

# **VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

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



**LAB RECORD**

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

*Submitted by*

**Shashank Patel C J (1BM22CS255)**

*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 **Shashank Patel C J (1BM22CS255)**, 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. Rekha G S 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.	1-4
2	18/10/2024	Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply	5-9
3	25/10/2024	Configure default route, static route to the router	10-15
4	08/11/2024	Configure DHCP within a LAN and outside LAN.	16-20
5	22/11/2024	Configure RIP routing Protocol in Routers	21-23
6	22/11/2024	Configure OSPF routing protocol	24-29
7	22/11/2024	Demonstrate the TTL/ Life of a Packet	30-32
8	08/11/2024	Configure Web Server, DNS within a LAN.	33-36
9	20/12/2024	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)	37-39
10	20/12/2024	To understand the operation of TELNET by accessing the router in server room from a PC in IT office.	40-42
11	20/12/2024	To construct a VLAN and make the PC's communicate among a VLAN	43-45
12	20/12/2024	To construct a WLAN and make the nodes communicate wirelessly	46-49

## **Index-Cycle-II**

<b>Sl. No.</b>	<b>Date</b>	<b>Experiment Title</b>	<b>Page No.</b>
1	15/11/2024	Write a program for error detecting code using CRC-CCITT (16-bits).	50-53
2	15/11/2024	Write a program for congestion control using Leaky bucket algorithm	54-55
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.	56-58
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.	59-61

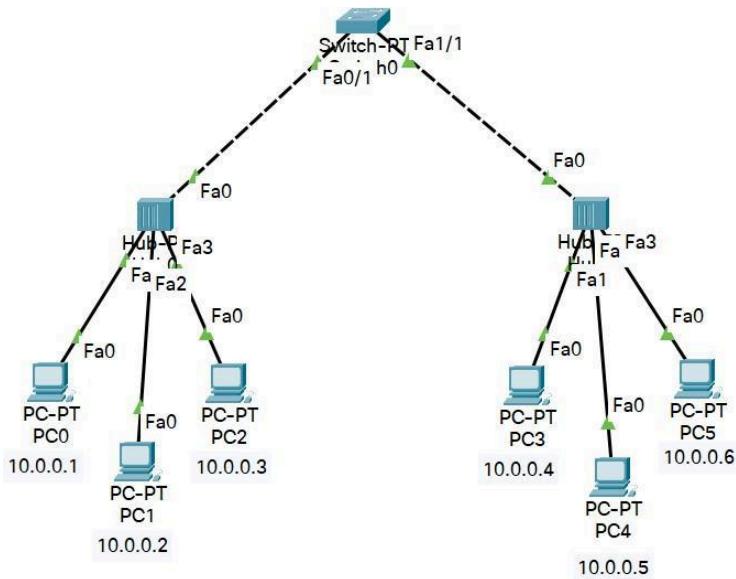
Github Link:

<https://github.com/SP212004/CN>

# Cycle-I

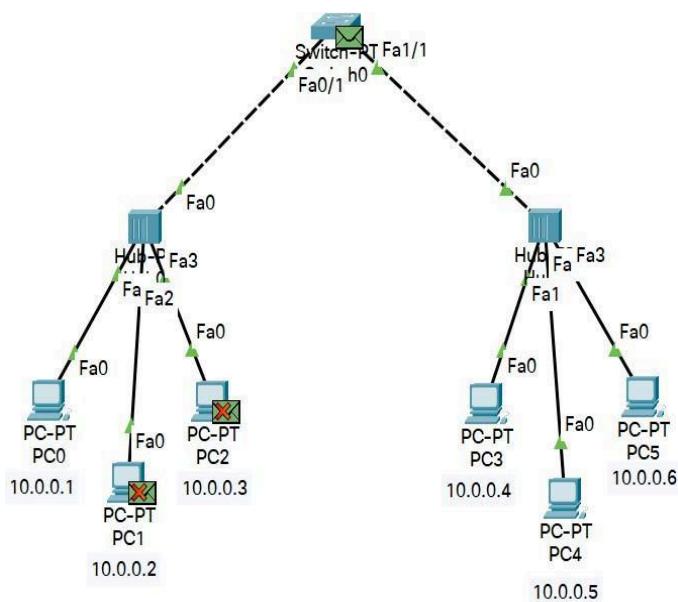
## Program 1

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

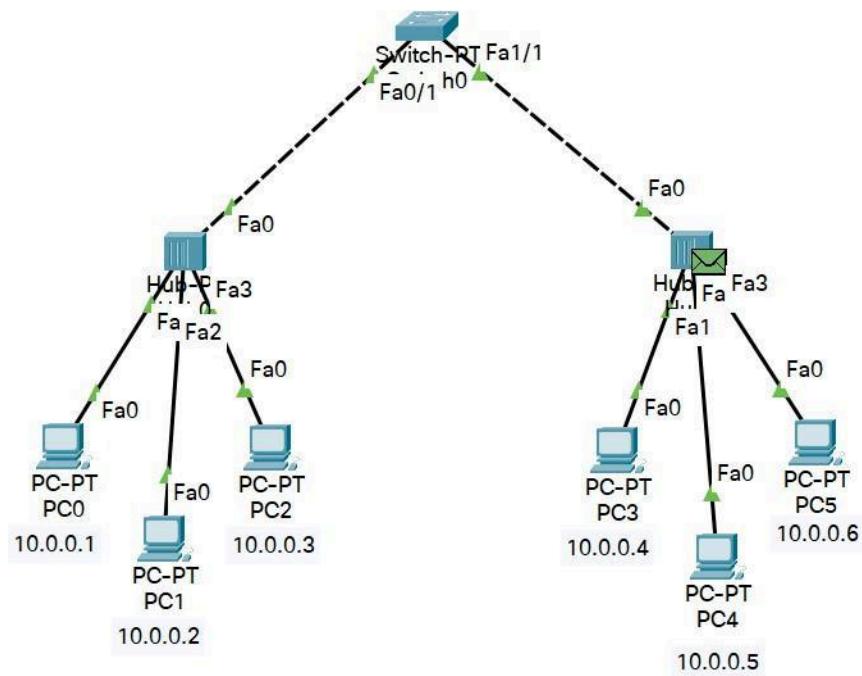


- iii. Screen shots/ output

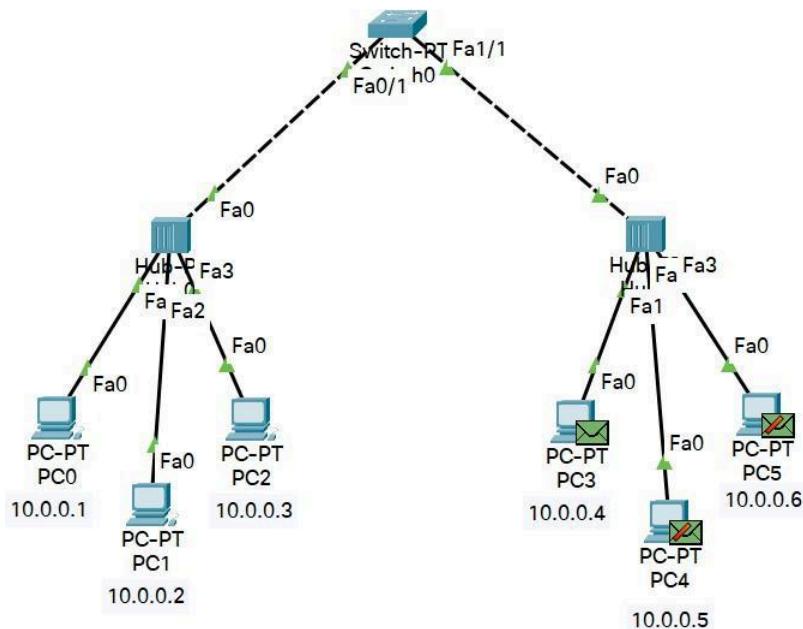
Hub behaviour at sending end



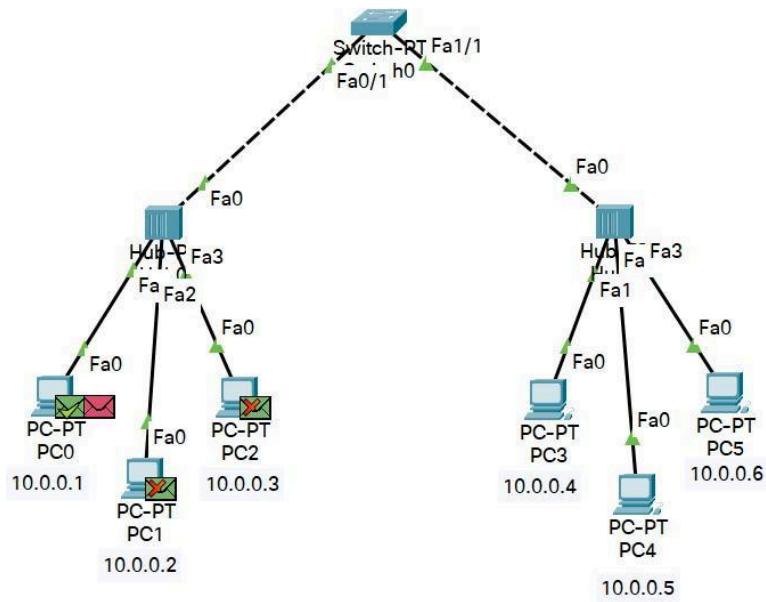
## Switch behaviour



## Hub behaviour at receiving end



## Hub behaviour when back to sender



Ping command to connectivity

Screenshot of the **PC0** window in Packet Tracer, showing the **Desktop** tab selected. The **Command Prompt** window displays the following output:

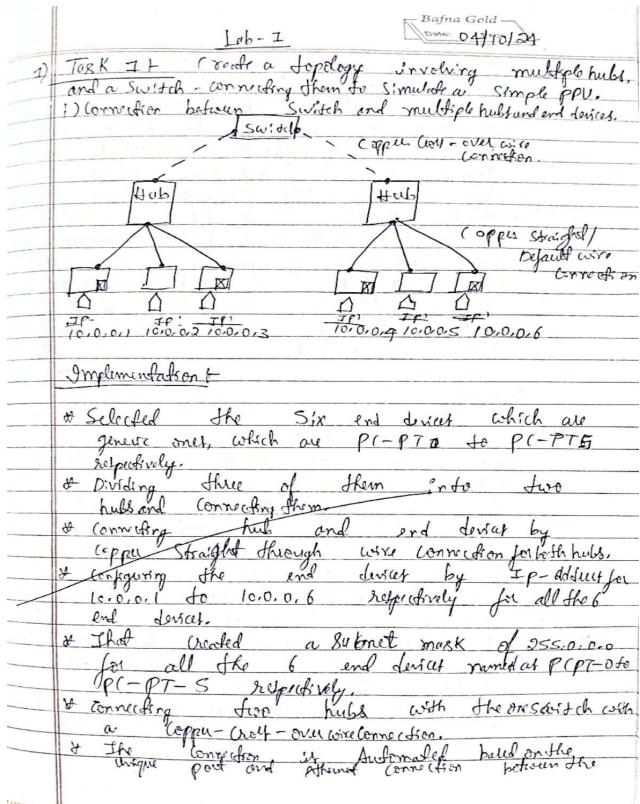
```
Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.4

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:\>
```

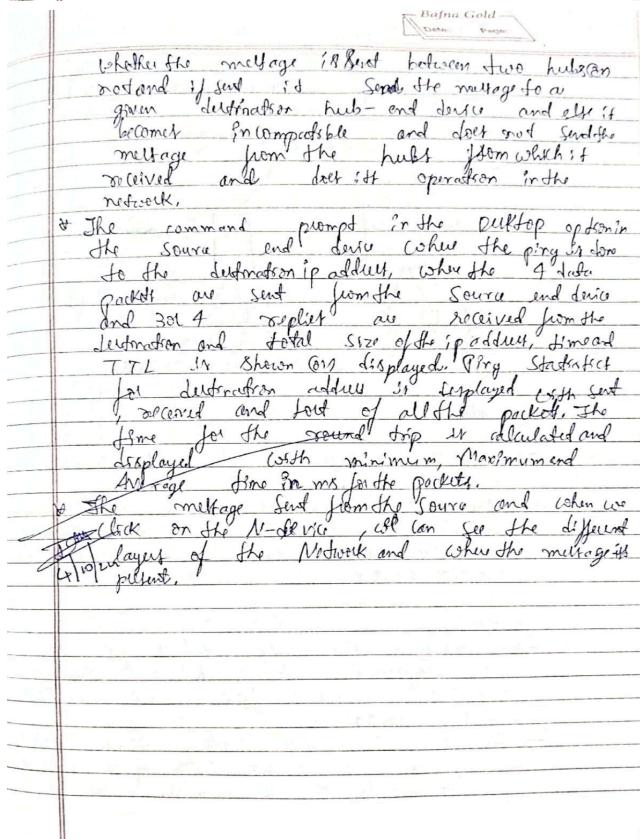
#### iv. Observation



- \* A message (Simple P2P) was passed from PC-PT-0 to PC-PT-2 in hub-1 and PC-PT-3 to PC-PT-5 in hub-2 in the forward to switch and in this network.
- \* After the successful message sent, we need to check for the source end device and go to desktop option, then selecting the command prompt and entering a command of ping destination where TTL (Time to live) for the end device will until it sends the 4 data packets and all are received by the destination PC and returns back.

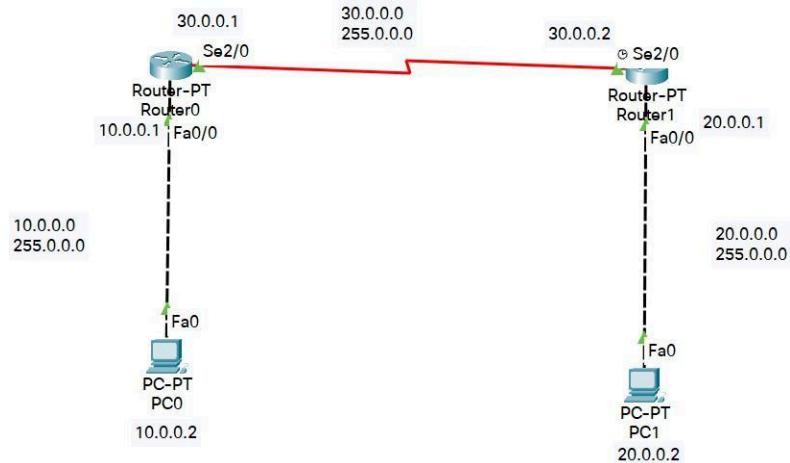
#### Observation -

- \* It is found that communication forms a star topology between all the N-devices in the network.
- \* A given signal at the bottom end is seen which indicates the successful connection.
- \* The star was successful and type of layer with ICMP for both the messages which were sent and the messages are received by the destination end devices.
- \* The hub broadcast the data to all the connected devices in the network and switch identify the MAC media access control address on the data header to forward the data only to the device that requested it, so here the switch checks



## **Program 2**

- i. Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply.
- 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

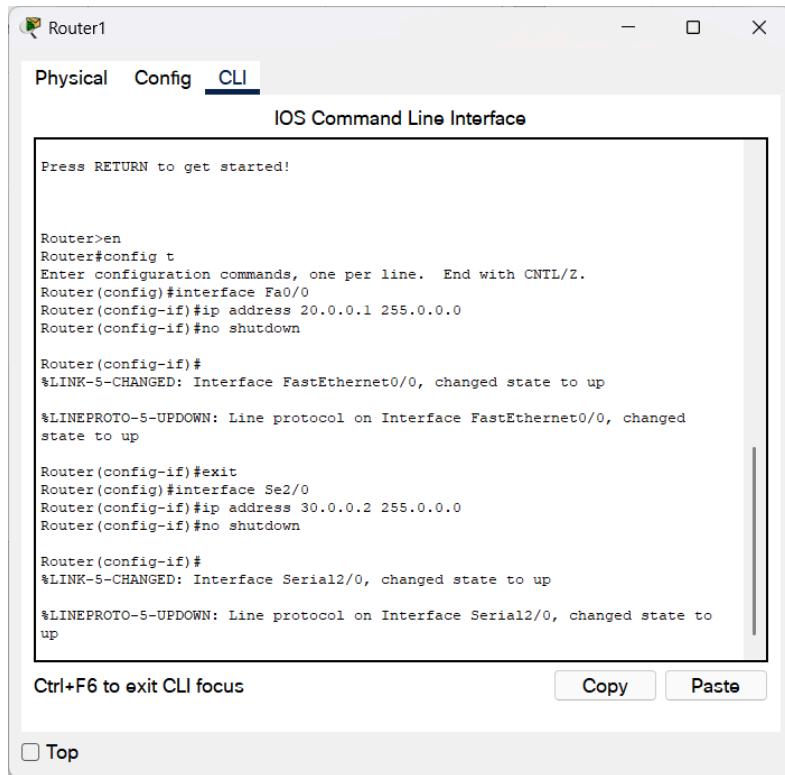
%LINK-5-CHANGED: Interface Serial2/0, changed state to down
Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

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

Ctrl+F6 to exit CLI focus     

Top

## Router1 configuration



The screenshot shows a Windows-style application window titled "Router1". The tab bar at the top has three tabs: "Physical", "Config", and "CLI", with "CLI" being the active tab. Below the tabs is a title bar "IOS Command Line Interface". 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". A checkbox labeled "Top" is also present.

