

# **VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

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



## **LAB RECORD**

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

*Submitted by*

**Shreyas K(1BM22CS271)**

*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 **Shreyas K(1BM22CS271)**, 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

<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:

<https://github.com/Shreyesss7/CN-LAB>

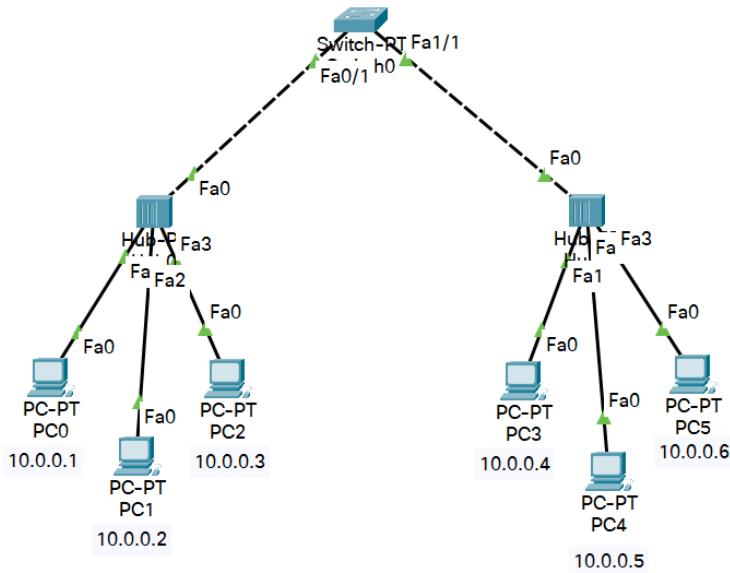
## **Index-Cycle-II**

<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

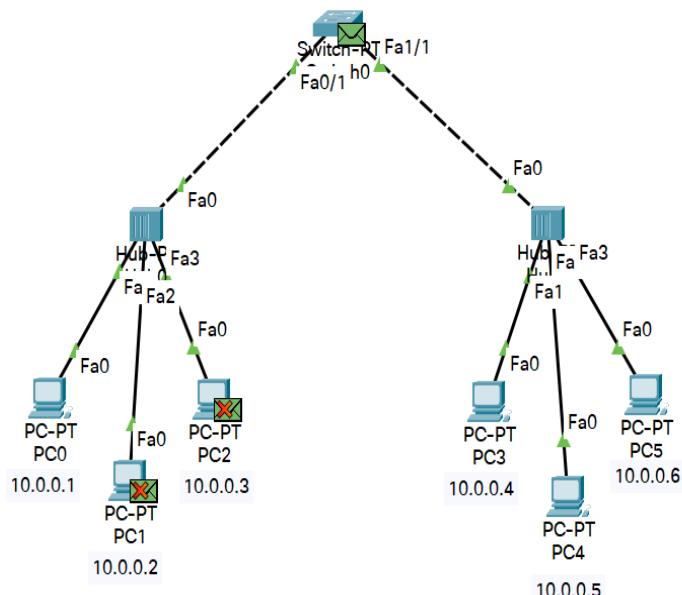
## Cycle-I

### **Program 1**

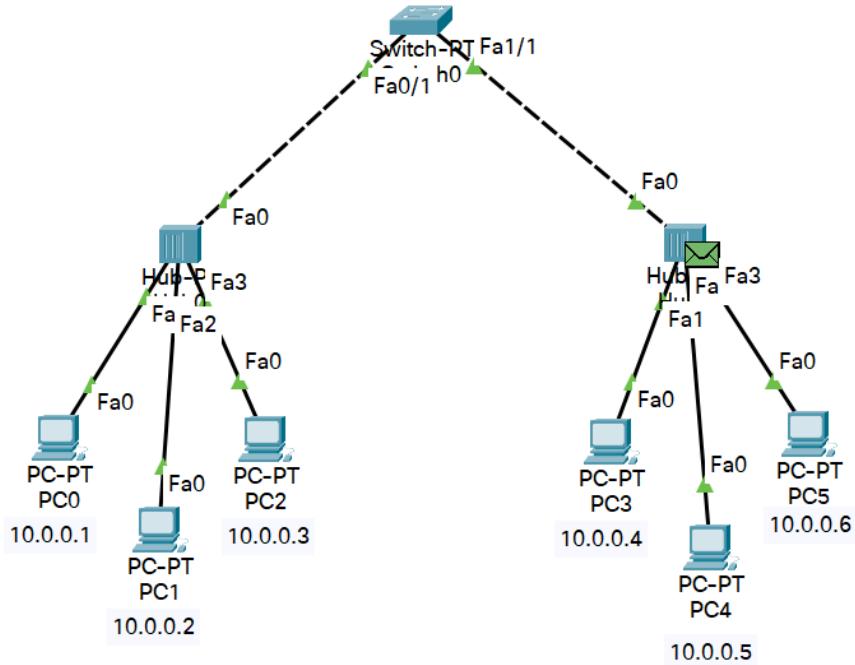
- i. Create a topology involving multiple hubs and a switch connecting them to simulate with simple PDU.
- ii. Procedure along with the topology



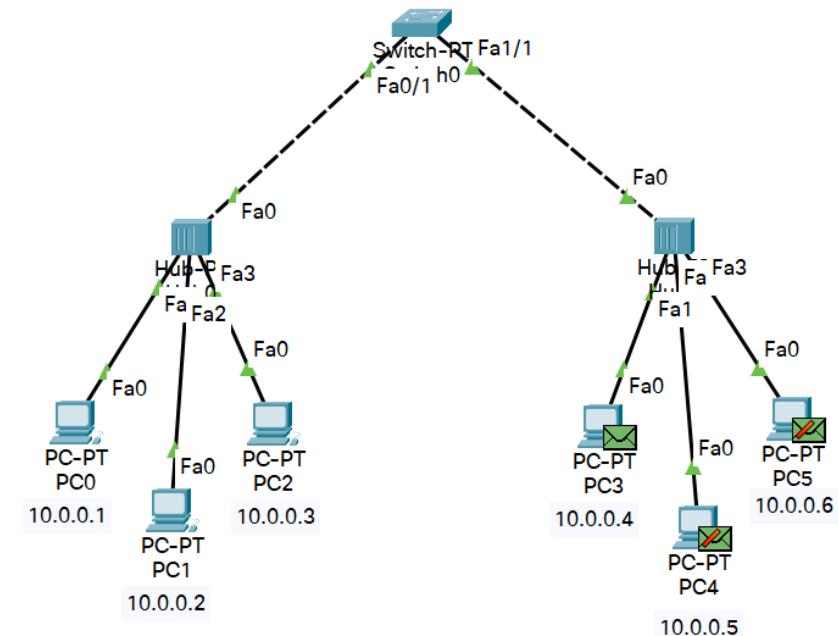
- iii. Screen shots/ output  
Hub behaviour at sending end



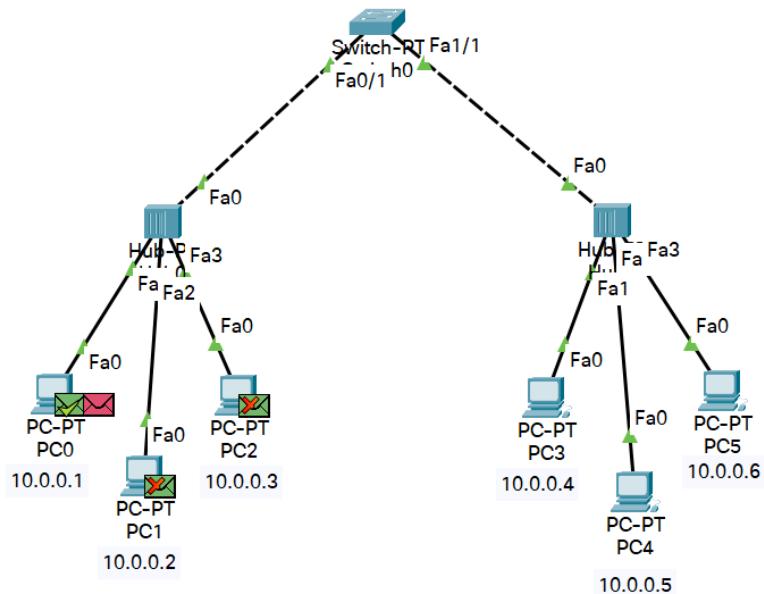
## Switch behaviour



## Hub behaviour at receiving end



## Hub behaviour when back to sender



## Ping command to connectivity

```

PC0
Physical Config Desktop Programming
Command Prompt X
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

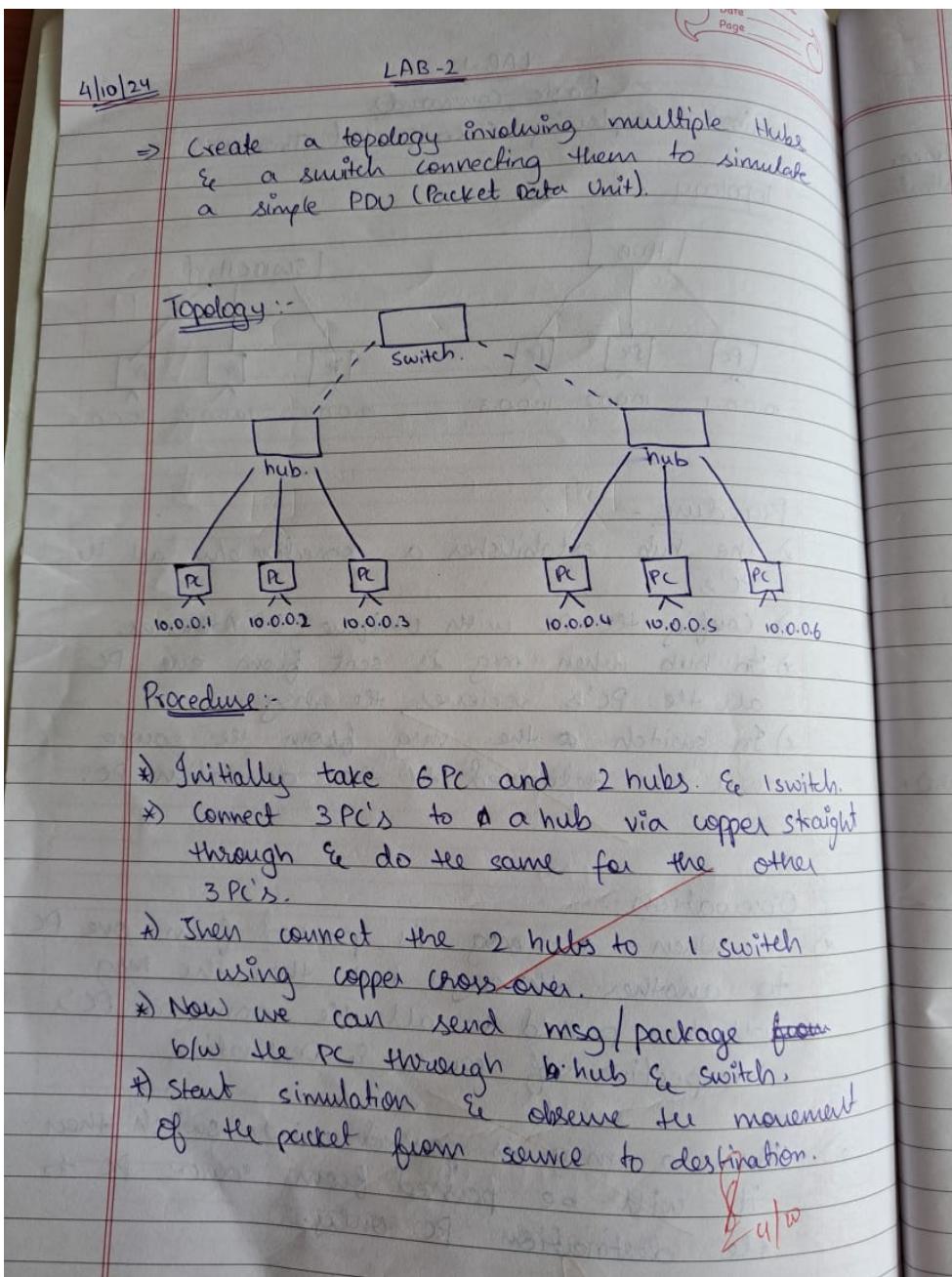
Ping statistics for 10.0.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>

```

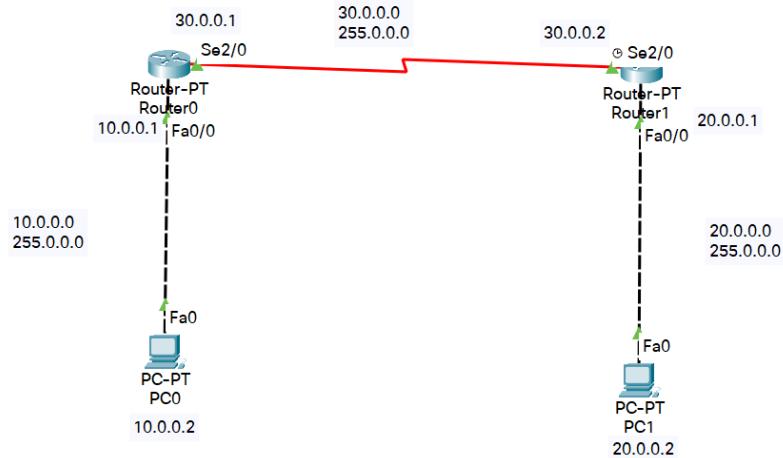
Top

#### iv. Observation



## **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

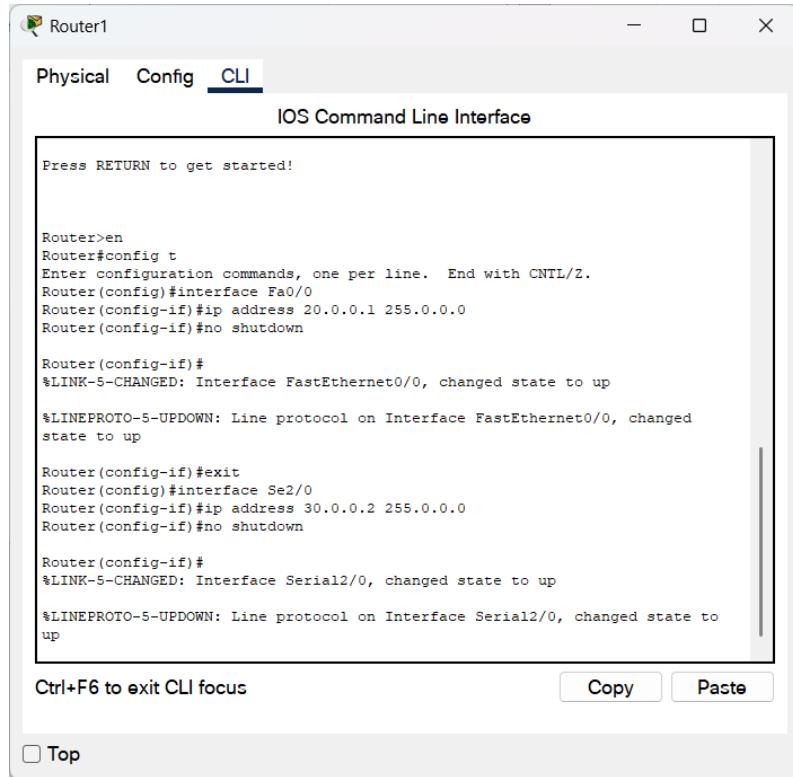
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed
state to up

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
```

## Router1 configuration



The screenshot shows a software application window titled "Router1". At the top, there are three tabs: "Physical", "Config", and "CLI", with "CLI" being the active tab. Below the tabs, it says "IOS Command Line Interface". A text area contains the following configuration commands:

```
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 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". To the right of the text area are two buttons: "Copy" and "Paste". Below the text area 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**

