VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB RECORD

Computer Network Lab (23CS5PCCON)

Submitted by

Sidhvin Vidyadhar Burli(1BM21CS211)

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 **Sidhvin Vidyadhar Burli(1BM21CS211)** 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.

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

Index

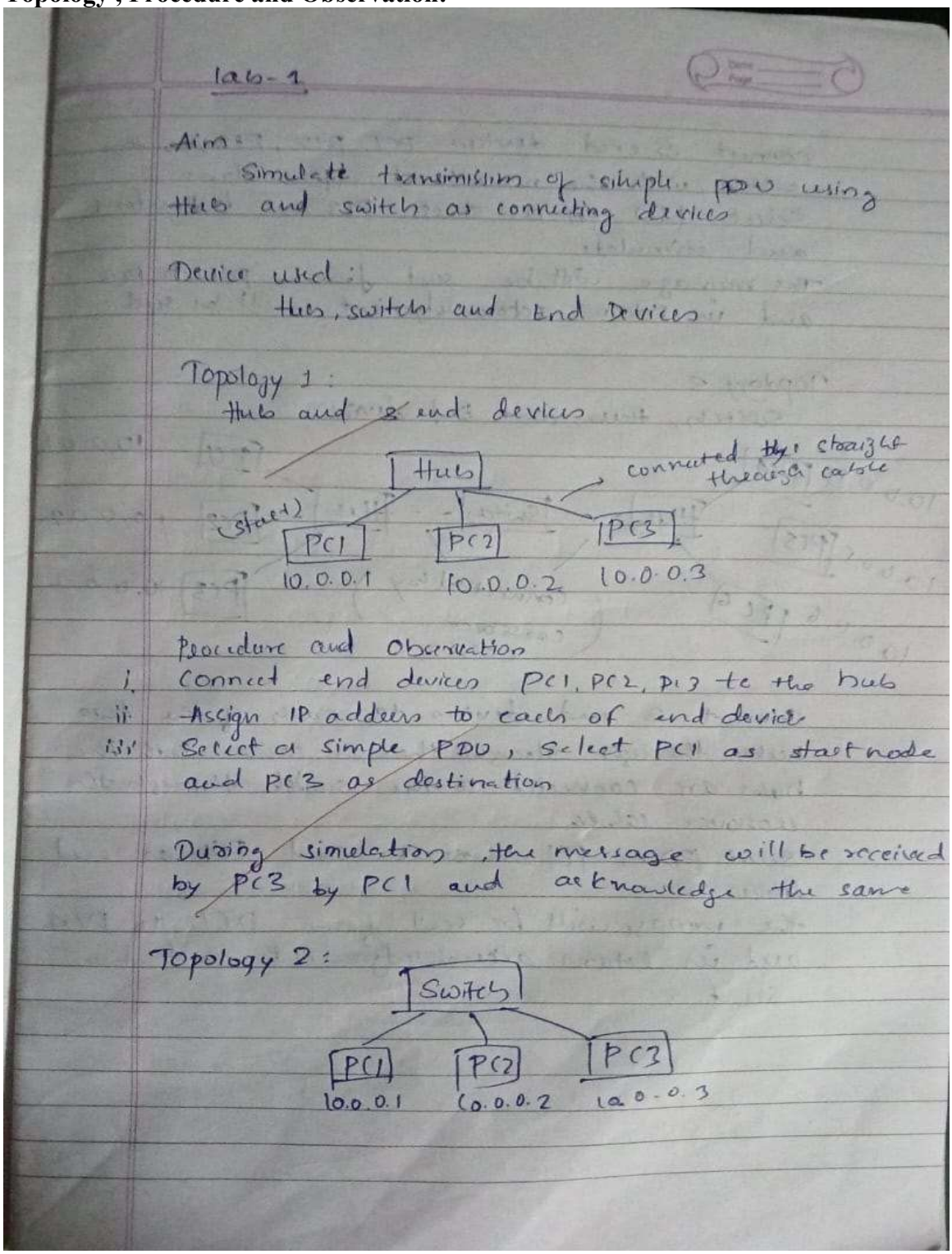
17	18-12-24	Using UDP sockets, write a client-server program to make the client send the file name and the server to send back the contents of the requested file if present.	62-64

Github Link: https://github.com/SidhvinBurli/CNLAB5TH

Program 1

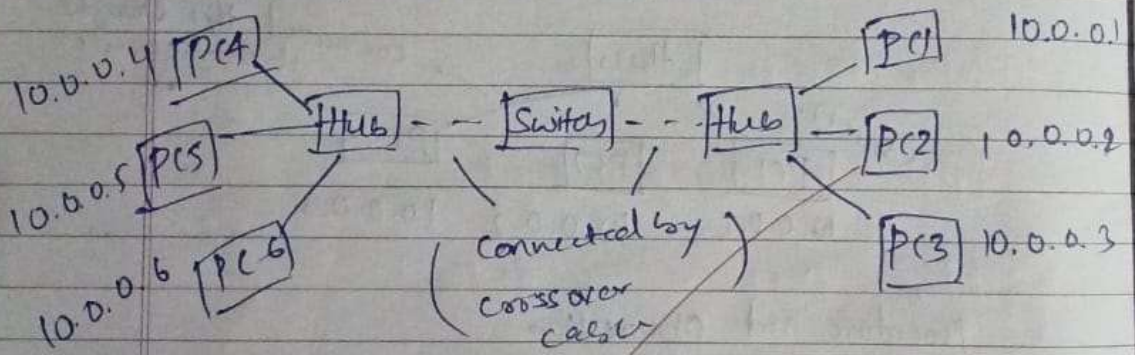
Aim: Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping messages.

Topology , Procedure and Observation:



Connect 3 end devices PC1, PC2, PC3, to the switch with the mentioned IP address
Select PDU, PC1 as start PC3 as destination and stimulate
The message will be sent from PC1 to PC3 and in returns acknowledgement will be send

