

# **VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

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



## **LAB RECORD**

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

*Submitted by*

**Shreya Mitawa (1BM22CS266)**

*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 **Shreya Mitawa (1BM222CS266)**, 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.

Megha J 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	23
7	22/11/2024	Demonstrate the TTL/ Life of a Packet	29
8	08/11/2024	Configure Web Server, DNS within a LAN.	31
9	20/12/2024	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)	35
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	41
12	20/12/2024	To construct a WLAN and make the nodes communicate wirelessly	45

Github Link:

<https://github.com/shreyamitawa/CN.git>

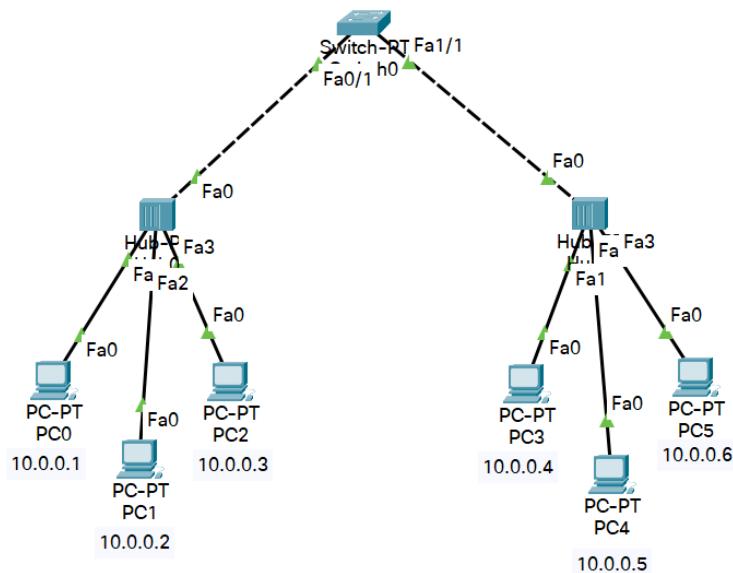
## **Index-Cycle-II**

<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).	49
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

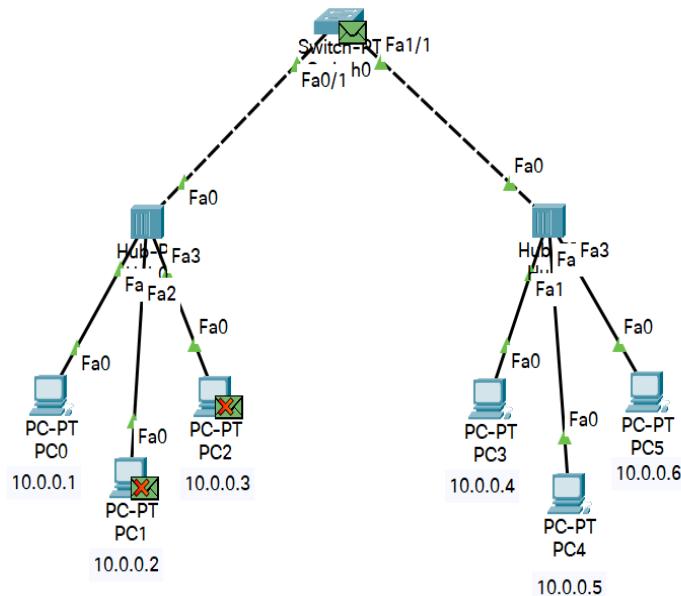
## Program 1

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

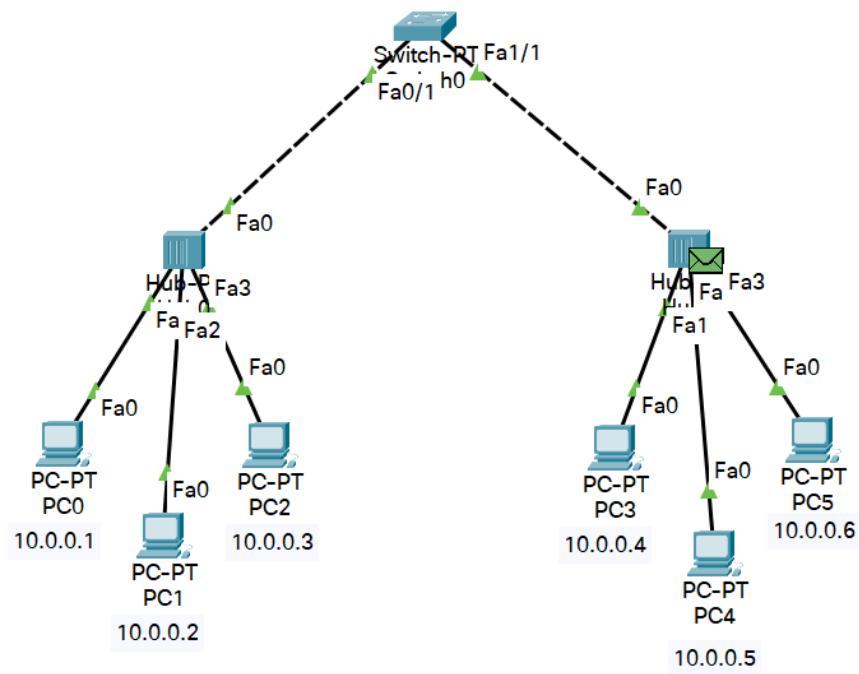


- iii. Screen shots/ output

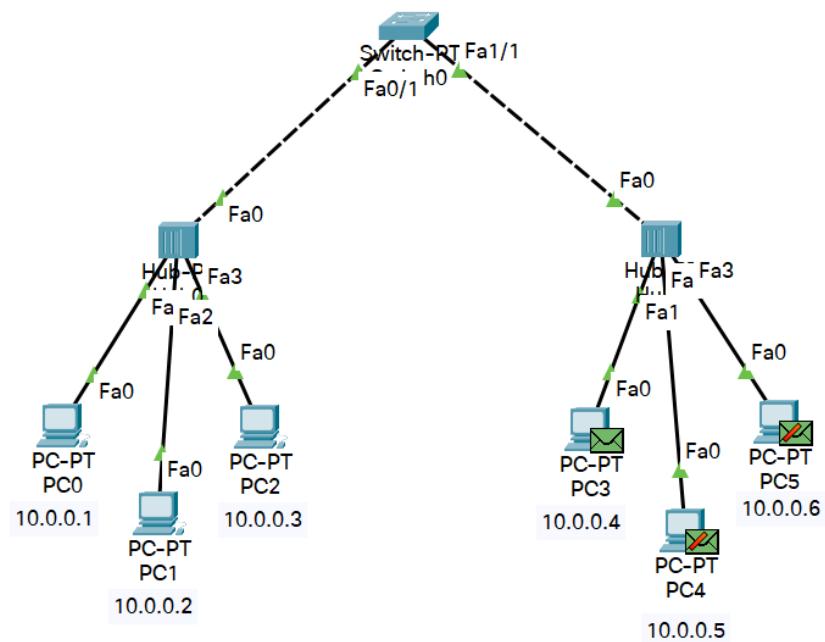
Hub behaviour at sending end



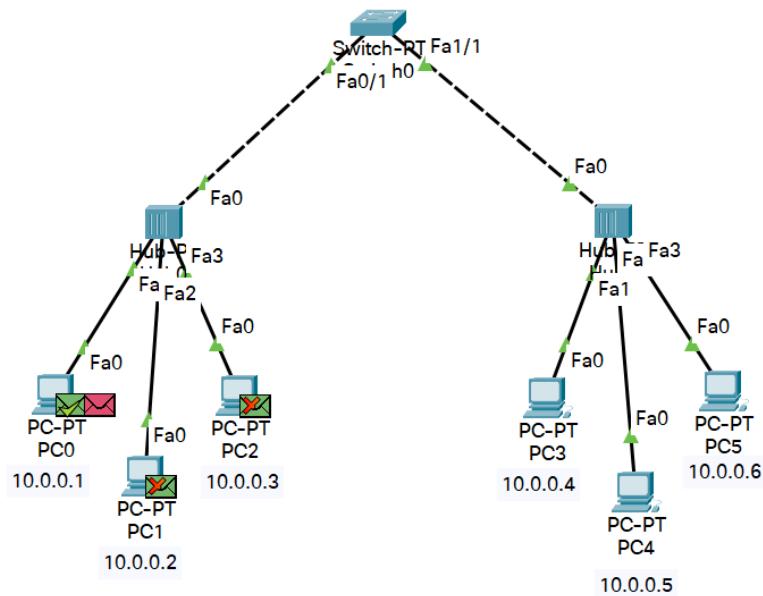
## Switch behaviour



## Hub behaviour at receiving end



## Hub behaviour when back to sender



Ping command to connectivity

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

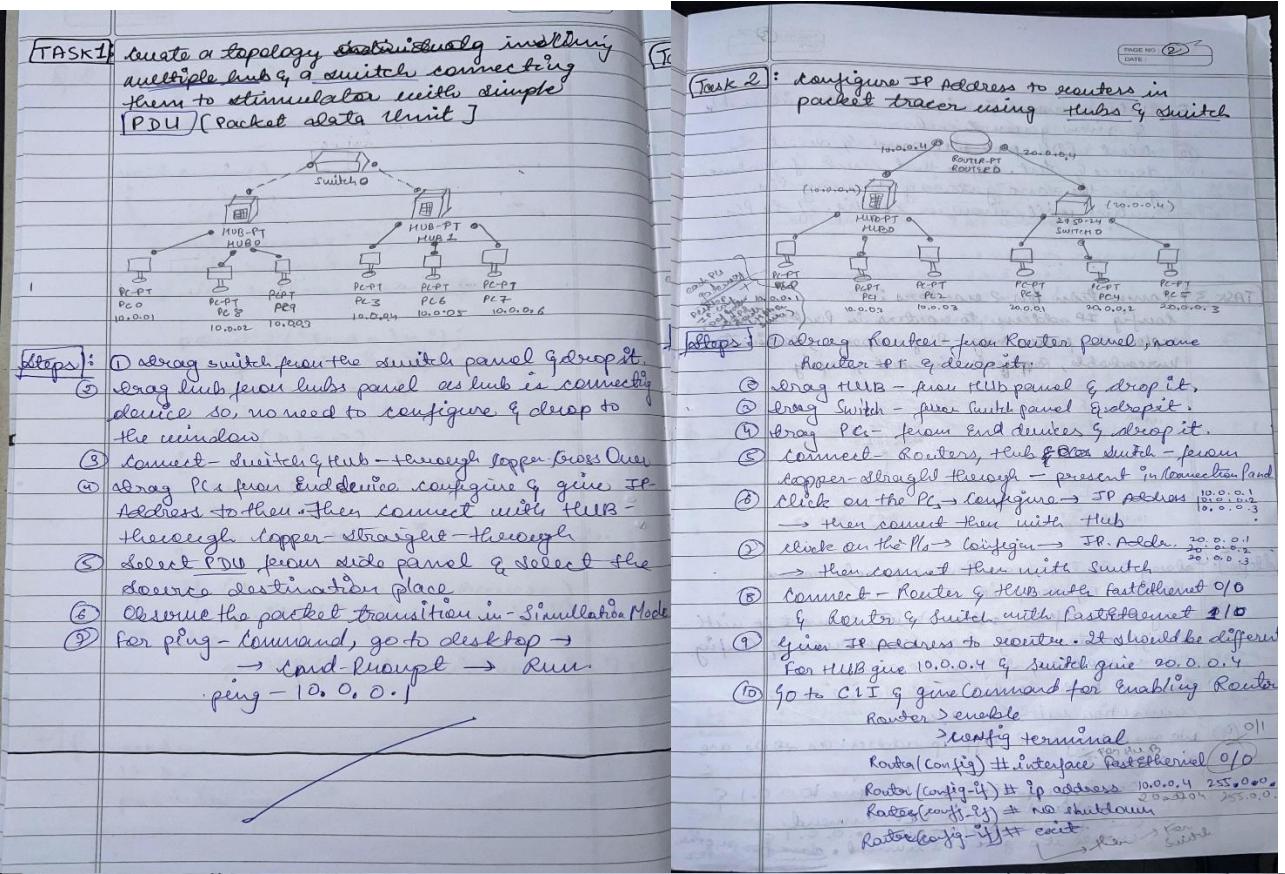
Pinging 10.0.0.4 with 32 bytes of data:

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:\>
```

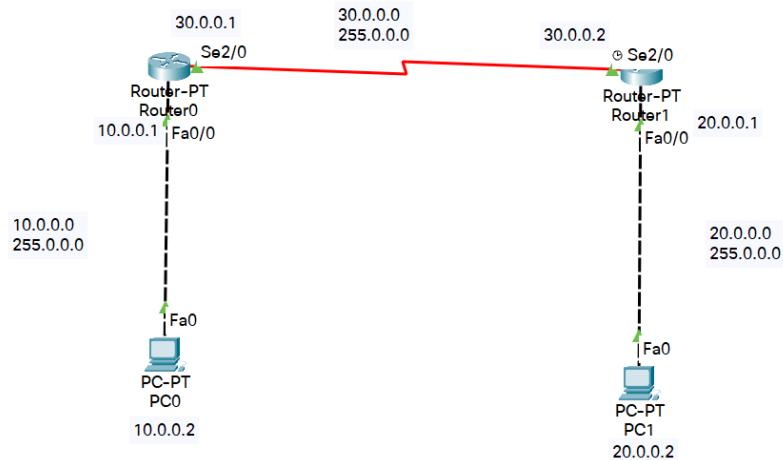
#### iv. Observation



- 11 If the command is ✓, then router will enable & gives green signal.
- 12 Select PDU from side panel & select the source & dest. before that click if you have given the same gateway to every PCs which will show in config. of the PCs.  
→ then Ping Down isk → comp. the capture / forward

## Program 2

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



- iii. Screen shots/ output

Router0 configuration

The screenshot shows the Cisco IOS Command Line Interface (CLI) for Router0. The window title is "Router0". The tabs at the top are "Physical", "Config", and "CLI", with "CLI" selected. The main area is titled "IOS Command Line Interface" and 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 10.0.0.1 255.0.0.0
Router(config-if)#no shutdown

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

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

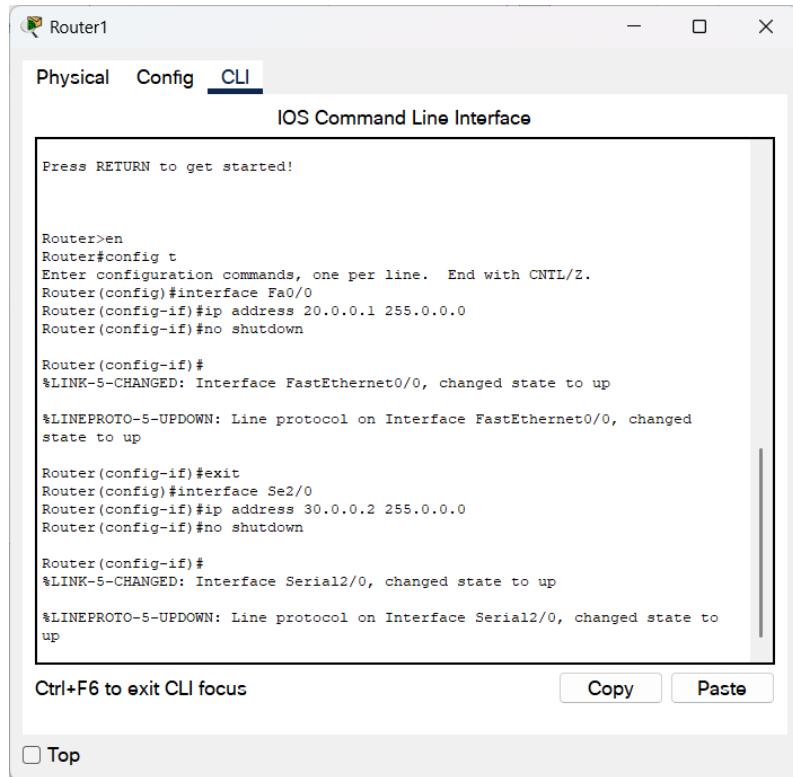
Router(config-if)#exit
Router(config)#interface Se2/0
Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#no shutdown

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

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

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

## 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". A message box says "Press RETURN to get started!". The main text 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 of the window, there are buttons for "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

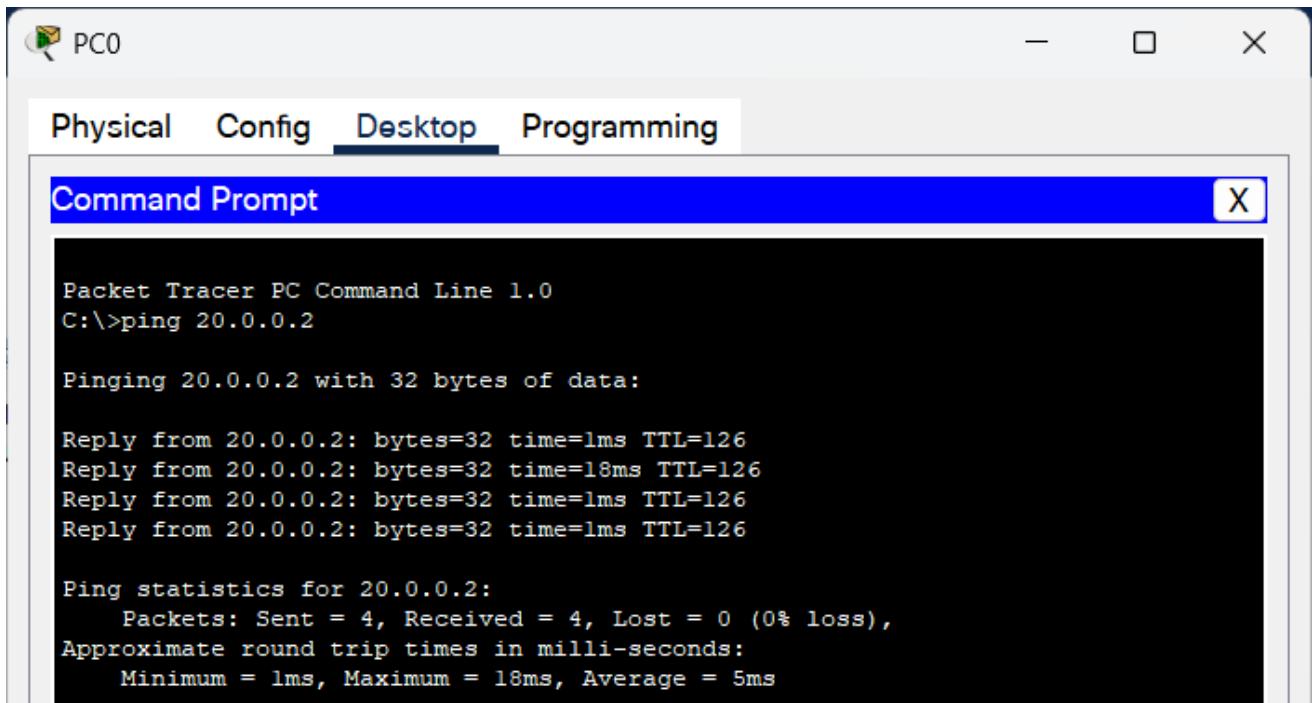
```
Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.2

