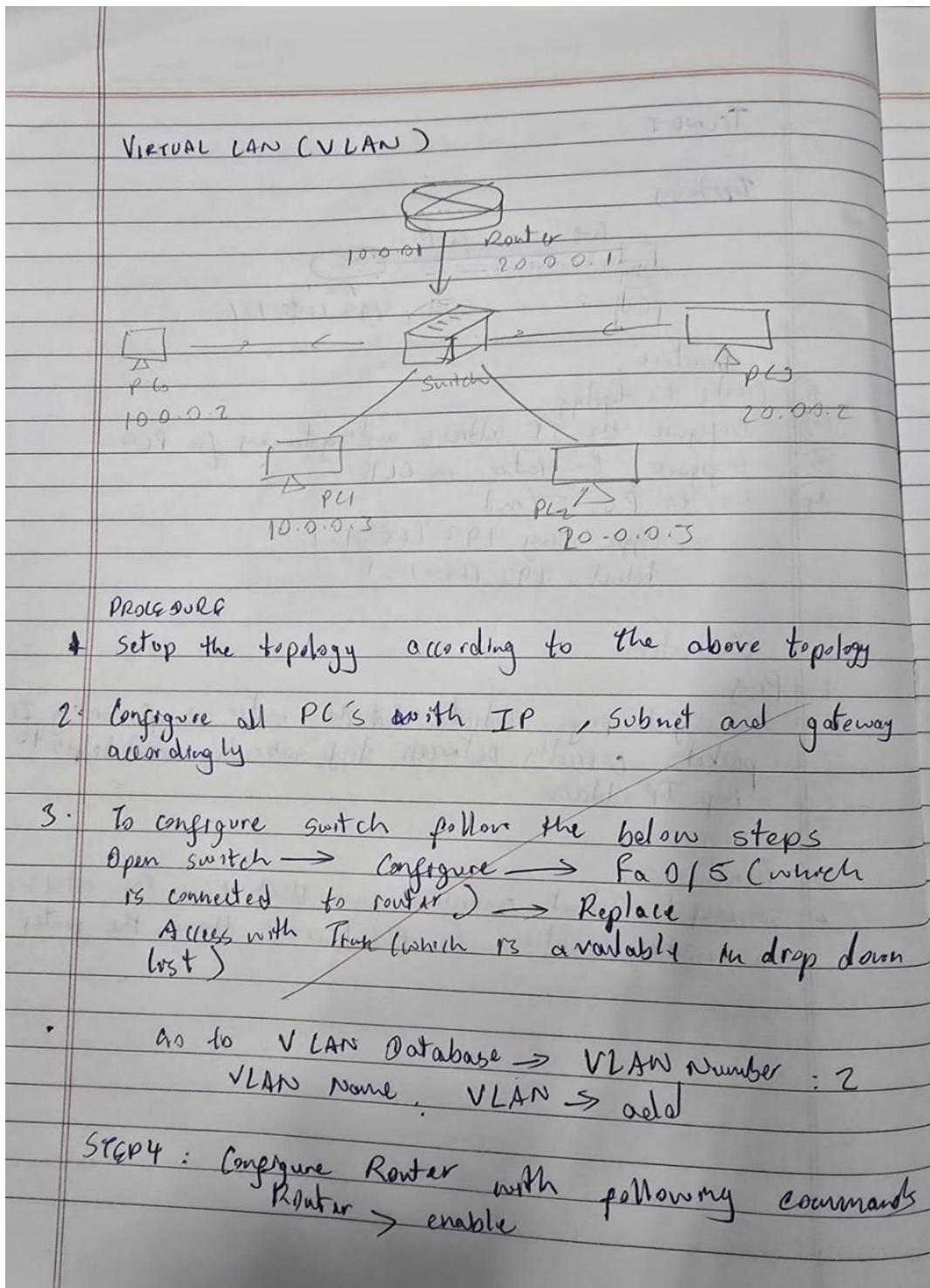
Router(config-subif)#encapsulation dot1q 2
Router(config-subif)#ip address 20.0.0.1 255.0.0.0
Router(config-subif)#no shutdown
Router(config-subif)#exit
Router(config)#

