

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB RECORD

Computer Network Lab (23CS5PCCON)

Submitted by

SANJANA SURESH(1BM22CS239)

in partial fulfillment 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 **Sanjana Suresh(1BM22CS239)**, 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

Sl. No.	Date	Experiment Title	Page No.
1	04.10.2024	Simple PDU message and ping responses	1-3
2	18.10.2024	Router IP address configuration	4-7
3	25.10.2024	Default and static routing	8-12
4	08.11.2024	DHCP Configuration within and outside a LAN	13-16
5	22.11.2024	RIP routing protocol	17-21
6	29.11.2024	OSPF routing protocol	22-27
7	22.11.2024	TTL/Life of a packet	28-31
8	08.11.2024	DNS configuration	32-35
9	20.12.2024	ARP	36-39
10	20.12.2024	TELNET protocol	40-42
11	20.12.2024	VLAN	43-46
12	20.12.2024	WLAN	47-51
13	15.11.2024	CRC-Error detection	52-54
14	15.11.2024	Leaky bucket algorithm	54-56
15	20.12.2024	Client-Server program using TCP/IP sockets	56-58
16	20.12.2024	Client- Server program using UDP sockets	58-61

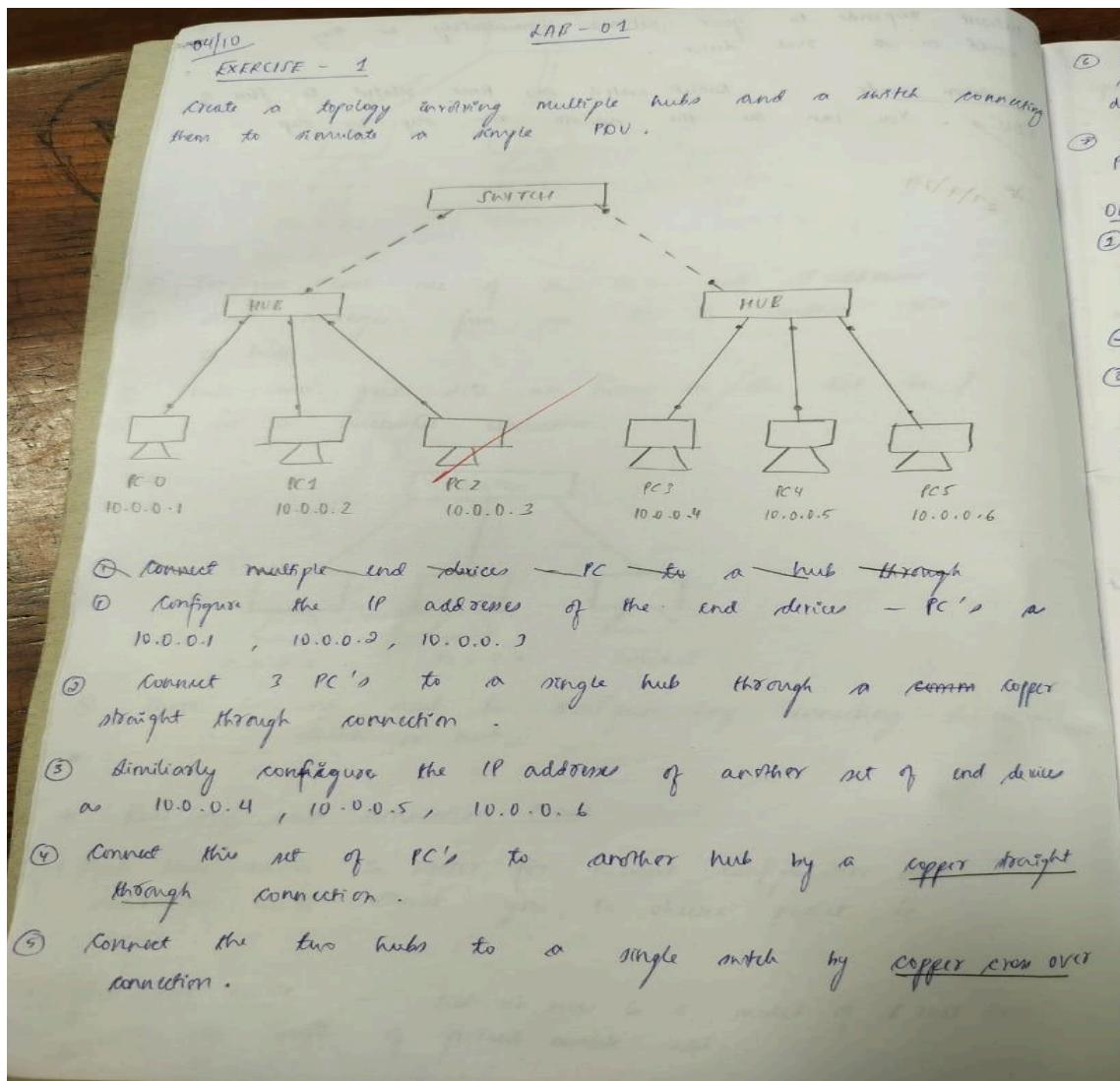
Github Link: https://github.com/SanjanaSuresh30/1BM22CS239_CN_LAB

Program 1

Aim of the program:

Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping messages.

Procedure along with the topology:



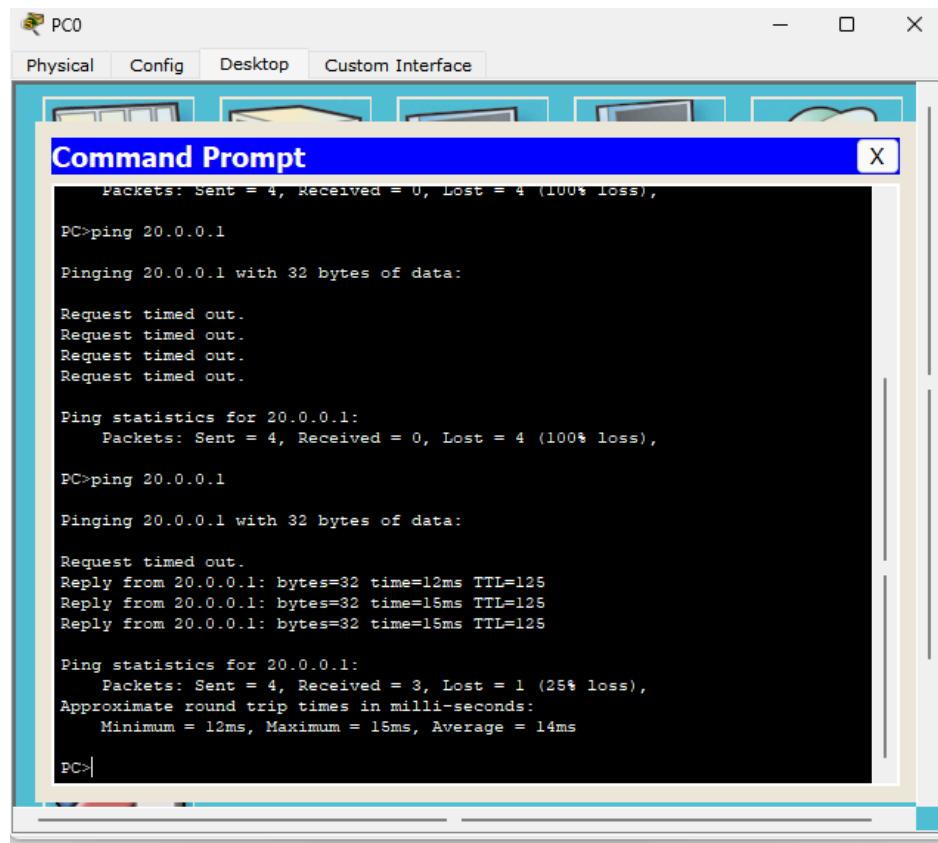
connecting

⑥ When the connection is available - green dots present on links, send a PDU (data packet) from PC0 to PC2 -
data packets through a hub.

⑦ Similarly, when the connection is available - green dots present on links, send a PDU (data packet) from PC1 to PC4 - through a switch.

Screen shots/ output :

```
Pinging 20.0.0.1 with 32 bytes of data:  
  
Request timed out.  
Request timed out.  
Request timed out.  
Request timed out.  
  
Ping statistics for 20.0.0.1:  
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),  
  
PC>ping 20.0.0.1  
  
Pinging 20.0.0.1 with 32 bytes of data:  
  
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127  
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127  
Reply from 20.0.0.1: bytes=32 time=1ms TTL=127  
Reply from 20.0.0.1: bytes=32 time=1ms TTL=127  
  
Ping statistics for 20.0.0.1:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 0ms, Maximum = 1ms, Average = 0ms  
  
PC>|
```



Observation:

OBSERVATIONS

- ① During the transmission of data packets, last status is 'in progress'. After completed, the last status changes to 'successful'.
- ② Type in the Event List as ICMP.
- ③ PDU information can be seen ~~in~~ only by clicking on the message icon which shows layers in OSI model.

TASK - 2

- ① Click on a PC that has received a data packet, go to Desktop tab and click on command prompt
- ② Run the command
PC > ping 10.0.0.4 // destination address

OBSERVATIONS

① The command prompt says - $\text{TTL} = \text{Time To Live}$

~~PC > ping 10.0.0.3~~

Reply from 10.0.0.3 : bytes = 32 time = 3ms TTL = 128

Reply from 10.0.0.3 : bytes = 32 time = 4ms TTL = 128

Reply from 10.0.0.3 : bytes = 32 time = 4ms TTL = 128

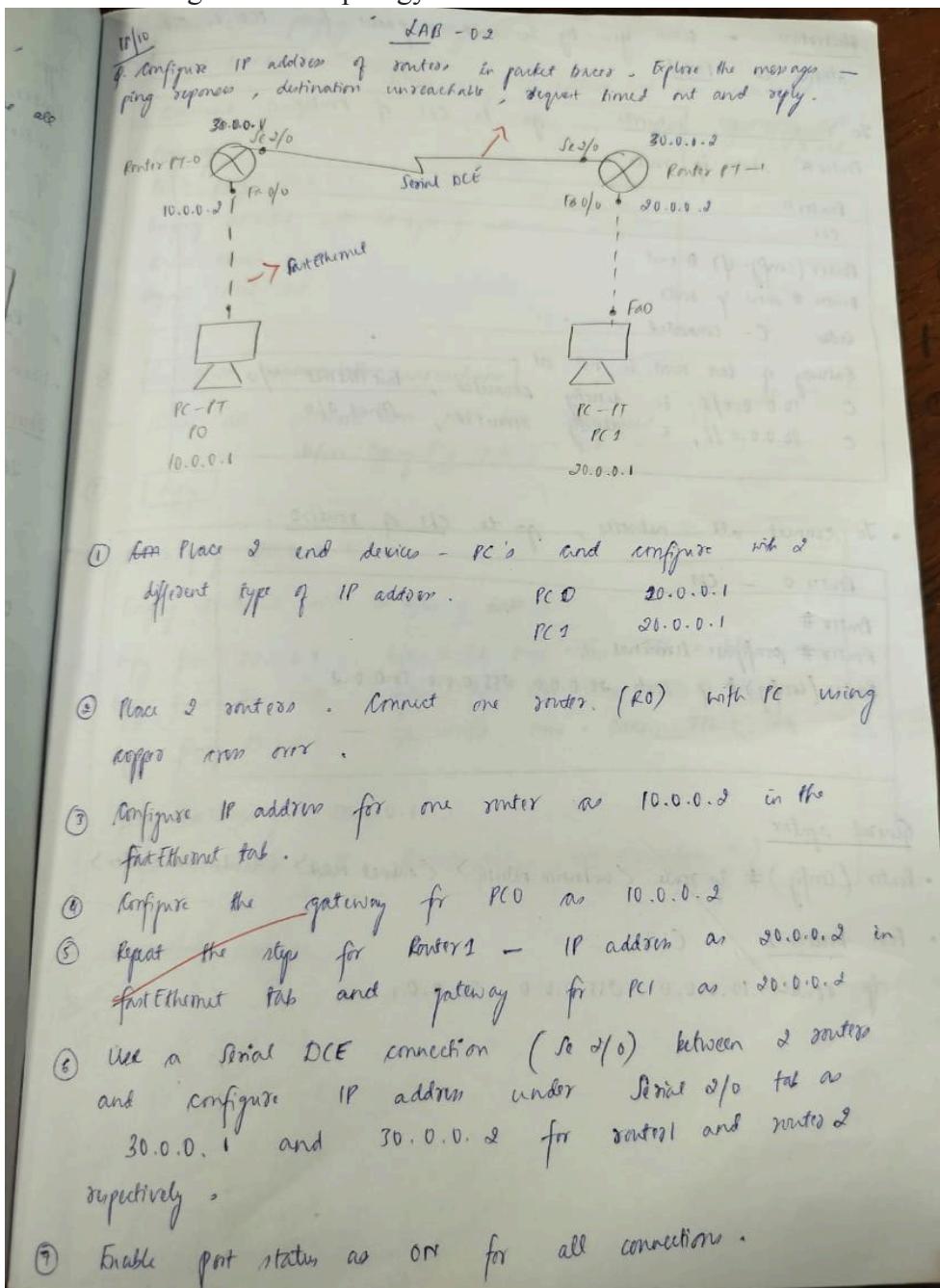
Ping statistics for 10.0.0.3
Packets : Sent = 4, Received = 4, Lost = 0

Program 2

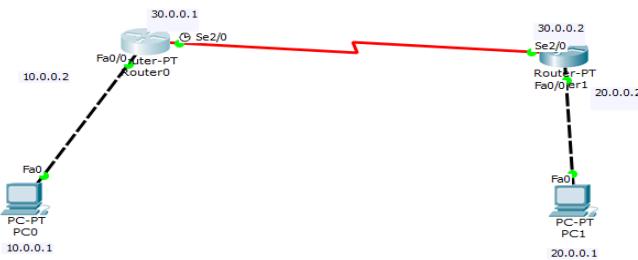
Aim of the program:

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

Procedure along with the topology



Screen shots/ output :



```
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 20.0.0.1:
  Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 30.0.0.2
Pinging 30.0.0.2 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 30.0.0.2:
  Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 30.0.0.1
Pinging 30.0.0.1 with 32 bytes of data:
```

Router0

Physical Config CLI

IOS Command Line Interface

```

Router>enable
Router#config terminal
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)#
Router(config)#interface Serial2/0
Router(config-if)#exit
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
S    20.0.0.0/8 [1/0] via 30.0.0.2
C    30.0.0.0/8 is directly connected, Serial2/0
Router#
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
Router(config)##interface FastEthernet0/0

```

Copy Paste

PC0

Physical Config Desktop Custom Interface

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=13ms TTL=254
Reply from 30.0.0.2: bytes=32 time=5ms TTL=254
Reply from 30.0.0.2: bytes=32 time=10ms TTL=254
Reply from 30.0.0.2: bytes=32 time=7ms TTL=254

Ping statistics for 30.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 5ms, Maximum = 13ms, Average = 8ms

PC>ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data:

Reply from 20.0.0.1: bytes=32 time=1ms TTL=126
Reply from 20.0.0.1: bytes=32 time=7ms TTL=126
Reply from 20.0.0.1: bytes=32 time=14ms TTL=126
Reply from 20.0.0.1: bytes=32 time=7ms TTL=126

Ping statistics for 20.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 14ms, Average = 7ms

PC>

```

Observation:

Observation → When you try to send a packet from PC0 to Router, status is Failed.

To recognize all networks, go to CLI of Router

