# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**"JnanaSangama", Belgaum -590014, Karnataka.**

**LAB RECORD**

# Computer Network Lab (23CS5PCCON)

***Submitted by***

**Swapnil Sahil (1BM22CS300)**

***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 **Swapnil Sahil (1BM22CS300),** 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.

| | |
|---|---|
| Dr. Shashikala<br>Associate Professor<br>Department of CSE, BMSCE | Dr. Kavitha Sooda<br>Professor & HOD<br>Department of CSE, BMSCE |

# Index

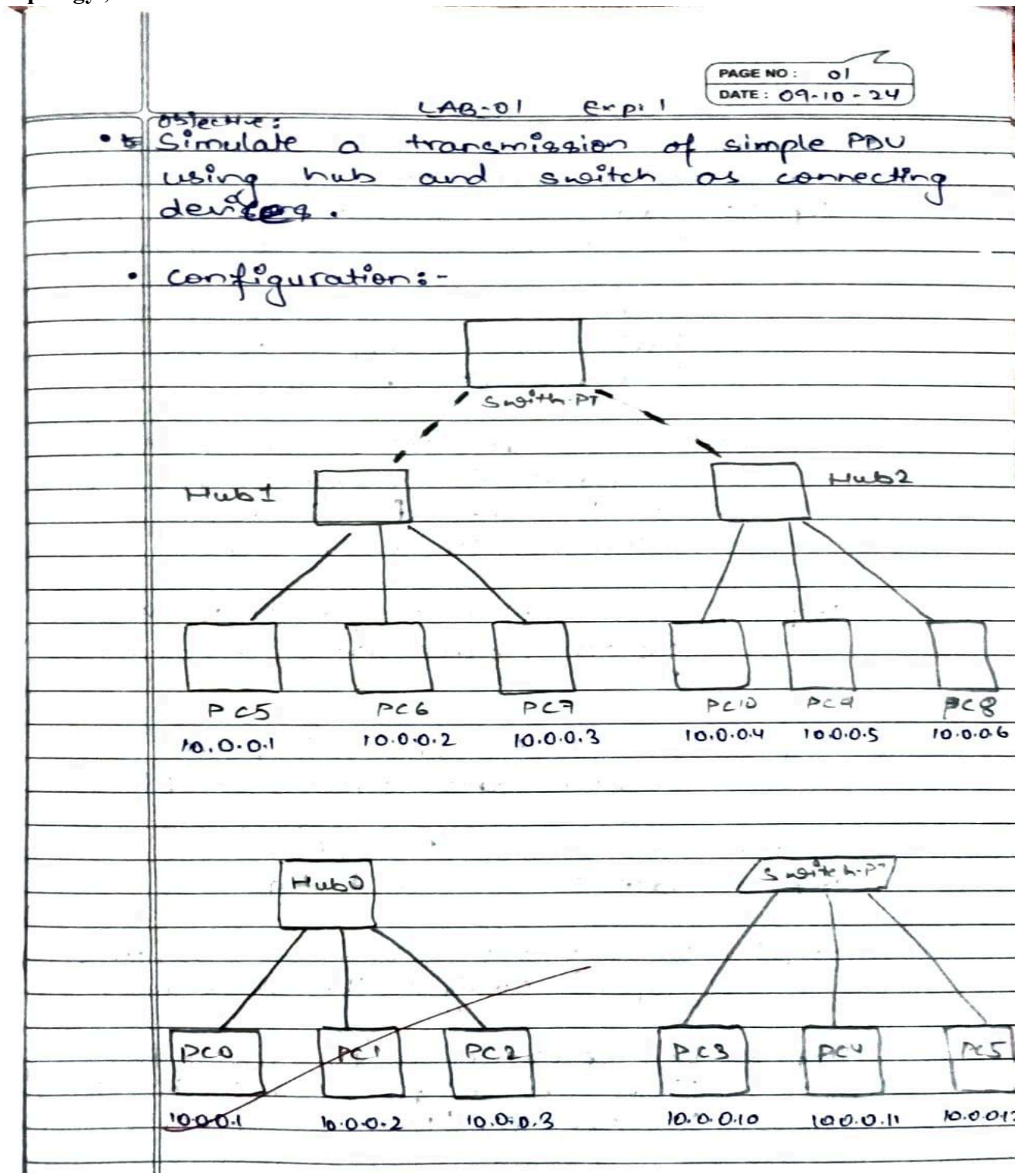


Github Link:
Git hub link

## 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 3 PC's along with 1 hub. All generics to be chosen.

step: 2 : connect all the PC's to the hub using copper straight-Through wire.

step: 3 :- Select each PC, then go to configure tab, select fastethernet and add IP addresses (10.0.0.1, 10.0.0.2, 10.0.03)

step: 4 :- Select Add simple PDU, then select source and destination PC's.

step: 5 :- Switch to simulation tab and observe after clicking on play.

2) 3 end devices and 1 Switch.

step: 1 :- place 3 PC's and 1 generic switch

Step 2 : connect all PC's to switch using copper straight through wires.

Step 3 : By selecting PC, go to configure then select fastethernet to set IP addresses (10.0.0.10, 10.0.0.11, 100.0

Step 4 : select add simple PDU and select

PAGE NO :
DATE :

source & destination PC's.

Step5:- Observe the simulation in simulation mode.

3. 6 end devices, 2 hubs and 1 switch

Step1:- Place 6 PC's and connect 3 PC's to one hub and other 3PC's to another hub using copper straight through wires.

Step2: Connect both the hubs to a switch using copper cross over wire.

Step3: Select each PC, then go to configure tab and set the IP addresses.

Step4: Select Add simple PDU, then select soure and destination PC's.

Step5: Switch to simulation tab and observe after clicking on play.

- Difference between hub and switch

| Hub | Switch |
|---|---|
| 1) Operated on physical layer. | 1) Operated on Data Link layer. |
| 2) It is broadcast type transmission. | 2) It is unicast, multicast and broadcast type transmission. |
| 3) Hubs have 4/12 ports. | 3) Switch can have 24 to 48 ports. |

PAGE NO :
DATE :

| | |
|---|---|
| 4) Hub is half duplex transmission mode. | 4) Switch is full duplex transmission mode |
| 5) cannot be used as repeater | 5) can be used as repeater. |

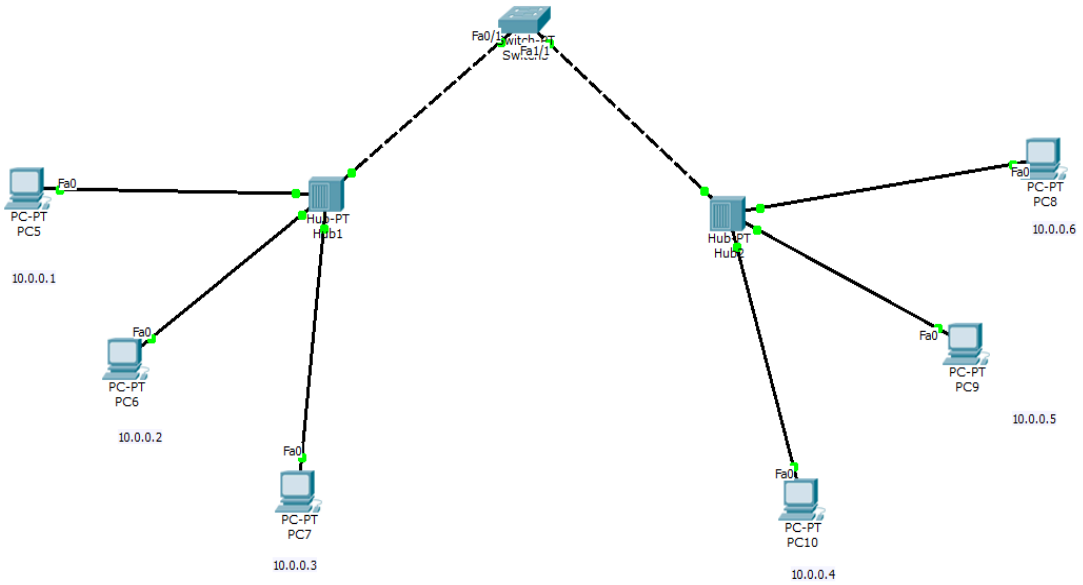
- observation:-

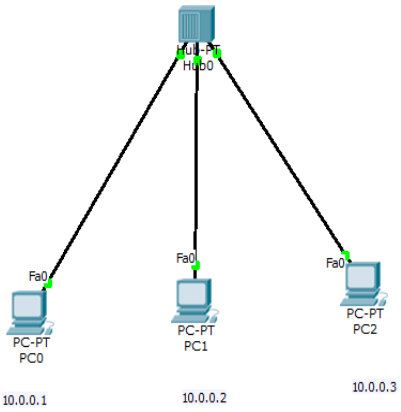
1. The hub sends the packet to all available devices. The destination PC accepts the packet and sends the acknowledgement back. All the remaining PC's repeat the packet.

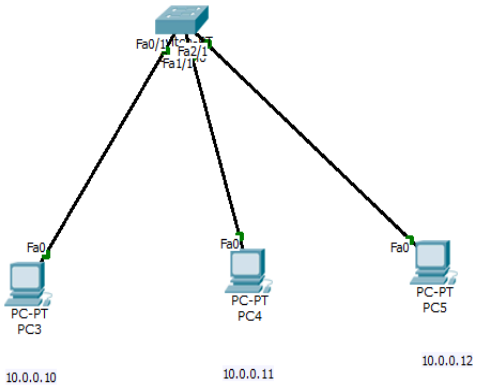
2. The switch sends the packet only to the destination PC which accepts and sends back acknowledgement.

3. The hub10 receive packet and sends to other PC and the switch, the switch sends it to hub1 which transmit it to all remaining PC's, the destination PC accept the packet and acknowledge by sending it back by hub1 → switch → hub10

**Screen Shots:**



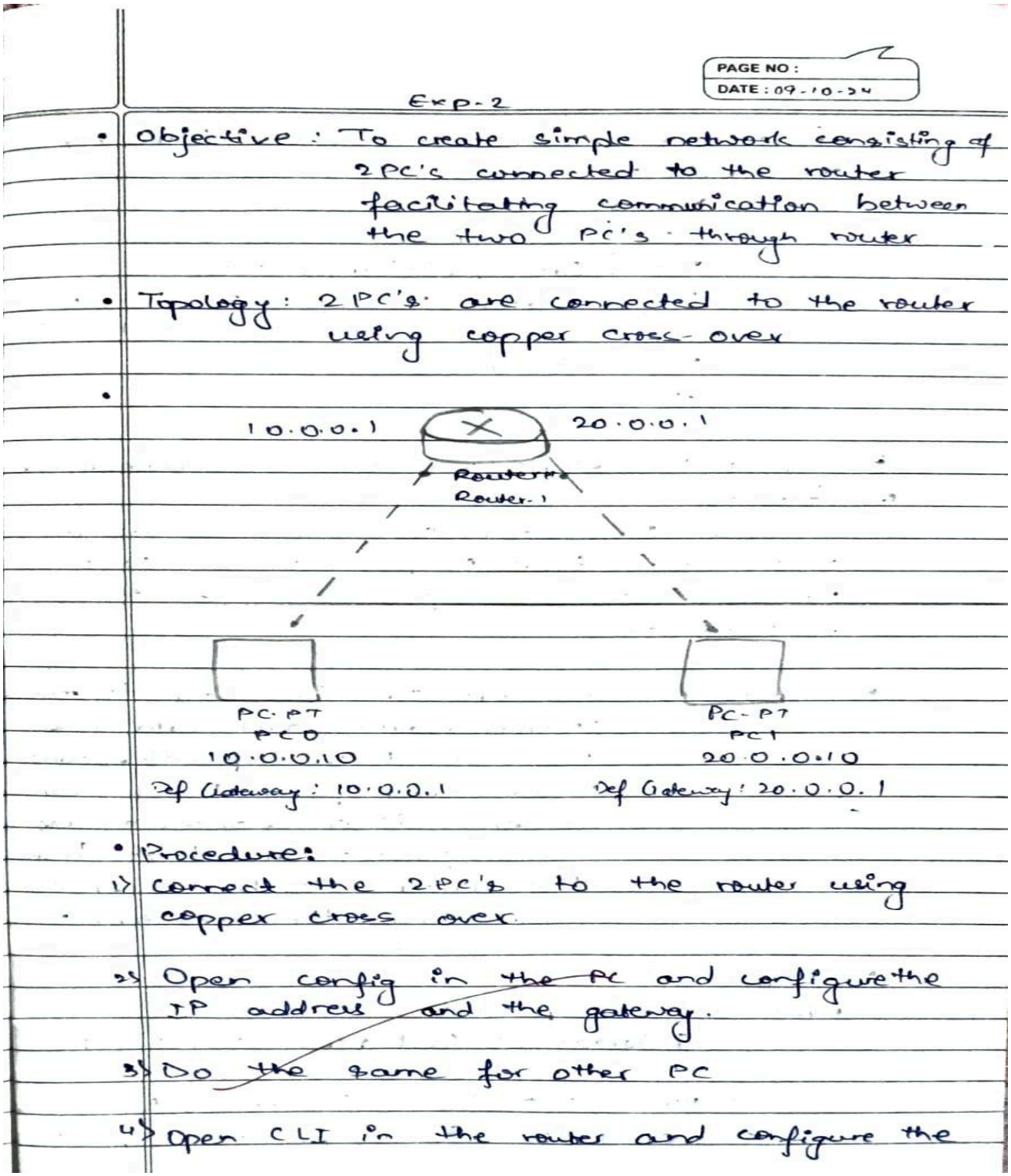



**Program 2**

**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 :

fastethernet connection by the following commands :-

> enable
> config terminal
> interface fastethernet 0/0
> ip address 10.0.0.1 255.0.0.0
gateway subnet mask
> no shutdown
exit

5) Repeat the steps for other PC connection.
6) Go to
6) Go to CLI for sending messages from a PC and type : ping 20.0.0.10 where 20.0.0.10 is the ip address of the receiver PC.

- Observation : Routers can be used to manage communication and data transfer between two different network. While doing ping test we can observe that chances of losing one packet are high because the router will be busy in establishing the connection.