## 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)#

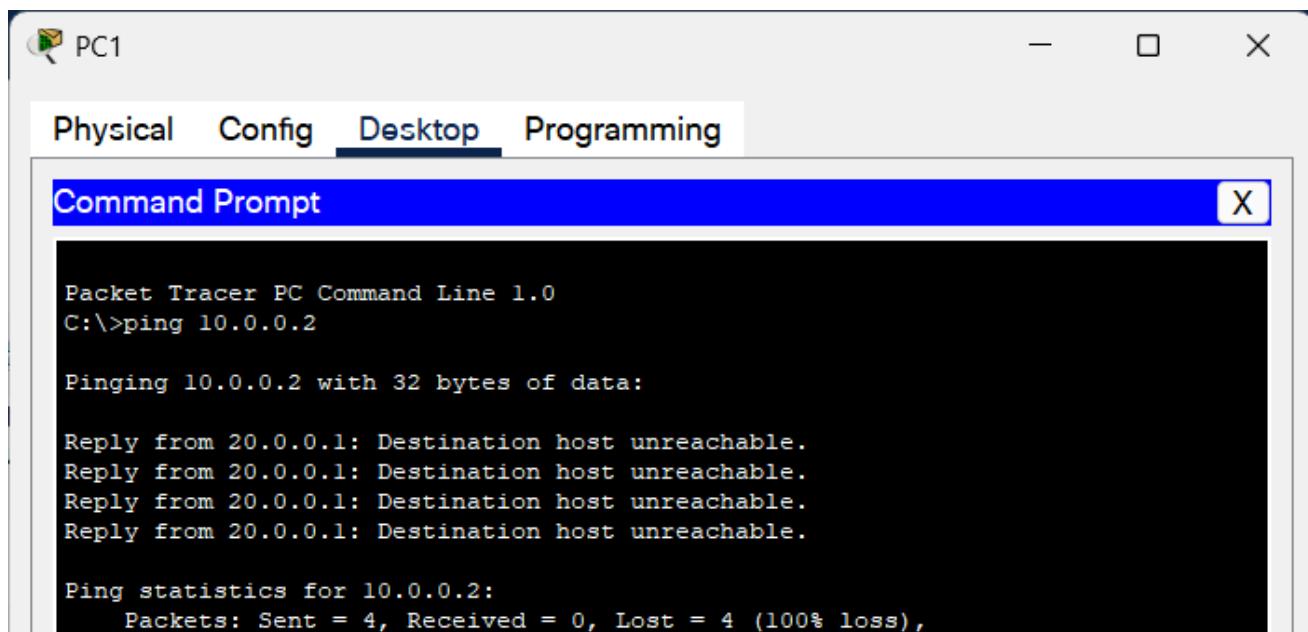
```

## 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

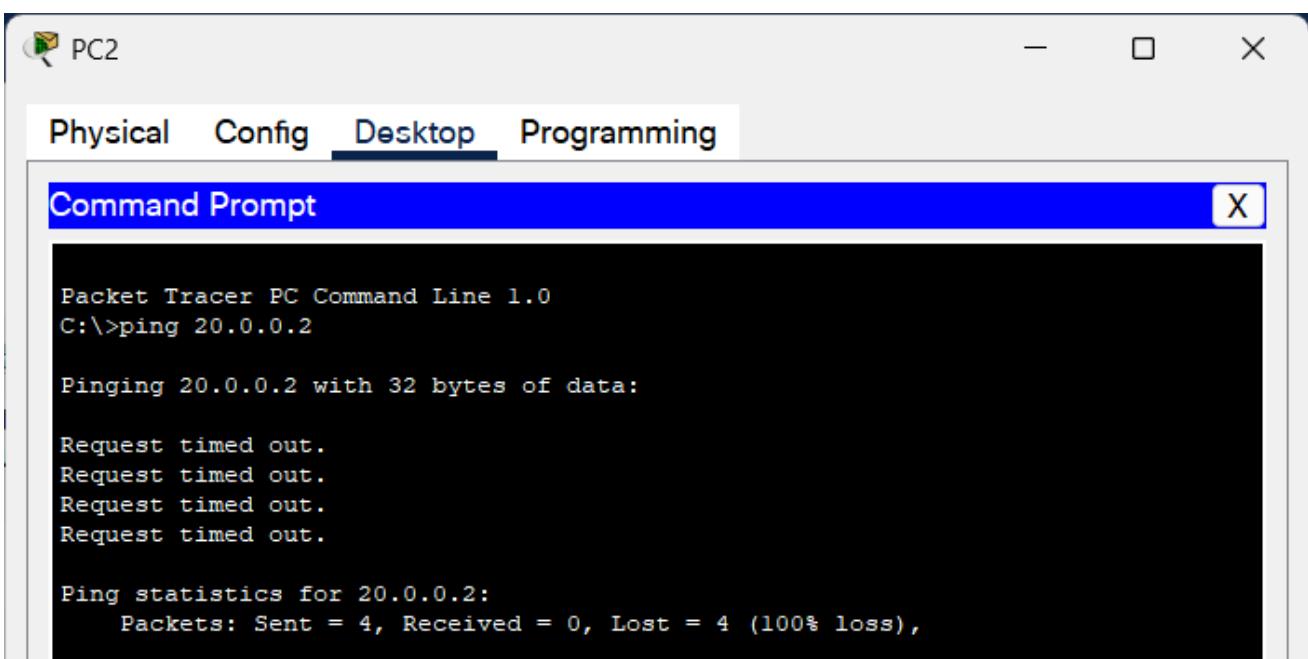
```
Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.2

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

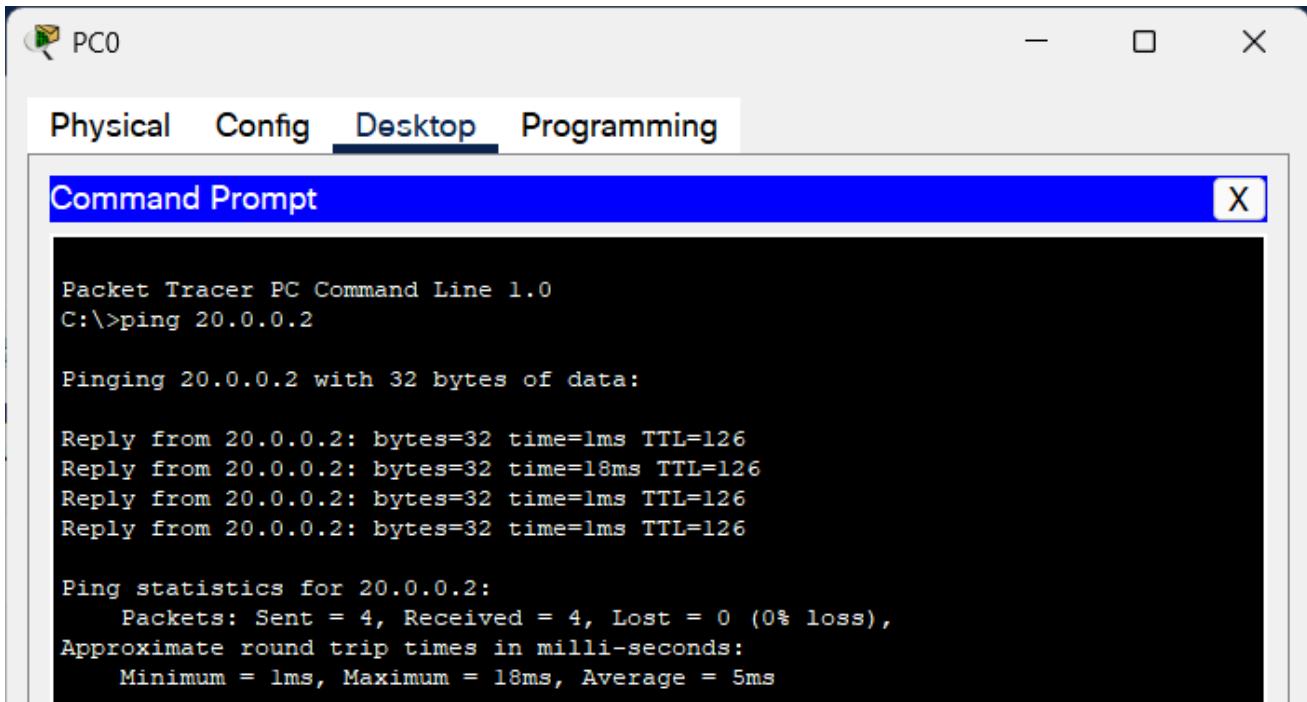
```
Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.2

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

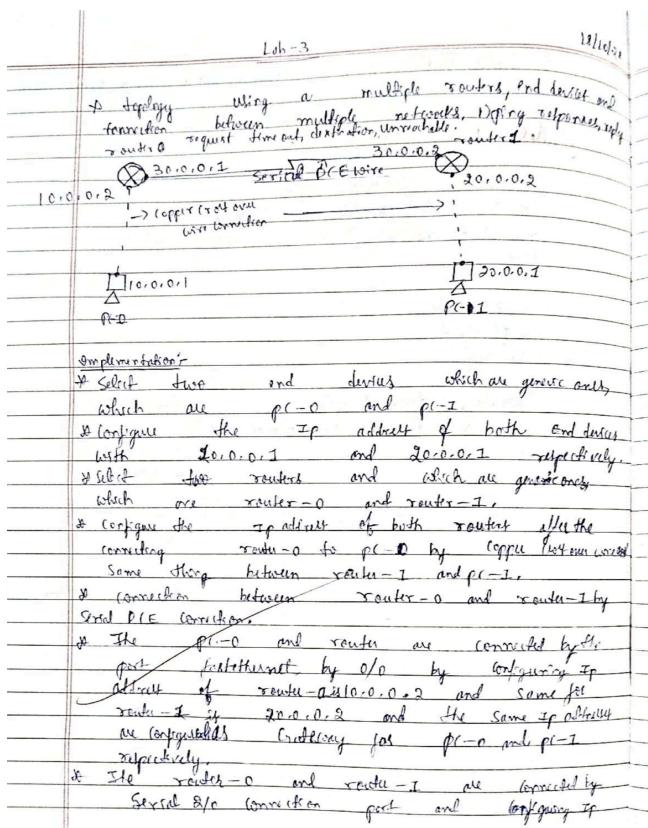
```
Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.2

Pinging 20.0.0.2 with 32 bytes of data:

Reply from 20.0.0.2: bytes=32 time=lms TTL=126
Reply from 20.0.0.2: bytes=32 time=18ms TTL=126
Reply from 20.0.0.2: bytes=32 time=lms TTL=126
Reply from 20.0.0.2: bytes=32 time=lms 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 = lms, Maximum = 18ms, Average = 5ms
```

#### iv. Observation



Bafna Gold  
Date: 11/10/21

address as 30.0.0.1 and 30.0.0.2 respectively.

~~Implementation~~

- # The message (Simple PDU) is sent between PC-0 and PC-1 but it didn't failed because of two different networks.
- # Now to get successfully first open CLI command type ip route 30.0.0.0 255.0.0.0 30.0.0.2 for router-0 and 10.0.0.0 255.0.0.0 30.0.0.1 for router-1 respectively.
- # After establishing connection, send from PC-0 and the message is successfully sent to PC-1.
- # Thus establishing the common connection between the two networks.

Request timed out

Request timed out

Request timed out

Request timed out

ping 30.0.0.1

pinging 30.0.0.1 with 32 bytes of data

Request: Send = 9 ; Receive = 0, Lost = 0 (0% Loss).

destination

ping 30.0.0.1

pinging 30.0.0.1 with 32 bytes of data

Reply from 30.0.0.1 : bytes = 32

ping 30.0.0.1 -

Request: Send = 9 ; Receive = 1, Lost = 0 (0% Loss)

unreachable

ping 30.0.0.1

ping 30.0.0.1 with 32 bytes of data

Reply from 30.0.0.1: Destination not reached

10/10/21

show ip route

CLI Command

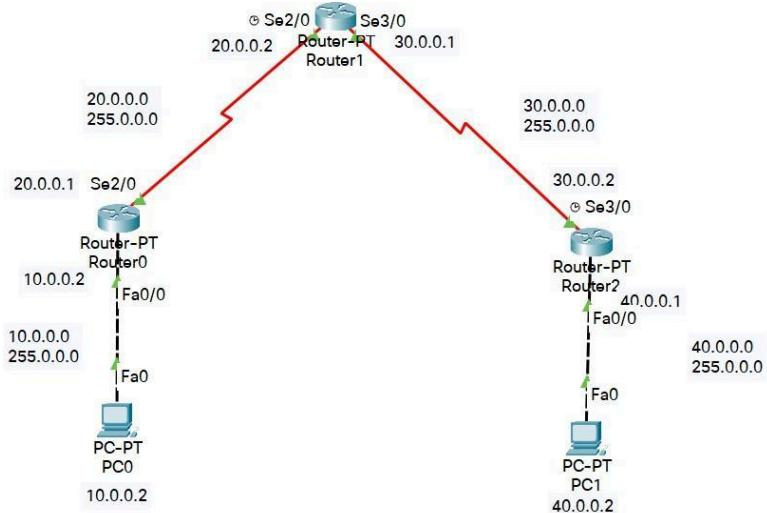
```
root# show ip route
code: r - Connected, S - Static, I - IGRP, R - RIP, H - mobile,
      B - BGP, D - EIGRP, E - OSPF External, O - OSPF,
      IA - OSPF Inter area,
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA
      internal type 2,
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, L1 - IS-IS L1, L2 - IS-IS L2, L3 - IS-IS
      P - Point-to-point, P2P - Multi-point Static routes, O - ODR
      P - periodic downloaded static route,
```

ip route	20.0.0.0	255.0.0.0	30.0.0.2
ip route	10.0.0.0	255.0.0.0	30.0.0.1

10/10/21

### **Program 3**

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



- iii. Screen  
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
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

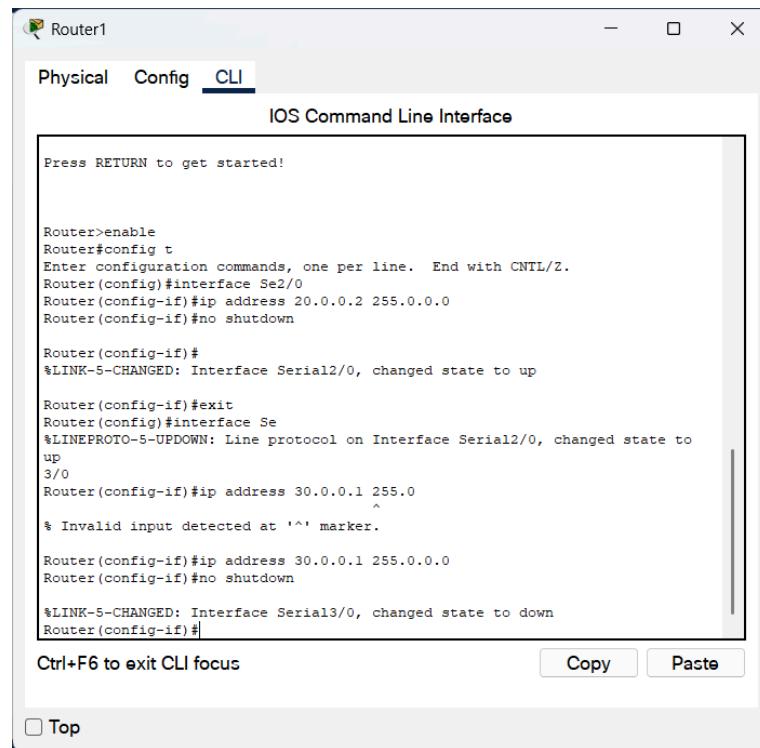
$LINK-5-CHANGED: Interface Serial2/0, changed state to down
Router(config-if)#

Ctrl+F6 to exit CLI focus
```

Top

Copy Paste

## Router1 configuration



The screenshot shows the Router1 CLI window. The title bar says "Router1". 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 commands:

```
Press RETURN to get started!

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

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

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

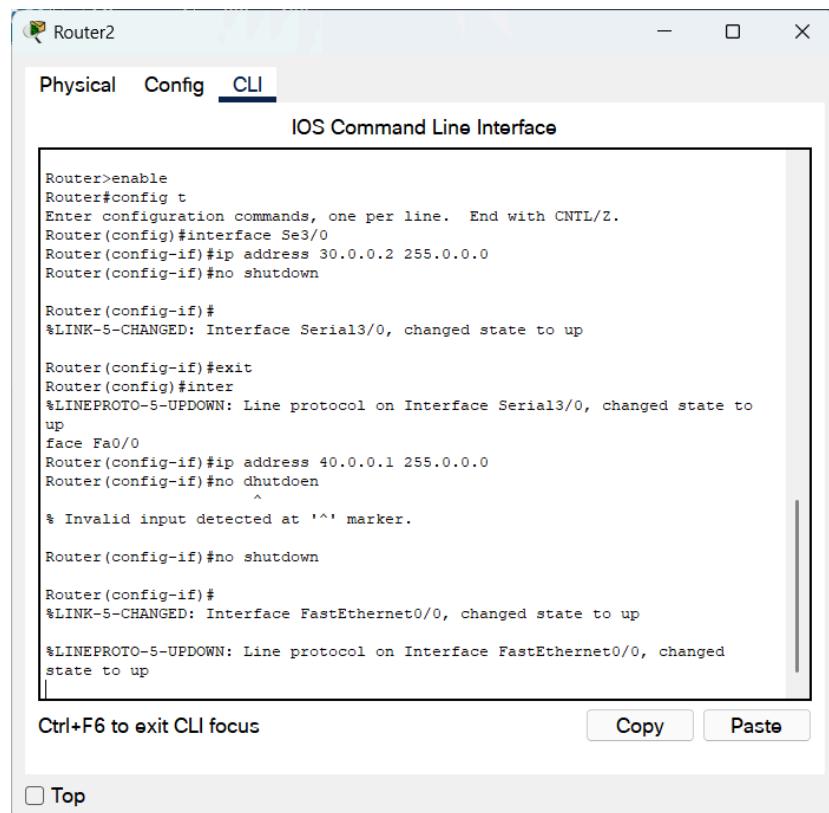
Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#no shutdown

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

Ctrl+F6 to exit CLI focus
```

At the bottom right of the main window are "Copy" and "Paste" buttons. At the bottom left is a checkbox labeled "Top".

## Router2 configuration



The screenshot shows the Router2 CLI window. 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 commands:

```
Press RETURN to get started!

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

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

Router(config-if)#exit
Router(config)#inter
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to
up
face Fa0/0
Router(config-if)#ip address 40.0.0.1 255.0.0.0
Router(config-if)#no 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

Ctrl+F6 to exit CLI focus
```

At the bottom right of the main window are "Copy" and "Paste" buttons. At the bottom left is a checkbox labeled "Top".

## Static Routing:

Router0

Router>enable  
Router#config t  
Enter configuration commands, one per line. End with CNTL/Z.  
Router(config)#ip route 30.0.0.0 255.0.0.0 20.0.0.2  
Router(config)#ip route 40.0.0.0 255.0.0.0 20.0.0.2  
Router(config)#exit  
Router#  
%SYS-5-CONFIG\_I: Configured from console by console  
  
Router#show ip route  
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP  
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area  
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2  
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP  
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter  
area  
\* - candidate default, U - per-user static route, o - ODR  
P - periodic downloaded static route  
  
Gateway of last resort is not set  
  
C 10.0.0.0/8 is directly connected, FastEthernet0/0  
C 20.0.0.0/8 is directly connected, Serial2/0  
S 30.0.0.0/8 [1/0] via 20.0.0.2  
S 40.0.0.0/8 [1/0] via 20.0.0.2  
  
Router#

Ctrl+F6 to exit CLI focus     

Top

Router1

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

Ctrl+F6 to exit CLI focus     

Top

## Router2

Router>enable  
Router#config t  
Enter configuration commands, one per line. End with CNTL/Z.  
Router(config)#ip route 10.0.0.0 255.0.0.0 30.0.0.1  
Router(config)#ip route 20.0.0.0 255.0.0.0 30.0.0.1  
Router(config)#exit  
Router#  
\$SYS-5-CONFIG\_I: Configured from console by console

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

Gateway of last resort is not set

S   10.0.0.0/8 [1/0] via 30.0.0.1
S   20.0.0.0/8 [1/0] via 30.0.0.1
C   30.0.0.0/8 is directly connected, Serial3/0
C   40.0.0.0/8 is directly connected, FastEthernet0/0

Router#
```

Ctrl+F6 to exit CLI focus     

Top

## Dynamic Routing:

### Route0

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

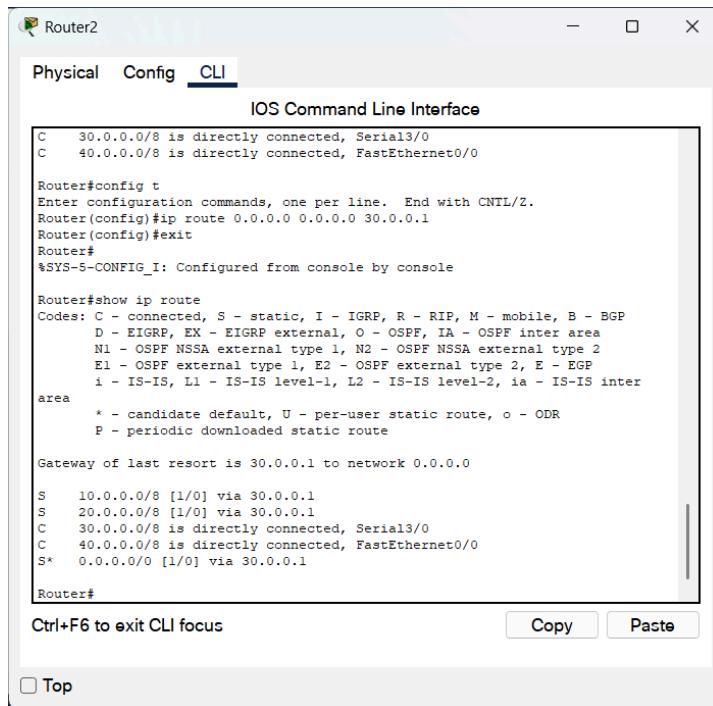
C   10.0.0.0/8 is directly connected, FastEthernet0/0
C   20.0.0.0/8 is directly connected, Serial2/0
S   30.0.0.0/8 [1/0] via 20.0.0.2
S   40.0.0.0/8 [1/0] via 20.0.0.2
S*  0.0.0.0/0 [1/0] via 20.0.0.2

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 output:

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

Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 0.0.0.0 0.0.0.0 30.0.0.1
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

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

Gateway of last resort is 30.0.0.1 to network 0.0.0.0

S 10.0.0.0/8 [1/0] via 30.0.0.1
S 20.0.0.0/8 [1/0] via 30.0.0.1
C 30.0.0.0/8 is directly connected, Serial3/0
C 40.0.0.0/8 is directly connected, FastEthernet0/0
S* 0.0.0.0/0 [1/0] via 30.0.0.1

Router#
```