```
Router# show ip route
```

Router 0
CLI
Router (config-if) # exit
Router # show ip route
Codes: C - connected
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, Serial 0/0

- To connect all networks, go to CLI of router0.

Router 0 - CLI
Router #
Router # configure terminal
Router (config)# ip route 20.0.0.0 255.0.0.0 30.0.0.2

General syntax

Router (config)# ip route <unknown network> <subnet mask> <unknown routes
For Router 1, CLI
ip route 10.0.0.0 255.0.0.0 30.0.0.1

Responses while pinging a message →

① [Request timeout] ✓

- When wrong IP address has been mentioned.
- Gateway not configured.

Command prompt
Ping
PC > ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Request timed out.
Request timed out.

② [Destination unreachable/unreachable]

- When all networks have not been recognized. *ip route* before typing (*ip route*)

③ [Reply]

PC > ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Reply from 20.0.0.1: bytes = 32 time = 1ms TTL = 126
Reply from 20.0.0.1: bytes = 32 time = 3ms TTL = 126
Reply from 20.0.0.1: bytes = 32 time = 11ms TTL = 126
Reply from 20.0.0.1: bytes = 32 time = 11ms TTL = 126
Ring statistics for 20.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)

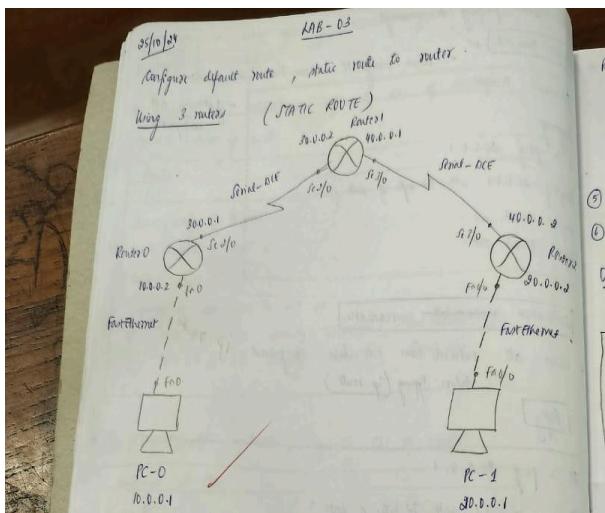
- Successful message

18/10

Program 3

Aim of the program:
Configure default route, static route to the Router

Procedure along with the topology



- ① Place 3 end devices PC-0 and PC-1 and configure them with 2 different IP address

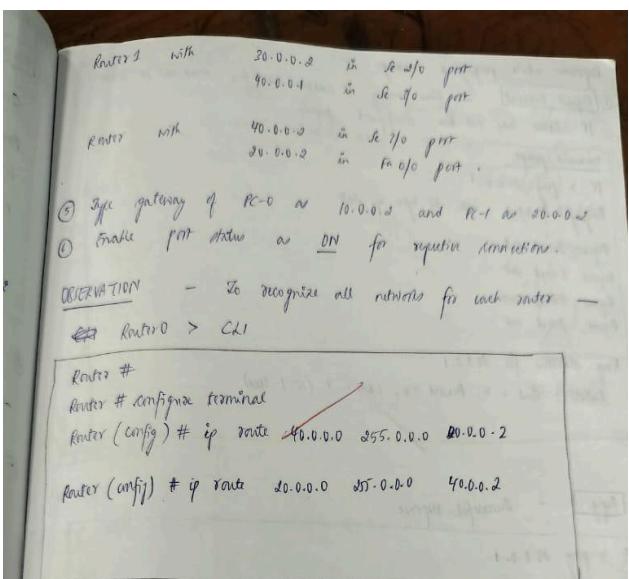
PC-0 10.0.0.1
PC-1 20.0.0.1

- ② In a 3 router configuration, there will be 4 different type of network.

- ③ Place 3 routers and connect each with Serial-RCE connection and connect Router-0 and Router-2 to end devices

PC-0 and PC-1 with FastEthernet respectively.

- ④ Configure Router 0 with 10.0.0.2 with Fa0/0 port.
30.0.0.1 in Se2/0 port.



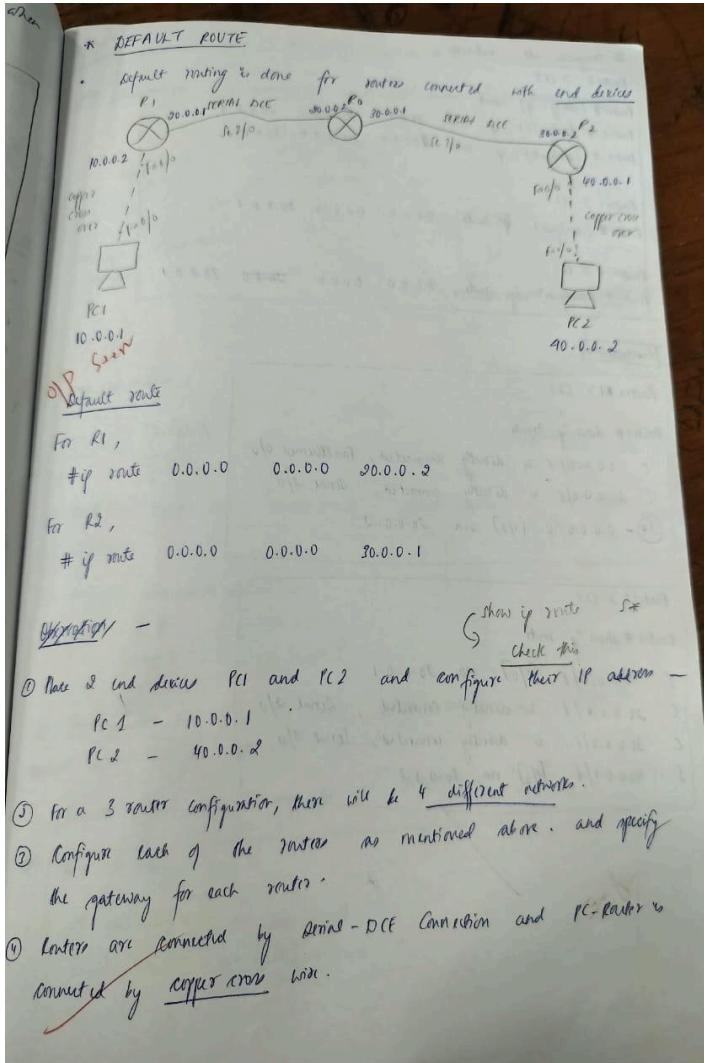
Similarly do the same for Router1 and Router2

Router1 > CLI

Router # configure terminal
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 40.0.0.2

Router 2 > CLI

Router # configure terminal
Router (config) # ip route 10.0.0.0 255.0.0.0 30.0.0.1
Router (config) # ip route 30.0.0.0 255.0.0.0 40.0.0.1



Screen shots/ output :

Router# Router0

Physical Config CLI

IOS Command Line Interface

```

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
S    20.0.0.0/8 [1/0] via 30.0.0.2
C    30.0.0.0/8 is directly connected, Serial2/0
S    40.0.0.0/8 [1/0] via 20.0.0.2
Router#enable
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
S    20.0.0.0/8 [1/0] via 30.0.0.2
C    30.0.0.0/8 is directly connected, Serial2/0
S    40.0.0.0/8 [1/0] via 20.0.0.2
Router#

```

Copy Paste

PC0

Physical Config Desktop Custom Interface

Command Prompt

```

PC>ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Request timed out.
Reply from 20.0.0.1: bytes=32 time=12ms TTL=125
Reply from 20.0.0.1: bytes=32 time=15ms TTL=125
Reply from 20.0.0.1: bytes=32 time=16ms TTL=125

Ping statistics for 20.0.0.1:
  Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 12ms, Maximum = 15ms, Average = 14ms

PC>ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Reply from 20.0.0.1: bytes=32 time=10ms TTL=125
Reply from 20.0.0.1: bytes=32 time=19ms TTL=125
Reply from 20.0.0.1: bytes=32 time=9ms TTL=125
Reply from 20.0.0.1: bytes=32 time=2ms TTL=125

Ping statistics for 20.0.0.1:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 19ms, Average = 10ms

PC>

```

Realtime

Scenario 0	Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	<input checked="" type="radio"/>	Failed	PC0	PC2	ICMP	■	0.000	N	0	(edit)	(delete)
	<input checked="" type="radio"/>	Successful	PC0	PC2	ICMP	■	0.000	N	1	(edit)	(delete)

New Delete Toggle PDU List Window

Router1

Physical Config CLI

IOS Command Line Interface

```
Router(config)#interface FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet1/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#ip address 20.0.0.2 255.0.0.0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial3/0
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)#
Router(config-if)#exit
Router(config)#interface FastEthernet1/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial2/0, changed state to down
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

%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)#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-if)#exit
Router(config)#
$SYS-5-CONFIG_I: Configured from console by console

