

# **VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

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



## **LAB RECORD**

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

*Submitted by*

**Shruti Khandelia (1BM22CS274)**

*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 **Shruti Khandelia (1BM22CS274)**, 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.

Prof. 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	6
3	25/10/2024	Configure default route, static route to the router	11
4	08/11/2024	Configure DHCP within a LAN and outside LAN.	17
5	22/11/2024	Configure RIP routing Protocol in Routers	21
6	22/11/2024	Configure OSPF routing protocol	25
7	22/11/2024	Demonstrate the TTL/ Life of a Packet	31
8	08/11/2024	Configure Web Server, DNS within a LAN.	33
9	20/12/2024	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)	37
10	20/12/2024	To understand the operation of TELNET by accessing the router in server room from a PC in IT office.	40
11	20/12/2024	To construct a VLAN and make the PC's communicate among a VLAN	43
12	20/12/2024	To construct a WLAN and make the nodes communicate wirelessly	47

Github Link:

<https://github.com/shrutikhandelia/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).	51
2	15/11/2024	Write a program for congestion control using Leaky bucket algorithm	53
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.	55
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.	58

## I N D E X

NAME: Shruti Khandelwal

BRUNNEN  
STD. 1 8

6

18M228934

STD...... 9

1

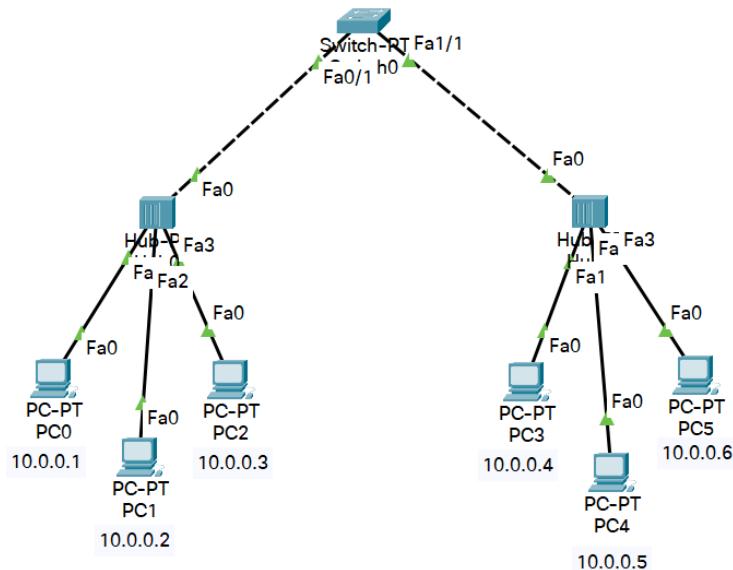
ROLL NO. \_\_\_\_\_ SUB. \_\_\_\_\_

(N.L.B)

# Cycle-I

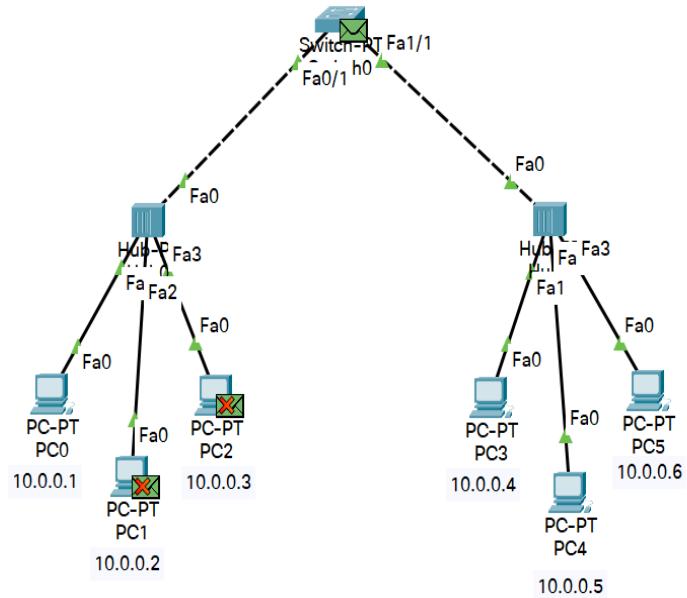
## Program 1

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

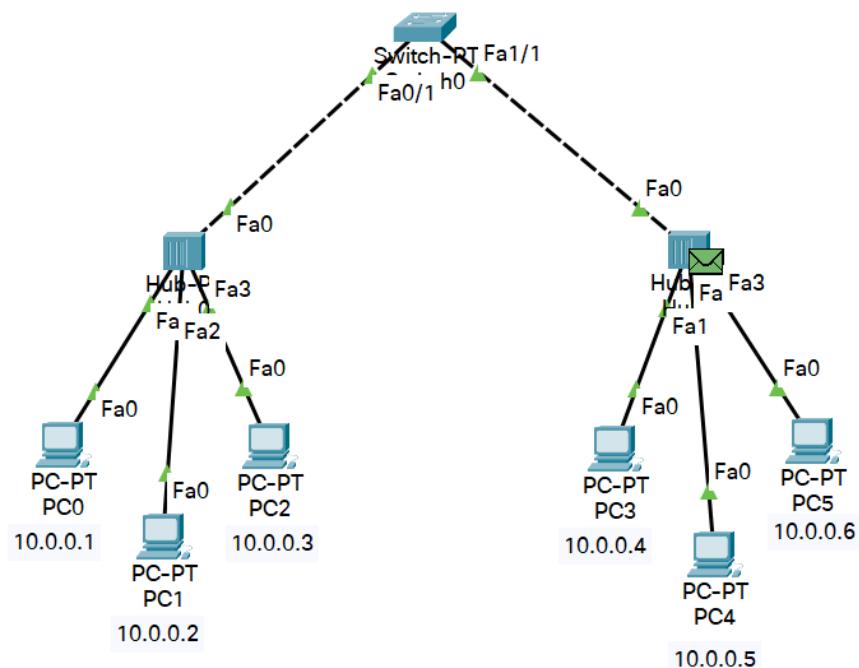


- iii. Screen shots/ output

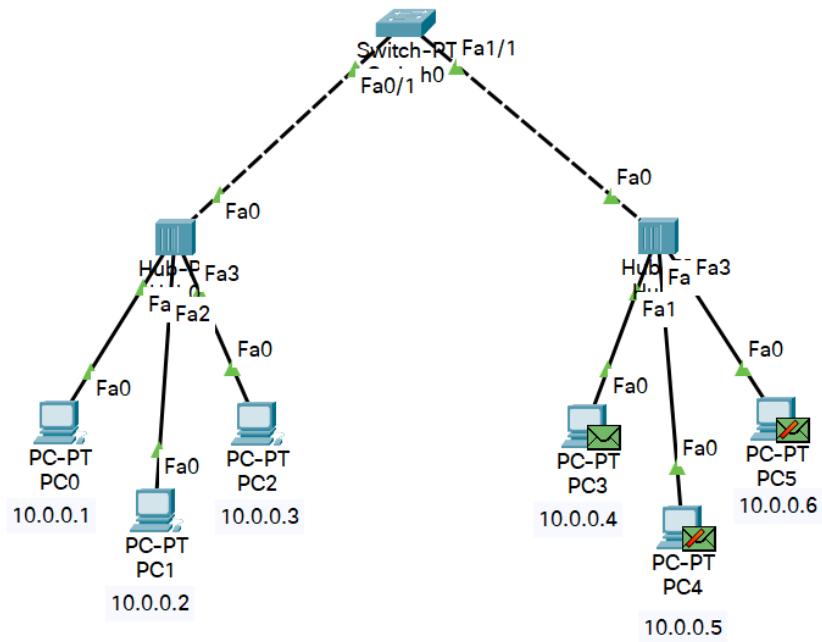
Hub behaviour at sending end



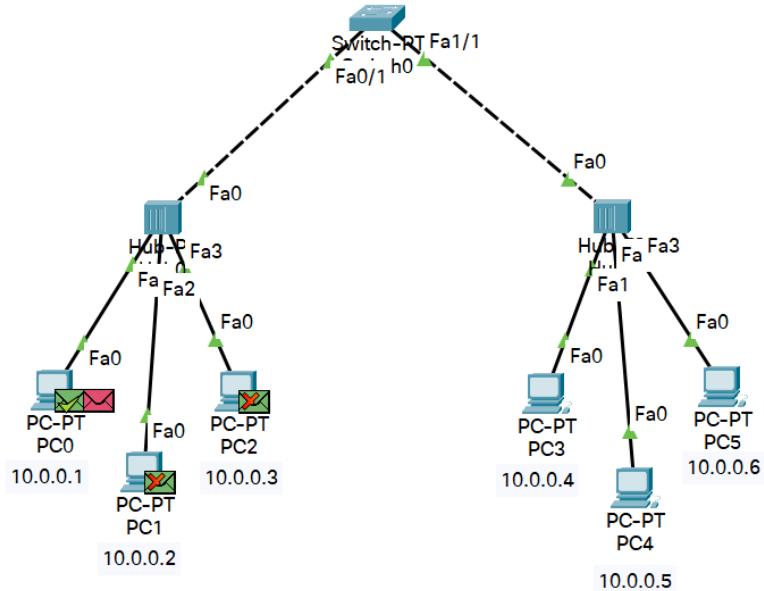
Switch behaviour



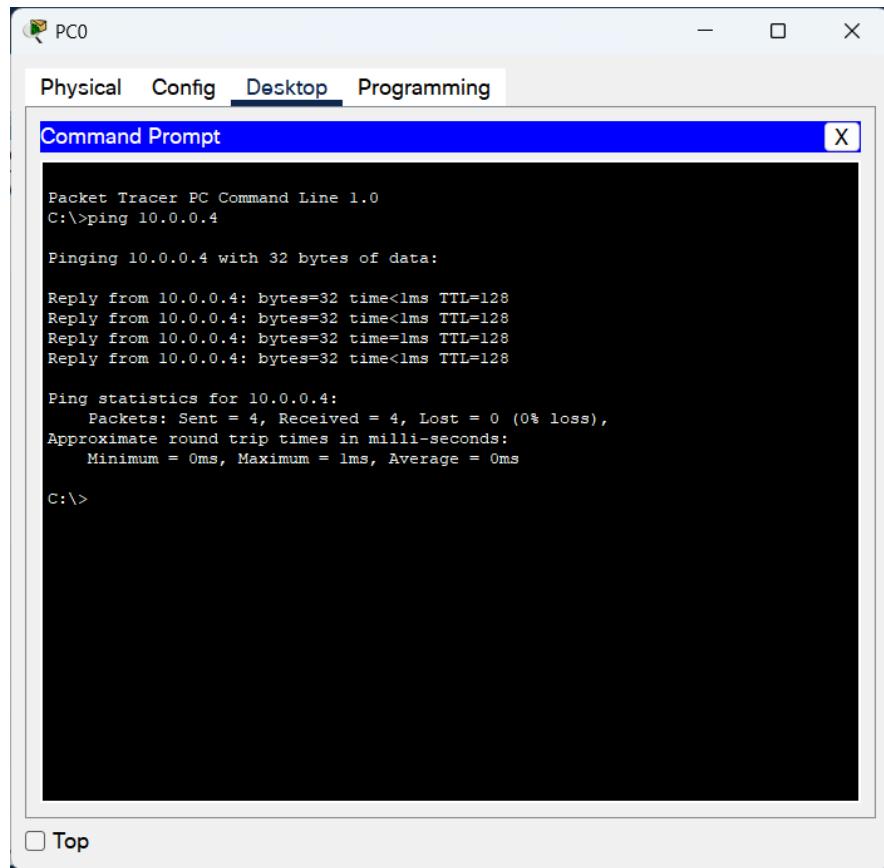
Hub behaviour at receiving end



## Hub behaviour when back to sender



## Ping command to connectivity



The screenshot shows a software interface titled "PC0" with a tab bar at the top. The "Desktop" tab is selected. Below it is a "Command Prompt" window with a blue header bar containing the title and a close button. The main area of the window displays the output of a "ping" command. The text reads:

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

Pinging 10.0.0.4 with 32 bytes of data:

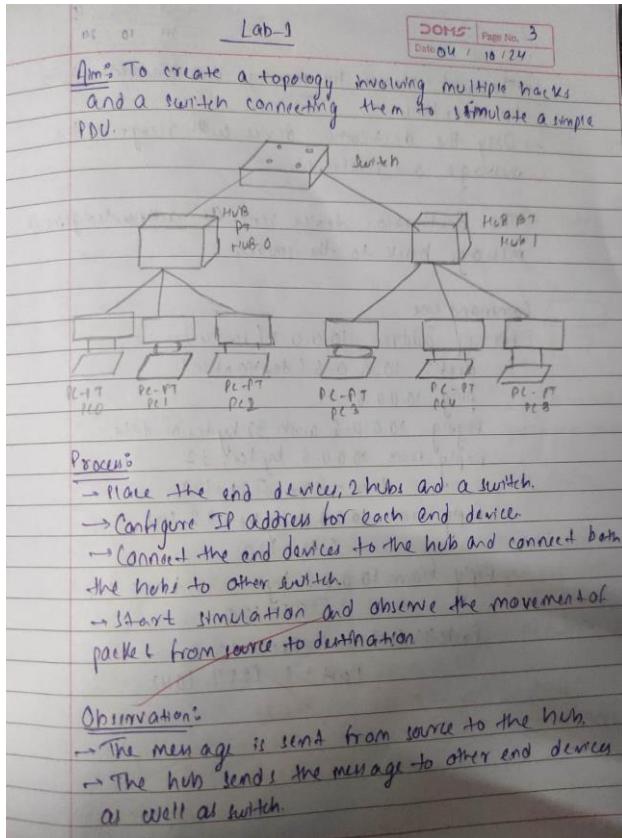
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time=<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time=<1ms TTL=128

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:\>
```

At the bottom left of the Command Prompt window, there is a checkbox labeled "Top".

#### iv. Observation



**DOMS** Page No. 4  
Date 04 / 10 / 24

- The switch receives the message and forwards it to the other hub.
- Only the destination device will recognize the message and receive it.
- The destination device sends an acknowledgement message back to the source.

Command line

for ip address 10.0.0.1 (source)  
dest: 10.0.0.6 (destination)  
ping 10.0.0.6

Pinging 10.0.0.6 with 32 bytes of data.  
Reply from 10.0.0.6 by test=32  
time = 9ms TTL=120

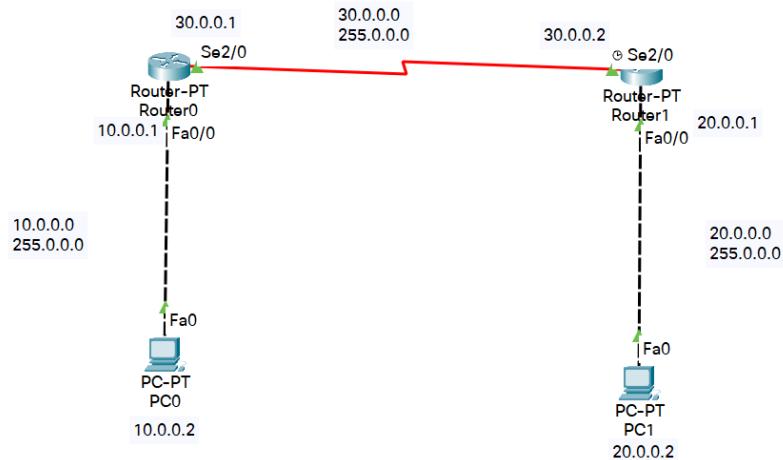
Reply from 10.0.0.6 by test=32 time = 9ms  
TTL=120

Reply from 10.0.0.6 by test=32 time = 9ms  
TTL=120

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

## 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

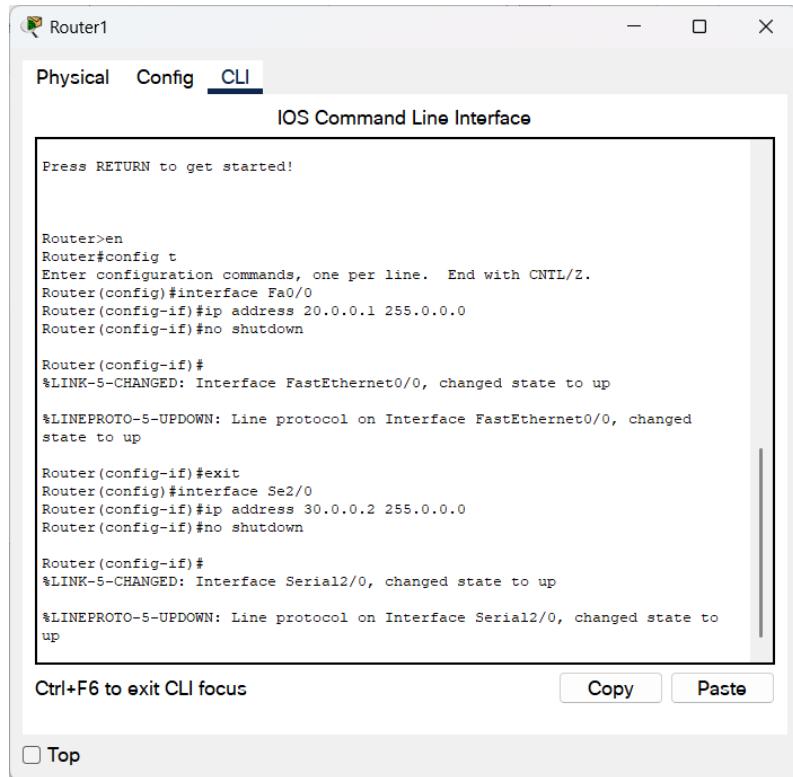
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed
state to up

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 window titled "Router1" with a tab bar containing "Physical", "Config", and "CLI". The "CLI" tab is selected, displaying the "IOS Command Line Interface". A message at the top says "Press RETURN to get started!". Below it, configuration commands are shown:

```
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", and 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)#