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

Pinging:

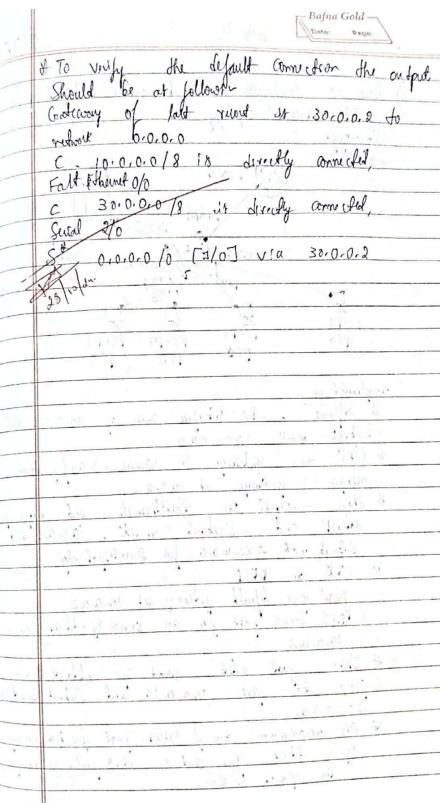
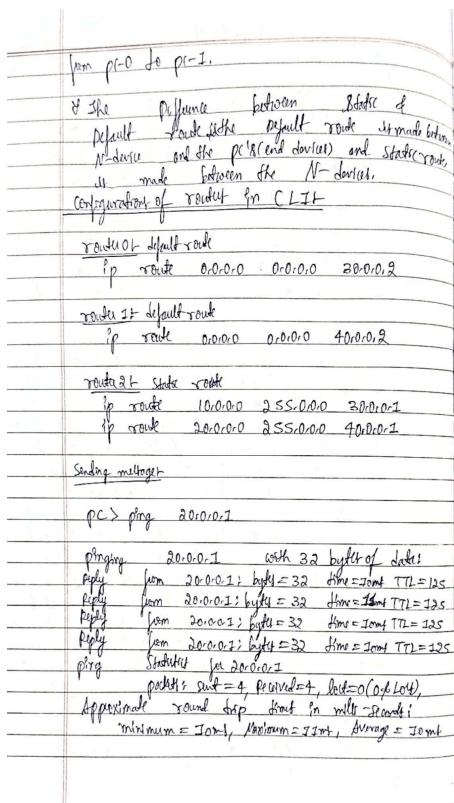
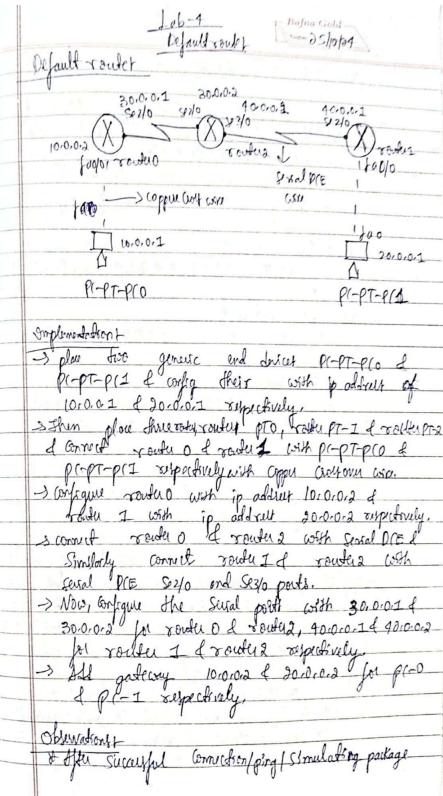
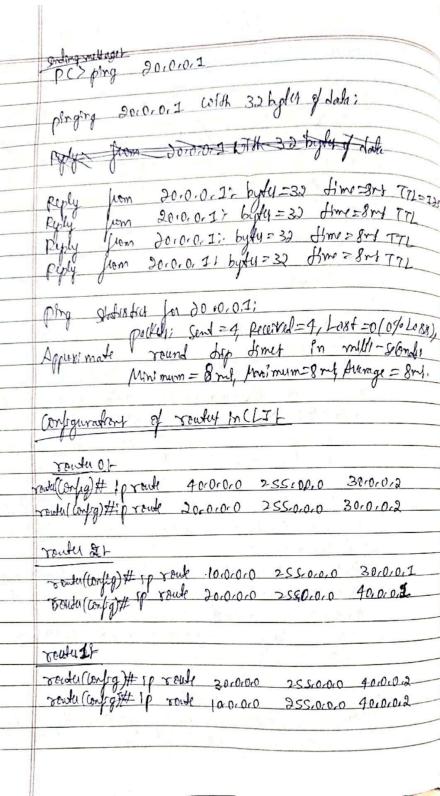
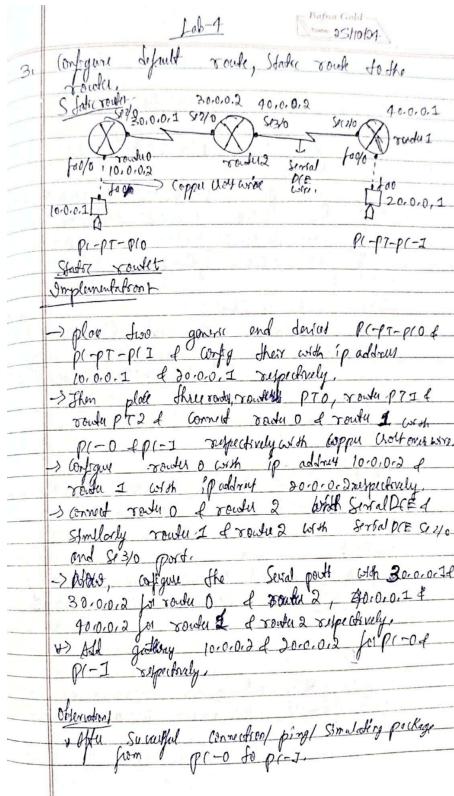
```
C:\>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

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

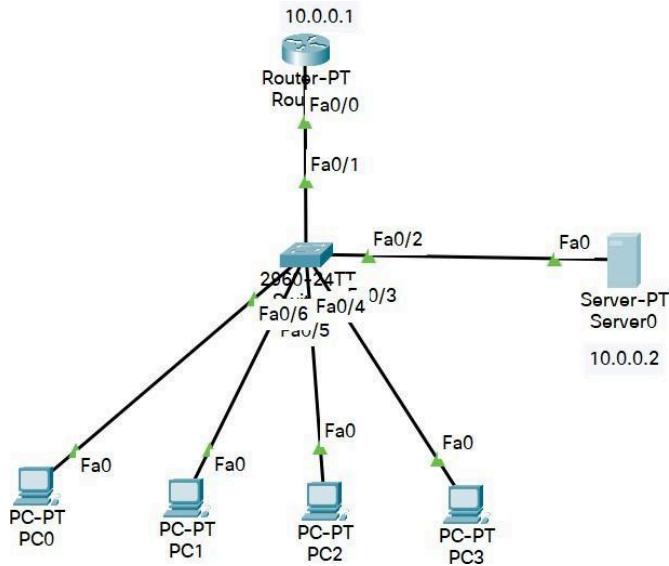
Ping statistics for 40.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 25ms, Average = 16ms
```

#### iv. Observation



## Program 4

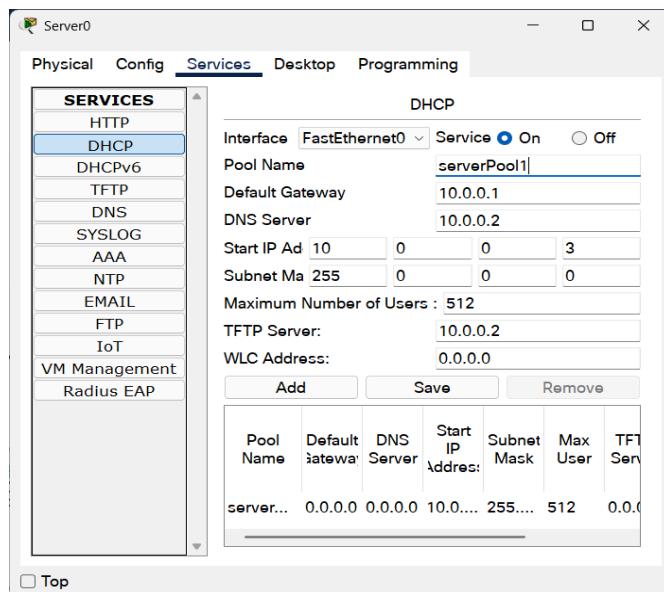
- i. Configure DHCP within a LAN and outside LAN.
- ii. Procedure along with the topology



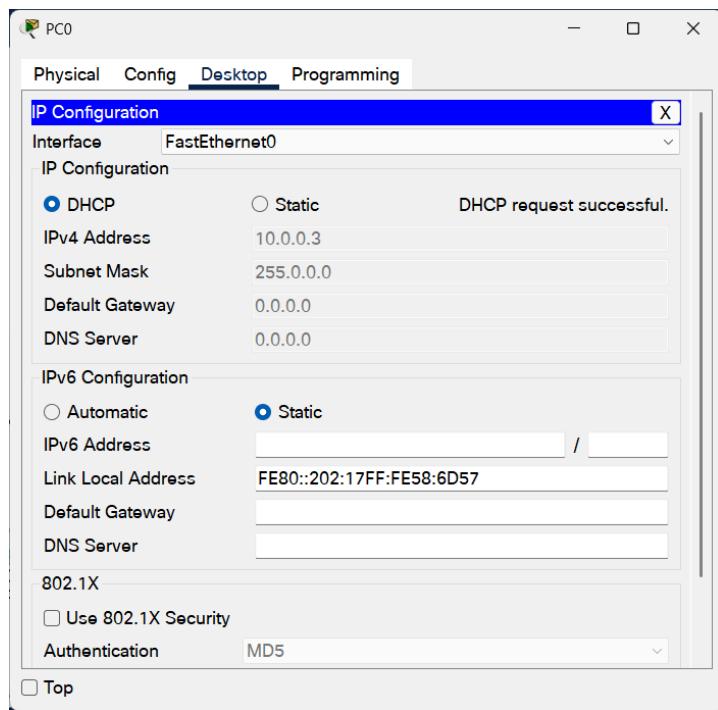
- iii. Screen shots/ output

DHCP Within LAN

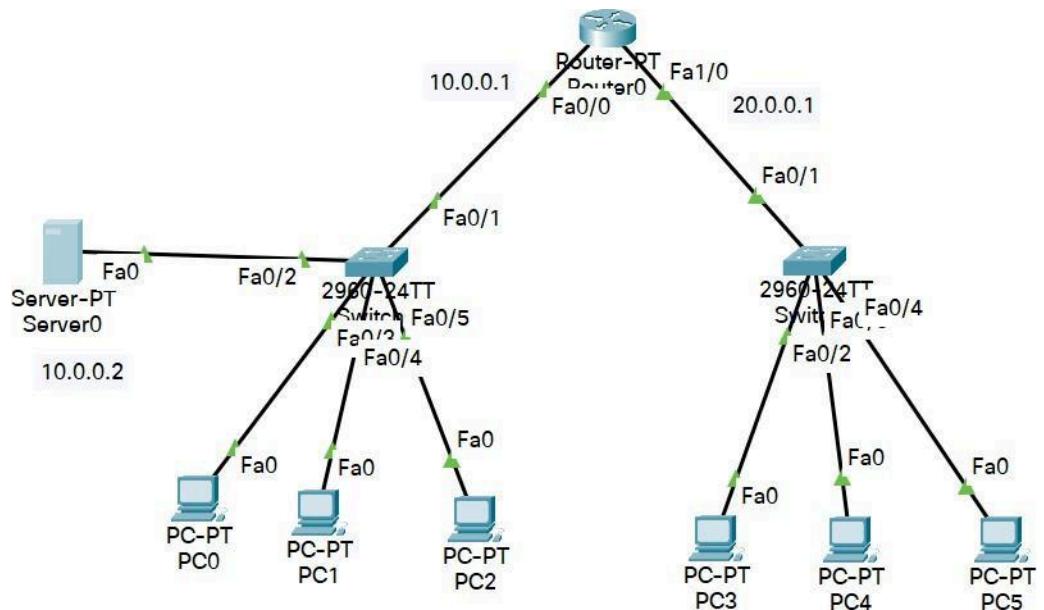
DHCP Configuration



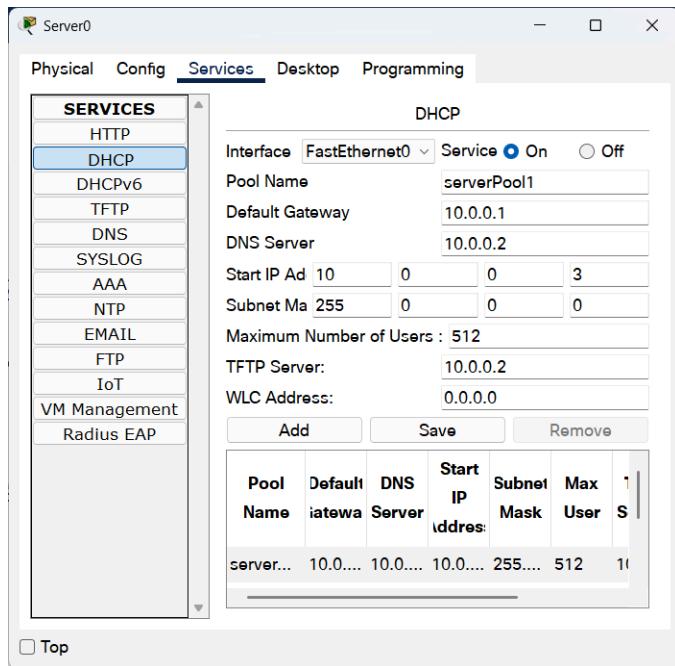
## PC settings



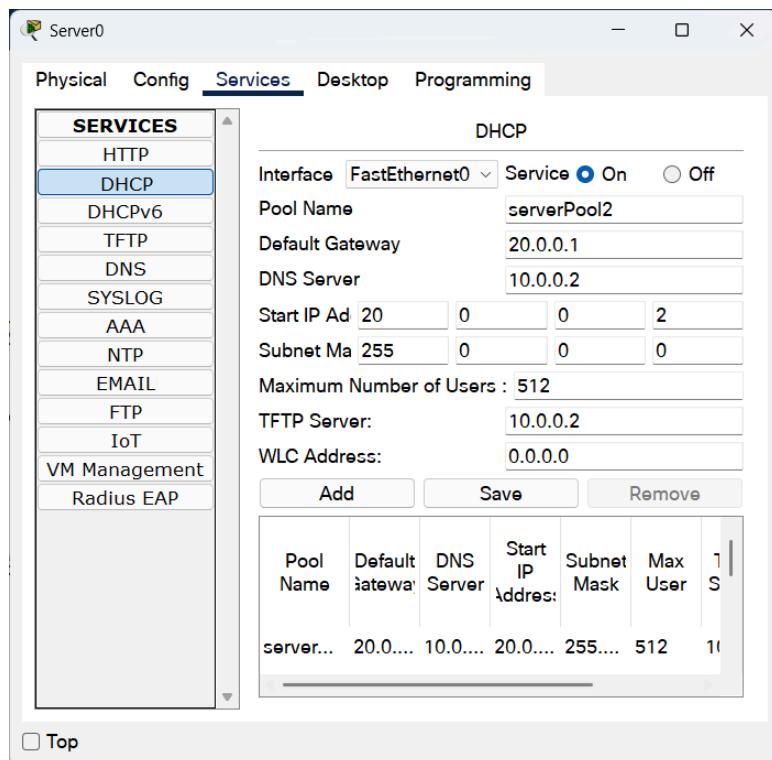
DHCP outside LAN:



## DHCP configuration for inside LAN



## DHCP configuration for outside LAN

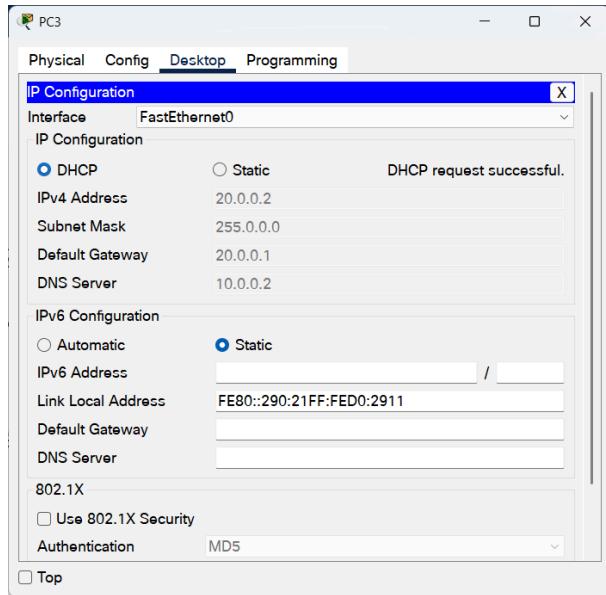


## Ip helper command in Router

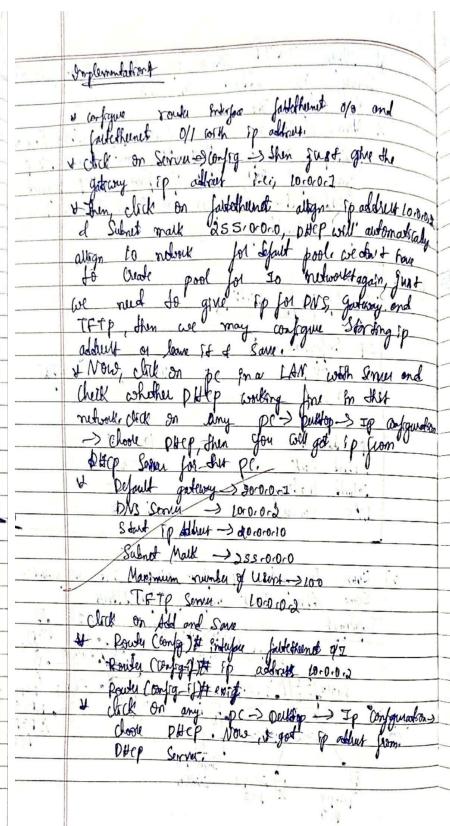
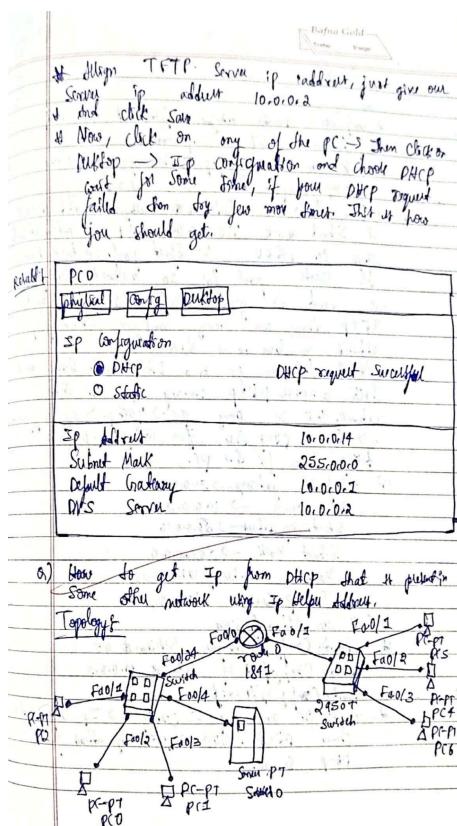
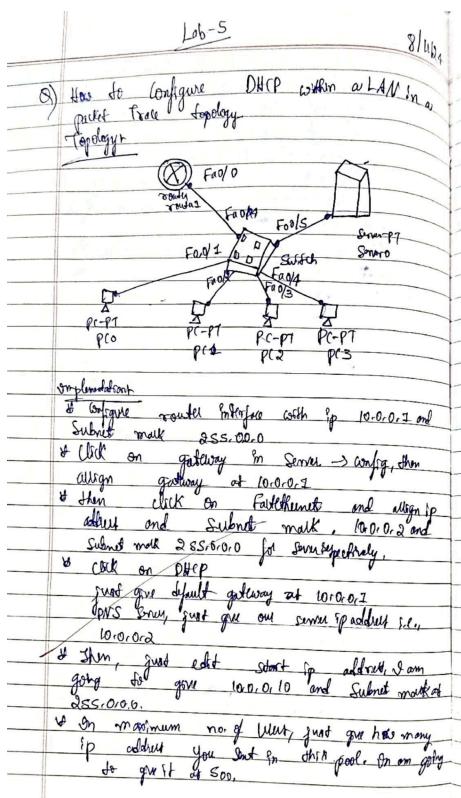
```
Router(config-if)#exit
Router(config)#interface Fa1/0
Router(config-if)#ip helper-address 10.0.0.2
Router(config-if)#

