

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB RECORD

Computer Network Lab (23CS5PCCON)

Submitted by

Shashidhar B M (1BM23CS257)

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 **Shashidhar B M (1BM23CS257)**, 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.

Rekha G S 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
2	18/10/2024	Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply	6
3	25/10/2024	Configure default route, static route to the router	11
4	08/11/2024	Configure DHCP within a LAN and outside LAN.	18
5	22/11/2024	Configure RIP routing Protocol in Routers	24
6	22/11/2024	Configure OSPF routing protocol	29
7	22/11/2024	Demonstrate the TTL/ Life of a Packet	36
8	08/11/2024	Configure Web Server, DNS within a LAN.	39
9	20/12/2024	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)	43
10	20/12/2024	To understand the operation of TELNET by accessing the router in server room from a PC in IT office.	46
11	20/12/2024	To construct a VLAN and make the PC's communicate among a VLAN	49
12	20/12/2024	To construct a WLAN and make the nodes communicate wirelessly	53

Github Link:

https://github.com/ShashidharM0118/CN_LAB-1BM22CS257

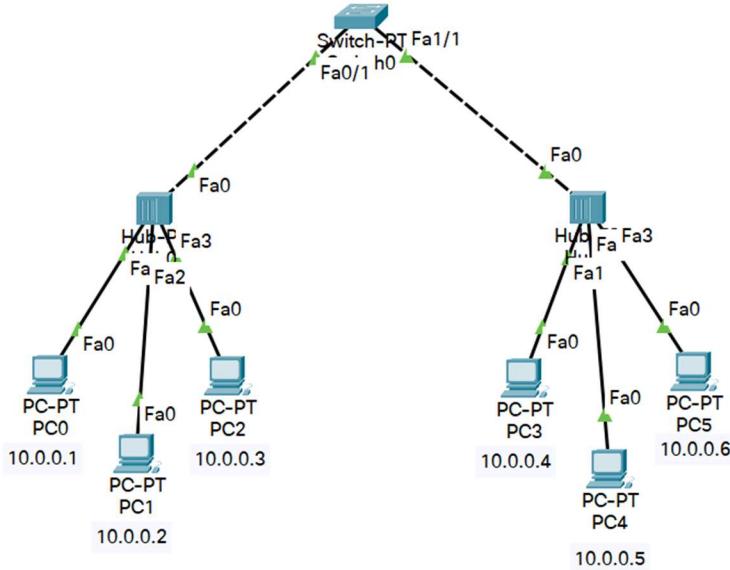
Index-Cycle-II

Sl. No.	Date	Experiment Title	Page No.
1	15/11/2024	Write a program for error detecting code using CRC-CCITT (16-bits).	57
2	15/11/2024	Write a program for congestion control using Leaky bucket algorithm	61
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.	63
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.	66

Cycle-I

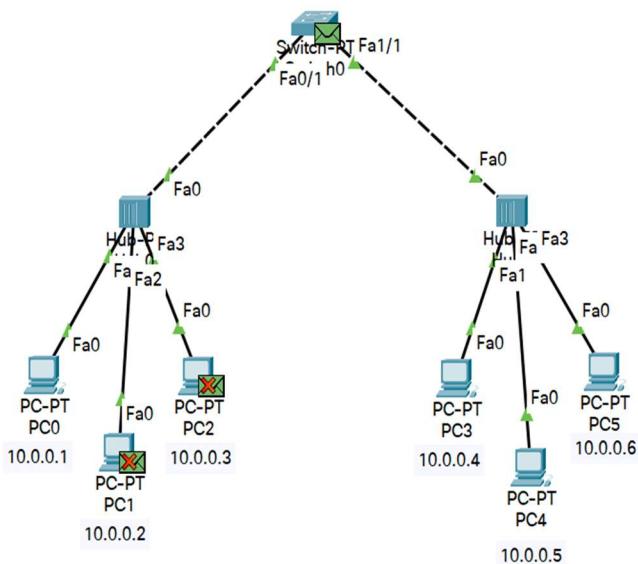
Program 1

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

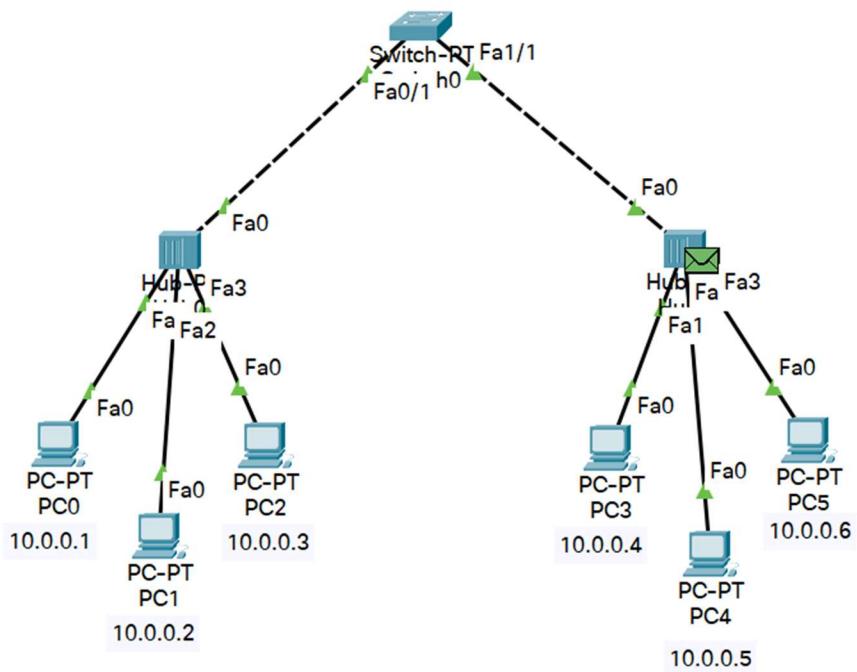


- iii. Screen shots/ output

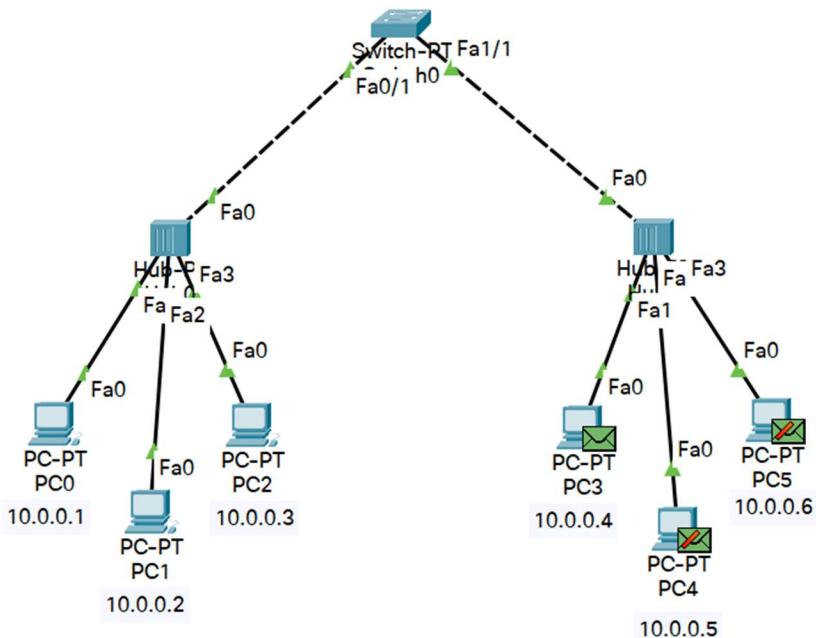
Hub behaviour at sending end



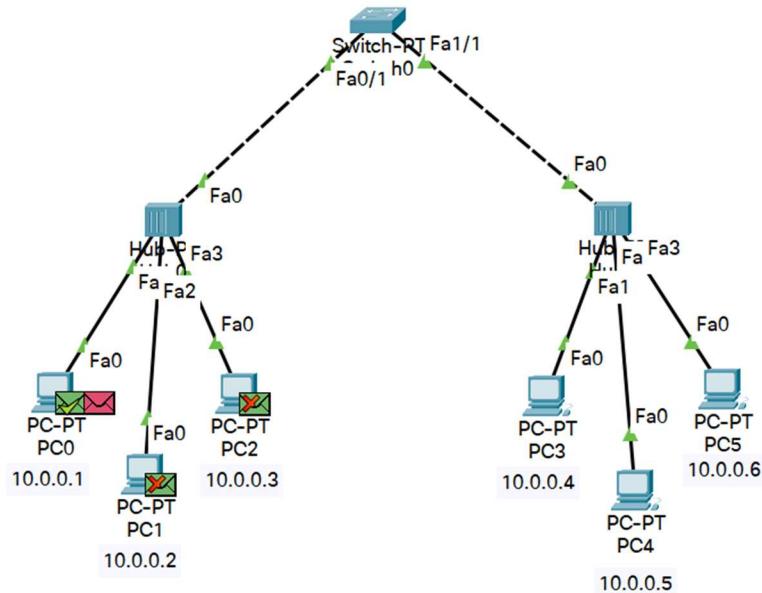
Switch behaviour



Hub behaviour at receiving end



Hub behaviour when back to sender



Ping command to connectivity

Screenshot of the Packet Tracer Command Prompt window titled "Command Prompt". The window shows the output of a ping command from host PC0 to host PC3. The output is as follows:

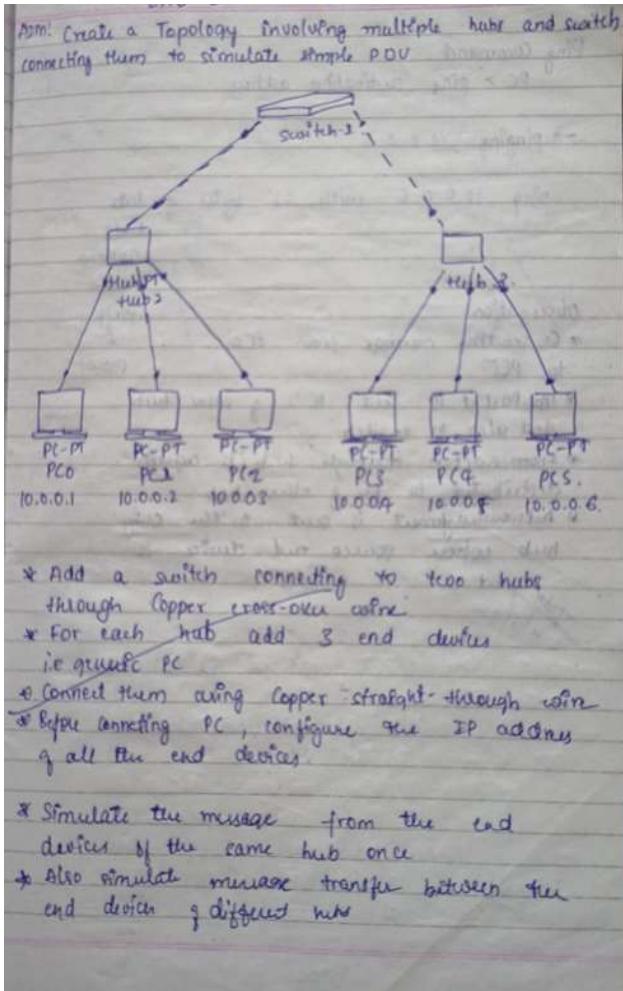
```
Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128

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



Ping Command
PC > ping destination address
 → pinging 10.0.0.6.
 ping 10.0.0.6 with 82 bytes q. date.
 Radio Options ↓
 prenum ↓
 Show ↓

Observation

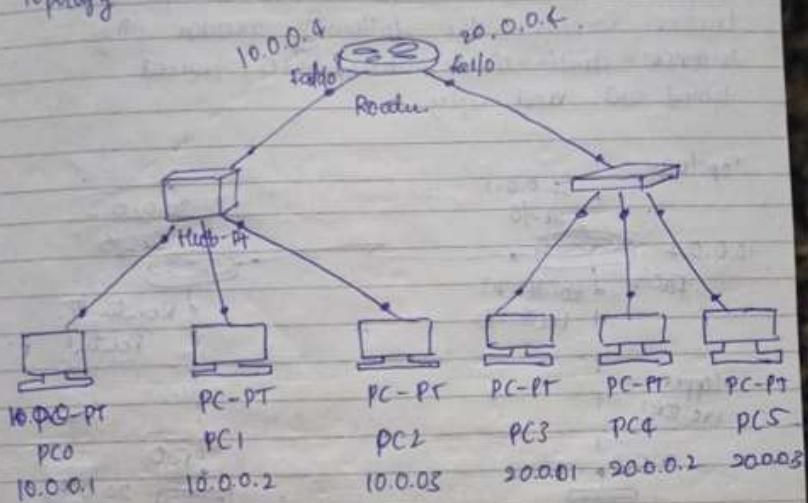
- * PC sending message from PC0 to PC5
- * The packet is sent PC's of same hub and also to switch
- * From switch it is sent to PC of other hub
- * Acknowledgment is sent to the only hub where source and device is present

~~4/10/20~~

Date : 04/10/24.

A.

AIM: Connecting and Configuring Router
Topology



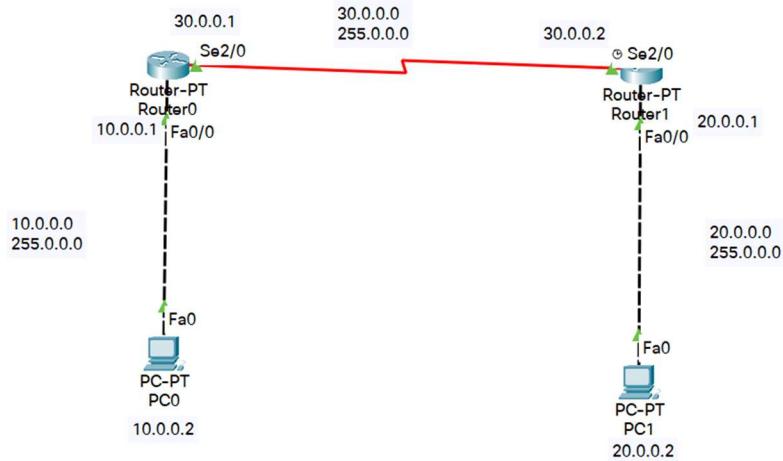
Steps:

- * Add a Hub and switch connecting to a Router.
- * For Hub and switch add 8 pc for each and config IP for them as shown.
- * For switch
 - click on switch → cl
 - type "no", double 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 shutdown
 - exit.

Q2. Explain the configuration of Cisco switch.

Program 2

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



- iii. Screen shots/ output

Router0 configuration

The screenshot shows the Router0 CLI interface. The window title is 'Router0'. The tabs at the top are 'Physical', 'Config', and 'CLI', with 'CLI' being the active tab. The main area is titled 'IOS Command Line Interface' and contains the following configuration session:

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

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

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

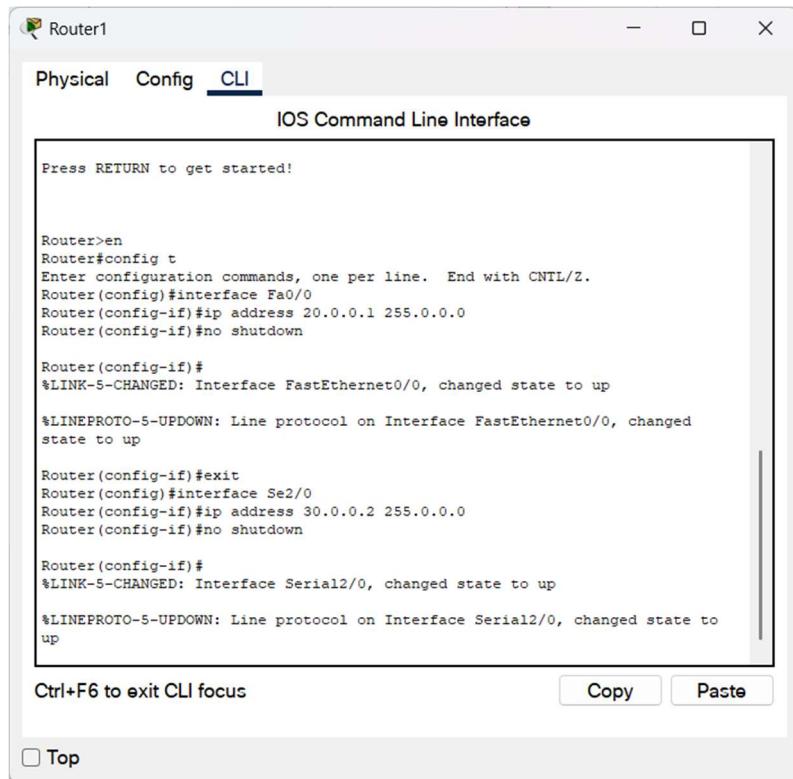
Router(config-if)#exit
Router(config)#interface Se2/0
Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#no shutdown

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

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

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

Ip route command in Router0

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

```

Ip route command in Router1

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

```

Destination host Unreachable (Before establishing network Fully)

PC1

Physical Config Desktop Programming

Command Prompt X

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

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 20.0.0.1: Destination host unreachable.

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

Request Timed Out

PC2

Physical Config Desktop Programming

Command Prompt X

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

Pinging 20.0.0.2 with 32 bytes of data:

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

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

Reply from Destination

PC0

Physical Config Desktop Programming

Command Prompt

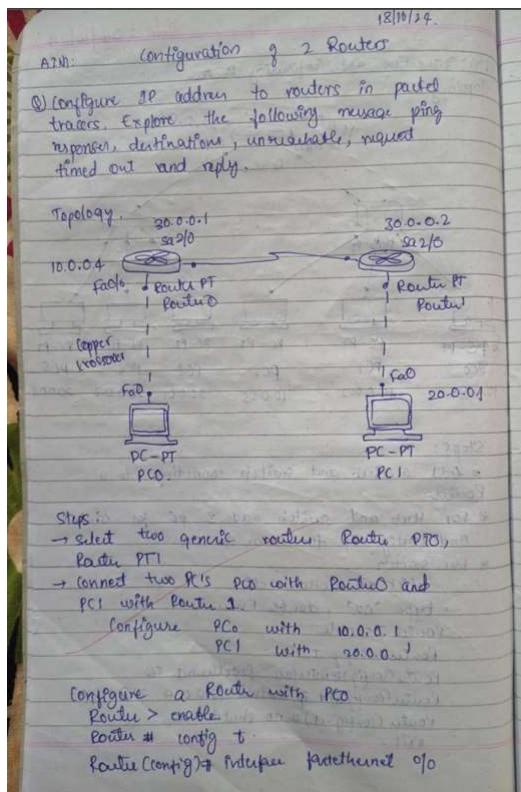
```
Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.2

Pinging 20.0.0.2 with 32 bytes of data:

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

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

iv. Observation



Router(Config-if)# ip address 10.0.0.4 255.0.0.0
Router(Config-if)# no shutdown
Router(Config-if)# exit
config PC1 with Router1
Router > enable
Router# config t
Router(Config)# interface fastethernet 1/0
Router(Config-if)# ip address 20.0.0.4 255.0.0.0
Router(Config-if)# no shutdown
Router(Config-if)# exit
configuration of two Routers
For Router1:
Router# show ip route
Gateway of last resort is not set
c 10.0.0.0/8 is directly connected, FastEthernet 0/0
c 30.0.0.0/8 is directly connected, FastEthernet 1/0
For Router1:
Router# show ip route
Gateway of last resort is not set
c 20.0.0.0/8 is directly connected, FastEthernet 1/0
c 30.0.0.0/8 is directly connected, FastEthernet 0/0
Router#
Router(Config)# ip route 20.0.0.0 255.0.0.0
30.0.0.2.

