

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB RECORD

Computer Network Lab (23CS5PCCON)

Submitted by

Sanjeet Prajwal Pandit (1BM22CS241)

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 **Sanjeet Prajwal Pandit (1BM22CS241)**, 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.

Ramya K M 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	5
3	25/10/2024	Configure default route, static route to the router	10
4	08/11/2024	Configure DHCP within a LAN and outside LAN.	16
5	22/11/2024	Configure RIP routing Protocol in Routers	21
6	22/11/2024	Configure OSPF routing protocol	24
7	22/11/2024	Demonstrate the TTL/ Life of a Packet	30
8	08/11/2024	Configure Web Server, DNS within a LAN.	32
9	20/12/2024	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)	36
10	20/12/2024	To understand the operation of TELNET by accessing the router in server room from a PC in IT office.	39
11	20/12/2024	To construct a VLAN and make the PC's communicate among a VLAN	42
12	20/12/2024	To construct a WLAN and make the nodes communicate wirelessly	46

Github Link: https://github.com/Sanjeet-108/CN_lab

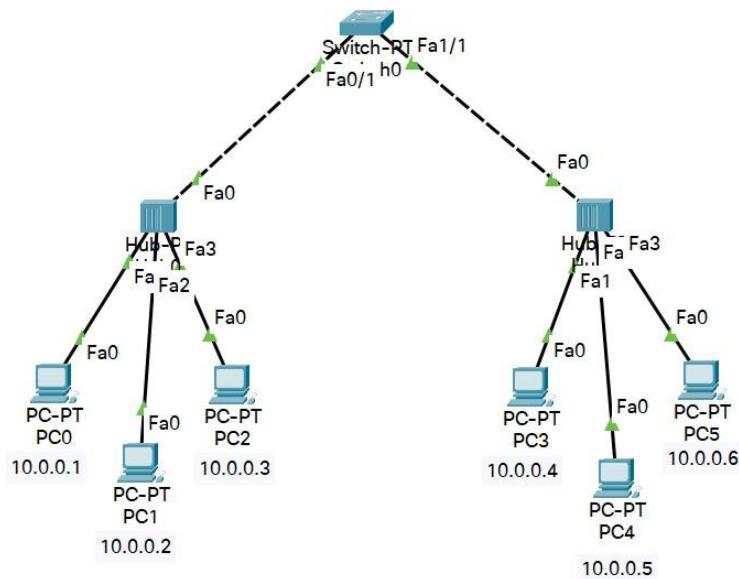
Index-Cycle-II

Sl. No.	Date	Experiment Title	Page No.
1	15/11/2024	Write a program for error detecting code using CRC-CCITT (16-bits).	50
2	15/11/2024	Write a program for congestion control using Leaky bucket algorithm	52
3	20/12/2024	Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.	54
4	20/12/2024	Using UDP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.	57

Cycle-I

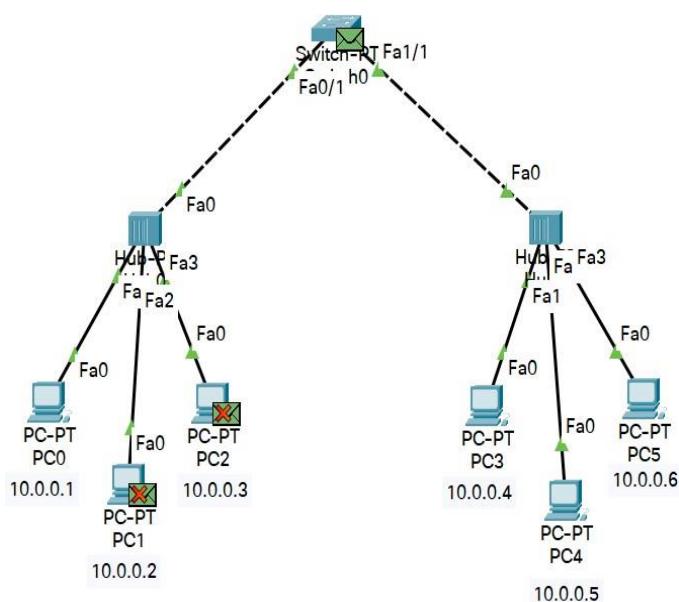
Program 1

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

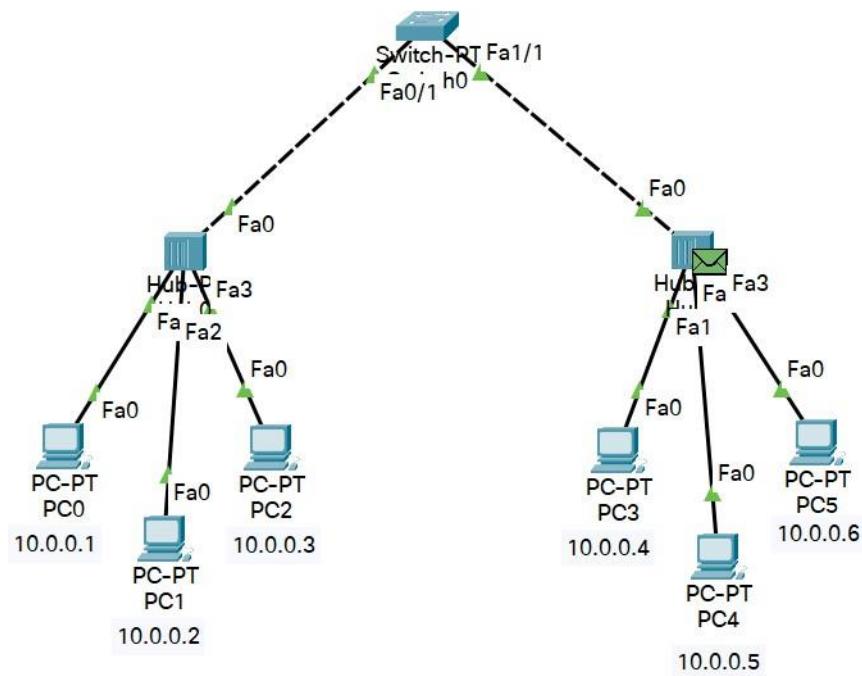


- iii. Screen shots/ output

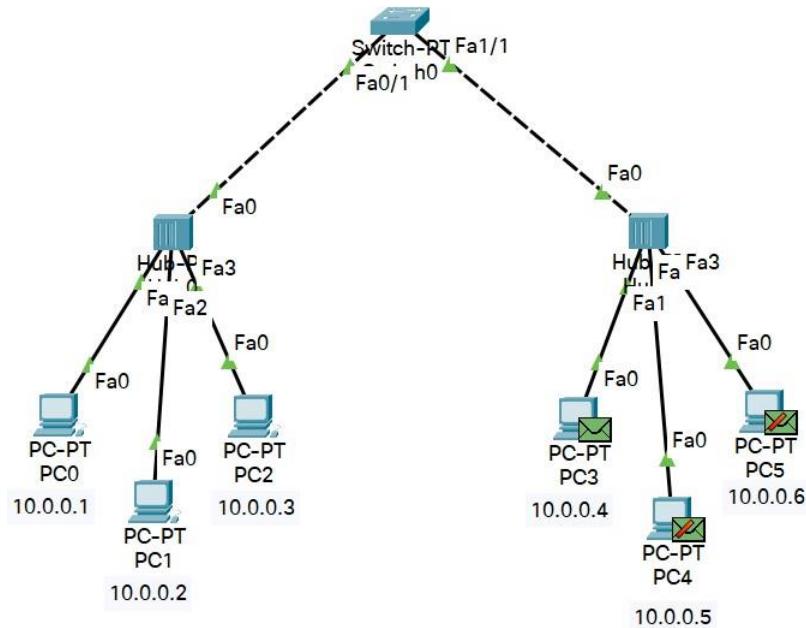
Hub behaviour at sending end



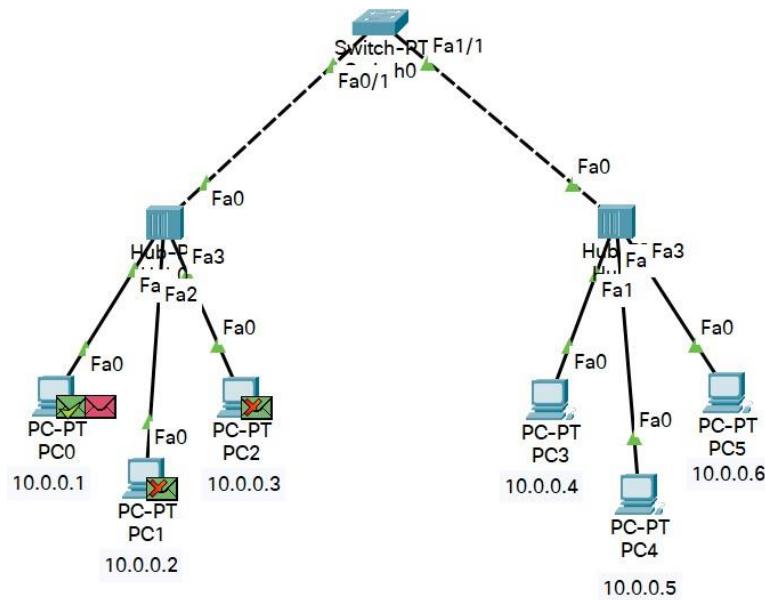
Switch behaviour



Hub behaviour at receiving end



Hub behaviour when back to sender



Ping command to connectivity

A screenshot of a Windows Command Prompt window titled "PC0". The window has tabs for Physical, Config, Desktop, and Programming, with Desktop selected. The Command Prompt window shows the following output:

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

Pinging 10.0.0.4 with 32 bytes of data:

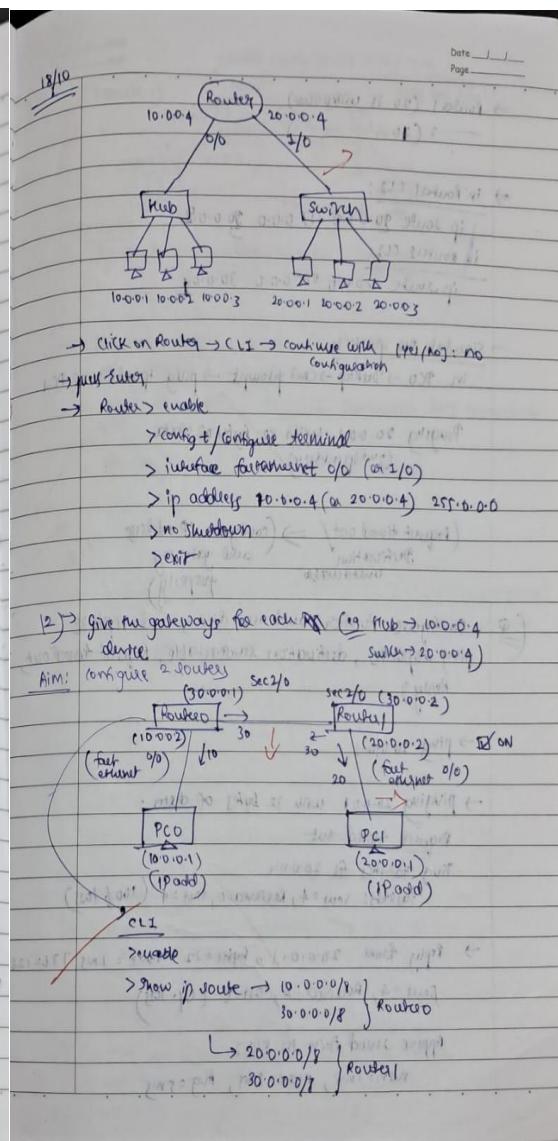
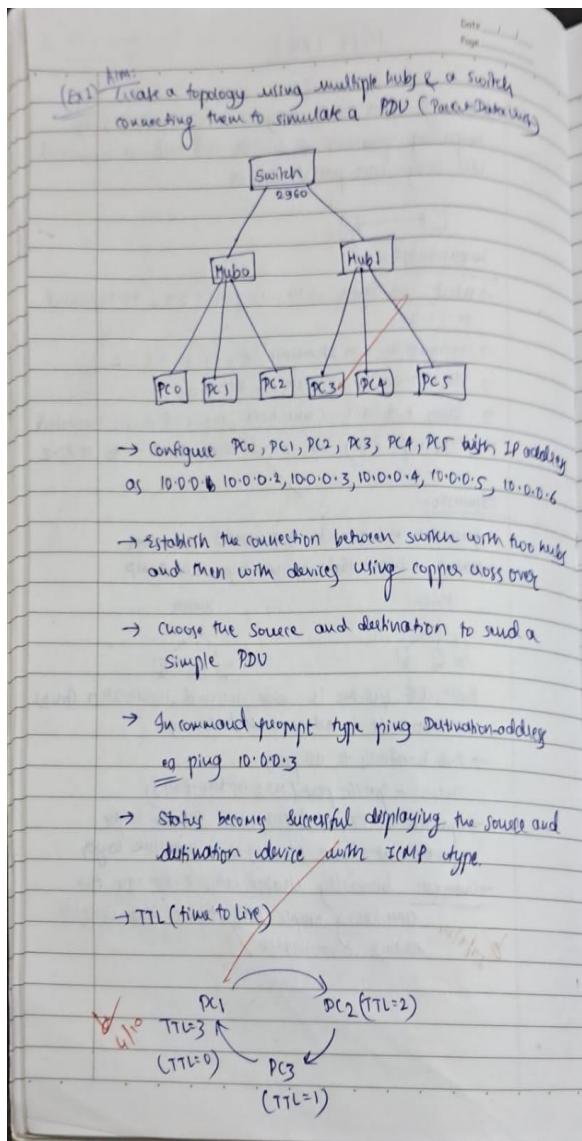
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time=<1ms TTL=128
Reply from 10.0.0.4: bytes=32 time<1ms TTL=128

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

C:\>
```

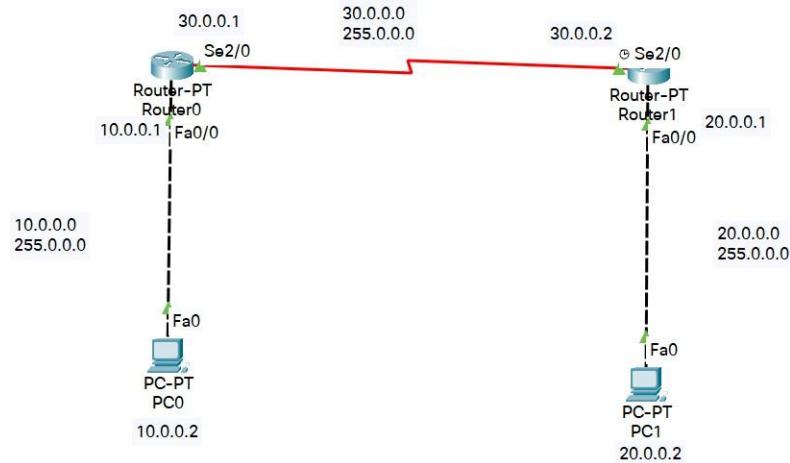
At the bottom left of the window, there is a checkbox labeled "Top".

iv. Observation



Program 2

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



- iii. Screen shots/ output

Router0 configuration

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

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

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

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

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

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

Router1 configuration

The screenshot shows a 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" 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 of the window, there are buttons for "Ctrl+F6 to exit CLI focus", "Copy", and "Paste". There is also 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

```

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

Pinging 20.0.0.2 with 32 bytes of data:

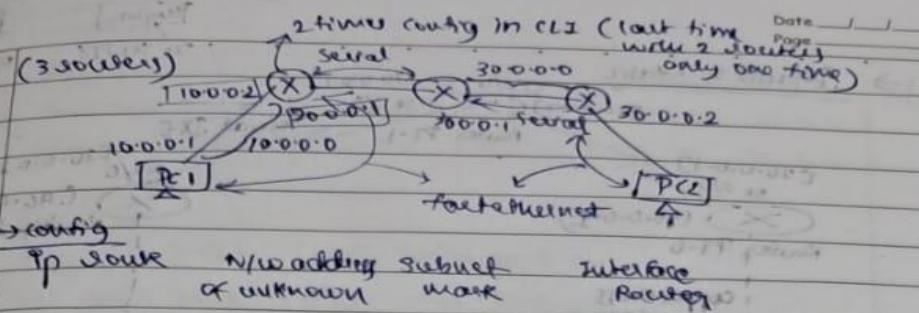
Reply from 20.0.0.2: bytes=32 time=1ms TTL=126
Reply from 20.0.0.2: bytes=32 time=18ms TTL=126
Reply from 20.0.0.2: bytes=32 time=1ms TTL=126
Reply from 20.0.0.2: bytes=32 time=1ms TTL=126

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 1 (20 is unknown)
 → 2 (20)
 → in Router 1 CL1:
 ip route 20.0.0.0 255.0.0.0 30.0.0.2
 in Router 2 CL2:
 ip route 10.0.0.0 255.0.0.0 30.0.0.1
 → Simulate the packets
 in PC0 → Desktop → Cmd prompt → ping ip.address of PC1
 Ping 20.0.0.1 with 32 bytes of data
 (2nd ip address)
 (Request timed out / Destination unreachable) ⇒ (Configure IP address and gateway properly)
 (Configure Router and end devices and replace ping responses, destination unreachable, Request timed out, Reply)
 → ping 20.0.0.1
 → Ping 20.0.0.1 with 32 bytes of data:
 Request timed out
 Ping statistics for 20.0.0.1
 Packets: sent=4, received=0, lost=4 (100% loss)
 → Reply from 20.0.0.1 bytes=32 time=1ms TTL=126
 Sent=4, Received=4, Lost=0 (0% loss)
 Approx round trip in ms:
 min=1ms, max=3ms, Avg=1ms



→ Simulate gateway from PRO to PC,
Ping 20.0.0.1

If: Reply from 10.0.0.2 Destination host unreachable
(Router unable to find default gw)

Router > Show ip route

✓

Outcome: Configured IP addresses on routers, and explored various ping messages including successful replies, requests timed out, destination unreachable errors.