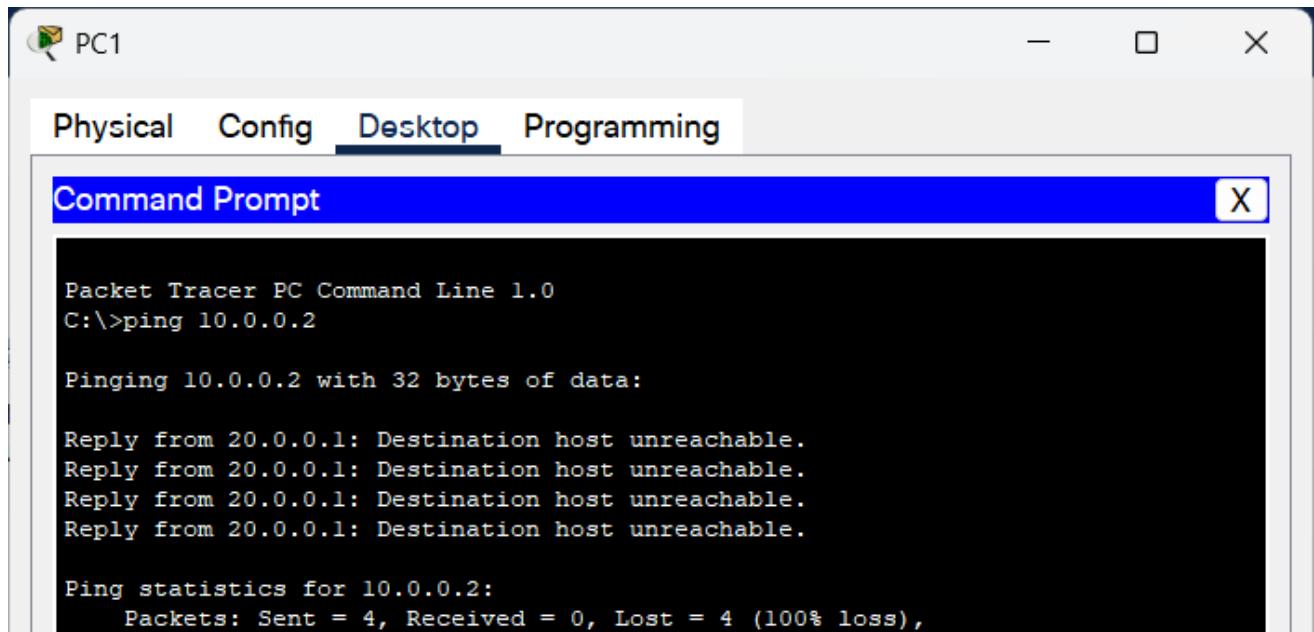
```

## 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

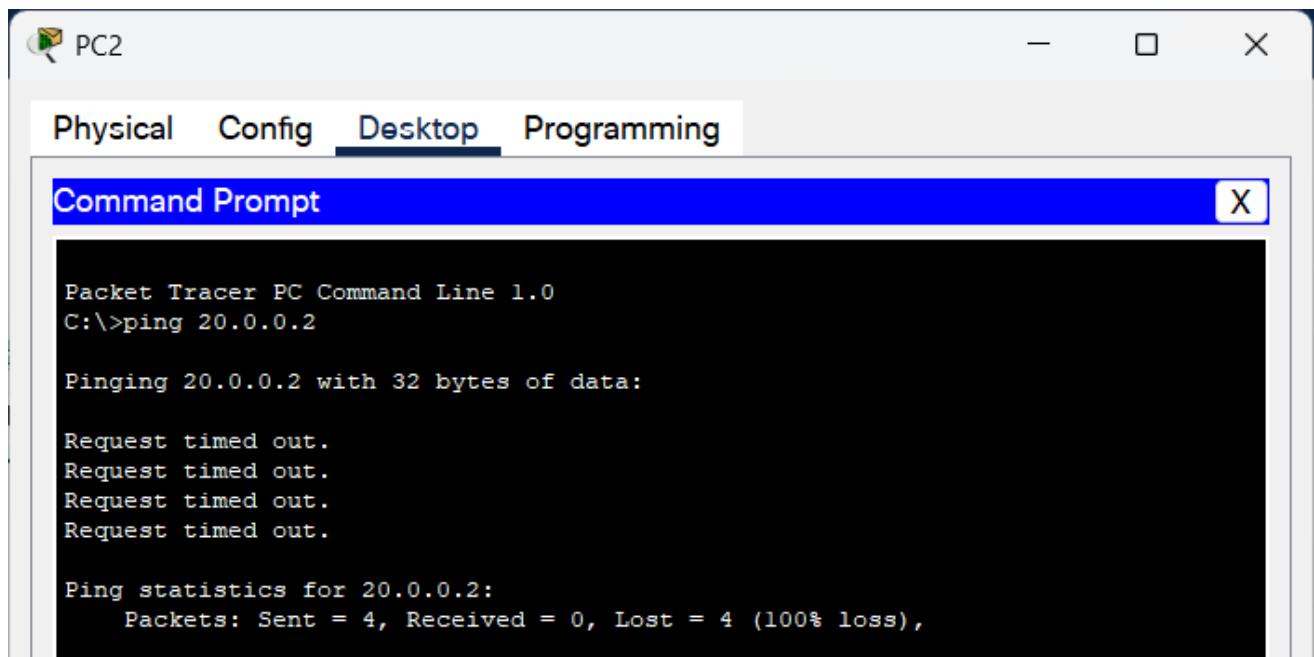
```
Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.2

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

PC

Physical Config Desktop Programming

**Command Prompt**

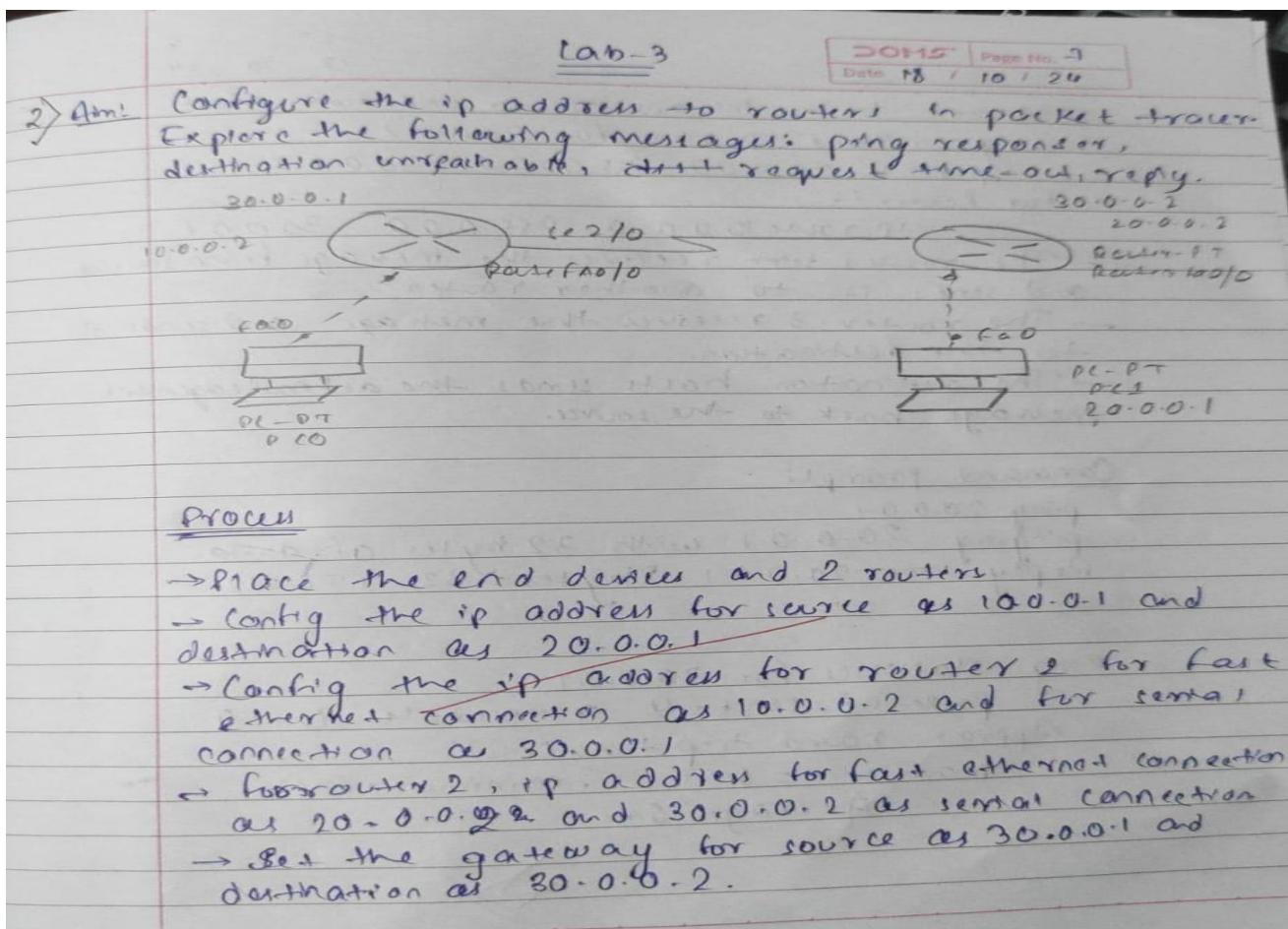
```
Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.2

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



- Connect both the routers to unknown host
- using command ip route for Router 2:
- ip route 20.0.0.0 255.0.0.0 30.0.0.2
- for Router 3:
- ip route 10.0.0.0 255.0.0.0 30.0.0.1
- The router 2 receives the message from source and sends it to another router
- The router 3 receives the message and sends it to the destination.
- The destination host sends the acknowledgement message back to the source.

#### Command prompt:

ping 20.0.0.1  
 pinging 20.0.0.1 with 32 bytes of data.  
 Reply from 20.0.0.1: bytes=32 TTL=12

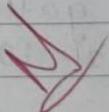
Ping statistics for 20.0.0.1

packets: sent=4 received=4 loss=0

Approx round trip time (ms):

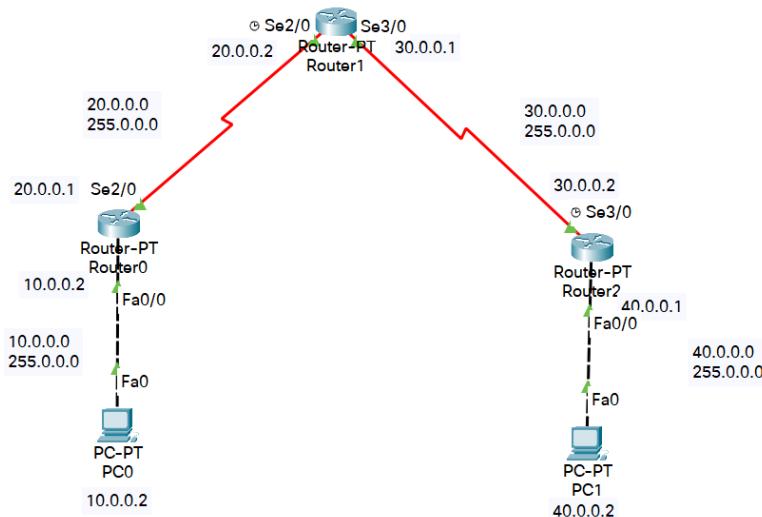
maximum=6ms, minimum=6ms

Average=6ms



### Program 3

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



- iii. Screen shots/ output

Router0 configuration

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

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

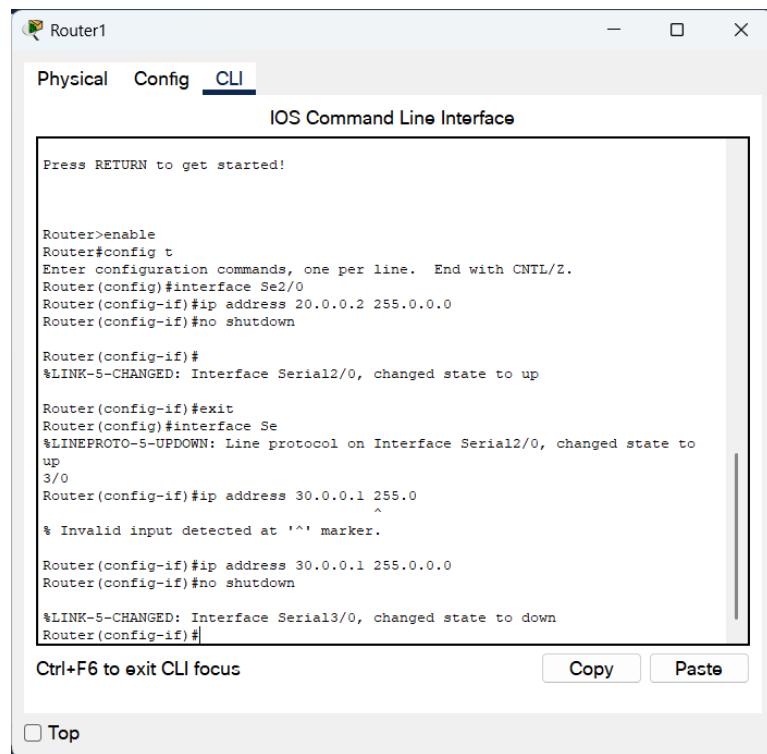
Router(config-if)#
$LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

$LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed
state to up
$IP-4-DUPLICADDR: Duplicate address 10.0.0.2 on FastEthernet0/0, sourced by
00C.CFC2.65B0

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

$LINK-5-CHANGED: Interface Serial2/0, changed state to down
Router(config-if)#
Ctrl+F6 to exit CLI focus
Copy Paste
Top
```

## Router1 configuration



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

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

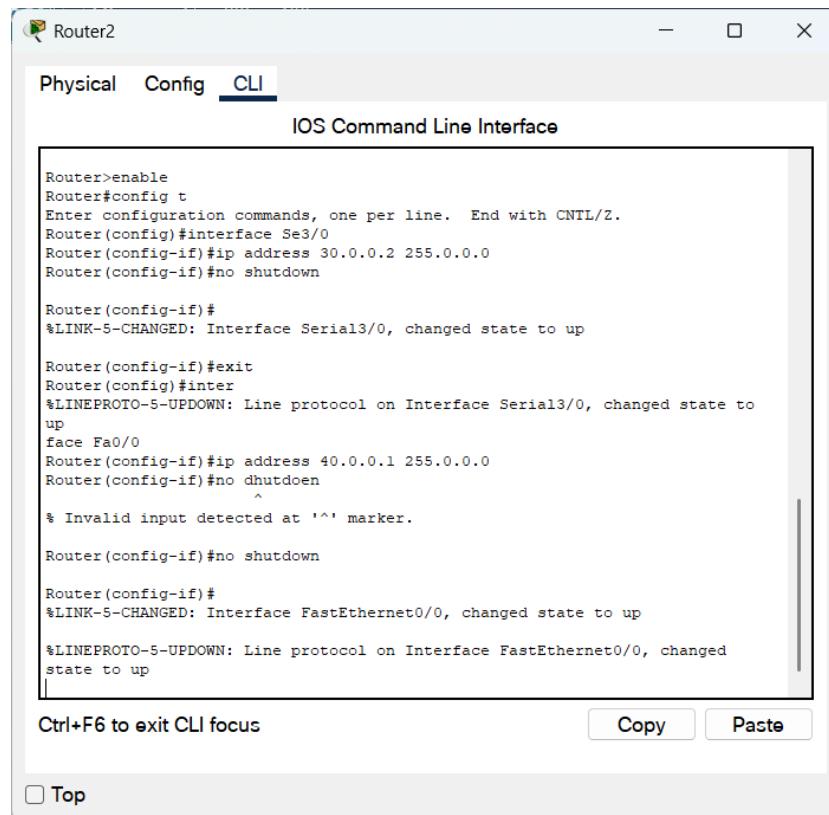
%LINK-5-CHANGED: Interface Serial3/0, changed state to down
Router(config-if)#

Ctrl+F6 to exit CLI focus
```

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)#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
```

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#

Ctrl+F6 to exit CLI focus     

Top

Router1

Router#  
%SYS-5-CONFIG\_I: Configured from console by console  
  
Router#config t  
Enter configuration commands, one per line. End with CNTL/Z.  
Router(config)#ip route 10.0.0.0 255.0.0.0 20.0.0.1  
Router(config)#ip route 40.0.0.0 255.0.0.0 30.0.0.2  
Router(config)#exit  
Router#  
%SYS-5-CONFIG\_I: Configured from console by console  
  
Router#show ip route  
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP  
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area  
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2  
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP  
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter  
area \* - candidate default, U - per-user static route, o - ODR  
P - periodic downloaded static route  
  
Gateway of last resort is not set  
  
S 10.0.0.0/8 [1/0] via 20.0.0.1  
C 20.0.0.0/8 is directly connected, Serial2/0  
C 30.0.0.0/8 is directly connected, Serial3/0  
S 40.0.0.0/8 [1/0] via 30.0.0.2  
  
Router#

Ctrl+F6 to exit CLI focus     

Top

## Router2

Router>enable  
Router#config t  
Enter configuration commands, one per line. End with CNTL/Z.  
Router(config)#ip route 10.0.0.0 255.0.0.0 30.0.0.1  
Router(config)#ip route 20.0.0.0 255.0.0.0 30.0.0.1  
Router(config)#exit  
Router#  
%SYS-5-CONFIG\_I: Configured from console by console  
  
Router#show ip route  
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP  
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area  
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2  
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP  
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter  
area  
\* - candidate default, U - per-user static route, o - ODR  
P - periodic downloaded static route  
  
Gateway of last resort is not set  
  
S 10.0.0.0/8 [1/0] via 30.0.0.1  
S 20.0.0.0/8 [1/0] via 30.0.0.1  
C 30.0.0.0/8 is directly connected, Serial3/0  
C 40.0.0.0/8 is directly connected, FastEthernet0/0  
  
Router#

Ctrl+F6 to exit CLI focus     

Top

## Dynamic Routing:

### Route0

Router0#enable  
Router0#config t  
Enter configuration commands, one per line. End with CNTL/Z.  
Router0(config)#ip route 0.0.0.0 0.0.0.0 20.0.0.2  
Router0(config)#exit  
Router0#  
%SYS-5-CONFIG\_I: Configured from console by console  
  