Router 1:
Router (config)# ip route 10.0.0.0 255.0.0.0
30.0.0.0

Packet Transmission Successful.

command prompt.
paired tracer PC command line 20

IC > ping 20.0.0.1
To check route configured or not
pinging 20.0.0.1 with 3c bytes

Reply from 20.0.0.1 by tel = 32
time = 8ms TTL = 12.6

ping statistics for 20.0.0.1:

packets: sent + Received =
lost = 0 (0% losses).

~~pc> ping 20.0.0.1~~
~~pinging 20.0.0.1 with 32 bytes of data~~

~~Request time off.~~

~~when gate way is not configured~~

~~It is 10 pm and it is not config and.~~

Dentists lost valuable
legally based art
Reply

(cohen gateway Not configured)

> ping 20.0.0.1
pinging 20.0.0.0 with 32 bytes of data
Request timed out.

> ping 50.0.0.1
pinging 50.0.0.1 with 32 bytes of data.

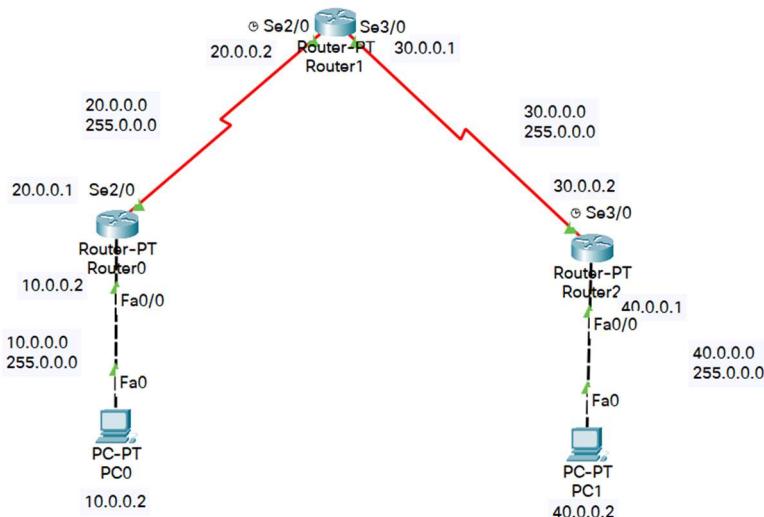
~~25/10/2014~~ Reply from 10.0.0.2: Destination host unreachable

Observation:

Configured routers and executed ping commands and observed output.

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

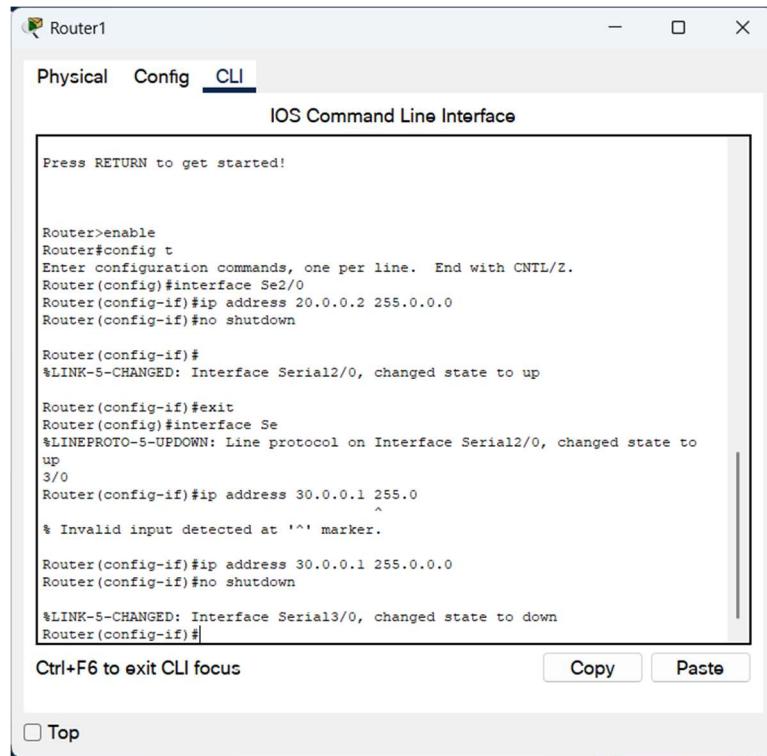
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed
state to up
%IP-4-DUPLICATE: Duplicate address 10.0.0.2 on FastEthernet0/0, sourced by
000C.CFC2.65B0

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

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

Router1 configuration



The screenshot shows a Windows-style application window titled "Router1". The tab bar at the top has three tabs: "Physical", "Config", and "CLI", with "CLI" being the active tab. The main area is labeled "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
^
% Invalid input detected at '^' marker.

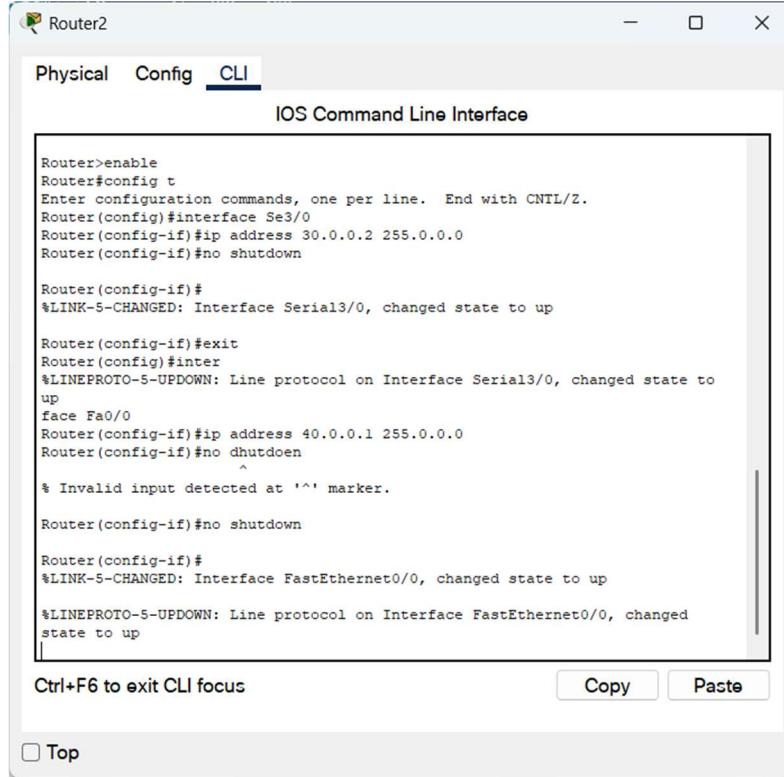
Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#no shutdown

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

Ctrl+F6 to exit CLI focus
```

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

Router2 configuration



The screenshot shows a Windows-style application window titled "Router2". The tab bar at the top has three tabs: "Physical", "Config", and "CLI", with "CLI" being the active tab. The main area is labeled "IOS Command Line Interface". It displays the following configuration commands:

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

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

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

Router(config-if)#no shutdown

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

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

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

Static Routing:

Router0

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)#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

Router1

Physical Config CLI

IOS Command Line Interface

```
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

Router2

Physical Config CLI

IOS Command Line Interface

```

Router>enable
Router>config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#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 **Copy** **Paste**

Top

Dynamic Routing:

Route0

Router0

Physical Config CLI

IOS Command Line Interface

```

S 30.0.0.0/8 [1/0] via 20.0.0.2
S 40.0.0.0/8 [1/0] via 20.0.0.2

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

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

Gateway of last resort is 20.0.0.2 to network 0.0.0.0

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

Router#

```

Ctrl+F6 to exit CLI focus **Copy** **Paste**

Top

Router2

The screenshot shows the Router2 CLI interface. The title bar says "Router2". Below it, there are tabs: "Physical", "Config", and "CLI", with "CLI" being the active tab. The main window is titled "IOS Command Line Interface". It displays the following command-line session:

```

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#
*SIS-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 CLI window, there are "Copy" and "Paste" buttons, and a checkbox labeled "Top".

Pinging:

```

C:\>ping 40.0.0.2

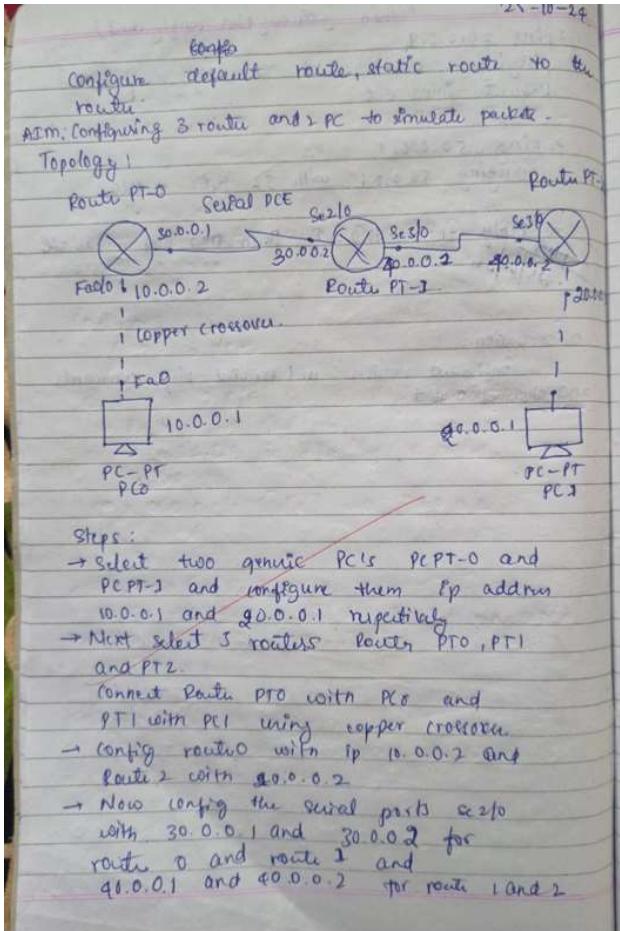
Pinging 40.0.0.2 with 32 bytes of data:

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



Packet transmission:
Select source and destination PC for the packet to be transferred.
Packet transmission successful.

ping Command:

```

pc > ping 40.0.0.1
pinging 40.0.0.1 with 32 bytes of data.
Reply from 40.0.0.1: bytes=32 time=15ms
" "
" "

```

ping statistics for 40.0.0.1
packets: sent = 4, received = 4, lost = 0.
Approximate round trip times in milliseconds
Minimum=2ms, Maximum=15ms, Average=8ms

Router 1:

```

Router(config)# ip route 10.0.0.0 255.0.0.0 20001
Router(config)# ip route 40.0.0.0 255.0.0.0 30002

```

Router 2:

```

ip route 20.0.0.0 255.0.0.0 20.0.0.2
ip route 10.0.0.0 255.0.0.0 20.0.0.1

```

ping Commands:

```

pc > ping 50.0.0.1
Reply from 10.0.0.2: Destination host unreachable.

```

Default Routing:

for Router 0:

Router# ip route 0.0.0.0 0.0.0.0 30.0.0.2

for Router 2:

ip route 0.0.0.0 0.0.0.0 40.0.0.1

Router 0:

show ip route

c 10.0.0.0/8 is directly connected,

FastEthernet 0/0

c 30.0.0.0/8 is directly connected,

Serial 2/0

s 0.0.0.0/0 [1/0] via 30.0.0.2

Observation (Static Routing)

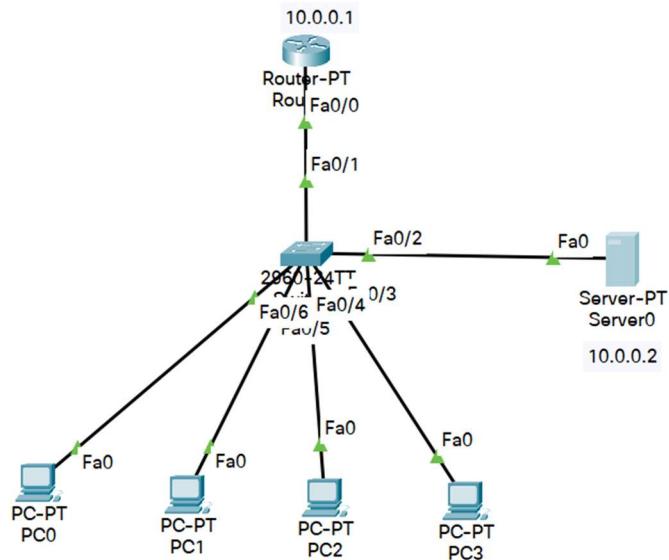
Executed ping command to test simulation,
learn how to configure 3 routers

Observation (Default Routing)

observed that default routing can be done
in case where more routers involved.

Program 4

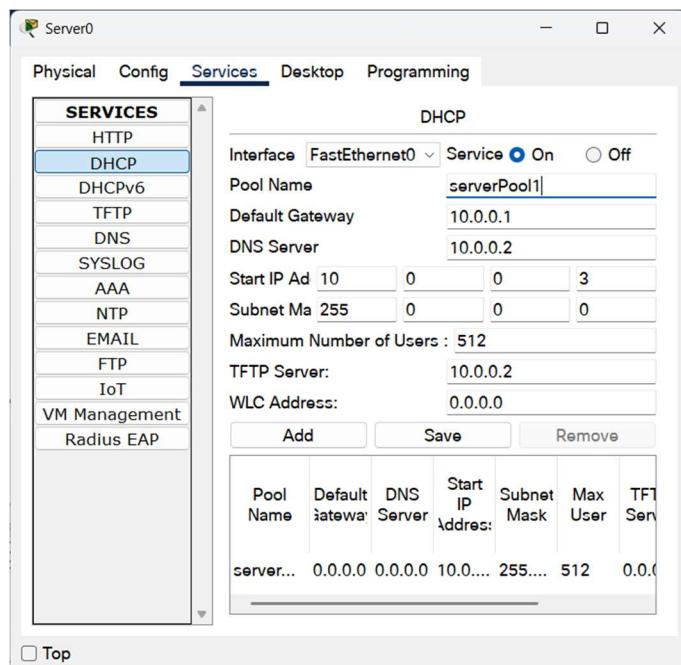
- i. Configure DHCP within a LAN and outside LAN.
- ii. Procedure along with the topology



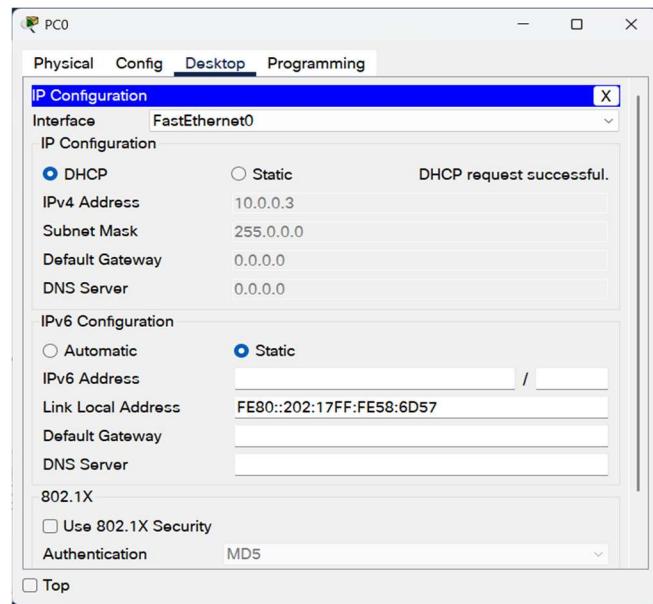
- iii. Screen shots/ output

DHCP Within LAN

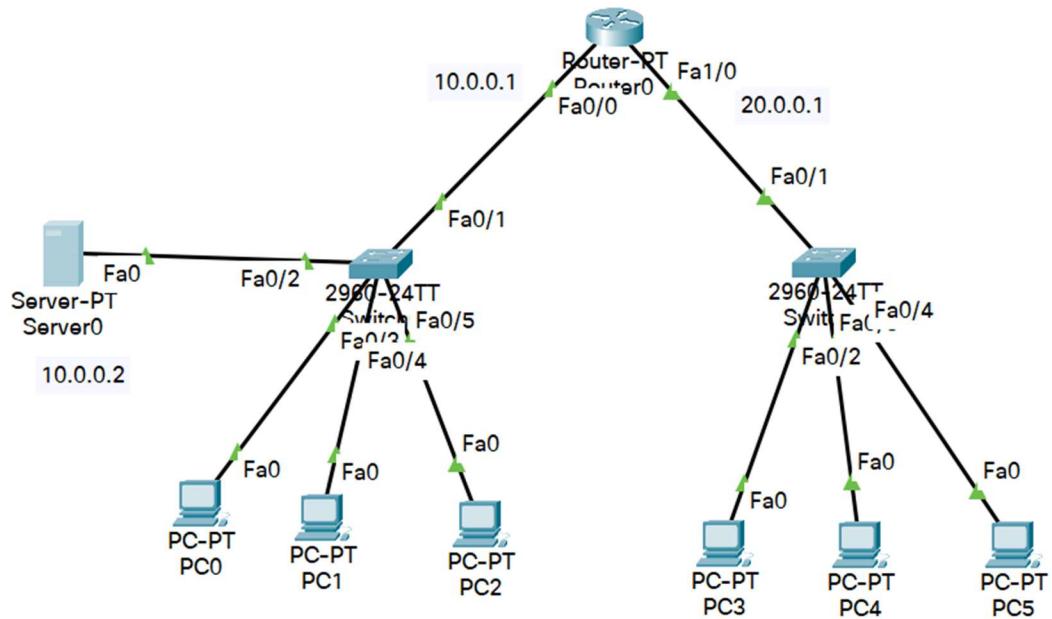
DHCP Configuration



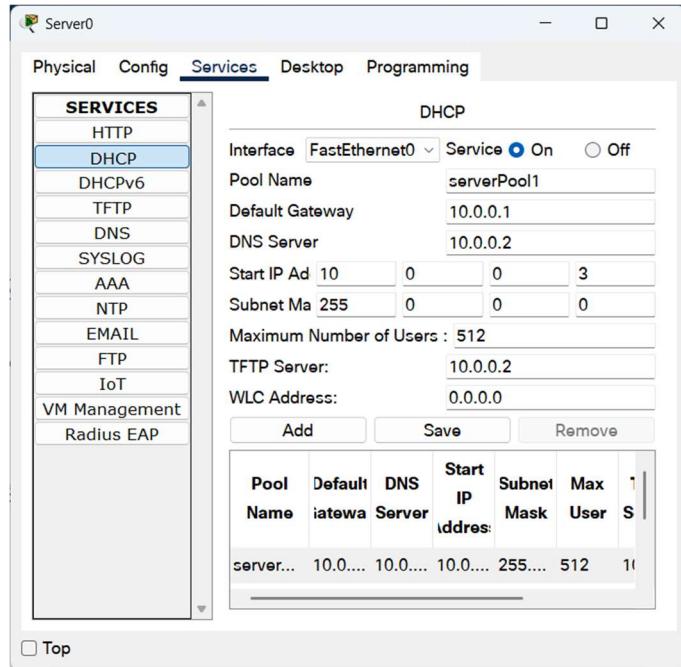
PC settings



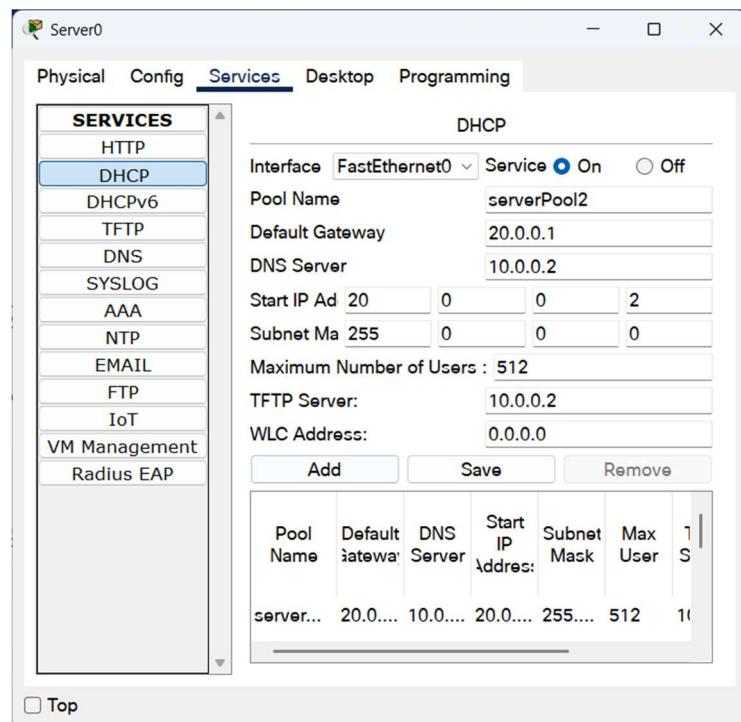
DHCP outside LAN:



DHCP configuration for inside LAN



DHCP configuration for outside LAN

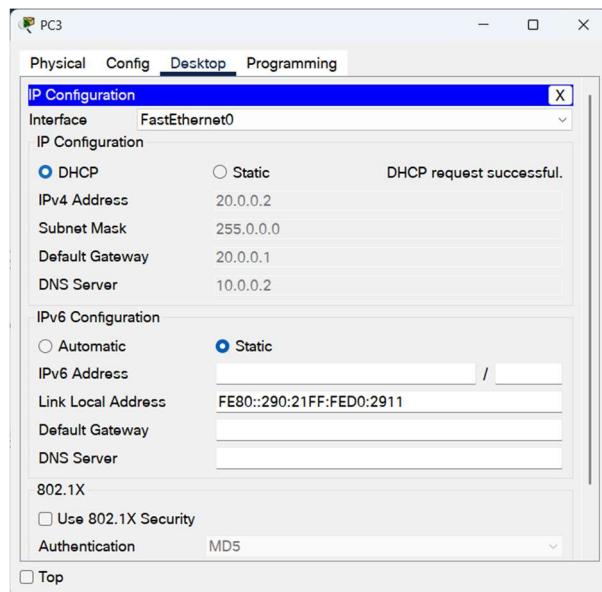


Ip helper command in Router

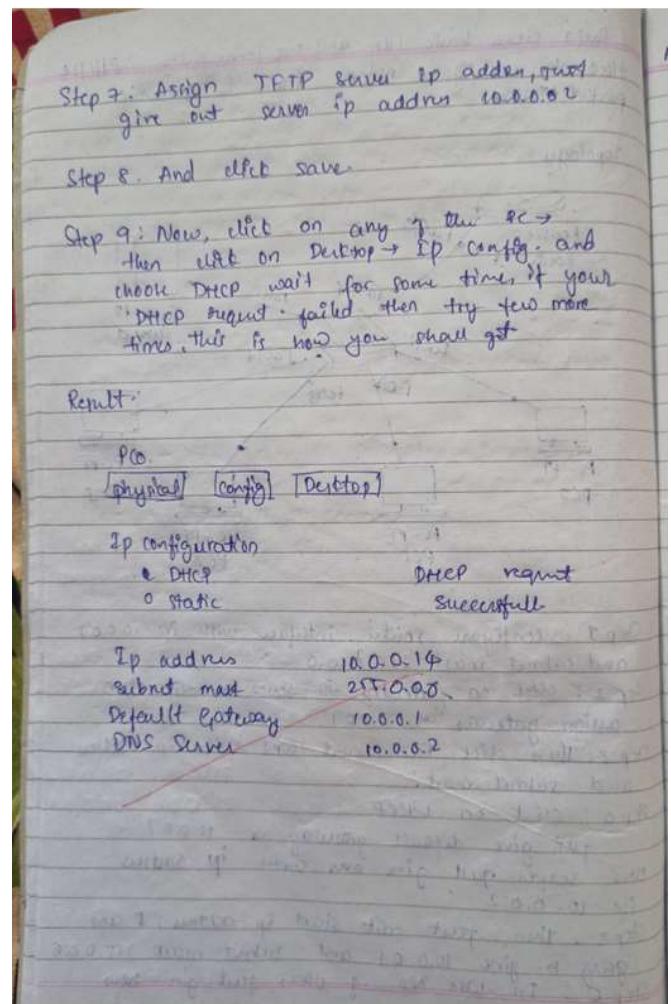
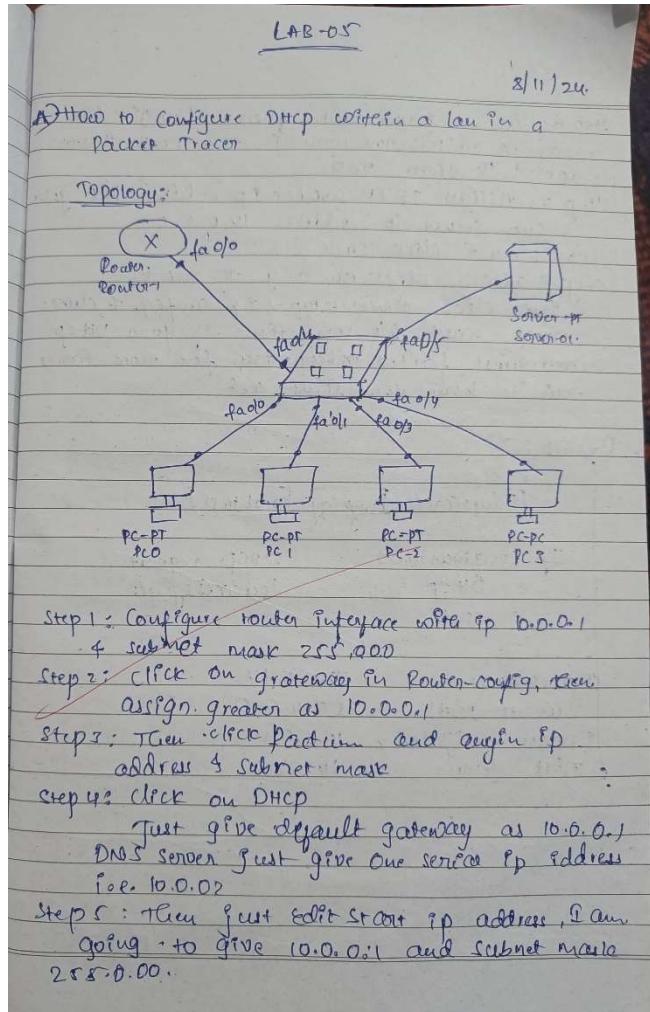
```
Router(config-if)#exit
Router(config)#interface Fa1/0
Router(config-if)#ip helper-address 10.0.0.2
Router(config-if)#

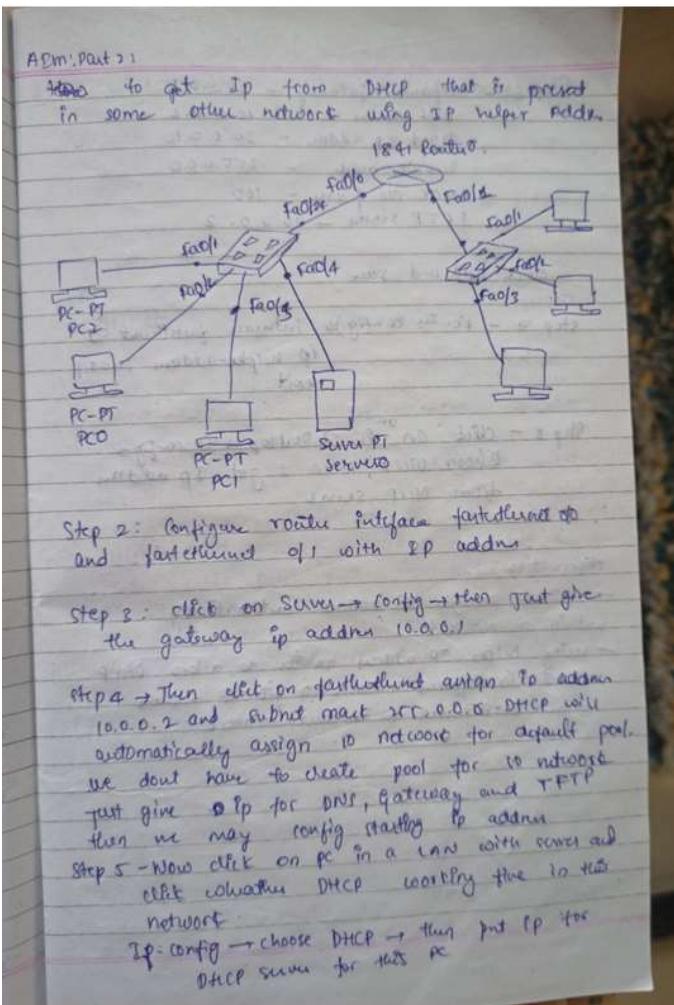
```

PC setting in another network



iv. Observation





Step 6 - Default Gateway - 20.0.0.1
 DNS server - 10.0.0.2
 Start IP address - 20.0.0.10
 subnet mask - 255.0.0.0
 max. No. of user - 100
 TFTP server - 10.0.0.2

click add and save

Step 7 - Router config → interface fastethernet 0/0
 ip helper-address 10.0.0.1
 exit

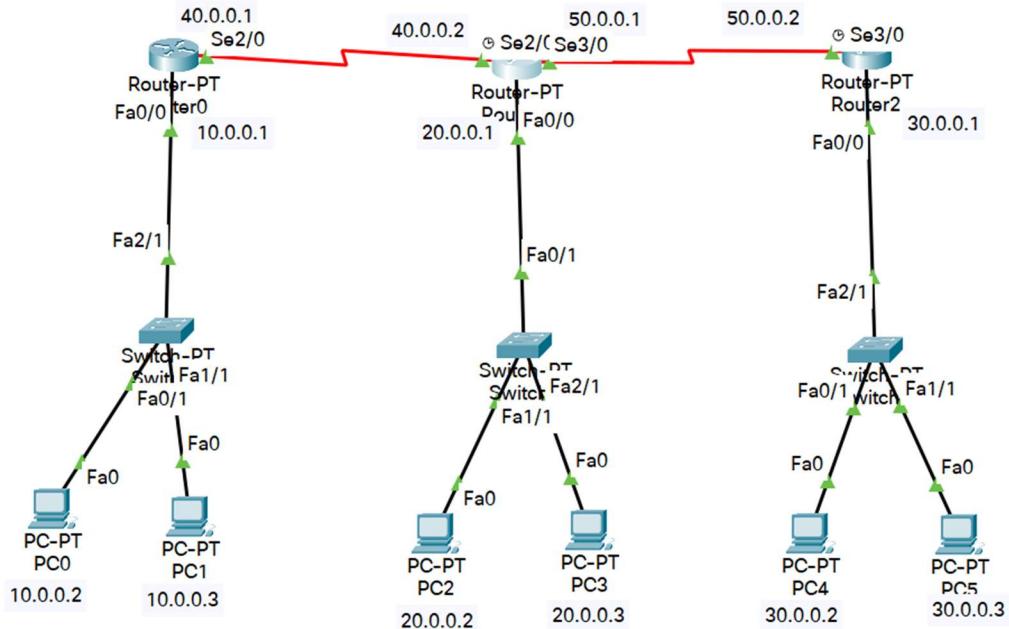
Step 8 - Click on ^{an} PC → Desktop → IP config
 Choose DHCP; Now I get IP address from DHCP server

Observation:

- DHCP assigns IP address automatically within a network
- Using helper IP address router to allow DHCP client in different network to get IP address from DHCP server to another network

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

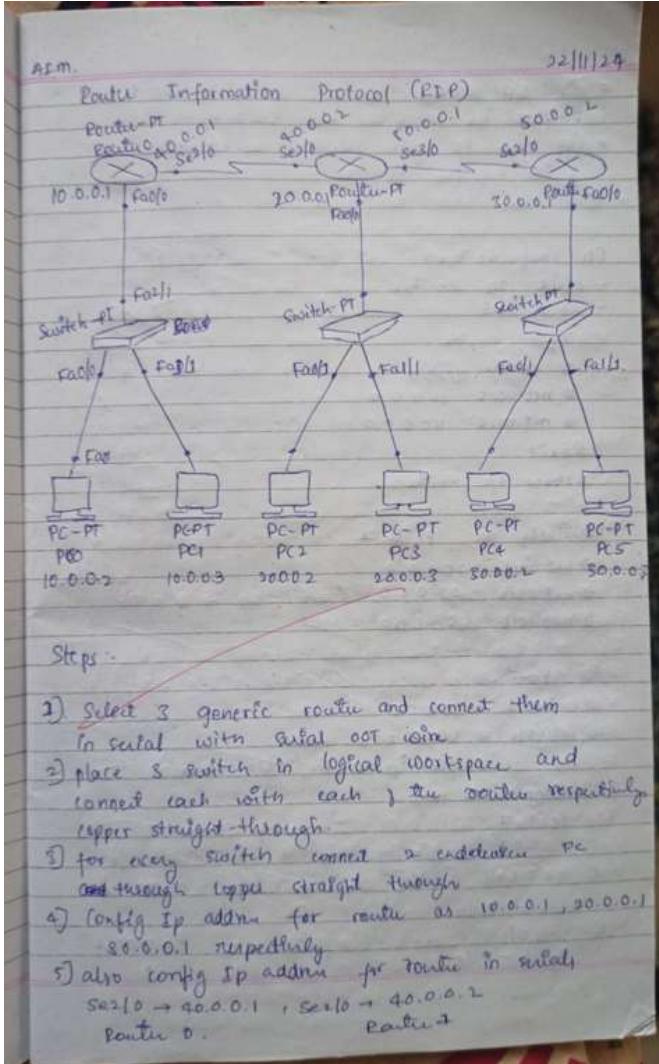
Pinging 20.0.0.2 with 32 bytes of data:

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



* Find IP address between Router 1 and Router 2 as 50.0.0.1 and 50.0.0.2.

* for each PC give gateway as IP address of its respective Router.

For each Router:

- click on Router
- go to CLI
- Type enable
- # route rip
- # network 10.0.0.0
- # network 20.0.0.0
- # network 30.0.0.0
- # exit
- > show ip route

Similarly for Router 1