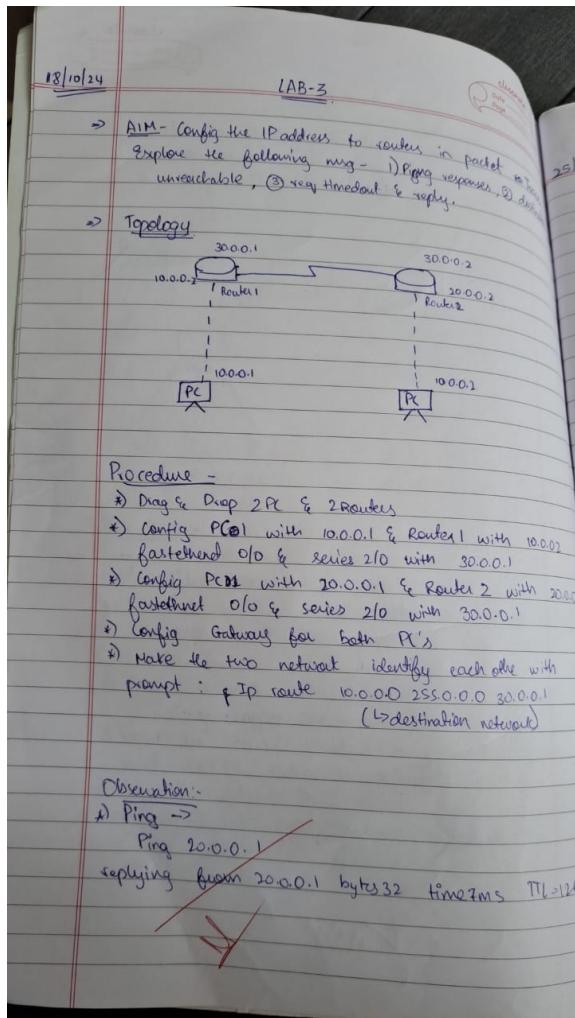
```
Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.2

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

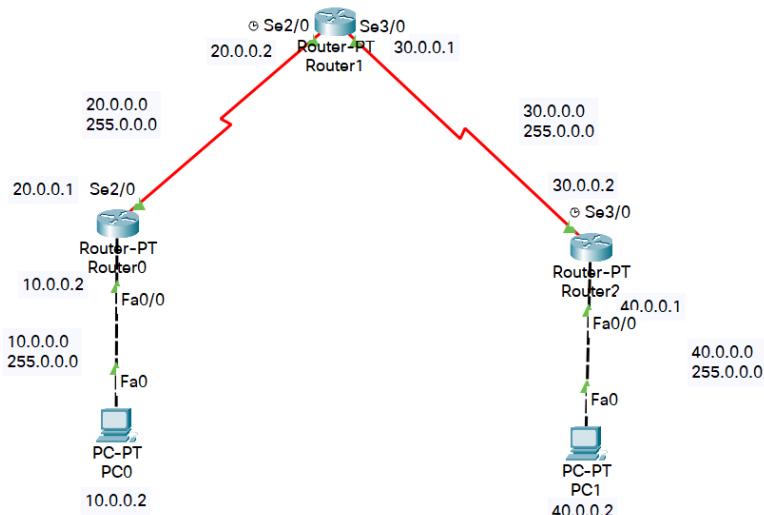
Ping statistics for 20.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 18ms, Average = 5ms
```

#### iv. Observation



## 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!

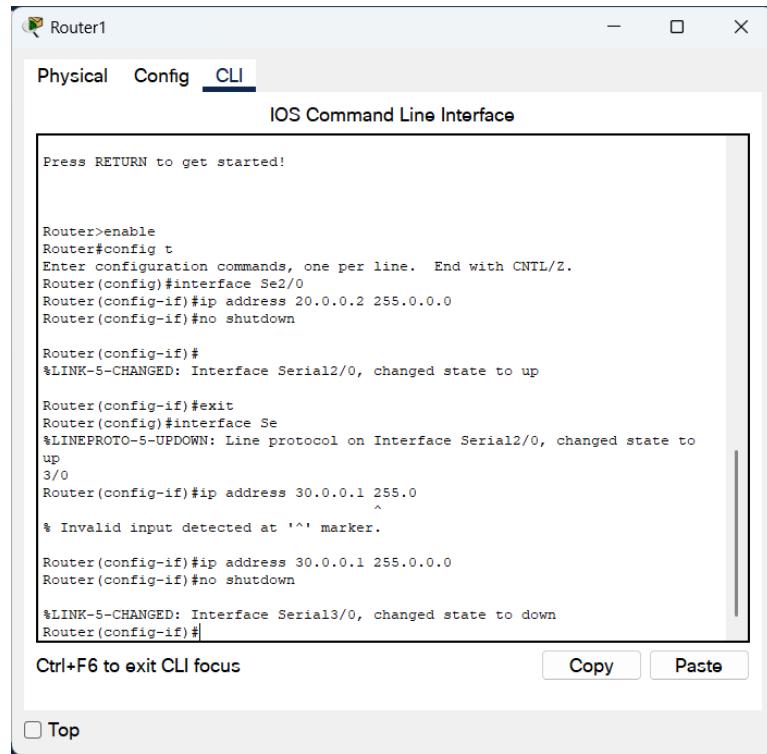
Router>en
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface Fa0/0
Router(config-if)#ip address 10.0.0.2 255.0.0.0
Router(config-if)#no shutdown

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-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)#
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. Below the tabs is a title bar "IOS Command Line Interface". The main area is a scrollable text box containing the configuration commands for Router1. The text box starts with "Press RETURN to get started!" followed by the configuration steps for Router1's interfaces.

```
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

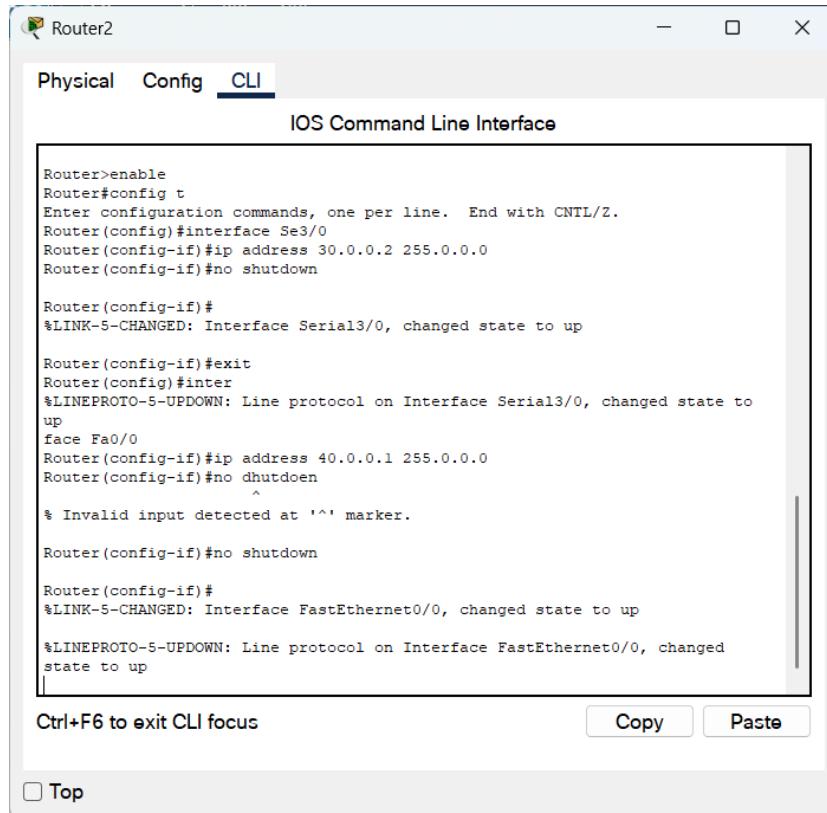
Router(config-if)#exit
Router(config)#interface Se
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to
up
3/0
Router(config-if)#ip address 30.0.0.1 255.0.0.0
^
% Invalid input detected at '^' marker.

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
```

## 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. Below the tabs is a title bar "IOS Command Line Interface". The main area is a scrollable text box containing the configuration commands for Router2's interfaces. The text box starts with "Press RETURN to get started!" followed by the configuration steps for Router2's interfaces.

```
Press RETURN to get started!

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

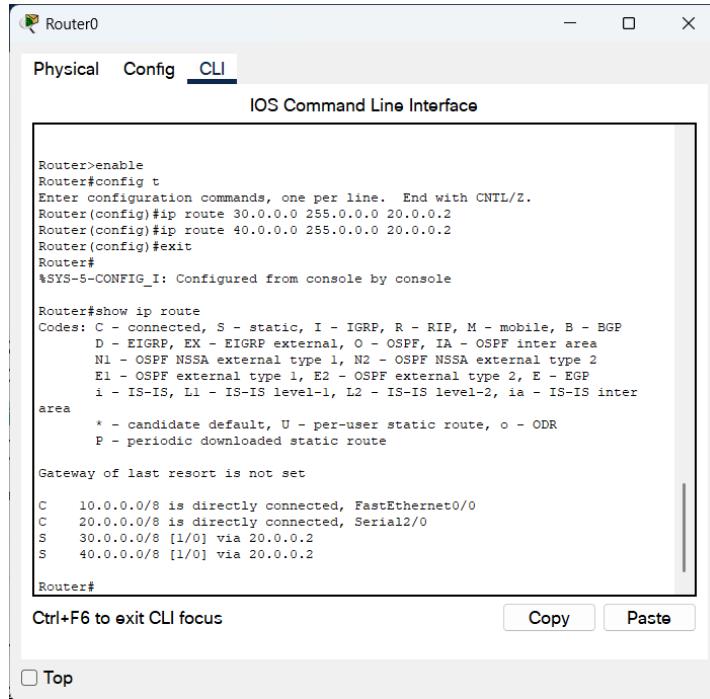
Router(config-if)# no shutdown

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
```

## 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

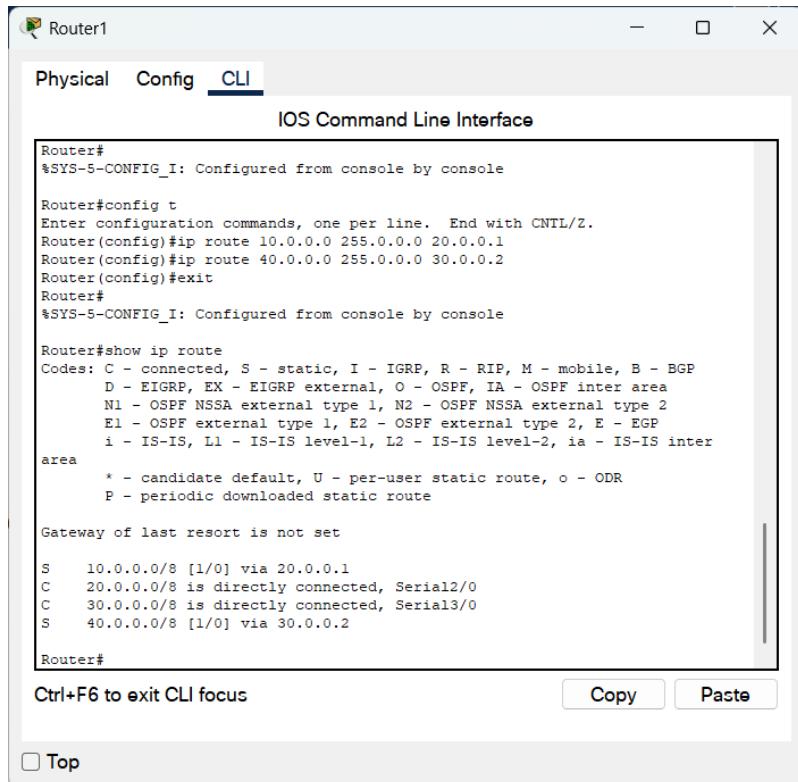
C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    20.0.0.0/8 is directly connected, Serial2/0
S    30.0.0.0/8 [1/0] via 20.0.0.2
S    40.0.0.0/8 [1/0] via 20.0.0.2

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#
```

## Dynamic Routing:

### Route0

```
Router>enable
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

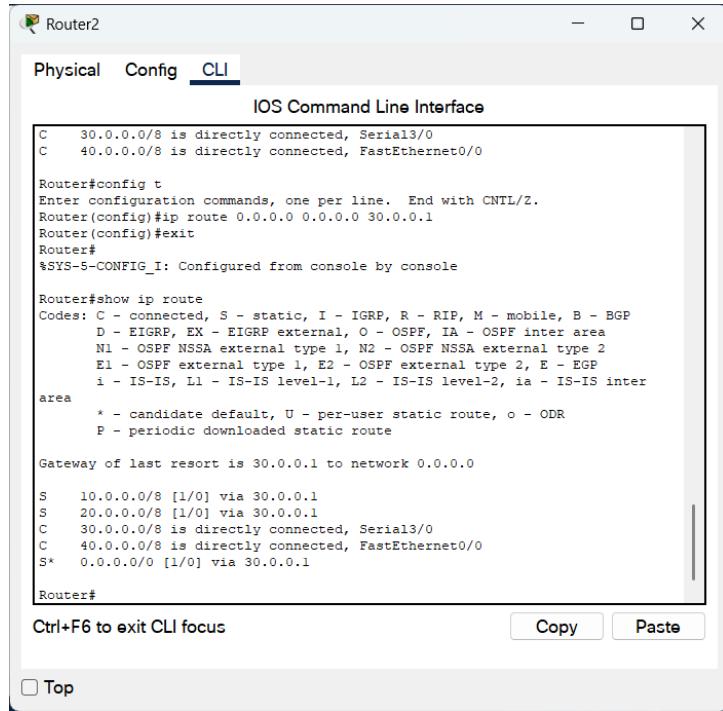
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      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#
```

## 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 output:

```
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. Below the window, there is a "Top" button.