(160) Voss) *Pitcairnia*, 2: 12. 1942.

1.0-0.8 0.6-0.4
old individuals from unmanaged forest and plant

(*see also* *the following* *approaches*) *see also* *see also*

3. *Tomarita* (Fr. *tomarita*) *utriculus*, 15 mm. - 30 mm.
3. *Tomarita* (Fr. *tomarita*) *utriculus*, 15 mm. - 30 mm.

As time progresses we see the same

3 7 2008

2000.00 810-00402

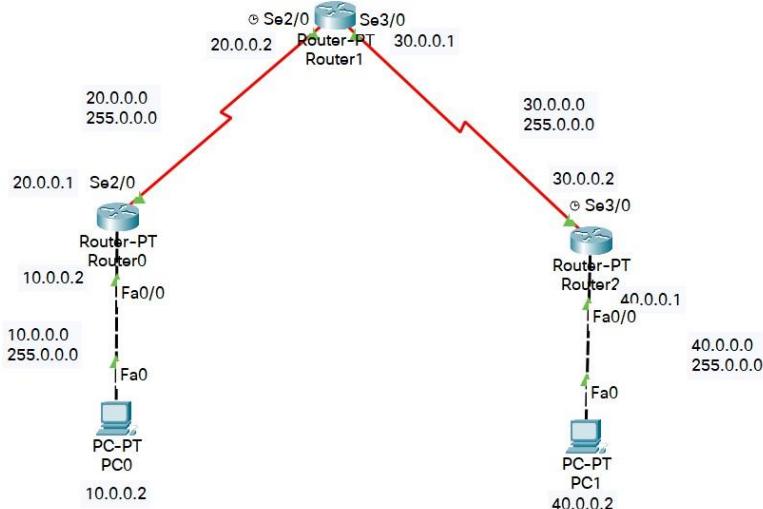
(one lesson) (one lesson)

1999-02-02 10:13:30.000000 +0000 : 800000

(3) *Individuo* pertence à *espécie* *humana*.

Program 3

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



- iii. Screen shots/ output

Router0 configuration

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

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

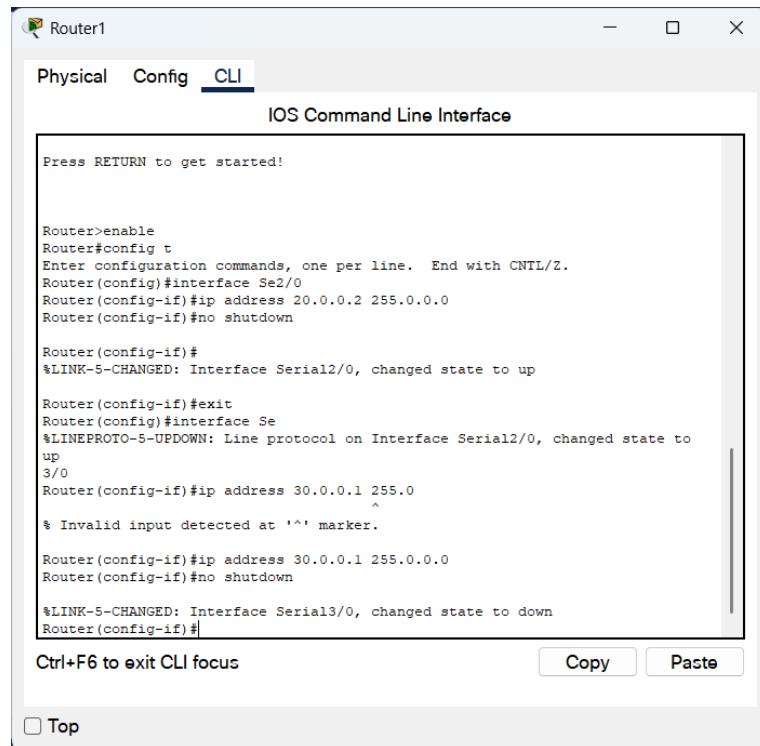
$LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed
state to up
$IP-4-DUPADDR: Duplicate address 10.0.0.2 on FastEthernet0/0, sourced by
00C.CFC2.65B0

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

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

Would you like to enter the initial configuration dialog? [yes/no]: n
Press RETURN to get started!
```

Router1 configuration



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

```
Press RETURN to get started!

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

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

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

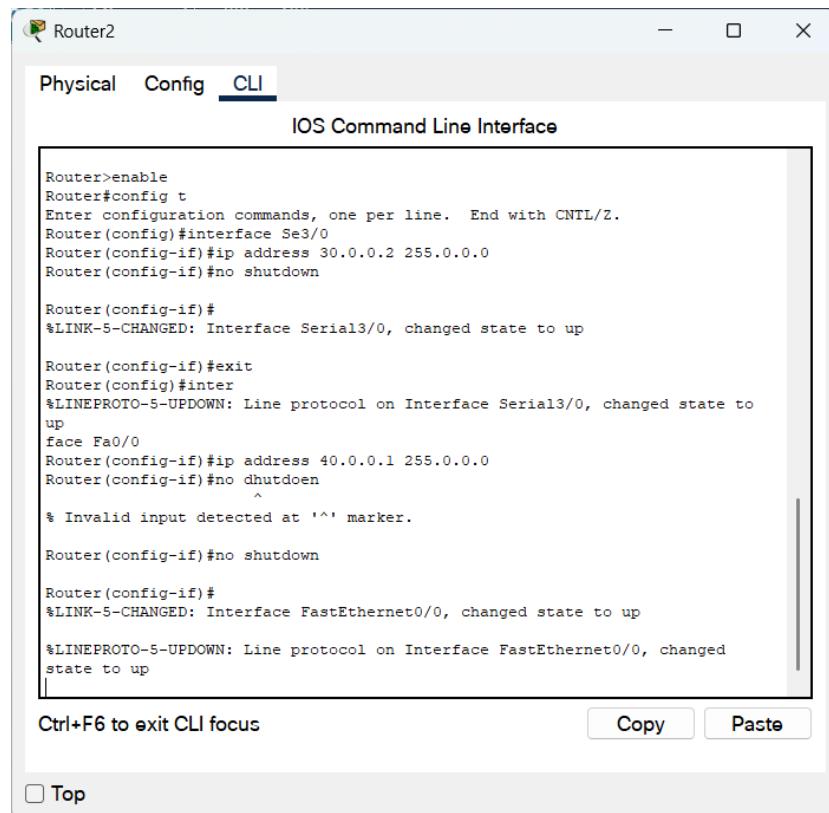
Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#no shutdown

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

Ctrl+F6 to exit CLI focus
```

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

Router2 configuration



The screenshot shows the configuration window for Router2. The title bar says "Router2". 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 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 dhtddo
      ^
% Invalid input detected at '^' marker.

Router(config-if)#no shutdown

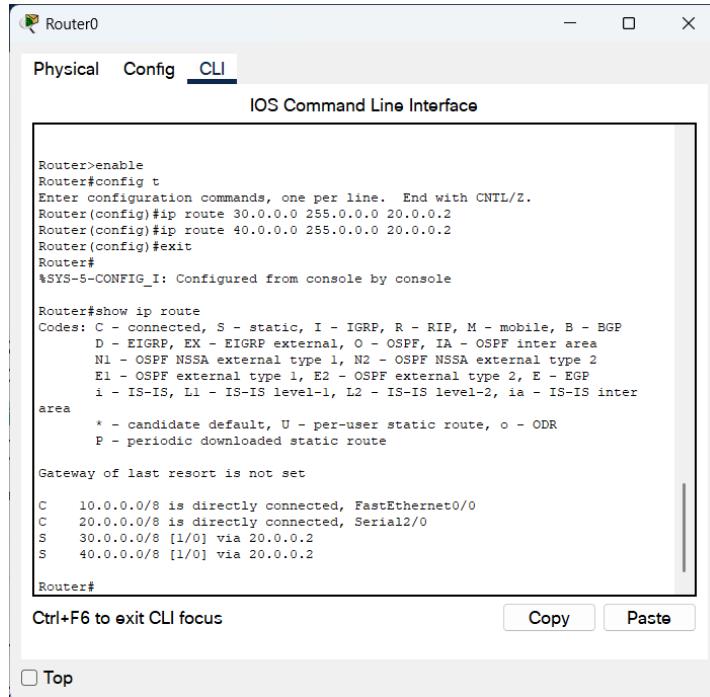
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

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

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

Static Routing:

Router0



Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 30.0.0.0 255.0.0.0 20.0.0.2
Router(config)#ip route 40.0.0.0 255.0.0.0 20.0.0.2
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

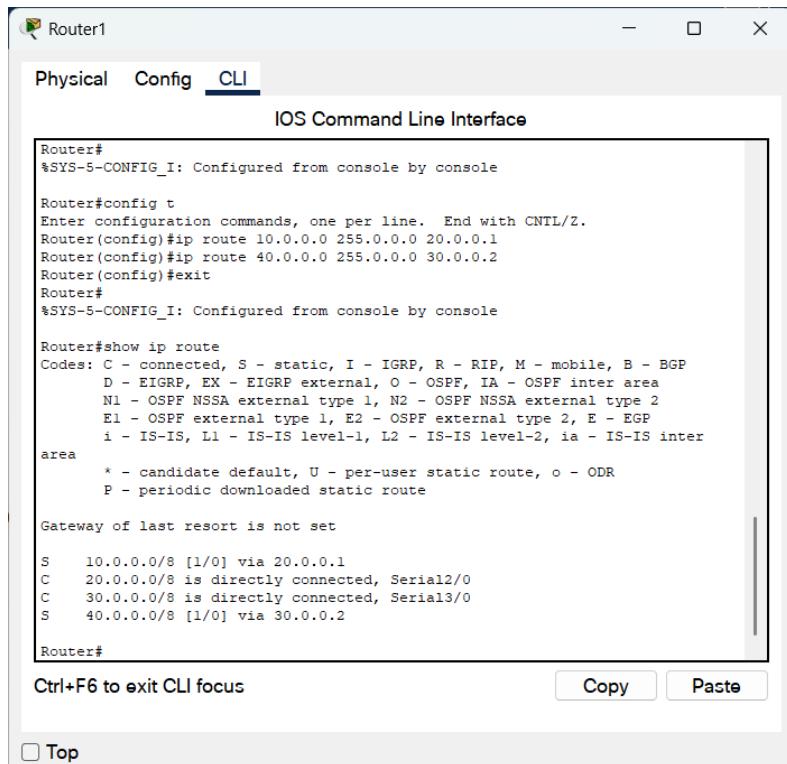
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS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

Gateway of last resort is not set

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

Router#

Router1



Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 10.0.0.0 255.0.0.0 20.0.0.1
Router(config)#ip route 40.0.0.0 255.0.0.0 30.0.0.2
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

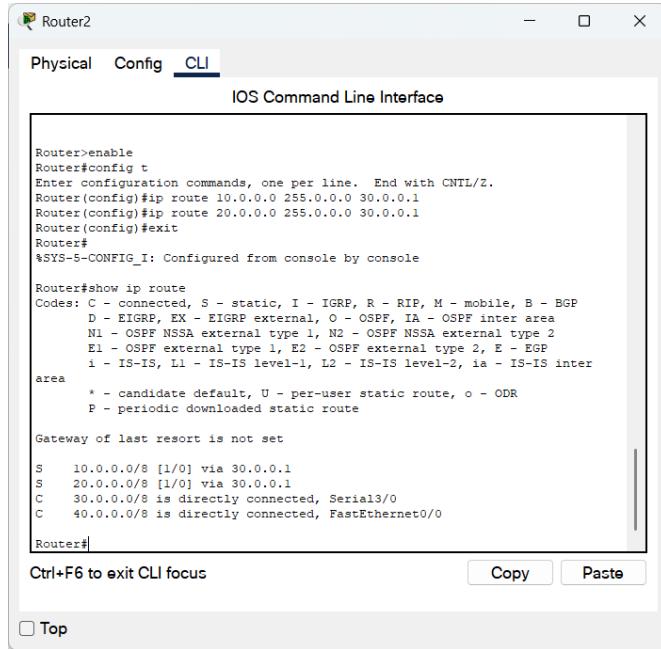
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS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

Gateway of last resort is not set

S 10.0.0.0/8 [1/0] via 20.0.0.1
C 20.0.0.0/8 is directly connected, Serial2/0
C 30.0.0.0/8 is directly connected, Serial3/0
S 40.0.0.0/8 [1/0] via 30.0.0.2

Router#

Router2



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

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

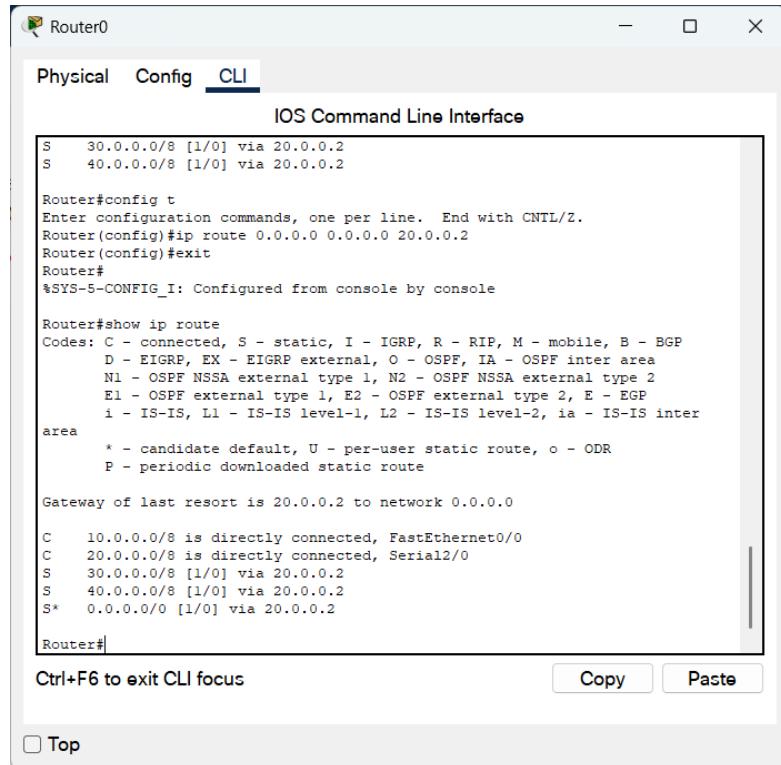
Gateway of last resort is not set

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

Router#
```

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

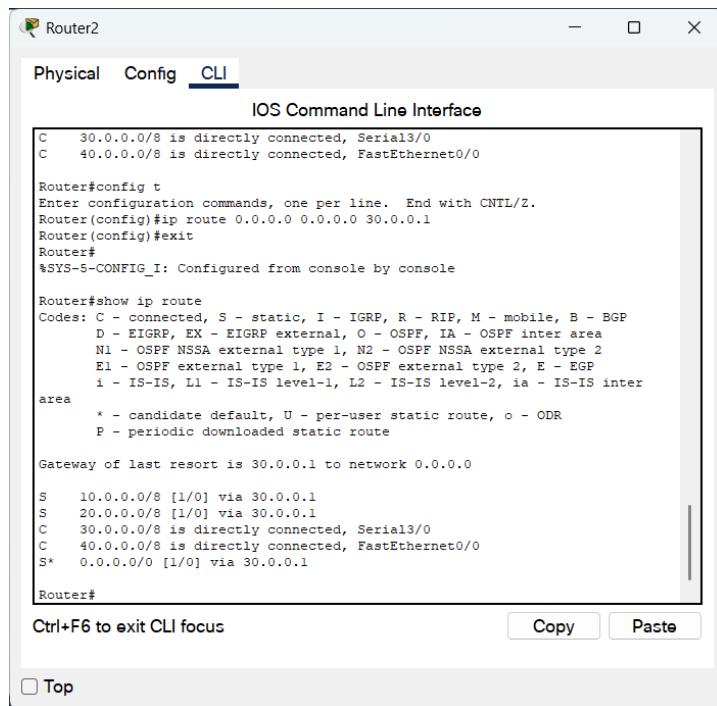
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      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#
```

Router2



The screenshot shows the Router2 CLI interface. The title bar says "Router2". The tabs at the top are "Physical", "Config", and "CLI", with "CLI" being the active tab. The main window title is "IOS Command Line Interface". The command-line area contains the following text:

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

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

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

Gateway of last resort is 30.0.0.1 to network 0.0.0.0

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

Router#
```

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

Pinging:

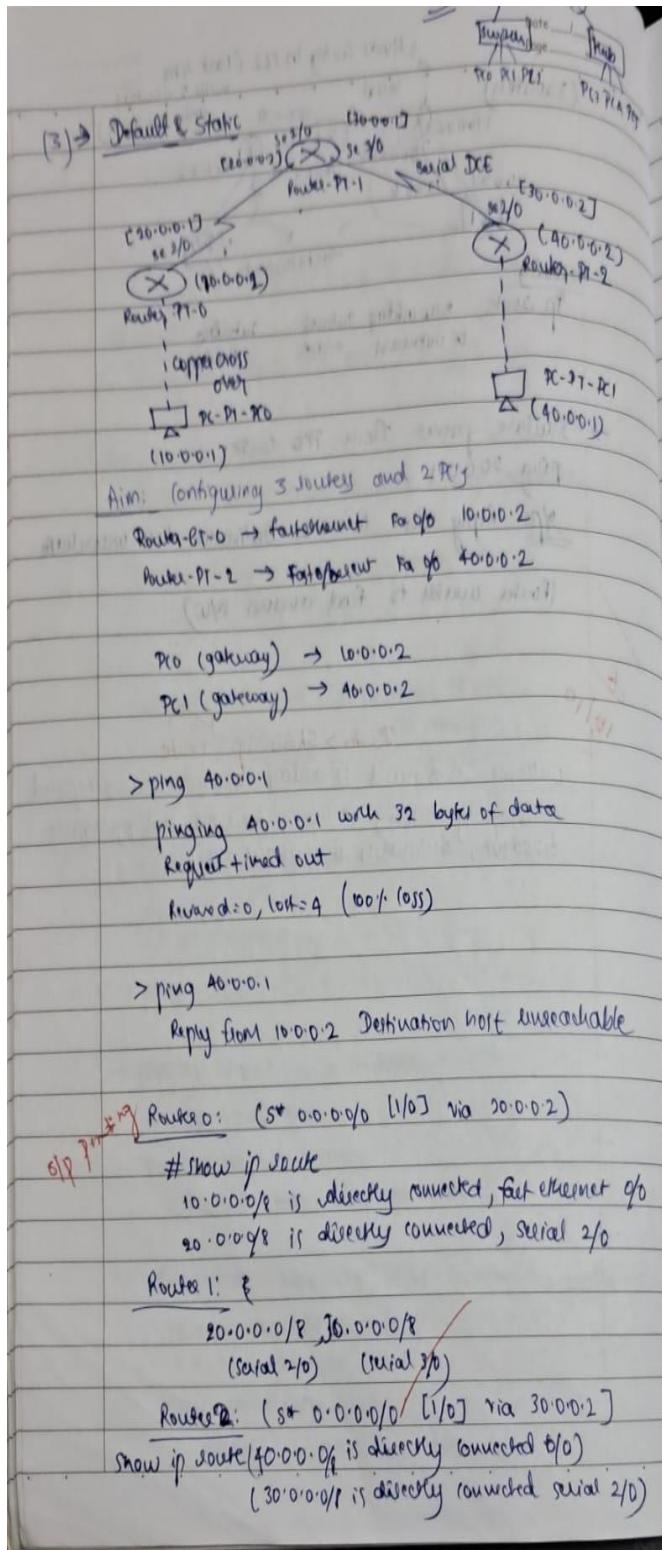
```
C:\>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

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

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

iv. Observation



Date / /
 Page /

Router 0:

```
Router(config)# ip route 0.0.0.0 0.0.0.0 20.0.0.2
Router(config)# ip route 0.0.0.0 0.0.0.0 30.0.0.1
```

Router 1:

```
S 10.0.0.0/8 [1/0] via 20.0.0.1
S 40.0.0.0/8 [1/0] via 30.0.0.2
```

*
 pc> ping 10.0.0.1
 pinging 10.0.0.1 with 32 bytes of data
 Reply from 10.0.0.1: bytes=32 time=15 ms TTL=155

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

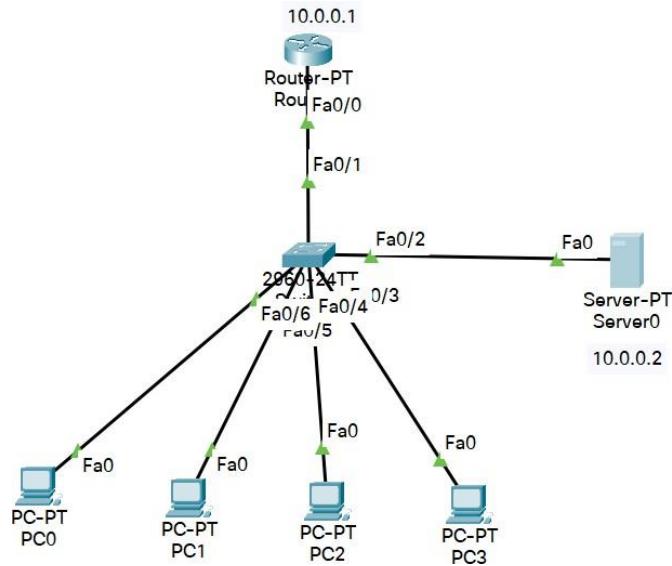
8/11

1) Request timed out
 2) Destination un

Outcome: Successfully configured default and static routes on routers, enabling proper routing of packets between different network segments.

Program 4

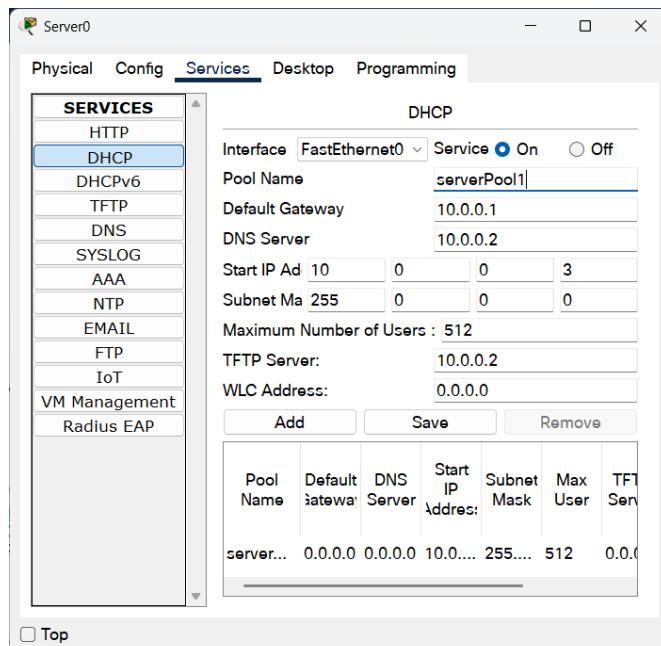
- i. Configure DHCP within a LAN and outside LAN.
- ii. Procedure along with the topology



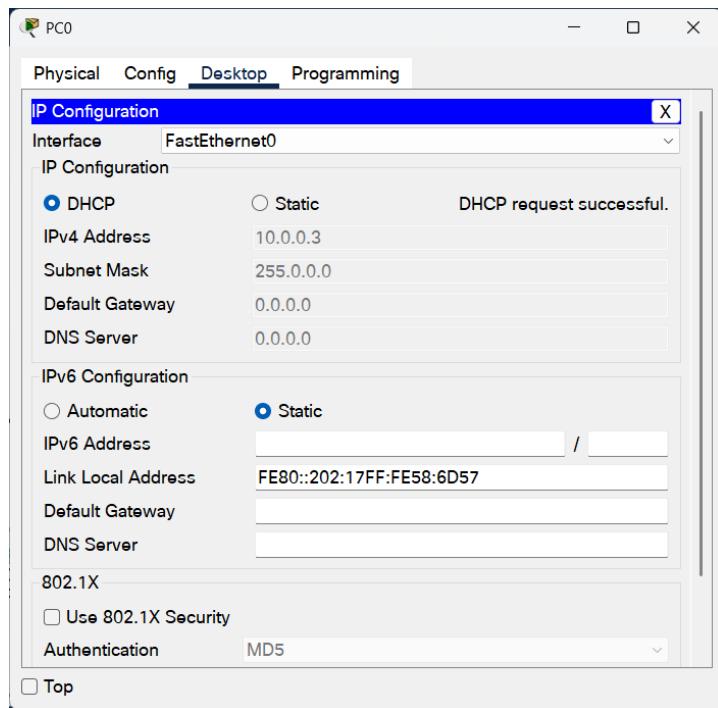
- iii. Screen shots/ output

DHCP Within LAN

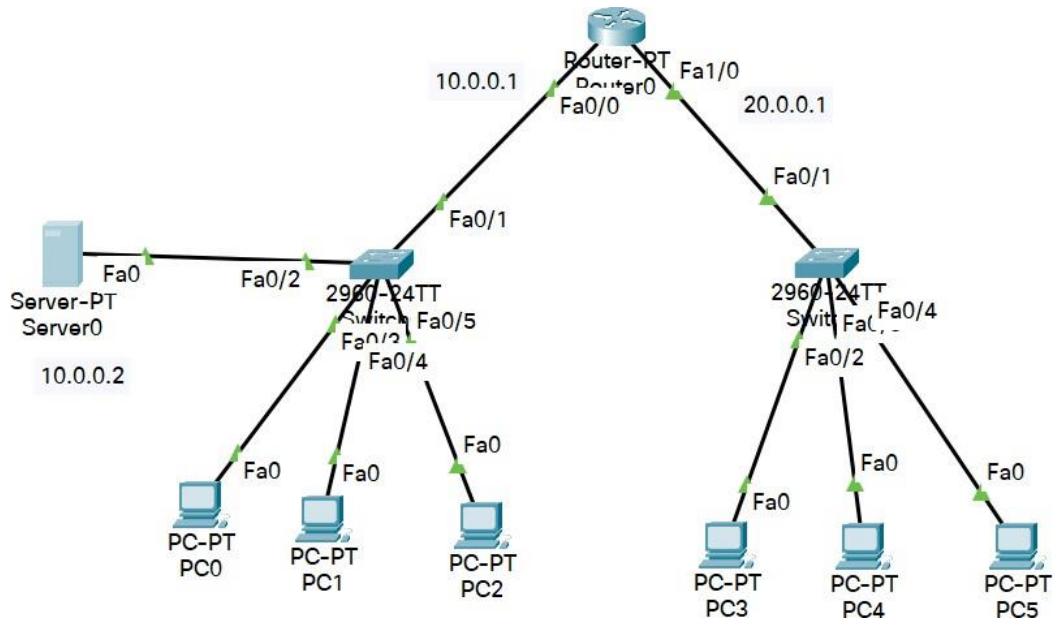
DHCP Configuration



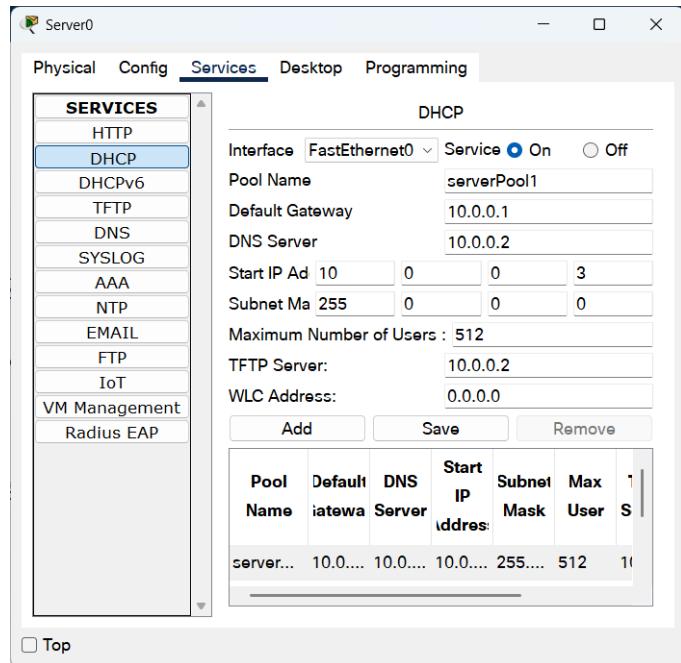
PC settings



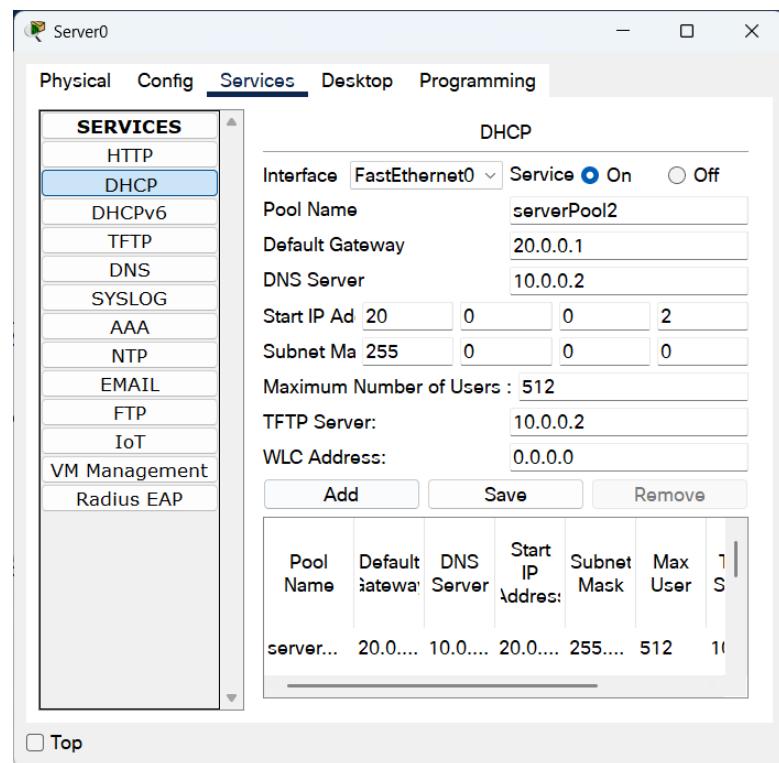
DHCP outside LAN:



DHCP configuration for inside LAN



DHCP configuration for outside LAN

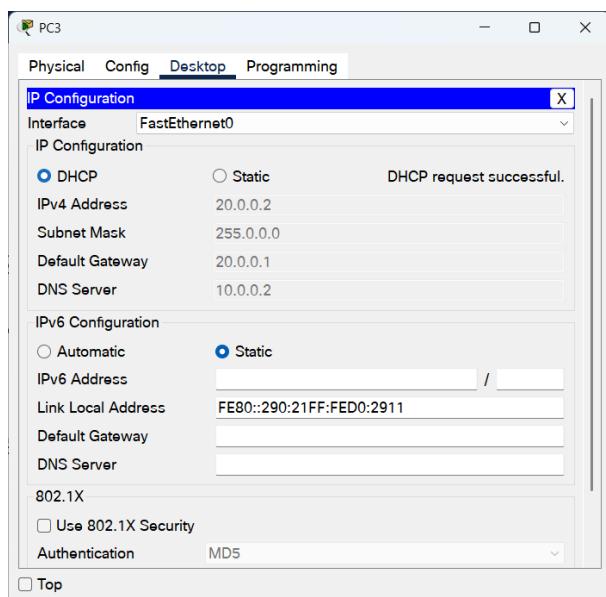


Ip helper command in Router

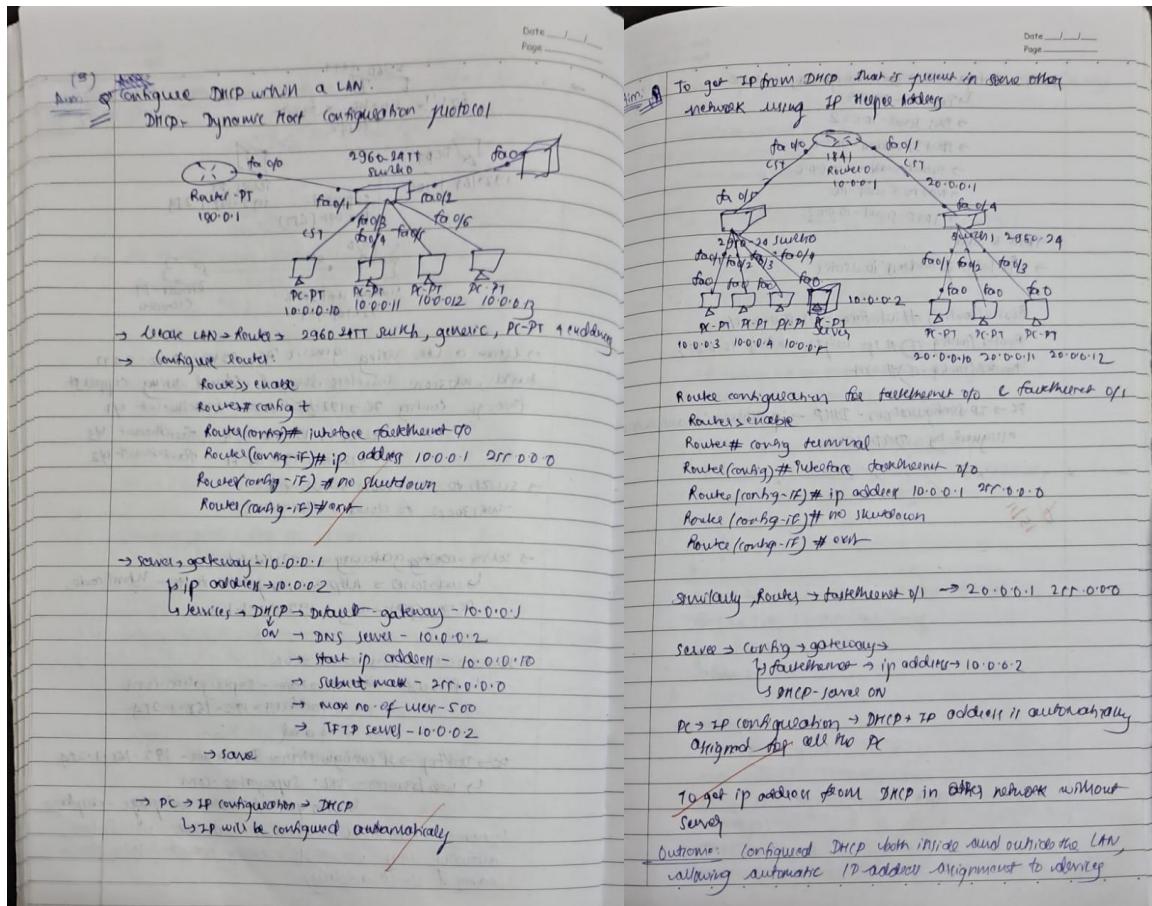
```
Router(config-if)#exit
Router(config)#interface Fa1/0
Router(config-if)#ip helper-address 10.0.0.2
Router(config-if)#

```

PC setting in another network

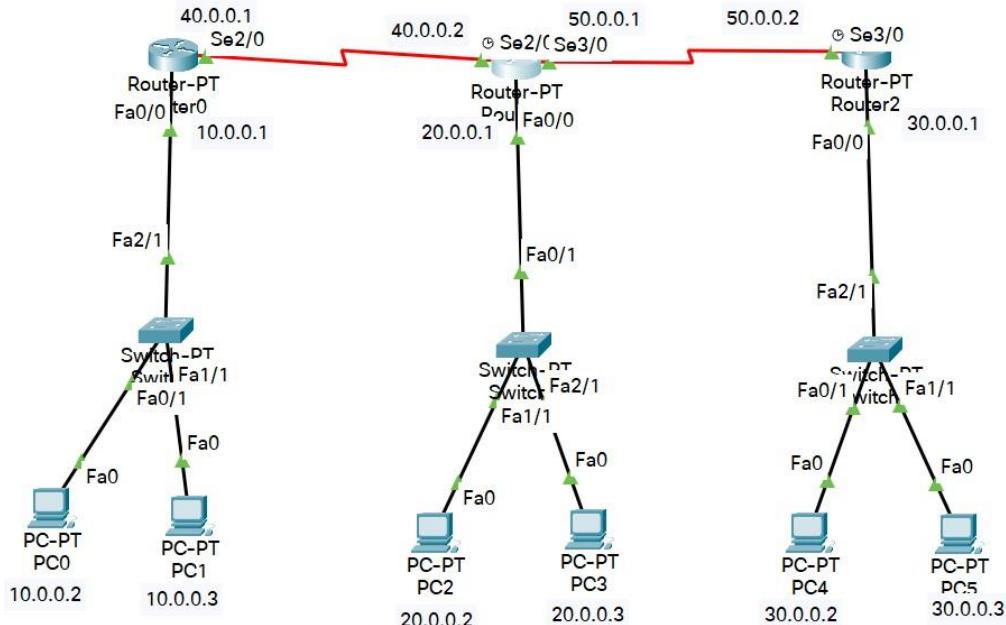


iv. Observation



Program 5

- i. Configure RIP routing Protocol in Routers
- ii. Procedure along with the topology



- iii. Screen shots/ output

Router0