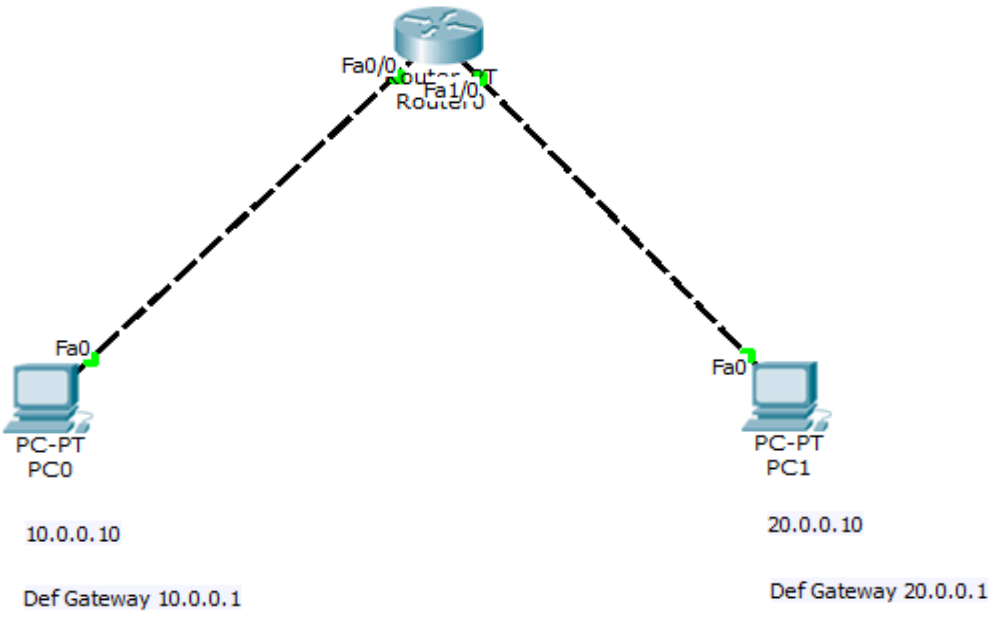
- Output :

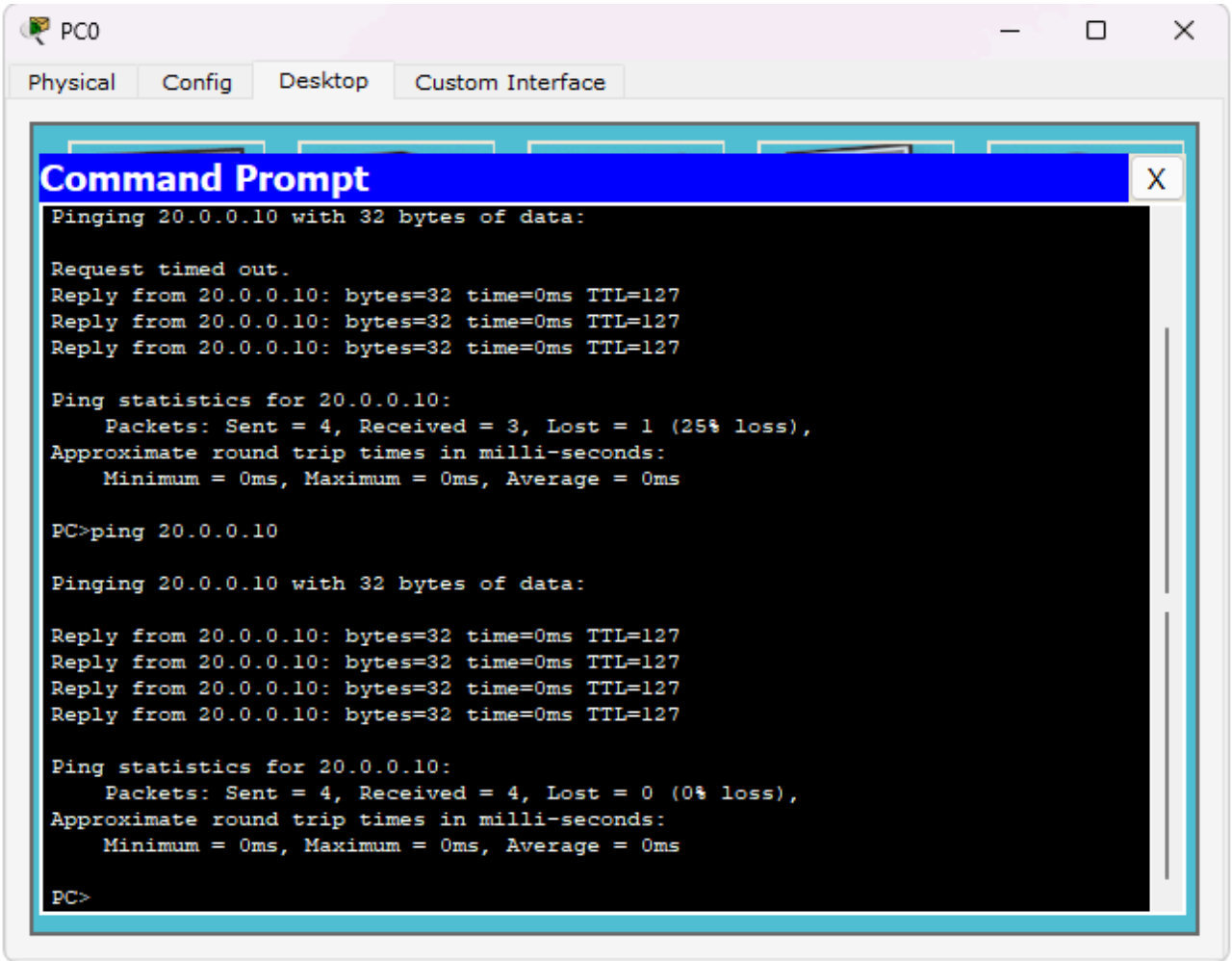
Router> show ip route
Gateway of last resort is not set

C 10.0.0.0/8 is directly connected to fastethernet
C 20.0.0.0/8 is directly connected to fastethernet

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

**Screen Shots:**



**Program 3**

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

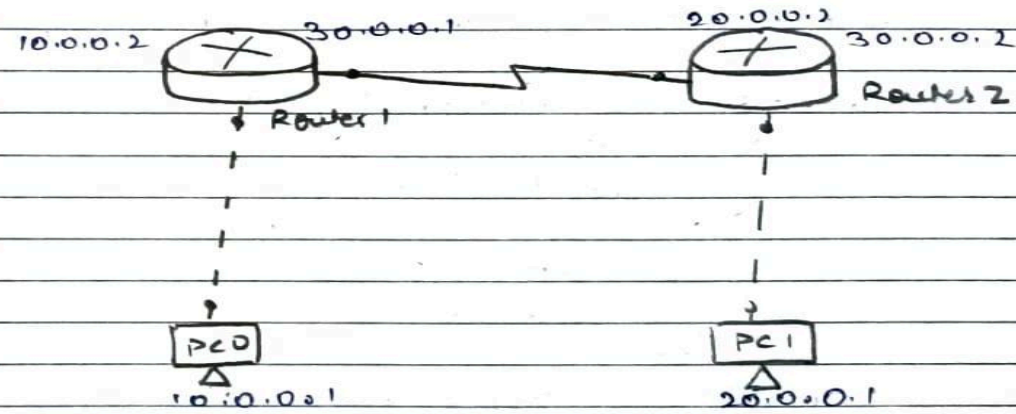
**Topology , Procedure and Observation:**

PAGE NO :
DATE : 16-10-24

LAB-02

- Objective: Configure default route, static route to the router. To connect two PC's on two different networks using two routers.

- Topology: 2 PC's are connected to two different routers, through copper cross-over and routers are connected through serial DCE.




Def gateway: 10.0.0.2

Def gateway: 20.0.0.2

- Procedure:

1) Connect two PC's to the two different routers using copper cross-over.

2) Connect the routers using serial DCE.

3) Open config in the PC and configure the IP address and the gateway.

PAGE NO :
DATE :

4) Do the same for the other PC.

5) Open CLI in the router and configure the fastethernet connection by the following commands :-

```
> enable
> config terminal
> interface fastethernet 0/0
> ip address 10.0.0.2 255.0.0.0
> no shutdown
> exit
```

6) Repeat the steps for other router.

7) Connect the routers by the following commands

```
> enable
> config terminal
> interface serial 2/0
> ip address 30.0.0.1 255.0.0.0
> no shutdown
  exit
```

8) Repeat the steps for other router.

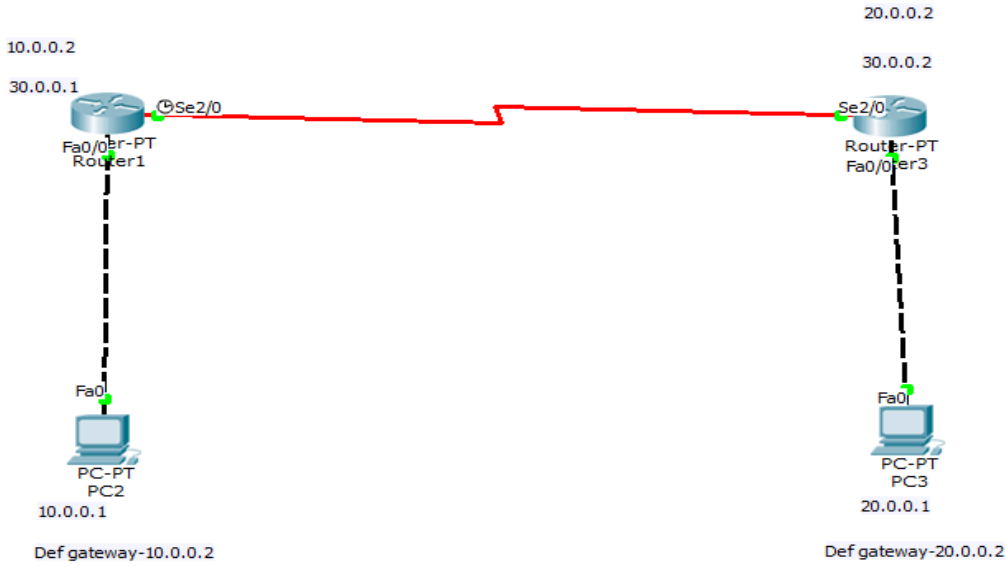
- Observation:

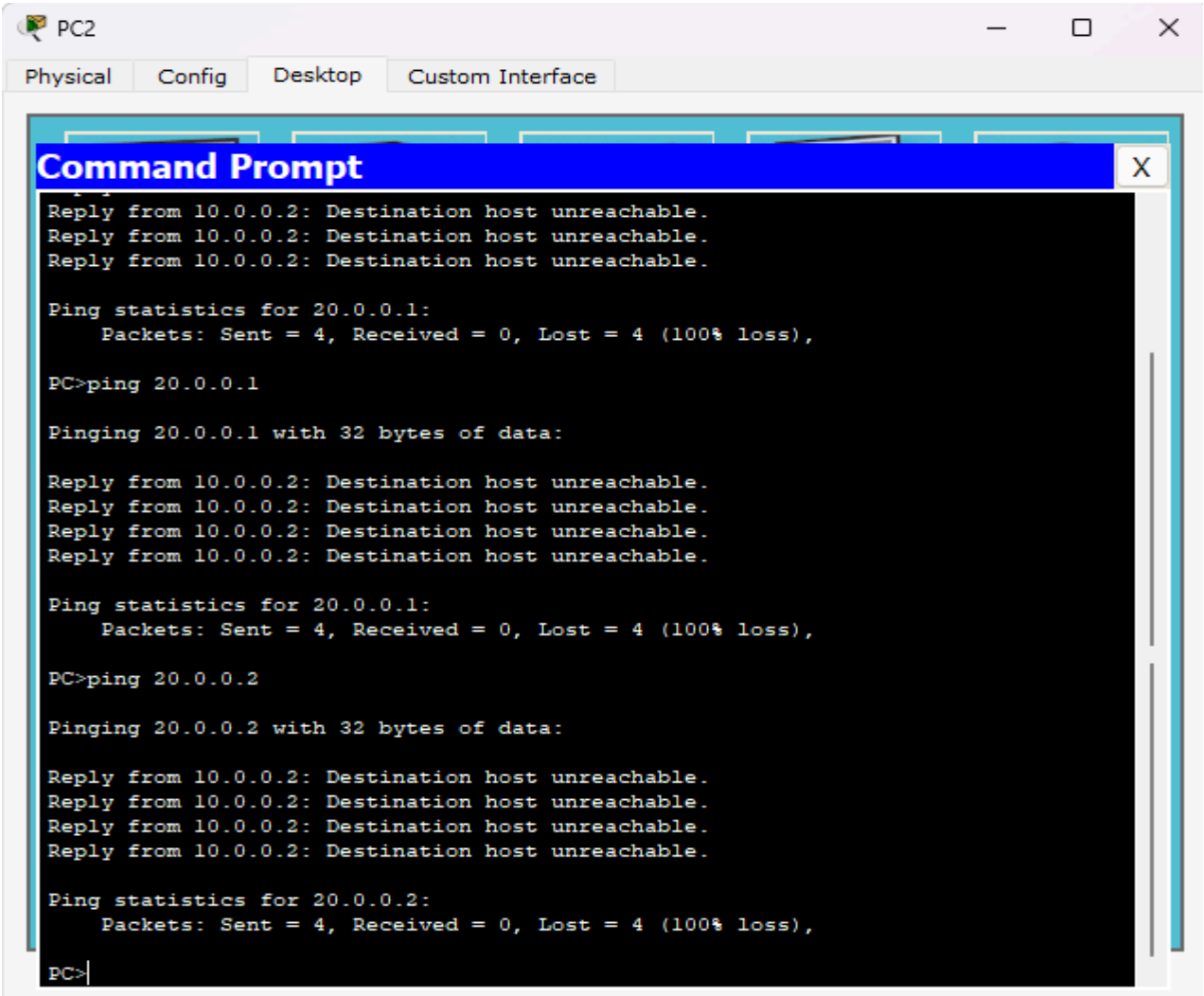
1) All PC's and routers are successfully connected

2) Ping 20.0.0.10 & Ping 30.0.0.2 are unsuccessful and show destination host unreachable. This happens because they are not neighbour network with 10.0.0.1.

3) Ping 30.0.0.1 will be successful.

**Screen Shots:**



**Program 4**

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

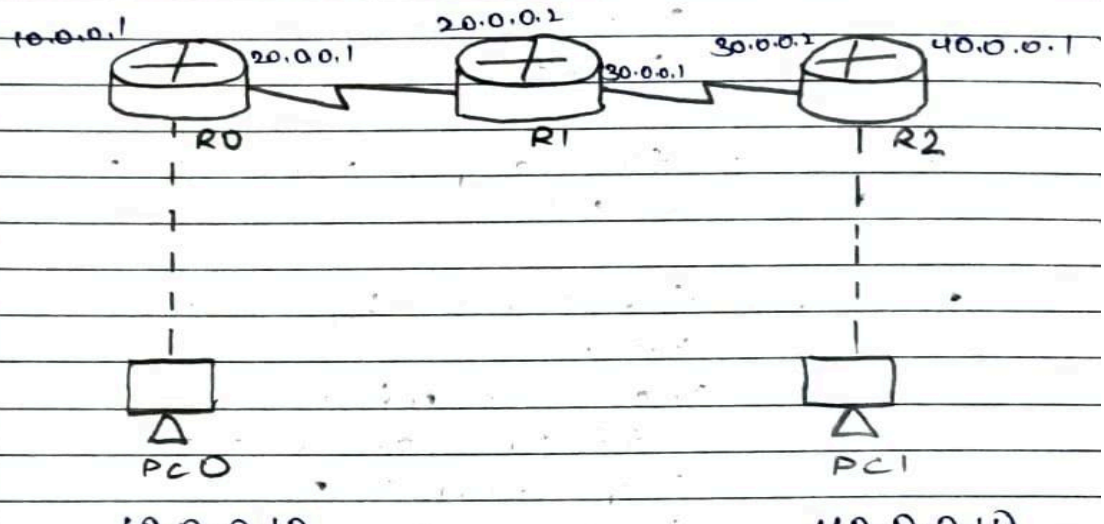
**Topology , Procedure and Observation:**

PAGE NO :
DATE : 23-10-24

LAB-03

- Objective: Configure one static routing and two default routings using three routers and two PC's.

- Topology:




- Procedure:

1) Connect two PC's with two different routers using copper cross over.

2) Connect both the routers using one more router using serial DCE.

3) Open config in the PC and configure the IP address and the gateway.

4) Do the same for the other PC.

PAGE NO
DATE

5) Open CLI in the router and configure a fastethernet connection by the following commands:

> enable
> config terminal
> interface fastethernet 0/0
> ip address 10.0.0.1 255.0.0.0
> no shutdown
> exit

6) Repeat the steps for the routers connected to other PC.

7) Connect the routers by the following commands:

> enable
> config terminal
> interface serial 2/0
> ip address 20.0.0.1 255.0.0.0
> no shutdown
> exit

8) Repeat the steps for the other routers

9) Now make the middle router as static router using following commands:

> enable
> 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

10) Make the other routers as Default routing

PAGE NO :
DATE :

by following commands:

> enable
> config terminal
> ip route 0.0.0.0 0.0.0.0 20.0.0.2

11) Repeat the steps for the routers connected to other PC.

- Observation:

1) All PC's and routers are connected successfully.

2) Default routing is set for routers connected to PC's (R0 & R2).

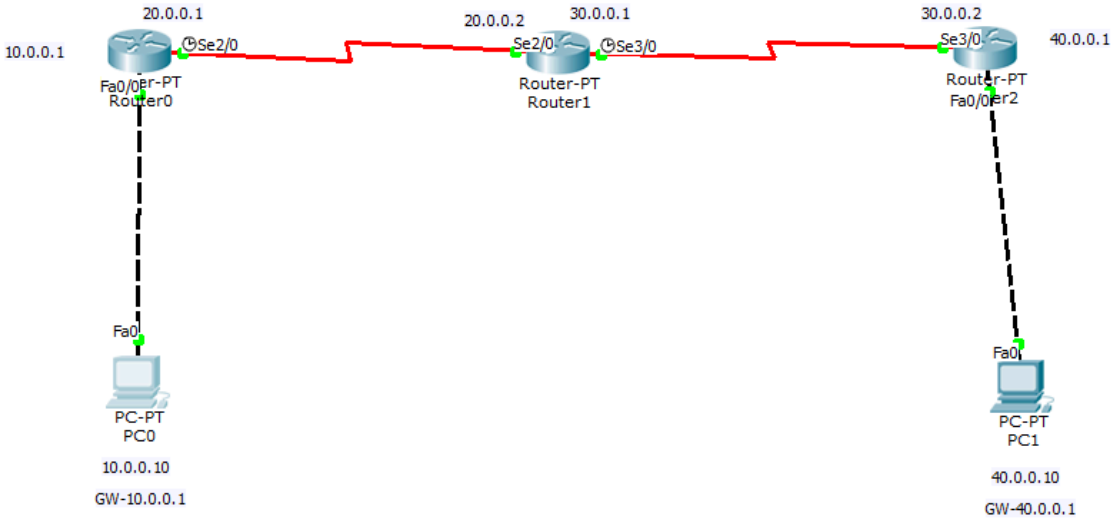
3) Static routing is set for R0.