Pinging:

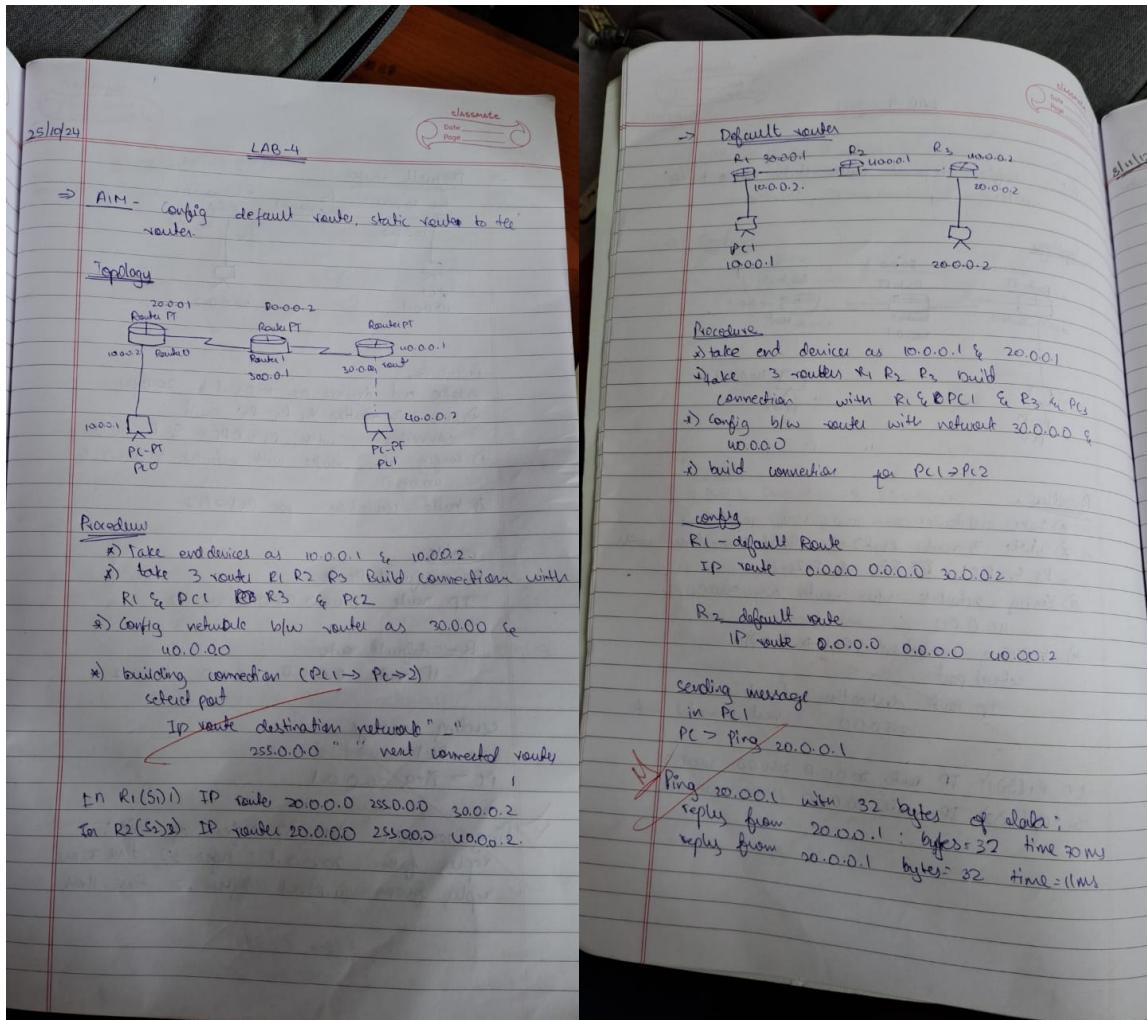
```
C:\>ping 40.0.0.2

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

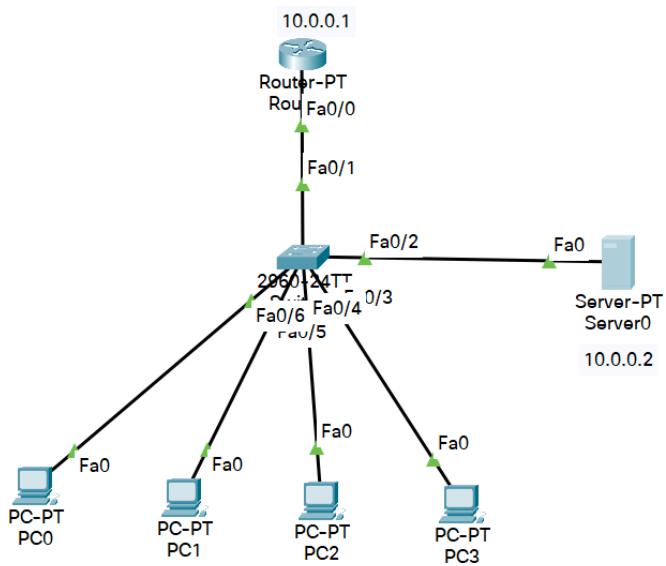
Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 25ms, Average = 16ms
```

#### iv. Observation



## Program 4

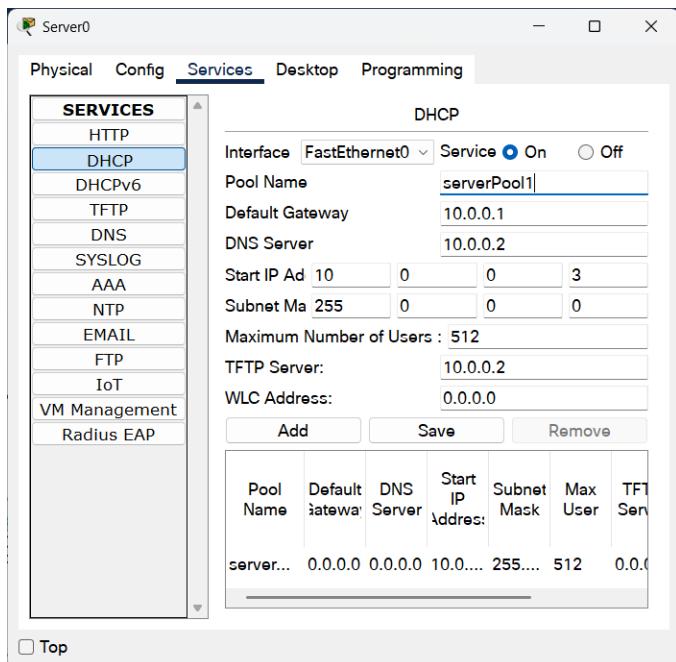
- i. Configure DHCP within a LAN and outside LAN.
- ii. Procedure along with the topology



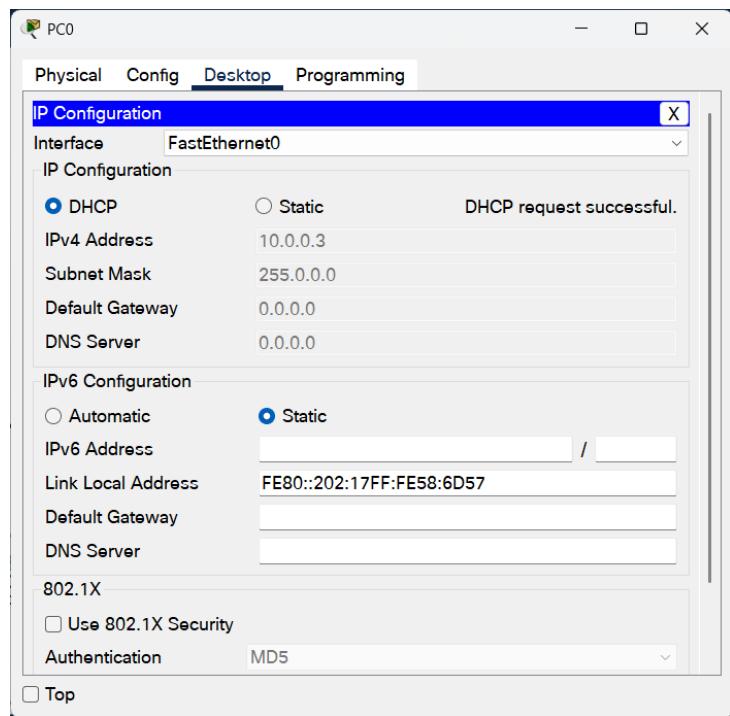
- iii. Screen shots/ output

DHCP Within LAN

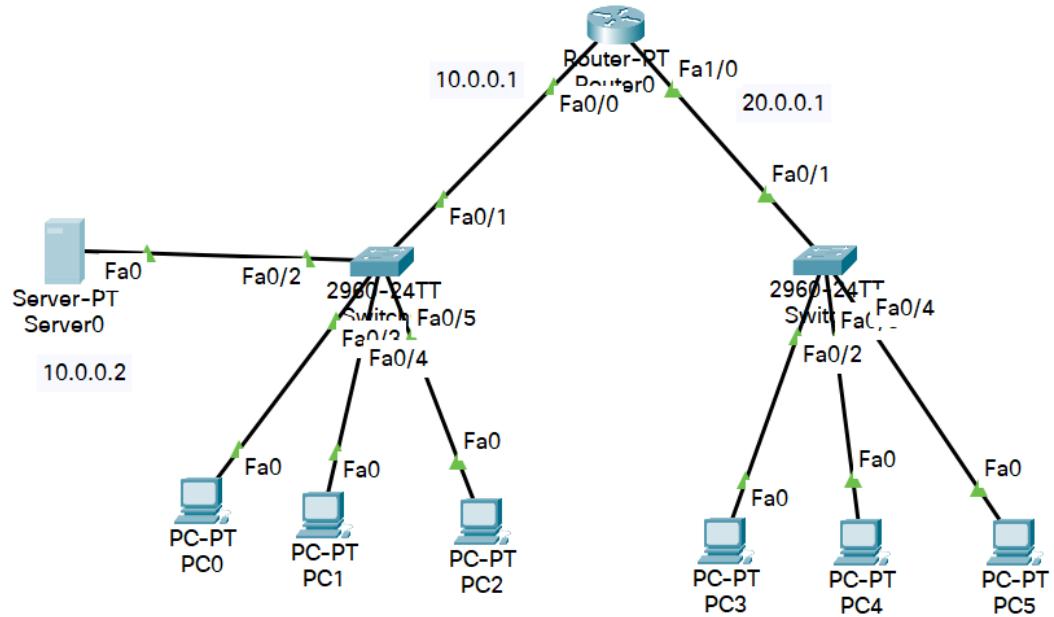
DHCP Configuration



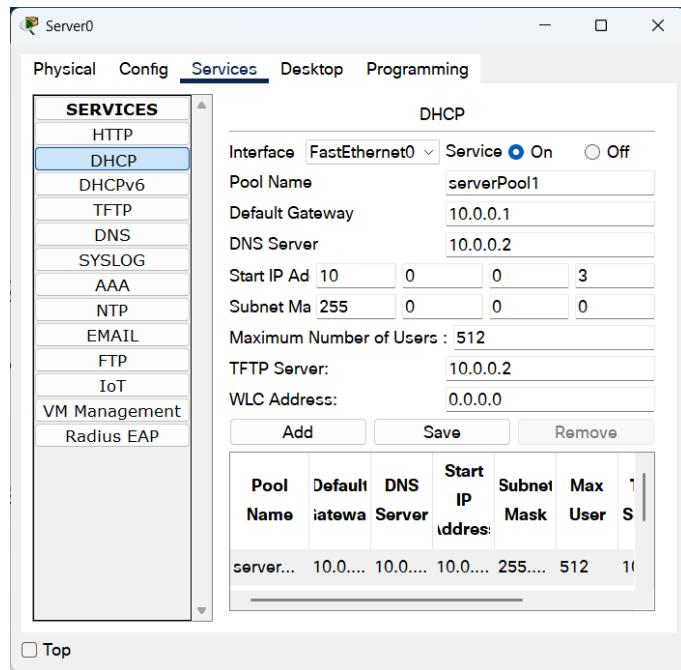
## PC settings



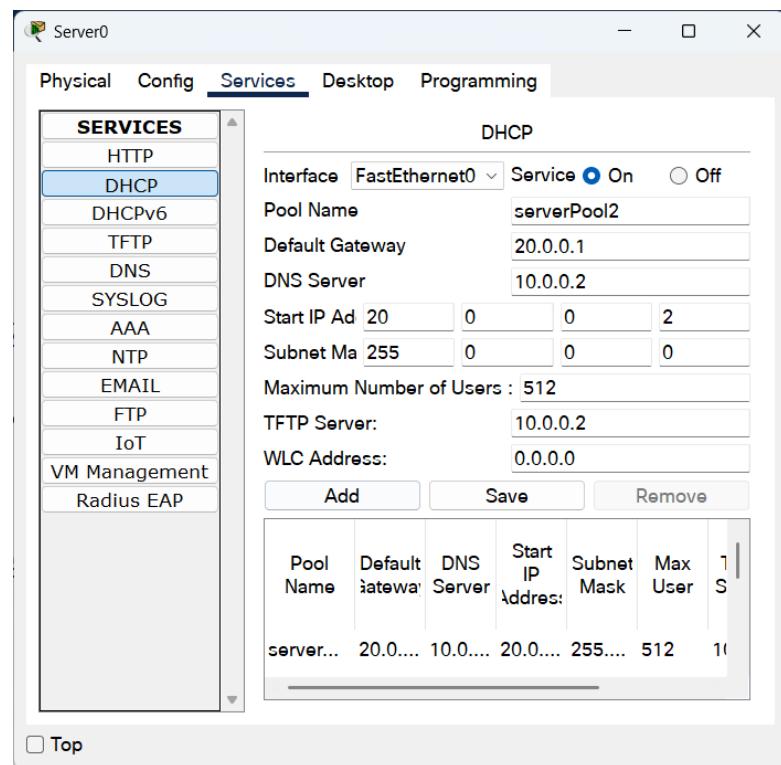
DHCP outside LAN:



DHCP configuration for inside LAN



DHCP configuration for outside LAN

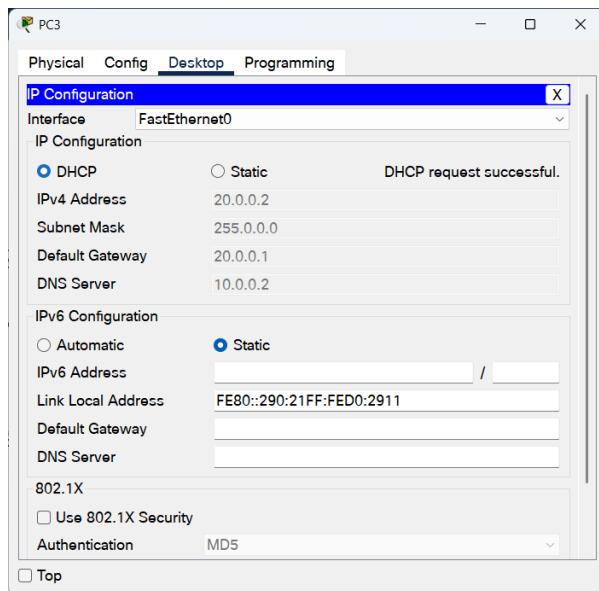


Ip helper command in Router

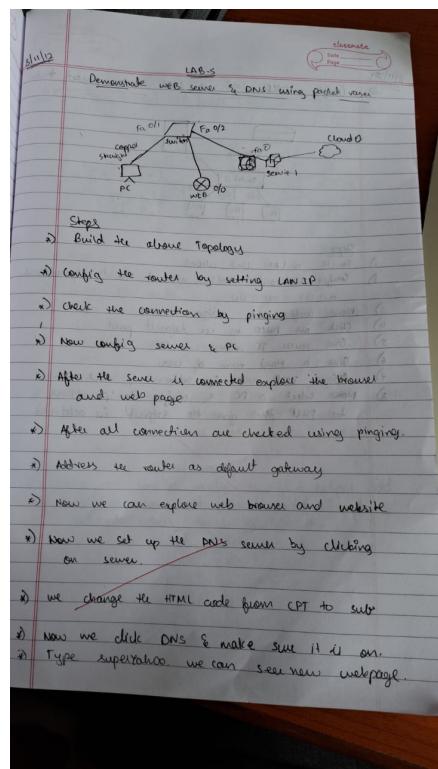
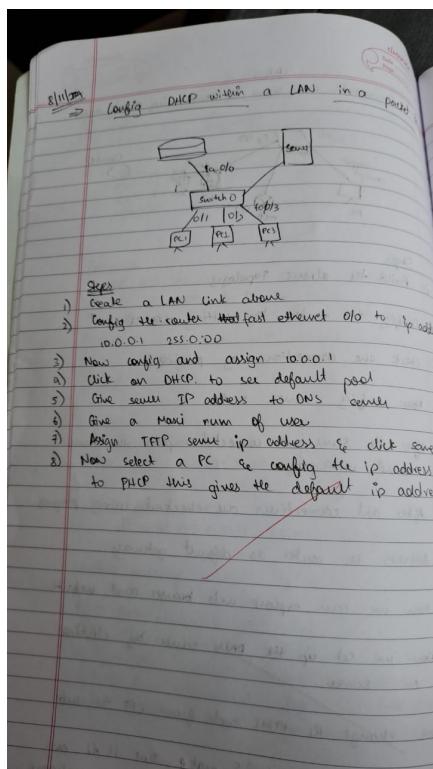
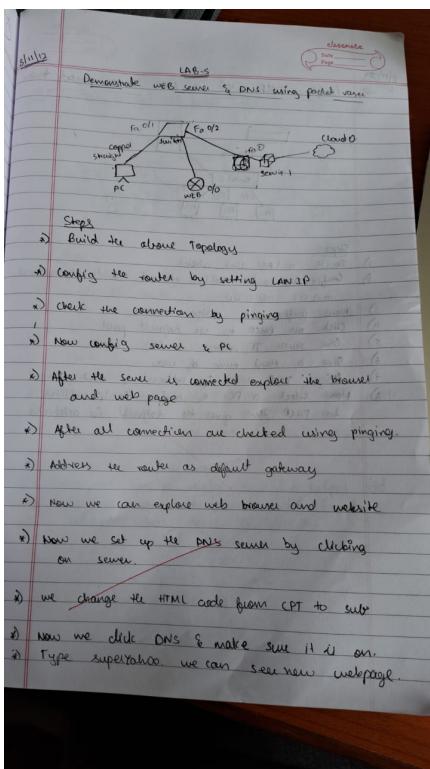
```
Router(config-if)#exit
Router(config)#interface Fa1/0
Router(config-if)#ip helper-address 10.0.0.2
Router(config-if)#