Router0#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  
  
Router0#

Ctrl+F6 to exit CLI focus     

Top

## Router2

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

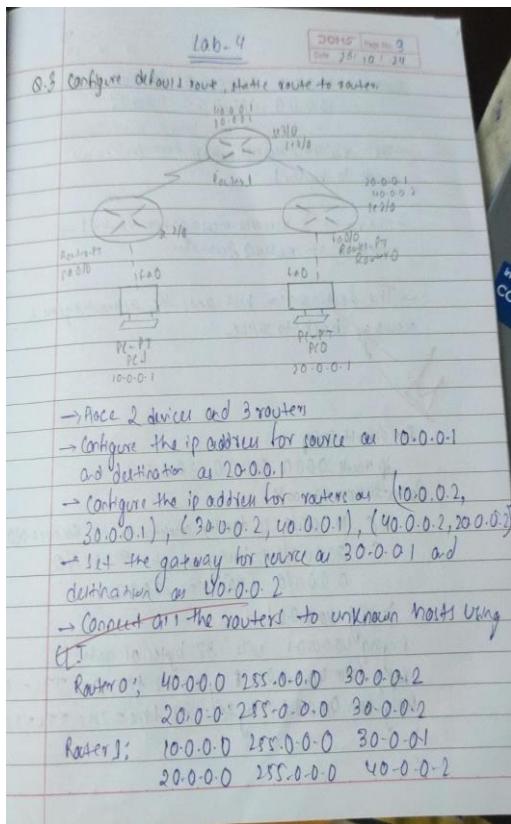
```
C:\>ping 40.0.0.2

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



20MS Page No. 10  
Date 28/10/20

Router 2: 30.0.0.0 255.0.0.0 40.0.0.1  
10.0.0.0 255.0.0.0 40.0.0.1

→ The Router 0 sends message from source to destination to Router 1

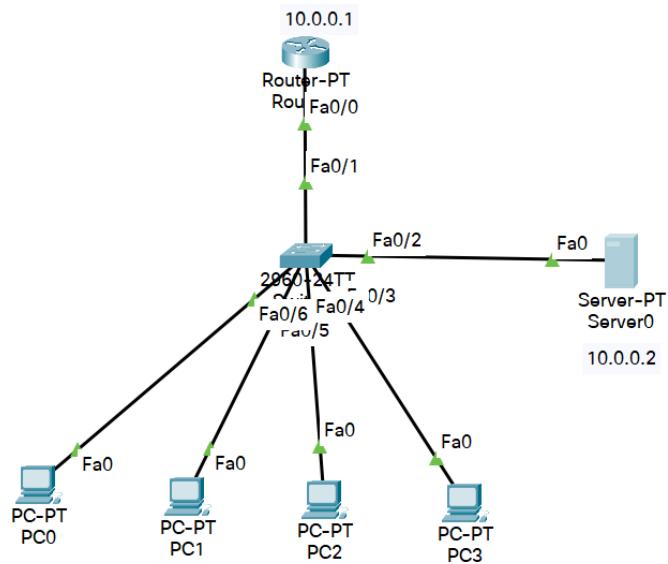
→ The Router 1 sends message from Router 1 to send to Router 2 destination.

→ The destination host sends the acknowledgement message back to source.

For default config  
ip route 0.0.0.0 255.0.0.0 20.0.0.2  
Router # show ip route  
C 10.0.0.0/8 is directly connected to Fast Ethernet 0/0  
C 20.0.0.0/8 is directly connected Serial 2/0  
0.0.0.0/0 [10] via 20.0.0.2  
> ping 40.0.0.1  
PING 40.0.0.1 (40.0.0.1) 32 bytes of data  
Reply from 40.0.0.1: bytes=32 time=2ms TTL=128

## Program 4

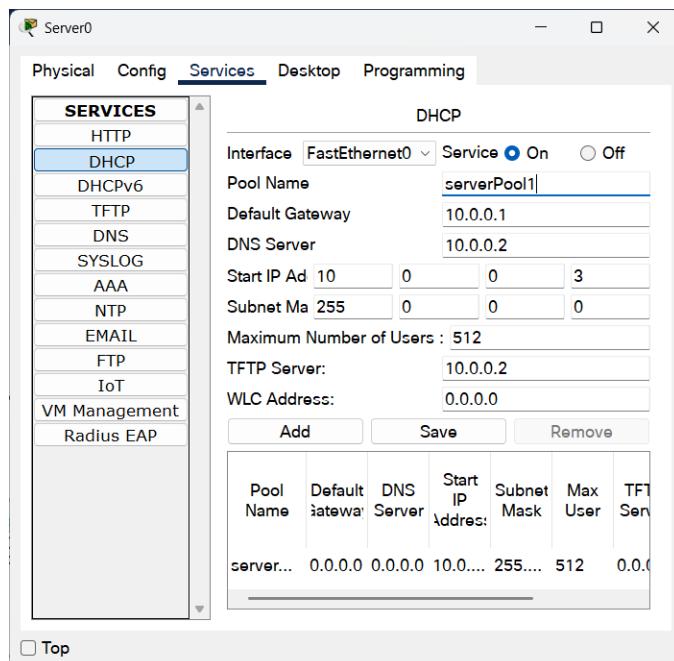
- i. Configure DHCP within a LAN and outside LAN.
- ii. Procedure along with the topology



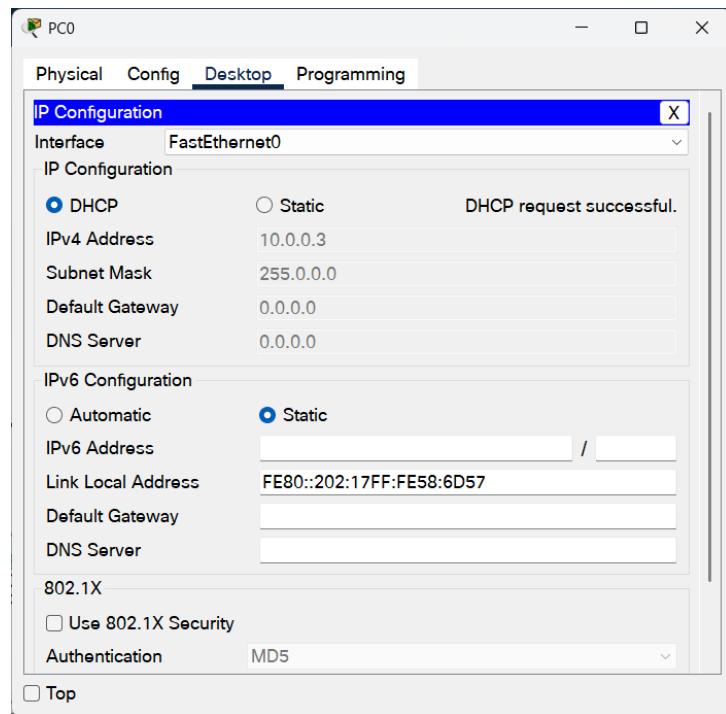
- iii. Screen shots/ output

DHCP Within LAN

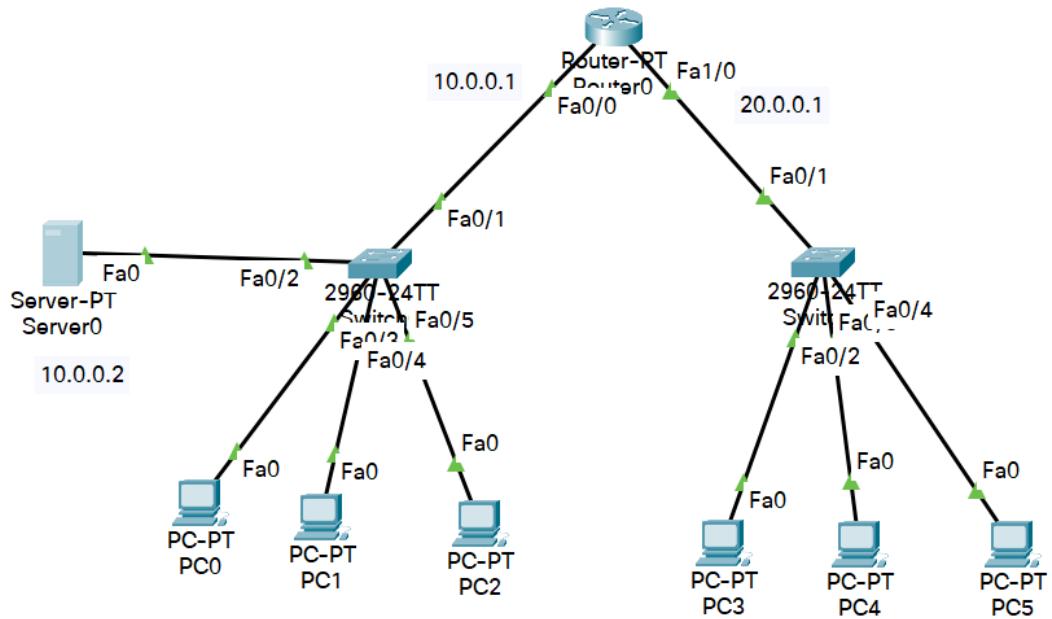
DHCP Configuration



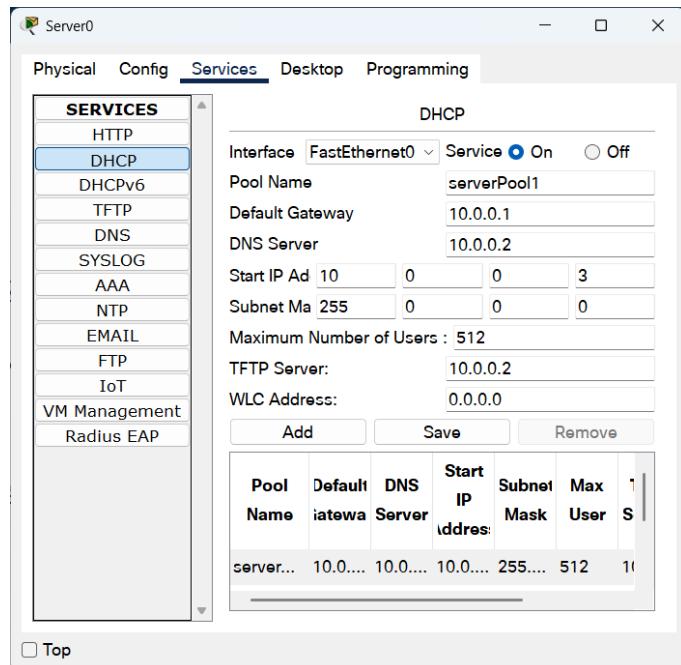
## PC settings



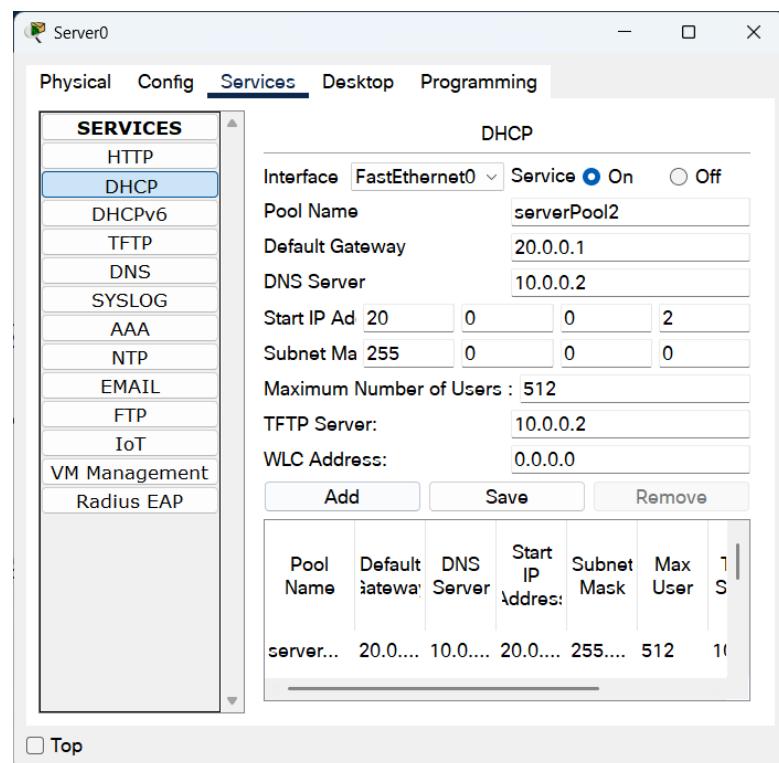
DHCP outside LAN:



DHCP configuration for inside LAN



DHCP configuration for outside LAN



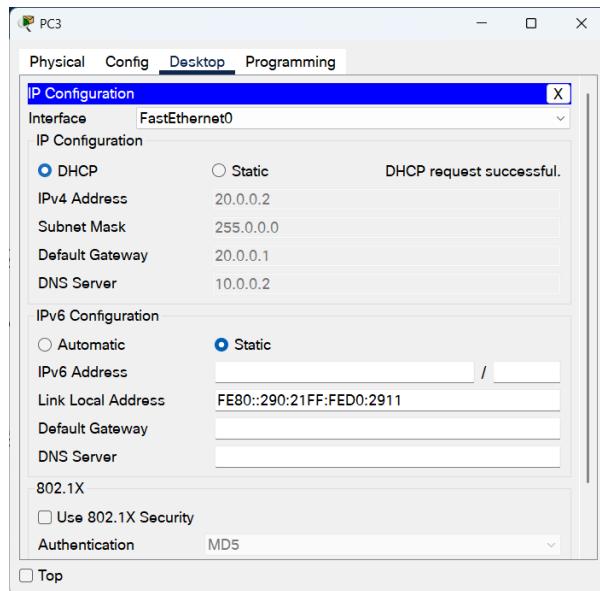
Ip helper command in Router

```

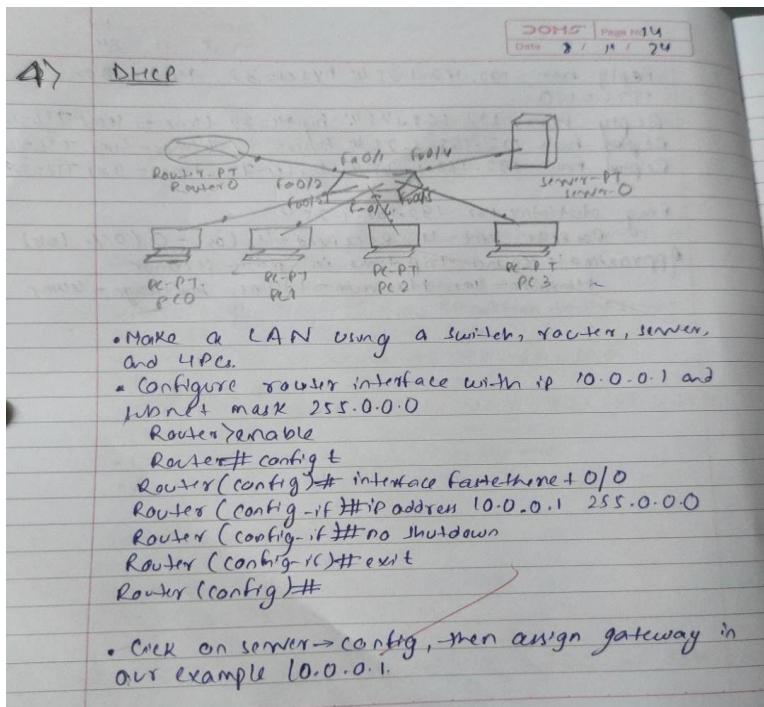
Router(config-if)#exit
Router(config)#interface Fa1/0
Router(config-if)#ip helper-address 10.0.0.2
Router(config-if)#

