VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB RECORD

Computer Network Lab (23CS5PCCON)

Submitted by

Priyanshu Kumar (1BM22CS210)

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 **Priyanshu Kumar (1BM22CS210),** 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.

Surabhi S Assistant Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE

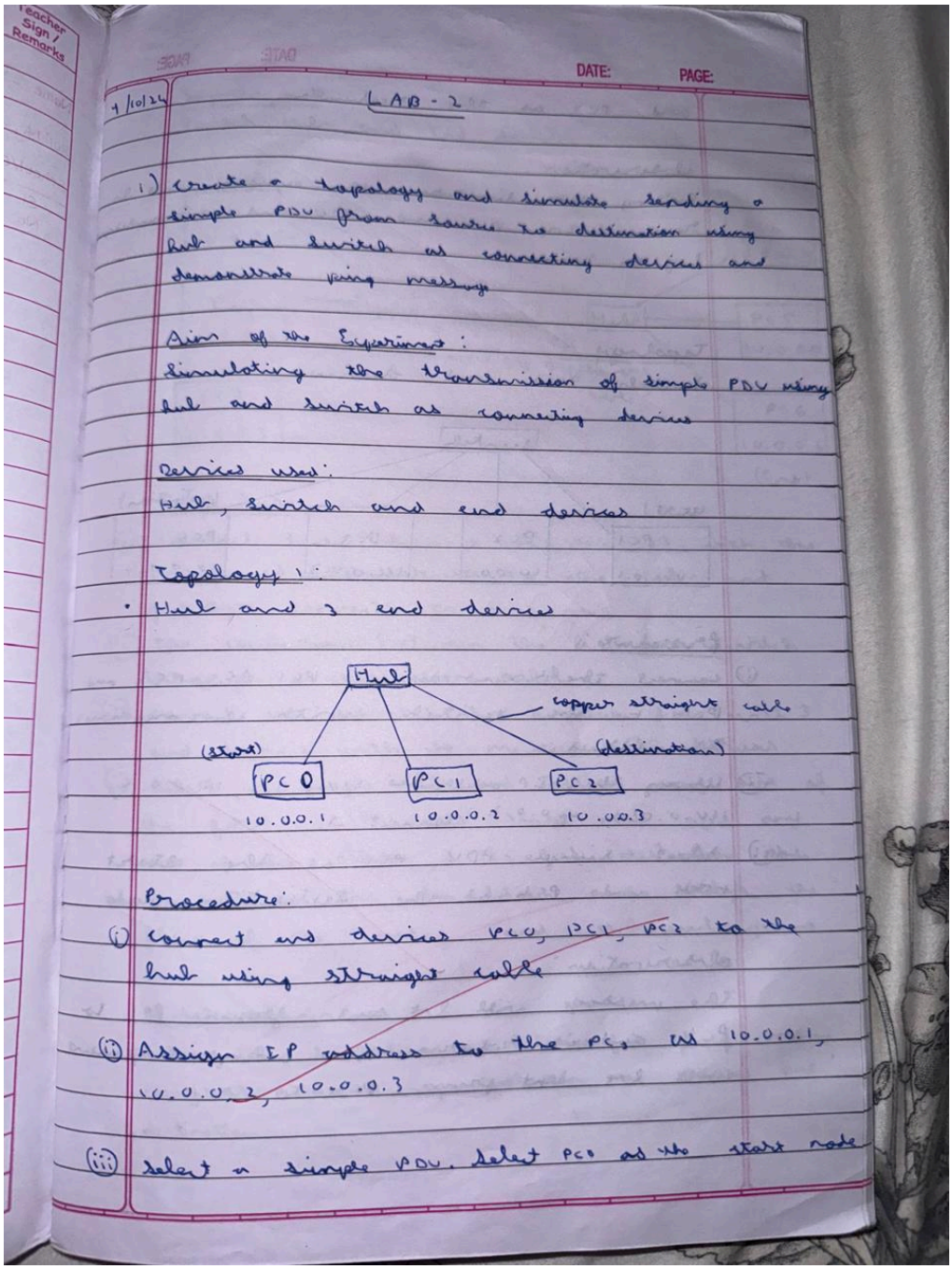
Index

Github Link:
https://github.com/Tanishmv/CN_Lab

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.

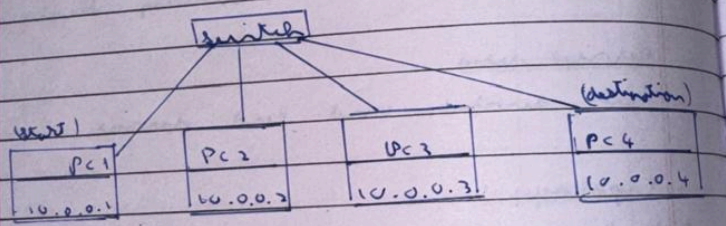
Topology , Procedure and Observation:



and PC3 as the destination

Observation:
During the simulation, the message will be received by PC2 and PC0 and acknowledged by the same.

Topology 2:
Switch and end device



Procedure:
(i) connect the 4 end devices PC1, PC2, PC3 and PC4 to the switch with the mentioned IP address.
(ii) Using the IP address 10.0.0.1, 10.0.0.2, 10.0.0.3, 10.0.0.4 connect it
(iii) select simple PDU, PC1 as the start node and PC4 as the destination node

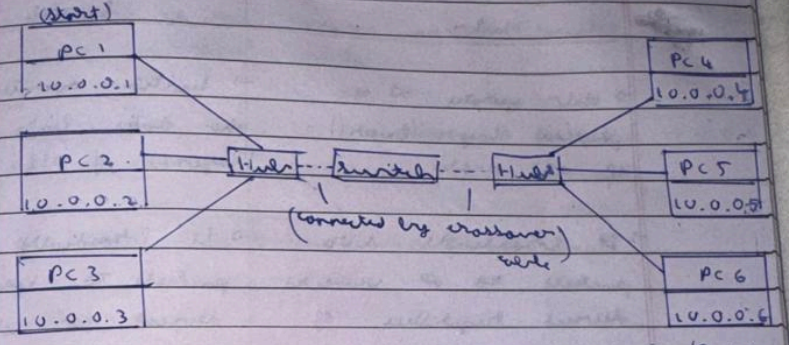
Observation:
The message will be sent from PC1 to PC4 and in return the acknowledgement will be sent from PC4 to PC1

PAGE:

DATE: PAGE:

Topology 3:

Switch, Hub and end devices



Procedure:

(i) connect 3 end devices PC1, PC2, PC3 with the mentioned IP addresses to the Hub and further connected to a switch.

(ii) the connection between the hub and switch through a cross over cable

(iii) connect the switch to another hub with 3 end devices with the mentioned IP address

(iv) select a simple PDU and assign any one of the first 3 PCs as start node and any one of the other 3 PCs as destination node

(v) Demonstrate the simulation and analyse the flow of the message and acknowledgement

Observation:

During the simulation, a successful ping message confirms connectivity between the source and destination

DATE: PAGE:

Difference between Hub and Switch

Hub	Switch
→ Hub operates at the physical layer (layer 1) of OSI model	→ Switch operates at the data link layer (layer 2) of OSI model
→ It broadcasts data packets to all connected devices regardless of intended recipient	→ It broadcasts data packets to specific devices for which data is intended
→ It is less efficient and supports lower speeds	→ It is more efficient and supports higher speeds

16/10/24

Procedure:

(i) Select a ge

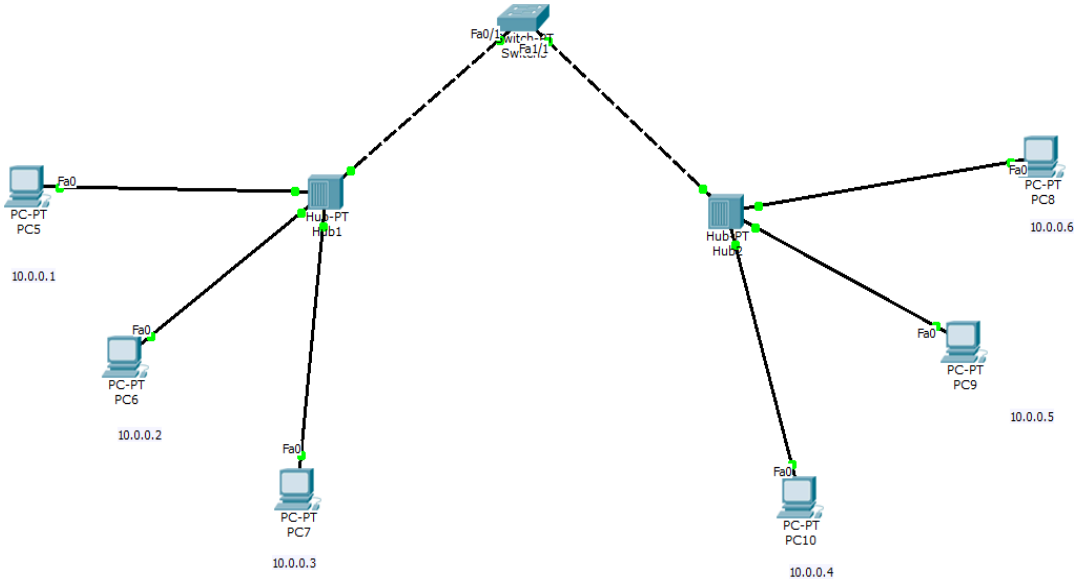
(ii) connect a copper cros

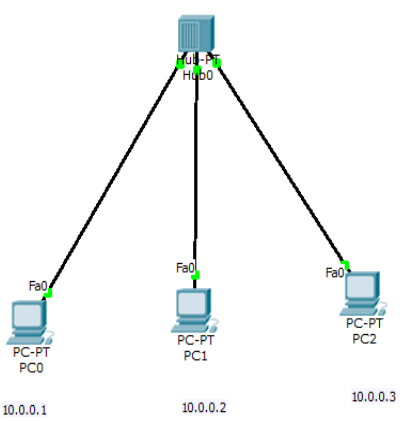
(iii) configure P and 10.0.0

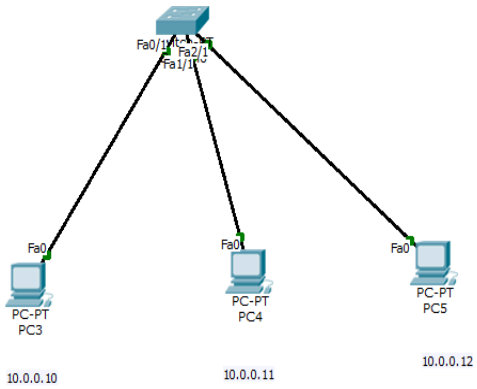
(iv) Select the

Router > en
Router #
Router (co
Router (confi
Router (conf
Router (co
Similarly

Screen Shots:



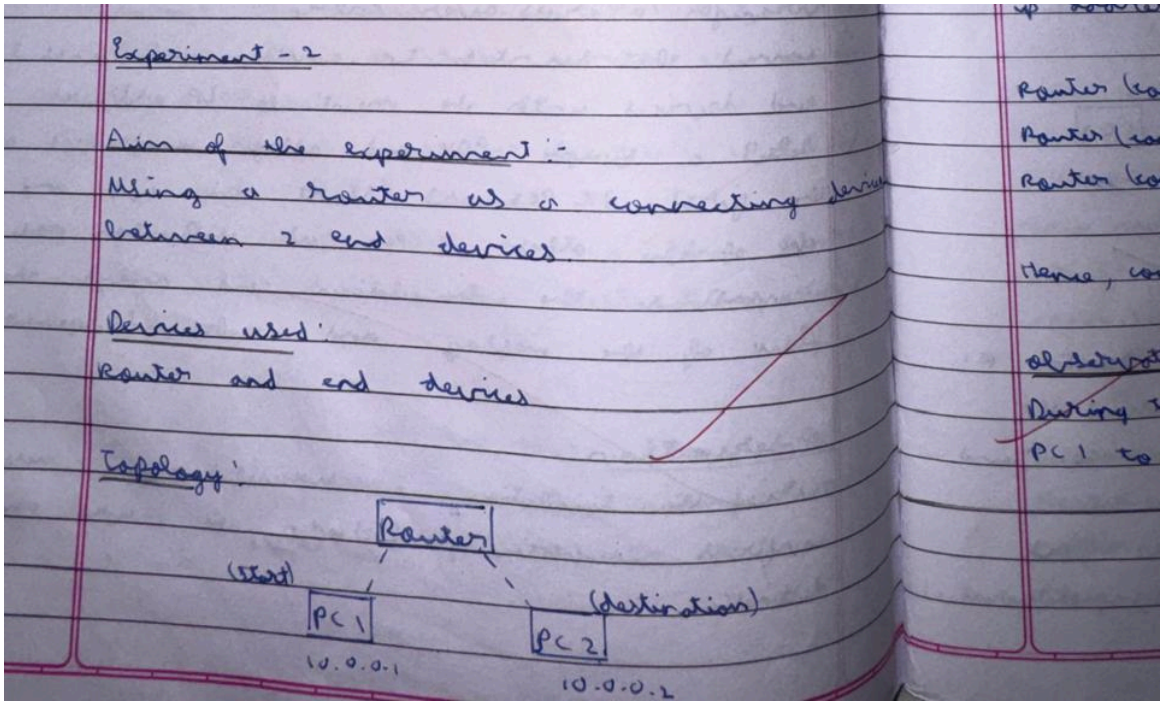




Program 2

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:



DATE: PAGE:

Procedure:

(i) Select a generic router R1

(ii) Connect 2 end devices to the router using copper cross over cables

(iii) Configure PC1 and PC2 with IP addresses 10.0.0.1 and 10.0.0.2

(iv) Select the router and go to the CLI:

Router > enable
Router # config terminal
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
Similarly do the same for PC2 but set the ip address as 10.0.0.2 this time in the CLI:

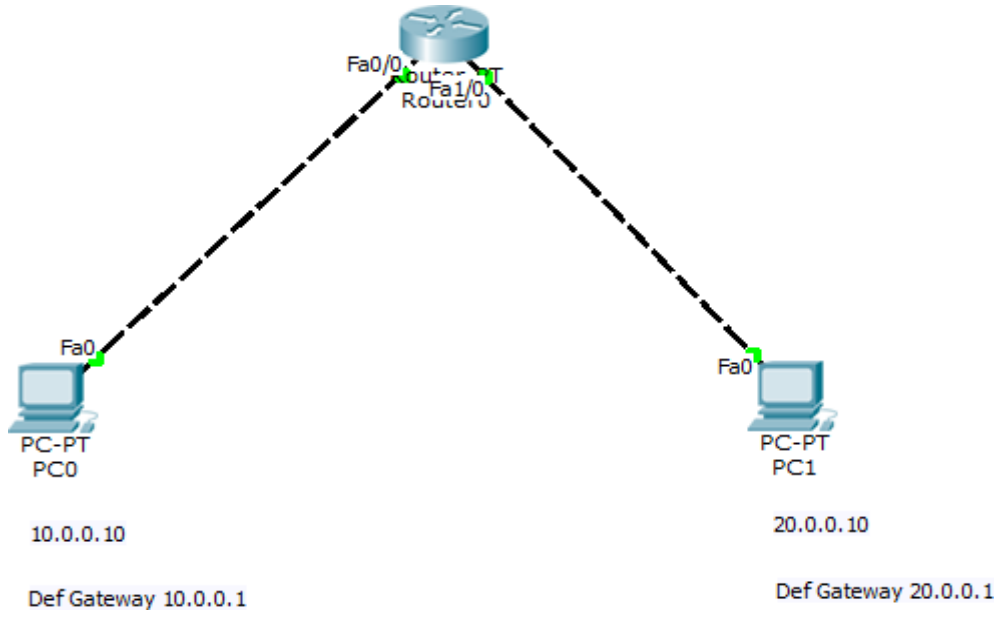
Router (config) # interface fastEthernet 1/0
Router (config-if) # ip address 10.0.0.2 255.0.0.0
Router (config-if) # no shutdown

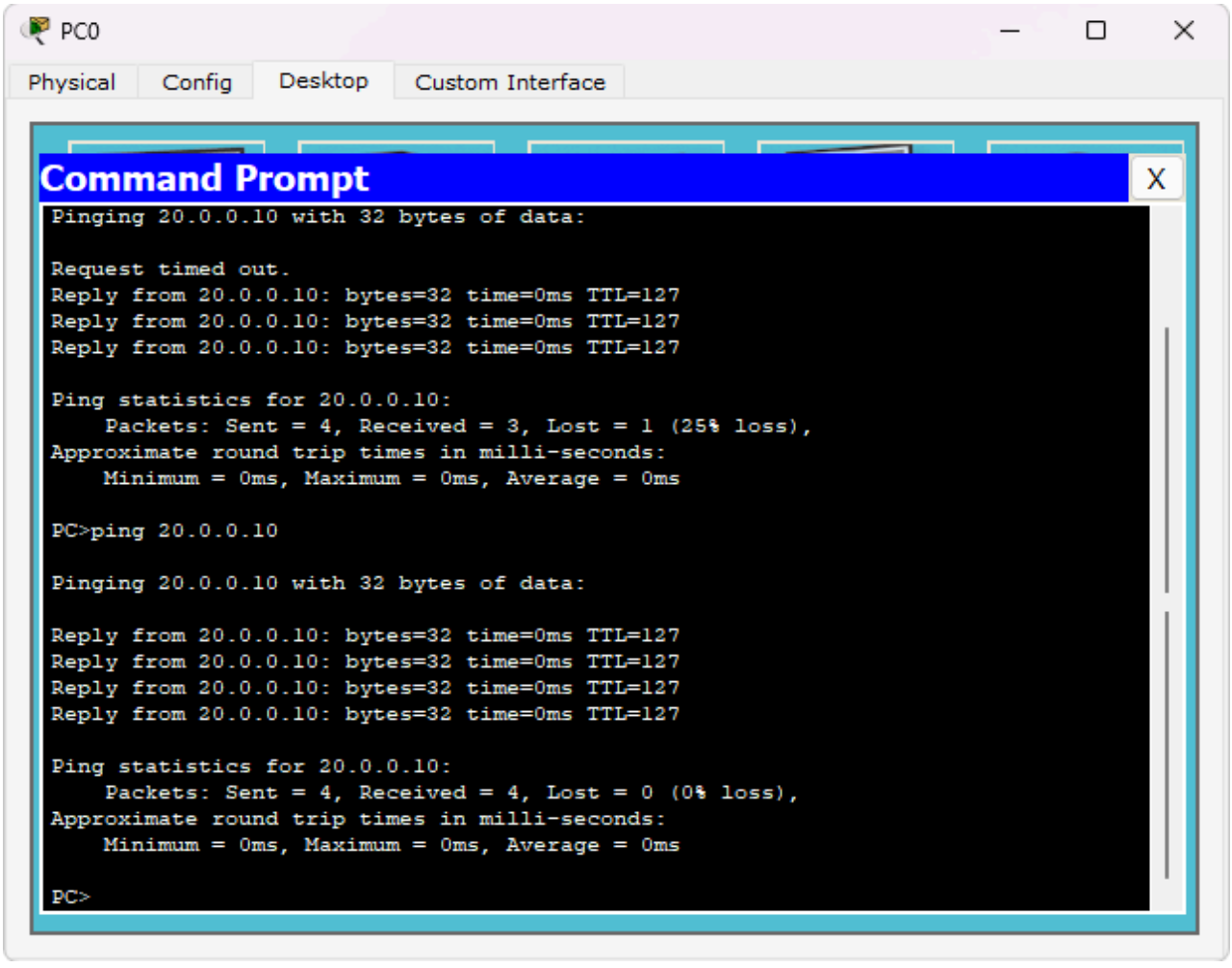
Hence, connection between router and PC is established

Observations:

During the simulation, the message is sent from PC1 to PC2 and acknowledgement is sent back.