Ctrl+F6 to exit CLI focus
```

Top

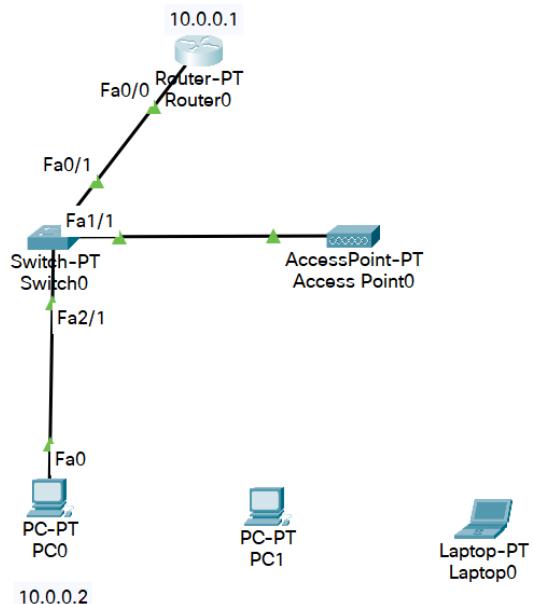
**Copy** **Paste**

iv. Observation

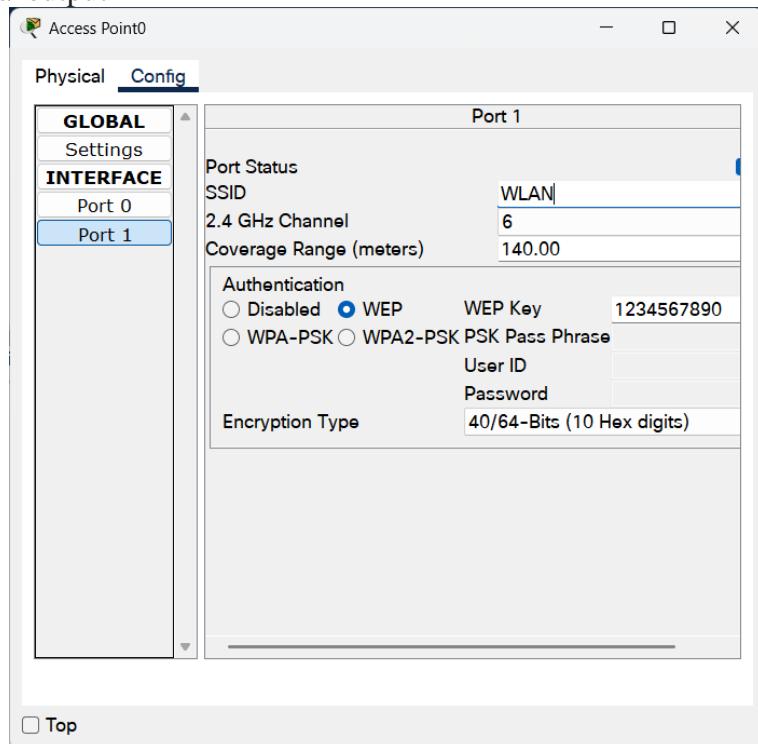


## Program 12

- 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

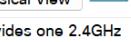
Physical Device View		
<a href="#">Zoom In</a>	<a href="#">Original Size</a>	<a href="#">Zoom Out</a>
		
 <span>Customize Icon in Physical View</span>		
 <span>Customize Icon in Logical View</span>		
		

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

Top

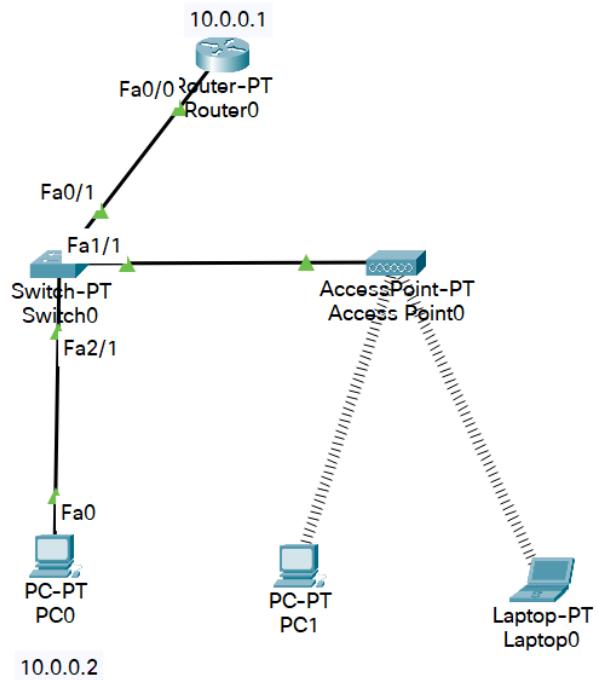
**Laptop0**

- Physical**
- Config
- Desktop
- Programming

Physical Device View		
<a href="#">Zoom In</a>	<a href="#">Original Size</a>	<a href="#">Zoom Out</a>
		
 <span>Customize Icon in Physical View</span>		
 <span>Customize Icon in Logical View</span>		
		

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

Top



Ping:

Screenshot of the Packet Tracer Command Prompt window titled "Laptop0". The window shows the following command and its output:

```

Physical Config Desktop Programming
Command Prompt X
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

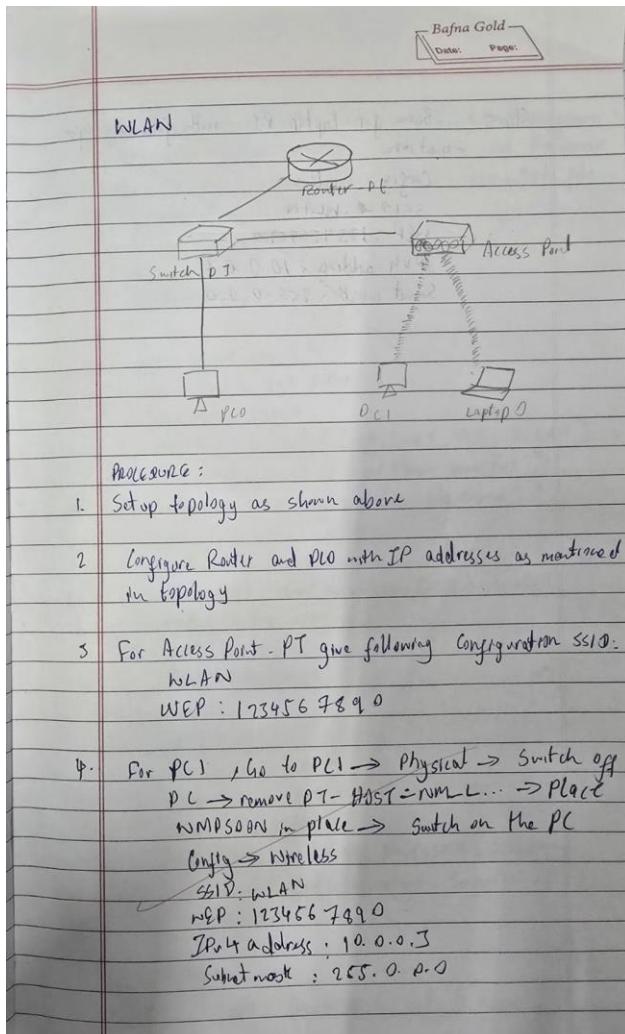
Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 24ms, Maximum = 40ms, Average = 28ms

C:\>

```

Top

#### iv. Observation



Steps : Same for Laptop PT with following parameter

config - wireless  
SSID : WLAN  
WEP : 1234567890  
IPv4 address : 10.0.0.4  
Subnet mask : 255.0.0.0

## 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

Write a program for Error Detection Using CRC-CCITT

(Code :)