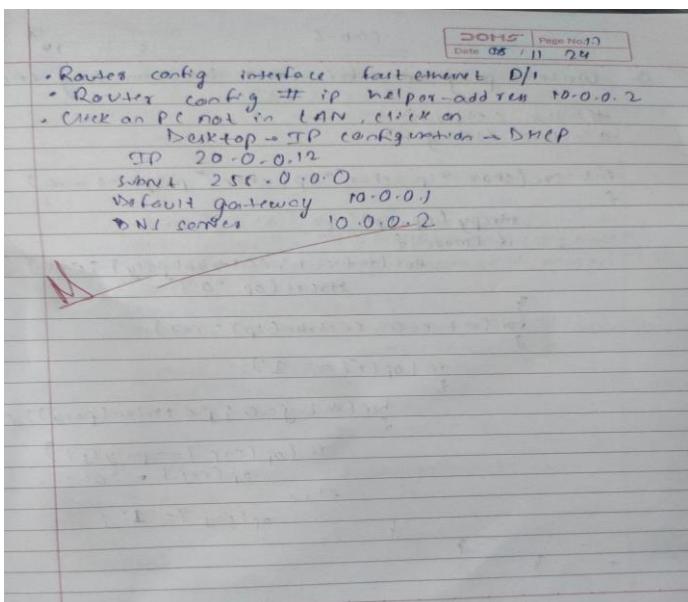
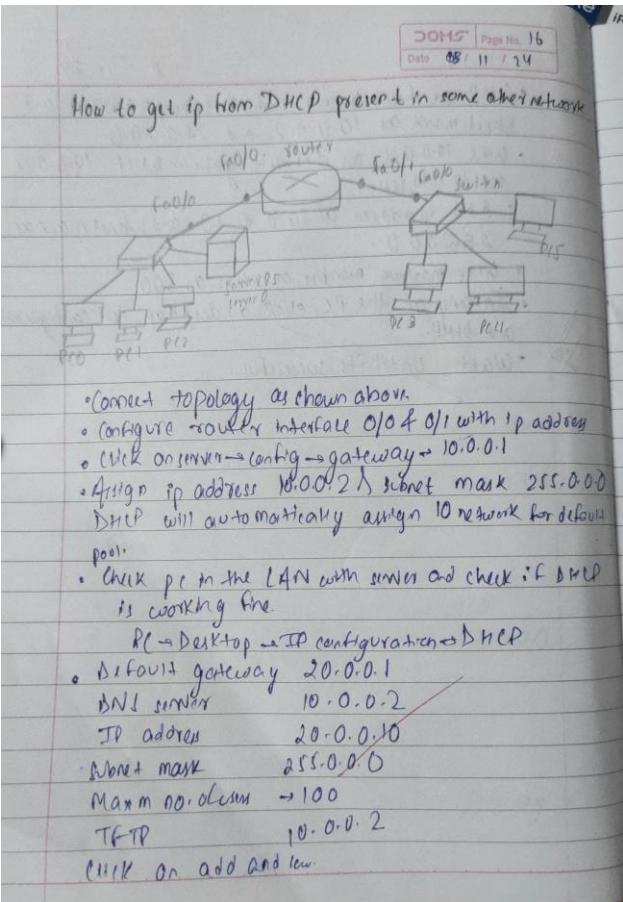
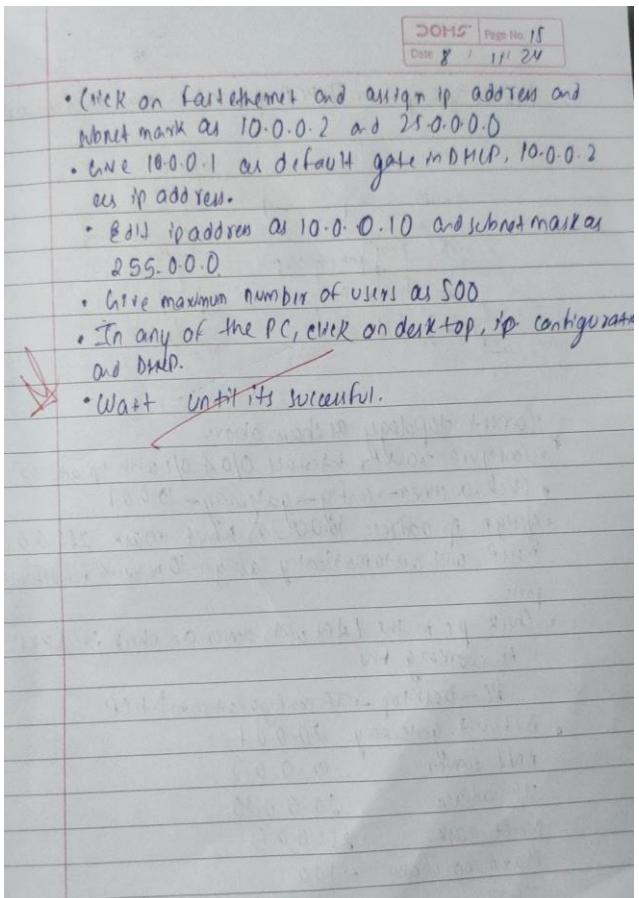
```

PC setting in another network



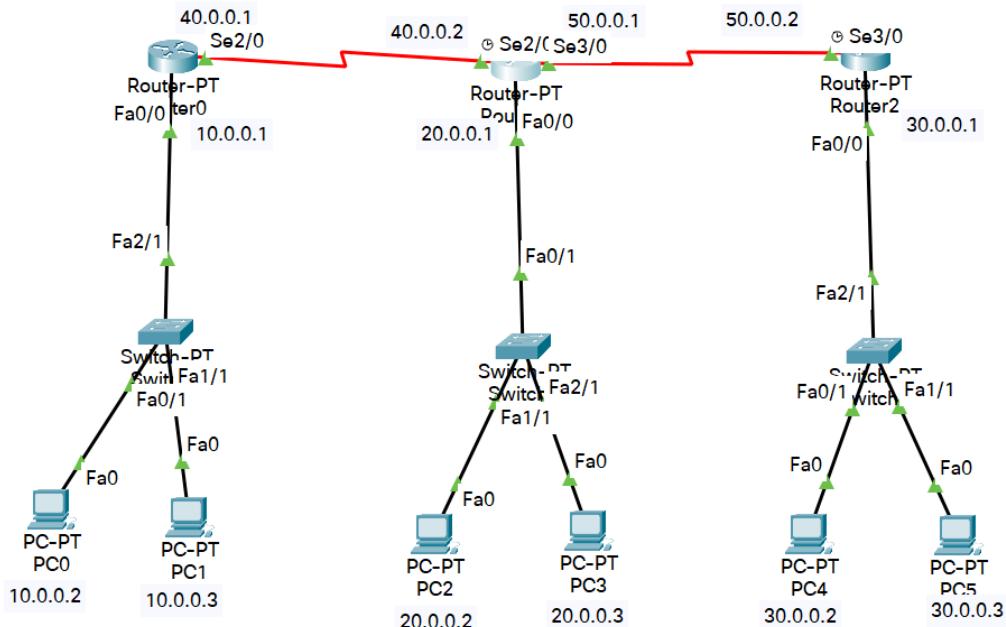
#### iv. Observation





## 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:

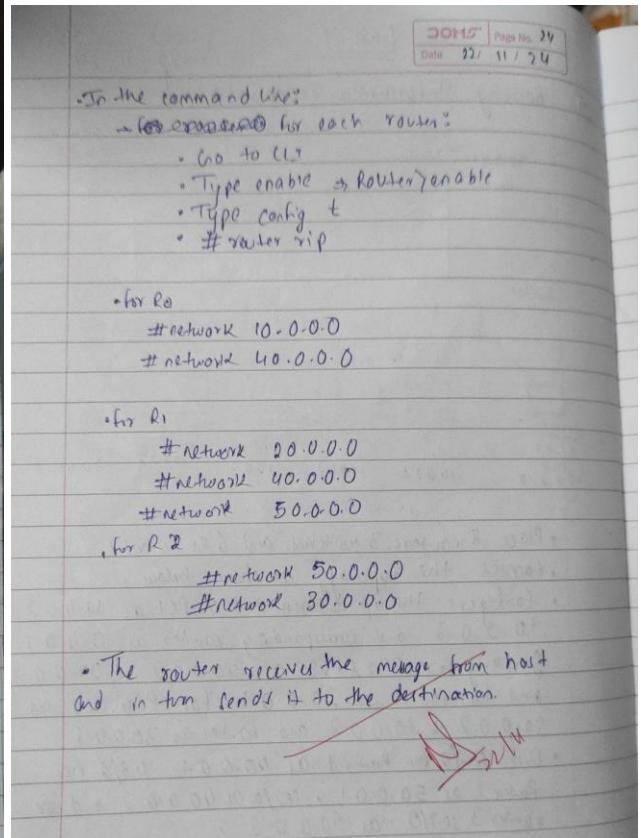
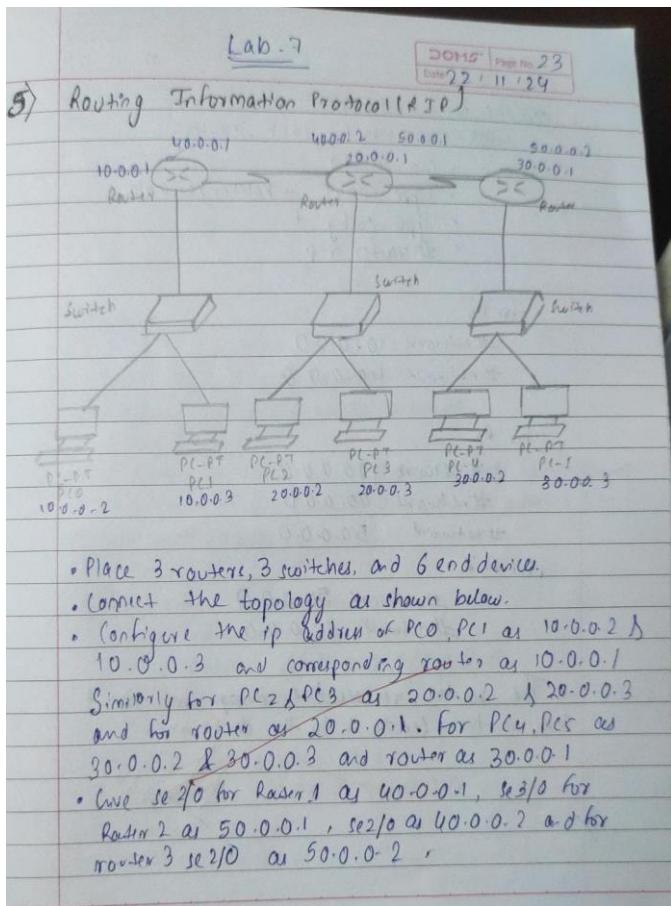
```
Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.2

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

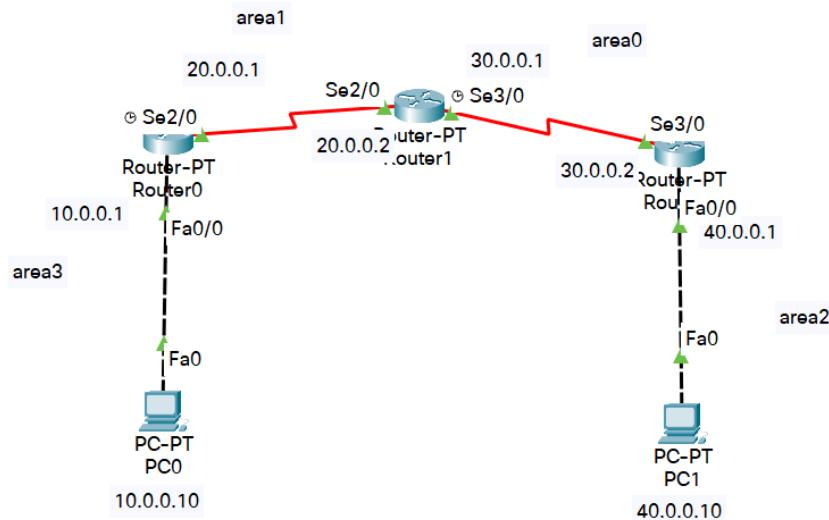
Ping statistics for 20.0.0.2:
  Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
  Minimum = 1ms, Maximum = 9ms, Average = 6ms
```

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

Router(config-if)#no shutdown
```

## Router2

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

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

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed
state to up
%IP-4-DUPADDR: Duplicate address 40.0.0.1 on FastEthernet0/0, sourced by
000D.BDDA.0123

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

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

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

## OSPF Routing Protocol

### Router0

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

Router#sho
00:27:19: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Serial2/0 from LOADING to FULL, Loading Done
w ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
      * - candidate default, U - per-user static route, o - ODR
      P - periodic downloaded static route

Gateway of last resort is not set

C    10.0.0.0/8 is directly connected, FastEthernet0/0
     20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        20.0.0.0/8 is directly connected, Serial2/0
C        20.0.0.2/32 is directly connected, Serial2/0
O  IA 30.0.0.0/8 [110/128] via 20.0.0.2, 00:00:02, Serial2/0
O  IA 40.0.0.0/8 [110/129] via 20.0.0.2, 00:00:02, Serial2/0
```

### Router1

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

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

Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
      * - candidate default, U - per-user static route, o - ODR
      P - periodic downloaded static route

Gateway of last resort is not set

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

## Router2

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

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

Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
      * - candidate default, U - per-user static route, o - ODR
      P - periodic downloaded static route

Gateway of last resort is not set

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

## Configure Loopback address

## Router0

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

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

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

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

## Router1

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

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

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

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

