

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB RECORD

Computer Network Lab (23CS5PCCON)

Submitted by

Shree Varna M (1BM22CS263)

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 **Shree Varna M (1BM22CS263)**, 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

Sl. No.	Date	Experiment Title	Page No.
1	04/10/2024	Create a topology involving multiple hubs and a switch connecting them to simulate with simple PDU.	1 - 5
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 -10
3	25/10/2024	Configure default route, static route to the router	11 -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 - 31
8	08/11/2024	Configure Web Server, DNS within a LAN.	32 - 35
9	13/11/2024	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)	36 - 38
10	13/11/2024	To understand the operation of TELNET by accessing the router in server room from a PC in IT office.	39 - 41
11	20/12/2024	To construct a VLAN and make the PC's communicate among a VLAN	42 - 44
12	20/12/2024	To construct a WLAN and make the nodes communicate wirelessly	45 - 48

Github Link:

https://github.com/Shree-varna/Computer_Network

Index-Cycle-II

Sl. No.	Date	Experiment Title	Page No.
1	15/11/2024	Write a program for error detecting code using CRC-CCITT (16-bits).	49 - 50
2	15/11/2024	Write a program for congestion control using Leaky bucket algorithm	51 - 52
3	20/12/2024	Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.	53 - 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.	56 - 58

I N D E X

Name Shree Varma M. Class BE 5th Sem.
 Roll No 1BM92CS263 Subject AF CN Lab School

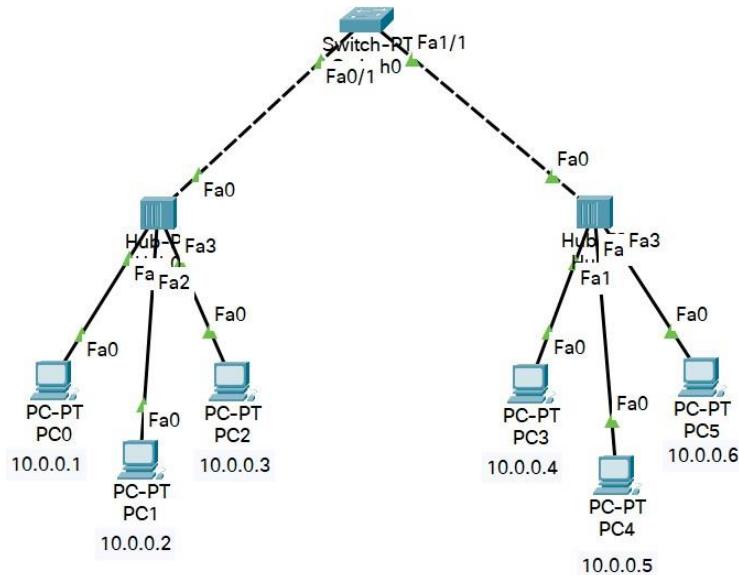
Sl No.	Date	Title	Page No.	Teacher Sign/ Remarks
		Artificial Intelligence		
		Computer Network Lab		
1)	27/9	Introduction	8/9	
2)	4/10	Routers and basic configuration of hub and Switch.	8/10	
3)	18/10	Two Router usage in Network	10	
4)	25/10	Two Routers Static Networks. 3 Routers Default Networks.	10	
5)	8/11/2024	DHCP & DNS	10/11	
6)	15/11	CRC & Leasage Bucket	10	
7)	22/11	ICMP & TTL	10/22/11	
8)	29/11	OSPF	10/29/11	

- 9) 18/11/2024 ART
Telnet
- 10) 20/11/2024 VLAN, WLAN
Socket programming

Cycle-I

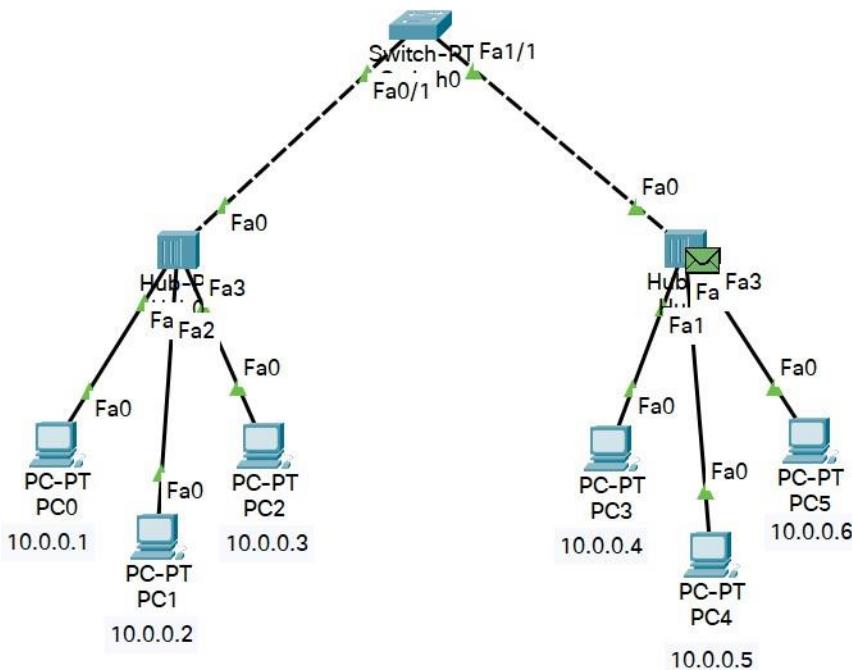
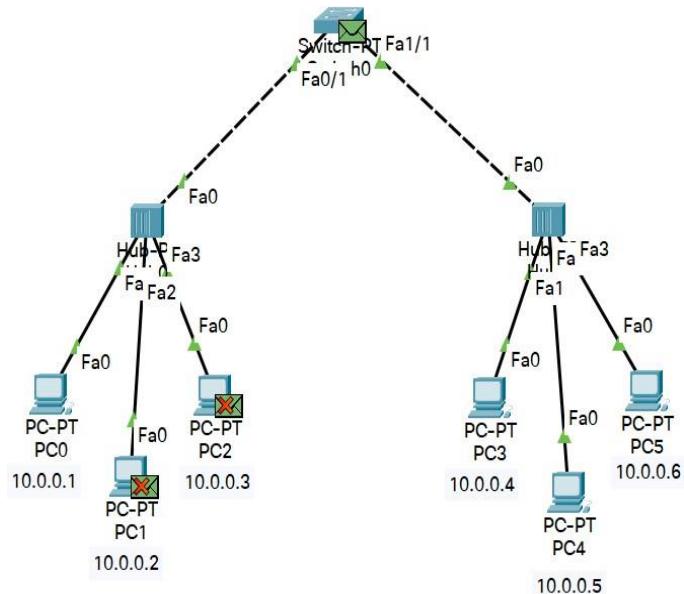
Program 1

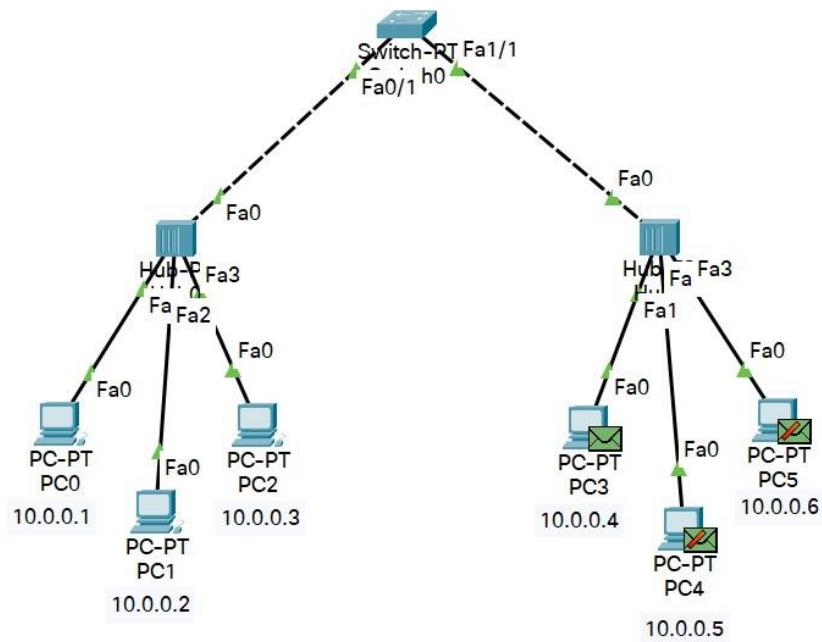
- i. Create a topology involving multiple hubs and a switch connecting them to simulate with simple PDU.
- ii. Procedure along with the topology



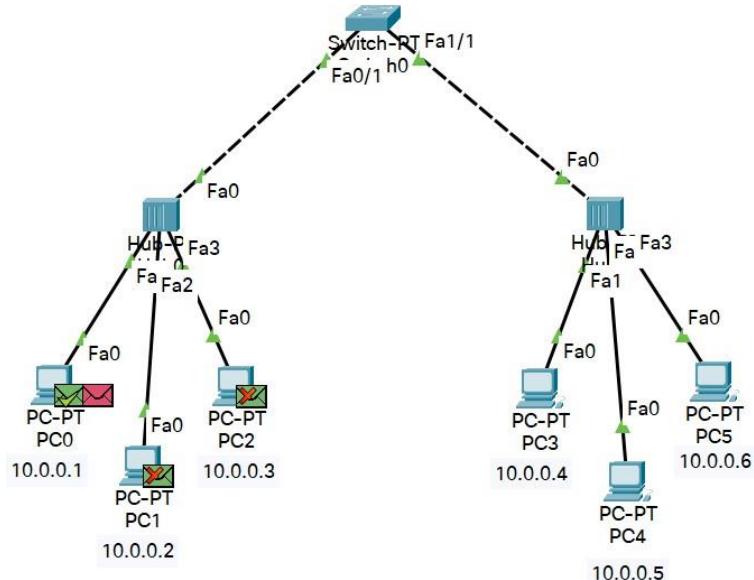
- iii. Screen shots/ output

Hub behaviour at sending end

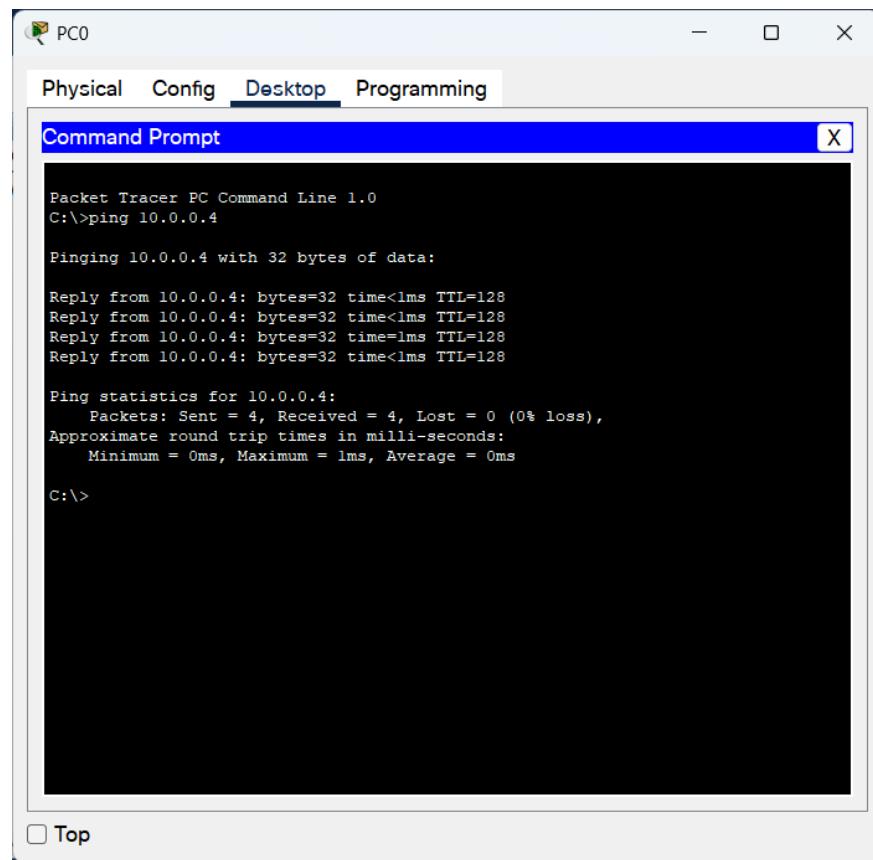




Hub behaviour when back to sender



Ping command to connectivity



The screenshot shows a software application window titled "PC0". The window has a menu bar with tabs: Physical, Config, Desktop, and Programming. The "Desktop" tab is currently selected. Below the menu is a title bar for a "Command Prompt" window. The main area of the Command Prompt window displays the output of a ping command. The output is as follows:

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

Pinging 10.0.0.4 with 32 bytes of data:

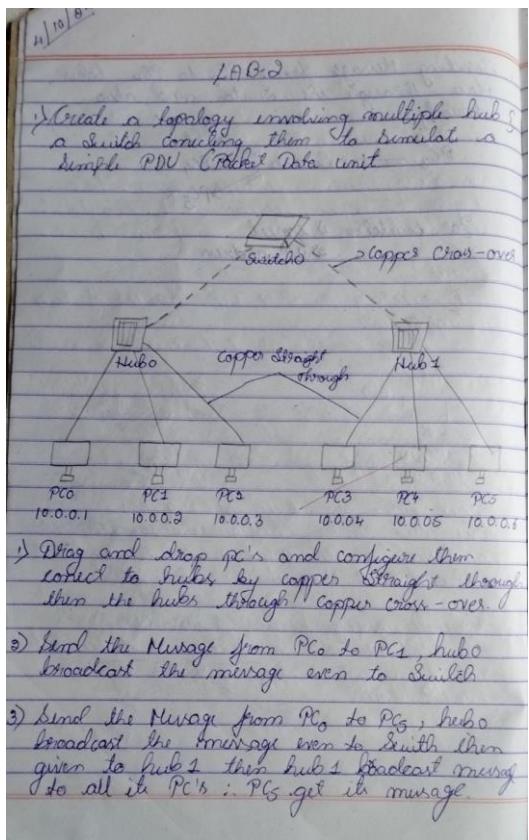
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time=1ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128

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

C:\>
```

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

iv. Observation



Router: Network layer used to connect a Network.

different Network by IP address 10.0.0.1
32 bits 10.0.0.1

Router is a configurable device by the
→ CLI interface (Command Line Interface)
→ port
→ port type no
→ port order
→ port enter

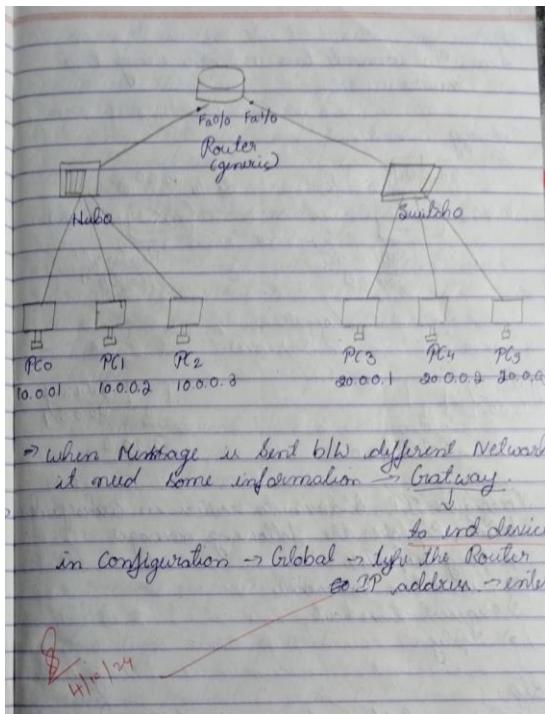
Router > enable
Router # config t

Router (Config) # interface fastethernet 0/0
Router (Config-if) # ip address 10.0.0.4 255.0.0.0
Router (Config-if) # no ip unShutdown

Router (Config-if) # exit

Router (Config) # interface fastethernet 1/0
Router (Config-if) # ip address 10.0.0.4 255.0.0.0
Router (Config-if) # no ip unShutdown

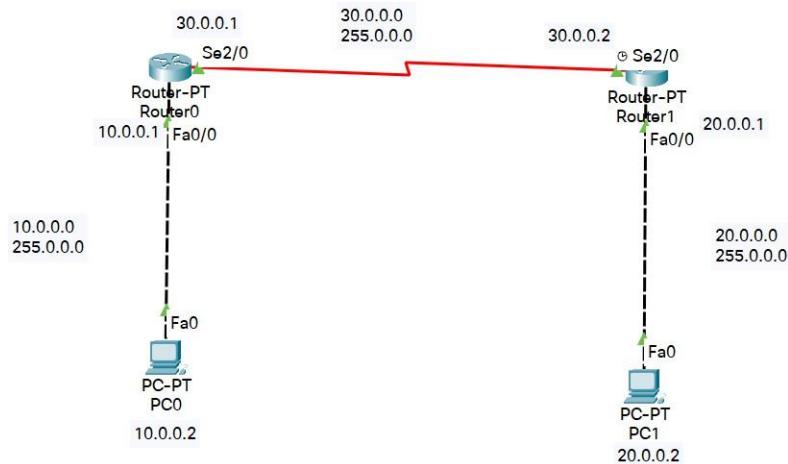
Router (Config-if) # exit



Program 2

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

- i. Procedure along with the topology

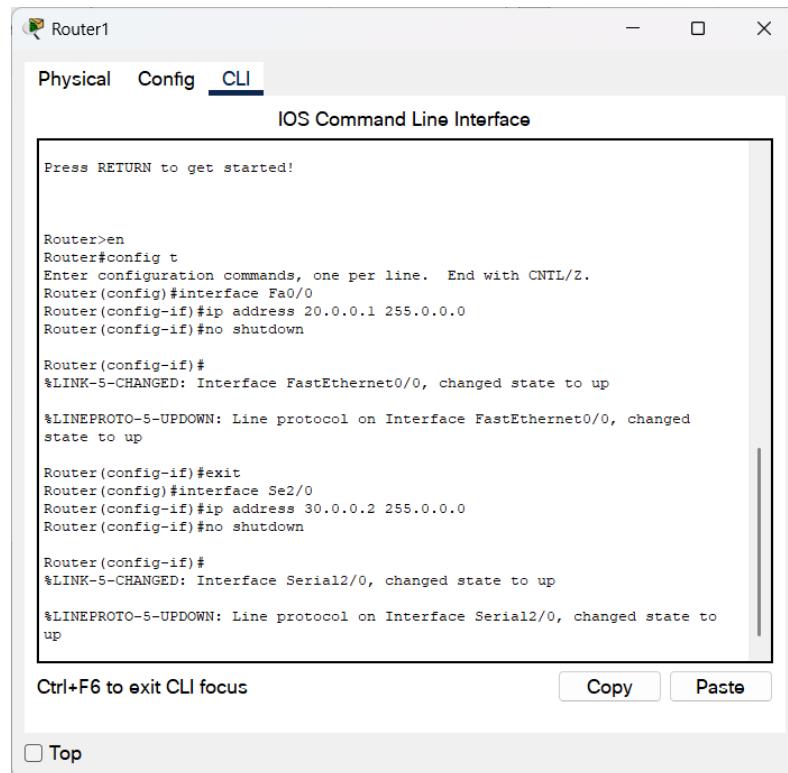


- ii. Screen shots/ output

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
Router(config-if)#
%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 image shows a window titled "Router1" with the tab "CLI" selected. The title bar also includes "Physical" and "Config". The main area is labeled "IOS Command Line Interface" and contains the following text:

```
Press RETURN to get started!

Router>en
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface Fa0/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#no shutdown

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

Router(config-if)#exit
Router(config)#interface Se2/0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to
up
```

At the bottom left is the text "Ctrl+F6 to exit CLI focus". At the bottom right are "Copy" and "Paste" buttons. Below the window is a checkbox labeled "Top".

Ip route command in Router0

```
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 20.0.0.0 255.0.0.0 30.0.0.2
Router(config)#

```

Ip route command in Router1

```
Router#config t
Enter configuration commands, one per line. End with CNTL/Z
Router(config)#ip route 10.0.0.0 255.0.0.0 30.0.0.1
Router(config)#

```

Destination host Unreachable (Before establishing network Fully)

PC1

Physical Config Desktop Programming

Command Prompt X

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

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 20.0.0.1: Destination host unreachable.

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

Request Timed Out

PC2

Physical Config Desktop Programming

Command Prompt X

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

Pinging 20.0.0.2 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 20.0.0.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Reply from Destination

PC0

Physical Config Desktop Programming

Command Prompt X

```

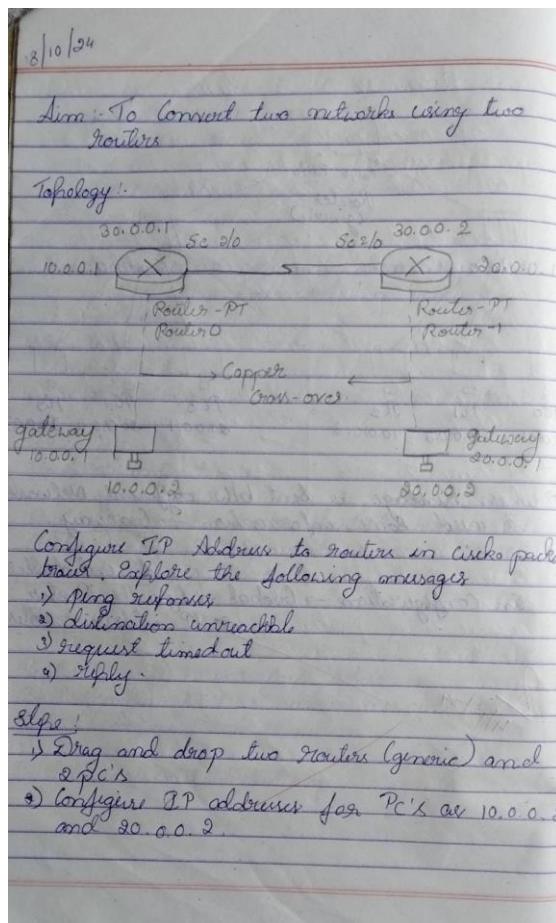
Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.2

Pinging 20.0.0.2 with 32 bytes of data:

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

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

iii. Observation



- 3) Connect routers to PC, and router 2 to PLC using Copper Cross-over wires
- 4) Click on PC A and give gateway as 10.0.0.1 and 30.0.0.1 respectively
- 5) Configure IP address for Router using fast ethernet 0/0 as 10.0.0.1 and 30.0.0.1 respectively
- 6) Connect the two routers with serial DTE cable using serial 0/0 & 2/0 ports
- 7) Configure IP address as 30.0.0.1 and 30.0.0.2 for routers
- 8) The steps to configure in CLI are
 - type **y**
 - enable
 - config t
 - interface serial 2/0
 - ip address 30.0.0.1 255.0.0.0
 - no shutdown
 - exit

9) Click on routers to go CLI and type **y**
exit until you see
Router#
Then type **show ip route**

10) Click on Router 1 & go to CLI prompt.
type → **config t**
ip route 20.0.0.0 255.0.0.0 30.0.0.1

Follow the same steps for Router 2 with ip →
10.0.0.0 255.0.0.0 30.0.0.1

- 3) Connect routers to PC, and router 2 to PLC using Copper Cross-over wires
- 4) Click on PC A and give gateway as 10.0.0.1 and 30.0.0.1 respectively
- 5) Configure IP address for Router using fast ethernet 0/0 as 10.0.0.1 and 30.0.0.1 respectively
- 6) Connect the two routers with serial DTE cable using serial 0/0 & 2/0 ports
- 7) Configure IP address as 30.0.0.1 and 30.0.0.2 for routers
- 8) The steps to configure in CLI are
 - type **y**
 - enable
 - config t
 - interface serial 2/0
 - ip address 30.0.0.1 255.0.0.0
 - no shutdown
 - exit

9) Click on routers to go CLI and type **y**
exit until you see
Router#
Then type **show ip route**

10) Click on Router 1 & go to CLI prompt.
type → **config t**
ip route 20.0.0.0 255.0.0.0 30.0.0.1

Follow the same steps for Router 2 with ip →
10.0.0.0 255.0.0.0 30.0.0.1

- 11) Pass the messages from PC1 to PC2 in simulation mode
- 12) Click the Status
- 13) Click on Router - go to CLI type **y** and exit until you see
Router#
Then type **show ip route**.
- 14) Click on PC1 and go to Command prompt and ping command as "ping 20.0.0.2" and check reply status
- 15) Do not give IP address for one PC and check destination unreachable
- 16) You can also view request timed out message

OUTPUT:

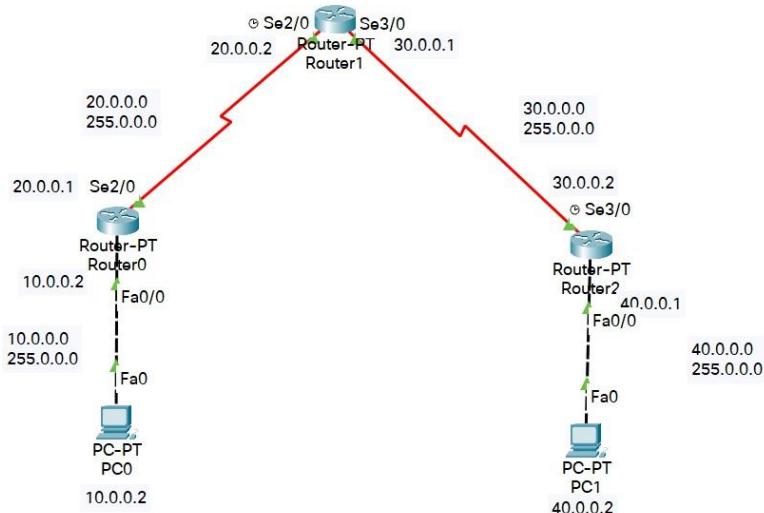
```

ping 20.0.0.2
pinging 20.0.0.2 with 32 bytes of data:
Reply from 20.0.0.2: bytes=32 time=2ms TTL=128
Reply from 20.0.0.2: bytes=32 time=1ms TTL=128
Reply from 20.0.0.2: bytes=32 time=8ms TTL=128
Reply from 20.0.0.2: bytes=32 time=1ms TTL=128
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 = 8ms, Average = 2ms
  
```

- 17) Remove gateways given in a network and pass messages and ping command.

Program 3

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



- iii. Screen shots/ output

Router0 configuration

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

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

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed
state to up
%IP-4-DUPADDR: Duplicate address 10.0.0.2 on FastEthernet0/0, sourced by
000C.CFC2.65B0

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

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

Router1 configuration

```

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

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

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

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

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

```

Ctrl+F6 to exit CLI focus

Top

Router2 configuration

```

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)#interface Serial3/0
%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       
```

Top

Static Routing:

```

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#
%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

```

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#
%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:

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

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

Gateway of last resort is 20.0.0.2 to network 0.0.0.0

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

Router#
```

Ctrl+F6 to exit CLI focus

Top

Router#enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 0.0.0.0 0.0.0.0 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#
```

Ctrl+F6 to exit CLI focus

Top

Pinging:

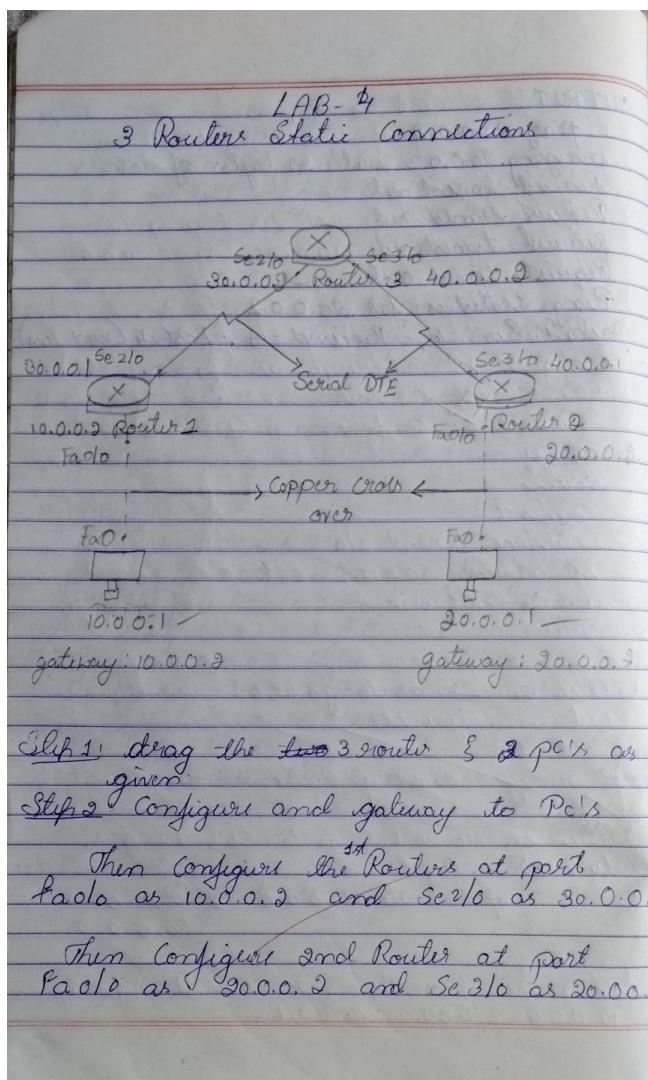
```
C:\>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

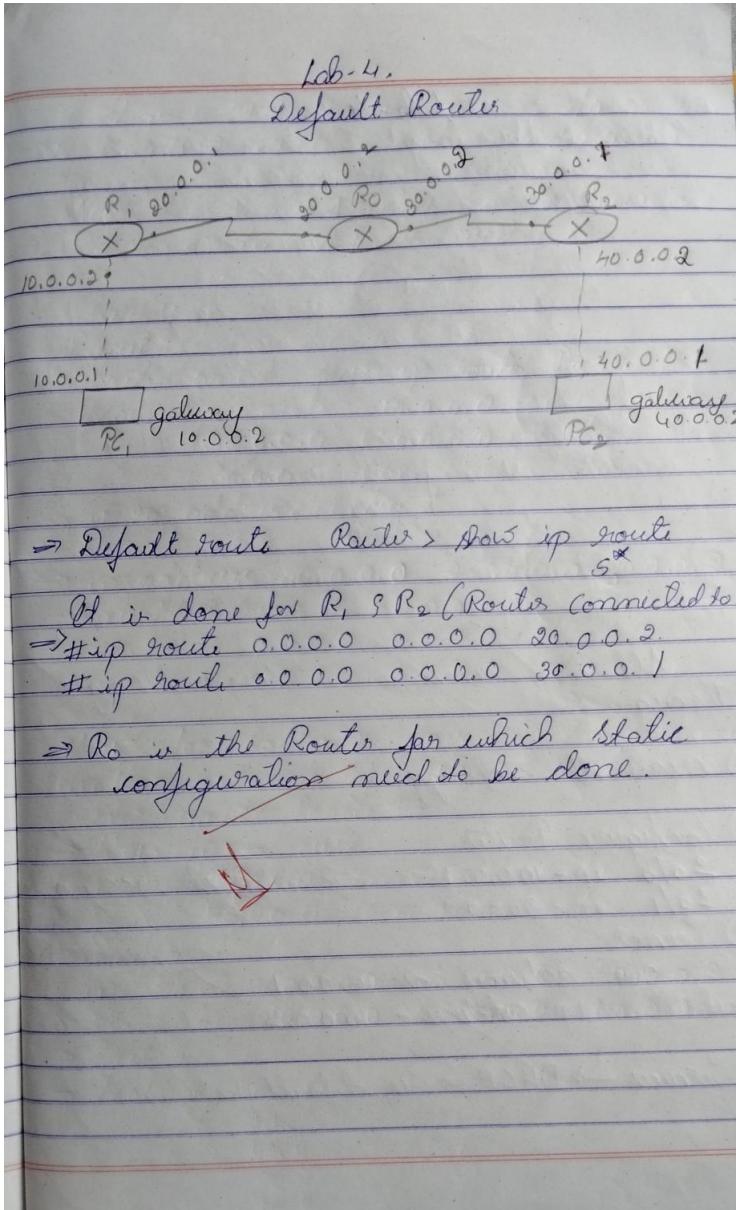
Reply from 40.0.0.2: bytes=32 time=21ms TTL=125
Reply from 40.0.0.2: bytes=32 time=17ms TTL=125
Reply from 40.0.0.2: bytes=32 time=25ms TTL=125
Reply from 40.0.0.2: bytes=32 time=2ms TTL=125

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
```

Observation:

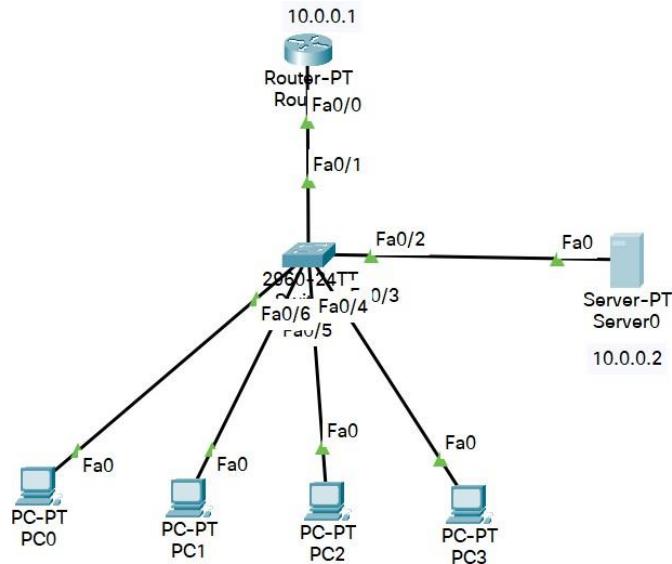


* Configure 3rd Router as Se 2/0 as ~~30.0.0.2~~ as 30.0.0.2 and Se 3/0 as 40.0.0.2
Now the Router 1 does not know about the ~~another~~ network so CLI is used.
→ Type 'y'
→ Exit
→ cat until shows ip route Router #
→ Shows ip route
10.0.0.0 connected fa0/0
30.0.0.0 Connected Se 2/0
→ ip ~~route~~ address config t
→ ip ~~route~~ 40.0.0.0 255.0.0.0 30.0.0.2
→ no shutdown 20.0.0.0 30.0.0.2
→ exit
Now for Router 2 does not know about ~~another~~ network of C1B is used.
→ exit until Router #
→ Shows ip ~~route~~
30.0.0.0 connected fa 0/0
40.0.0.0 Connected Se 3/0.
→ Config t
→ ip route 30.0.0.0 255.0.0.0 40.0.0.2
10.0.0.0 255.0.0.0 40.0.0.2.
→ Exit
→ Shows ip route
30.0.0.0
40.0.0.0
30.0.0.0
10.0.0.0