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 X

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

Pinging 20.0.0.2 with 32 bytes of data:

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

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

iv. Observation

**TASK 3: Connection b/w 2 routers :-**

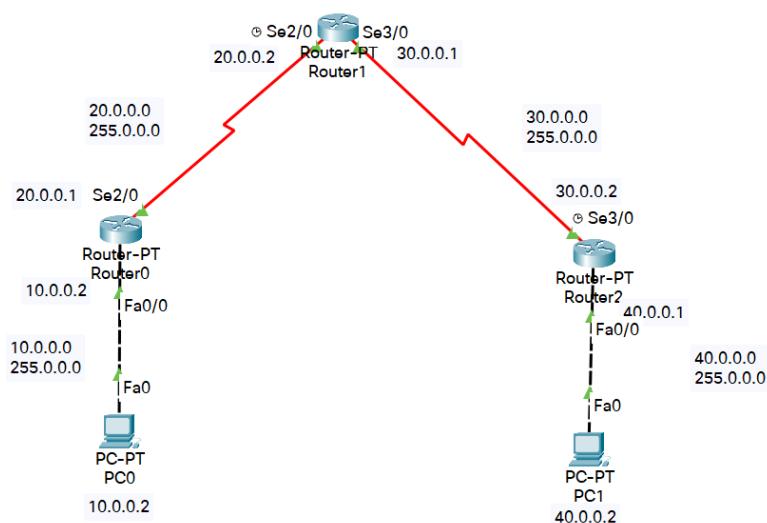
config IP address to routers in Packet tracer, explore the follows. msg - Ping response, Destination unreachable, Request time out, reply

**Steps:**

- ① Select 2 routers - from Router panel - name Router PT & connect it.
- ② Drag 2 PCs - from Endless Device, connect it with 2 diff routers using Copper-wire over & config + PC as 10.0.0.2 & no pcas 20.0.0.2. Connect both routers with Serial DTE - connection with S2/0. Add serial Port in physical to both router, connect.
- ③ We need to give 2 diff ip address as there are 2 connections. Router 1 IP 10.0.0.1 & Router 2 IP 20.0.0.1. One = for fastethernet 0/0 - give 10.0.0.1 & enable it or give clt command. One = for serial 1/0 - give 20.0.0.1 & enable it or give clt command. Same for another router.
- ④ Go to CLI exit it & come to Router give command & Router # show ip route gateway of last router it not set
- ⑤ Router# show ip route
  - 10.0.0.0/8 is directly connected, fastethernet 0/0
  - 30.0.0.0/8 is directly connected, serial 2/0
  - 20.0.0.0/8 is directly connected, fastethernet 0/0
  - 30.0.0.0/8 is directly connected, serial 2/0
- ⑥ For sending msg, we need to connect all the IP-Address with each other. For that, give command Router# config t
- Router(config)# ip route 20.0.0.0 255.0.0.0 30.0.0.2
  - so same thing for another Router
  - Router(config)# ip route 20.0.0.0 255.0.0.0 30.0.0.1
- ⑦ Now, send the PDU, it will show successful!

### **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

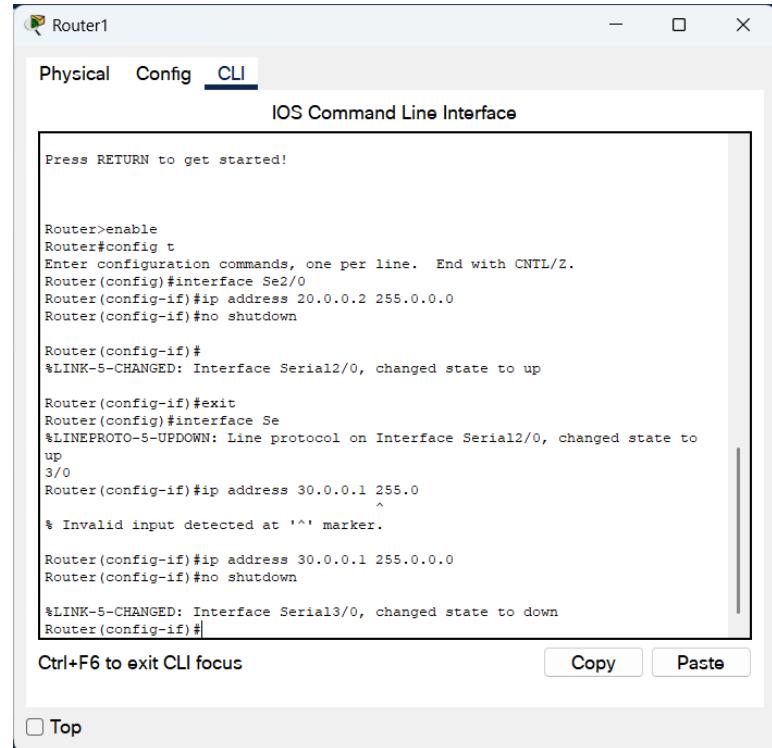
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%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



Router1

Physical Config CLI

IOS Command Line Interface

```
Press RETURN to get started!

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

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

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

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

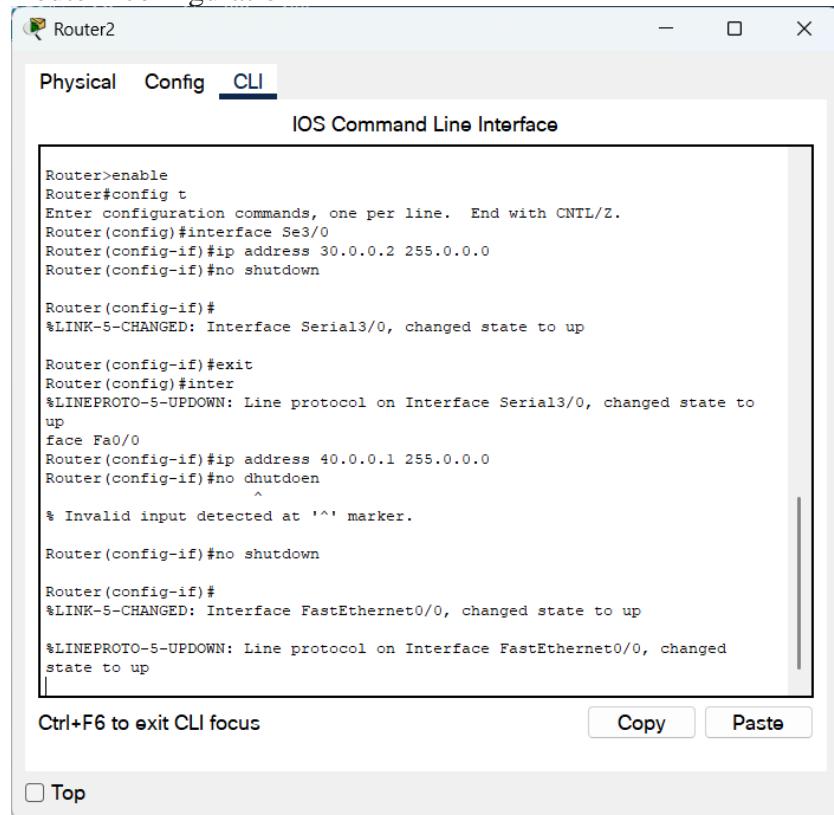
%LINK-5-CHANGED: Interface Serial3/0, changed state to down
Router(config-if)#

Ctrl+F6 to exit CLI focus
```

Top

Copy Paste

## Router2 configuration



Router2

Physical Config CLI

IOS Command Line Interface

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

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

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

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
```

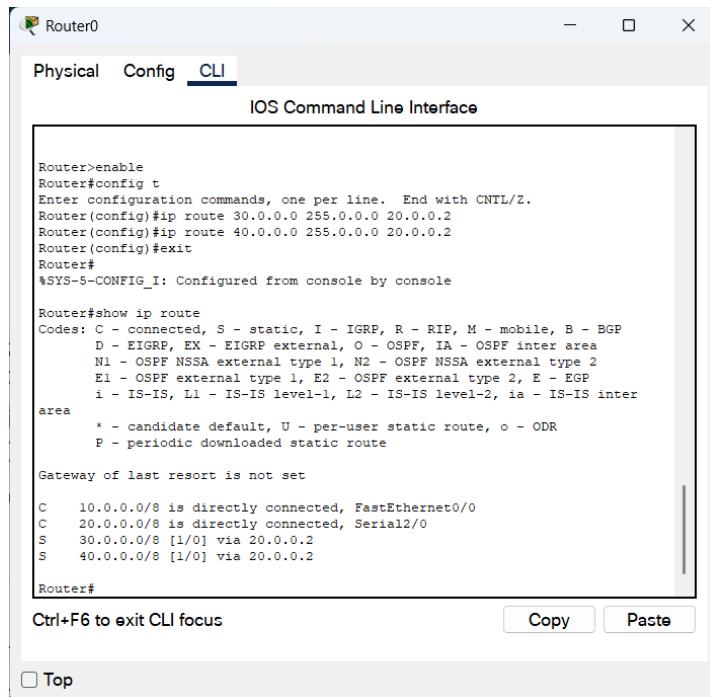
Top

Ctrl+F6 to exit CLI focus

Copy Paste

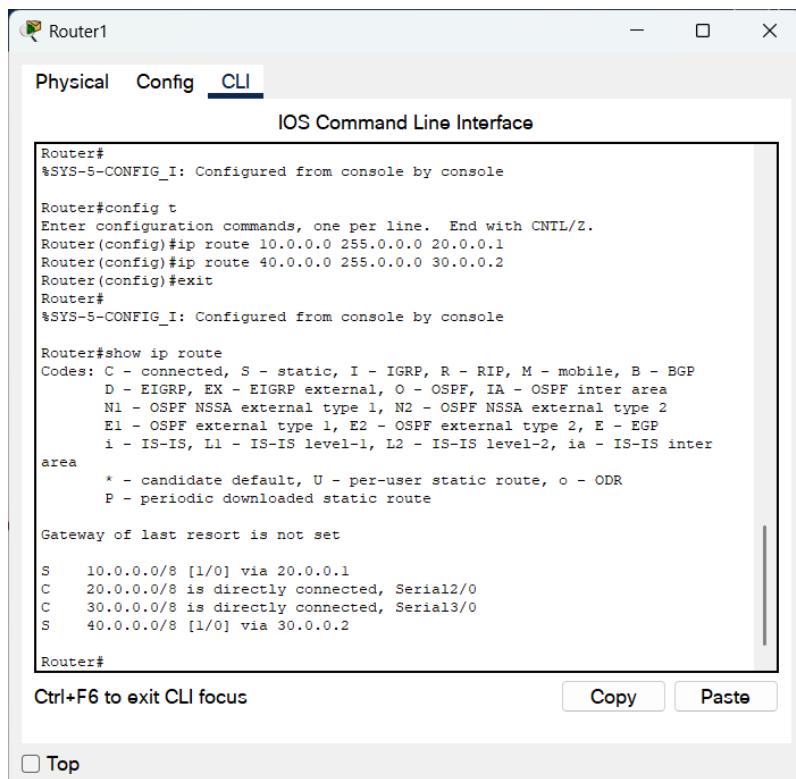
## Static Routing:

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#

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#

## 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

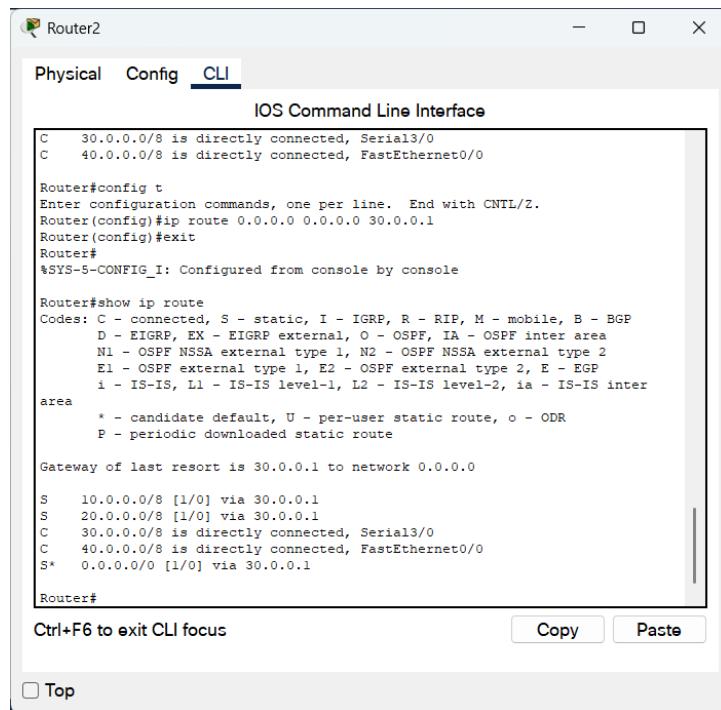
C  10.0.0.0/8 is directly connected, FastEthernet0/0
C  20.0.0.0/8 is directly connected, Serial2/0
S  30.0.0.0/8 [1/0] via 20.0.0.2
S  40.0.0.0/8 [1/0] via 20.0.0.2
S* 0.0.0.0/0 [1/0] via 20.0.0.2

Router#
```