```
# network 20.0.0.0
# network 40.0.0.0
# network 50.0.0.0
```

for Router 2

```
# network 30.0.0.0
# network 50.0.0.0
```

show ip route for each.

Observation:

For Router 2

```
C 10.0.0.0/8 is directly connected FastEthernet 0/0
R 20.0.0.0/8 [120/1] via 40.0.0.2 ; 00:00:06
Serial 1/0
R 30.0.0.0/8 [256/2] via 40.0.0.2 00:00:06
Serial 1/0
```

C 40.0.0.0/8 is directly connected, via Serial 1/0
R 50.0.0.0/8 [12010] via 40.0.0.2 00:00:06!
Serial 2/0

Similarly show IP route for R1 at Port 3 and Port 2.

elicit on PC1 → command prompt

PC> ping 30.0.0.2

pinging 30.0.0.2 with 32 bytes of data:

Reply from 30.0.0.2: bytes=32 time=4ms TTL=125

Reply from 30.0.0.2: bytes=32 time=4ms TTL=125

Reply " " " "

Reply " " " "

ping statistics for 30.0.0.2

Approximate round trip times:

Packets: sent=4 received=4, lost=0

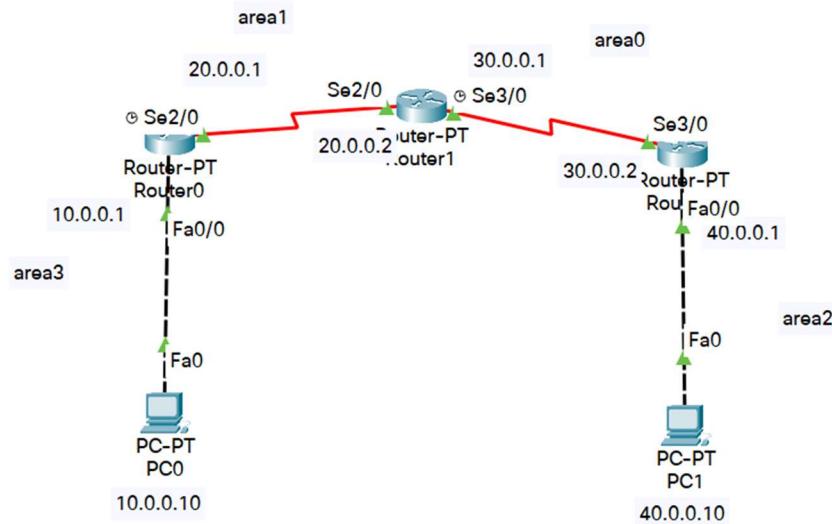
Approximate round trip time in milliseconds

Minimum=4ms, maximum=4ms, average=4ms

2/1/13

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)#

Router2
```

Create Virtual Link

Router0

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

Router1
```

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

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

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

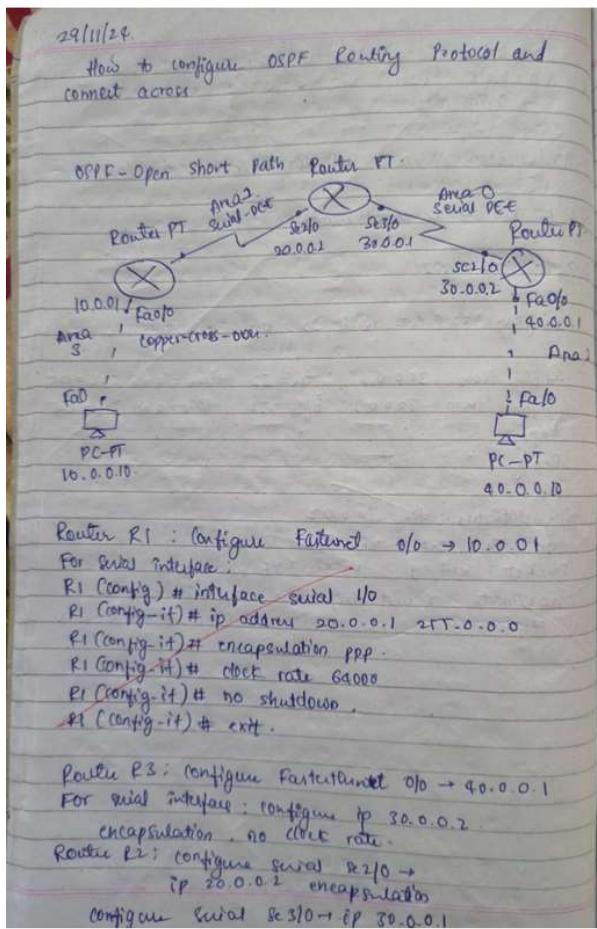
Router(config-router)#area 1 v
01:11:41: %OSPF-4-ERRRCV: Received invalid packet: mismatch area ID, from
backbone area must be virtual-link but not found from 20.0.0.2, Serial2/0
irtual-link 1.1.1.1
Router(config-router)#
01:11:56: %OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on OSPF_VL0 from LOADING to
FULL,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

Pinging

```
C:\>ping 40.0.0.10
Pinging 40.0.0.10 with 32 bytes of data:
Reply from 40.0.0.10: bytes=32 time=24ms TTL=125
Reply from 40.0.0.10: bytes=32 time=18ms TTL=125
Reply from 40.0.0.10: bytes=32 time=18ms TTL=125
Reply from 40.0.0.10: bytes=32 time=20ms TTL=125

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

iv. Observation



Configuring OSPF routing protocol

Router R1:

- R1 (config)# router ospf 1
- R1 (config-router) # area 0
- R1 (config-router) # router-id 1.1.1.1
- R1 (config-router) # network 10.0.0.0 0.255.255.255 area 0
- R1 (config-router) # exit

Router R2:

- R2 (config) # router ospf 2
- R2 (config) # route-id 2.2.2.2
- R2 (config-router) # network 20.0.0.0 0.255.255.255 area 0

Router R3:

- R3 (config) # router ospf 3
- R3 (config-router) # route-id 3.3.3.3
- R3 (config-router) # network 30.0.0.0 0.255.255.255 area 2

Configure loopback address

R1:

- R1 (config) # interface loopback 0
- R1 (config-if) # ip add 172.16.1.252 255.255.0.0
- R1 (config-if) # no shutdown

R2:

- R2 (config) # interface loopback 0
- R2 (config-if) # ip add 172.16.1.157 255.255.0.0
- R2 (config-if) # no shutdown

R1:

Router# show ip route

- c 10.0.0.0/8 is directly connected, FastEthernet0/0
- c 20.0.0.0/8 is variably subnetted, 2 subnets, 2
 c 20.0.0.0/8 is directly connected, serial 2/0
- c 20.0.0.0/8 is directly connected, serial 2/0
- c 20.0.0.0/8 [110/128] via 20.0.0.2, 00:00:11
 serial 2/0
- c 172.16.0.0/16 is directly connected, loopback 0.

R2:

```

Router (config) # route-map OSPF 1
Router (config-route) # area 0 virtual-link 1.1.1.1

```

~~R2~~

```

Router (config) # route-map OSPF 2
Router (config-route) # area 1 virtual-link 1.1.1.1

```

pc > ping 20.0.0.10

pinging 20.0.0.10 with 32 bytes of data.
Request timed out.

Reply from 20.0.0.10: bytes=32 time=16ms TTL=128

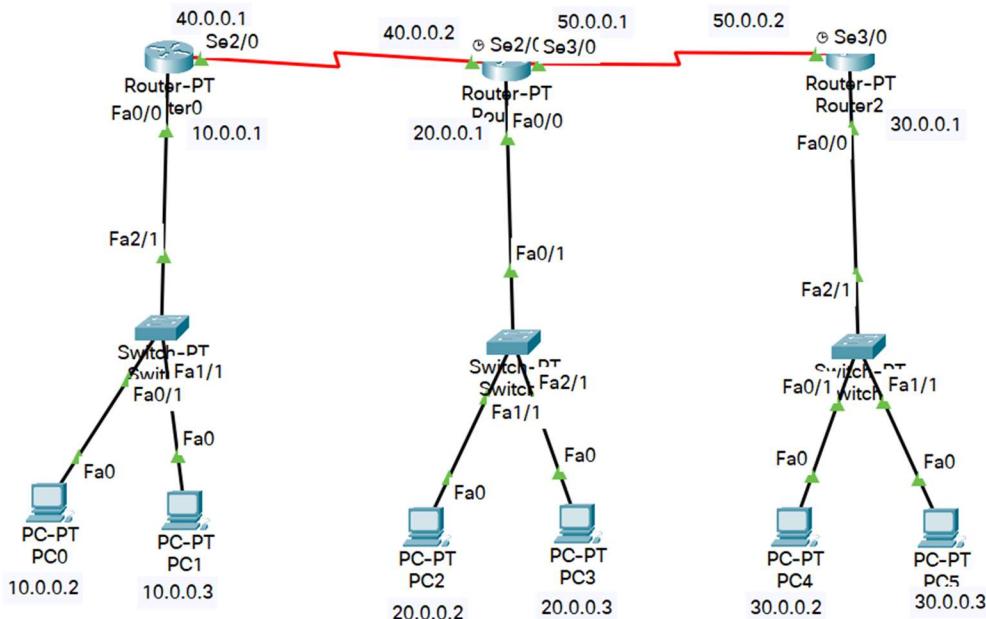
ping statistics for 20.0.0.10:
packets: sent=4, received=3, lost=1
(25% loss)

#approximate round trip time 20 milliseconds
minimum = 2ms, maximum = 60ms
average = 8ms