Router#show ip routes
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, 0 - 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#
```

Observation:

Router while pinging a router -
 ① **Request timeout** - When wrong IP address has been mentioned or
 IP address has not been configured properly.

Command prompt

PC > ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data:

Request timed out.

Request timed out.

Request timed out.

Request timed out.

Ring statistics for 20.0.0.1 :

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

② **Reply** - Successful response

PC > ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data:

Request timed out.

Reply from 20.0.0.1 : bytes = 32 time = 16 ms TTx = 18.5

Reply from 20.0.0.1 : bytes = 32 time = 15 ms TTx = 12.5

Reply from 20.0.0.1 : bytes = 32 time = 16 ms TTx = 12.5

Ring statistics for 20.0.0.1 :

Packets : Sent = 4 Received = 3 Lost = 1 (25% loss).

To recognize all networks:

Router0 > CLI

Router# (config) # ip route 10.0.0.0 255.0.0.0 20.0.0.1

Router# (config) # ip route 0.0.0.0 255.0.0.0 20.0.0.2

Router# (config) # ip route 0.0.0.0 0.0.0.0 20.0.0.2

Router# (config) #

Router# (config) # ip route 0.0.0.0 0.0.0.0 20.0.0.2

Router# (config) #

Router# (config) # ip route 0.0.0.0 0.0.0.0 20.0.0.1

Observations

Router#1 > CLI

Router# show ip route

C 10.0.0.0/8 is directly connected, FastEthernet 0/0

C 20.0.0.0/8 is directly connected, Serial 0/0

S* 0.0.0.0/0 [1/0] via 20.0.0.2

// default
routing

Router0 > CLI

Router# show ip route

S 10.0.0.0/8 [1/0] via 20.0.0.1

C 20.0.0.0/8 is directly connected, Serial 0/0

C 30.0.0.0/1 is directly connected, Serial 0/0

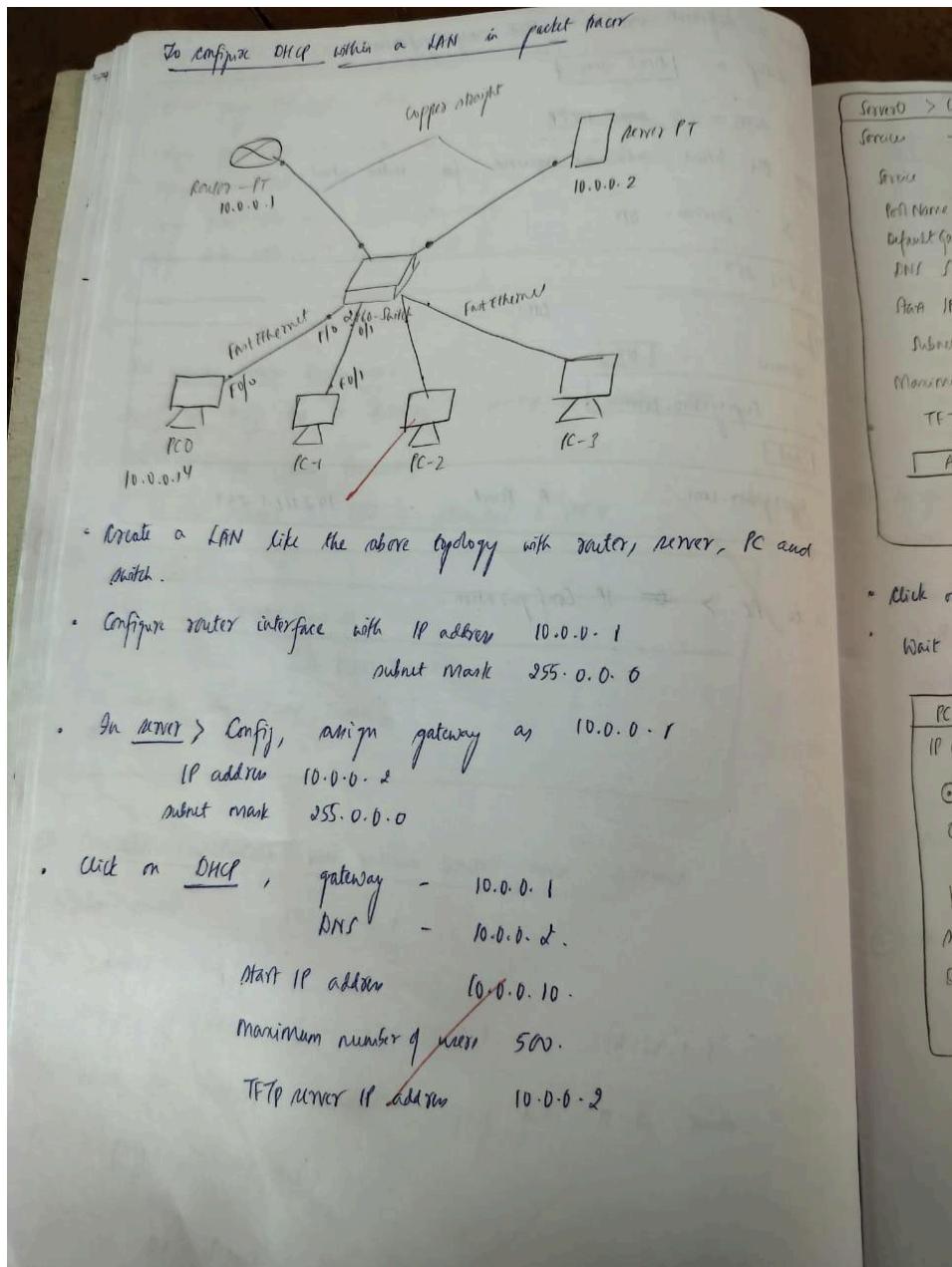
S 40.0.0.0/8 [1/0] via 20.0.0.2

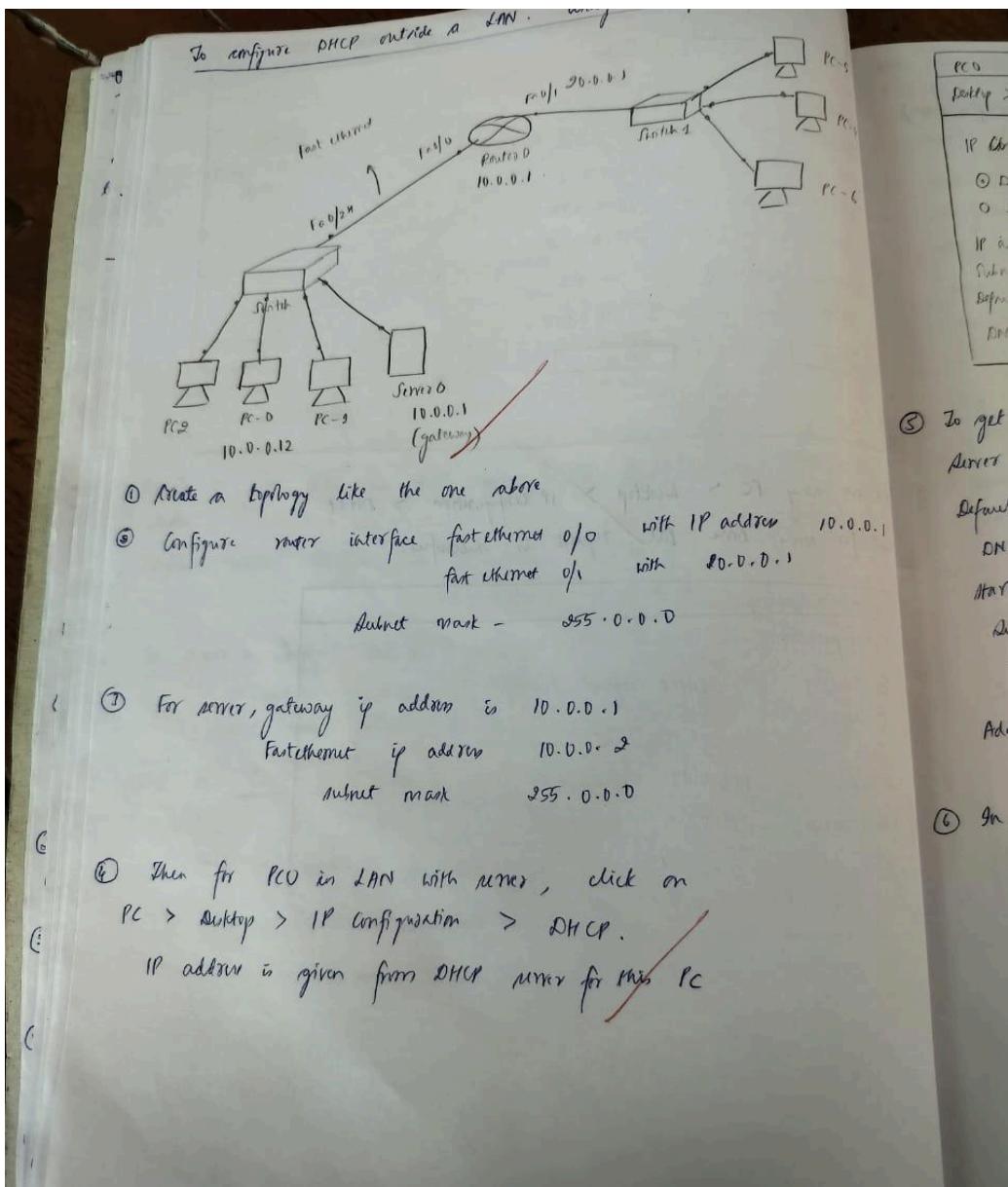
8/11

Program 4

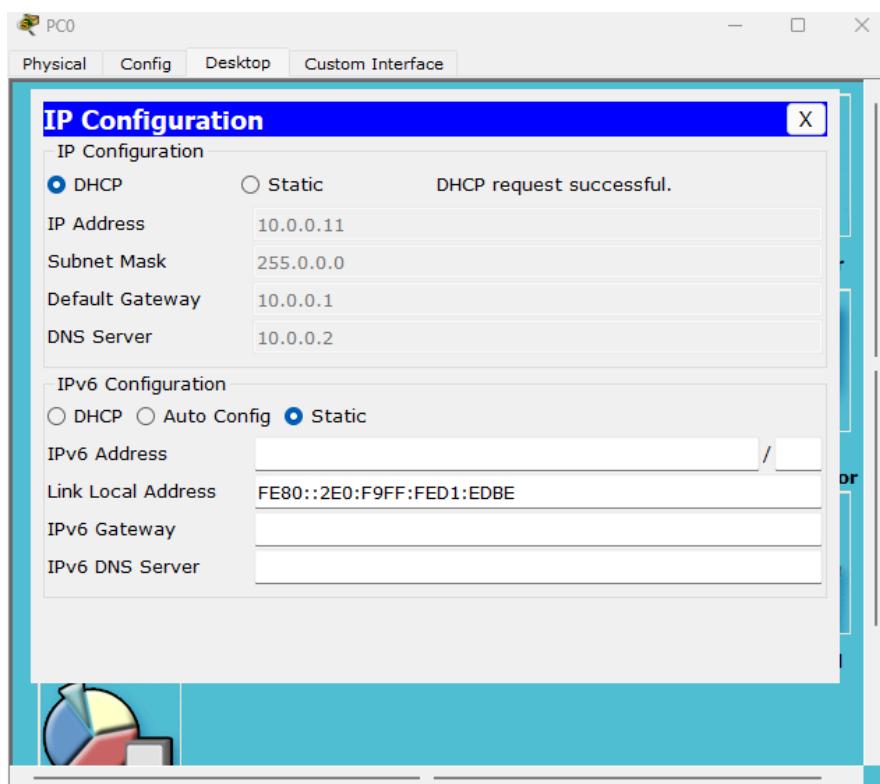
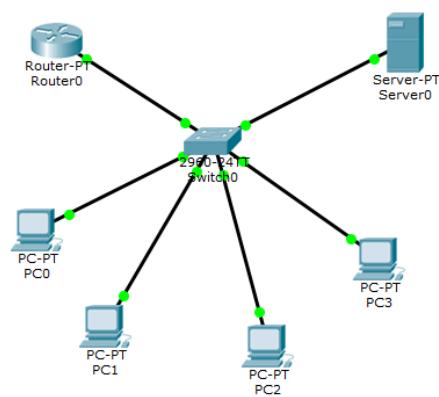
Aim of the program:
Configure DHCP within a LAN and outside LAN.

Procedure along with the topology:

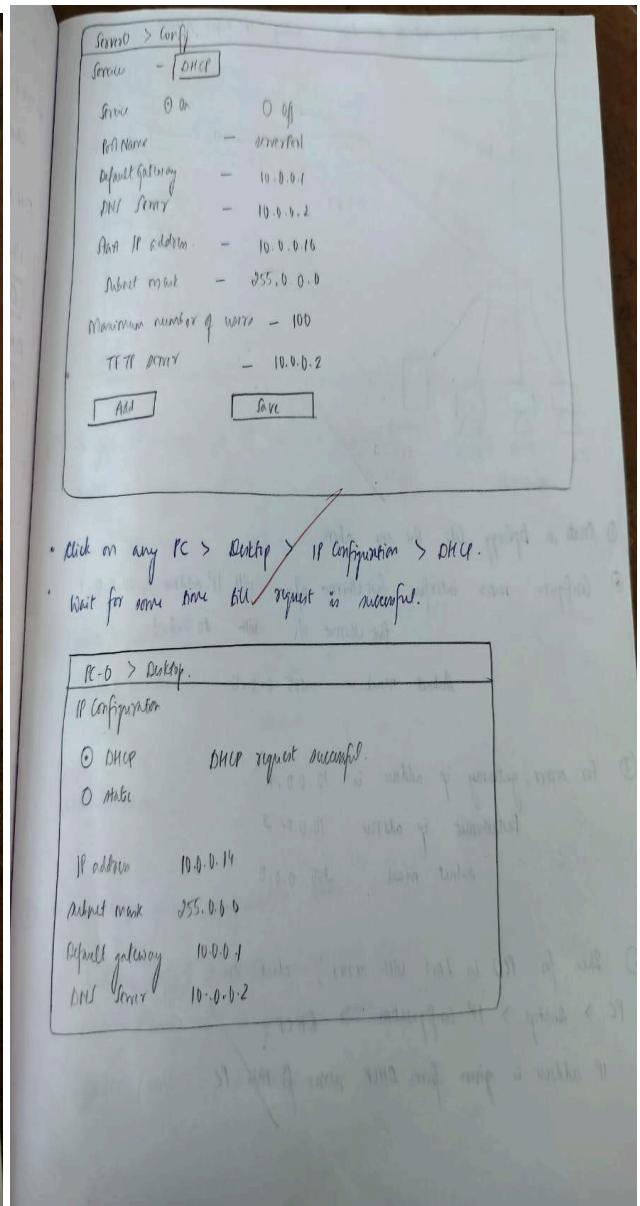
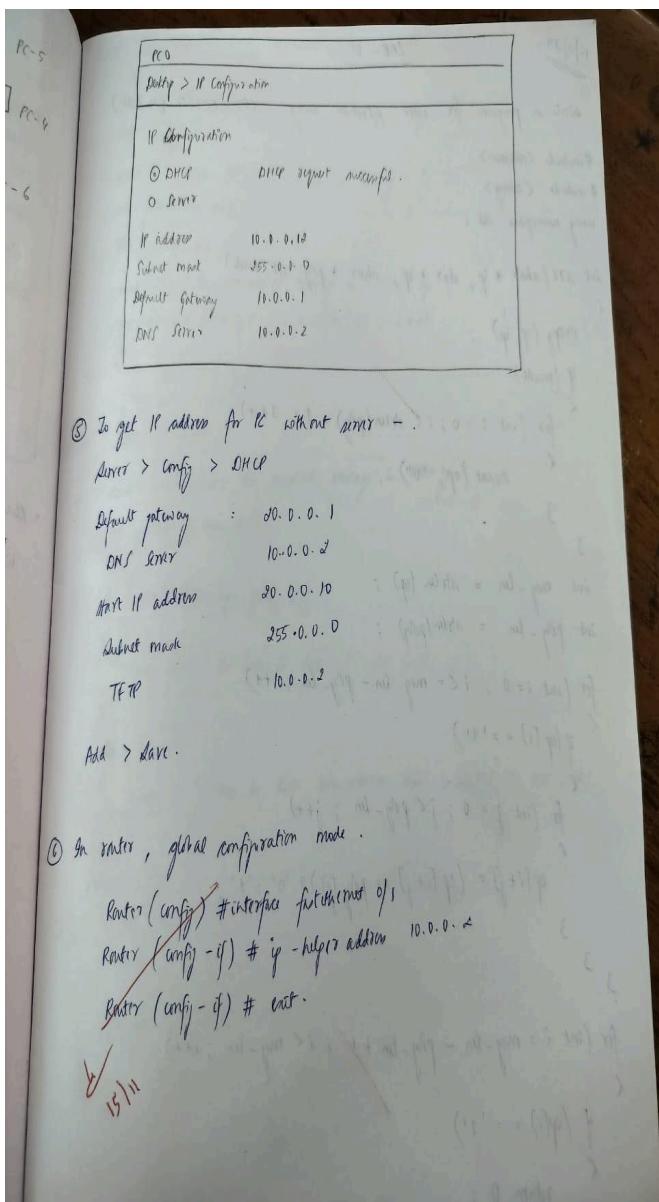




Screen shots/ output :



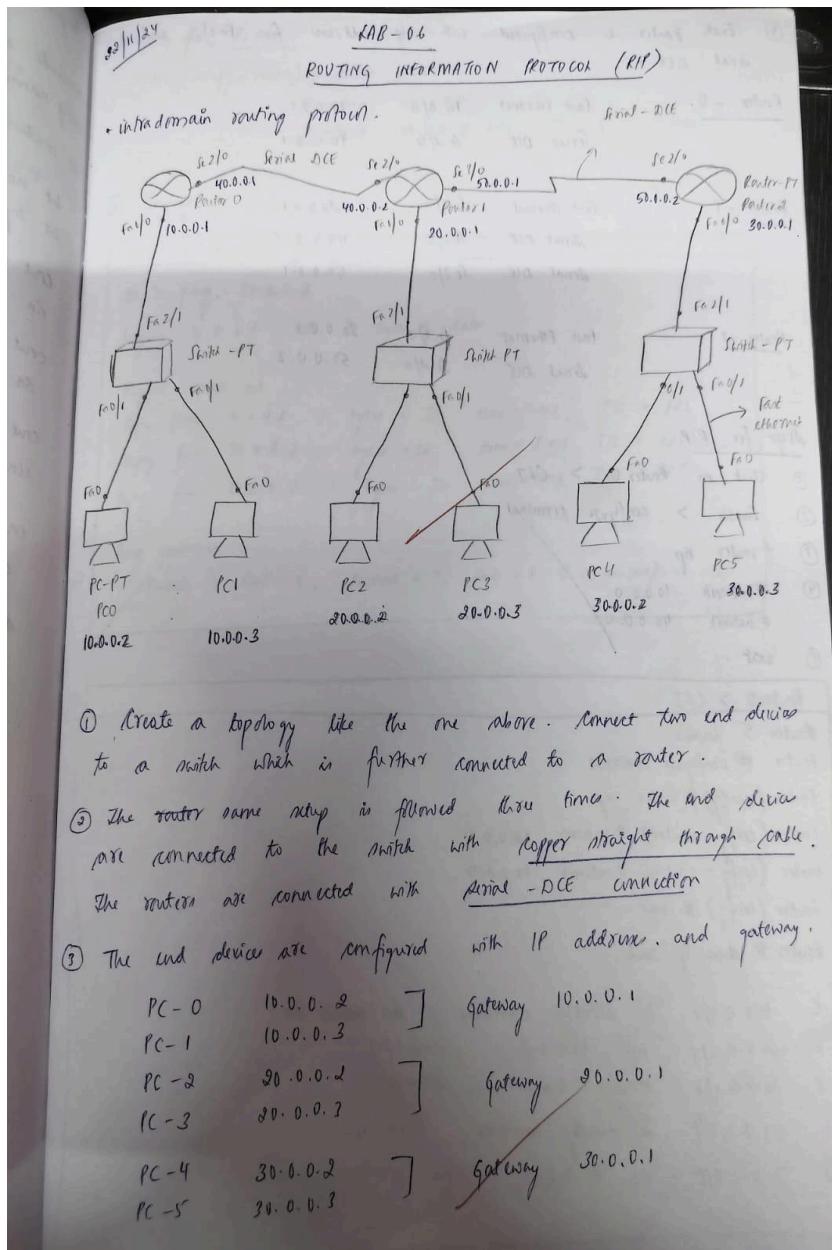
Observation:

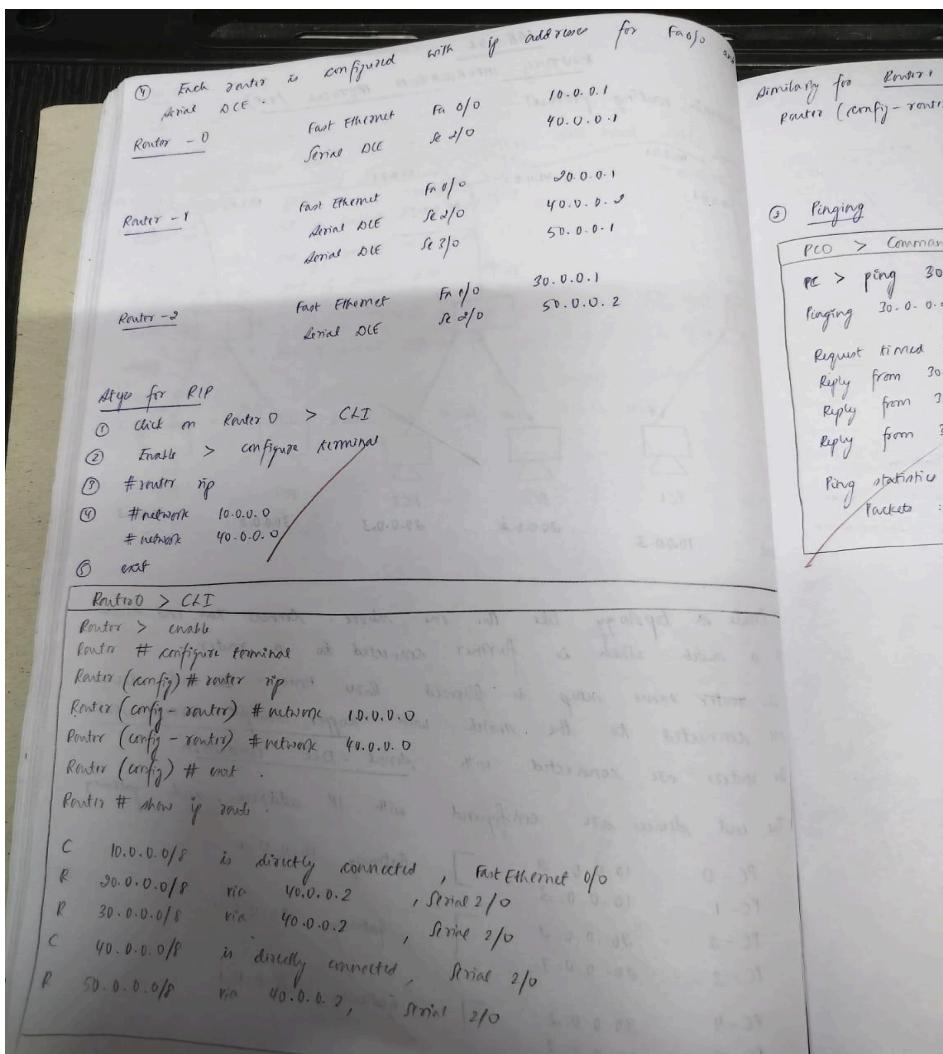


Program 5

Aim of the program:
Configure RIP routing Protocol in Routers

Procedure along with the topology:





Similarly for Router 1
Router (config)- router

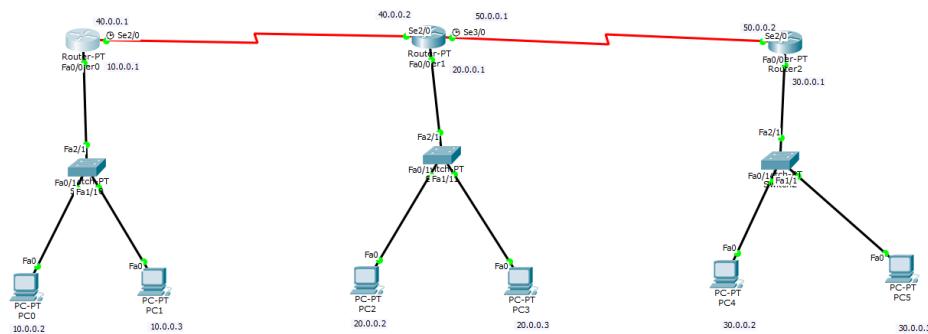
③ Pinging

PC0 > Command

```

PC > ping 30.
Pinging 30.0.0.2
Request timed out
Reply from 30.0.0.2
Reply from 30.0.0.2
Reply from 30.0.0.2
Ping statistics:
    Packets: = 
  
```

Screen shots/ output :



Router0

Physical Config CLI

IOS Command Line Interface

```
Router>enable
Router#configure terminal
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)#exit
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
R    20.0.0.0/8 [120/1] via 40.0.0.2, 00:00:12, Serial2/0
R    30.0.0.0/8 [120/2] via 40.0.0.2, 00:00:12, Serial2/0
C    40.0.0.0/8 is directly connected, Serial2/0
R    50.0.0.0/8 [120/1] via 40.0.0.2, 00:00:12, Serial2/0
Router#
```

Copy Paste

PC0

Physical Config Desktop Custom Interface

Command Prompt

```
Packet Tracer PC Command Line 1.0
PC>ping 30.0.0.2

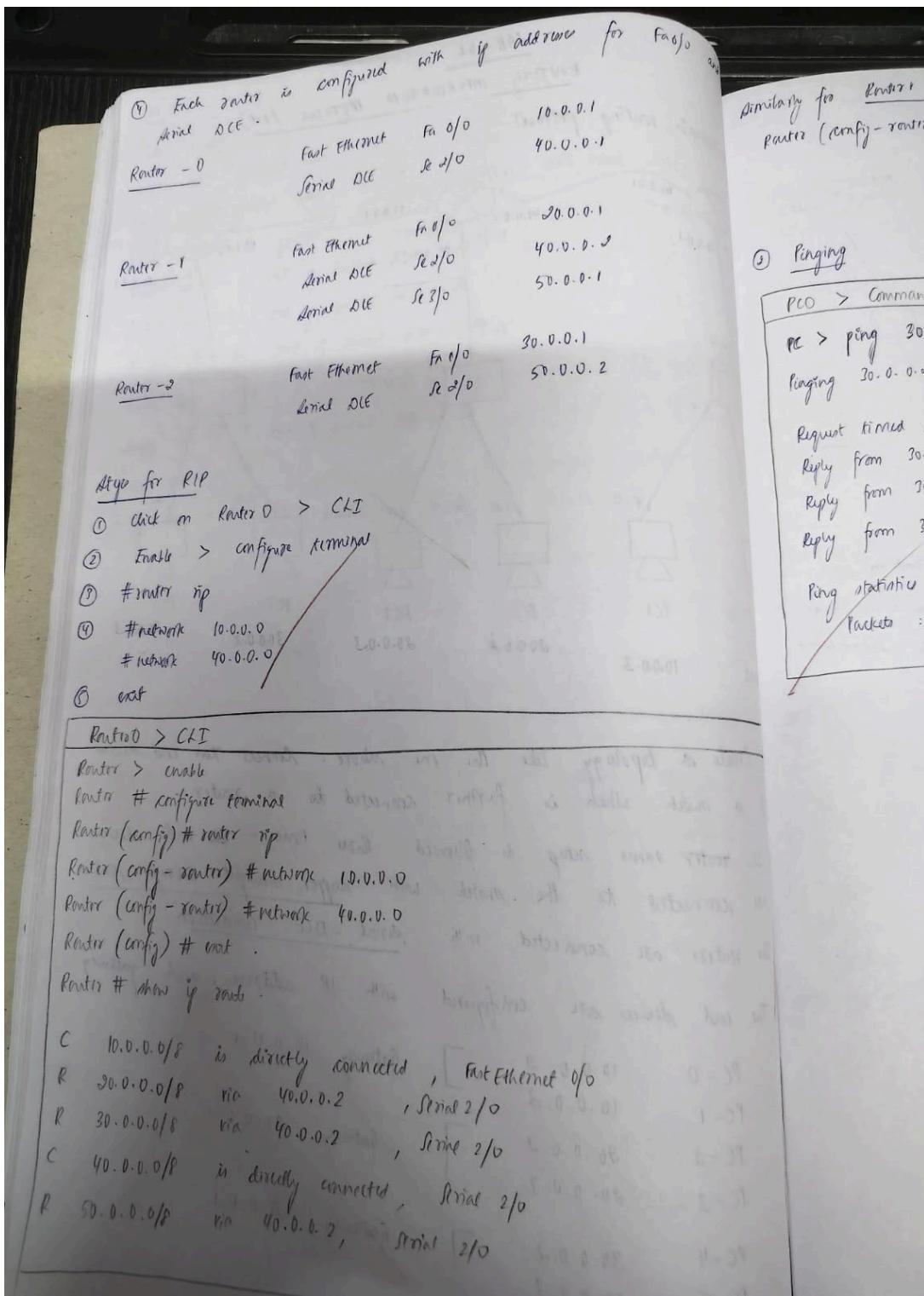
Pinging 30.0.0.2 with 32 bytes of data:

Request timed out.
Reply from 30.0.0.2: bytes=32 time=9ms TTL=125
Reply from 30.0.0.2: bytes=32 time=8ms TTL=125
Reply from 30.0.0.2: bytes=32 time=10ms TTL=125

Ping statistics for 30.0.0.2:
  Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 8ms, Maximum = 10ms, Average = 9ms