Screen Shots:

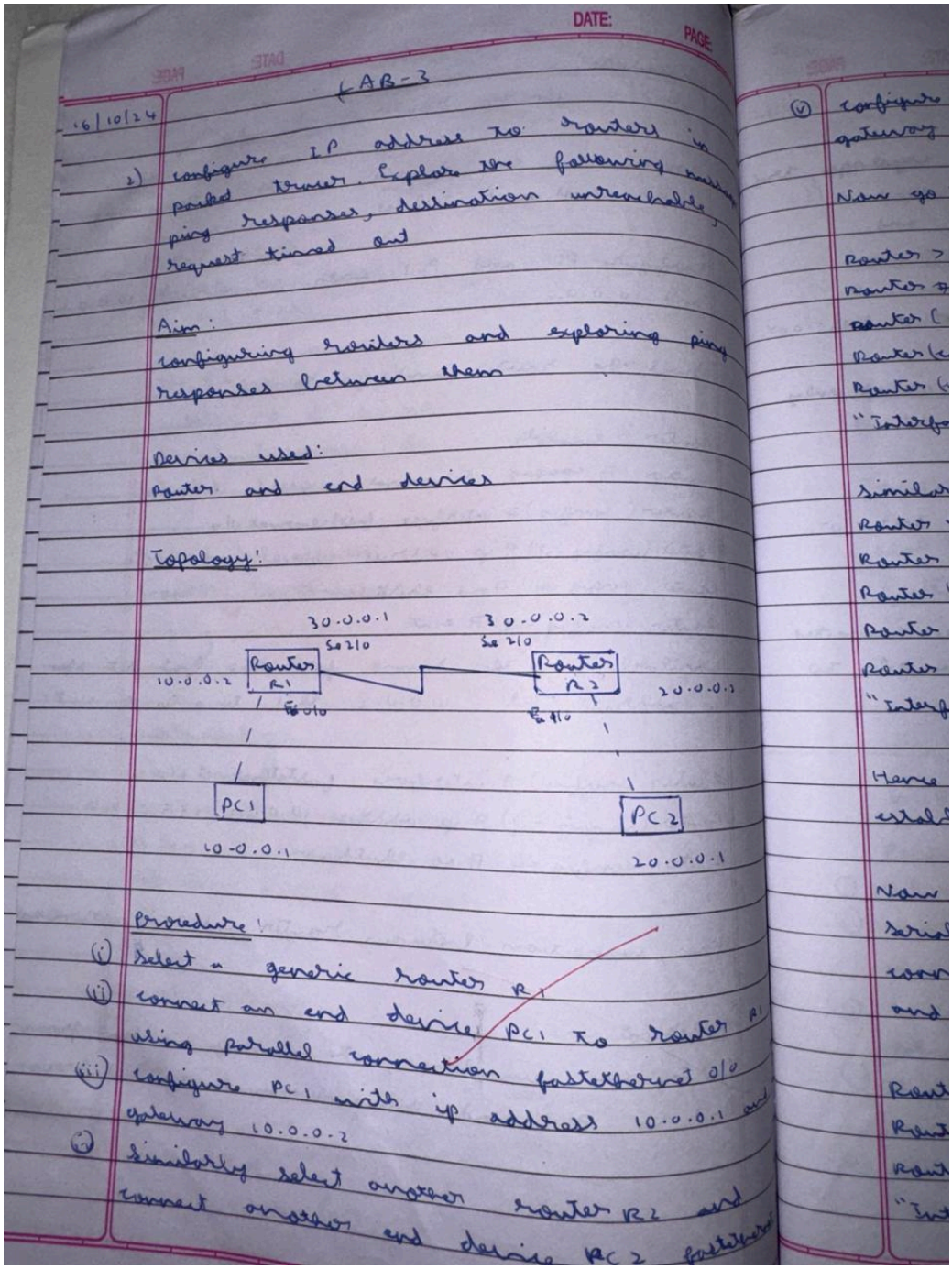


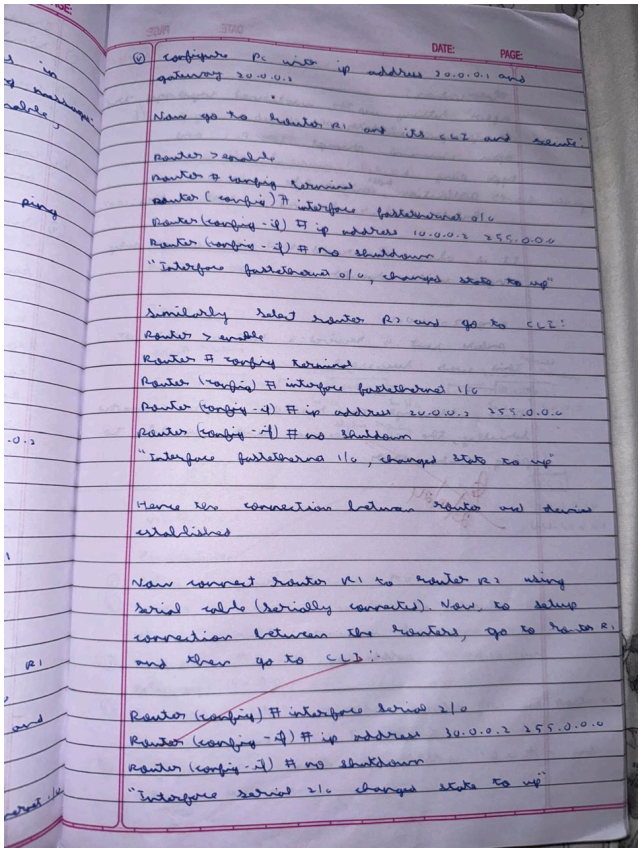


Program 3

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

Topology , Procedure and Observation:





v) Configure PC with ip address 10.0.0.1 and gateway 10.0.0.2

Now go to Router R1 and its CLI and execute:

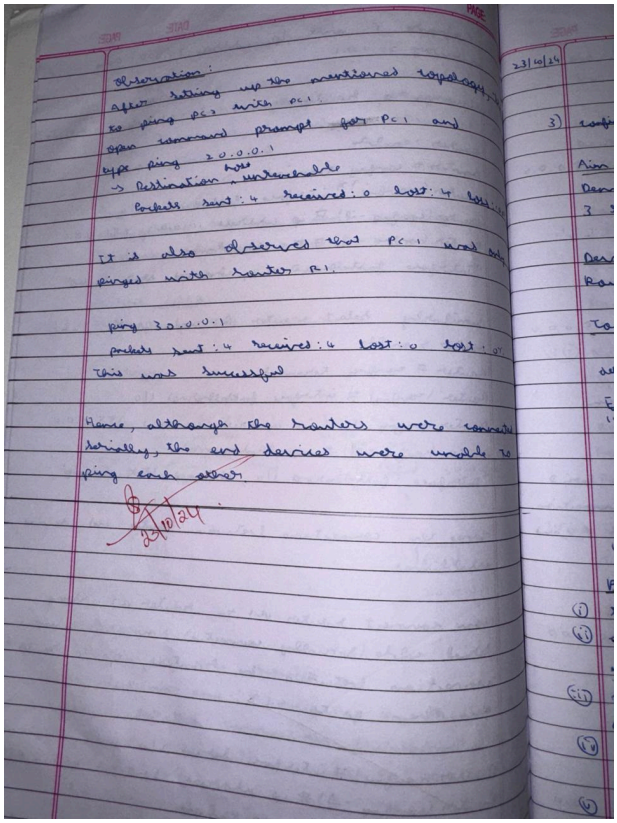
Router > enable
Router # config terminal
Router (config) # interface fastethernet 0/0
Router (config-if) # ip address 10.0.0.2 255.0.0.0
Router (config-if) # no shutdown
"Interface fastethernet 0/0, changed state to up"

Similarly select router R2 and go to CLI:
Router > enable
Router # config terminal
Router (config) # interface fastethernet 1/0
Router (config-if) # ip address 20.0.0.2 255.0.0.0
Router (config-if) # no shutdown
"Interface fastethernet 1/0, changed state to up"

Here the connection between router and device established

Now connect Router R1 to Router R2 using serial cable (serially connected). Now, to setup connection between the routers, go to router R1 and then go to CLI:

Router (config) # interface serial 2/0
Router (config-if) # ip address 10.0.0.2 255.0.0.0
Router (config-if) # no shutdown
"Interface serial 2/0 changed state to up"



Observation:
After setting up the mentioned topology, to ping PC2 with PC1
Open command prompt for PC1 and type ping 20.0.0.1
-> Destination host unreachable
Packets sent: 4 received: 0 lost: 4 loss: 100%

It is also observed that PC1 was not pinged with router R1.

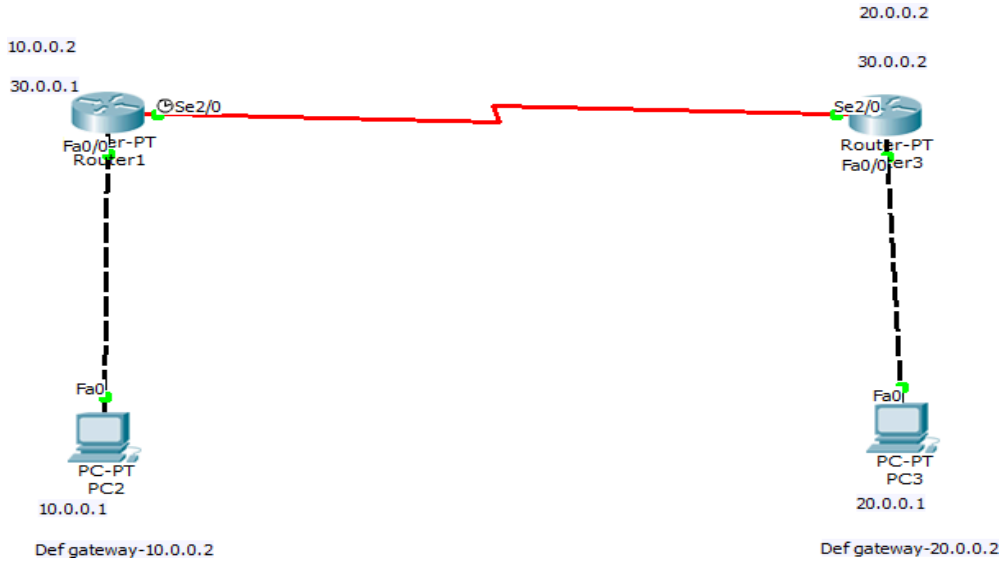
ping 30.0.0.1
packets sent: 4 received: 4 lost: 0 loss: 0%
This was successful

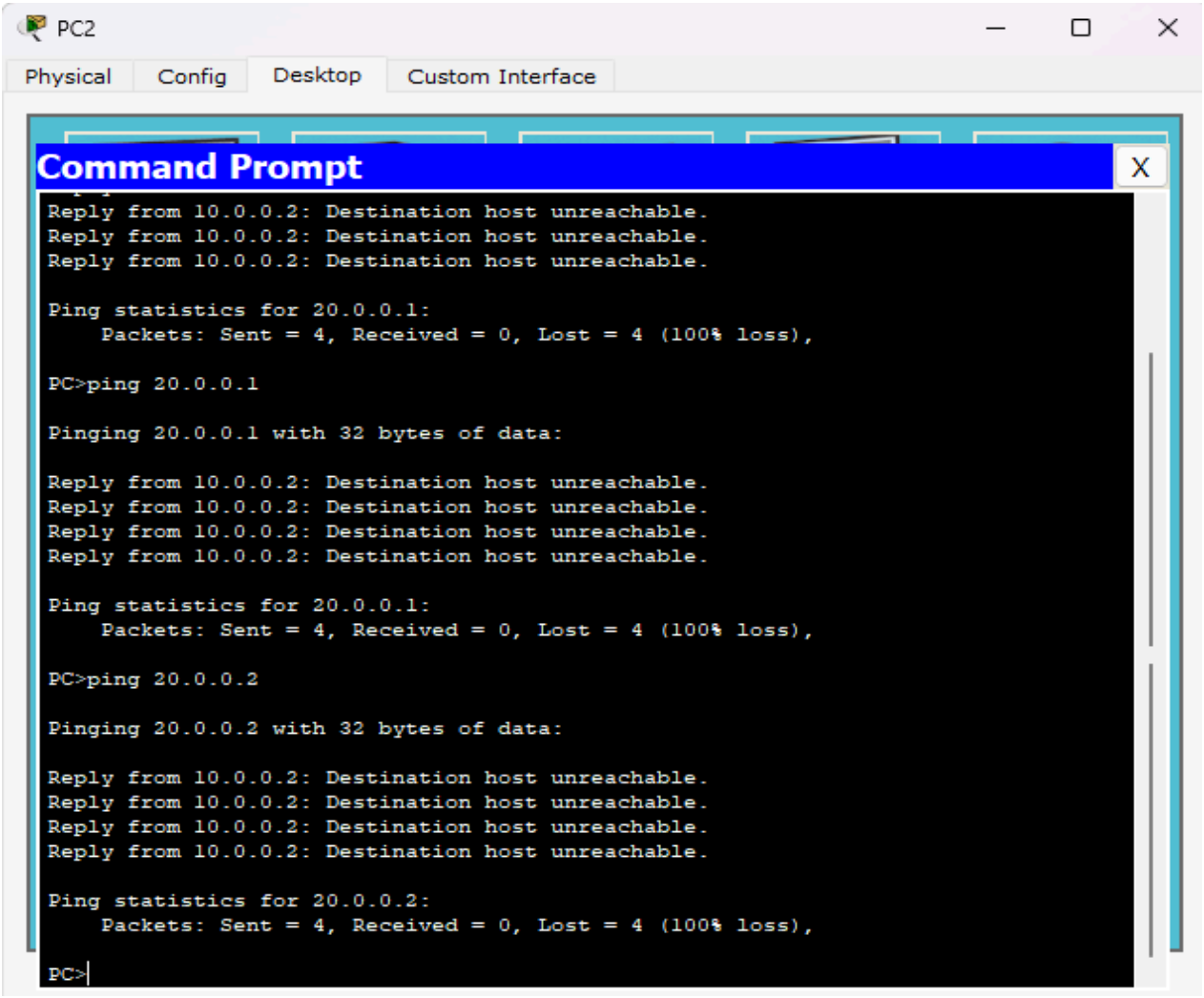
Hence, although the routers were connected serially, the end devices were unable to ping each other.

23/10/24

23/10/24

Screen Shots:

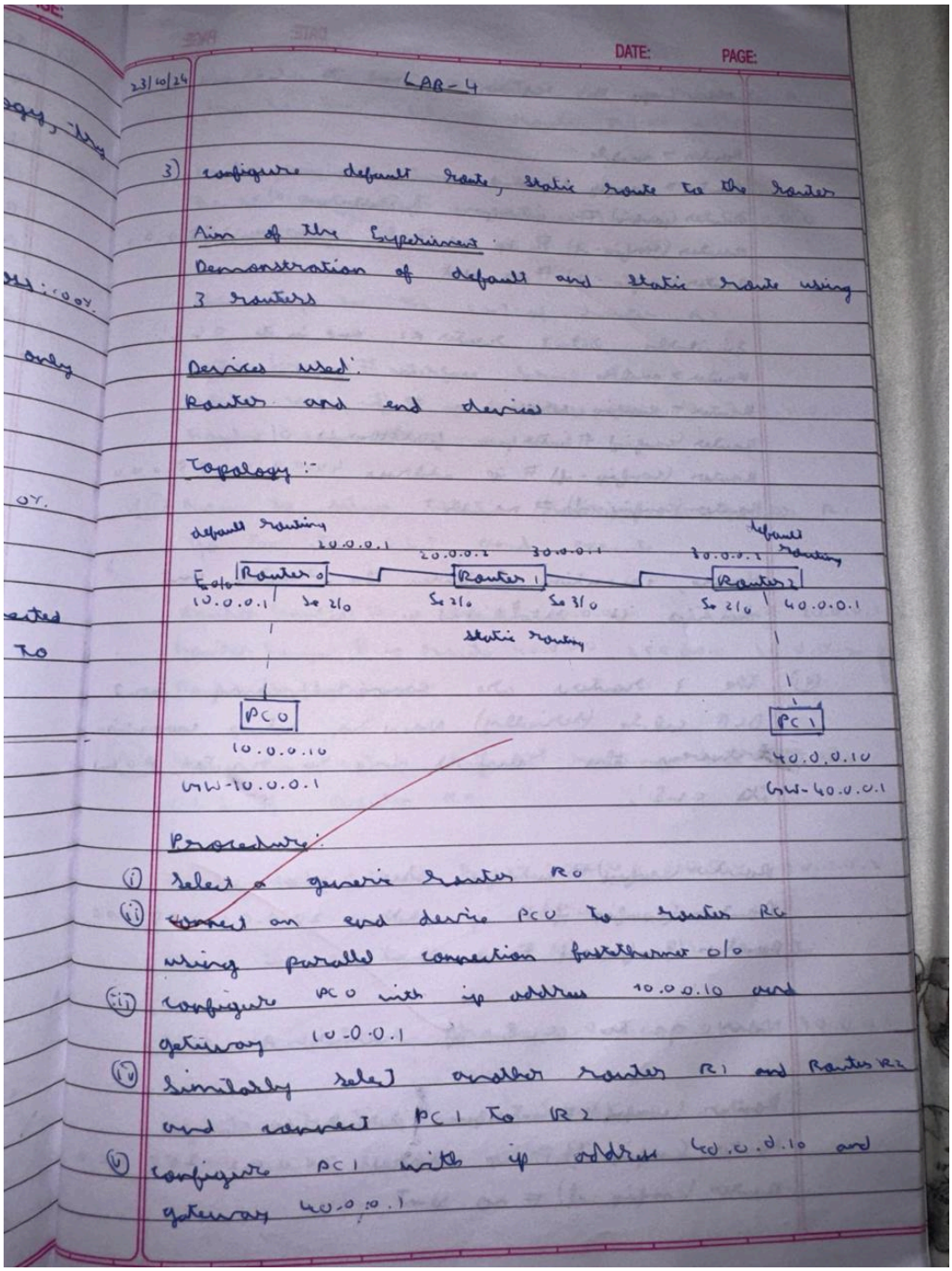




Program 4

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

Topology , Procedure and Observation:



DATE: PAGE:

Now go to router R0 and its CLI and

Router > enable
Router # config terminal
Router (config) # interface fastEthernet 0/0
Router (config-if) # ip address 10.0.0.10 255.0.0.
Router (config-if) # no shut

Similarly select router R2 and in its CLI
Router > enable
Router # config terminal
Router (config) # interface fastEthernet 0/0
Router (config-if) # ip address 40.0.0.10 255.0.0.
Router (config-if) # no shut

Hence, connection between the router and device is established.

(vi) The 2 routers are connected using serial DCE cable (serially). Now to setup connection between the routers. Go to router R0 in its CLI,

Router (config) # interface serial 2/0
Router (config-if) # ip address 20.0.0.1 255.0.0.
Router (config-if) # no shut

Now go to CLI of router R1

Router (config) # interface serial 2/0
Router (config-if) # ip address 20.0.0.2 255.0.0.
Router (config-if) # no shut

(vii) Now

(viii) Now

Now to connect Router R1 and Router R2:-
Go to the CLI of Router R1:-

Router (config) # interface serial 3/0
Router (config-if) # ip address 30.0.0.1 255.0.0.0
Router (config-if) # no shut

Now go to the CLI of Router R2:-

Router (config) # interface serial 2/0
Router (config-if) # ip address 40.0.0.1 255.0.0.0
Router (config-if) # no shut

(vii) Now to setup static routing in router R1 go to its CLI and execute:-

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

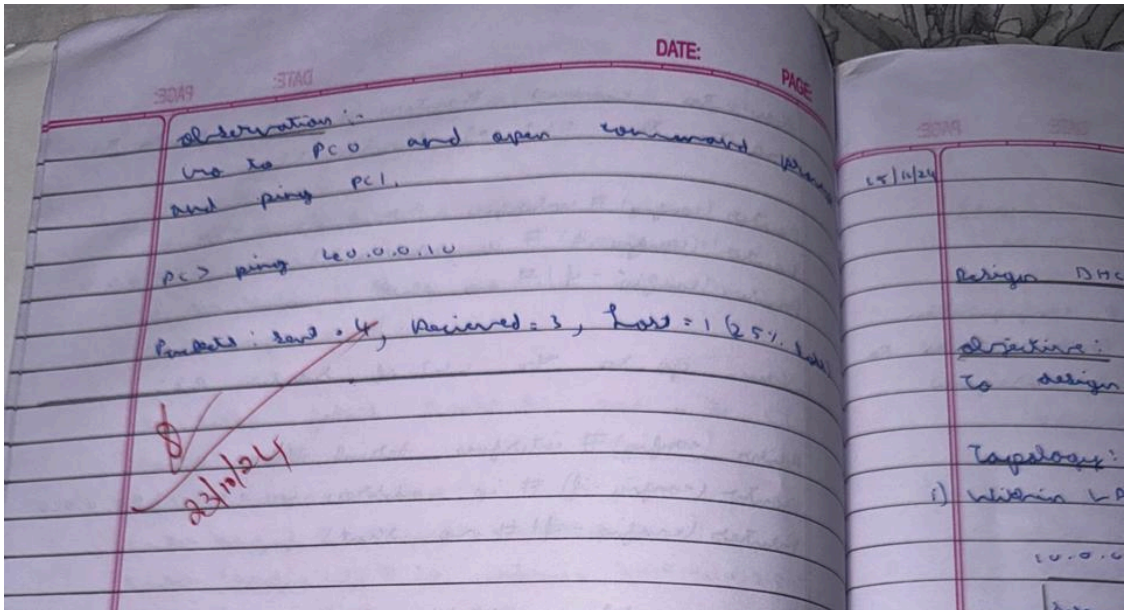
(viii) Now to setup default routing. Go to the CLI of Router R0

Router (config) # ip route 0.0.0.0 0.0.0.0 20.0.0.2

Now in the CLI of Router R2 execute:-

Router (config) # ip route 0.0.0.0 0.0.0.0 30.0.0.1

Default routing is now setup



DATE:

PAGE:

Observation:

Use to PC0 and open command prompt and ping PC1.