```

PC setting in another network



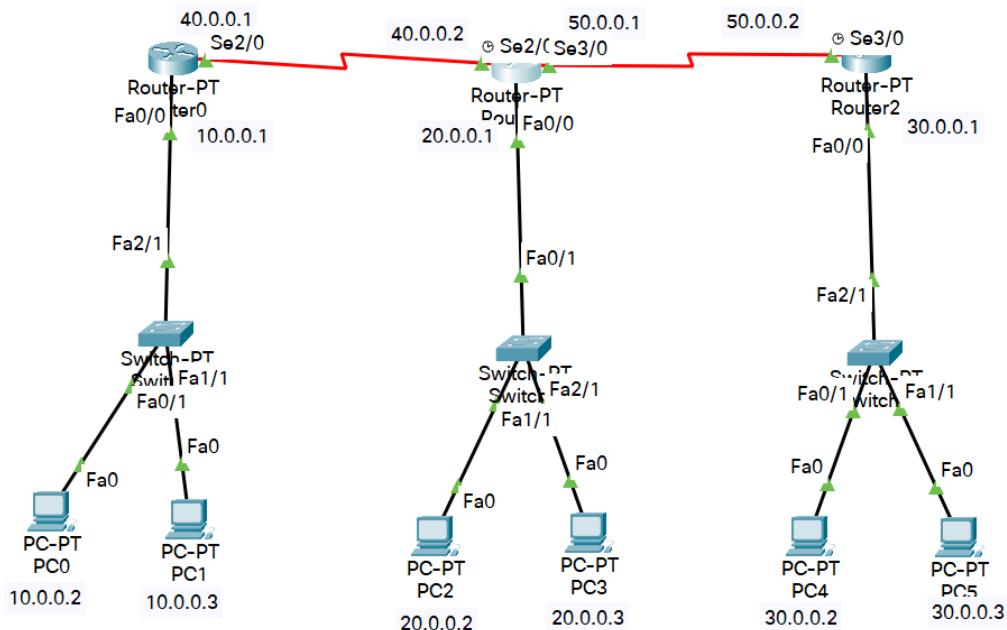
iv. Observation



## 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:

```

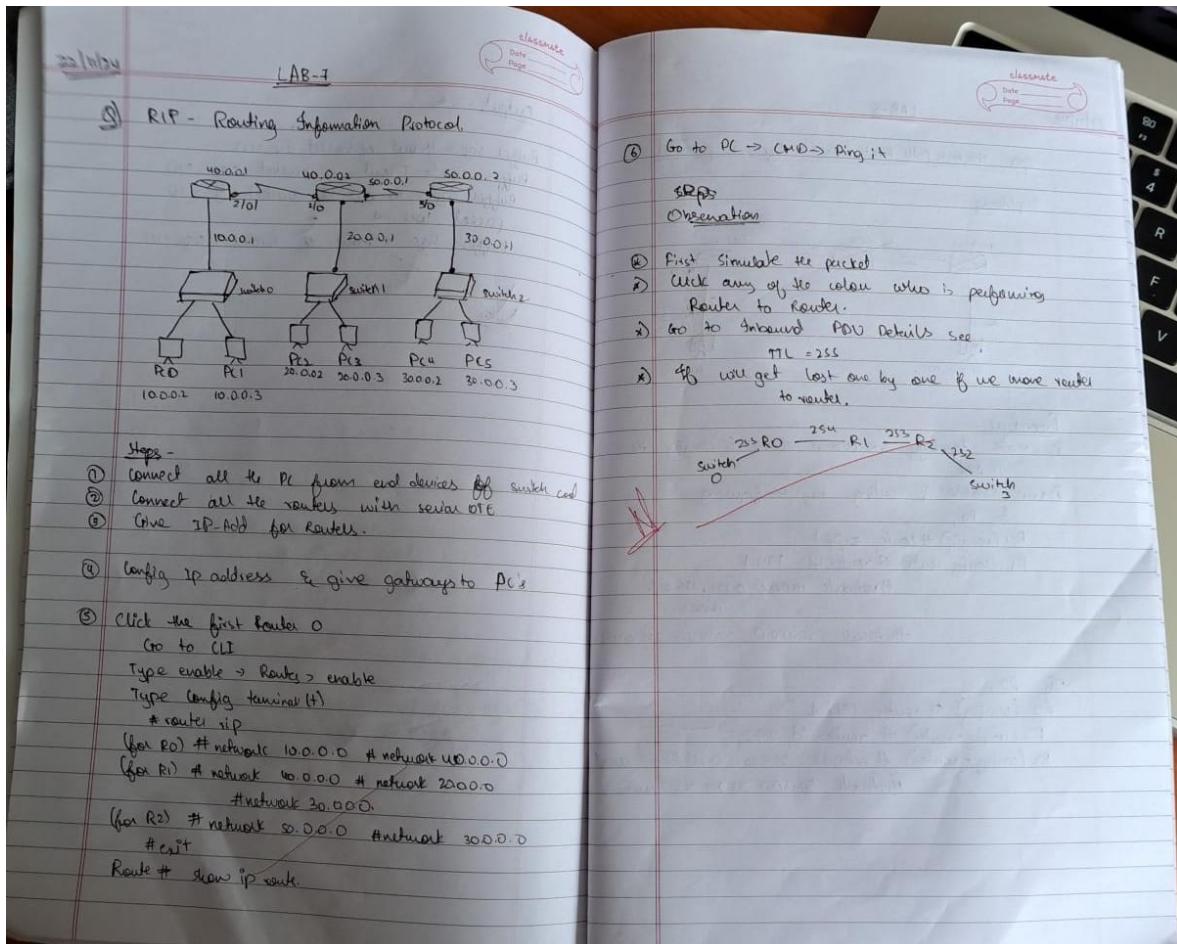
Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.2

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

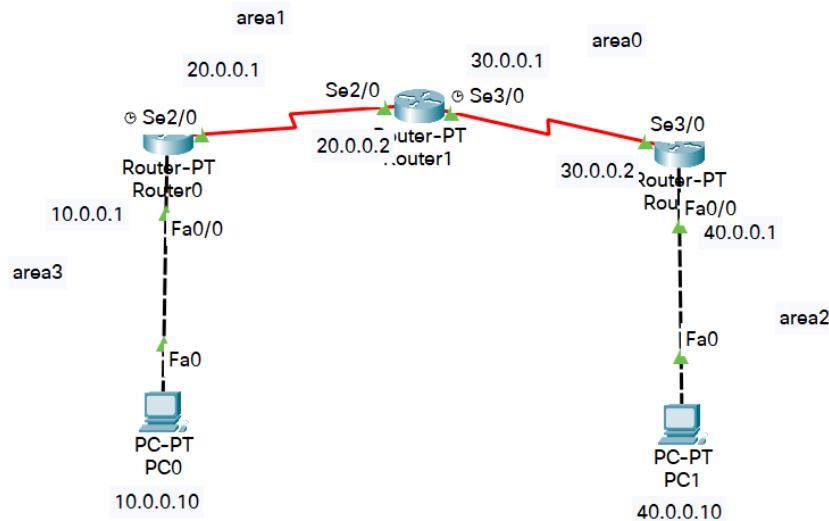
Ping statistics for 20.0.0.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 9ms, Average = 6ms
  
```

## 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 dhutdown
^
% 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

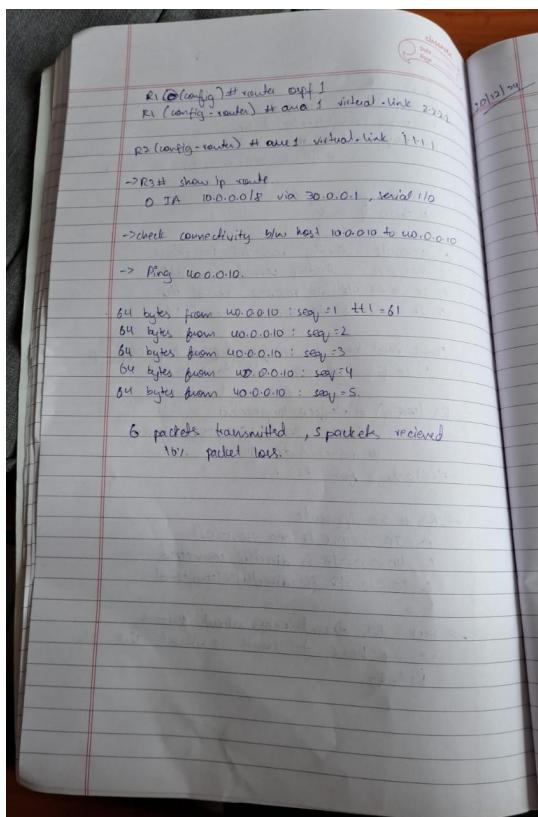
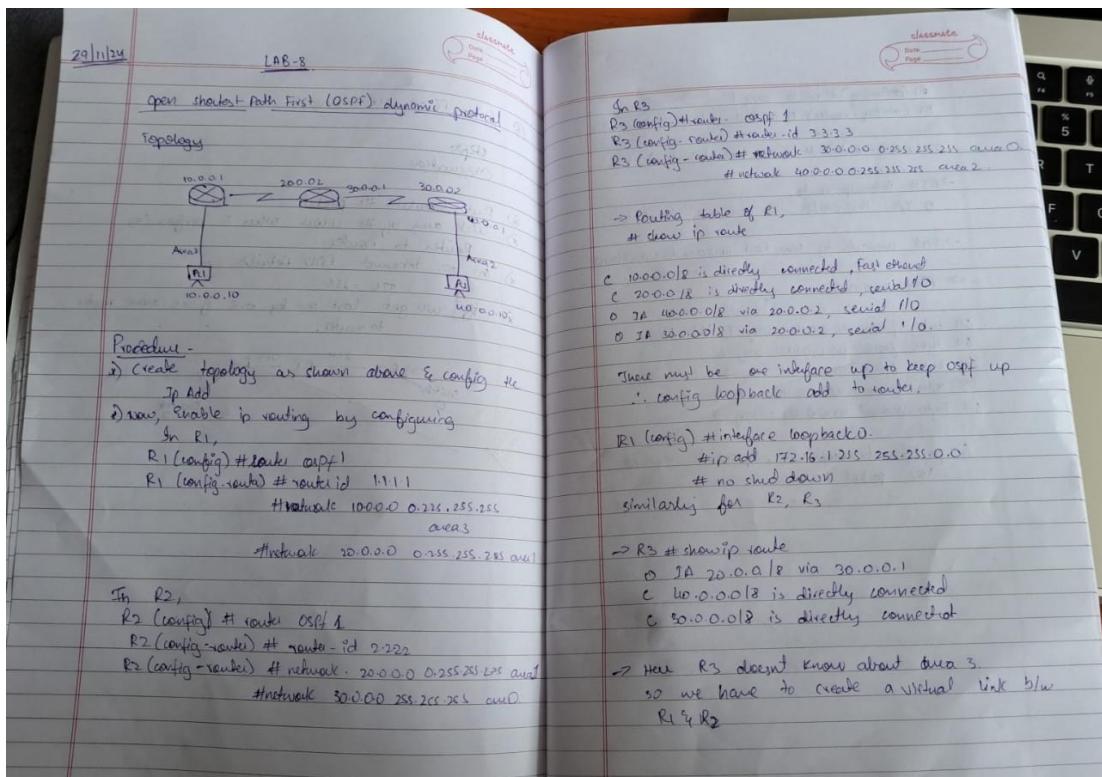
```
C:\>ping 40.0.0.10

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

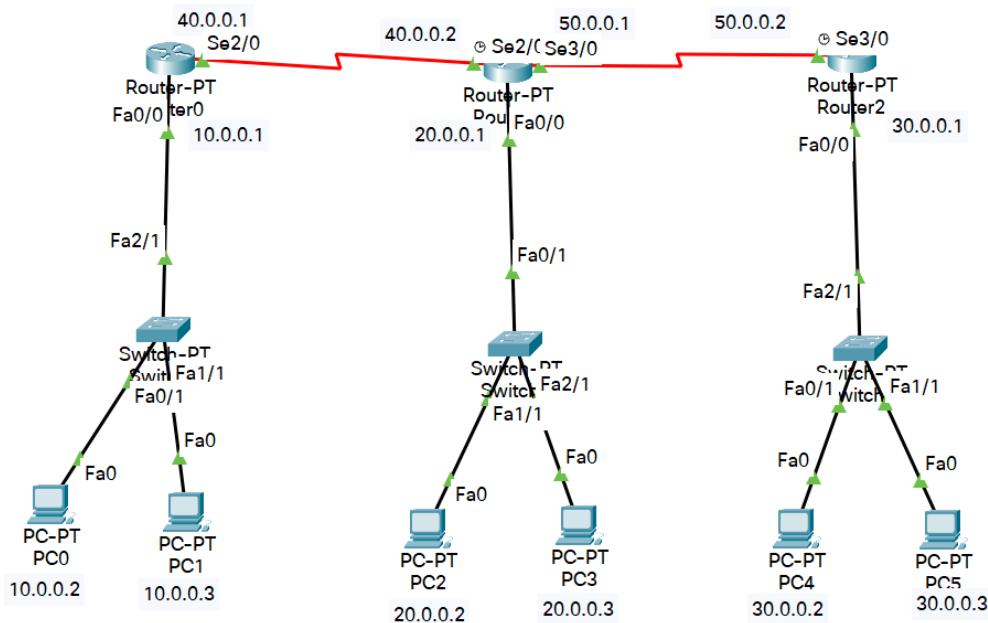
Ping statistics for 40.0.0.10:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 18ms, Maximum = 24ms, Average = 20ms
```

#### iv. Observation



## Program 7

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



- Screen shots/ output

### Packet at Router0

PDU Information at Device: Router0

OSI Model Inbound PDU Details Outbound PDU Details

PDU Formats

IP			
VER:4	IHL:5	DSCP:0x00	TL:28
ID:0x0005		FLAGS: 0x0	FRAG OFFSET:0x000
TTL:255	PRO:0x01	CHKSUM	
SRC IP:10.0.0.2			
DST IP:30.0.0.2			
DATA (VARIABLE LENGTH)			

