VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB RECORD

Computer Network Lab (23CS5PCCON)

Submitted by

Vachan.D.H (1BM22CS314)

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 **Vachan.D.H(1BM22CS314),** who is a 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.

Sneha.P Associate Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE

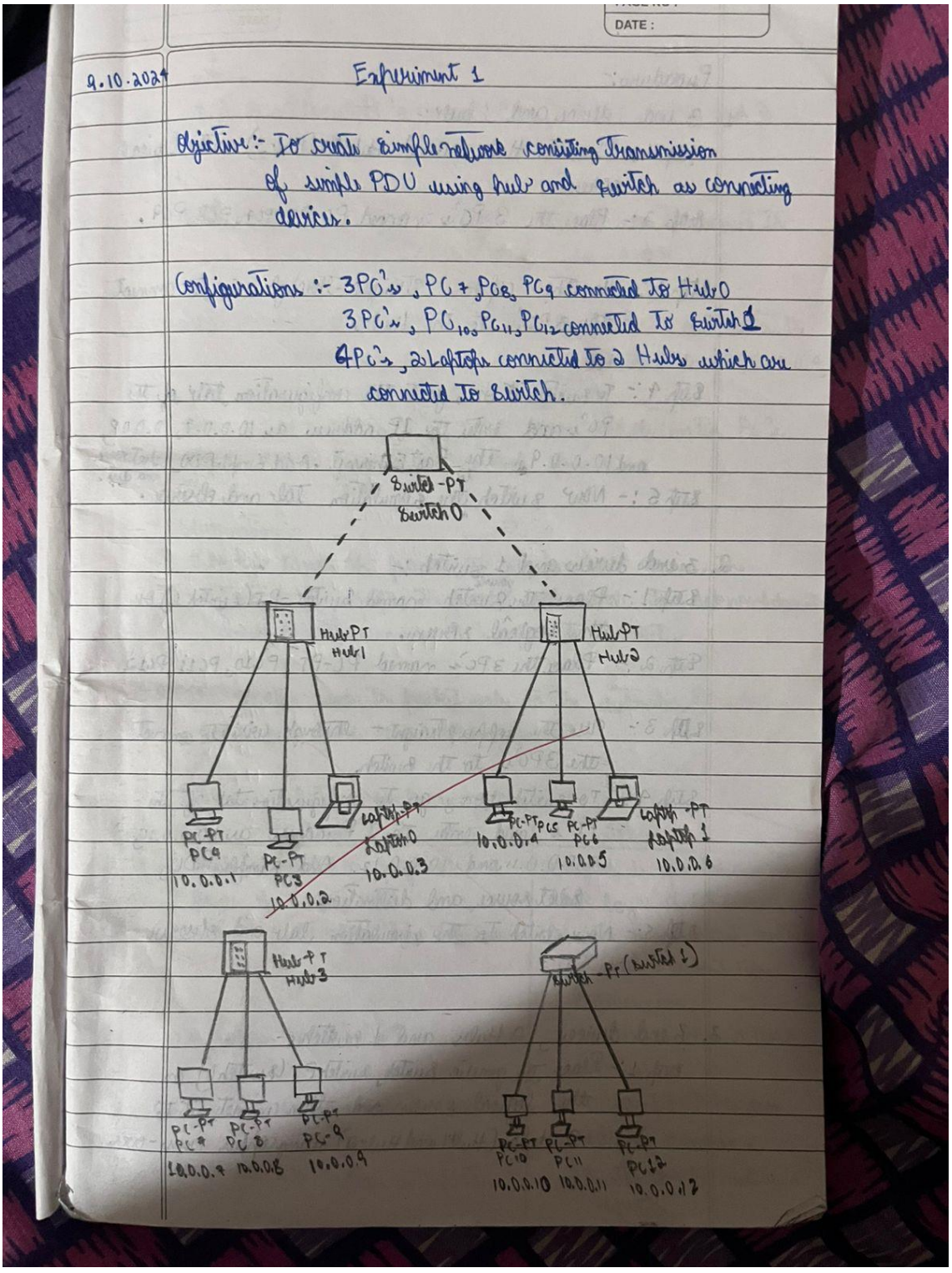
Index

Github Link:
https://github.com/Vachandh/CN-LAB.git

Program 1

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

Topology , Procedure and Observation:



PAGE NO :
DATE :

Procedure :-

1. 3 end devices and 1 hub :-

Step 1 :- Place the Hub named Hub-PT (Hub 3) on the Logical screen.

Step 2 :- Place the 3 PC's named PC-PT PC7, PC8, PC9.

Step 3 :- Use the copper straight-through wire to connect the 3 PC's to the Hub.

Step 4 :- To switch it on, go to the configuration tab of the PC's and enter the IP address as 10.0.0.7, 10.0.0.8 and 10.0.0.9 in the FastEthernet. Add simple PDU, select source and dest.

Step 5 :- Now switch to the simulation tab and observe.

2. 3 end devices and 1 switch :-

Step 1 :- Place the generic Switch named Switch-PT (Switch 1) on the Logical screen.

Step 2 :- Place the 3 PC's named PC-PT PC10, PC11, PC12

Step 3 :- Use the copper straight-through wire to connect the 3 PC's to the Switch.

Step 4 :- To switch it on, go to configuration tab of the PC's and enter the IP address as 10.0.0.10, 10.0.0.11 and 10.0.0.12. Add simple PDU, select source and destination.

Step 5 :- Now switch to the simulation tab and observe.

3. 3 end devices, 2 Hubs and 1 Switch :-

Step 1 :- Place the generic Switch, Switch-PT (Switch 0) on the logical screen and then connect it to 2 Hubs (Hub 1 and Hub 2) using copper cross-over.

PAGE NO :
DATE :

wire.

Step 2:- Place 4 PC's named PC_1, PC_2, PC_3, PC_4 and 2 Laptop's Laptop 1, Laptop 2.

Step 3:- Use the copper straight - Through wire to connect the 4 PC's and 2 Laptops to Hubs 1 and 0.

Step 4:- Go to the configuration Tab of the PC's and enter the IP address as 10.0.0.1, 10.0.0.2, 10.0.0.3, 10.0.0.4, 10.0.0.5 and 10.0.0.6 respectively.

Step 5:- Add simple PDU's, Select source and destination PC's.

Step 6:- Go to simulation Tab and observe.

Observation:-

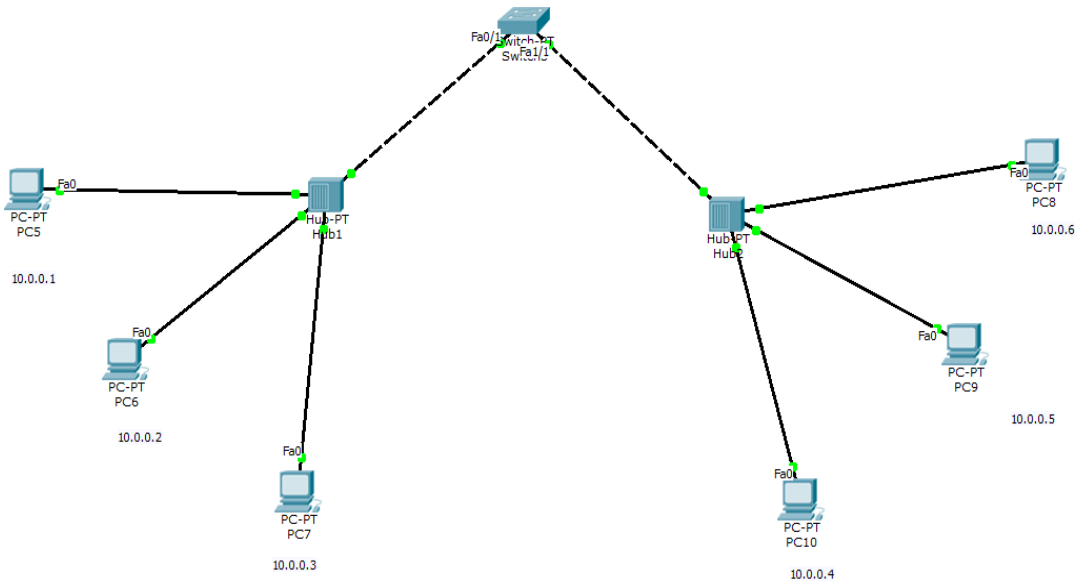
1. The hub sends the packet to all available devices. The destination PC accepts the packet and sends the acknowledgement back. All the remaining PC's reject the packet.

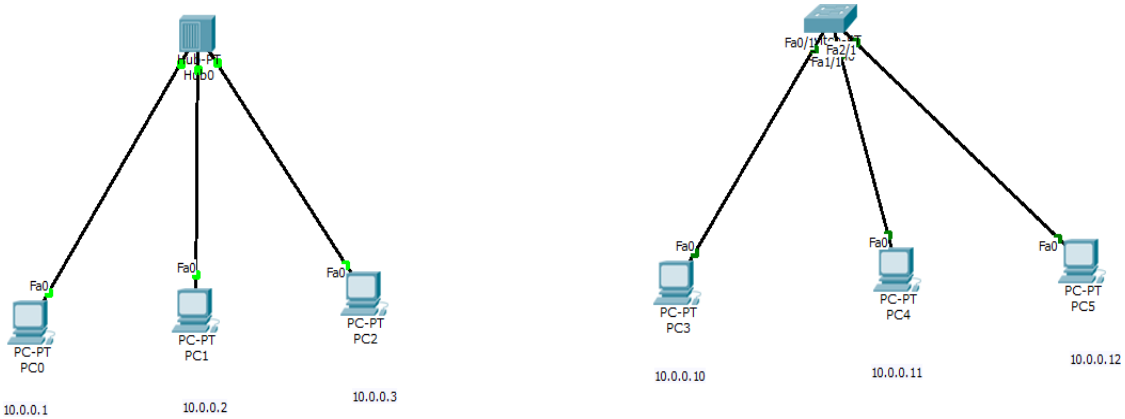
2. The switch sends the packet only to the destination PC which accepts and sends back acknowledgement.

Difference between Hub and Switch:-

Hub	Switch
→ Hub is operated on physical layer of OSI model.	→ Switch is operated on Data link layer of OSI model.
→ Hub is a broadcast type transmission.	→ Switch is a unicast, multicast and broadcast type transmission.
→ Hubs have 4/12 ports	→ Switch can have 24 to 48 ports.
→ There is only one collision domain.	→ Different ports have own collision domain.
→ Hub can't be used as a repeater.	→ Switch can be used as a repeater.

Screen Shots:

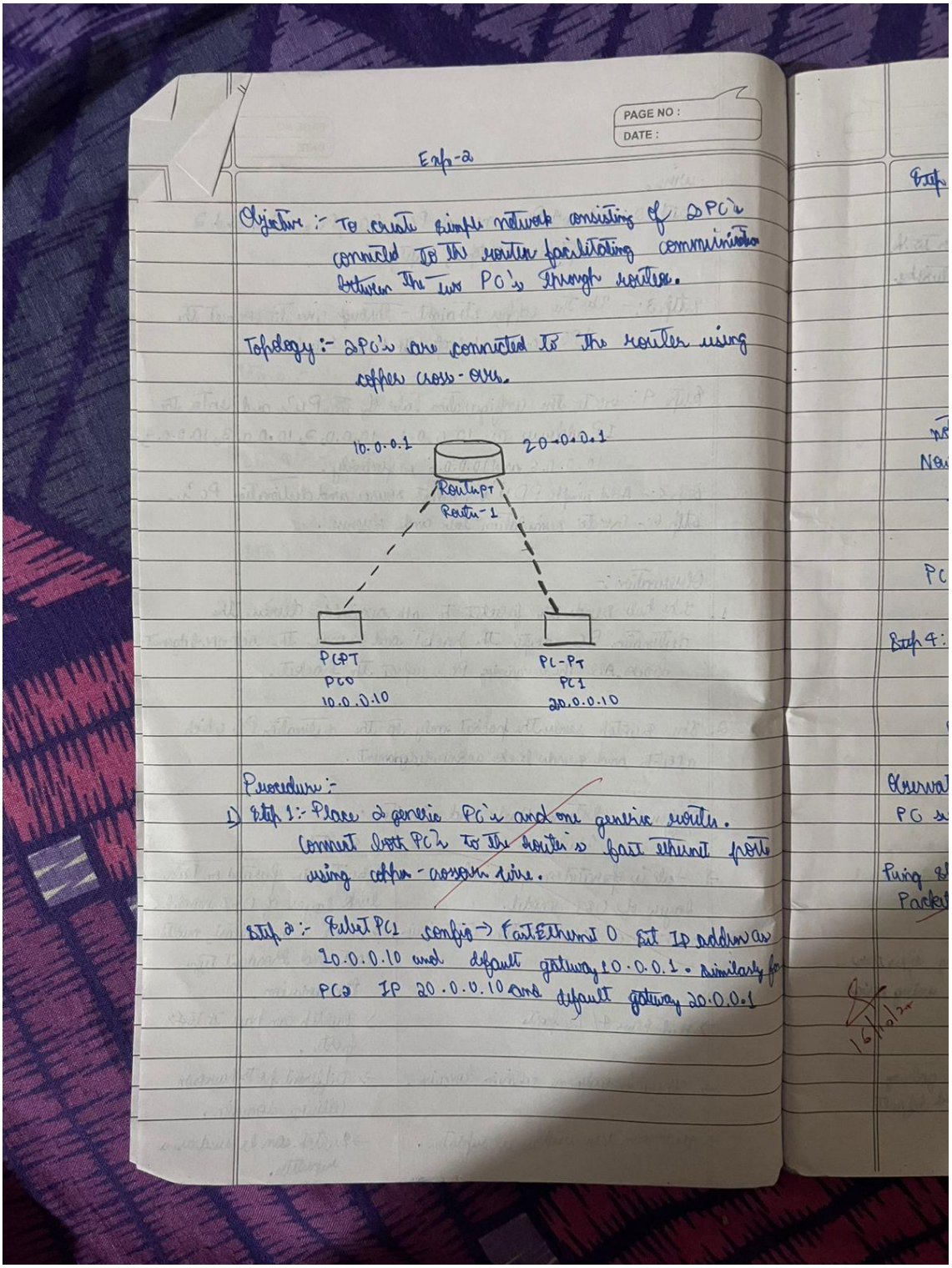




Program 2

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

Topology , Procedure and Observation:



PAGE NO :
DATE :

Step 3:- Select router and go to the CLI, execute the following commands.

enable
config Terminal
interface Fast Ethernet 1/0
ip address 10.0.0.1 255.0.0.0
no shutdown
exit

notice PC0 and router0 are successfully connected.

Now run for PC1

interface fast Ethernet 1/0
ip address 20.0.0.1 255.0.0.0
no shutdown.

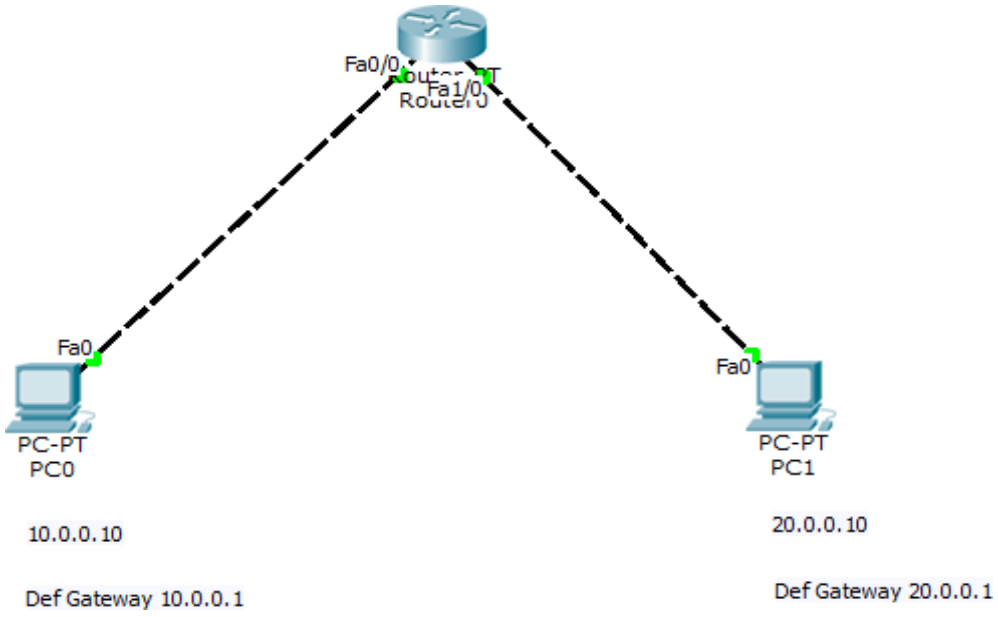
PC1 is also successfully connected.

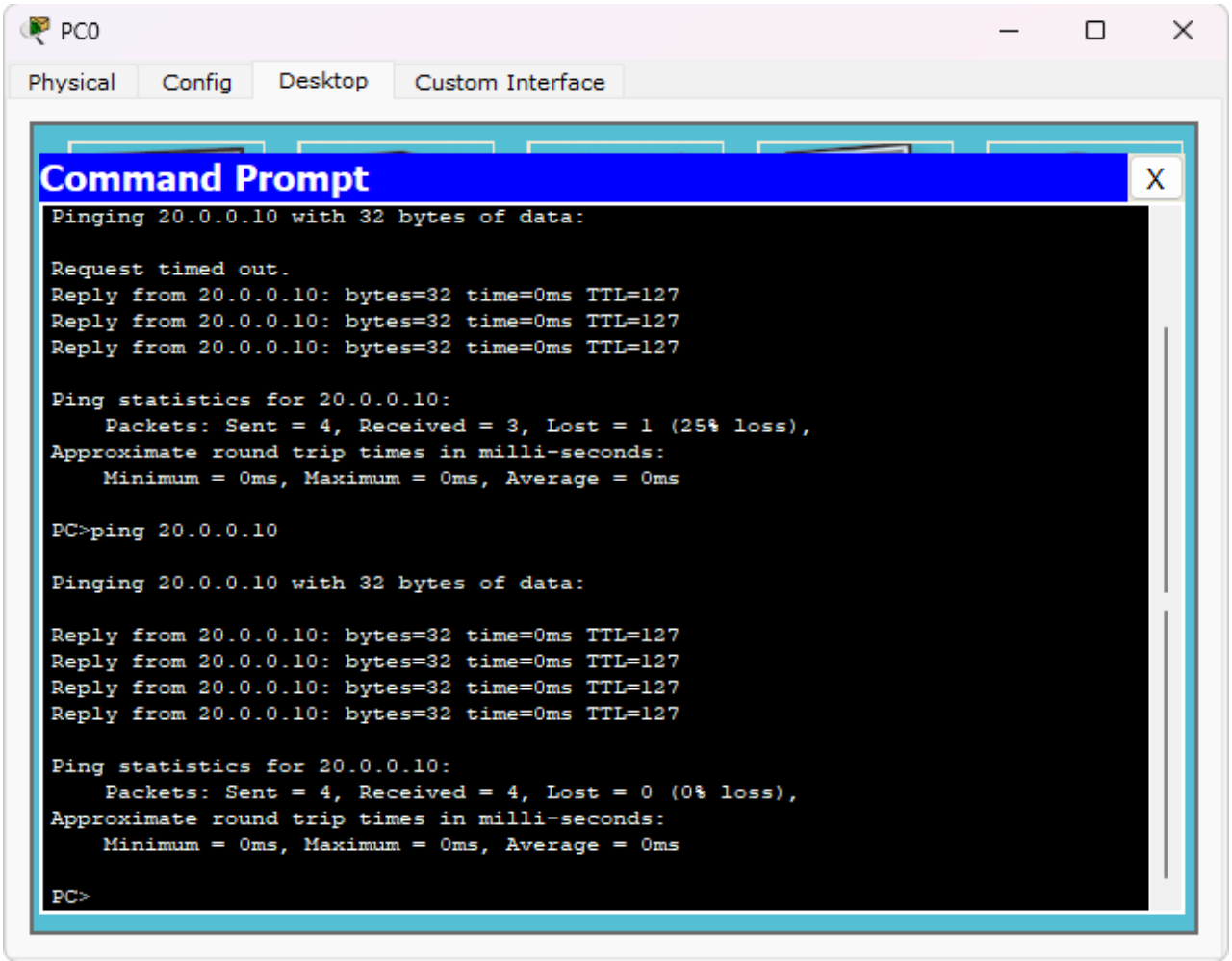
Step 4:- Select PC0 → Desktop → command prompt & ping PC1 by running command
→ ping 20.0.0.10
Observe the output.