Ctrl+F6 to exit CLI focus     

Top

## Router2



The screenshot shows the IOS Command Line Interface window titled "Router2". The "CLI" tab is selected. The main pane displays the following configuration and routing 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:

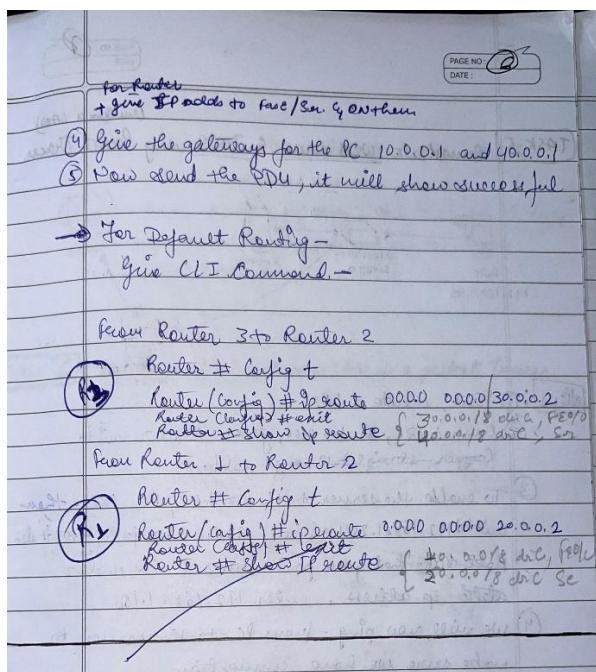
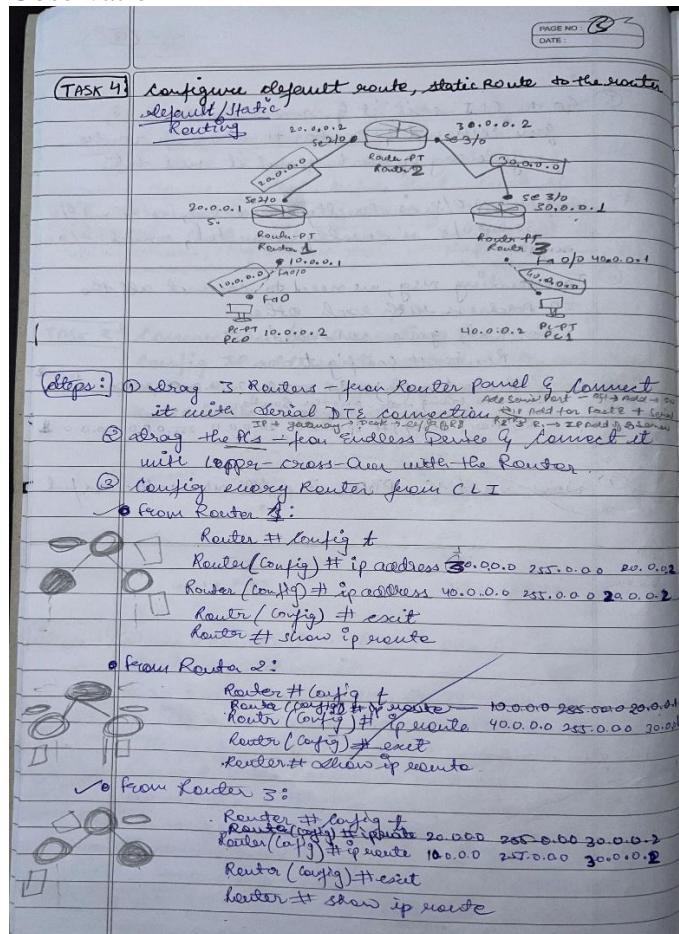
```
C:\>ping 40.0.0.2

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

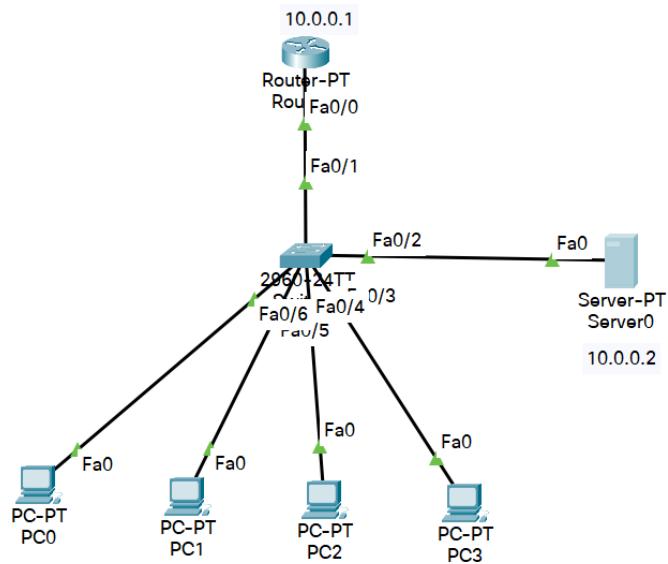
Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 25ms, Average = 16ms
```

#### iv. Observation



## Program 4

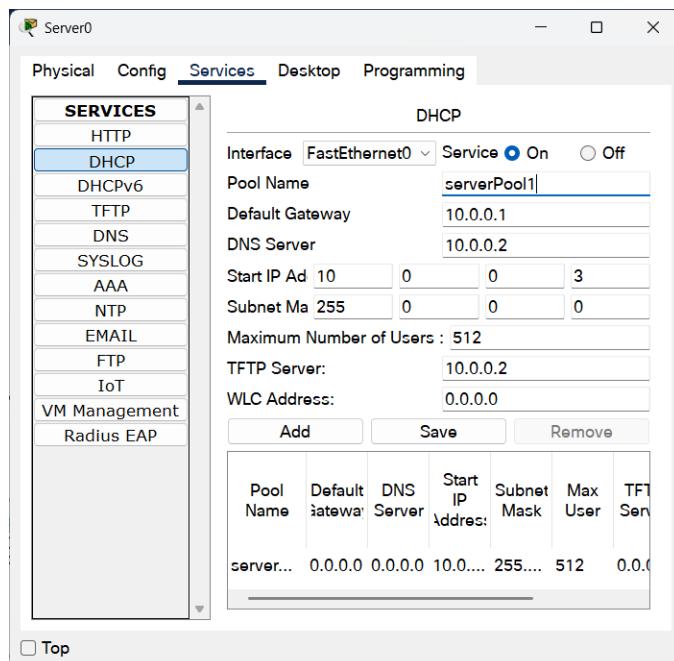
- i. Configure DHCP within a LAN and outside LAN.
- ii. Procedure along with the topology



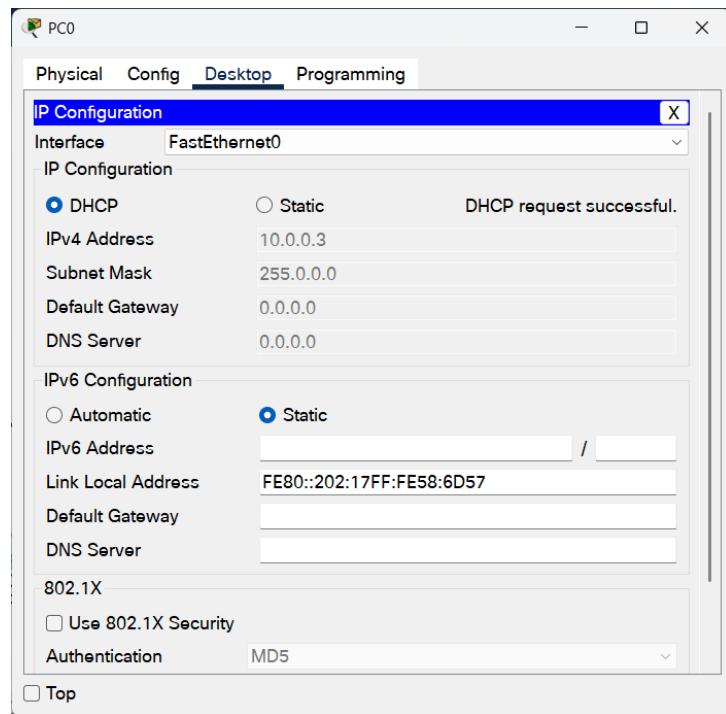
- iii. Screen shots/ output

DHCP Within LAN

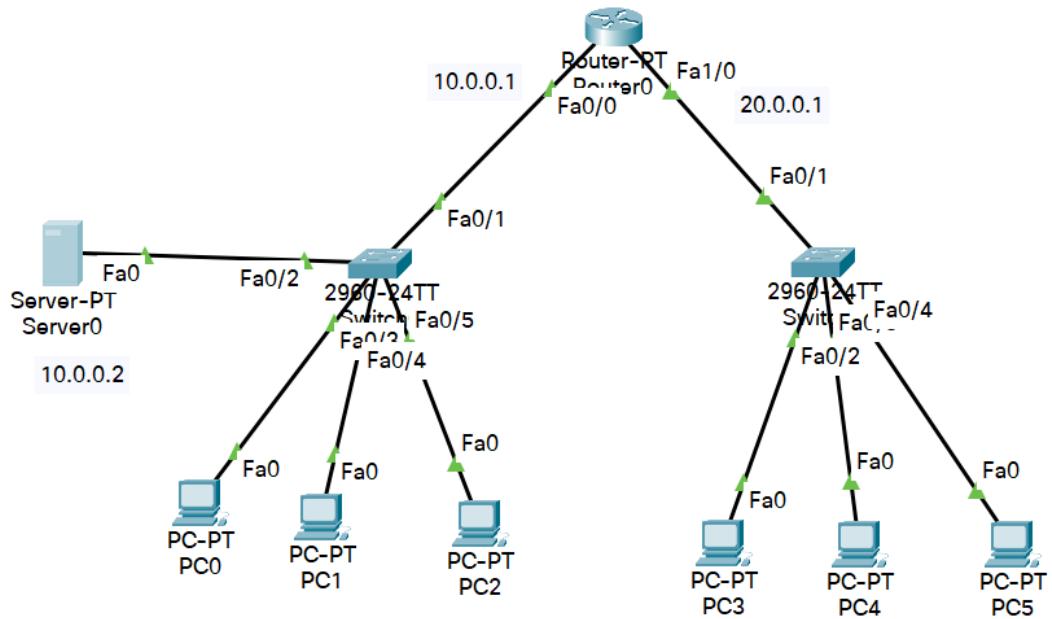
DHCP Configuration



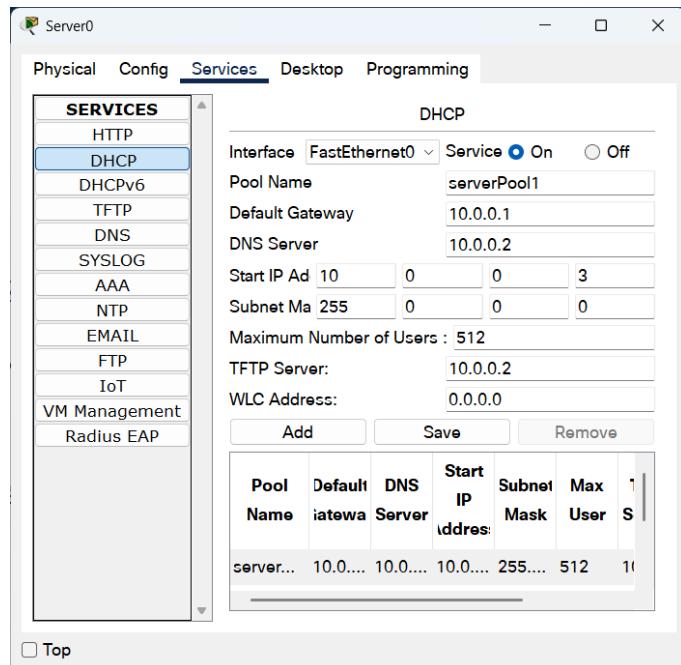
## PC settings



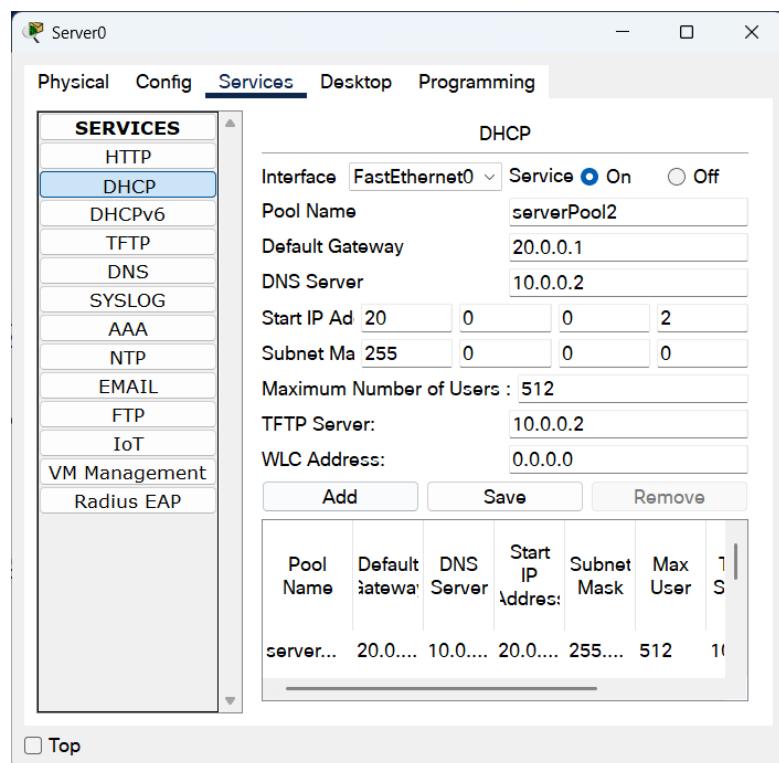
DHCP outside LAN:



DHCP configuration for inside LAN



DHCP configuration for outside LAN



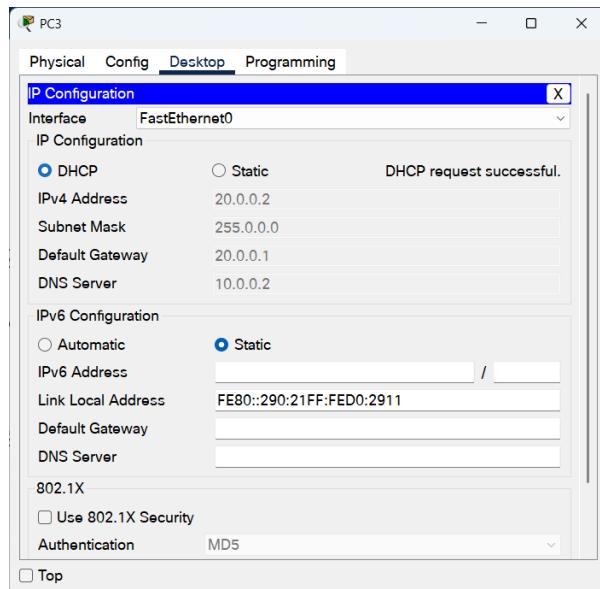
Ip helper command in Router

```

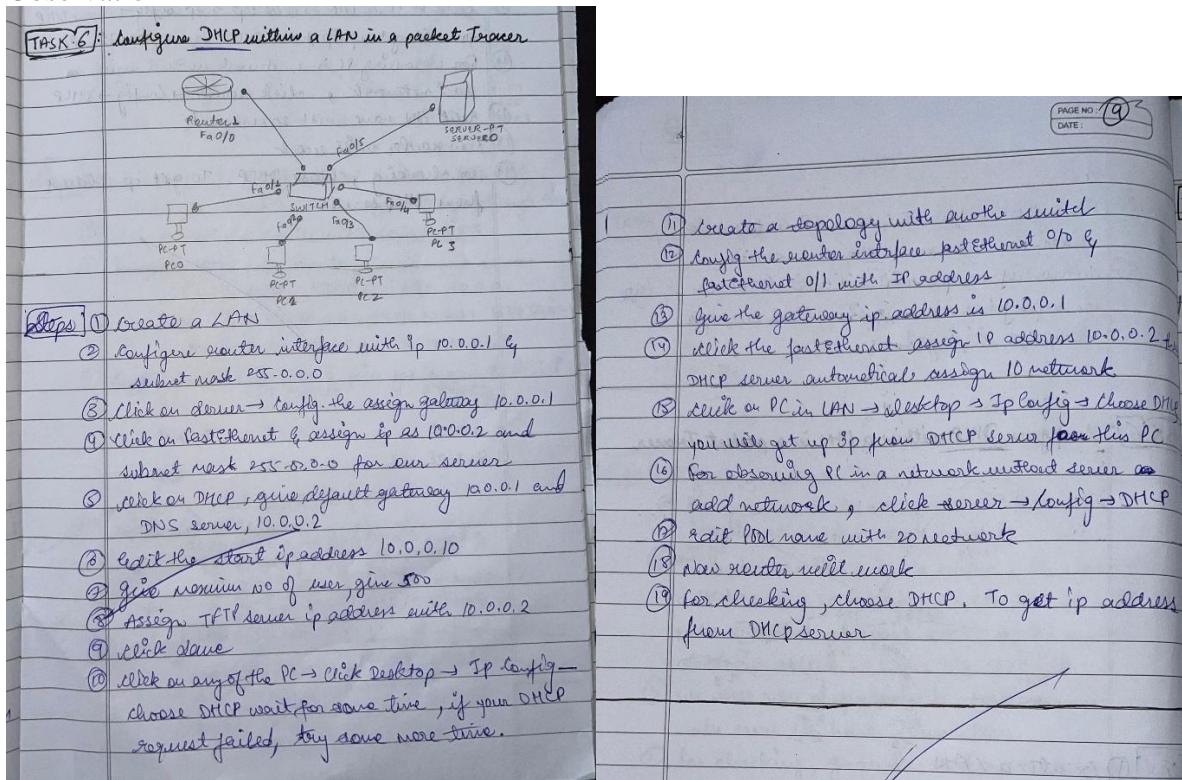
Router(config-if)#exit
Router(config)#interface Fa1/0
Router(config-if)#ip helper-address 10.0.0.2
Router(config-if)#

```

PC setting in another network

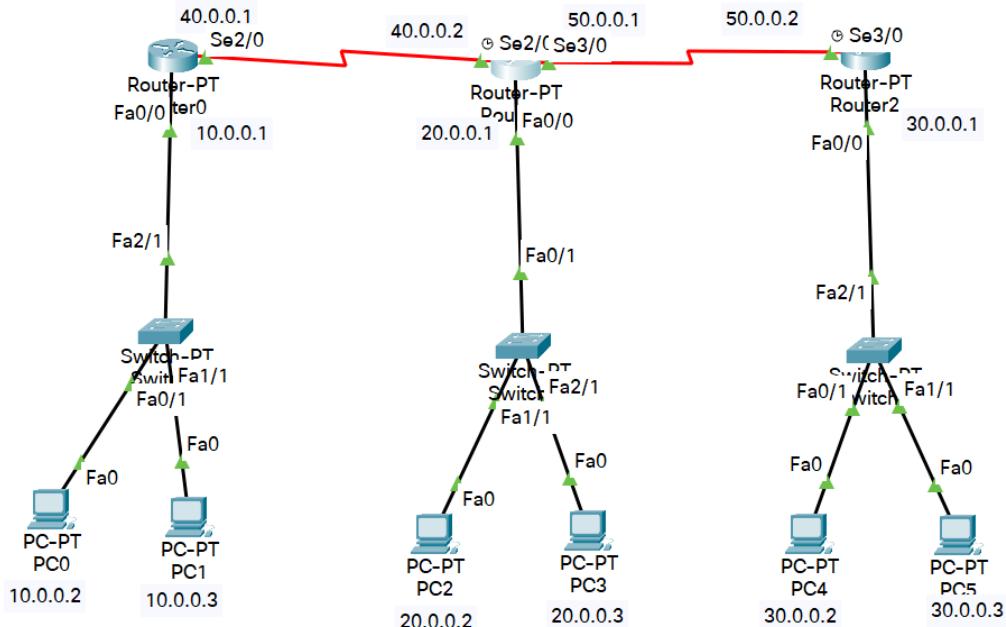


#### iv. Observation



## 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:

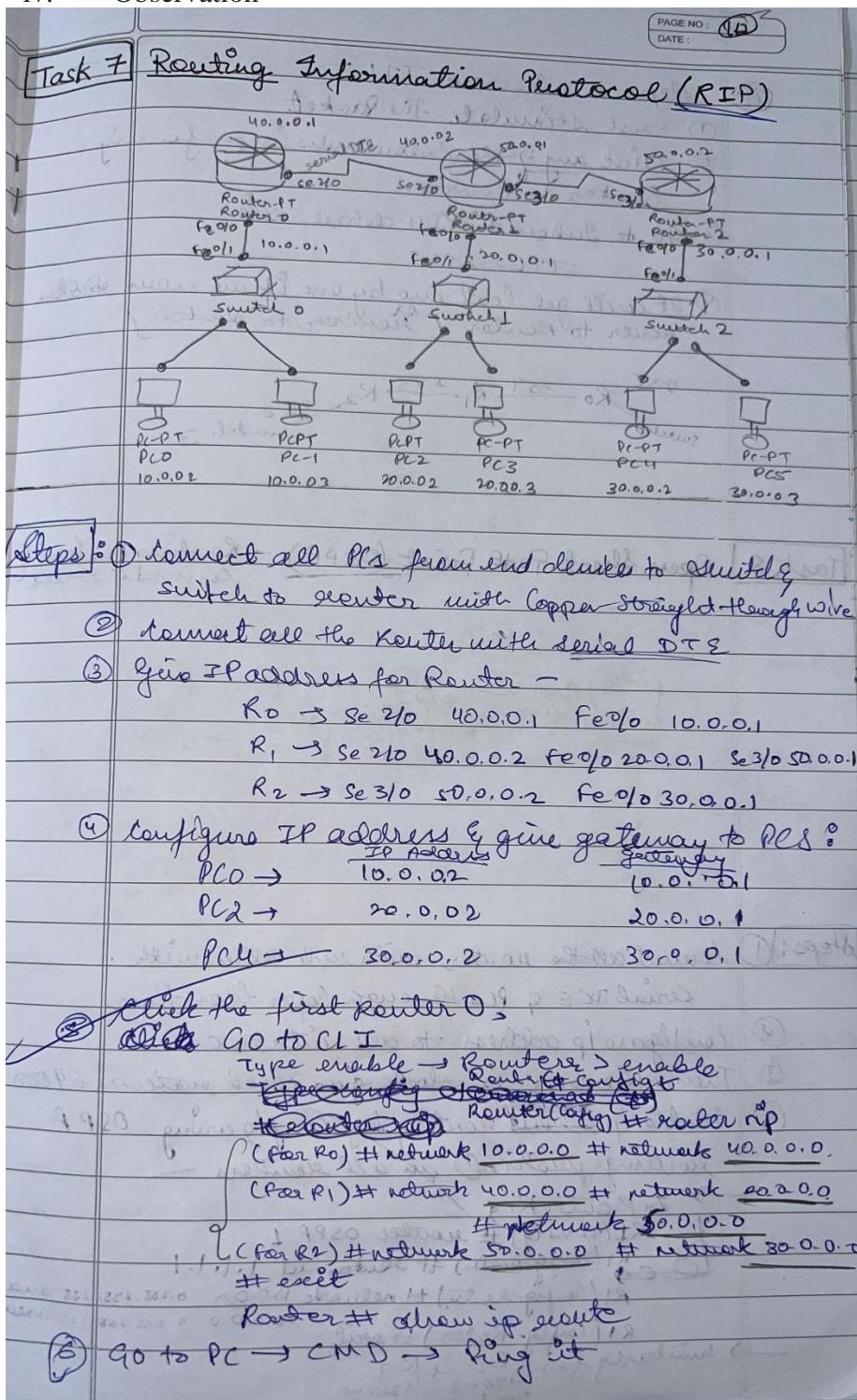
```
Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.2

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

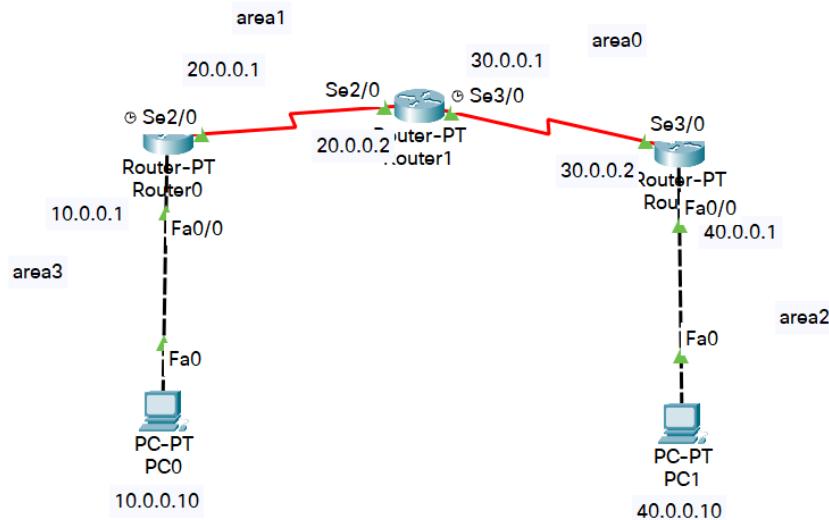
Ping statistics for 20.0.0.2:
  Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
  Minimum = 1ms, Maximum = 9ms, Average = 6ms
```