~~20/12/20~~

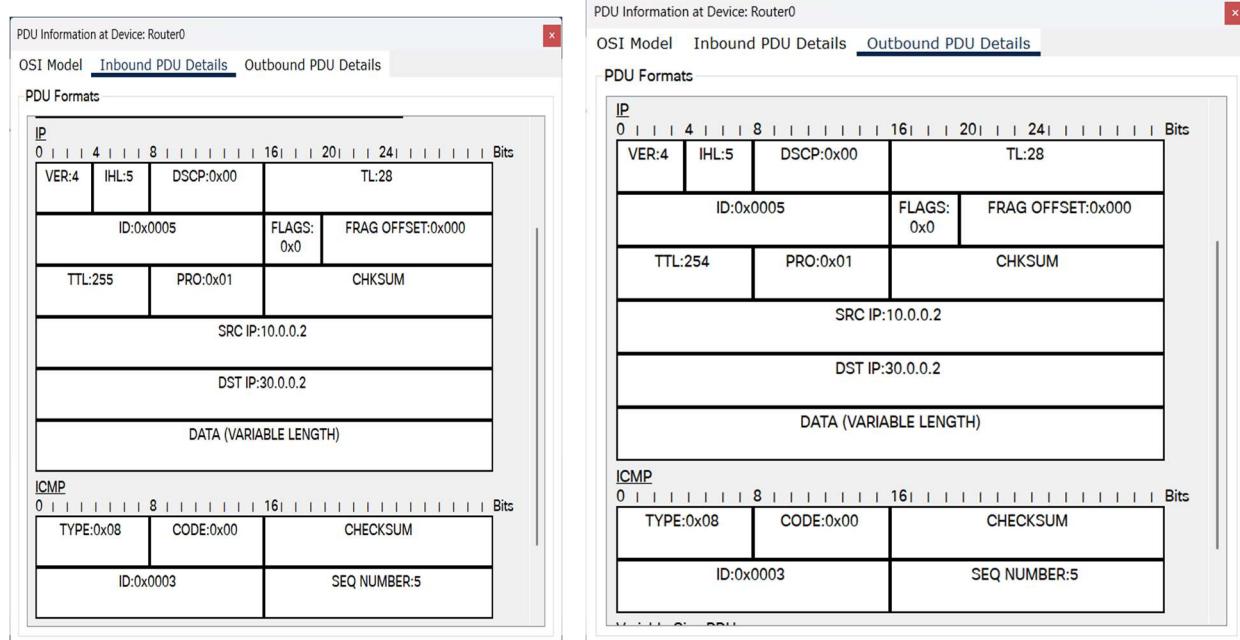
Program 7

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

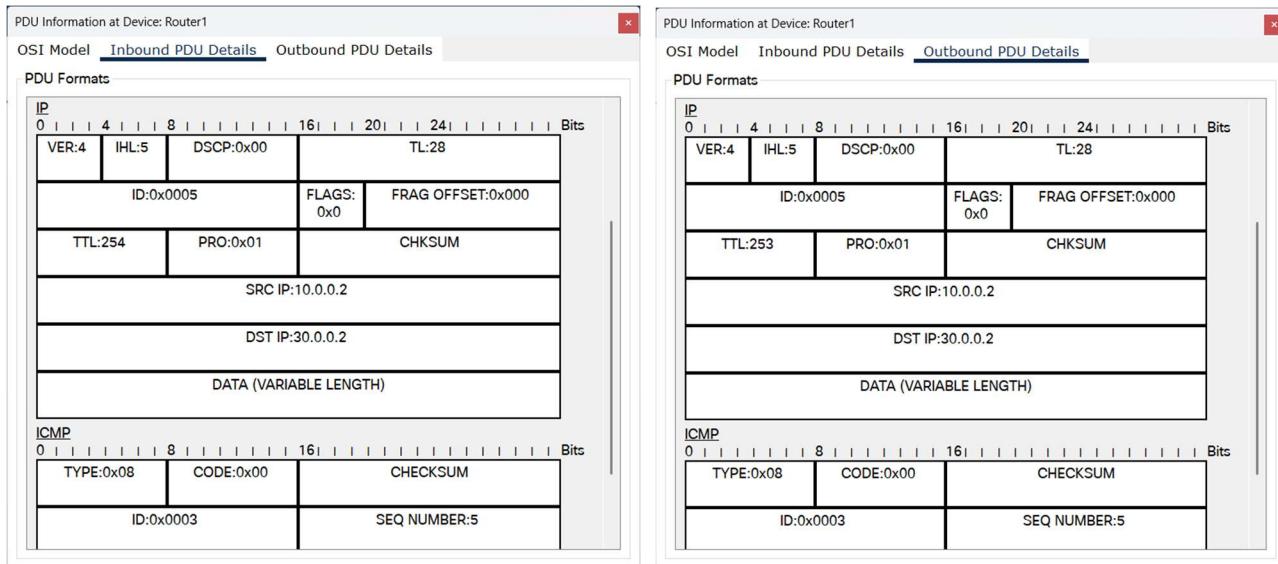


- Screen shots/ output

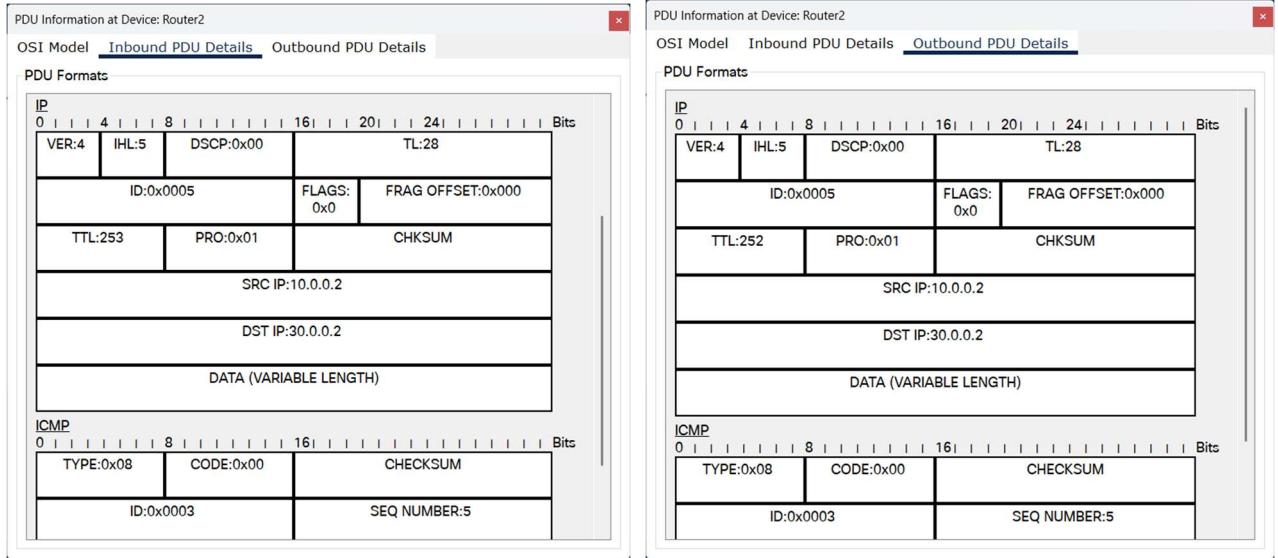
Packet at Router0



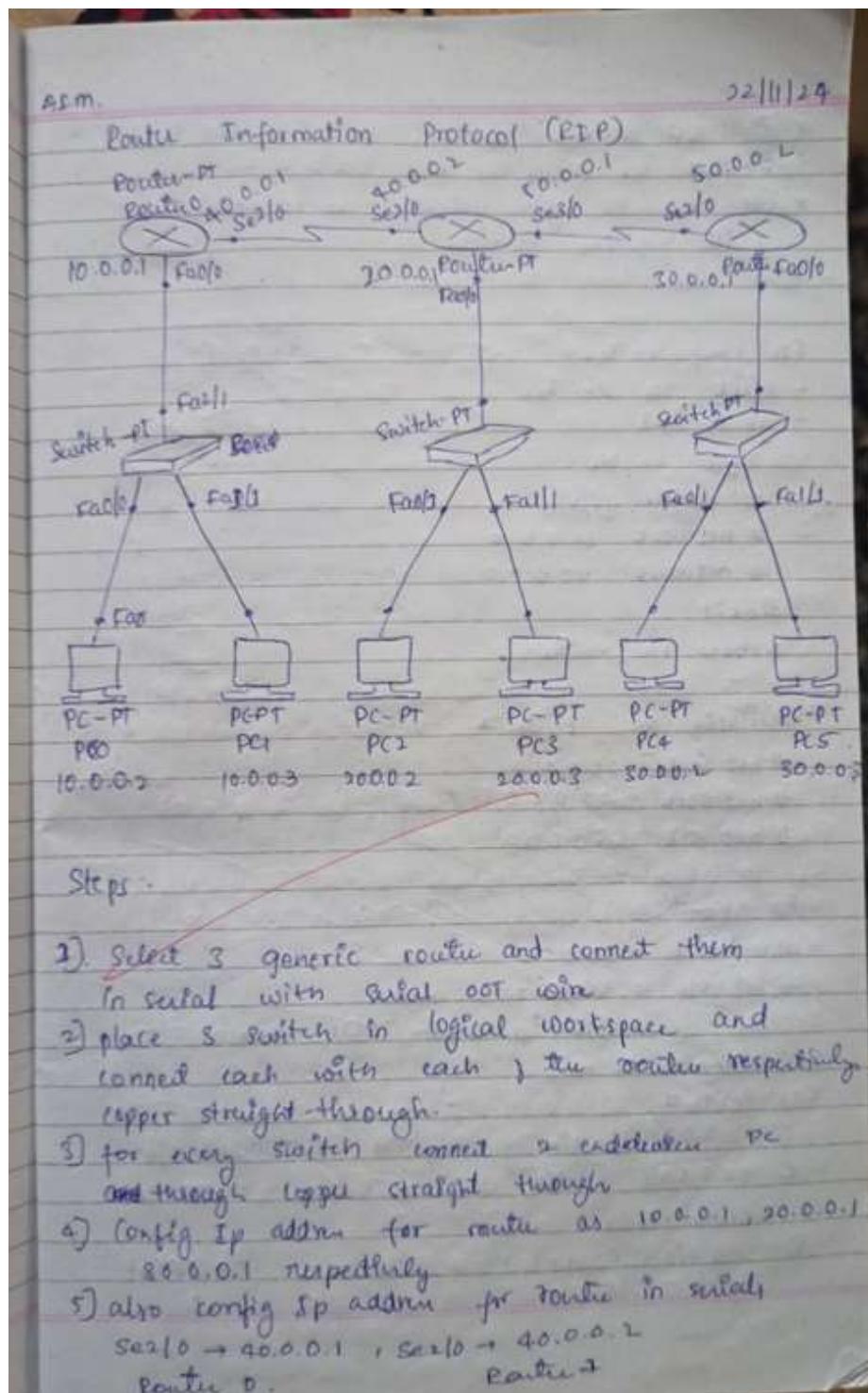
Packet at Router1



Packet at Router2

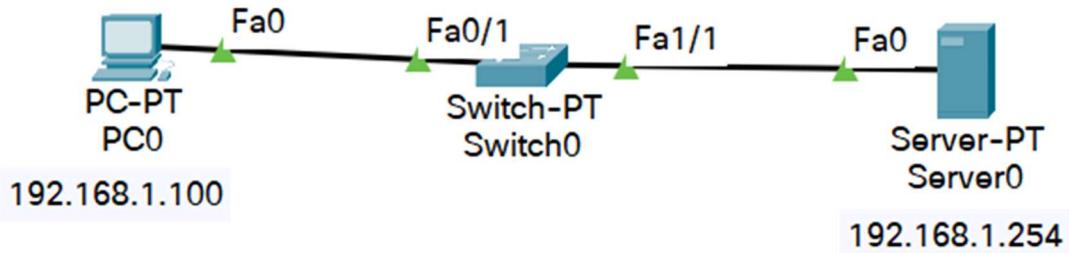


iv. Observation



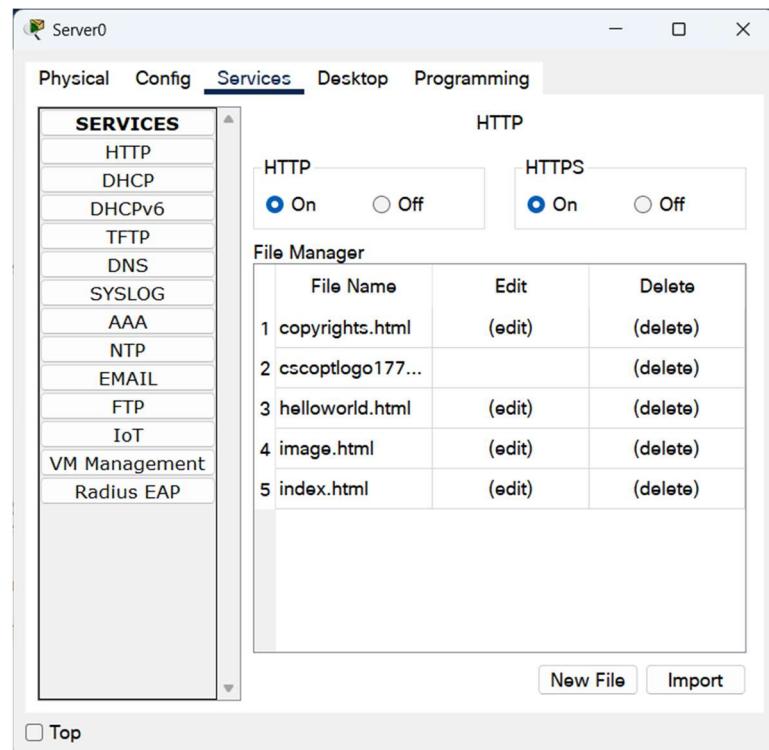
Program 8

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

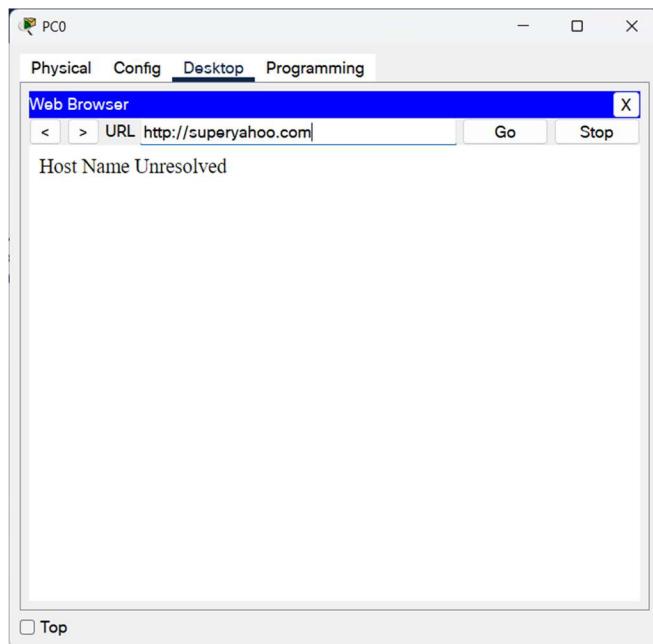
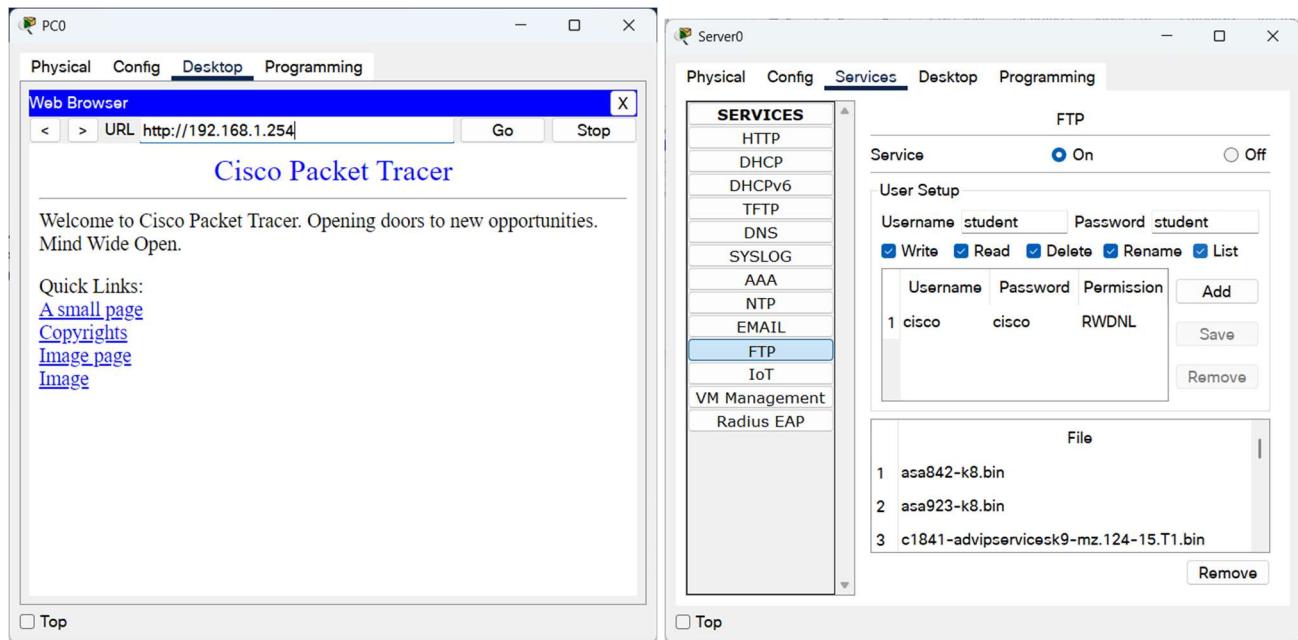


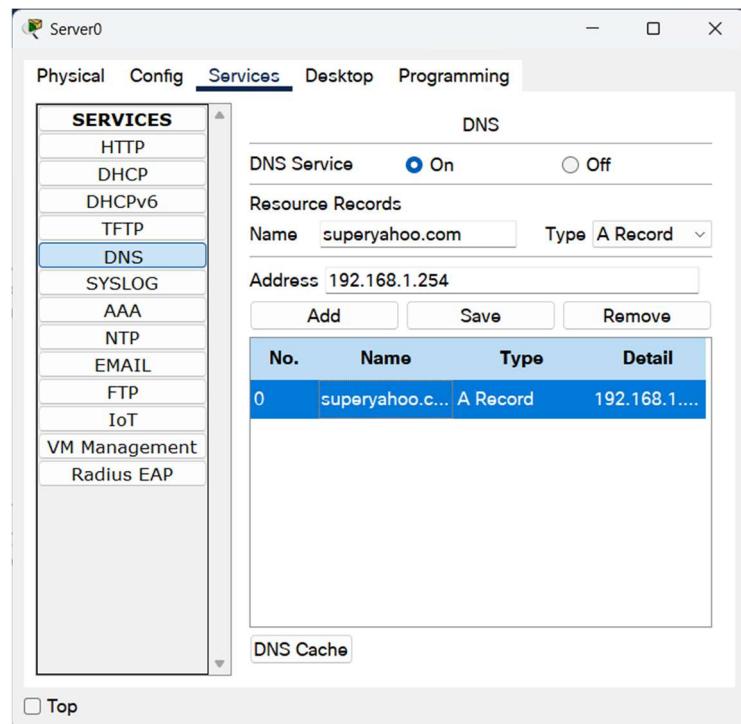
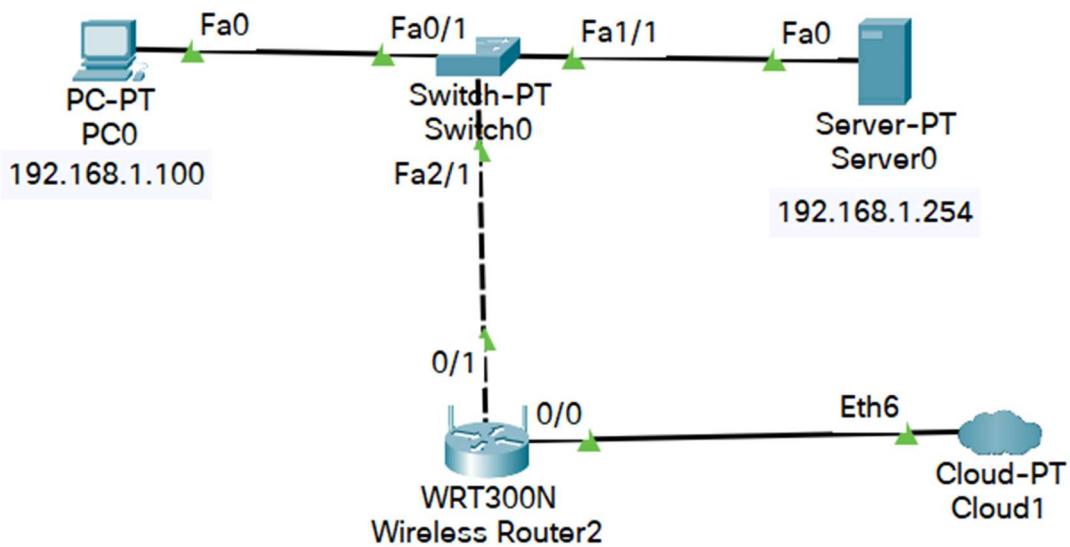
- iii. Screen shots/ output

Server's services

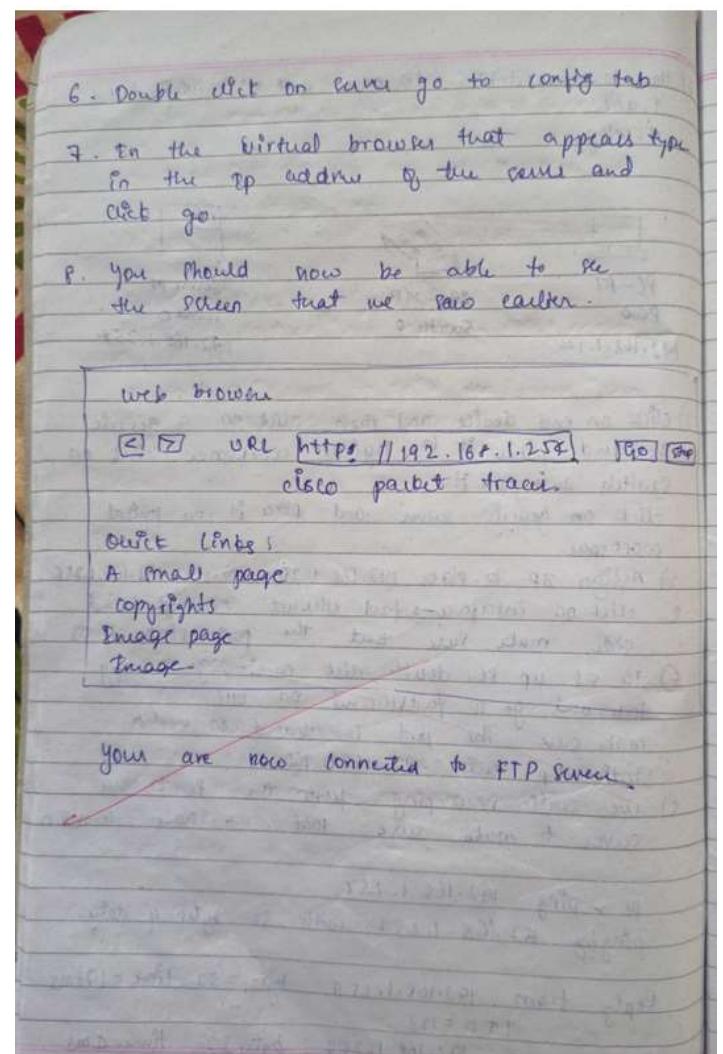
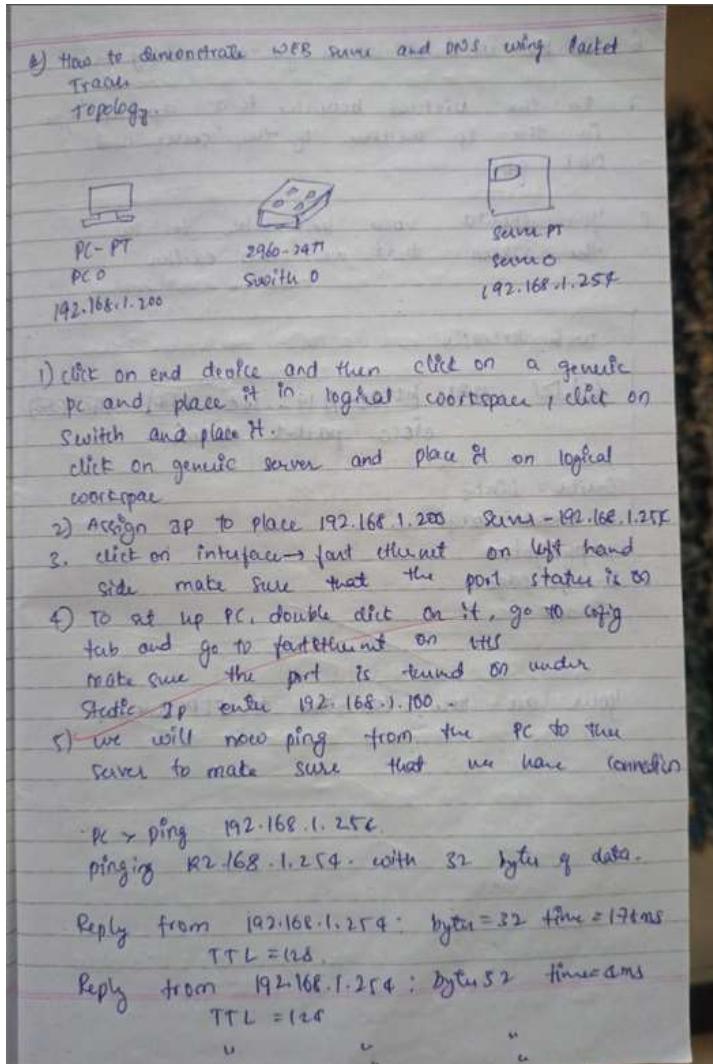