Program 4

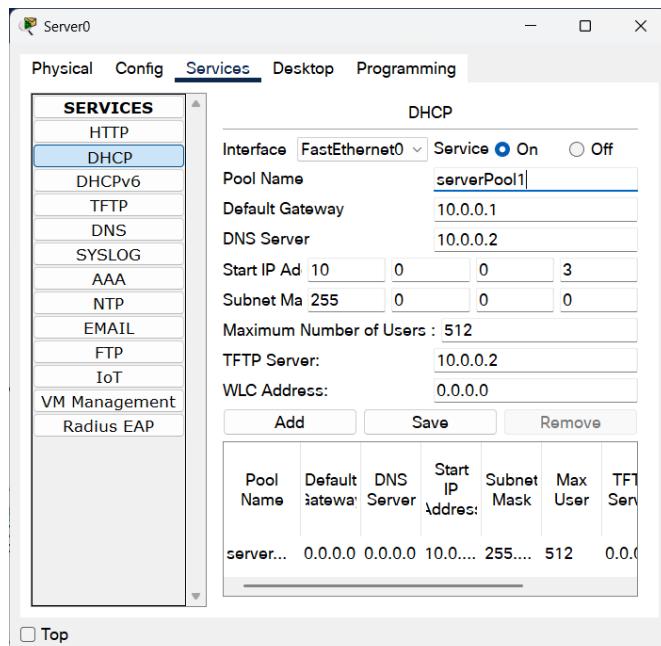
- i. Configure DHCP within a LAN and outside LAN.
 - ii. Procedure along with the topology



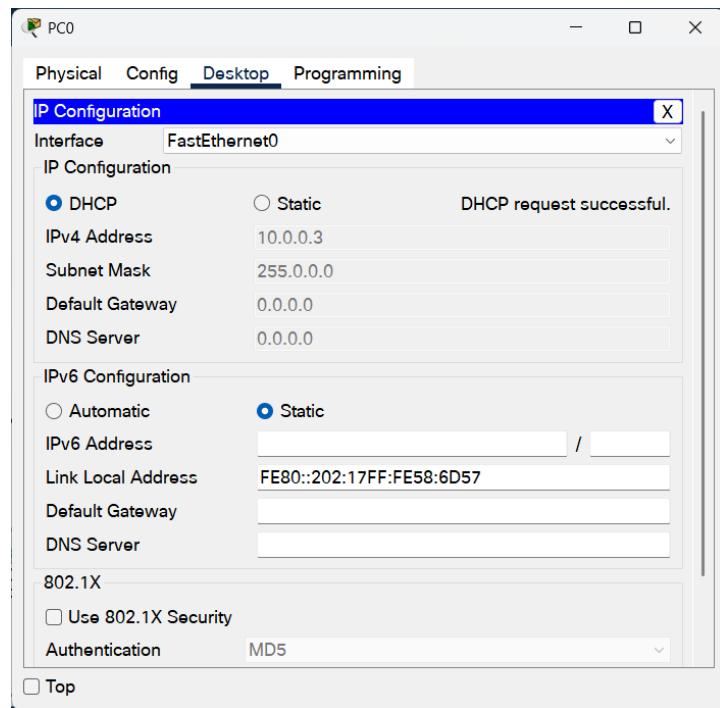
- ### iii. Screen shots/ output

DHCP Within LAN

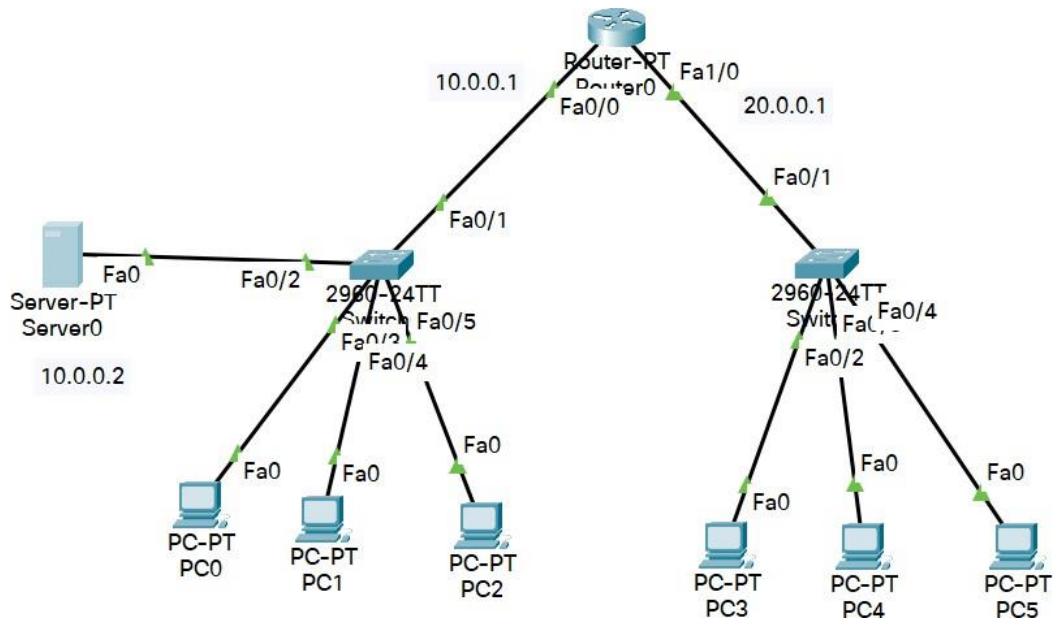
DHCP Configuration



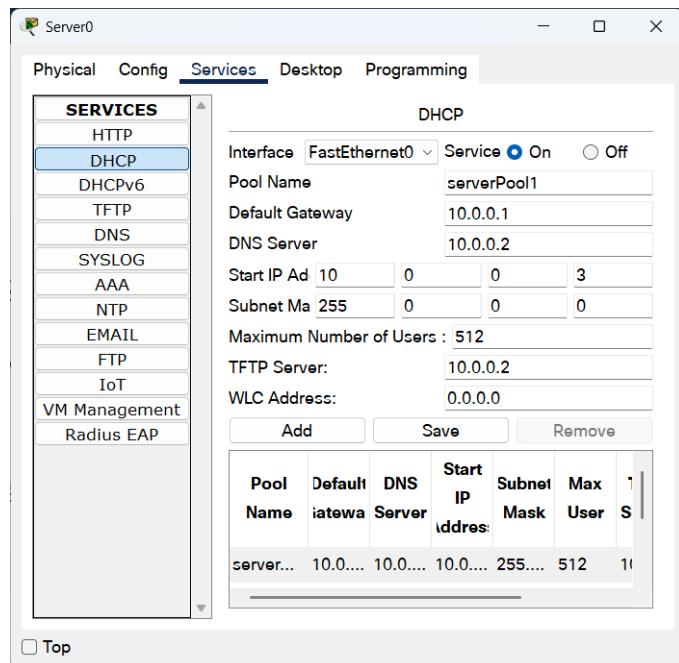
PC settings



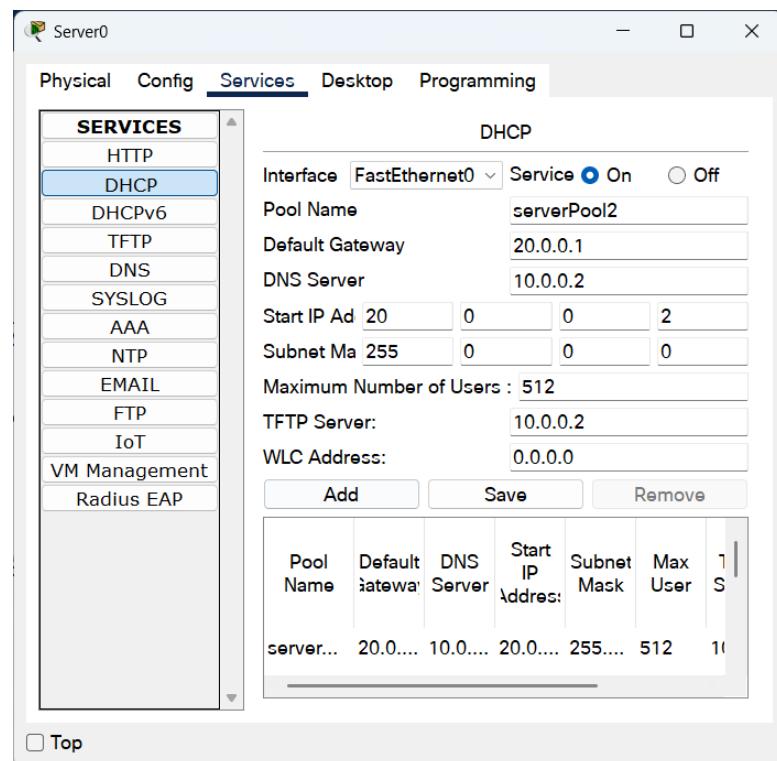
DHCP outside LAN:



DHCP configuration for inside LAN



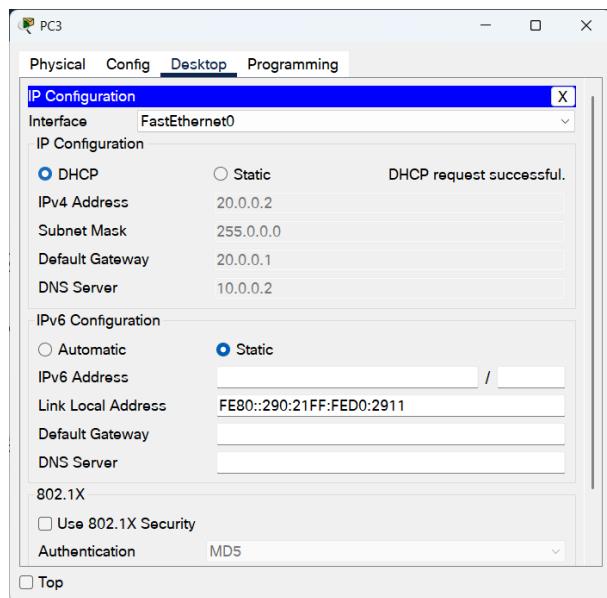
DHCP configuration for outside LAN



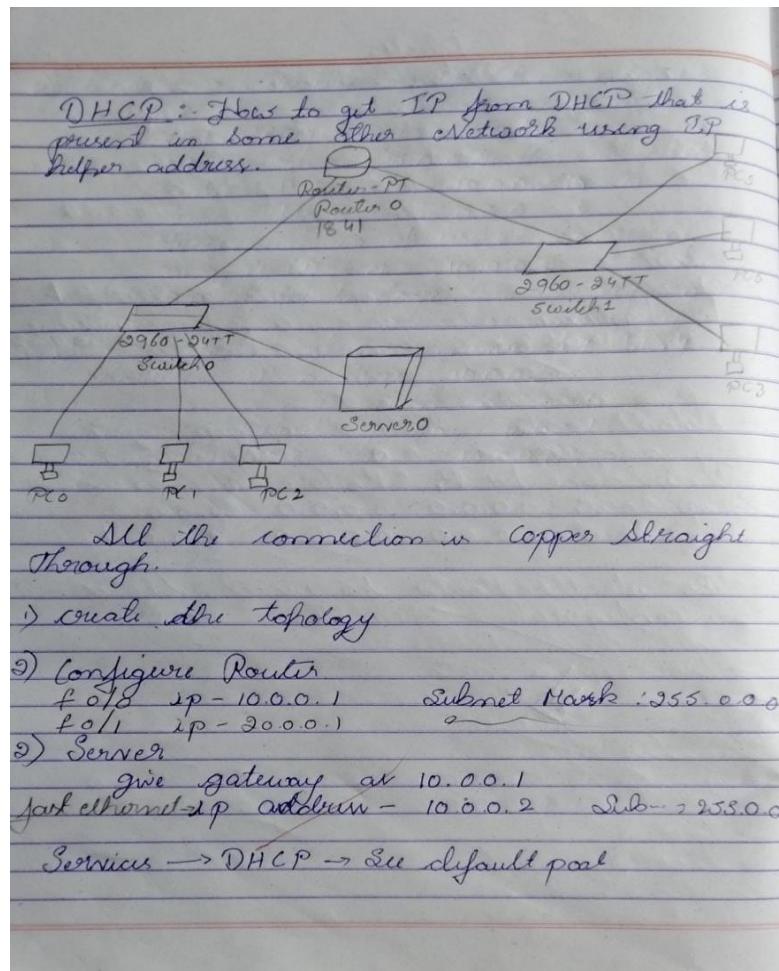
```
Ip helper command in Router
Router(config-if)#exit
Router(config)#interface Fa1/0
Router(config-if)#ip helper-address 10.0.0.2
Router(config-if)#

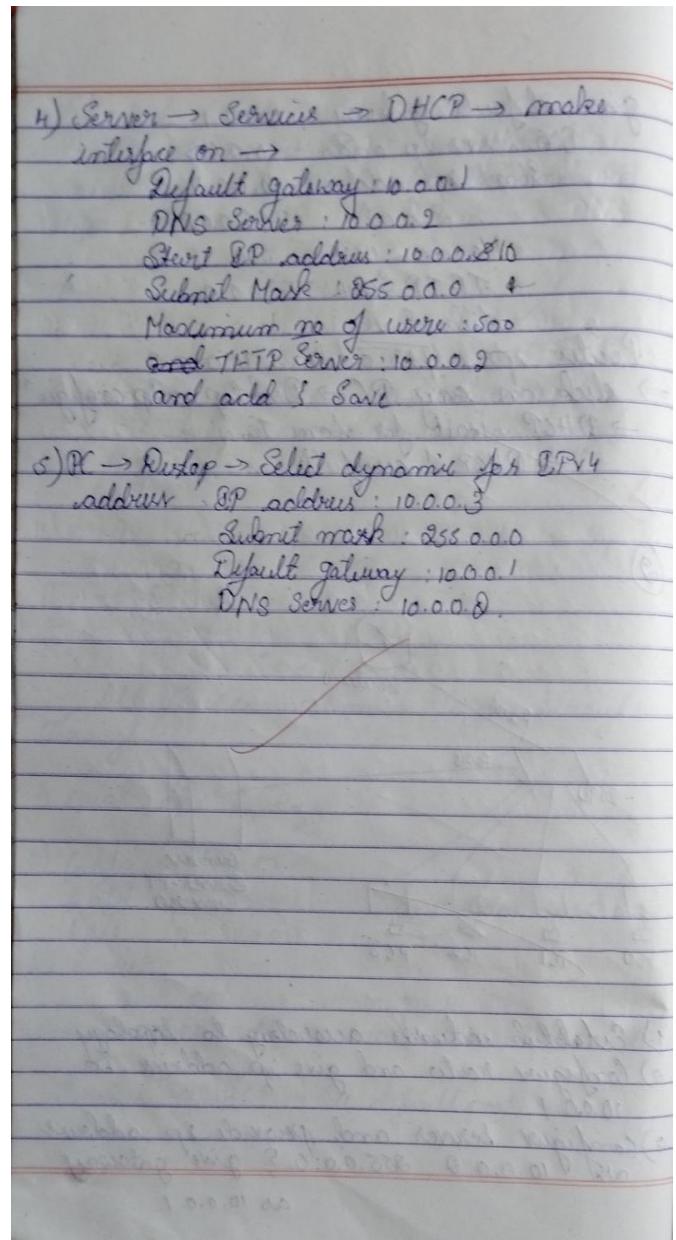
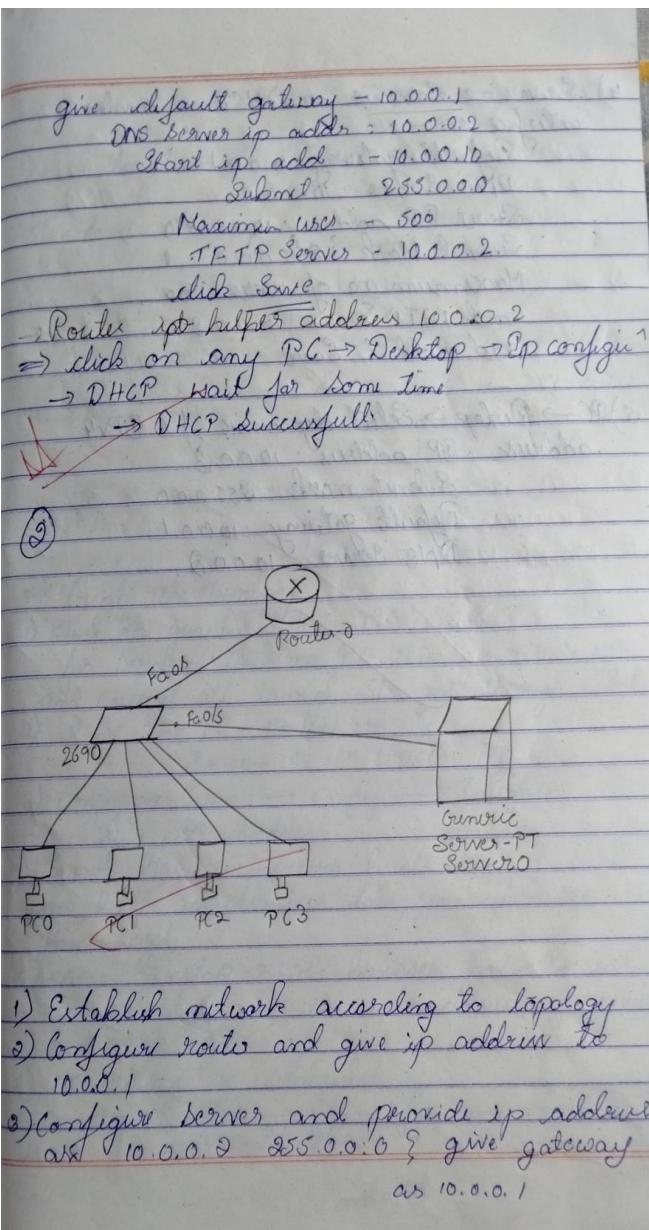
```