PC>
```

Observation:



Similarly for Router

Router (config-router) # network 20.0.0.0
network 40.0.0.0
network 50.0.0.0

② Pinging

PC > Command prompt

PC > ping 30.0.0.2

Pinging 30.0.0.2 with 32 bytes of data :

Request timed out.

Reply from 30.0.0.2 : bytes = 32 time = 9ms TTL = 125

Reply from 30.0.0.2 : bytes = 32 time = 8ms TTL = 125

Reply from 30.0.0.2 : bytes = 32 time = 10ms TTL = 125

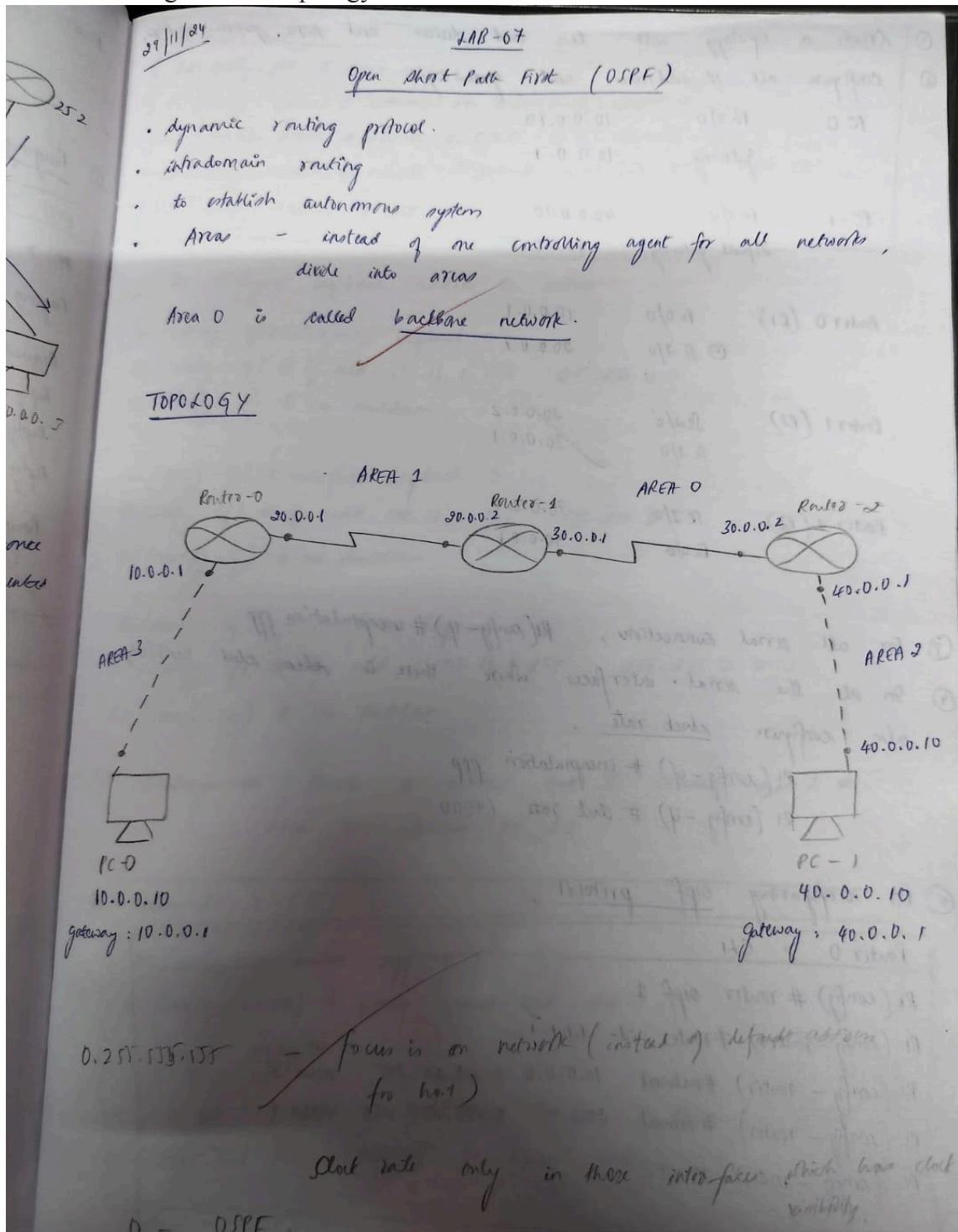
Ping statistics :

Packets : Sent = 4, Received = 3, Lost = 1 (25.0% loss)

Program 6

Aim of the program:
Configure OSPF routing protocol

Procedure along with the topology:



① Create a topology with two end devices and three generic routers.

② Configure all IP addresses and gateway.

Router 0 Fa 0/0 10.0.0.10
PC 0 Fa 0/0 10.0.0.1
 Gateway

Router 1 Fa 0/0 40.0.0.10
PC -1 Fa 0/0 40.0.0.1
 Default gateway

Router 0 (R1) Fa 0/0 10.0.0.1
 Serial 2/0 20.0.0.1

Router 1 (R2) Serial 2/0 30.0.0.2
 Serial 1/0 30.0.0.1

Router 2 (R3) Serial 1/0 30.0.0.2
 Fa 0/0 40.0.0.1

Router 2 >
R2 (config)-Y
R2 (config)-Z
R2 (config)-
R2 (config)-
R2 (config)-

⑥ To config

R1 (config)-
R1 (config)-
R1 (config)-

R2 (config)-
R2 (config)-
R2 (config)

R3 (config)

R3 (config)

R3 (config)

⑦ Router

⑦ For all serial connection, R1(config-y) # encapsulation ppp

⑧ In all the serial interfaces where there is ~~area~~ clock visibility, also configure clock rate.

R1(config-if) # encapsulation ppp

R1(config-y) # clock rate 64000

⑨ For configuring ospf protocol,

Router 0 > CLI

R1(config) # router ospf 1

R1(config-router) # router-id 1.1.1.1

R1(config-router) # network 10.0.0.0

R1(config-router) # network 20.0.0.0 0.255.255.255 area 3

R1(config-router) # exit

In 200

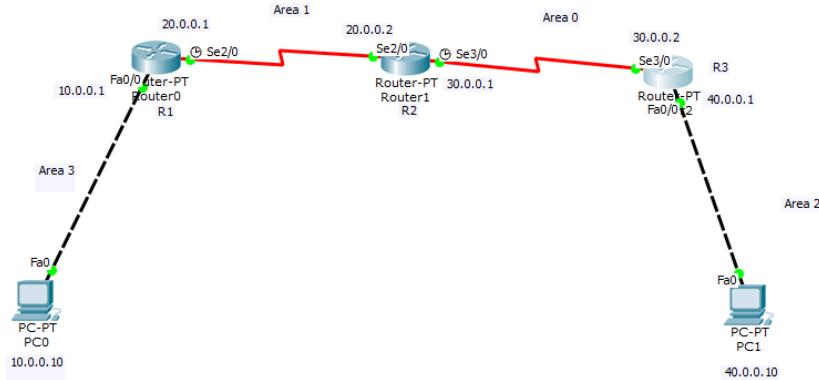
R1

R1

R1

*

Screen shots/ output :



Router0

Physical Config CLI

IOS Command Line Interface

```

Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface fastethernet 0/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface serial 2/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to down
Router(config-if)#clock rate 64000
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
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)#exit
Router(config)#
00:43:24: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Serial2/0 from LOADING to
FULL, Loading Done
    
```

Copy Paste

Router0

Physical Config CLI

IOS Command Line Interface

```

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
     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, 00:01:28, Serial2/0
O  IA 40.0.0.0/8 [110/129] via 20.0.0.2, 00:00:13, Serial2/0
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
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 add 172.16.1.252 255.255.0.0
Router(config-if)#no shutdown
Router(config-if)#

```

Copy Paste

PC0

Physical Config Desktop Custom Interface

Command Prompt

```

Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.10: bytes=32 time=8ms TTL=125
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Reply from 40.0.0.10: bytes=32 time=2ms TTL=125

Ping statistics for 40.0.0.10:
  Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 8ms, Average = 5ms

PC>

```

Observation:

Router 2 > CLI

```
R2(config)# router ospf 1
R2(config-router)# network 0.0.0.0 0.255.255.255 area 1
R2(config-router)# network 0.0.0.0 0.255.255.255 area 0
R2(config-router)# network 0.0.0.0 0.255.255.255 area 0
R2(config-router) # exit
```

⑥ To reconfigure loopback address to router -

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

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

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

⑦ Routing to Create a virtual link to connect area 1 to area 0 between R1, R2.

In *Router R1 > CLI*