Observation:-
PC0 successfully ping PC1 with 32 bytes of data

Ping statistics :-
Packets :-
Sent = 4, Received = 4, Lost = 0

Screen Shots:

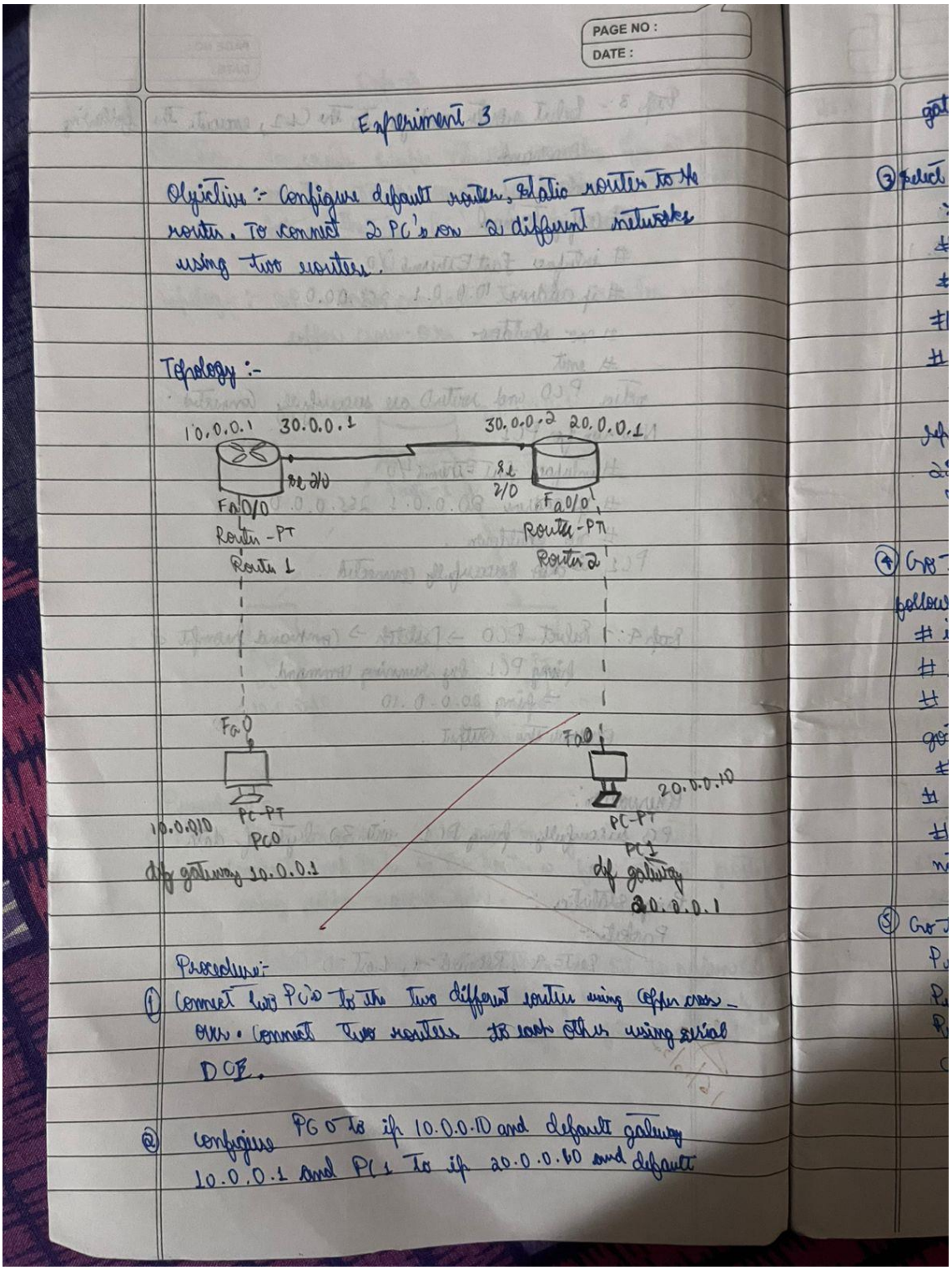




Program 3

Aim: Configure default route, static route to the Router(Part 1).

Topology , Procedure and Observation:



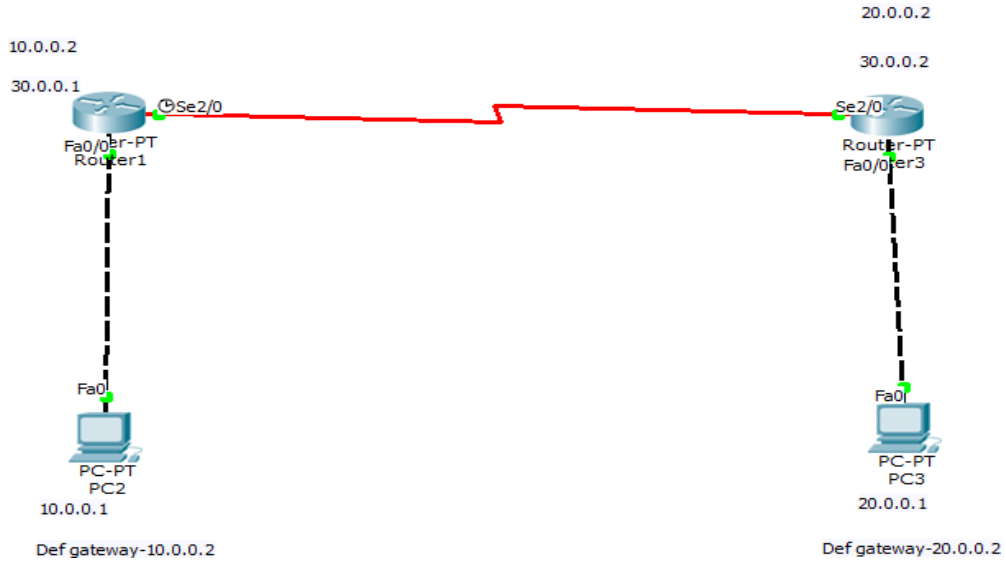
gateway 20.0.0.1.

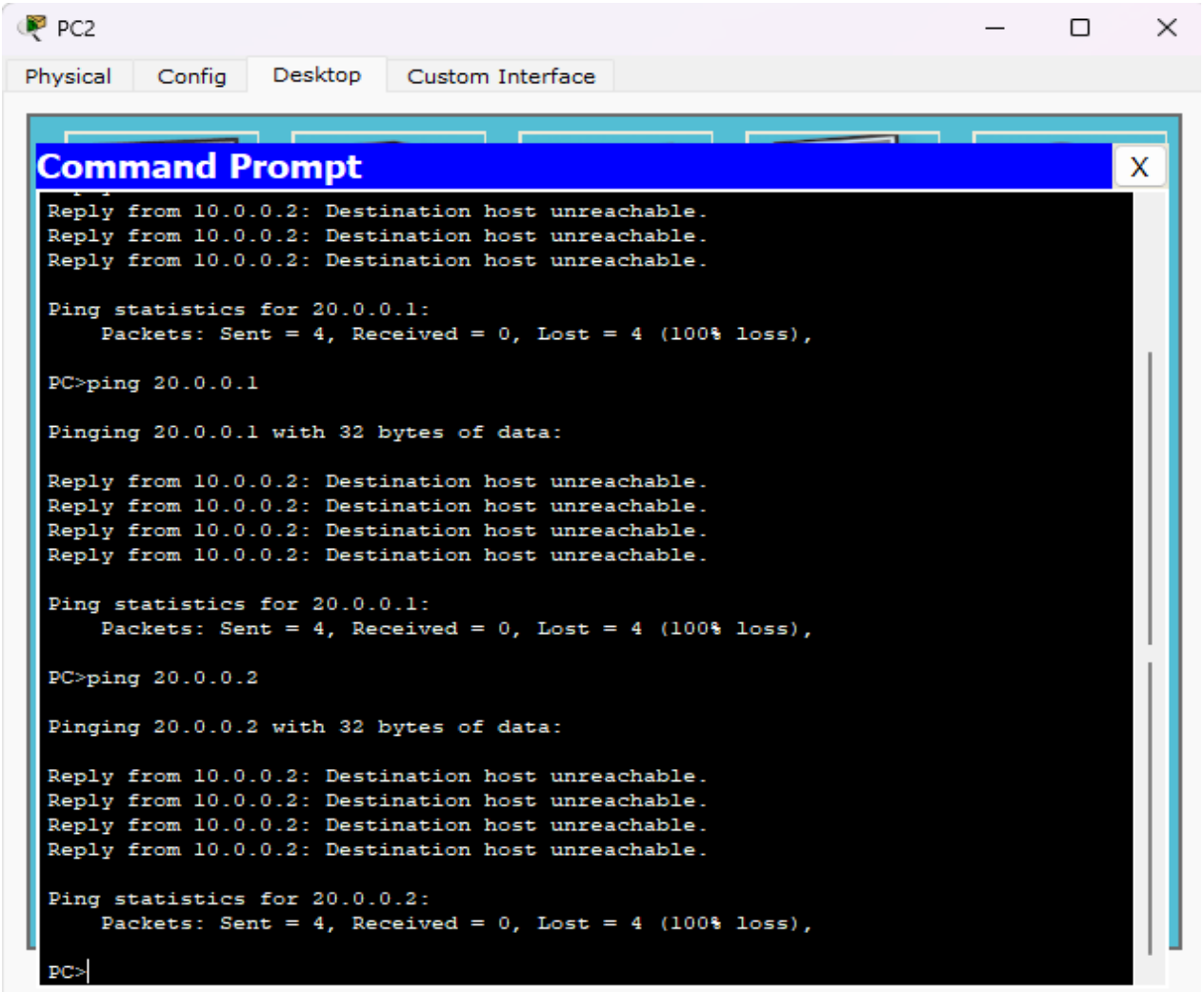
③ Select router 0 → CLI and execute the following commands
> enable
config terminal
interface FastEthernet 0/0
ip address 10.0.0.1 255.0.0.0
no shut
notice PC0 & router 0 are successfully connected.
Select the same procedure for router 1 with ip address 20.0.0.1 255.0.0.0
notice PC1 & router 1 are also successfully connected.

④ Go to router 0 → CLI config terminal and execute the following
interface serial 2/0
ip address 30.0.0.1 255.0.0.0
no shut
go to router 1 → CLI config terminal and execute
interface serial 2/0
ip address 30.0.0.2 255.0.0.0
no shut
notice router 0 & router 1 are successfully connected

⑤ Go to pc0 → Desktop → Command prompt and run
Ping 20.0.0.10
Ping 30.0.0.2
Ping 30.0.0.1
and observe the outputs

Screen Shots:

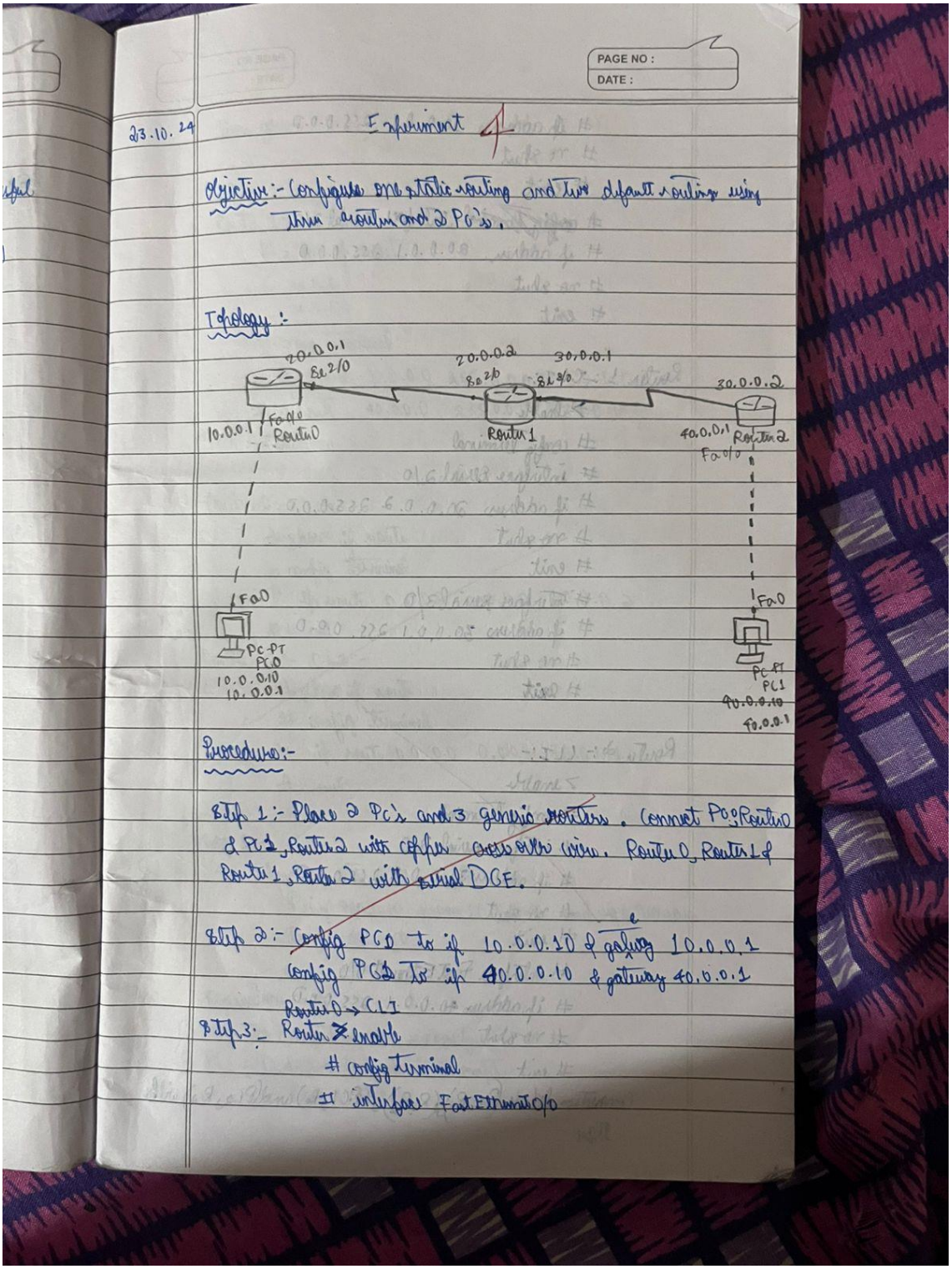




Program 4

Aim: Configure default route, static route to the Router(Part 2).

Topology , Procedure and Observation:



PAGE NO :
DATE :

ip address 10.0.0.1 255.0.0.0
no shut
exit
config terminal interface serial 2/0
ip addrss 20.0.0.1 255.0.0.0
no shut
exit

Router 1 :- CLI :-

> enable
config terminal
interface serial 2/0
ip address 20.0.0.2 255.0.0.0
no shut
exit
interface serial 3/0
ip address 30.0.0.1 255.0.0.0
no shut
exit

Router 2 :- CLI :-

> enable
config terminal
interface serial 2/0
ip address 30.0.0.2 255.0.0.0
no shut
exit
interface Fast Ethernet 0/0
ip address 40.0.0.1 255.0.0.0
no shut
exit

Connection b/w (R_0, R_1), (R_1, R_2), (PC_1, R_0) and (PC_2, R_2) will be

PAGE NO :
DATE :

Turn green.

Step 4:-

Router 1:- CLI :-

show ip route
C 20.0.0.0
C 30.0.0.0
config terminal
ip route 10.0.0.0 255.0.0.0 20.0.0.1
ip route 40.0.0.0 255.0.0.0 30.0.0.2
exit

Router 0:- CLI :-

show ip route
config terminal
ip route 0.0.0.0 0.0.0.0 20.0.0.2
exit

Router 2:- CLI :-

show ip route
config terminal
ip route 0.0.0.0 0.0.0.0 30.0.0.1
exit

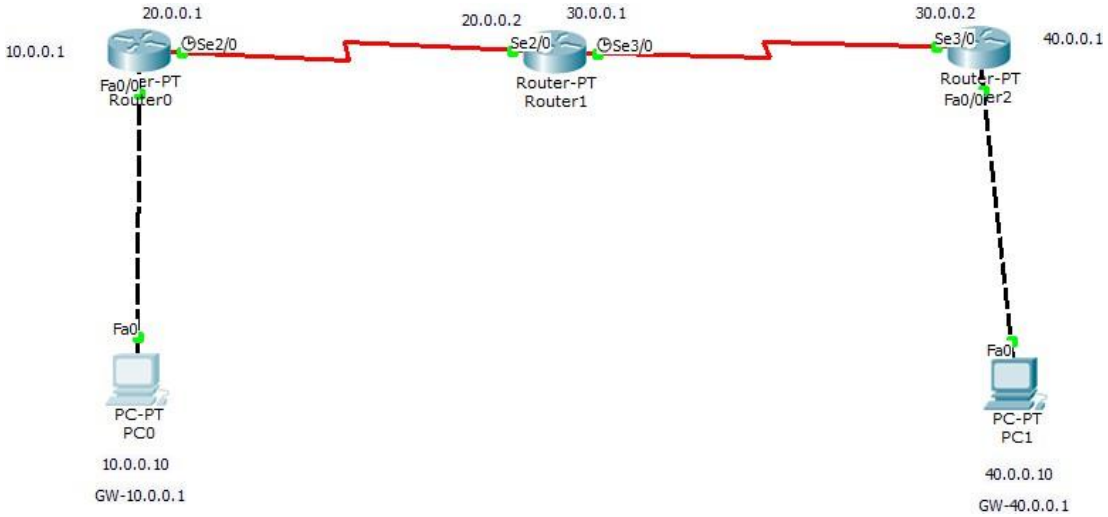
Step 5:-

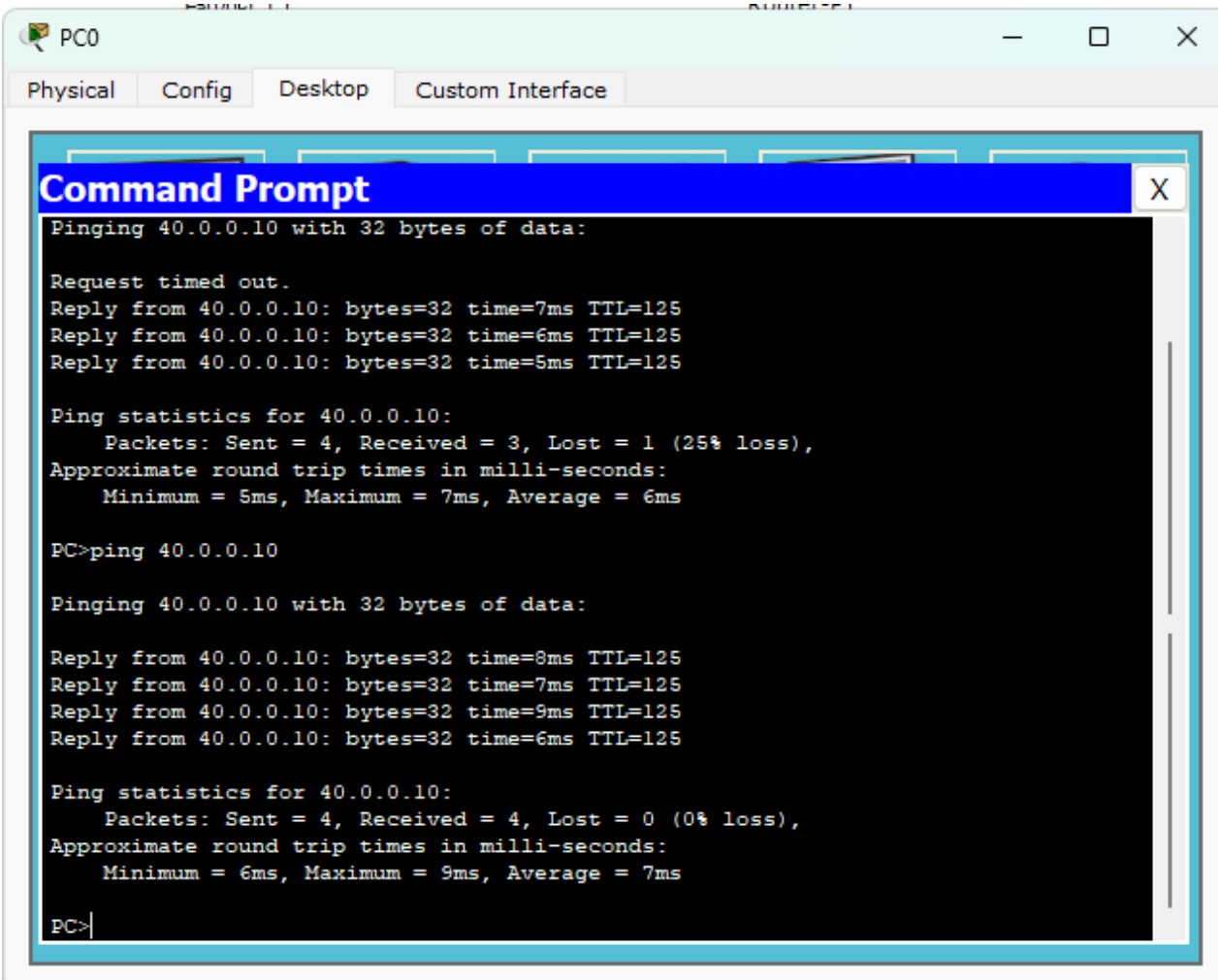
Show IP route in the router
Ping 40.0.0.10 from 10.0.0.10 and vice versa.
Ping 10.0.0.10

Observation :-

(i) All the connection are successful
(ii) PC1 -> Desktop -> command prompt
Ping 40.0.0.10 is successful

Screen Shots:

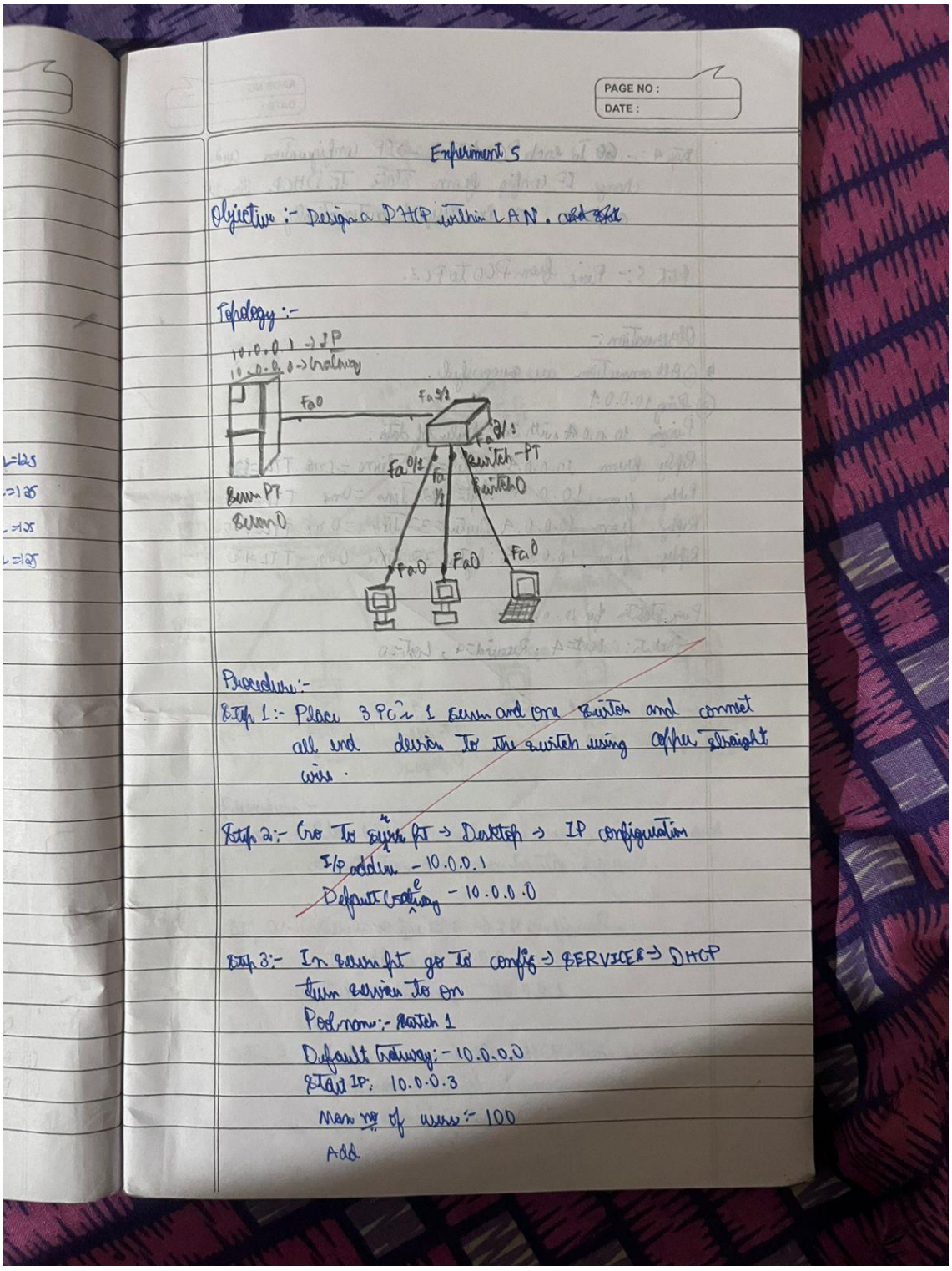




Program 5

Aim: Configure DHCP within a LAN and outside LAN.