```
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 10.0.0.0
Router(config-router)#network 40.0.0.0
Router(config-router)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

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

Gateway of last resort is not set

C    10.0.0.0/8 is directly connected, FastEthernet0/0
C    40.0.0.0/8 is directly connected, Serial2/0
```

Router1

```
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 40.0.0.0
Router(config-router)#network 50.0.0.0
Router(config-router)#network 20.0.0.0
Router(config-router)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

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

Gateway of last resort is not set

R    10.0.0.0/8 [120/1] via 40.0.0.1, 00:00:08, Serial2/0
C    20.0.0.0/8 is directly connected, FastEthernet0/0
R    30.0.0.0/8 [120/1] via 50.0.0.2, 00:00:10, Serial3/0
C    40.0.0.0/8 is directly connected, Serial2/0
C    50.0.0.0/8 is directly connected, Serial3/0
```

Router2

```
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 30.0.0.0
Router(config-router)#network 50.0.0.0
Router(config-router)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

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

Gateway of last resort is not set

R    10.0.0.0/8 [120/2] via 50.0.0.1, 00:00:28, Serial3/0
R    20.0.0.0/8 [120/1] via 50.0.0.1, 00:00:28, Serial3/0
C    30.0.0.0/8 is directly connected, FastEthernet0/0
R    40.0.0.0/8 [120/1] via 50.0.0.1, 00:00:28, Serial3/0
C    50.0.0.0/8 is directly connected, Serial3/0
```

Pinging:

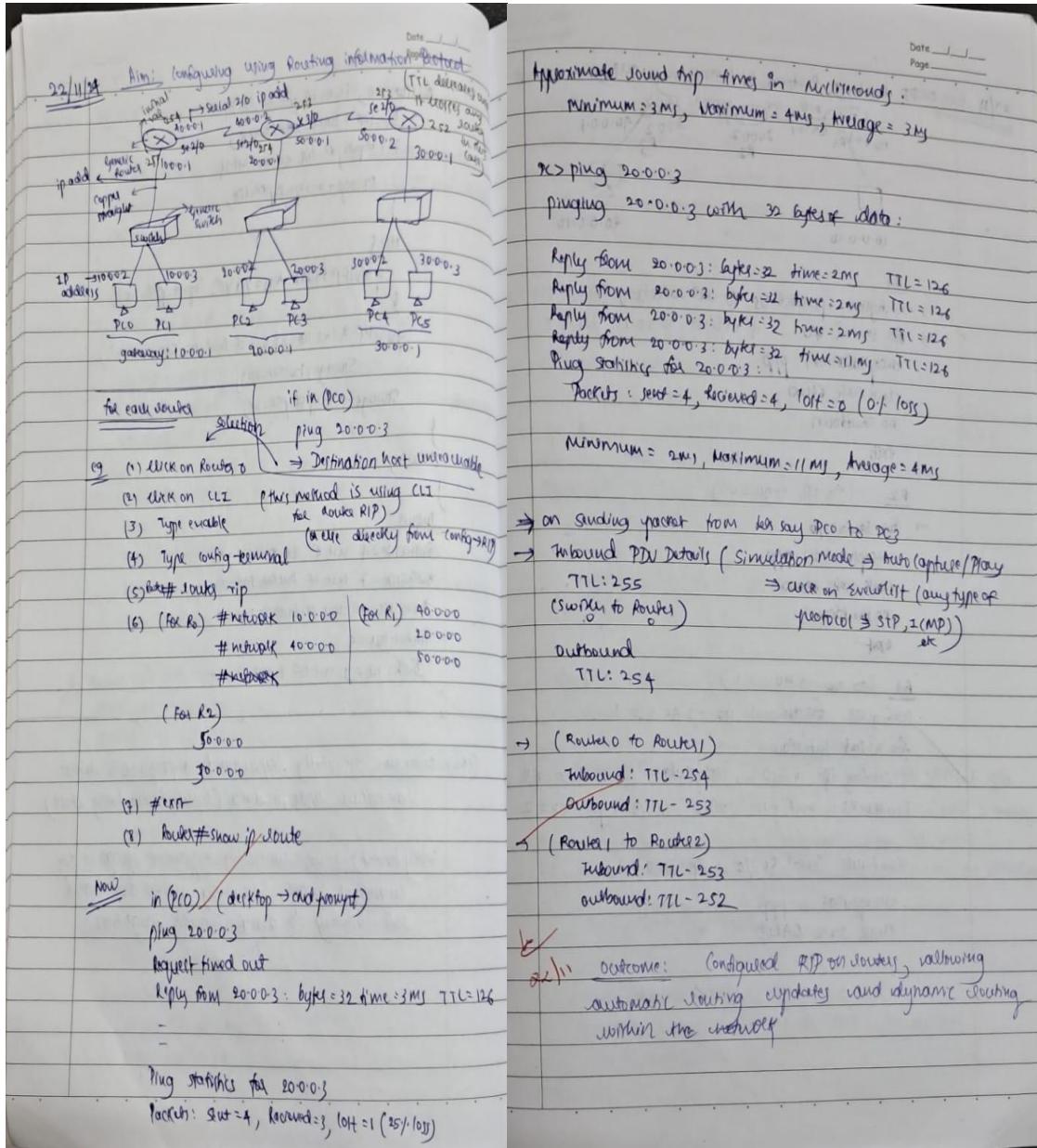
```
Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.2

Pinging 20.0.0.2 with 32 bytes of data:

Request timed out.
Reply from 20.0.0.2: bytes=32 time=9ms TTL=126
Reply from 20.0.0.2: bytes=32 time=1ms TTL=126
Reply from 20.0.0.2: bytes=32 time=9ms TTL=126

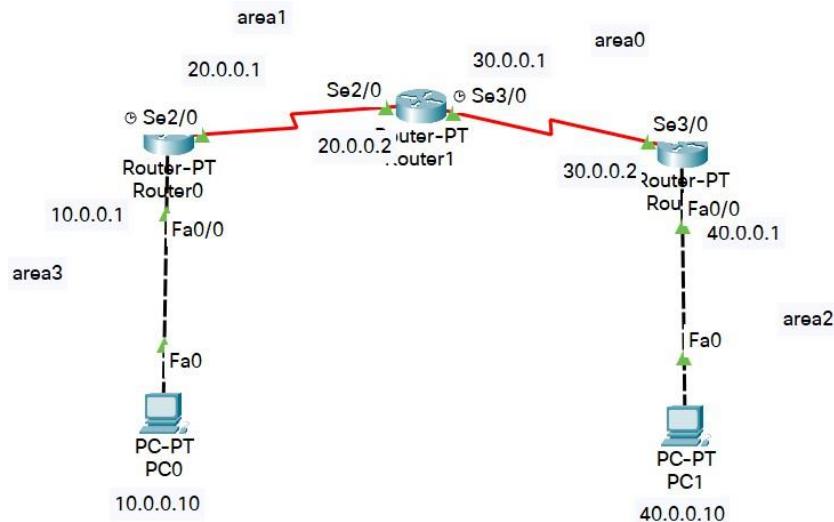
Ping statistics for 20.0.0.2:
  Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
  Minimum = 1ms, Maximum = 9ms, Average = 6ms
```

iv. Observation



Program 6

- i. Configure OSPF routing protocol
- ii. Procedure along with the topology



- iii. Screen shots/ output

Encapsulation:

Router0

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

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

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

Router(config-if)#exit
Router(config)#interface Se2/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#clock rate 64000
Router(config-if)#no shutdown

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

```

Router1

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

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

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

Router(config-if)#exit
Router(config)#interface Se3/0
Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#clock rate 64000
Router(config-if)#no dhutdown
^
% Invalid input detected at '^' marker.

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

Router2

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

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

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

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

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

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

OSPF Routing Protocol

Router0

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

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

Gateway of last resort is not set

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

Router1

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

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

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

Gateway of last resort is not set

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

Router2

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

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

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

Gateway of last resort is not set

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

Configure Loopback address

Router0

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

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

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

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

Router1

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

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

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

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

```

Router2

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

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

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

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

Create Virtual Link

Router0

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

Router1

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

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

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

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

Pinging

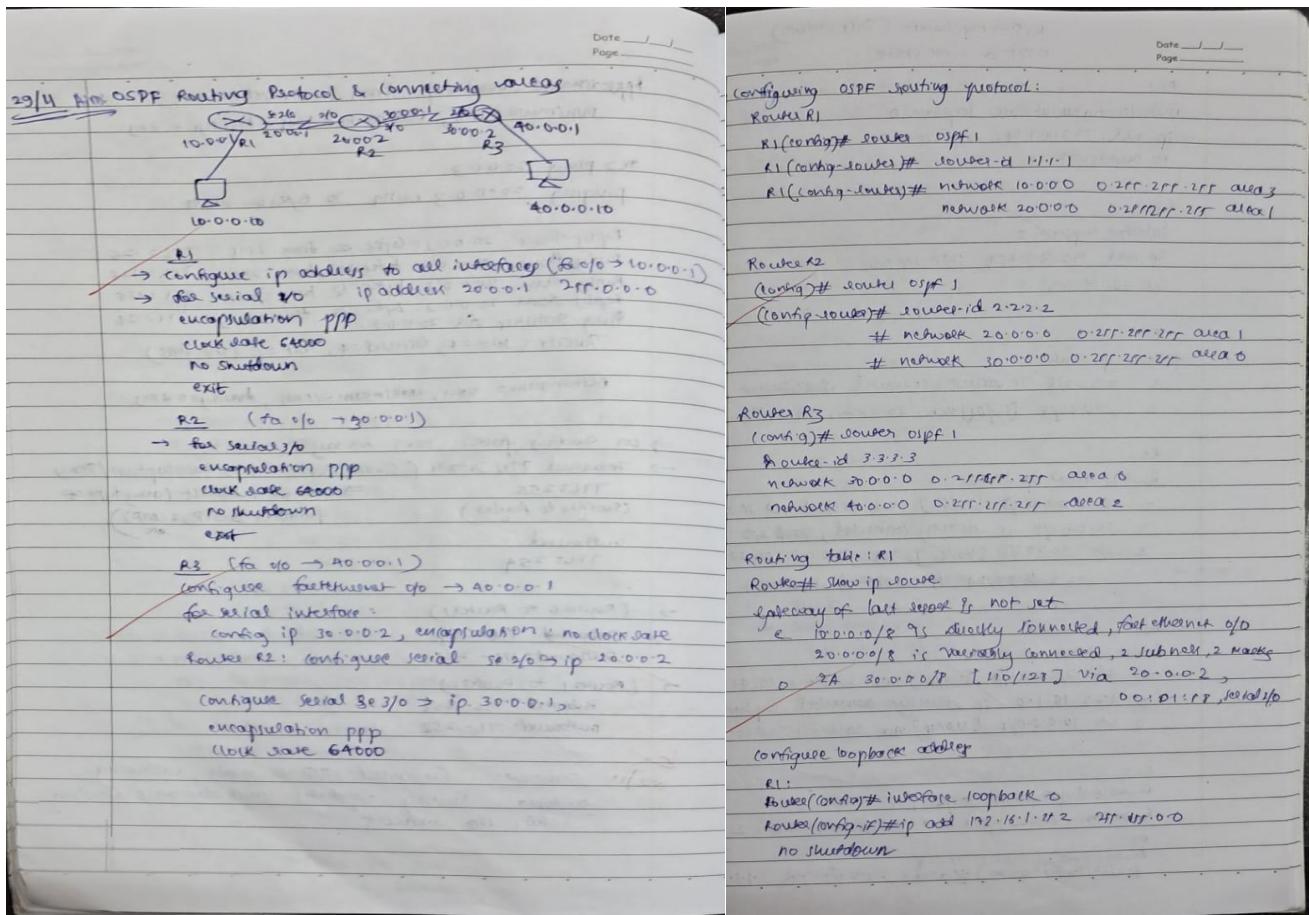
```
C:\>ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data:

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

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

iv. Observation



RIP → Hop count (Diff vector)

OSPF → Link state

Date / /
Page / /

R2:

Router(config)# interface loopback 0
ip add 172.16.1.253 255.255.0.0
no shutdown

R3:

interface loopback 0
ip add 172.16.1.254 255.255.0.0
no shutdown

R1:

show ip route
c 10.0.0.0/8 is directly connected, FastEthernet 0/
o 30.0.0.0/8 [110/0] via 20.0.0.2, 00:11:11 serial 2/0

R2:

Router> show ip route
o 1A 10.0.0.0/8 [110/65] via 20.0.0.1, 00:27:28, serial 2/0
c 20.0.0.0/8 is directly connected, serial 2/0
o 2A 40.0.0.0/8 [110/65] via 30.0.0.2, 01:27:47,
172.16.0.0/24 is subnetted, 1 subnet serial 3/0

R3:

Router> show ip route
o 2A 20.0.0.0/8 [110/21] via 30.0.0.1, 00:48:20, serial
c 172.16.1.0 is directly connected, loopback 0 2/0
o 2A 10.0.0.0/8 [110/12] via 30.0.0.1 00:00:09 serial 2/0

R1:

Router(config)# Router ospf 1

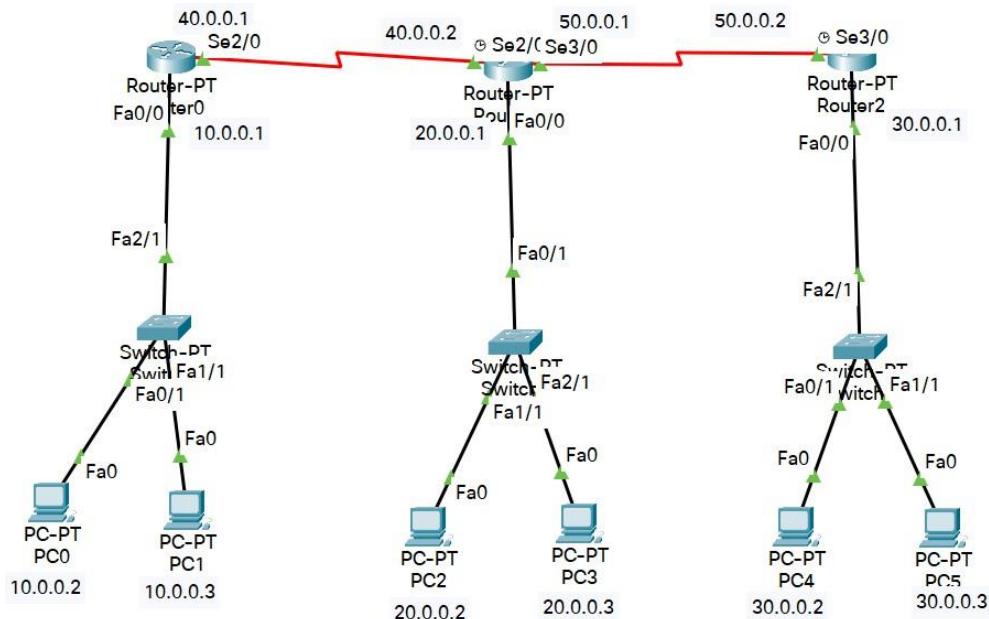
Router(config-router)# area 1 virtual-link 2.2.2.2

R2:

Router(config)# Router ospf 1
Router(config-router)# area 1 virtual-link 1.1.1.1

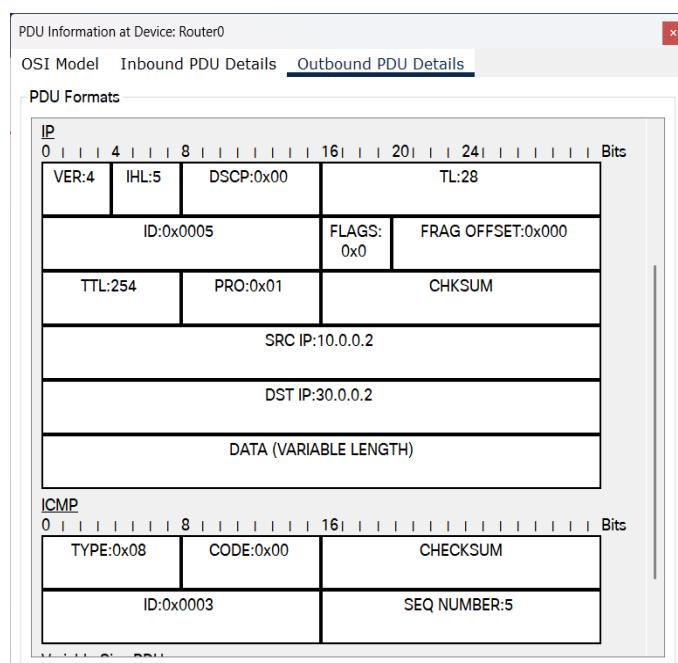
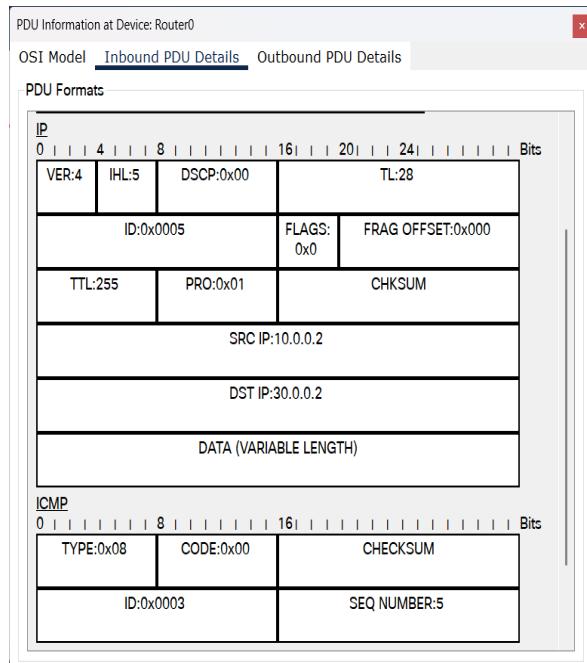
Program 7

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

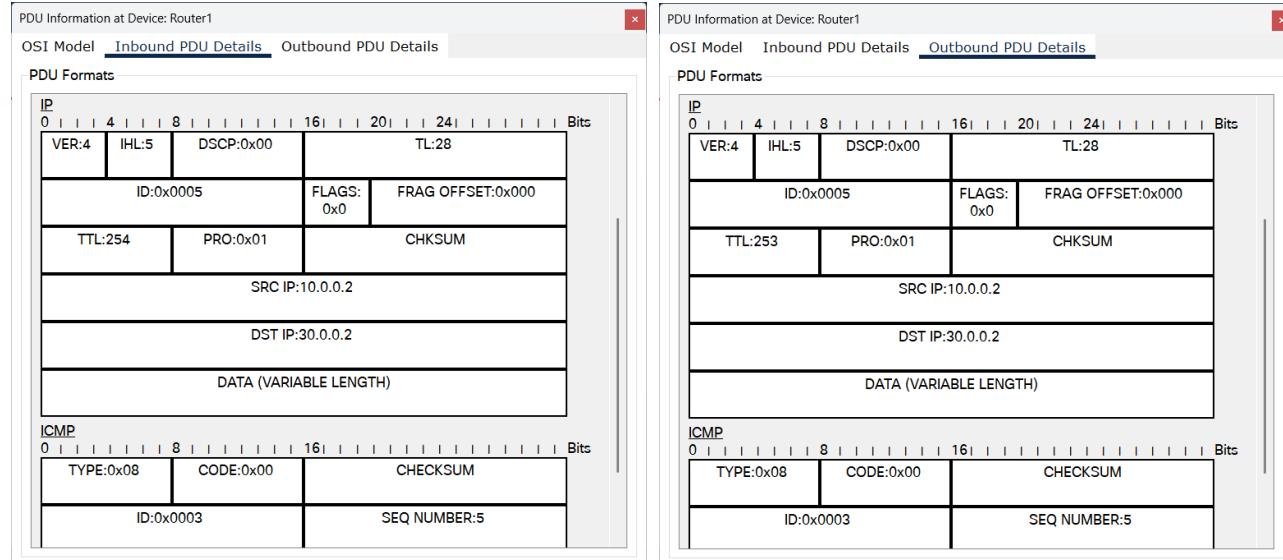


- Screen shots/ output

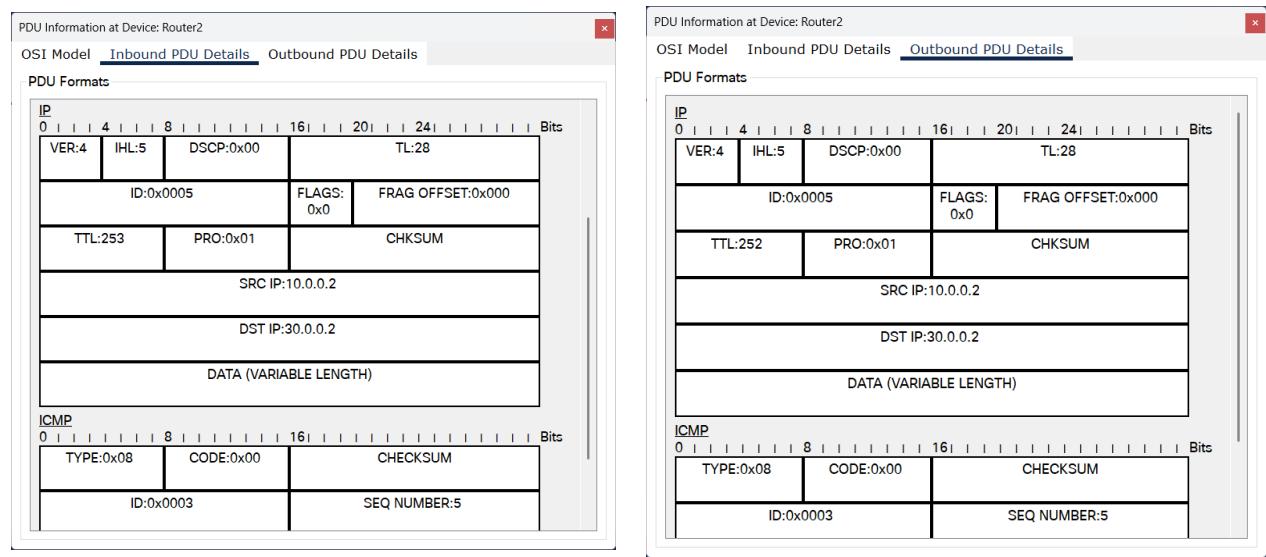
Packet at Router0



Packet at Router1

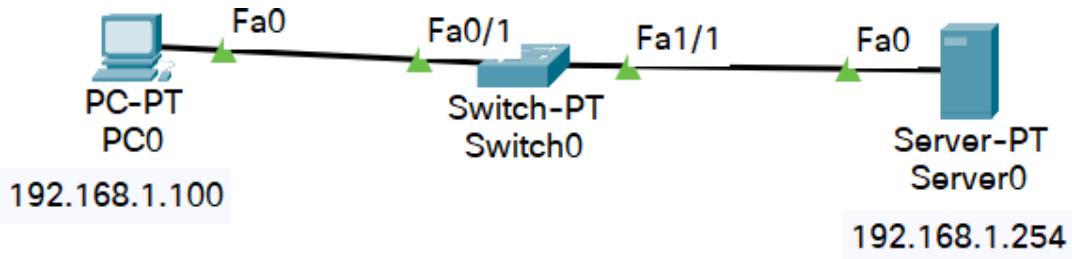


Packet at Router2



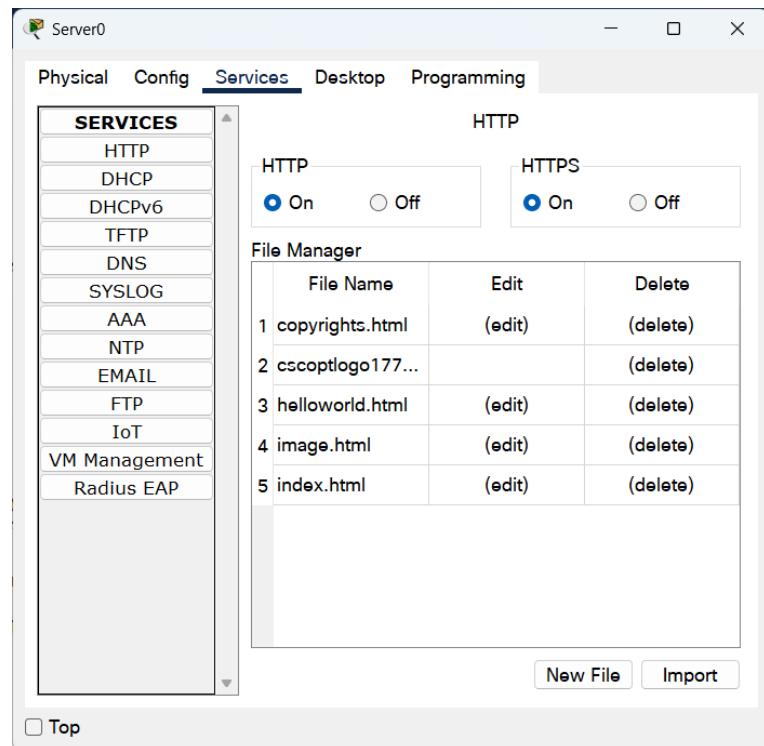
Program 8

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

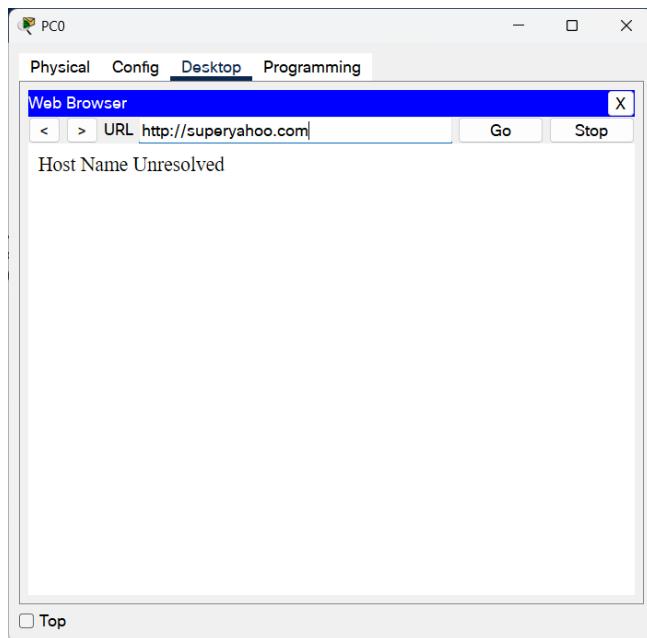
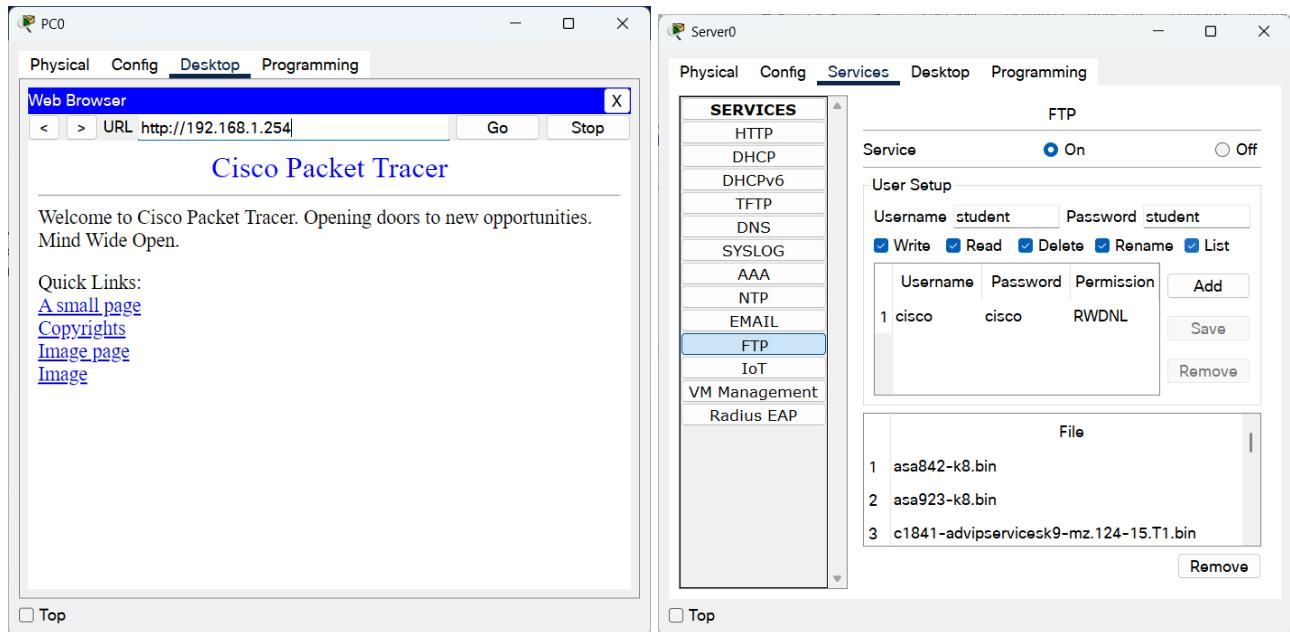


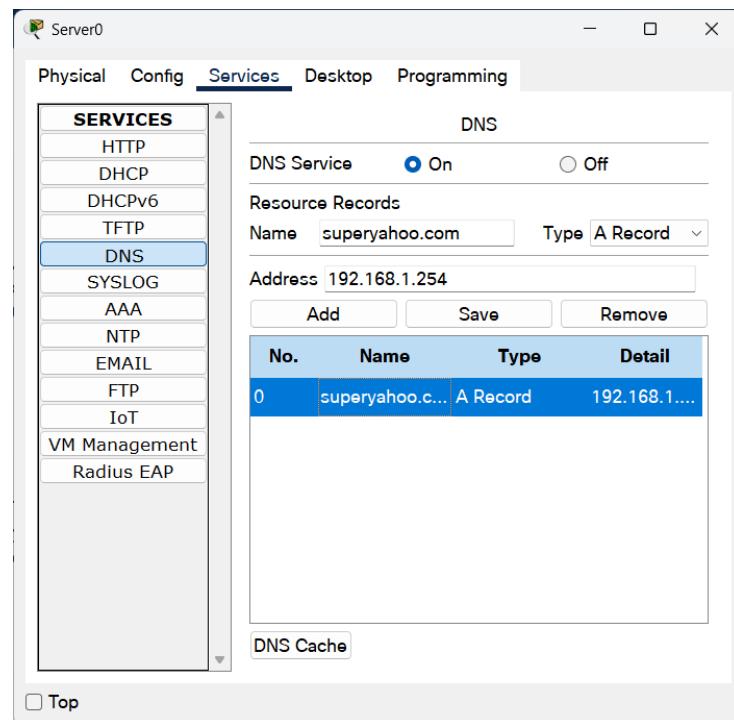
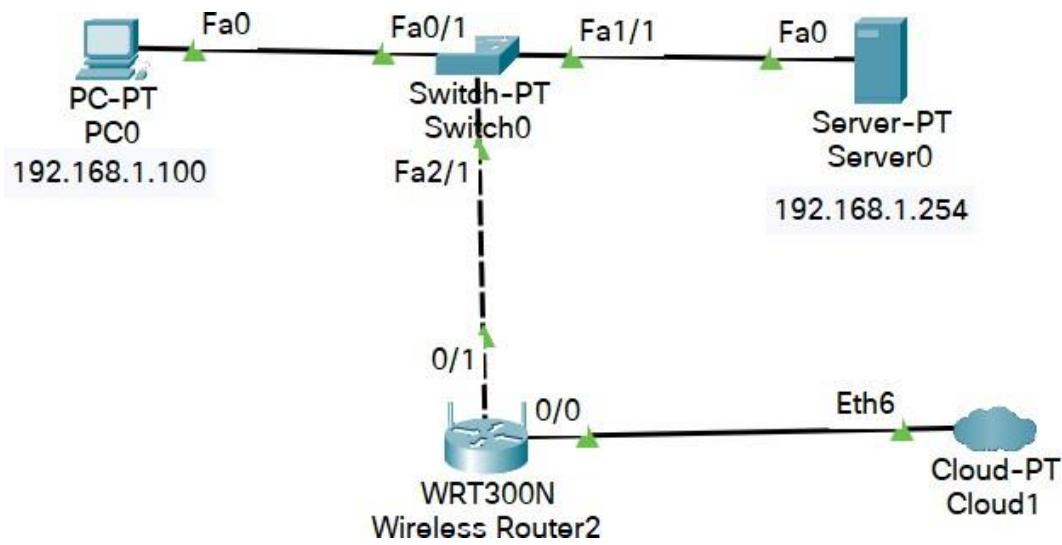
- iii. Screen shots/ output