PC> ping 10.0.0.10

Packets: Sent = 4, Recieved = 3, Lost = 1 (25% loss)

23/10/24

15/11/24

Design DHC

Objective:

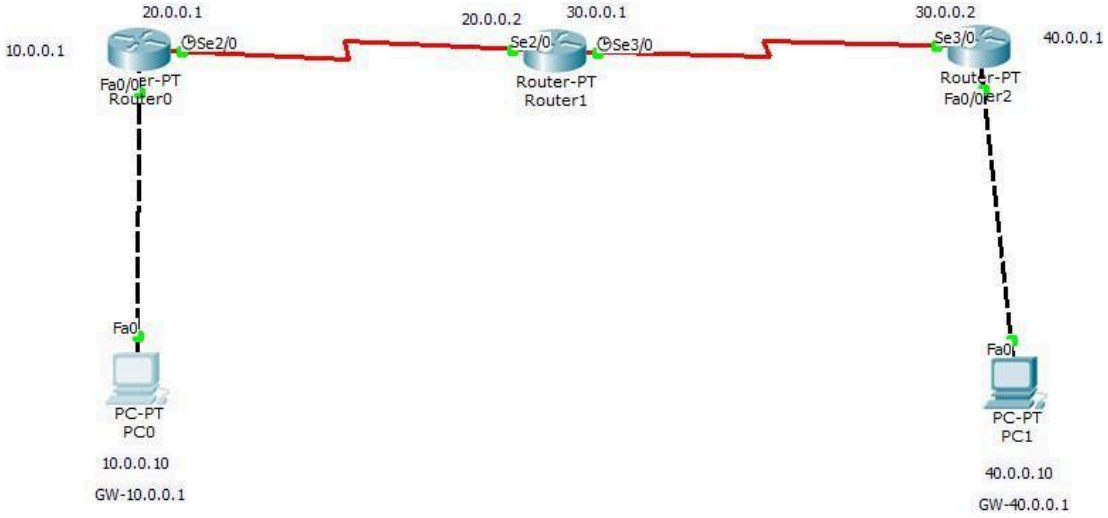
To design

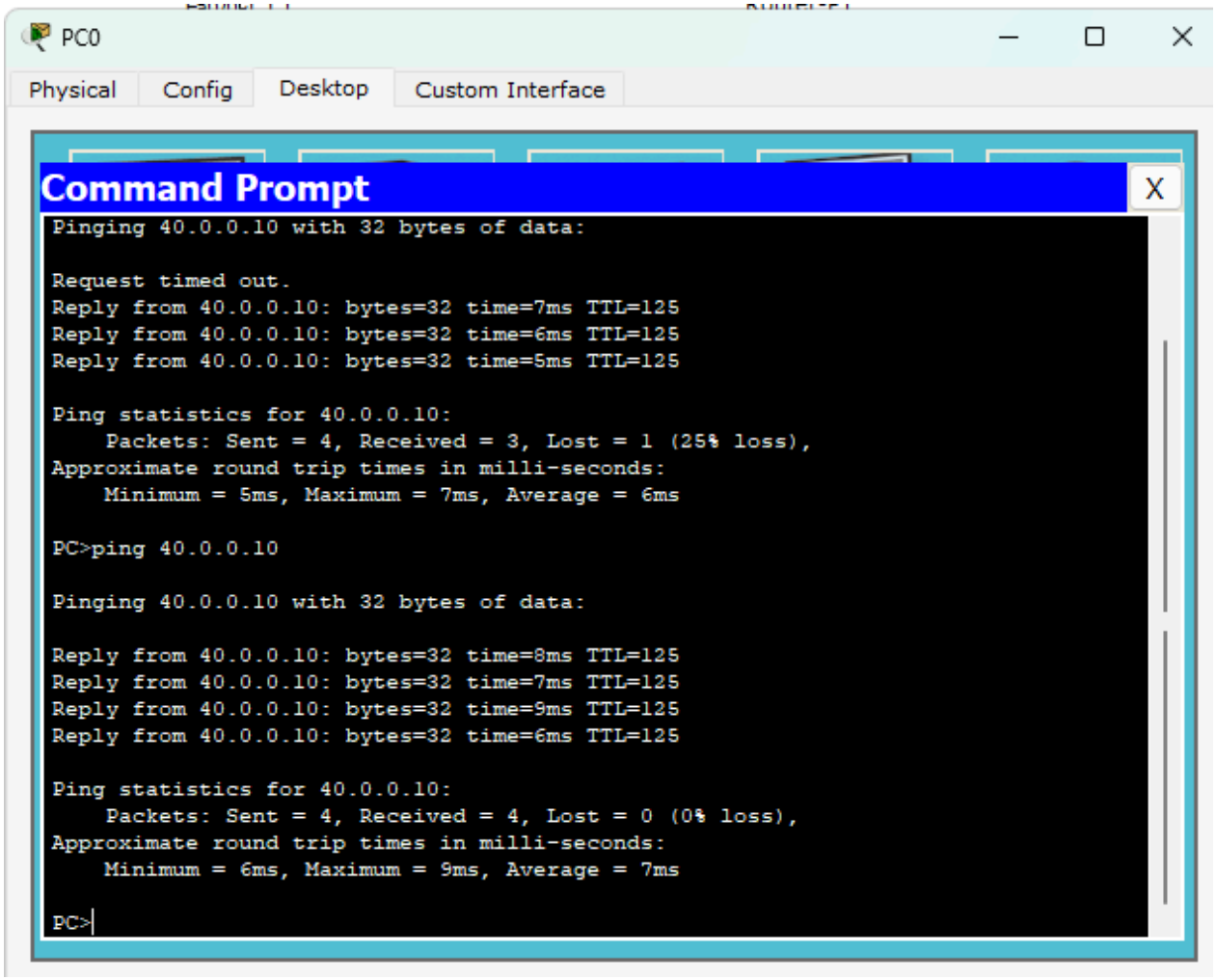
Topology:

i) Within LA

10.0.0

Screen Shots:

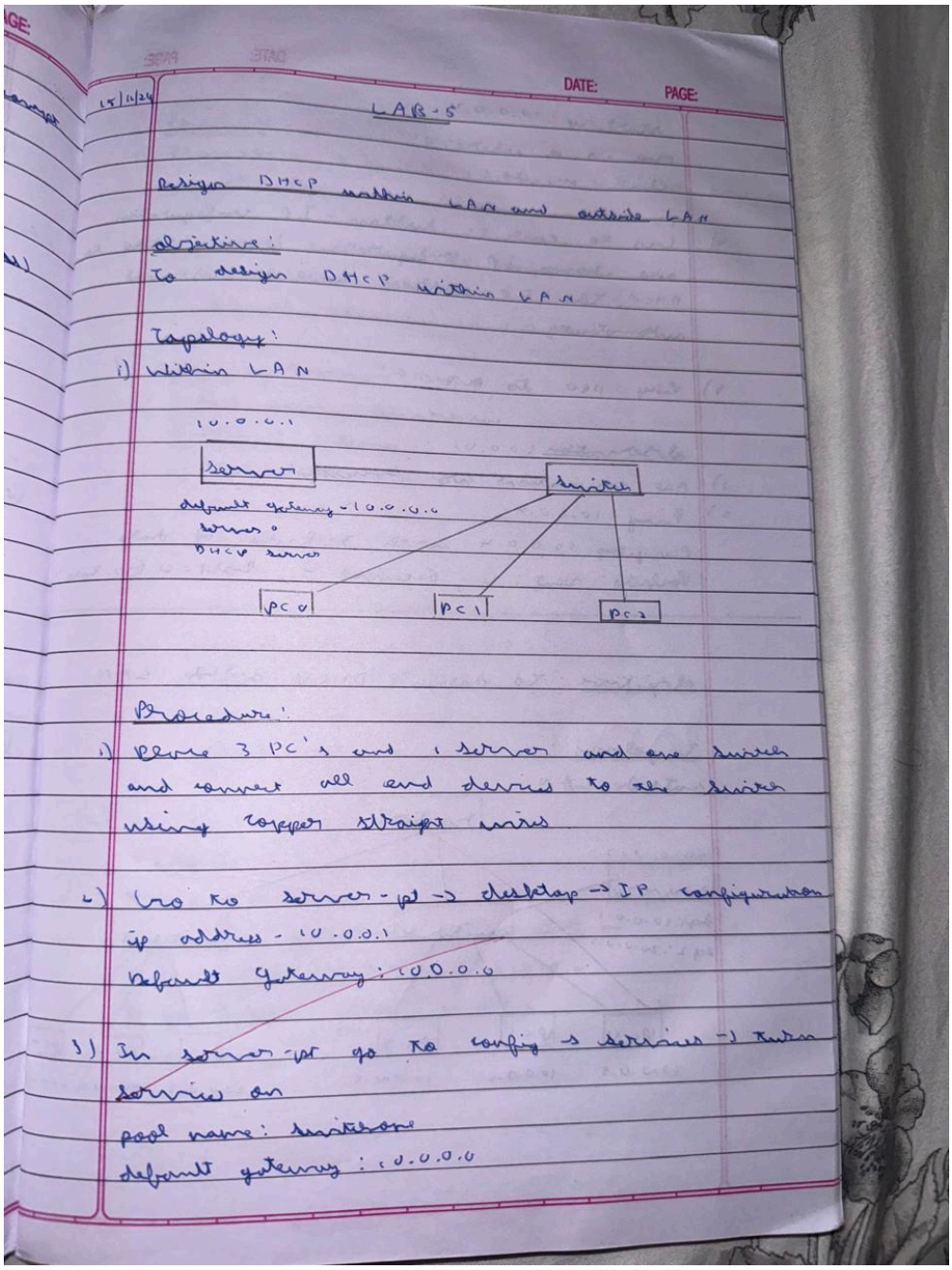




Program 5

Aim: Configure DHCP within a LAN and outside LAN.

Topology , Procedure and Observation:



DATE: PAGE:

start ip : 10.0.0.3
Max no. of users : 100
click on add

4) Go to each PC desktop - IP configuration and change IP configuration from static to DHCP. The IP address will be assigned automatically

5) Ping PC0 to PC2

observation :

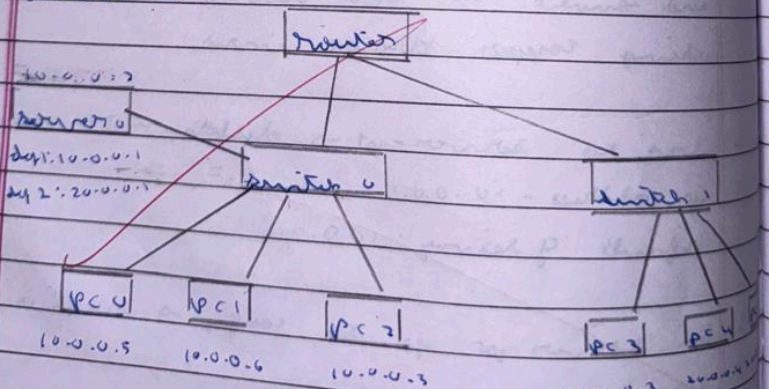
1) All connections are successful
2) Ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data
Packets sent : 4, Received 4, Lost = 0 (0% loss)

objective : To design DHCP outside LAN

Topology :

outside LAN



PAGE:

DATE: PAGE:

Procedure:

1) Place 6 PC's, 2 switch, 1 server, 1 router and connect them as shown in the figure.

2) Server → Desktop → IP configuration
IP address : 10.0.0.2
default gateway : 10.0.0.1

3) Config → Service → DHCP,
Pool name : switchone
default gateway : 10.0.0.1
start IP : 10.0.0.3
max users : 100
click add

pool name : switchtwo
default gateway : 20.0.0.1
start IP : 20.0.0.3
max users : 100
click add

4) Go to router CLI
> enable
config terminal
interface fastethernet 0/0
ip address 10.0.0.1 255.0.0.0
ip helper-address 10.0.0.2
no shut
exit

DATE: PAGE:

interface fastethernet 0/0
ip address 20.0.0.1 255.0.0.0
ip helper-address 10.0.0.2
no shut
exit

All router switches connections go up

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

6) Ping PC0 to PC5

Observation :

1) All connections are successful
2) All the PC's got assigned DHCP ip address

13/11/24.

20/11/24

configu

Aim :
To
router

Devices
Router

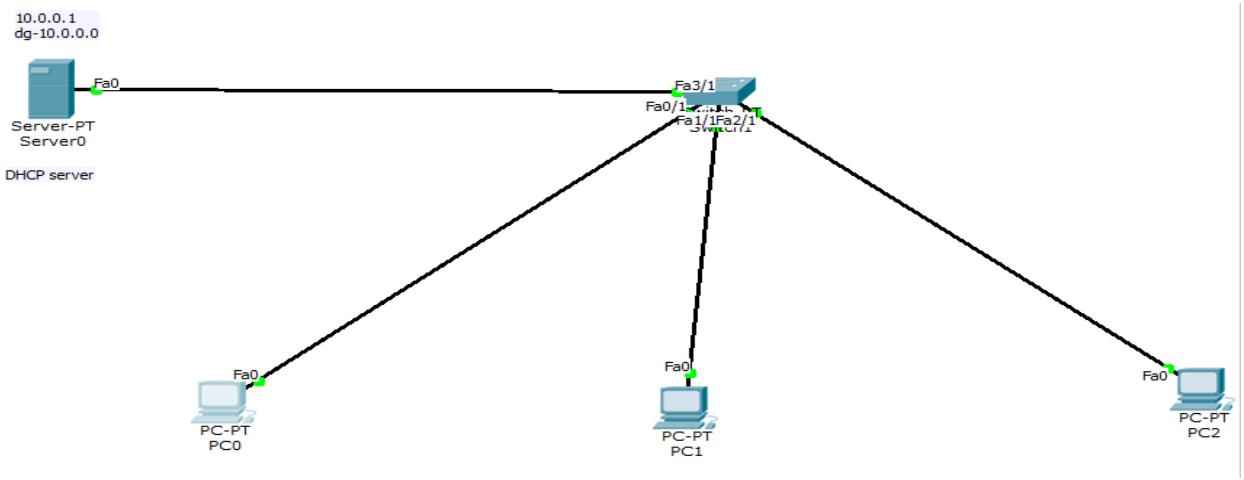
Topol

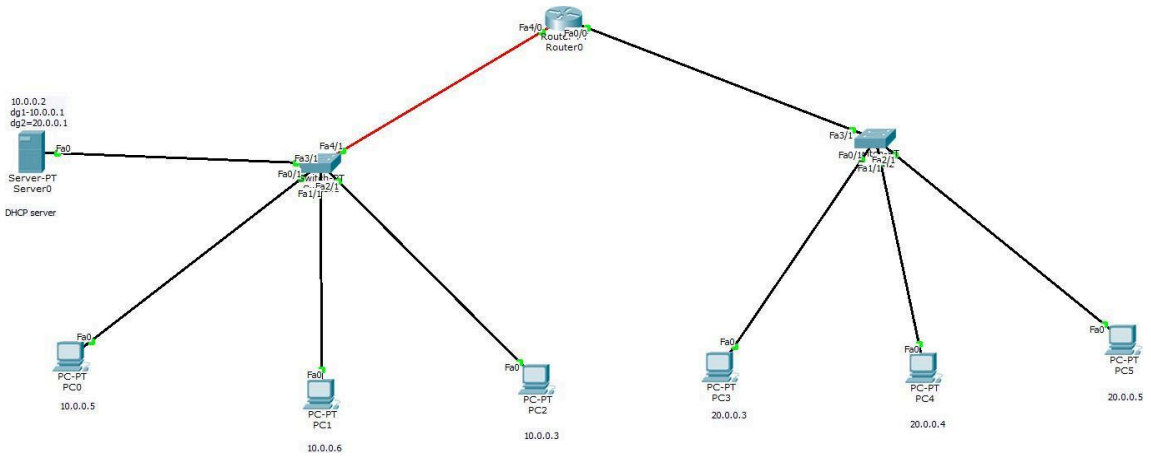
10.0

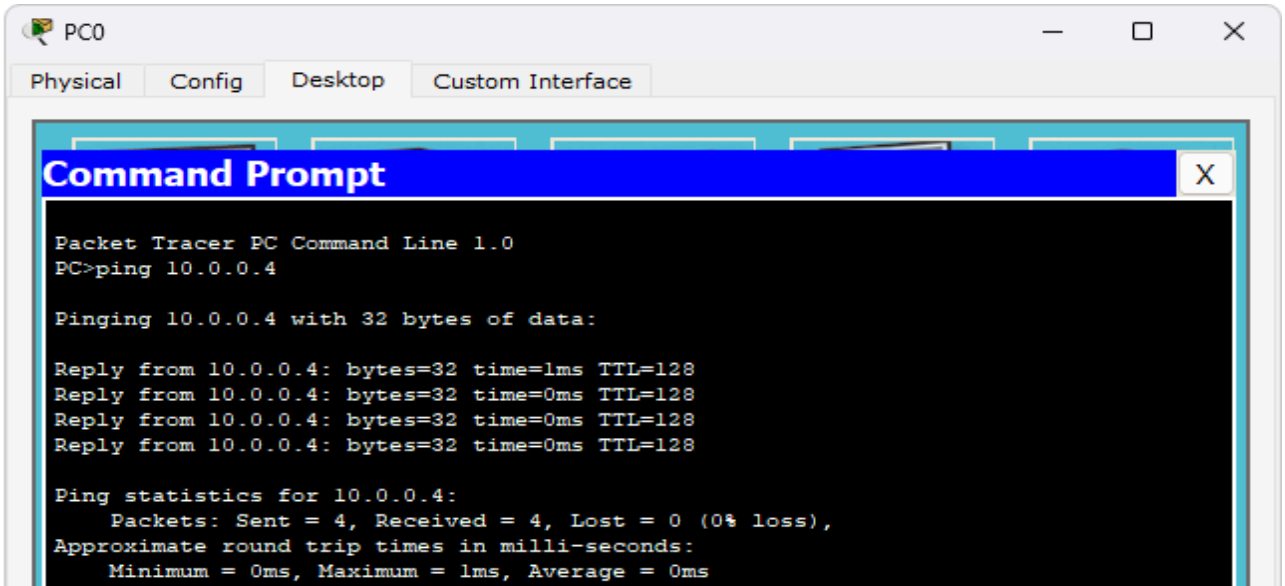
PC
10.0

Pro
(i) Con
a
(ii) Con

Screen Shots:



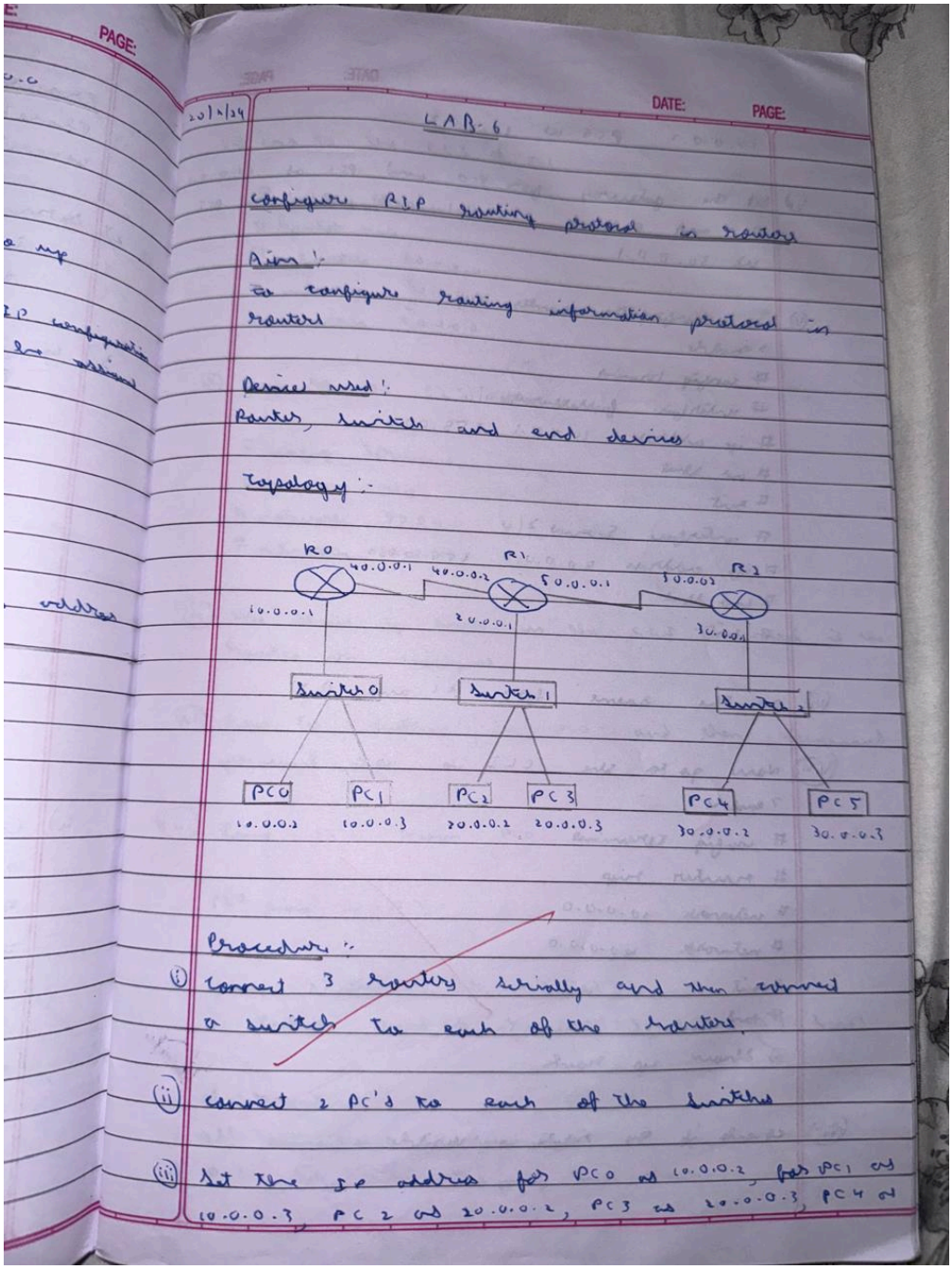


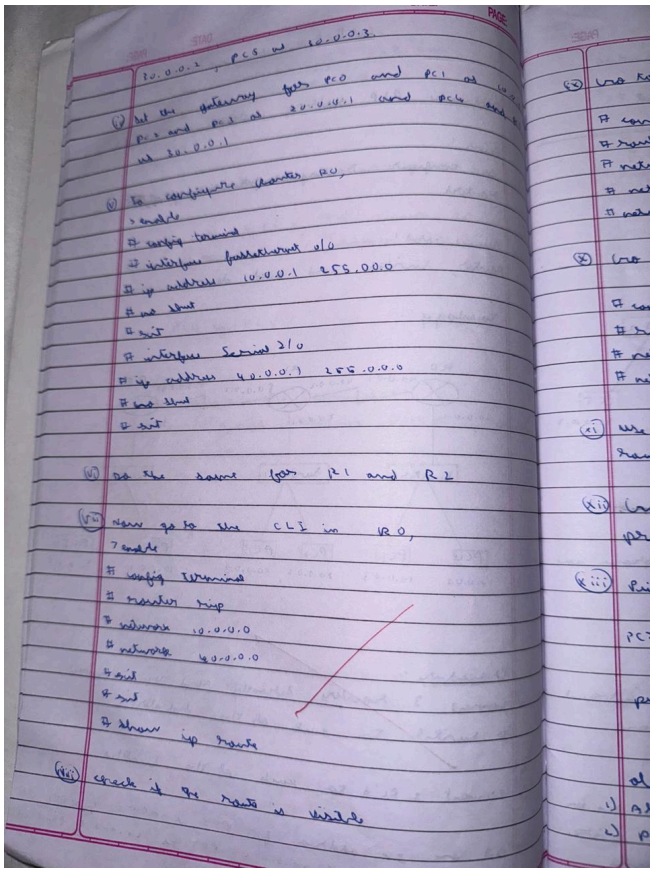


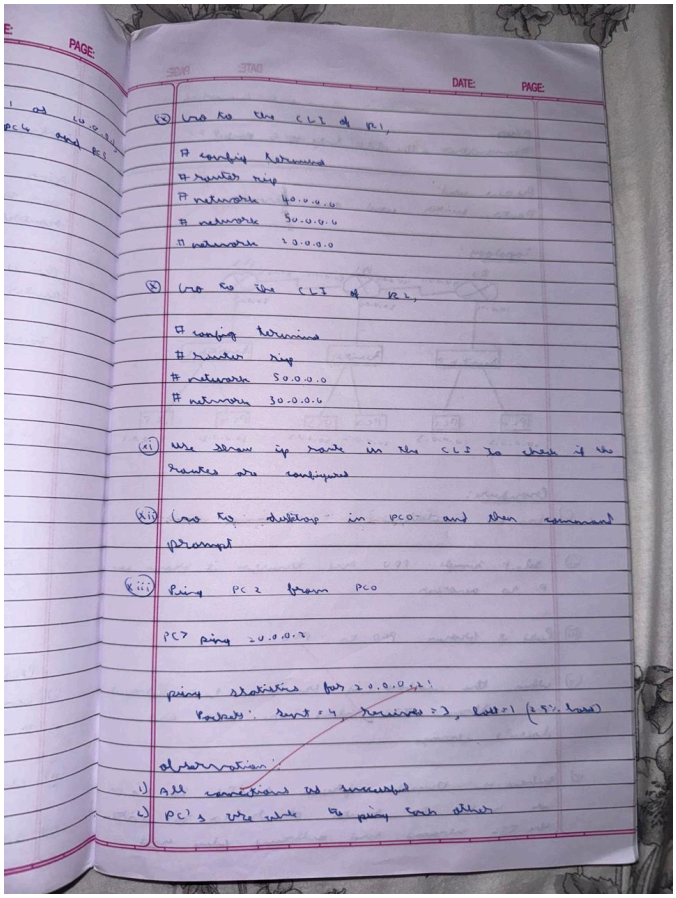
Program 6

Aim: Configure RIP routing Protocol in Routers .

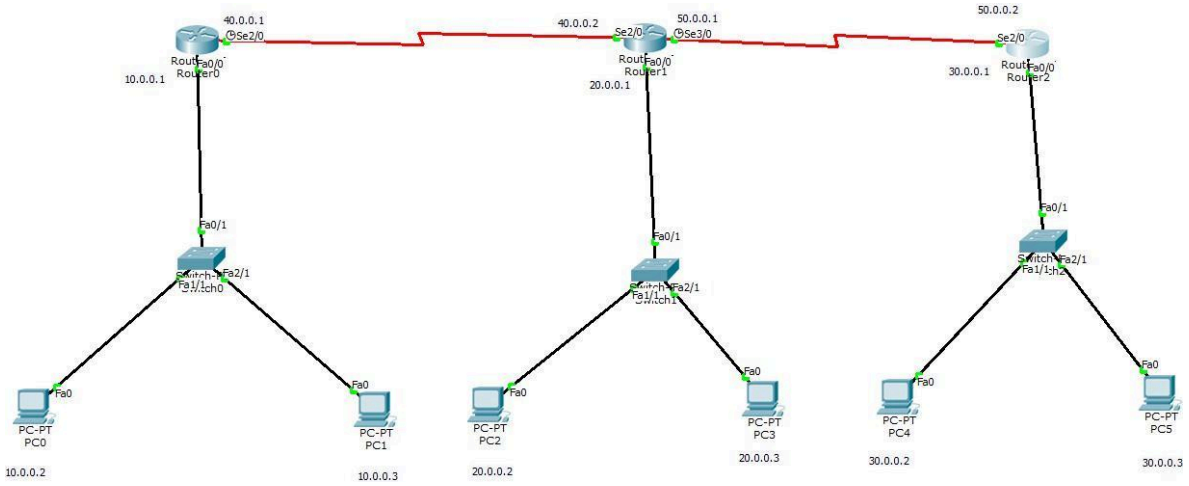
Topology , Procedure and Observation:

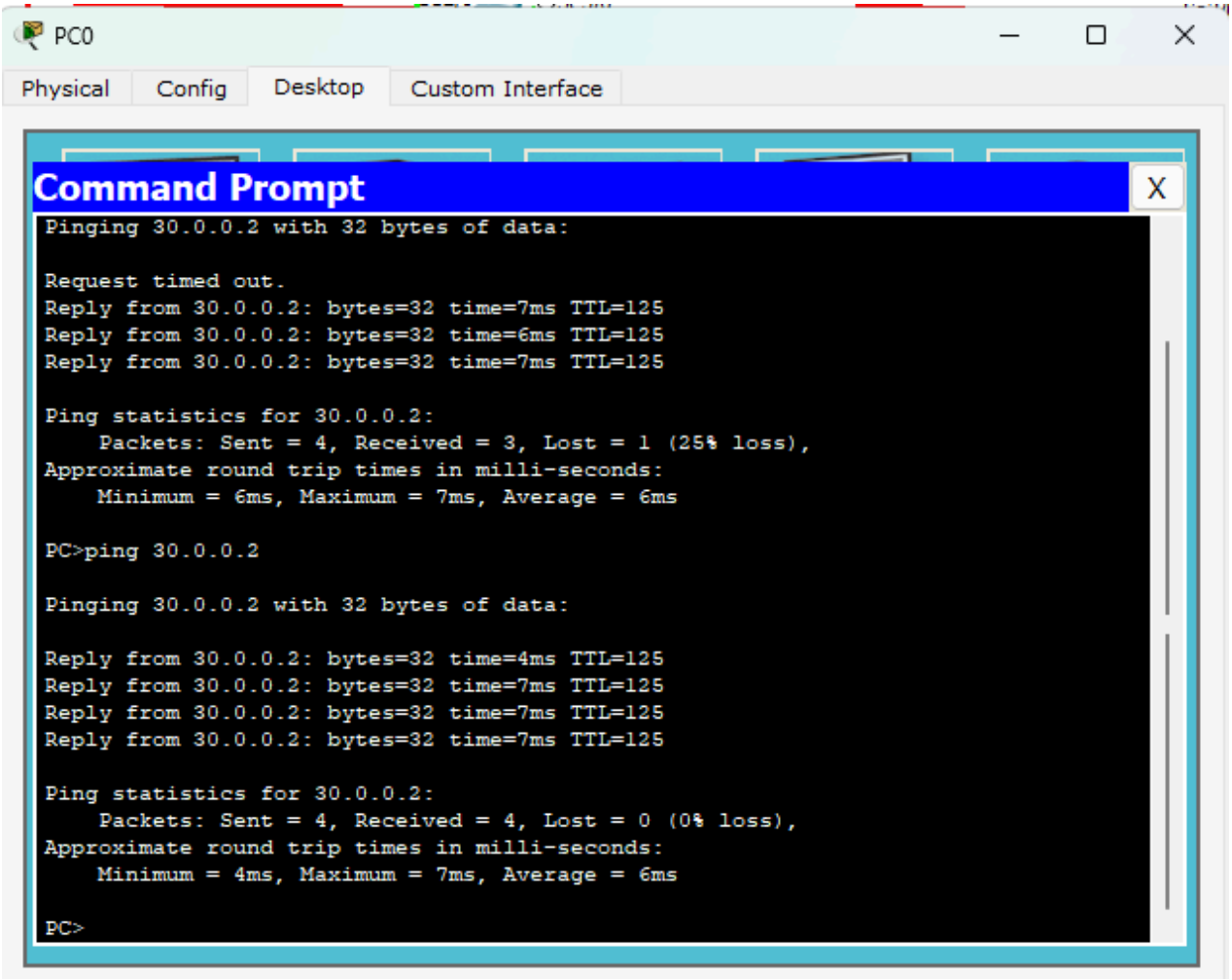






Screen Shots:

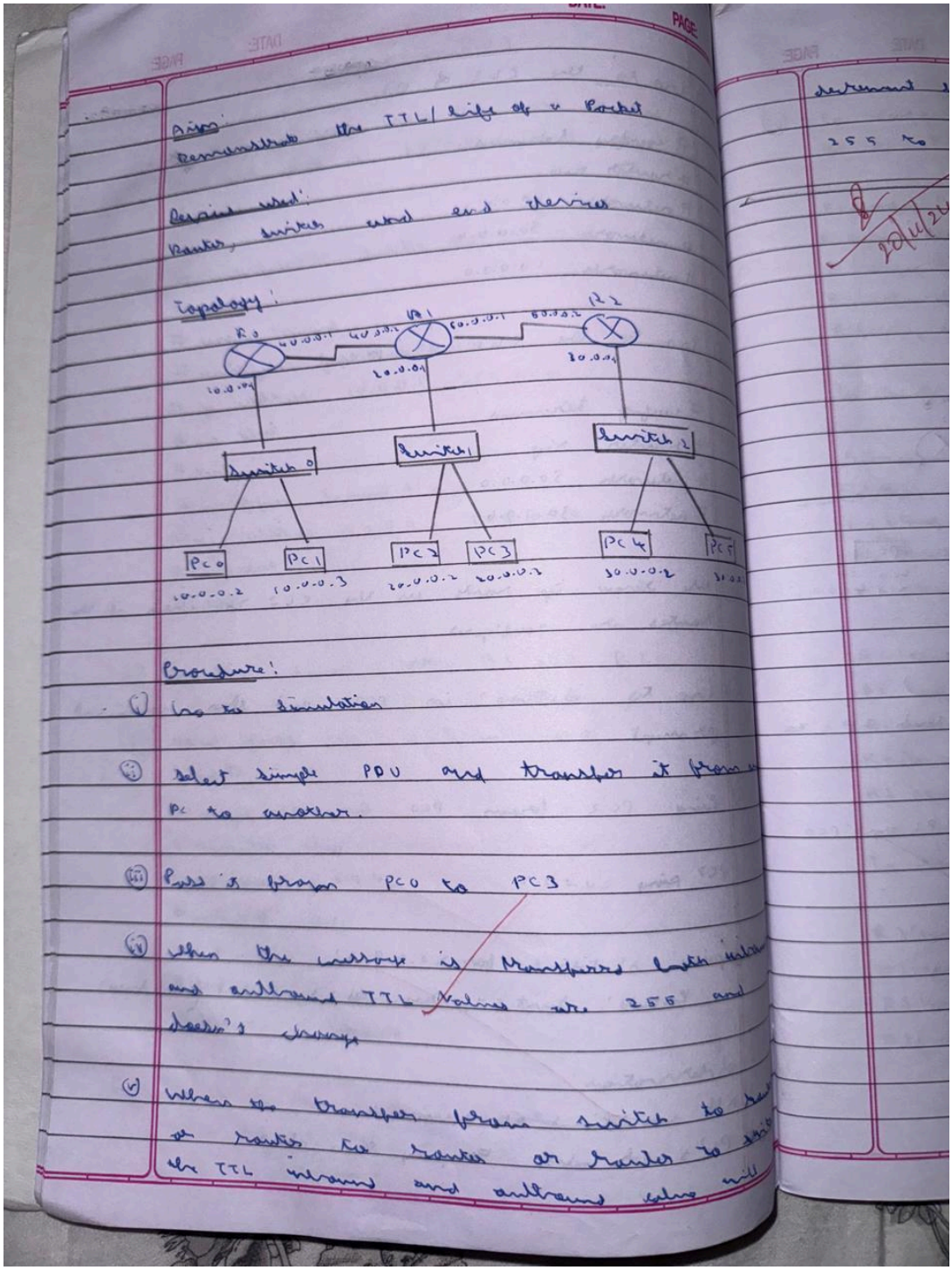


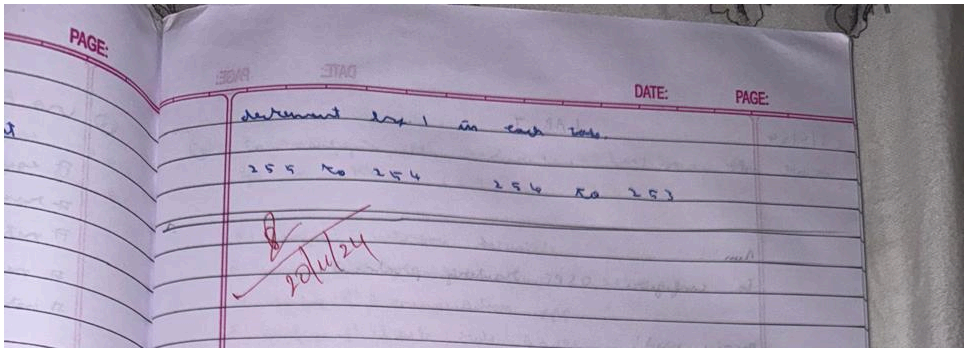


Program 7

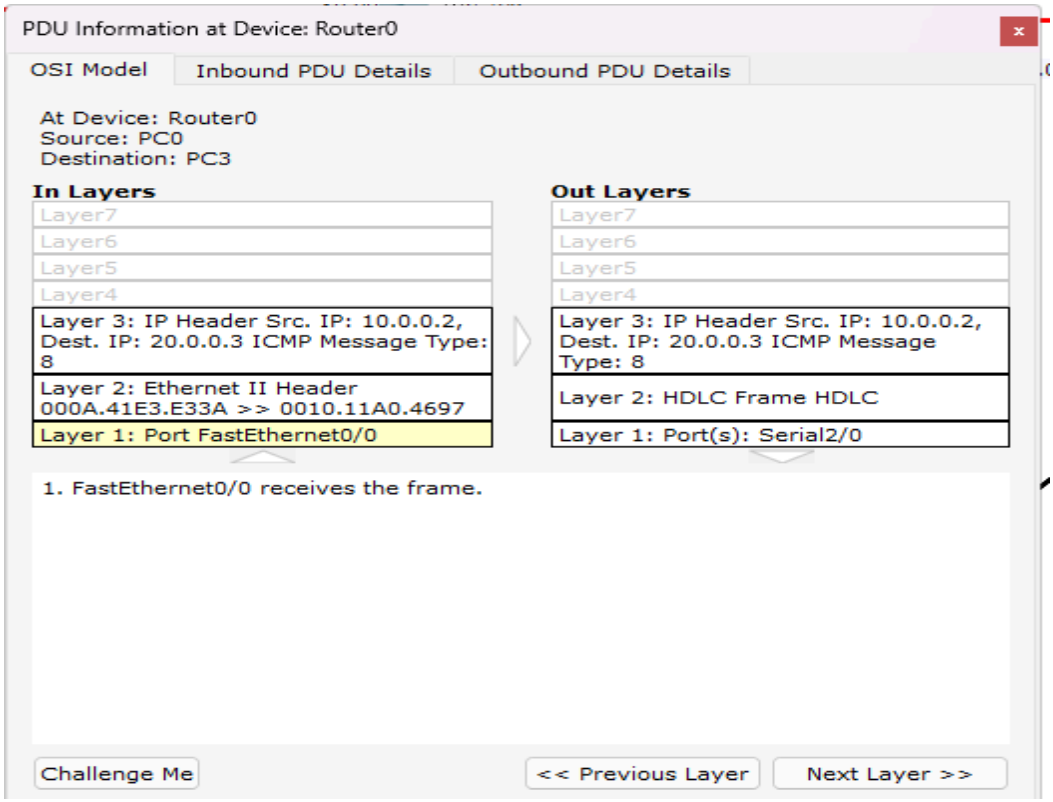
Aim: Demonstrate the TTL/ Life of a Packet .

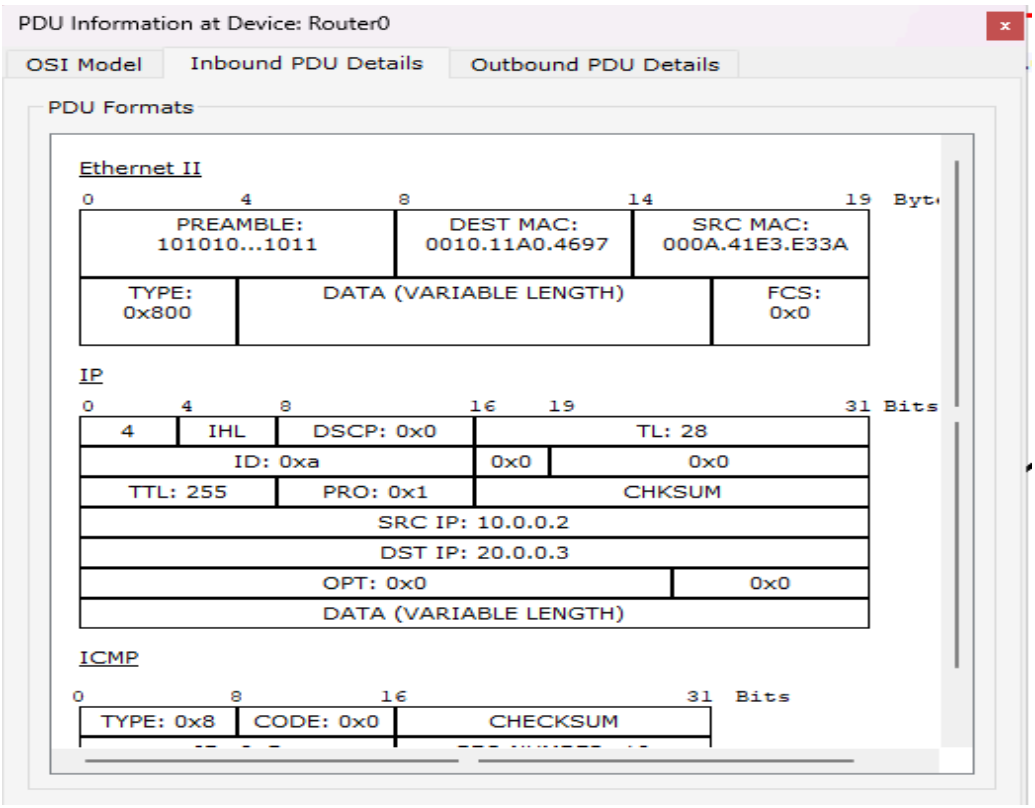
Topology , Procedure and Observation:

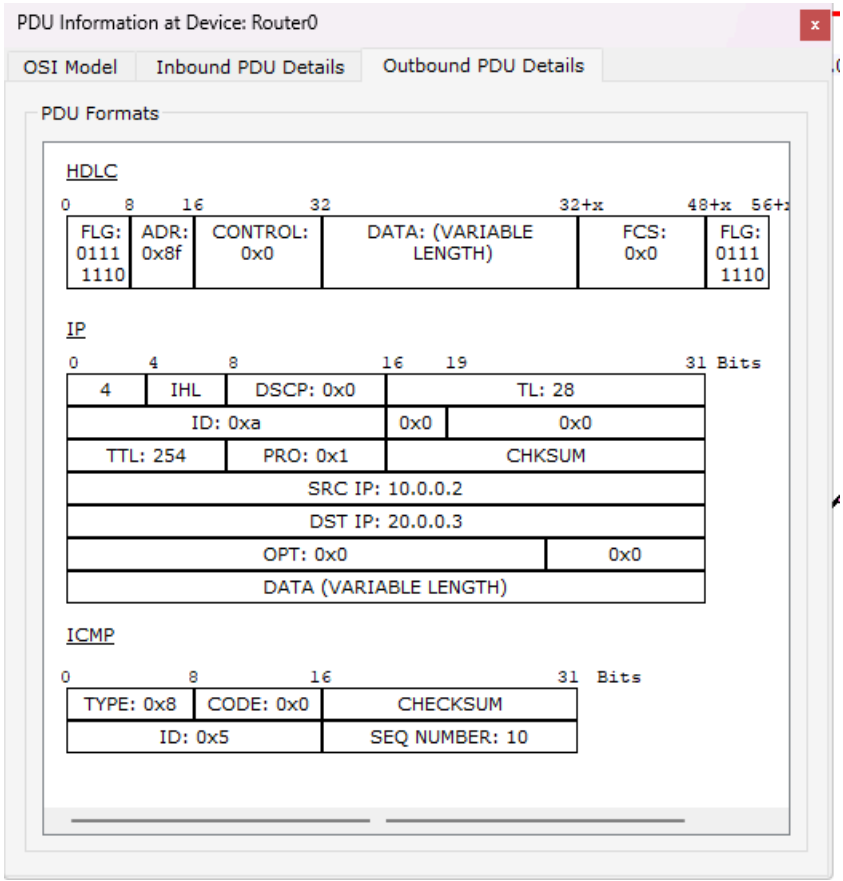




Screen Shots:



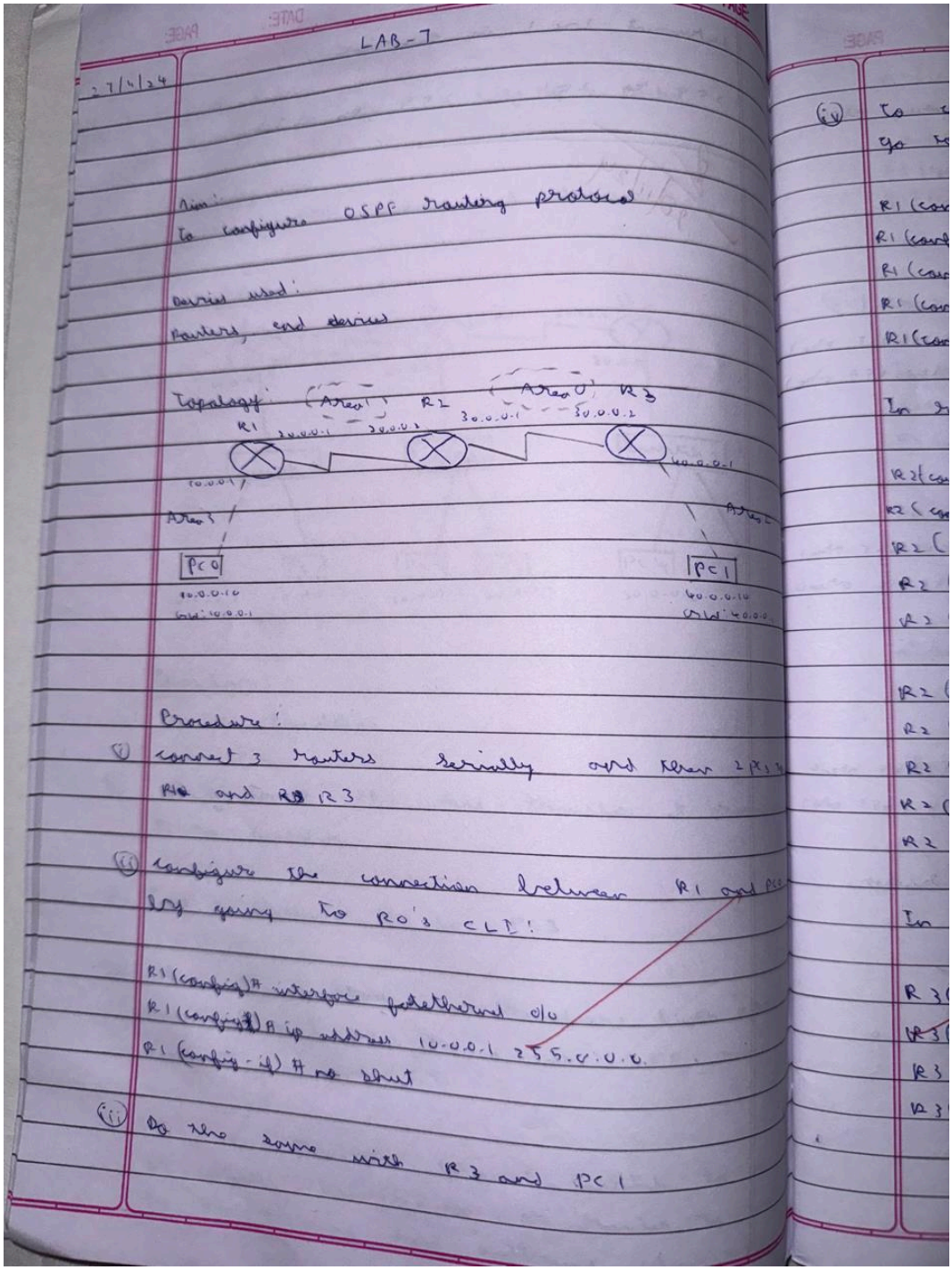

PDU Information at Device: Router0
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
0 4 8 14 19 Byt
PREAMBLE: 101010...1011
DEST MAC: 0010.11A0.4697
SRC MAC: 000A.41E3.E33A
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
0 4 8 16 19 31 Bits
4
IHL
DSCP: 0x0
TL: 28
ID: 0xa
0x0
0x0
TTL: 255
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


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+
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:

DATE: PAGE:

(iv) To configure the connections between the routers, go to R0's CLI

R1 (config) # interface serial 2/0
R1 (config-if) # ip address 20.0.0.1 255.0.0.0
R1 (config-if) # encapsulation ppp
R1 (config-if) clock rate 64000
R1 (config-if) # no shut

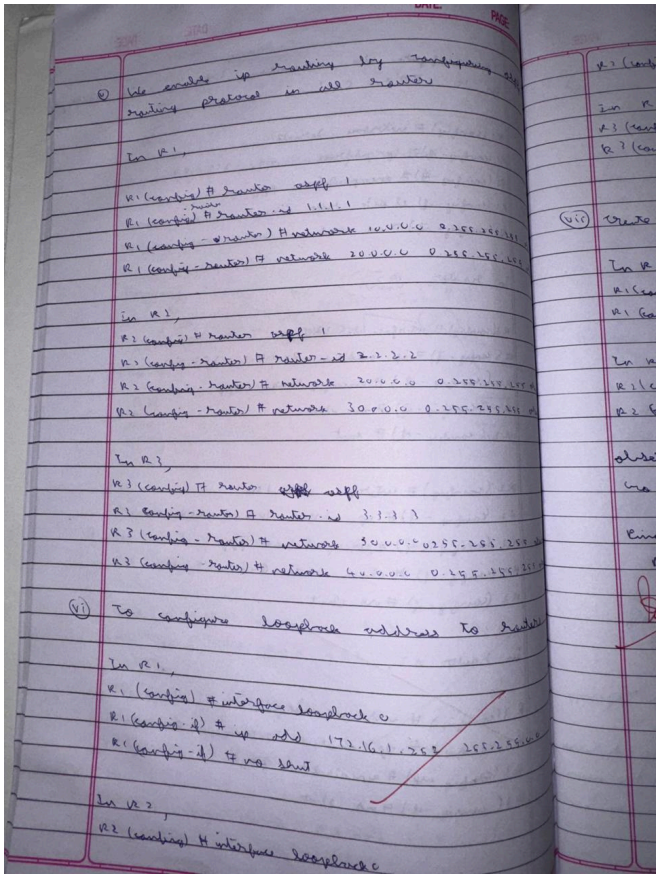
In router R2,

R2 (config) # interface serial 2/0
R2 (config-if) # ip address 20.0.0.2 255.0.0.0
R2 (config-if) # encapsulation ppp
R2 (config-if) # no shut
R2 (config-if) # exit

R2 (config) # interface Serial 3/0
R2 (config-if) # ip address 30.0.0.1 255.0.0.0
R2 (config-if) # encapsulation ppp
R2 (config-if) # clock rate 64000
R2 (config-if) # no shut

In router R3,

R3 (config) # interface Serial 2/0
R3 (config-if) # ip address 30.0.0.2 255.0.0.0
R3 (config-if) # encapsulation ppp
R3 (config-if) # no shut



(v) We enable ip routing by configuring ospf routing protocol on all routers

In R1,

R1 (config) # router ospf 1

R1 (config) # router-id 1.1.1.1

R1 (config-router) # network 10.0.0.0 0.255.255.255

R1 (config-router) # network 20.0.0.0 0.255.255.255

In R2,

R2 (config) # router ospf 1

R2 (config-router) # router-id 2.2.2.2

R2 (config-router) # network 20.0.0.0 0.255.255.255

R2 (config-router) # network 30.0.0.0 0.255.255.255

In R3,

R3 (config) # router ospf 1

R3 (config-router) # router-id 3.3.3.3

R3 (config-router) # network 30.0.0.0 0.255.255.255

R3 (config-router) # network 40.0.0.0 0.255.255.255

(vi) To configure loopback address to routers

In R1,

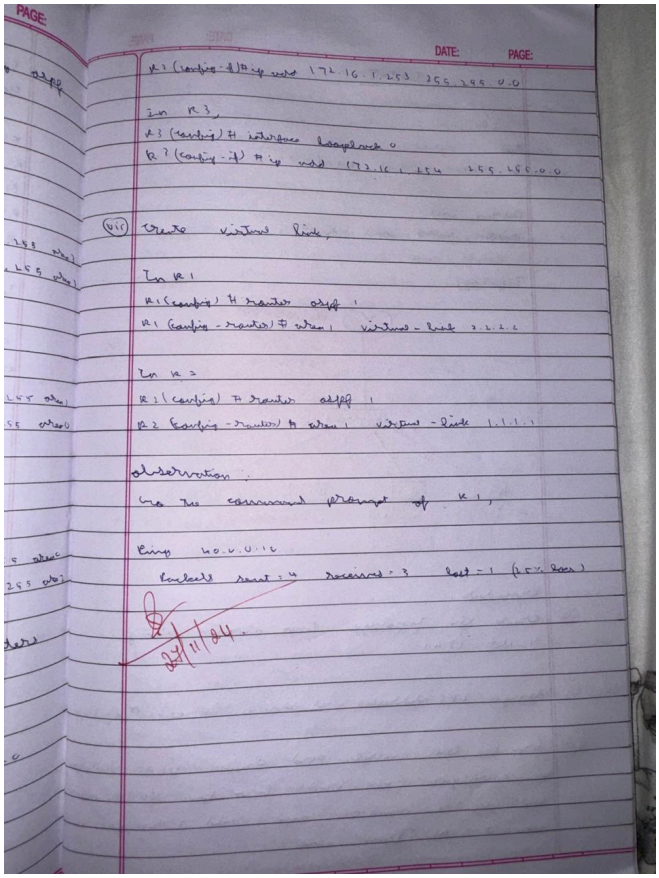
R1 (config) # interface loopback 0

R1 (config-if) # ip address 172.16.1.254 255.255.255.0

R1 (config-if) # no shut

In R2,

R2 (config) # interface loopback 0



PAGE:

DATE: PAGE:

R3 (config-if)# ip add 172.16.1.253 255.255.0.0

In R3,

R3 (config)# interface loopback 0

R3 (config-if)# ip add 172.16.254.254 255.255.0.0

(viii) Create virtual link,

In R1

R1 (config)# router ospf 1

R1 (config-router)# area 1 virtual-link 3.3.3.3

In R3

R3 (config)# router ospf 1

R3 (config-router)# area 1 virtual-link 1.1.1.1

Observation:

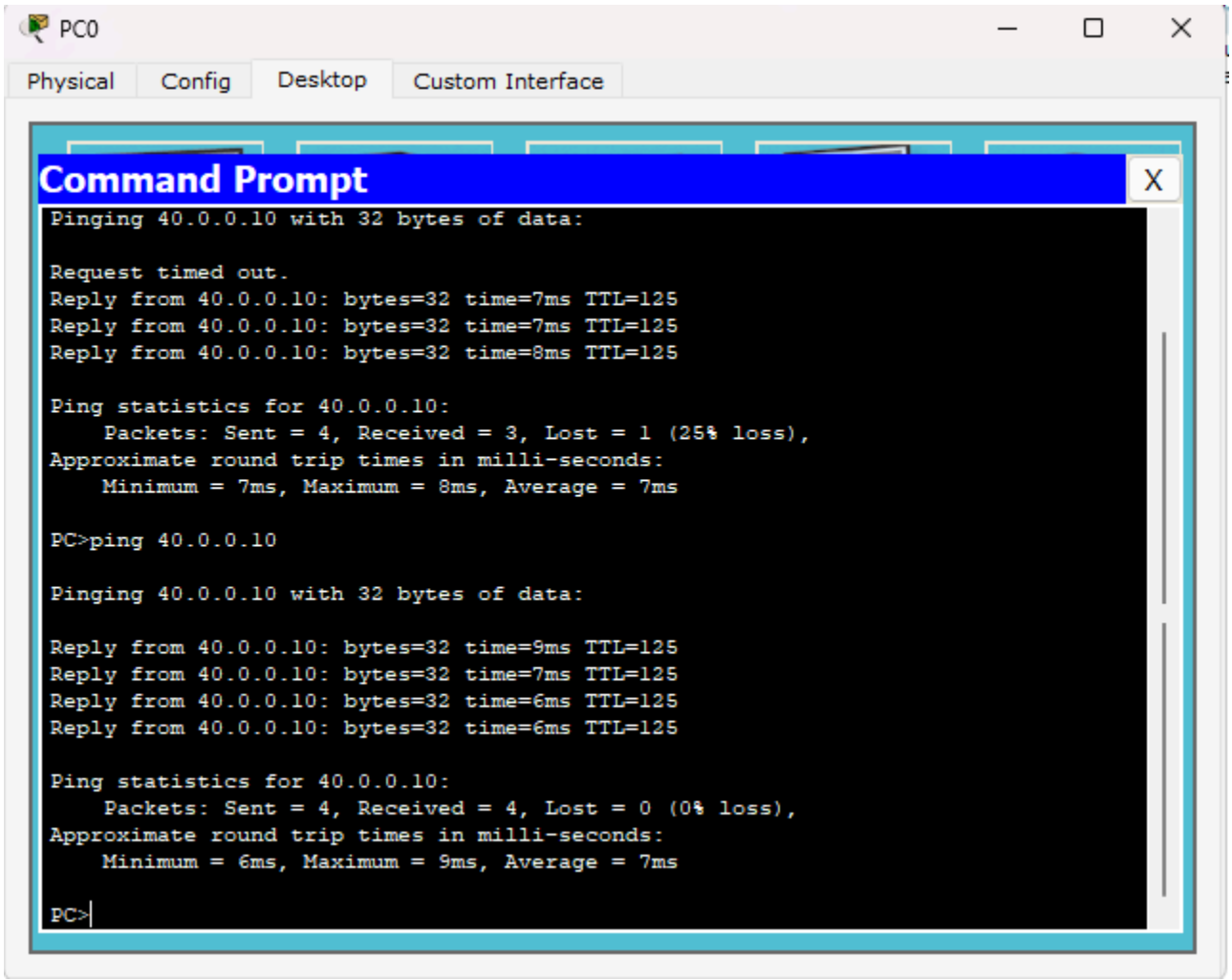
In the command prompt of R1,

Ping 40.0.0.10

Packets sent = 4 received = 3 lost = 1 (25% loss)

27/11/24

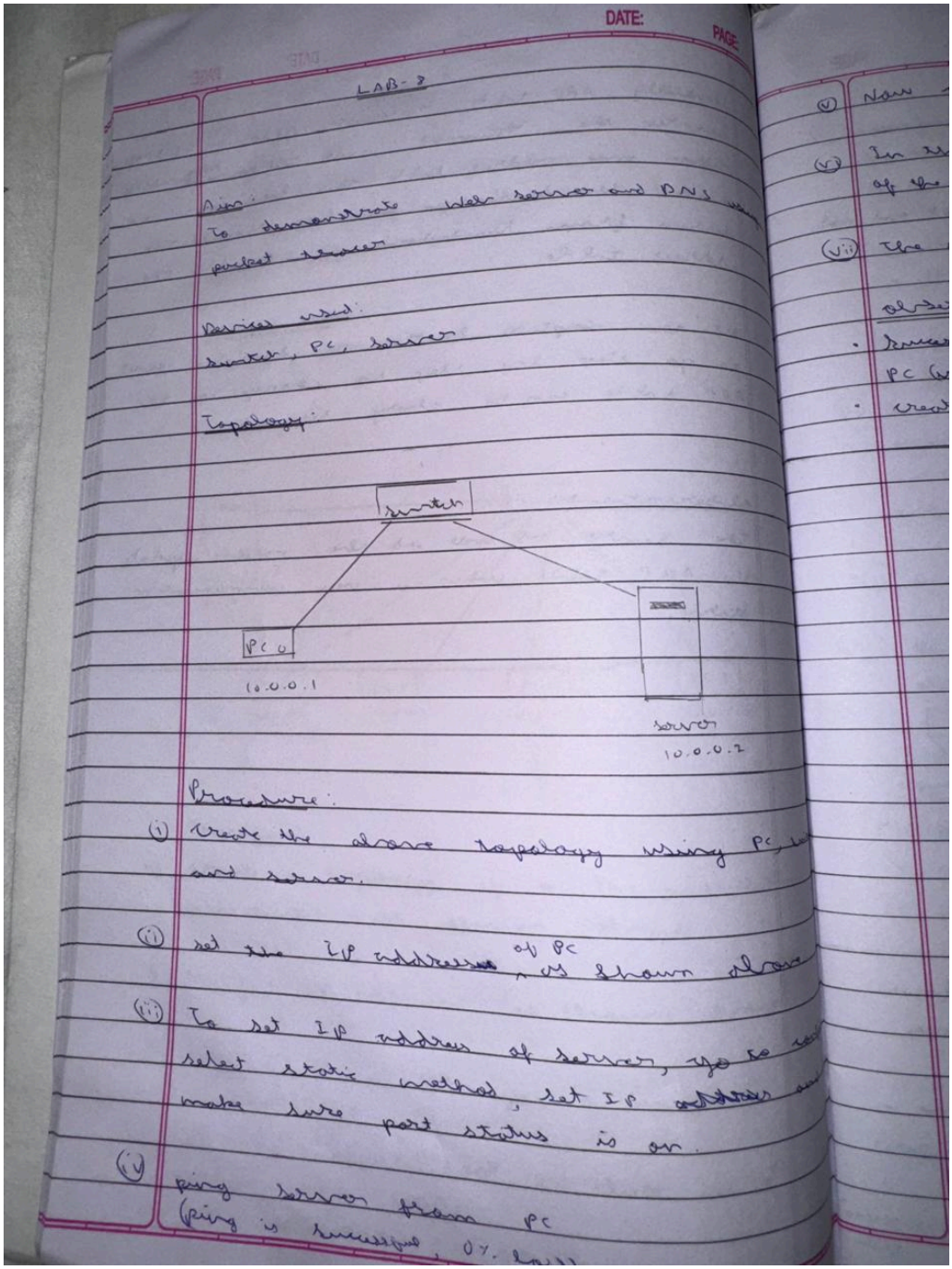
Screen Shots:

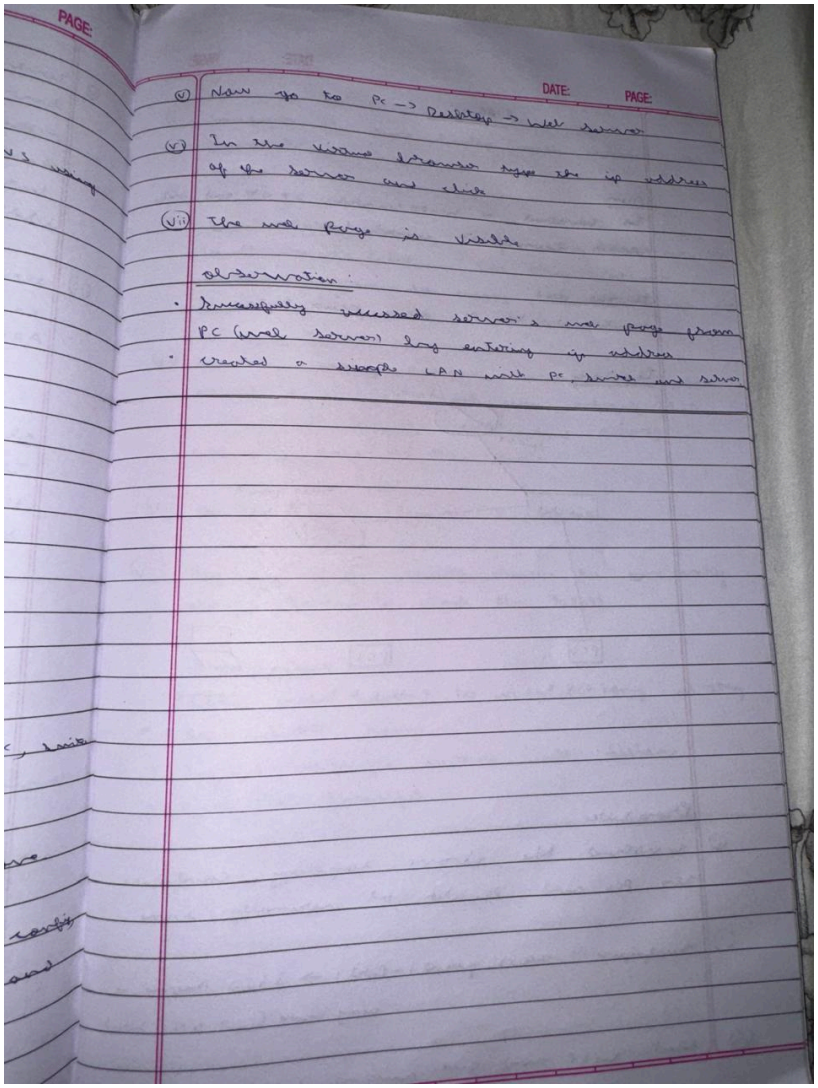


Program 9

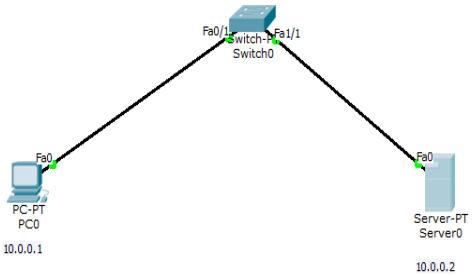
Aim: Configure Web Server, DNS within a LAN.