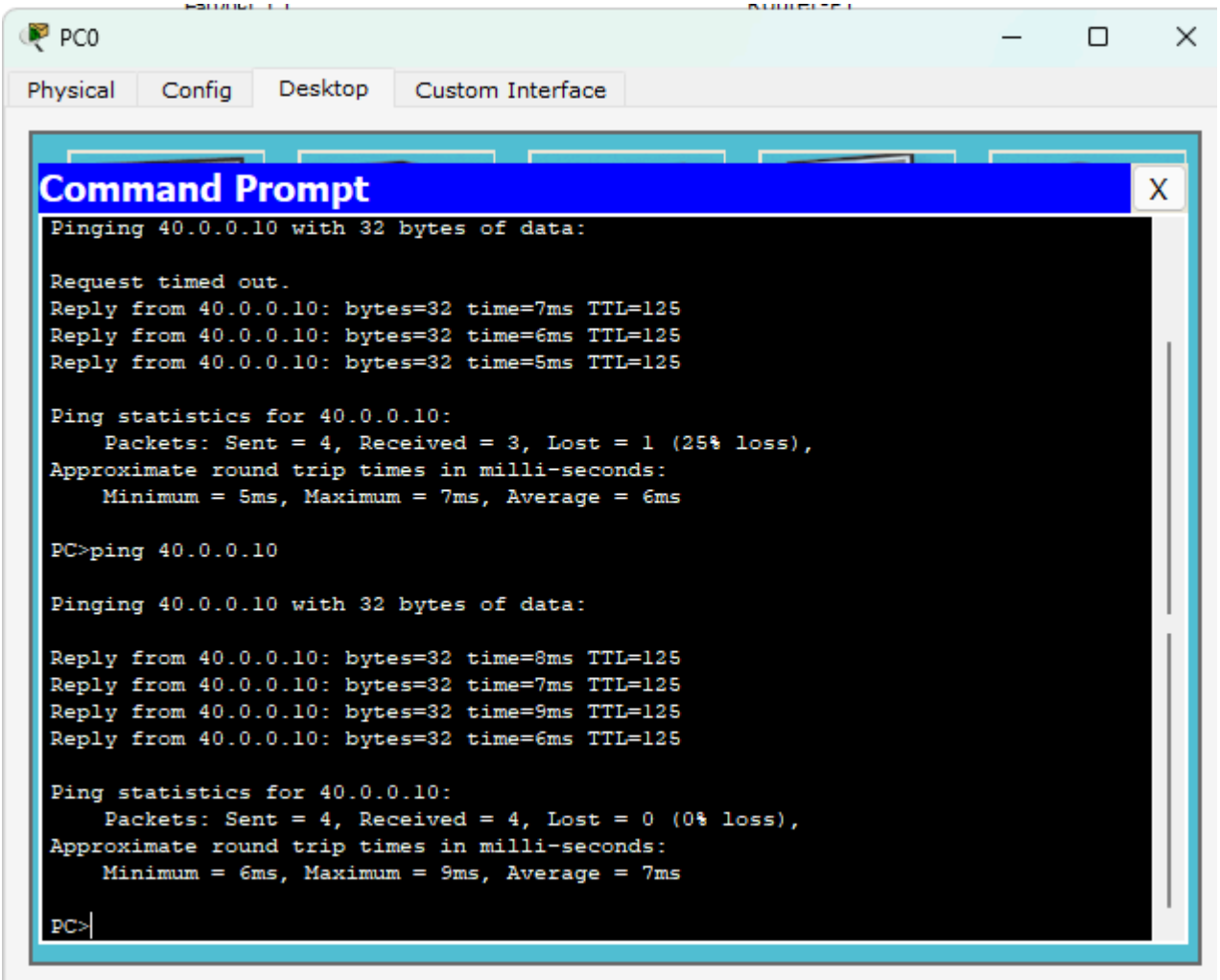
4) From PC0, ping 40.0.0.10 will be successful and all packets will be received.

23/10/24.

10/10

**Screen Shots:**



**Program 5**

**Aim:** Configure DHCP within a LAN and outside LAN.

**Topology , Procedure and Observation:**

PAGE NO :
DATE : 13 - 11 - 24

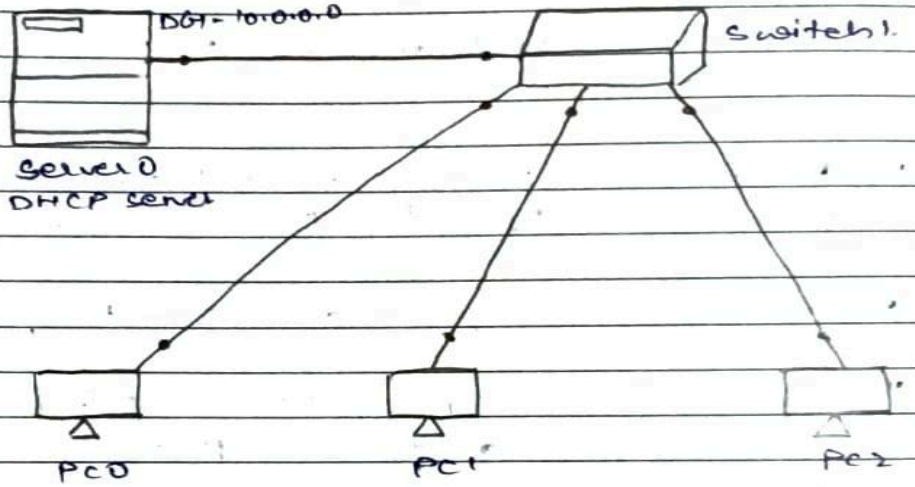
LAB-04

# Design a DHCP within LAN and Outside LAN

- Objective : To design a DHCP within LAN ~~and outside LAN~~.

- Topology :

i) within LAN




- Procedure :

1) Place 3 PC's 1 server and one switch and connect all end devices to the switch using copper straight wire.

2) Go to server0 → Desktop → IP configuration
IP address - 10.0.0.1
Default Gateway : 10.0.0.0

3) In server0 go to config → services → DHCP turn services to on

PAGE NO :
DATE :

pool name : switchone
Default gateway : 10.0.0.0
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 addresses will be assigned automatically.

5. Ping from PC0 to PC3.

- Observation:

1) All connections are successful

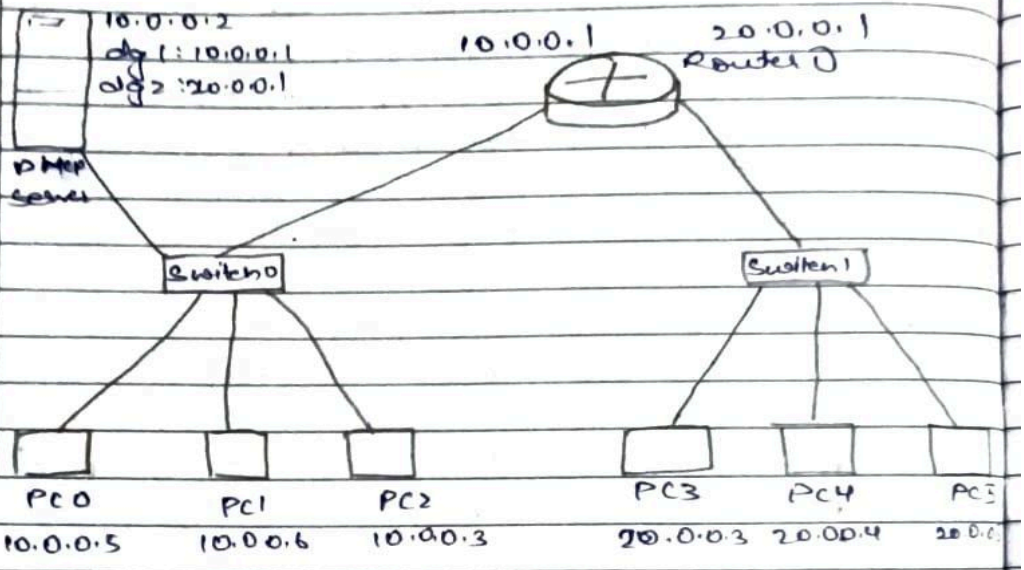
2) Ping 10.0.0.4

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

PAGE NO :
DATE :

- Objective : To design a DHCP Outside LAN
- Topology :

Outside LAN




- 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
Def. gateway - 10.0.0.1

3) Config → Services → DHCP, turn services to on
Pool name : switch one
Def. gateway : 10.0.0.1
start IP : 10.0.0.3

PAGE NO :
DATE :

max users : 100
click on Add.

pool name : switchtwo
Def. gateway : 20.0.0.1
start IP : 20.0.0.3
max users : 100
click on Add

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

# 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 switch connections go up

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

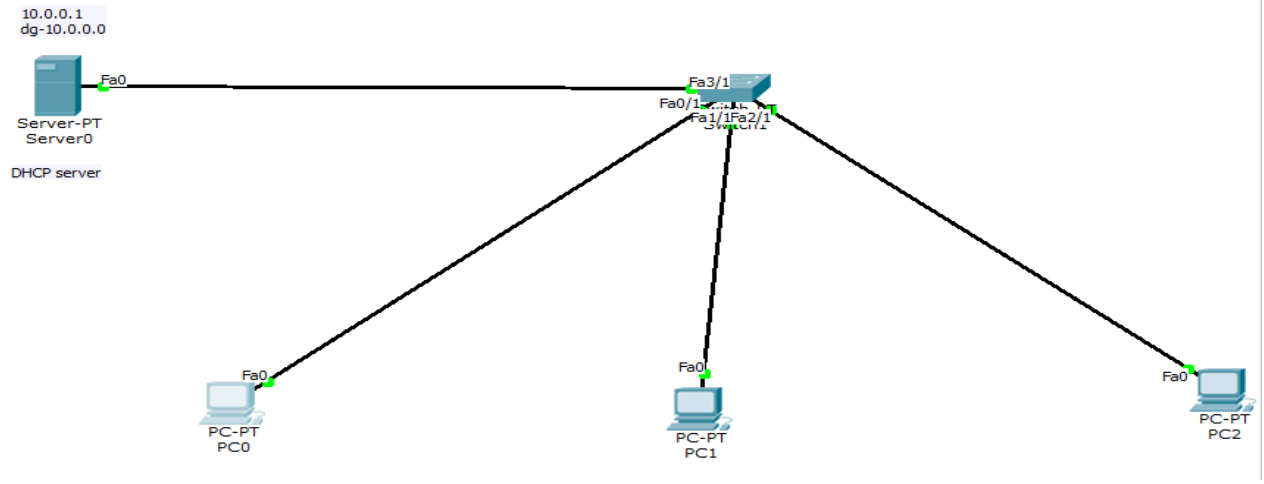
6) Ping PC0 to PC5

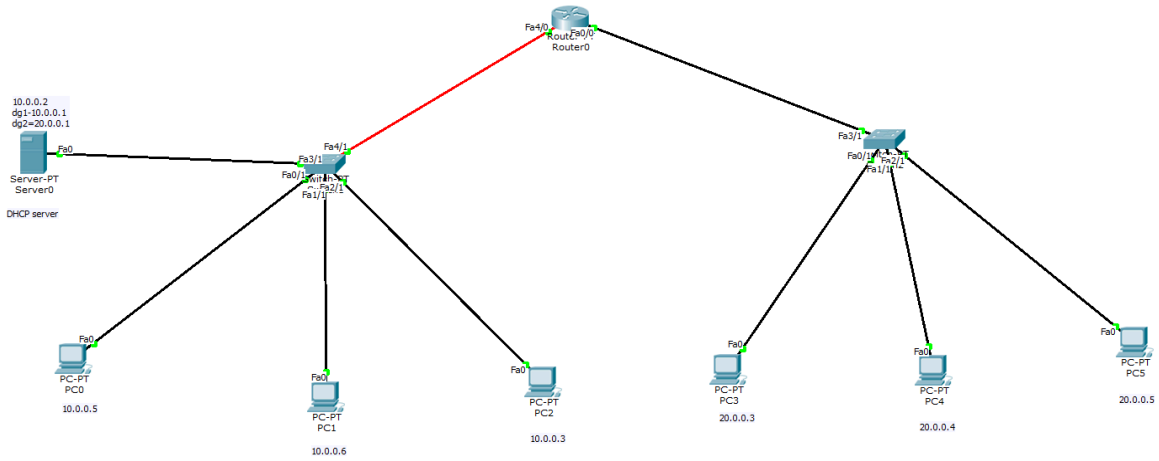
13/11/24

PAGE NO :
DATE :

- Observation
1) All connections are successfull.
2) All the PC's get assigned DHCP ip address.

**Screen Shots:**






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**

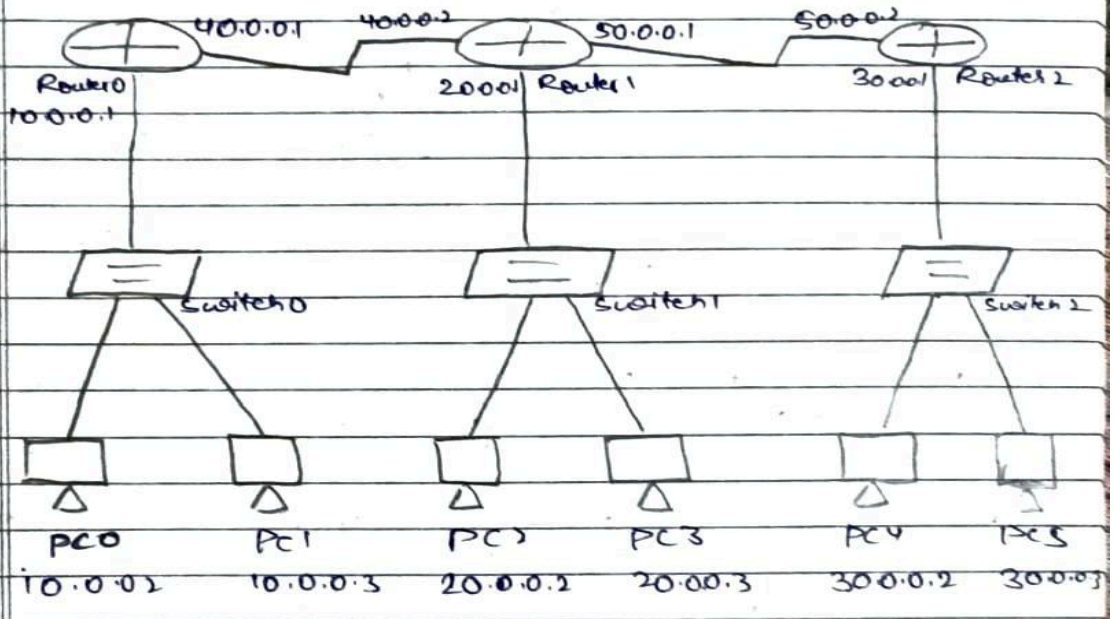
**Aim:**Configure RIP routing Protocol in Routers .

**Topology , Procedure and Observation:**

PAGE NO :
DATE : 20-11-24

LAB-05

- objective : Configure Routing Information Protocol in Router.
- Topology :




- Procedure:

i) Place 6 PC's, 3 routers and 3 switches

ii) Connect each pair of PC's to each switch and each switch to a router.

iii) Connect all the routers together.

iv) Set the ip addresses of the end systems along with gateway.