```
R1(config)# router ospf 1
R1(config-router)# area1 virtual-link 0.0.0.0
R1(config-router)#
* 1:03:52 : / OSPF-5-ADSG: Process 1, Nbr 0.0.0.0 on OSPF-VL0 from LOADING to FULL Loading Done.
```

In router R2 > CLI

R2 (config) -> [router] #

R2 (config) -> [router] # area 1 virtual-link 1.1.1.1

R2 (config) -> [router] # end

01:07:50 : 7.057F-5 - ADTUG : Processor, Mhz 1.1.1.1 on U8P1

90/12

To

① R3 # Show ip route of router R3

R3 > CLI

R3 # show ip route

O IA 30.0.0.0/8 [110/128] via 30.0.0.1 00:01:50, Area 3/
C 30.0.0.0/8 is directly connected, Fast Ethernet 0/0

O IA 10.0.0.0/8 [110/128] via 30.0.0.1 00:01:50 Area 3/

C 30.0.0.0/8 is directly connected, serial 3/0.

C 172.16.0.0/16 is directly connected, loopback 0

② Check connectivity

PC0 > Command prompt

PC > ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data:

Request timed out.

Reply from 40.0.0.10 :

bytes = 32 time = 8ms TTR = 125

Reply from 40.0.0.10 :

bytes = 32 time = 7ms TTR = 125

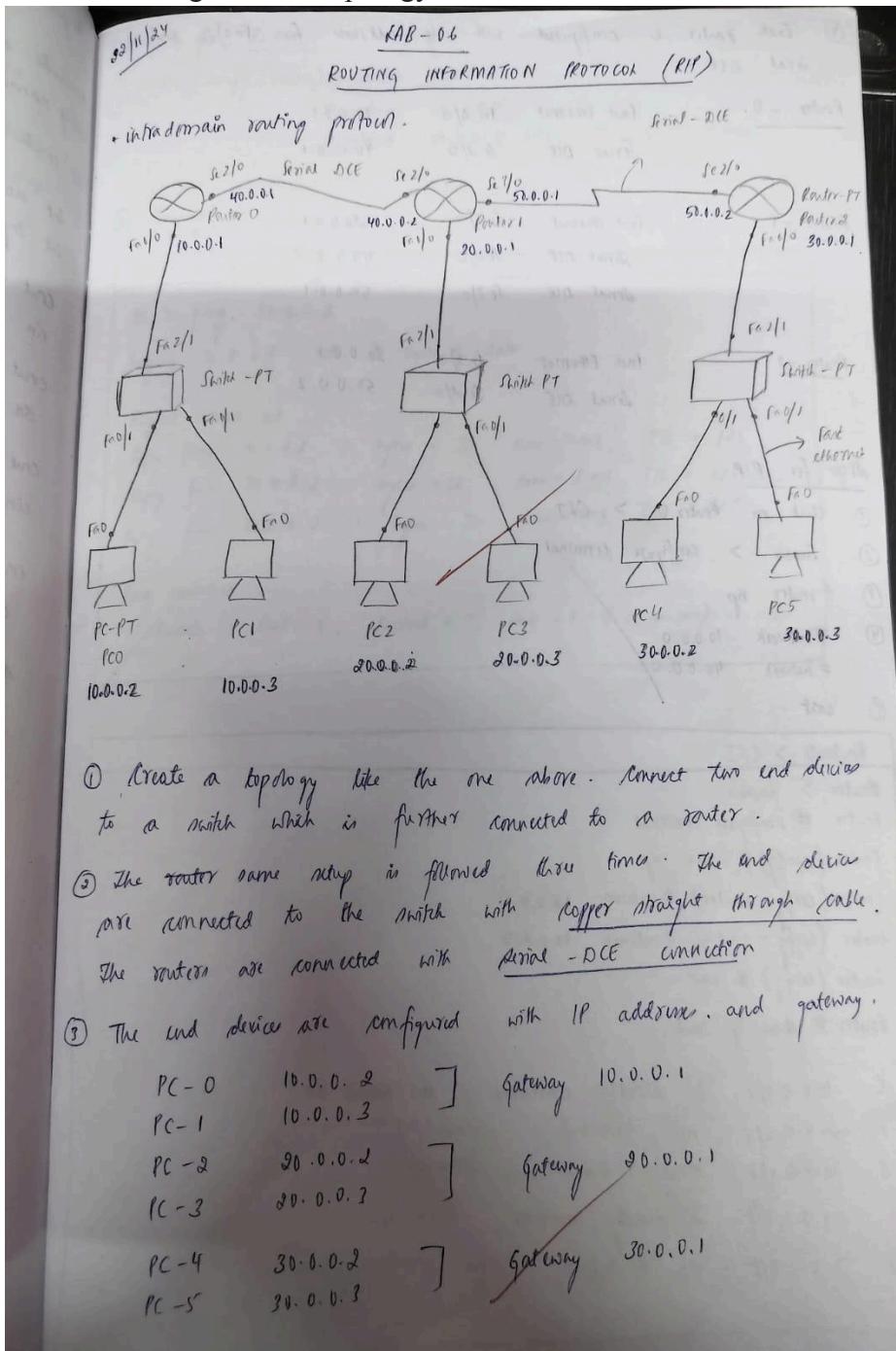
Reply from 40.0.0.10 :

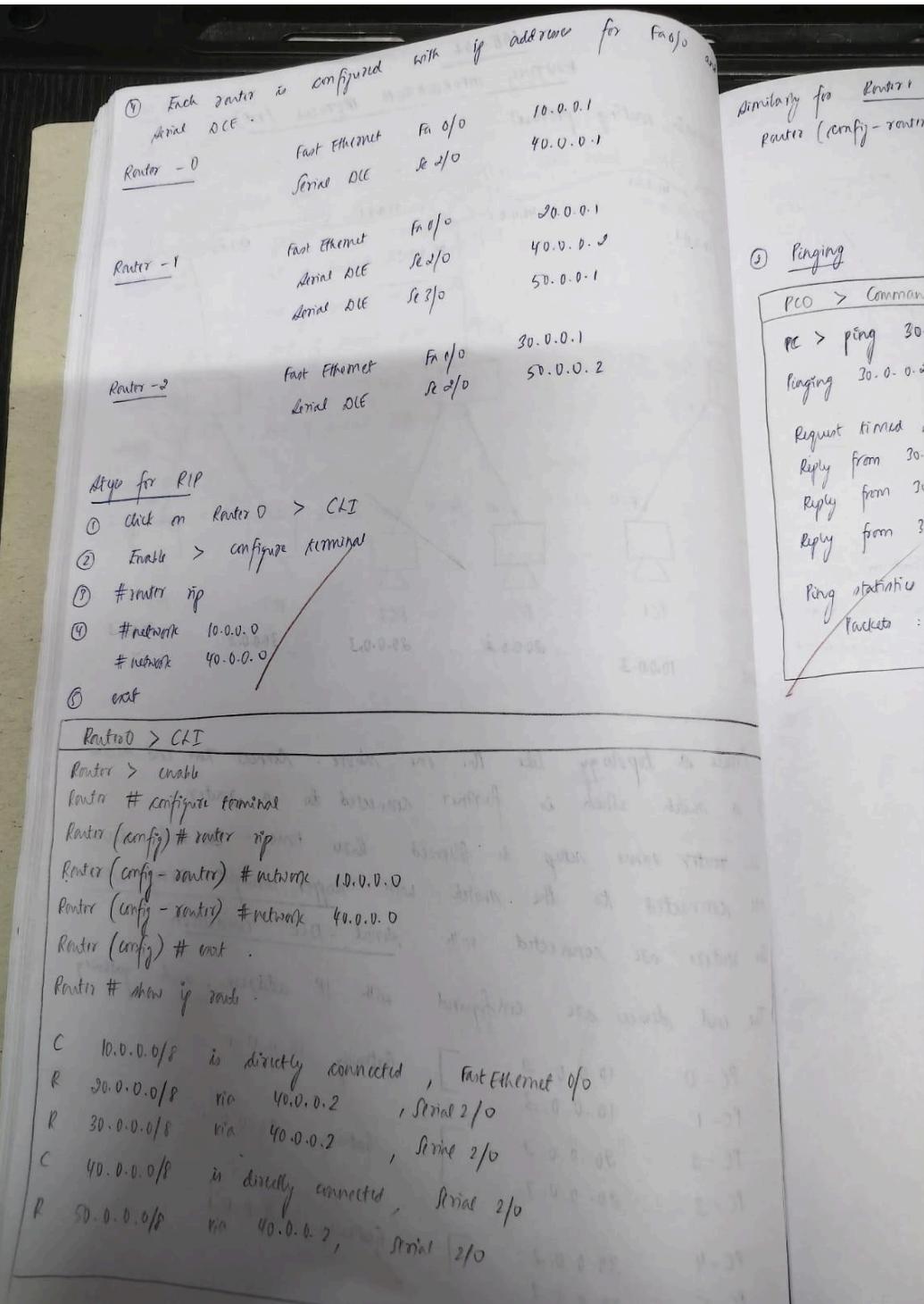
bytes = 32 time = 8ms TTR = 125

Program 7

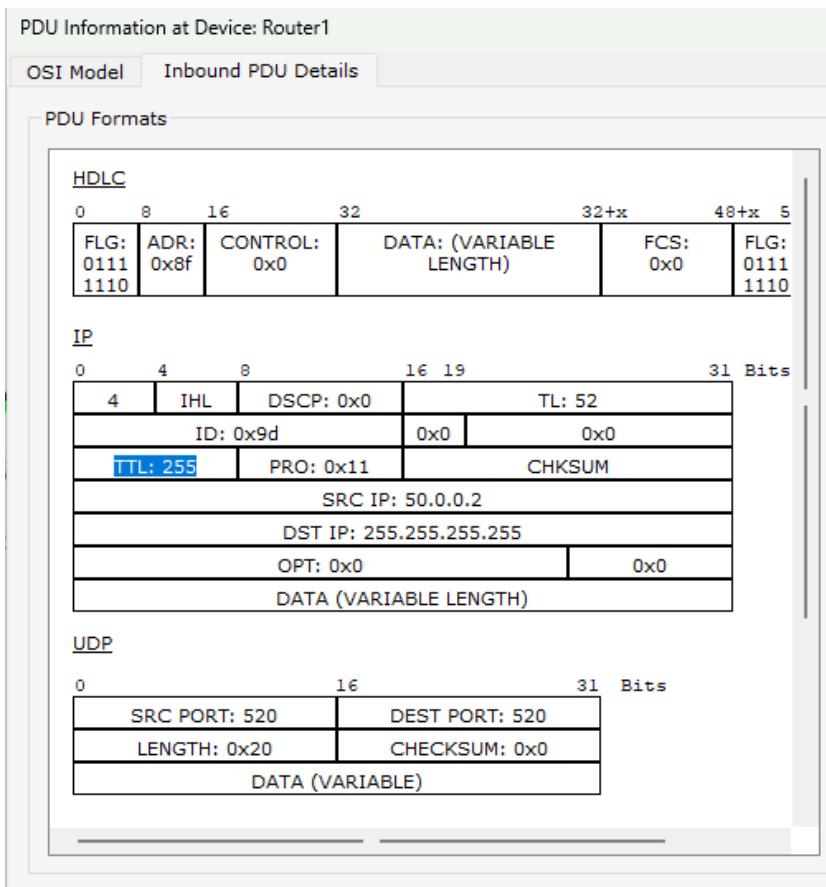
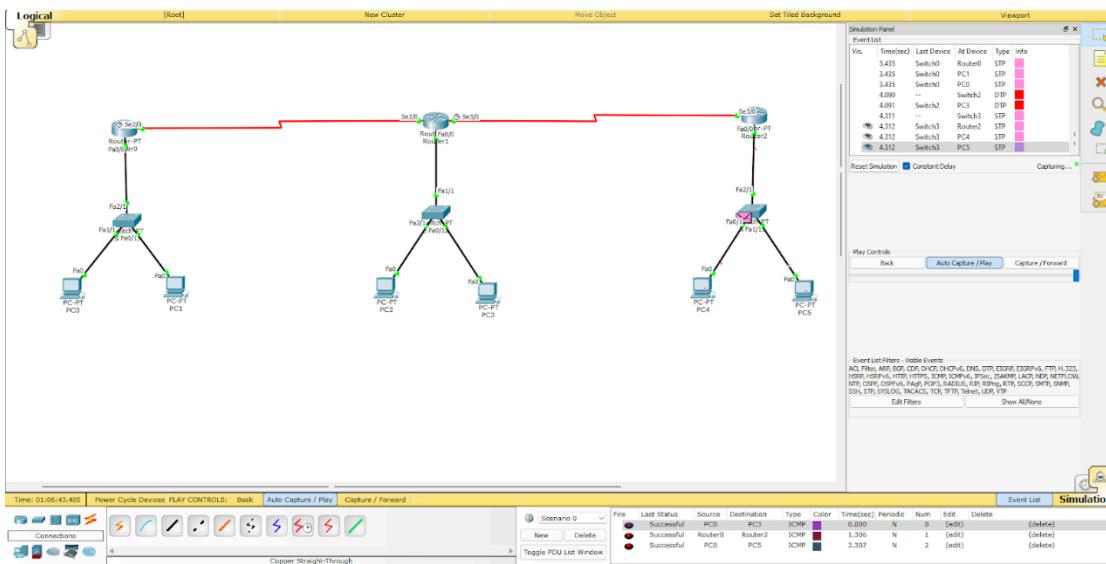
Aim of the program:
Demonstrate the TTL/ Life of a Packet

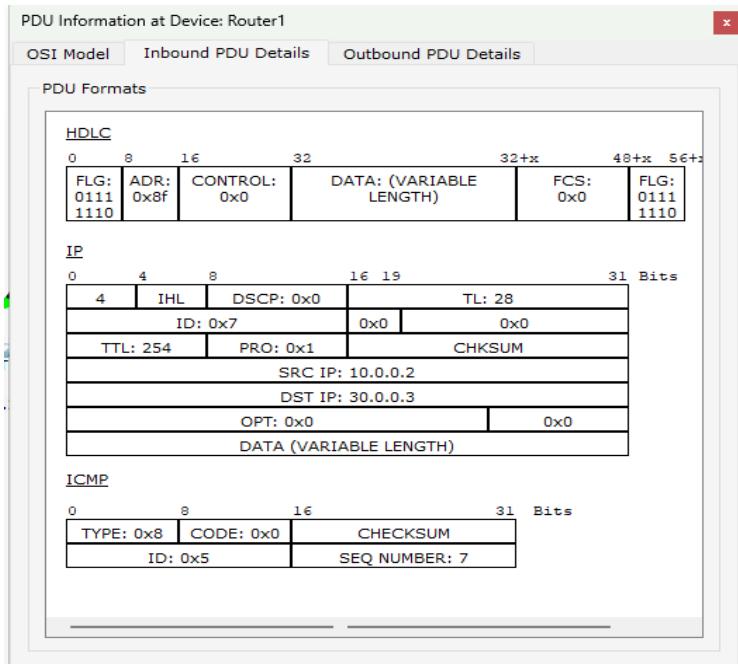
Procedure along with the topology



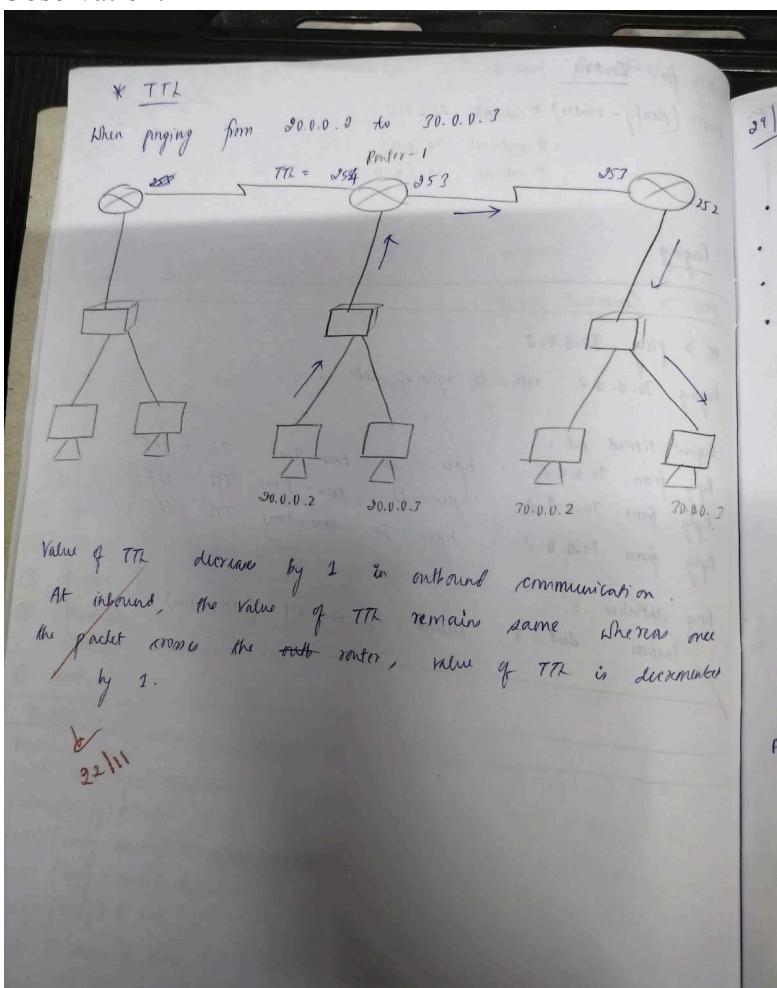


Screen shots/ output :





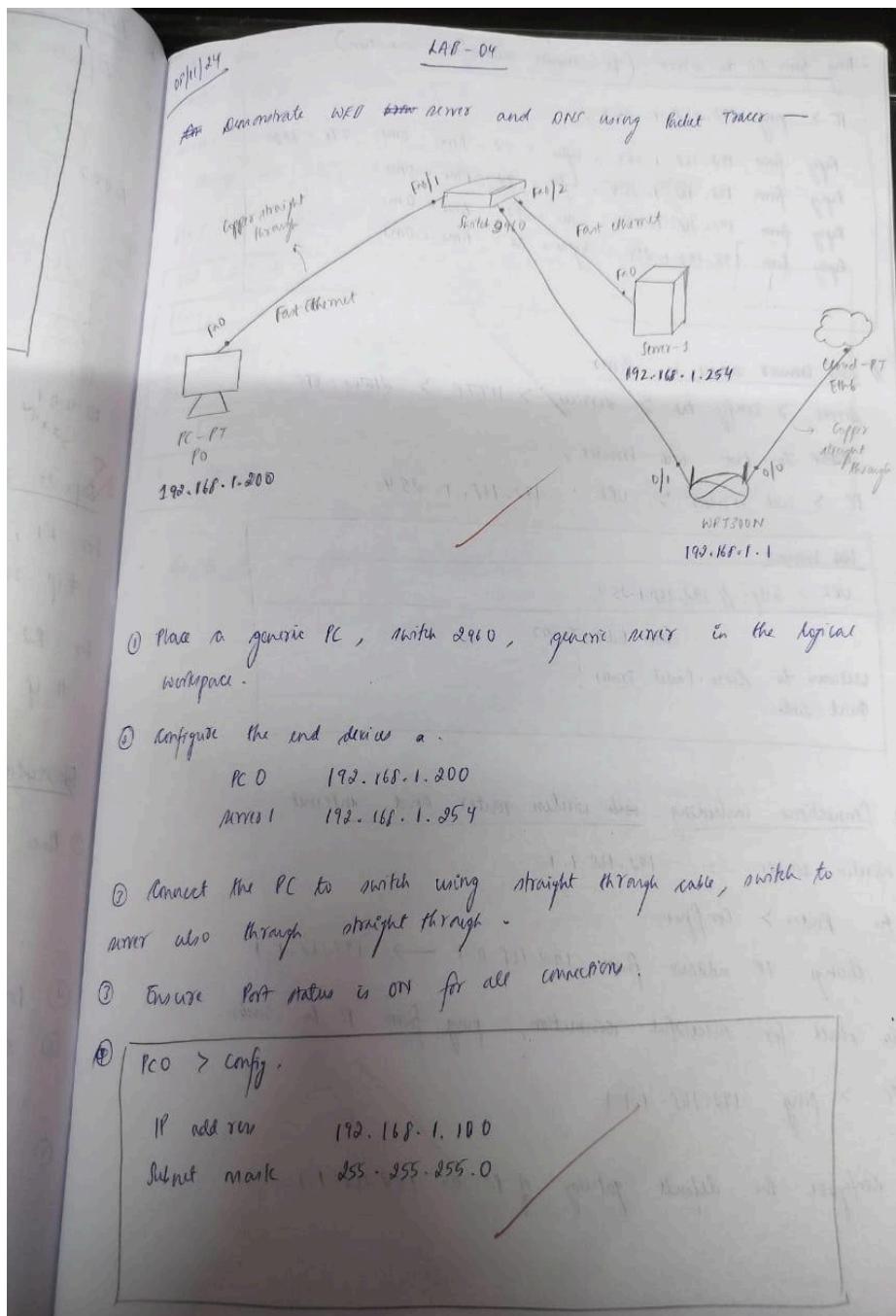
Observation:



Program 8

Aim of the program:
Configure Web Server, DNS within a LAN.

Procedure along with the topology:



• Ping from PC to router. (to ensure successful connection)