```

## Router2

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

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

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

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

## Create Virtual Link

### Router0

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

### Router1

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

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

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

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

## Pinging

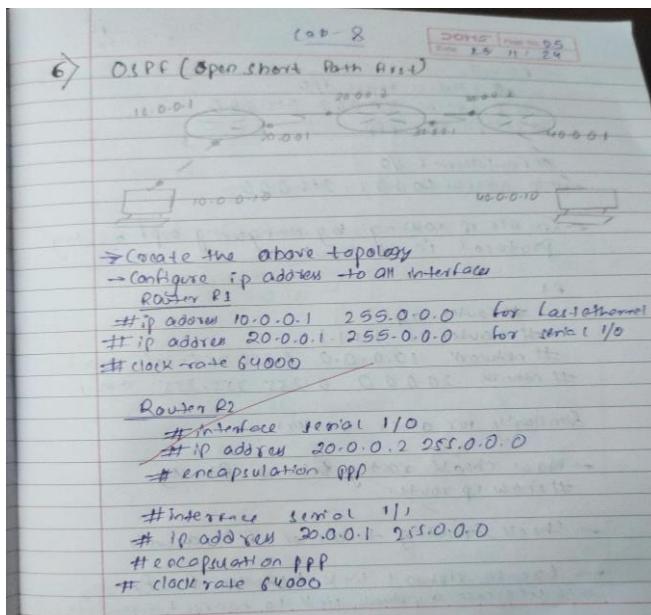
```
C:\>ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data:

Reply from 40.0.0.10: bytes=32 time=24ms TTL=125
Reply from 40.0.0.10: bytes=32 time=18ms TTL=125
Reply from 40.0.0.10: bytes=32 time=18ms TTL=125
Reply from 40.0.0.10: bytes=32 time=20ms TTL=125

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

#### iv. Observation



Router 3  
 # Interface serial 1/0  
 # IP address 30.0.0.2 255.0.0.0  
 # encapsulation PPP  
 # Interface serial 1/1  
 # IP address 40.0.0.1 255.0.0.0  
 # encapsulation PPP  
 # clockrate 64000

→ Enable IP routing by configuring OSPF routing protocol in all routers

R1  
 # Router OSPF 1  
 # Router-ID 1.1.1.1  
 # Network 10.0.0.0 0.255.255.255 area 0  
 # Network 20.0.0.0 0.255.255.255 area 1

Similarly for other routers

→ Now check routing table of R1 by  
 # show ip route

→ Check routing table of R3 as well

→ Create virtual link between R1, R2; by this we create a virtual link to connect Area B to Area A

Area 0  
Router 1  
 # Router OSPF 1  
 # Area 0 virtual link 2.2.2.2

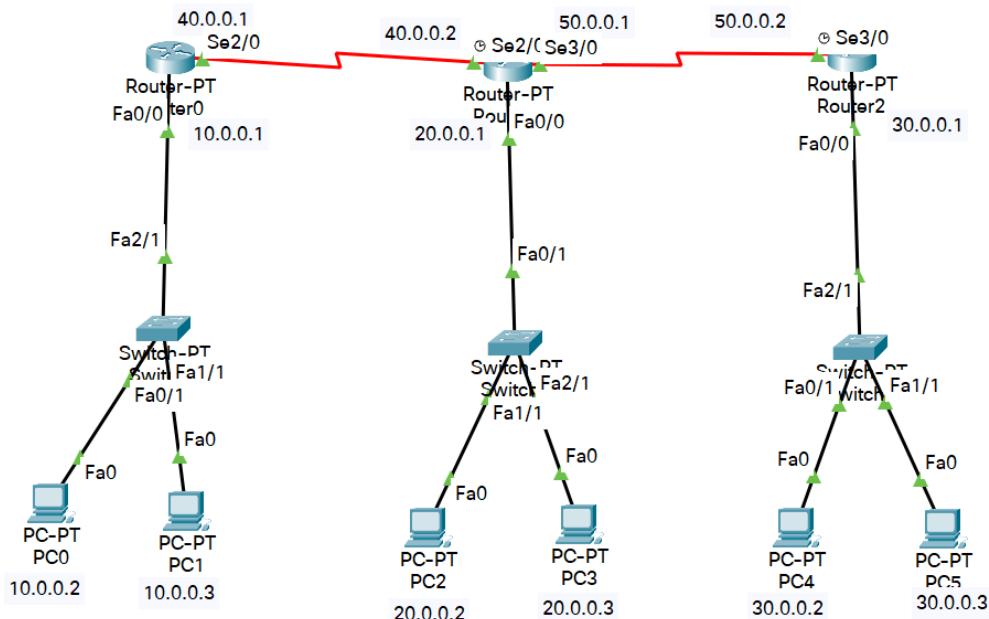
Router 2  
 # Area 0 virtual link 1.1.1.1

→ Now, check routing table of R3

→ Check connectivity between host 10.0.0.10 → 40.0.0.10

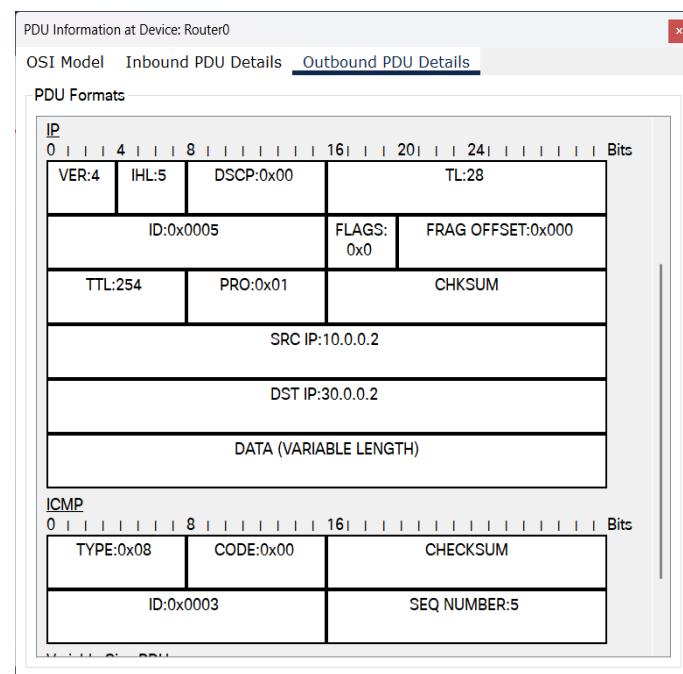
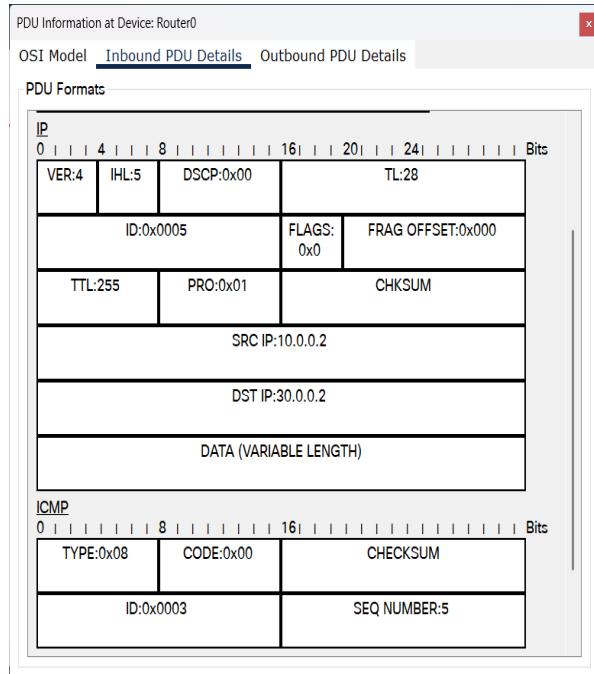
## Program 7

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

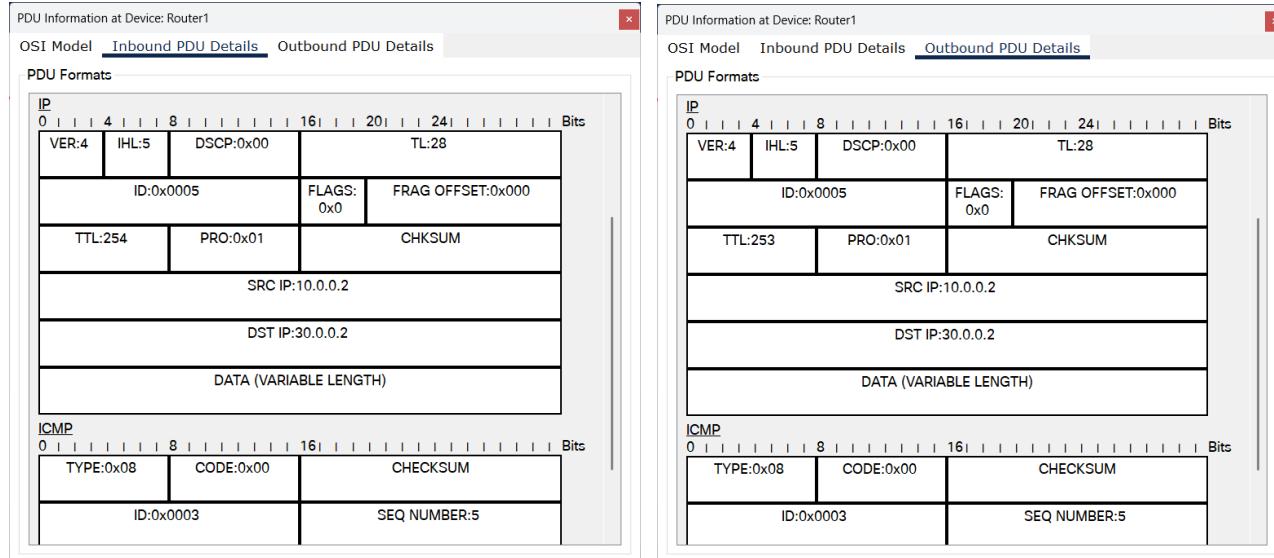


- ### iii. Screen shots/ output

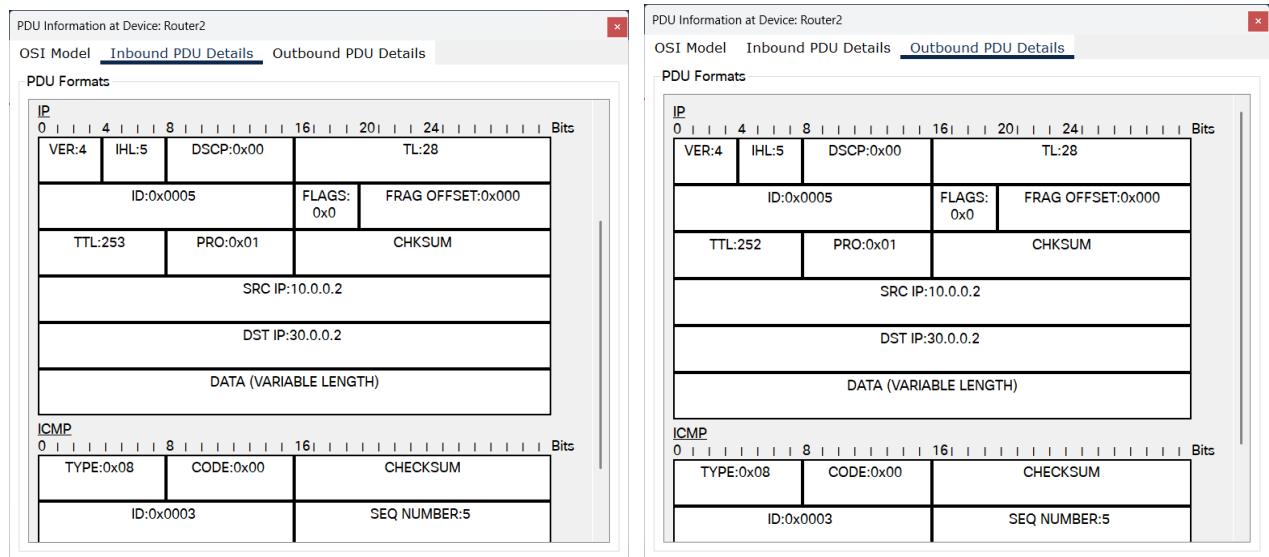
## Packet at Router0



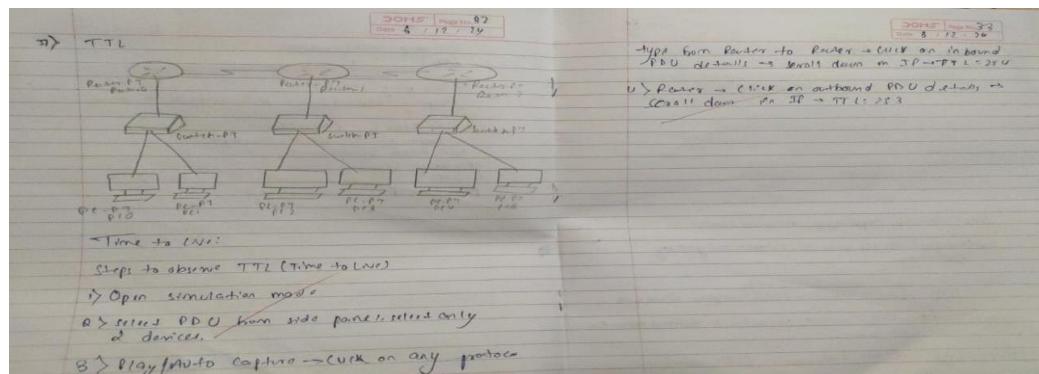
## Packet at Router1



## Packet at Router2

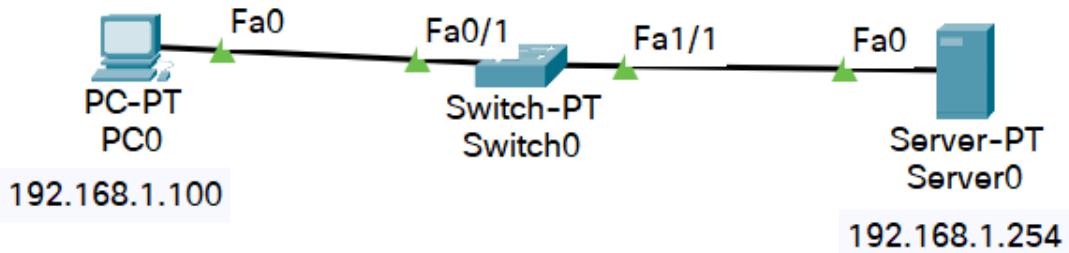


### iv. Observation



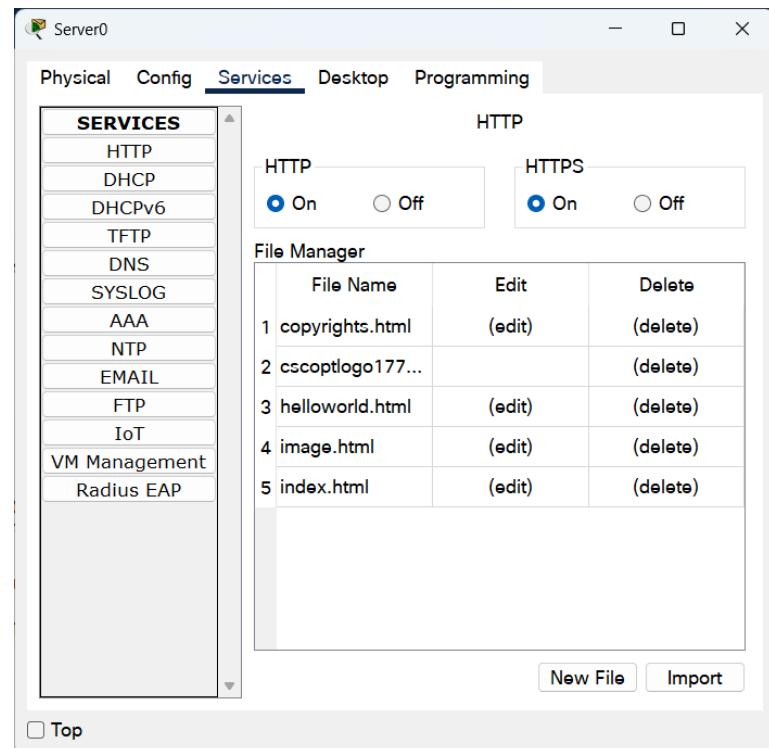
## Program 8

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

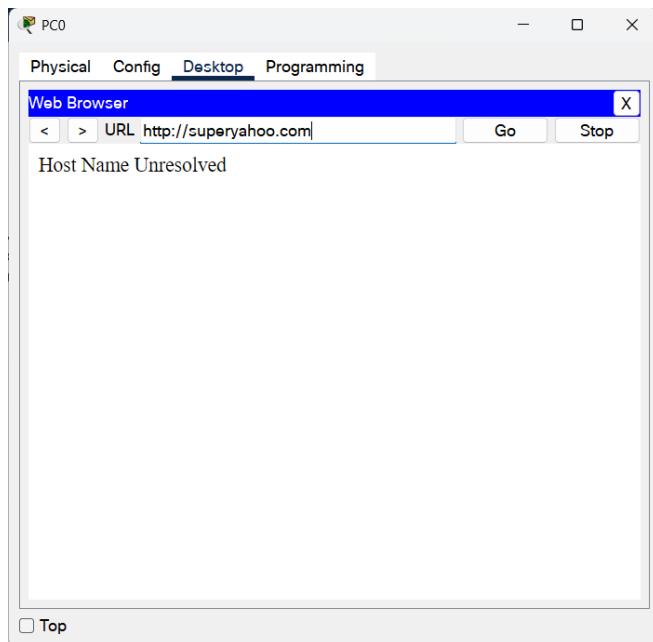
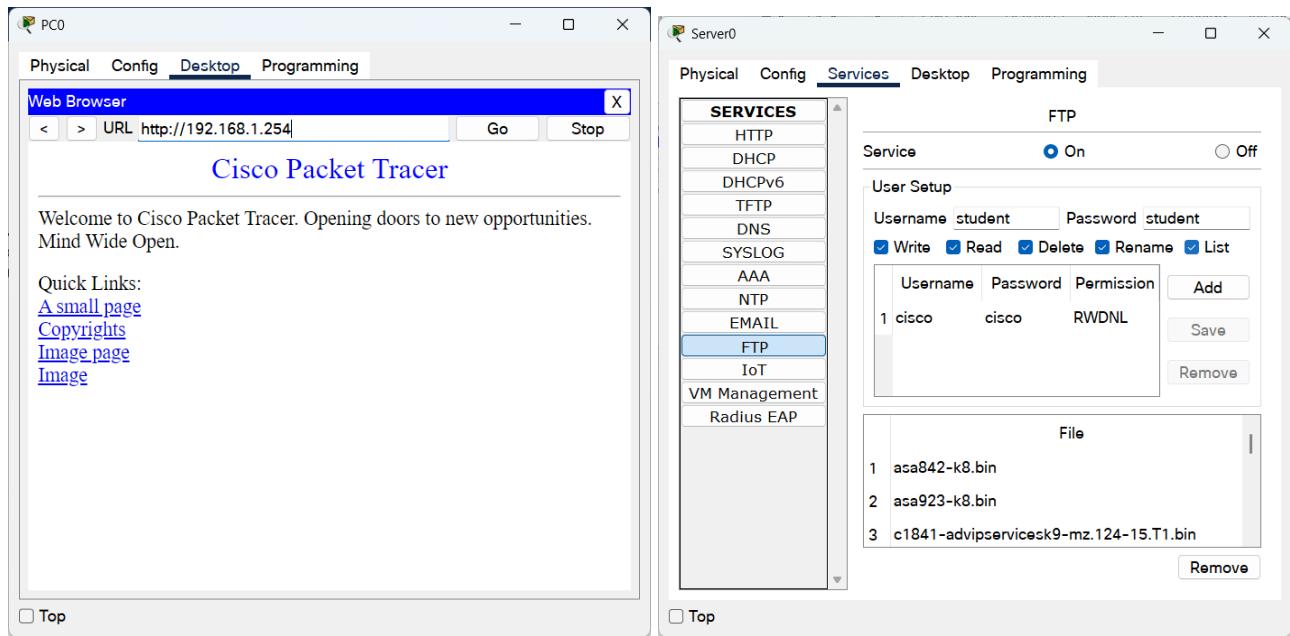


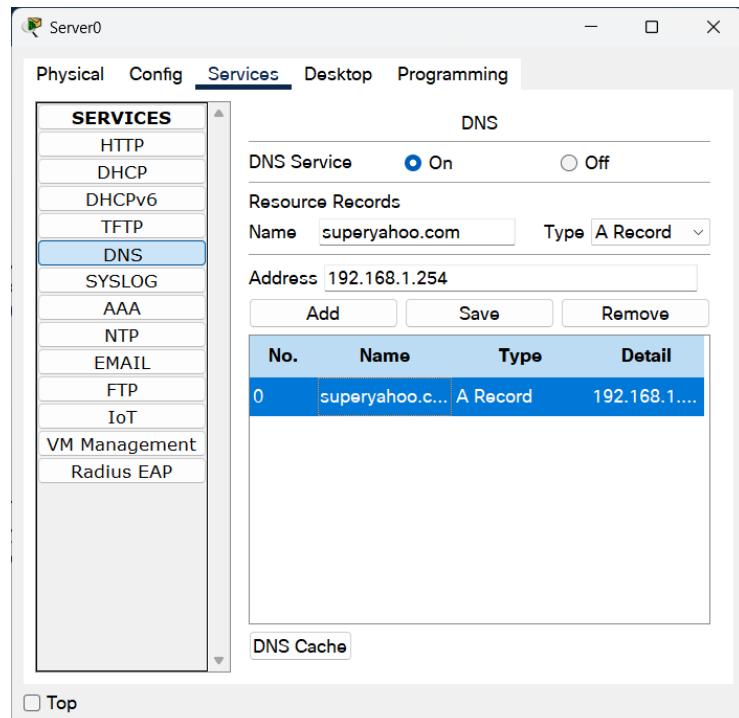
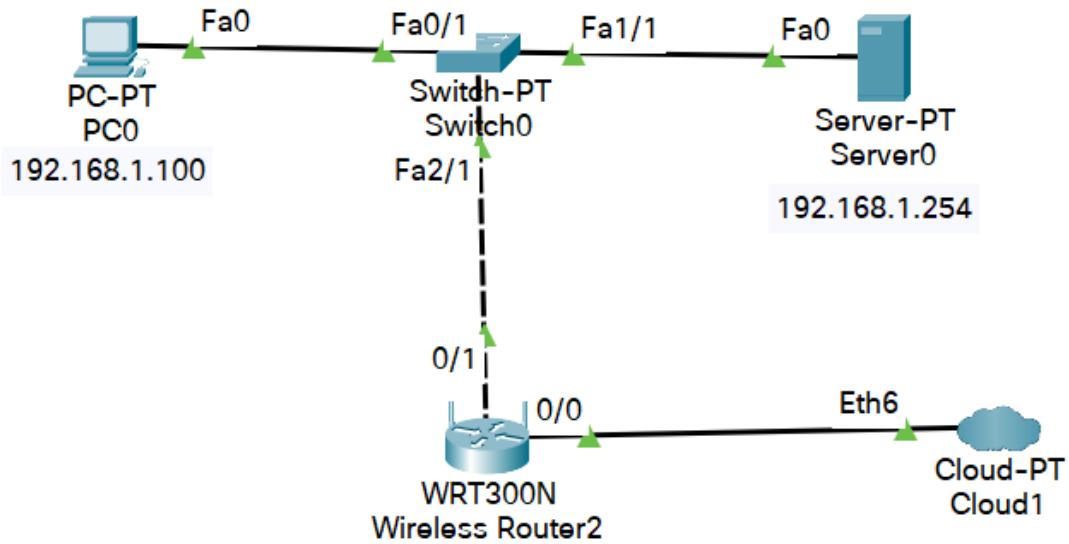
- iii. Screen shots/ output

Server's services



PC's Web Browser





#### iv. Observation

DOMS Page No. 12  
Date 8 / 11 / 24

Lab 5

8) DNS