PC's Web Browser



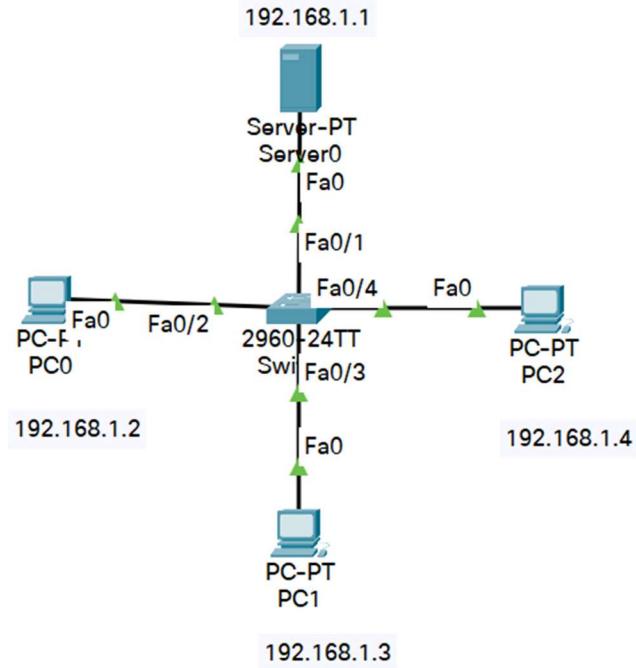


iv. Observation



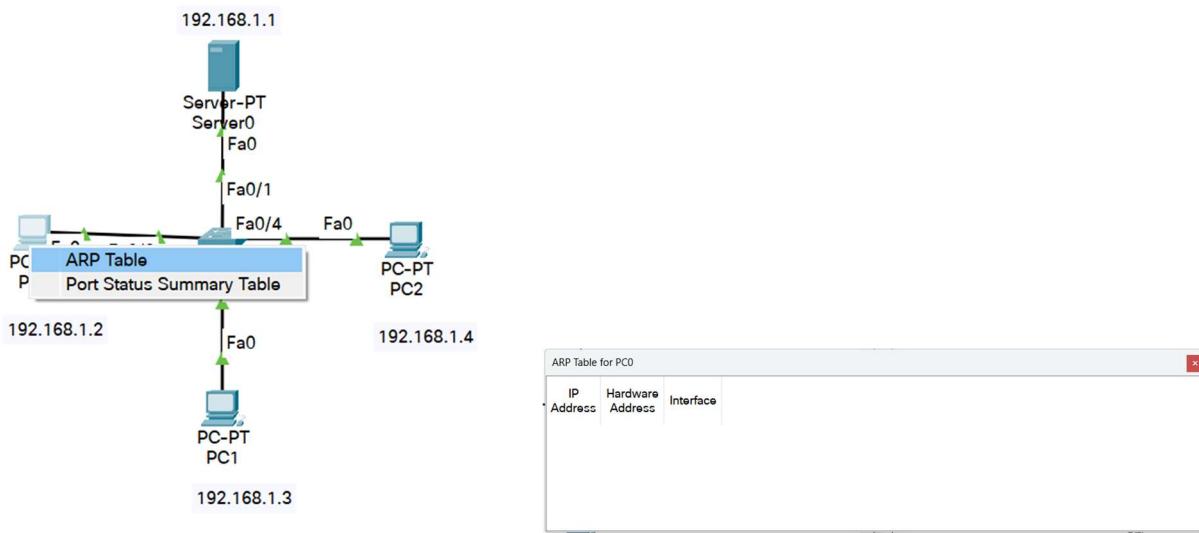
Program 9

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

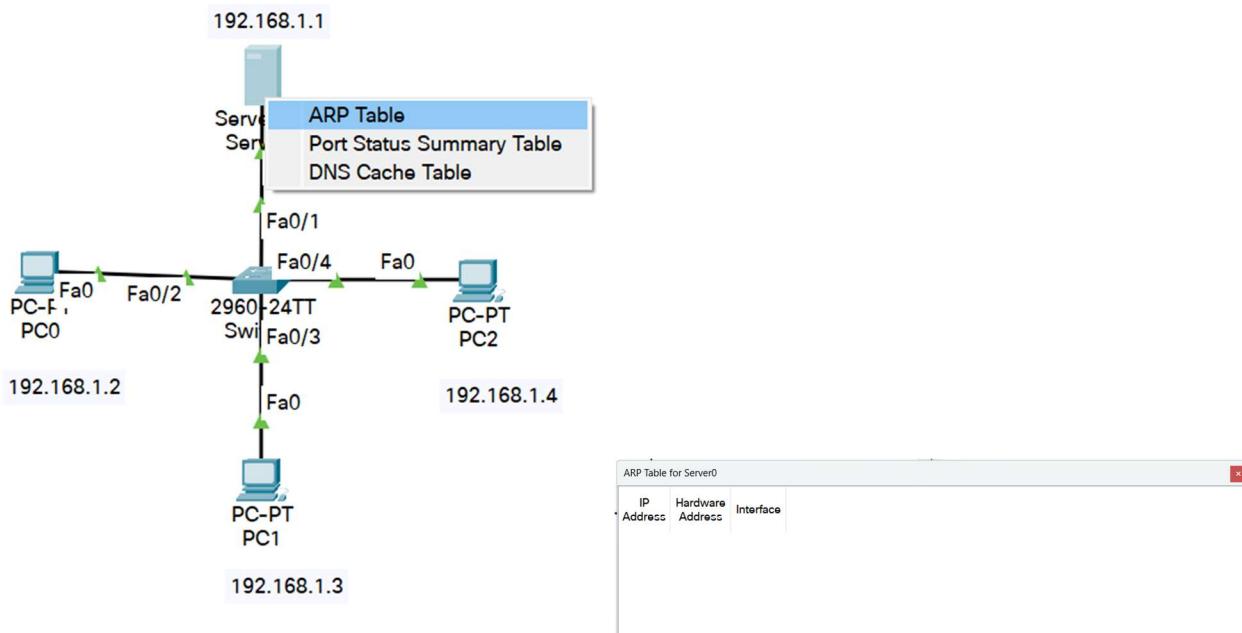


- iii. Screen shots/ output

ARP Table of PC



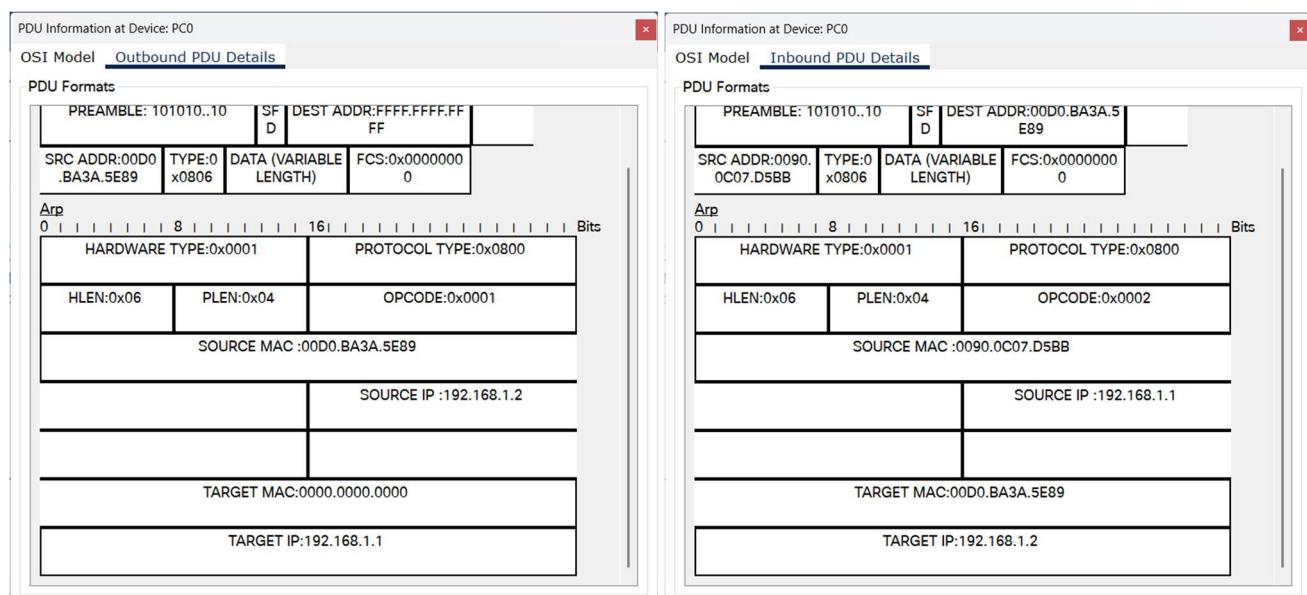
ARP Table of Server



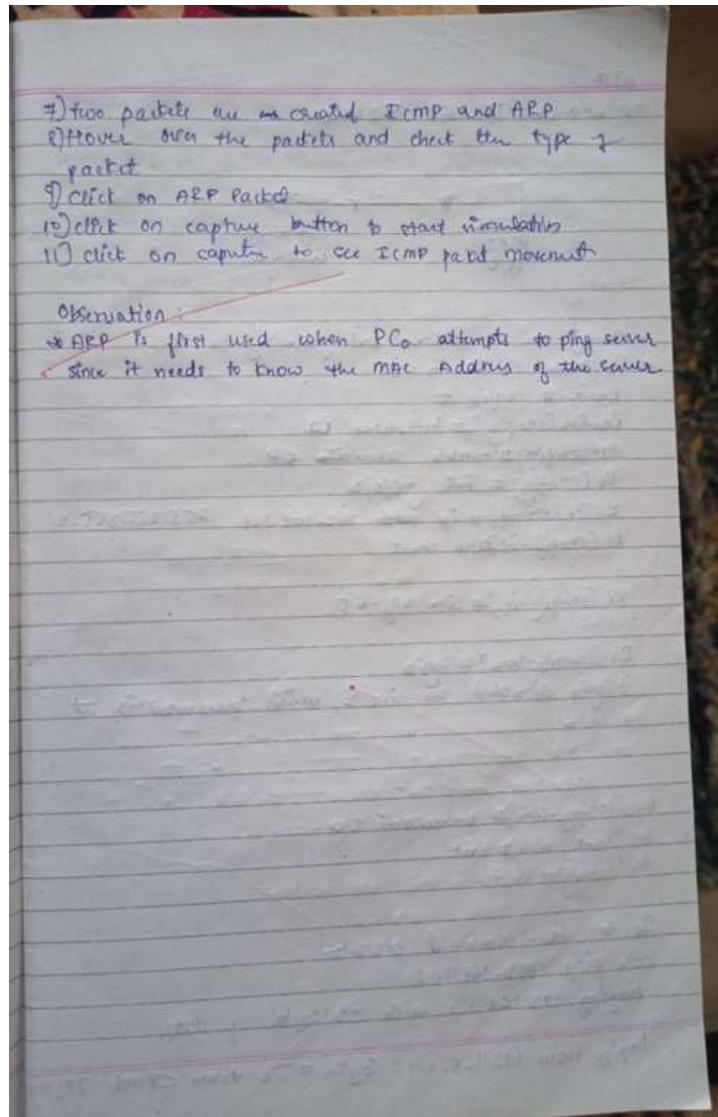
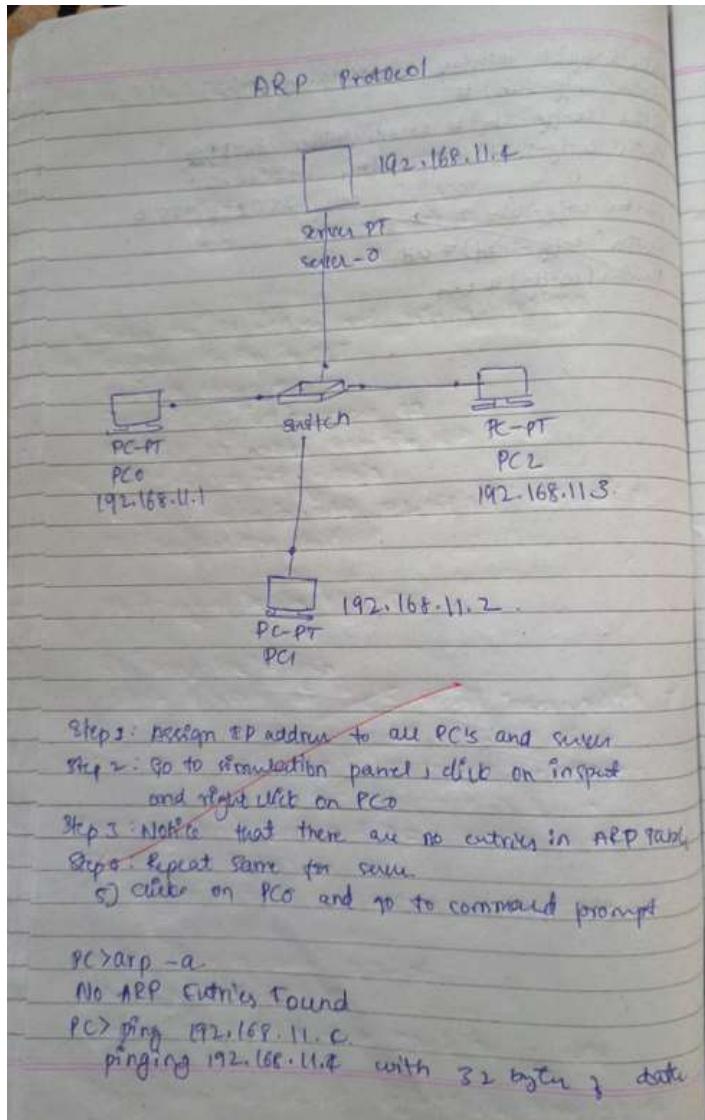
Command at PC