PAGE NO :
DATE :

v) Configure routers with ip addresses of fastethernet and serial ports.

vi) For each router perform following after going to CLI:
> enable
> config terminal
> router rip
> network 10.0.0.0 (for router 1)
> network 40.0.0.0 (for router 1 & 3)
> network 50.0.0.0 (for router 2 & 3)
> network 20.0.0.0 (for router 2)
> network 30.0.0.0 (for router 3)

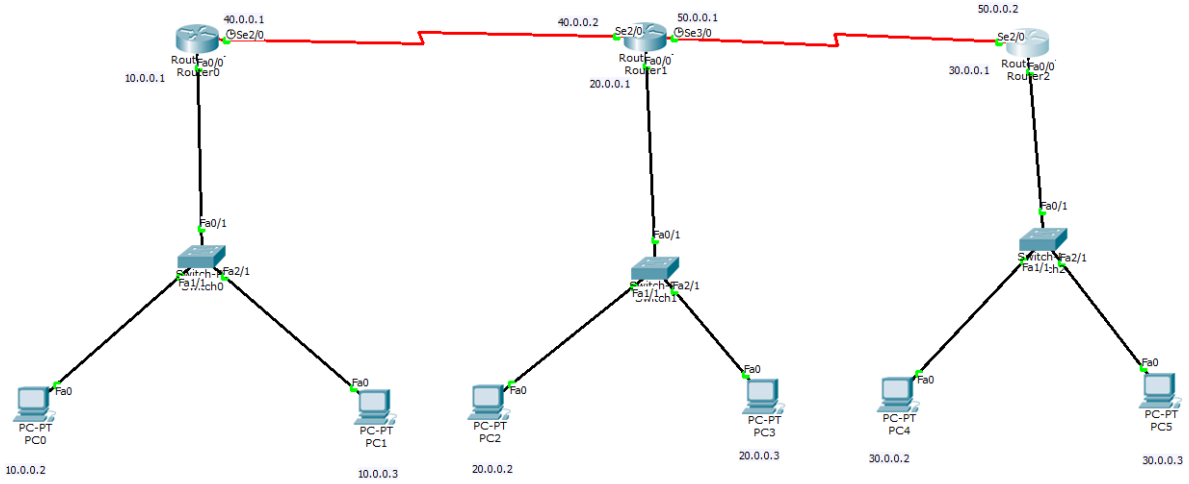
vii) For each router check the routes by
> show ip route.

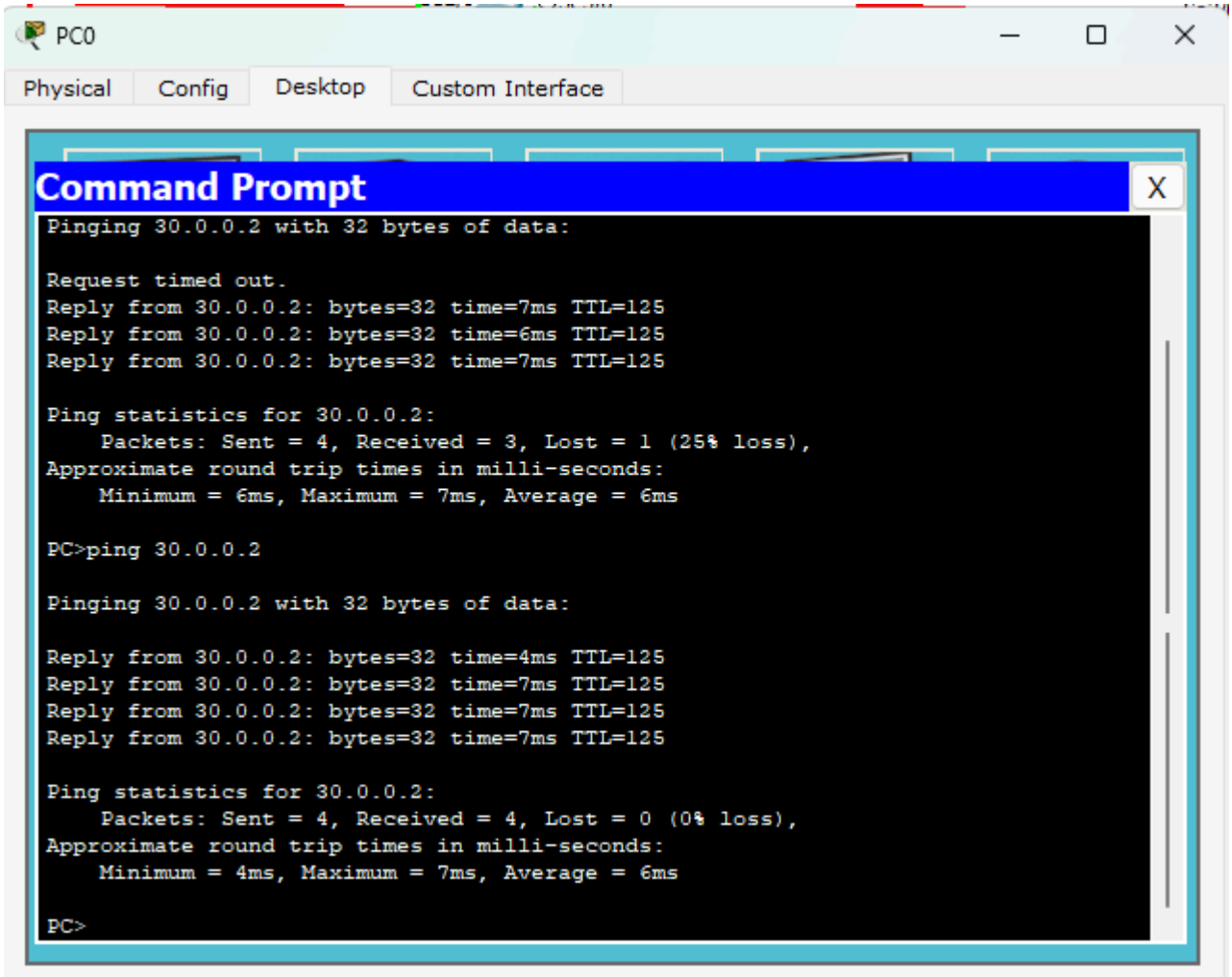
viii) Now Ping from one PC of a router to another PC of another router.
ie For PC0 (10.0.0.2) : Ping 30.0.0.2

- Observation:

1. All device connected successfully.
2. Ping 30.0.0.2 from PC0
ping statistics for 30.0.0.2
Packets: Sent=4, Received=4, Lost=0 (0% Loss)

**Screen Shots:**

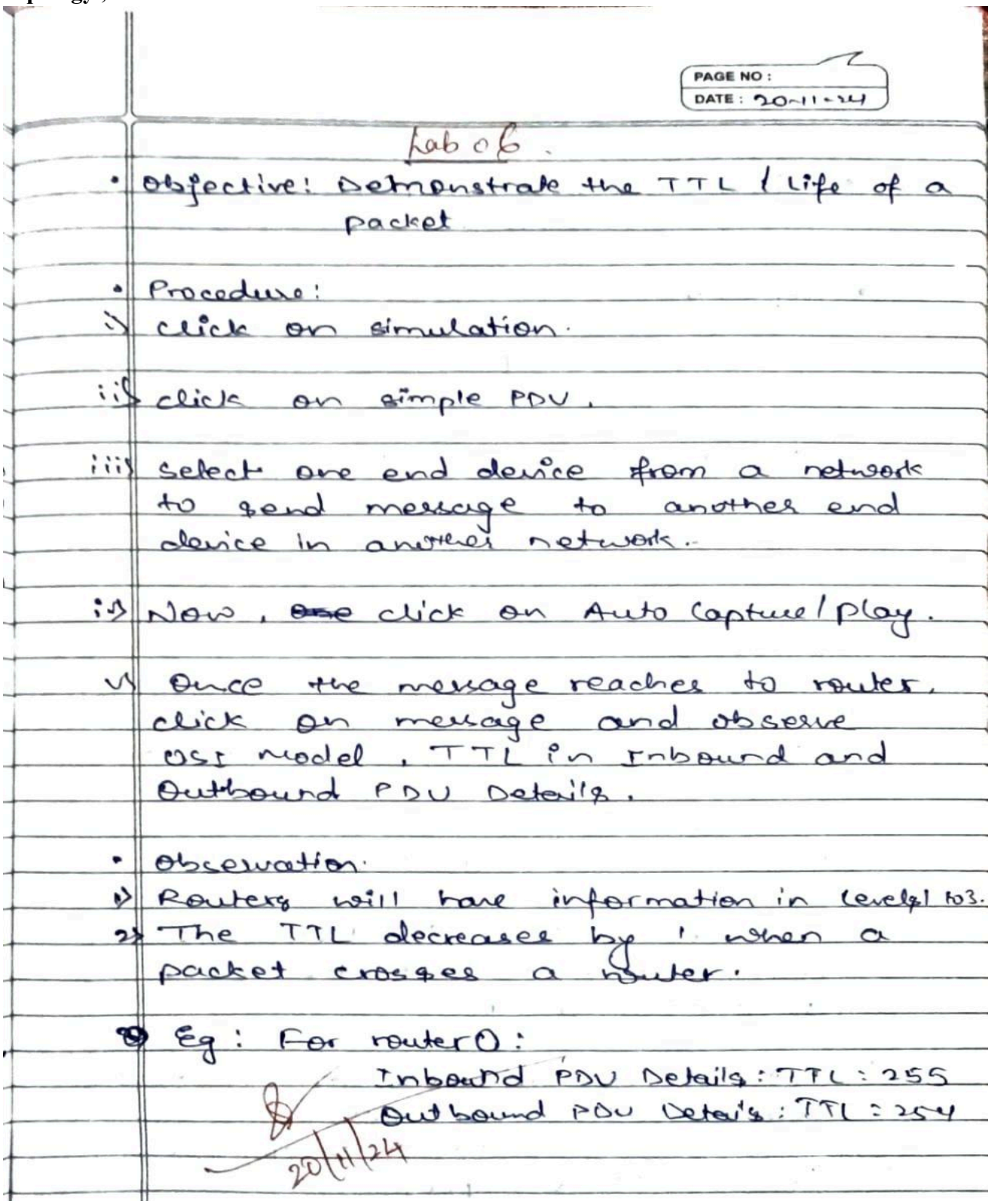


**Program 7**

**Aim:**Demonstrate the TTL/ Life of a Packet .

**Topology , Procedure and Observation:**



PAGE NO :
DATE : 20-11-24

Lab 06.

- Objective: Demonstrate the TTL / Life of a packet

- Procedure:

i) click on simulation.

ii) click on simple PDU.

iii) select one end device from a network to send message to another end device in another network.

iv) Now, click on Auto Capture/play.

v) Once the message reaches to router, click on message and observe OSI model, TTL in Inbound and Outbound PDU Details.

- Observation:

1) Routers will have information in Level 1 to 3.

2) The TTL decreases by 1 when a packet crosses a router.

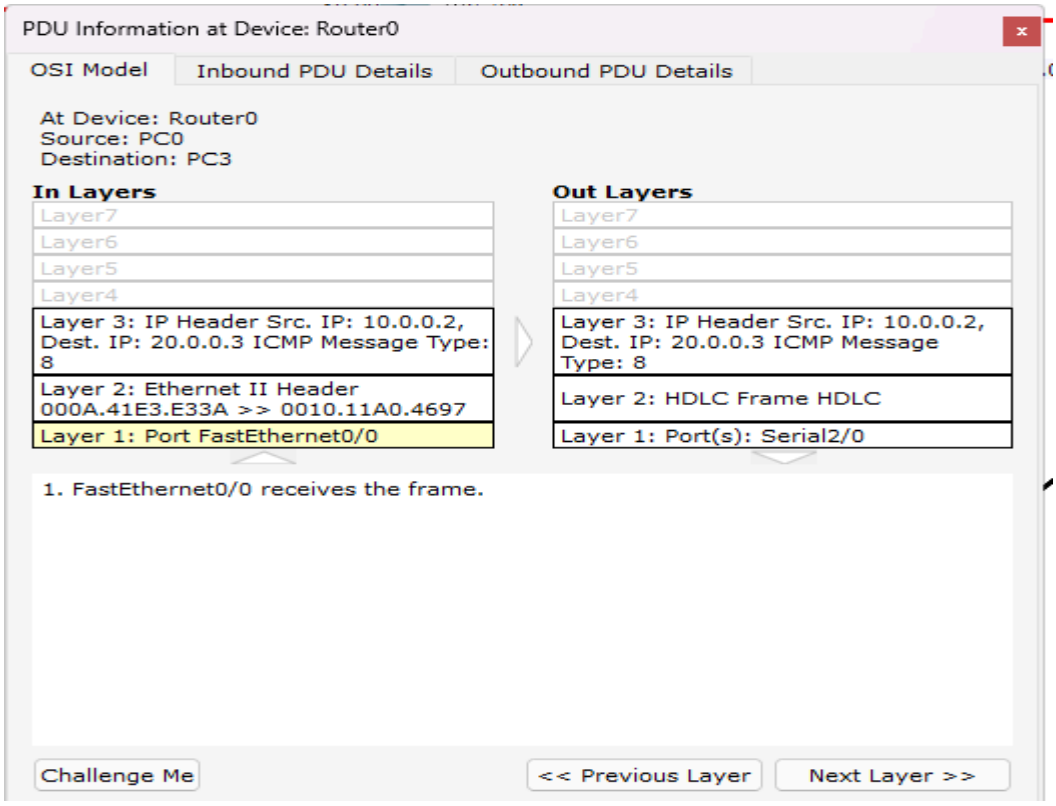
Eg: For router 0:

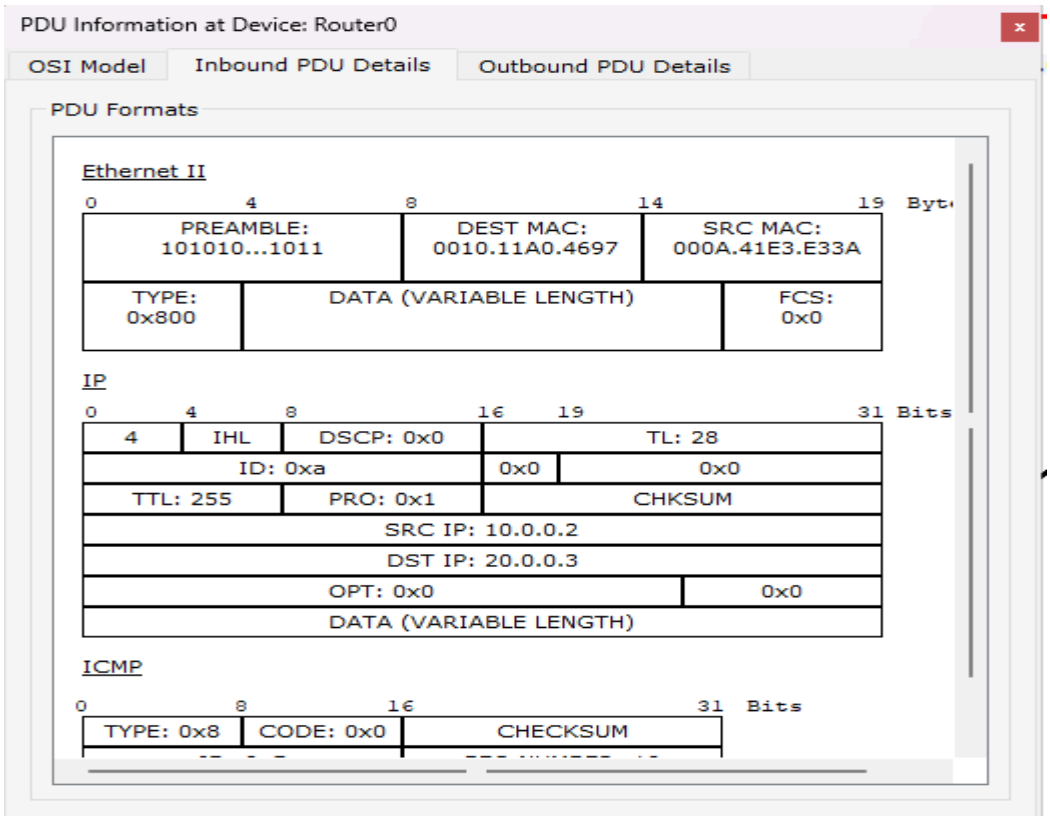
Inbound PDU Details: TTL: 255

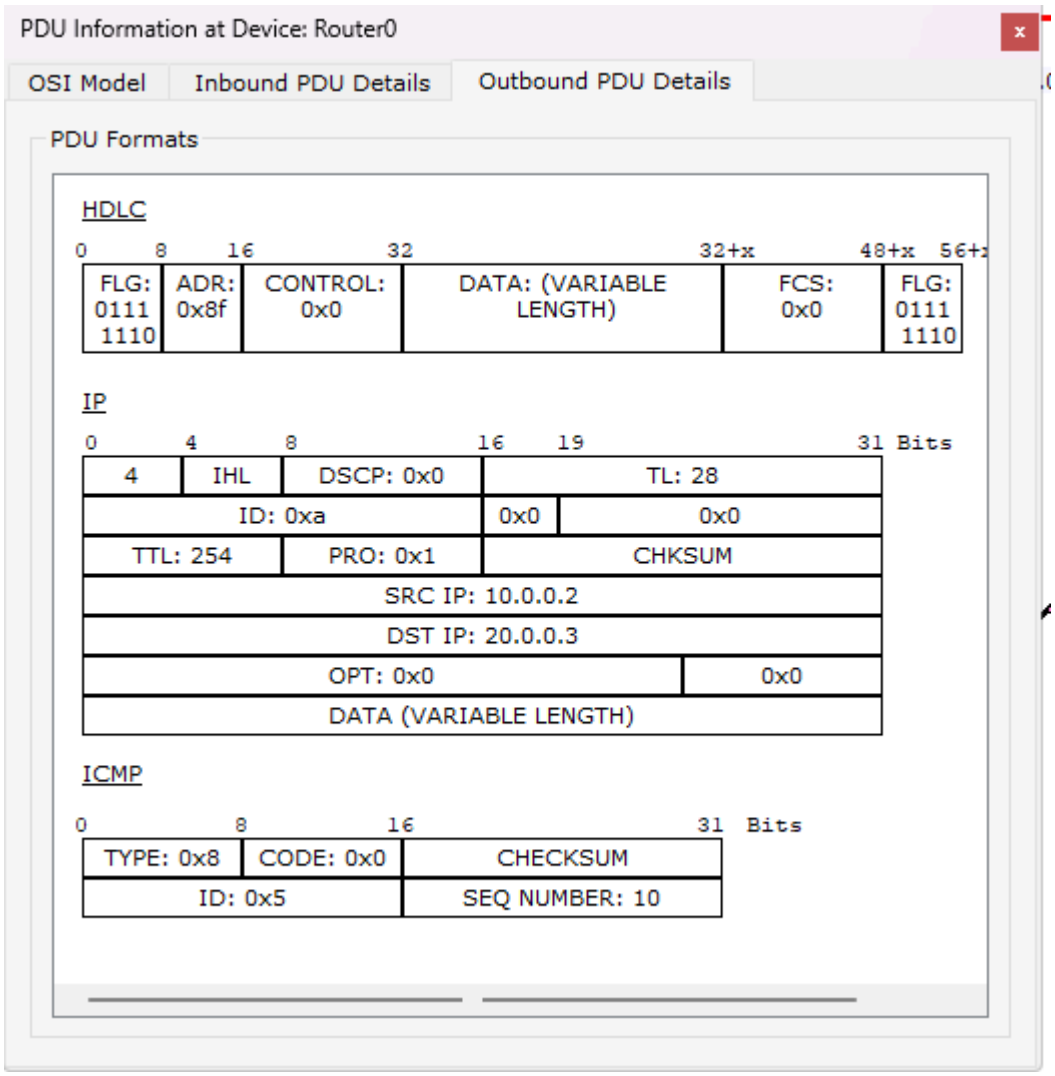
Outbound PDU Details: TTL: 254

20/11/24

**Screen Shots:**



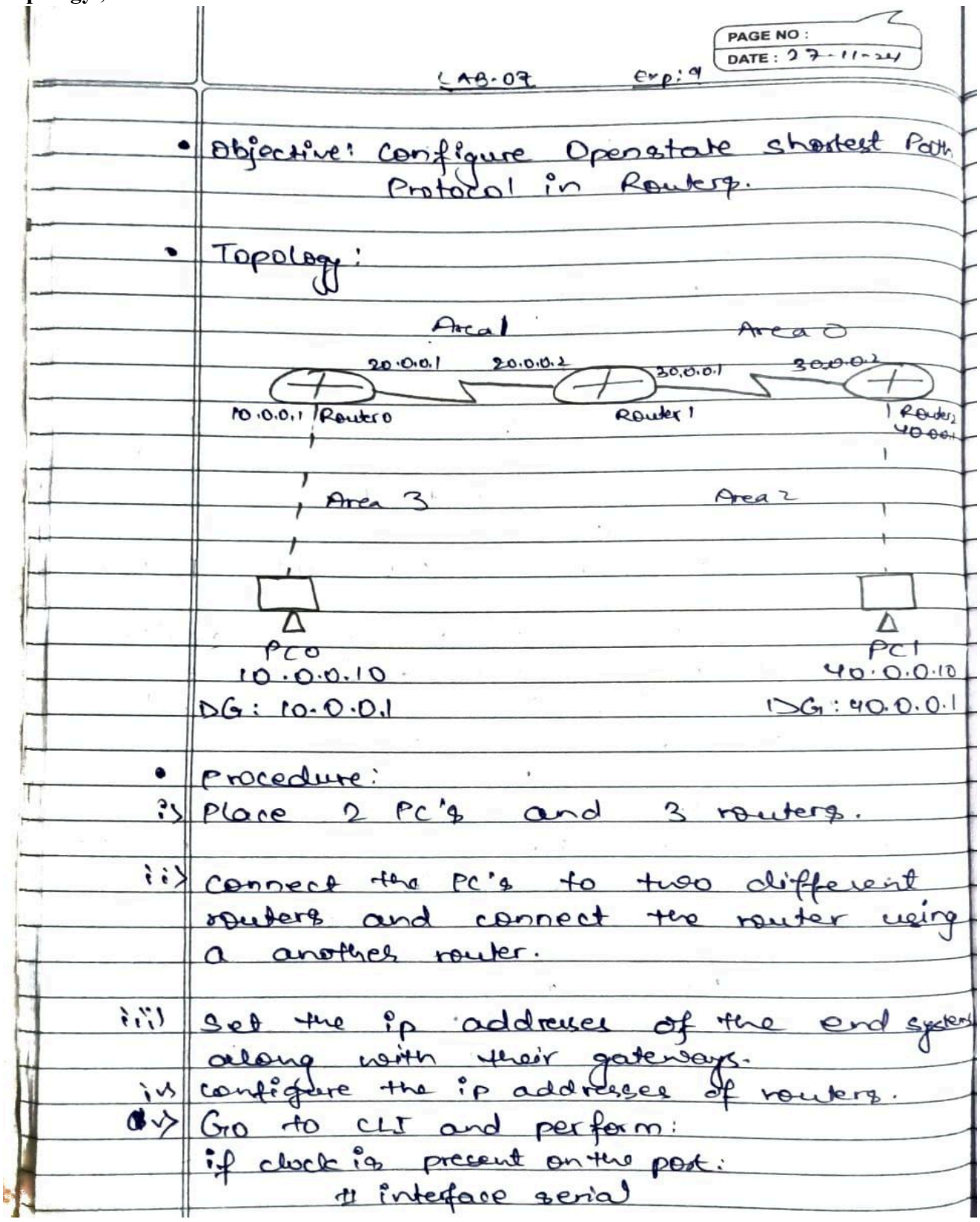




PDU Information at Device: Router0
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
0 8 16 32 32+x 48+x 56+x
FLG: 0111 1110
ADR: 0x8f
CONTROL: 0x0
DATA: (VARIABLE LENGTH)
FCS: 0x0
FLG: 0111 1110
IP
0 4 8 16 19 31 Bits
4
IHL
DSCP: 0x0
TL: 28
ID: 0xa
0x0
0x0
TTL: 254
PRO: 0x1
CHKSUM
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 :

(config-if) # encapsulation PPP
(config-if) # clock rate 64000
(config-if) # no shut

if clock is not present
(config-if) # encapsulation PPP
(config-if) # no shut

vi) Giving id's to Routers :-
Router 0 :
(config) # 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 :
# 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 :
# 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

vii) Giving Loopback :
Router 0 :
(config-if) # interface loopback 0
# ip add 170.16.1.252 255.255.0.0
# no shut

PAGE NO :
DATE :

Router 1 :
#! interface loopback 0
#! ip add 176.16.1.253 255.255.0.0
#! no shut

Router 2: #! interface loopback 0
#! ip add 176.16.1.254 255.255.0.0
#! no shut

viii) Create virtual link between R1 and R2
In R1 :
(config)#! router ospf 1
#! area 1 virtual link 2.2.2.2
#! exit

In R2
(config)#! router ospf 1
#! area 1 virtual link 1.1.1.1
#! exit

ix) Ping PC0 to PC1

- Observation:

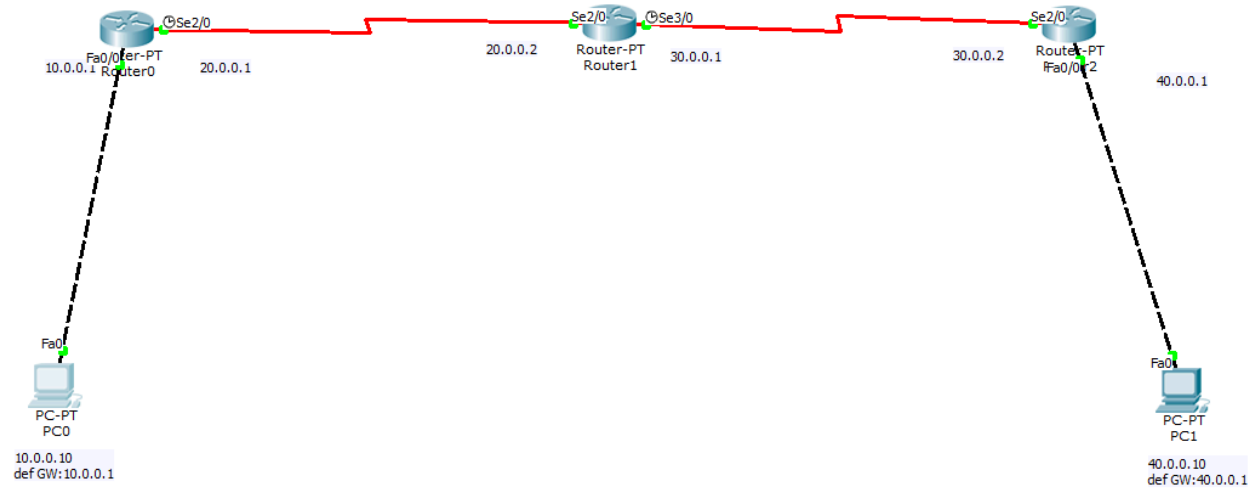
1) All the devices are connected successfully

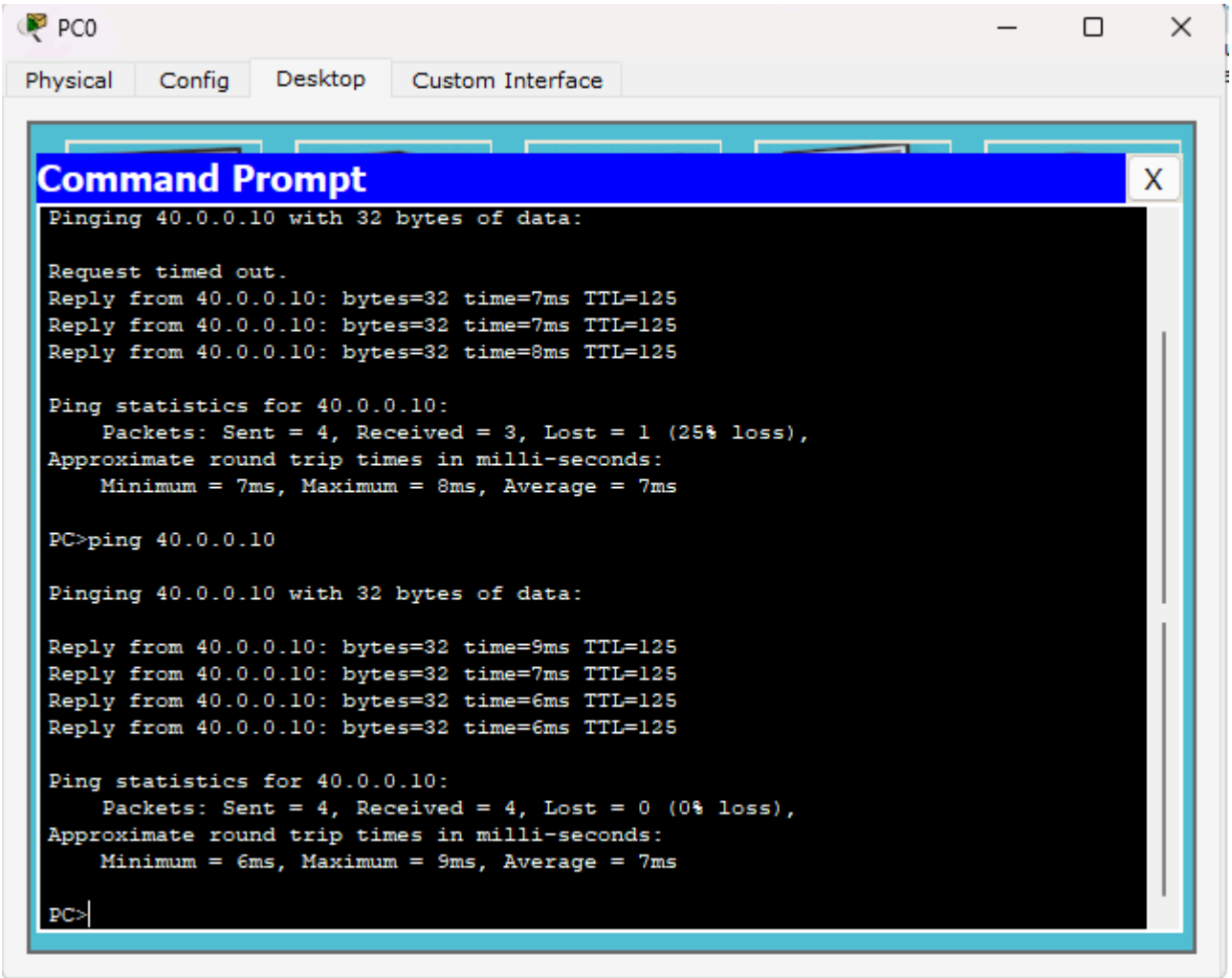
2) Ping 40.0.0.10 from PC0
ping statistics for 40.0.0.10

Packets: Sent:4, Received:4, Lost:0 (0%

24/4/24

**Screen Shots:**

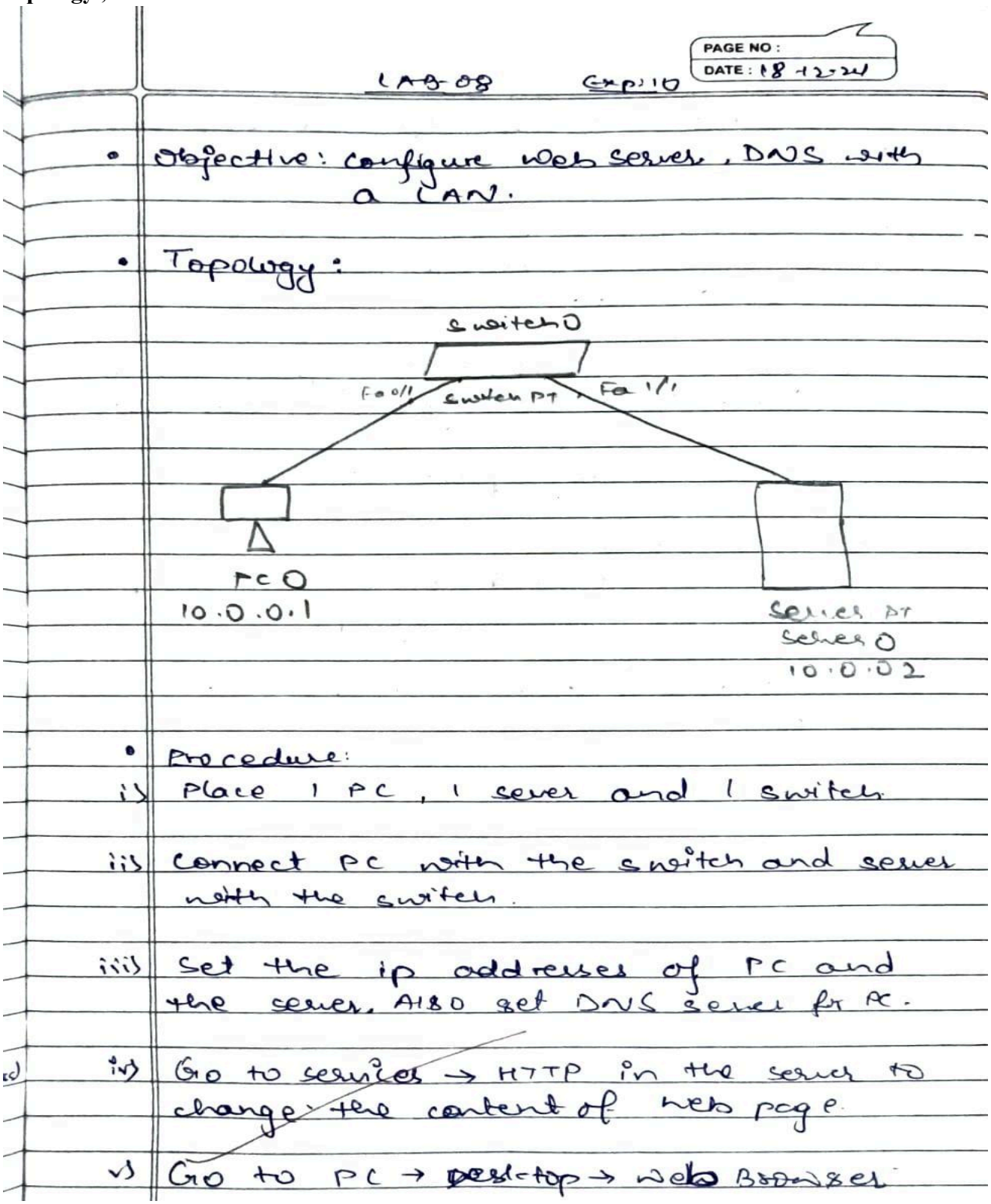


**Program 9**

**Aim:**Configure Web Server, DNS within a LAN.

**Topology , Procedure and Observation:**



PAGE NO :
DATE :

vi) In the URL type ip address of server to get the content.

vii) To configure, go to server → DNS and give, Name, type and address, click Add.

viii) Again go to PC → Desktop → web browser and in the URL type the address.

ix) Now the web page is visible.