```

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

def mod_2div(dividend, divisor):
    pick = len(divisor)
    temp = dividend[0:pick]
    while pick <= len(dividend):
        if temp[0] == '1':
            temp = XOR(divisor, temp) + dividend[pick]
        else:
            temp = XOR('0'*pick, temp)
        pick += 1
    if temp[0] == '1':
        temp = XOR(divisor, temp)
    else:
        temp = XOR('0', temp)
    return temp

def encode(data, key):
    appended_data = data + '0' * (len(key) - 1)
    remainder = mod_2div(appended_data, key)
    codeword = data + remainder
    return codeword

def verify_data(received_data, key):
    remainder = mod_2div(received_data, key)
    if remainder == '0':
        print("Data is error-free")
    else:
        print("Data has errors")

```

Buffer: 11  
Data:

return 0;  $\tan(\text{key}) - 1 \rangle = \text{remainder}$

~~if - name - == "main"~~

~~data = input("Enter the binary: ")~~

~~if rs valid~~  
~~print ("Correct")~~  
~~else~~  
~~print ("Corrupted")~~

Output :

Enter the binary message: 1101011011  
 received binary 1101011011110110110

~~86~~ ~~86~~  
Leaky

## **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

Write a program for leaky bucket Algorithm

```
import time
import random
from collections import deque

class LeakyBucket:
    def __init__(self, bucket_capacity, output_rate):
        self.bucket_capacity = bucket_capacity
        self.output_rate = output_rate
        self.bucket = deque()

    def add_packet(self, packet_size):
        if len(self.bucket) < self.bucket_capacity:
            self.bucket.append(packet_size)
            print(f"Packet of size {packet_size} added. Current bucket size: {len(self.bucket)}")
        else:
            print(f"Packet of size {packet_size} is dropped. Current bucket size: {len(self.bucket)}")

    def leak(self):
        packet_sent = 0
        while packet_sent < self.output_rate and self.bucket:
            packet_size = self.bucket.pop()
            print(f"Packet of size {packet_size} is sent")
            packet_sent += 1
```

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

```
packet_sent = 1
if packets_sent == 0:
    print("Bucket is empty. No packets to serve")

if __name__ == "__main__":
    bucket_capacity = int(input("Enter bucket capacity (number of packets):"))
    output_rate = int(input("Enter output rate (packets per time unit):"))
    leaky_bucket = LeakyBucket(bucket_capacity, output_rate)

for _ in range(10):
    packet_size = random.randint(1, 5)
    print(f"Leaky packets: {leaky_packet_size}")
    leaky_bucket.leak()
    time.sleep(1)

Output:
Incoming packet size=2
Packet size 2 added. current bucket size
leaking packets
Packet of size 2 sent

Incoming packet size=1
Packet of size @ added. current bucket size
leaking packets
Packet of size 1 sent.
```

### **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:

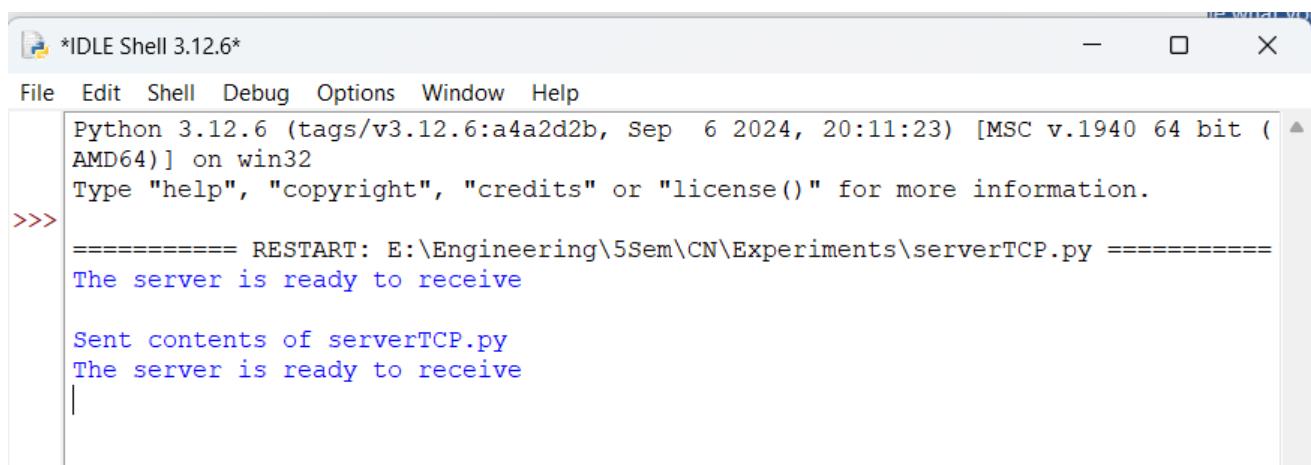
from socket import *
serverName="127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print ("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()

    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

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

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

Code:

```
Client TCP.py
from socket import *
Server Name = "192.0.0.1"
Server Port = 12000
clientSocket = socket (AF_INET, SOCK_STREAM)
clientSocket.connect ((Server Name, Server Port))
sentence = input ("\\n Enter file Name")
clientSocket.send (sentence.encode ())
file contents = clientSocket.recv (1024).decode ()
print ("\\n from Server: " + sentence)
print (file contents)
clientSocket.close ()
```

Server TCP.py

```
from socket import *
serverName = "192.0.0.1"
serverPort = 12000
serverSocket = socket (AF_INET, SOCK_STREAM)
serverSocket.bind ((serverName, serverPort))
serverSocket.listen ()
```

while 1:

print ("The server is ready to receive")

connectionSocket = serverSocket.accept ()  
 sentence = connectionSocket.recv (1024).decode ()  
 file = open (sentence, "r")  
 t = file.read (1024)  
 connectionSocket.send (t.encode ())  
 print ("\\n Sent contents of " + sentence)  
 file.close ()  
 connectionSocket.close ()

## 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(sentence.encode("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)

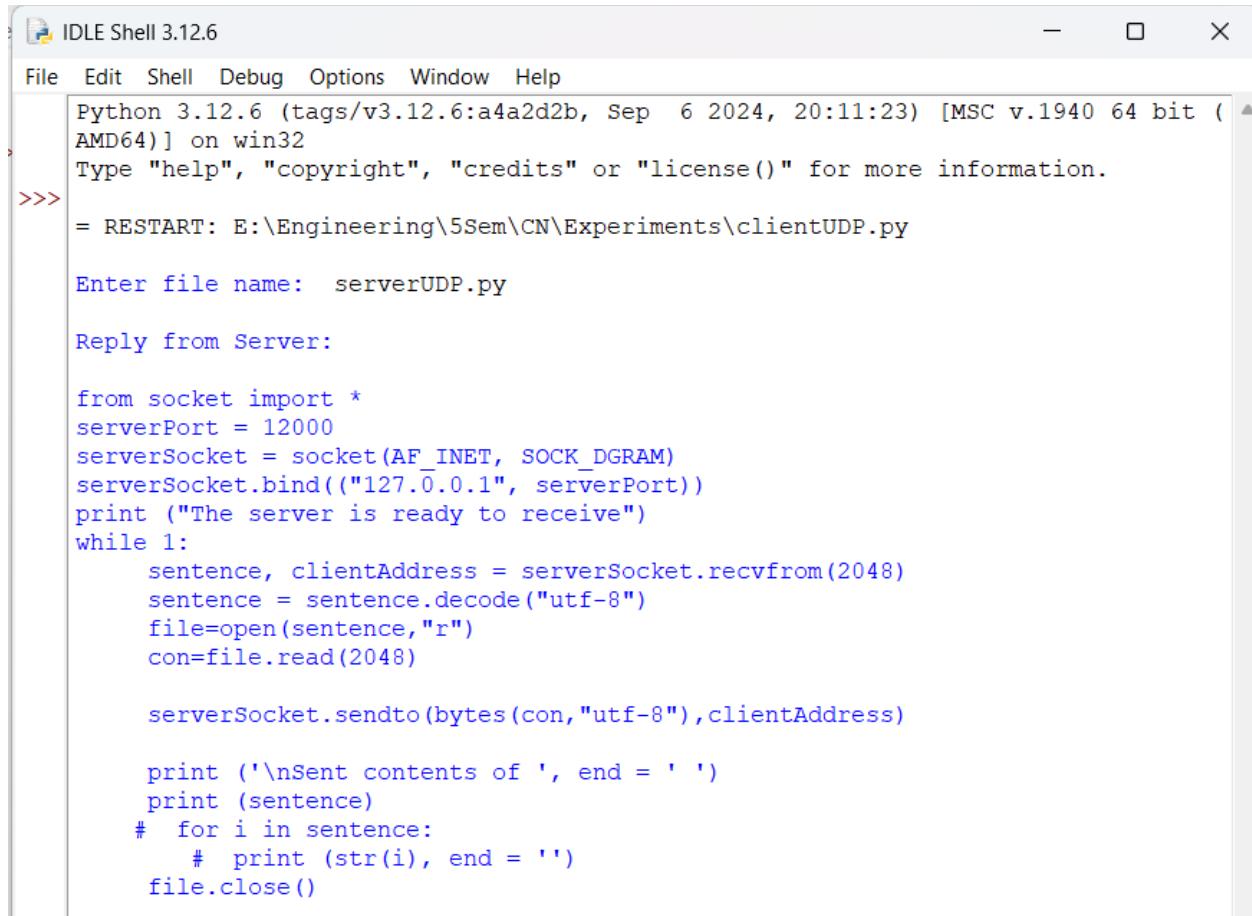
    serverSocket.sendto(bytes(con,"utf-8"),clientAddress)

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

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

### iii. Screen shots/ output

#### Client