Server's services

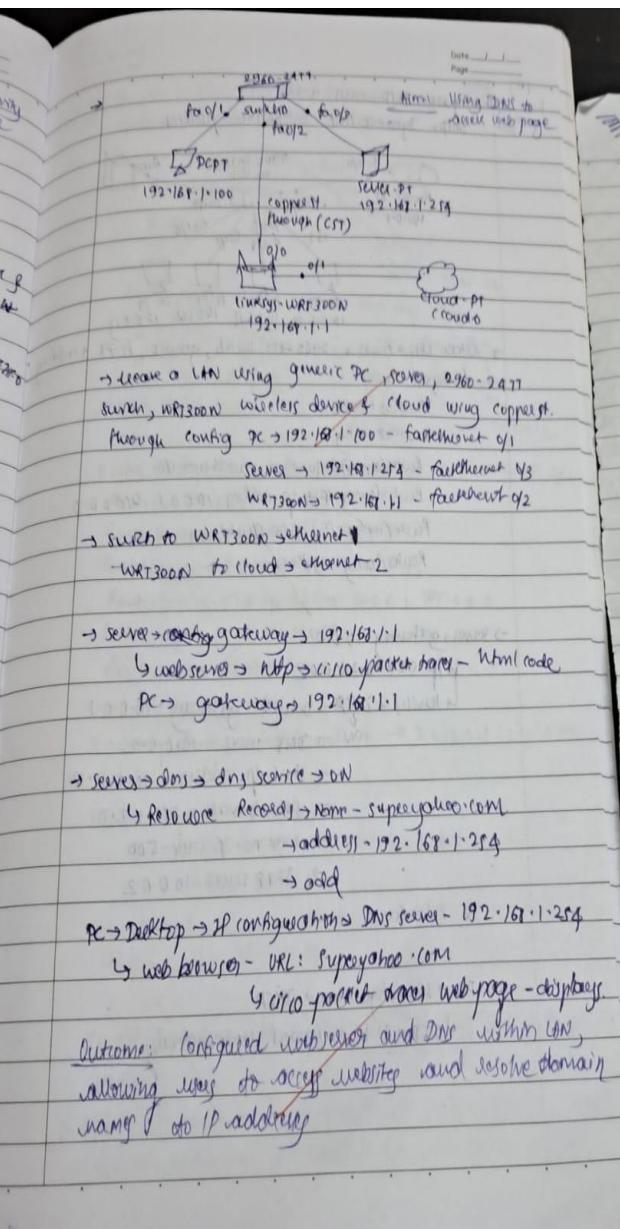
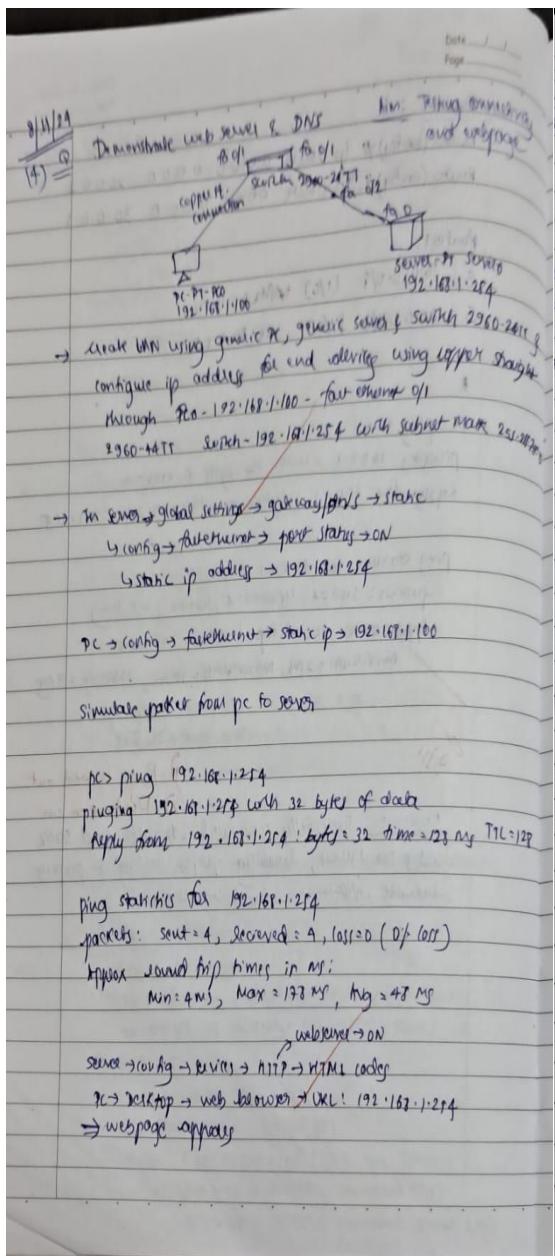


PC's Web Browser



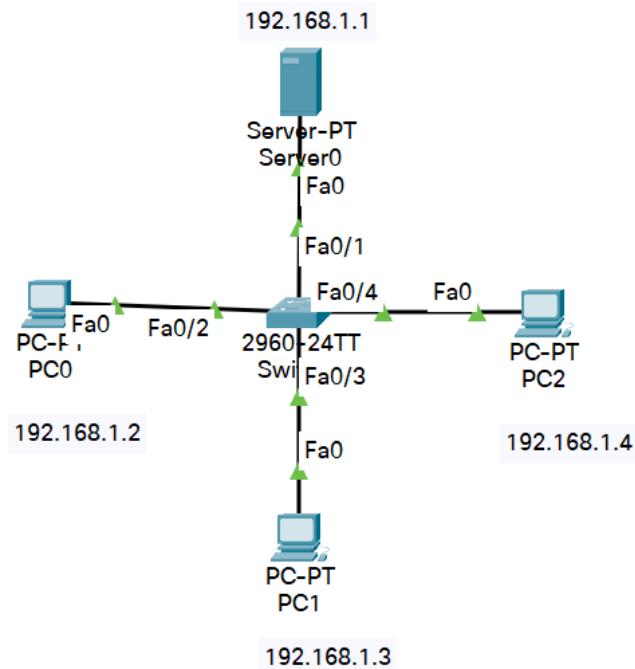


iv. Observation



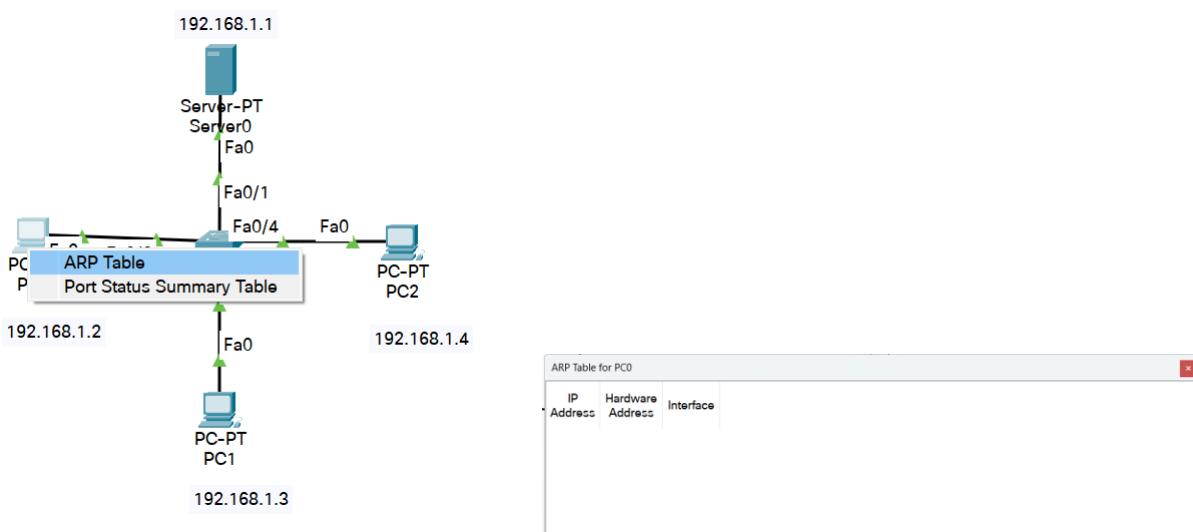
Program 9

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

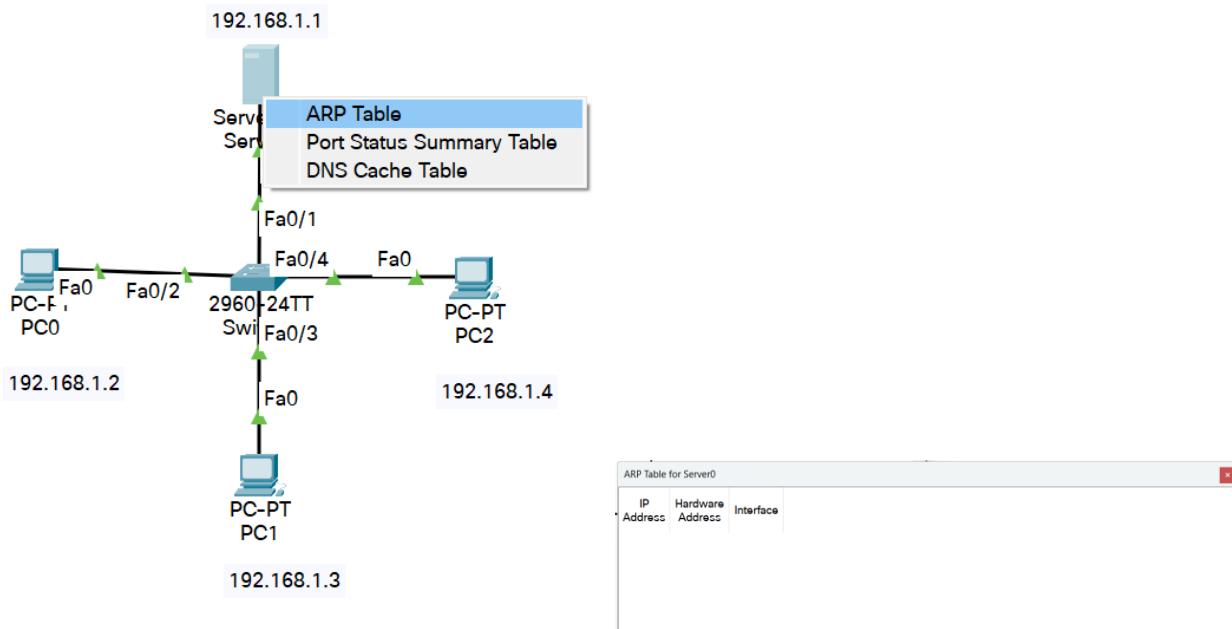


- iii. Screen shots/ output

ARP Table of PC



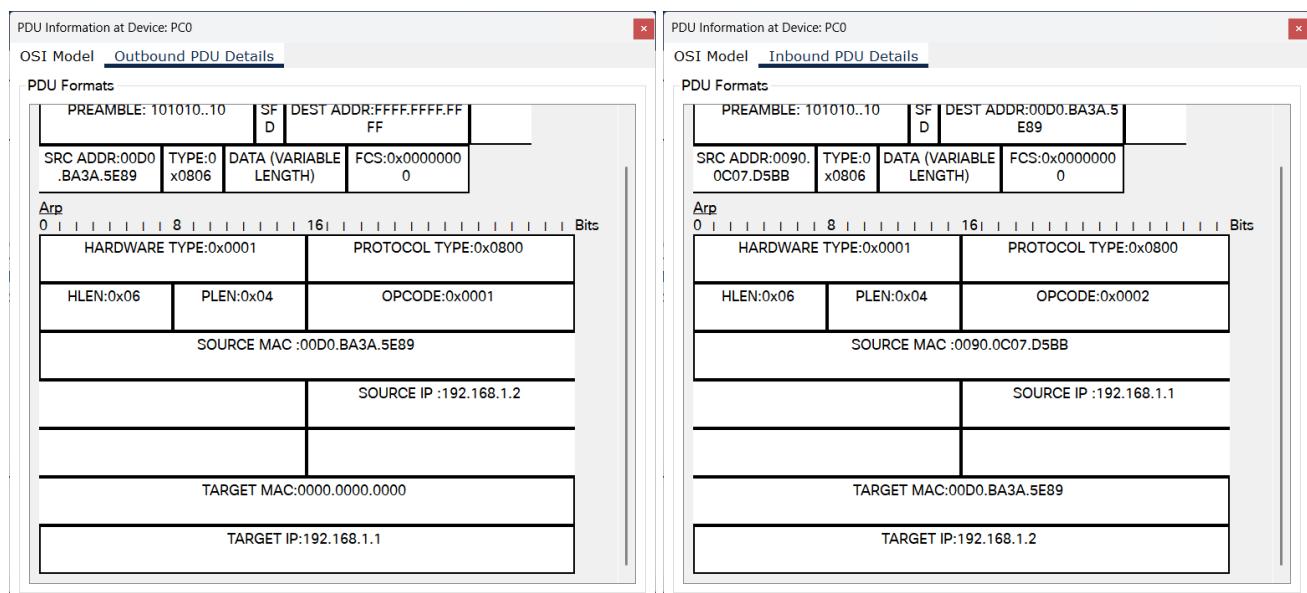
ARP Table of Server



Command at PC

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

Pinging in Simulation Mode



Observation

Date _____
Page _____

Q) ARP (Address Resolution Protocol) Aim: To construct simple LAN and understand concept of ARP

→ Simulation panel → inspect → PC1 (Right click)

- ↳ ARP-table (empty)
- ↳ Server-ARP-table (empty)

→ Ping from PC1 to Server

```
>arp-a
No entries found
```

>ping 192.168.11.4

Reply from 192.168.11.4: bytes=32, time=8ms, TTL=128
 packets: sent=4, received=4, lost=0 (0% loss)
 minimum=4ms, maximum=8ms, average=6ms

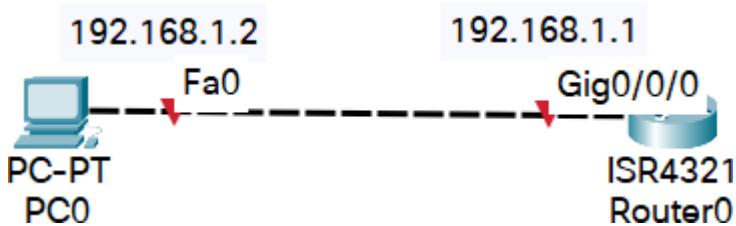
→ ARP table shows the IP address of PC1 and then PC1 delivers the acknowledgement, if ARP table shows IP address of server

→ Router → 192.168.1.1

Outcome: Built a simple LAN and demonstrated ARP operation, carries map IP addresses to MAC addresses for communication

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

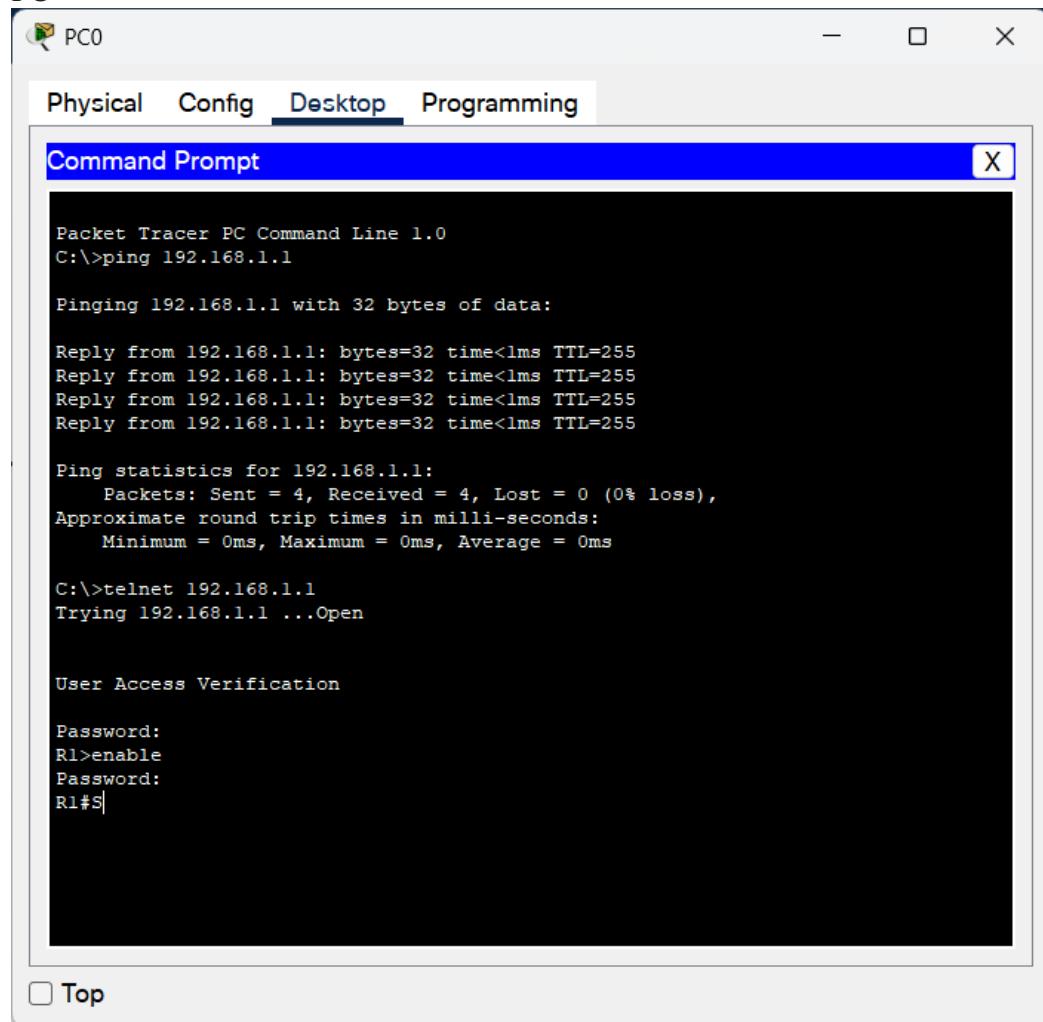
R1(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to up

%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

Date _____
Page _____

CLI

Router > enable
Router# config t
Router (config)# hostname R1
enable secret 3p
interface Fa0/0
ip address 192.168.1.1 255.255.255.0
no shutdown
line vty 0 5
Router (config-line)# login
password tp
Router# exit
configured from console by console
R1#exit
Building configuration...
[OK]

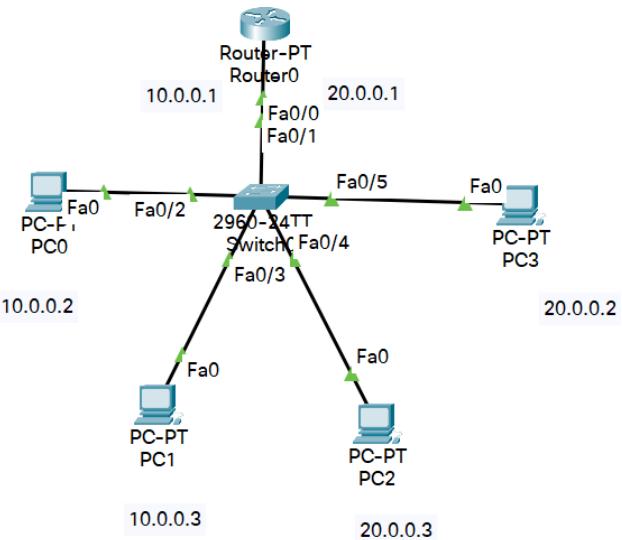
> ping 192.168.1.1
Reply from 192.168.1.1: bytes=32 time=1ms TTL=255
Ping statistics for 192.168.1.1:
Packets: sent=4, Received=4, lost=0 (0% loss)
minimum=0ms, maximum=1ms, Average=0ms

PC> telnet 192.168.1.1
trying 192.168.1.1 ... open
User Access Verification
Password: tp
Router>en
Password: tp

Outcome: Successfully accessed the router via serial port using TELNET from PC demonstrating remote management.