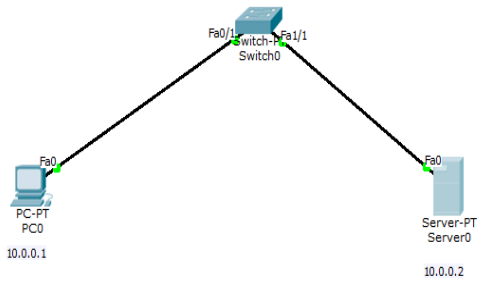
- observation:

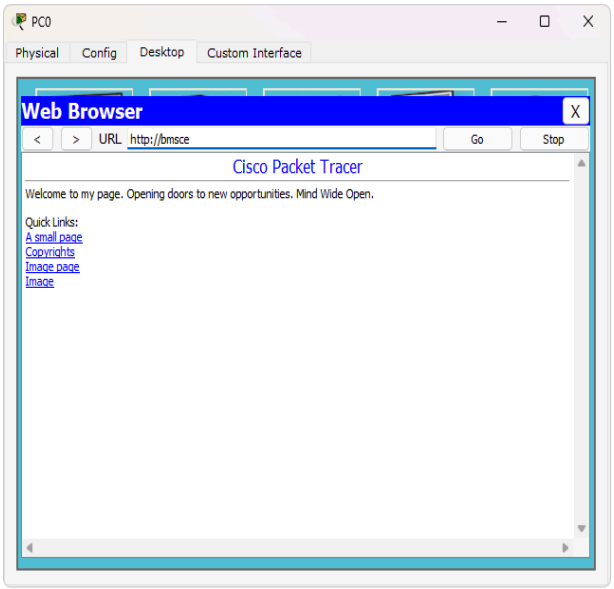
→ Successfully accessed the server web page from PC by entering ip address.

→ configured web, Server, DNS with a LAN.

26/12/24

**Screen Shots:**



**Program 10**

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

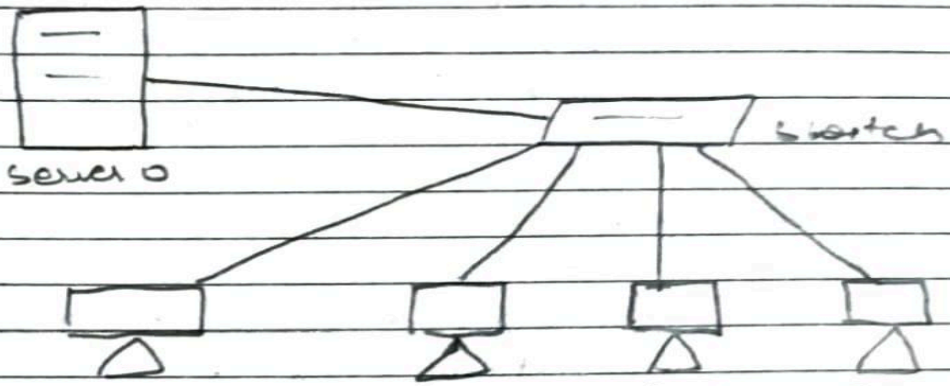
**Topology , Procedure and Observation:**

PAGE NO :
DATE : 18-12-24

LAB 09

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

- Topology :




- Procedure :

i) Create a topology of 4 PCs, a server and a switch.

ii) Assign IP addresses to all.

iii) Connect them through a switch.

iv) Use the inspect tool to click on a PC to see the ARP Table.
(command in CLI for same is arp -a)

PAGE NO :
DATE :

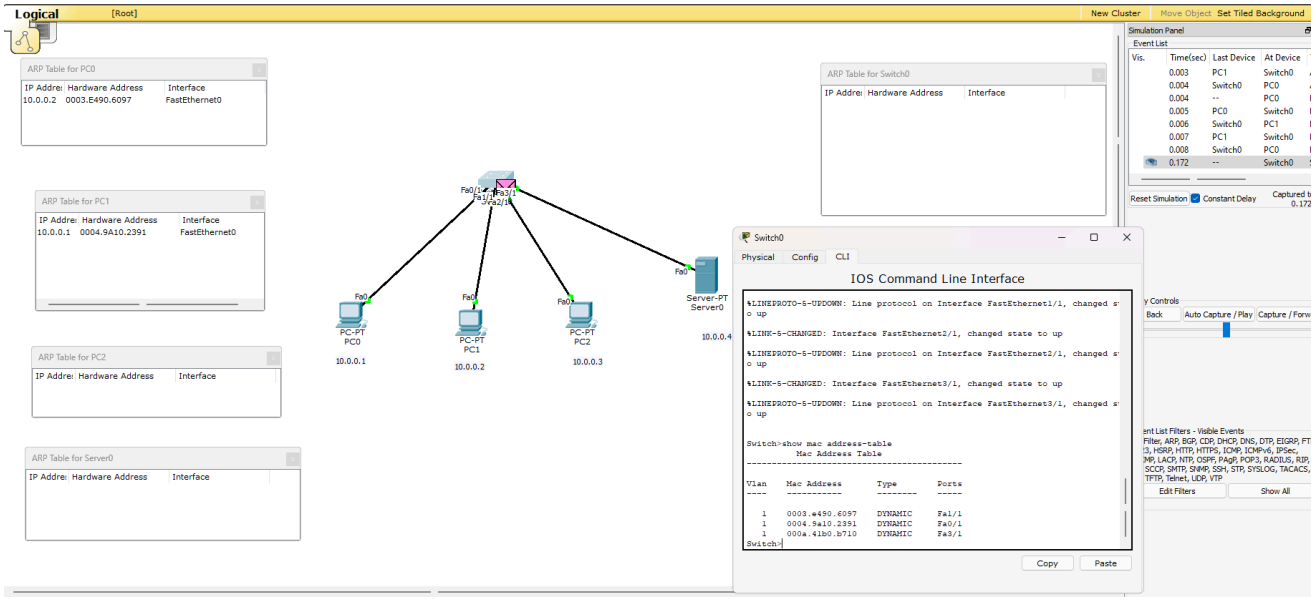
4) Initially ARP table is empty.
(Also in CLI of switch, the command show mac address-table can be given on every transaction to see how the switch learns from transactions and build the address-table.

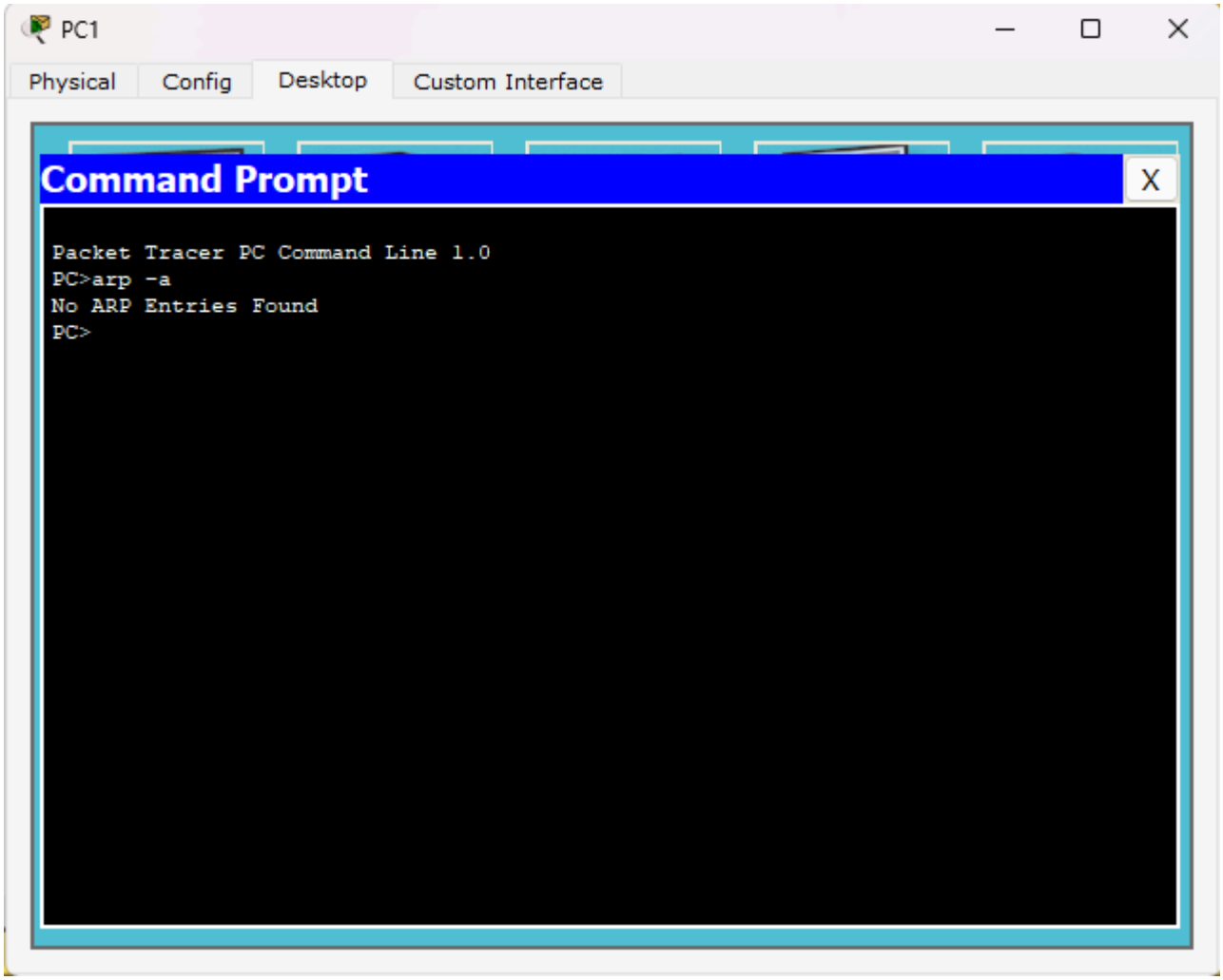
5) Use the capture in the simulation panel to go step by step so that the change in ARP can be clearly noted.

- Observations:
→ Switch as well the nodes update the ARP Table as and when a new communication starts.

26/12/24.

## 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 NO :
DATE : 18-12-24

LAB-10

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

- Topology :


10.0.0.2 10.0.0.1

- Procedure

i) Make a simple topology with one router and one PC.

ii) Commands in Router

```
> enable
# config terminal
# hostname R1
# enable secret P1
# interface fastethernet 0/0
# ip address 10.0.0.1 255.0.0.0
# no shut

# line vty 0 5
# login
# password p0
# exit
> exit
```

PAGE NO :
DATE :

wr → to save changes

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

- Password for user access verification is P0
- Password for enable is P1.

iv) Accessing router CLI from PC
> telnet 10.0.0.1

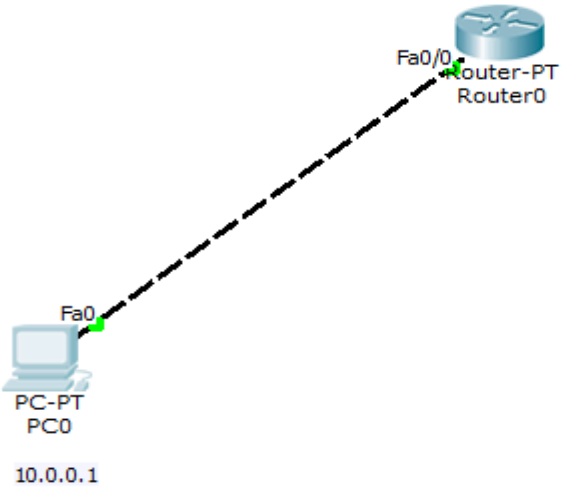
- User Access Verification
Password :
R1 > enable
password :
R1 # show ip route
codes
Area

- observation :

→ The admin in PC is able to run commands as run in router CLI and see the result from PC.