PC setting in another network



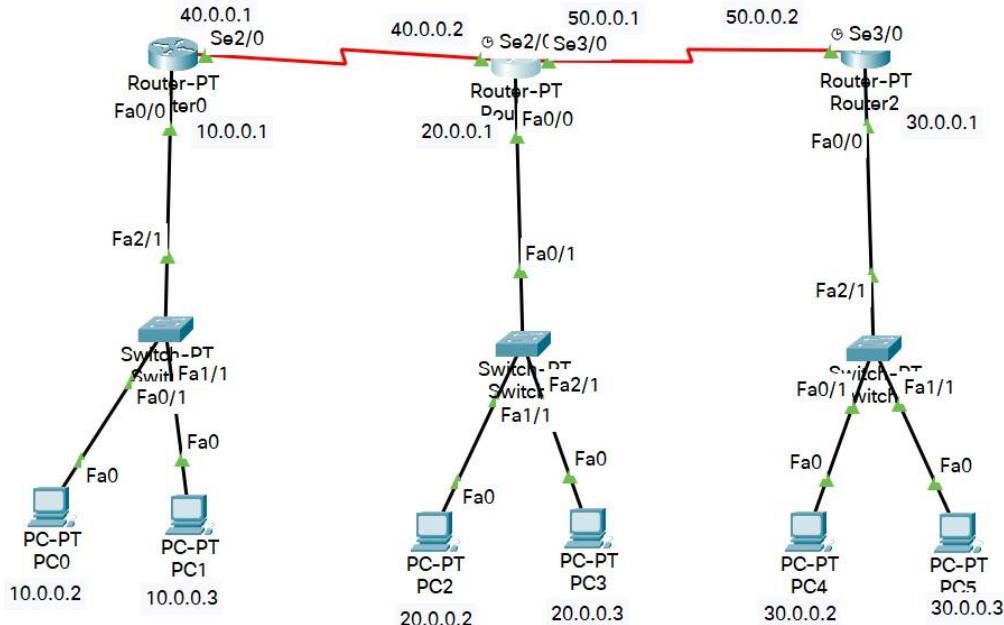
observation





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:

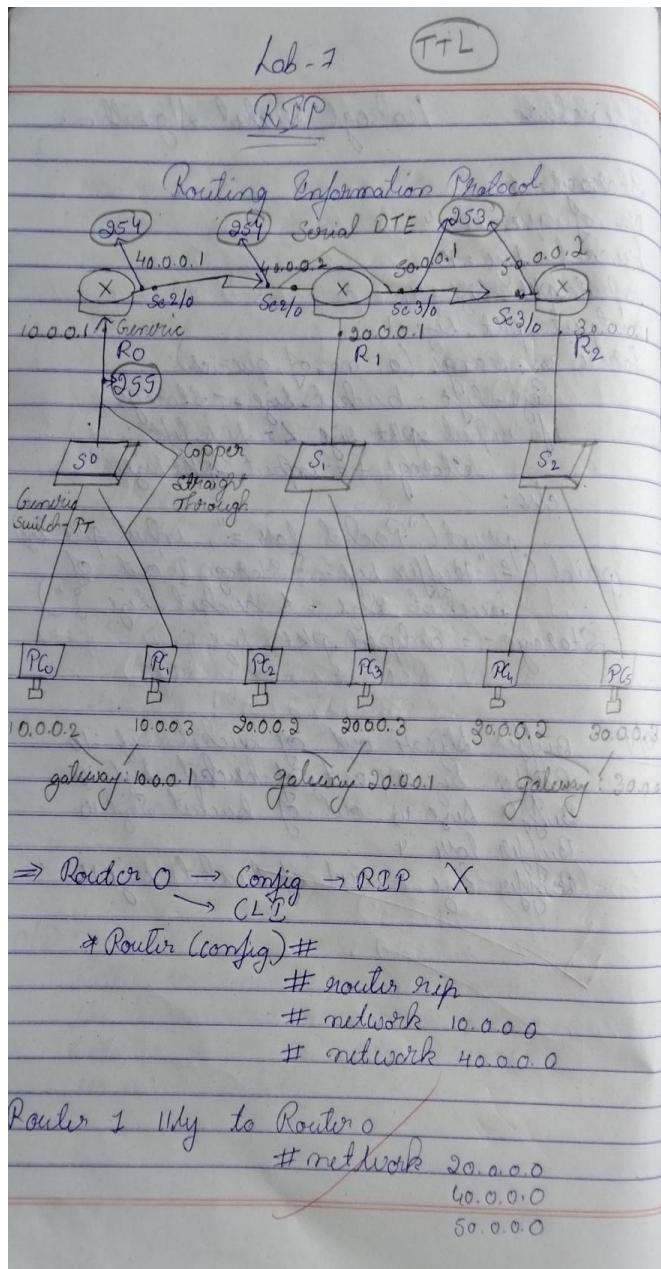
```
Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.2

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



→ Router 2
network 30.0.0.0
network 30.0.0.0

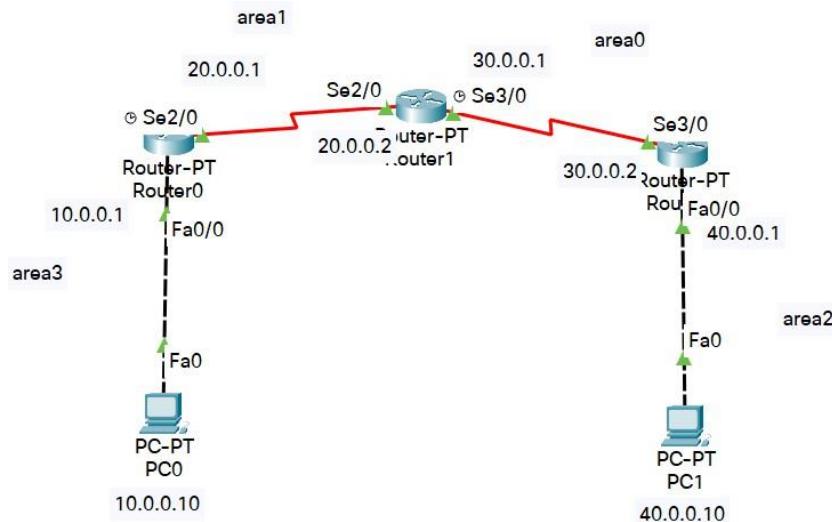
Show ip route
o/p: 30.0.0.0 - - - for o/o
30.0.0.0 - - - Serial 3/0

Output: Router/PC
pc> ping 20.0.0.0 / PC> ping 30.0.0.0

✓

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

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

Router2

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

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

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

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

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

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

OSPF Routing Protocol

Router0

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

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

Gateway of last resort is not set

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

Router1

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

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

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

Gateway of last resort is not set

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

Router2

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

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

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

Gateway of last resort is not set

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

Configure Loopback address

Router0

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

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

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

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

Router1

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

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

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

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

```

Router2

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

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

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

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

Create Virtual Link

Router0

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

Router1

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

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

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

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

Pinging

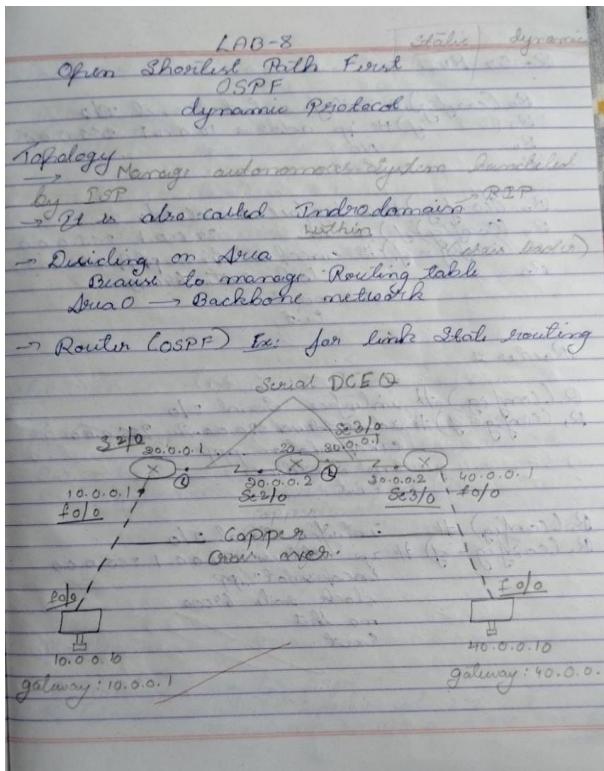
```
C:\>ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data:

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

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

Observation :



Router R0

```
R0(config)# interface FastEthernet 0/0
R0(config-if)# ip address 10.0.0.1 255.0.0.0
R0(config-if)# no shutdown
R0(config-if)# end
```

```
R0(config)# interface Serial 2/0
R0(config-if)# ip address 20.0.0.1 255.0.0.0
R0(config-if)# encapsulation ppp
R0(config-if)# clock rate 64000
R0(config-if)# no shutdown
R0(config-if)# end
```

Router 1

```
R1(config)# interface Serial 2/0
R1(config-if)# ip address 20.0.0.2 255.0.0.0
R1(config-if)# encapsulation ppp
R1(config-if)# no shutdown
R1(config-if)# end
```

```
R1(config)# interface Serial 3/0
R1(config-if)# ip address 30.0.0.1 255.0.0.0
R1(config-if)# encapsulation pp
R1(config-if)# clock rate 64000
R1(config-if)# no shutdown
R1(config-if)# end
```

Router 2

Same Machine or Customer's Local host

Phase 3:
 loopback: check its own stage (active)

Router D

```
R0(config-if)# interface loopback 0
R0(config-if)# ip add 172.16.1.255 255.255.0.0
R0(config-if)# no shutdown
```

Router 2

```
R2(config-if)# interface loopback 0
R2(config-if)# ip add 172.16.1.253 255.255.0.0
R2(config-if)# no shutdown
```

Router 3

```
R3(config-if)# interface loopback 0
R3(config-if)# ip add 172.16.1.254 255.255.0.0
R3(config-if)# no shutdown
```

→ Show up Router (R2)

Output:
 C 0.0.0.0/8 [110/128] via 30.0.0.1 Serial3/0
 C 40.0.0.0/8 is directly connected, F2/0
 C 30.0.0.0/8

Step 6:
Phase 4:

Router 0