ICMP

ICMP		
TYPE:0x08	CODE:0x00	CHECKSUM
ID:0x0003		SEQ NUMBER:5

PDU Information at Device: Router0

OSI Model Inbound PDU Details Outbound PDU Details

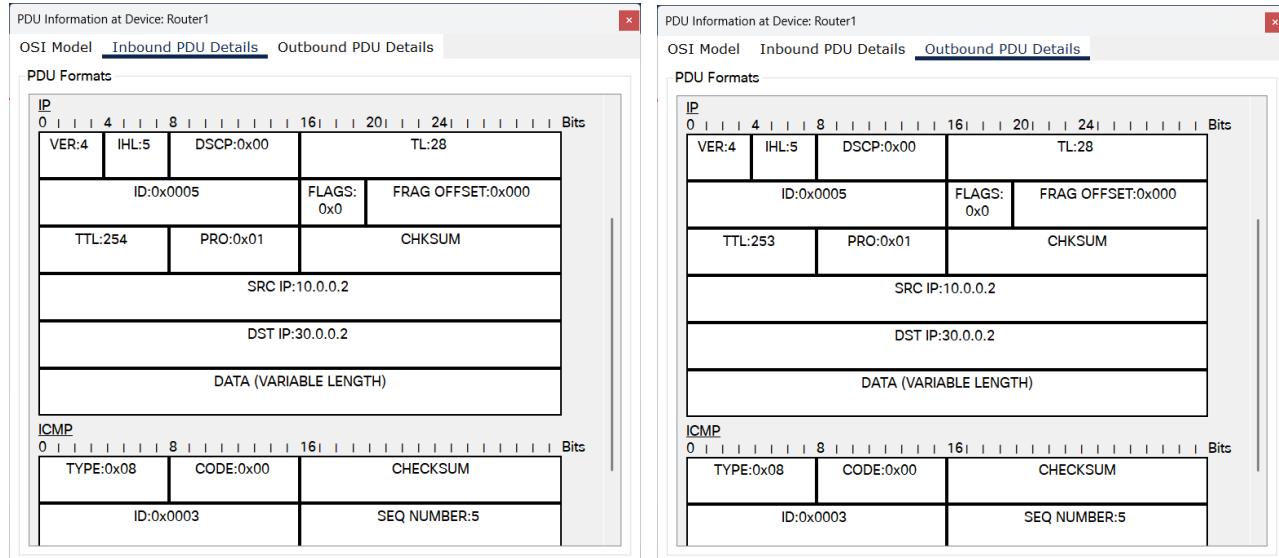
PDU Formats

IP			
VER:4	IHL:5	DSCP:0x00	TL:28
ID:0x0005		FLAGS: 0x0	FRAG OFFSET:0x000
TTL:254	PRO:0x01	CHKSUM	
SRC IP:10.0.0.2			
DST IP:30.0.0.2			
DATA (VARIABLE LENGTH)			

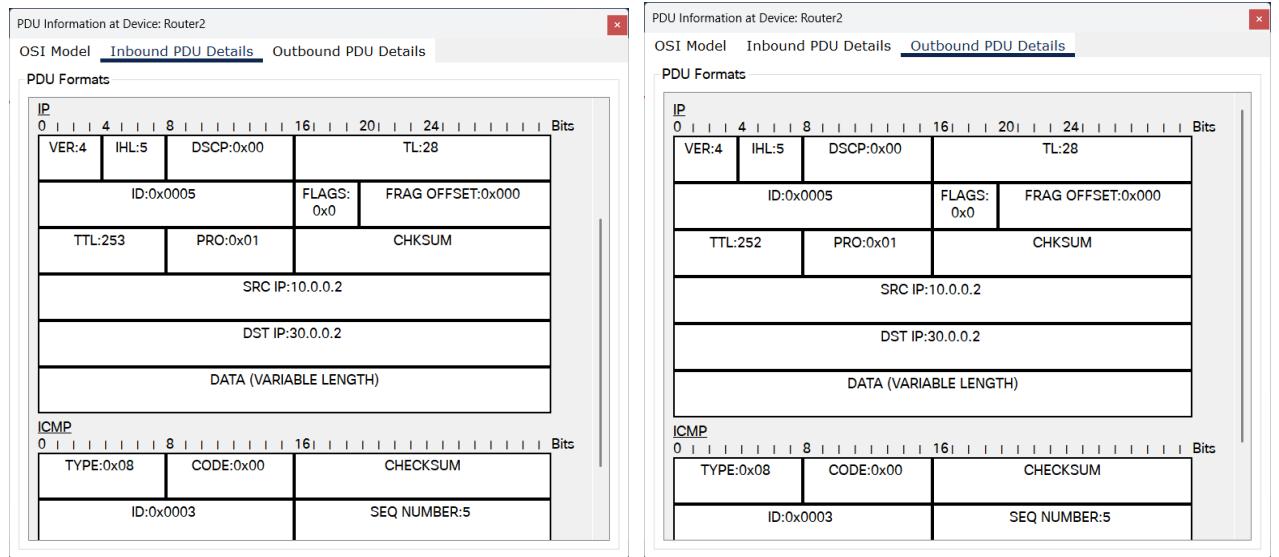
ICMP

ICMP		
TYPE:0x08	CODE:0x00	CHECKSUM
ID:0x0003		SEQ NUMBER:5

## Packet at Router1

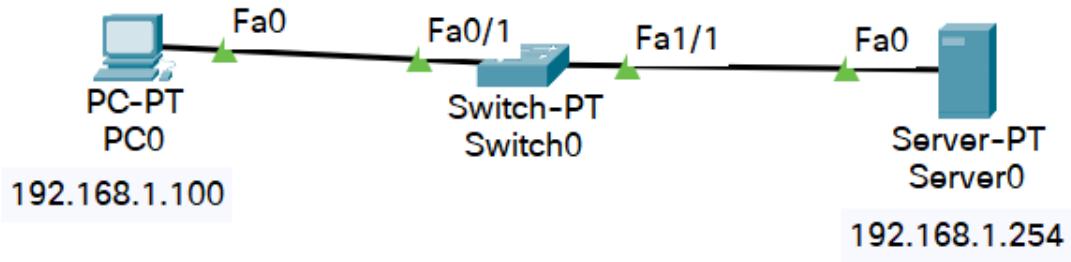


## Packet at Router2



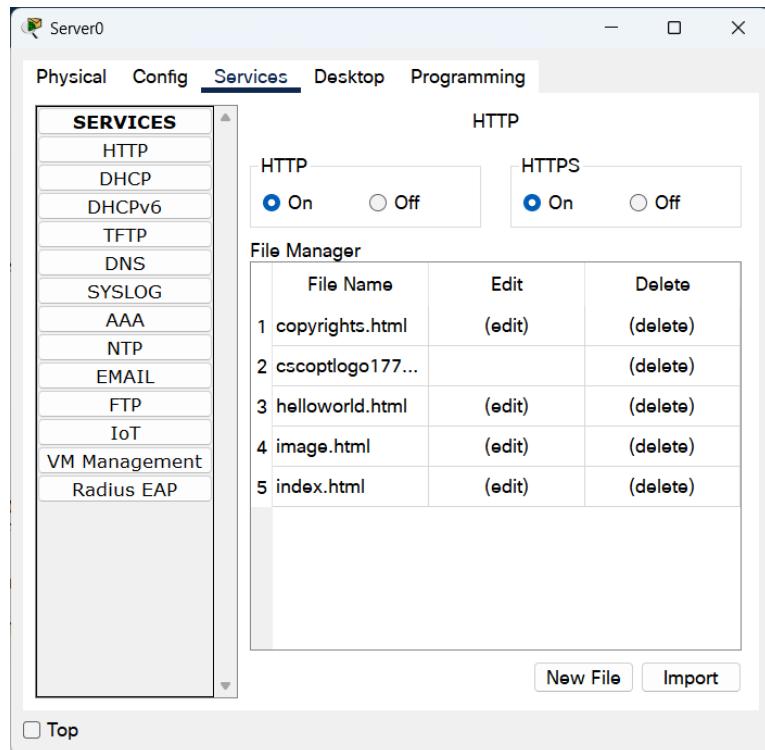
## Program 8

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

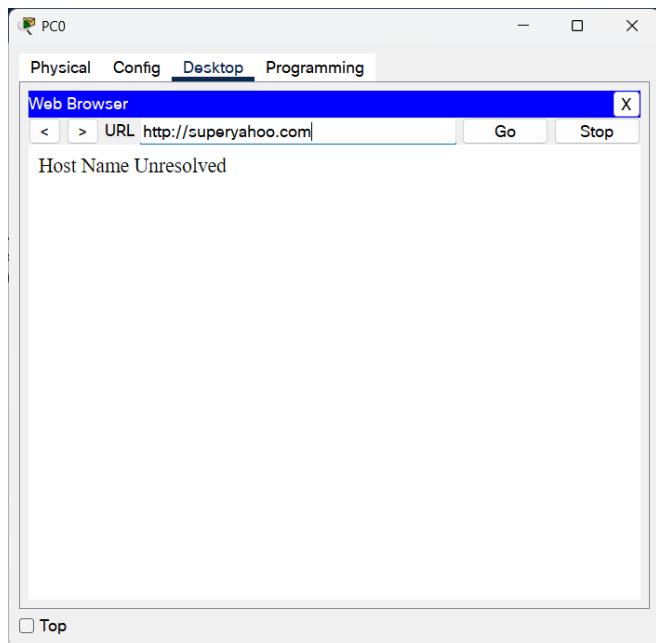
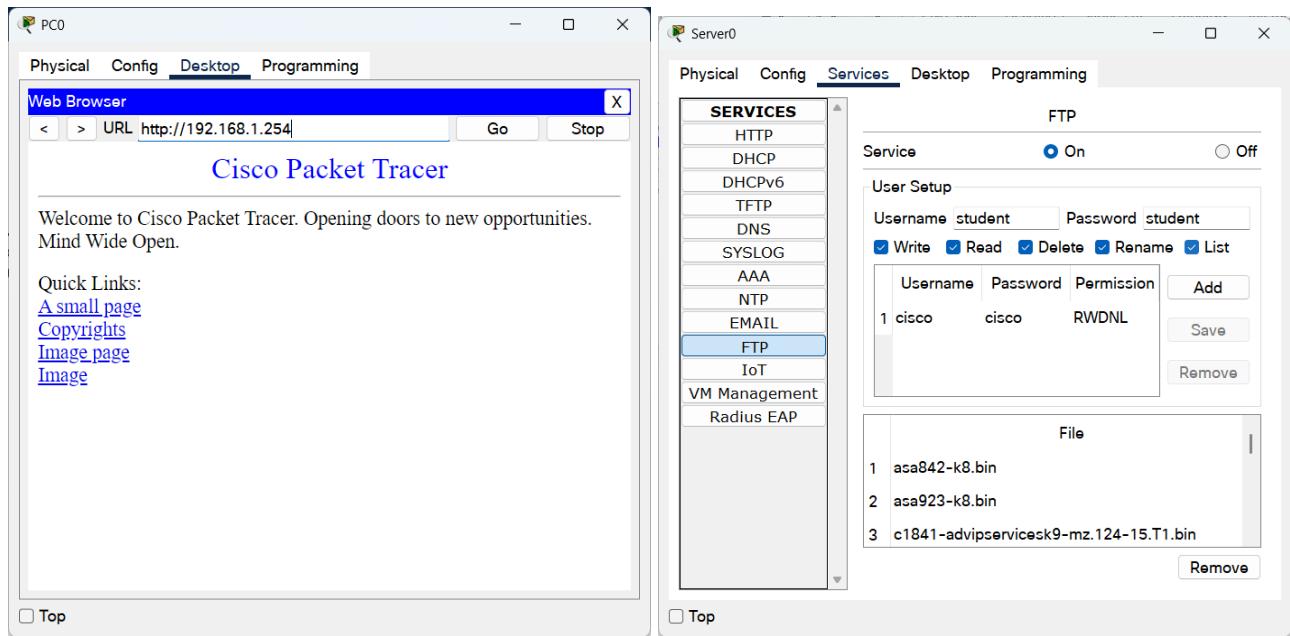


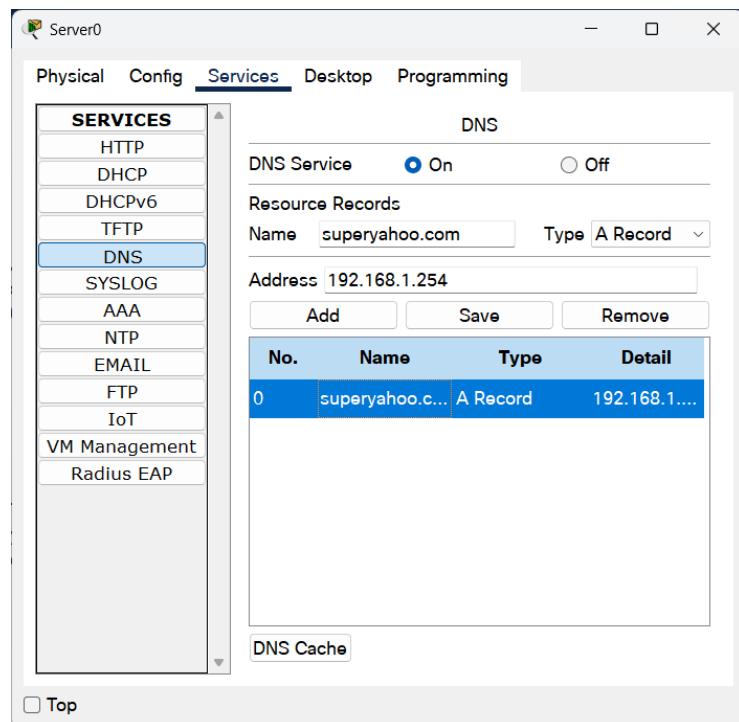
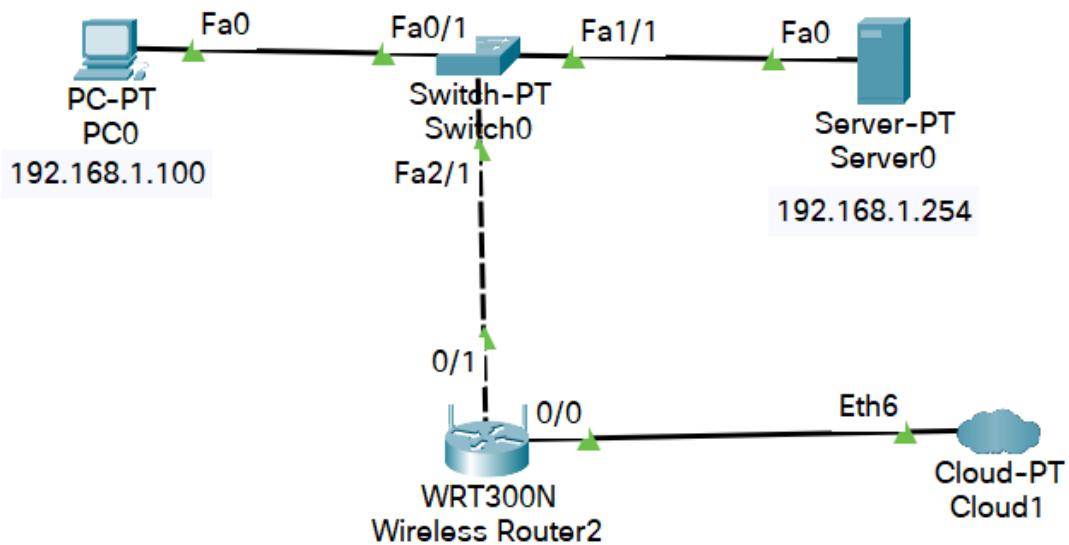
- iii. Screen shots/ output