26/2/24

**Screen Shots:**



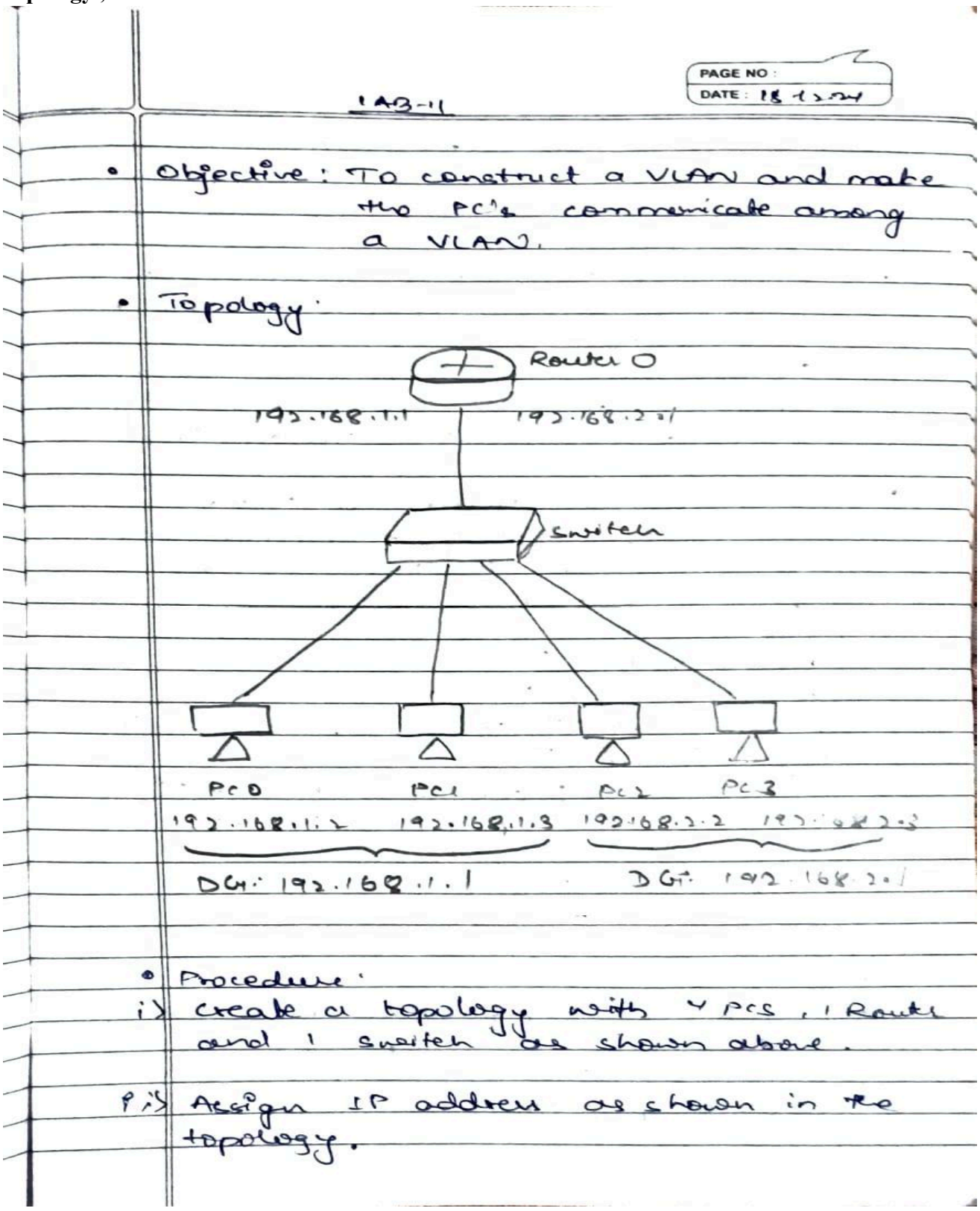

**Command Prompt**

```
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 :

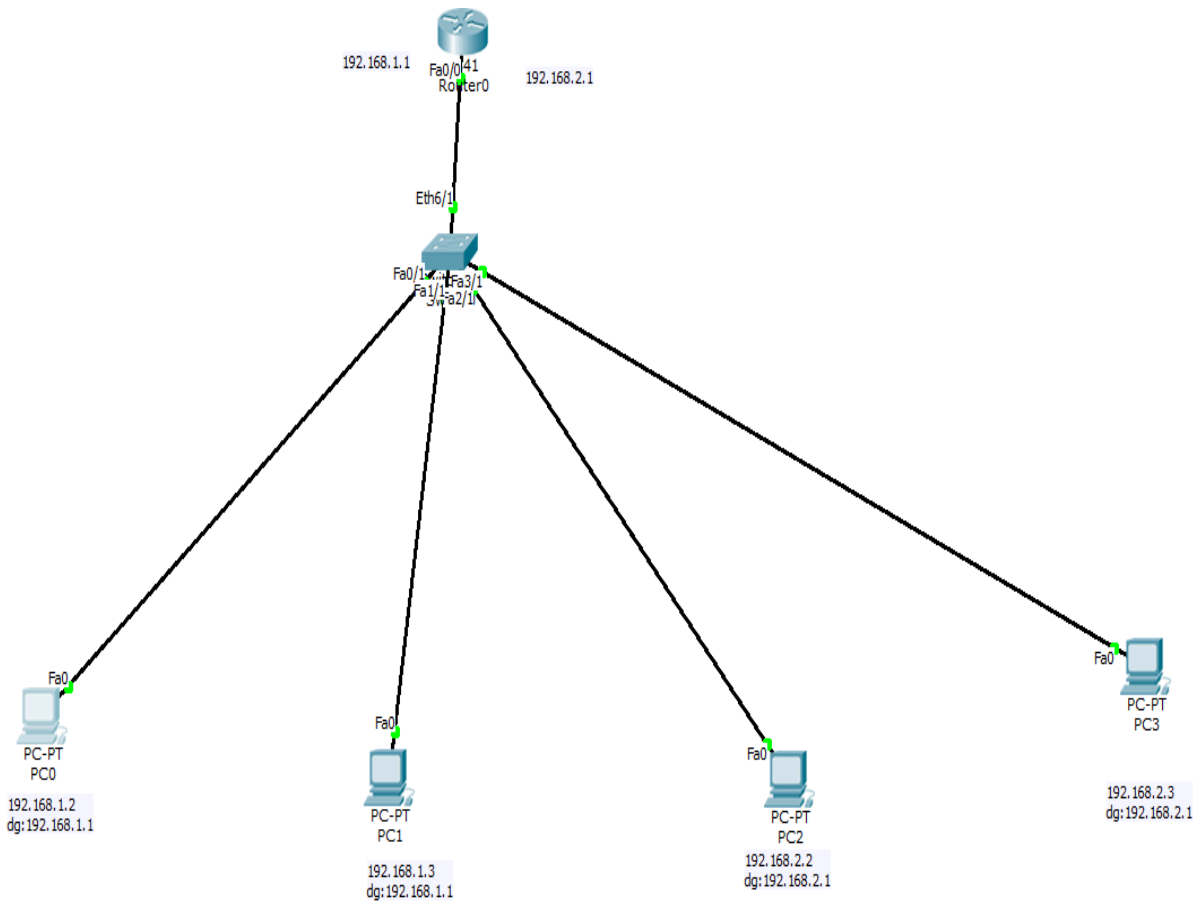
iii) Go to the switch → VLAN Database to configure the VLAN. Give VLAN number and VLAN name, add it.

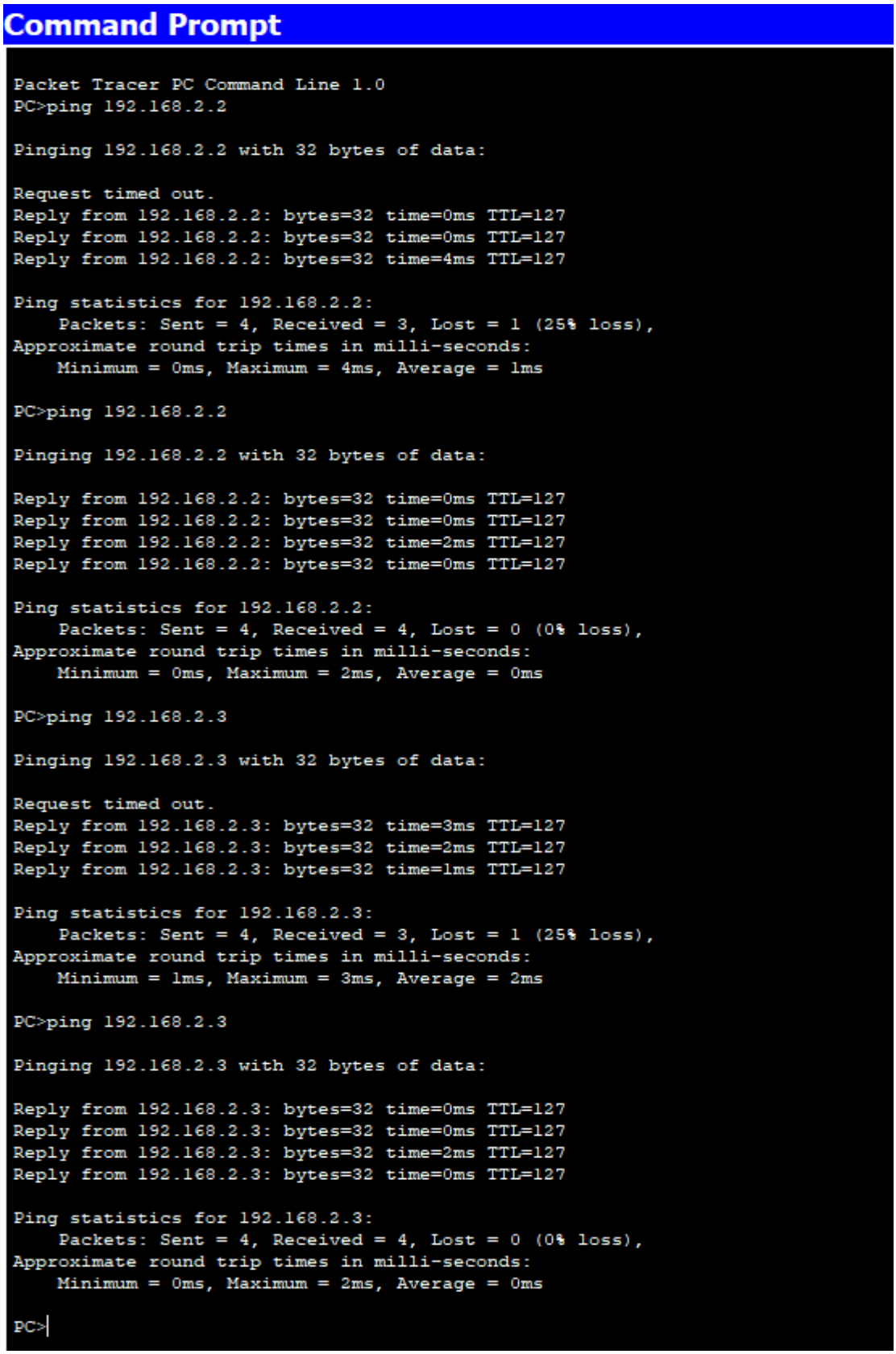
iv) Select the interface ie fastethernet 0/1 (near the switch from router) and make it trunk.
(VLAN trunking allows switches to forwards frames from different VLANs over a single link called trunk)

v) To make router understand VLAN.
Go to config tab of router, select VLAN database, enter the member and name of VLAN created.
Go to CLI
Router(vlan) #exit
Router # config t
Router(config)# interface fastether 0/0.1
Router(config)#
# encapsulation dot1q 2
# ip address 192.168.2.1 255.
# 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:**

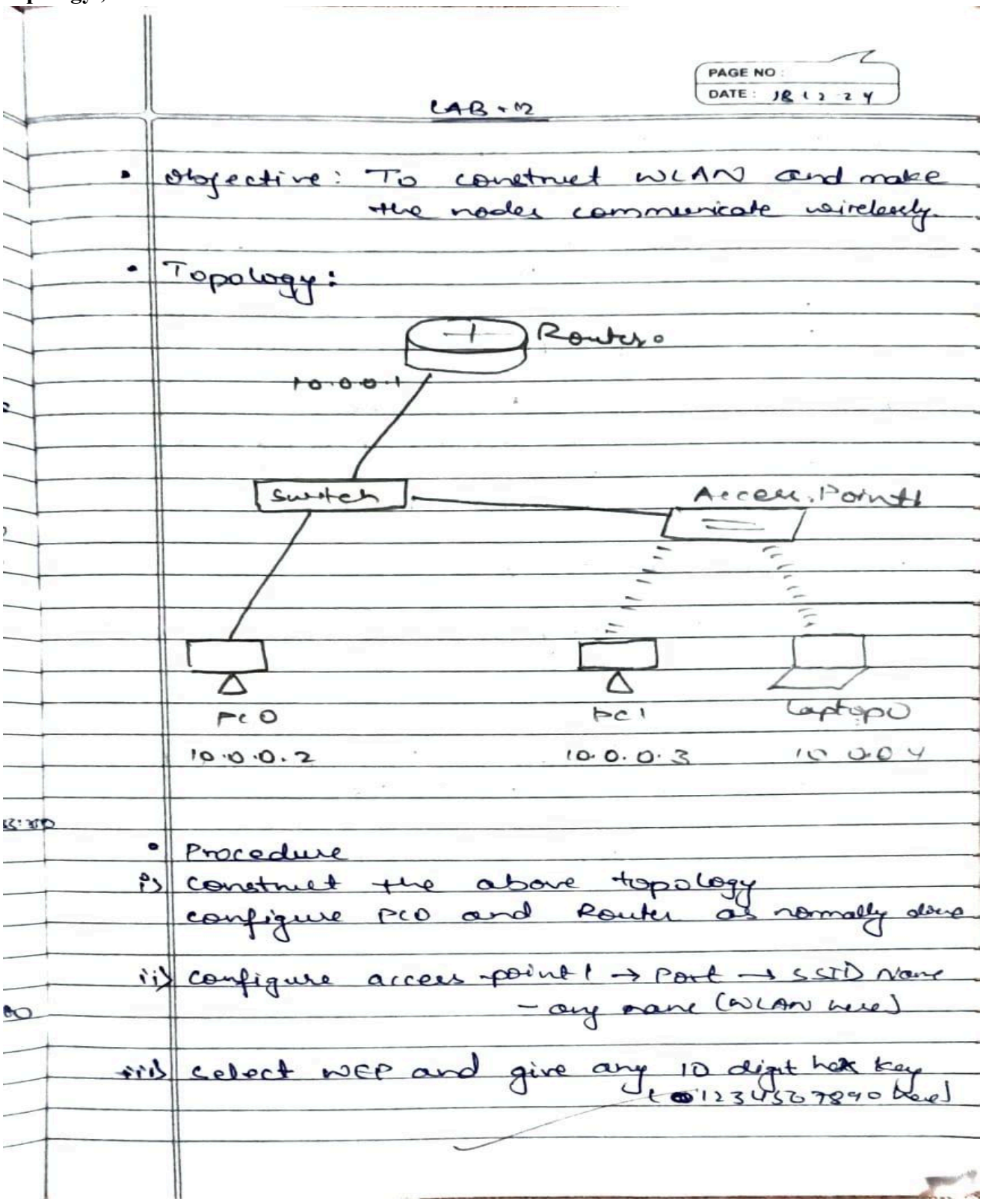



Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 192.168.2.2
Pinging 192.168.2.2 with 32 bytes of data:
Request timed out.
Reply from 192.168.2.2: bytes=32 time=0ms TTL=127
Reply from 192.168.2.2: bytes=32 time=0ms TTL=127
Reply from 192.168.2.2: bytes=32 time=4ms TTL=127
Ping statistics for 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

iv) Configure PC1 and laptop with wireless standards

v) In PC1
Switch off the device, drag the existing PT-HOST-NM-1AM to the component listed in LHS. Drag WMP300N wireless interface to the empty port. Switch On the device.

vi) 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.

vii) DO similar in laptop.

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

- Observation:

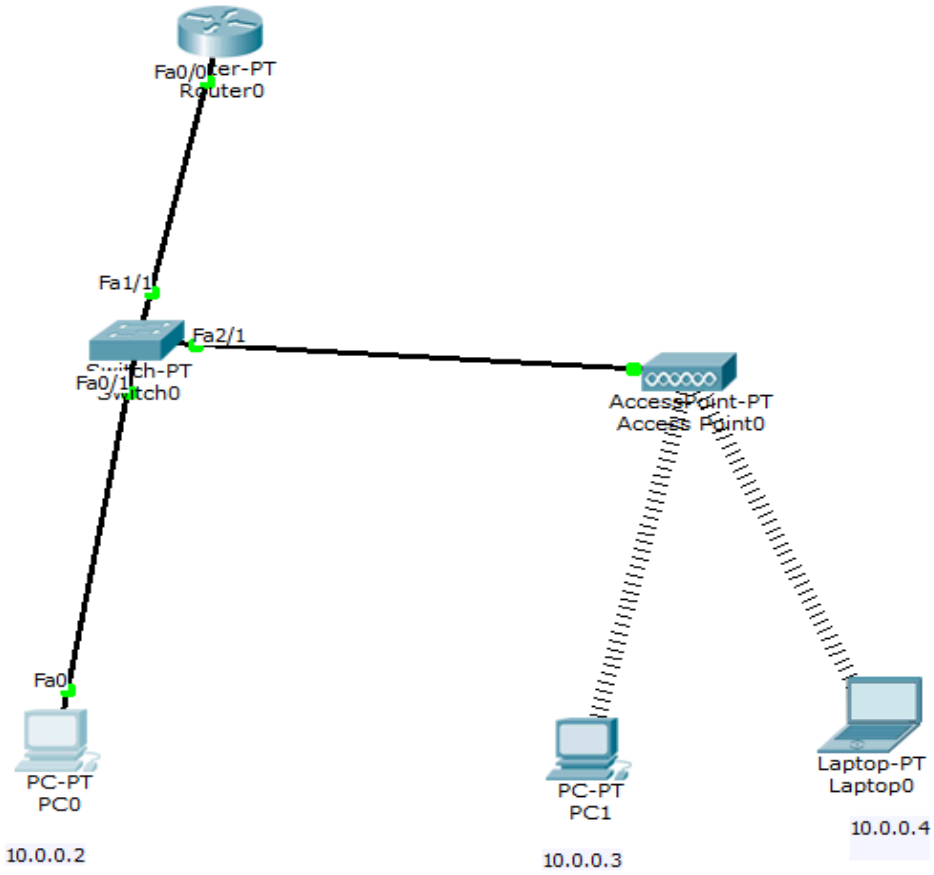
→ Device could connect to WLAN as long as they are in the network range.

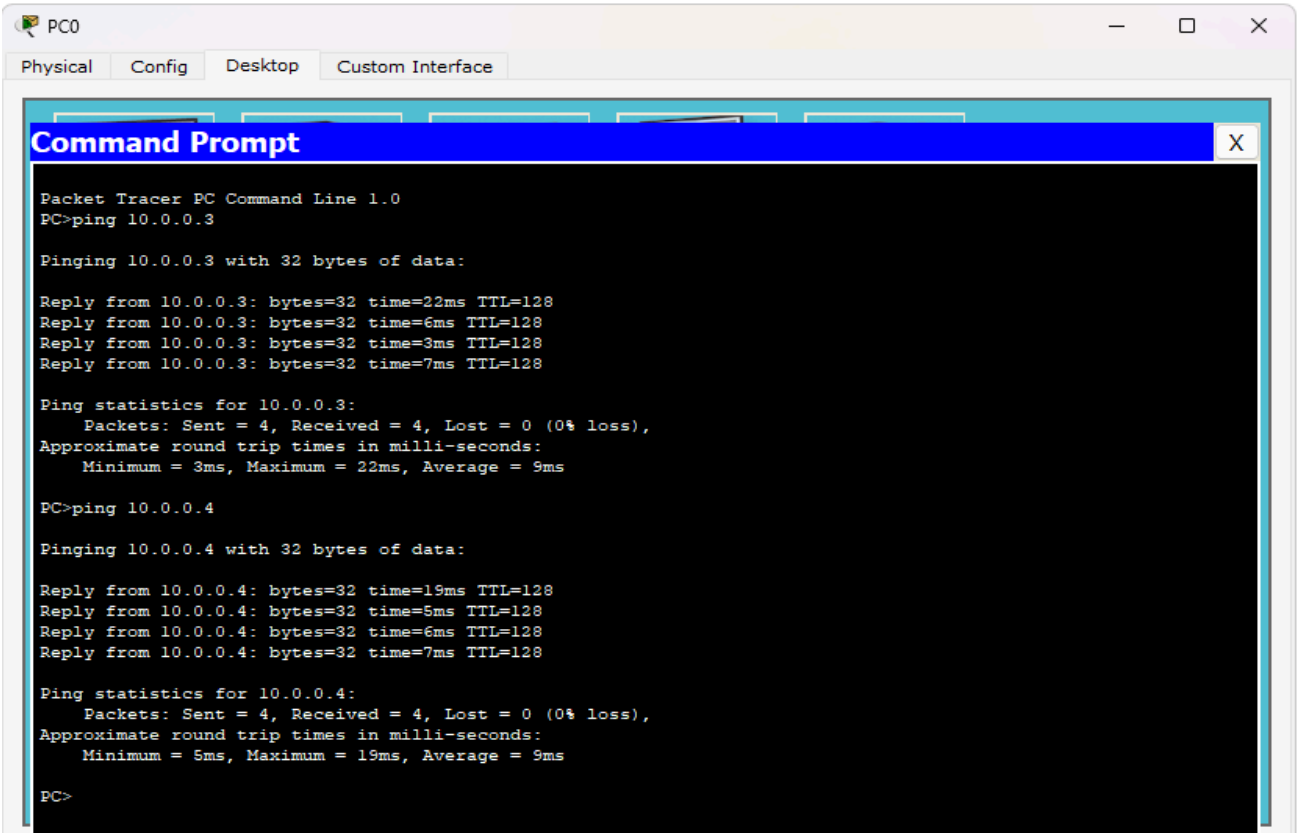
Bad Idea

signal strength decreases 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 : 22-12-24

Part B

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