```
R0(config)# interface ospf 1
R0(config-router)# area 1 virtual-link 2/0
R0(config-router)# area 0 is central it knows Area 1 & Area 2
R0(config-router)# area 0 is unaware of 10 network
```

Router 1

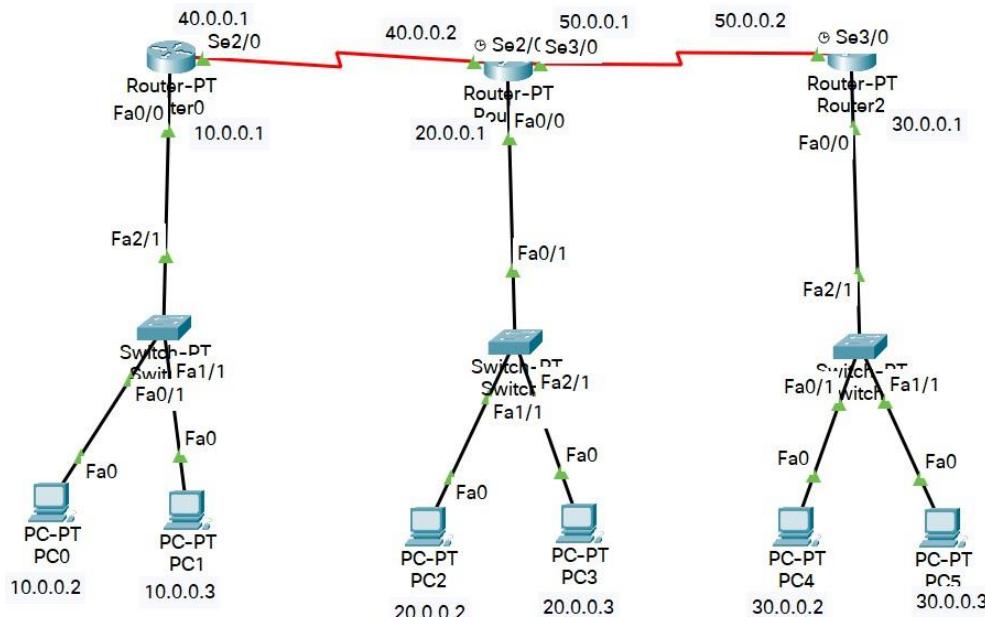
```
R1(config-router)# area 1 virtual-link 1/1
R1(config-router)# area 1 virtual-link 3/0
```

Step 7:
 Shows ip route (R3)

Output:
 O 0.0.0.0/8 [110/128] via 30.0.0.1 Serial3/0
 C 40.0.0.0/8 F2/0
 O 0.0.0.0/8 [110/129] via 20.0.0.1 Serial 3/0
 C 30.0.0.0/8 Serial3/0

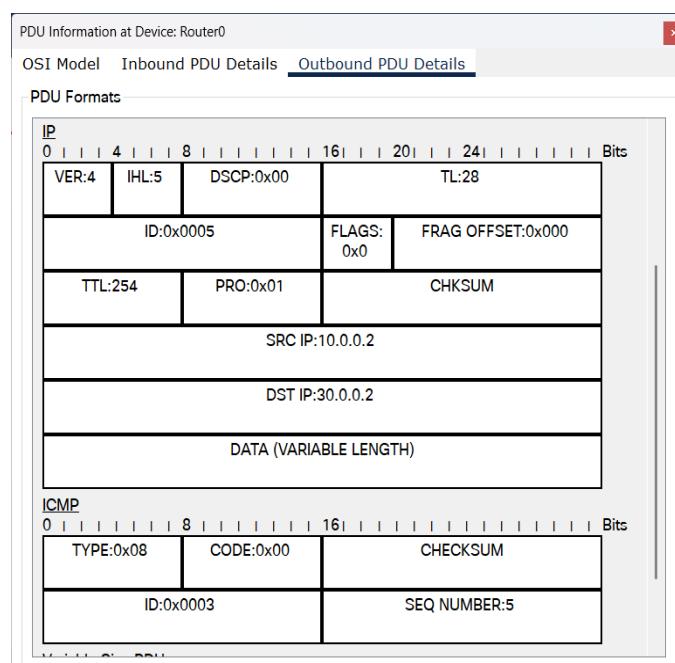
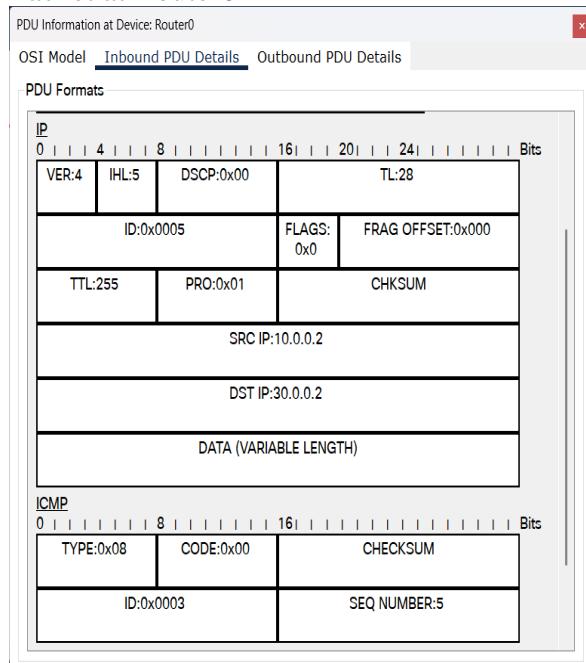
Program 7

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



- Screen shots/ output

Packet at Router0



Packet at Router1

PDU Information at Device: Router1

OSI Model Inbound PDU Details Outbound PDU Details

PDU Formats

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

PDU Information at Device: Router1

OSI Model Inbound PDU Details Outbound PDU Details

PDU Formats

IP						
0	4	8	16	20	24	Bits
VER:4	IHL:5	DSCP:0x00	TL:28			
ID:0x0005		FLAGS: 0x0	FRAG OFFSET:0x000			
TTL:253	PRO:0x01	CHKSUM				
SRC IP:10.0.0.2						
DST IP:30.0.0.2						
DATA (VARIABLE LENGTH)						
ICMP				Bits		
0	8	16				
TYPE:0x08	CODE:0x00	CHECKSUM				
ID:0x0003		SEQ NUMBER:5				

Packet at Router2

PDU Information at Device: Router2

OSI Model Inbound PDU Details Outbound PDU Details

PDU Formats

IP						
0	4	8	16	20	24	Bits
VER:4	IHL:5	DSCP:0x00	TL:28			
ID:0x0005		FLAGS: 0x0	FRAG OFFSET:0x000			
TTL:253	PRO:0x01	CHKSUM				
SRC IP:10.0.0.2						
DST IP:30.0.0.2						
DATA (VARIABLE LENGTH)						
ICMP				Bits		
0	8	16				
TYPE:0x08	CODE:0x00	CHECKSUM				
ID:0x0003		SEQ NUMBER:5				

PDU Information at Device: Router2

OSI Model Inbound PDU Details Outbound PDU Details

PDU Formats

IP						
0	4	8	16	20	24	Bits
VER:4	IHL:5	DSCP:0x00	TL:28			
ID:0x0005		FLAGS: 0x0	FRAG OFFSET:0x000			
TTL:252	PRO:0x01	CHKSUM				
SRC IP:10.0.0.2						
DST IP:30.0.0.2						
DATA (VARIABLE LENGTH)						
ICMP				Bits		
0	8	16				
TYPE:0x08	CODE:0x00	CHECKSUM				
ID:0x0003		SEQ NUMBER:5				

iv. Observation

TTL

click on right bar top
observe

Router → Router []
OSI layer

0
TTL:255 → after ping

After packet transfer from 10.0.0.0 → 30.0.0.0
also captured in eight-tap

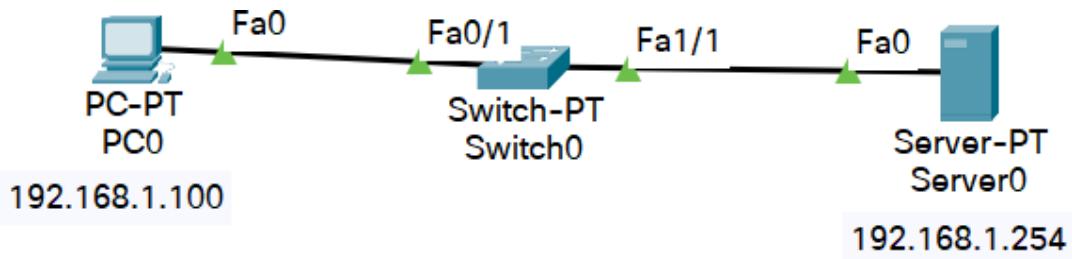
↓
wait until
Router → Router []
last time at device Click
observe Inbound PDU Details
TTL: 254

outbound PDU Details
TTL: 254

But initially it will be 255

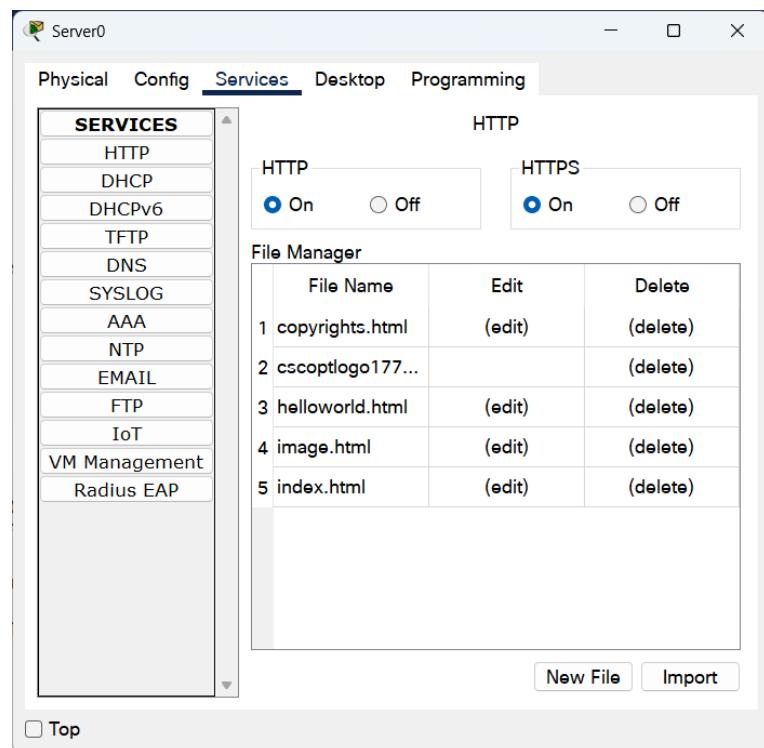
Program 8

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

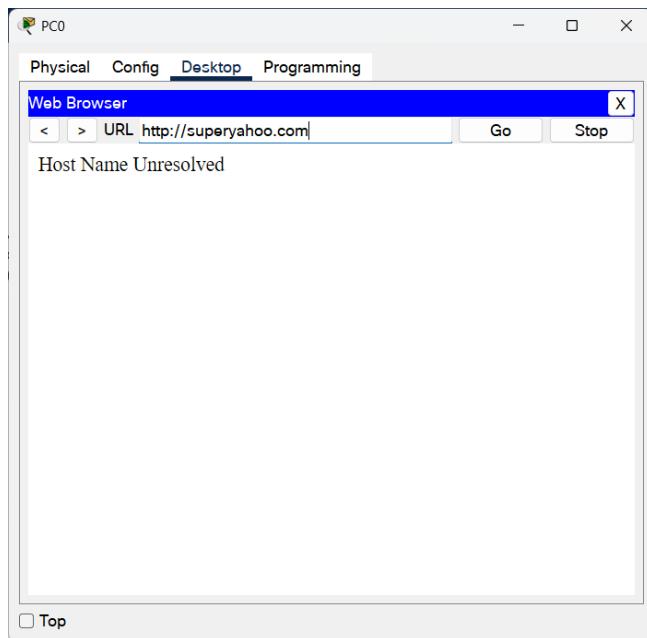
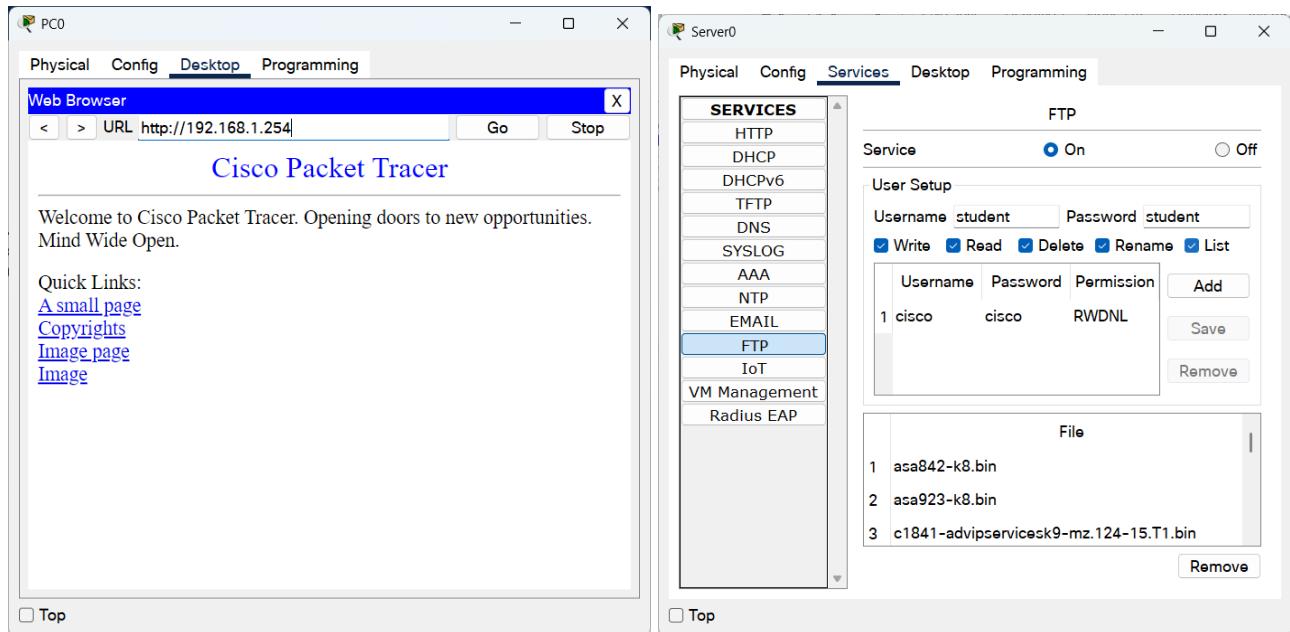


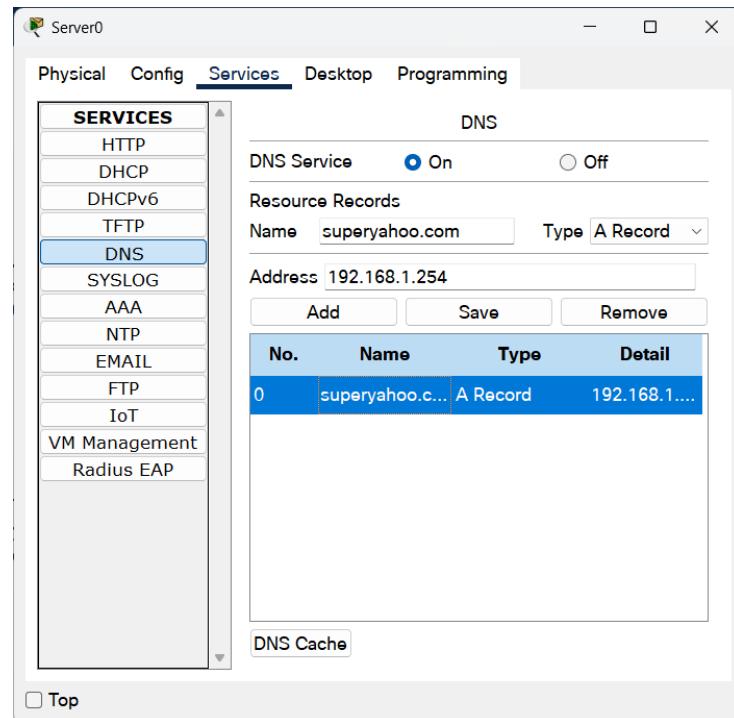
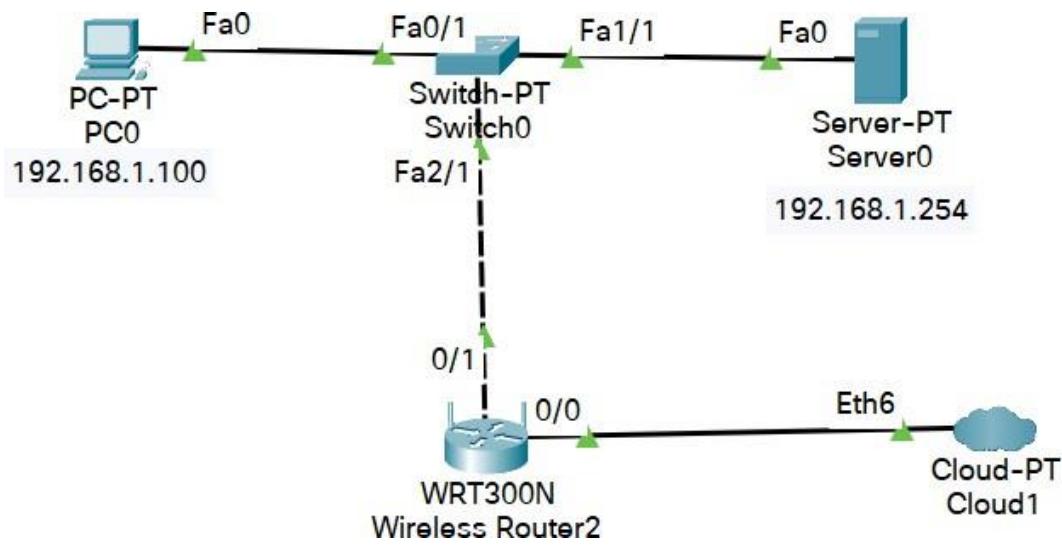
- iii. Screen shots/ output

Server's services

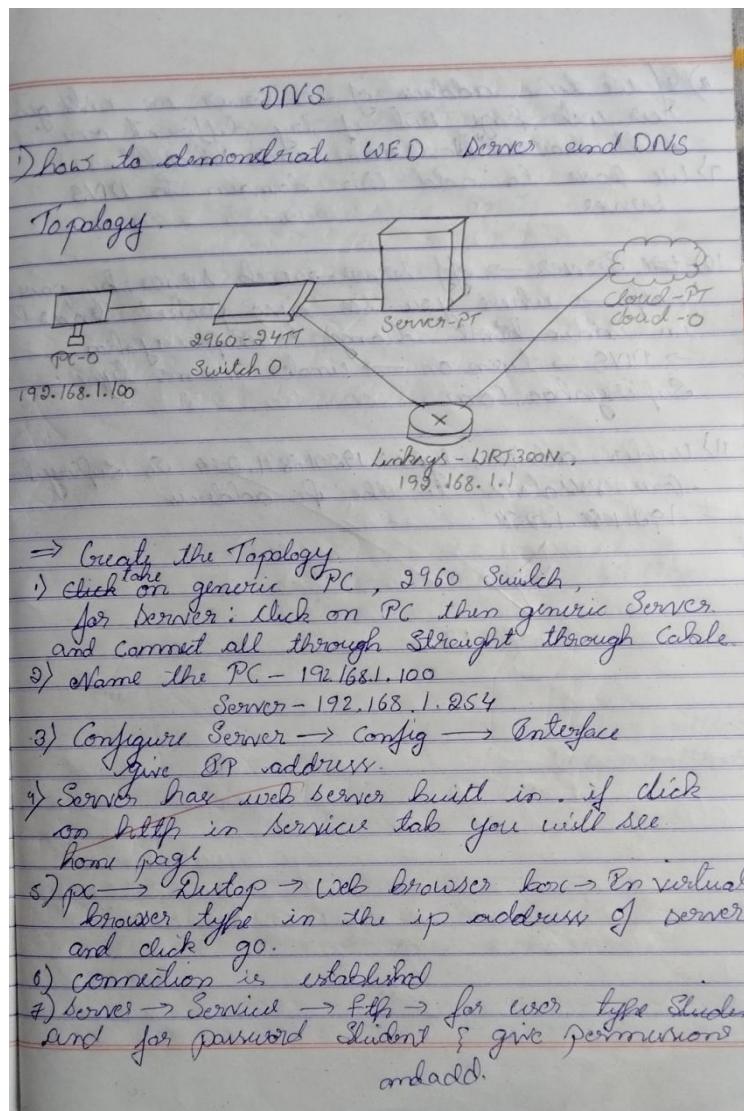


PC's Web Browser





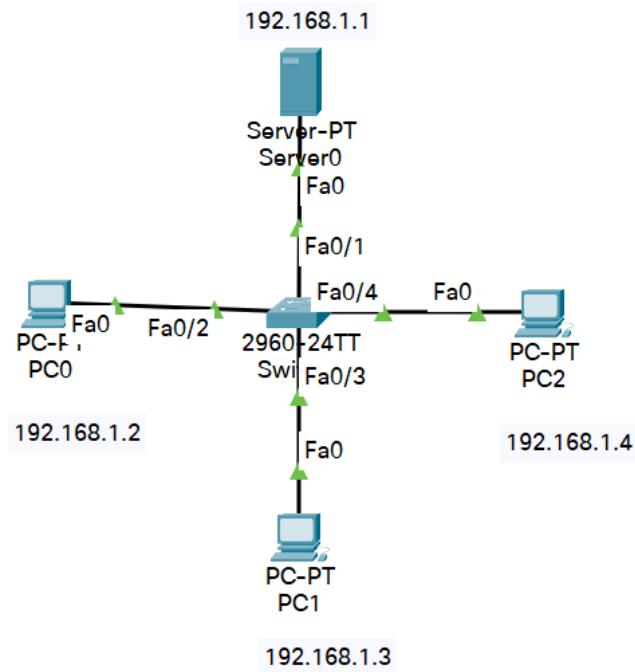
iv. Observation



- 8) If we type address of our server we will get our web page but if try different one it will show "Host name unreachable".
 - 9) We have to add this domain to DNS server.