- Place a generic PC, 2960 switch, generic server in the logical workspace.
- Connect these devices to make a LAN.
- Give 192.168.1.254 as IP address to the server.
- Give 192.168.1.200 as PC label.
- Give PC the IP address as 192.168.1.100.
- Ping from PC to the server.
- In the command prompt, ping 192.168.1.254.
- The web server has built-in now.
- Click on http of server config tab.
- A HTML homepage will be visible.

(Command prompt)

Packet Tracer PC Command Line IO

PC > ping 192.168.1.254

Pinging 192.168.1.254 with 32 bytes of data:

DOMS Page No. 13  
Date 8 / 11 / 24

Reply from 192.168.1.254: bytes=32 time=170ms TTL=120

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

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

Reply from 192.168.1.254: bytes=32 time=9ms TTL=128

Ping statistics for 192.168.1.254

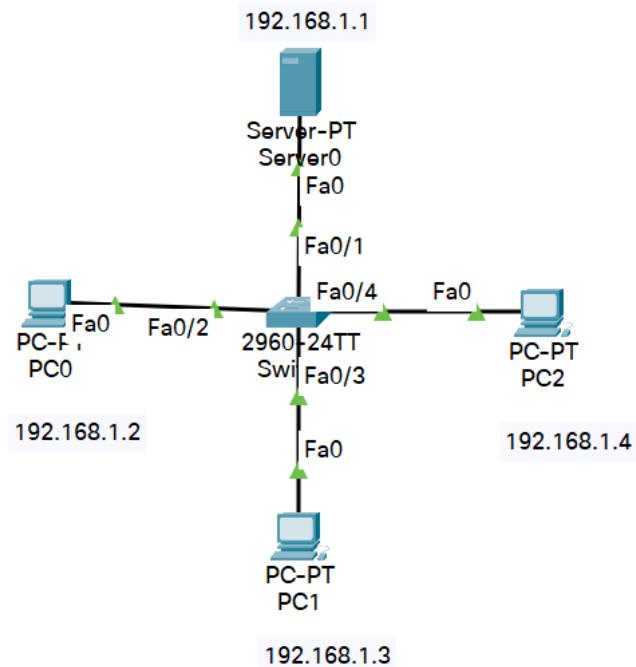
Packets: Sent=4, Received=4, Lost=0 (0% loss)

Approximate round trip time in milli-seconds

Minimum=4ms, Maximum=170ms, Average=40ms

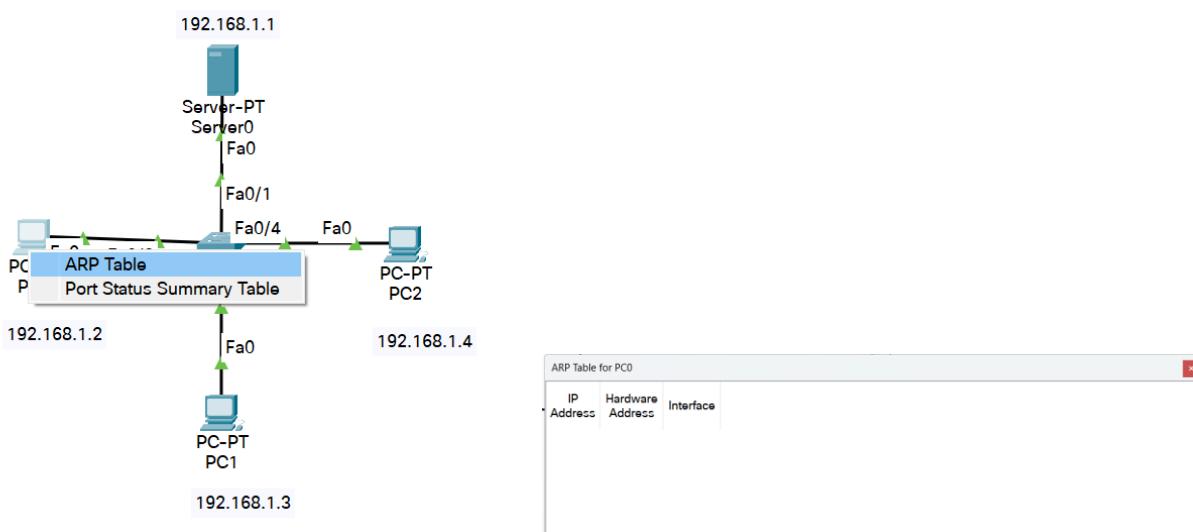
## Program 9

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

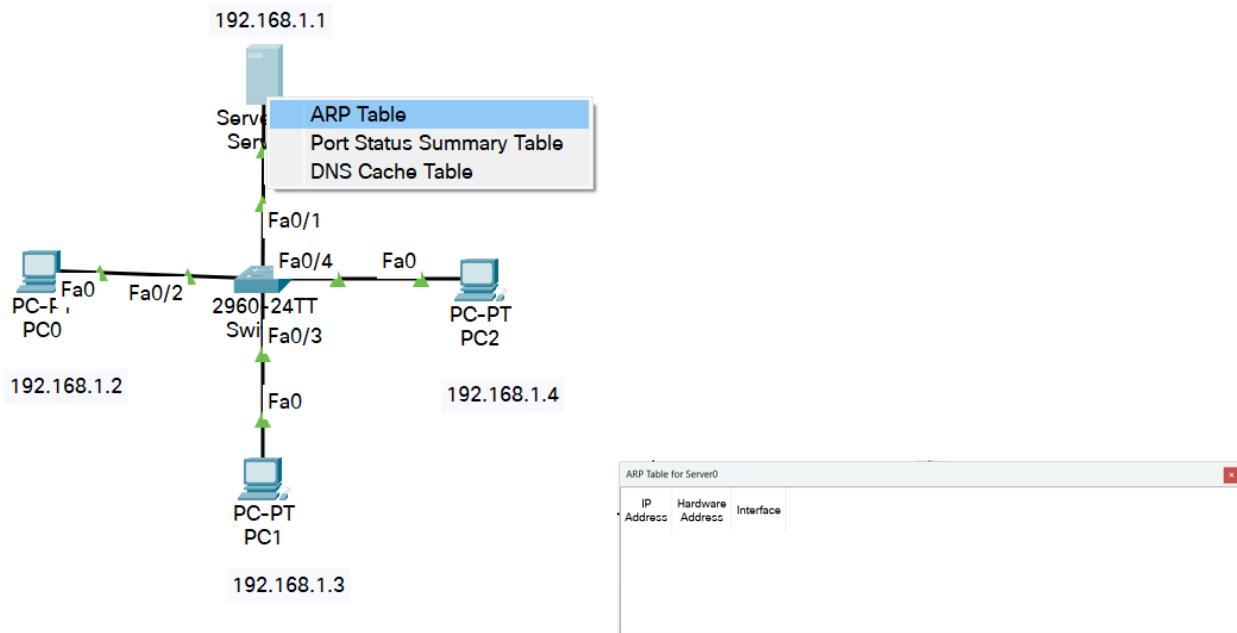


- iii. Screen shots/ output

ARP Table of PC



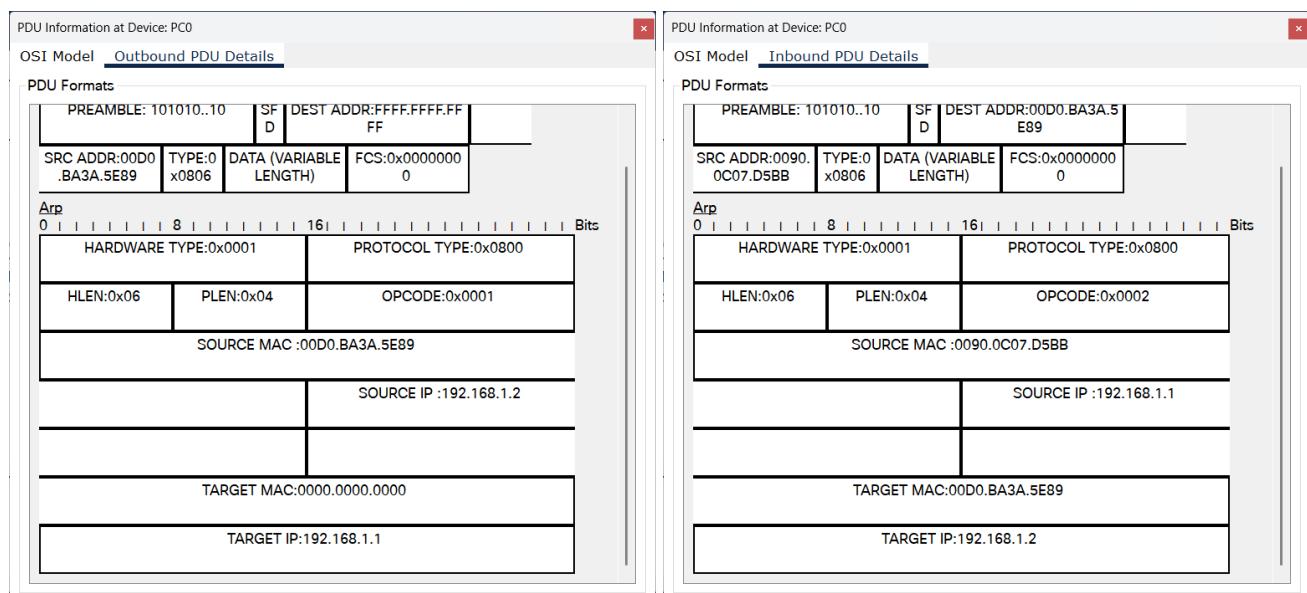
ARP Table of Server



### Command at PC

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

### Pinging in Simulation Mode



#### iv. Observation

DOMS | Page No. 28  
Date 6 / 12 / 24

Lab-9

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

Step 1: Assign IP addresses to all PCs and server

Step 2: Go to simulation panel, click on inspect and right click on PC0

Step 3: There will be no entries in the ARP Table

Step 4: Click on PC0 again and go to command prompt

Step 5: Gyp -a

DOMS | Page No. 29  
Date 6 / 12 / 24

ping 192.168.11.4

Step 6: Two packets are created ICMP and ARP

Step 7: Move over the packets → check the type of packets

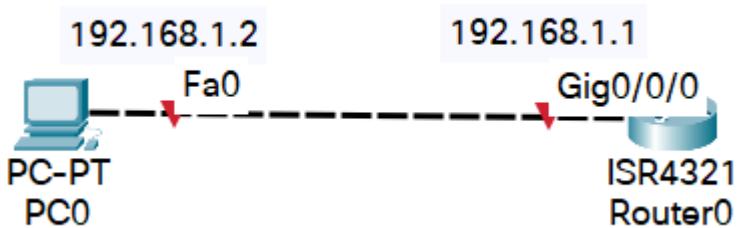
Step 8: Click on ARP.  
Click on capture button to start the simulation

Step 9: Ping request from PC0 to switch with number  
Again click on capture.  
Click on capture twice to see the ping acceptance  
from server 0 to PC0

Step 10: Click on capture to see ICMP packet movement

## **Program 10**

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



- iii. Screen shots/ output

### Router

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

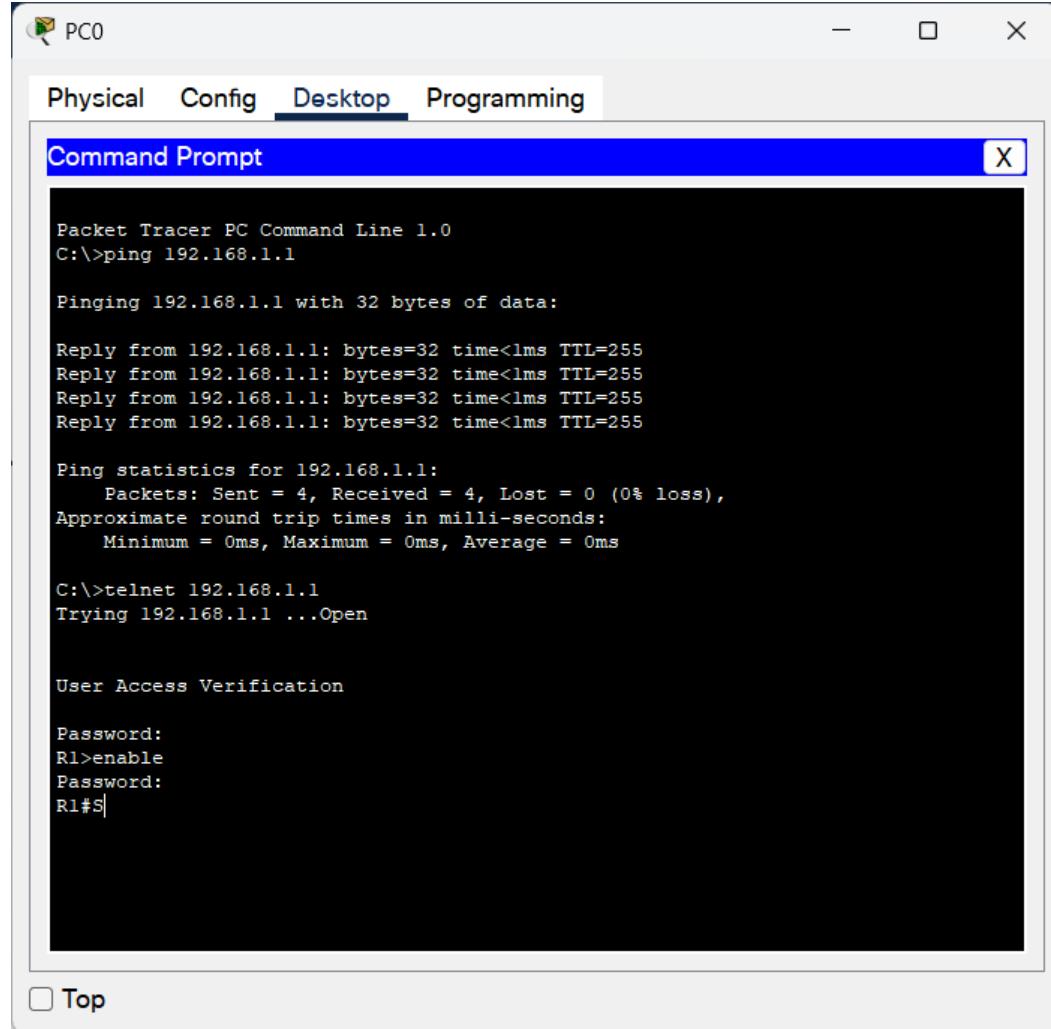
R1(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to up

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

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

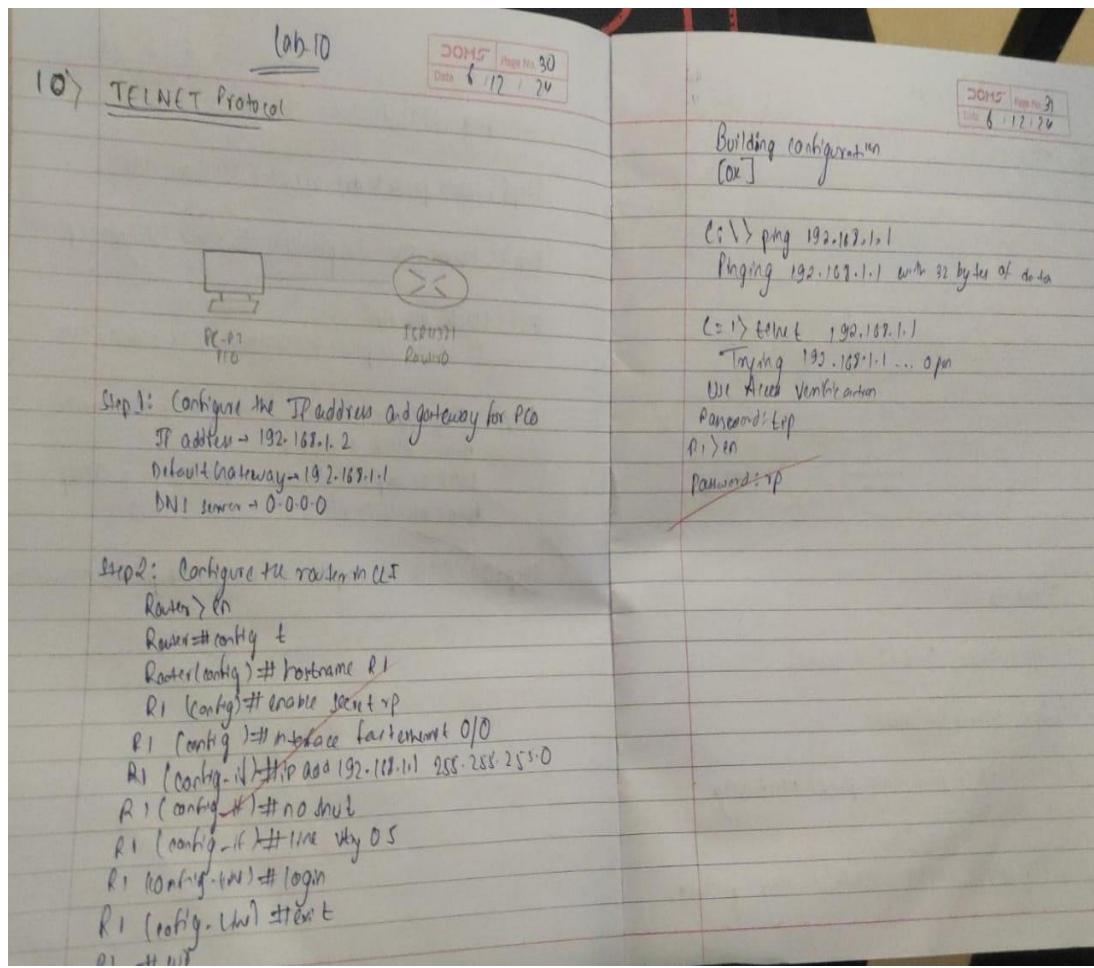
R1#wr
Building configuration...
[OK]
R1#
```