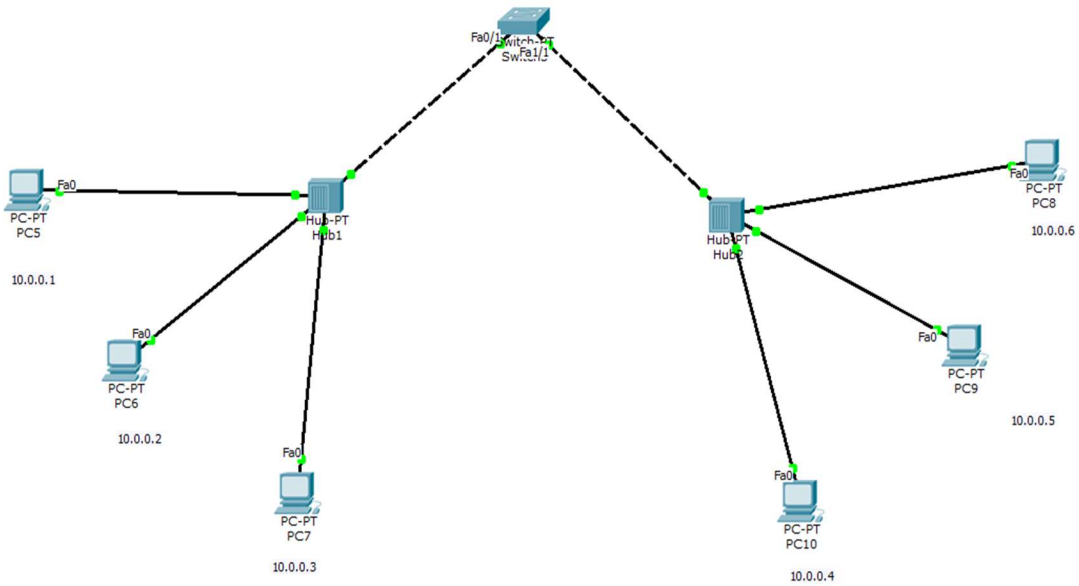
Topology 3:
Switch, Hub and End device

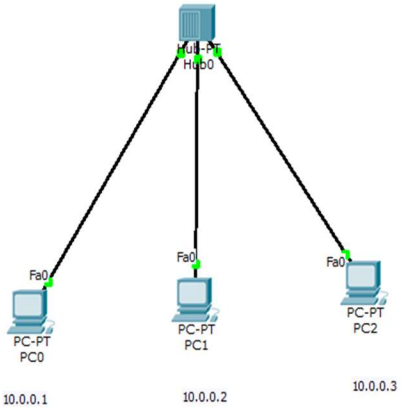


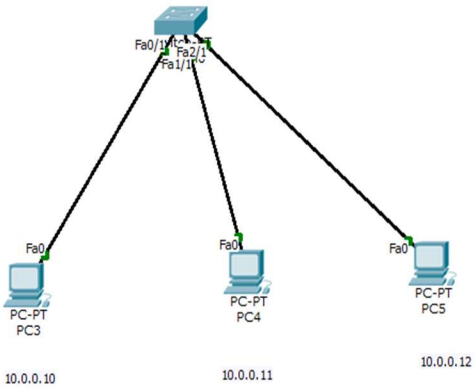
Connect 3 end devices to each of the two hubs through straight through cables. Both the hubs are connected to the switch through the crossover cables
Select PDU, PC1 as start and PC4 as end devices
The message will be sent from PC1 to PC4 and in return acknowledgement will be sent

Switch
24/10/24

Screen Shots:



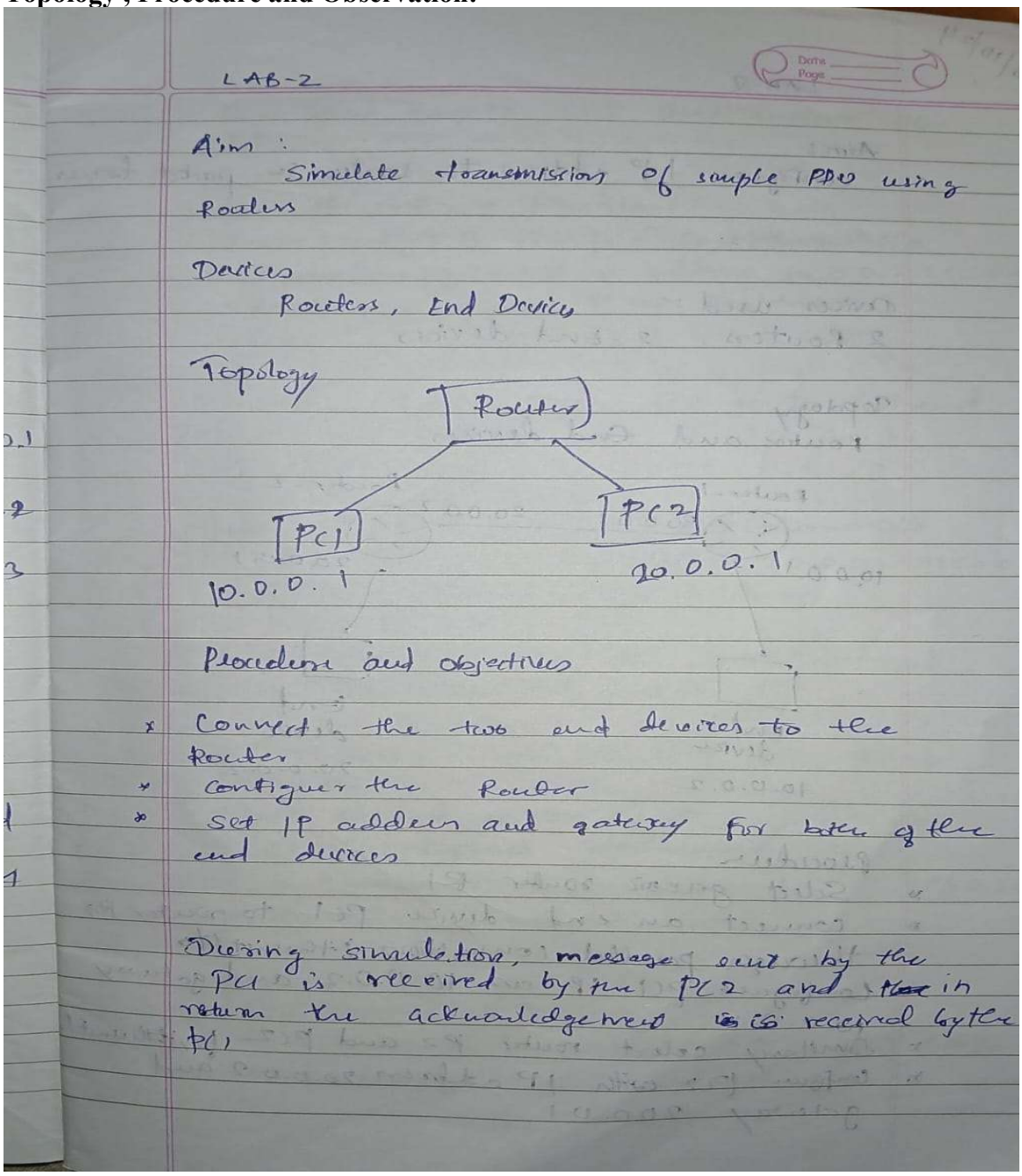




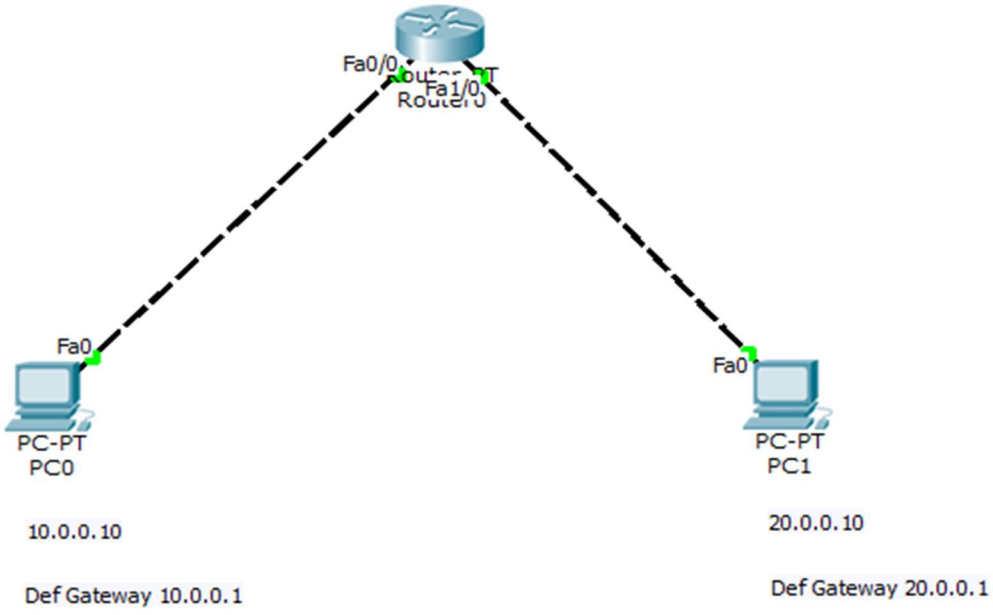
Program 2

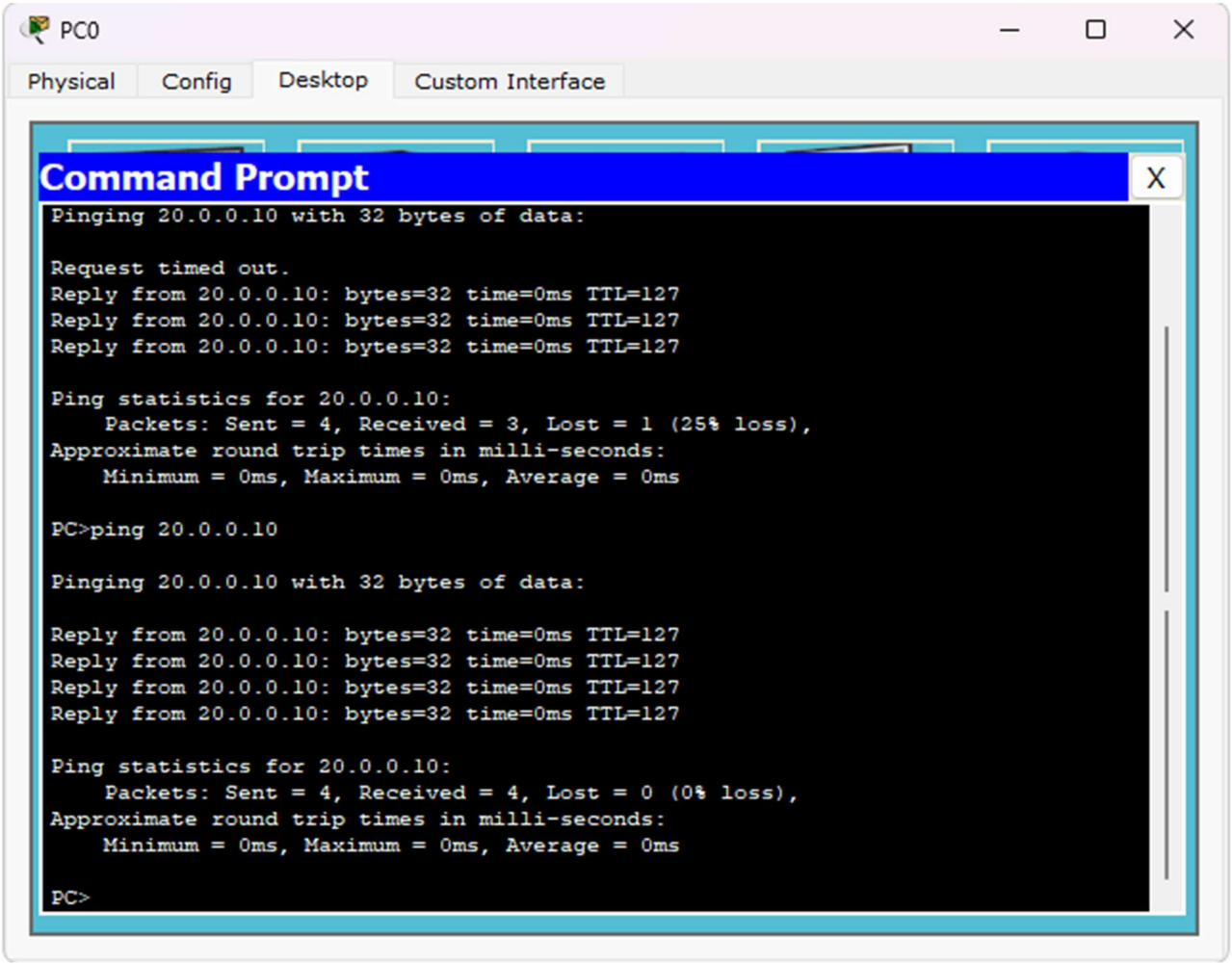
Aim: Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply.

Topology , Procedure and Observation:



Screen Shots:

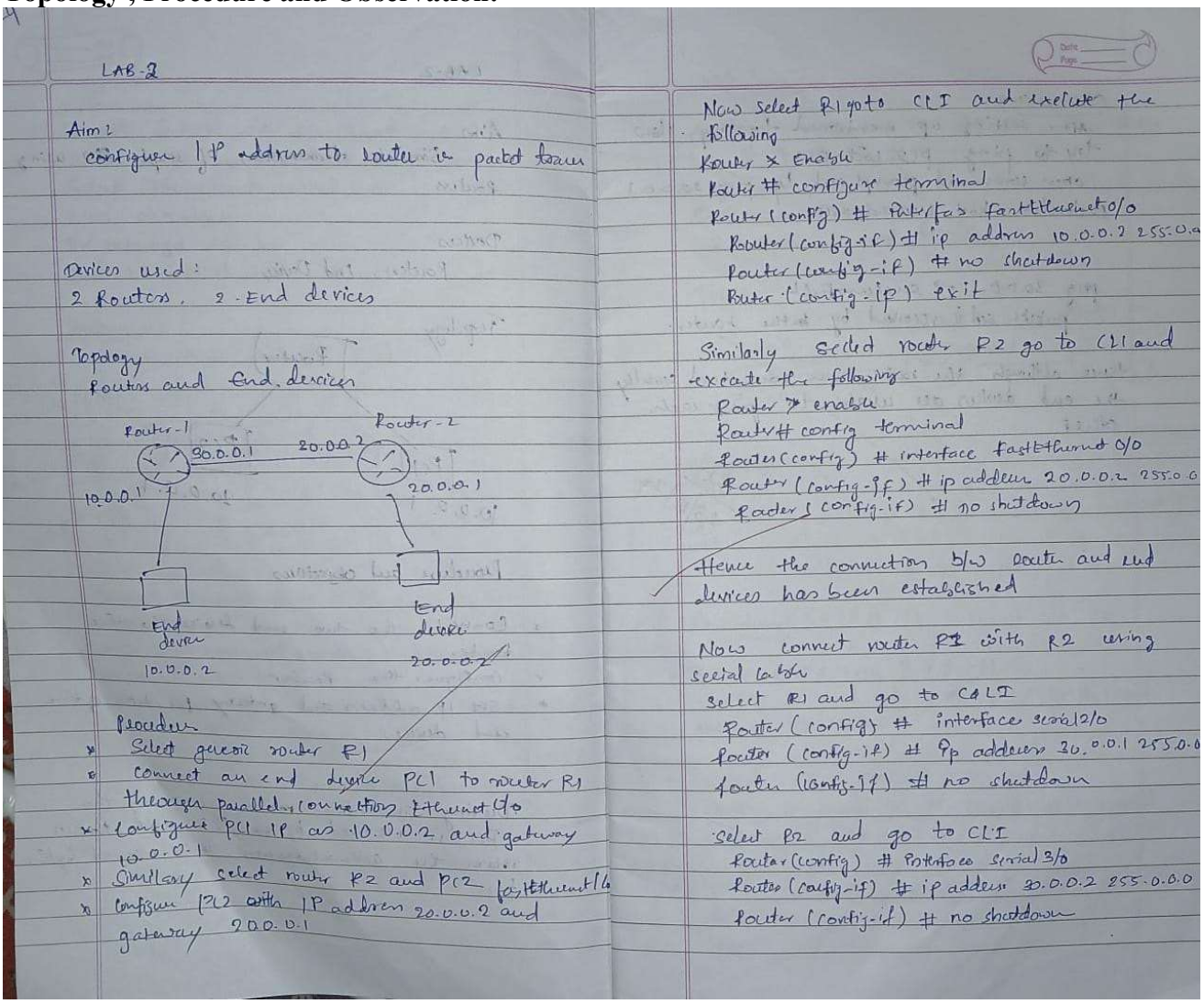




Program 3

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

Topology , Procedure and Observation:



LAB-2

Aim:
configure IP address to router in packet tracer

Devices used:
2 Routers, 2 End devices

Topology
Routers and End devices

Procedure
- Select generic router R1
- Connect an end device PC1 to router R1 through parallel connection Ethernet 0/0
- Configure PC1 IP as 10.0.0.2 and gateway 10.0.0.1
- Similarly select router R2 and PC2 fastEthernet 1/0
- Configure PC2 with IP address 20.0.0.2 and gateway 20.0.0.1

Now select R1 goto CLI and execute the following
Router > Enable
Router # configure terminal
Router (config) # interface fastEthernet 0/0
Router (config-if) # ip addres 10.0.0.2 255.0.0
Router (config-if) # no shutdown
Router (config-ip) exit

Similarly select router R2 go to CLI and excute the following
Router > enable
Router# config terminal
Router (config) # interface fastEthernet 0/0
Router (config-if) # ip address 20.0.0.2 255.0.0
Router (config-if) # no shutdown

Hence the connection b/w router and end devices has been established

Now connect router R1 with R2 using serial cable
select R1 and go to CLI
Router (config) # interface serial 2/0
Router (config-if) # ip address 30.0.0.1 255.0.0
Router (config-if) # no shutdown

select R2 and go to CLI
Router (config) # interface serial 2/0
Router (config-if) # ip address 30.0.0.2 255.0.0.0
Router (config-if) # no shutdown

Observations:

After setting up mentioned topology. Now try to ping PC2 with PC1

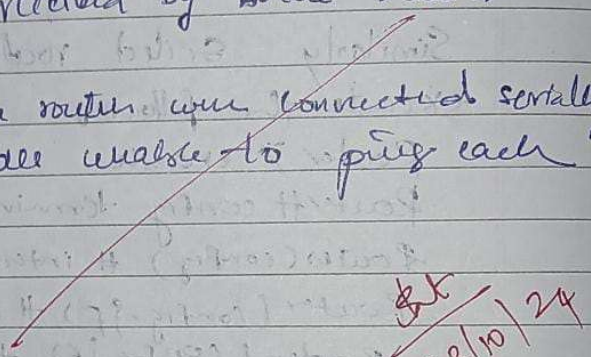
open command prompt and type ping 20.0.0.1