```
PC > ping 192.168.1.254.
Reply from 192.168.1.254: bytes = 32 time = 0ms TTX = 128
Reply from 192.168.1.254: bytes = 32 time = 0ms
Reply from 192.168.1.254: bytes = 32 time = 0ms
Reply from 192.168.1.254: bytes = 32 time = 0ms
Reply from 192.168.1.254: bytes = 32 time = 0ms
```

// Web browser service in Router

Router > config tab > Services > HTTP > Status: ON

Keep to test web browser,

PC > web browser > URL : 192.168.1.254.

Web browser

URL : http://192.168.1.254.

Cisco Packet Tracer

Welcome to Cisco Packet Tracer!

Quick Links:

② Connections including web wireless router and internet

Wireless router : 192.168.1.1

In Router > Configure.

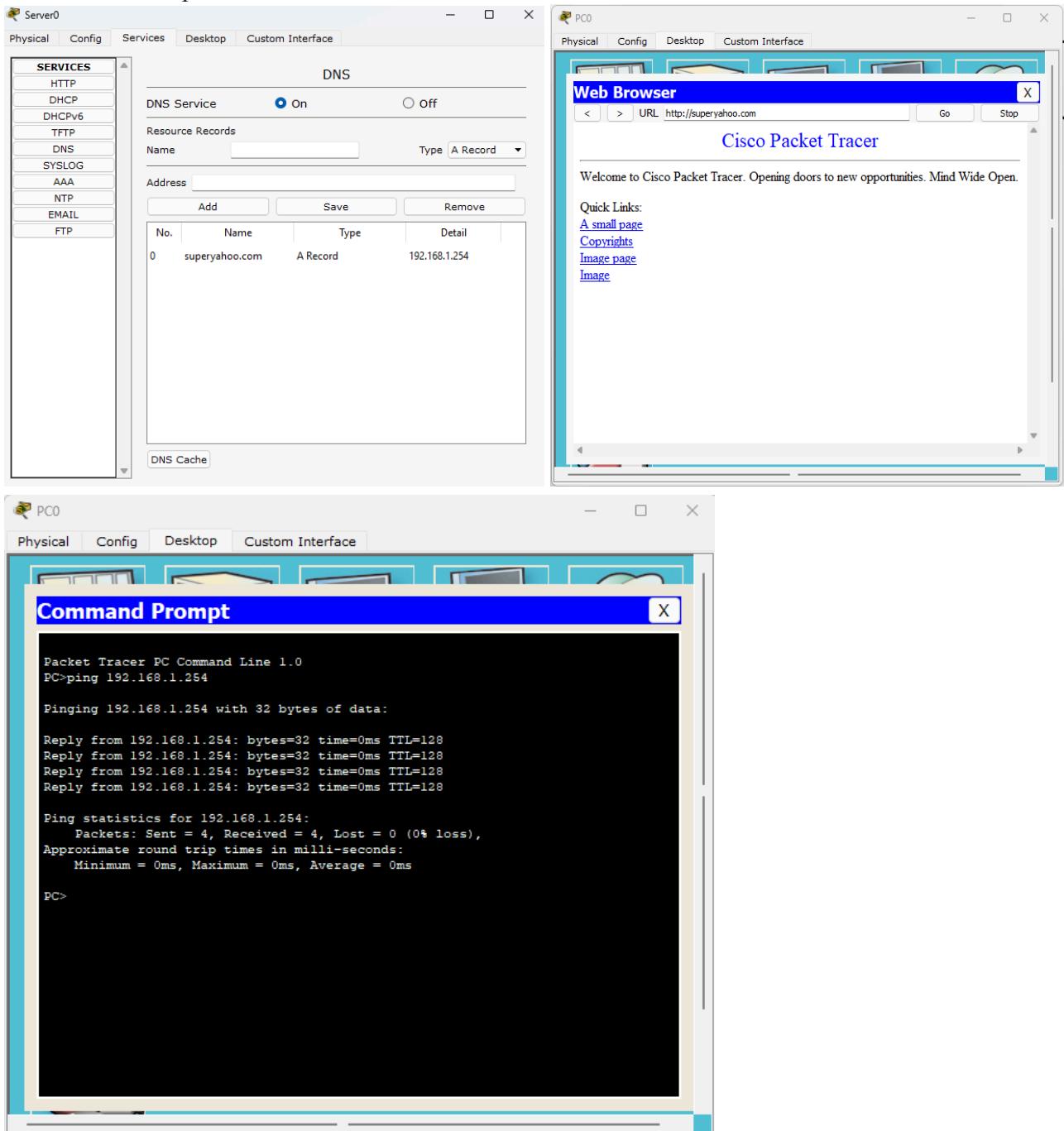
Change IP address from 192.168.0.1 → 192.168.1.1.

To check for successful connection, ping from PC to router.

PC > ping 192.168.1.1.1

• Configure the default gateway of PC as 192.168.1.1

Screen shots/ output :



Observation:

// To view a different website called superyahoo.com
⇒ setup a DNS entry

Server > Services > ~~HTTP~~ HTTP.
change the html code as required for index.html.

DNS > Services: ON.

192.168.1.254

Config DNS Services ON

Name: superyahoo.com Add

1. superyahoo.com A Record 192.168.1.254

• Go to PC > ~~IP~~ IP Configuration.

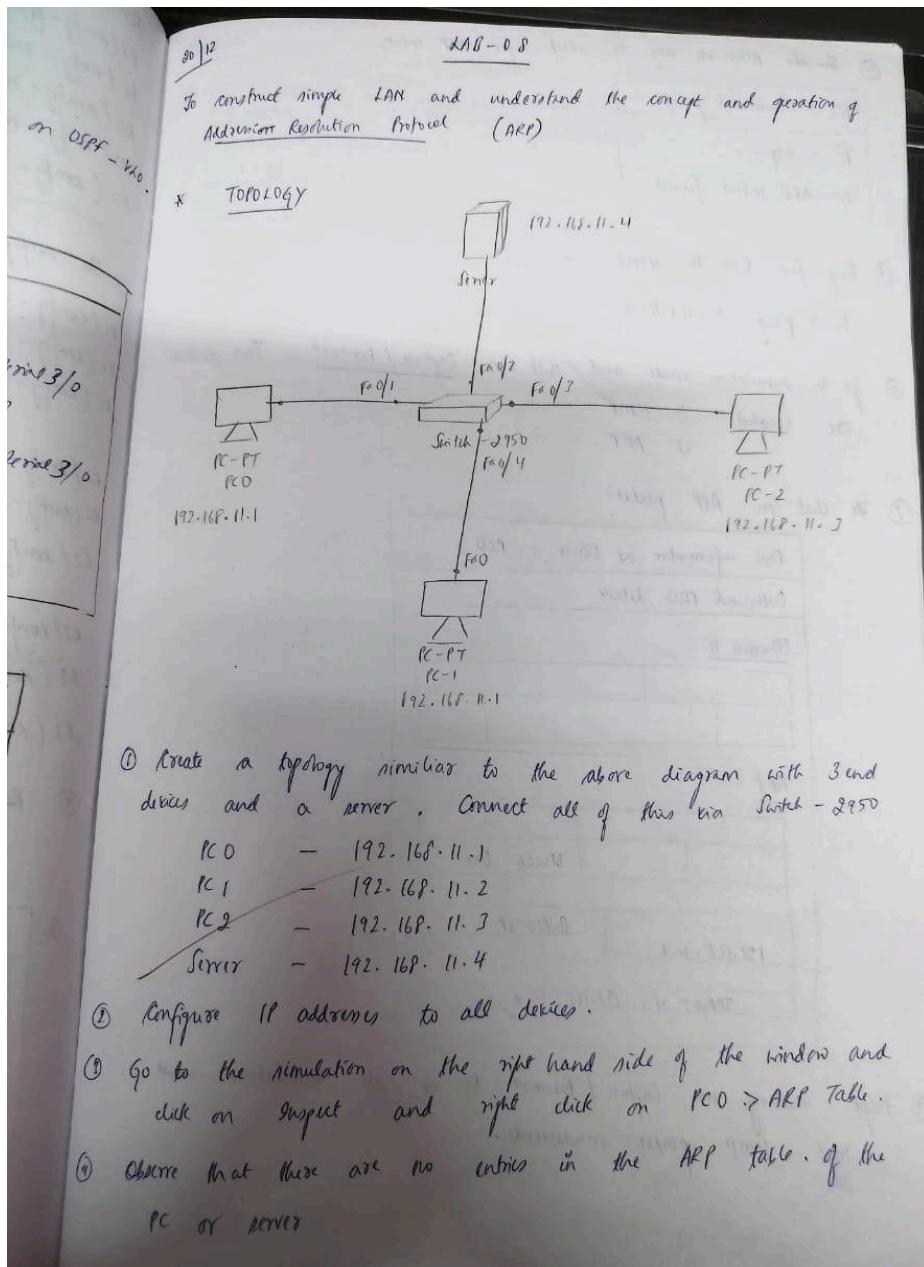
0.0.0.254 static route
192.168.1.254 gateway
0.0.0.254 static route
192.168.1.254 gateway

Program 9

Aim of the program:

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

Procedure along with the topology:



⑤ To do alternate way to check for APP entries,

(a) PCD > command prompt

PCD > app -a

NO APP entries found

Observations

APP 1
IP address
192.168.

⑥ ping from PCD to RMM

PCD > ping 192.168.11.4

⑦ go to simulation mode and click on Capture / Forward. Then ^{packets} _{are created}

- 1) ICMP
- 2) APP

⑧ To click on APP packet

(a) PDU information at Device : PCD

Outbound PDU details

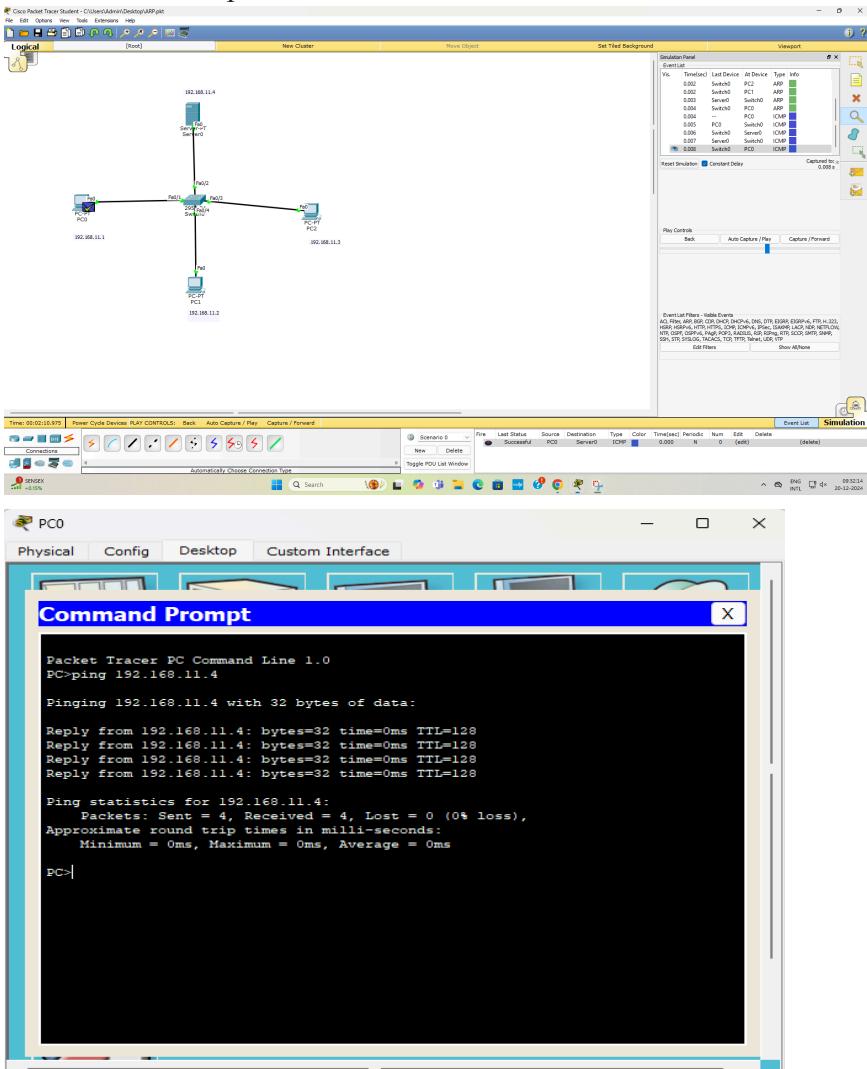
Ethernet II

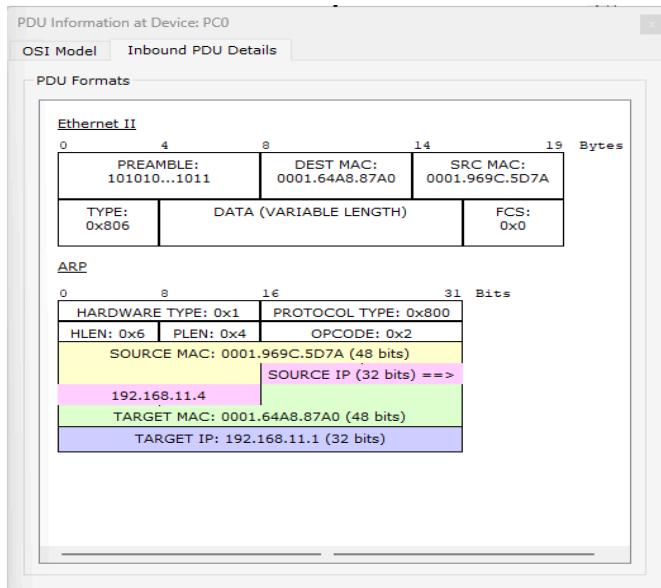
APP

	OPCODE: 0x1
192.168.11.1	SOURCE IP
TARGET IP: 192.168.11.4	

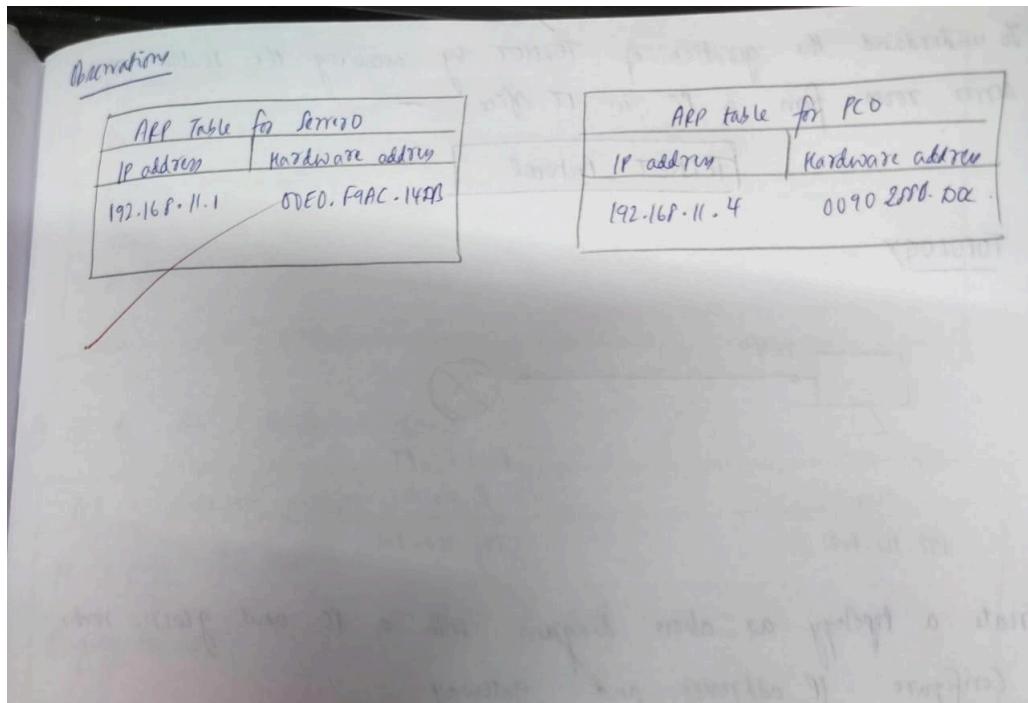
⑨ kept clicking on Capture / Forward to see all simulations and track ICMP packet movements.

Screen shots/ output :





Observation:

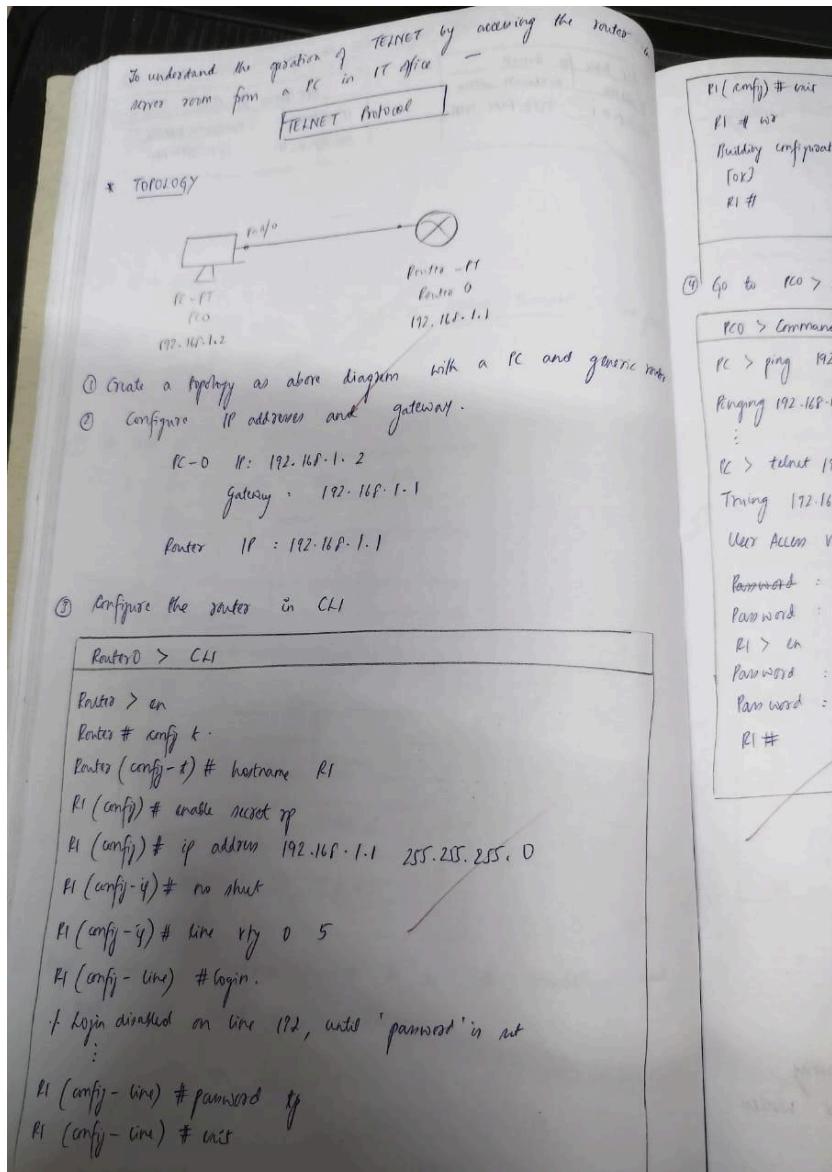


Program 10

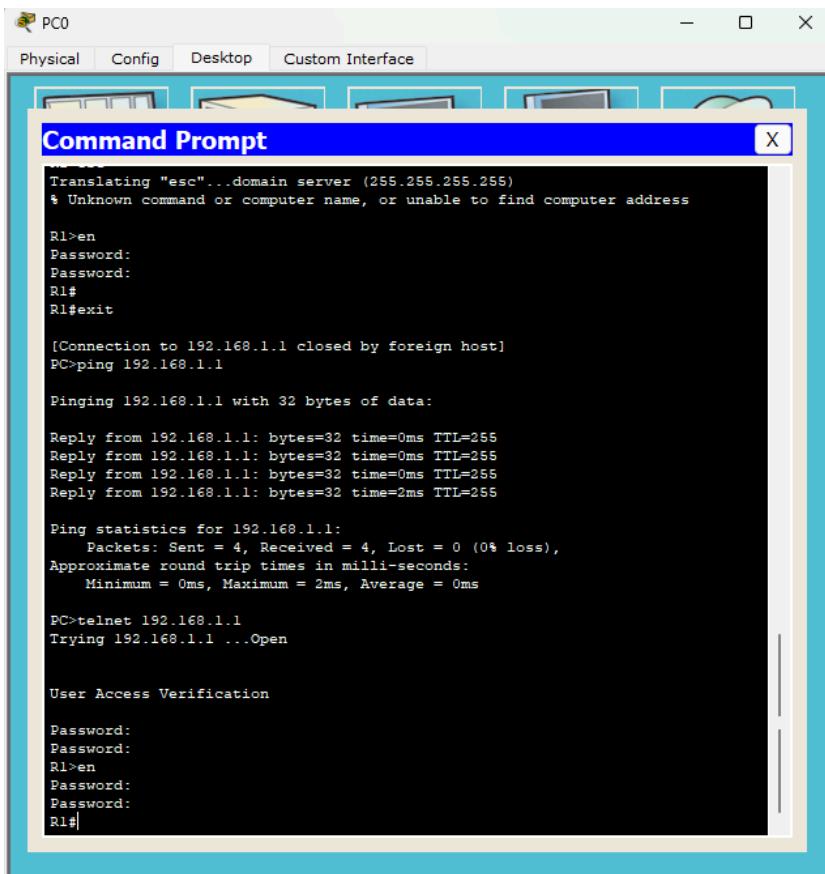
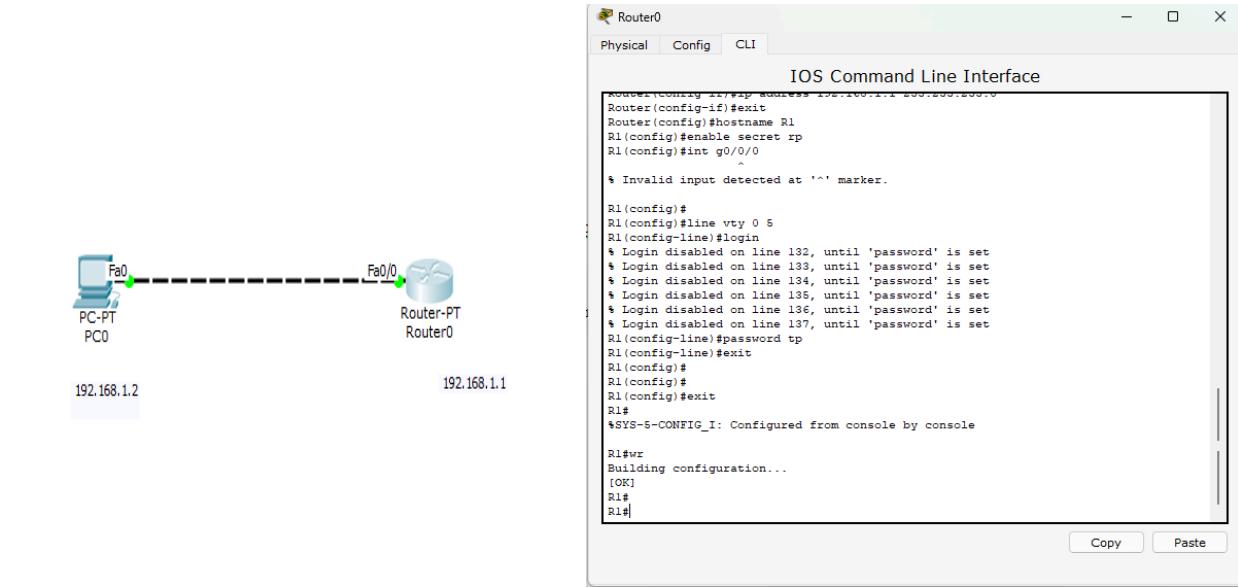
Aim of the program:

To understand the operation of TELNET by accessing the router in server room from a PC in IT office.

Procedure along with the topology:



Screen shots/ output :



Observation:

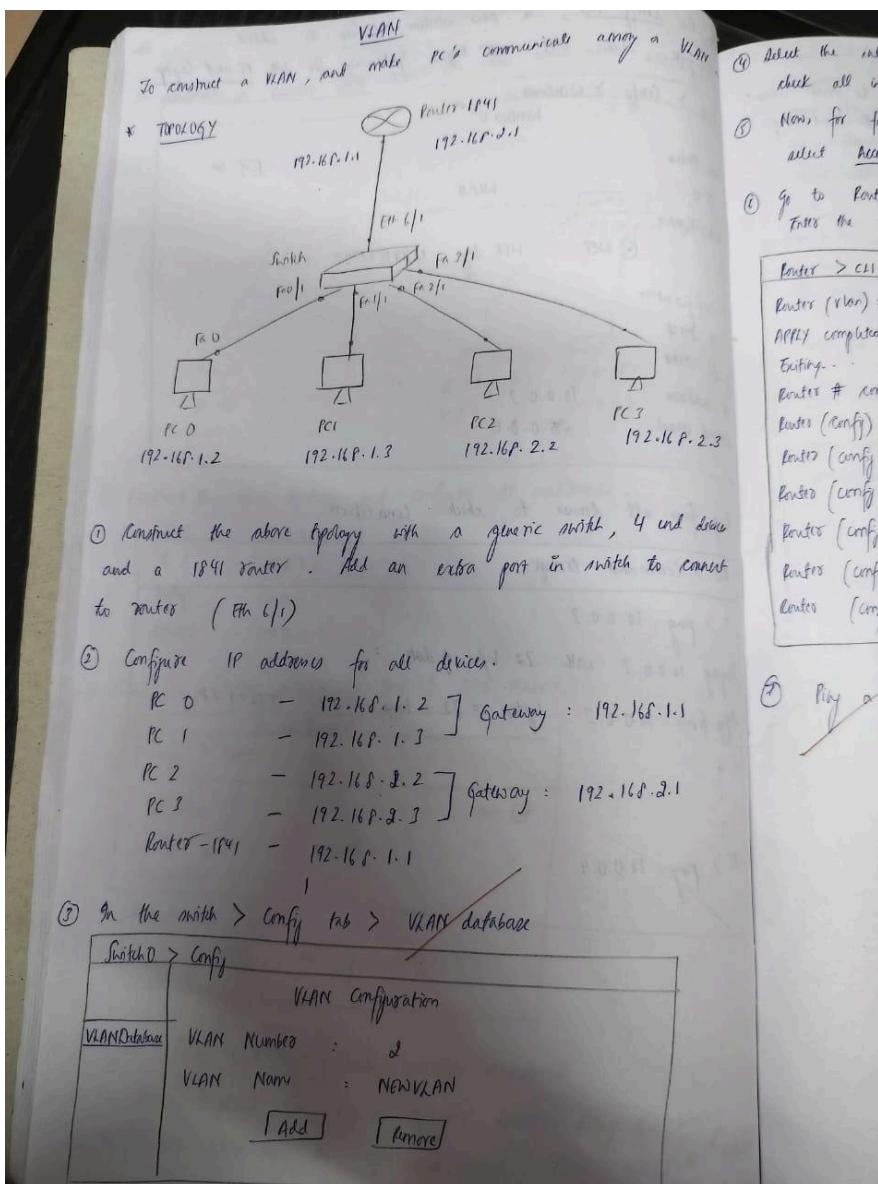
```
R1(config)# write  
R1# Building configuration...  
[OK]  
R1#  
  