Topology , Procedure and Observation:



PAGE NO :
DATE :

Step 4 :- Go to each PC desktop → IP configuration and change IP config from Static to DHCP. The IP addresses will be assigned automatically.

Step 5 :- Ping from PC0 to PC3.

Observation :-

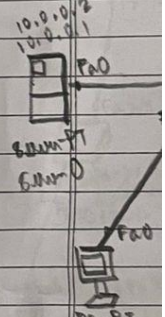
1. All connections are successful.
2. Ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data :

Reply from 10.0.0.4 : bytes=32 Time=1ms TTL=120
Reply from 10.0.0.4 : bytes=32 Time=0ms TTL=120
Reply from 10.0.0.4 : bytes=32 Time=0ms TTL=120
Reply from 10.0.0.4 : bytes=32 Time=0ms TTL=120

Ping Statistics for 10.0.0.4

Packets : Sent=4 ; Received=4, Lost=0

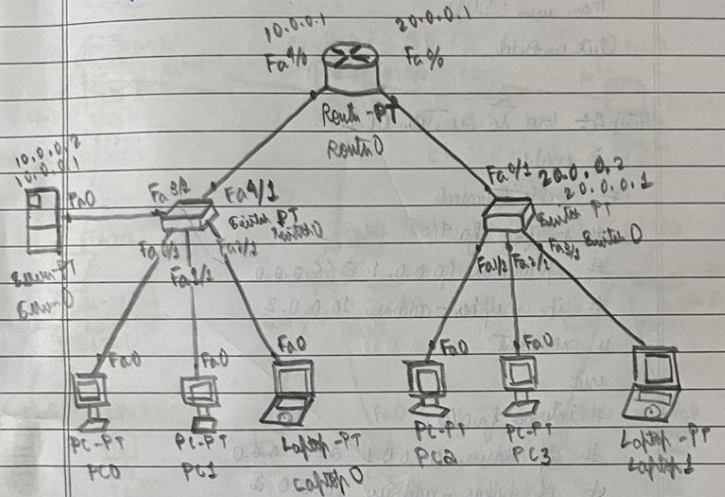


PAGE NO :
DATE :

Experiment - 6

Objective :- Design DHCP outside LAN

Topology :-



Procedure :-

Step 1 :- Place 6 PCs, 2 switch, 1 server, 1 router and connect them as shown in the figure.

Step 2 :- Server → Desktop → IP configuration
IP address - 10.0.0.2
Default Gateway - 10.0.0.1

Step 3 :- Config → Service → DHCP, Turn service to ON
Pool name :- Switch 1
Default gateway :- 10.0.0.1

PAGE NO :
DATE :

Start IP: 10.0.0.3
Man usm: 100
Click on Add.

Pool name: switch 2
Default Gateway: 20.0.0.1
Default start IP: 20.0.0.3
Man usm: 100
Click on Add

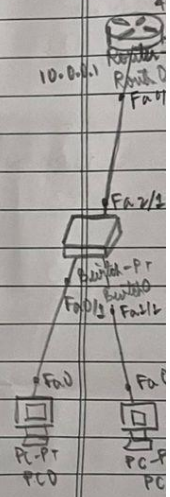
Step 4:- Go to router CLI
> enable
config terminal
interface fa 9/0
ip address 10.0.0.1 255.0.0.0
ip helper-address 10.0.0.2
no shut
exit
interface fa 0/0
ip address 20.0.0.1 255.0.0.0
ip helper-address 10.0.0.2
no shut

All router switch connection go up

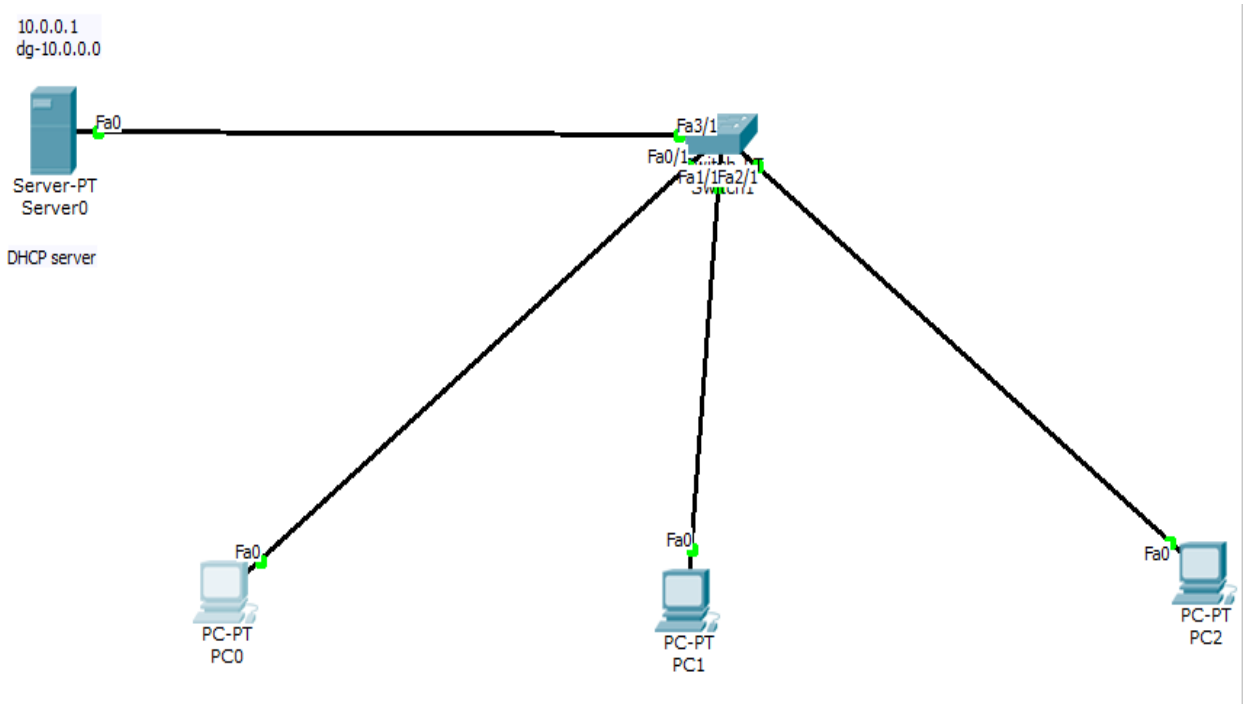
Step 5:- Go to all 6 PC's and change IP configuration from static to DHCP address will be automatically assigned.

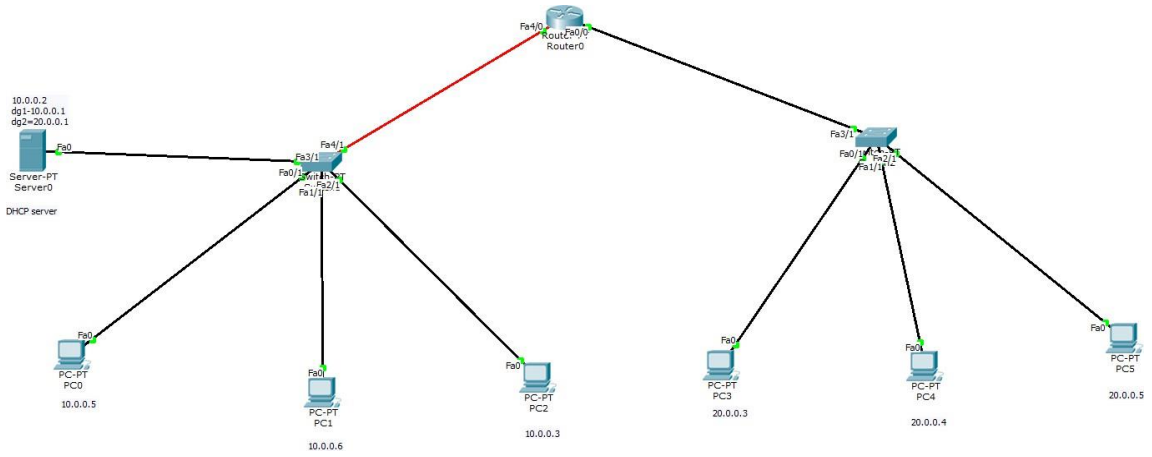
Step 6:- Ping PC0 to PC6.

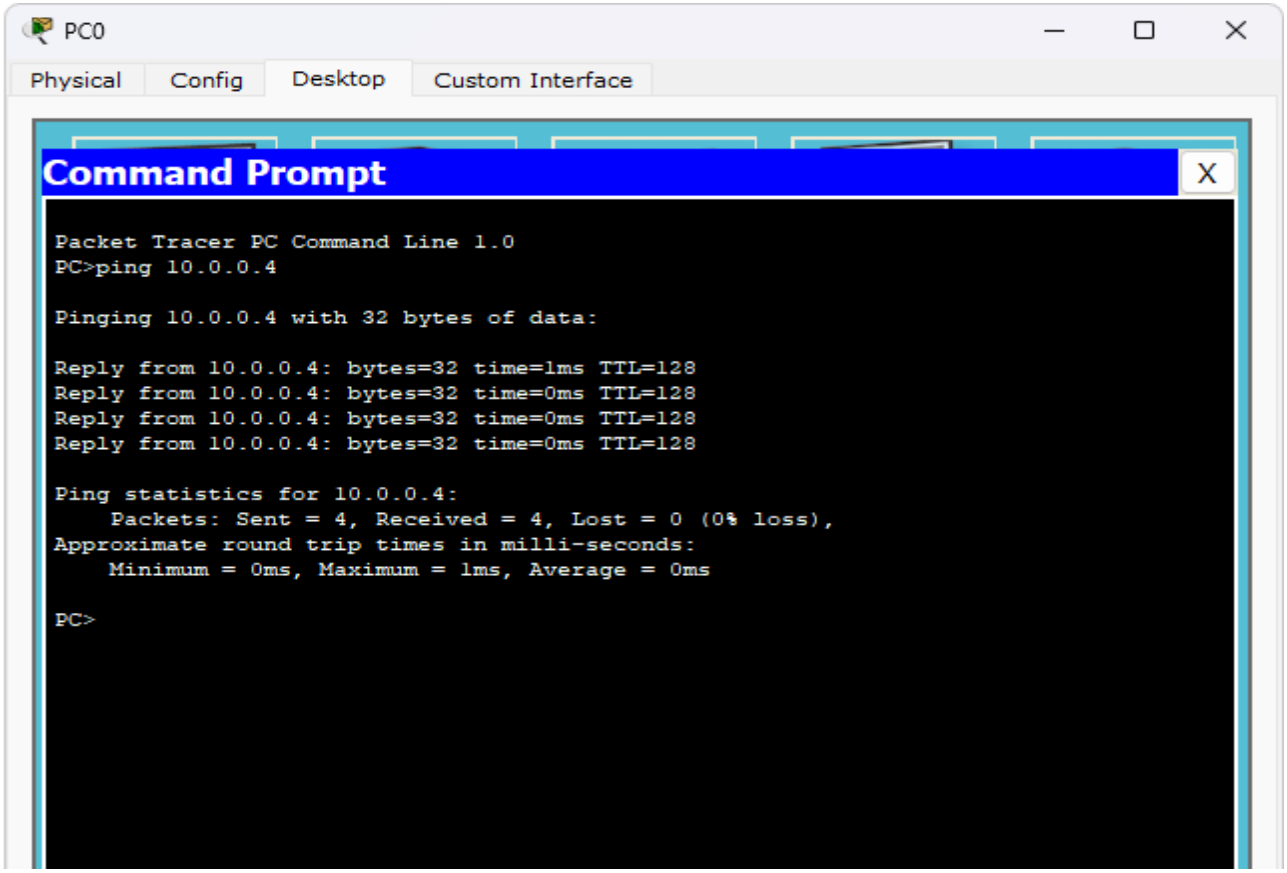
Observation:- (1) All connection are successful
(2) All the PC's are assigned.



Screen Shots:



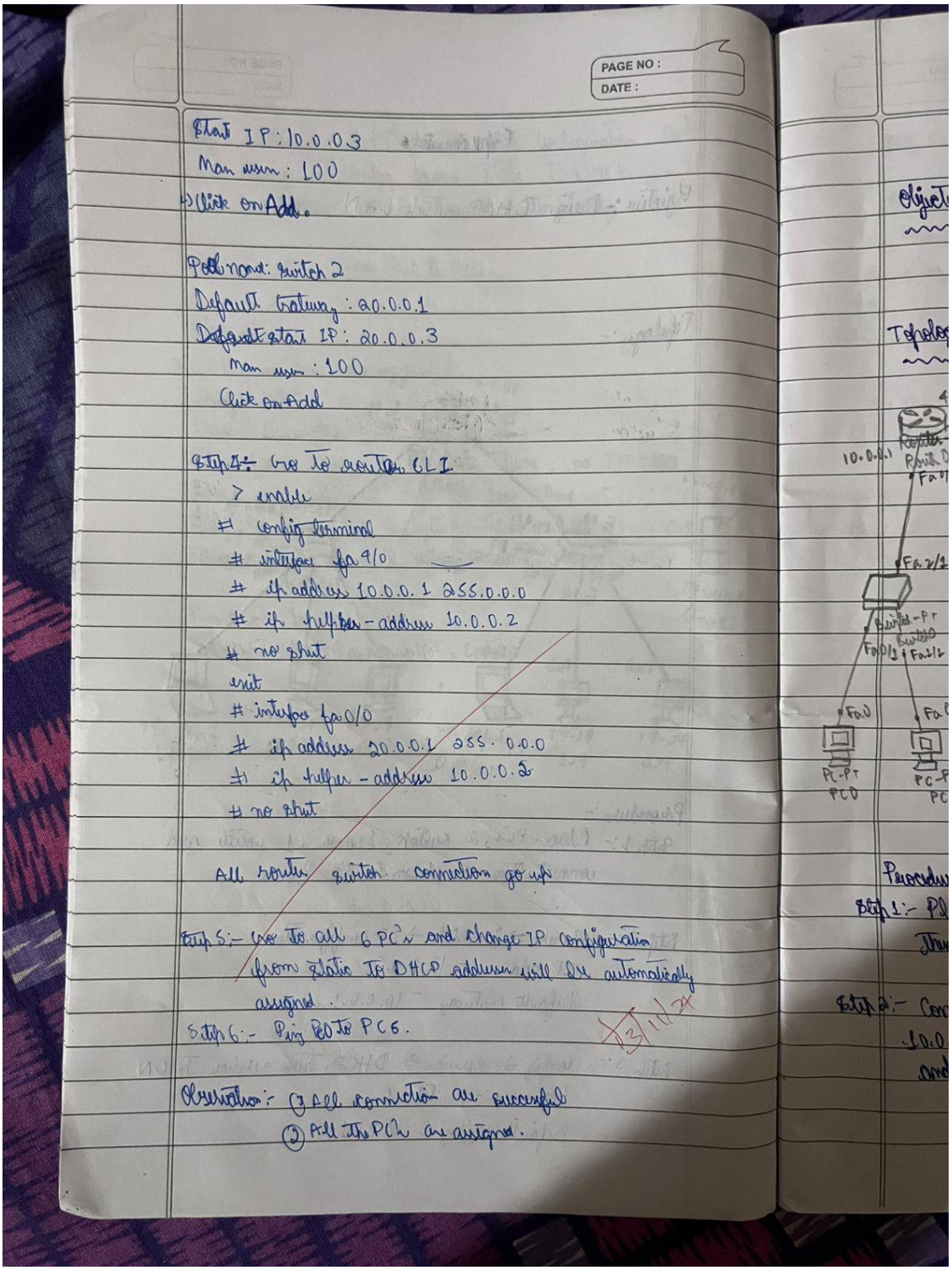

Router0
Server-PT
Server0
DHCP server
10.0.0.2
dg1-10.0.0.1
dg2=20.0.0.1
PC-PT
PC0
10.0.0.5
PC-PT
PC1
10.0.0.6
PC-PT
PC2
10.0.0.3
PC-PT
PC3
20.0.0.3
PC-PT
PC4
20.0.0.4
PC-PT
PC5
20.0.0.5

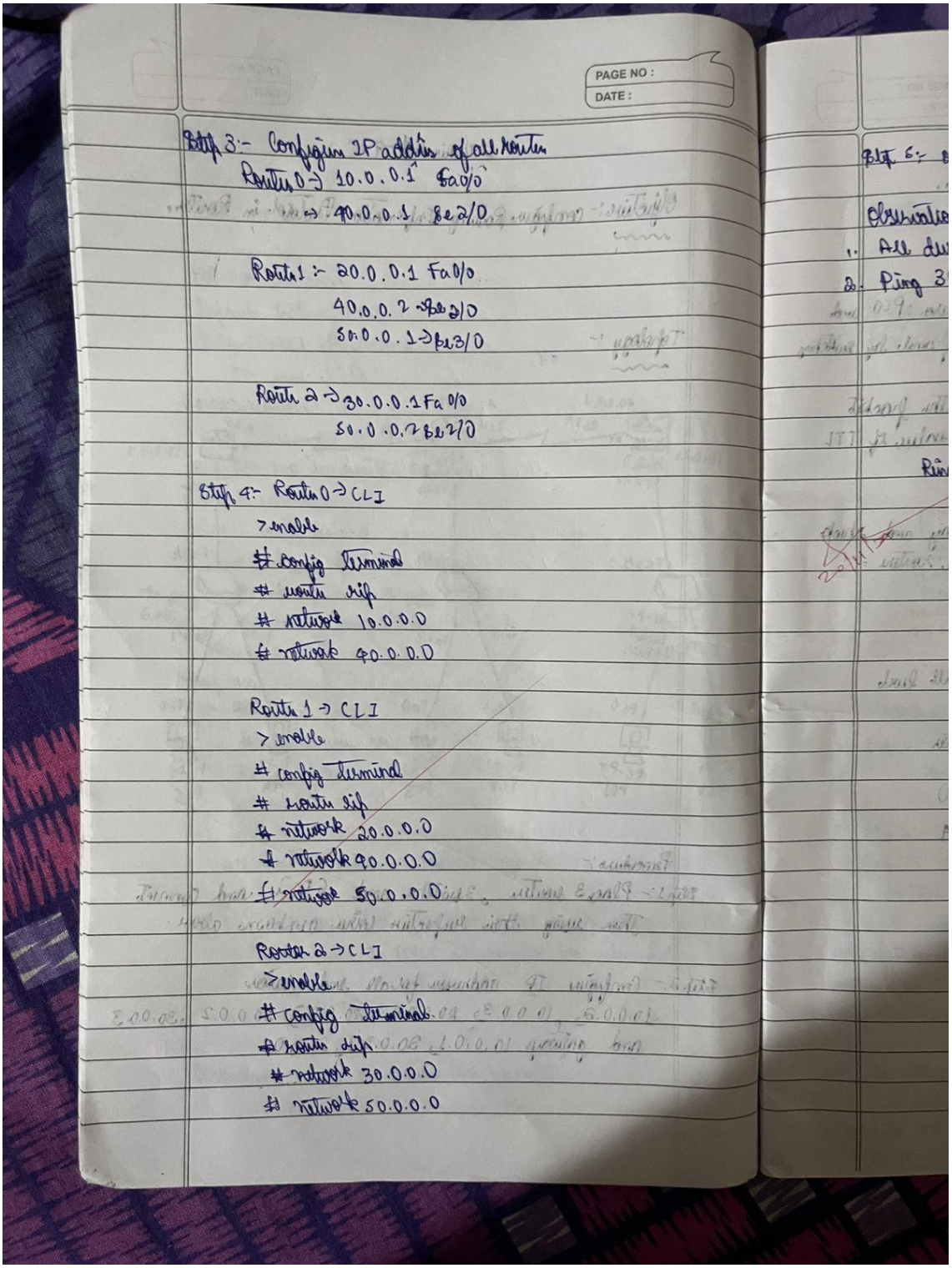

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.4
Pinging 10.0.0.4 with 32 bytes of data:
Reply from 10.0.0.4: bytes=32 time=1ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms 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
PC>

Program 6

Aim: Configure RIP routing Protocol in Routers .