Destination host unreachable

ping 30.0.0.1 → successful

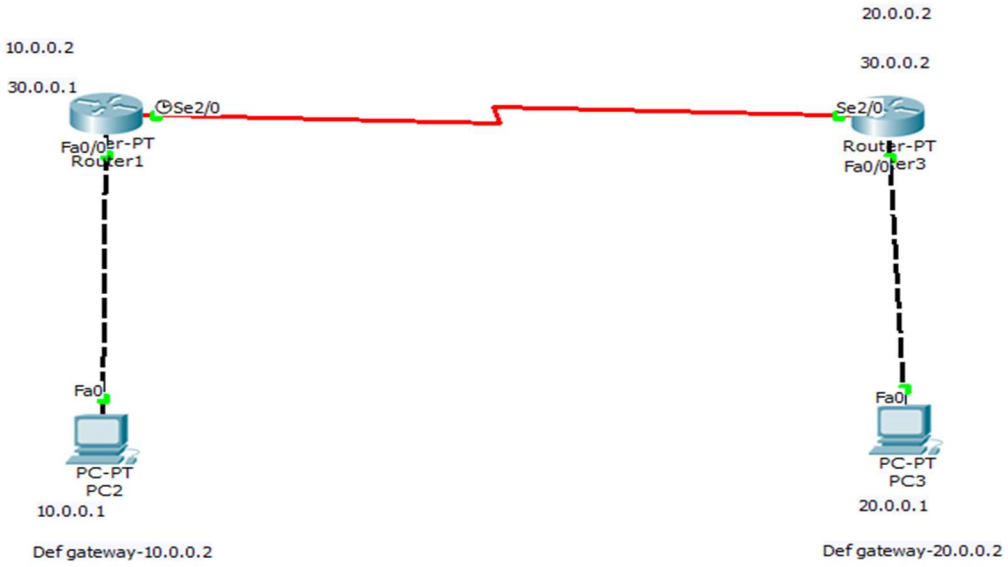
packets sent is received by the router

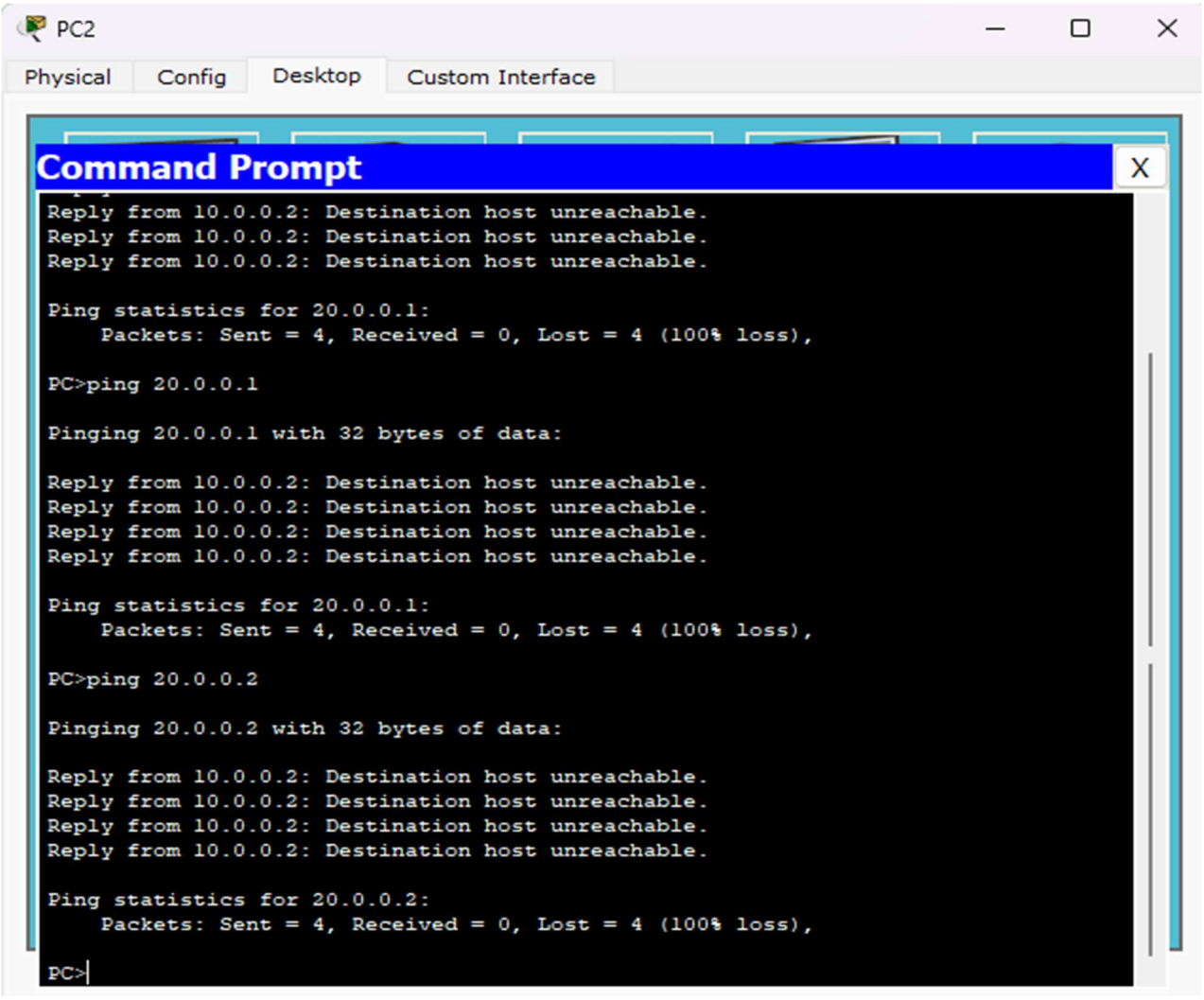
Hence although the routers are connected serially the end devices are unable to ping each other



23/10/24

Screen Shots:

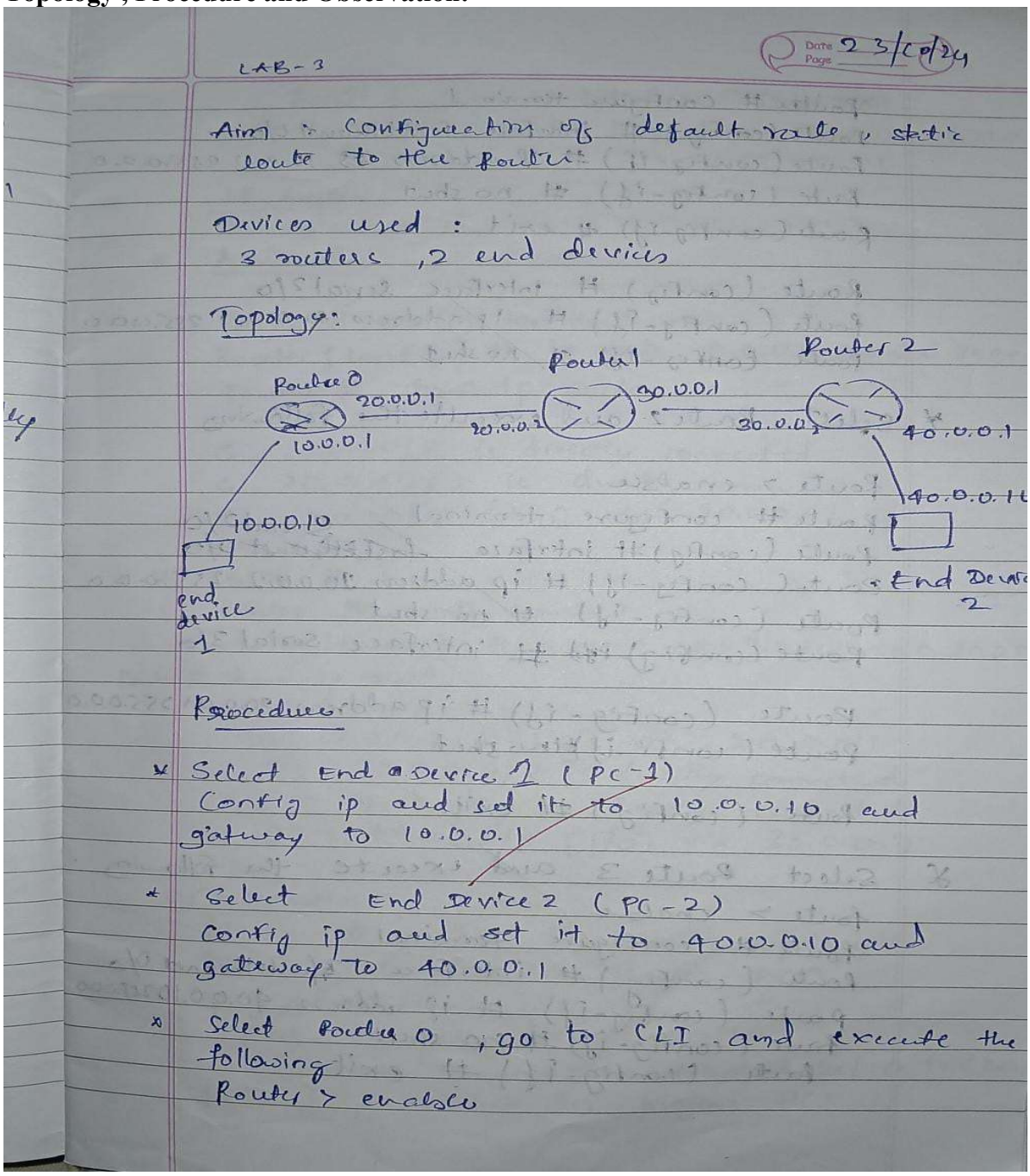



PC2
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.
Ping statistics for 20.0.0.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.
Ping statistics for 20.0.0.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.
Reply from 10.0.0.2: Destination host unreachable.
Ping statistics for 20.0.0.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>