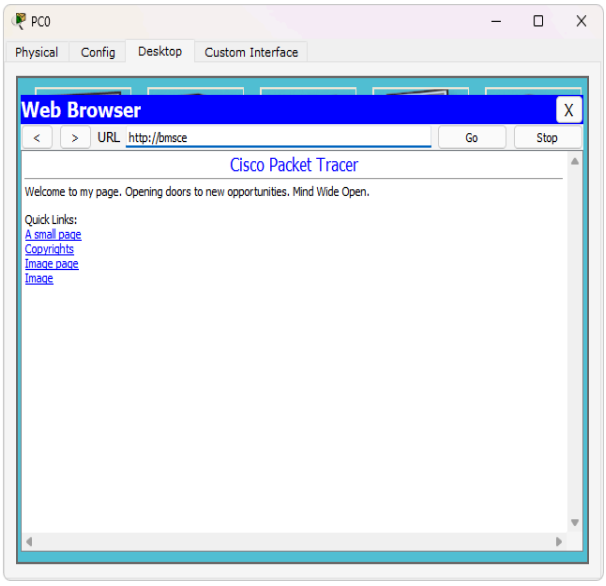
Topology , Procedure and Observation:





Screen Shots:

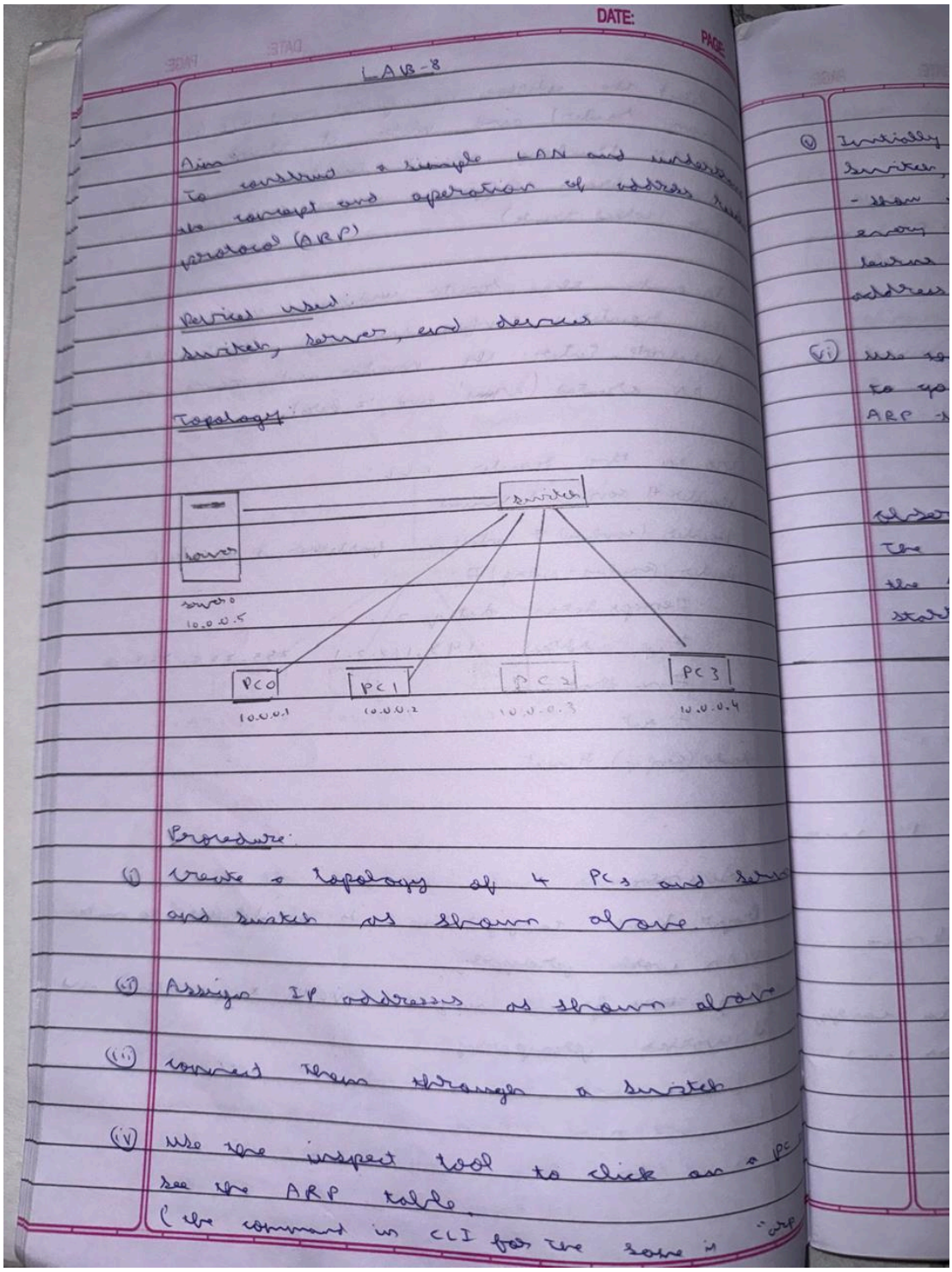




Program 10

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

Topology , Procedure and Observation:



DATE: PAGE:

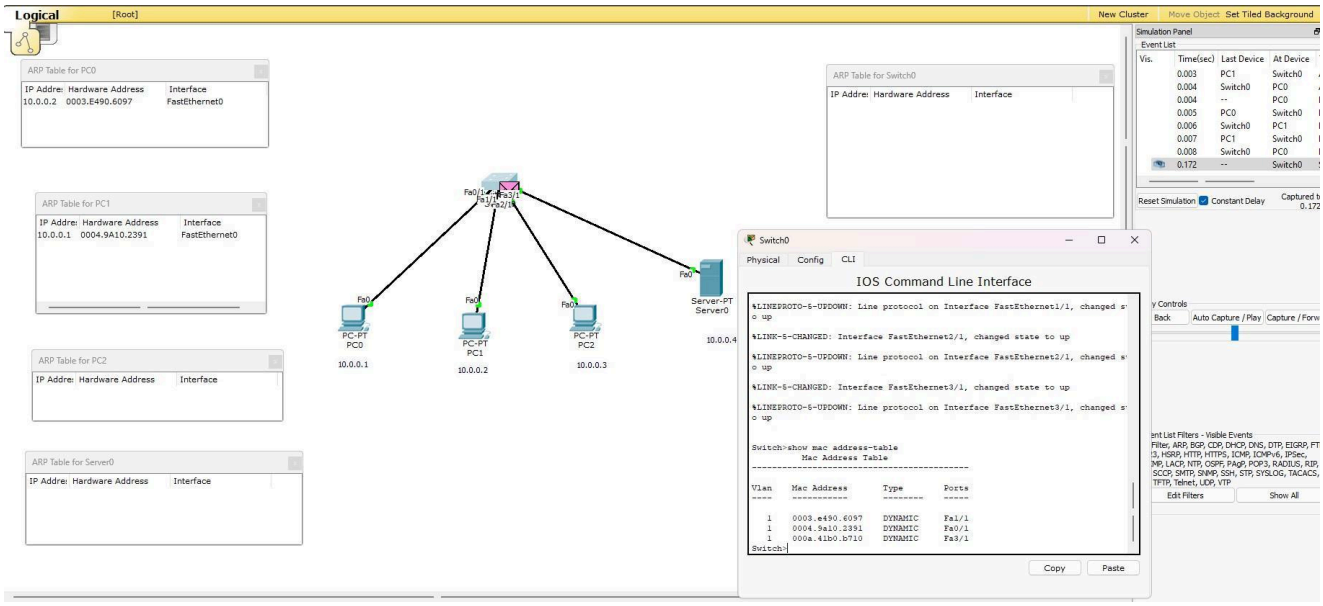
(v) Initially ARP table is empty. Also in CLI of switch, the command ~~show mac address-table~~ can be given on every transaction to see how the switch learns from transactions and builds the address table.

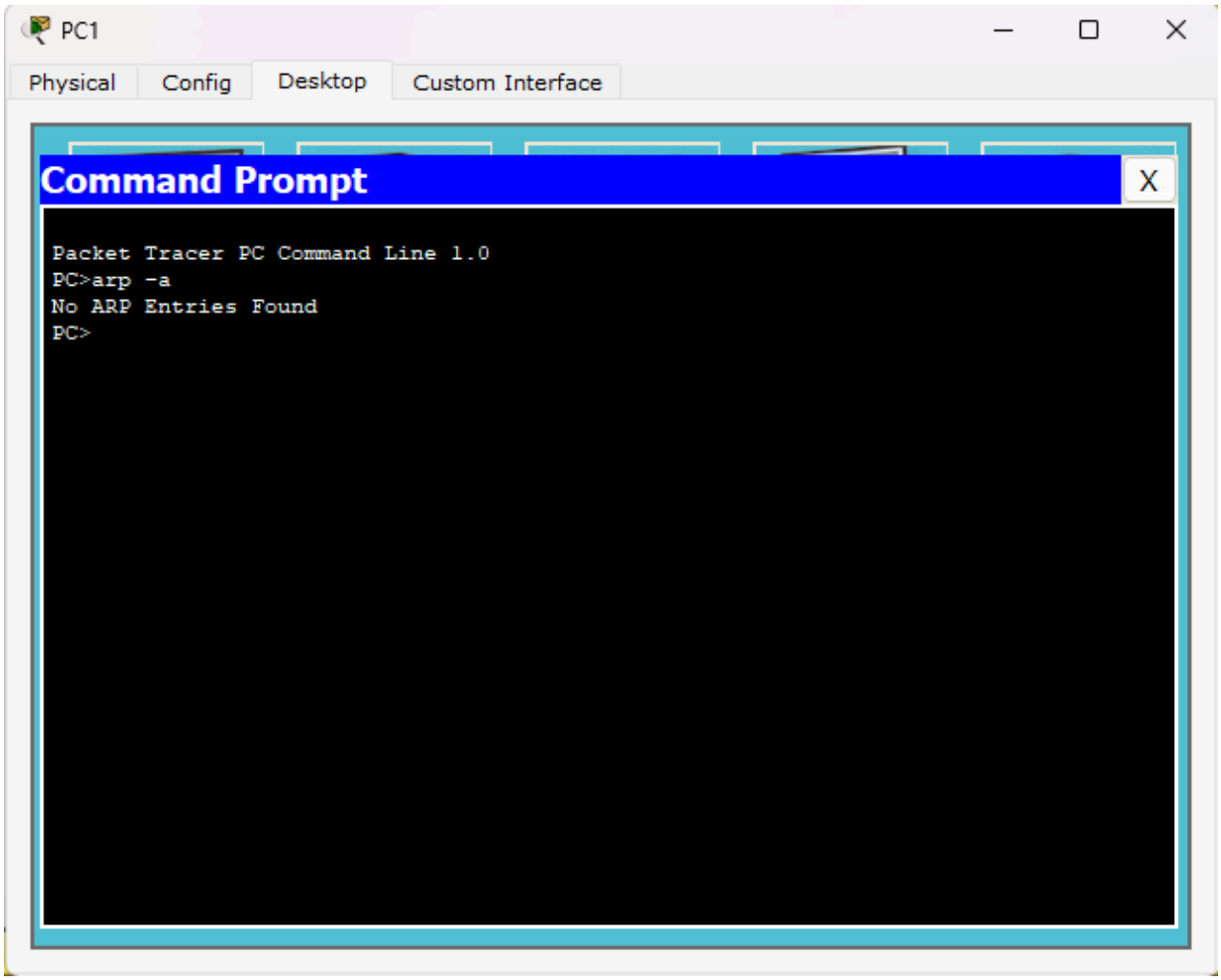
(vi) Use the capture button in simulation panel to go step by step so changes in the ARP table can be clearly seen.

Observations:

The switch as well as the router update the ARP table when a new communication starts.

Screen Shots:

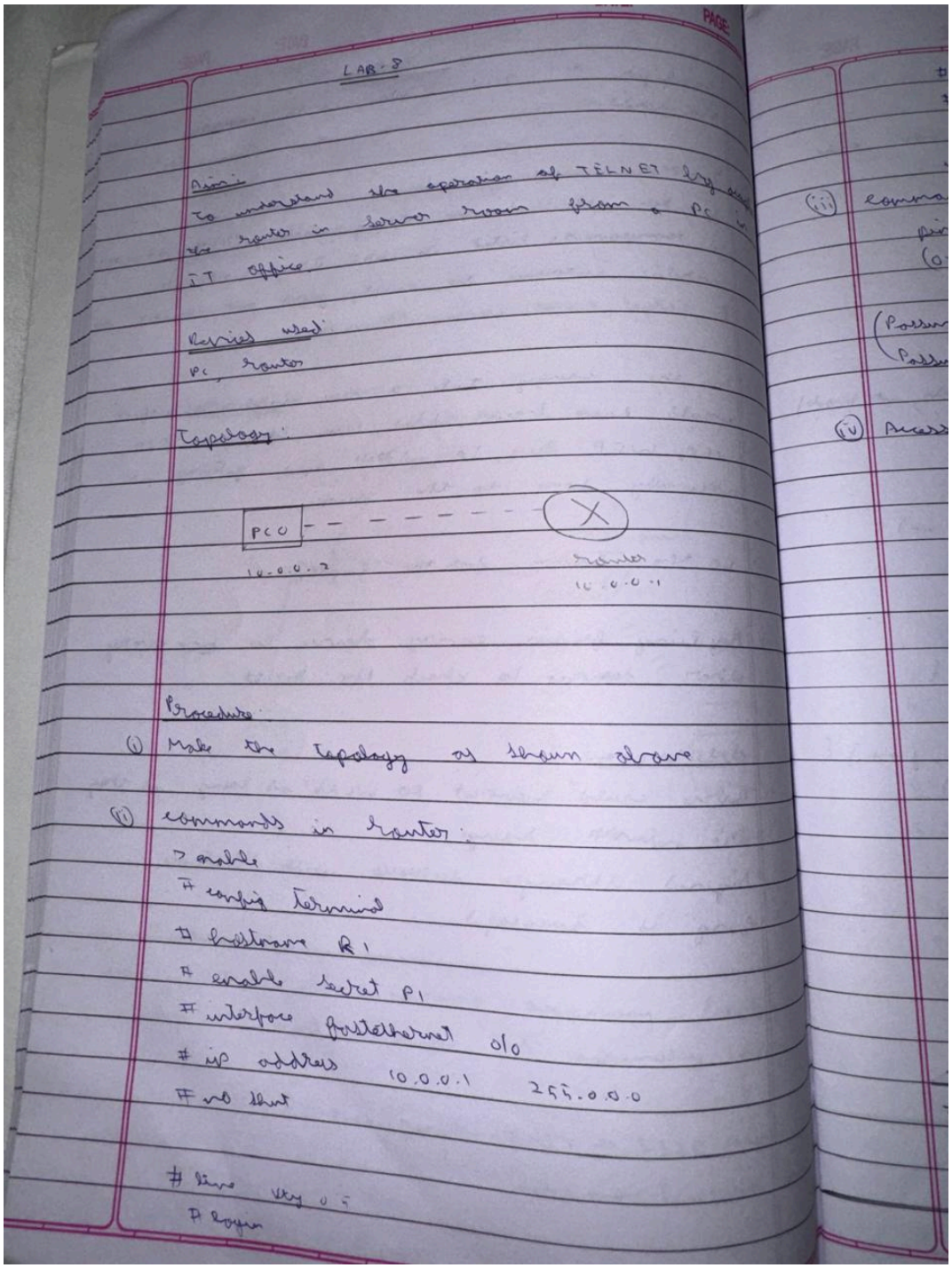




Program 11

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:



PAGE:

DATE: PAGE:

password p0
exit
exit

wr - to save changes in router

(iii) commands in PC

ping 10.0.0.1
(0% loss)

(Password for user mode verification is p0
Password for enable is p1)

(iv) Accessing router CLI from PC

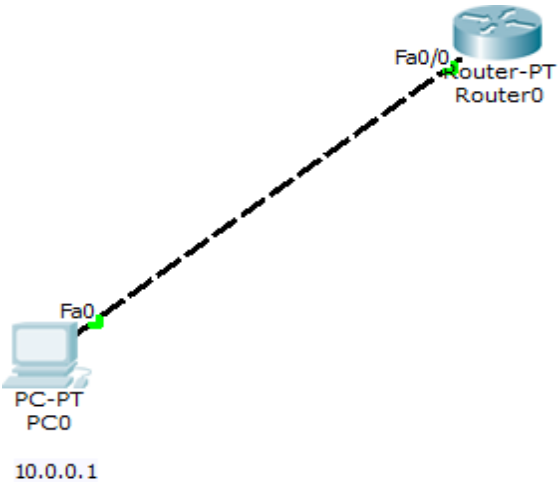
user exec verification
Password:
>1 > enable
Password:
>1 # show ip route
codes: C - - -
D - - -
N1 - - -
E1 - -
A - - -

area
* - - -
IP - - - -

(Gateway of last resort is not set)

observation:
The admin in PC is able to run commands as run in router CLI and see results from PC.

Screen Shots:



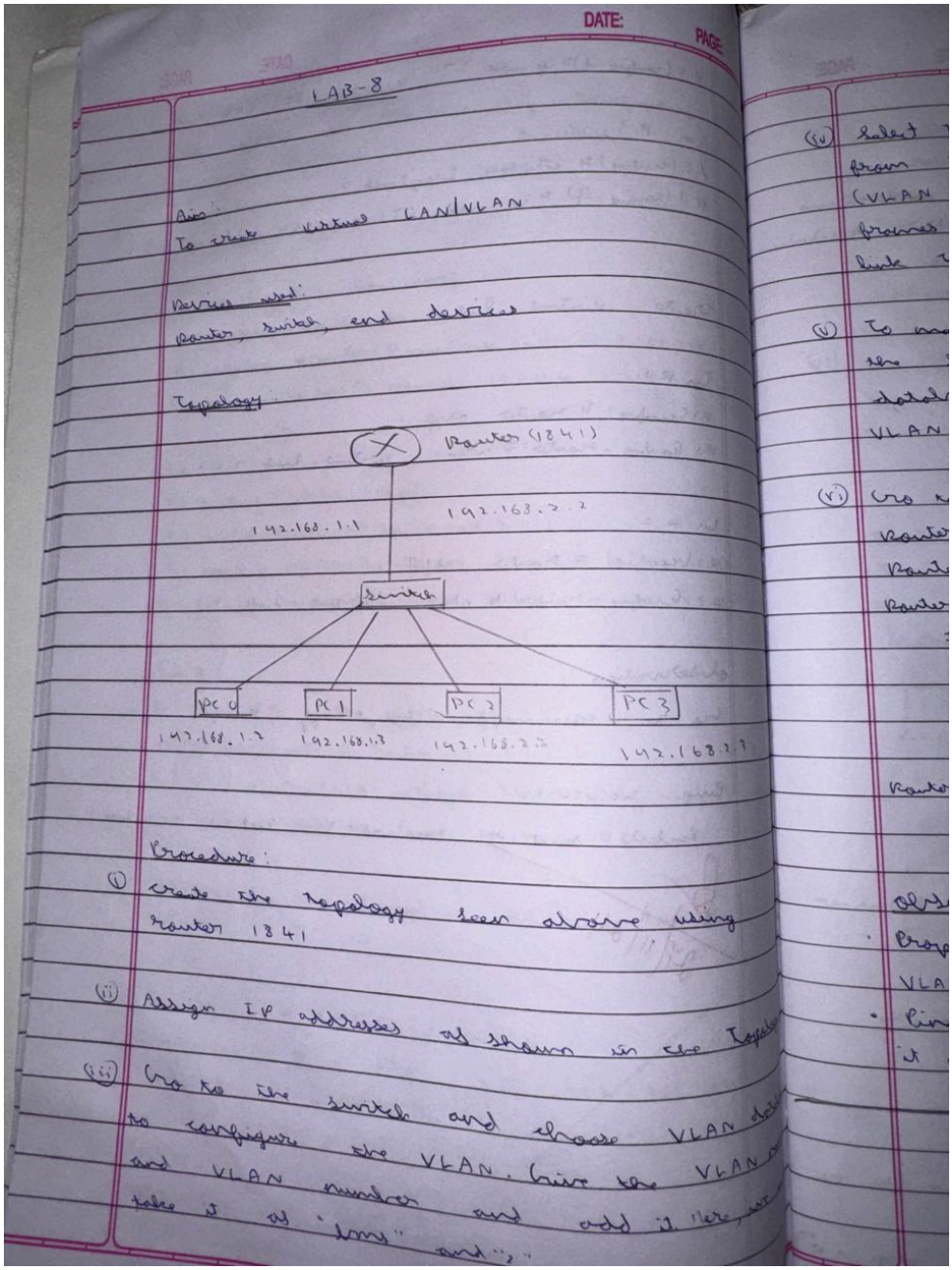
Command Prompt

```
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:



DATE: PAGE:

(iv) Select the interface fastethernet 4/1 (port switch from router) and make it trunk.
(VLAN trunking allows switches to forward frames from different VLANS over a single link called trunk)

(v) To make the router understand VLAN, go to the router's config tab and select VLAN database. Enter the number and name of VLAN created ('two' and '2' here).

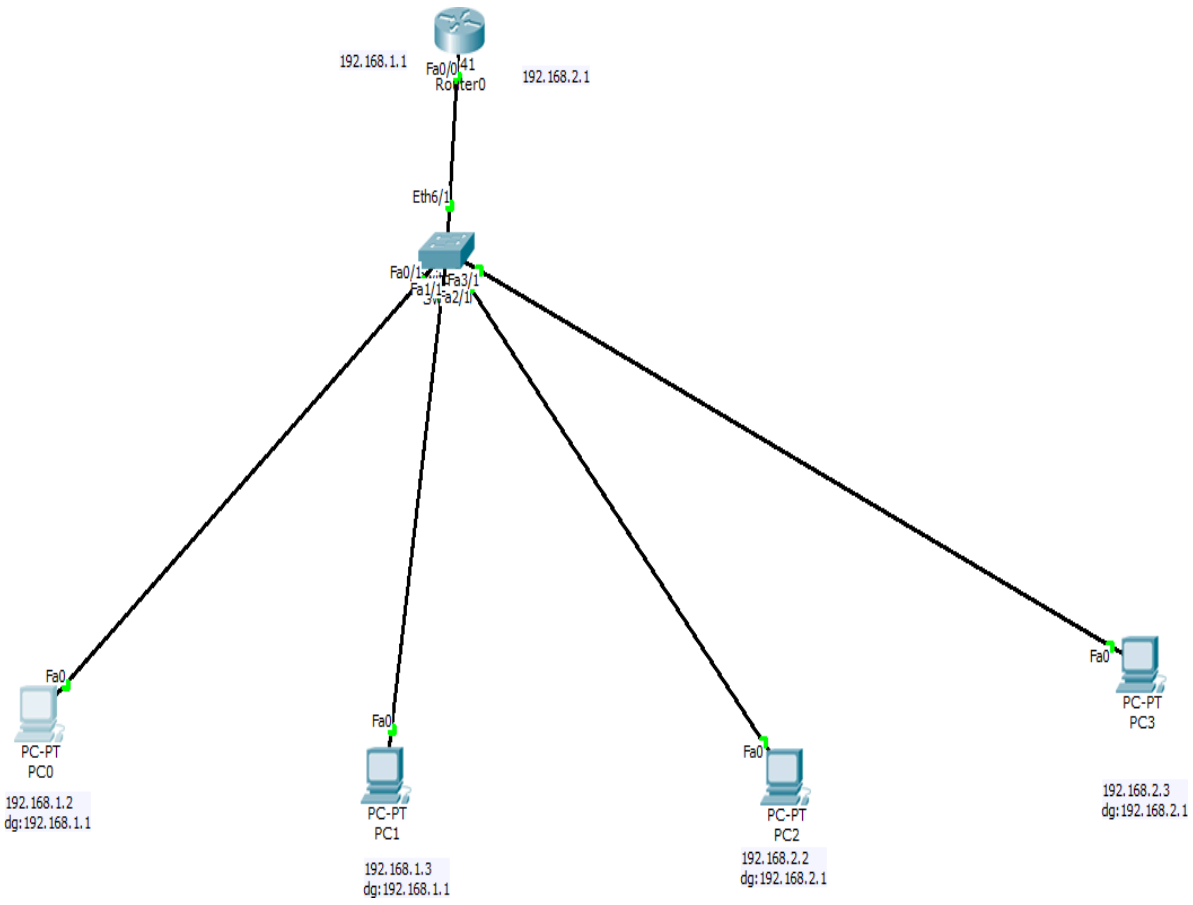
(vi) Go to the router CLI:

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

Observation:

- Proper trunk configuration is established to make VLAN work properly
- Ping from any one VLAN to another and it works properly.

Screen Shots:



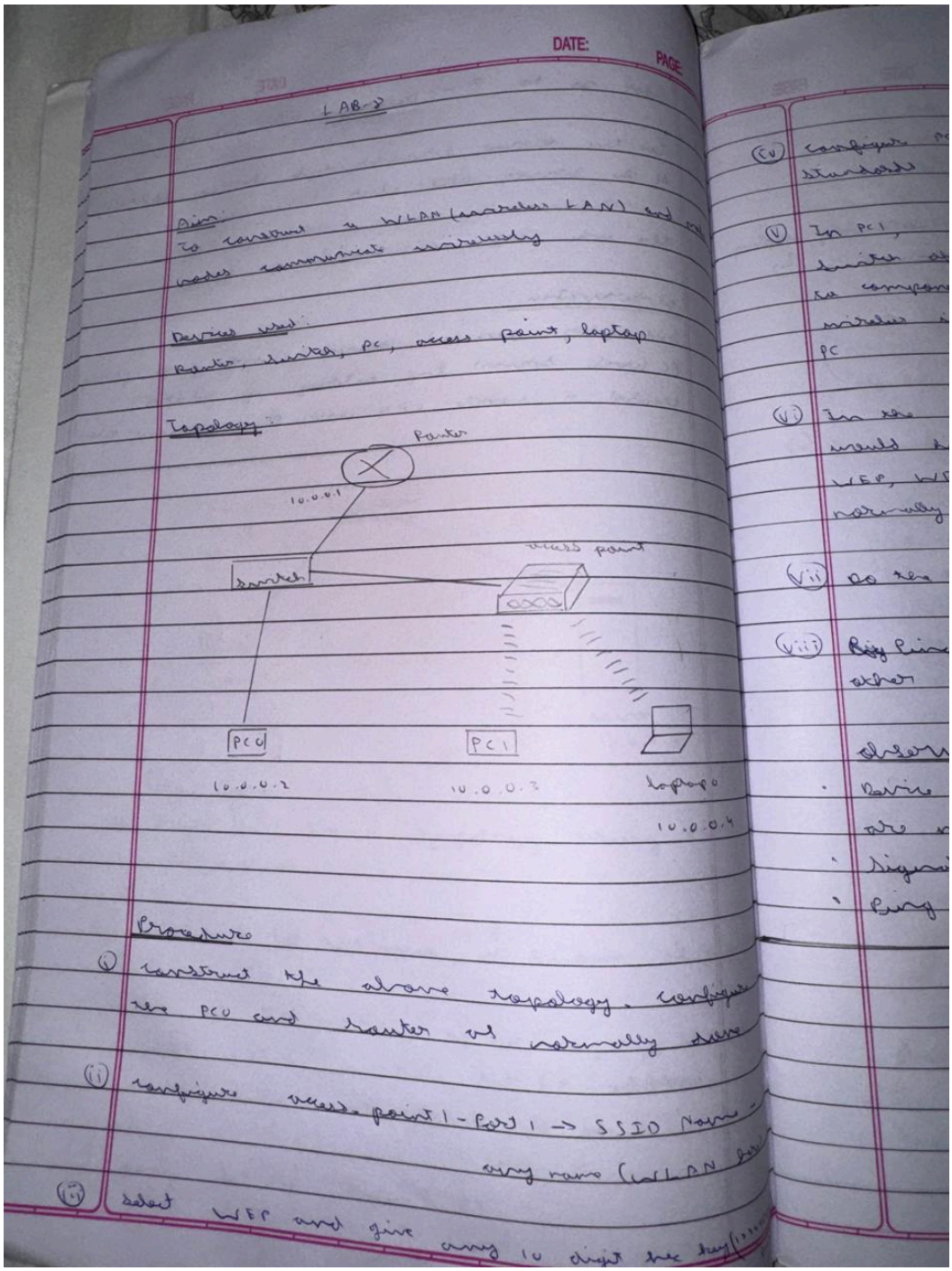
Command Prompt

```
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:



DATE: PAGE:

(iv) configure PC1 and laptop with wireless wireless standards

(v) In PC1,
switch off device, drag existing PT-HOST-NM-1AM to components listed in LHS. Drag LINKSYS WPC300N wireless interface to empty port and switch on PC

(vi) In the config tab, a new wireless interface would have been added. Now configure SSID, WEP, WEP key, IP address and gateway as normally done to the device.

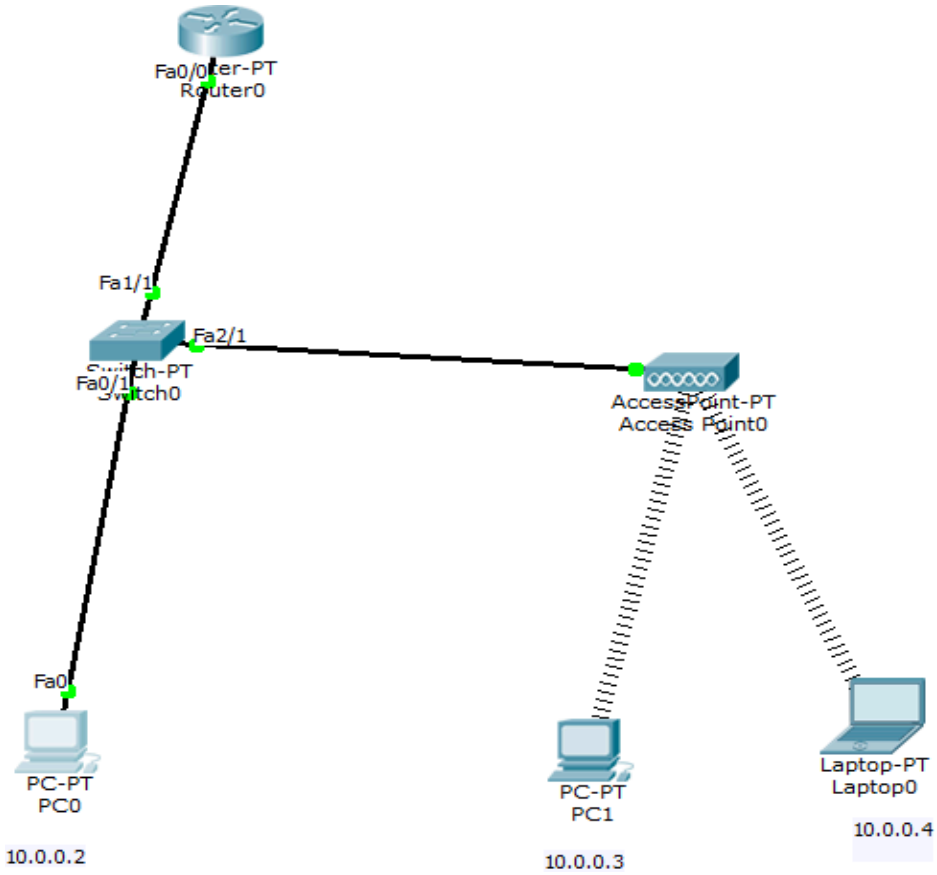
(vii) Do the same for the laptop

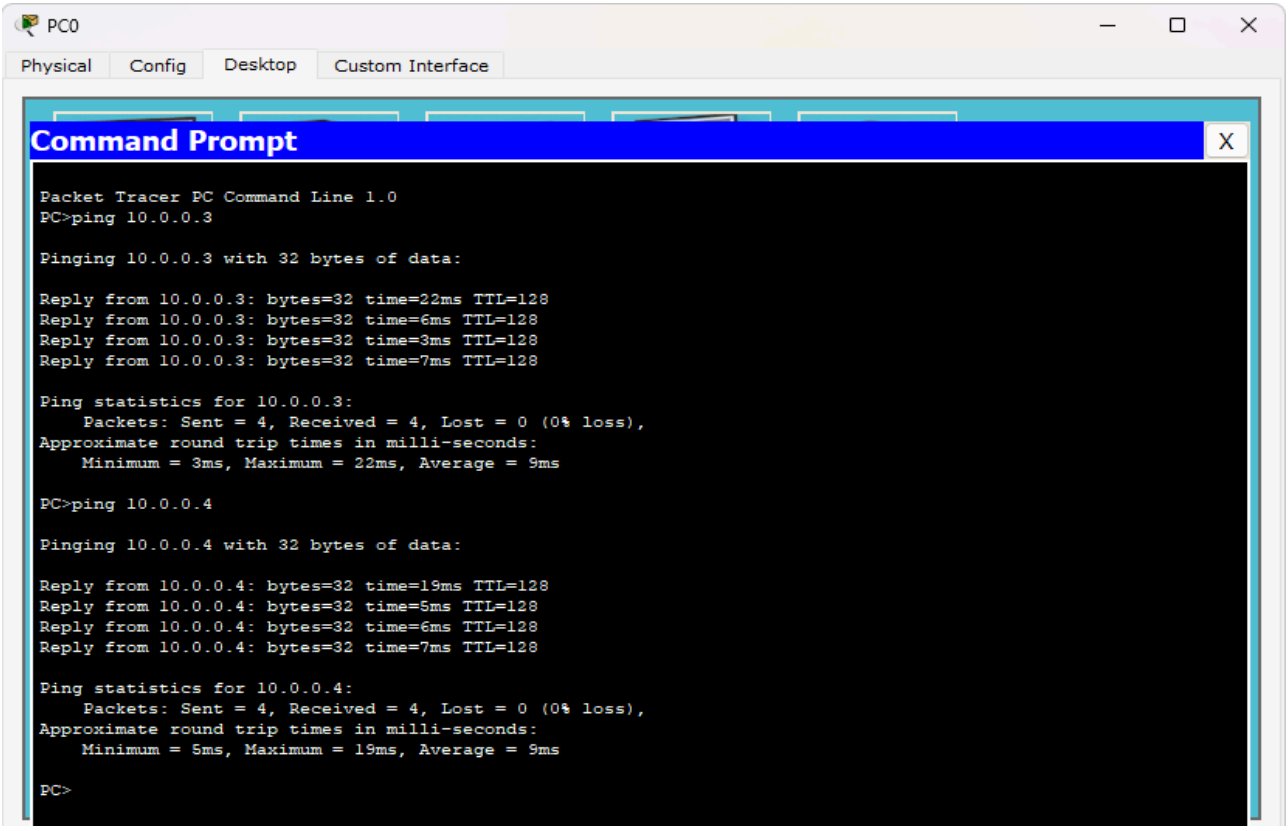
(viii) Ping from every device to every other device to check the result.

Observations:

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

Screen Shots:



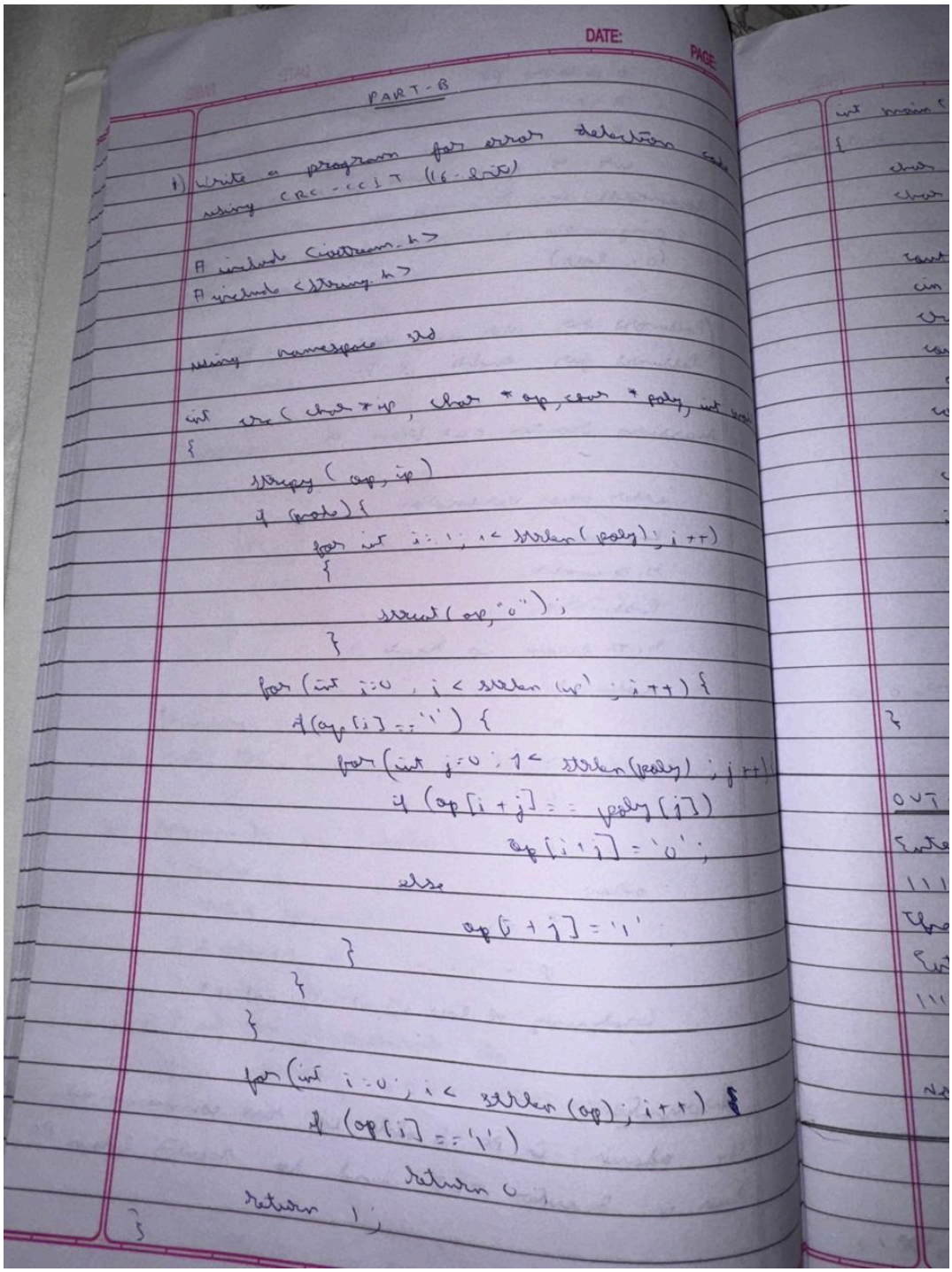


PART-B

Program 14

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

Code and Output:



```
int main()
{
    char ip[50], op[50], recv[50];
    char poly[7] = "1000100000100001";

    cout<<"Enter 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
                                occured"<<endl;
    return 0;
}
```

OUTPUT:

Enter the input message in binary
111101

The transmitted message is: 111101011001011

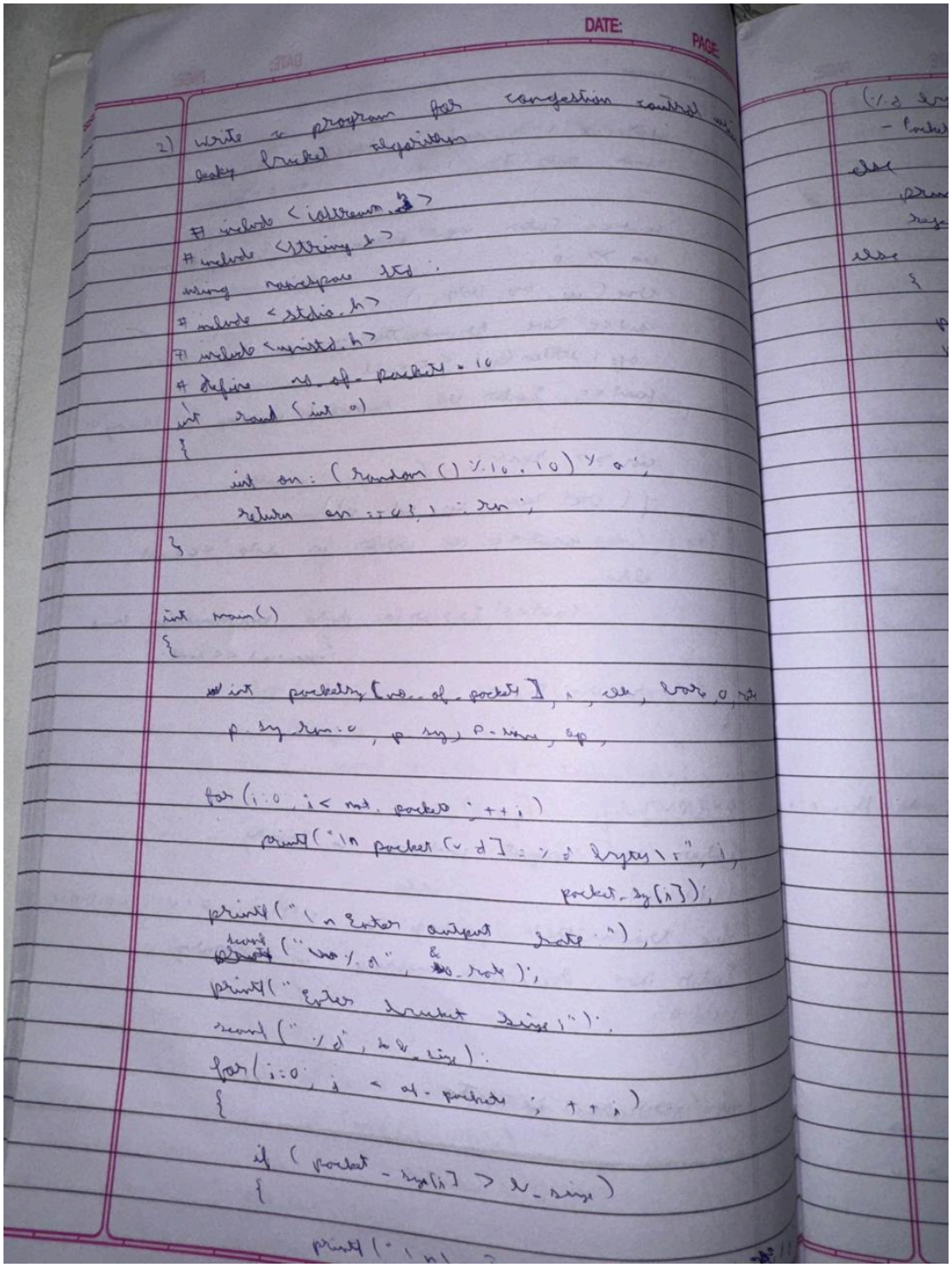
Enter the received message in binary
111101

No error in data

Program 15

Write a program for congestion control using Leaky bucket algorithm.