```

## PC setting in another network

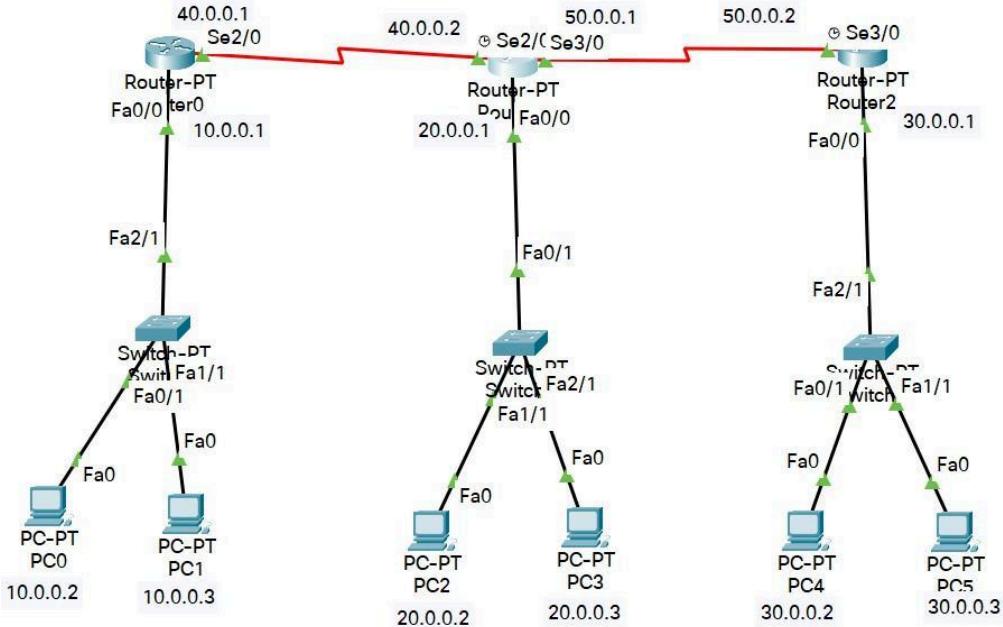


#### iv. Observation



## 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:

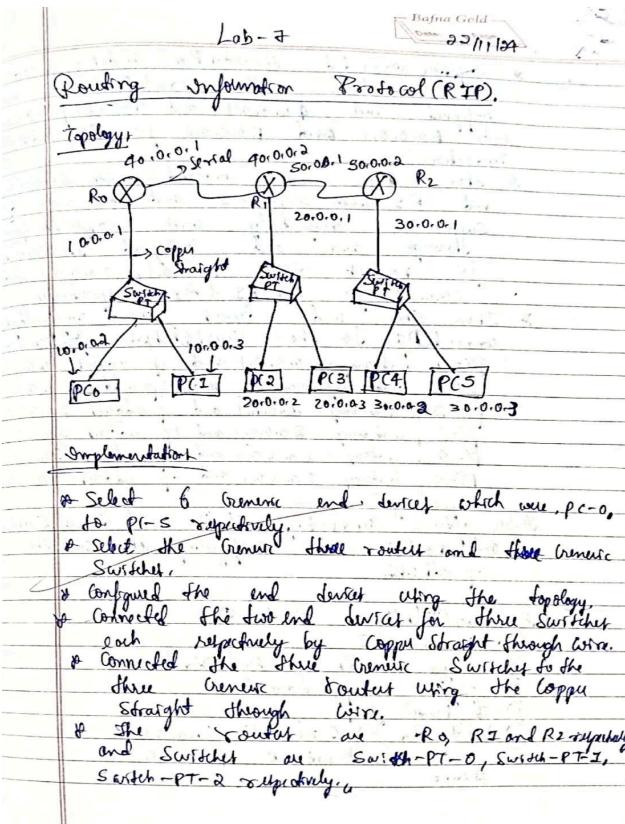
```
Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.2

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



**Routefit Show ip route.**

For R0 → # route print for exist routers R1 & R2.  
# network 20.0.0.0  
# network 30.0.0.0  
# network 50.0.0.0  
# exit

**Routefit Show ip route**  
# route rip.

For R1 → # network 30.0.0.0  
# network 50.0.0.0  
# exit

**Routefit Show ip route.**

For R2 → Show ip route (route 2).  
R 10.0.0.0/8 [120/2] via 50.0.0.1, 00:00:25, Serial 2/0  
R 20.0.0.0/8 [120/2] via 50.0.0.1, 00:00:25, Serial 2/0  
C 30.0.0.0/8 is directly connected, FastEthernet 0/0  
R 40.0.0.0/8 [120/2] via 50.0.0.2, 00:00:25, Serial 2/0  
C 50.0.0.0/8 is directly connected, Serial 2/0  
like show route 0 and route 1, same five connections.

# send Simple PDU (message) from PC0 to PC4 and observe the observations.

**Observations:**

- After sending the packet from PC0 to PC4, the packet is received at PC4 and the acknowledgement is sent to the PC0.
- The status of the message is successful using this protocol.
- We can do this process using ping also.

\* Configuration connected three routers R0 and R1 and R2 with serial ports were and configured with 10.0.0.1 and 20.0.0.2 for Router R0 and 30.0.0.1 and 50.0.0.2 for R1 and R2 connection.

\* Then, R0 is connected with Switch-0 and R2 with Switch-2 respectively with copper straight through wire and configured the connection for router with 10.0.0.1, 20.0.0.1 and 30.0.0.1 for R0, R1 & R2 respectively.

\* Then, the end device are configured connecting to the switch with port numbers, PC-0, gateway - 10.0.0.1 and ip address 10.0.0.2, PC-1, gateway - 20.0.0.1 and ip address 10.0.0.3, PC-2, gateway - 20.0.0.1 and ip address 20.0.0.3, PC-3, gateway - 30.0.0.1 and ip address 30.0.0.3, PC-4, gateway - 30.0.0.1 and ip address 30.0.0.4, PC-5, gateway - 30.0.0.1 and ip address 30.0.0.5 respectively.

\* So, after connecting all the end device, switches and routers and configuring them with the IP addresses.

\* Now Click on route ①

- Go to CLI
- Type enable → Route Enable
- Type Config terminal
- # route rip
- (# for R0) → # network 10.0.0.0:  
# network 40.0.0.0
- # exit

\* After on PC0  
\* Go to MikroTik  
\* Click on Command prompt and type ping command,  
\* PC> ping 30.0.0.2

Pinging 30.0.0.2 with 82 bytes of data:

Request timed out  
Reply from 30.0.0.2: bytes=32 time=2ms TTL=64  
Reply from 30.0.0.2: bytes=32 time=8ms TTL=64  
Reply from 30.0.0.2: bytes=32 time=12ms TTL=64  
Ping Statistics for 30.0.0.2:  
Packets: Sent = 9, received = 3, Lost = 1 (25% loss).  
Approximate round trip time in milliseconds:  
Minimum = 2ms, Maximum = 12ms, Average = 6ms

② For R0 show ip route (route 0)

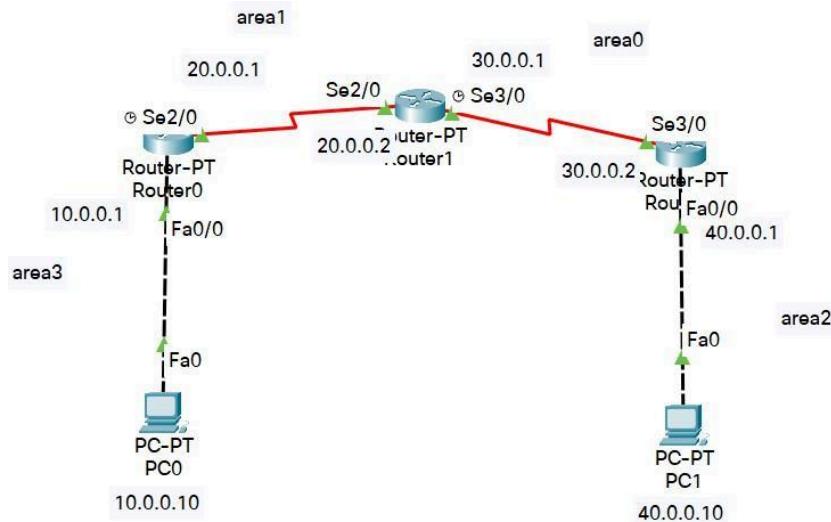
C 10.0.0.0/8 is directly connected FastEthernet 0/0  
R 20.0.0.0/8 [120/2] via 40.0.0.2, 00:00:19, Serial 2/0  
R 30.0.0.0/8 [120/2] via 40.0.0.2, 00:00:19, Serial 2/0  
C 40.0.0.0/8 is directly connected Serial 2/0  
R 50.0.0.0/8 [120/2] via 40.0.0.2, 00:00:19, Serial 2/0

③ For R1 show ip route (route 1)

R 10.0.0.0/8 [120/2] via 40.0.0.1, 00:00:19, Serial 2/0  
C 20.0.0.0/8 is directly connected, FastEthernet 0/0  
R 30.0.0.0/8 [120/2] via 50.0.0.2, 00:00:22, Serial 3/0  
C 50.0.0.0/8 is directly connected, Serial 2/0  
C 60.0.0.0/8 is directly connected, Serial 3/0

## 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
virtual-link 1.1.1.1
Router(config-router)#
01:11:56: %OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on OSPF_VL0 from LOADING to
FULL, Loading Done
```