Program 4

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

Topology , Procedure and Observation:



Router # Configure terminal
Router (config) # interface fastEthernet 0/0
Router (config-if) # ip address 10.0.0.1 255.0.0.0
Router (config-if) # no shut
Router (config-if) # exit

Router (config) # interface serial 2/0
Router (config-if) # ip address 30.0.0.1 255.000
Router (config-if) # no shut

* Select Router 2 and execute the following

Router > enable
Router # configure terminal
Router (config) # interface Serial 2/0
Router (config-if) # ip address 20.0.0.2 255.0.0.0
Router (config-if) # no shut
Router (config) # interface serial 3/0

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

Router (config) # exit

* Select Router 3 and execute the following
Router > enable
Router # configure terminal
Router (config) # interface 3/fastEthernet 0/0
Router (config-if) # ip address 40.0.0.10 255.000
Router (config-if) # no shut
Router (config-if) # exit

Router (config) # interface serial 3/0
Router (config-if) # ip address 30.0.0.2 255.0.0.0

IP Routes

* Router R0

Router # config terminal
Router (config) # ip route 0.0.0.0 0.0.0.0 20.0.0.2
Router # show ip route

C 10.0.0.0/8 is directly connected
C 20.0.0.0/8 is directly connected
S 0.0.0.0/0 [1/0] via 20.0.0.2

* Router R1
Router # config terminal
Router (config) # ip route 10.0.0.0 255.0.0.0 20.0.0.1
Router # show ip route

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

Observation:
After setting up the config topology, PC1 is successfull in pinging PC2
PC>ping 40.0.0.10
Ping statistics for 40.0.0.10:
Packets: sent = 4 received = 4 lost = 0 (0% loss)

Open CLI of PC2

PC > ping 10.0.0.10

Pinging 10.0.0.10 with 32 bytes of data

Ping statistics for 10.0.0.10:
Packets: Sent = 4, received = 4, Lost = 0 (0% loss)

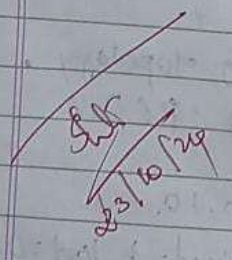
Hence, the connection between end devices is established successfully