Server's services



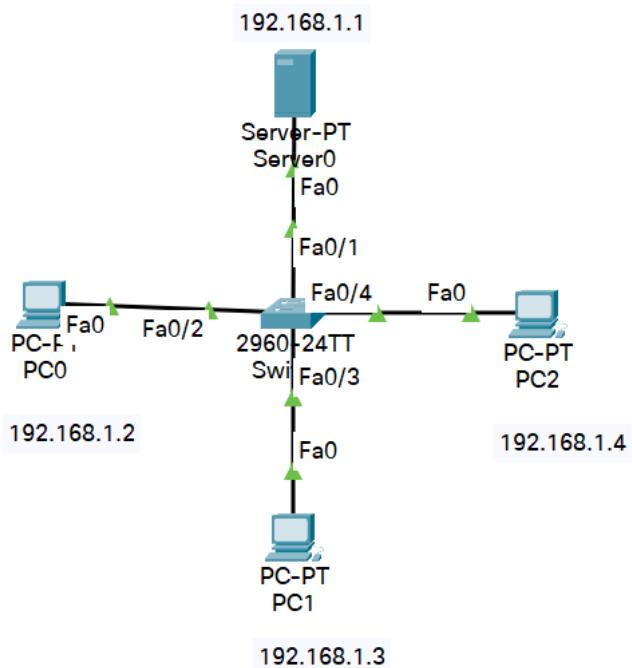
PC's Web Browser





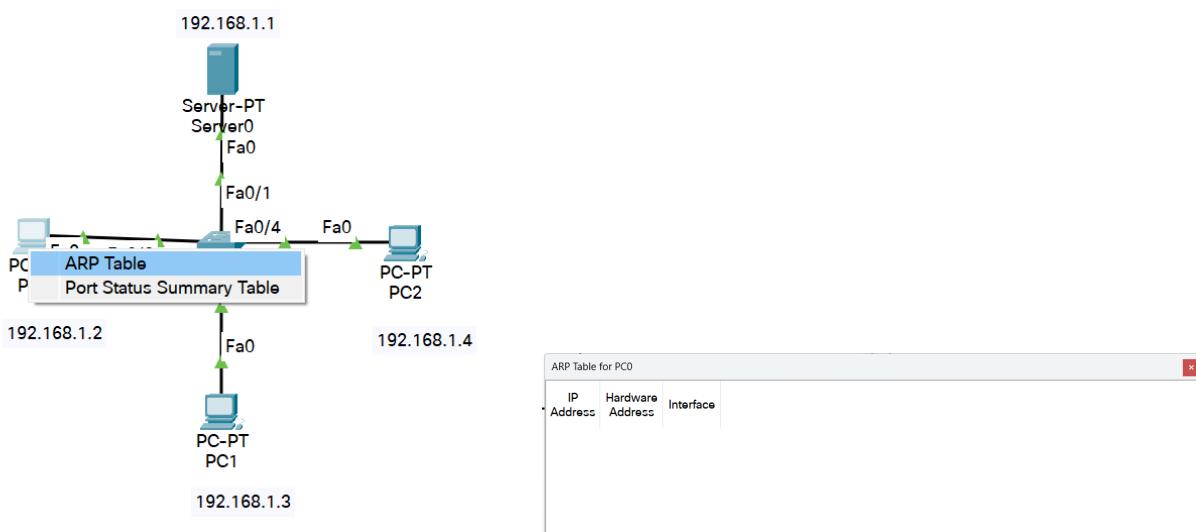
## Program 9

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

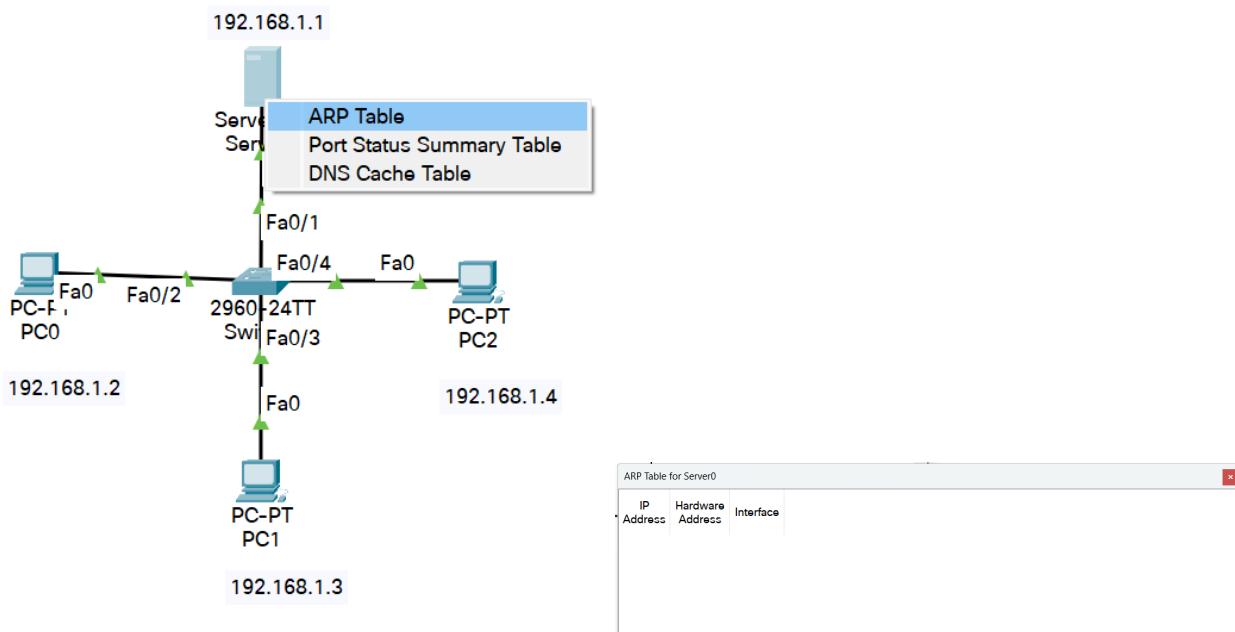


- iii. Screen shots/ output

ARP Table of PC



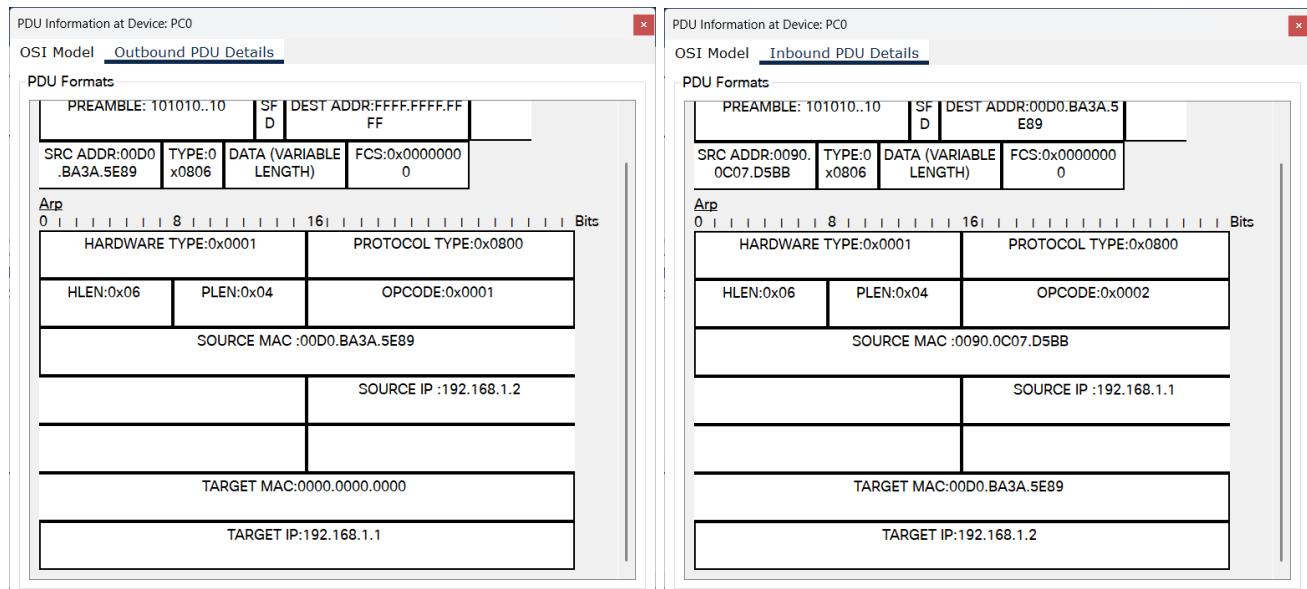
## ARP Table of Server



## Command at PC

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

## Pinging in Simulation Mode



#### iv. Observation

2.2  
01/12/04

LAB-9

AIM: To construct simple LAN & understand the concept & operation of ARP

Topology -

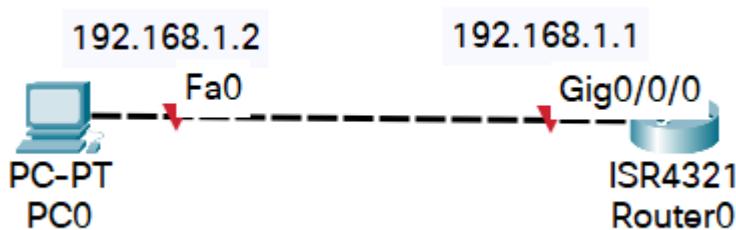
```
graph TD; Server[server PT 192.168.11.4] --- Switch((switch)); Switch --- PC1[PC1 192.168.11.2]; Switch --- PC2[PC2 192.168.11.3]; Switch --- PC3[PC3 192.168.11.1]
```

Procedure -

- Create a topology as shown above & config them with respective IP address.
- Simulation panel → inspect → click PRO  
→ ARP Table → NO entries.
- Repeat for server.
- PRO command prompt  
→ arp -a  
NO ARP entries found.
- Ping server → ping 192.168.11.4
- Two packets are created ICMP & ARP
- click capture to start simulation
- We can see the packet movement of ICMP & ARP from PC to switch to server & vice versa.

## 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

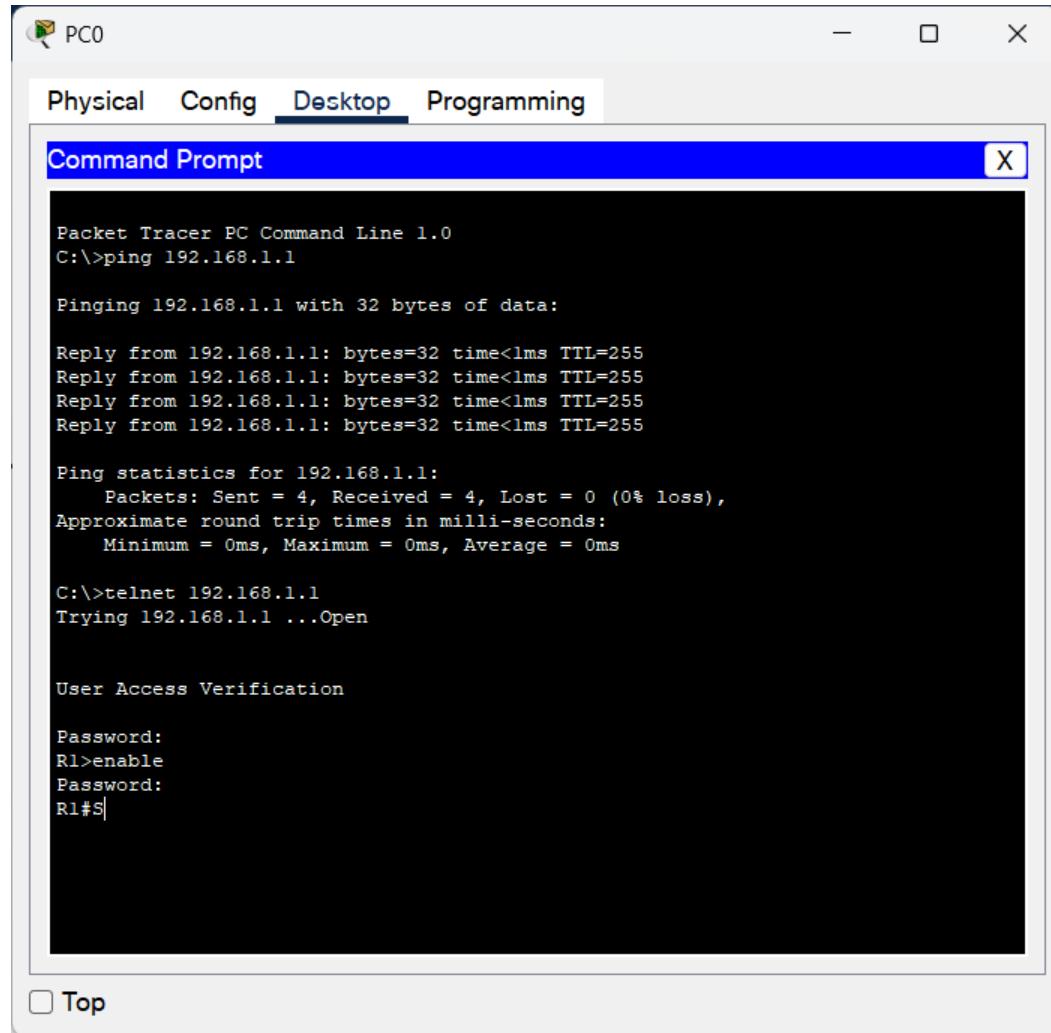
R1(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to up

%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

20/12/24

classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

### TELNET

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

Topology:

PC --- Router (R1) --- X

IP address: 192.168.1.2

Procedure-

- \* Assign IP address & Default gateway for PC
- \* Config router in R1

Router # config t

Router (config) # hostname R1

R1 (config) # enable secret rp

R1 (config-if) # ip add 192.168.1.1 255.255.255.0

R1 (config-if) # no shutdown

→ Go to PC command prompt

C:\> telnet 192.168.1.1

Trying 192.168.1.1... open

User access verification.  
Password:

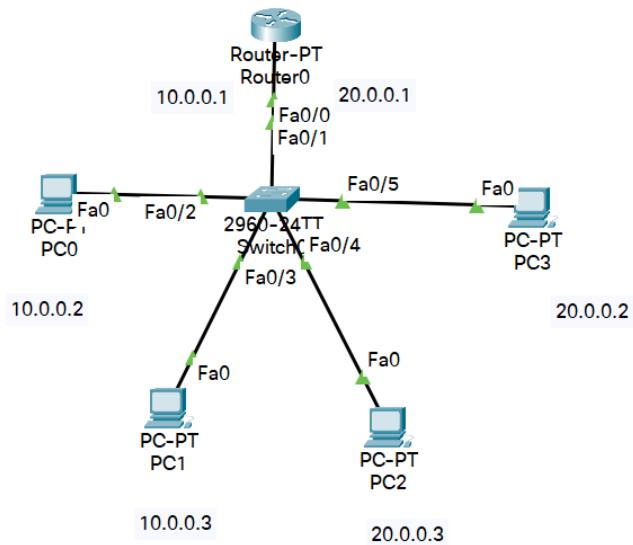
R1> en

Password:

R1# rp

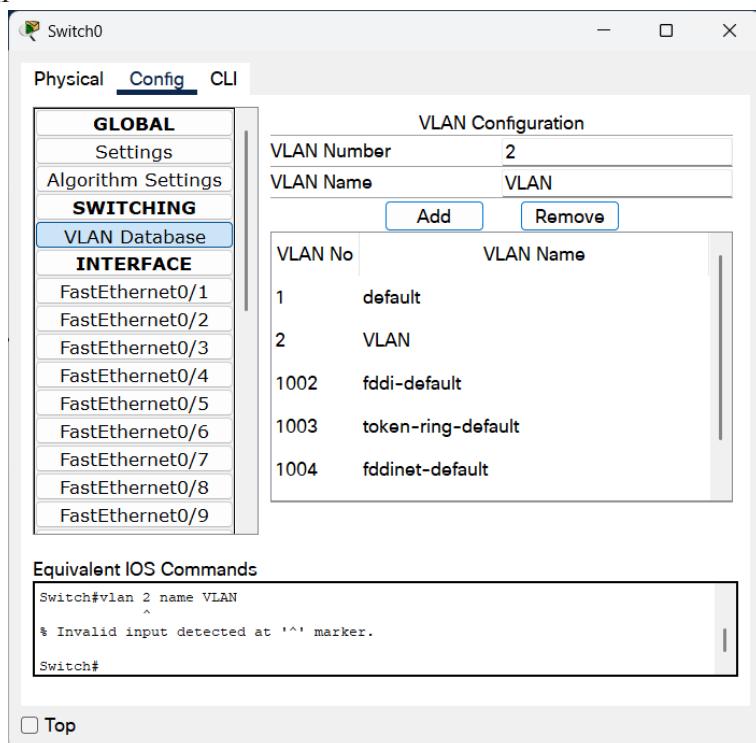
## Program 11

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

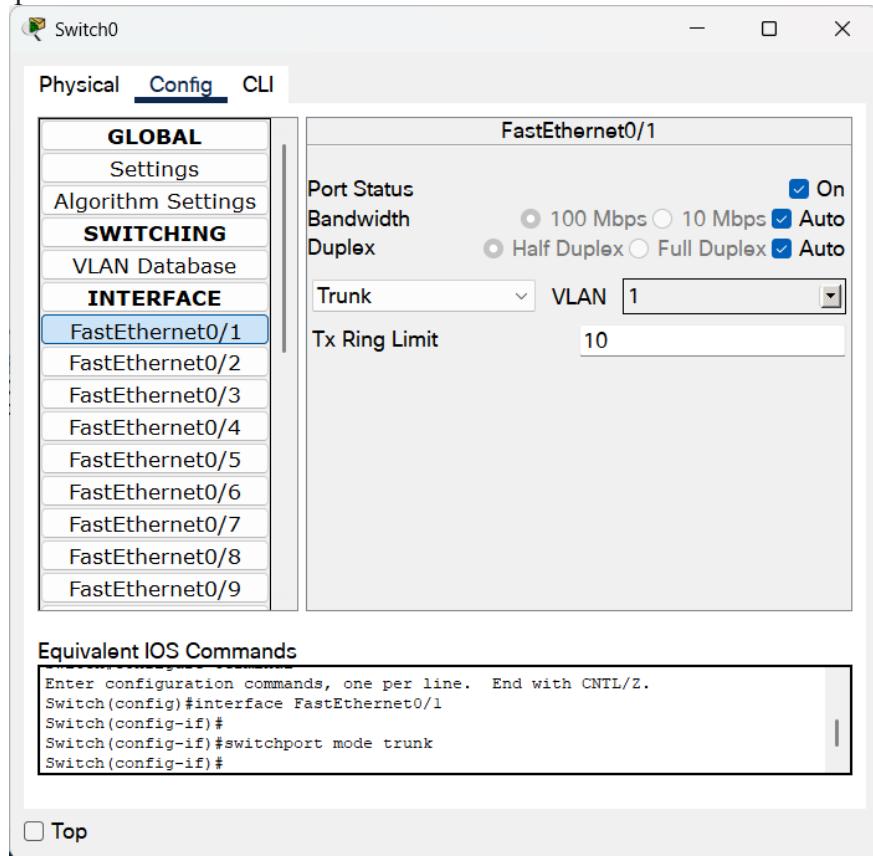


- iii. Screen shots/ output

Switch Configuration



## Configuration of port connected to router



## Configuration of ports connected to other networks

This screenshot shows the configuration interface for "Switch0" with two ports selected: "FastEthernet0/4" and "FastEthernet0/5".

- FastEthernet0/4 Configuration:** The "Access" dropdown is set to "VLAN 2".
- FastEthernet0/5 Configuration:** The "Access" dropdown is set to "VLAN 2". The "VLAN 2" dropdown shows three entries: "1:default", "2:VLAN", and "1002:fdi-default".

Below each port's configuration, there is an "Equivalent IOS Commands" section:

- FastEthernet0/4:**

```

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

```
- FastEthernet0/5:**

```

Switch(config)#interface FastEthernet0/5
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

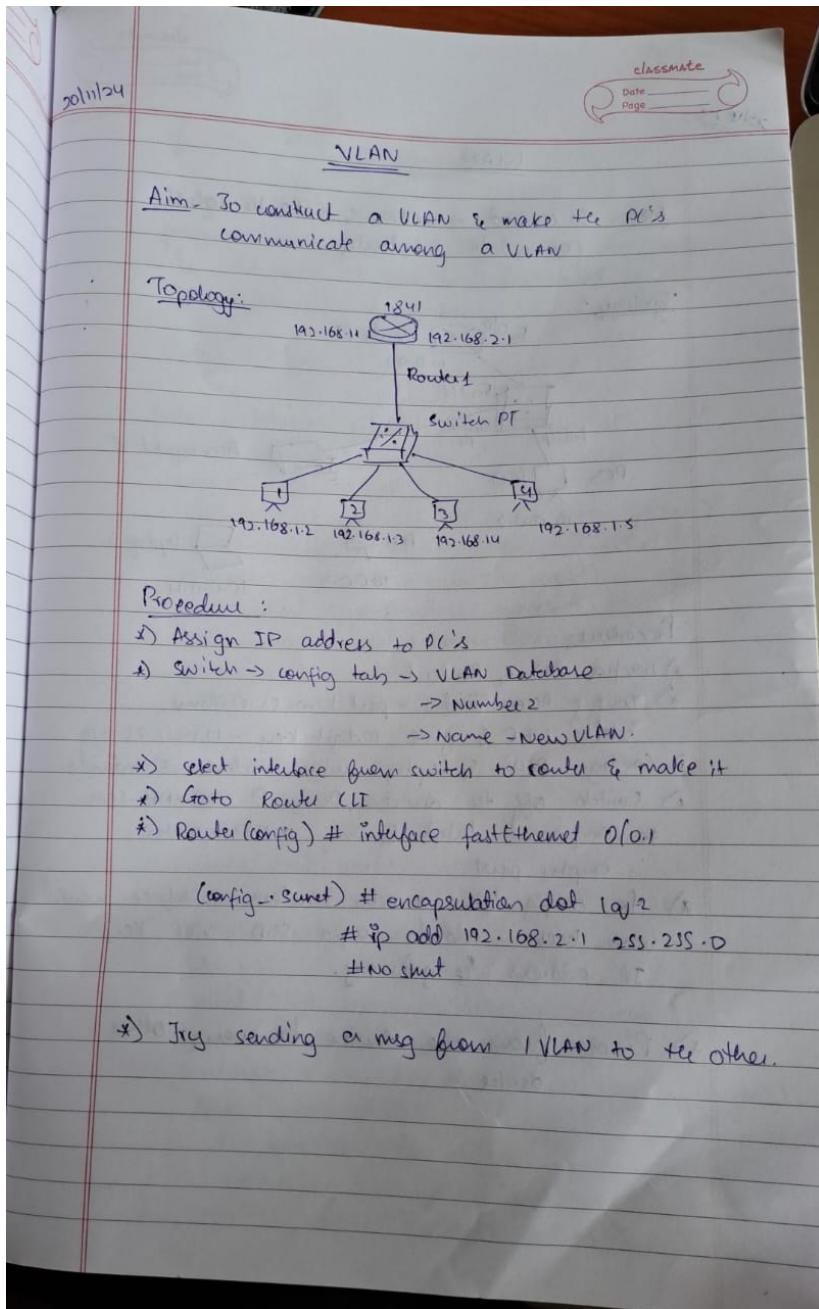
Router(config-subif)#encapsulation dot1q 2
Router(config-subif)#ip address 20.0.0.1 255.0.0.0
Router(config-subif)#no shutdown
Router(config-subif)#exit
Router(config)#

Ctrl+F6 to exit CLI focus
```

Top

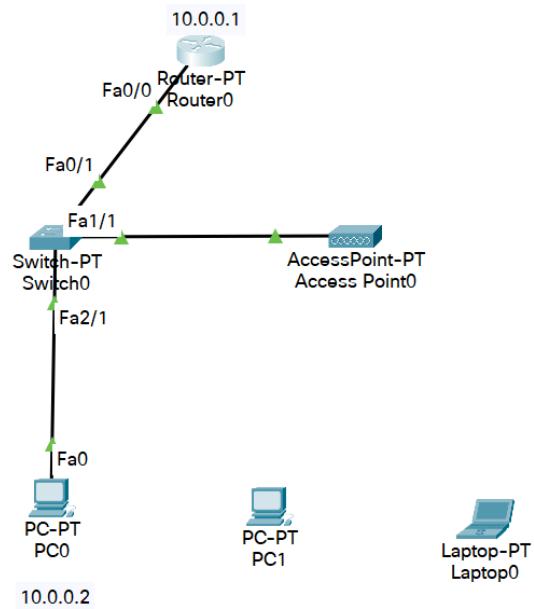
**Copy** **Paste**

#### iv. Observation

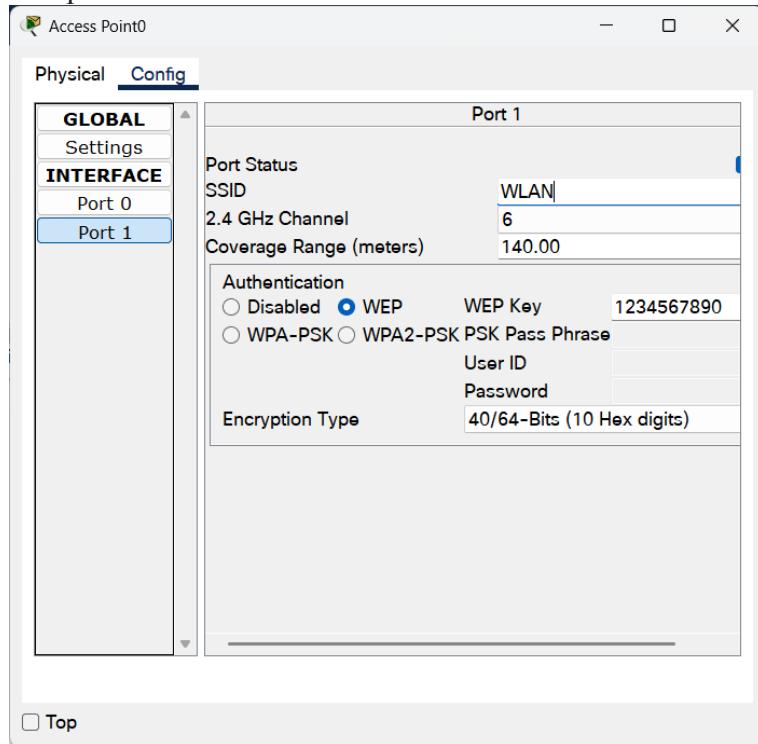


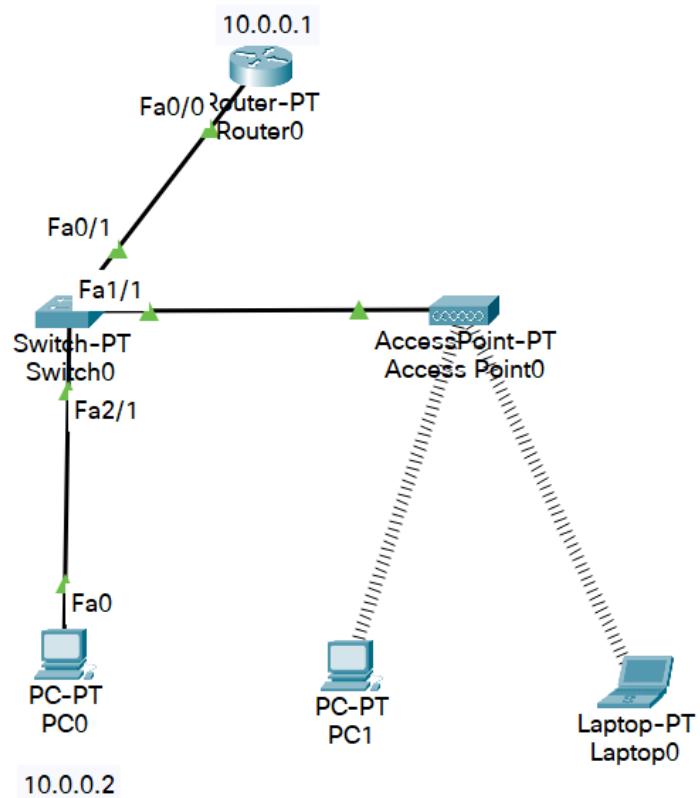
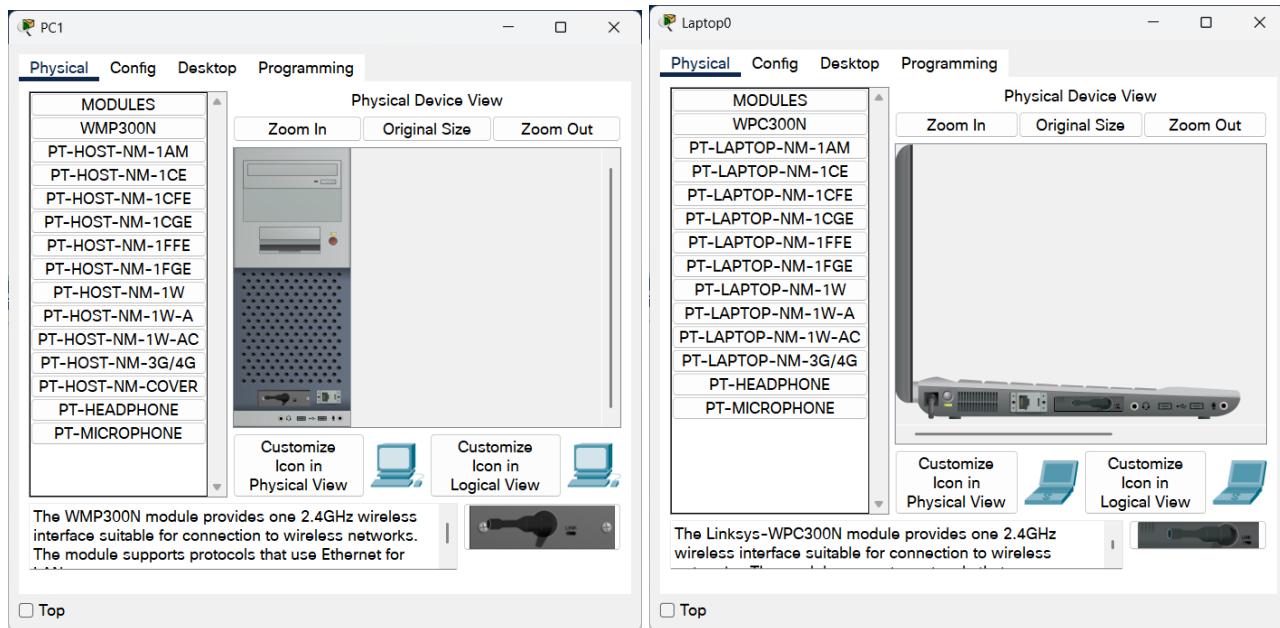
## Program 12

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

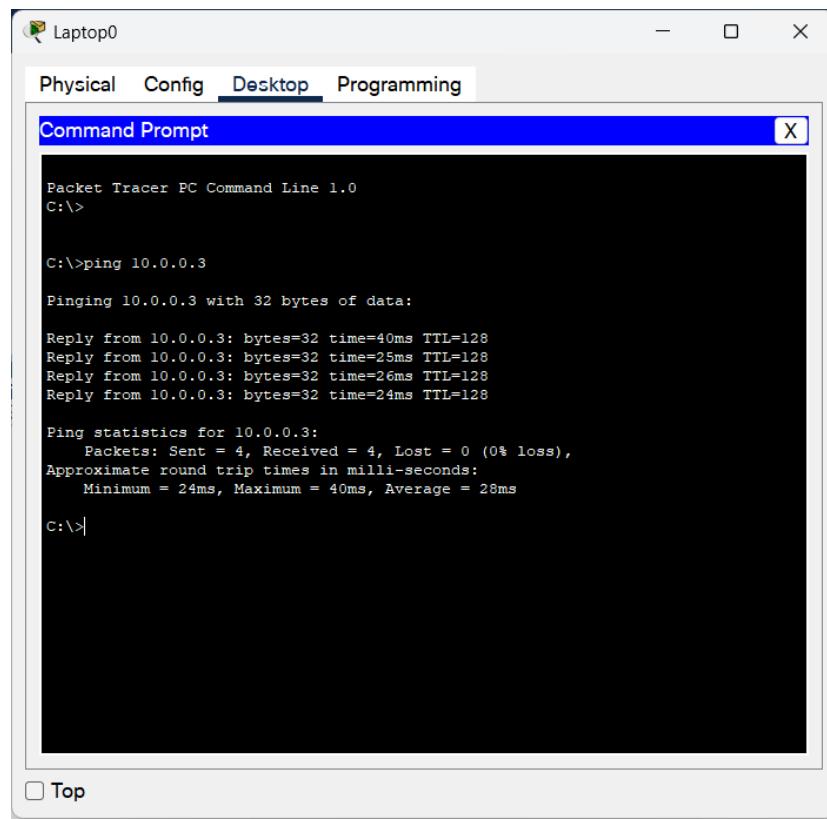


- iii. Screen shots/ output





Ping:



Packet Tracer PC Command Line 1.0  
C:\>

C:\>ping 10.0.0.3

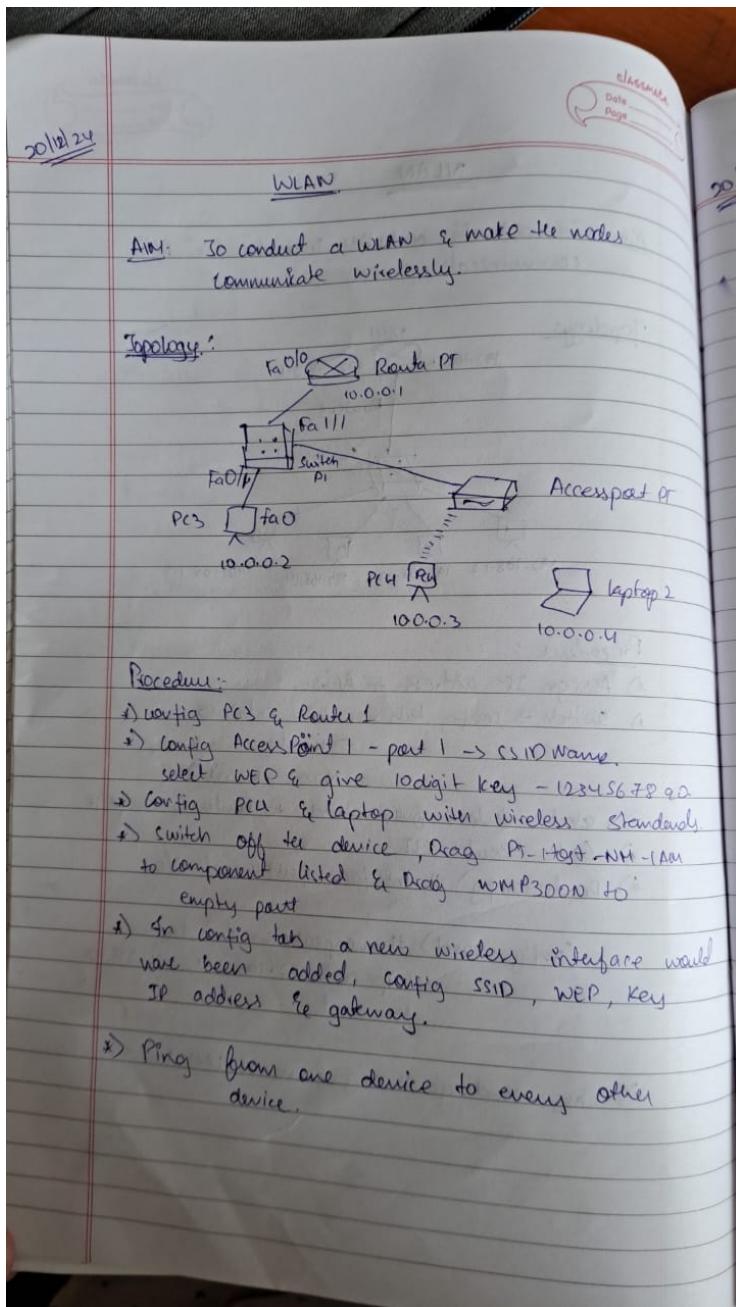
Pinging 10.0.0.3 with 32 bytes of data:

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:\>

#### iv. Observation



## 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., 11010011101100110110110111000011): "
)

# 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

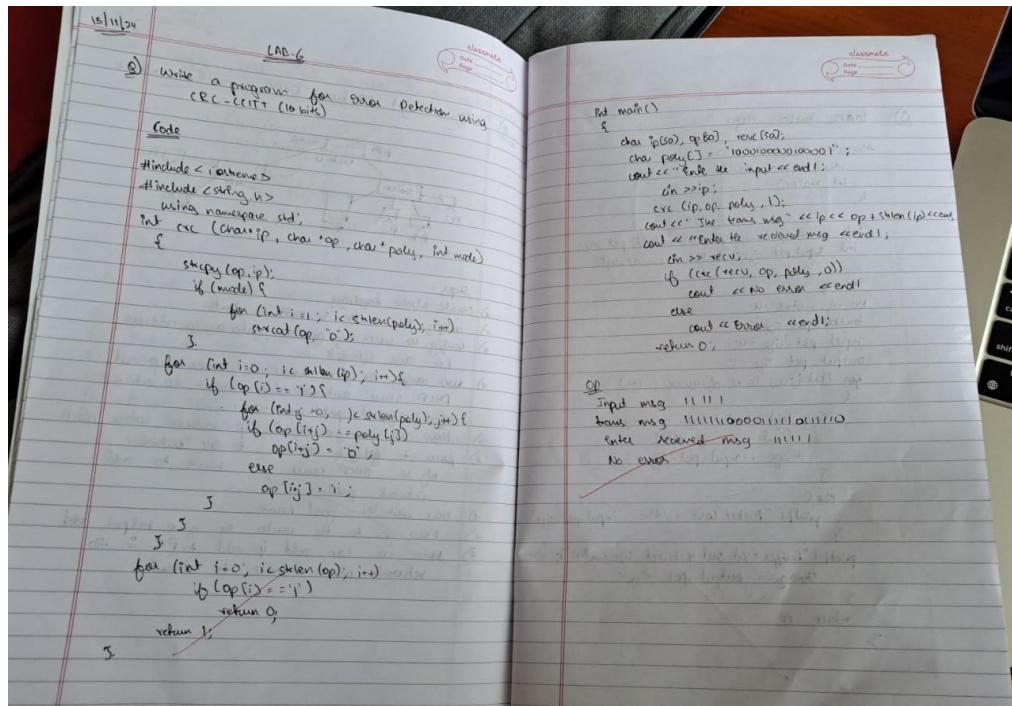
In [2]: runcell(0, 'E:/python_files/untitled2.py')

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



## 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

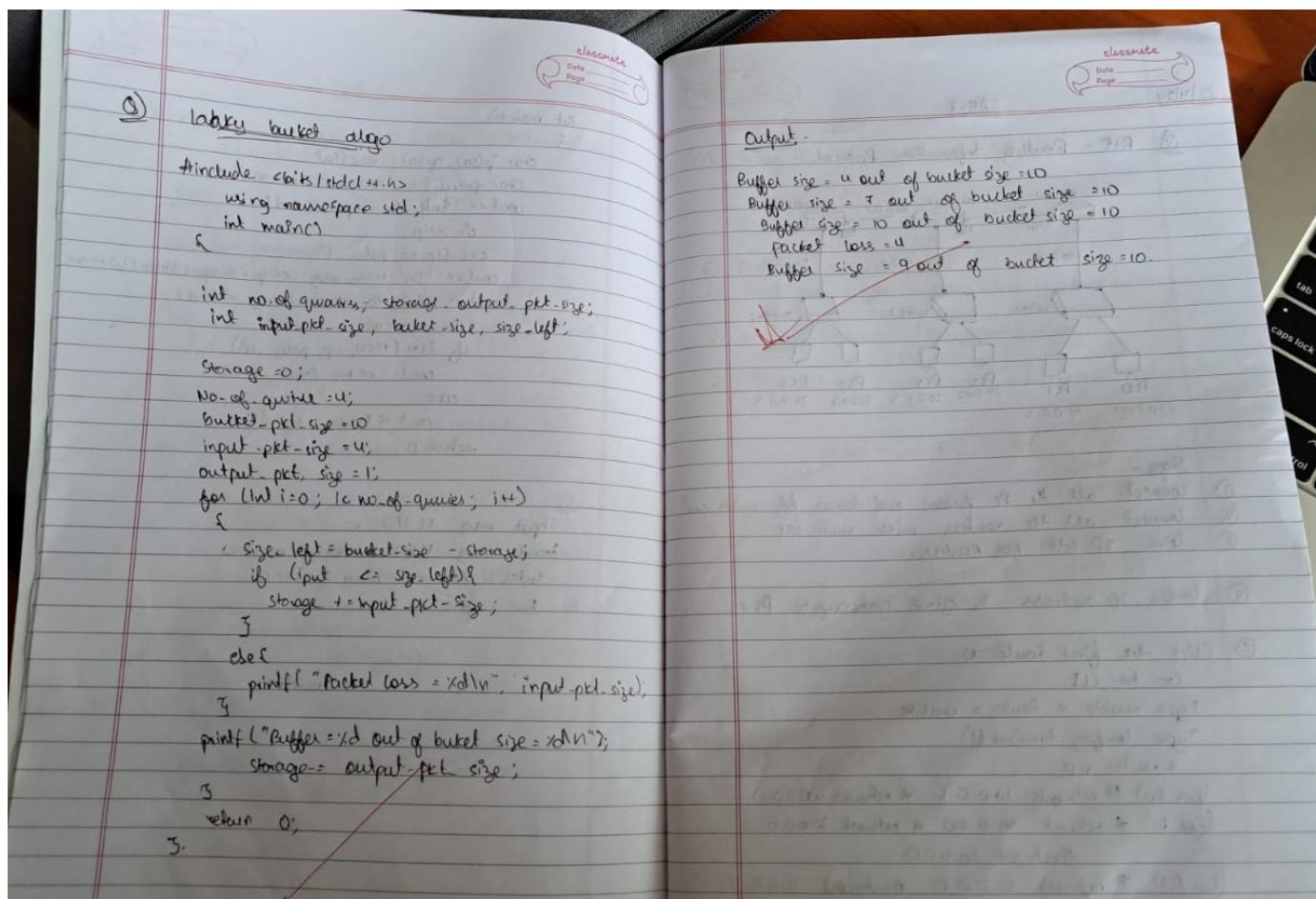
if __name__ == "__main__":
```

## 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



### **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



The screenshot shows the Python IDLE Shell interface. The title bar reads "IDLE Shell 3.12.6". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python code:

```
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

TCP/IP Sockets

Using TCP/IP sockets, write a Client - server program to make client sending the filename to server to send back the content of the 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 name:")
clientSocket.send(sentence.encode())
fileContent = clientSocket.recv(1024).decode()
print(fileContent)
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(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("In sent contents of " + sentence)
```

file.close()  
connectionSocket.close()

Output:

server TCP.py  
The server is ready to receive  
sent contents of server TCP.py  
The server is ready to receive

Client TCP.py  
Enter file name : server TCP.py

from server:  
from socket import \*

:::  
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)
```

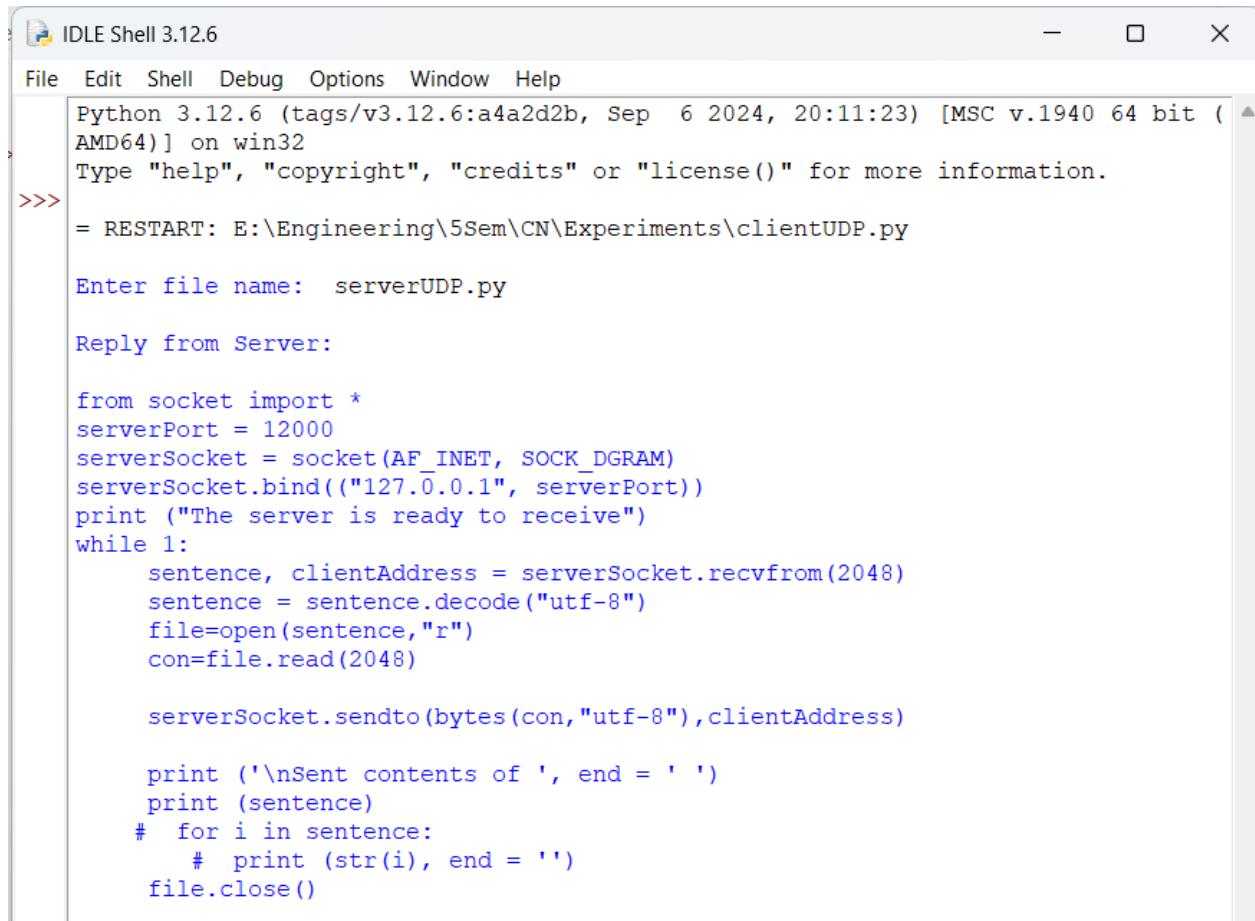
```

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

```

### iii. Screen shots/ output

#### Client



The screenshot shows the Python IDLE Shell interface. The title bar reads "IDLE Shell 3.12.6". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python code:

```

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

20/10/24

UDP sockets

Using UDP sockets, write a client-server program to make client sending file name & the server to send back the content of requested file if present.

Client UDP.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("Enter file name")
clientSocket.sendto(sentence.encode("utf-8"), (serverName, serverPort))
fileContent, serverAddress = clientSocket.recvfrom(2048)
print("Reply from server")
print(fileContent.decode("utf-8"))
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 True:
    sentence, clientAddress = serverSocket.recvfrom(1024)
    sentence = sentence.decode("utf-8")
    file = open(sentence, 'r')
    content = file.read(2048)
    file.close()
    serverSocket.sendto(content.encode("utf-8"), clientAddress)
```

Output:

server UDP.py

The server is ready to receive  
Sent contents of server UDP.py  
The server is ready to receive

Client UDP.py

Enter file name: server UDP.py  
Reply from server

from socket import \*

=====

file.close().