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

Pinging in Simulation Mode

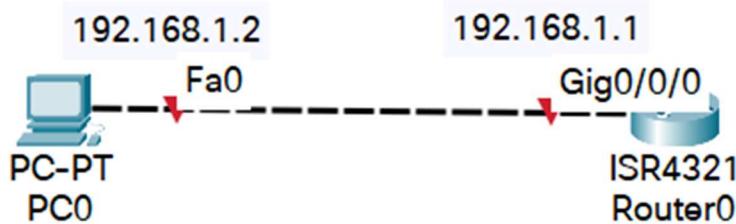


iv. Observation



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

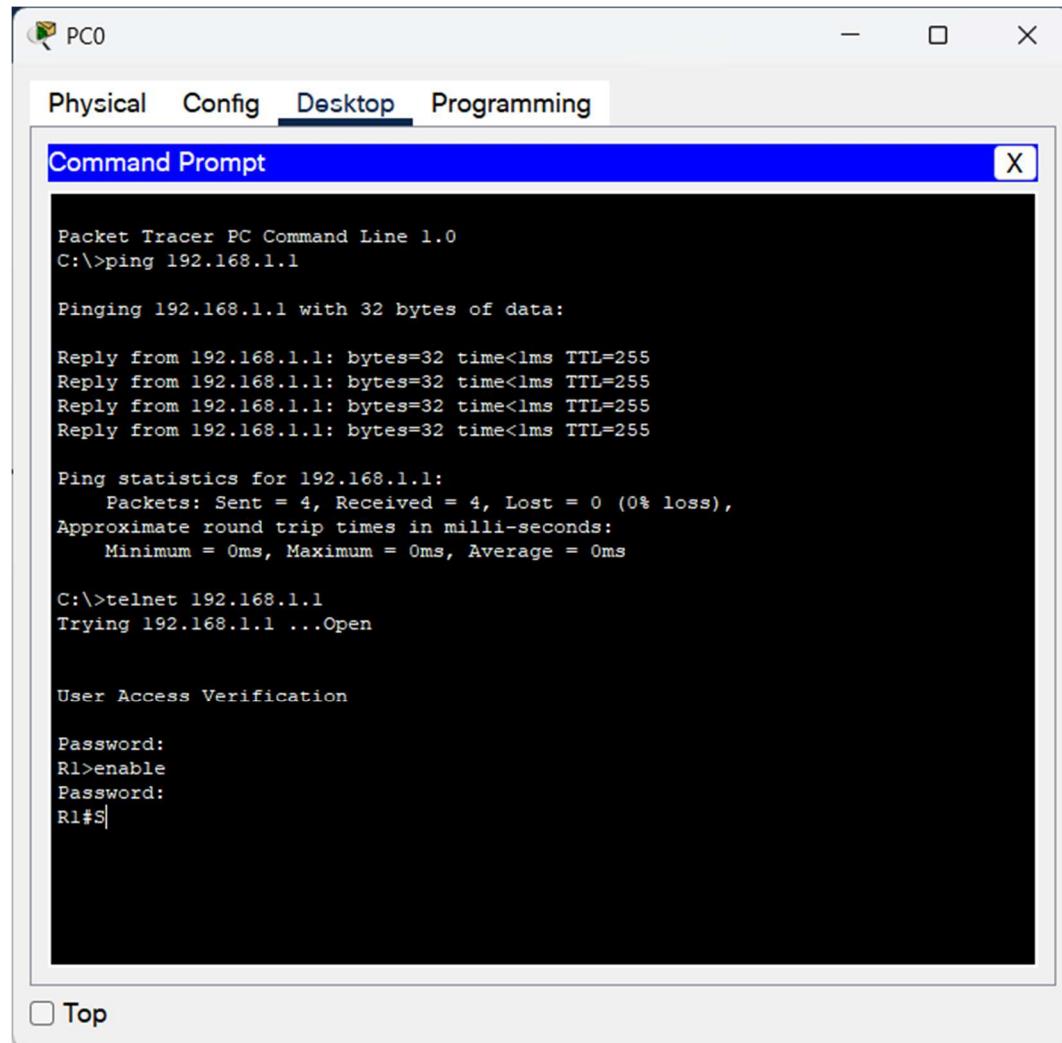
R1(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to up

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

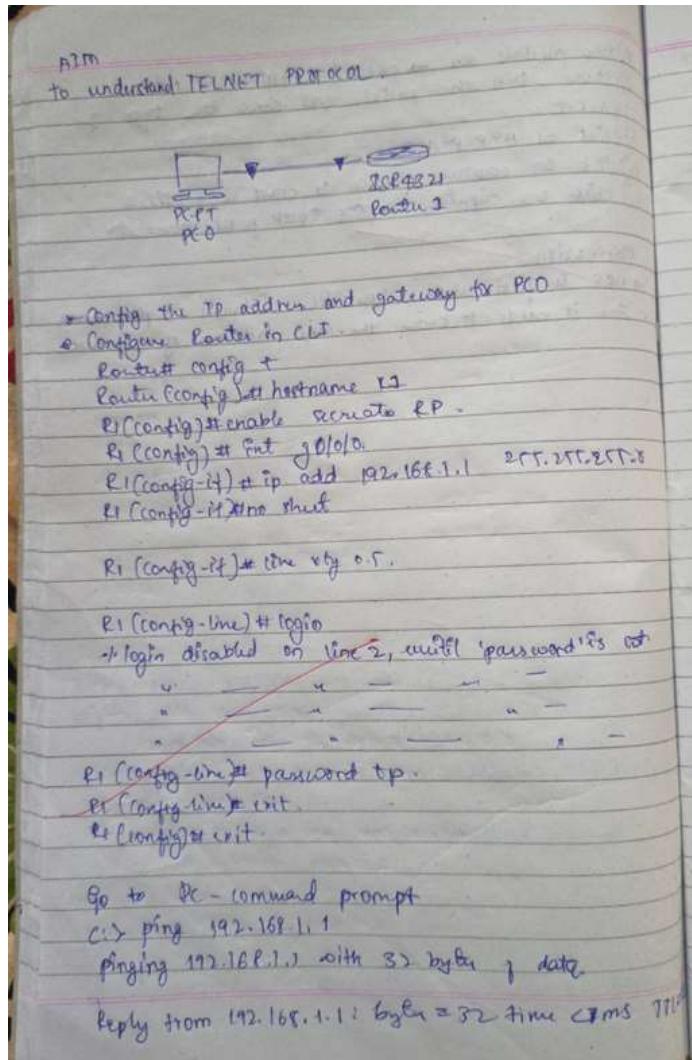
R1(config-if)#line vty 0 5
R1(config-line)#login
% Login disabled on line 2, until 'password' is set
% Login disabled on line 3, until 'password' is set
% Login disabled on line 4, until 'password' is set
% Login disabled on line 5, until 'password' is set
% Login disabled on line 6, until 'password'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



Reply from 192.168.1.1 bytes=32 time <1ms TTL=255

Ping statistics for 192.168.1.1:
 Packets: sent = 4, Received = 4, Lost = 0 (0.0% loss)
 Approximate round trip times in milli-seconds:
 Minimum = 0 ms, Maximum = 0 ms, Average = 0 ms

```

C:> telnet 192.168.1.1
Trying 192.168.1.1...Open
User Access Verification:
Password:
R1>
R1:>
R1:> en
R1:> Password:
R1#>
R1#
    
```

Observation.

Ding

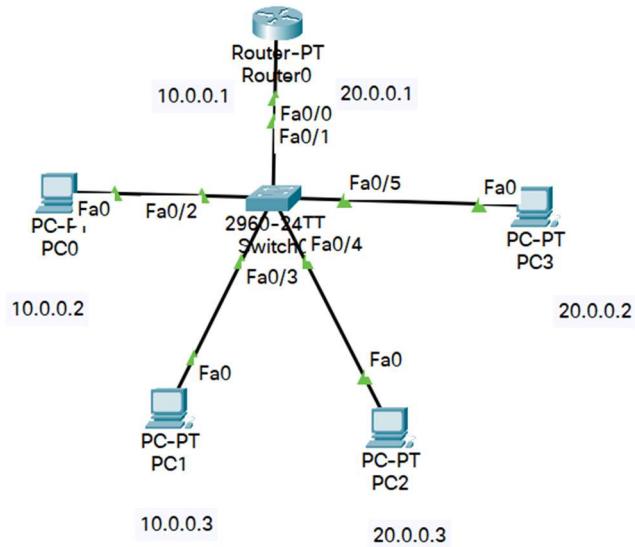
Successful pings indicate that the router is forwarding ICMP packets correctly between different subnets. ARP helps to map IP address.

Telnet

Successful telnet connections show that PC0 can establish a remote connection to PC1 / server through the router.

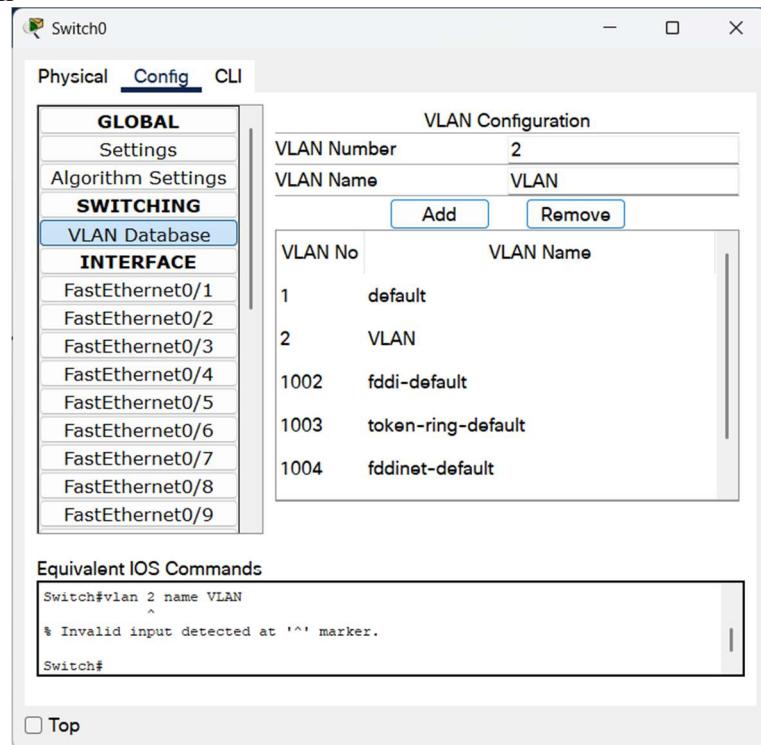
Program 11

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

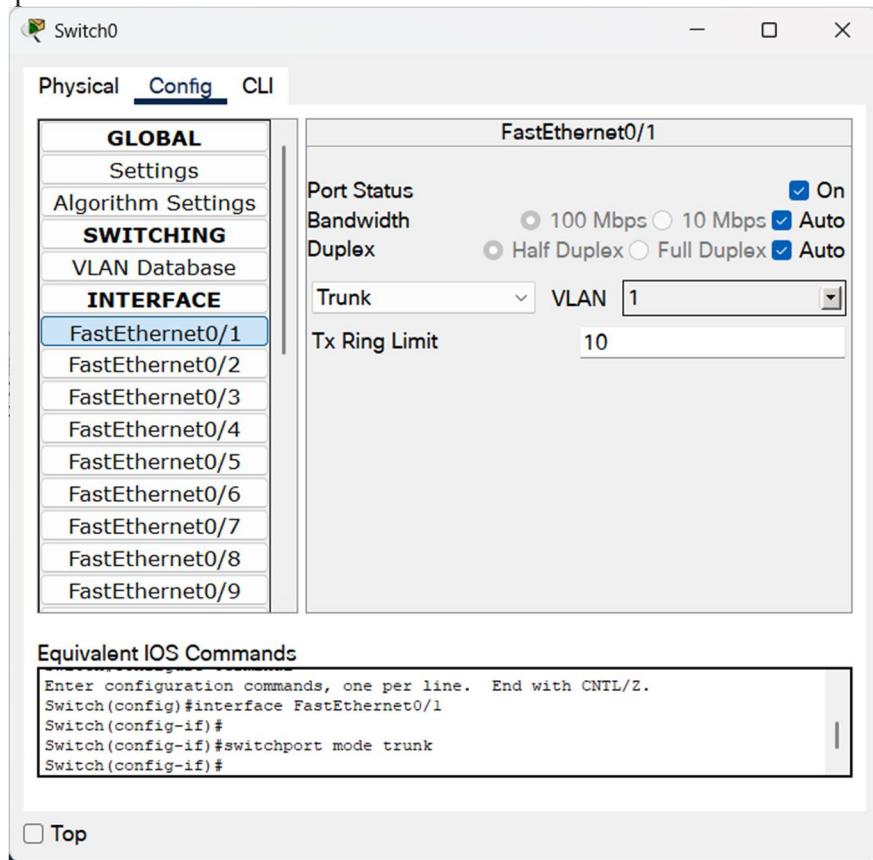


- iii. Screen shots/ output

Switch Configuration



Configuration of port connected to router



Configuration of ports connected to other networks

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

FastEthernet0/4 Configuration:

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

Equivalent IOS Commands:

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

```

FastEthernet0/5 Configuration:

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

The "VLAN" dropdown for FastEthernet0/5 shows three entries: "1:default" (unchecked), "2:VLAN" (checked), and "1002:fddi-default" (unchecked).

Equivalent IOS Commands:

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

```

A "Top" button is located at the bottom left of each interface panel.

Configuartion of Router

Router0

Physical Config **CLI**

IOS Command Line Interface

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

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

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

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

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

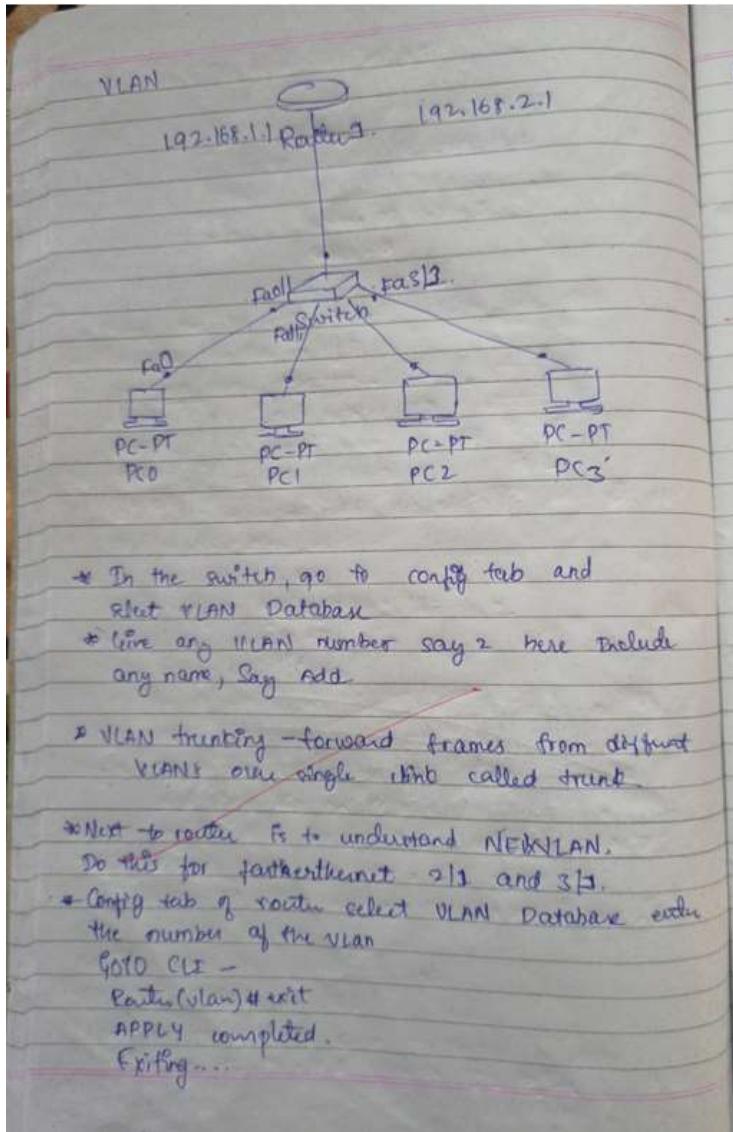
Router(config-subif)#encapsulation dot1q 2
Router(config-subif)#ip address 20.0.0.1 255.0.0.0
Router(config-subif)#no shutdown
Router(config-subif)#exit
Router(config)#

Ctrl+F6 to exit CLI focus
```

Top

Copy **Paste**

iv. Observation



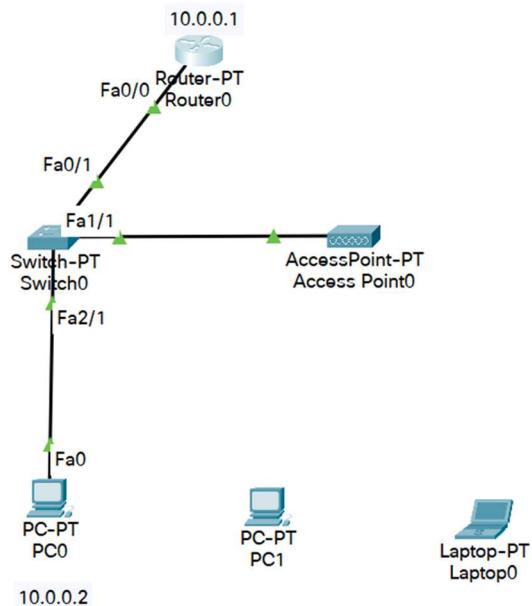
```

Router(config)# interface fastethernet 0/0.1
Router(config-subif)#
Router(config-subif)# encapsulation dot1q 2
Router(config-subif)# ip address 192.168.2.1
                           255.255.255.0
Router(config-subif)# no shutdown
Router(config-subif)# exit
Router(config)# exit