IP Route

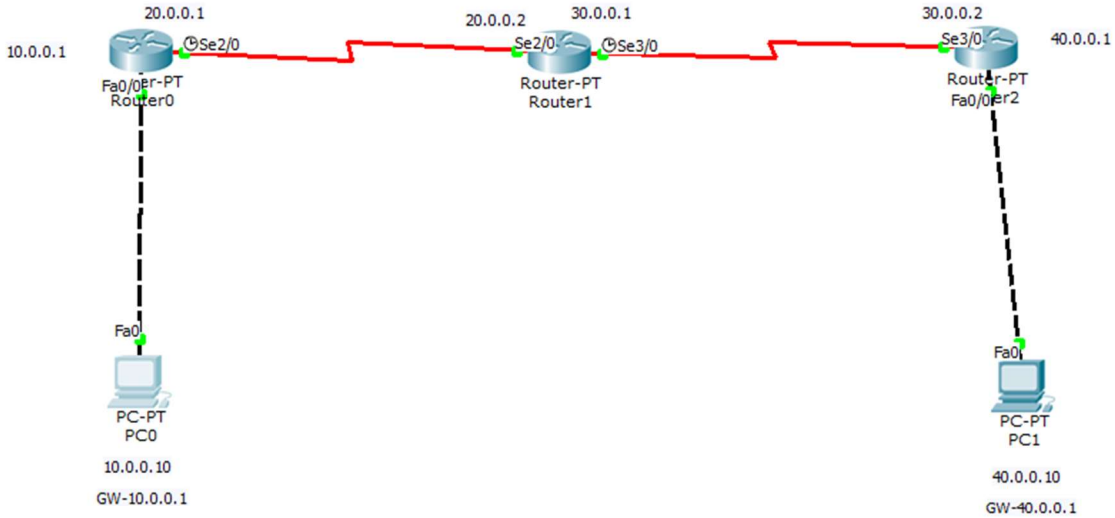
Router - 2

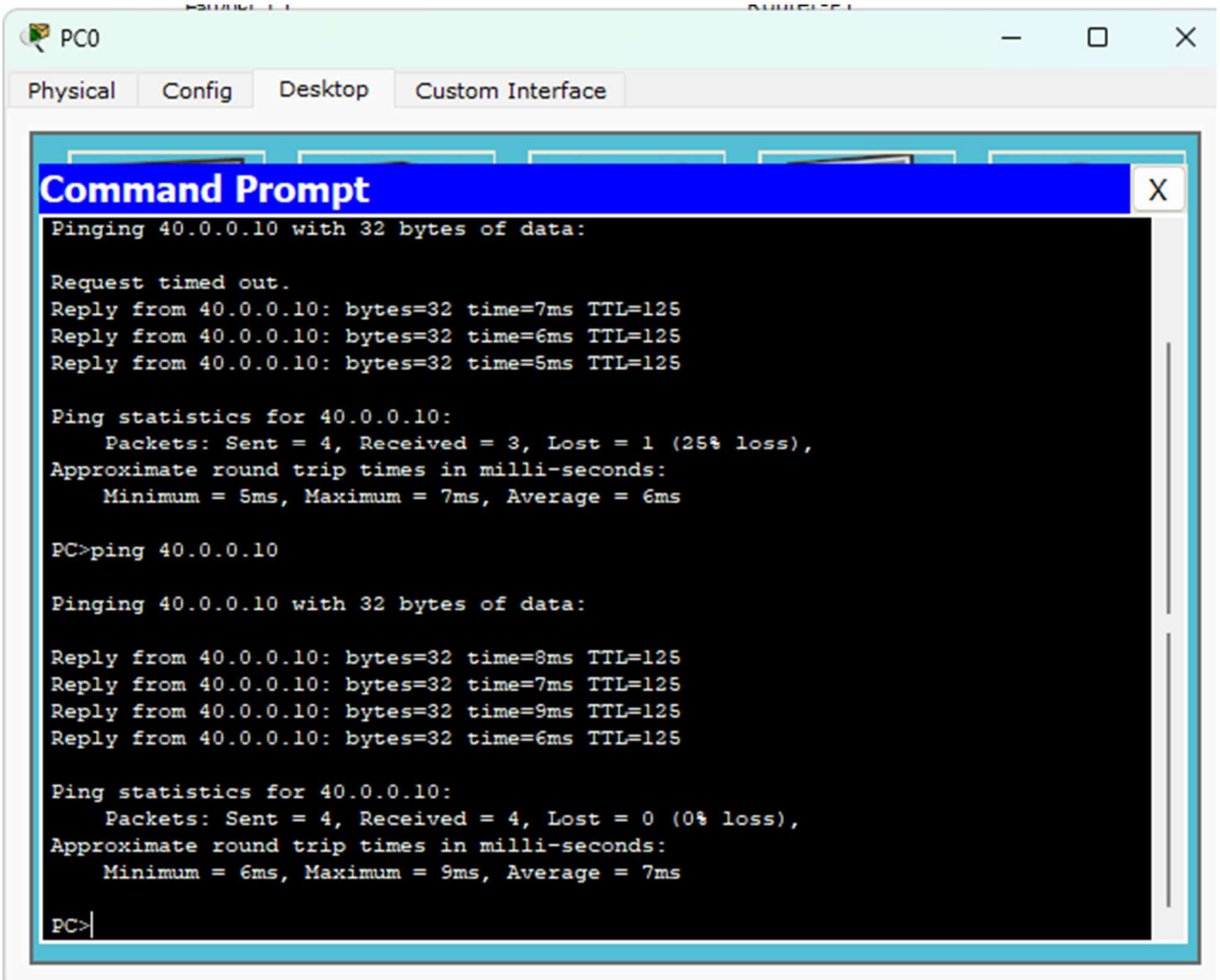
Route # show ip route

C 30.0.0.0/8 is directly connected
C 40.0.0.0/8 is directly connected
S 0.0.0.0/0 [1/0] via 30.0.0.1



Screen Shots:

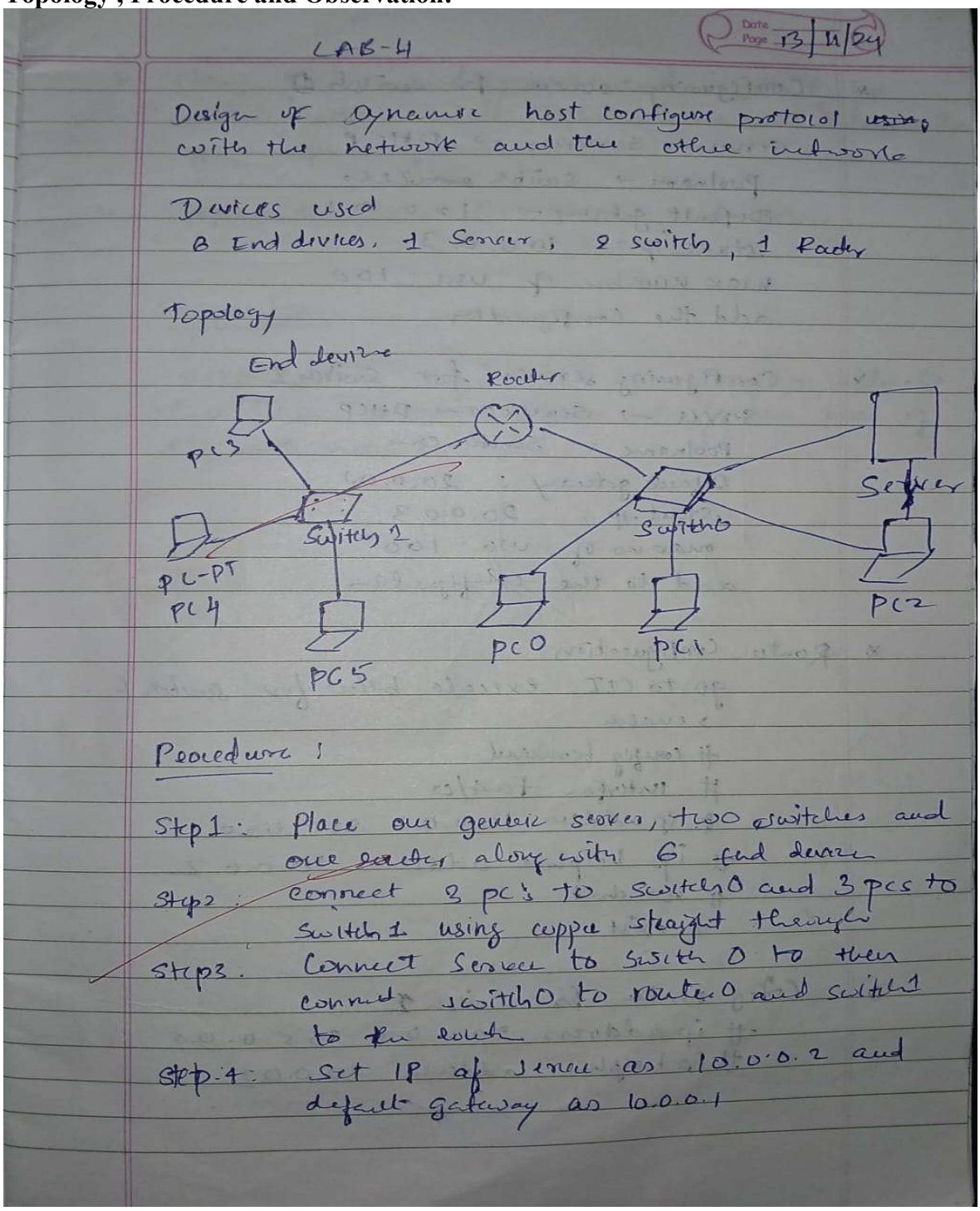




Program 5

Aim: Configure DHCP within a LAN and outside LAN.

Topology , Procedure and Observation:



* Configuring server for switch 0

Server → Services → DHCP
Pool name → switch ~~server~~
Default gateway → 10.0.0.1
start ip: 10.0.0.3
max number of use: 100
add the configuration

* Configuring server for switch 1
Server → Services → DHCP
Pool name: switch one
Default gateway: 20.0.0.1
start ip: 20.0.0.3
max no of use: 100
add to the configuration

* Router configuration
go to CLI, execute below for switch 1
> enable
config terminal
interface Fa4/0
ip address 10.0.0.1 255.0.0.0
ip helper address 10.0.0.2
no shut
exit

* Config router for switch 1
# ip address 20.0.0.1 255.0.0.0
# ip helper address 20.0.0.2

* Now go to end system and select ip configurations, select DHCP. Server will automatically give ip to systems