```
#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");
        }
    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';
            }
        }
    }
    for(int i=0; i<strlen(op); i++)
        if(op[i]=='1')
            return 0;
    return 1;
}
```

PAGE NO :
DATE :

```
int main ()
{
    char ip[50], op[50], recv[50];
    char poly[] = "10001000000100001";

    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 bin" <<
                                        << 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

1111101

The transmitted message is : 11111011010111100111101

Enter the received message in binary

1111101

No error in data

**Program 15**

Write a program for congestion control using Leaky bucket algorithm.

**Code and Output:**

PAGE NO :
DATE : 23-11-24

2. write a program for congestion control using leaky bucket algorithm.

```
#include<iostream>
#include<string.h>
using namespace std;
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#define nof_packets 10
int rand(int a)
{
   int rn=(random()%10)%a;
   return rn==0?1:rn;
}

int main()
{
   int packet_sz[nof_packets], i, clk, b_size, o_rate,
       p_sz_rm=0, p_sz, p_time, op;

   for(i=0; i<nof_packets; ++i)
      packet_sz[i]=rand(6)*10;

   for(i=0; i<nof_packets; ++i)
      printf("\n packet[%d]: %d bytes\t", i,
             packet_sz[i]);
   printf("\n Enter the Output rate");
   scanf("%d", &o_rate);
   printf("Enter the Bucket Size");
   scanf("%d", &b_size);
```

PAGE NO :
DATE :

```
for (i=0; i<no_of_packets; ++i)
{
  if ((packet_sz[i] + p_sz_rm) > b_size)
    if (packet_sz[i] > b_size)
      printf("\n\nIncomming packet size
      (%ddbytes) is Greater than bucket
      capacity (%dbytes) - Packet Rejected",
      packet_sz[i], b_size);
  else
    printf("\n\nBucket capacity exceeded.
        Packets rejected!!");
else
 {
  p_sz_rm += packet_sz[i];
  printf("\n\nIncomming Packet size %d",
                packet_sz[i]);
  printf("\nBytes remaining to Transmit: %d",
                p_sz_rm);
  p_time = rand(4) * 10;
  printf("\nTime left for transmission: %d
        units", p_time);
  for (clk = 10; clk <= p_time; clk += 10)
   {
    sleep(1);
    if (p_sz_rm)
     {
      if (p_sz_rm <= o_rate)
        op = p_sz_rm, p_sz_rm = 0;
      else
        op = o_rate, p_sz_rm = p_sz_rm - o_rate;
      printf("\nPacket of size %d Transmitted",
                op);
```

```
                printf("...Bytes Remaining to Transmit:
                        %d", p->sz - rm);
            }
            else
            {
                printf("\nTime left for transission: %d
                        unit", p->time_clk);
                printf("\nNo 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 Bucket size: 50

Incoming Packet size 30
Bytes remaining to Transmit: 30
Time left for transmission: 20 units
Packet of size 30 Transmitted - Bytes Remaining
                                to Transmit: 0
Time left for transmission: 0 units
NO packets to tranemit!!
```

PAGE NO.
DATE

Incoming packet size: 10
Bytes remaing to Transmit: 10
Time left for trasmission: 30
Packet of size 10 Transmitted - Bytes remaing to Trasmt: 0
Time left for trasmission: 10 units
No packets to trasmit!!
Time left for trasmission: 0 units
No packets to trasmit!!

Incoming packet size: 10
Bytes remaing to Trasmt: 10
Time left for trasmission: 10 units
Packet of size 10 Transmitted - Bytes remaing to Trasmt: 0

Incoming Pacet size: 50
Bytes remaing to trasmt: 50
Time left for trasmission: 10 units
Packet of size 50 Trasmitted - Bytes remaing to Trasmit: 0

Incoming Packet size: 30
Bytes remaing to Trasmt: 30
Time left for trasmission: 30 units
Packet of size 30 trasmitted - Bytes remaing to transmit: 0
Time left for trasmission: 10 units
No packets to trasmt!!
Time left for trasmission: 0 units
No packets to trasmit!!

**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:**

PAGE NO :
DATE : 21-12-24

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

→ client side

```
#Include<stdio.h>
int main()
{
    int soc,n;
    char buffer[1024], fname[50];
    struct sockaddr_in addr;

    soc=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");

    while(connect(soc,(struct sockaddr*)&addr,
          sizeof(addr)));
    printf("\nClient is connected to server.");
    printf("\nEnter file name:");
    scanf("%s", fname);

    send(soc, fname, sizeof(fname), 0);
    printf("\nReceived response\n");

    while((n = recv(soc, buffer, sizeof(buffer),0))>0)
        printf("%s", buffer);
    return 0;
}
```

PAGE NO :
DATE :

» Server.c

```
#include<stdio.h>
#include<arpa/inet.h>
#include<fcntl.h>
#include<unistd.h>

int main()
{
    int welcome, new_soc, fd, n;
    char buffer[1024], fname[50];
    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_soc = accept(welcome, NULL, NULL);
    recv(new_soc, fname, 50, 0);
    printf("\n Requesting for file: %s\n", fname);
    fd = open(fname, O_RDONLY);

    if(fd < 0)
        send(new_soc, "\nFile not found\n",
                    15, 0);
```

```
else
    while((n=read(fd,buffer,sizeof(buffer)))
                                        >0)
        send(new_soc,buffer,n,0);

printf("\nRequest sent\n");
close(fd);

return 0;

}
```

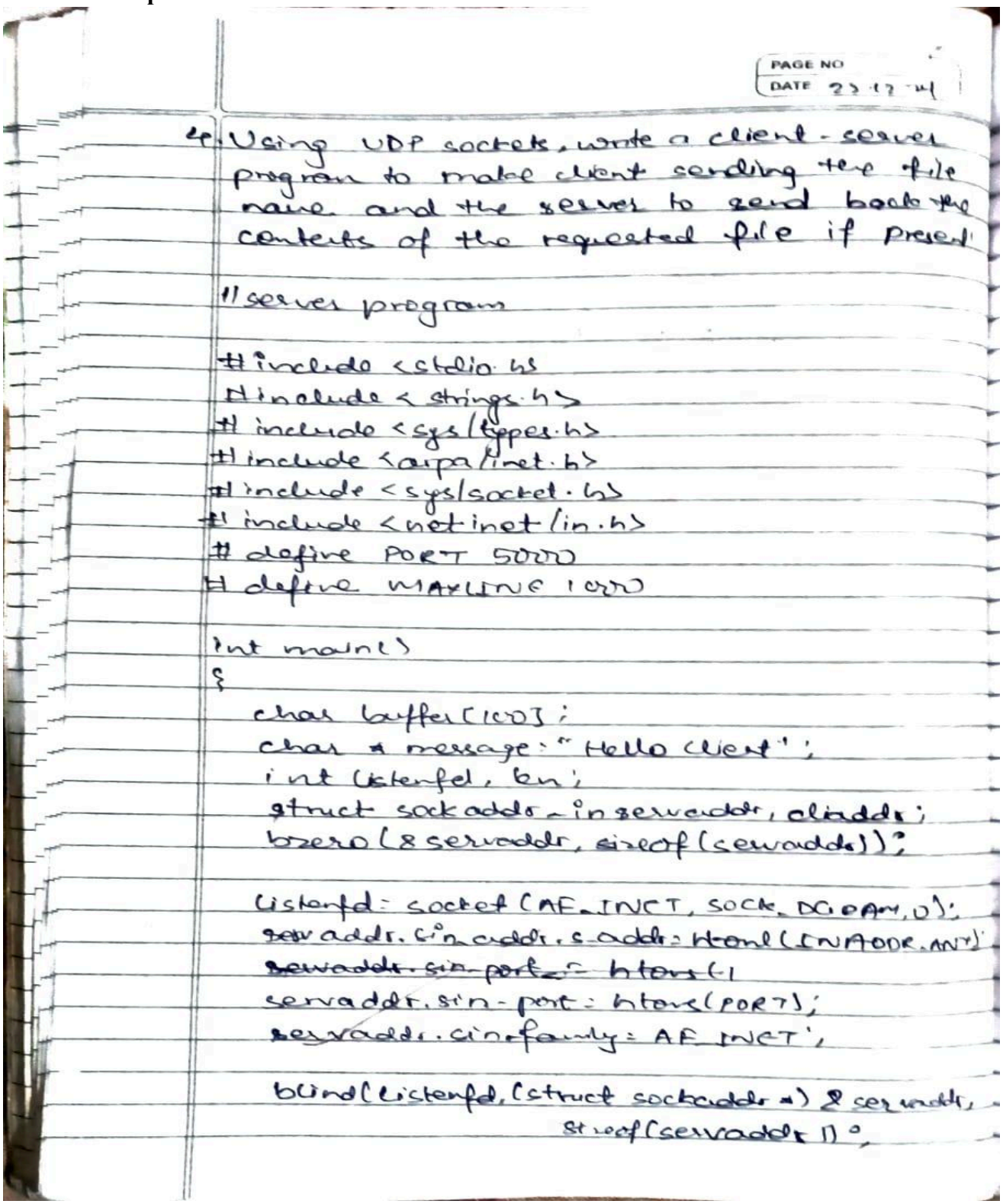
26/12/24

```
//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:**



```
PAGE NO
DATE 23.12.24

4. 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

// server program

#include <stdio.h>
#include <strings.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, bn;
    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));
```

PAGE NO :
DATE :

~~function~~

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

sendto(listenfd, message, MAXLINE, 0,
       (struct sockaddr *) &cliaddr,
       sizeof(cliaddr));
}
```

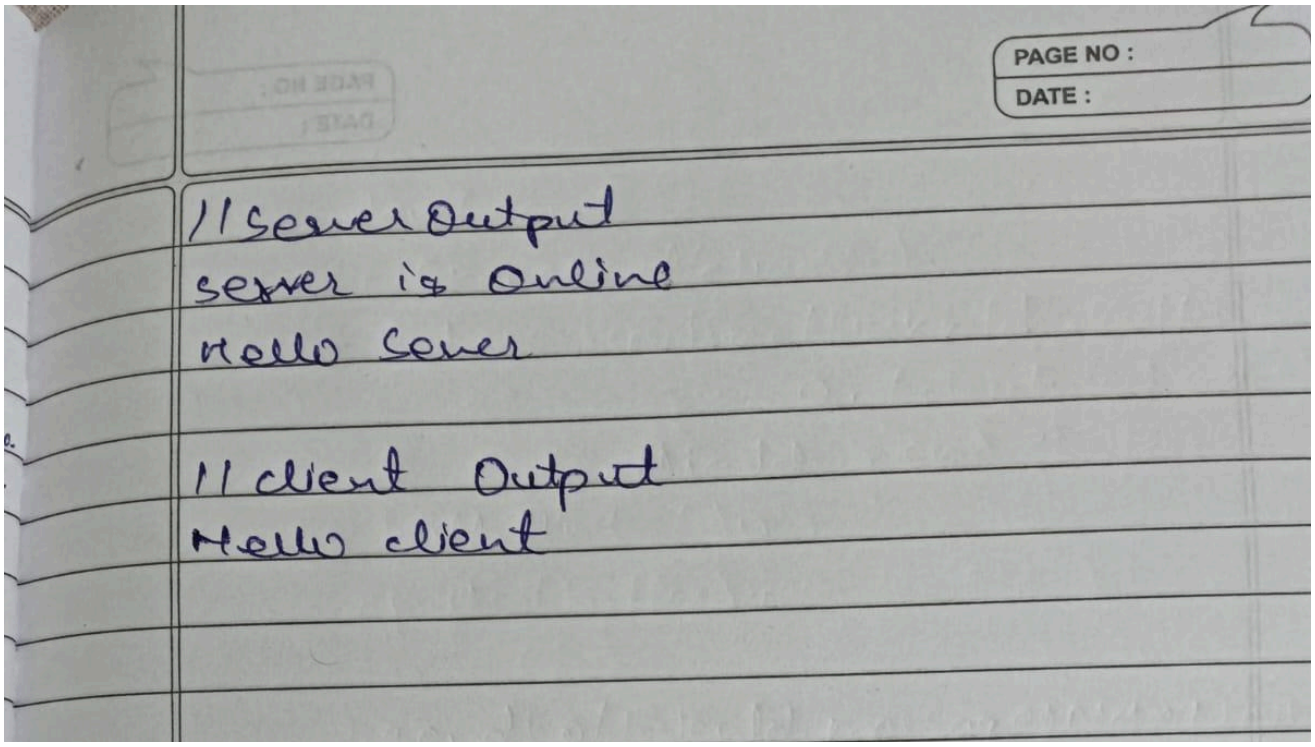
// client driver program

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

#define PORT 5000
#define MAXLINE 1000

int main()
{
    char buffer[100];
    char *message = "Hello server";
    int sockfd, n;
```

PAGE NO :
DATE :

```
struct sockaddr_in servaddr;

bzero(&servaddr, sizeof(servaddr));
servaddr.sin_addr.s_addr = inet_addr(
servaddr.sin_port = htons(PORT);
servaddr.sin_family = AF_INET;

sockfd = socket(AF_INET, SOCK_DGRAM, 0)

if (connect (sockfd, (struct sockaddr*) &
    servaddr, sizeof (servaddr)) < 0)
{
    printf("\n Error : Connect Failed \n");
    exit(0);
}

sendto(sockfd, message, MAXLINE, 0,
        (struct sockaddr*) NULL,
        sizeof (servaddr));

recvfrom(sockfd, buffer, sizeof(buffer), 0,
        (struct sockaddr*) NULL, NULL);
puts(buffer);

close(sockfd);
}
```

26/12/24


PAGE NO :
DATE :
// Server Output
server is Online
Hello Sever
// client Output
Hello client