 - 10) tell Server → off server → web server by passing http → where you see circled packet trace in index.html change to Super yahoo → DNS → turn on → under name type: Superyahoo.com.
 - 11) Under address type: 192.168.1.254 So Superyahoo.com resolves to the ip address 192.168.1.254
- ↙ ↘

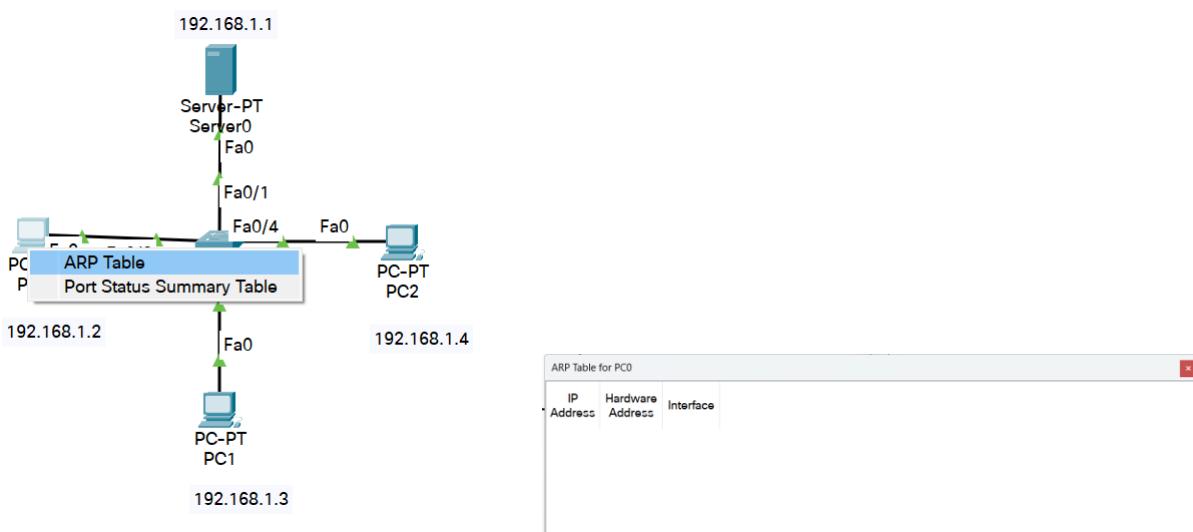
Program 9

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

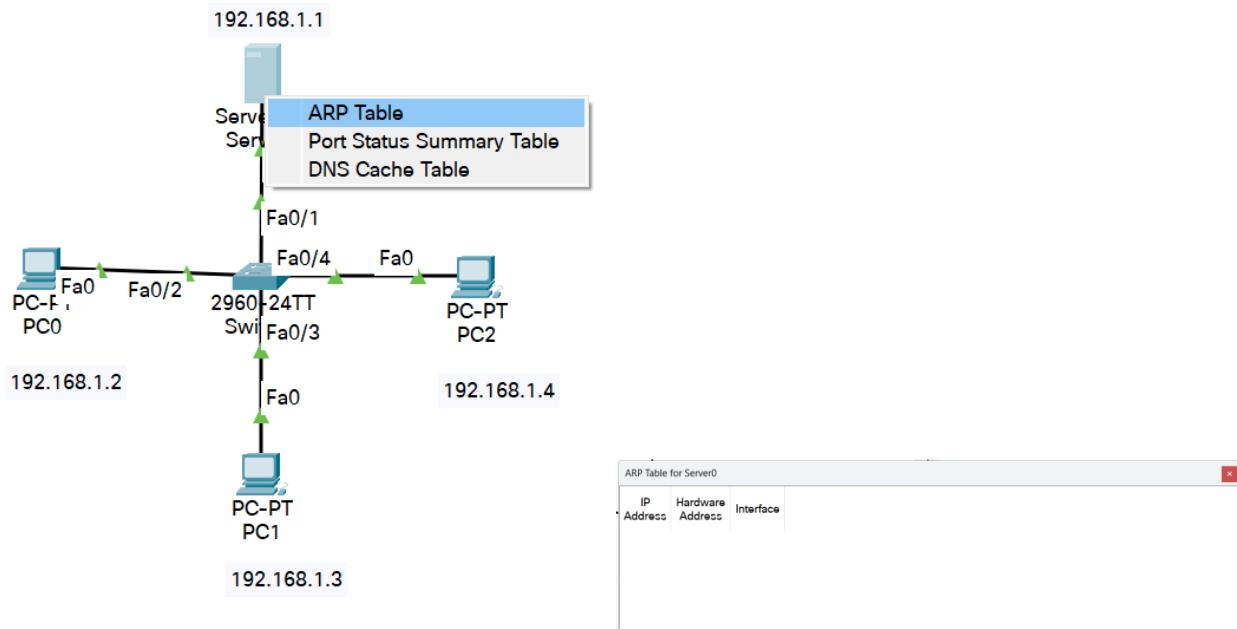


- iii. Screen shots/ output

ARP Table of PC



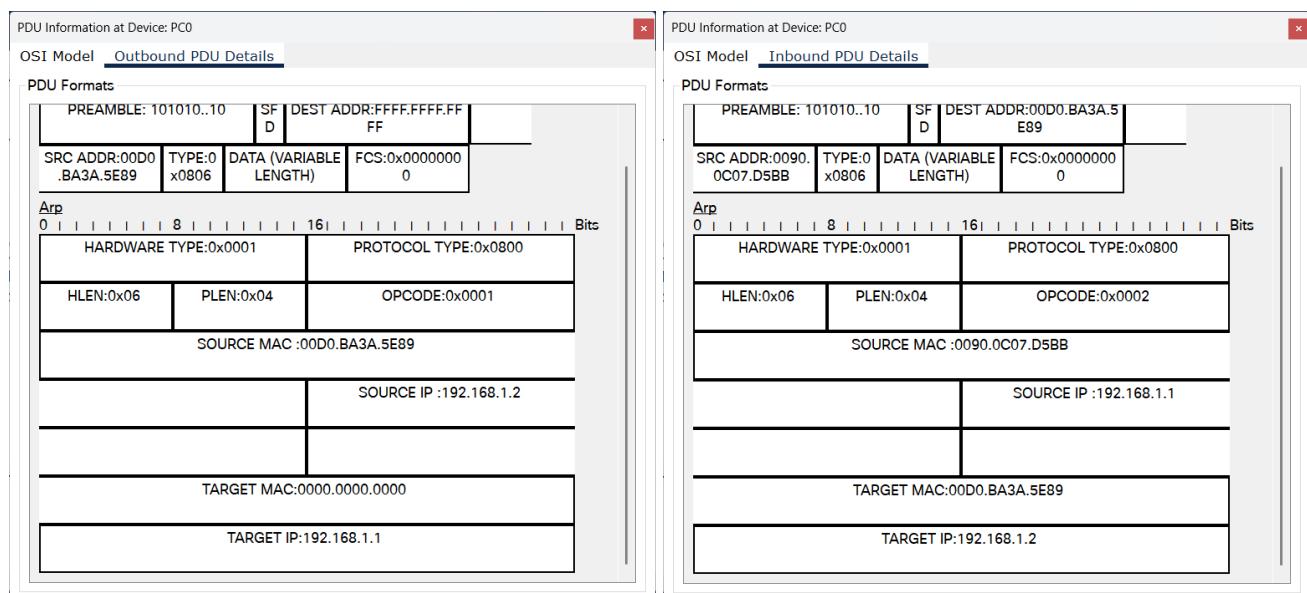
ARP Table of Server



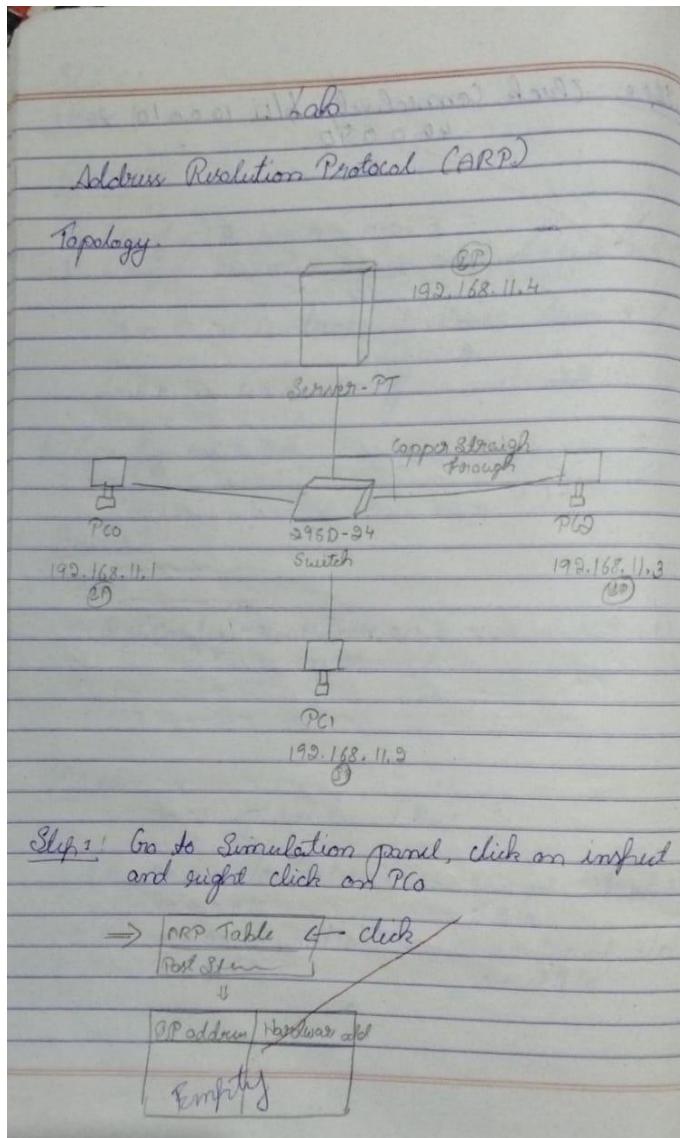
Command at PC

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

Pinging in Simulation Mode



iv. Observation



→ O/P

check some for Server and make sure its table is also Empty

Step 2: PC → Command prompt

PC > arp -a
→ No ARP Entries Found.
PC > ping 192.168.11.4

→ pinging 192.168.11.4 with 32 bytes of data:
at this time 2 packets are created
ICMP & ARP at right top

Step 3: Hover on the packet on PC0 to check the type of packet
Right-ARP

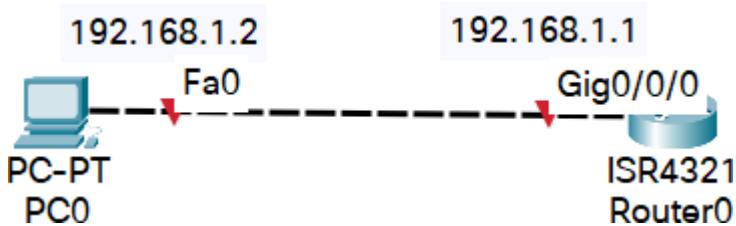
click on ARP packet
→ Outbound PDU Details to See packet Structure

Step 4: click on capture/forward button to start the simulation
→ ping req from PC0 to Switch (See in right top)
click again (Switch to Server)
click again twice (Server to PC0)

Step 5: click capture to See ICMP packet Movements
click ARP table & see entries.

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

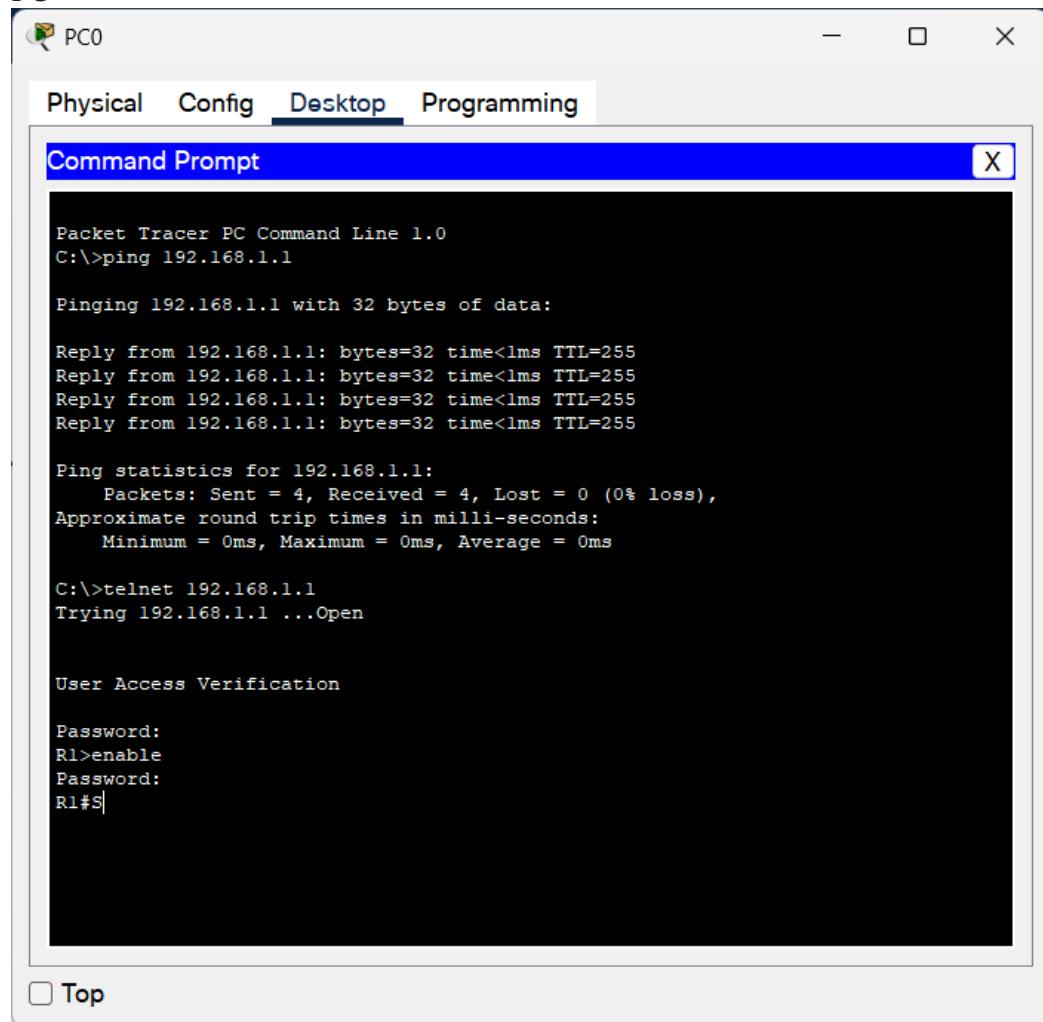
R1(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to up

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

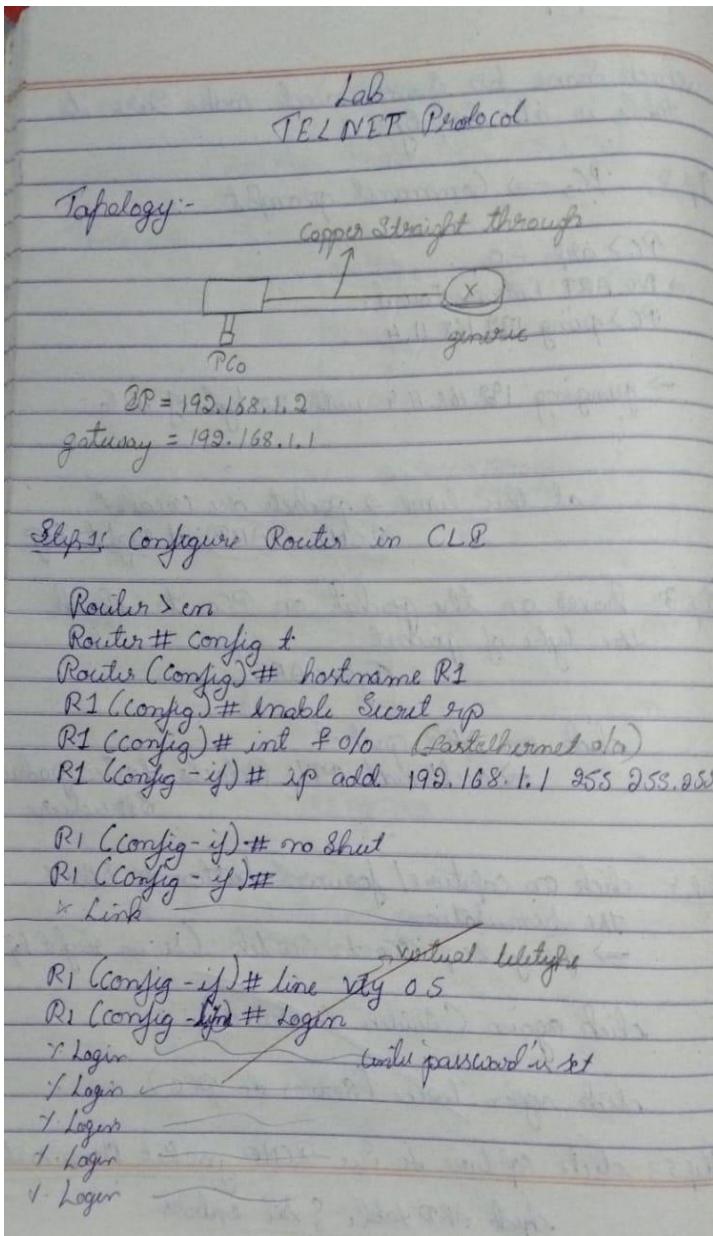
R1(config-if)#line vty 0 5
R1(config-line)#login
% Login disabled on line 2, until 'password' is set
% Login disabled on line 3, until 'password' is set
% Login disabled on line 4, until 'password' is set
% Login disabled on line 5, until 'password' is set
% Login disabled on line 6, until 'password' 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



R1 (config-line)# password tp
R1 (config-line)# exit
R1 (config)#exit
R1#
! sys

R1 # (Wr)
→ Building Configuration
POK

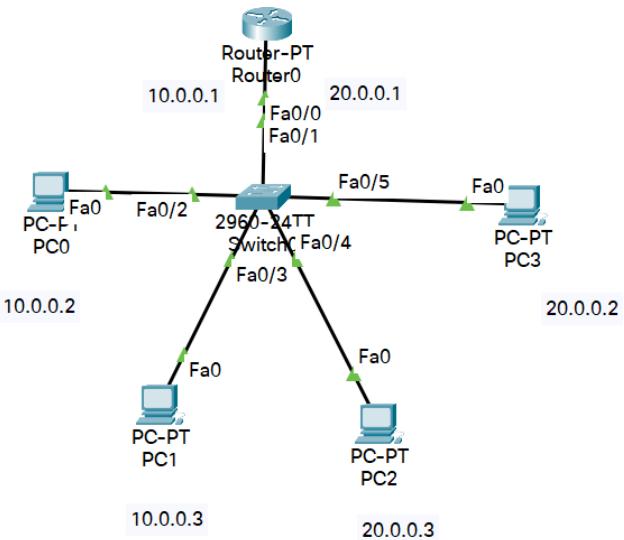
R1#
!

Step 2! PC → Command prompt

PC> ping 192.168.1.1

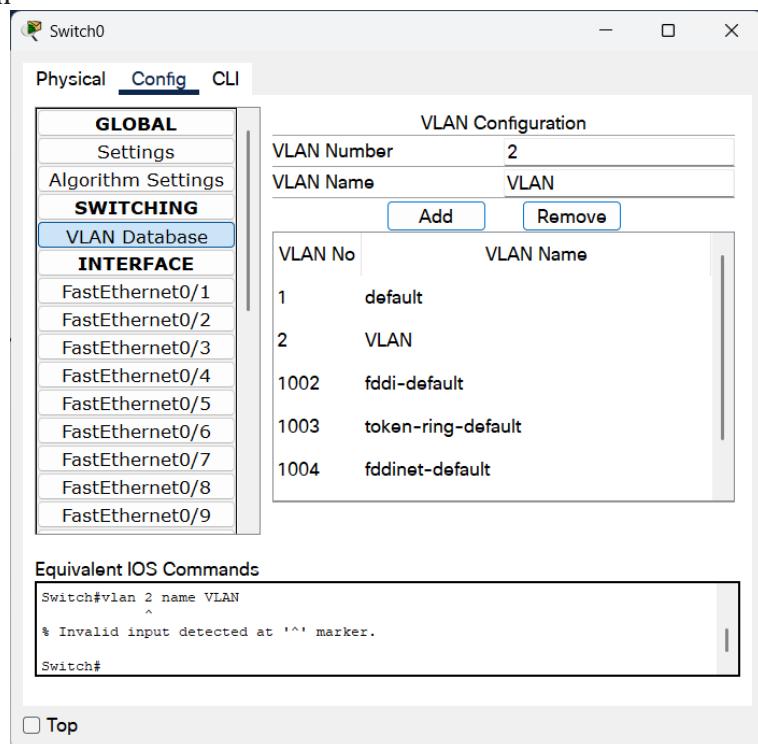
Program 11

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

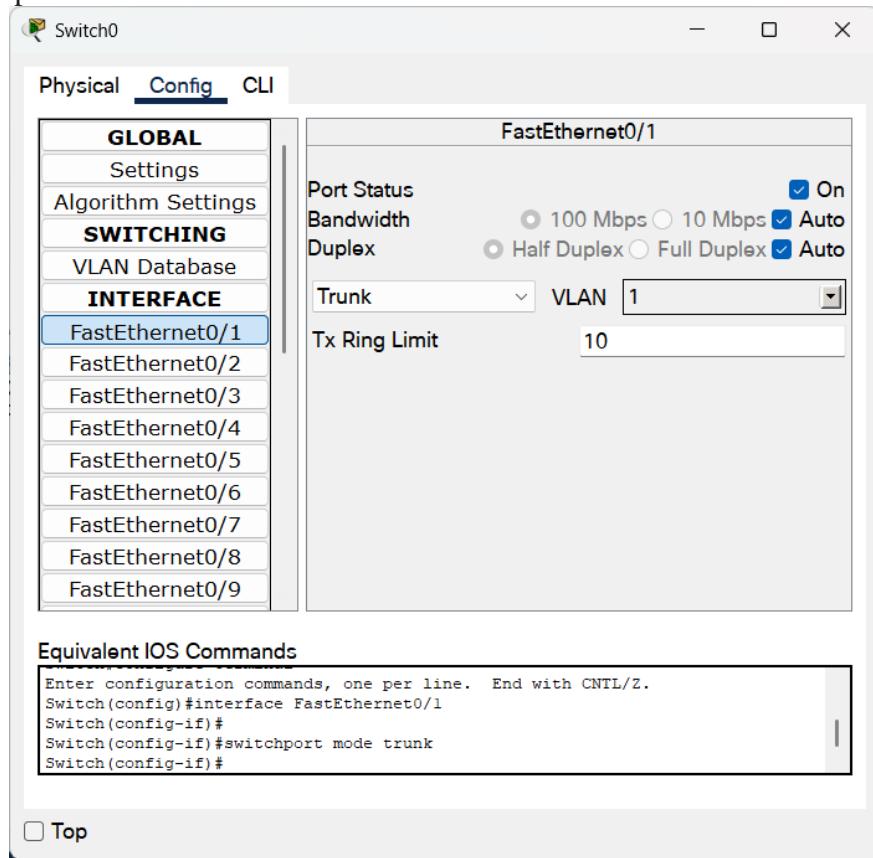


- iii. Screen shots/ output

Switch Configuration



Configuration of port connected to router



Configuration of ports connected to other networks

This screenshot shows the configuration interface for "Switch0" with two separate panels for "FastEthernet0/4" and "FastEthernet0/5".

- FastEthernet0/4 Configuration:** The "Access" dropdown is set to "VLAN" and the "VLAN" dropdown is set to "2". The "Tx Ring Limit" is set to "10".
- FastEthernet0/5 Configuration:** The "Access" dropdown is set to "VLAN" and the "VLAN" dropdown is set to "2". The "Tx Ring Limit" is set to "10". The "VLAN" dropdown also shows "1:default" and "1002:fddi-default" as options.

Below each panel is an "Equivalent IOS Commands" section:

- FastEthernet0/4:**