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

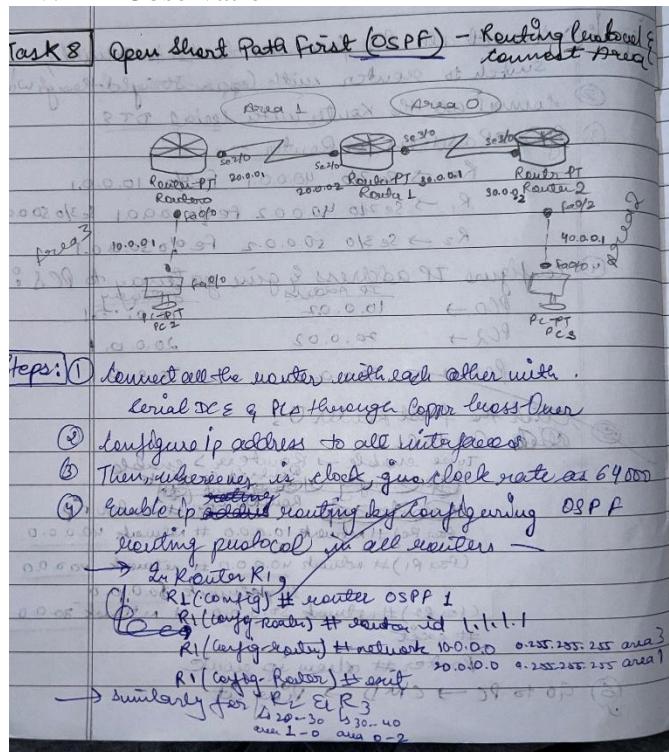
```
C:\>ping 40.0.0.10

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



PAGE NO: 12  
 DATE:

⑤ Now check Routing table of all Routers - R1, R2, R3  
 Router # show ip route

→ Here all the devices are linked together.  
 → There must be interface to keep OSPF process up.  
 to configure loop back address to routers.

R1(config-if)# interface loopback 0  
 + ip add 172.16.1.252 255.255.0.0  
 + no shutdown → 253 → R1 for  
 Similar for R2 & R3

⑥ Check Routing table for R3  
 R3# show ip route

→ Here R3 doesn't know the area 3, so we create virtual link.  
 In Router R1  
 R1(config)# router OSPF 1  
 R1(config-router)# area 0 virtual link 2.2.2.2

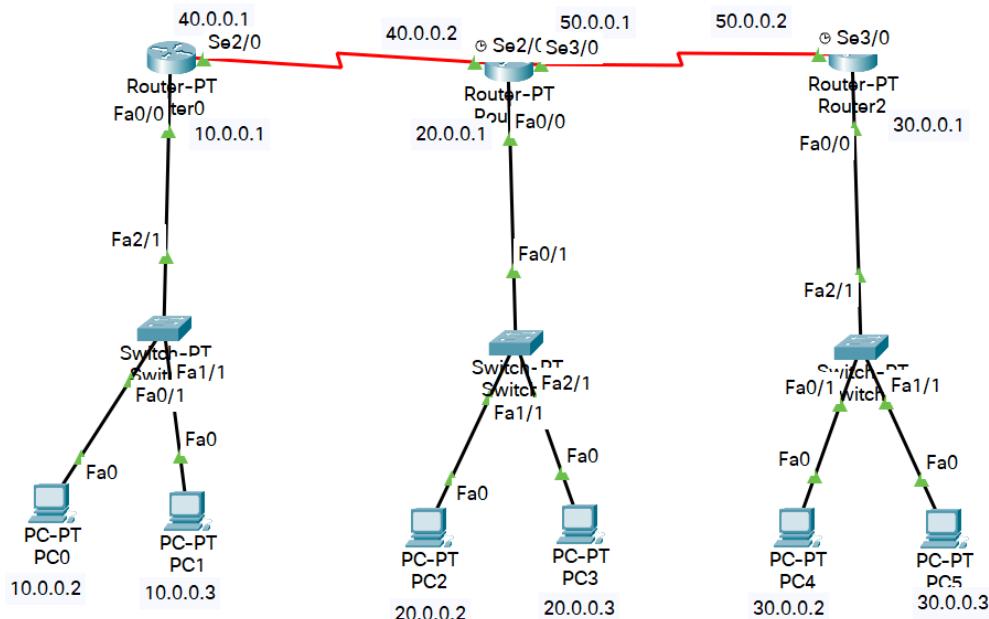
In Router R2,  
 R2(config-router)# area 0 virtual link 1.1.1.1

⑦ R2 & R3 get update without area 3, check it.

⑧ Check immediately how host 10.0.0.10 to 40.0.0.10  
 by giving ping command → ping 40.0.0.10

## 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							
0	1	4	8	16	20	24	Bits
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				
0	1	8	16	Bits
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							
0	1	4	8	16	20	24	Bits
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				
0	1	8	16	Bits
TYPE:0x08	CODE:0x00	CHECKSUM		
ID:0x0003		SEQ NUMBER:5		

## Packet at Router1

PDU Information at Device: Router1

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

PDU Information at Device: Router1

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:253	PRO:0x01	CHKSUM	
SRC IP:10.0.0.2			
DST IP:30.0.0.2			
DATA (VARIABLE LENGTH)			

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

## Packet at Router2

PDU Information at Device: Router2

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:253	PRO:0x01	CHKSUM	
SRC IP:10.0.0.2			
DST IP:30.0.0.2			
DATA (VARIABLE LENGTH)			

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

PDU Information at Device: Router2

OSI Model Inbound PDU Details Outbound PDU Details

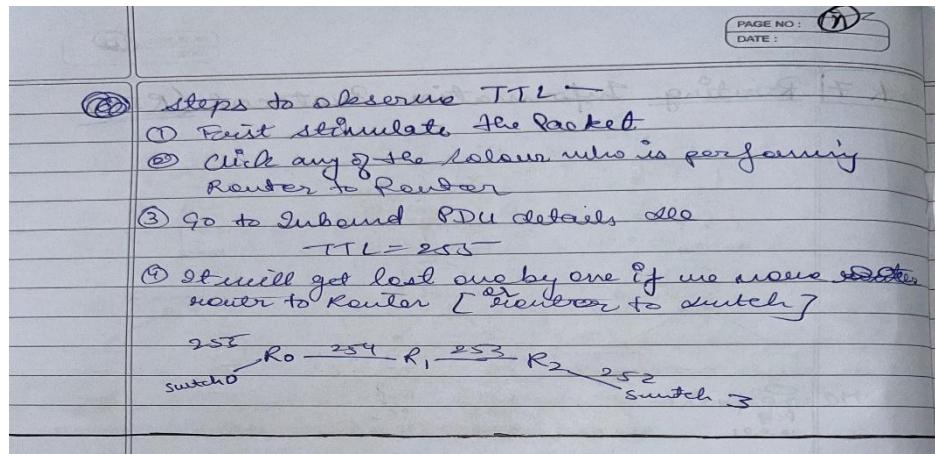
PDU Formats

IP			
VER:4	IHL:5	DSCP:0x00	TL:28
ID:0x0005		FLAGS: 0x0	FRAG OFFSET:0x000
TTL:252	PRO:0x01	CHKSUM	
SRC IP:10.0.0.2			
DST IP:30.0.0.2			
DATA (VARIABLE LENGTH)			

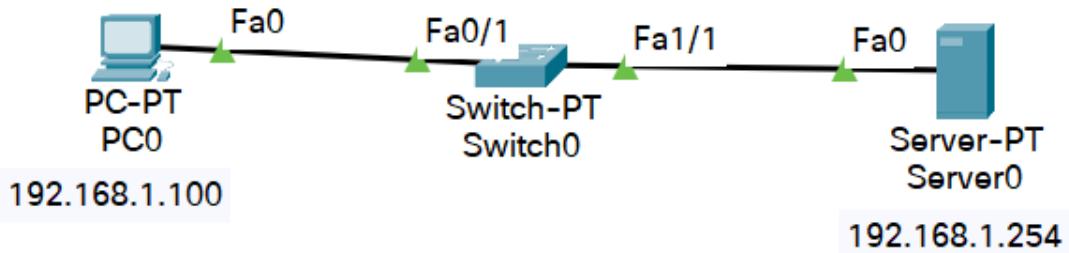
ICMP			
TYPE:0x08	CODE:0x00	CHECKSUM	
ID:0x0003		SEQ NUMBER:5	

### iv. Observation



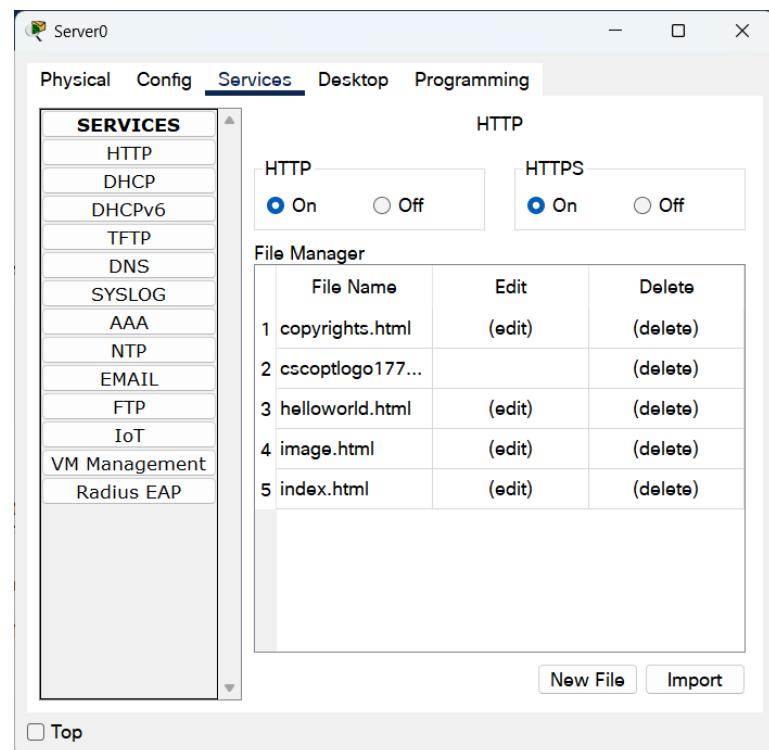