Code and Output:



DATE: PAGE:

```
(%d bytes) is greater than bucket capacity (%d bytes)
 - Packet rejected", packet_sz[i], b_size);
else
    printf("\n\n Bucket capacity exceeded, packet
    rejected");
else
    {
     p_sz_rem = packet_sz[i];
     printf("\n\n Incoming packet size %d",
                                   packet_sz[i]);
     printf("\n Bytes remaining to transmit %d",
                                   p_sz_rem);
     p_time = rand(4) * 10;
     printf("\n Time left for transmission %d",
                                   p_time);
    for (clk = 10; clk <= p_time; clk += 10)
    {
       sleep(1);
       if (p_sz_rem)
       {
          if (p_sz_rem <= o_rate)
               op = p_sz_rem, p_sz_rem = 0;
       }
       else
            op = o_rate, p_sz_rem -= o_rate;
       printf("\n Packet of size %d transmitted", op);
       printf("Bytes remaining to transmit: %d",
                                   p_sz_rem);
       }
       else
       {
       printf("\n time left for transmission %d",
                                   p_time - clk);
```

DATE: PAGE:

printf("\n no of packets to transmit");
}
}
}
}
}

OUTPUT:

packet[0] = 30 bytes
Packet[1] : 10 bytes
packet[2] : 10 bytes
packet[3] : 50 bytes
packet[4] : 30 bytes

Enter the output rate : 100
Enter the packet size : 10

For incoming packet no 30
Bytes remaining to transmit : 30

Time left for transmission : 10 sec
Packet of size 30 transmitted - bytes remaining to tr

Time left for transmission : 0 sec
No packets to transmit

Incoming packet no : 10
Bytes remaining to transmit : 10
Time left for transmission : 10
Packet of size 10 transmitted - Bytes remaining to
transmit : 0

DATE: PAGE:

Time left for transmission : 10 units
No packets to transmit
Time left for transmission : 0 units
No packets to transmit

Incoming packet size : 10
Bytes remaining to transmit : 10
Time left for transmission : 10 units
Packet of size 10 transmitted - Bytes remaining to transmit : 0

Incoming packet size 50
Bytes remaining to transmit : 50
Time left for transmission : 10 units
Packet of size 50 transmitted - Bytes remaining to transmit : 0

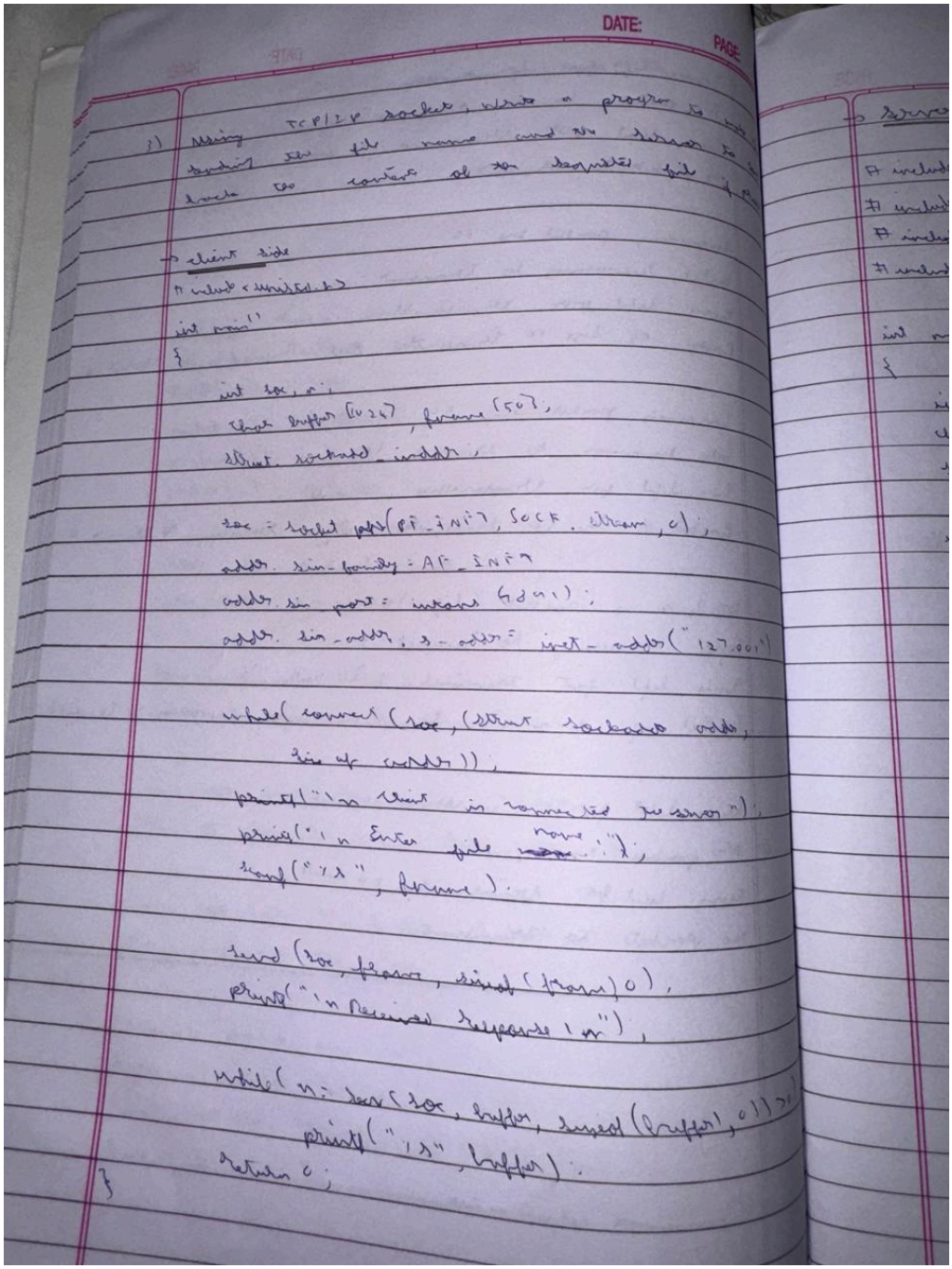
Incoming packet size : 30
Bytes remaining to transmit : 30
Time left for transmission : 30 units
Packet of size 30 transmitted - Bytes remaining to transmit : 0

Time left for transmission : 10 units
No packets to transmit
Time left for transmission : 0 units
No packets to transmit.

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.

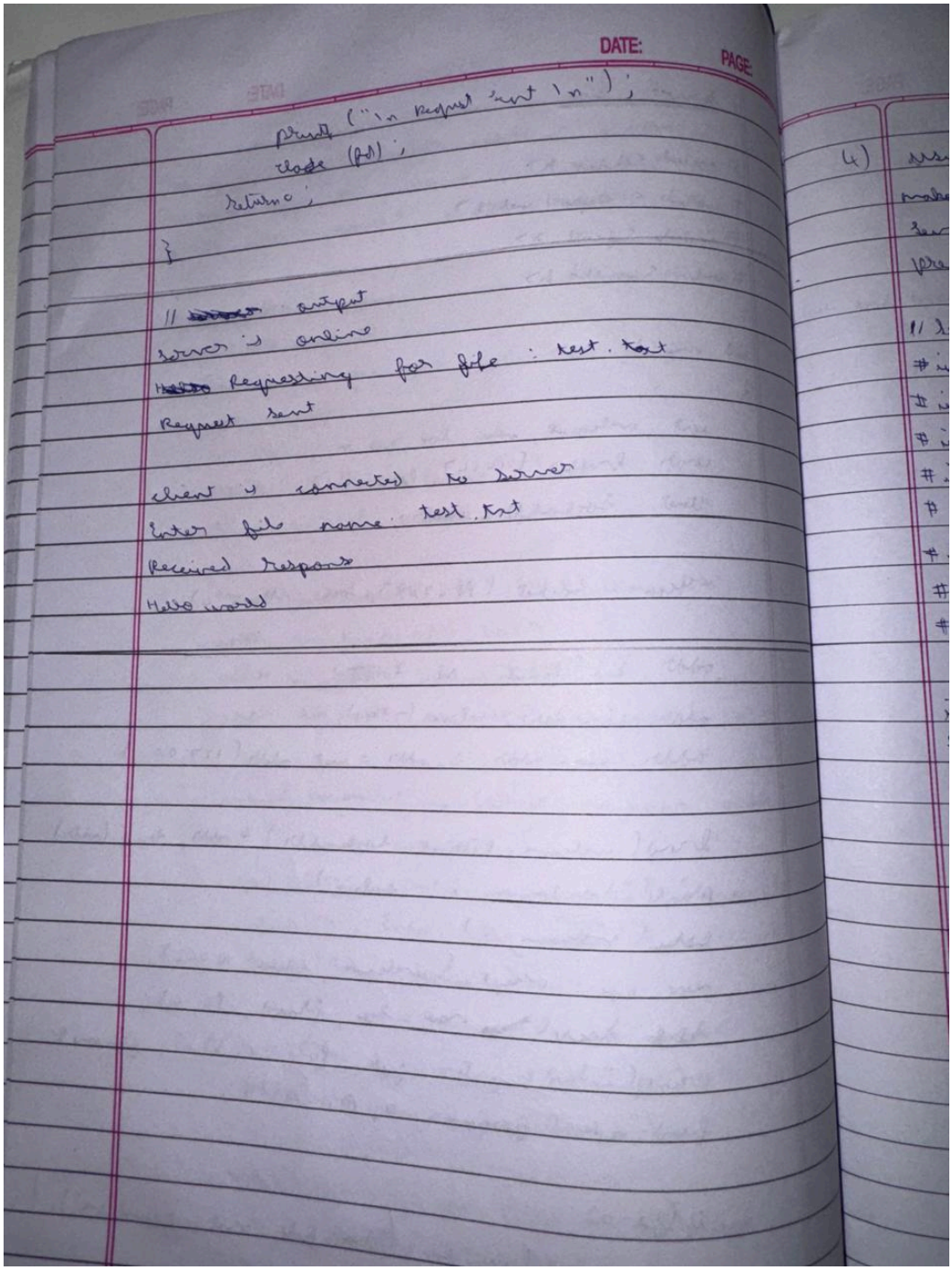
Code and Output:



DATE: PAGE:

→ Server side

```
#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <fcntl.h>
#include <unistd.h>

int main()
{
    int welcome, new_sock, fd, n;
    char buffer[1024], fname[10];
    struct sockaddr_in addr;

    welcome = socket(PF_INET, SOCK_STREAM, 0);

    addr.sin_family = AF_INET;
    addr.sin_port = htons(7891);
    addr.sin_addr.s_addr = inet_addr("127.0.0.1");

    bind(welcome, (struct sockaddr*)&addr, sizeof(addr));
    printf("\n Server is online");
    listen(welcome, 5);
    new_sock = accept(welcome, NULL, NULL);
    recv(new_sock, fname, 10, 0);
    printf("\n Requesting for file: %s\n", fname);
    fp = open(fname, O_RDONLY);

    if (fp < 0)
        send(new_sock, "\n file not found !!", 15, 0);
    else
        while((n = read(fp, buffer, sizeof(buffer)) > 0)
            send(new_sock, buffer, n, 0);
```



printf("\n Request sent \n");
close(fd);
return 0;
}

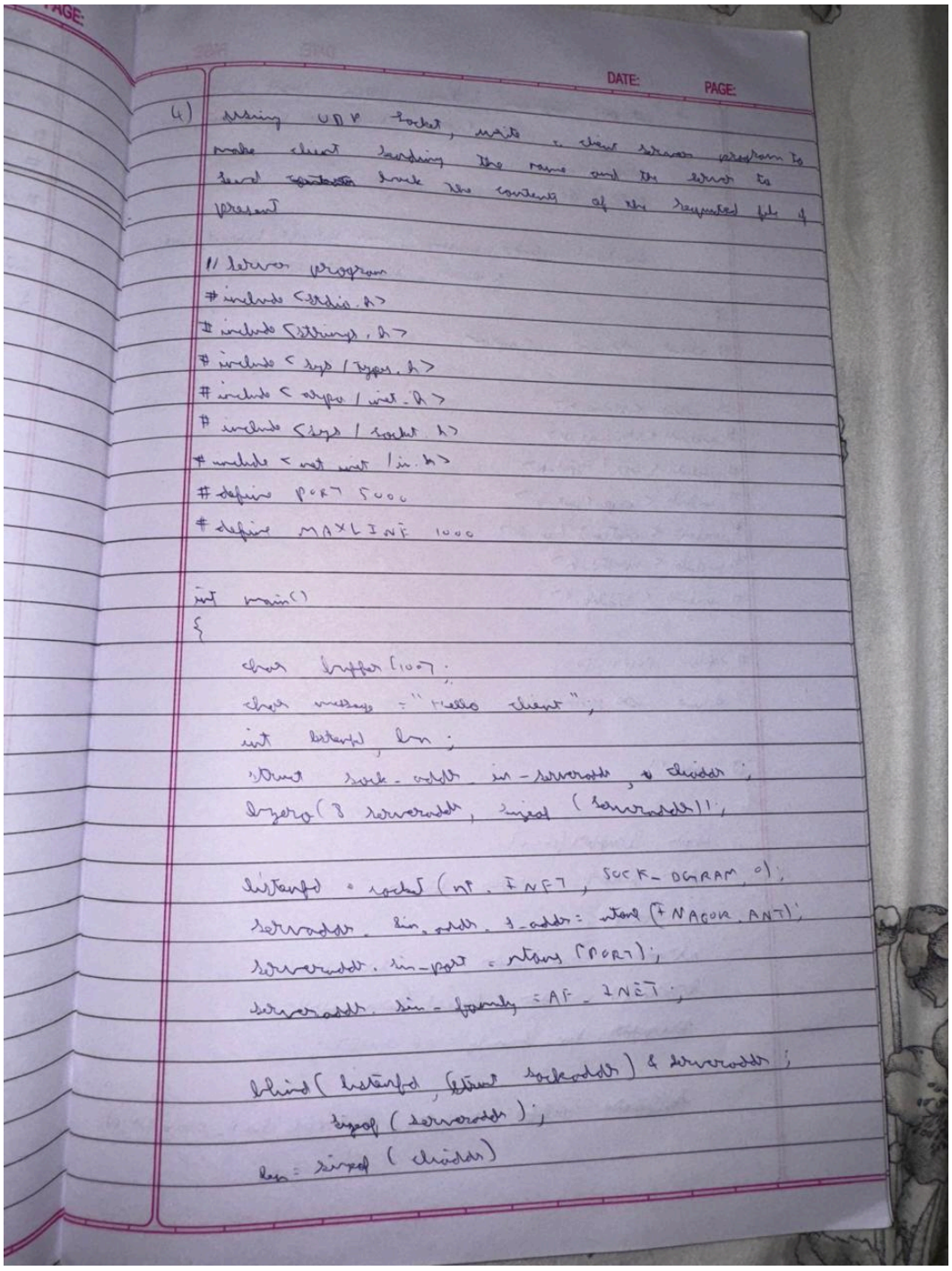
// output
Server is online
Requesting for file : test.txt
Request sent

client is connected to server
Enter file name: test.txt
Received response
Hello world

Program 17

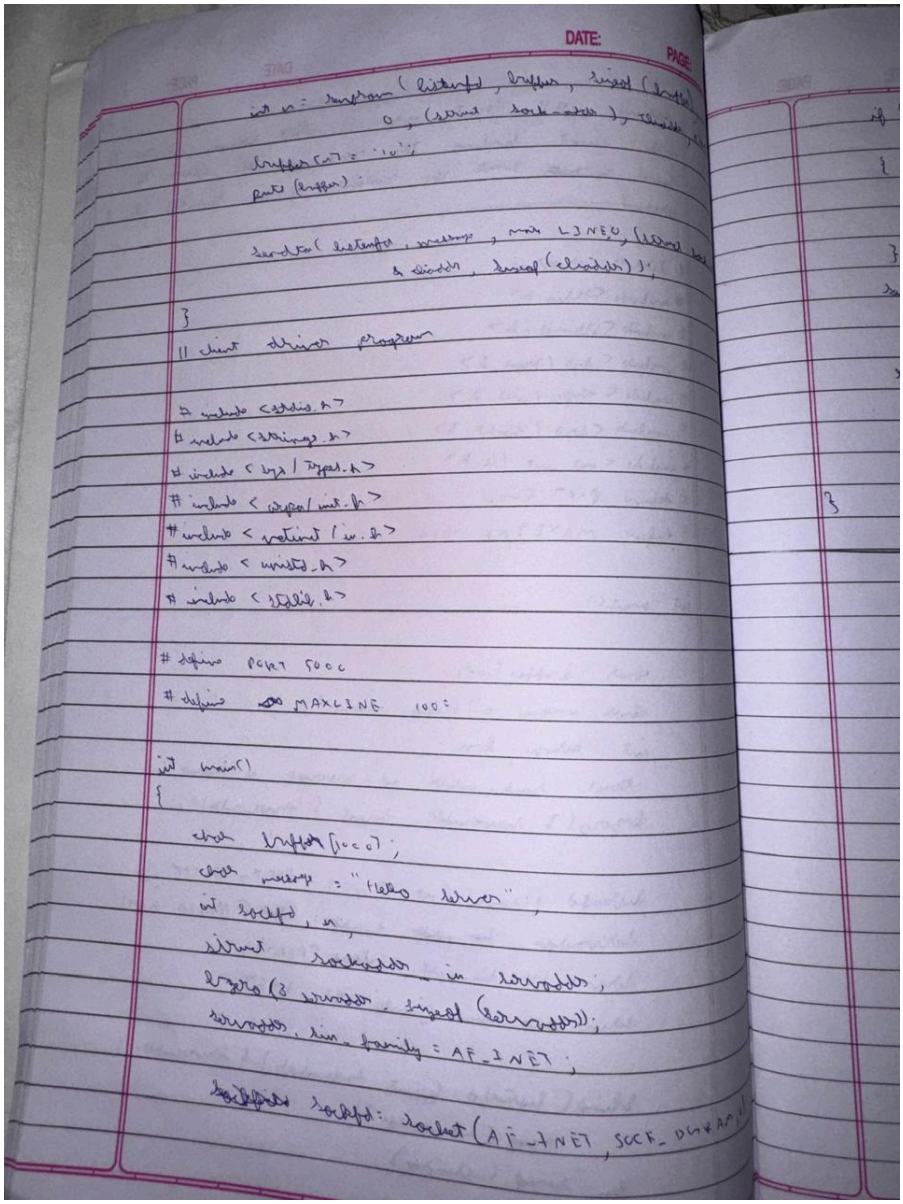
Using UDP sockets, write a client-server program to make the client send the file name and the server to send back the contents of the requested file if present.

Code and Output:



4) using UDP socket, write a client server program to make client sending the name and the server to send back the contents of the requested file if present

```
// server program
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <netinet/in.h>
#define PORT 5000
#define MAXLINE 1000

int main()
{
    char buffer[100];
    char *message = "Hello client";
    int listenfd, len;
    struct sockaddr_in servaddr, cliaddr;
    bzero(&servaddr, sizeof(servaddr));

    listenfd = socket(AF_INET, SOCK_DGRAM, 0);
    servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
    servaddr.sin_port = htons(PORT);
    servaddr.sin_family = AF_INET;

    bind(listenfd, (struct sockaddr*)&servaddr, sizeof(servaddr));
    len = sizeof(cliaddr);
```



DATE:
PAGE:

```
int n = recvfrom(listenfd, buffer, sizeof(buffer)
                 0, (struct sockaddr *), &len);
buffer[n] = '\0';
puts(buffer);

sendto(listenfd, message, MAXLINE, (struct sockaddr
       *)&cliaddr, sizeof(cliaddr));
}
// client driver program

#include <stdio.h>
#include <strings.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>

#define PORT 5000
#define MAXLINE 1000

int main()
{
    char buffer[100];
    char *message = "Hello Server";
    int sockfd, n;
    struct sockaddr_in servaddr;
    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_family = AF_INET;

    sockfd = socket(AF_INET, SOCK_DGRAM,
```

DATE: PAGE:

```
if (connect (sockfd, (struct sockaddr) &server,
                sizeof (server)) < 0)
{
    printf ("\n Error : connection failed");
    exit (0);
}

sendto (sockfd, message, MAX (INT, 0, (struct sockaddr),
                NULL, sizeof (server));

recvfrom (sockfd, buffer, sizeof (buffer), 0,
                (struct sockaddr), NULL, NULL);
puts (buffer);
close (sockfd);
}
```

// server output
Server is online
Hello Server

// client output
Hello client