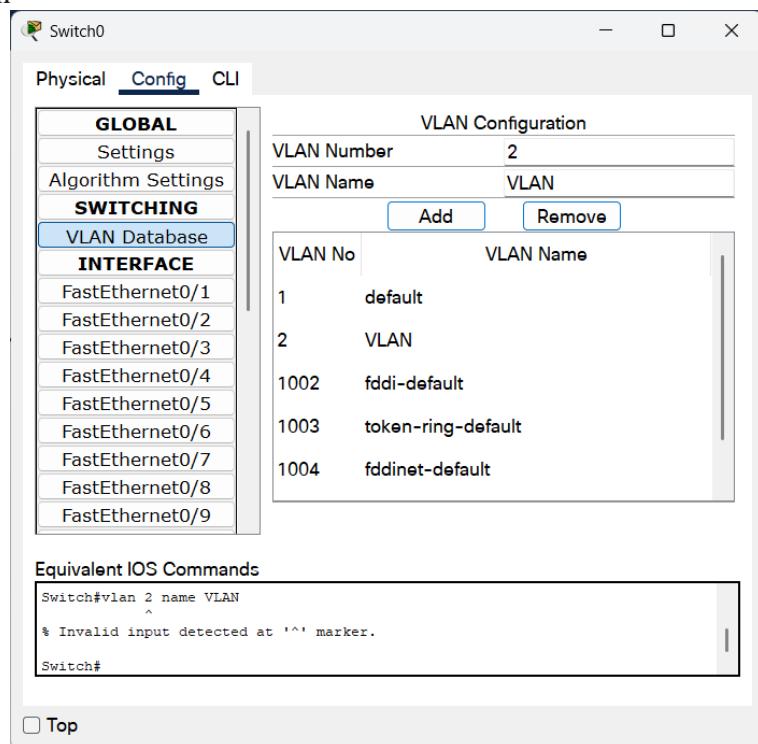
Program 11

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

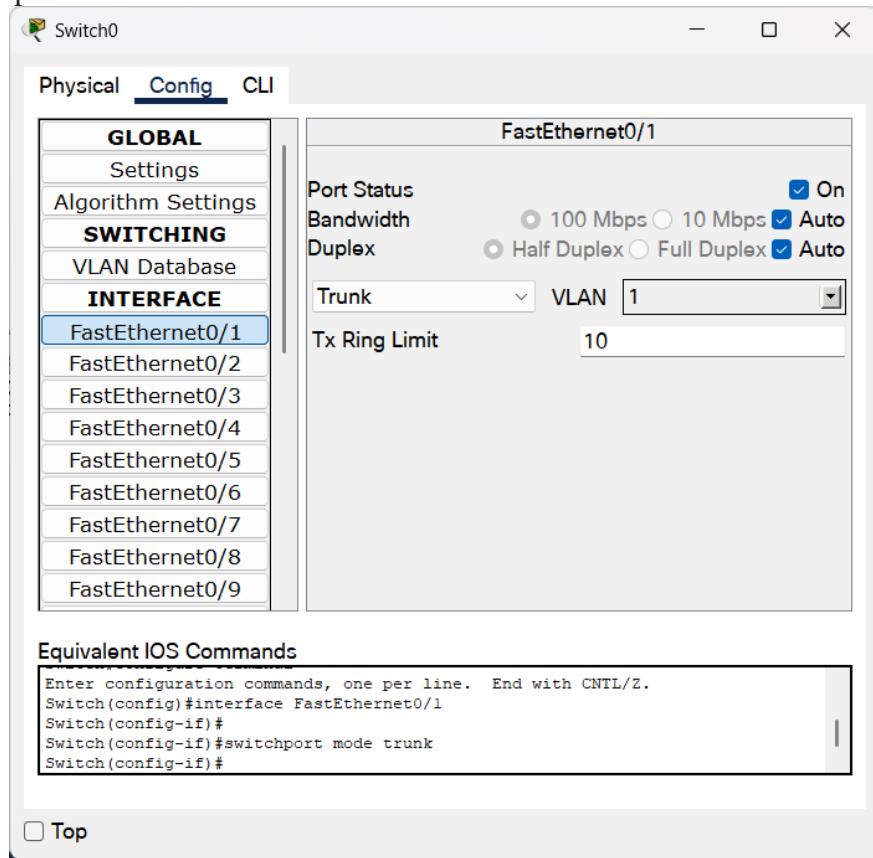


- iii. Screen shots/ output

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 "Port Status" is "On" (checked). Bandwidth is set to "100 Mbps" (radio button). Duplex is set to "Full Duplex" (radio button) with "Auto" checked. 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 "Port Status" is "On" (checked). Bandwidth is set to "100 Mbps" (radio button). Duplex is set to "Full Duplex" (radio button) with "Auto" checked. 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 for FastEthernet0/5 shows three entries: "1:default", "2:VLAN" (which is checked), and "1002:fddi-default".

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

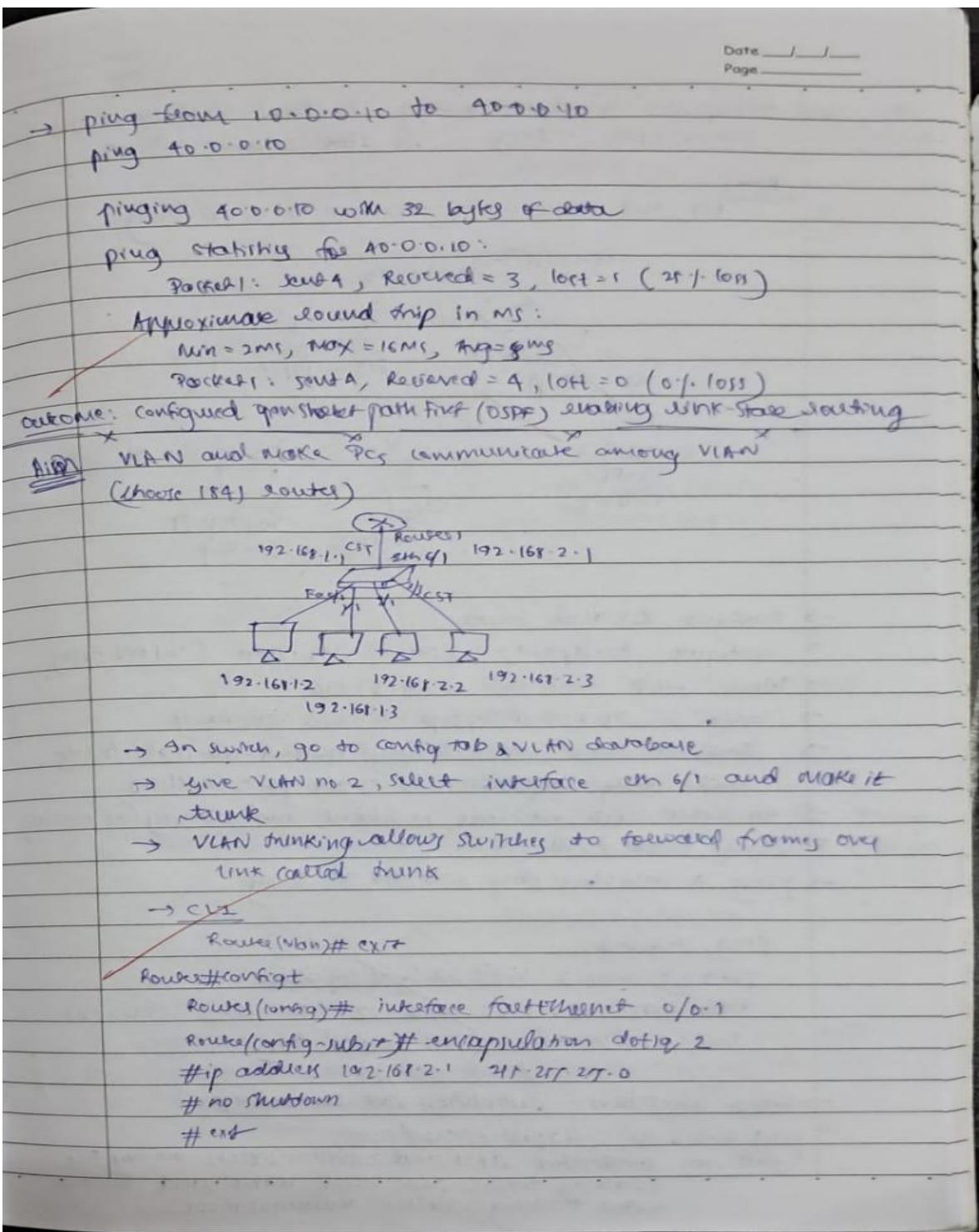
Router(config-subif)#encapsulation dot1q 2
Router(config-subif)#ip address 20.0.0.1 255.0.0.0
Router(config-subif)#no shutdown
Router(config-subif)#exit
Router(config)#

Ctrl+F6 to exit CLI focus
```

Top

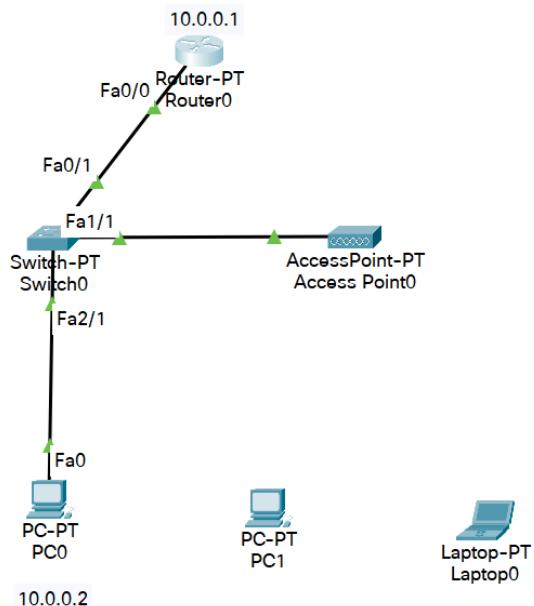
Copy **Paste**

iv. Observation

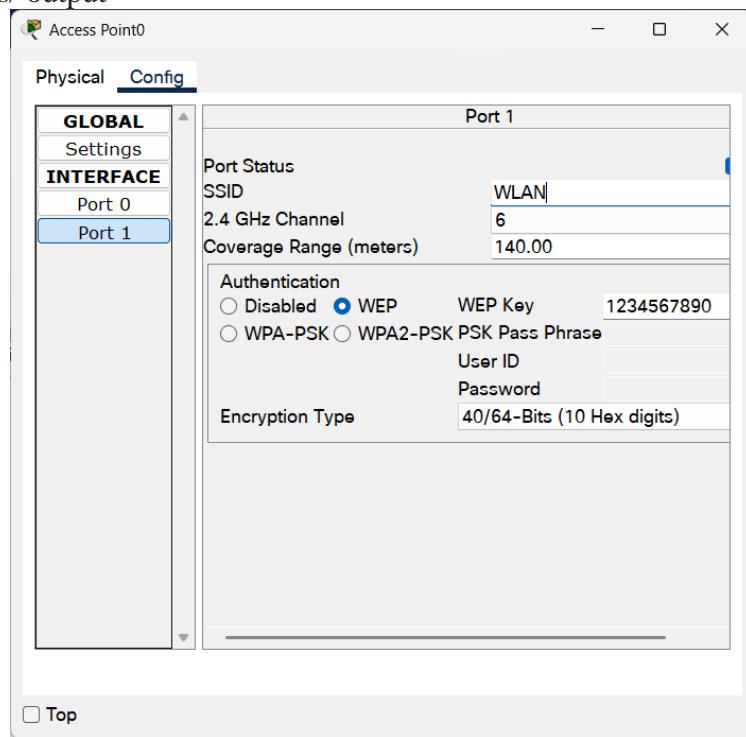


Program 12

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



- iii. Screen shots/ output



PC1

- Physical**
- Config
- Desktop
- Programming

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

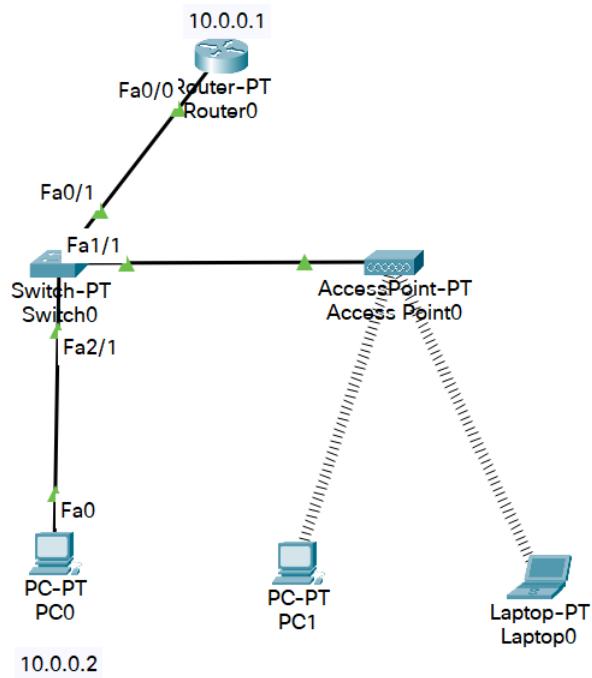
Laptop0

- 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 showing the output of a ping command:

```

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

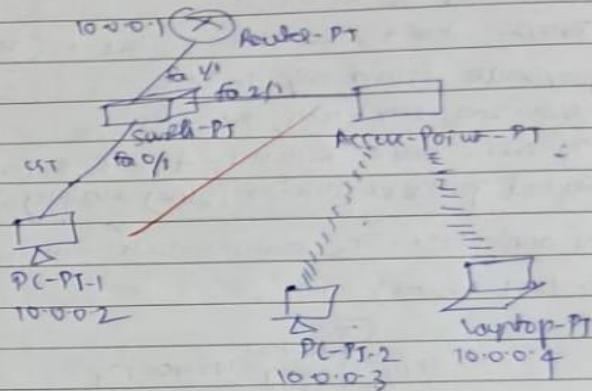
C:\>

```

iv. Observation

→ IEEE 802.11 is networking standard that supports virtual LANs (VLAN) on an IEEE 802.11 network.

After: WLAN and wireless communicate wirelessly



- configure PC1 and laptop
- configure Access-point-SSID name - any name (WLAN here)
- Set WEP and key - 1234567890
- configure PC-2 and laptop wireless standards
- switch off device, drag existing Pt-Host-Nat-Item to Lng and drag WMP300N to empty port, switch on
- go config, SSID, WEP, key, IP address and gateway (as normally done) to the device
- ping to every other device and check the result

> ping 10.0.0.3

pinging 10.0.0.3 with 32 bytes of data

Reply from 10.0.0.3: bytes=32 time=16ms TTL=128

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

→ Before connection: Destination Host Unreachable

→ ping 10.0.0.10 : Request timed out

Outcome: Constructed VLAN and communication among PCs. Rotating traffic from other VLANs and built a WLAN enabling wireless communication.

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: 1111111111111111111100000

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

iv. Observation

Date / /
Page / /

29 Work a program for Error Detection using CRC-CITT (16 bits)

Code:

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

int crc(char*ip, char*op, char*poly, int mode) {
    strcpy(op, ip);
    if (mode) {
        for (int i=1; i< strlen(poly); i++)
            strcat(ip, "0");
    }
    // Perform XOR on msg with selected polynomial
    for (int i=0; i< strlen(ip); i++) {
        if (ip[i] == '1') {
            for (int j=0; j< strlen(poly); j++)
                if (ip[i+j] == poly[j]) op[i+j] = '0';
                else op[i+j] = '1';
        }
    }
    // Work for error, return 0 if error detected
    for (int i=0; i< strlen(ip); i++)
        if (op[i] == '1') return 0;
    return 1;
}

int main() {
    char ip[10], op[10], serv[5];
    char poly[] = "1000100000100001";
    cout << "Enter input message in binary" << endl;
    cin >> ip;
    crc(ip, op, poly, 1);
    cout << "The transmitted message is:" << ip << op + strlen(ip) - 1 << endl;
}
```

Date / /
Page / /

29 "Enter received message in binary" << endl;
cin >> ip;
if (ip == "No error in data") cout << "No error in data" << endl;
else cout << "Error in data transmission has occurred" << endl;
return 0;

Output:
→ Enter the input message in binary
1010
The transmitted message is: 101010000101001010
Enter the received message in binary: 101010000101001010
No error in data

→ Enter the input message in binary
1010
The transmitted message is: 101010100001010010
Enter the received message in binary: 101010100001010010
Error in data transmission has occurred.