## Program 8

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

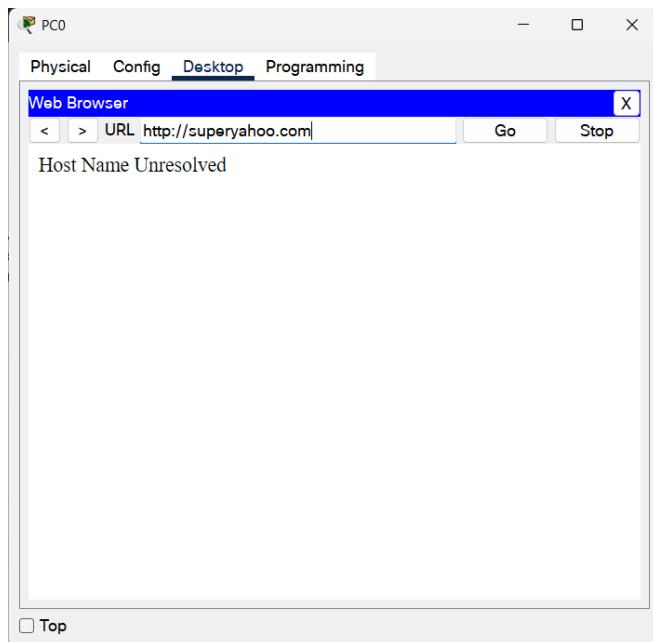
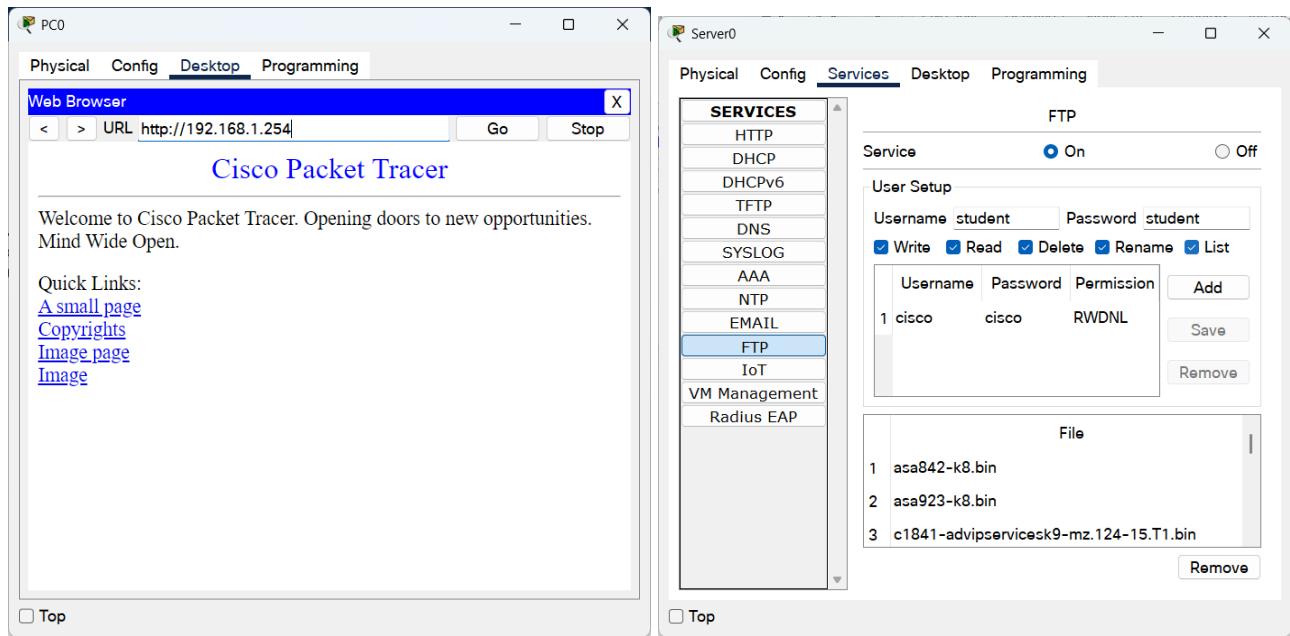


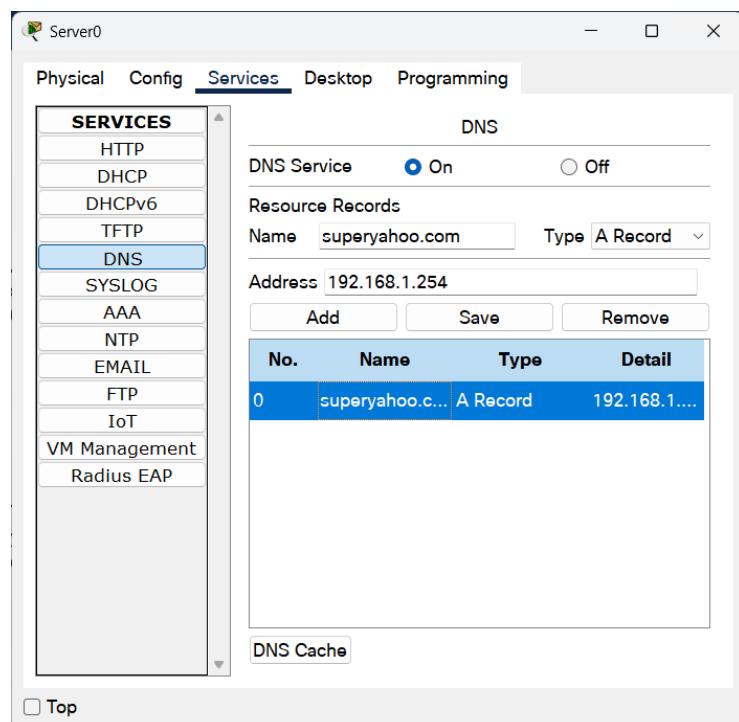
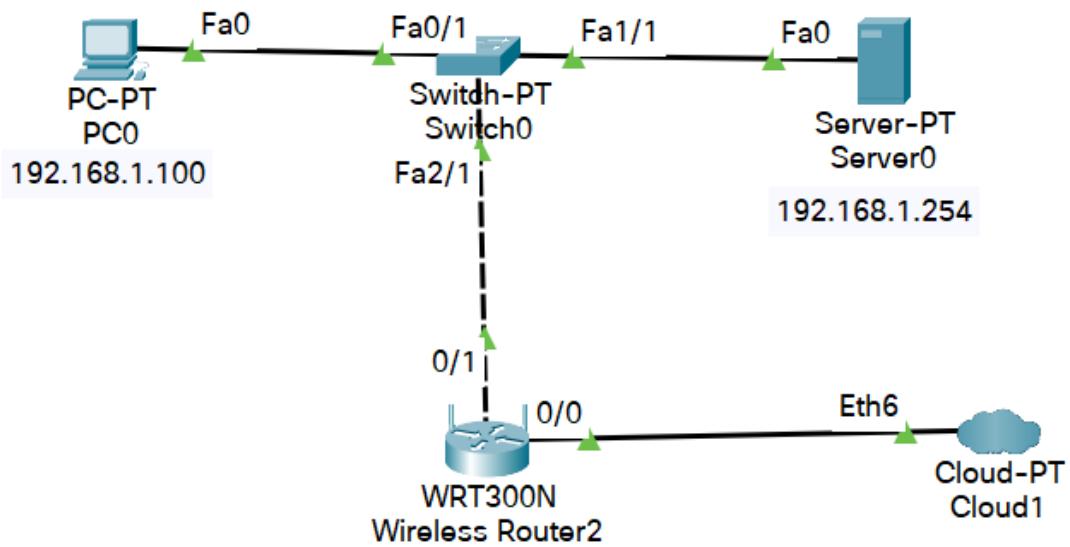
- iii. Screen shots/ output

Server's services

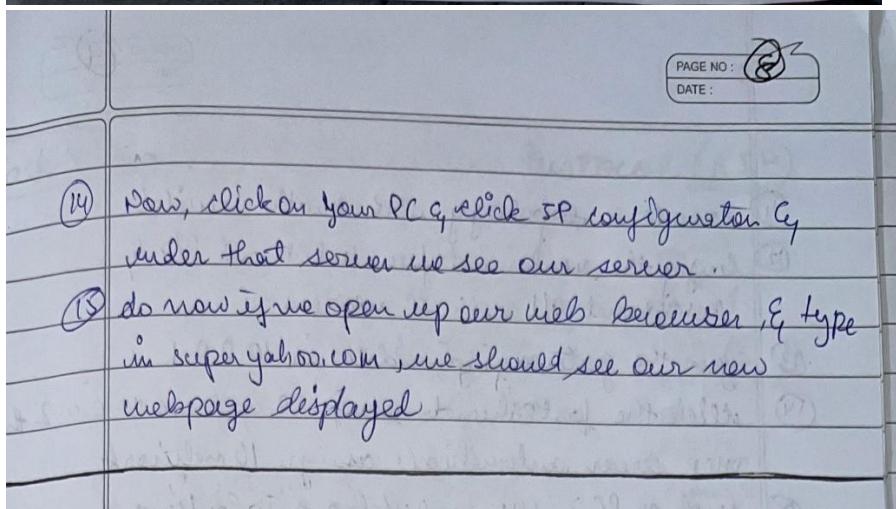
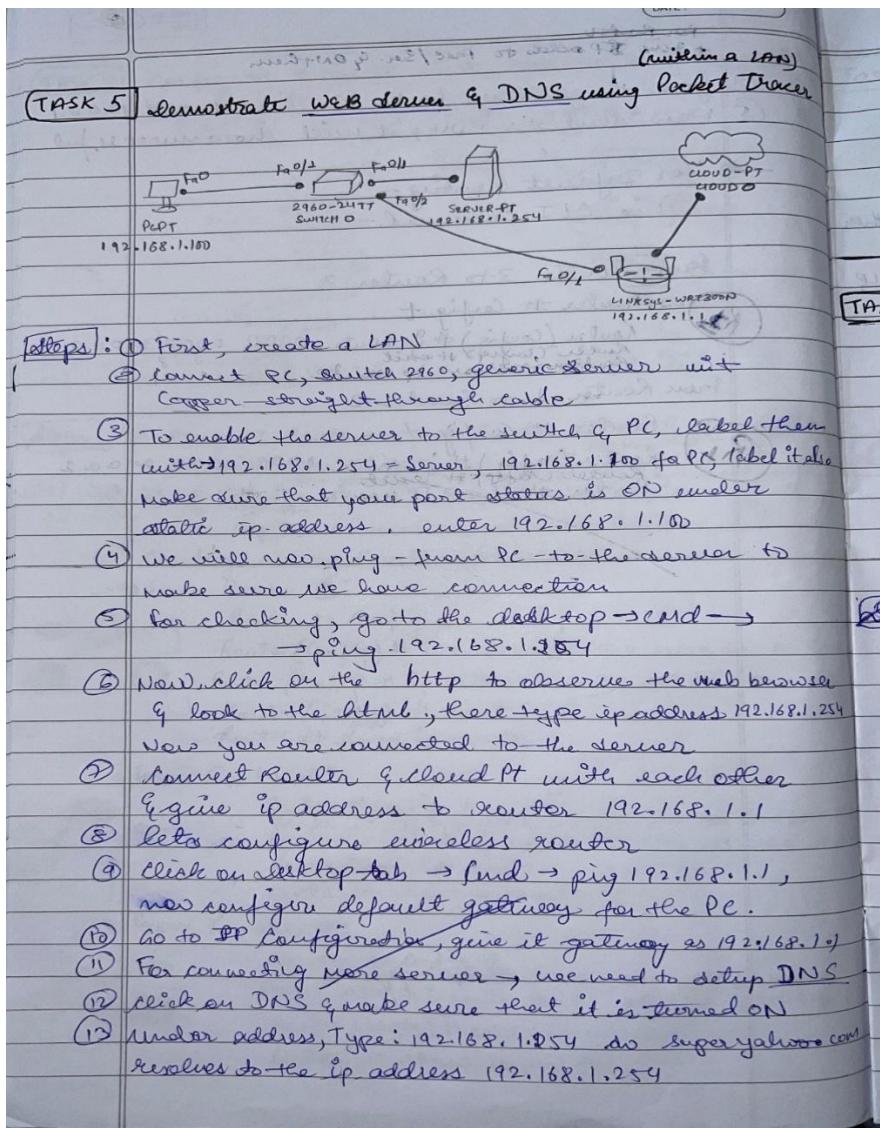


PC's Web Browser



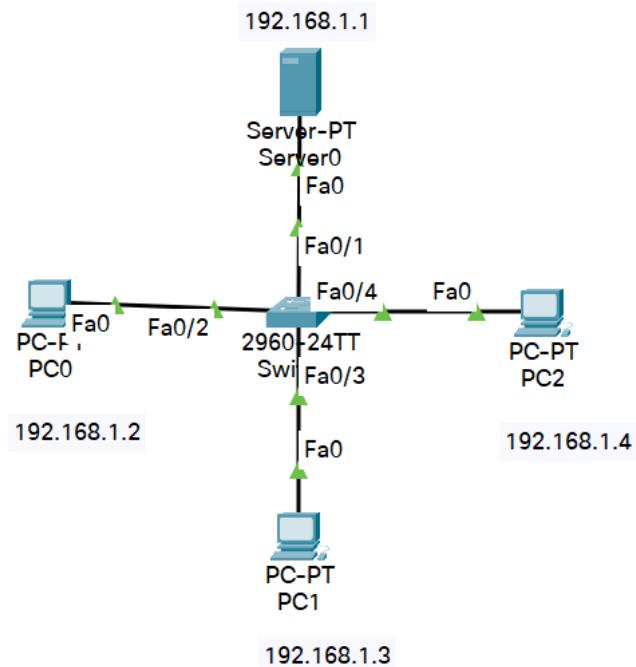


#### iv. Observation



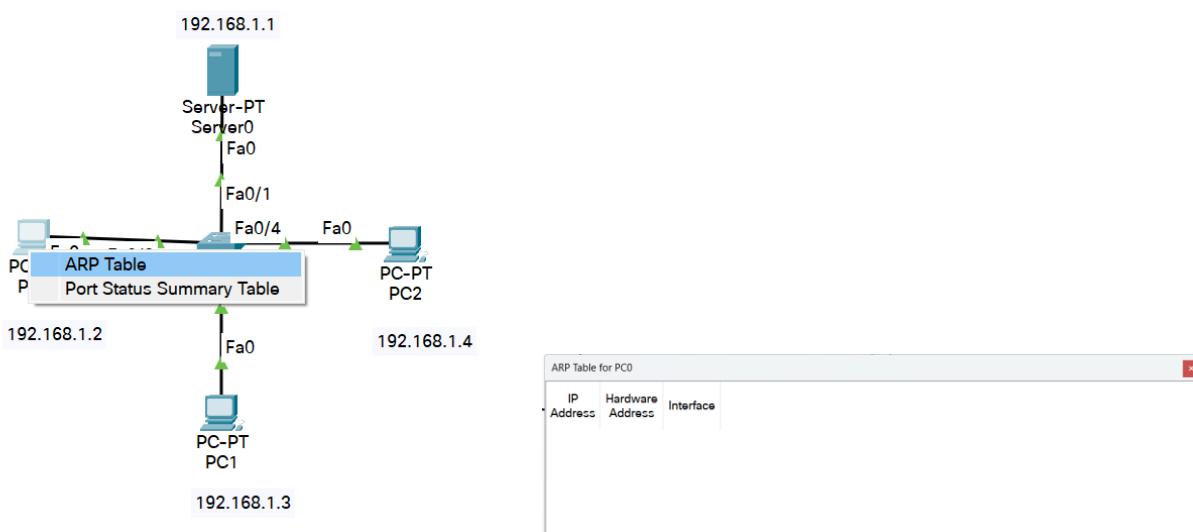
## Program 9

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

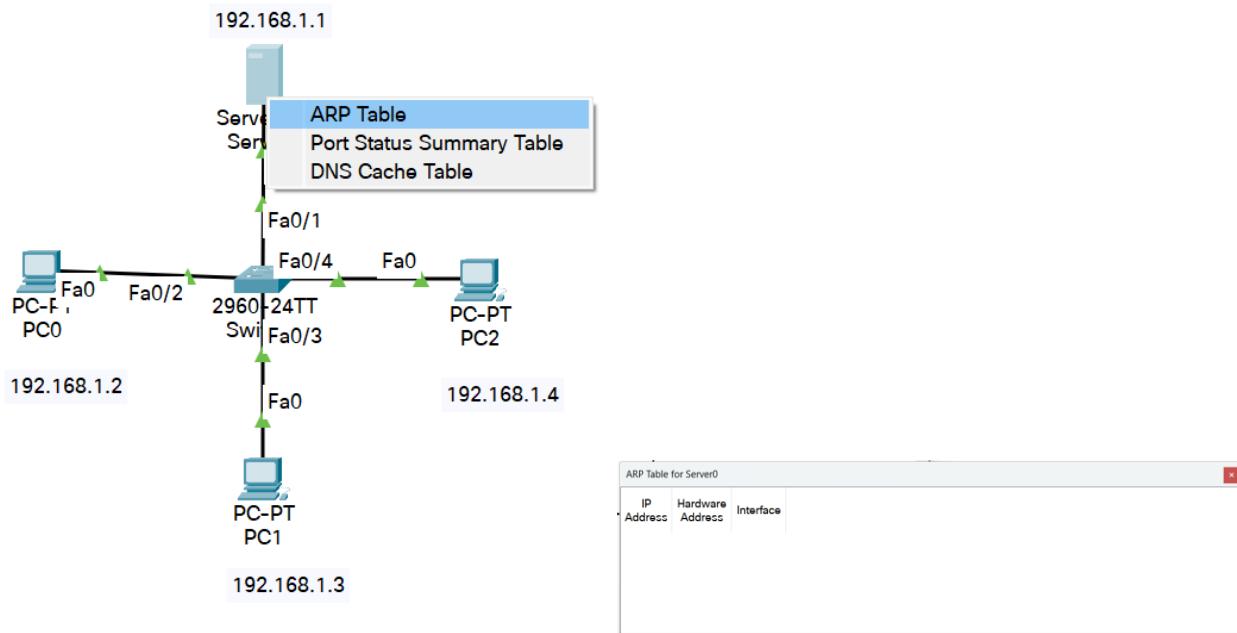


- iii. Screen shots/ output

ARP Table of PC



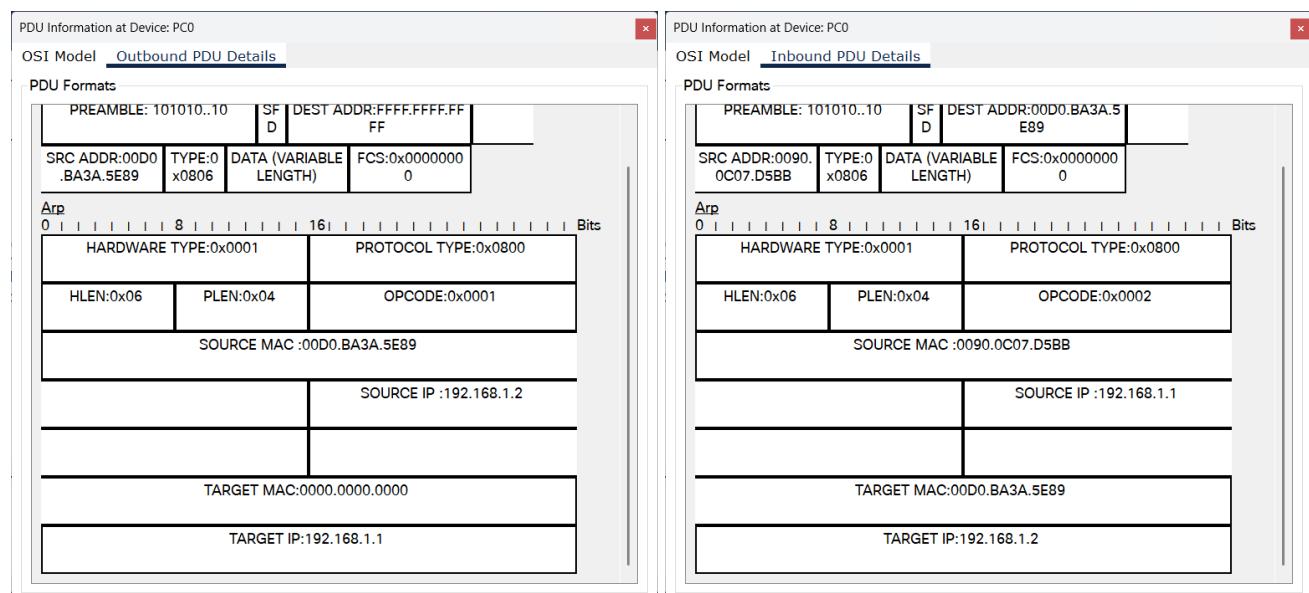
ARP Table of Server



### Command at PC

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

### Pinging in Simulation Mode

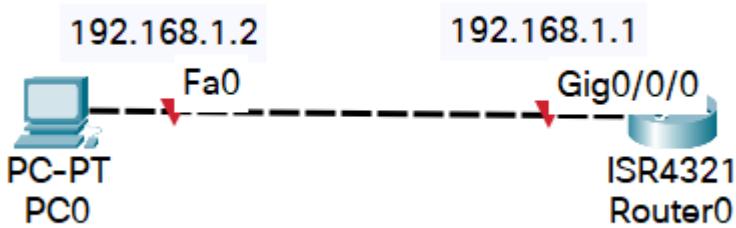


iv. Observation

<b>TASK 9</b>	<b>Simple LAN &amp; (ARP) Address Resolution Protocol</b>
<p><u>Steps:</u></p> <ol style="list-style-type: none"> <li>① drag Switch from switch panel, Server &amp; PT from tool bar &amp; 2 PCs from endless library. Connect all with switch using copper straight through wire</li> <li>② Assign IP Address to all PCs, server with 192.168.11.1/24</li> <li>③ Go to EtherealNet Panel, click on Inspect (Q) &amp; right click on PC1 (ie Search Option)</li> <li>④ After clicking on PC1, → Click on ARP Table &amp; Notice that entries are there or not in ARP Table (Empty) • Repeat same for server</li> <li>⑤ Click on PC1 → go to Cmd Prompt, type Type arp -a {Initially there are no ARP entries}</li> <li>⑥ Try Pinging from PC1 to Server → give command → ping 192.168.11.4</li> <li>⑦ Send package from PC1 to Server; Notice 2 packets → ICMP = created</li> <li>⑧ Click on ARP packet &amp; then click on Capture Section to start the stimulation</li> <li>⑨ Ping request from PC1 to switch will happen</li> <li>⑩ Click on capture to see the ICMP packet movement</li> <li>⑪ After doing everything, click on inspect option &amp; click to PC1 to check ARP Table</li> </ol> <p>→ Now you can see that there is entries present.</p>	

## **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

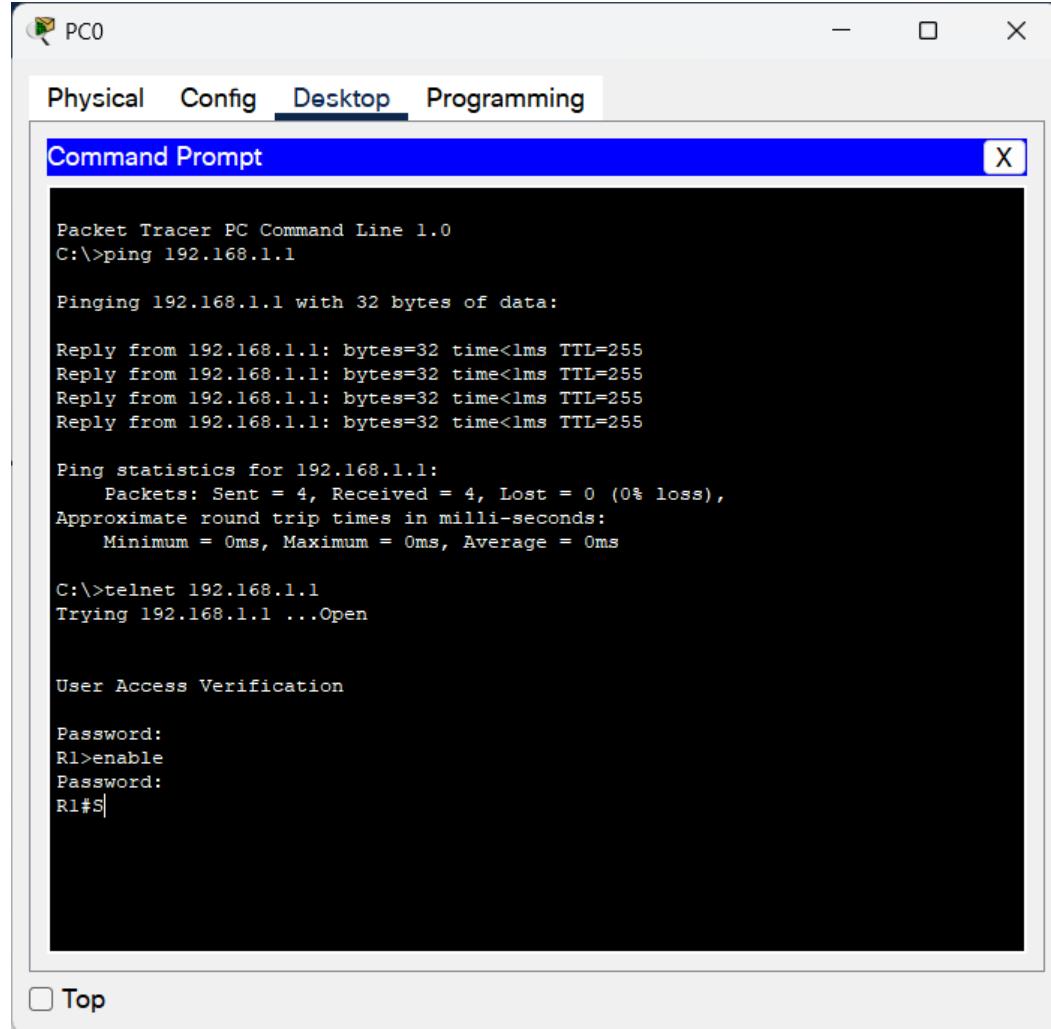
R1(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to up

%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

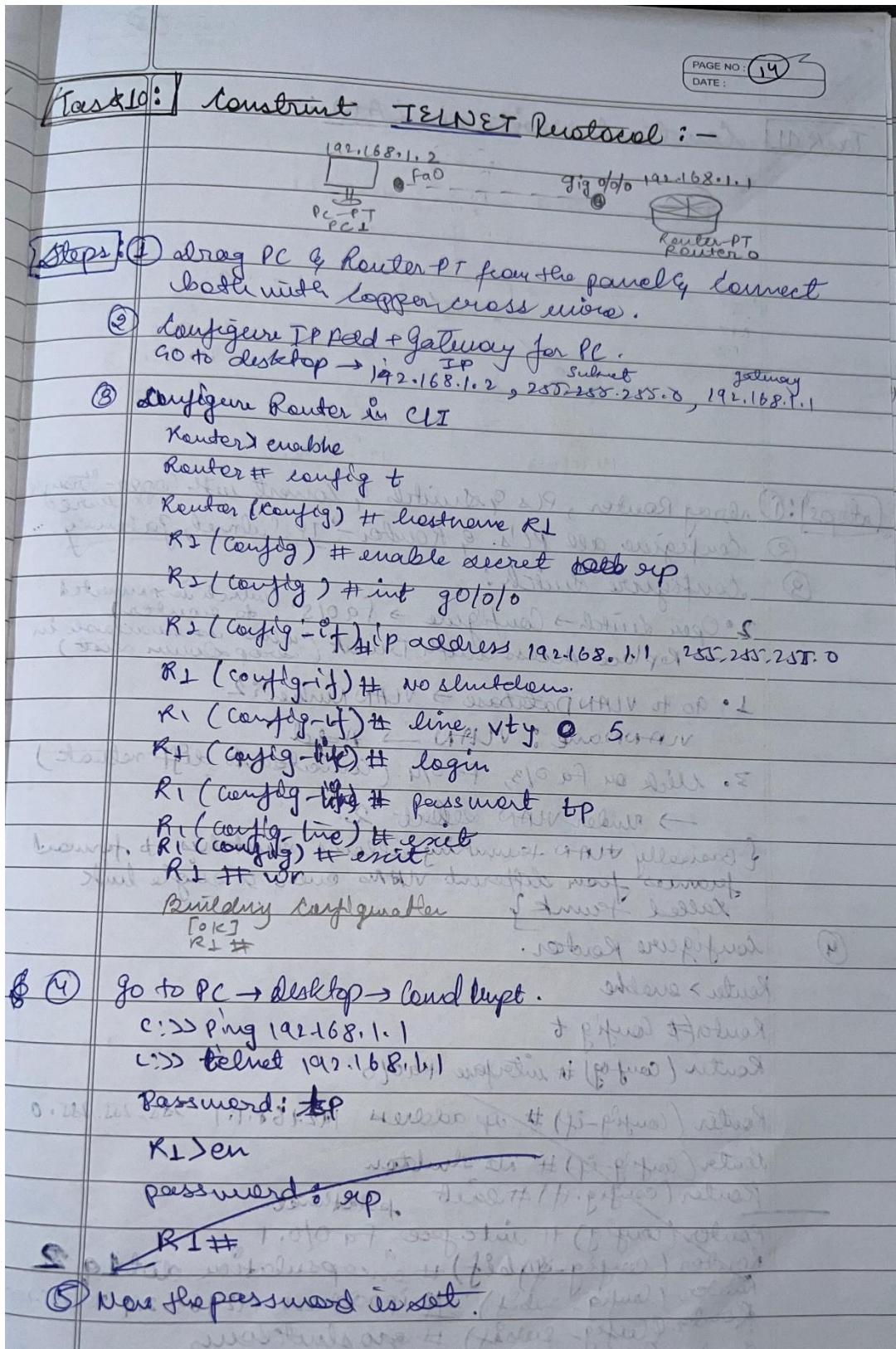
R1#wr
Building configuration...
[OK]
R1#|
```