Topology , Procedure and Observation:





PAGE NO :
DATE :

Step 3:- Configure IP address of all routers
Router 0 → 10.0.0.1 fa0/0
→ 40.0.0.1 se2/0

Router 1 :- 20.0.0.1 Fa0/0
40.0.0.2 → se2/0
50.0.0.1 → fa3/0

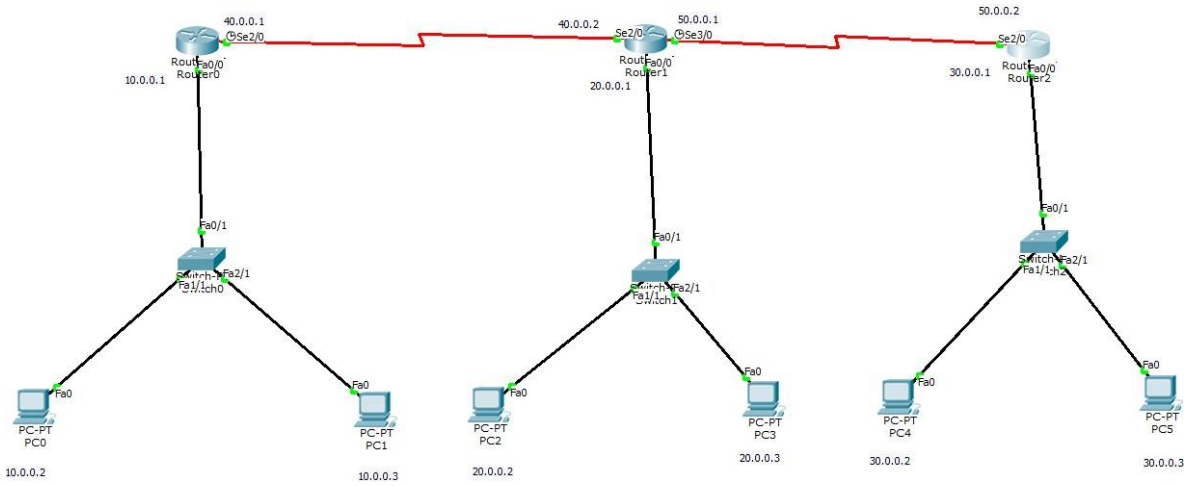
Router 2 → 30.0.0.1 Fa 0/0
50.0.0.2 se2/0

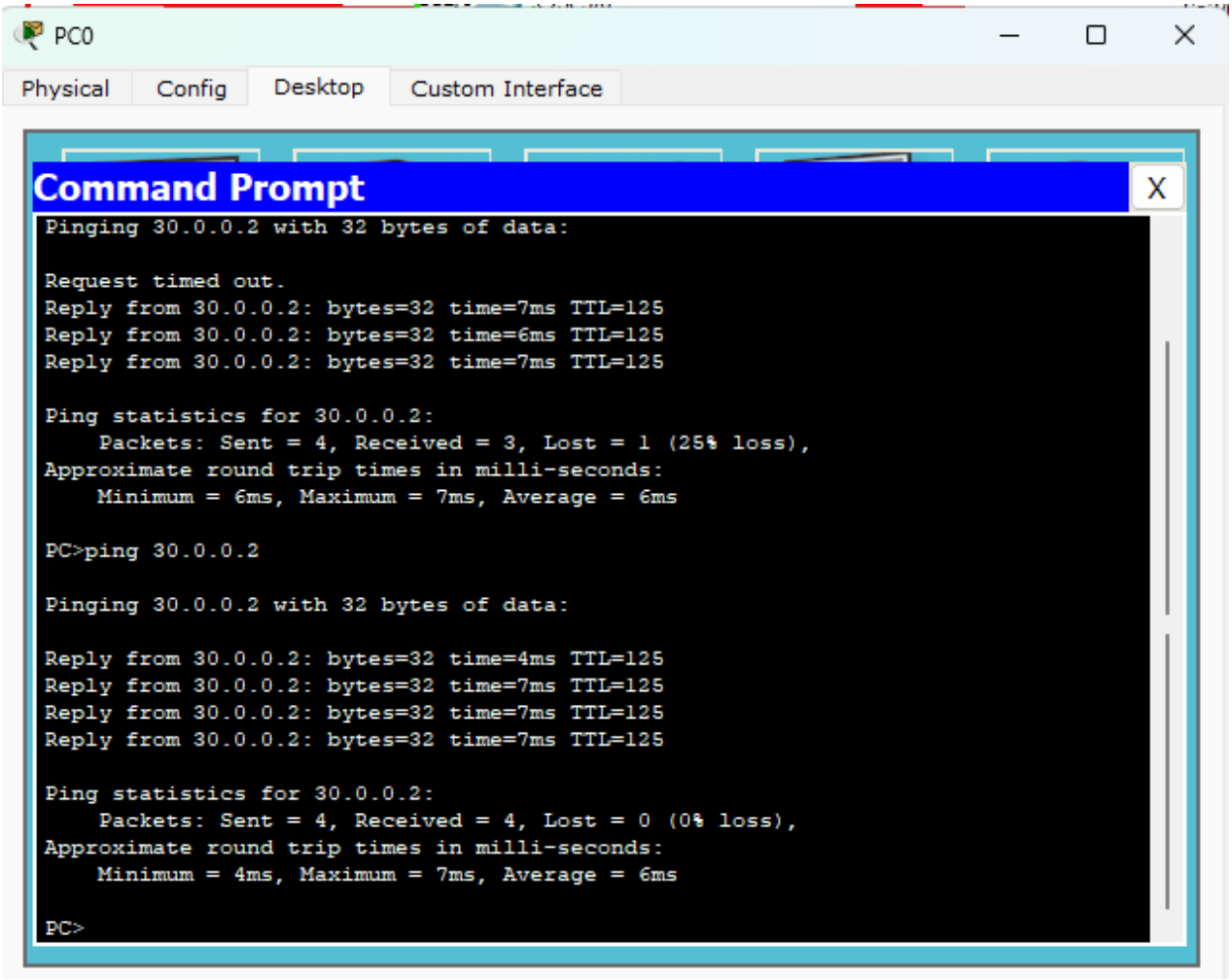
Step 4:- Router 0 → CLI
> enable
config terminal
router rip
network 10.0.0.0
network 40.0.0.0

Router 1 → CLI
> enable
config terminal
router rip
network 20.0.0.0
network 40.0.0.0
network 50.0.0.0

Router 2 → CLI
> enable
config terminal
router rip
network 30.0.0.0
network 50.0.0.0

Screen Shots:

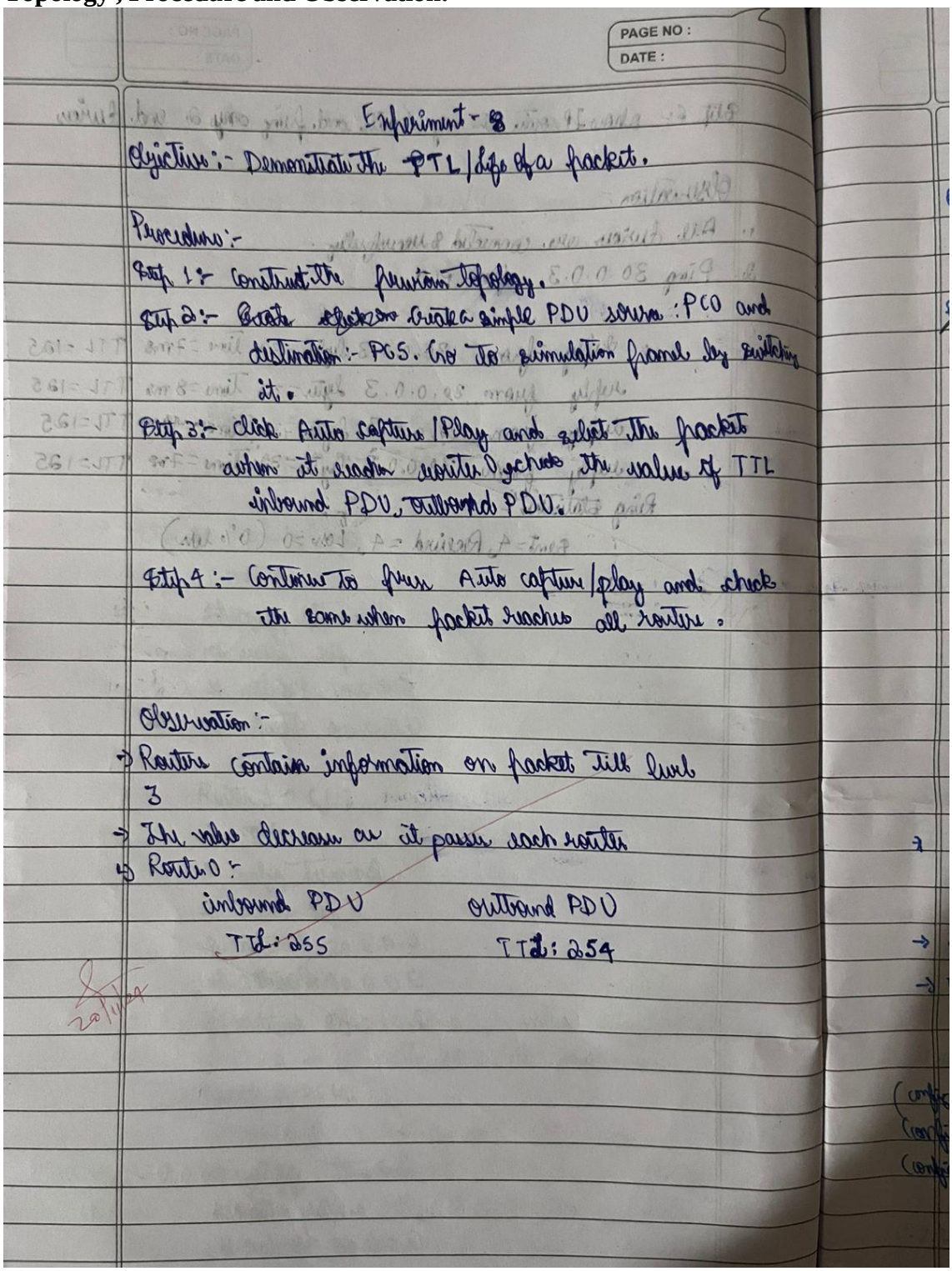




Program 7

Aim: Demonstrate the TTL/ Life of a Packet .

Topology , Procedure and Observation:



Experiment - 8

Objective :- Demonstrate the PTL / life of a packets.

Procedure :-

Step 1 :- Construct the given topology.

Step 2 :- Create a simple PDU source : PC0 and destination :- PC5. Go to simulation panel by switching it.

Step 3 :- click Auto capture / Play and select the packet when it reaches router 0 check the value of TTL inbound PDU, outbound PDU.

Step 4 :- Continue to press Auto capture / play and check the same when packet reaches all routers.

Observation :-

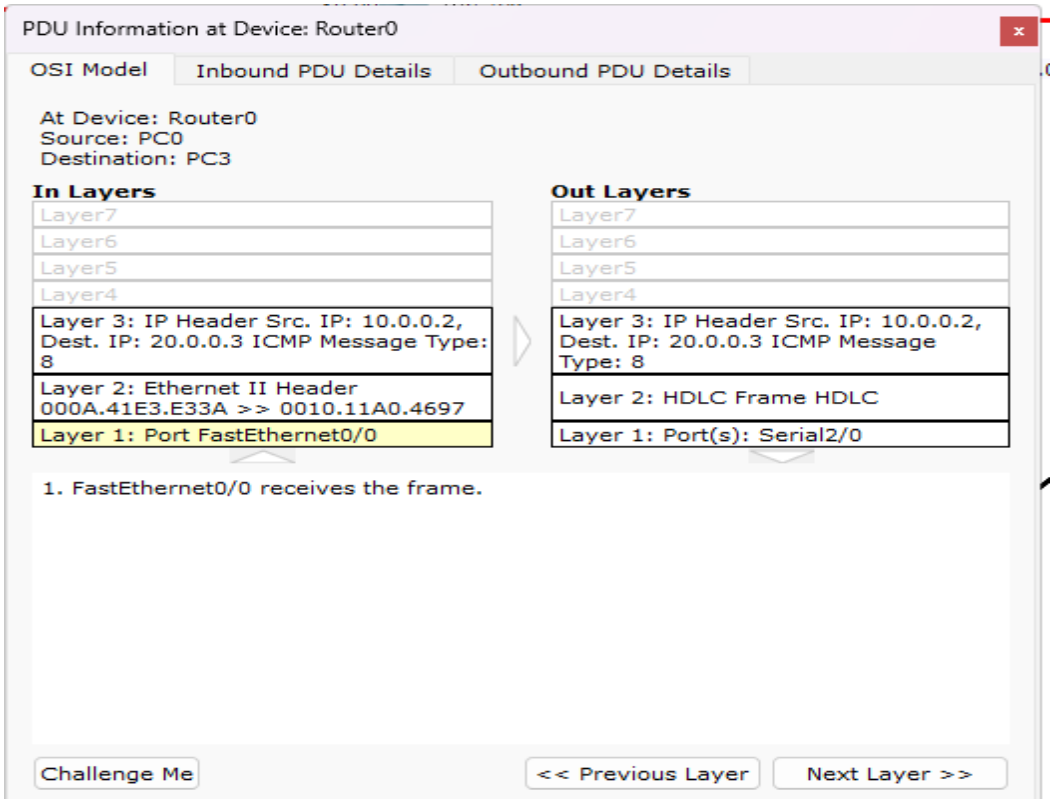
→ Routers contains information on packet till level 3

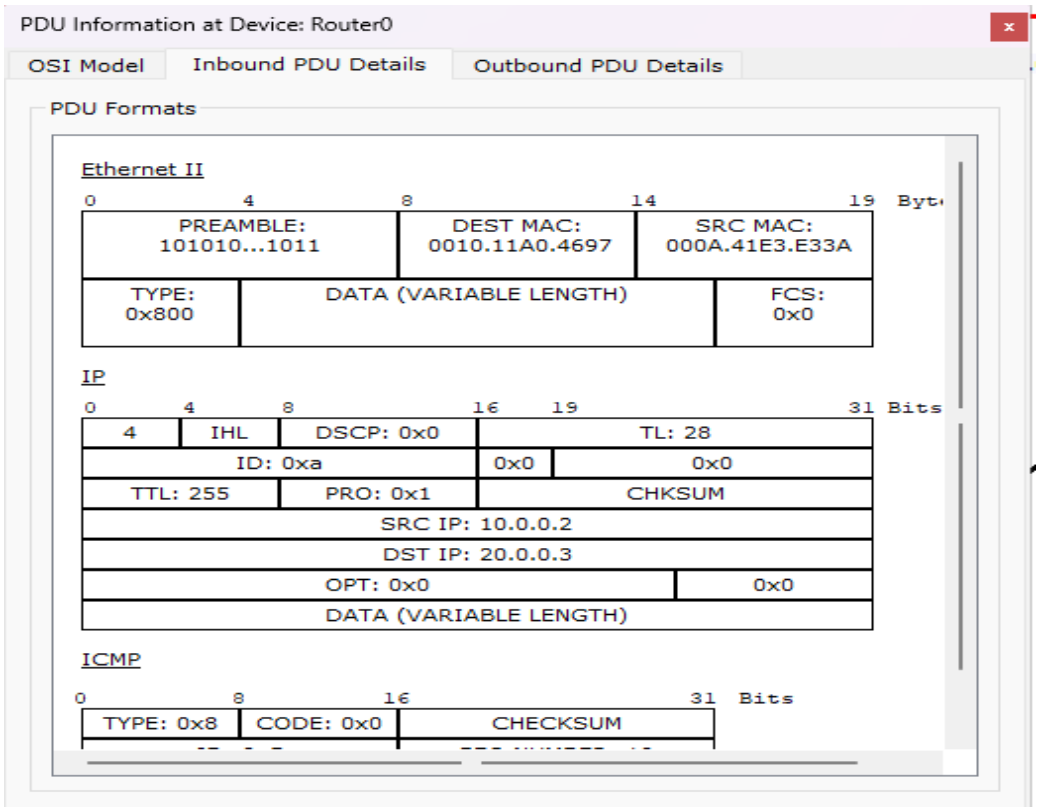
→ The value decreases as it passes each router

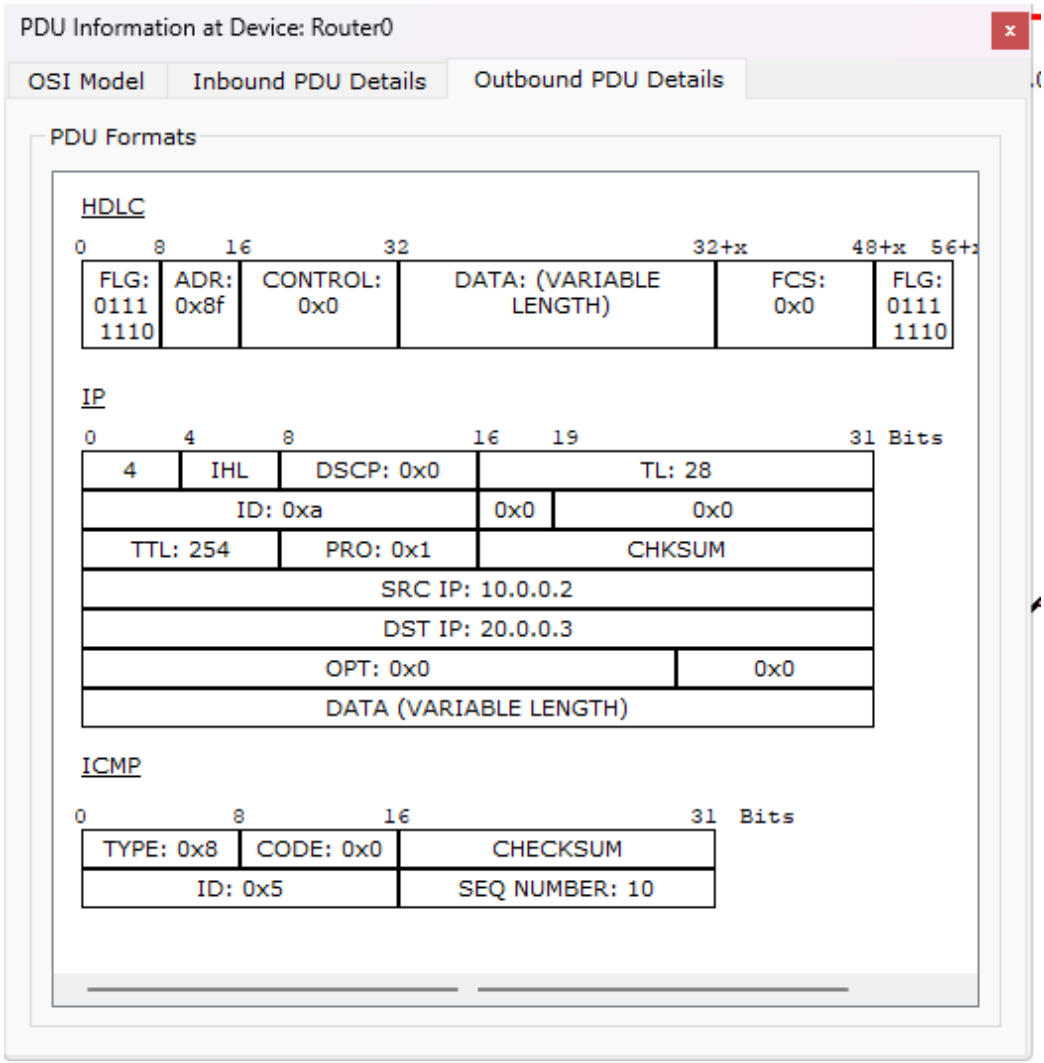
→ Router 0 :-

inbound PDU	outbound PDU
TTL : 255	TTL : 254

Screen Shots:



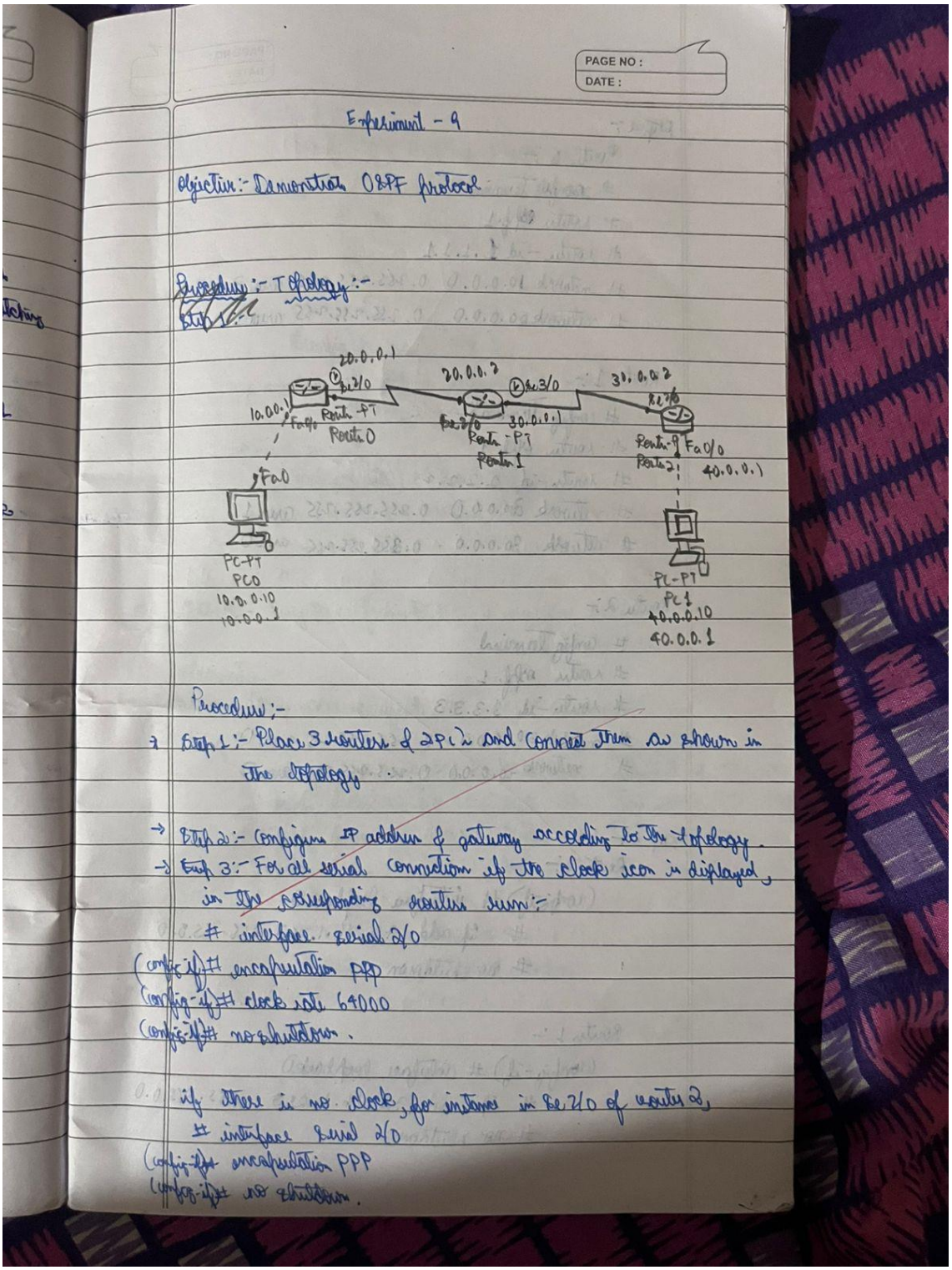


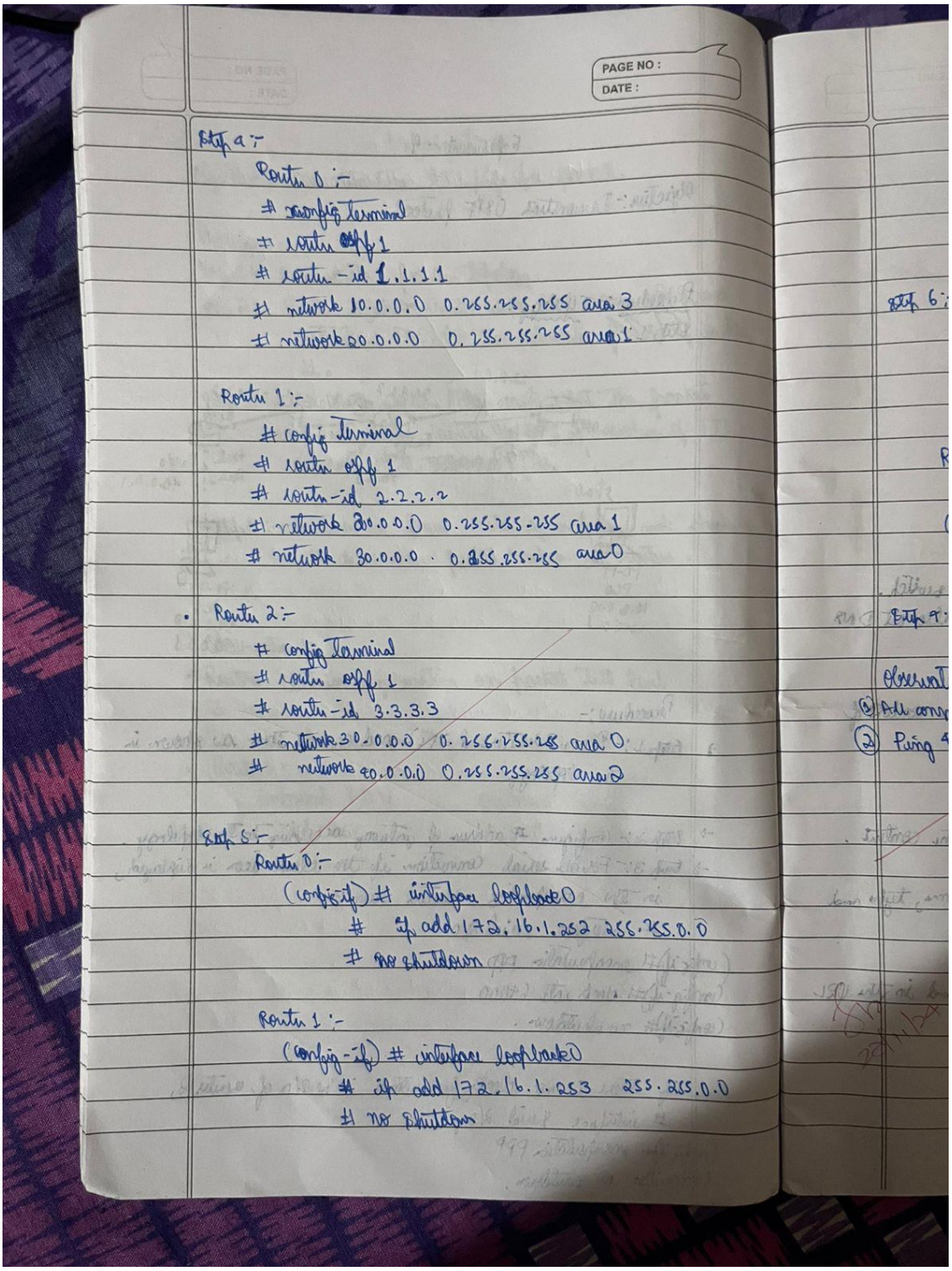

PDU Information at Device: Router0
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
0 8 16 32 32+x 48+x 56+x
FLG: 0111 1110
ADR: 0x8f
CONTROL: 0x0
DATA: (VARIABLE LENGTH)
FCS: 0x0
FLG: 0111 1110
IP
0 4 8 16 19 31 Bits
4
IHL
DSCP: 0x0
TL: 28
ID: 0xa
0x0
0x0
TTL: 254
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.2
DST IP: 20.0.0.3
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
0 8 16 31 Bits
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x5
SEQ NUMBER: 10

Program 8

Aim: Configure OSPF routing protocol .

Topology , Procedure and Observation:





PAGE NO :
DATE :

Step 4 :-

Router 0 :-

config terminal
router ospf 1
router-id 1.1.1.1
network 10.0.0.0 0.255.255.255 area 3
network 20.0.0.0 0.255.255.255 area 1

Router 1 :-

config terminal
router ospf 1
router-id 2.2.2.2
network 20.0.0.0 0.255.255.255 area 1
network 30.0.0.0 0.255.255.255 area 0

Router 2 :-

config terminal
router ospf 1
router-id 3.3.3.3
network 30.0.0.0 0.255.255.255 area 0
network 40.0.0.0 0.255.255.255 area 2

Step 5 :-

Router 0 :-

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

Router 1 :-

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

PAGE NO :
DATE :

Router 2 :-

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

step 6 :-

Router 0 :-

R0(config) # router ospf 1
(config-router) # area 1 virtual-link 2.2.2.2
exit

Router 1 :-

R1(config-router) # router ospf 1
(config-router) # area 1 virtual-link 1.1.1.1
exit

Step 9 :- Ping PC1 from PC0

Observation :-

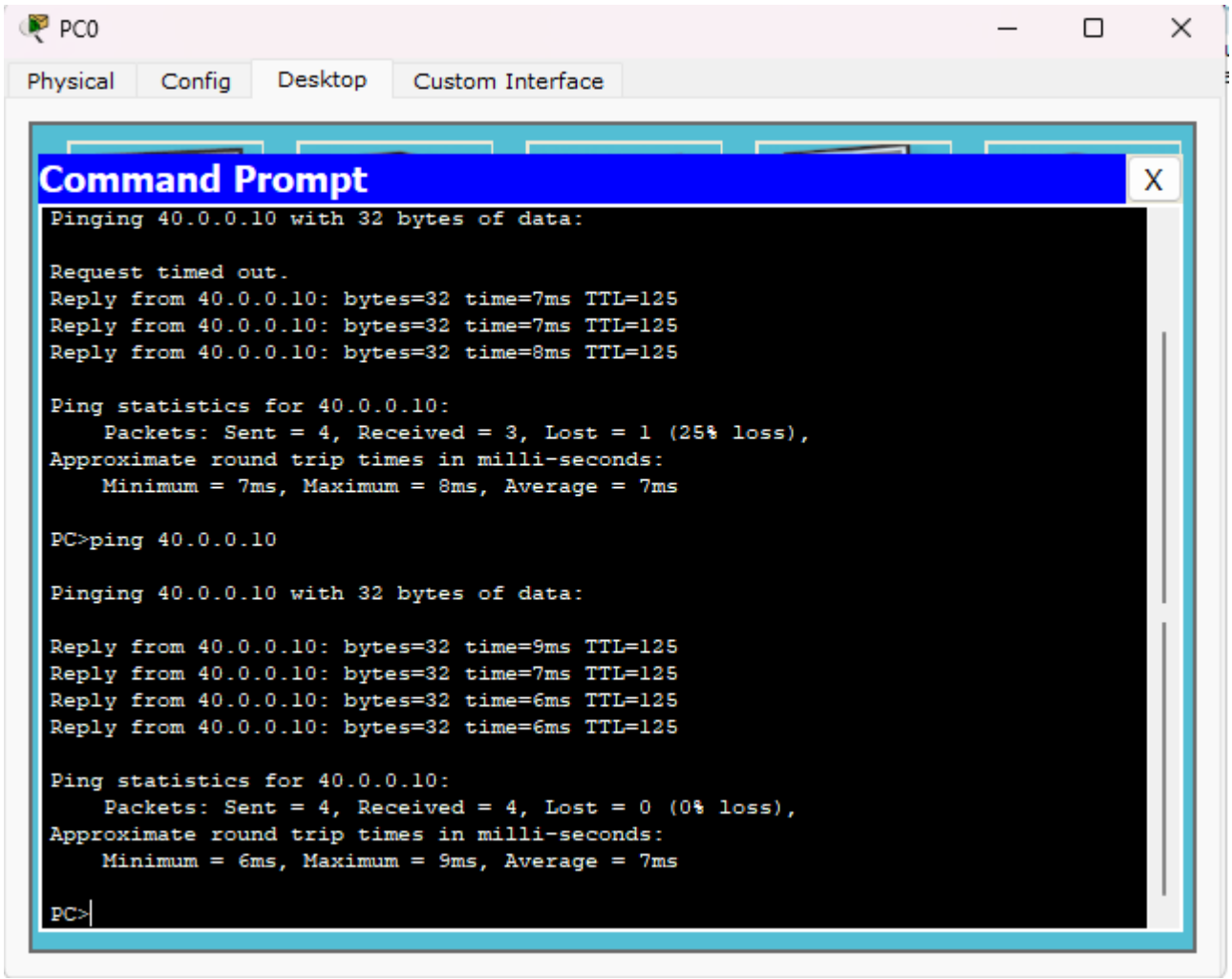
① All connections are successfull.

② Ping 40.0.0.10

reply from 40.0.0.10 bytes : 32 Time = 7ms TTL = 125
reply from 40.0.0.10 bytes : 32 Time = 6ms TTL = 125
reply from 40.0.0.10 bytes : 32 Time = 7ms TTL = 125
reply from 40.0.0.10 bytes : 32 Time = 7ms TTL = 125

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

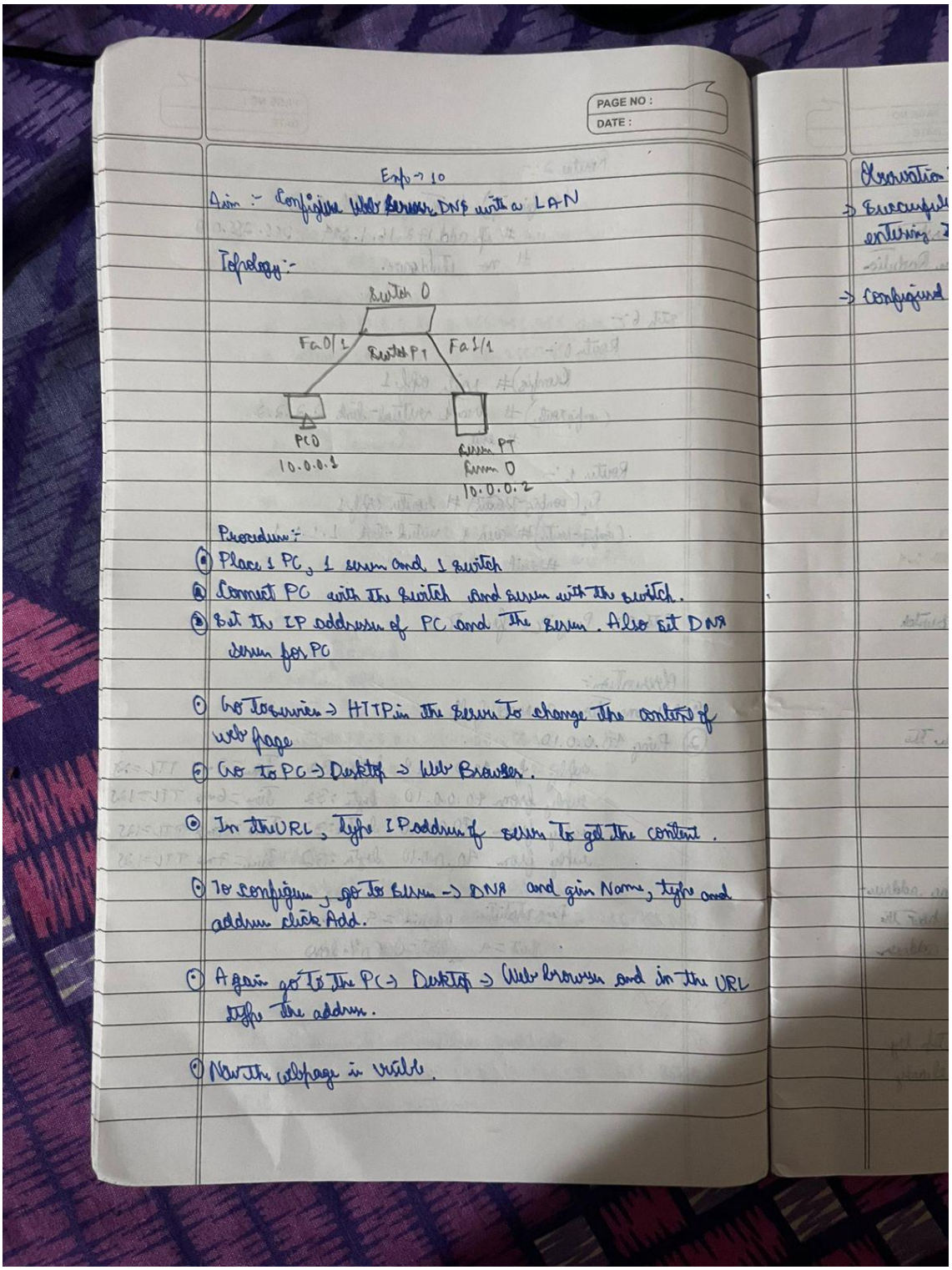
Screen Shots:

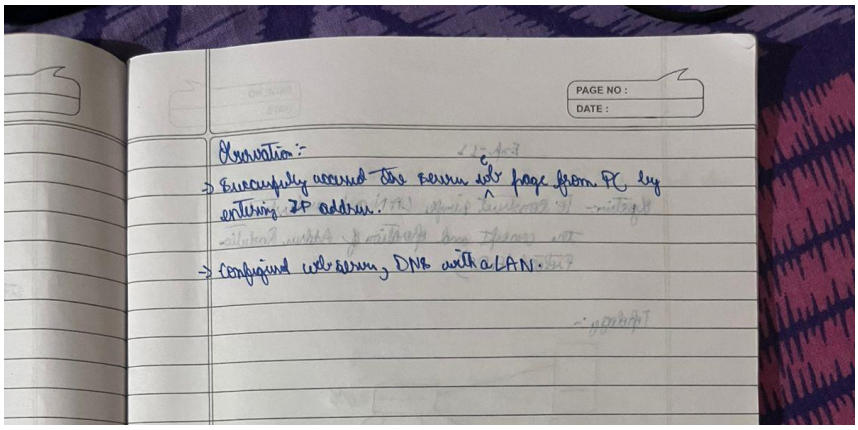


Program 9

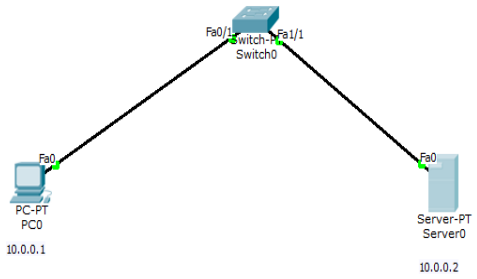
Aim: Configure Web Server, DNS within a LAN.

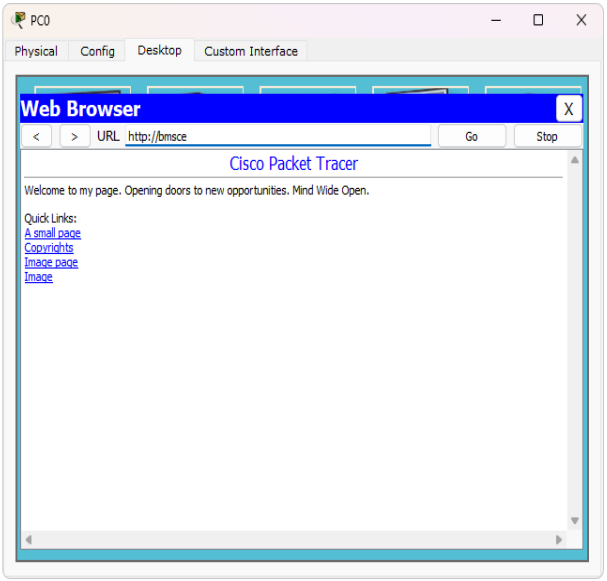
Topology , Procedure and Observation:





Screen Shots:

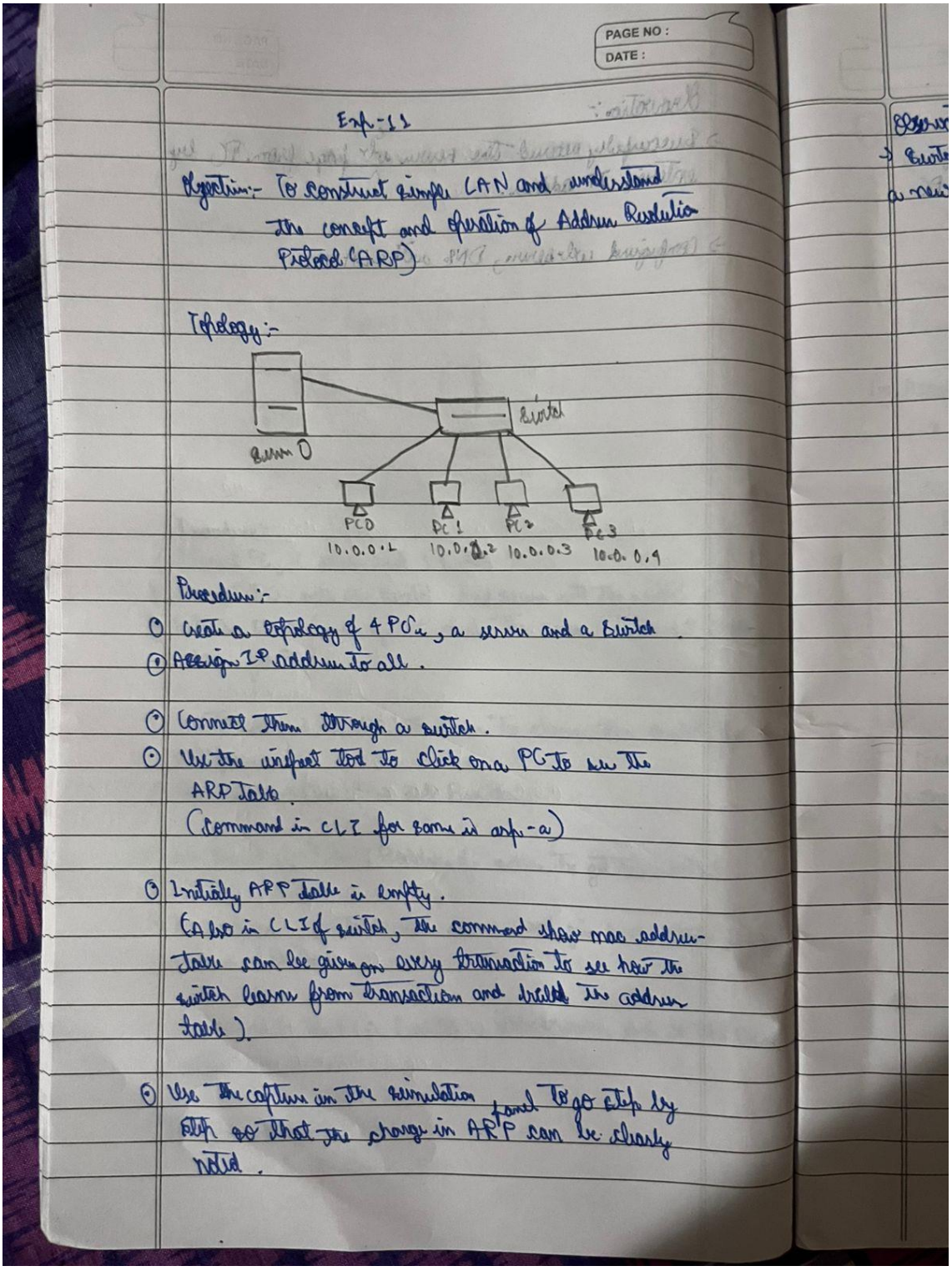


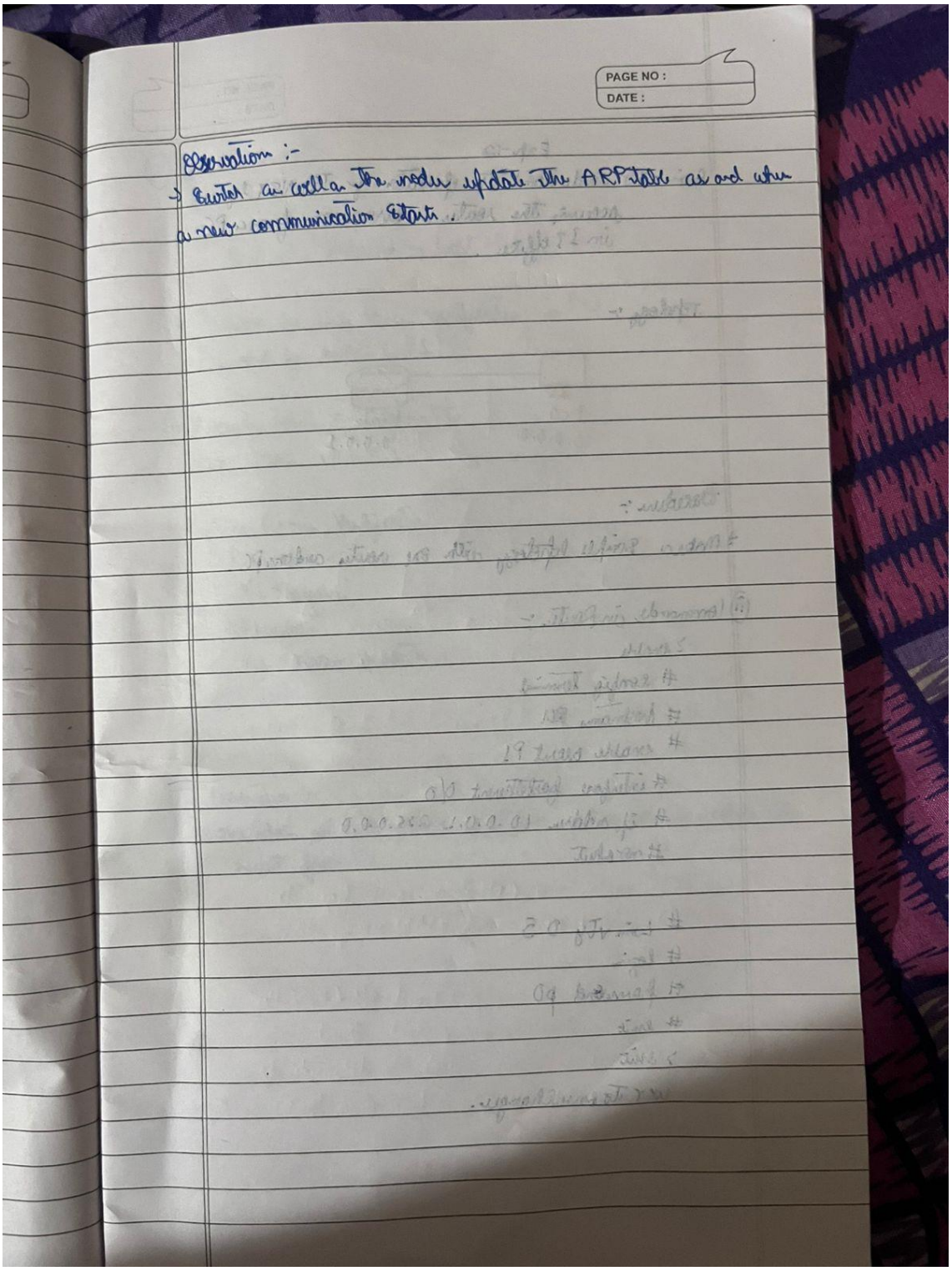


Program 10

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

Topology , Procedure and Observation:

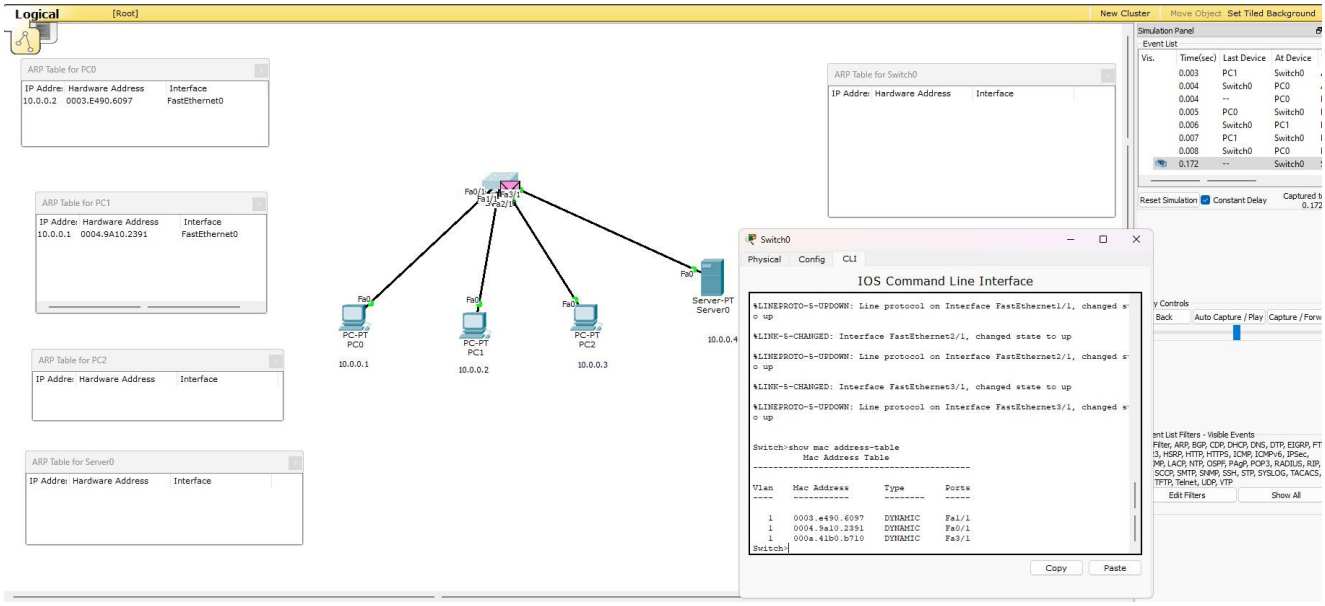


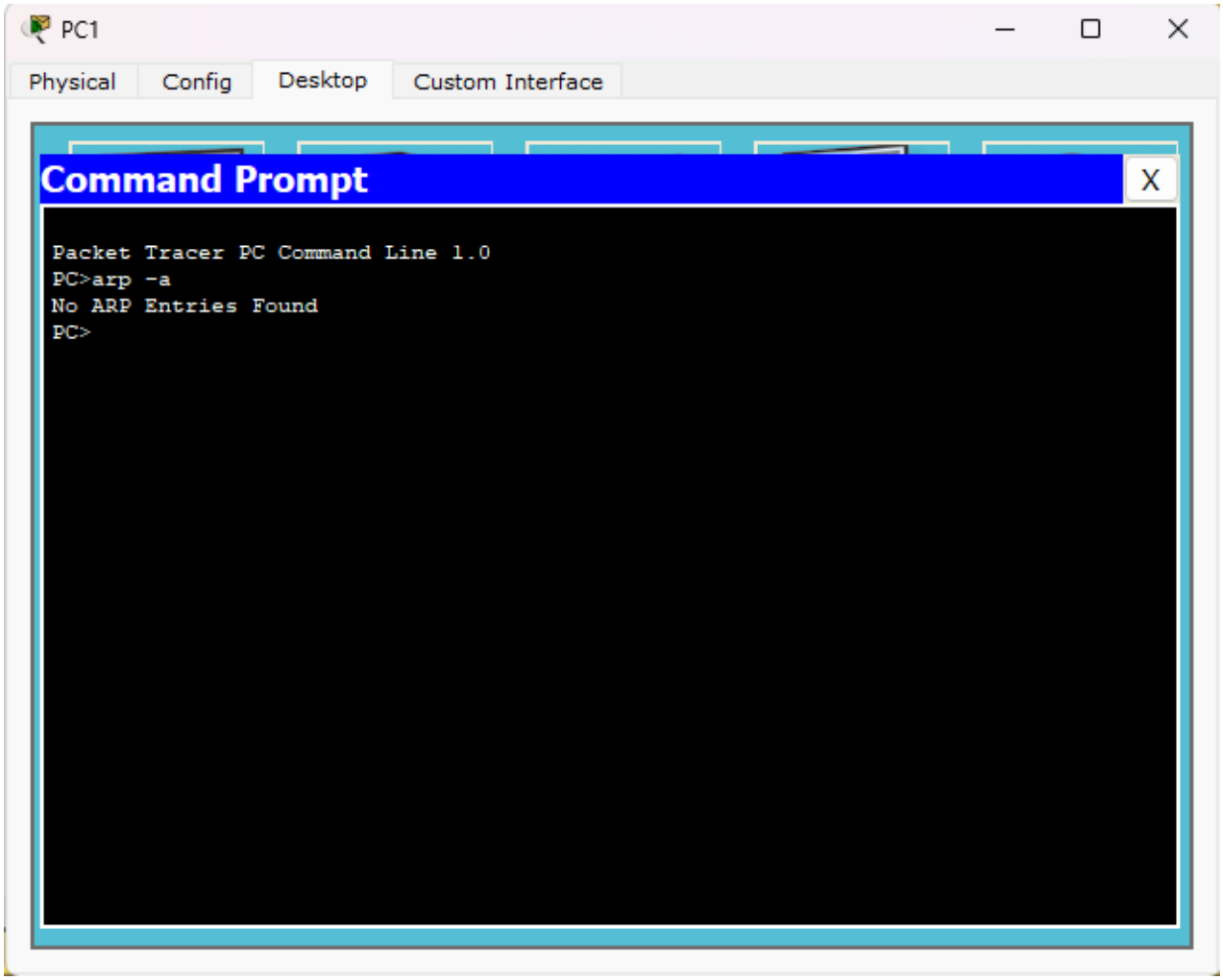

PAGE NO :
DATE :

Observation :-

→ Switch as well as the nodes update the ARP table as and when a new communication starts.

Screen Shots:

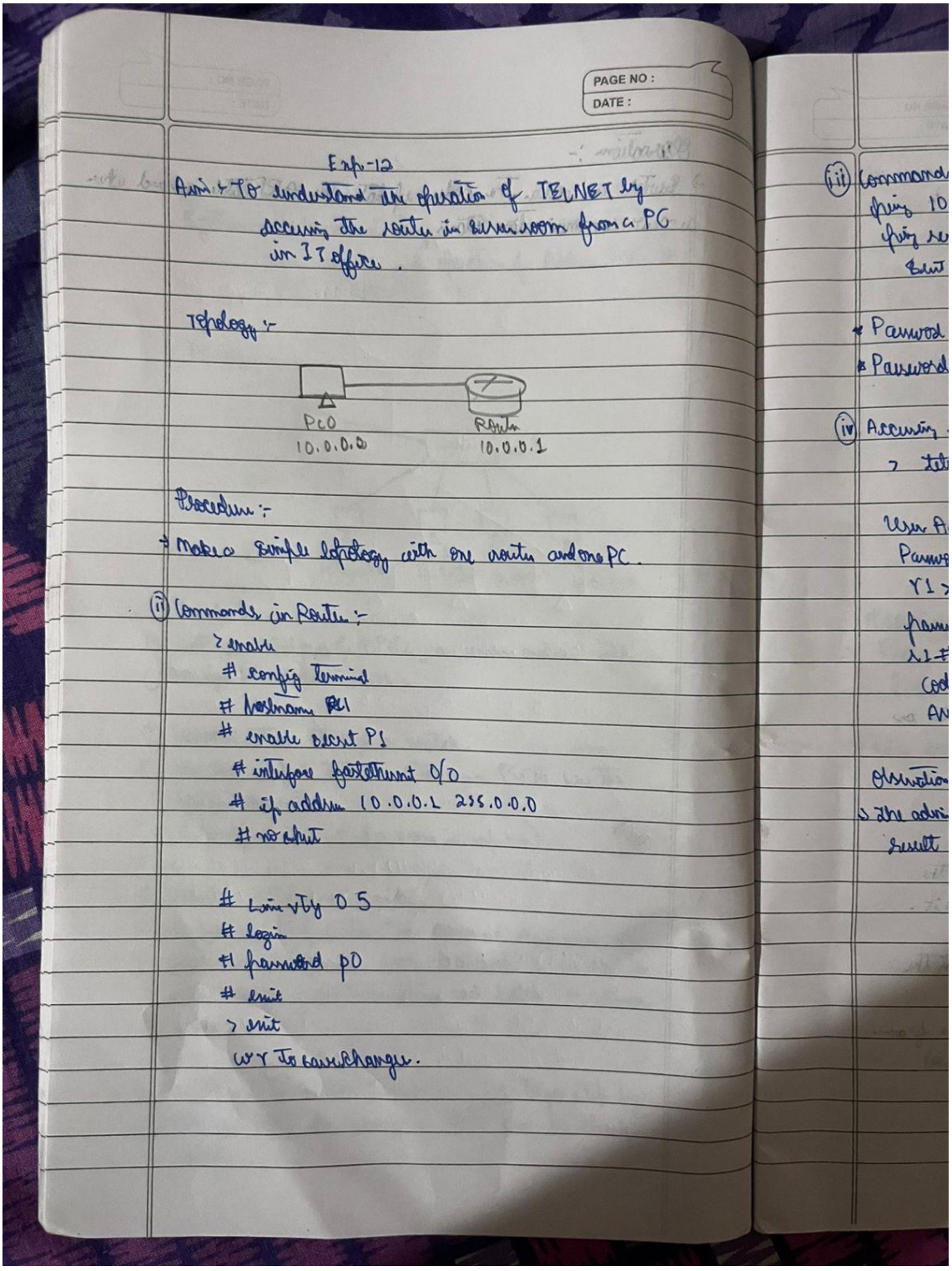


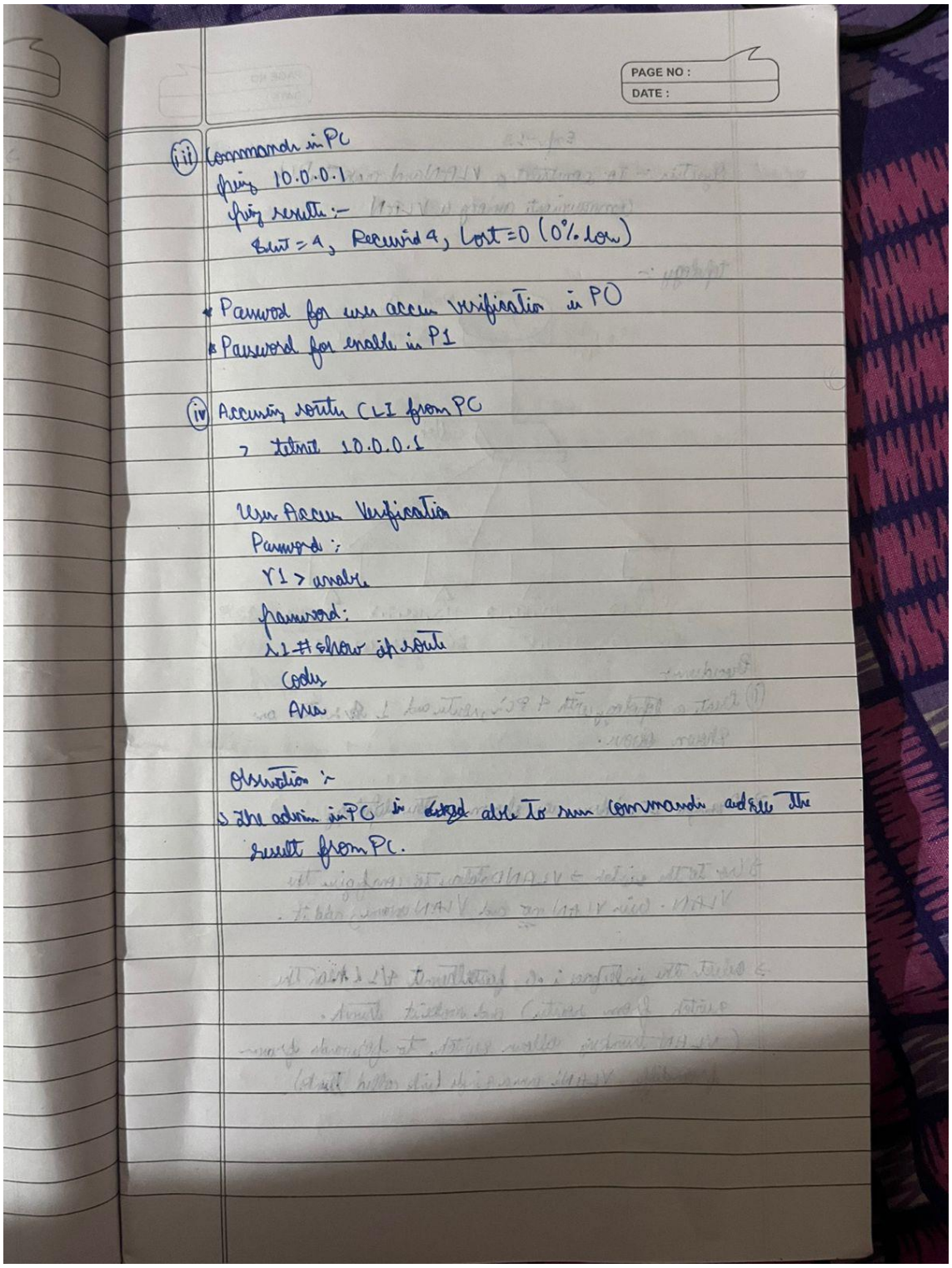


Program 11

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

Topology , Procedure and Observation:





(iii) Commands in PC
ping 10.0.0.1
ping results :-
Sent = 4, Received 4, Lost = 0 (0% loss)

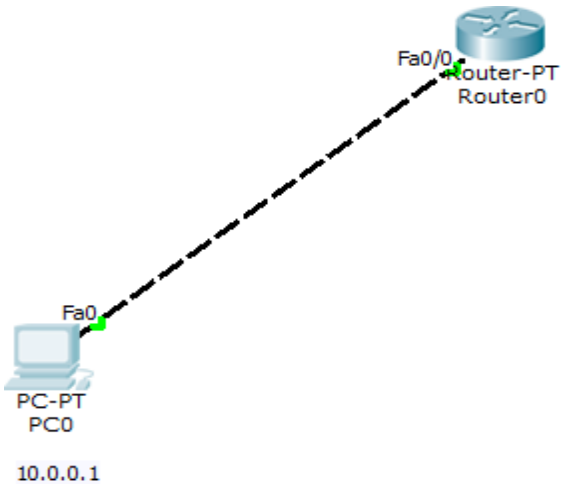
* Password for user access verification in P0
* Password for enable in P1

(iv) Accessing router CLI from PC
> telnet 10.0.0.1

User Access Verification
Password :
R1 > enable
Password:
R1 # show ip route
Codes
Area

Observation :-
The admin in PC is able to run commands and see the results from PC.

Screen Shots:



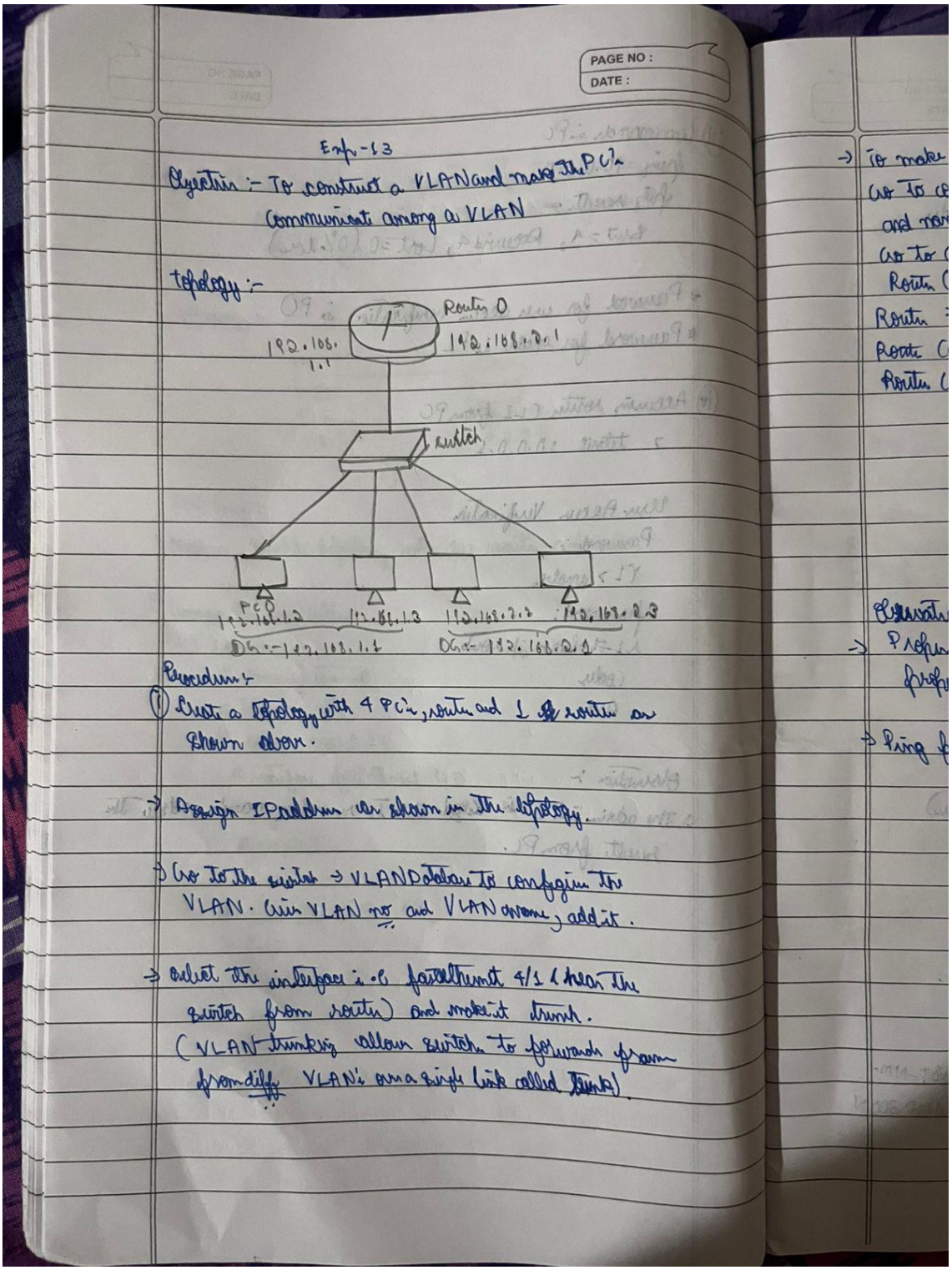
Command Prompt

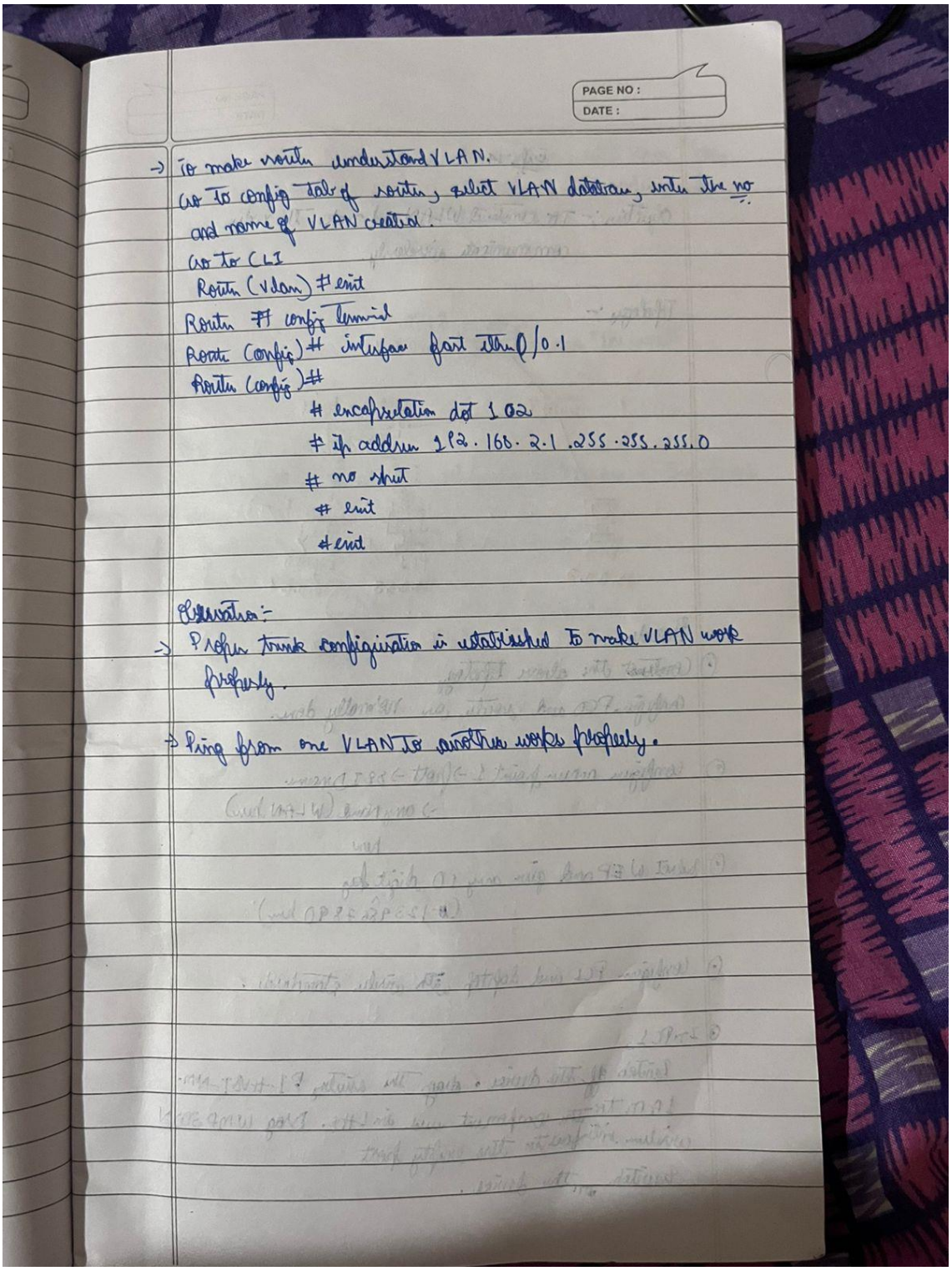
```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>telnet 10.0.0.2
Trying 10.0.0.2 ...Open


User Access Verification

Password:
R1>enable
Password:
R1#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
R1#
```