8. Leaky Bucket Algorithm
#include <iostream>
#include <bits/stdc++.h>
using namespace std;
int main() {
 int no_of_queries, storage, output_pkt_size;
 int input_pkt_size, bucket_size, size;
 storage = 0;
 no_of_queries = 4;
 output_pkt_size = 1;
 bucket_size = 10;
 input_pkt_size = 4;
 size = 10;
 for (int i=0; i< no_of_queries; i++) {
 if (storage <= size) {
 if (storage <= bucket_size) {
 storage += input_pkt_size;
 cout << "Data packet accepted" << endl;
 } else {
 cout << "Data packet rejected" << endl;
 }
 } else {
 cout << "Data packet rejected" << endl;
 }
 }
}

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

Date / /
Page / /

```
leaky" enter received message in binary" <eval>
clm, decy;
if (rcv(pck,poly,0)) rcv = "no error in data" <eval>
else rcv = "error in data transmission has occurred" <eval>
return;
```

Output:

→ Enter the input message in binary
1010
The transmitted message is: 10101010000101001010
Enter the received message in binary: 1010101000100101
no error in data

→ Enter the input message in binary
1010
The transmitted message is: 101010100001010010
Enter the received message in binary: 10101010001010011
error in data transmission has occurred.

Leaky Bucket Algorithm

```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;
int main(){
    int no_of_queries, storage, output_pkt_size;
    int input_pkt_size, bucket_size, size_left;
    storage = 0;
    no_of_queries = 4;
    input_pkt_size = 10;
    bucket_size = 10;
    output_pkt_size = 4;
    cout << "Leaky Bucket Algorithm" << endl;
    cout << "Bucket size = " << bucket_size << endl;
    cout << "Input packet size = " << input_pkt_size << endl;
    cout << "Output packet size = " << output_pkt_size << endl;
    cout << "Number of queries = " << no_of_queries << endl;
    cout << "Storage = " << storage << endl;
    cout << "Size left = " << size_left << endl;
    for (int i = 0; i < no_of_queries; i++) {
        if (input_pkt_size <= size_left) {
            cout << "Packet loss = 0" << endl;
            cout << "Bucket size = " << bucket_size << endl;
            cout << "Input packet size = " << input_pkt_size << endl;
            cout << "Output packet size = " << output_pkt_size << endl;
            cout << "Storage = " << storage << endl;
            cout << "Size left = " << size_left << endl;
            cout << "Data transmitted = " << input_pkt_size << endl;
            cout << "Data received = " << output_pkt_size << endl;
            cout << "Storage after transmission = " << storage << endl;
            cout << "Size left after transmission = " << size_left << endl;
            cout << endl;
            storage -= output_pkt_size;
            size_left -= input_pkt_size;
        } else {
            cout << "Packet loss = 1" << endl;
            cout << "Bucket size = " << bucket_size << endl;
            cout << "Input packet size = " << input_pkt_size << endl;
            cout << "Output packet size = " << output_pkt_size << endl;
            cout << "Storage = " << storage << endl;
            cout << "Size left = " << size_left << endl;
            cout << "Data transmitted = " << input_pkt_size << endl;
            cout << "Data received = " << output_pkt_size << endl;
            cout << "Storage after transmission = " << storage << endl;
            cout << "Size left after transmission = " << size_left << endl;
            cout << endl;
        }
    }
}
```

Date / /
Page / /

```
for (int i = 0; i < no_of_queries; i++) {
    size_left = bucket_size - storage;
    if (input_pkt_size <= size_left) {
        storage += input_pkt_size;
    } else {
        cout << "Packet loss = 1" << endl;
        cout << "Bucket size = " << bucket_size << endl;
        cout << "Input packet size = " << input_pkt_size << endl;
        cout << "Output packet size = " << output_pkt_size << endl;
        cout << "Storage = " << storage << endl;
        cout << "Size left = " << size_left << endl;
        cout << "Data transmitted = " << input_pkt_size << endl;
        cout << "Data received = " << output_pkt_size << endl;
        cout << "Storage after transmission = " << storage << endl;
        cout << "Size left after transmission = " << size_left << endl;
        cout << endl;
    }
}

return;
```

Output:

Bucket size = 4 out of bucket size = 10
Bucket size = 7 out of bucket size = 10
Bucket size = 10 out of bucket size = 10
Packet loss = 4
Bucket size = 9 out of bucket size = 10

(RC) Outcome: successfully implemented a program to detect transmission errors in data (cycle redundancy check)

(RC) Leaky bucket algorithm successfully implements a program for congestion control, simulating packet flow and self-limiting to avoid network congestion.

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()
```

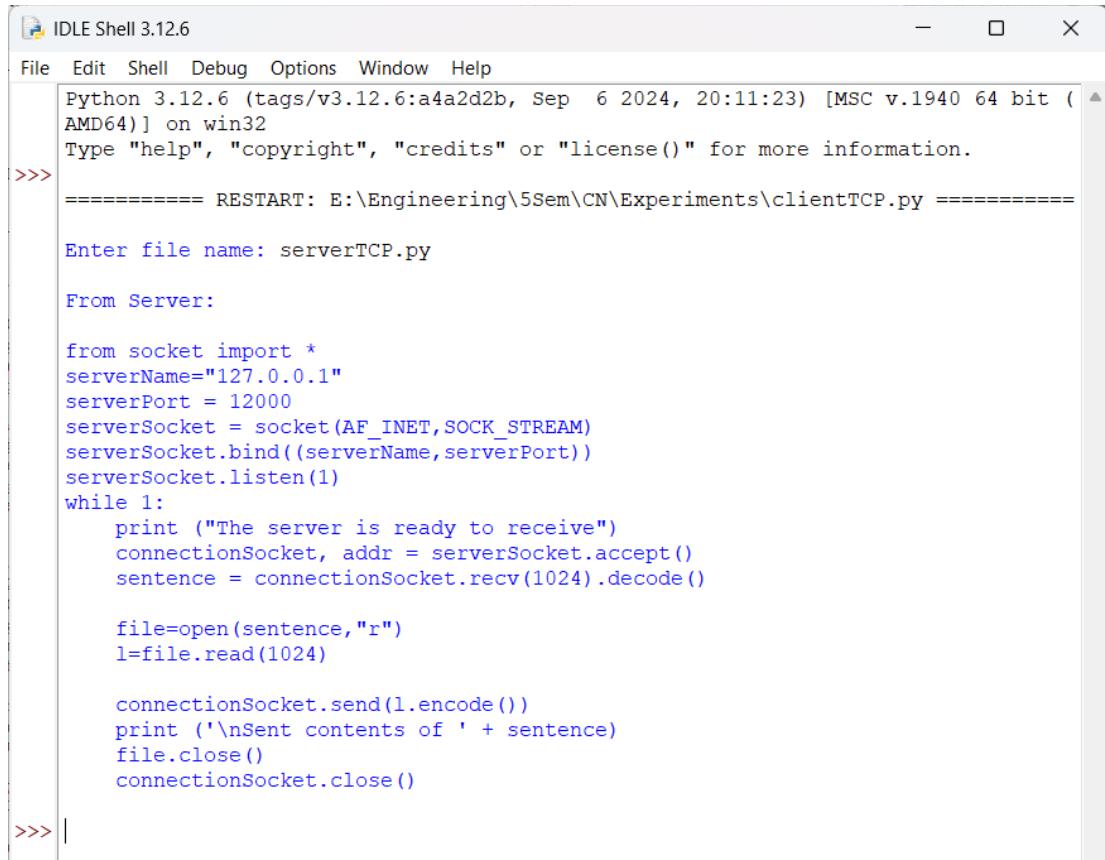
```
serverTCP.py
from socket import *
serverName="127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print ("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence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

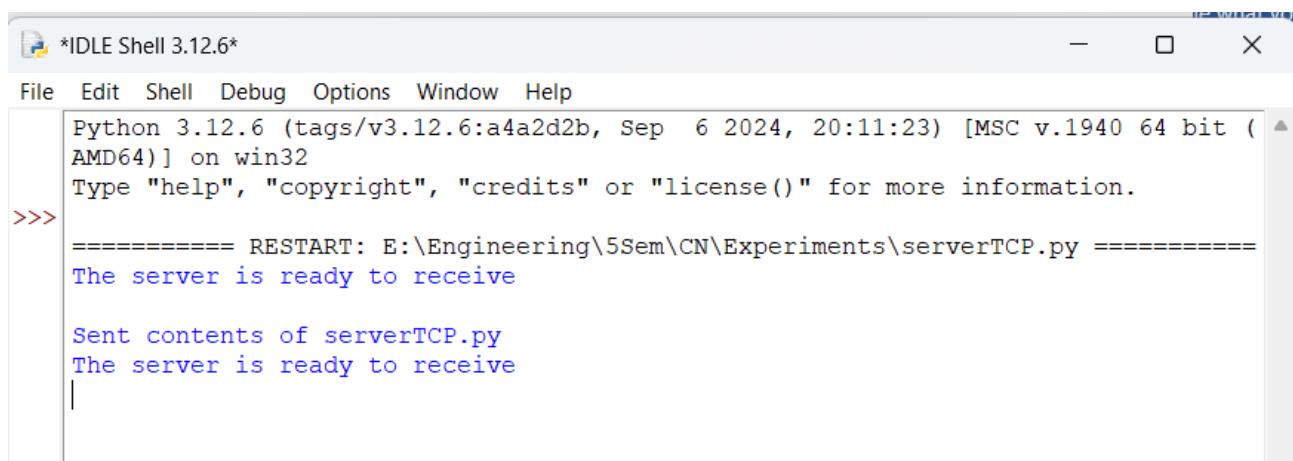
from socket import *
serverName="127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print ("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()

    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

Date _____
Page _____

Socket Programming Aim: Build client/server applications that communicate using sockets

TCP: (Using TCP/IP, write circuit server program for server and client sending filename) (Transmission Control Protocol Internet Protocol)

serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket (AF_INET, SOCK_STREAM)
clientSocket. connect ((serverName, serverPort))
~~sentence = input ("In what file name: ")~~
~~clientSocket.send (sentence.encode ())~~
~~print ("From server\n")~~
~~clientSocket.close ()~~

serverTCP.py

from socket import *
serverName = "127.0.0.1"
serverPort = 12000
serverSocket = socket (AF_INET, SOCK_STREAM)
serverSocket.bind ((serverName, serverPort))
while 1:
 print ("Server is ready to receive")
 file = open (sentence, "r")
 l = file.read (1024)
 connectionSocket.send (l.encode ())
 file.close()
 connectionSocket.close ()

Output:

Server is ready to receive
Sent contents of server TCP.py
Server is ready to receive

Program 4

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

- ii. Procedure

clientUDP.py

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

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

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

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

serverUDP.py

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file=open(sentence,"r")
    con=file.read(2048)
```

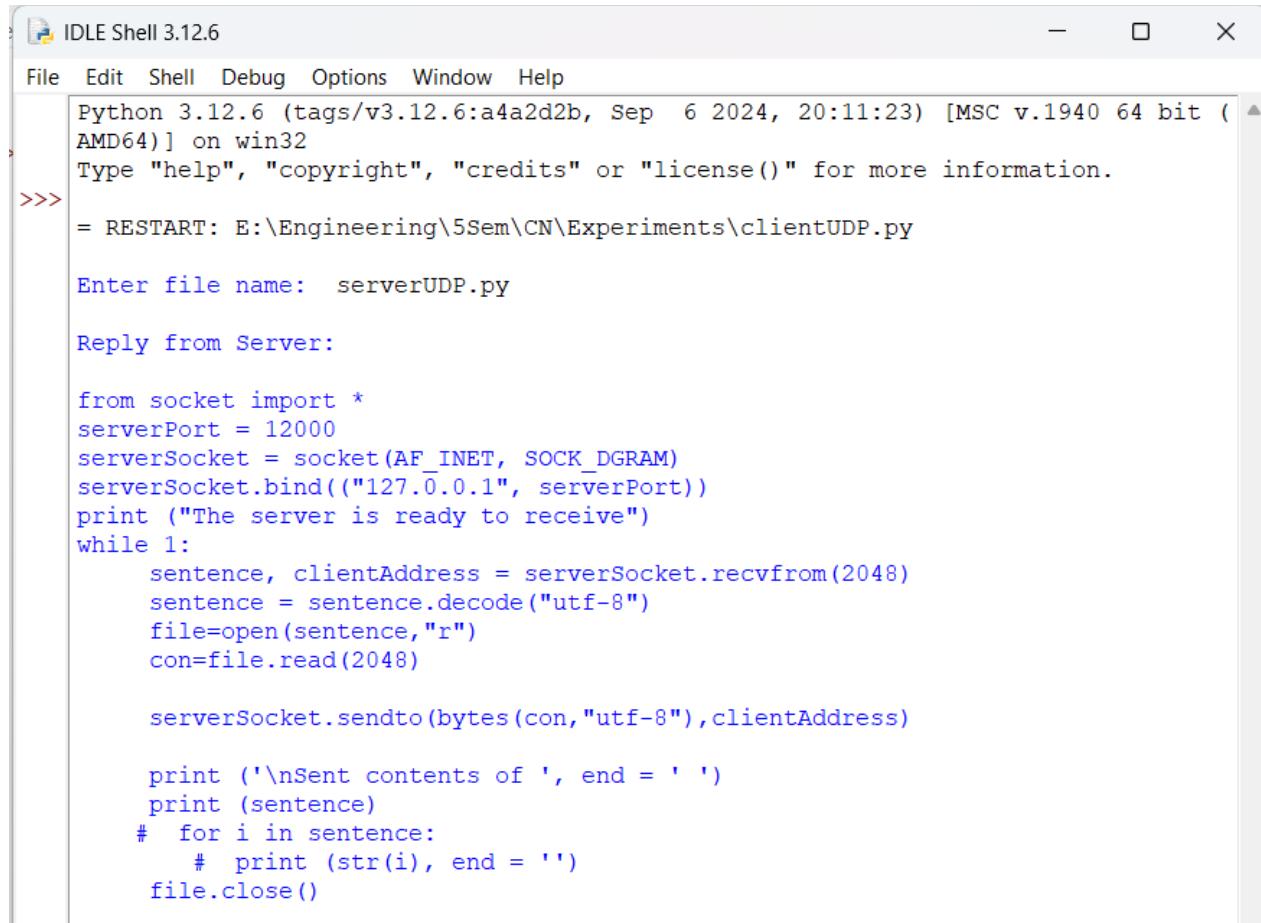
```
    serverSocket.sendto(bytes(con,"utf-8"),clientAddress)
```

```
    print ('\nSent contents of ', end = ' ')
    print (sentence)
    # for i in sentence:
```

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

iii. Screen shots/ output

Client



```
IDLE Shell 3.12.6
File Edit Shell Debug Options Window Help
Python 3.12.6 (tags/v3.12.6:a4a2d2b, Sep  6 2024, 20:11:23) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:\Engineering\5Sem\CN\Experiments\clientUDP.py

Enter file name: serverUDP.py

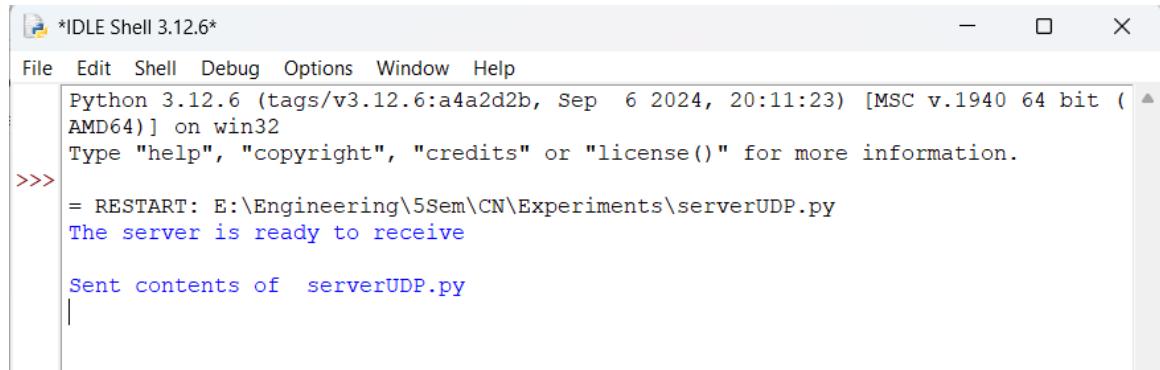
Reply from Server:

from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file=open(sentence,"r")
    con=file.read(2048)

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

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

Server



```
*IDLE Shell 3.12.6*
File Edit Shell Debug Options Window Help
Python 3.12.6 (tags/v3.12.6:a4a2d2b, Sep  6 2024, 20:11:23) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:\Engineering\5Sem\CN\Experiments\serverUDP.py
The server is ready to receive

Sent contents of  serverUDP.py
```

iv. Observation

Page _____

VDP (using UDP socket, write client-server program to make client reading file name and server to send back contents of requested file if present from socket import * (file Datagram protocol)
 ServerPort = "139.0.0.1"
 SERVERPORT = 12000
 CLIENTPORT = 10240 (AF_INET, SOCK_DGRAM)
 sentence = input("Enter file name: ")
 filecontents, ClientAddress = SERVERSOCKET.recvfrom(2048)
 print(filecontents.decode("UTF-8"))
 # ClientSocket.close()
 # ClientSocket.close()

ServerVDP.py

from socket import *
 SERVERPORT = 12000
 SERVERSOCKET = socket(AF_INET, SOCK_DGRAM)
 while True:
 sentence, ClientAddress = SERVERSOCKET.recvfrom(2048)
 sentence = sentence.decode("UTF-8")
 file = open(sentence, "r")
 on = file.read(2048)
 SERVERSOCKET.sendto(on, (ClientAddress, "UTF-8"), ClientAddress)
 print("File content of : ", end="")
 print(file.read())
 file.close()

Output: Server is ready to receive
 Sent contents of Server VDP.py
 Server is ready to receive

Page _____

Outcome:

(TCP): Implemented TCP Client-server program where Client requests file by name and server sends back the file contents if it exists demonstrating reliable file transfer.

(VDP): Implemented UDP Client-server program where Client sends file name and server sends file content supporting connectionless communication

26/12