PC



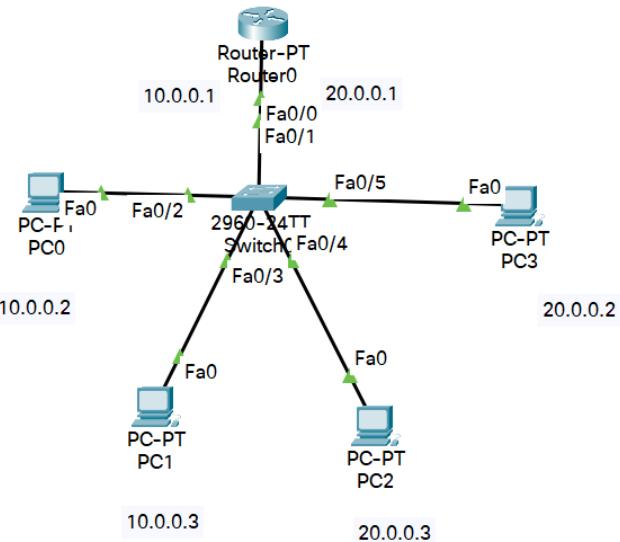
Top

iv. Observation



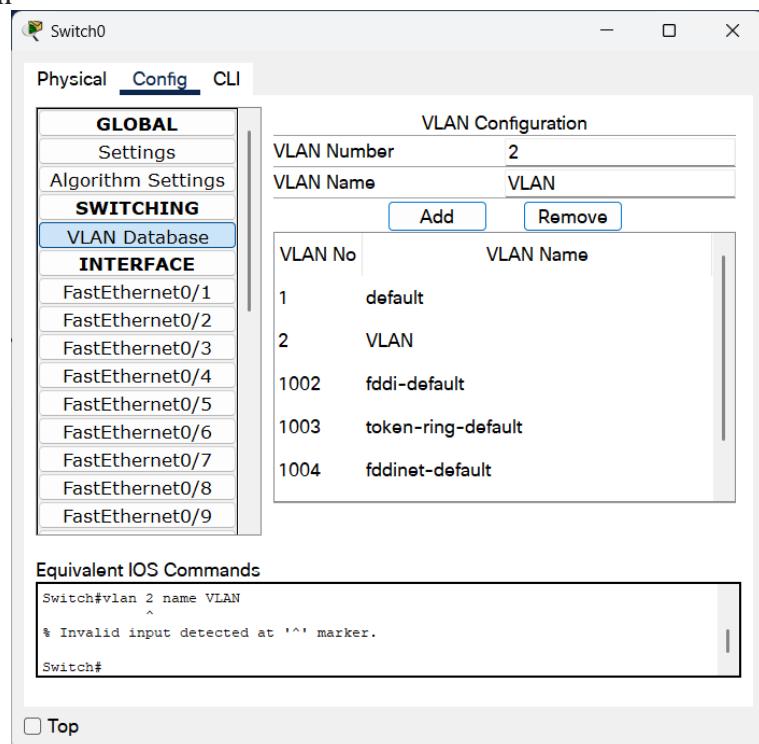
## Program 11

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

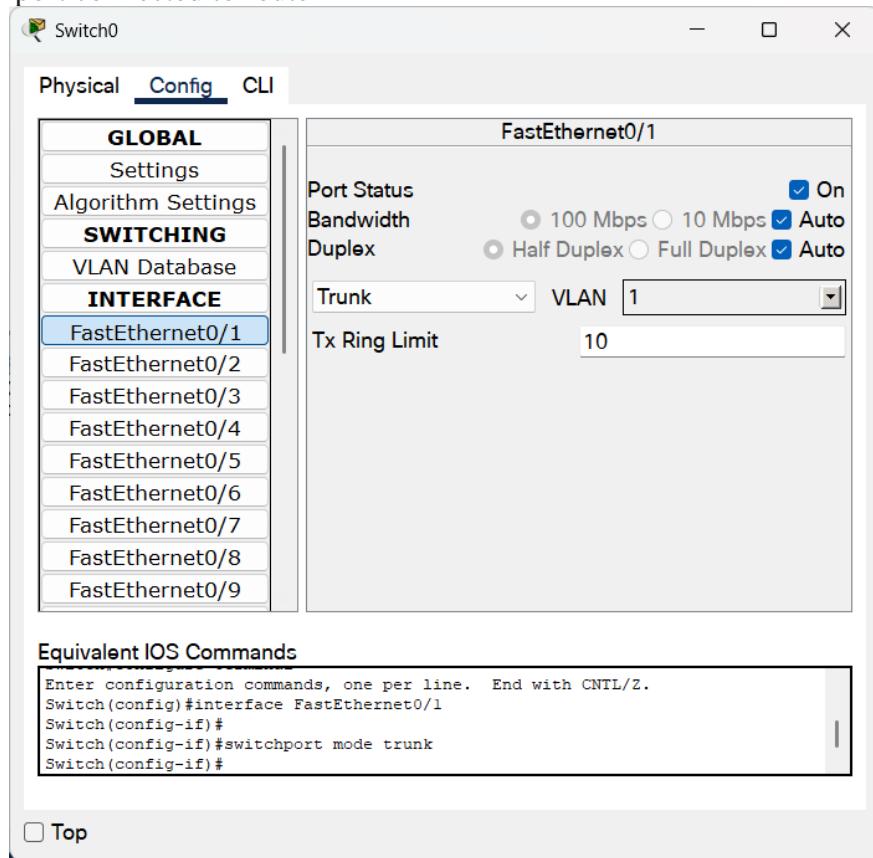


- iii. Screen shots/ output

Switch Configuration



## Configuration of port connected to router



## Configuration of ports connected to other networks

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

The left panel shows the configuration for "FastEthernet0/4". The "Port Status" is "On" (checked). Under "Bandwidth", "100 Mbps" is selected. Under "Duplex", "Auto" is selected. The "Access" dropdown is set to "VLAN", and the "VLAN" dropdown is set to "2". The "Tx Ring Limit" is set to "10".

The right panel shows the configuration for "FastEthernet0/5". The "Port Status" is "On" (checked). Under "Bandwidth", "100 Mbps" is selected. Under "Duplex", "Auto" is selected. The "Access" dropdown is set to "VLAN", and the "VLAN" dropdown is set to "2". The "Tx Ring Limit" is set to "10".

Both panels have an "Equivalent IOS Commands" section at the bottom:

Left panel:

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

```

Right panel:

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

```

At the bottom left of each panel is a "Top" button.

## 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

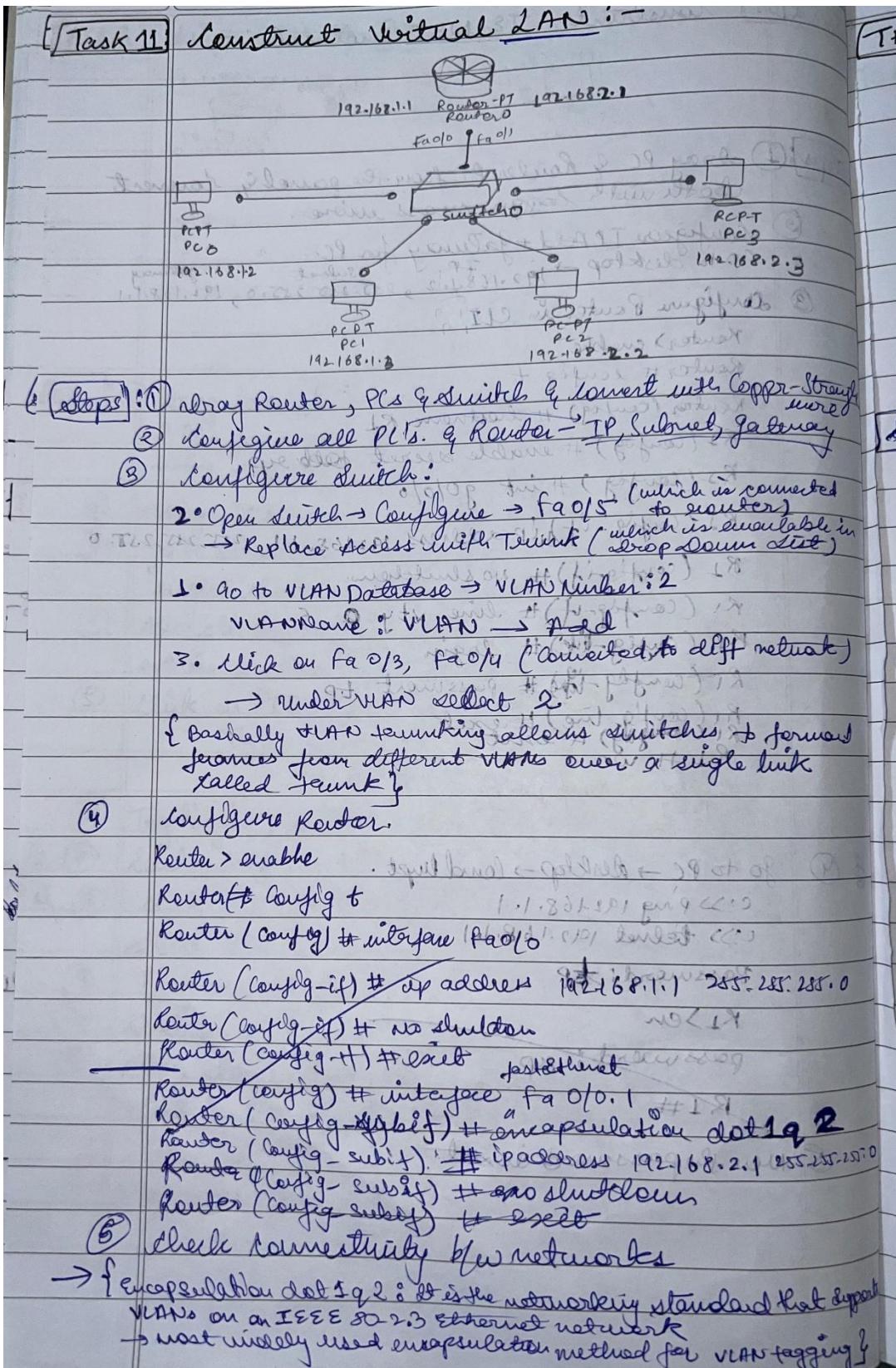
Router(config-subif)#encapsulation dot1q 2
Router(config-subif)#ip address 20.0.0.1 255.0.0.0
Router(config-subif)#no shutdown
Router(config-subif)#exit
Router(config)#

Ctrl+F6 to exit CLI focus
```

Top

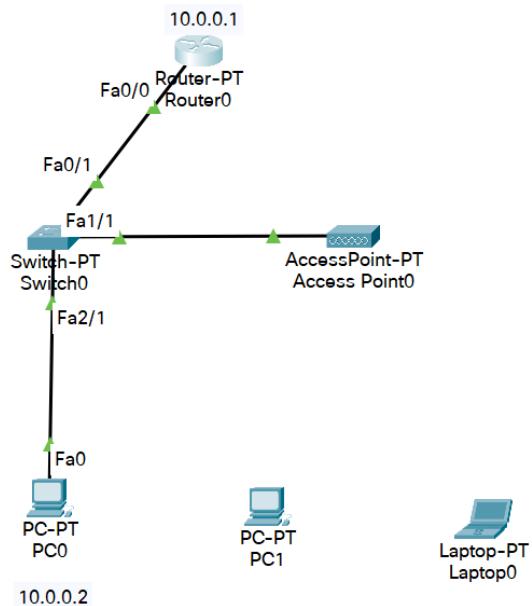
**Copy** **Paste**

iv. Observation

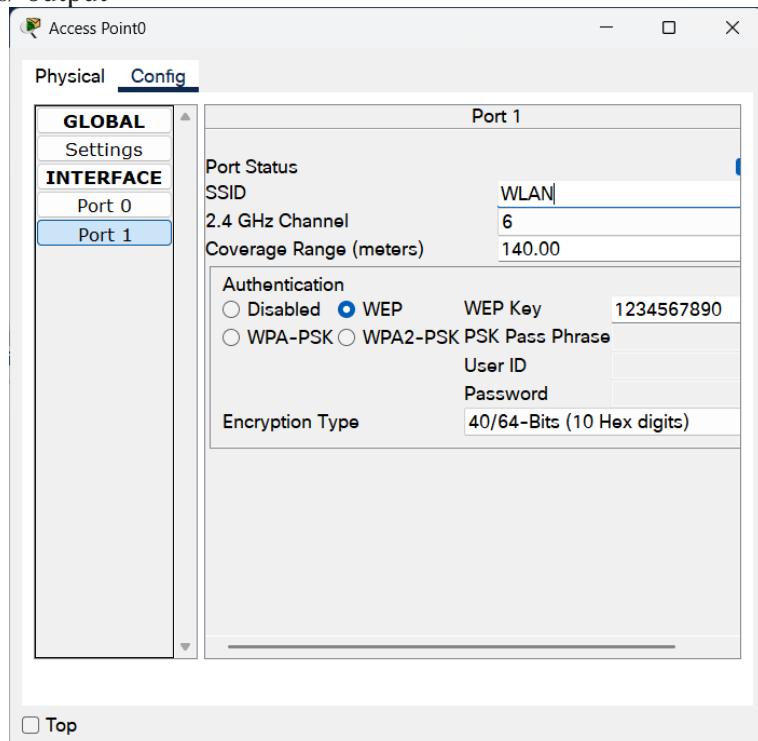


## Program 12

- 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

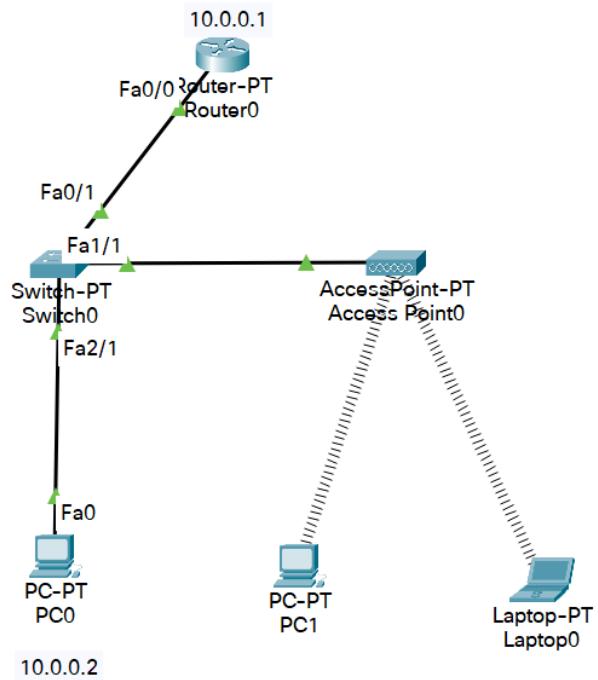
**Laptop0**

- 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 for **Laptop0**:

```

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

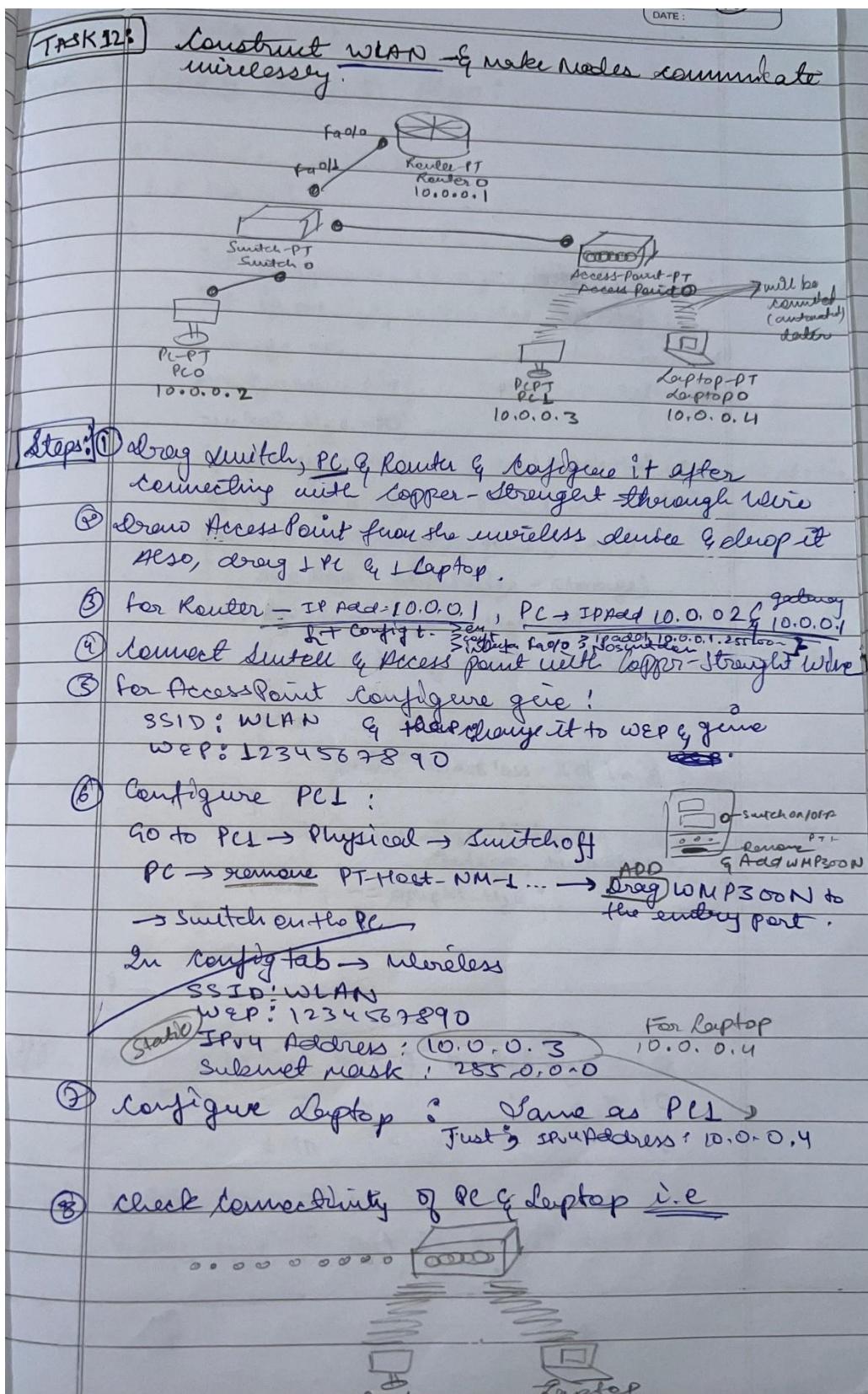
Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 24ms, Maximum = 40ms, Average = 28ms

C:\>

```

Top

iv. Observation



## 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): 1111111111111111100000  
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

**TASK 2:**

Error detection using CRC-CCITT (16 Bits)

```

#include <stdio.h>
#include <string.h>

int rec(char *ip, char *op, char *poly, int node)
{
    strcpy(op, ip); // copy ip to op
    if (node)
    {
        // append 0s to op
        for (int i=1; i<strlen(poly); i++)
        {
            strcat(op, "0");
        }
        // perform XOR
        for (int i=0; i<strlen(ip); i++)
        {
            if (op[i] == '1')
            {
                for (int j=0; j<strlen(poly); j++)
                {
                    if (ip[i+j] == '1')
                    {
                        op[i+j] = '0';
                    }
                    else
                    {
                        op[i+j] = '1';
                    }
                }
            }
            else
            {
                for (int j=0; j<strlen(poly); j++)
                {
                    if (ip[i+j] == '1')
                    {
                        op[i+j] = '1';
                    }
                    else
                    {
                        op[i+j] = '0';
                    }
                }
            }
        }
        return 1; // no error
    }
}

int main()
{
    char ip[50], op[50], recv[50];
    char poly[] = "10001000000100001";
    printf("Enter IP msg in binary: ");
    scanf("%s", ip);
}

```

PAGE NO. 32  
DATE:

// take in CRC & get transmission msg  
rec(ip, op, poly, 1);  
printf("The transmitted msg = %s\n", op);  
printf("Enter received msg in binary: ");  
scanf("%s", recv);  
// check for errors in received msg  
if (rec(recv, op, poly, 0))
{
 printf("No error in data\n");
 else
 {
 printf("Error in data transmission has occurred");
 }
}

C/F Enter IP msg in binary: 1111111111111111  
the transmission msg = 000001110001111011110  
Enter the received msg in binary: 1111111111111111  
error in data transmission has occurred.

② Enter IP msg in binary = 1111111111111111  
the transmission msg = 000001110001111011110  
Enter the received msg in binary: 1111111111111111  
No error in data.

## **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

\* Requirements:

Task 1: Leaky Bucket Algo:

```
#include <stdio.h>
int main()
{
    int no_of_queries, storage, ip_pkt_size;
    int ip_pkt_size, bucket_size, size_left;
    storage = 0; // initial packets in bucket
    no_of_queries = 4; // no of time bucket is checked
    bucket_size = 10;
    ip_pkt_size = 4; // no of packets that enter in bkt at a time
    op_pkt_size = 10 - ip_pkt_size; // exit ...
    for(int i=0; i<no_of_queries; i++)
    {
        if(i < size_left) bucket_size - storage;
        else if(ip_pkt_size <= size_left)
        {
            storage += ip_pkt_size; // update storage
        }
        else
        {
            printf("Packet loss = %d \n", ip_pkt_size);
            storage = op_pkt_size;
        }
    }
    return 0;
}
```

O/P

~~Buffer size = 4 out of bucket size = 10  
 " " = 7 " " = 10  
 " " = 10 out of bucket size = 10  
 Packet loss = 4  
 Buffer size = 9 out of bucket size = 10~~

### **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

**TASK 3 : socket Programming :-**

```

i (tcp.py) :->
    i (file name)
    i (port number)
    = i (year, "2016") name

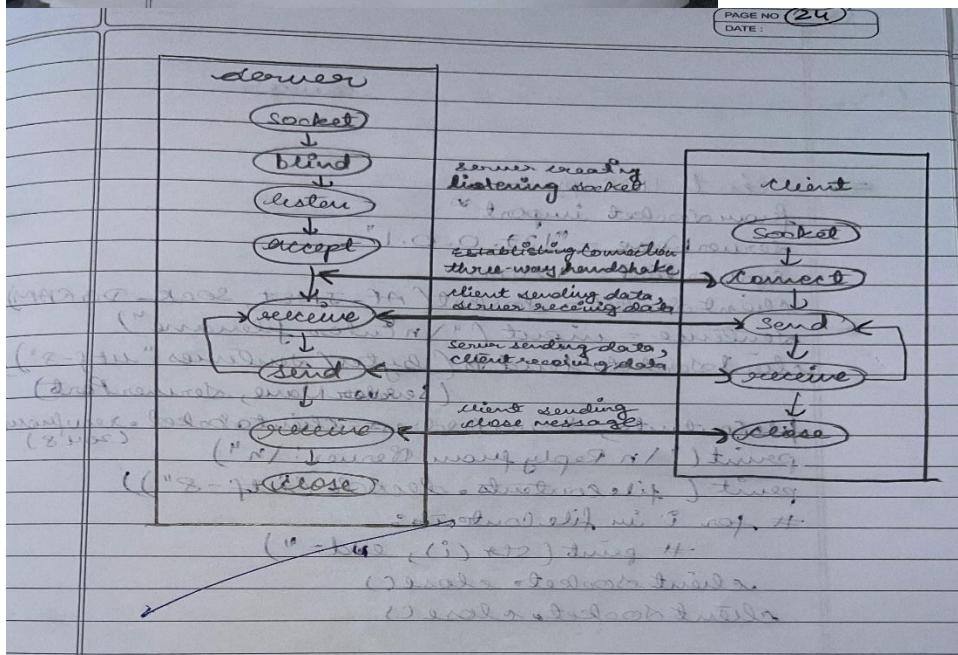
→ Client TCP.py -> question :-?
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = input("Enter file name:")
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print("From Server: " + sentence)
print(filecontents)
clientSocket.close()
    
```

→ Server TCP.py -> code in part q/ also

```

from socket import *
serverName = '127.0.0.1'
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(5)
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("Sent contents of " + sentence)
    file.close()
    connectionSocket.close()
    
```

PAGE NO (24)  
DATE :



## **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)
```

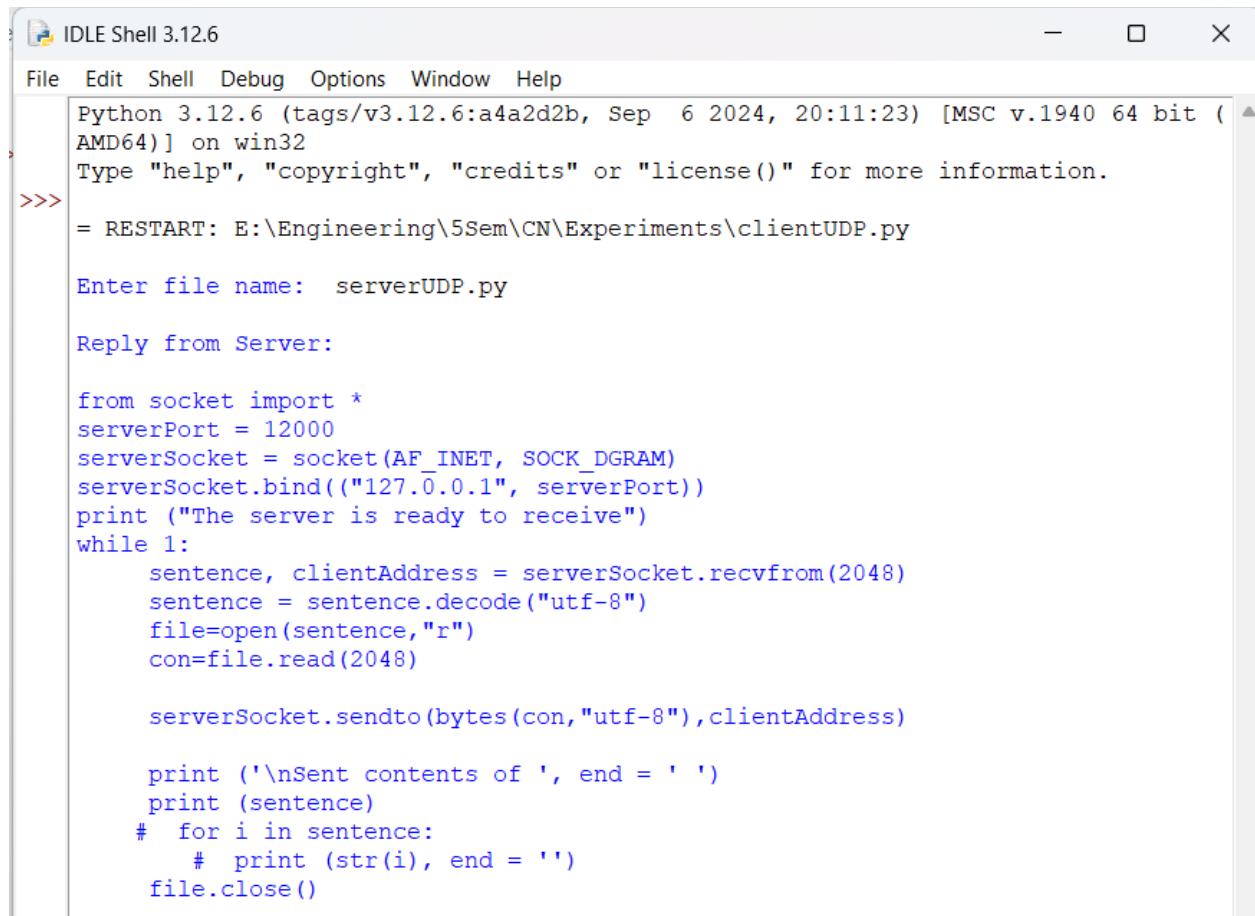
```
serverSocket.sendto(bytes(con,"utf-8"),clientAddress)
```

```
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

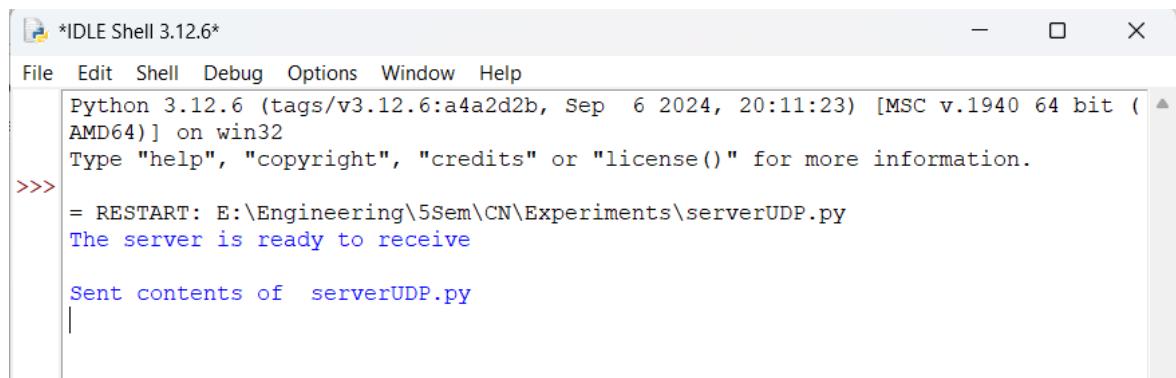
Reply from Server:

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

⇒ Client UDP.py —

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("\nEnter filename")
clientSocket.sendto(sentence.encode("utf-8"), (serverName, serverPort))
fileContents, clientAddress = clientSocket.recvfrom(2048)
print("\nReply from Server:\n", fileContents.decode("utf-8"))
# for i in fileContents:
#     print(str(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(2048)
    sentence = sentence.decode("utf-8")
    file = open(sentence, "r")
    ren = file.read(2048)
    serverSocket.sendto(ren.encode("utf-8"), clientAddress)
```

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

W  
23/12