⑨ Go to PC > Command prompt  
  
PC > Command prompt  
PC > ping 192.168.1.1  
Pinging 192.168.1.1 with 32 bytes of data:  
:  
PC > telnet 192.168.1.1  
Trying 192.168.1.1 ... Open  
User Access Verification  
Password : tp  
Password : tp  
R1> en  
Password : tp  
Password : tp  
R1#
```

Program 11

Aim of the program:

To construct a VLAN and make the PC's communicate among a VLAN

Procedure along with the topology:



- ④ Select the interface - ethernet 0/1 and make it trunk and check all in VLAN field.
- ⑤ Now for fastethernet 0/1 and fastethernet 1/1 select Access and VLAN : 2
- ⑥ Go to Router > Config tab > VLAN Database. Find the name and number of VLAN created.

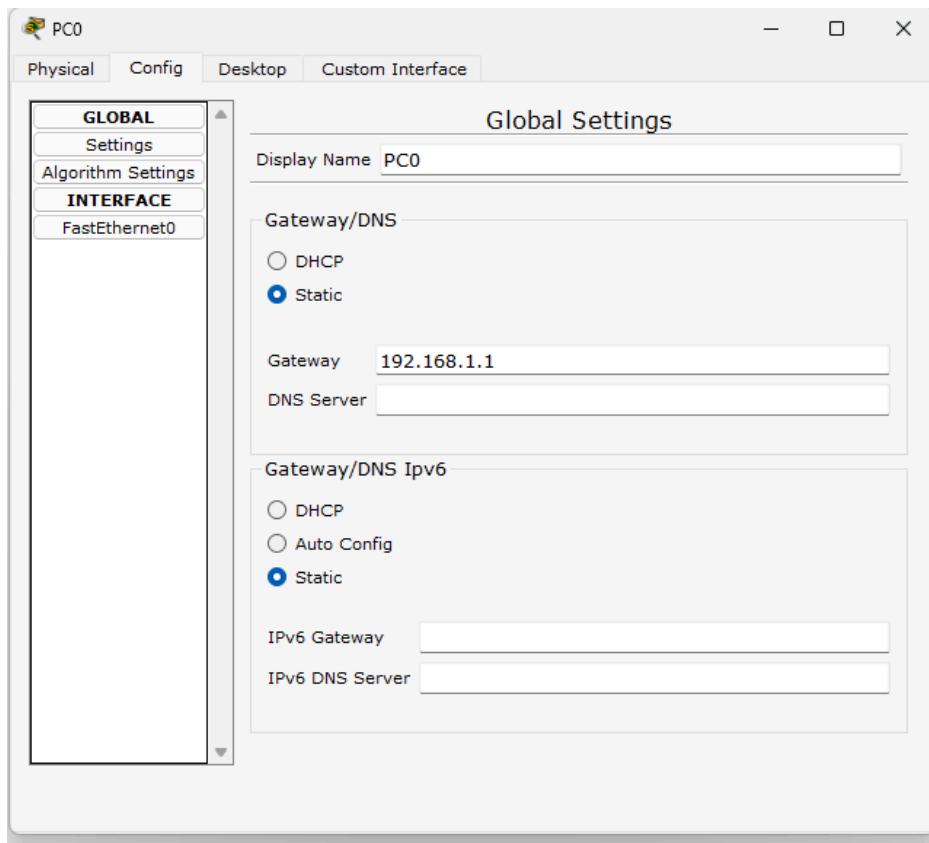
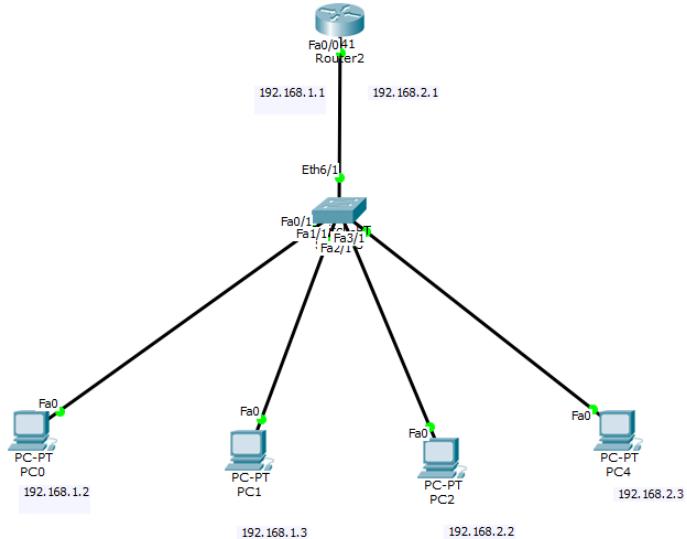
```

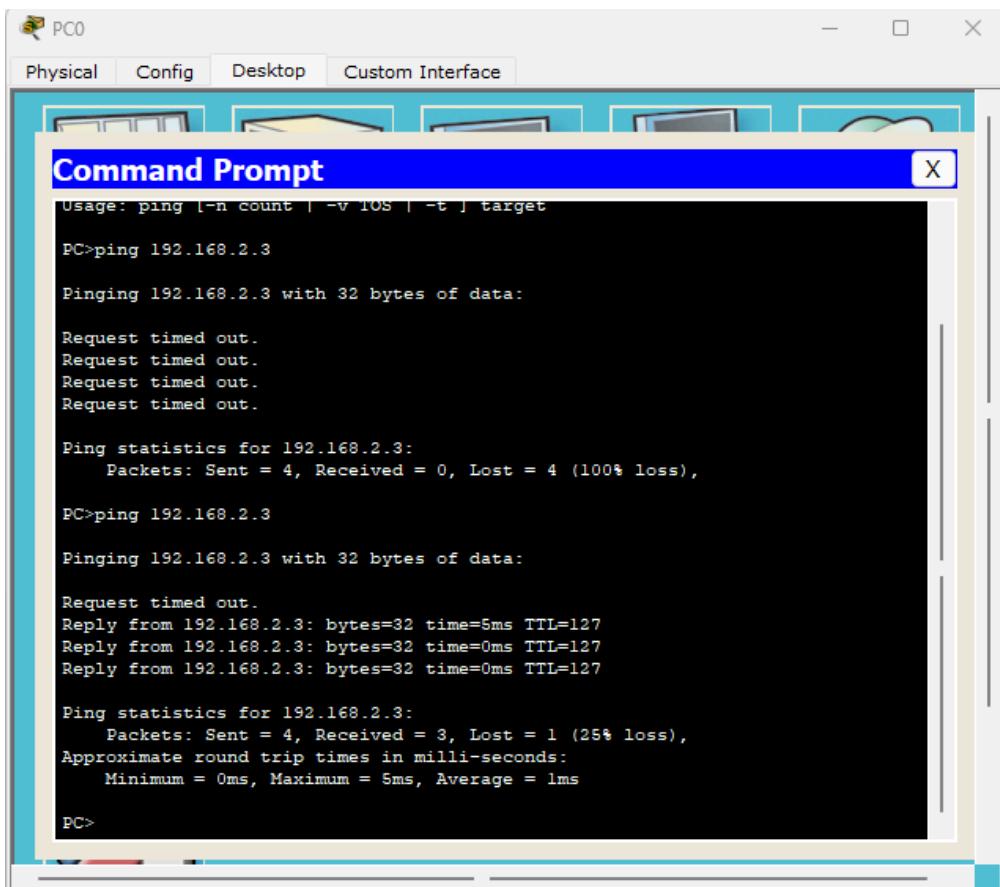
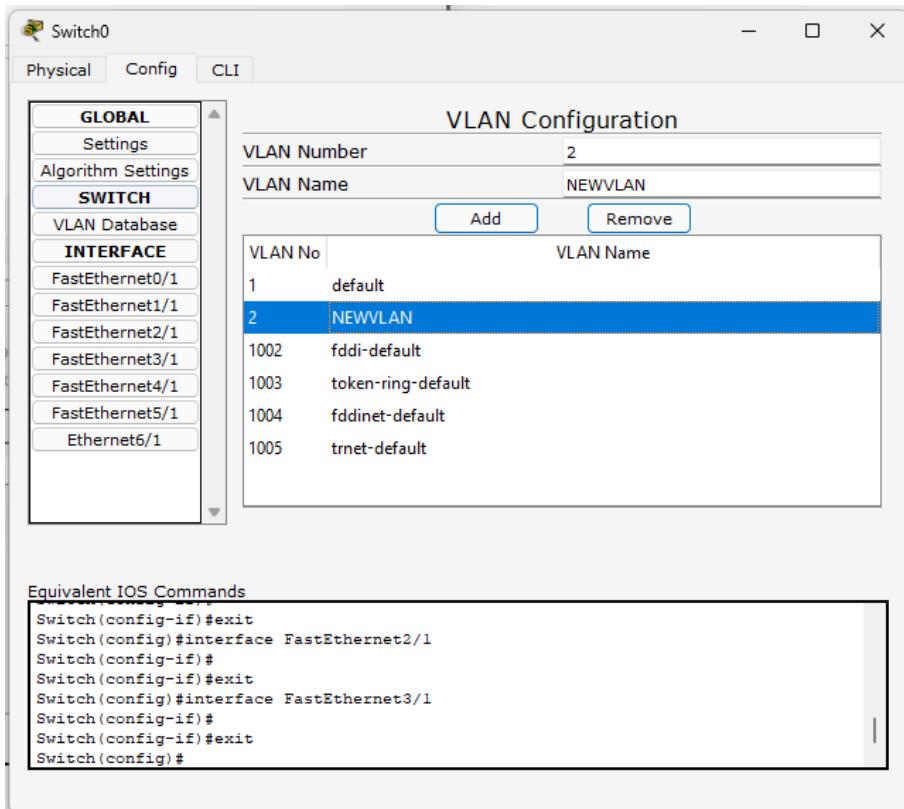
Router > c21
Router (vlan) # exit
Memory completed.
Exiting.
Router # config t
Router (config) # interface fastEthernet 0/0.1
Router (config-subif) # encapsulation dot1Q 2
Router (config-subif) # ip address 192.168.2.1 255.255.255.0
Router (config-subif) # no shutdown
Router (config-subif) # exit
Router (config) # exit

```

- ⑦ Play a message from one device to another.

Screen shots/ output :



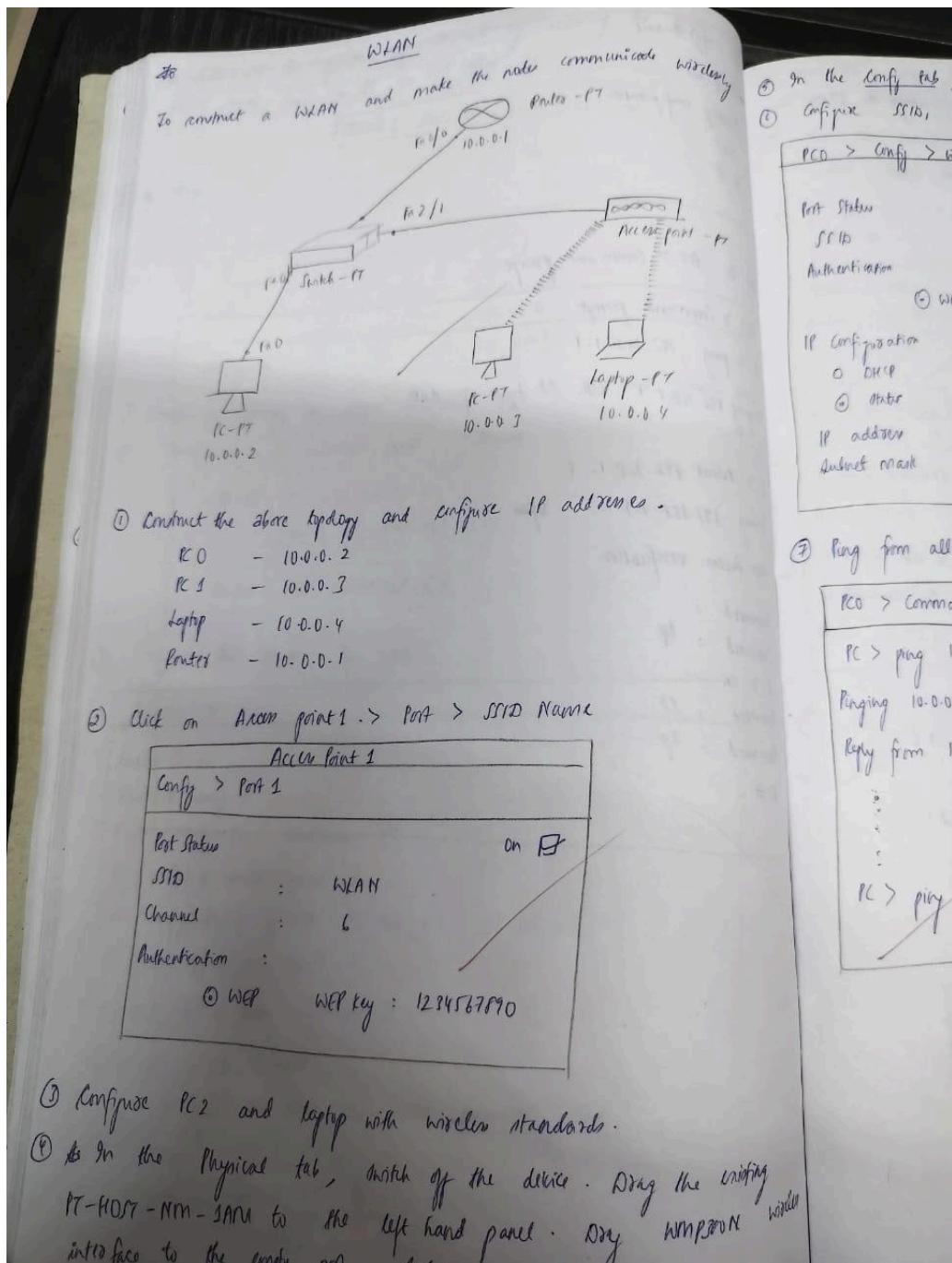


Program 12

Aim of the program:

To construct a WLAN and make the nodes communicate wirelessly

Procedure along with the topology:



- ⑥ In the Config Tab, a new wireless interface is added.
 ⑦ Configure SSID, WEP, WEP key, gateway for both PC and Laptop.

PC0 > Config > Wireless

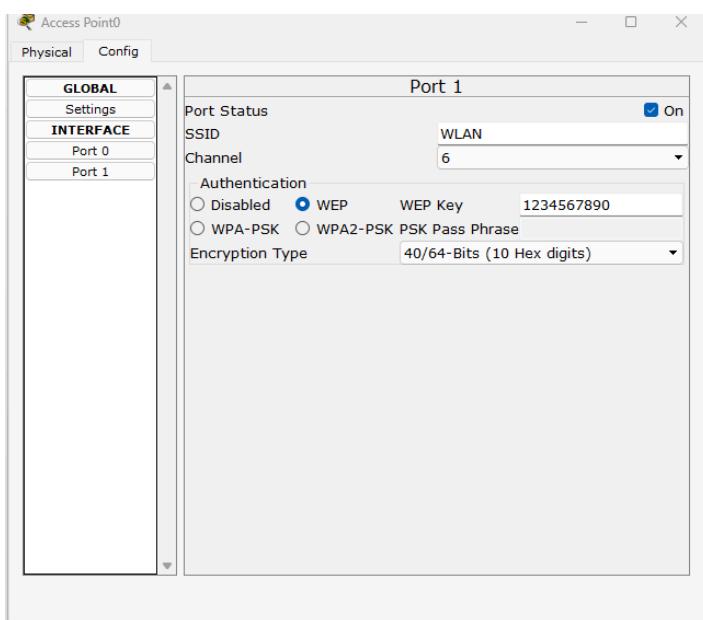
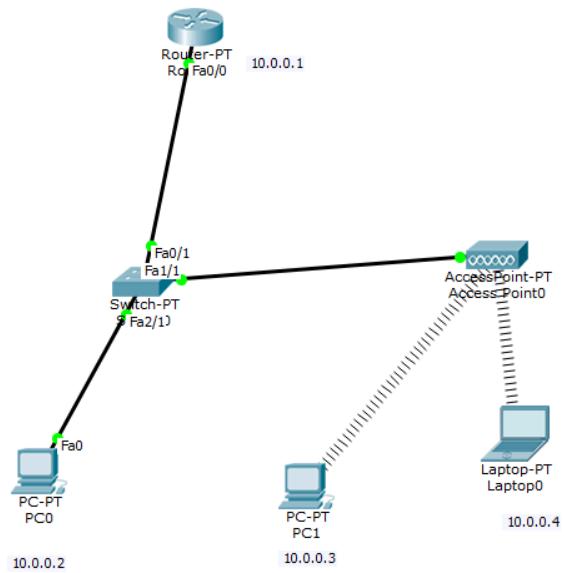
Wireless 0	
Port Status	<input checked="" type="checkbox"/> On
SSID	WLAN
Authentication	
④ WEP	WEP Key 1234567890
IP Configuration	
① Static	② DHCP
IP address	10.0.0.3
Subnet mask	255.0.0.0

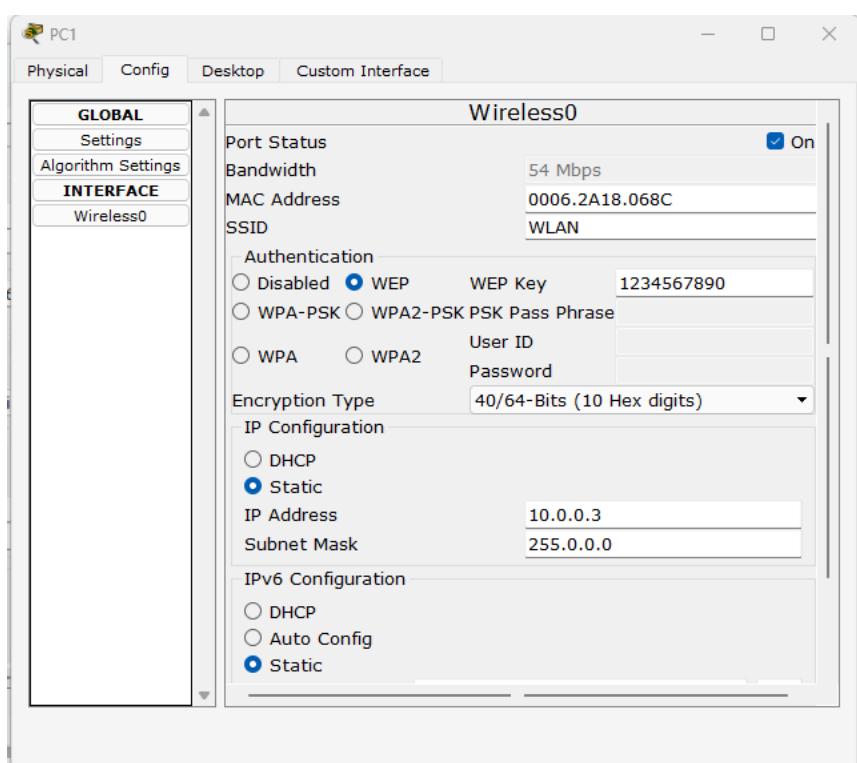
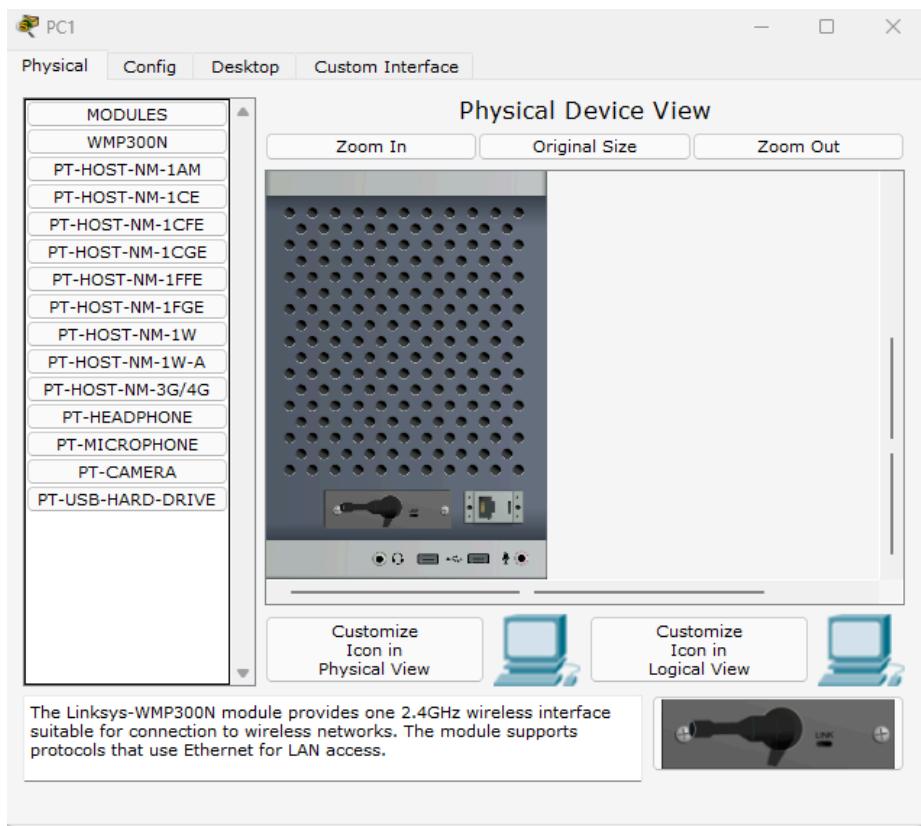
- ⑧ Ping from all devices to check connection.

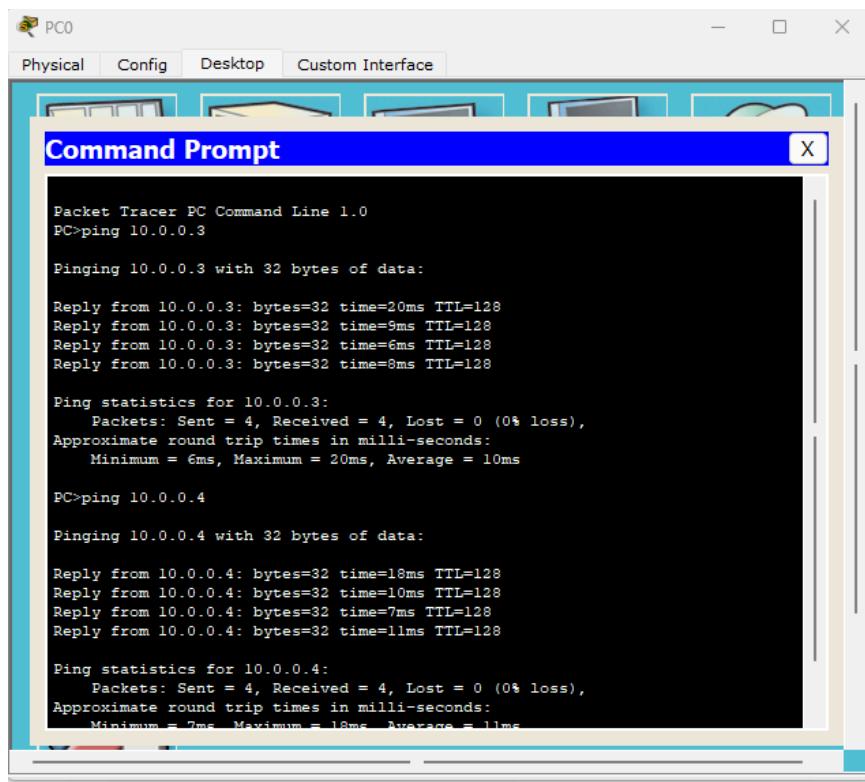
PC0 > Command Prompt

PC > ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data:
Reply from 10.0.0.3: bytes = 32 time = 9ms TTL=128
PC > ping 10.0.0.4

Screen shots/ output :







PC0

Physical Config Desktop Custom Interface

Command Prompt