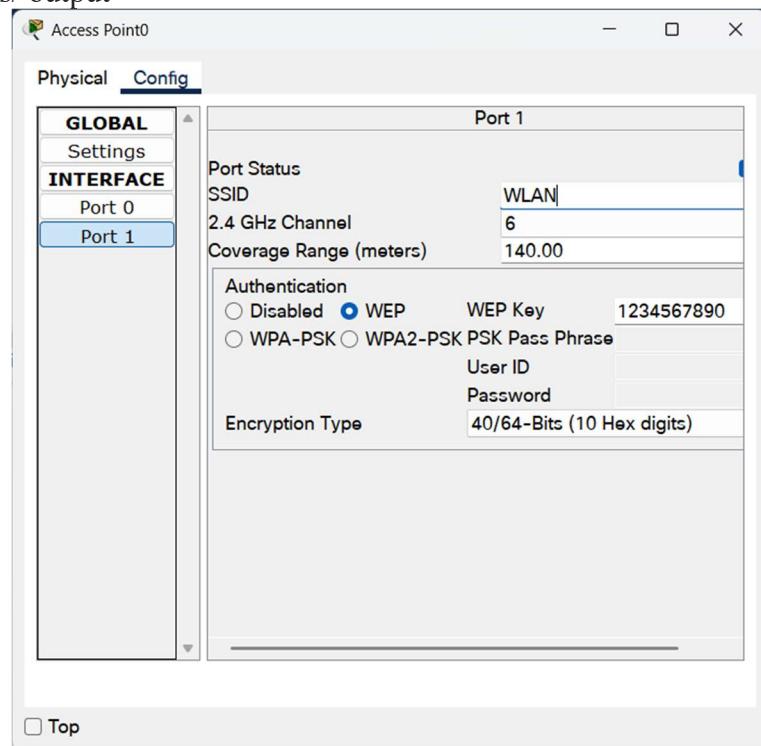
```

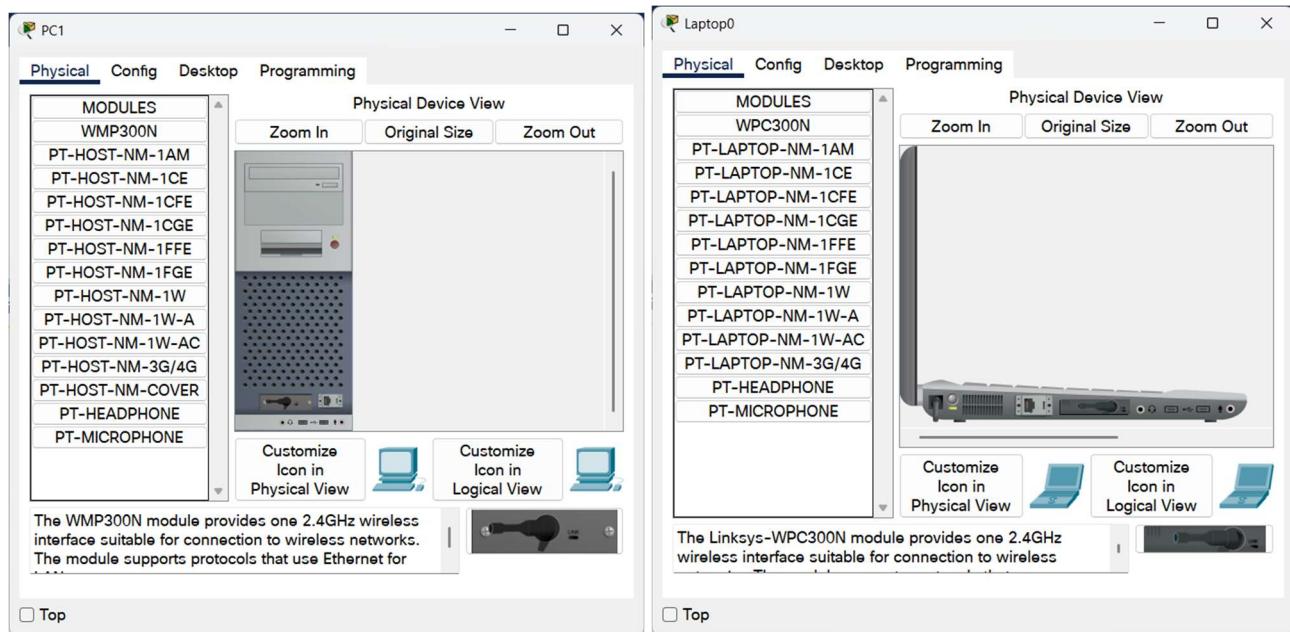
Program 12

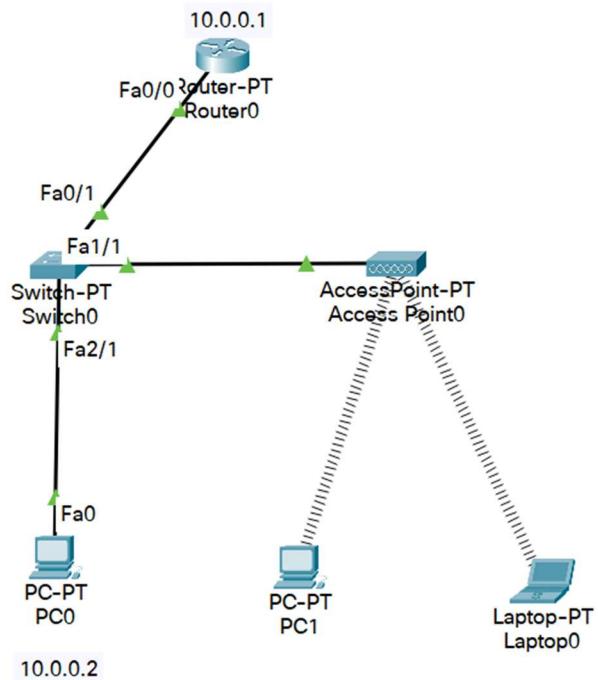
- i. To construct a WLAN and make the nodes communicate wirelessly.
- ii. Procedure along with the topology



- iii. Screen shots/ output







Ping:

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

```

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

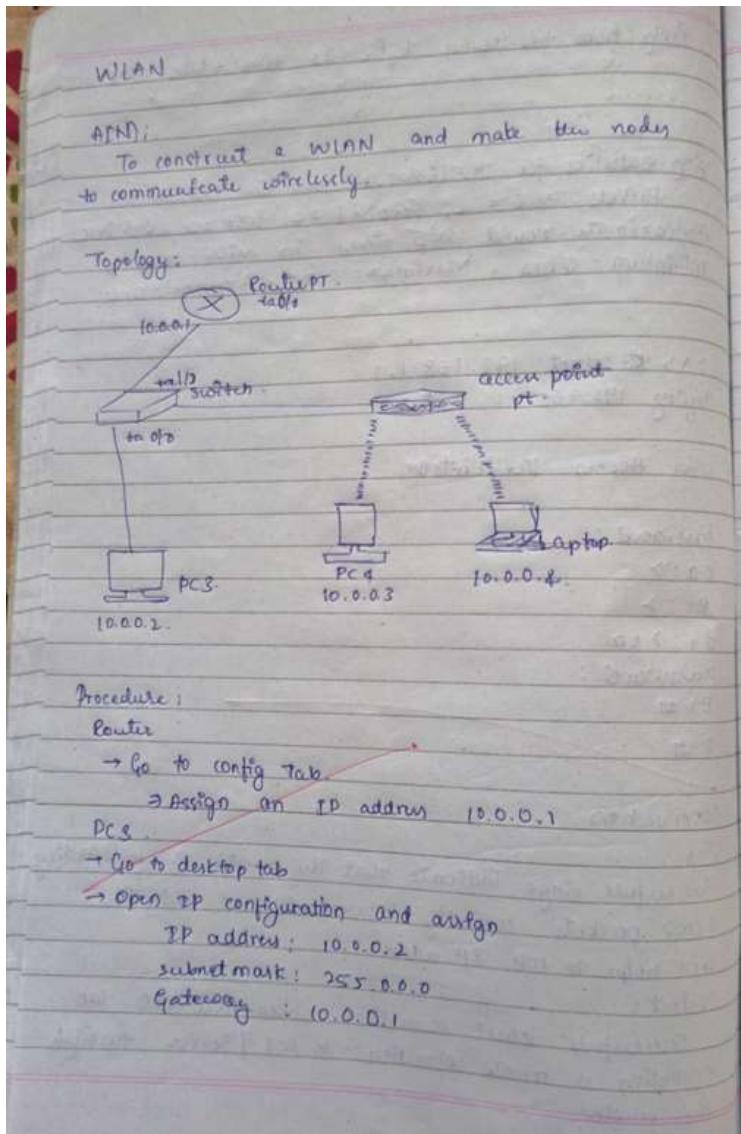
Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 24ms, Maximum = 40ms, Average = 28ms

C:\>

```

Top

iv. Observation



* Configuring Access Point:
set SSID
→ Go to config tab, select port
SSID name: WLAN
security mode: WEP
key: 1234567890(10-digit hex)

* Configuring PC4 and Laptop for wireless
Install wireless NIC
- Switch off PC4 and laptop.
→ Drag the WMP board wireless interface and to their empty ports.
→ Switch on device
Configure wireless setting:
→ Go to config tab
→ Select new wireless interface
→ set SSID : WLAN
Security mode: WEP
WEP key: 123-4567890
IP address : eg: PC4 : 10.0.0.5.
laptop : 10.0.0.4

~~31/12/24~~

Ring:
Open cmd
ping every other device

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)  
  
if __name__ == "__main__":  
    # User input for original bitstream  
    message_bits = input("Enter the original bitstream (e.g., 11010011101100): ")  
  
    # Calculate and append CRC  
    bitstream_with_crc = append_crc_to_bitstream(message_bits)  
    print(f'Bitstream with CRC: {bitstream_with_crc}')
```

```

# User input for verification
user_bitstream = input(
    "Enter the received bitstream for verification (e.g., 11010011101100110110110111000011): "
)

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

```

iii. Screen shots/ output

```

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

Enter the original bitstream (e.g., 11010011101100): 11111
Bitstream with CRC: 1111111111111100000

Enter the received bitstream for verification (e.g., 11010011101100110110110111000011): 111111111111111100000
CRC valid: True

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

Enter the original bitstream (e.g., 11010011101100): 11111
Bitstream with CRC: 1111111111111100000

Enter the received bitstream for verification (e.g., 11010011101100110110110111000011): 111111111111111100001
CRC valid: False

```

iv. Observation

ADM

15/11/24

Write a program for error detection using CRC.

```
#include <iostream>
#include <string>
using namespace std;

int crc(const string &p, string &op, const string &poly,
        int mode) {
    op = p;
    if (mode) {
        op.append(poly.length() - 1, '0');
    }
    for (int i = 0; i < op.length(); i++) {
        if (op[i] == '1') {
            for (int j = 0; j < poly.length(); j++) {
                op[i + j] = (op[i + j] == poly[j]) ?
                    '0' : '1';
            }
        }
    }
    for (int i = 0; i < op.length(); i++) {
        if (op[i] == '1') return 0;
    }
    return 1;
}

int main() {
    string ip, op, recv;
    string poly = "1000100000100001";
    cout << "Enter Input: ";
    cin >> ip;
    crc(ip, op, poly, 1);
}
```

```
cout << "Transmitted message: " <<  
    ip + op.substr(ip.length(), 1) << endl;
```

```
cout << "Enter the received message in  
binary: ";  
cin >> recv;
```

```
if (crc(recv, op, poly, 0)) {  
    cout << "No error in data" << endl;  
} else
```

```
    cout << "Error in data transmission  
has occurred" << endl;
```

3.

Output: ~~C:\Users\HP\OneDrive\Documents\GitHub\Project 1\src\main.cpp~~
> g++ main.cpp
12.out

Enter input

11111.

Transmitted message:

~~11111100011110111110~~

Enter the received message in binary

11111

No error in data.

Output 2.

Enter input

11111

The transmitted message ~~1111110001110~~

Enter received message

1111

Error in data transmission has occurred.

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

A3m
 To explore Leaky Bucket Algorithm.

#include <stdio.h>
 using namespace std;

```

int main() {
    int no, storage, output-pkt-size;
    int input-pkt-size, bucket-size; sz-left;

    storage = 0;
    n = 4;
    bucket-size = 10;
    input-pkt-size = 4;
    output-pkt-size = 1;

    for (int i=0; i<n; i++) {
        sz-left = bucket-size - storage;
        if (Input-pkt-size <= sz-left) {
            storage += Input-pkt-size;
        } else {
            cout << "packet loss = " << i+1 << endl;
            *Input-pkt-size;
        }
        cout << "Buffer size = " << storage << endl;
        cout << "out of bucket size = " << bucket-size;
    }

    storage = output-pkt-size;
}

```

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

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.

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 file name: ")
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print("In From 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 True:
    print("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
```

try:
 file("The server is ready to receive")
 file = open(sentence, "r")
 t = file.read(1024)
 connectionSocket.send(t.encode())
 print("To Sent contents of "+ sentence)
 file.close()
except FileNotFoundError:
 connectionSocket.send("File not found".encode())
 print("File content not found")
connectionSocket.close()

Output:

⇒ python ServerTCP.py python ClientTCP.py.
The server is ready to receive.
sent contents of ServerTCP.py

⇒ python 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)
#whole code above)

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)
```

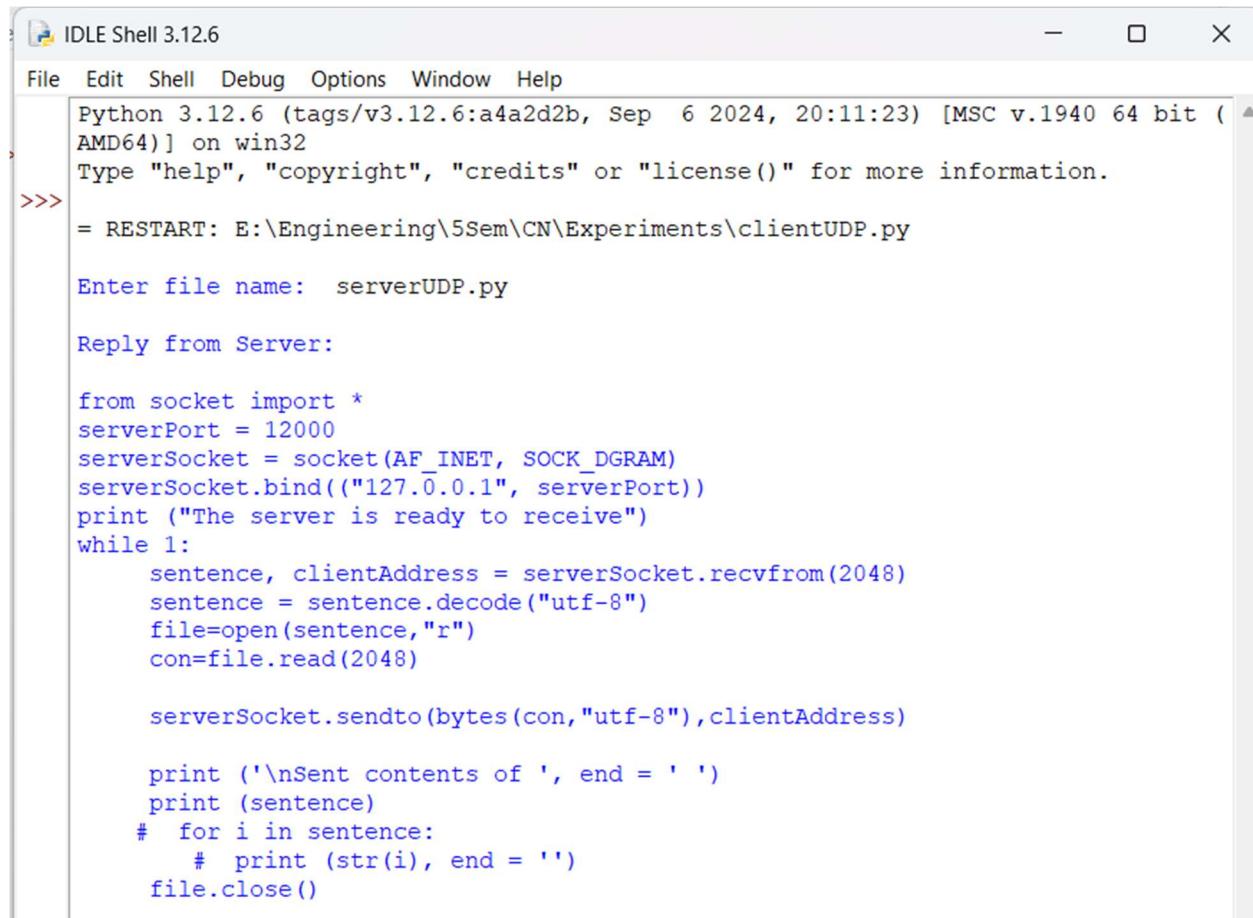
```
serverSocket.sendto(bytes(con,"utf-8"),clientAddress)
```

```
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

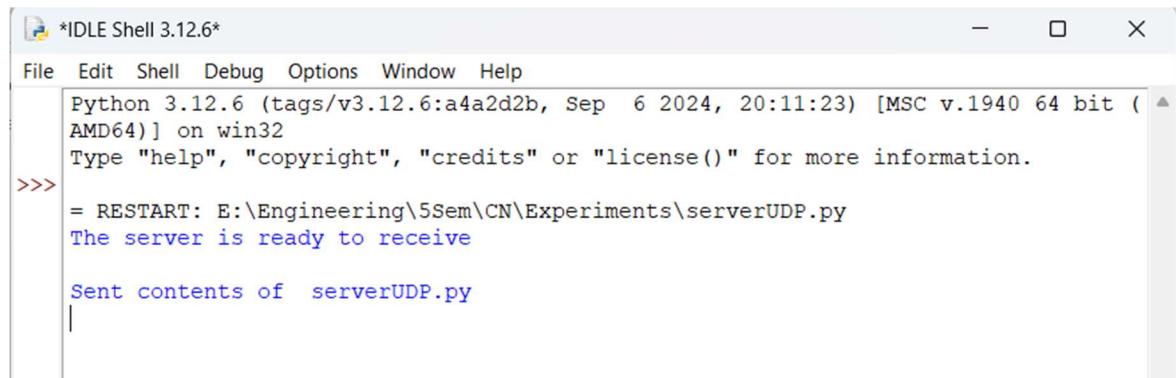
Reply from Server:

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

Using UDP sockets, write a client-server program to make client sending the file name and the send the bat f contents + the requested file if present.

Solution

```
ClientUDP.py
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
```

```
sentence = input("In Enter the file")
clientSocket = socket(AF_INET, SOCK_DGRAM)
clientSocket.sendto(sentence.encode('utf-8'), (serverName, serverPort))

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

~~ServerUDP~~

```
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)
```

ServerSocket,sendto(Coys([con,"utf-8"],clientAddress))

```
print("In Sent contents : ", end=" ")
print(sentence)
for i in sentence:
    print(str(i), end=" ")
fileclose()
```

Solution:

Python serverUDP.py:

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

The server is ready to receive

Python ClientUDP.py.

Reply from Server

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
#whole code.
```