```
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\clientUDP.py

Enter file name: serverUDP.py

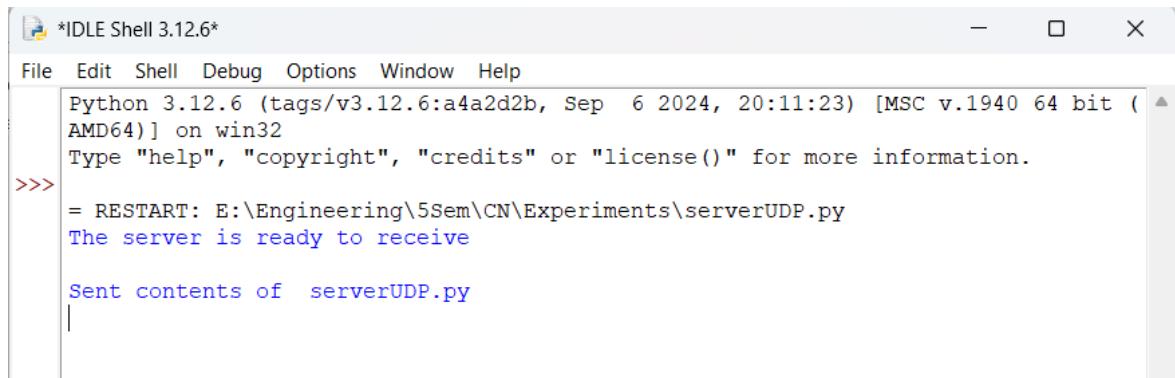
Reply from Server:

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



```
*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\serverUDP.py
The server is ready to receive

Sent contents of  serverUDP.py
```

iv. Observation

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

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

Code:

Client UDP.py

```
from socket import *
```

Server Name = "127.0.0.1"

Server Port = 12000

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

Sentence = input("In file file name: ")

clientSocket.sendto(Sentence, ("utf-8")) (ServerName, ServerPort)

fileContents, ServerAddress = clientSocket.recvfrom(2048)

print("In Reply from Server: ")

print(fileContents.decode("utf-8"))

# for i in file contents :

print(f'{i}.end = ')

clientSocket.close()

clientSocket.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(1024)  
sentence = sentence.decode("utf-8")  
file = open(sentence + ".r")  
con = file.read(2048)

serverSocket.sendto(bytes(con, "utf-8"), ClientAddress)  
print("Sent contents of", sentence)  
print(sentence)

for i in sentence:  
 print(str(i).end = 1)  
 file.close()