```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time=20ms TTL=128
Reply from 10.0.0.3: bytes=32 time=9ms TTL=128
Reply from 10.0.0.3: bytes=32 time=6ms TTL=128
Reply from 10.0.0.3: bytes=32 time=8ms 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 = 6ms, Maximum = 20ms, Average = 10ms

PC>ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:

Reply from 10.0.0.4: bytes=32 time=18ms TTL=128
Reply from 10.0.0.4: bytes=32 time=10ms TTL=128
Reply from 10.0.0.4: bytes=32 time=7ms TTL=128
Reply from 10.0.0.4: bytes=32 time=11ms 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 = 7ms, Maximum = 18ms, Average = 11ms
```

CYCLE-2

Program 1

Aim of the program:

Write a program for error detecting code using CRC-CCITT (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(op, "0");
    }

    // Perform XOR on the message with the selected polynomial
    for (int i = 0; i < strlen(ip); i++) {
        if (op[i] == '1') {
            for (int j = 0; j < strlen(poly); j++) {
                if (op[i + j] == poly[j])
                    op[i + j] = '0';
                else
                    op[i + j] = '1';
            }
        }
    }

    // Check for errors. Return 0 if error detected
    for (int i = 0; i < strlen(op); i++) {
        if (op[i] == '1')
            return 0;
    }
    return 1;
}

int main() {
    char ip[50], op[50], recv[50];
    char poly[] = "1000100000100001";

    cout << "Enter the input message in binary" << endl;
    cin >> ip;
```

```

    crc(ip, op, poly, 1);
    cout << "The transmitted message is: " << ip << op + strlen(ip) << endl;

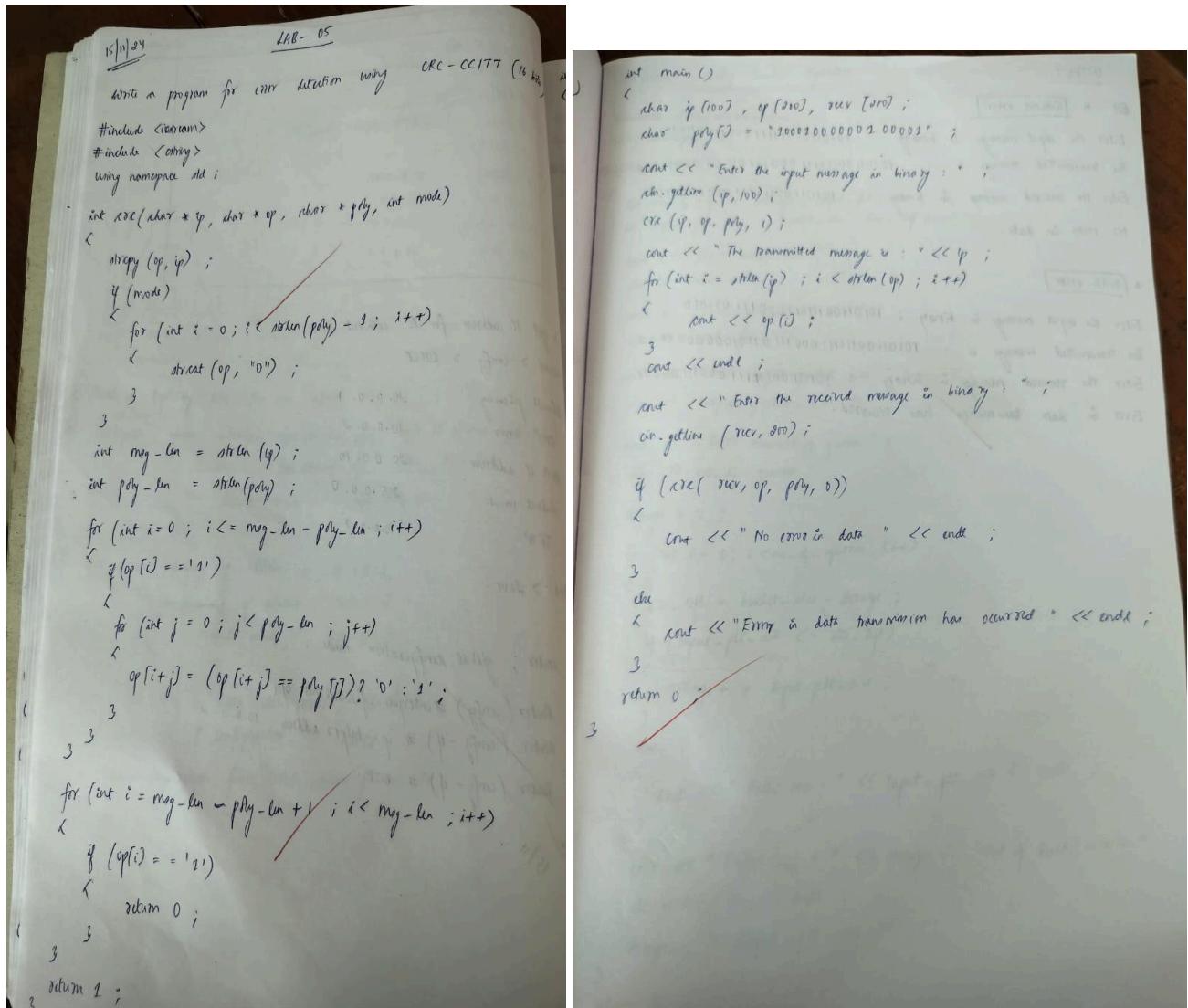
    cout << "Enter the received message in binary" << endl;
    cin >> recv;

    if (crc(recv, op, poly, 0))
        cout << "No error in data" << endl;
    else
        cout << "Error in data transmission has occurred" << endl;

    return 0;
}

```

OUTPUT:



OUTPUT

* [Without error]

Enter the input message in binary : 1010110011111000111011010
The transmitted message is : 1010110011111000111011010000000
Enter the received message in binary : 10101100111110001110110101
- No error in data.

* [With error]

Enter the input message in binary : 1010110011111000111011010
The transmitted message is : 101011001111100011101101000000000
Enter the received message in binary : 1010110011111000111011011
~~Error in data transmission has occurred.~~

Program 2

Aim of the program:

Write a program for congestion control using Leaky bucket algorithm.

Code:

```
#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;

    // 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 enters the bucket at a time
    input_pkt_size = 4;
```

```

// Number of packets that exits the bucket at a time
output_pkt_size = 1;

for (int i = 0; i < no_of_queries; i++) {
    // Space left in the bucket
    size_left = bucket_size - storage;

    if (input_pkt_size <= size_left) {
        // Update storage
        storage += input_pkt_size;
    } else {
        printf("Packet loss = %d\n", input_pkt_size);
    }

    printf("Buffer size = %d out of bucket size = %d\n",
           storage, bucket_size);

    // Decrease storage by packets exiting the bucket
    storage -= output_pkt_size;
}

return 0;
}

```

OUTPUT:

Implementation Linky Bucket Algorithm

```

#include <iostream>
using namespace std;

int main()
{
    int no_of_query, storage, output_pkt_size;
    int input_pkt_size, bucket_size, size_left;
    cout << "Enter bucket size ";
    cin >> bucket_size;
    cout << "Enter input packet size ";
    cin >> input_pkt_size;
    cout << "Enter output packet size ";
    cin >> output_pkt_size;
    cout << "Enter number of queries ";
    cin >> no_of_query;
    storage = 0;
    for (int i = 0; i < no_of_query; i++)
    {
        size_left = bucket_size - storage;
        if (input_pkt_size <= size_left)
        {
            storage += input_pkt_size;
        }
        else
        {
            cout << "Packet loss " << input_pkt_size << endl;
        }
        cout << "Buffer size = " << storage << "Out of bucket size = "
             << bucket_size << endl;
        storage -= output_pkt_size;
    }
    return 0;
}

```

OUTPUT
 Enter bucket size : 10
 Enter input packet size : 4
 Enter output packet size : 1
 Enter number of queries : 5
 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 len = 4
 Buffer size = 9 out of bucket size = 10
 Packet len = 4
 Buffer size = 8 out of bucket size = 10

Program 3

Aim of the program:

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

Code:

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()
```

OUTPUT:

The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left has a tree view with 'CN LAB' expanded, showing files like ClientTCP.py, ServerTCP.py, VLAN.pkt, and WLAN.pkt. The main editor area displays the code for 'ServerTCP.py'. The terminal at the bottom shows the output of running the script:

```
PS C:\Users\dell\Desktop\CN_LAB> & C:/Users/dell/AppData/Local/Programs/Python/Python312/python.exe c:/Users/dell/Desktop/CN_LAB/ServerTCP.py
The server is ready to receive
Sent contents ofServerTCP.py
The server is ready to receive
```

The status bar at the bottom indicates the file is Python 3.12.6 64-bit.

The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left has a tree view with 'CN LAB' expanded, showing files like ClientTCP.py, ServerTCP.png, ServerTCP.py, VLAN.pkt, and WLAN.pkt. The main editor area displays the code for 'ClientTCP.py'. The terminal at the bottom shows the output of running the script:

```
PS C:\Users\dell\Desktop\CN_LAB> & C:/Users/dell/AppData/Local/Programs/Python/Python312/python.exe c:/Users/dell/Desktop/CN_LAB/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()
```

The status bar at the bottom indicates the file is Python 3.12.6 64-bit.

Program 4

Aim of the program:

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.

Code:

```
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()
```

```
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()
```

OUTPUT:

The screenshot shows a code editor interface with several tabs at the top: ClientUDP.py, ServerUDP.py (which is currently selected), and ClientTCP.py. Below the tabs, there is a code editor window containing the following Python code:

```
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()
```

Below the code editor, there is a terminal window showing the execution of the program:

```
PS C:\Users\dell\Desktop\CN_LAB> & C:/Users/dell/AppData/Local/Programs/Python/Python312/python.exe c:/Users/dell/Desktop/CN_LAB/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)
```

The screenshot shows a Python code editor interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** CN_LAB
- Toolbar:** Standard window controls (minimize, maximize, close).
- Sidebar:** Includes icons for file operations (New, Open, Save, Find, Copy, Paste, Delete), a search icon, and a refresh icon.
- Code Editor:** The active tab is "ServerUDP.py". The code is as follows:

```
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()
```

- Terminal:** Shows the command line output of running the script:

```
PS C:\Users\dell\Desktop\CN_LAB> & C:/Users/dell/AppData/Local/Programs/Python/Python312/python.exe c:/Users/dell/Desktop/CN_LAB/ServerUDP.py
The server is ready to receive
Sent contents ofServerUDP.py
```

- Bottom Status Bar:** Shows the current terminal tab is "Python: ServerUDP".