## Pinging

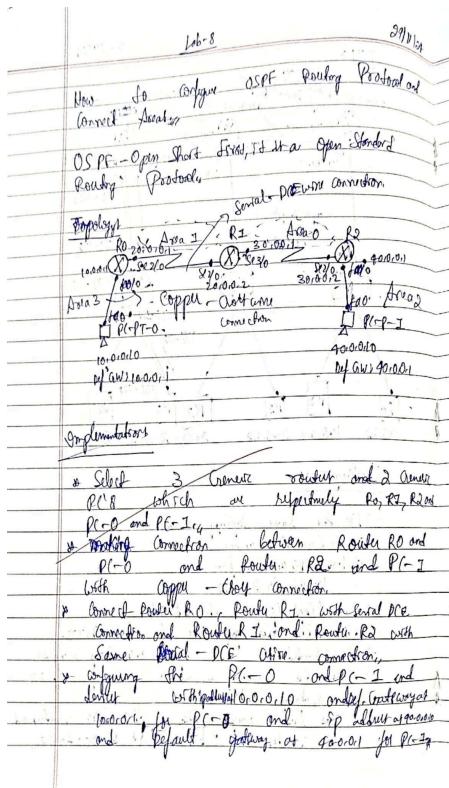
```
C:\>ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data:

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

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

iv. Observation



Bafna Gold  
Topic: Router

R3(config)# interface gigabitEthernet 0/0  
R3(config-if)# no shutdown  
R3(config-if)# exit

R3(config)# interface fastEthernet 2/0  
R3(config-if)# ip address 192.0.2.255.0.0.0  
R3(config-if)# no shutdown  
R3(config-if)# exit

Step 3/3: Verify if routing by configuring ospf  
 broadcast protocol in all routers.

On router R1:  
R1(config)#router OSPF 1  
R1(config-router)#network 1.1.1.1  
R1(config-router)#network 10.0.0.0 0.0.255.255.255.255.0.0.3

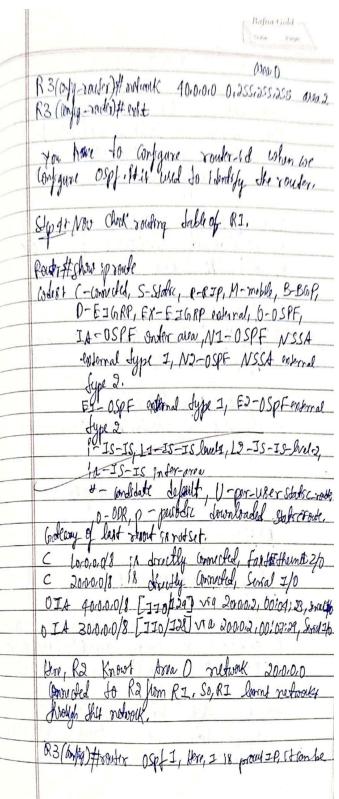
R1(config-router)#network 20.0.0.0 0.0.255.255.255.255.0.0.3

On router R2:  
R2(config)#router OSPF 2  
R2(config-router)#network 2.2.2.2  
R2(config-router)# network 20.0.0.0 0.0.255.255.255.255.0.0.3

R2(config-router)#network 30.0.0.0 0.0.255.255.255.255.0.0.3

R2(config-router)#exit

On router R3:  
R3(config)#router OSPF 2  
R3(config-router)#router id 3.3.3.3  
R3(config-router)#network 3.0.0.0 0.0.255.255.255.255.0.0.3



1 - 15535, 1 Subsidy Obj/price.

There must be one interface up to the esp, point up. So it better do multiple lookahead addrs to result, or it is virtual interface name generation and we configure.

R3 (config) # Interface loopback0  
 R3 (config)# ip add 122.16.1.252 255.255.0.0.  
 R3 (config) # no shutdown

R3 (config) # Interface loopback0  
 R3 (config)# ip add 122.16.1.253 255.255.0.0  
 R3 (config) # no shutdown

R3 (config) # Interface loopback0  
 R3 (config)# ip add 122.16.1.254 255.255.0.0  
 R3 (config) # no shutdown

Rajna Gold  
Date \_\_\_\_\_

Sight (break virtual link between R2/R3/R4/R5)  
we break a virtual link to ComSIS area 3 to  
area 0.

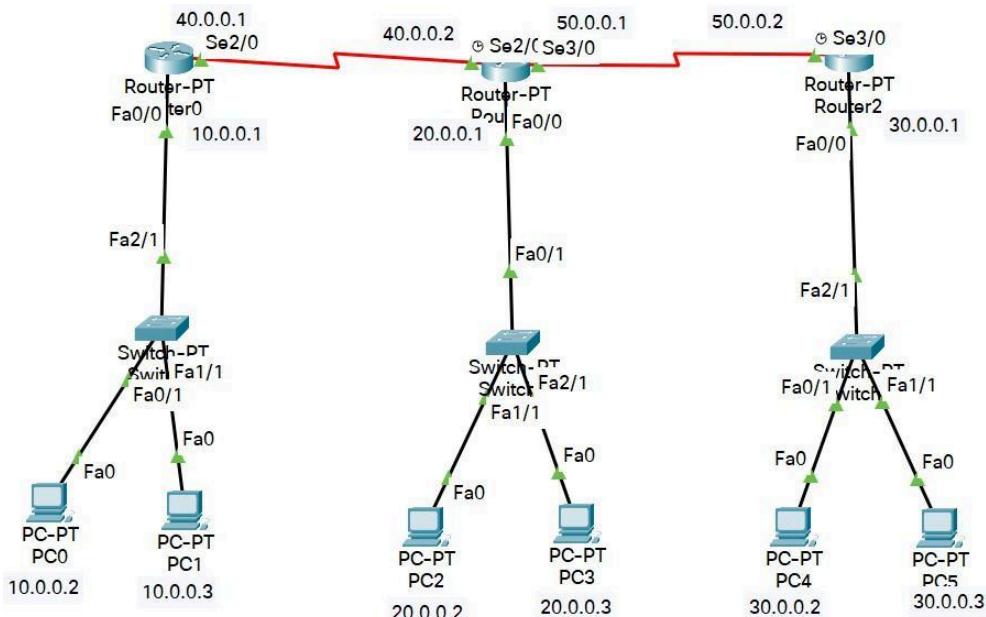
On Router R1:  
R1(config)#router OSPF1  
R1(config-router)#area 1 virtual-link 2.2.2.2  
R1(config-router)#  
Networking to full, handing done.

On Router R2:  
Virtual Serial port mismatch see If from  
link to area 0 will be virtual link between R2 and  
area 0.0.0.2, Serial 3/0a,  
from 200.0.0.2, Serial 3/0a,  
R2(config)#router ospf  
R2(config-router)#area 1 virtual-link 1.1.1.1  
R2(config-router)#  
from handing to full handing done.

Step 8: Check connectivity between host 10.0.0.10 &  
 90.0.0.30  
 \$ ping 90.0.0.30  
 pinging 90.0.0.30 with 32 bytes of data:  
 Reply from 90.0.0.30: bytes=32 time=9ms TTL=255  
 Reply from 90.0.0.30: bytes=32 time=8ms TTL=255  
 Reply from 90.0.0.30: bytes=32 time=9ms TTL=255  
 Reply from 90.0.0.30: bytes=32 time=8ms TTL=255  
 ping statistics for 90.0.0.30:  
 packets: sent=4, received=4, lost=0 (0%),  
 Approximate round trip delay in milliseconds:  
 Minimum = 8ms, Maximum = 9ms, Average = 8ms.

## 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

0			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