```

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

```
- FastEthernet0/5:**

```

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

```

Configuartion of Router

Router0

Physical Config CLI

IOS Command Line Interface

```

Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface Fa0/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shutdown

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

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

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

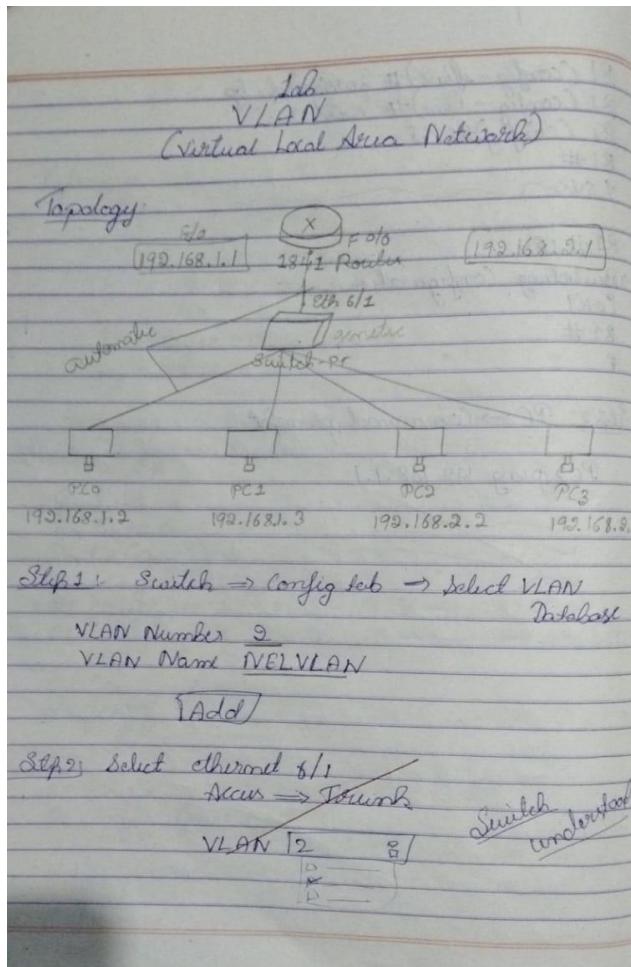
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0.1, changed
state to up

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

```

Ctrl+F6 to exit CLI focus

Top



Step 3: Router → Config tab → VLAN
Enter 2
NEWVLAN
Router undocked

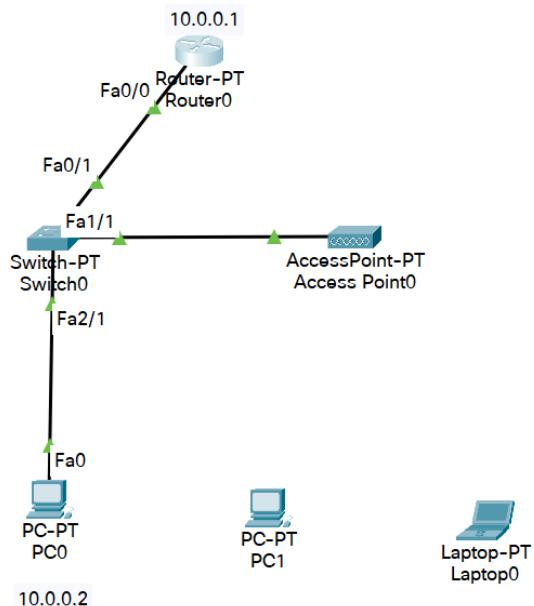
Go to CLI

Router(vlan)# exit
→ Apply Configuration
exit.

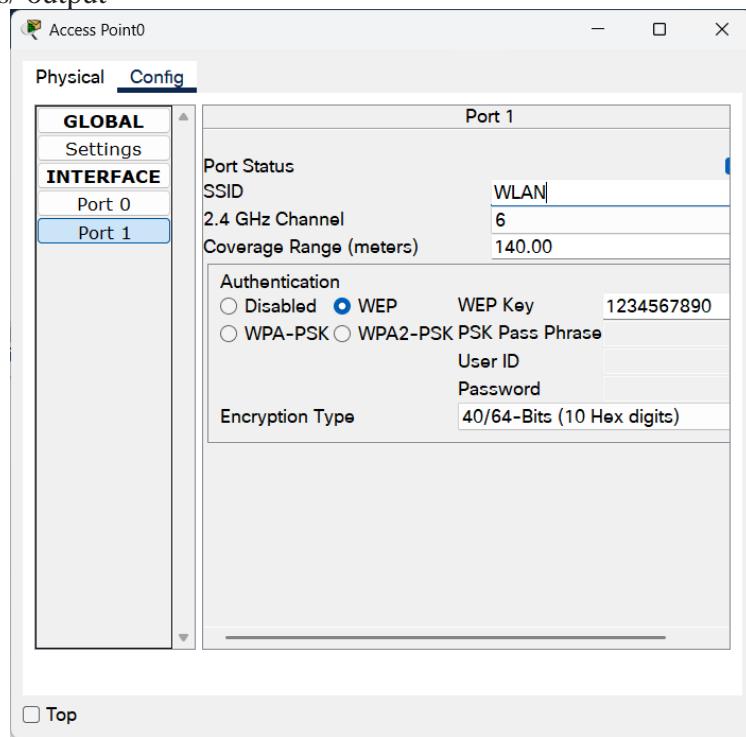
Router# Config t
Router(Config)# interface FastEthernet 0/0.1
Router(Config-Subif)# encapsulation dot1q 2
Router(Config-Subif)# ip add 192.168.2.1
255.255.255.0
no shut
card
card

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

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

Top

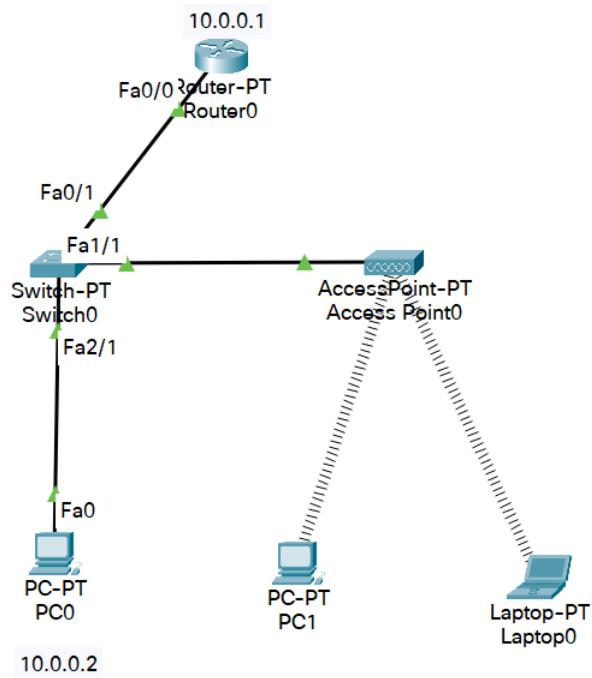
Laptop0

- 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

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

Top



Ping:

Screenshot of the **Laptop0** Command Prompt window (Packet Tracer PC Command Line 1.0) showing the output of a ping command:

```

C:\>ping 10.0.0.3

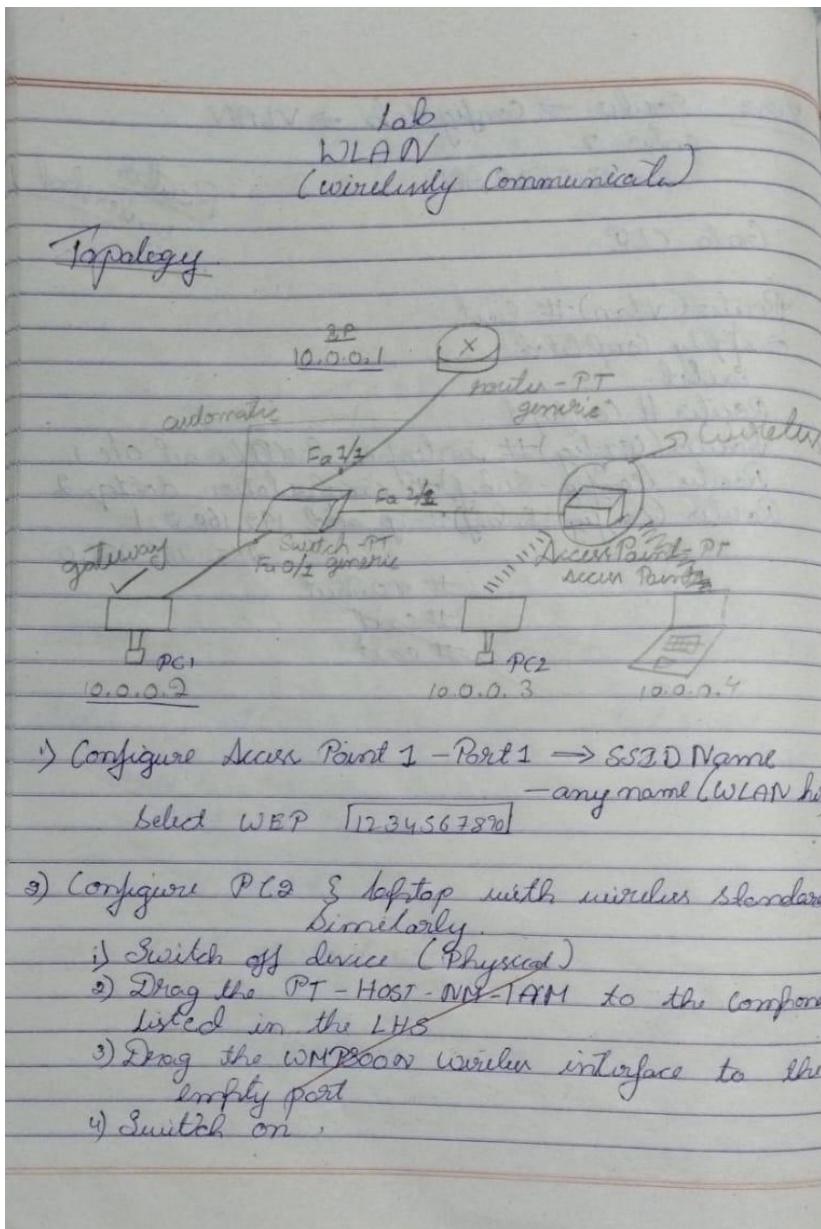
Pinging 10.0.0.3 with 32 bytes of data:
Reply from 10.0.0.3: bytes=32 time=40ms TTL=128
Reply from 10.0.0.3: bytes=32 time=25ms TTL=128
Reply from 10.0.0.3: bytes=32 time=26ms TTL=128
Reply from 10.0.0.3: bytes=32 time=24ms TTL=128