PC



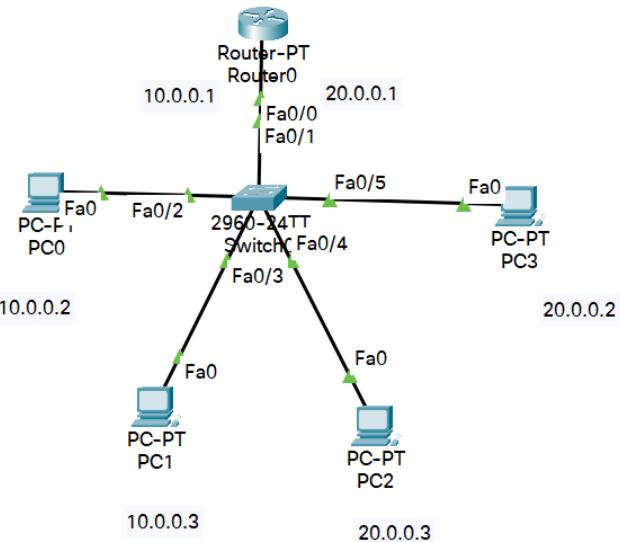
Top

#### iv. Observation



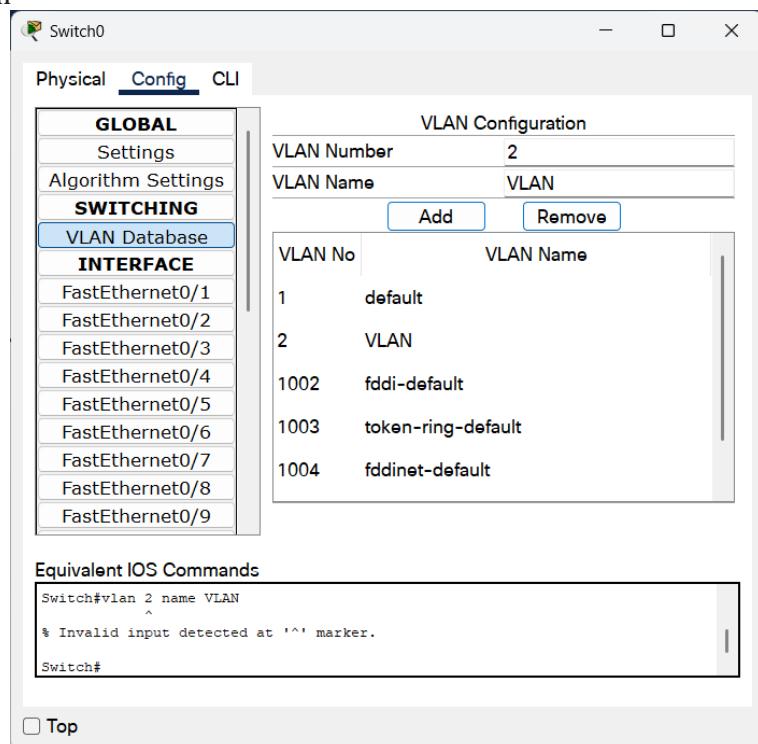
## Program 11

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

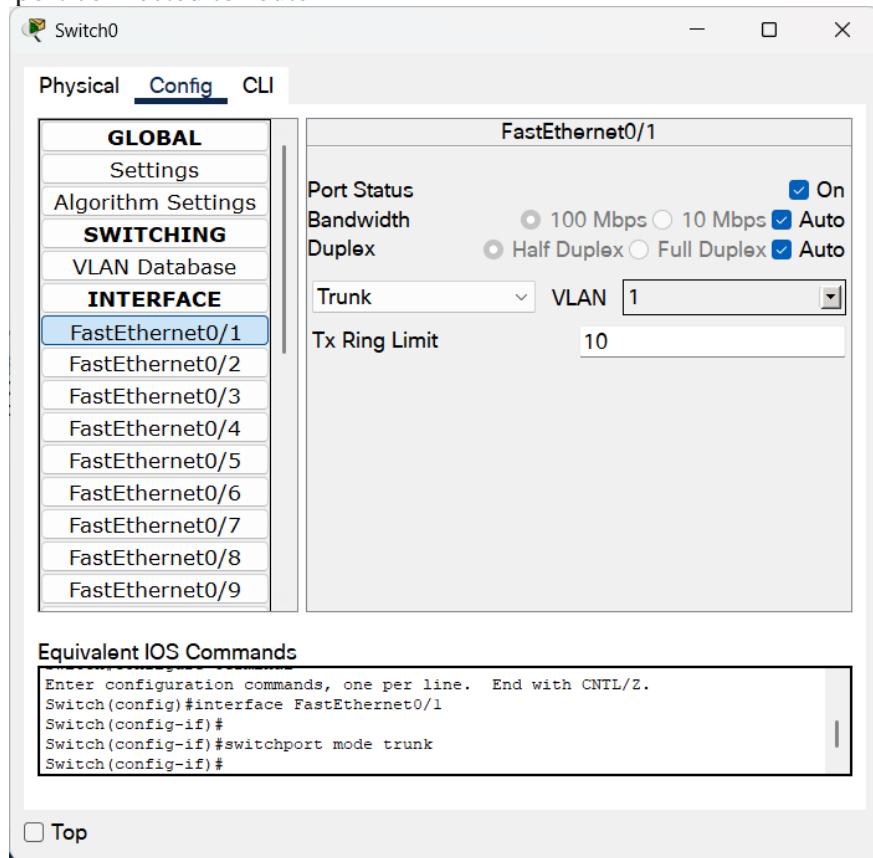


- iii. Screen shots/ output

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

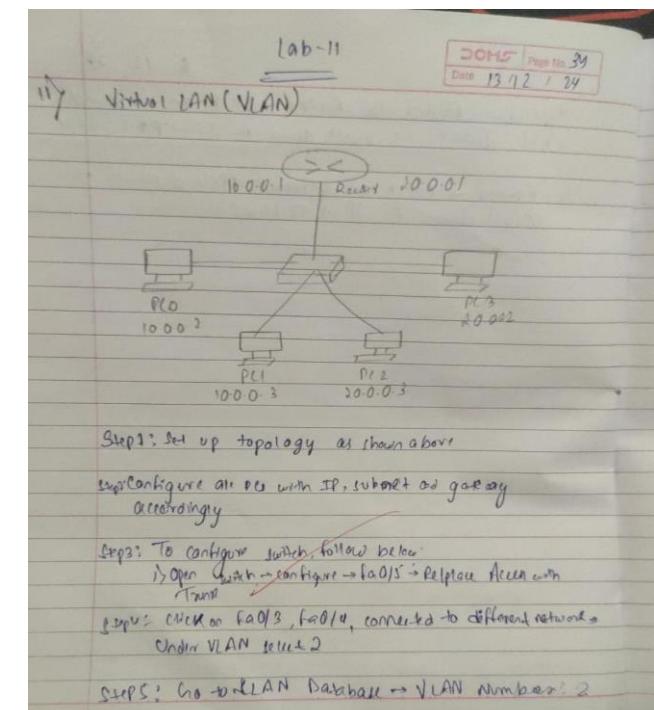
Router(config-subif)#encapsulation dot1q 2
Router(config-subif)#ip address 20.0.0.1 255.0.0.0
Router(config-subif)#no shutdown
Router(config-subif)#exit
Router(config)#

Ctrl+F6 to exit CLI focus
```

Top

**Copy** **Paste**

#### iv. Observation



DOMS Page No. 35  
Date 13/12/24

VLAN name: VLAN → Add

Step 5: Configure router with following commands:

```

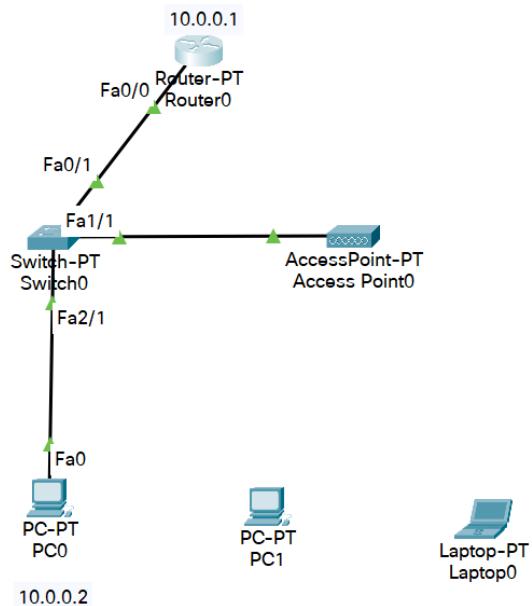
Router> enable
Router> config t
Config# interface fa0/0
(Config-if)# ip address 192.168.1.1 255.255.255.0
(Config-if)# no shutdown
(Config-if)
# interface fa0/0.1
# encapsulation dot1q 2
# ip address 192.168.2.1 255.255.255.0
# no shutdown

```

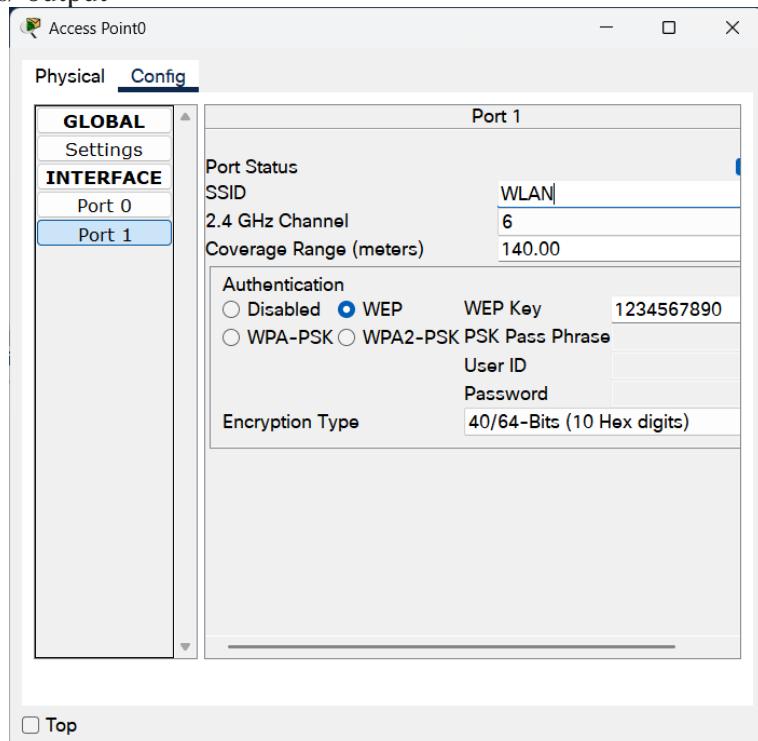
Step 6: Check connectivity between networks

## Program 12

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



- iii. Screen shots/ output



**PC1**

- Physical**
- Config
- Desktop
- Programming

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



Customize Icon in Physical View
Customize Icon in Logical View

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

 Top

**Laptop0**

- Physical**
- Config
- Desktop
- Programming

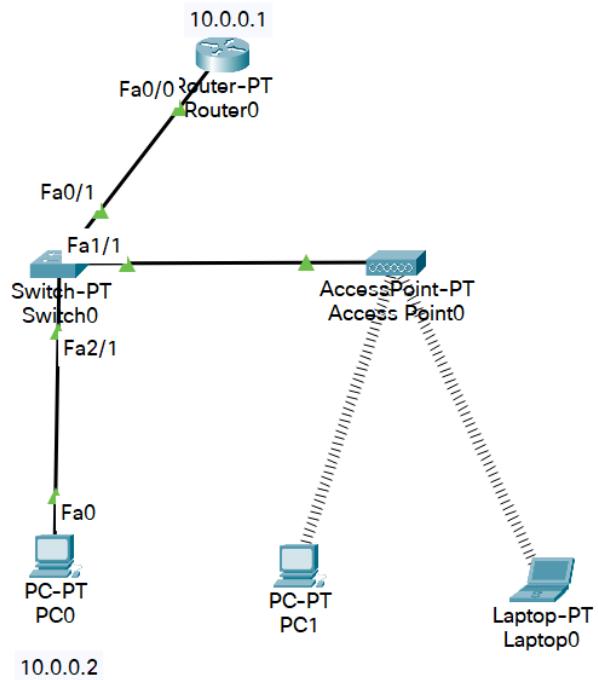
MODULES
WPC300N
PT-LAPTOP-NM-1AM
PT-LAPTOP-NM-1CE
PT-LAPTOP-NM-1CFE
PT-LAPTOP-NM-1CGE
PT-LAPTOP-NM-1FFE
PT-LAPTOP-NM-1FGE
PT-LAPTOP-NM-1W
PT-LAPTOP-NM-1W-A
PT-LAPTOP-NM-1W-AC
PT-LAPTOP-NM-3G/4G
PT-HEADPHONE
PT-MICROPHONE



Customize Icon in Physical View
Customize Icon in Logical View

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

 Top



Ping:

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

```

Physical Config Desktop Programming
Command Prompt X
Packet Tracer PC Command Line 1.0
C:\>

C:\>ping 10.0.0.3

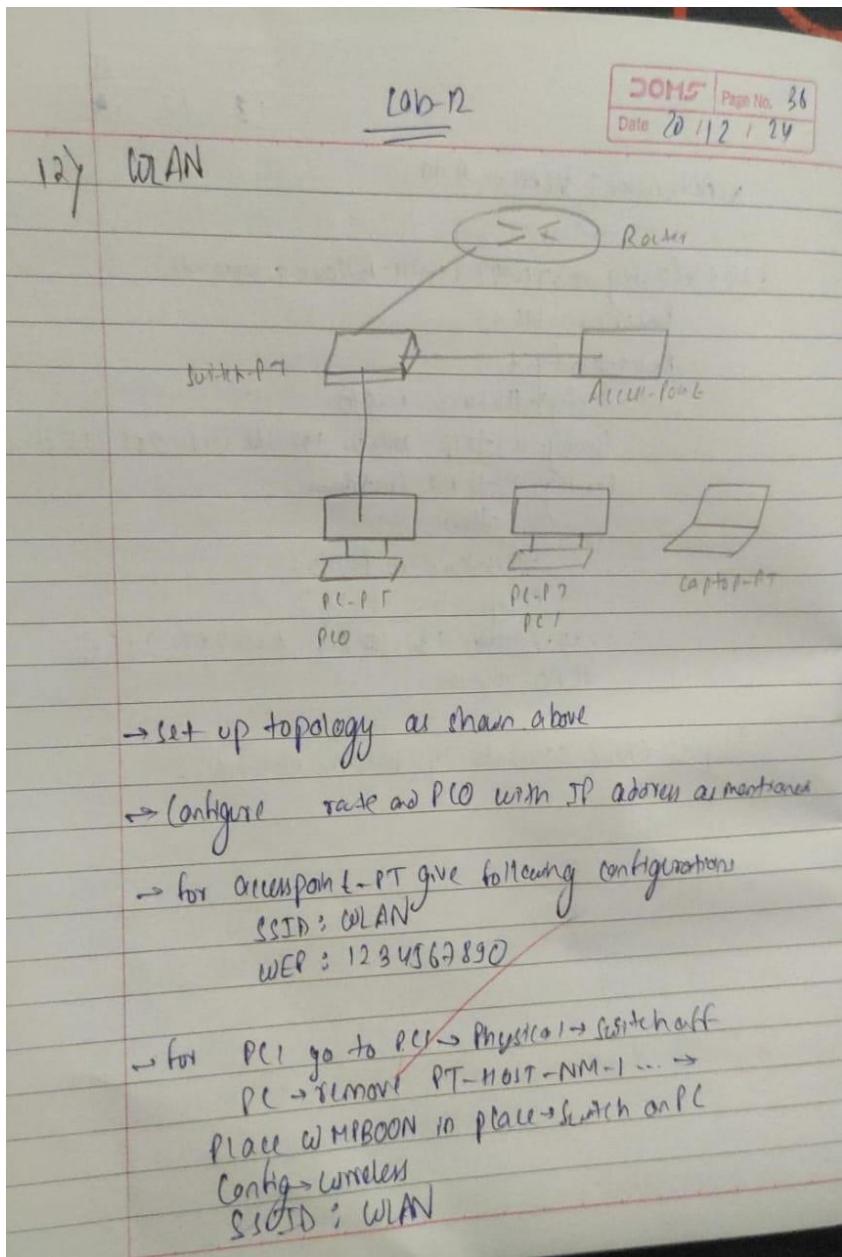
Pinging 10.0.0.3 with 32 bytes of data:
Reply from 10.0.0.3: bytes=32 time=40ms TTL=128
Reply from 10.0.0.3: bytes=32 time=25ms TTL=128
Reply from 10.0.0.3: bytes=32 time=26ms TTL=128
Reply from 10.0.0.3: bytes=32 time=24ms TTL=128

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

```

Top

iv. Observation



DOMS Page No. 36  
20/12/24

IP EPS : 10.0.0.5  
 IP V4 Address : 10.0.0.3  
 Subnet mask : 255.0.0.0

→ Same for Laptop-PT with following config -  
 WIRELESS  
 L1 ID : WMPBOON  
 WEP : 1234567890  
 IP V4 address : 10.0.0.4  
 Subnet mask : 255.0.0.0

→ Check connectivity of PC / Laptop

## Cycle-II

### Program 1

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

```
def crc_ccitt_16_bitstream(bitstream: str, poly: int = 0x1021, init_crc: int = 0xFFFF) -> int:  
    crc = init_crc  
    for bit in bitstream:  
        crc ^= int(bit) << 15 # Align the bit with CRC's uppermost bit  
        for _ in range(1): # Process the single bit  
            if crc & 0x8000: # Check if the leftmost bit is set  
                crc = (crc << 1) ^ poly  
            else:  
                crc <<= 1  
            crc &= 0xFFFF # Ensure CRC remains 16-bit  
    return crc  
  
def append_crc_to_bitstream(bitstream: str) -> str:  
    crc = crc_ccitt_16_bitstream(bitstream)  
    crc_bits = f'{crc:016b}' # Convert CRC to a 16-bit binary string  
    return bitstream + crc_bits  
  
def verify_crc_bitstream(bitstream_with_crc: str) -> bool:  
    if len(bitstream_with_crc) < 16:  
        return False # Not enough bits to contain CRC  
    data, received_crc = bitstream_with_crc[:-16], bitstream_with_crc[-16:]  
    calculated_crc = crc_ccitt_16_bitstream(data)  
    return calculated_crc == int(received_crc, 2)  
  
# Example usage:  
if __name__ == "__main__":  
    # User input for original bitstream  
    message_bits = input("Enter the original bitstream (e.g., 11010011101100): ")  
  
    # Calculate and append CRC  
    bitstream_with_crc = append_crc_to_bitstream(message_bits)  
    print(f"Bitstream with CRC: {bitstream_with_crc}")  
  
    # User input for verification  
    user_bitstream = input("Enter the received bitstream for verification (e.g., 1101001110110011011011000011): "
```

)

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

### iii. Screen shots/ output

```
In [1]: runcell(0, 'E:/python_files/untitled2.py')

Enter the original bitstream (e.g., 11010011101100): 11111
Bitstream with CRC: 111111111111111100000

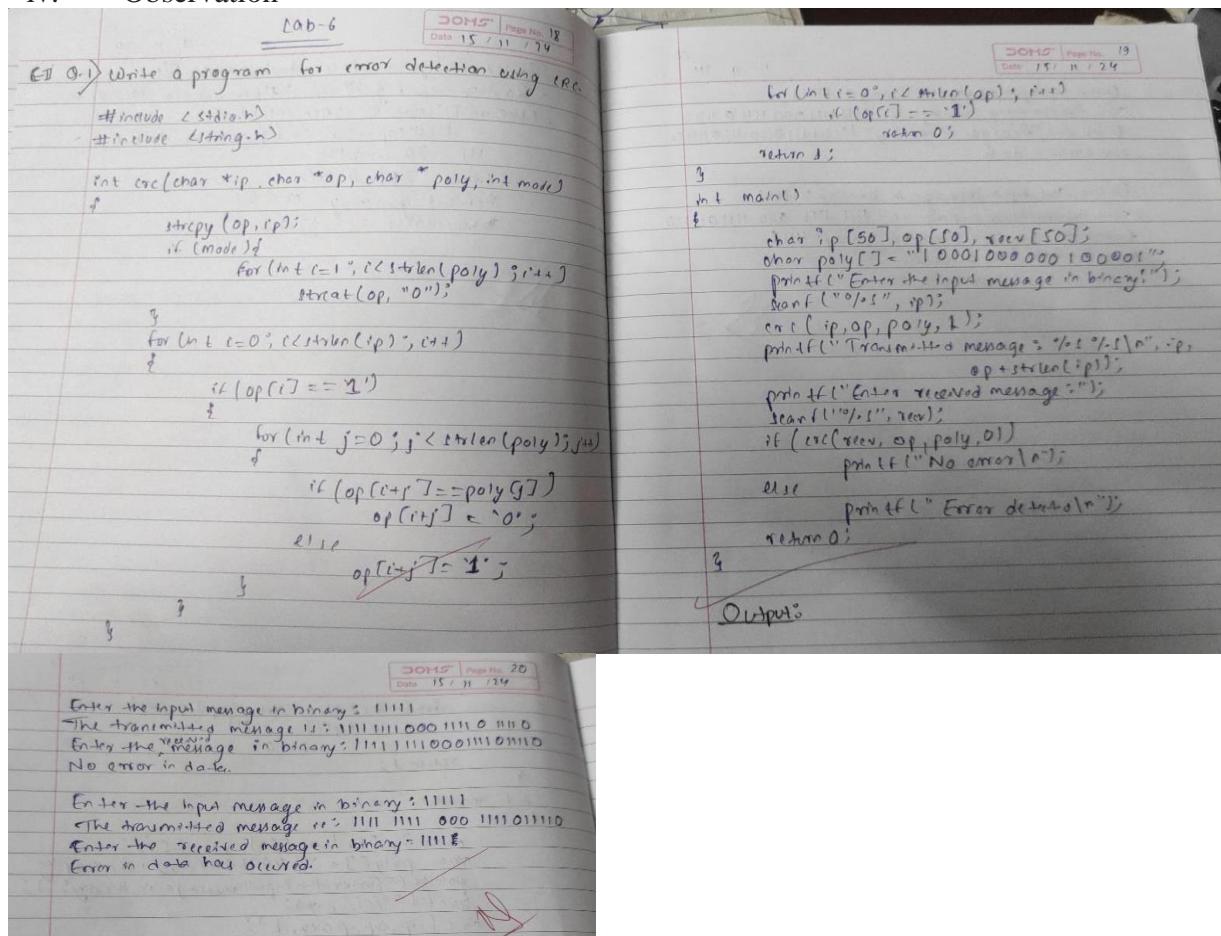
Enter the received bitstream for verification (e.g., 11010011101100110110110111000011): 1111111111111111100000
CRC valid: True