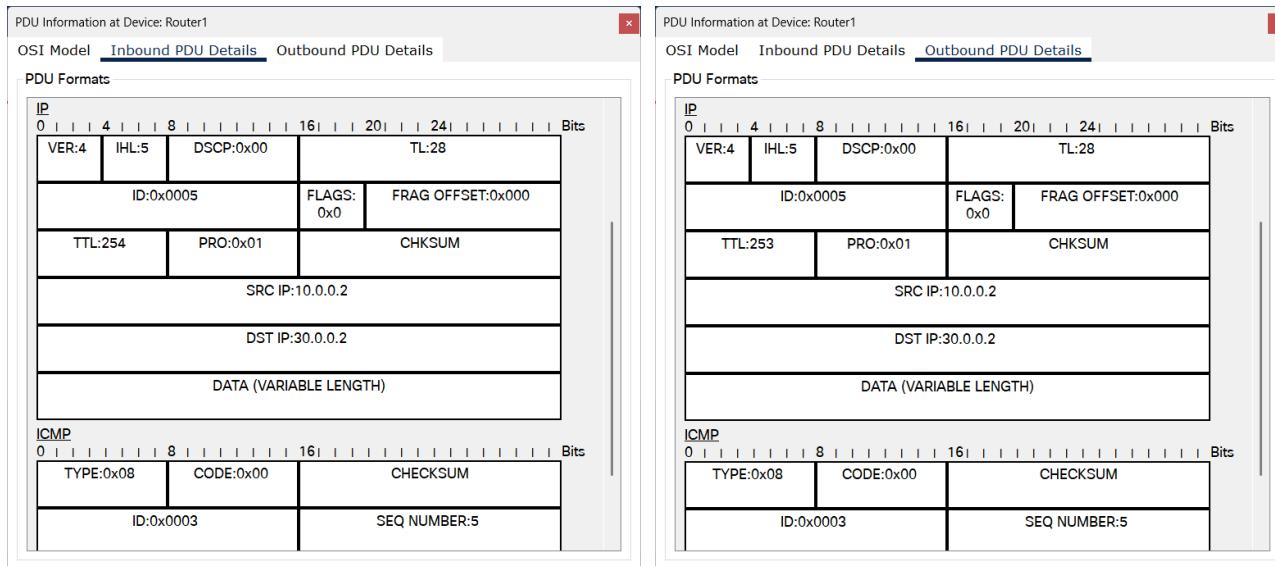
PDU Formats

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)							

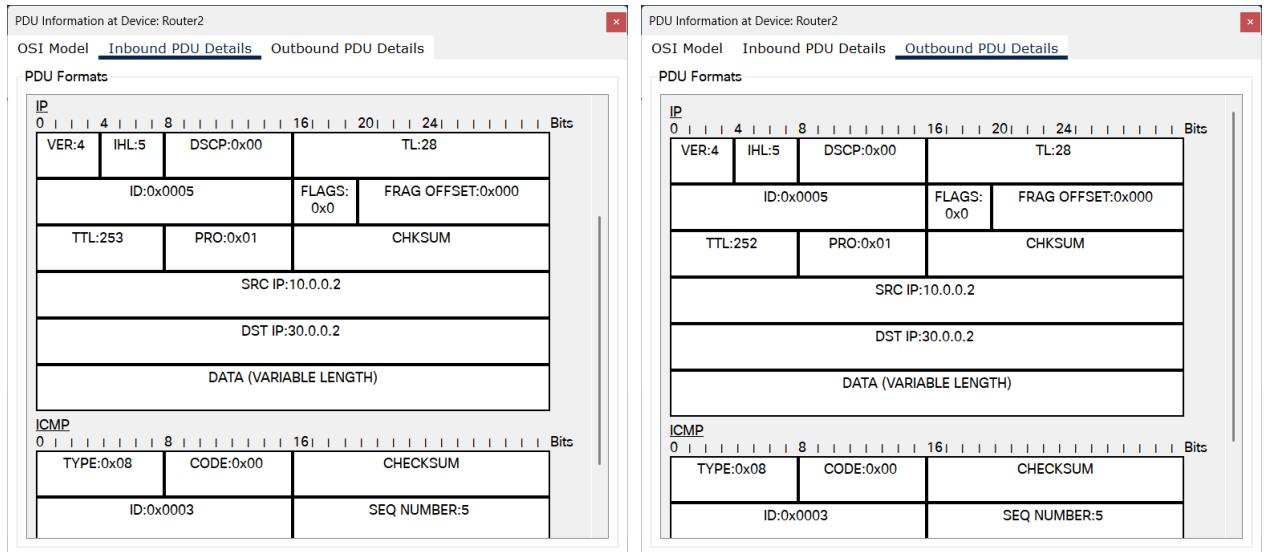
ICMP

0			8			16			Bits		
TYPE:0x08	CODE:0x00	CHECKSUM									
ID:0x0003			SEQ NUMBER:5								

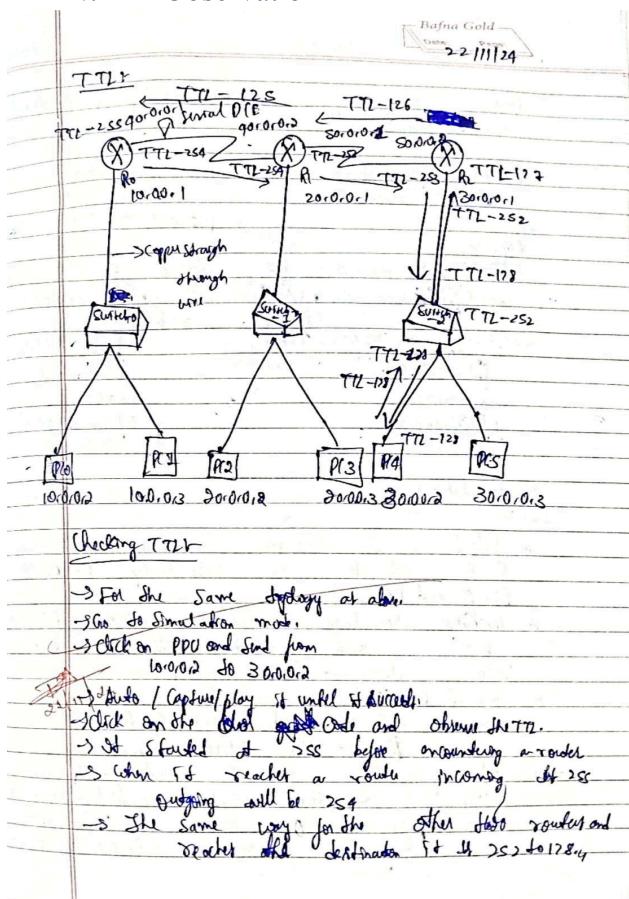
## Packet at Router1



## Packet at Router2

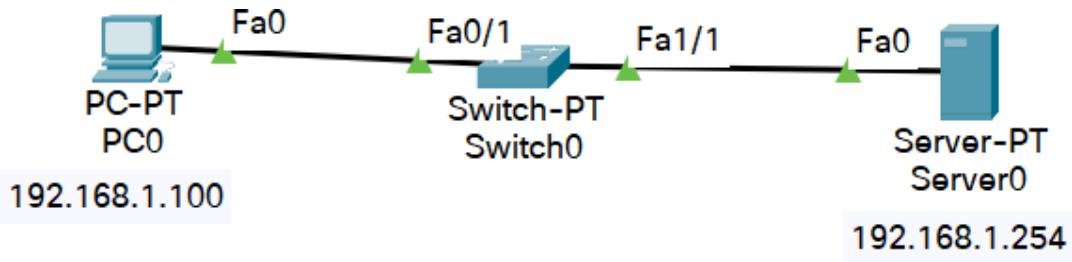


#### iv. Observation



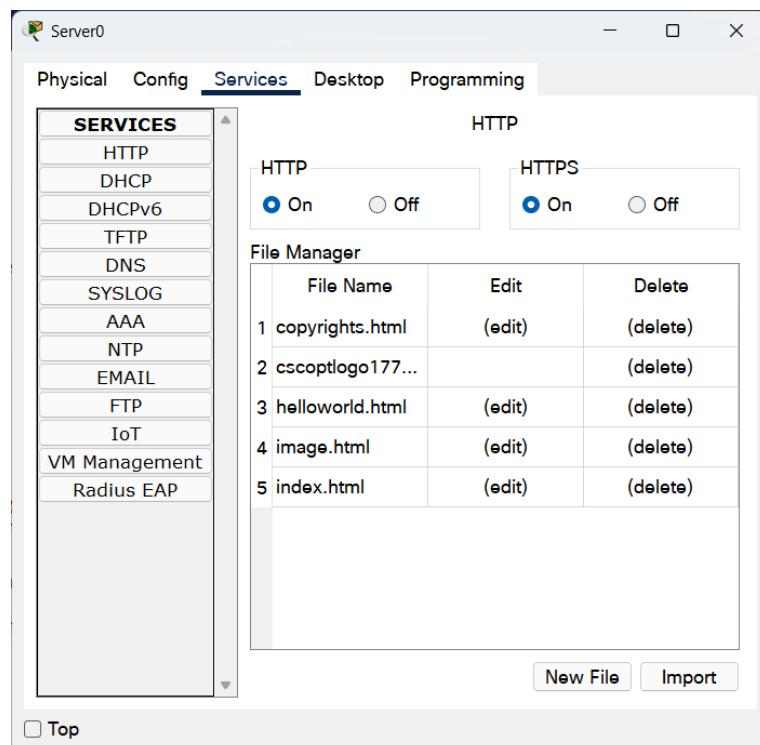
## Program 8

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

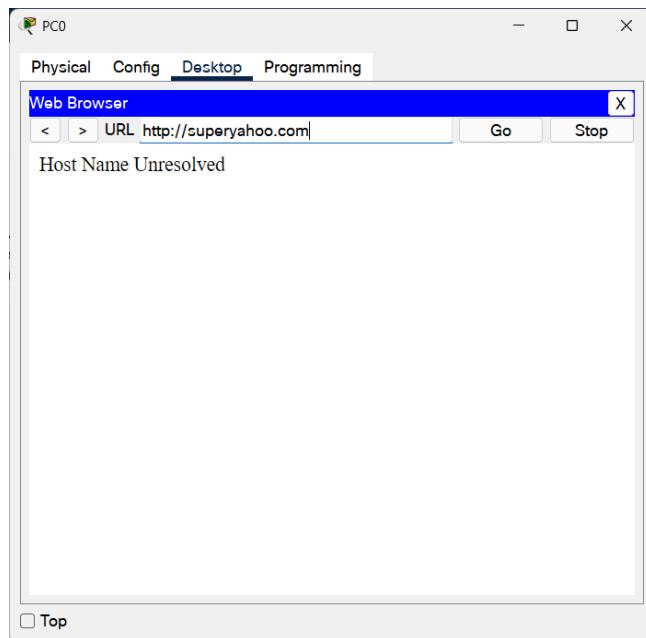
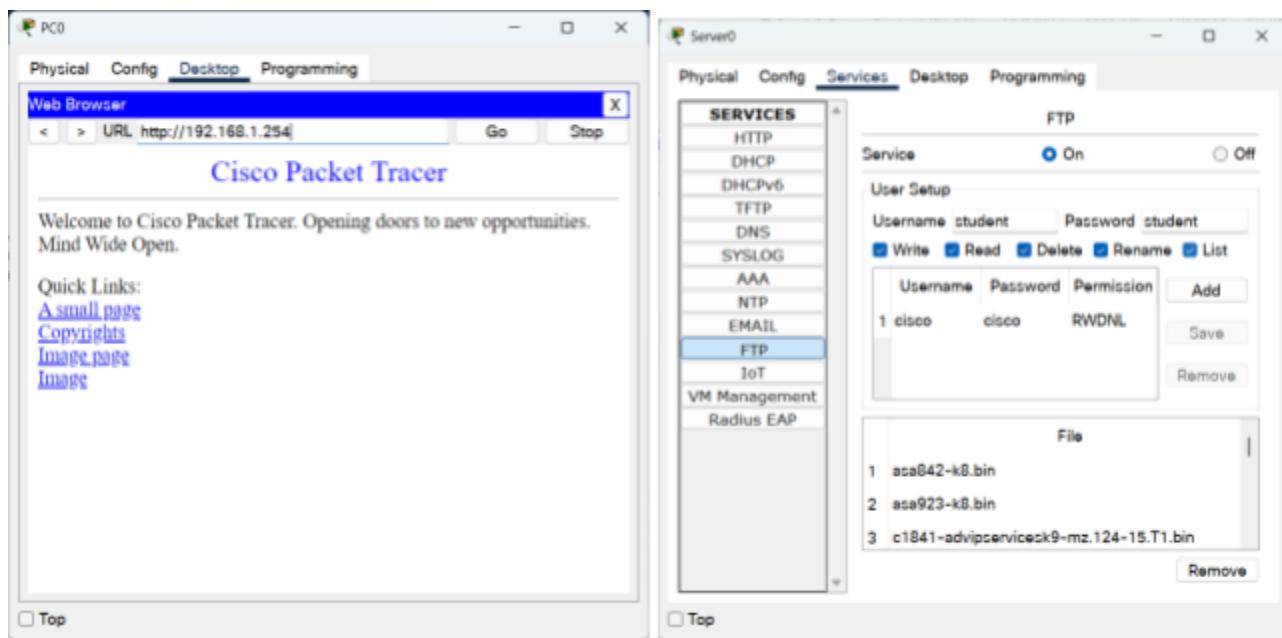


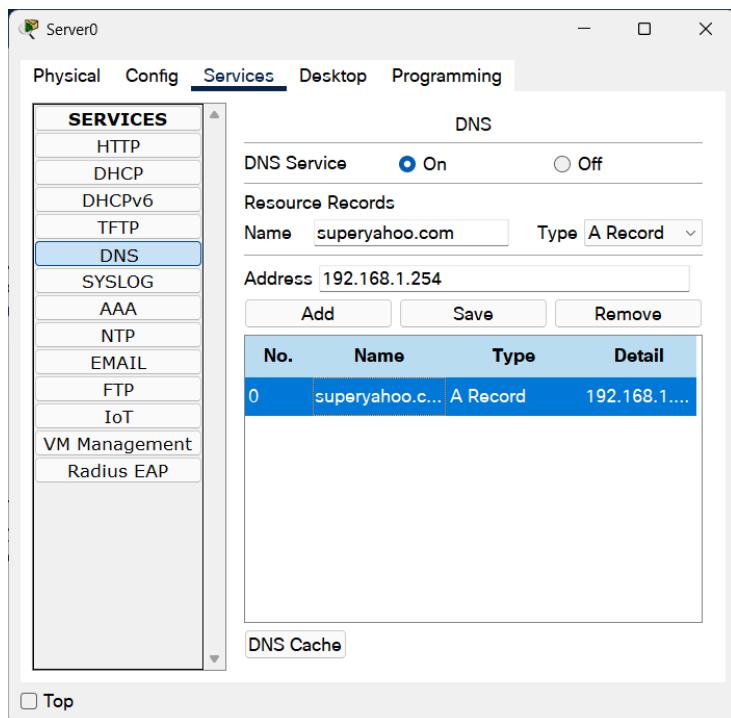
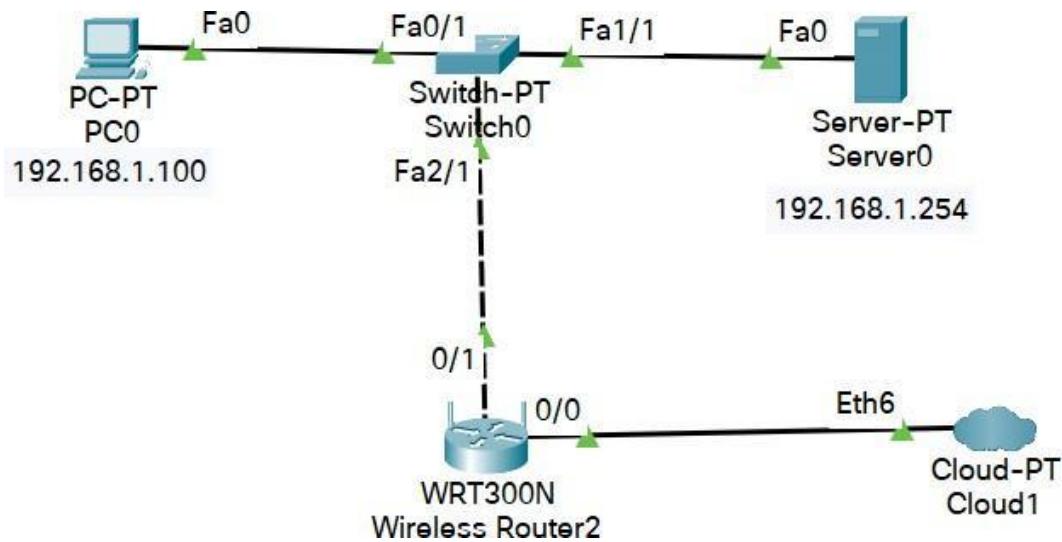
- iii. Screen shots/ output

Server's services

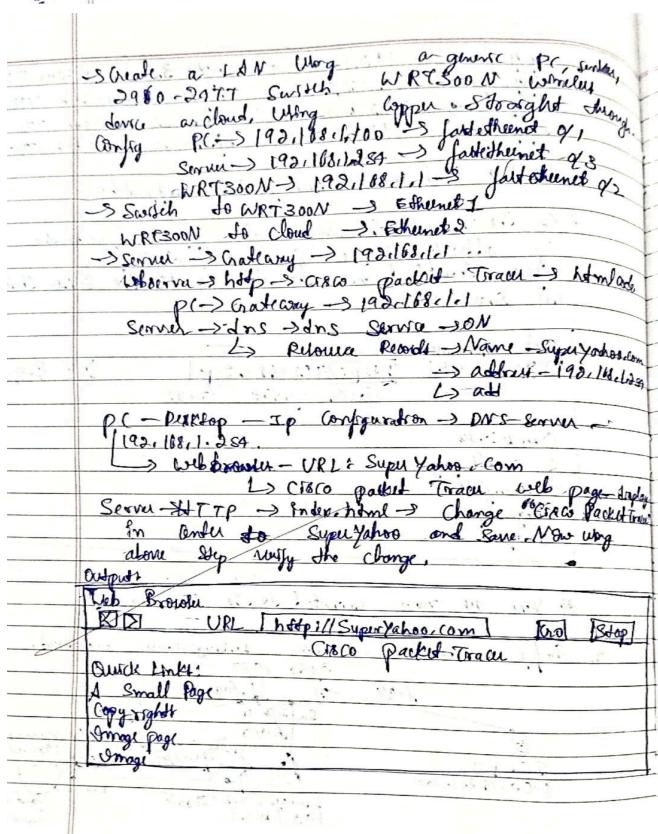
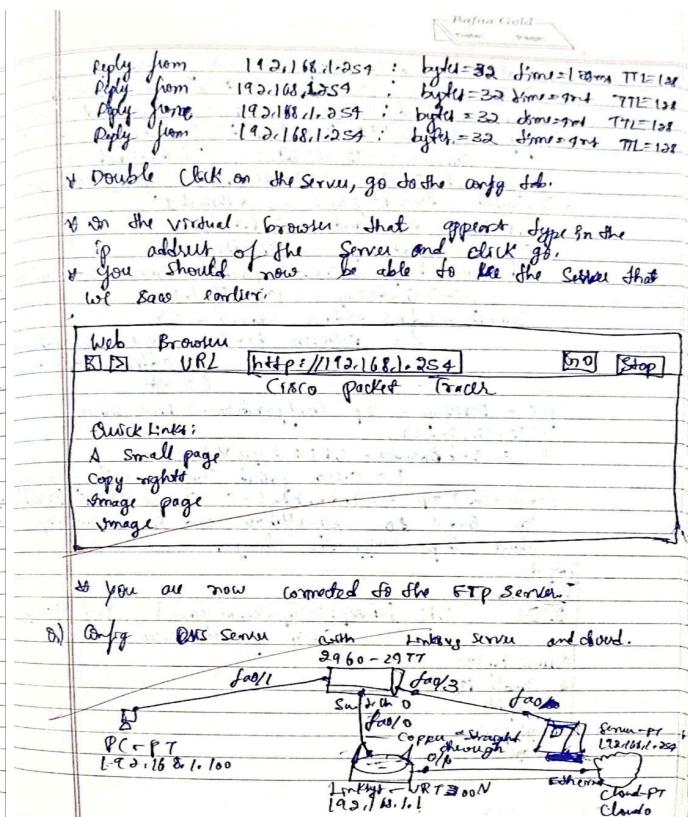
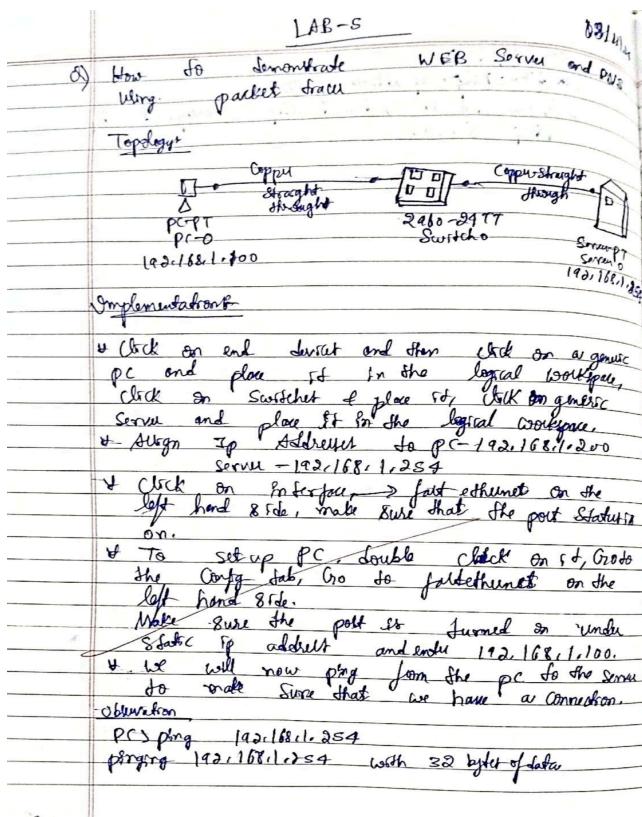


PC's Web Browser



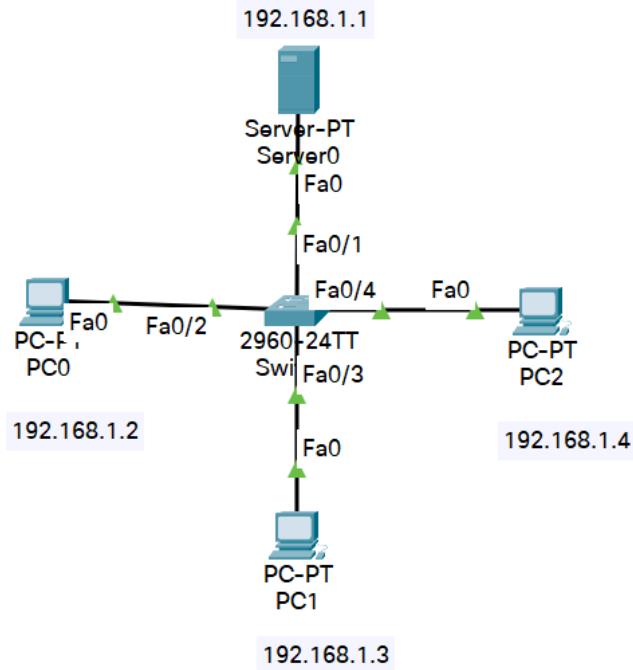


#### iv. Observation

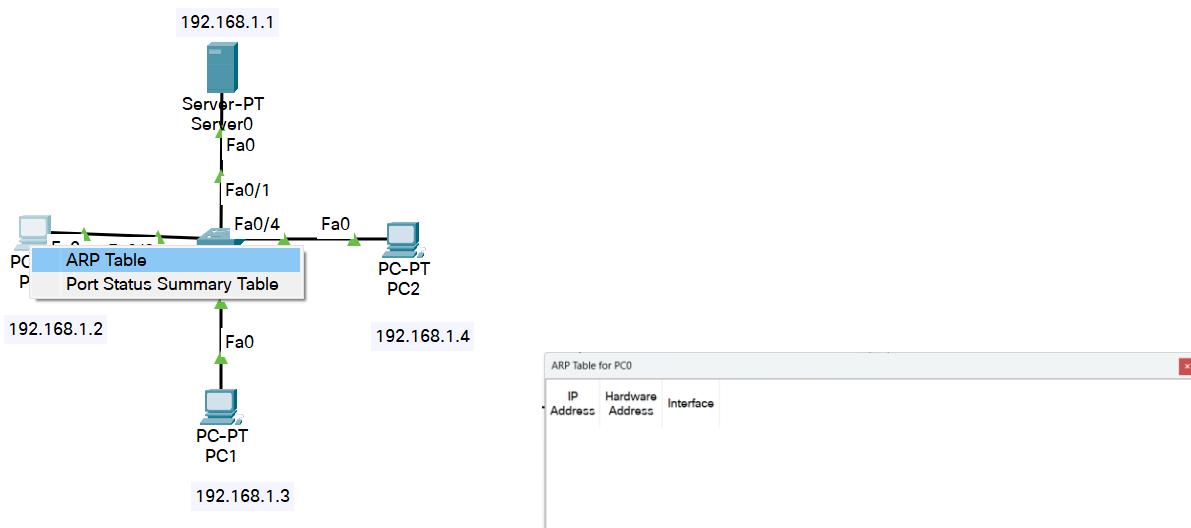


## Program 9

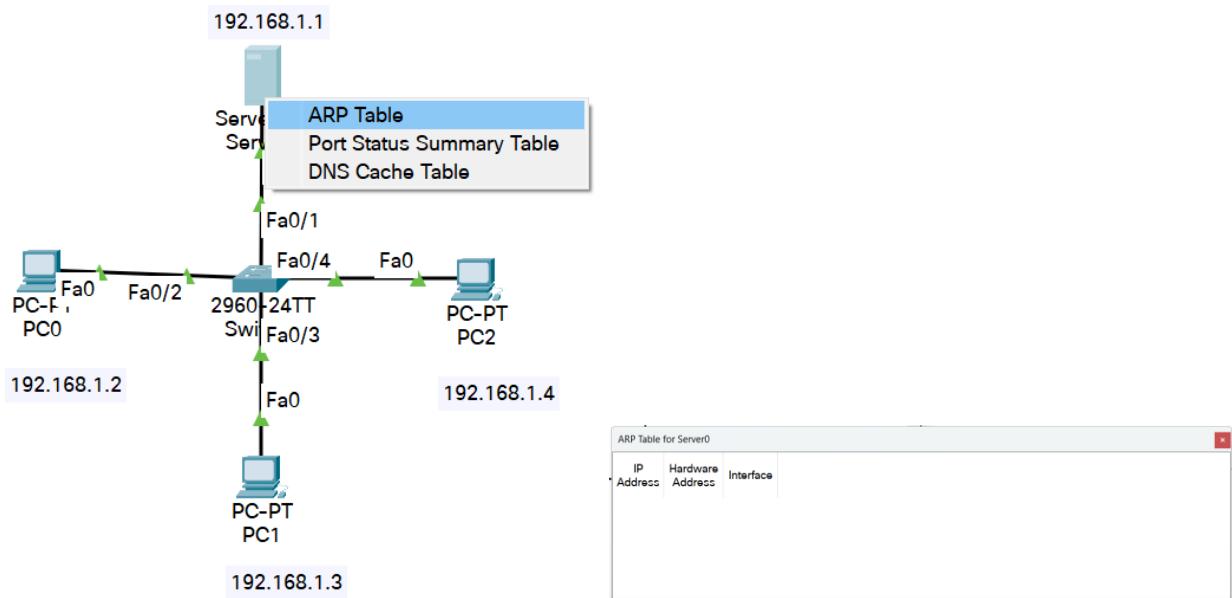
- i. To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)
- ii. Procedure along with the topology



- iii. Screen shots/ output  
ARP Table of PC



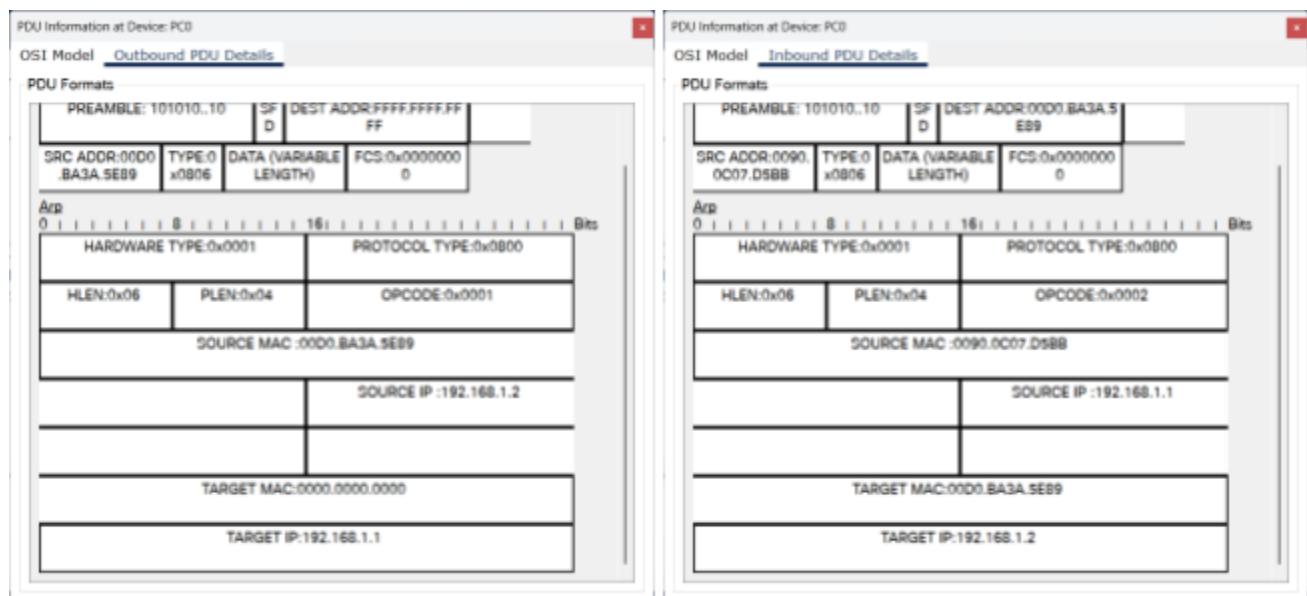
ARP Table of Server



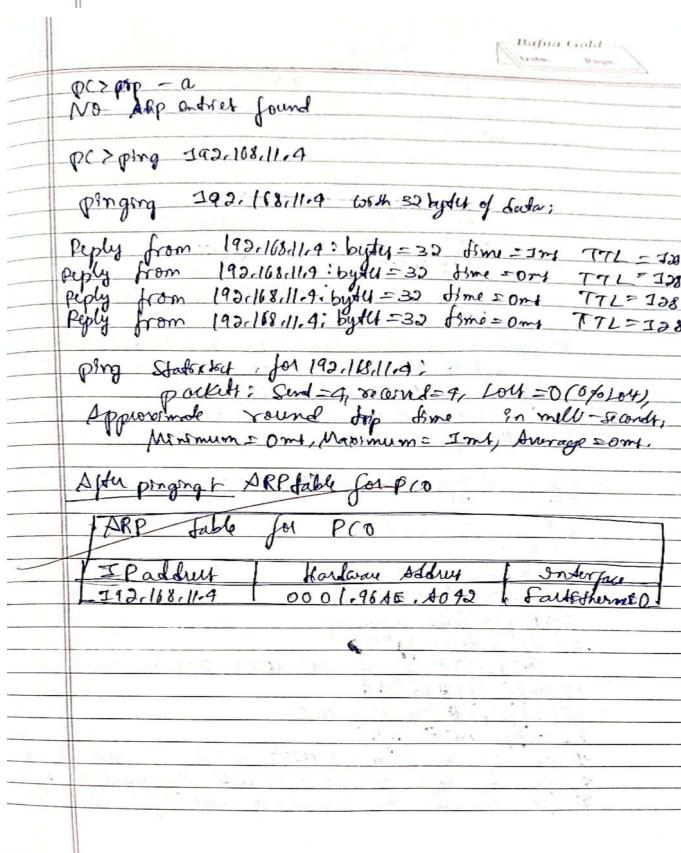
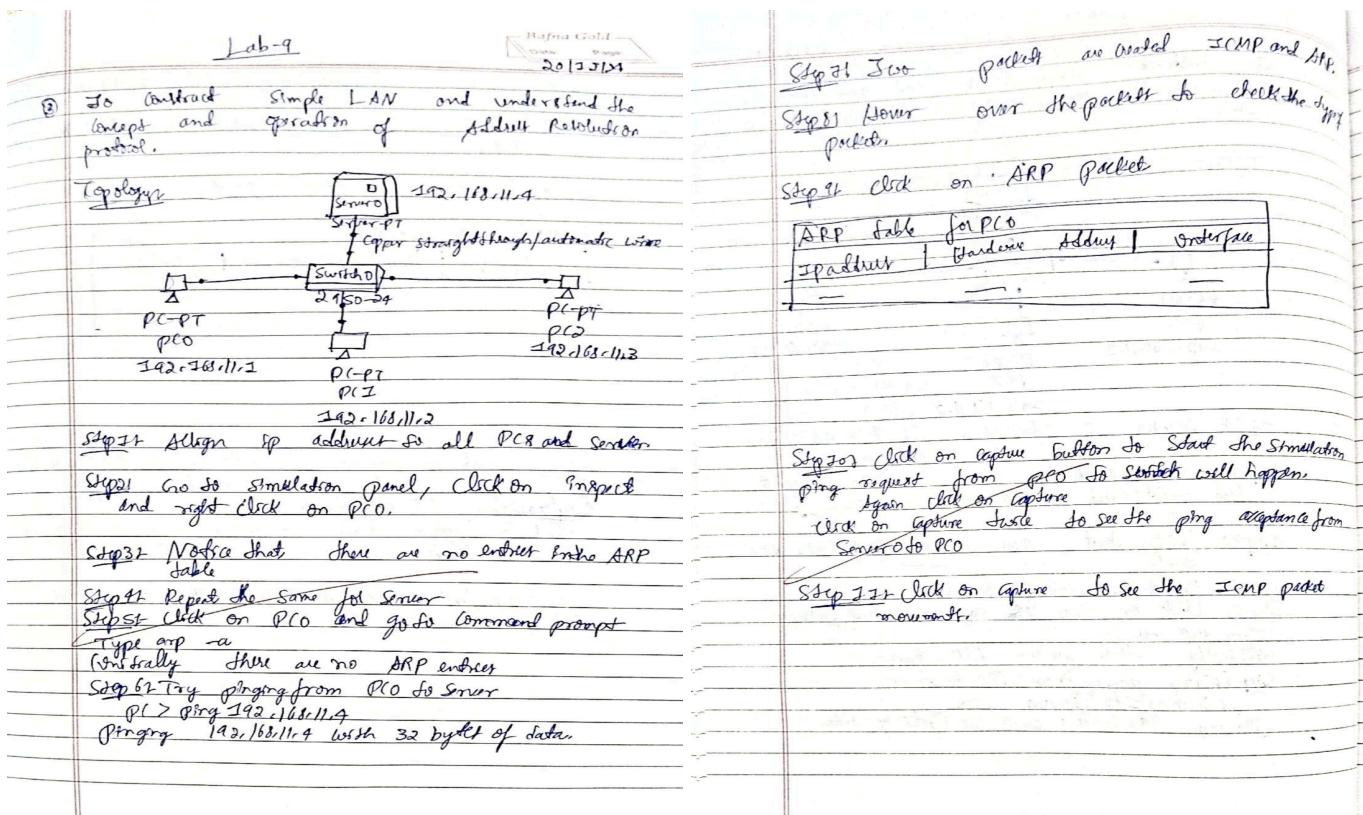
### Command at PC

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

### Pinging in Simulation Mode

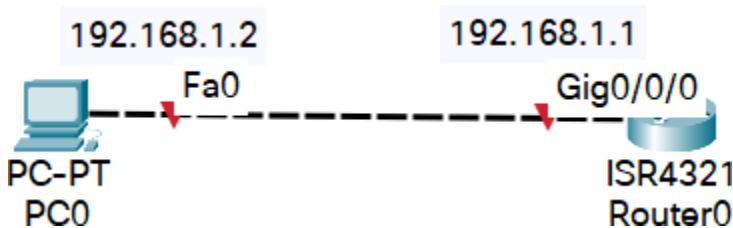


#### iv. Observation



## **Program 10**

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



- iii. Screen shots/ output  
Router

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

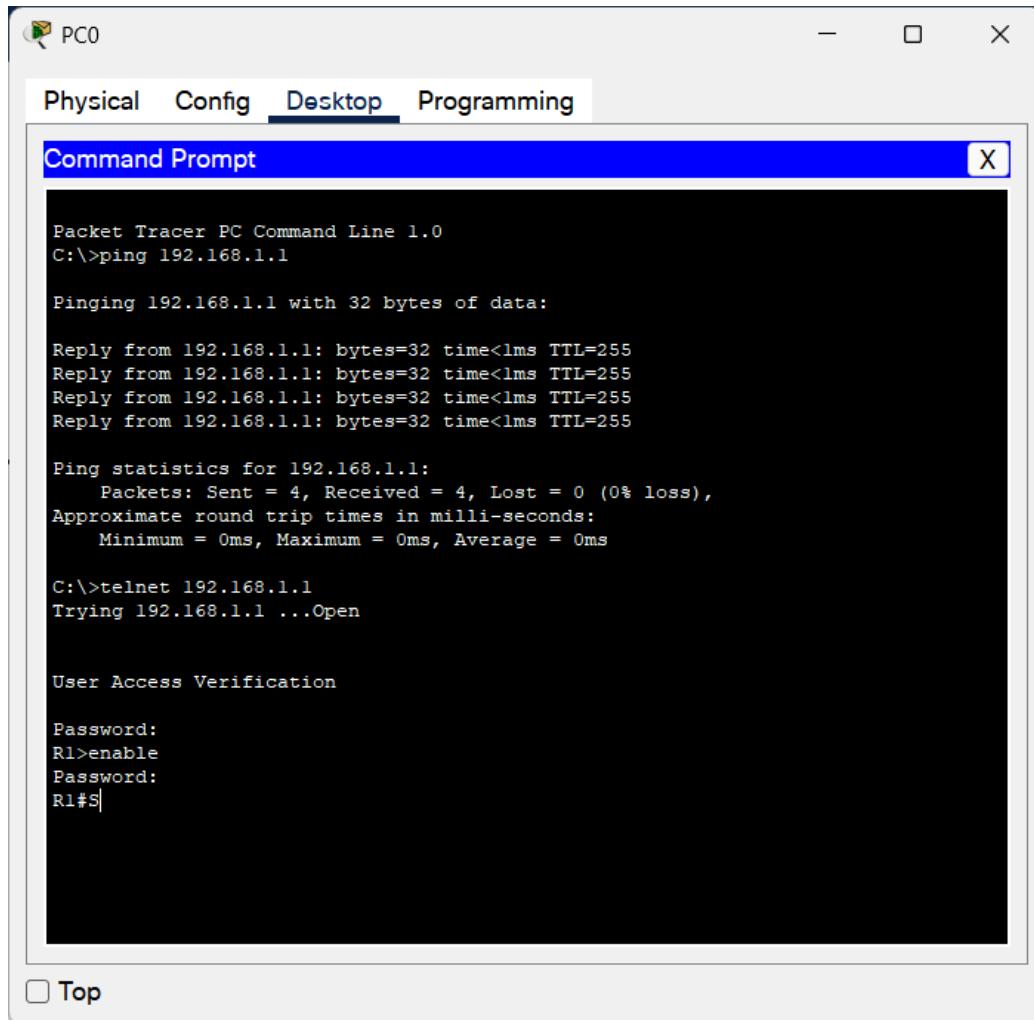
R1(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to up

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

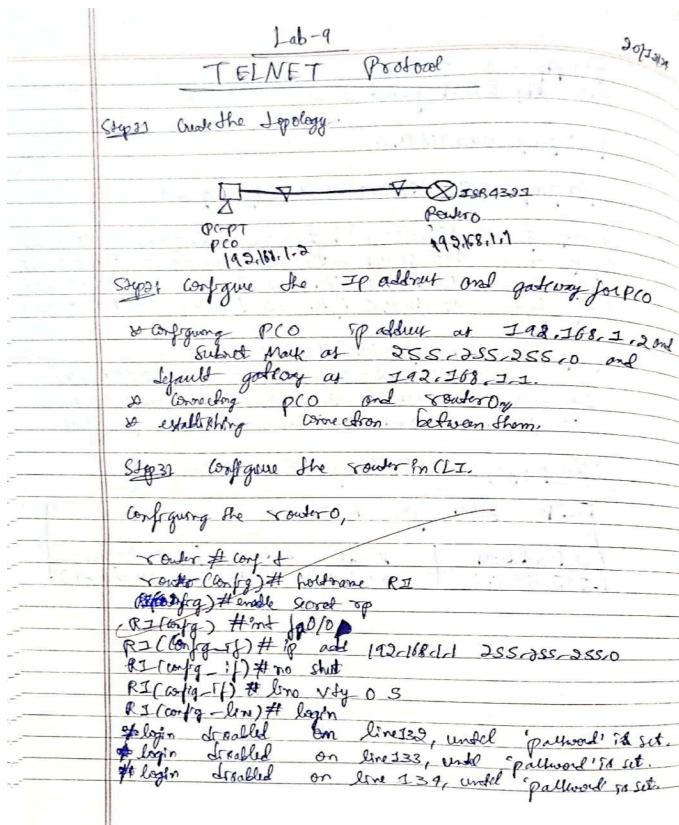
R1(config-if)#line vty 0 5
R1(config-line)#login
% Login disabled on line 2, until 'password' is set
% Login disabled on line 3, until 'password' is set
% Login disabled on line 4, until 'password' is set
% Login disabled on line 5, until 'password' is set
% Login disabled on line 6, until 'password'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

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

PC



#### iv. Observation



Bafna Gold  
Version 1.0

Router# login disabled on line 33, until 'password' is set.  
 Router# login disabled on line 135, until 'password' is set.  
 Router# login disabled on line 33, until 'password' is set.  
 Router(config-line)# password fp  
 Router(config-line)# exit  
 Router(config)# exit  
 Router#  
 Router# sys-s-config-i : Configured from console by console.  
 Router# hw Building configuration  
 Router# [OK]  
 Router#  
 Router#

Step 24 Go to PC> command prompt and ping the 192.168.1.1 (router) and telnet and enter the password.

PC> ping 192.168.1.1

pinging 192.168.1.1 with 32 bytes of data:

```

Reply from 192.168.1.1: bytes=32 time=0ms TTL=255
  
```

ping Statistics for 192.168.1.1:  
 packet: send=4, received=4, lost=0 (0% loss),  
 Approximate round trip times in milliseconds - seconds:  
 Minimum = 0ms, Maximum = 0ms, Average = 0ms.

Telnet

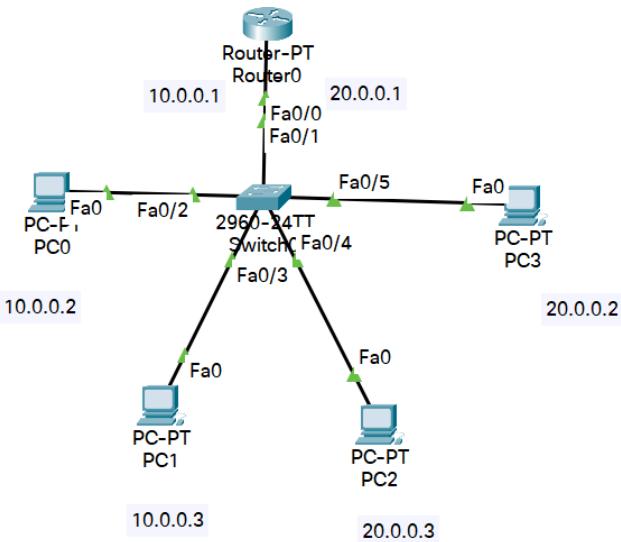
PC> telnet 192.168.1.1  
 Trying 192.168.1.1 -- Open

User Account Verification

Password:  
 Router#  
 Router#  
 Router#  
 Router#

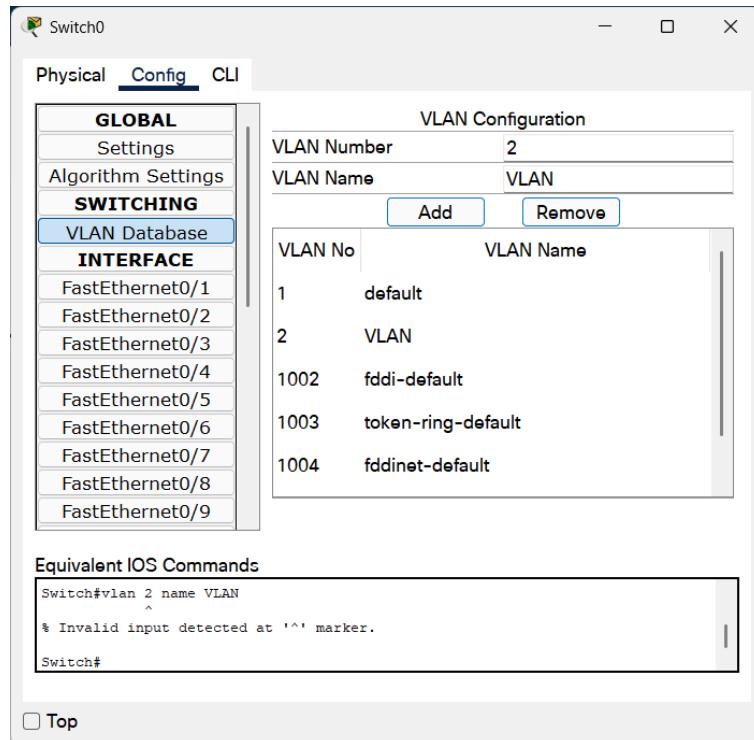
## Program 11

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

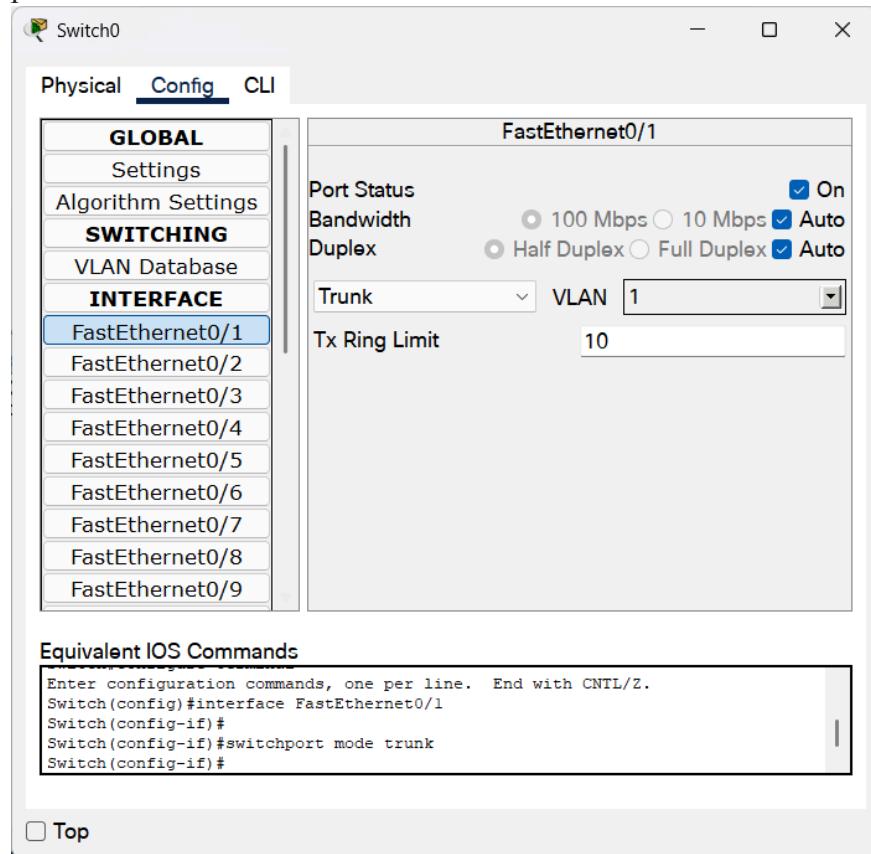


- iii. Screen shots/ output

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".

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

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

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

- Port Status:** On (checked)
- Bandwidth:** 100 Mbps (radio button)
- Duplex:** Half Duplex (radio button)
- Access:** VLAN 2 (selected from dropdown)
- Tx Ring Limit:** 10
- VLAN Selection:** A dropdown menu shows three options: "1:default" (unchecked), "2:VLAN" (checked), and "1002-fddi-default" (unchecked).

**Equivalent IOS Commands (Left):**

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

```

**Equivalent IOS Commands (Right):**

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

```

Top

## 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

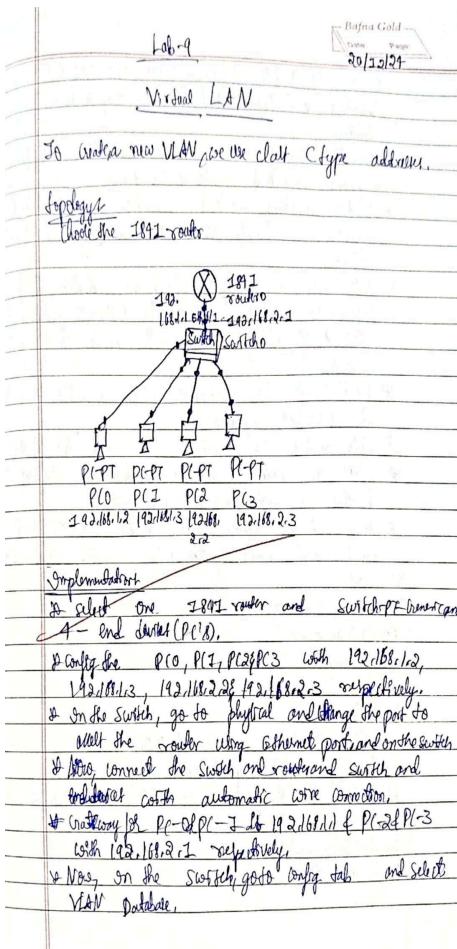
Router(config-if)#exit
Router(config)#interface Fa0/0.1
Router(config-subif)#
%LINK-5-CHANGED: Interface FastEthernet0/0.1, changed state to up

%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
```

iv. Observation



- \* On the VLAN Portable, Give any VLAN Number and Name and Number say 2 here, say Add
- \* Select the Subinterface i.e., 6/2 (over the Switch from standard mode if its trunk, from VLAN trunking almost switch to forward frame from different VLANs over a single link called trunk).
- \* Still is done by adding an additional header information called tag to the Ethernet frame, the process of adding this small header is called VLAN tagging.
- \* Look onto the interface of the switch with the configuration (Switch)

(8) Ping 192.168.2.2 from PC1 to PC2 and check if the result, the message is received.

(9) ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

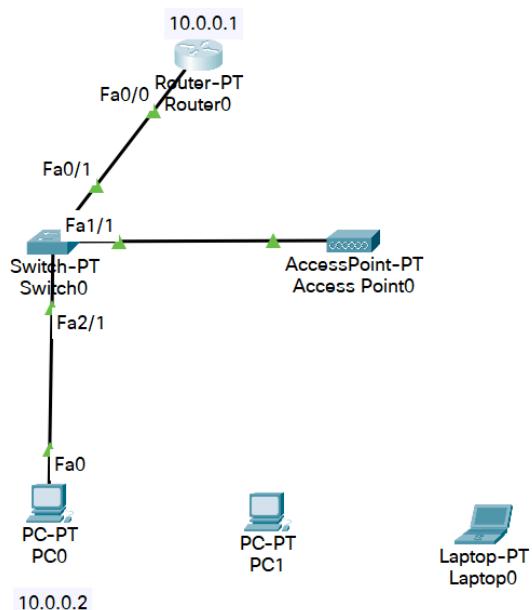
Reply from 192.168.2.2: bytes=32 time=2ms TTL=127  
 Reply from 192.168.2.2: bytes=32 time=1ms TTL=127  
 Reply from 192.168.2.2: bytes=32 time=2ms TTL=127  
 Reply from 192.168.2.2: bytes=32 time=1ms TTL=127

ping Statistics for 192.168.2.2:

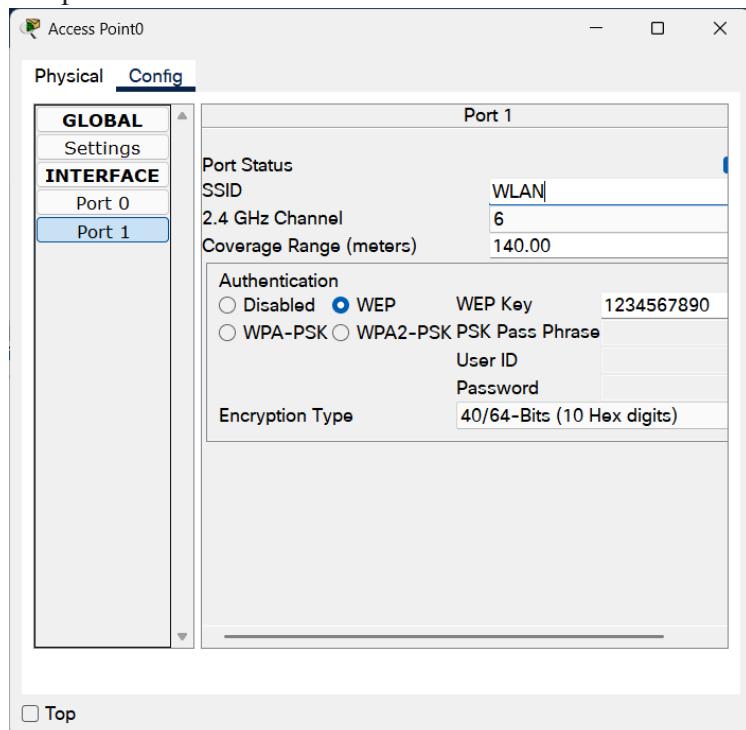
packets sent = 4, received = 4, lost = 0 (0% loss),  
 Approximate round trip times in milli-Seconds  
 Minimum = 1ms, Maximum = 2ms, Average = 2ms.

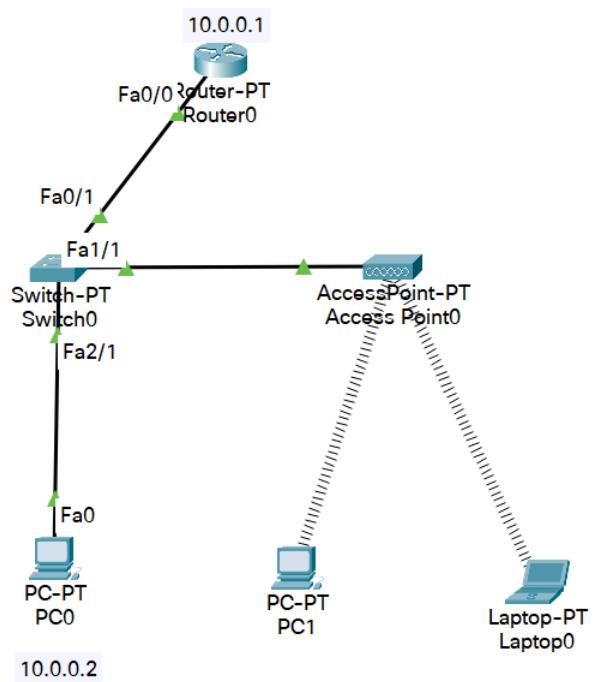
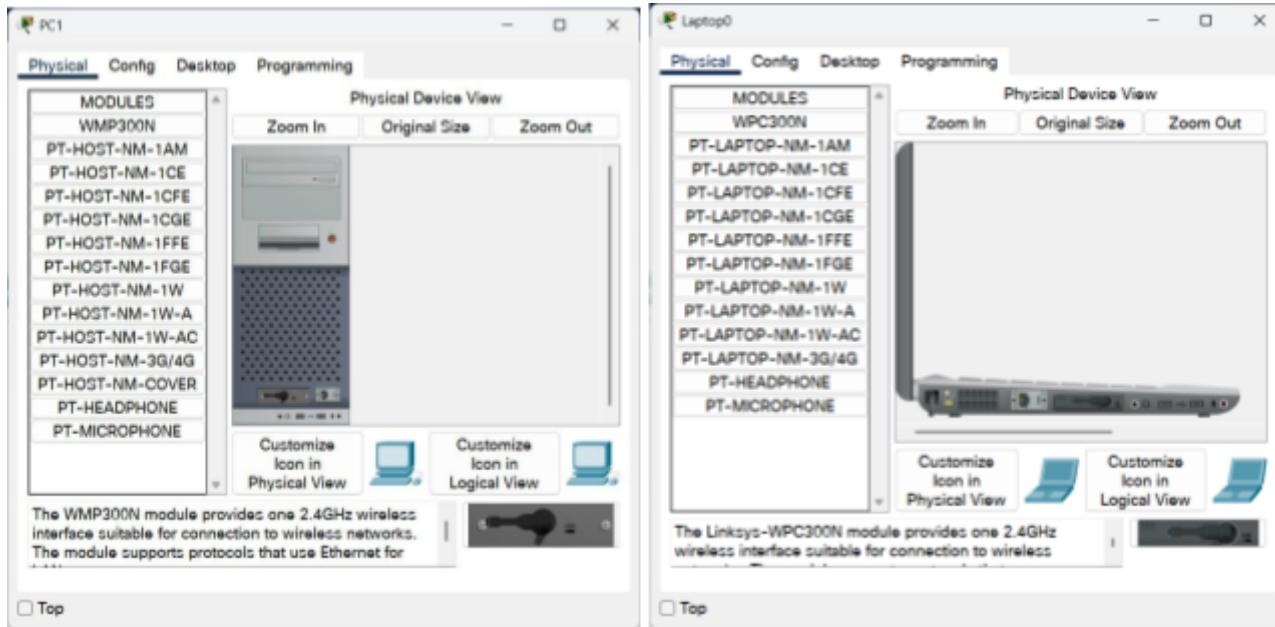
## Program 12

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

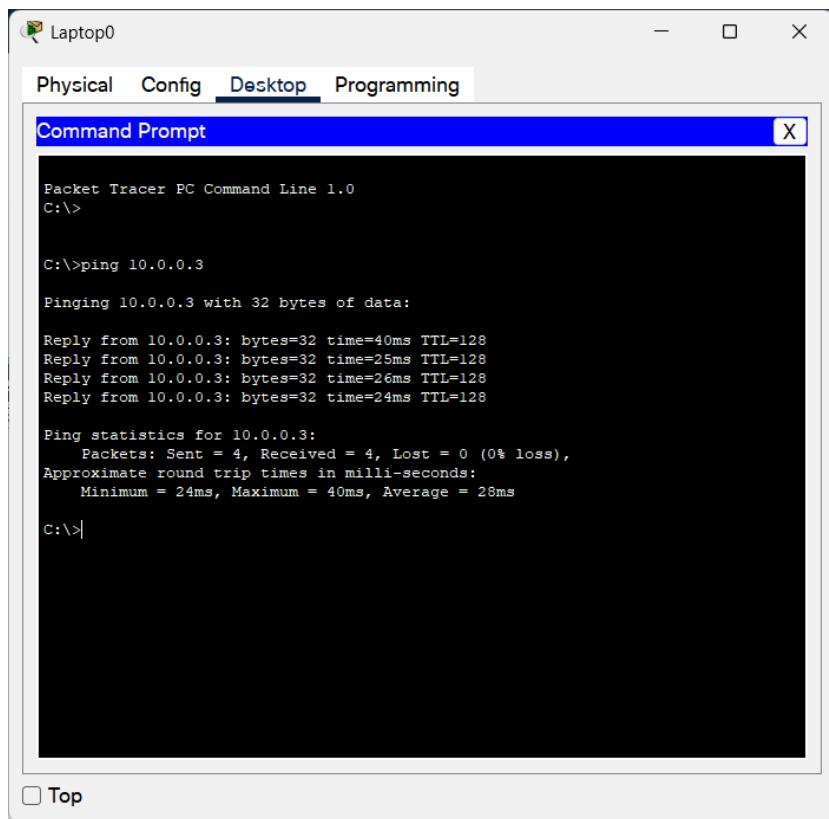


- iii. Screen shots/ output





Ping:



Laptop0

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:

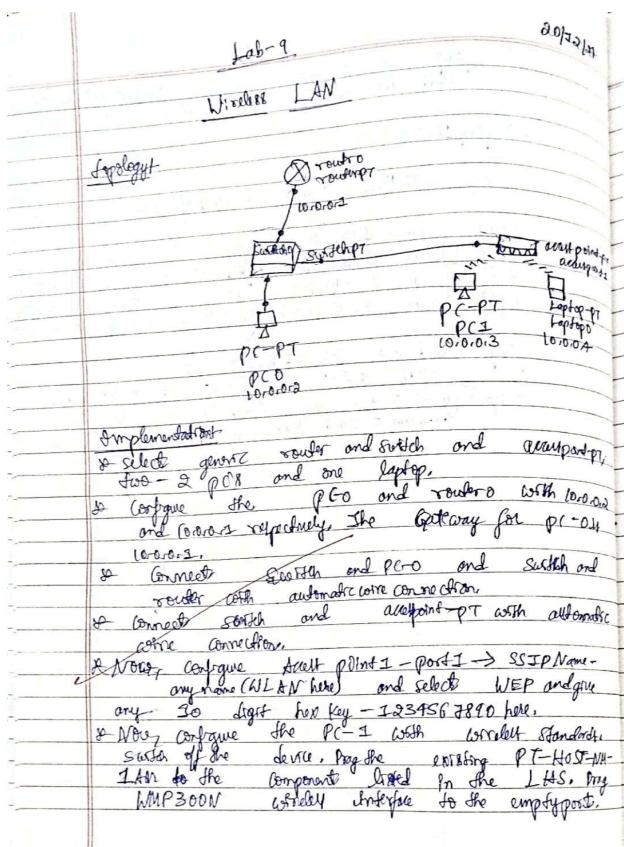
Reply from 10.0.0.3: bytes=32 time=40ms TTL=128
Reply from 10.0.0.3: bytes=32 time=25ms TTL=128
Reply from 10.0.0.3: bytes=32 time=26ms TTL=128
Reply from 10.0.0.3: bytes=32 time=24ms TTL=128

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



Bafna Gold  
Date: 27/11/2014

Switch on the device. On the config file, a new wireless interface would have been added. Now configure SSID, WEP, WEP Key, IP address and Gateway (at normally done) of the device and IP default at 10.0.0.3.