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

C:\>

```

iv. Observation



go to Config tab → wireless 0

SSID → change → WLAN

WEP

WEP Key → 1234567890

(static) IP address → 10.0.0.3 {Normally?}

At this time we get wireless connection

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., 110100111011001101101101111000011): 1111111111111111111100000
CRC valid: True

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

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

Enter the received bitstream for verification (e.g., 110100111011001101101101111000011): 1111111111111111111100001
CRC valid: False
```

iv. Observation

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

int Crc(char *ip, char *op, char *poly, int m)
{
    strcpy(op, ip);
    if (strcmp(ip, ""))
        for (int i=1; i< strlen(poly); i++)
            strcat(op, "0");
    for (int i=0; i< strlen(ip); i++)
        if (ip[i] == '1')
            for (int j=0; j< strlen(poly); j++)
                if (op[i+j] == poly[j])
                    op[i+j] = '0';
                else
                    op[i+j] = '1';
    for (int i=0; i< strlen(op); i++)
        if (op[i] == '1')
            return 1;
    return 0;
}
```

```
int main()
{
    char ip[50], op[50], mac[50];
    char poly[] = "10111";
    printf("Enter input message in binary\n");
    scanf("%s", ip);
    Crc(ip, op, poly, 5);
    printf("The transmitted message is: %s\n", op);
    printf("Enter received message\n");
    scanf("%s", mac);
    if (Crc(mac, op, poly, 5))
        printf("No error\n");
    else
        printf("Error occurred\n");
    return 0;
}

O/P:
Enter the input message: 101001111
The transmitted message is 1010011110101
Enter received message 1010010110101
No errors in data.
```

Program 2

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

ii. Procedure

```
def main():
    # Initial packets in the bucket
    storage = 0

    # Total number of times bucket content is checked
    no_of_queries = 4

    # Total number of packets that can be accommodated in the bucket
    bucket_size = 10

    # Number of packets that enter the bucket at a time
    input_pkt_size = 4

    # Number of packets that exit the bucket at a time
    output_pkt_size = 1

    for _ in range(no_of_queries):
        # Space left in the bucket
        size_left = bucket_size - storage

        if input_pkt_size <= size_left:
            # Update storage
            storage += input_pkt_size
        else:
            print(f"Packet loss = {input_pkt_size}")

        print(f"Buffer size = {storage} out of bucket size = {bucket_size}")

        # Remove packets from storage
        storage -= output_pkt_size

if __name__ == "__main__":
    main()
```

iii. Screen shots/ output

```
In [3]: runcell(0, 'E:/Engineering/5Sem/CN/Experiments/untitled3.py')
Buffer size = 4 out of bucket size = 10
Buffer size = 7 out of bucket size = 10
Buffer size = 10 out of bucket size = 10
Packet loss = 4
Buffer size = 9 out of bucket size = 10
```

iv. Observation

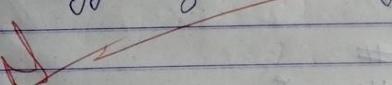
~~#!/usr/bin/python~~ Leakage Bucket Algorithm

Storage = 0
 No. of queries = 4
 bucket_size = 10
 input_pkt_size = 4
 output_pkt_size = 1
 for i in range (0, no_of_queries):
 size_left = bucket_size - storage
 if input_pkt_size <= size_left:
 storage += input_pkt_size
 else:

```
print ("Packet loss = ", input_pkt_size)
print ("Buffer size = {} Storage {} out of
       bucket_size = {} bucket_size {}")
```

storage = output_pkt_size

O/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
 Buffer loss 4
~~Buffer Size 9 out of bucket size 10~~



Program 3

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

```
clientTCP.py
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input("\nEnter file name: ")

clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print ('\nFrom Server:\n')
print(filecontents)
clientSocket.close()
```

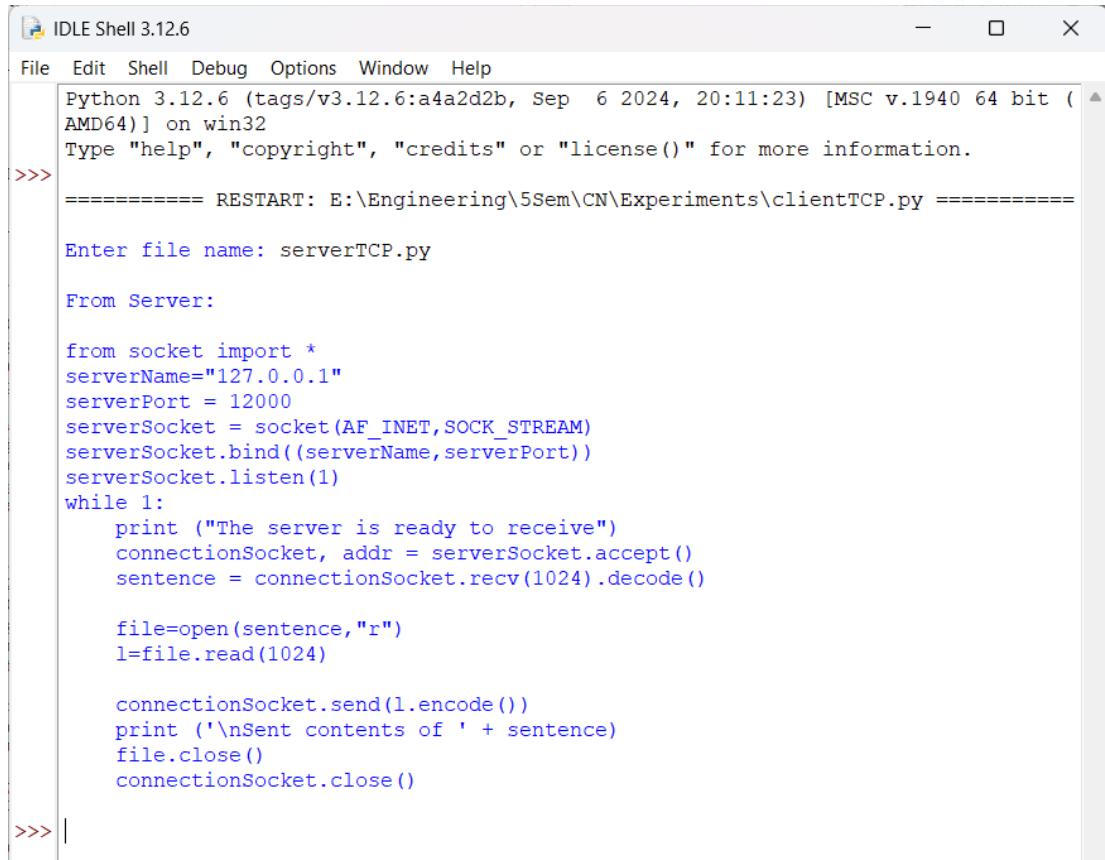
```
serverTCP.py
from socket import *
serverName="127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print ("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence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

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

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

iii. Screen shots/ output

Client



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

>>> ===== RESTART: E:\Engineering\5Sem\CN\Experiments\clientTCP.py =====

Enter file name: serverTCP.py

From Server:

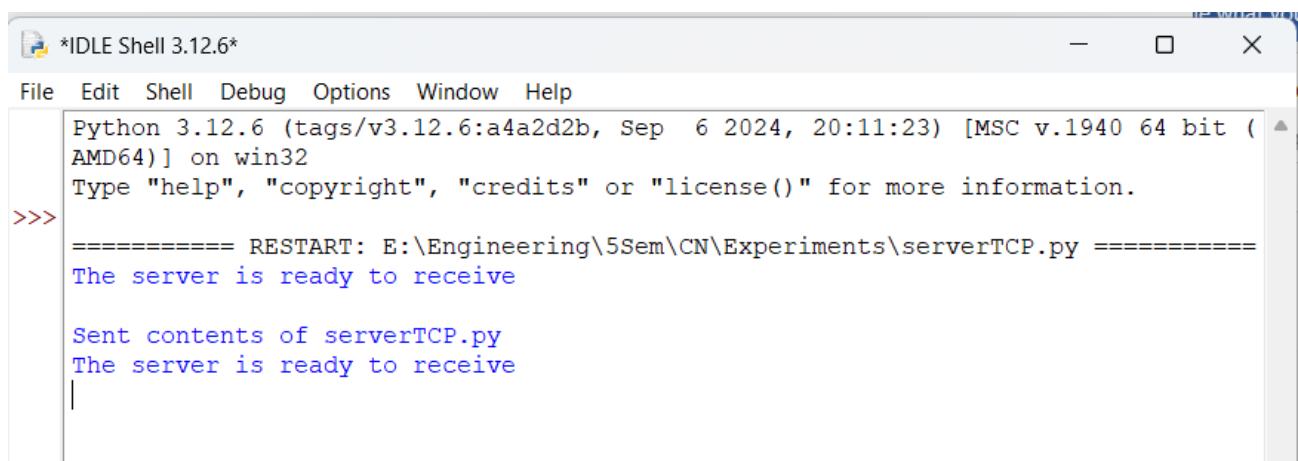
from socket import *
serverName="127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print ("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()

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

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

>>>
```

Server



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

>>> ===== RESTART: E:\Engineering\5Sem\CN\Experiments\serverTCP.py =====
The server is ready to receive

Sent contents of serverTCP.py
The server is ready to receive
```

iv. Observation

Lab
Socket programming

1) TCP :

ClientTCP.py.

```
from socket import *
ServerName = "192.0.0.1"
ServerPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((ServerName, ServerPort))
sentence = input("Enter file name: ")
```

```
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print("In From Server: " + sentence)
print(filecontents)
clientSocket.close()
```

ServerTCP.py

```
from socket import *
ServerName = '192.0.0.1'
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")
f = file.read(1024)
connectionSocket.send(f.encode())
print('In Sent content of ' + sentence)
file.close()
connectionSocket.close()
```

O/P: Our Server is ready to receive.

Sent Content of ServerTCP.py.

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)
```

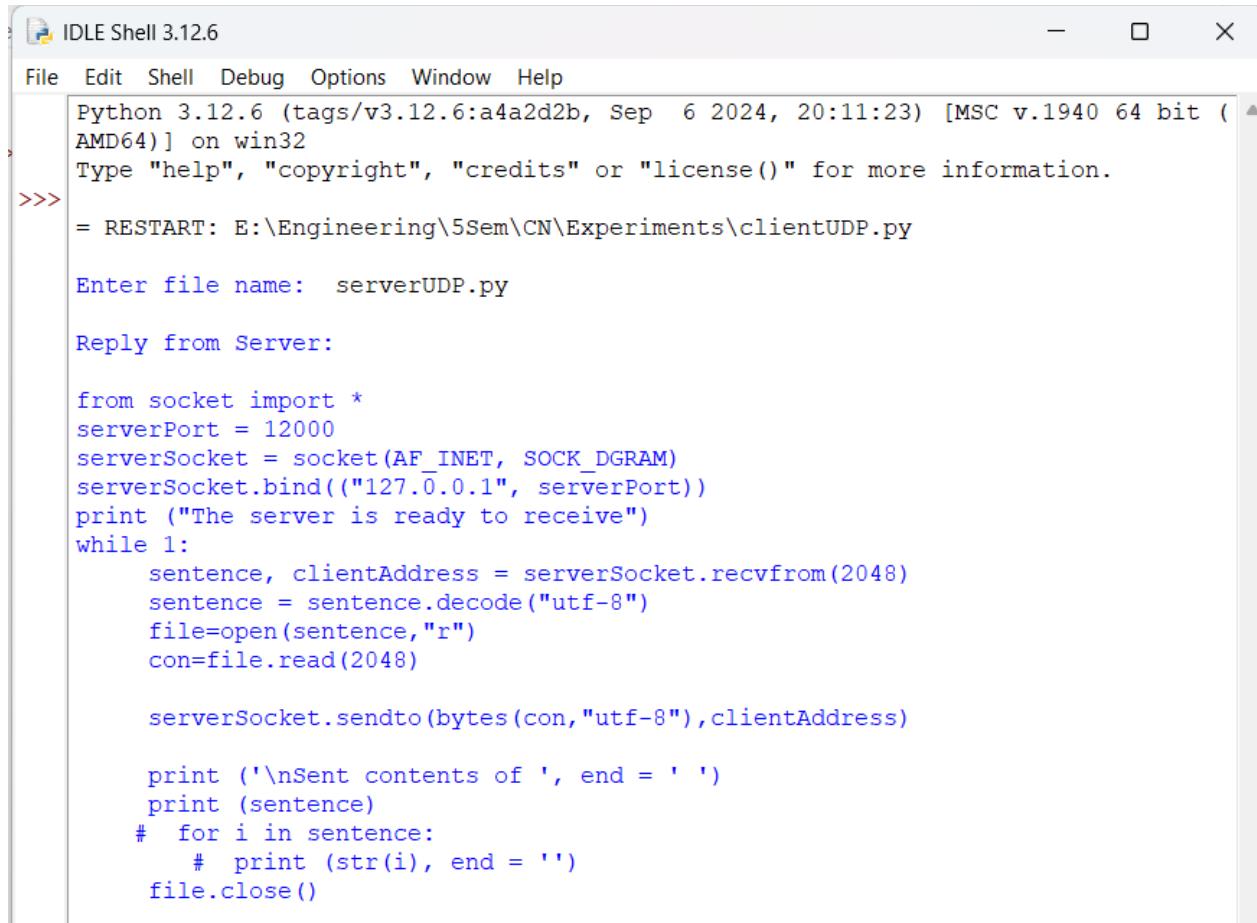
```
serverSocket.sendto(bytes(con,"utf-8"),clientAddress)
```

```
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

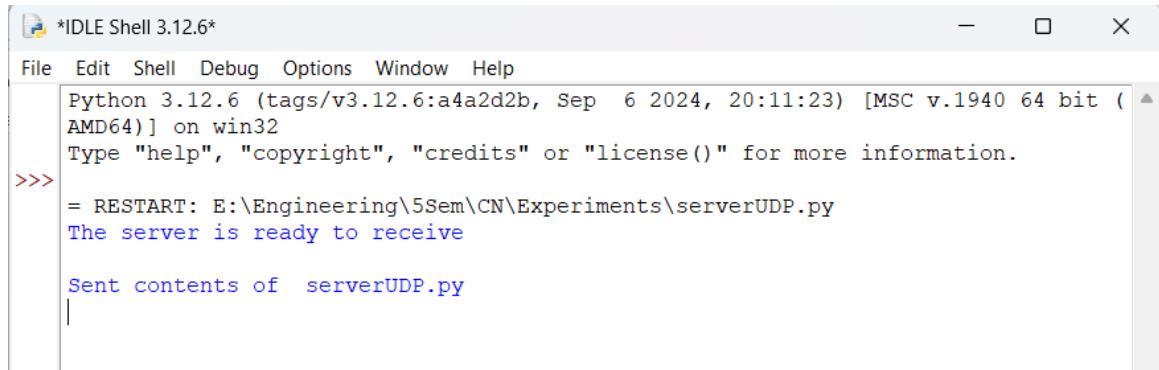
Reply from Server:

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

2) UDP

ServerUDP.py

```
from socket import *
PortServerPort = 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("file.txt", "r")
    con = file.read(2048)
    serverSocket.sendto(bytearray(con, "utf-8"), clientAddress)
    print("In Sent Content of file :")
    print(sentence)
    file.close()
```

ClientUDP.py

```
from socket import *
ServerName = "127.0.0.1"
ServerPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("Enter file Name :")
clientSocket.sendto(sentence.encode("utf-8"), (ServerName, ServerPort))
fileContent, serverAddress = clientSocket.recvfrom(2048)
print("In Reply from Server :")
print(fileContent.decode("utf-8"))
clientSocket.close()
clientSocket.close()
```

Output :

Server is ready to receive.
Sent content of Server UDP.py
The server is ready to receive