Program 12

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

Topology , Procedure and Observation:





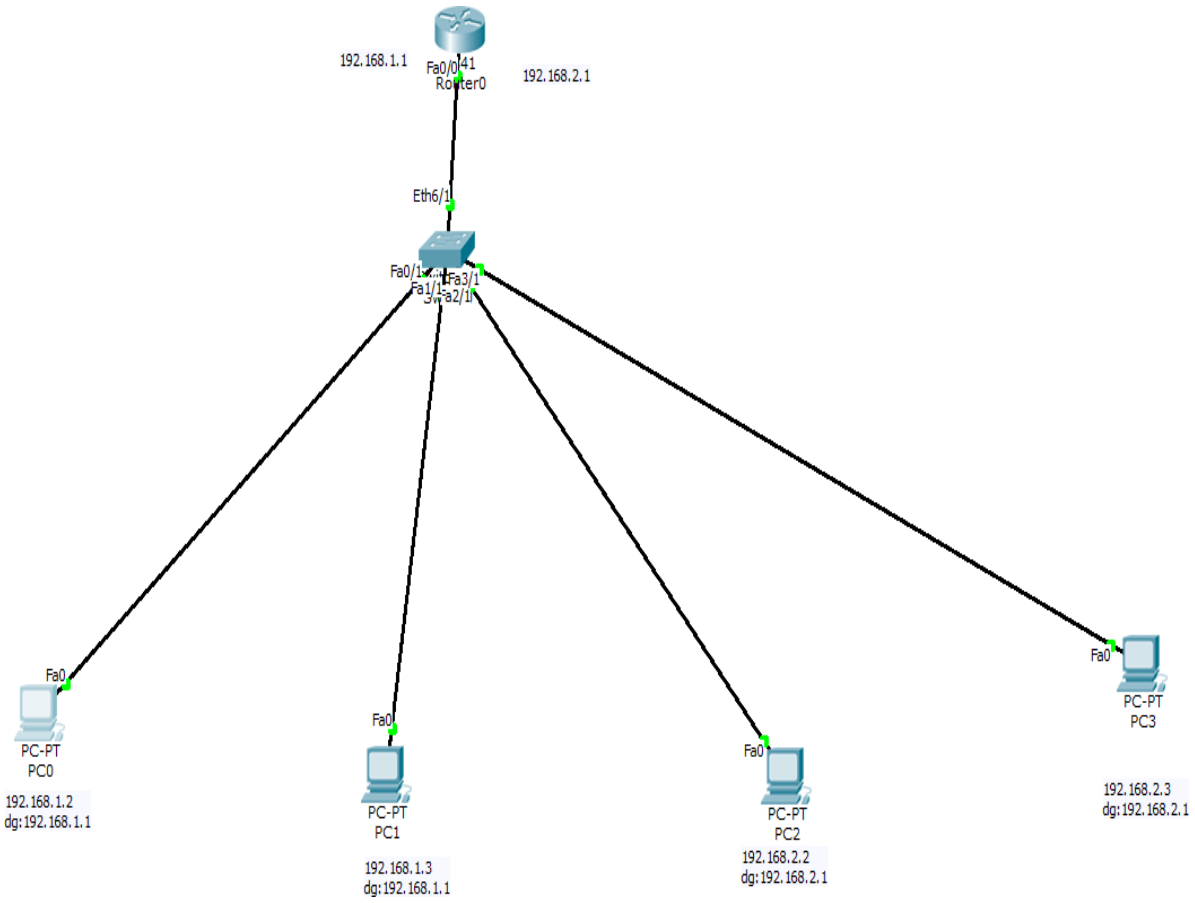
PAGE NO :
DATE :

→ To make router understand VLAN.
Go to config tab of router, select VLAN database, write the no and name of VLAN created.
Go to CLI
Router (vlan) # exit
Router # config terminal
Router (config) # interface fast ethernet 0/0.1
Router (config) #
encapsulation dot1Q 2
ip address 192.168.2.1 255.255.255.0
no shut
exit
exit

Observation :
→ Proper trunk configuration is established to make VLAN work properly.

→ Ping from one VLAN to another works properly.

Screen Shots:



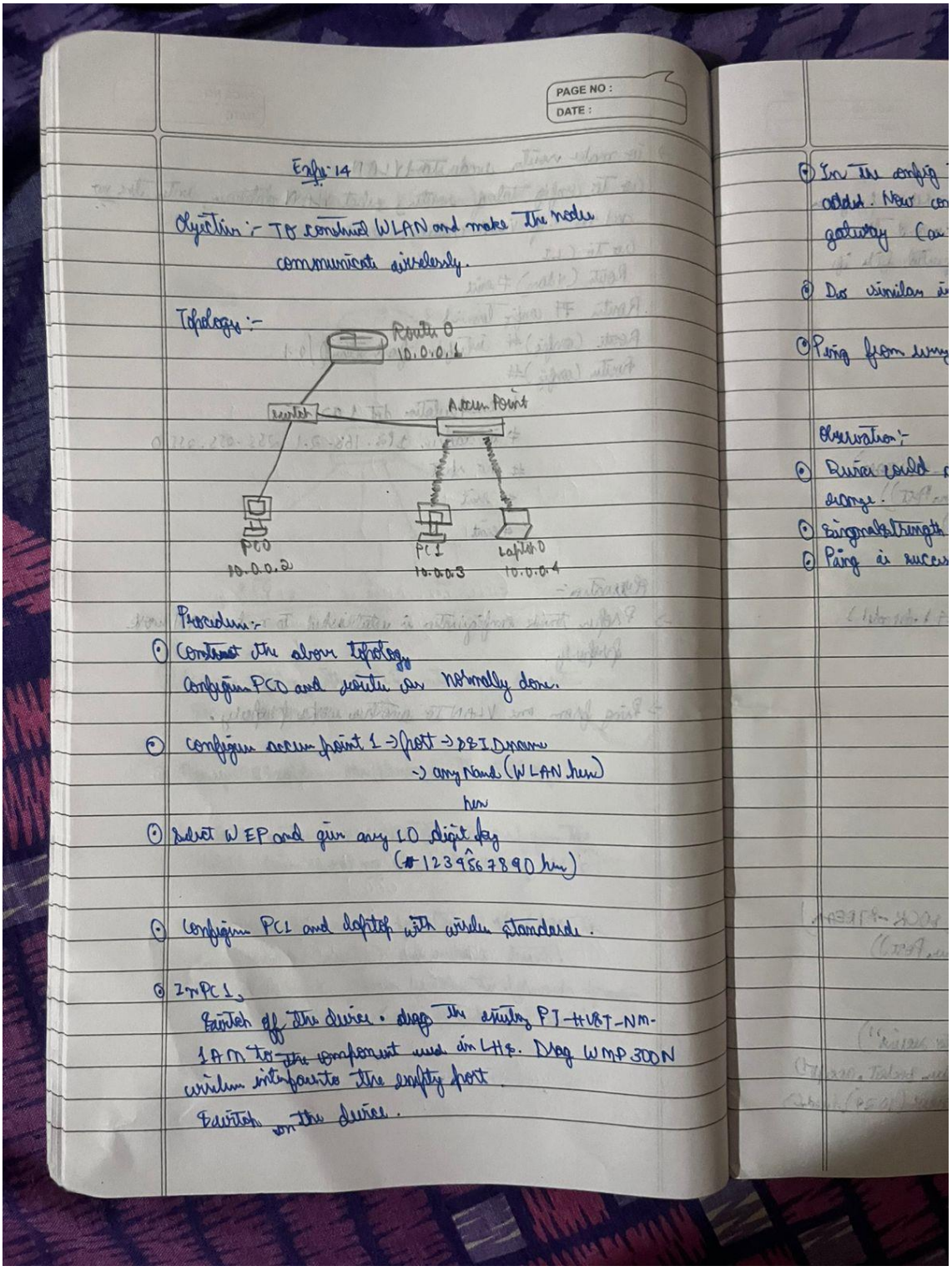
Command Prompt

```
Packet Tracer PC Command Line 1.0
PC>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Request timed out.
Reply from 192.168.2.2: bytes=32 time=0ms TTL=127
Reply from 192.168.2.2: bytes=32 time=0ms TTL=127
Reply from 192.168.2.2: bytes=32 time=4ms TTL=127

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 4ms, Average = 1ms

PC>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Reply from 192.168.2.2: bytes=32 time=0ms TTL=127
Reply from 192.168.2.2: bytes=32 time=0ms TTL=127
Reply from 192.168.2.2: bytes=32 time=2ms TTL=127
Reply from 192.168.2.2: bytes=32 time=0ms TTL=127

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms

PC>ping 192.168.2.3

Pinging 192.168.2.3 with 32 bytes of data:

Request timed out.
Reply from 192.168.2.3: bytes=32 time=3ms TTL=127
Reply from 192.168.2.3: bytes=32 time=2ms TTL=127
Reply from 192.168.2.3: bytes=32 time=1ms TTL=127

Ping statistics for 192.168.2.3:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 3ms, Average = 2ms

PC>ping 192.168.2.3

Pinging 192.168.2.3 with 32 bytes of data:

Reply from 192.168.2.3: bytes=32 time=0ms TTL=127
Reply from 192.168.2.3: bytes=32 time=0ms TTL=127
Reply from 192.168.2.3: bytes=32 time=2ms TTL=127
Reply from 192.168.2.3: bytes=32 time=0ms TTL=127

Ping statistics for 192.168.2.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms

PC>
```

Program 13

Aim: To construct a WLAN and make the nodes communicate wirelessly.

Topology , Procedure and Observation:



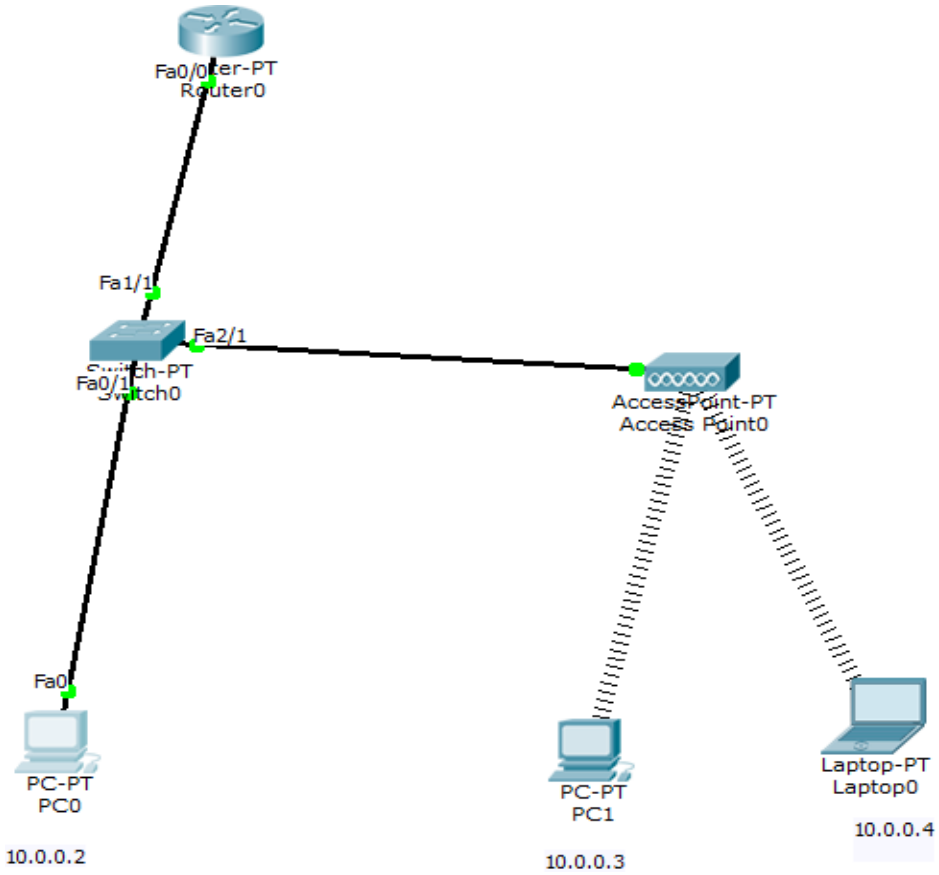
PAGE NO :
DATE :

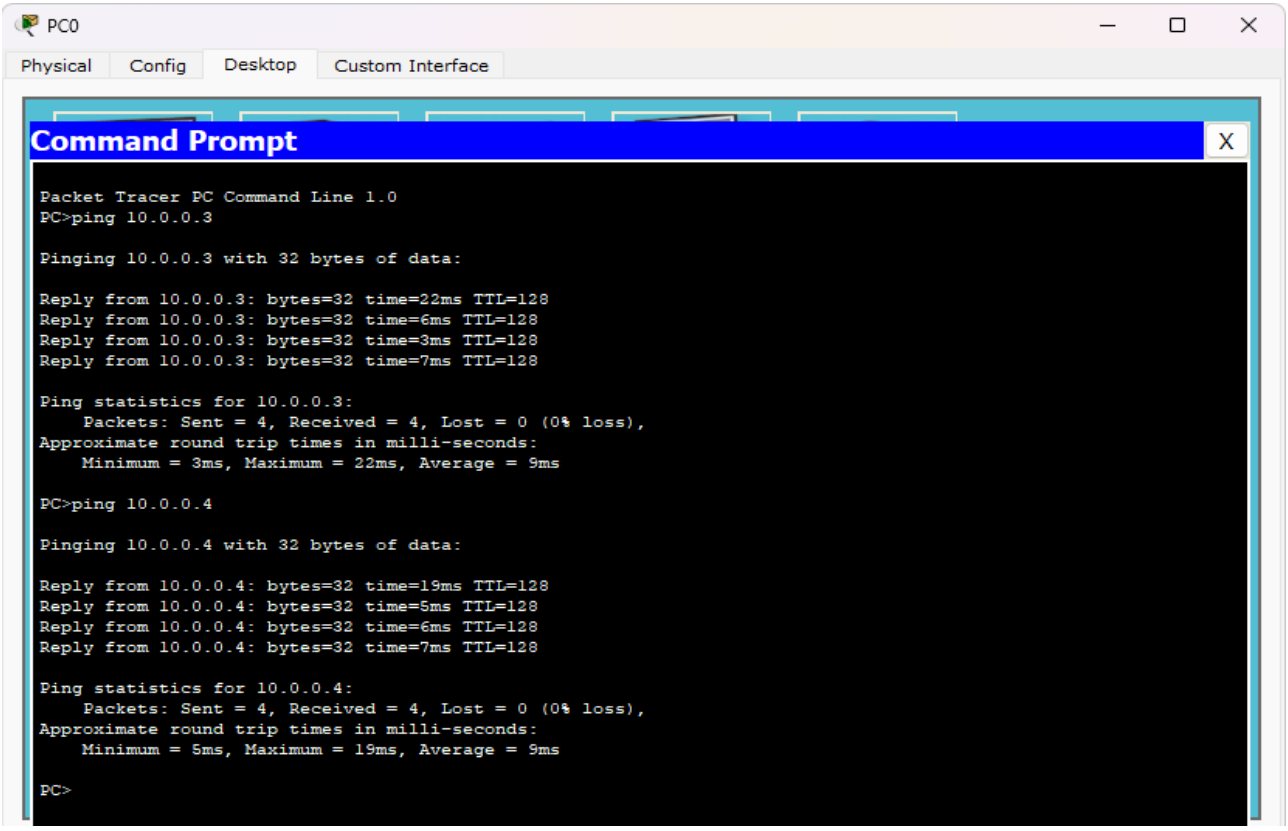
- In the config tab a new wireless interface would have been added. Now configure SSID, WEP key, IP address and gateway (as normally done) to the device.
- Do similar in laptop.
- Ping from any device to every other to check the result.

Observation :-

- Device could connect to WLAN as long as they are in the network range.
- Signal strength decrease with increase in distance.
- Ping is successful.

Screen Shots:



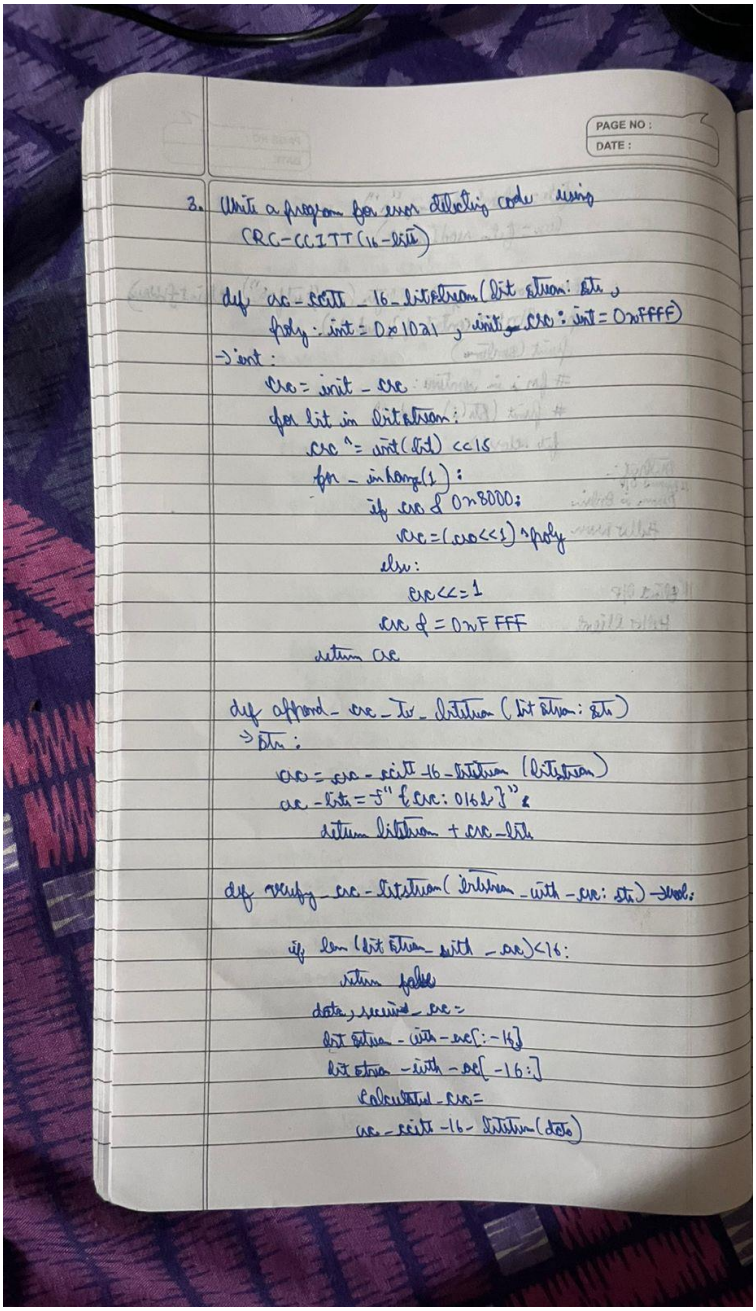


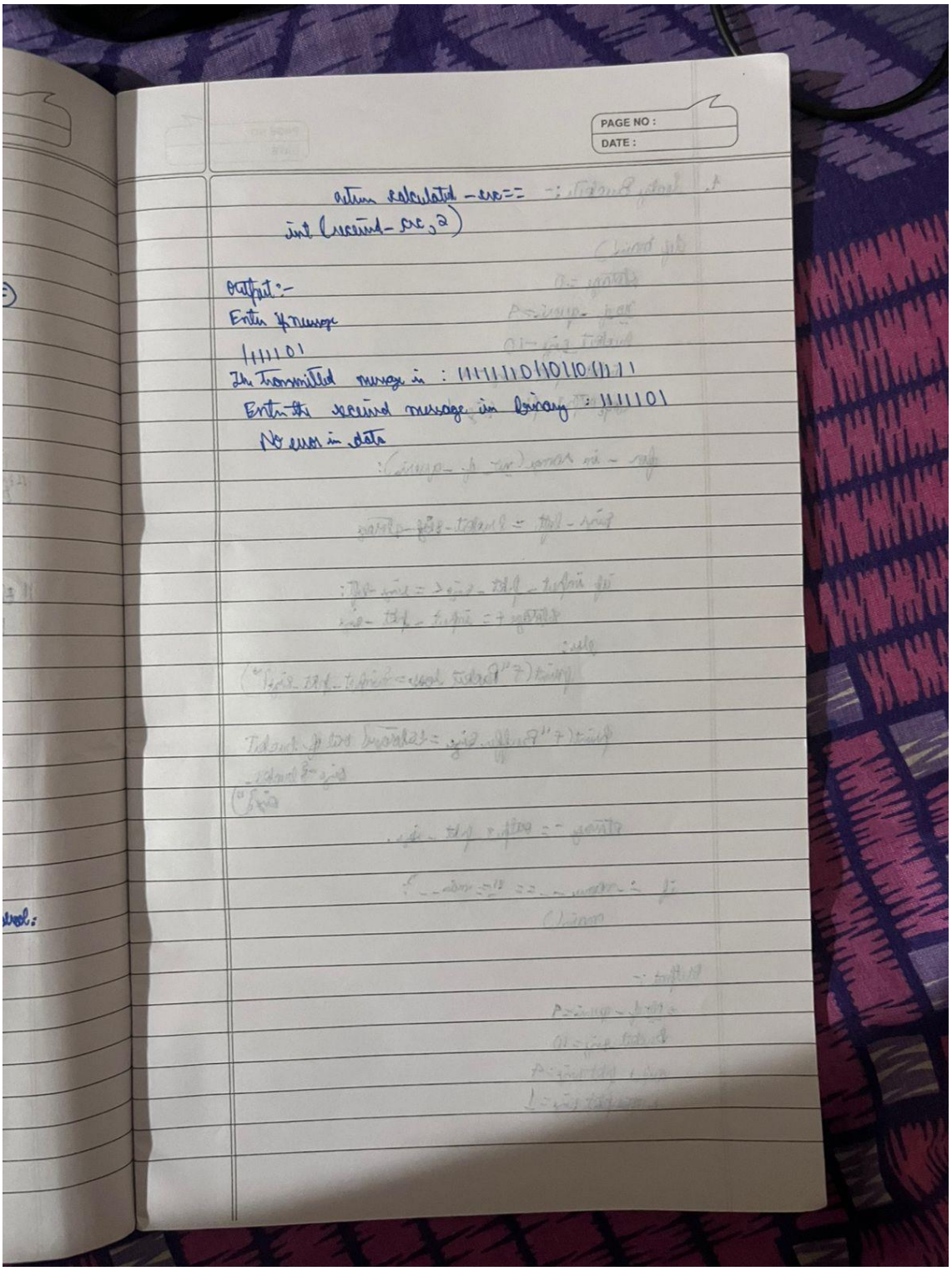
PART-B

Program 14

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

Code and Output:





PAGE NO :
DATE :

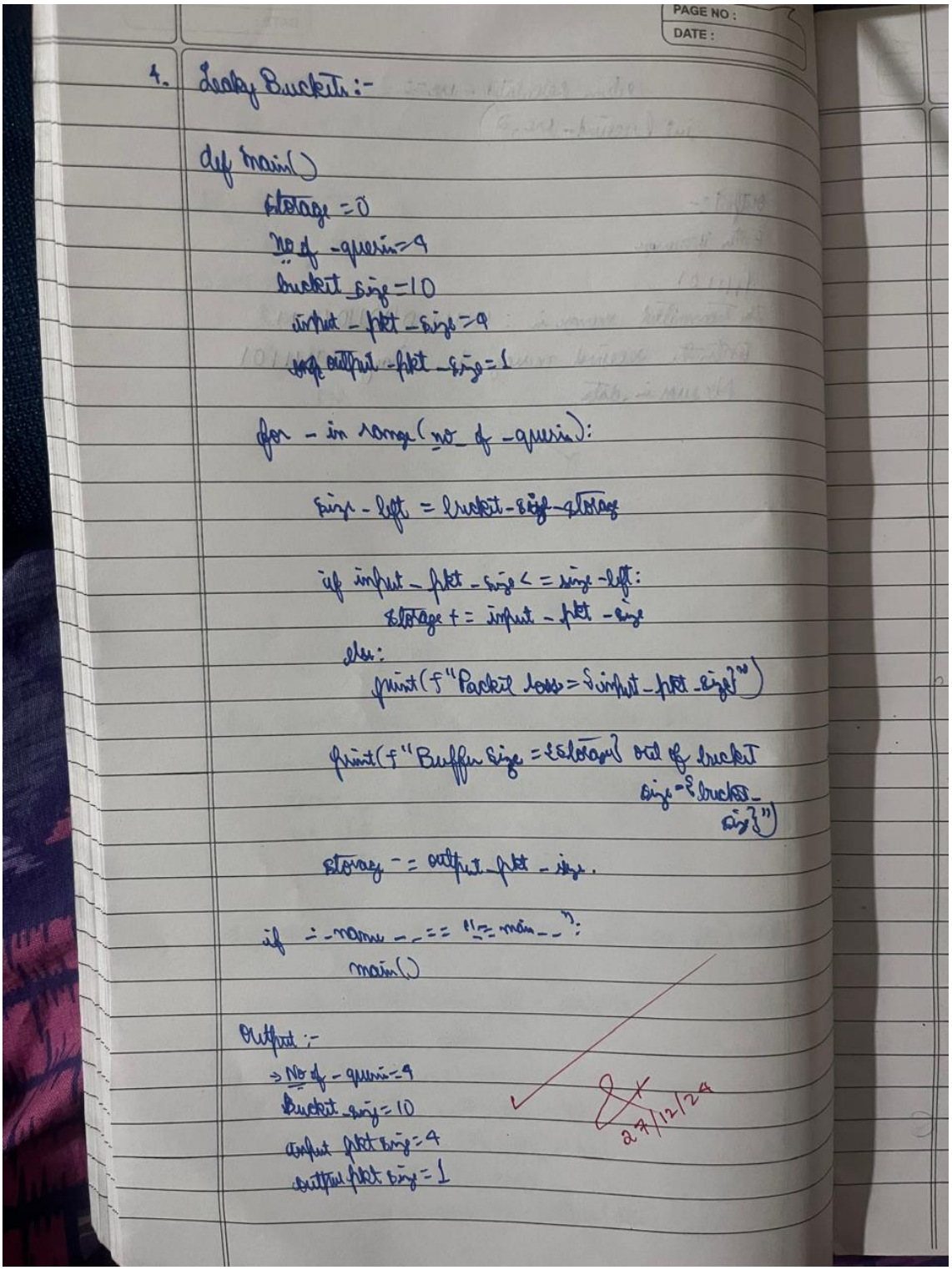
return calculated_crc;
int (received_crc, 2)

Output:-
Enter the message
111101
The transmitted message is : 11111101101010111
Enter the received message in binary : 111101
No error in data

Program 15

Write a program for congestion control using Leaky bucket algorithm.

Code and Output:



1. Leaky Bucket :-

```
def main()
    storage = 0
    no_of_queries = 4
    bucket_size = 10
    input_pkt_size = 4
    output_pkt_size = 1

for _ in range(no_of_queries):

    size_left = bucket_size - storage

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

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

    storage -= output_pkt_size.

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

Output :-

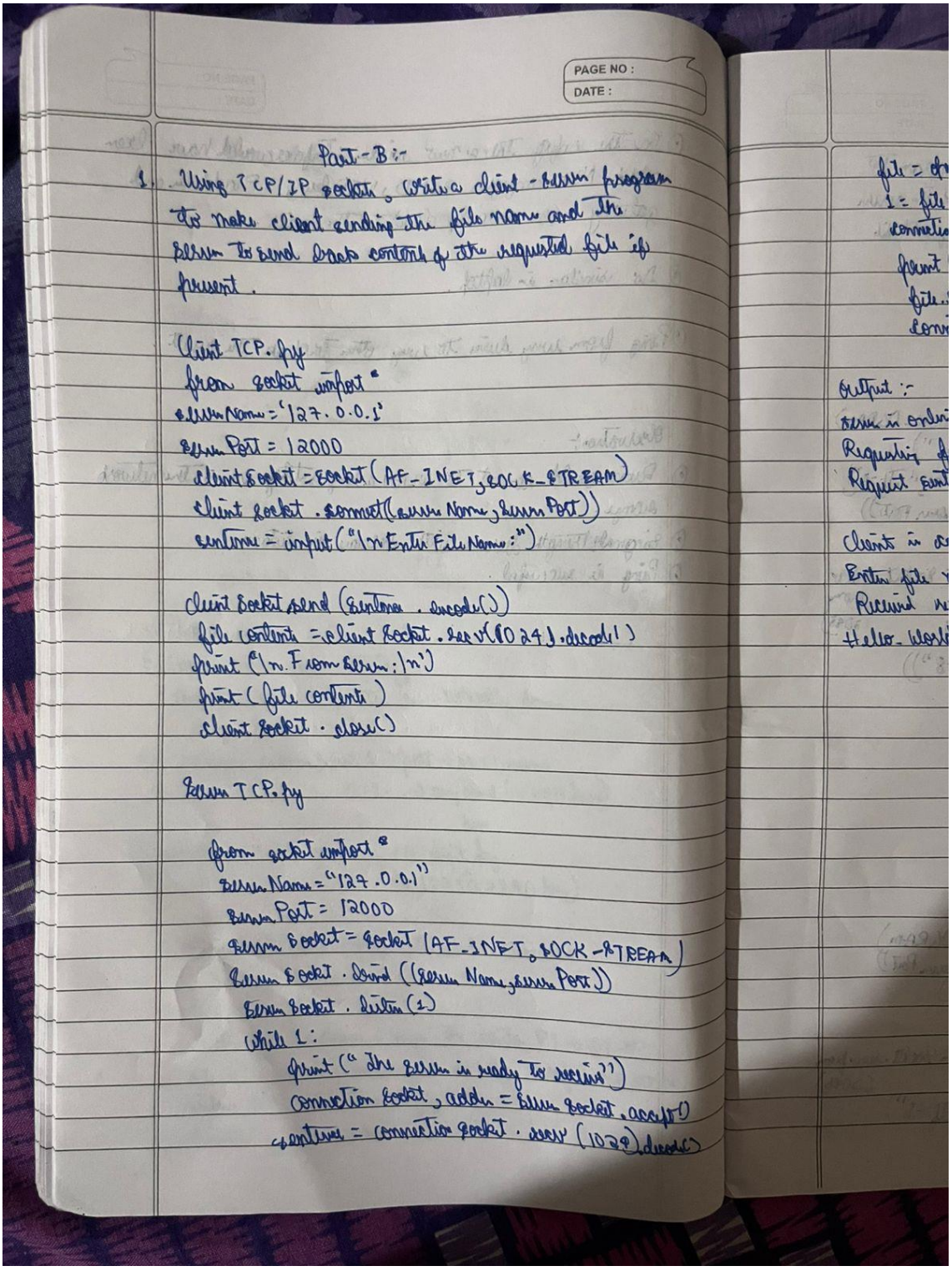
→ No of queries = 4
Bucket size = 10
input pkt size = 4
output pkt size = 1

27/12/24

Program 16

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

Code and Output:



Part - B :-

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

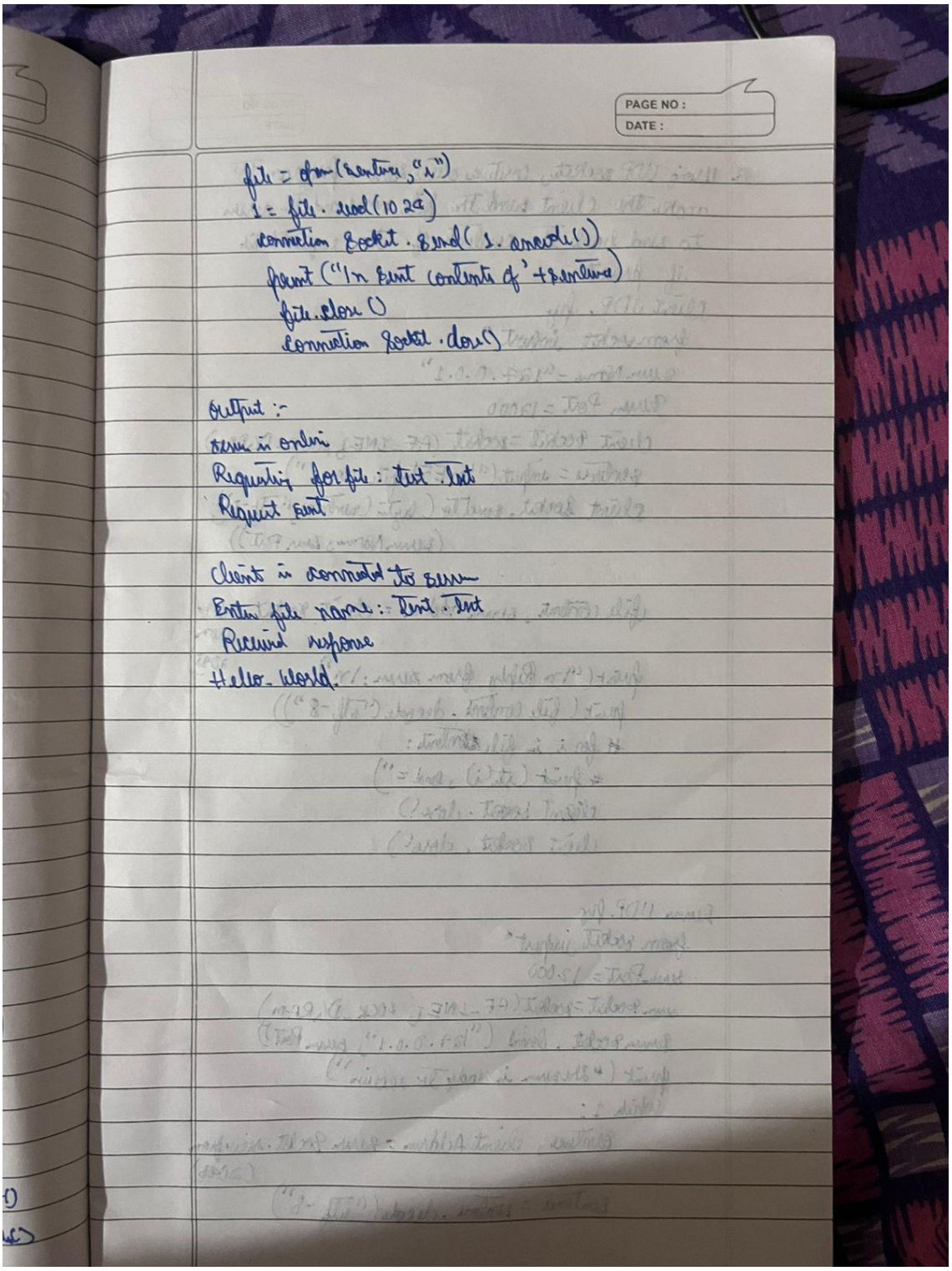
Client TCP.py

```
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(10241).decode()
print("\nFrom Server:\n")
print(fileContents)
clientSocket.close()
```

Server TCP.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()
```

Output :-



PAGE NO :
DATE :

```
file = open(filename, "r")
l = file.read(1024)
connection_socket.send(l.encode())
print("\n Sent contents of " + filename)
file.close()
connection_socket.close()
```

Output :-

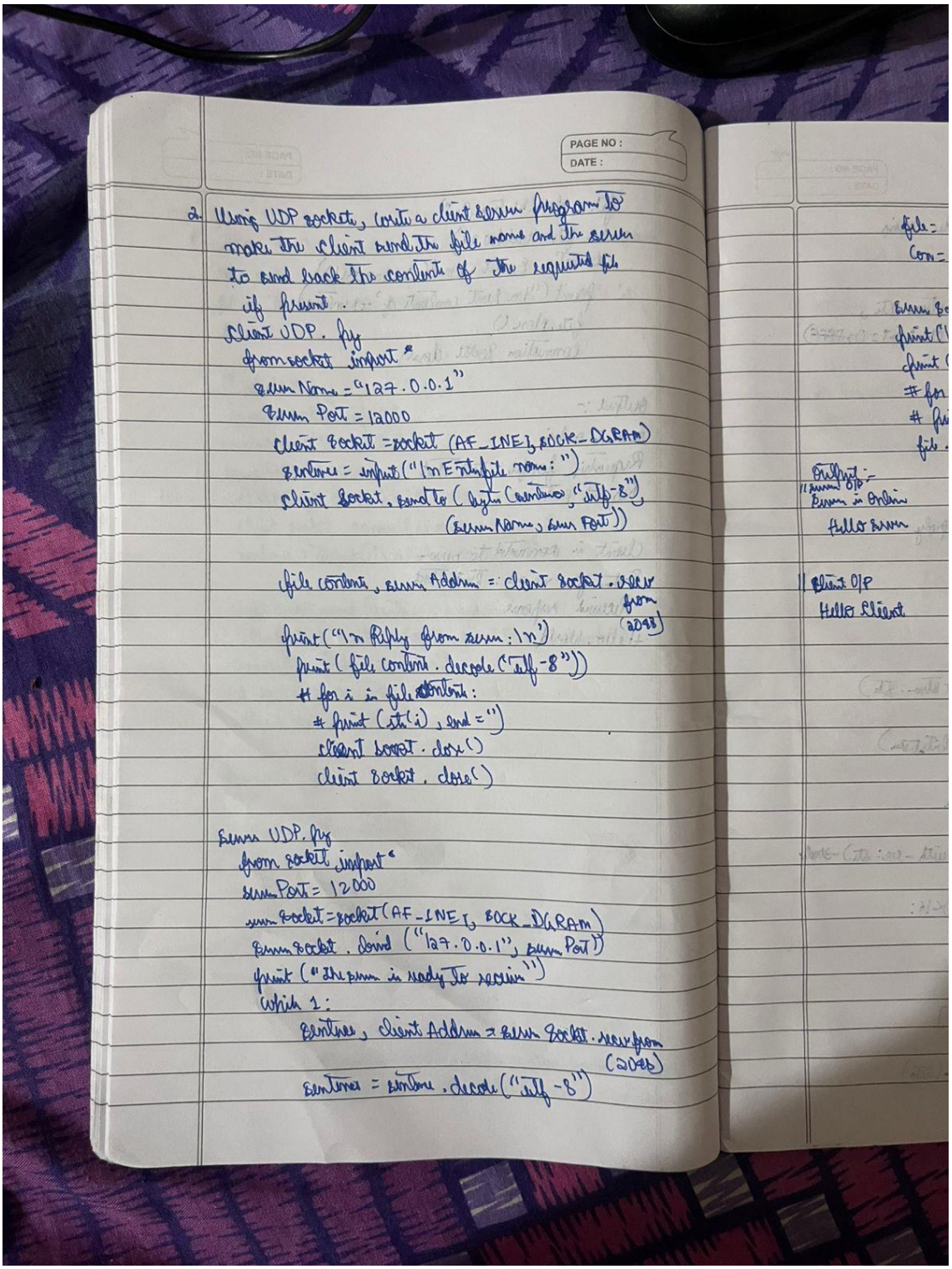
server is ready
Requesting for file : test.txt
Request sent

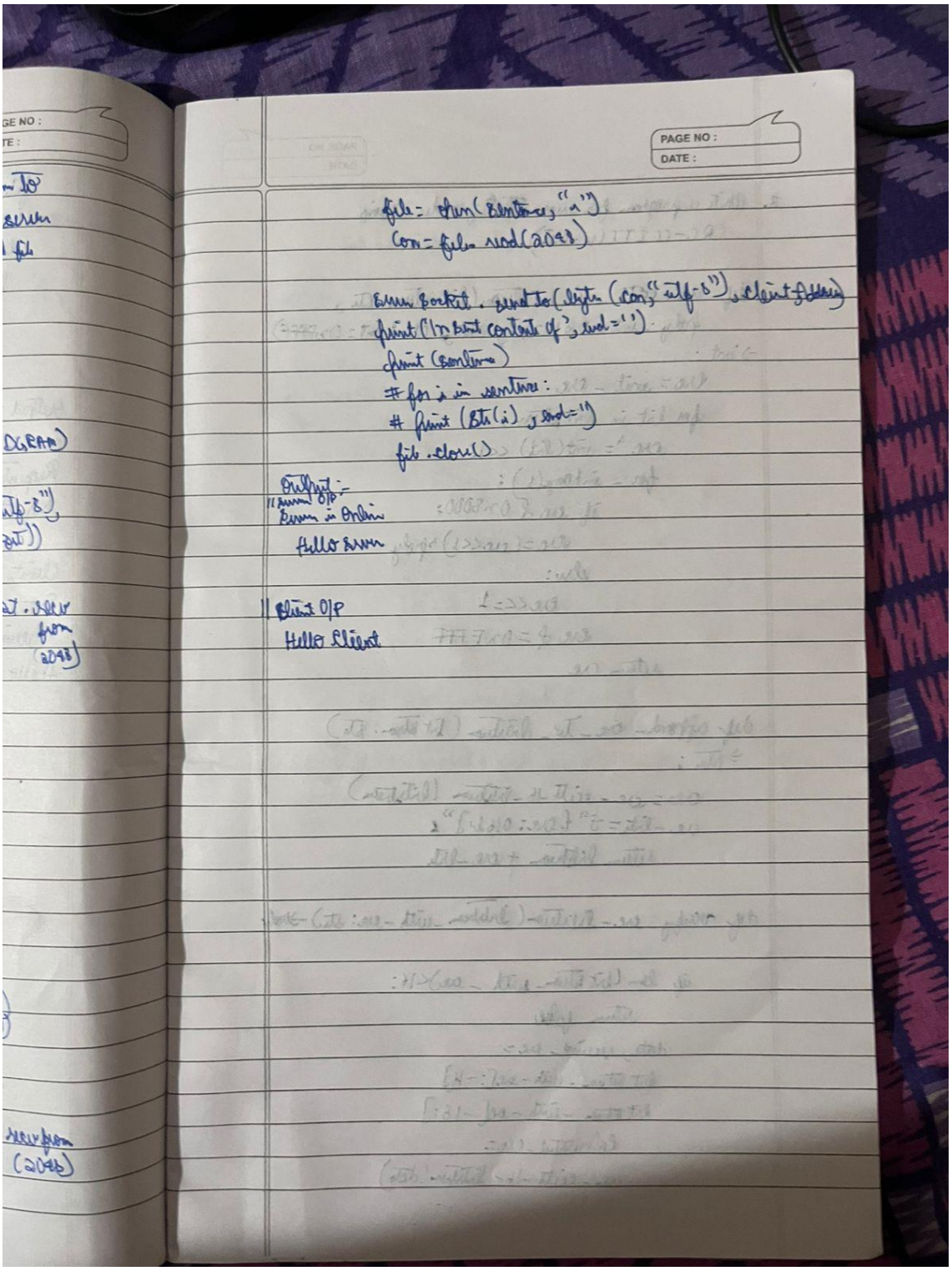
Client is connected to server
Enter file name : test.txt
Received response
Hello World

Program 17

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

Code and Output:





PAGE NO :
DATE :

```
file = open(filename, "a")
con = file.read(2048)

Server_socket.send_to(bytes(con, "utf-8"), client_address)
print("\n Data content of", end="")
print(filename)
# for i in contents:
# print(str(i), end="")
file.close()
```

Output :-
// Server O/P
Server is Online
Hello Server

// Client O/P
Hello Client