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

Enter the original bitstream (e.g., 11010011101100): 11111
Bitstream with CRC: 1111111111111111100000

Enter the received bitstream for verification (e.g., 11010011101100110110110111000011): 1111111111111111100001
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

<p>Q2 Write a program to implement Leaky Bucket Algorithm</p> <pre>#include &lt;stdio.h&gt;  int main() {     int no_of_packets, storage, output_pkts_size;     int input_pkts_size, bucket_size_left;      storage = 0;     no_of_packets = 4;     bucket_size_left = 10;     input_pkts_size = 4;     output_pkts_size = 3;     for (int i = 0; i &lt; no_of_packets; i++) {         if (input_pkts_size &lt; bucket_size_left) {             storage += input_pkts_size;             input_pkts_size = 0;         } else {             printf("Packet loss = %d\n", input_pkts_size - bucket_size_left);             input_pkts_size -= bucket_size_left;         }         printf("Buffer size = %d out of bucket size = %d\n", storage, bucket_size_left);     } }</pre>	<p>Storage = output_pkts_size; if (storage &lt; 0) storage = 0;</p> <p>return 0;</p> <p><u>Output</u></p> <p>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 = 3 Buffer size = 9 out of bucket size = 10</p>
---	---

### **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()
```

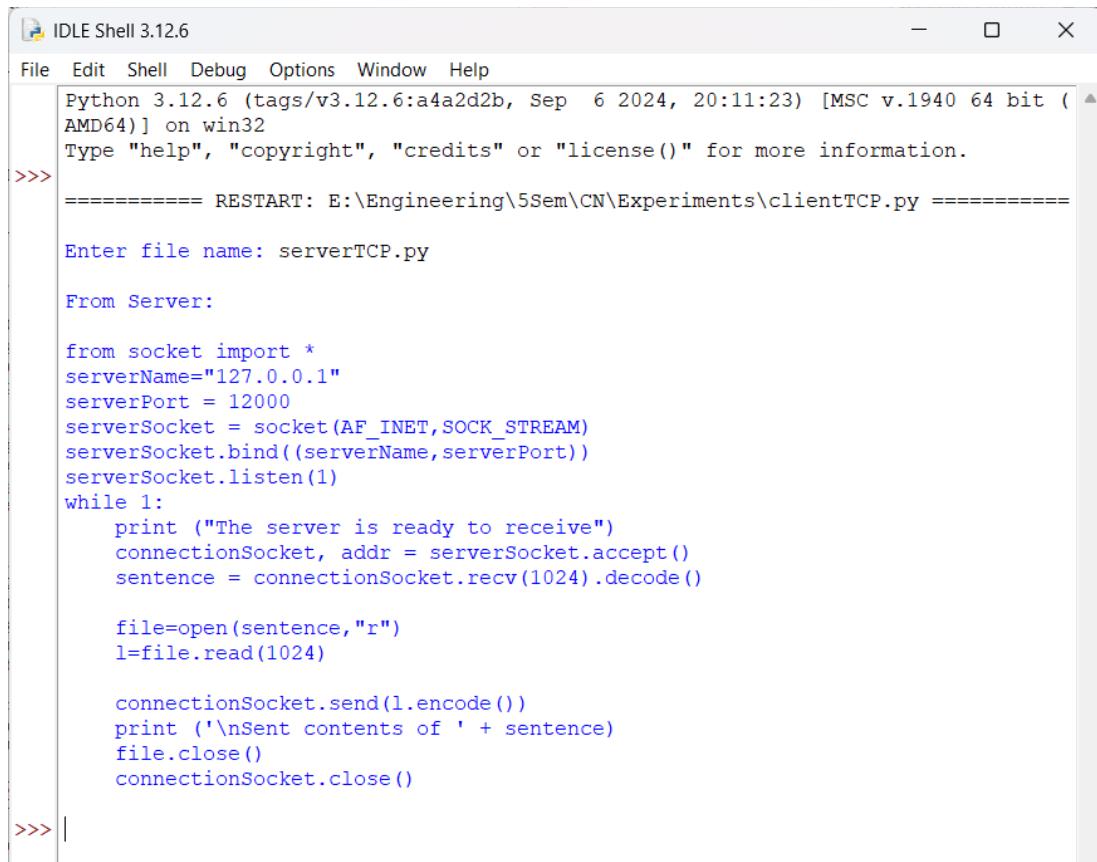
```
serverTCP.py
from socket import *
serverName="127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print ("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence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

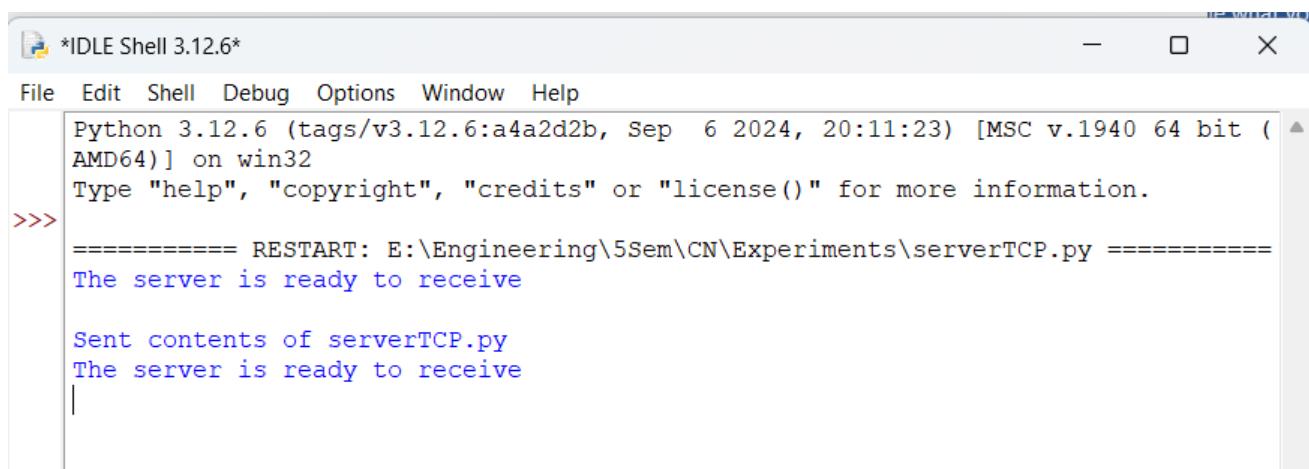
from socket import *
serverName="127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print ("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()

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

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

>>> |
```

Server

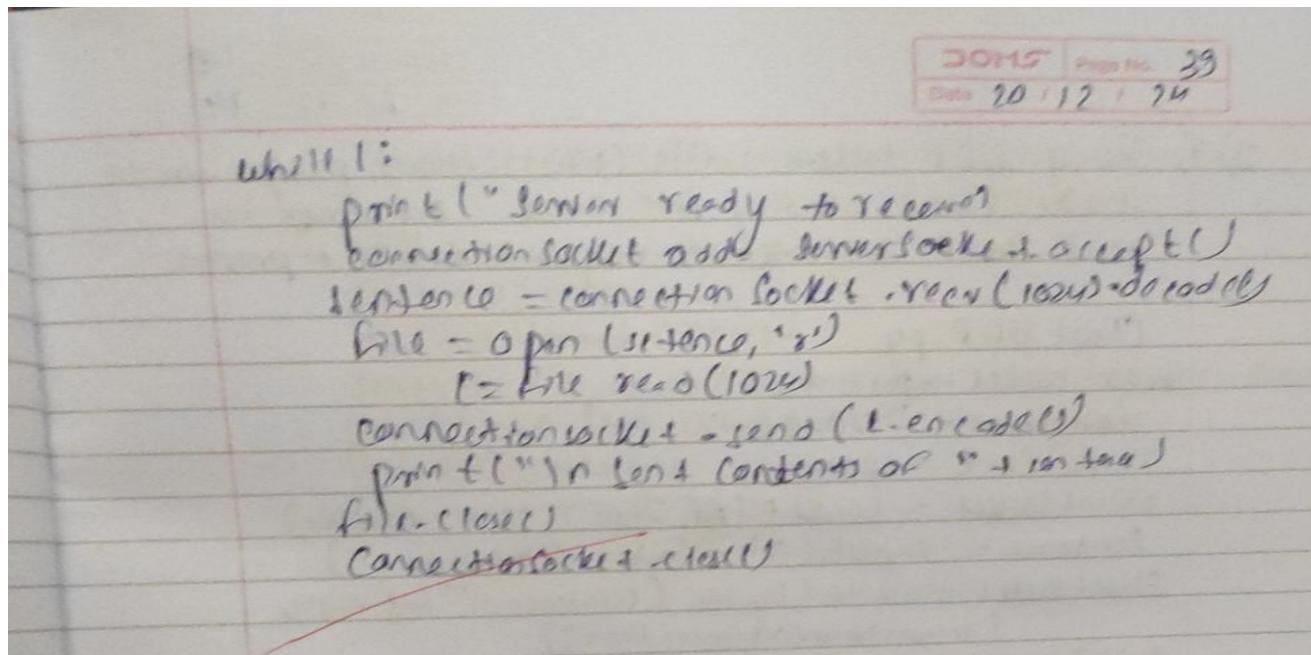
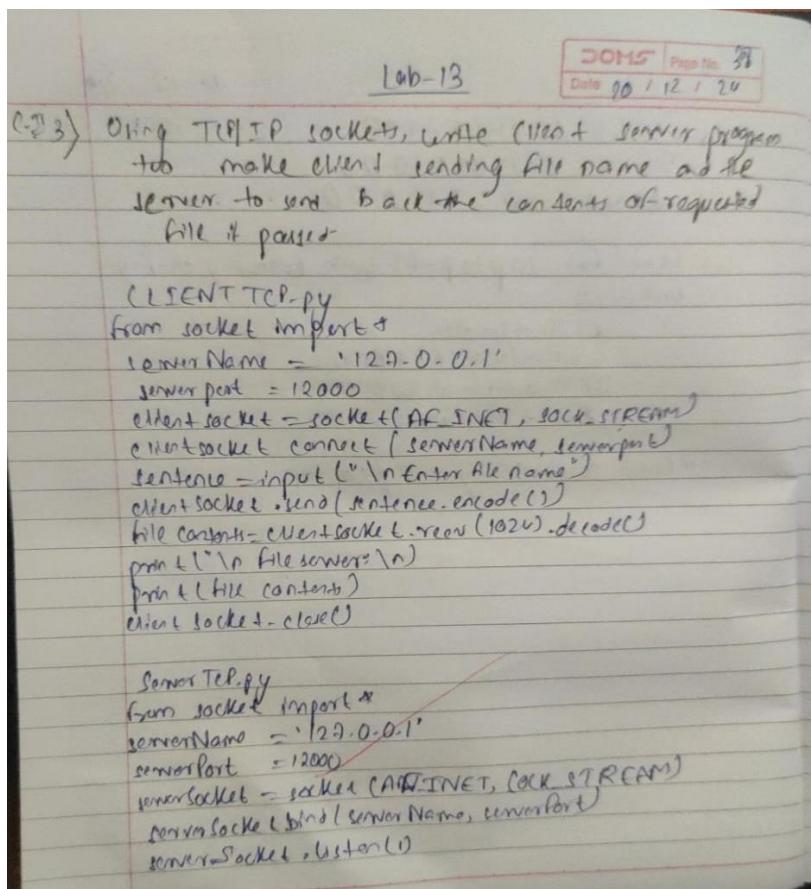


```
*IDLE Shell 3.12.6*
File Edit Shell Debug Options Window Help
Python 3.12.6 (tags/v3.12.6:a4a2d2b, Sep 6 2024, 20:11:23) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> ===== 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



## 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)
```

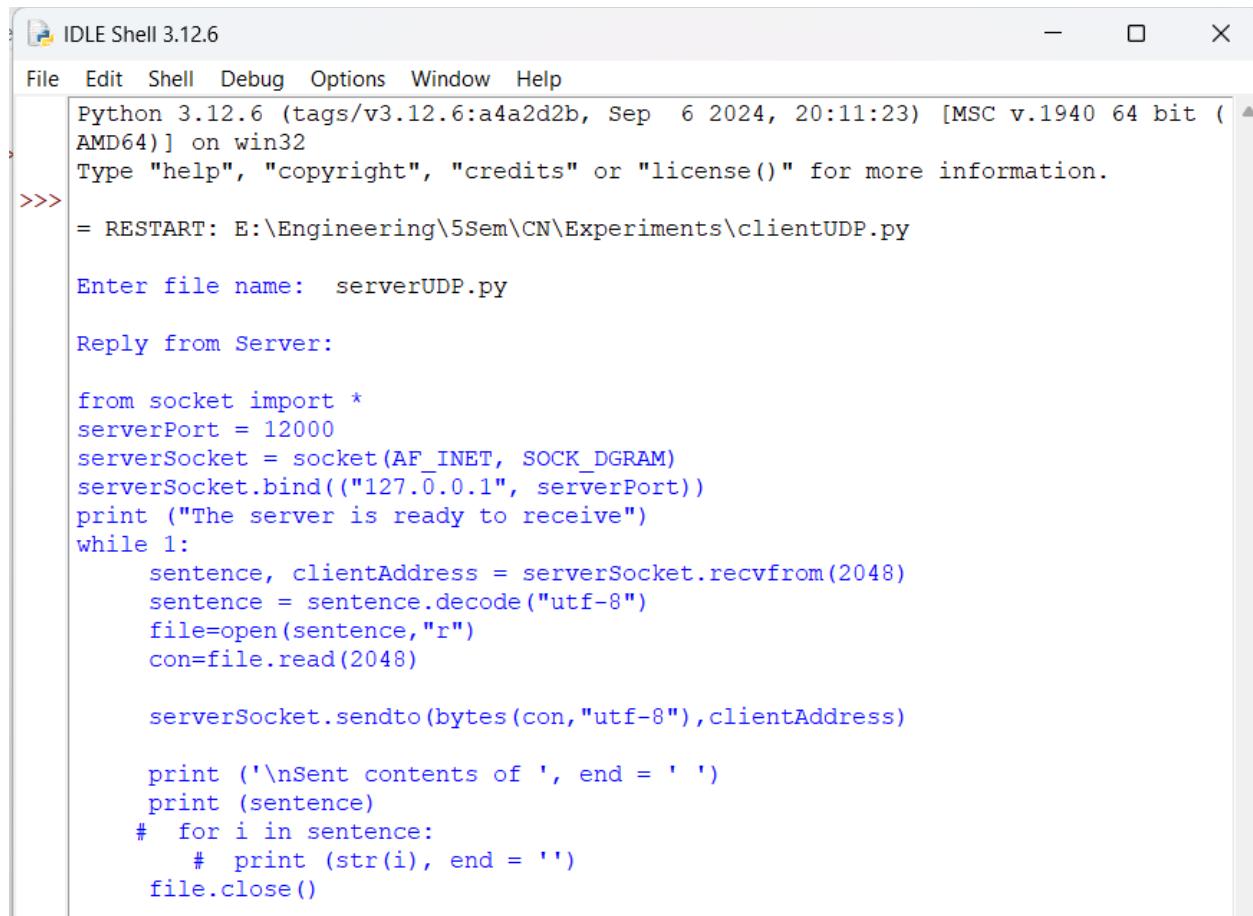
```
serverSocket.sendto(bytes(con,"utf-8"),clientAddress)
```

```
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

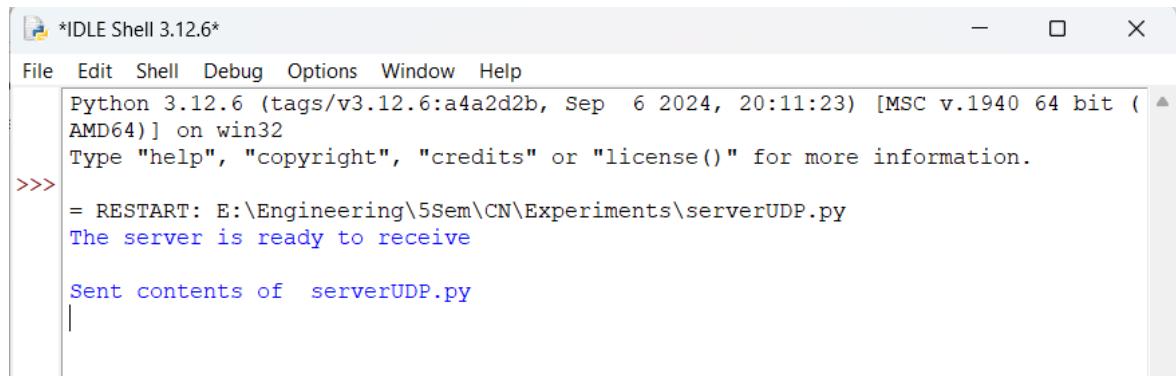
Reply from Server:

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

DOMS | Page No. 40  
Date 20/12/24

Q4) Using UDP sockets, write Client Server program to make Client sending file name and server to send back contents 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("\n Enter file Name :")
clientSocket.sendto(sentence.encode(), (serverName, serverPort))
fileContent, serverAddress = clientSocket.recvfrom(2048)
print("\n Reply from server :\n")
print(fileContent.decode("utf-8"))
for i in fileContent:
    print(str(i), end="")
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(2048)
    sentence = sentence.decode("utf-8")
    file = open(sentence, "r")
    content = file.read()
    file.close()
    print("Send content of :")
    print(content)
    for i in content:
        print(str(i), end="")

```

DOMS | Page No. 41  
Date 20/12/24

sentence, clientAddress = serverSocket.recvfrom(2048)

sentence = sentence.decode("utf-8")  
 file = open(sentence, "r")  
 content = file.read()  
 serverSocket.sendto(content.encode(), clientAddress)  
 print("Send content of :")  
 print(content)  
 for i in content:  
 print(str(i), end="")  
 file.close()

~~✓~~