Repeat the same process for Laptop-O and assign/config IP address at 10.0.0.4.

Now, the wireless connection is established from output points to the PC-I and Laptop-O.

**Observations**

- ping the message from every device to every other device and the successful ping was done.
- Pinging 10.0.0.3 with 32 bytes of data:

```

Reply from 10.0.0.3: bytes=32 time=2ms TTL=128
Reply from 10.0.0.3: bytes=32 time=1ms TTL=128
Reply from 10.0.0.3: bytes=32 time=1ms TTL=128
Reply from 10.0.0.3: bytes=32 time=9ms TTL=128
  
```

ping statistics for 10.0.0.3:

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

Approximate round trip time in milli-seconds:  
Minimum = 6 ms, Maximum = 25 ms, Average = 13 ms

## Cycle-II

### Program 1

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

```
def crc(input_msg, poly, mode):  
    input_len = len(input_msg)  
    poly_len = len(poly)  
    padded_input = list(input_msg) # Convert to list for mutable operations  
  
    # If in mode 1 (transmission), append (poly_len - 1) zeros to the input message  
    if mode == 1:  
        # Append (poly_len - 1) zeros to the message for CRC calculation  
        padded_input += ['0'] * (poly_len - 1)  
  
    # Now, padded_input contains the original message with zeros appended for CRC calculation  
    padded_input = list(padded_input) # Ensure mutable  
  
    # Perform XOR division using the polynomial  
    for i in range(input_len):  
        if padded_input[i] == '1': # Start XOR operation only when the current bit is '1'  
            for j in range(poly_len):  
                # XOR each corresponding bit of the padded_input and polynomial  
                padded_input[i + j] = '0' if padded_input[i + j] == poly[j] else '1'  
  
    # After the division, padded_input contains the message with the CRC remainder  
    output = list(input_msg) # Copy the original message  
  
    # Append the remainder (CRC) to the output message  
    output += padded_input[input_len:] # Copy the remainder (CRC part) from padded_input  
    return ''.join(output) # Return as a string  
  
  
def check_crc(received, poly):  
    received_len = len(received)  
    poly_len = len(poly)  
    received = list(received) # Convert to list for mutable operations  
  
    # Perform XOR division to check for errors  
    for i in range(received_len - poly_len + 1):  
        if received[i] == '1':  
            for j in range(poly_len):  
                # XOR each corresponding bit of the received message and polynomial  
                received[i + j] = '0' if received[i + j] == poly[j] else '1'  
  
    # If all bits are '0', the message is error-free  
    for i in range(received_len - poly_len + 1, received_len):  
        if received[i] == '1':
```

```

        return False # Error detected

    return True # No error detected

def main():
    # CRC polynomial (same as in the C code)
    poly = "1000100000100001"

    # Input message
    ip = input("Enter the input message in binary: ")

    # Calculate the transmitted message with CRC (mode = 1)
    transmitted_message = crc(ip, poly, 1)
    print(f"The transmitted message is: {transmitted_message}")

    # Simulate receiving the message and check for errors
    recv = input("Enter the received message in binary: ")

    # Check if the received message has errors
    if check_crc(recv, poly):
        print("No error in data")
    else:
        print("Error in data transmission has occurred")

if __name__ == "__main__":
    main()

```

iii. Screen shots/ output

---

→ Enter the input message in binary: 11111

The transmitted message is: 111111100011101110

Enter the received message in binary: 111111101011101110

Error in data transmission has occurred

→ Enter the input message in binary: 11111

The transmitted message is: 111111100011101110

Enter the received message in binary: 111111100011101110

No error in data

#### iv. Observation

Lab-6

Bafna Gold  
15/11/2019

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

```

def Crc (Input-msg, poly, mode):
    input_len = len(Input-msg)
    poly_len = len(poly)
    padded_input = list(Input-msg)

    if mode == 1:
        padded_input += [0] * (poly_len - 1)
    padded_input = list(padded_input)

    for i in range(input_len):
        if padded_input[i] == '1':
            for j in range(poly_len - 1):
                padded_input[i+j] ^= poly[j]
            padded_input[i+poly_len-1] = '1'
        else:
            padded_input[i+poly_len-1] = '0'

    output = list(Input-msg)
    output += padded_input[-poly_len:]
    return ''.join(output)

def check_CRC(received_poly):
    received_len = len(received_poly)
    poly_len = len(poly)
    received = list(received_poly)
    for i in range(received_len - poly_len + 1):
        if received[i] == '1':
            for j in range(poly_len - 1):
                received[i+j] ^= poly[j]
            received[i+poly_len-1] = '1'
        else:
            received[i+poly_len-1] = '0'

```

$\text{if received}[i:j] == \text{poly}[j:i]$   
 for i in range(received\_len - poly\_len + 1):  
 $\text{received}[i:j]$   
 $\text{if received}[i:j] \leq 1:$   
 $\text{return False}$   
 $\text{return True}$

```

def main():
    poly = "1000100000100001"
    ip = input("Enter the Input message in binary:")
    transmitted_message = Crc(ip, poly, 1)
    print("The transmitted message is: ", transmitted_message)
    recv = input("Enter the received message in binary:")
    if check_CRC(recv, poly):
        print("No error in data")
    else:
        print("Error in data transmission has occurred")
    if main == "main":
        main()

```

Output of Successful call:  
 Enter the input message in binary: 11111  
 The transmitted message is: 11111100011101110  
 Enter the received message in binary: 11111100011101110  
 No error in data.

Output of failure/error case:  
 Enter the input message in binary: 11111  
 The transmitted message is: 11111100011101110  
 Error in data transmission has occurred.

## **Program 2**

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

```
def main():  
    # Initial values  
    storage = 0 # Initial packets in the bucket  
    no_of_queries = 4 # Total number of times bucket content is checked  
    bucket_size = 10 # Total number of packets that can be accommodated in the bucket  
    input_pkt_size = 4 # Number of packets that enter the bucket at a time  
    output_pkt_size = 1 # Number of packets that exit the bucket at a time  
  
    for i in range(no_of_queries):  
        # Space left in the bucket  
        size_left = bucket_size - storage  
  
        if input_pkt_size <= size_left:  
            # Update storage if space is available  
            storage += input_pkt_size  
        else:  
            print(f'Packet loss = {input_pkt_size}')  
  
        print(f'Buffer size= {storage} out of bucket size= {bucket_size}')  
  
        # Decrease storage by the output packet size (packets exiting the bucket)  
        storage -= output_pkt_size  
  
if __name__ == "__main__":  
    main()
```

iii. Screen shots/ output

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

iv. Observation

Lab-6  
15/11/2021

a) Bucket Bucket Problem

```
def main():
    Storage = 0
    no_of_packets = 4
    bucket_size = 10
    Input_pkt_size = 4
    Output_pkt_size = 1

    for i in range(no_of_packets):
        size_left = bucket_size - Storage
        if Input_pkt_size <= size_left:
            Storage += Input_pkt_size
        else:
            print("packet lost = ", Input_pkt_size)
    print("Buffer size = ", Storage, " out of bucket size = ", bucket_size)

    Storage -= Output_pkt_size

    if name == "main":
        main()
    else:
        print("Output")
        print("Buffer Size = 4 out of bucket size=10")
        print("Buffer Size = 7 out of bucket size=10")
        print("Buffer size = 10 out of bucket size=10")
        print("Packet lost = 4")
        print("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

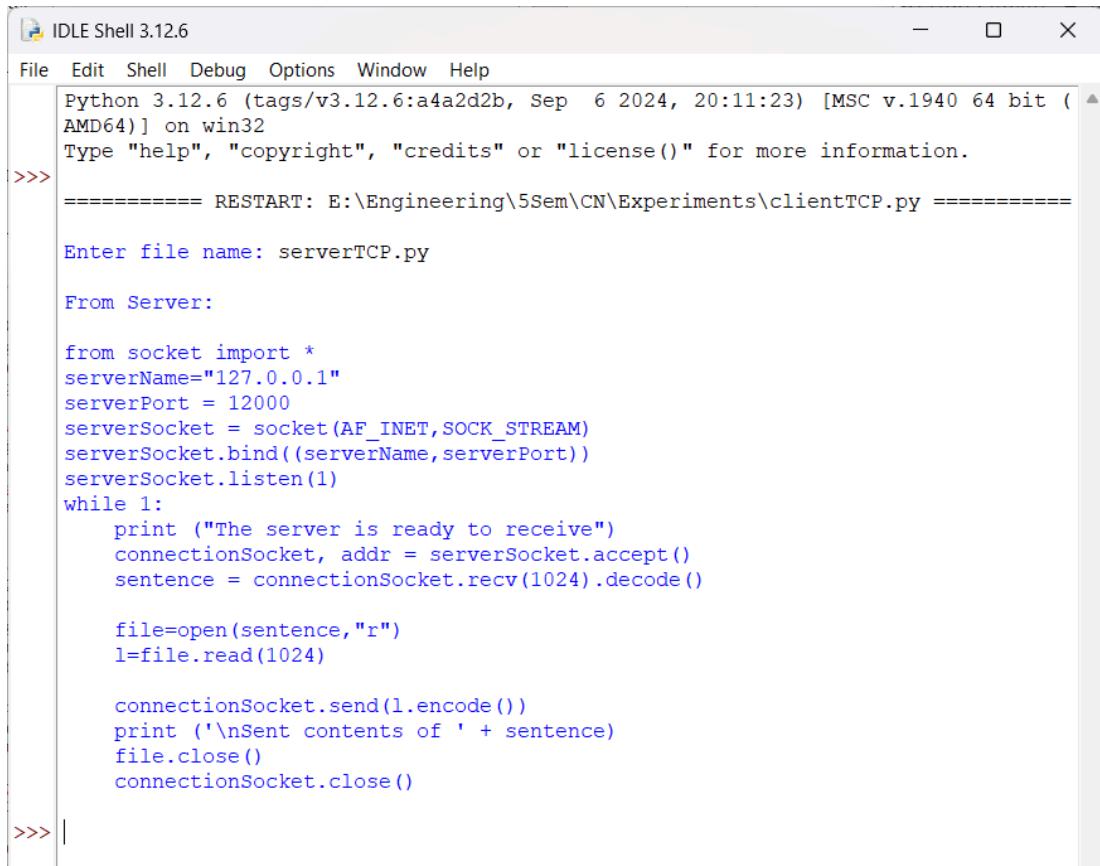
```
from socket import *
serverName="127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print ("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()

    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:

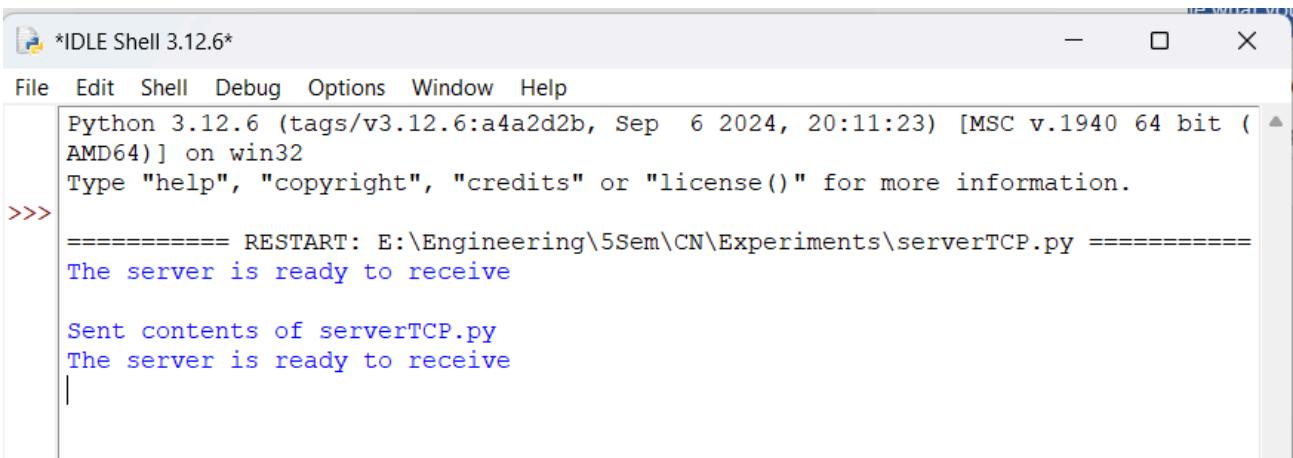
from socket import *
serverName="127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print ("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()

    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

Lab-9

Bafna Gold  
Today Date  
20/12/21

1. Using TCP/IP Sockets, write a Client-Server program to make Client sending the file name and Server to send back the contents of the required file.

ClientTCP.py

```
from socket import*
ServerName = "127.0.0.1"
ServerPort = 12000
ClientSocket = socket(AF_INET, SOCK_STREAM)
ClientSocket.connect((ServerName, ServerPort))
Sentence = input("In Enter filename:")
ClientSocket.send(Sentence.encode())
fileContent = ClientSocket.recv(1024).decode()
print("From Server:", fileContent)
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()
while 1:
    print("The server is ready to receive")
    ConnectionSocket, add = ServerSocket.accept()
    Sentence = ConnectionSocket.recv(1024).decode()
    file = open(Sentence, "r")
```

```
l = file.read(1024)
ConnectionSocket.send(l.encode())
print("In Sent contents of file Sentence")
file.close()
ConnectionSocket.close()
```

Output

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

20 12 20

## **Program 4**

- i. Using UDP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.
- ii. Procedure

clientUDP.py

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

sentence = input("\nEnter file name: ")

clientSocket.sendto(bytes(sentence,"utf-8"),(serverName, serverPort))

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

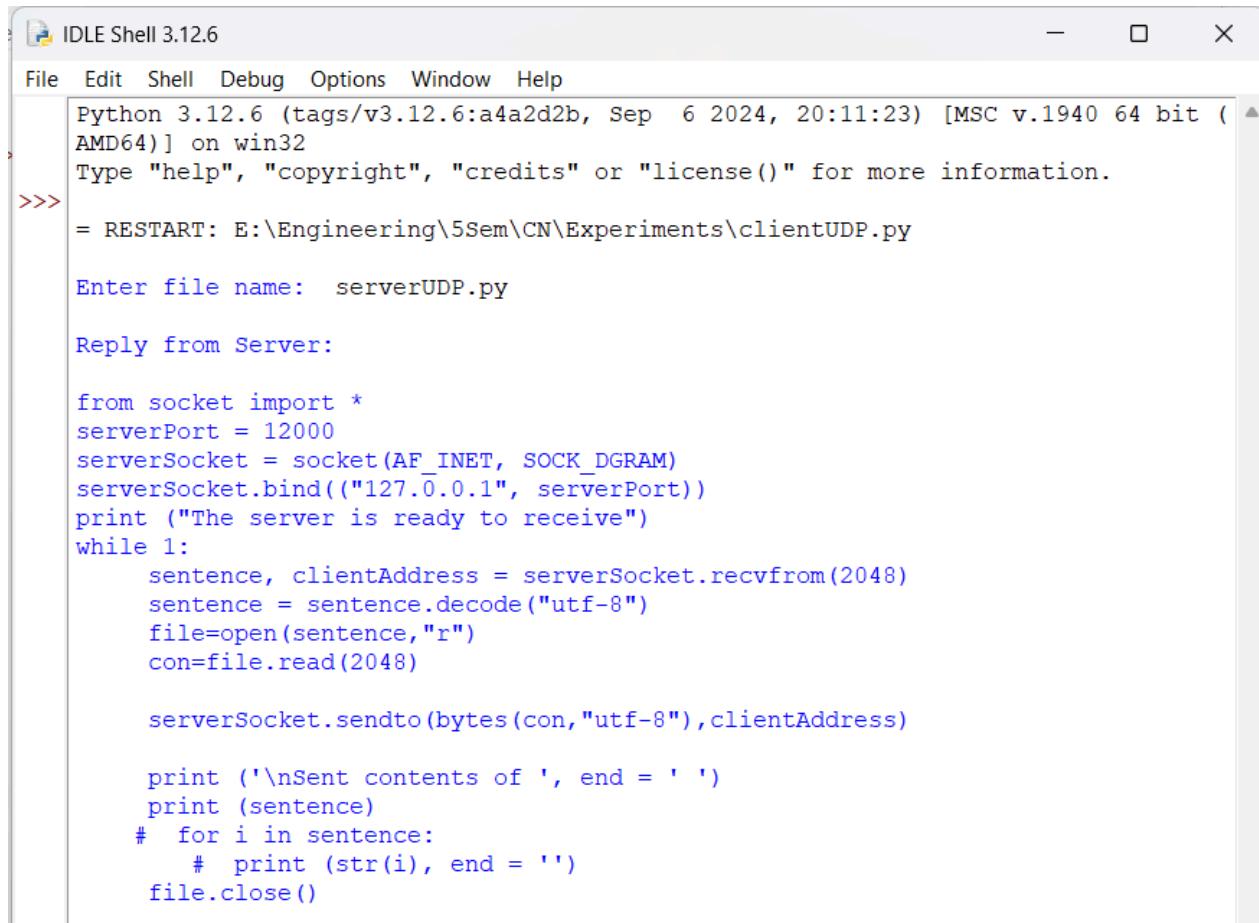
serverUDP.py

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

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

### iii. Screen shots/ output

#### Client



```
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

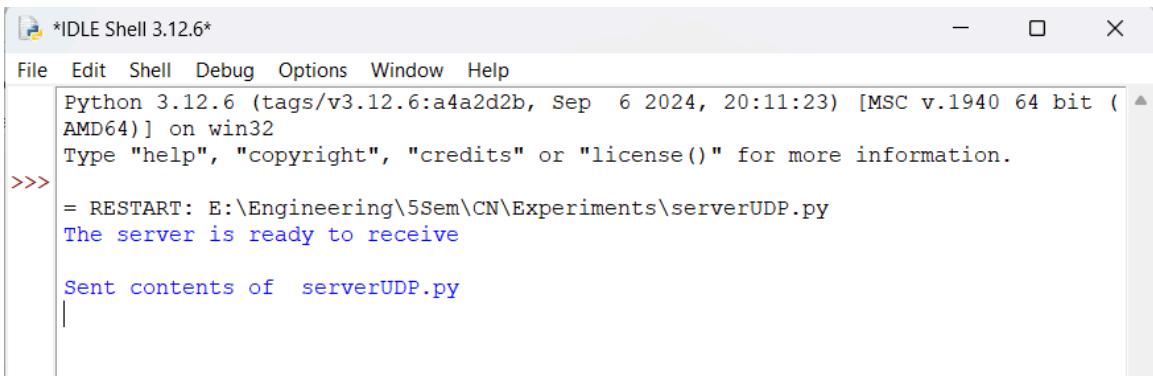
Reply from Server:

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

Lab-9  
Bafna Gold  
Date: \_\_\_\_\_  
Page: \_\_\_\_\_  
30/3/2024

2. 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 required file if present.

ClientUPP.py'

```
from socket import *
ServerName = "127.0.0.1"
ServerPort = 12000
ClientSocket = socket(AF_INET, SOCK_DGRAM)
Sentence = input("In Enter file name:")
ClientSocket.sendto(bytes(Sentence, "UTF-8"), (ServerName, ServerPort))
fileContent, serverAddress = ClientSocket.recvfrom(2048)
print("\nReply from Server:\n")
print(fileContent.decode("UTF-8"))
for i in fileContent:
    # print(str(i), end=' ')
ClientSocket.close()
ClientSocket.close()
```

ServerUPP.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(2048)  
 ServerSocket.sendto(bytes(content, "UTF-8"), ClientAddress)  
 print("In Sent Content of, end =")  
 print(Sentence)  
 for i in Sentence:  
 # print(str(i), end=' ')  
 file.close()

Output:  
 The  
 Sent  
 The  
 X  
 server is ready to receive.  
 X go/127.0.0.1 ready to receive.