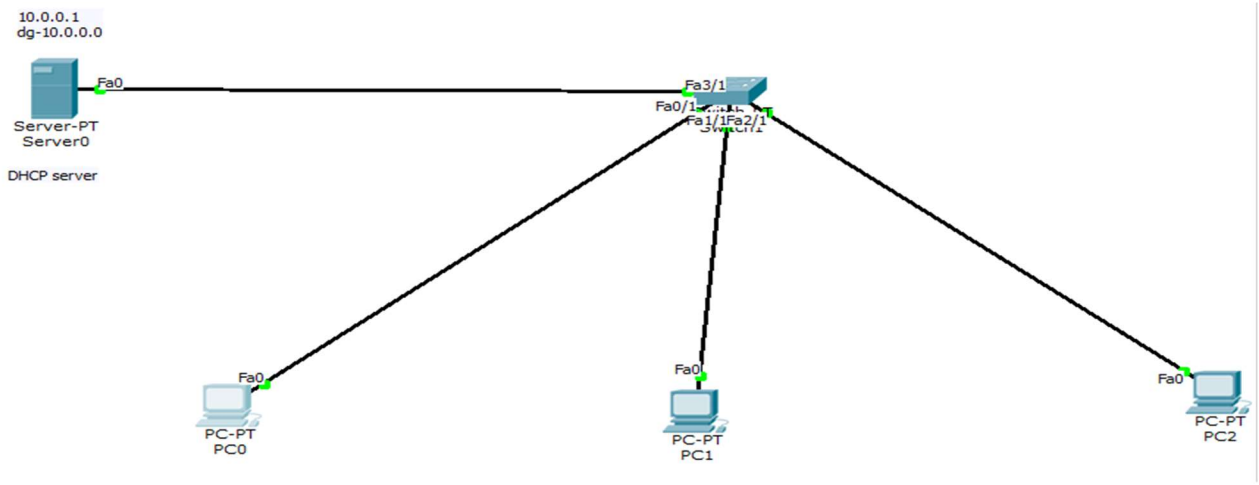
* Repeat the same for other end systems

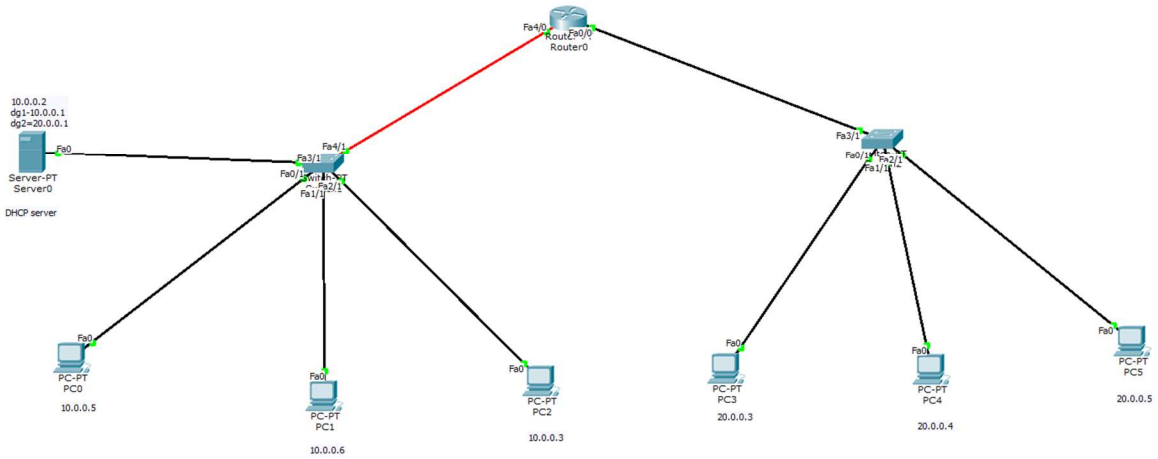
Observations:

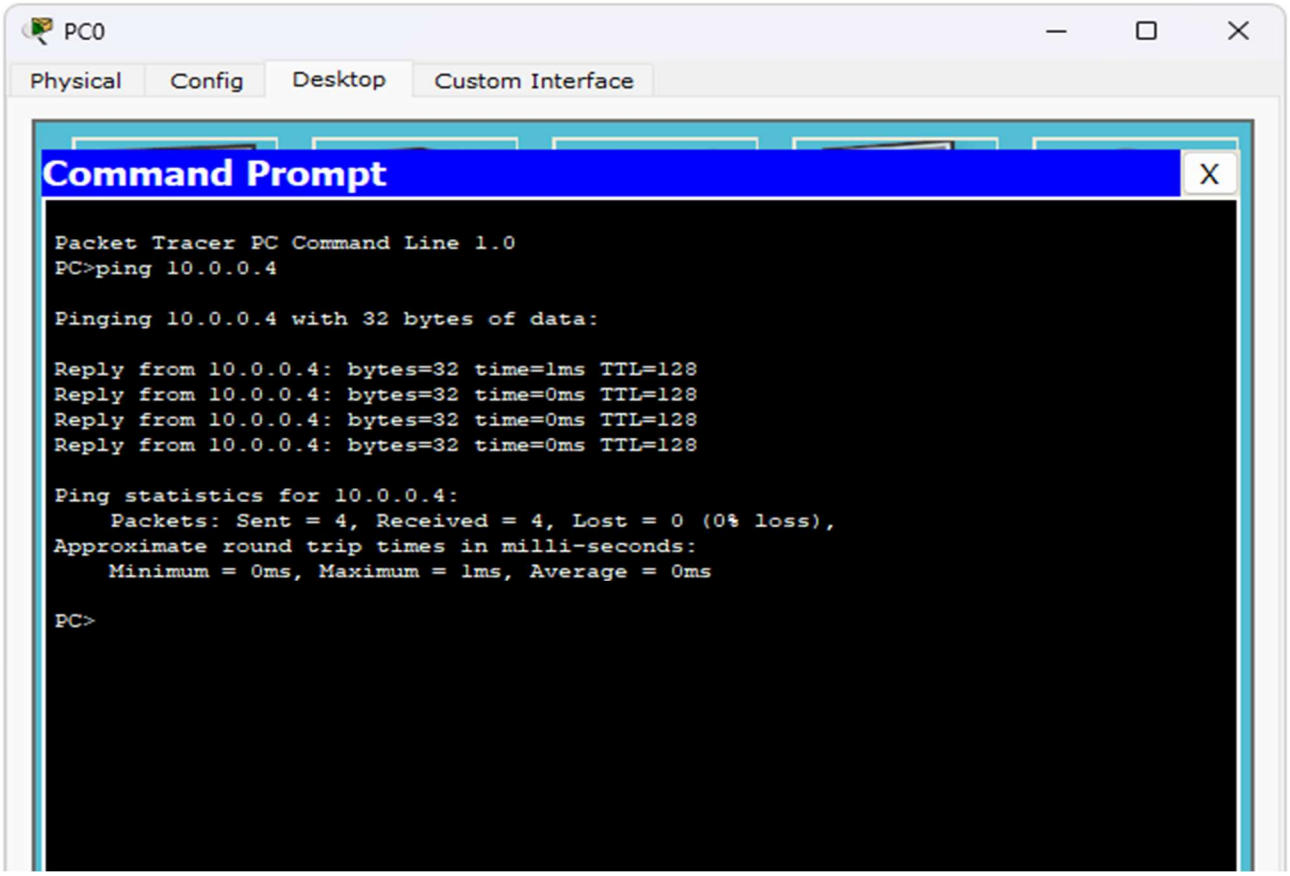
DHCP can be used to dynamically allocate IP address to end systems which are part of different network

Sri
20/11/24

Screen Shots:



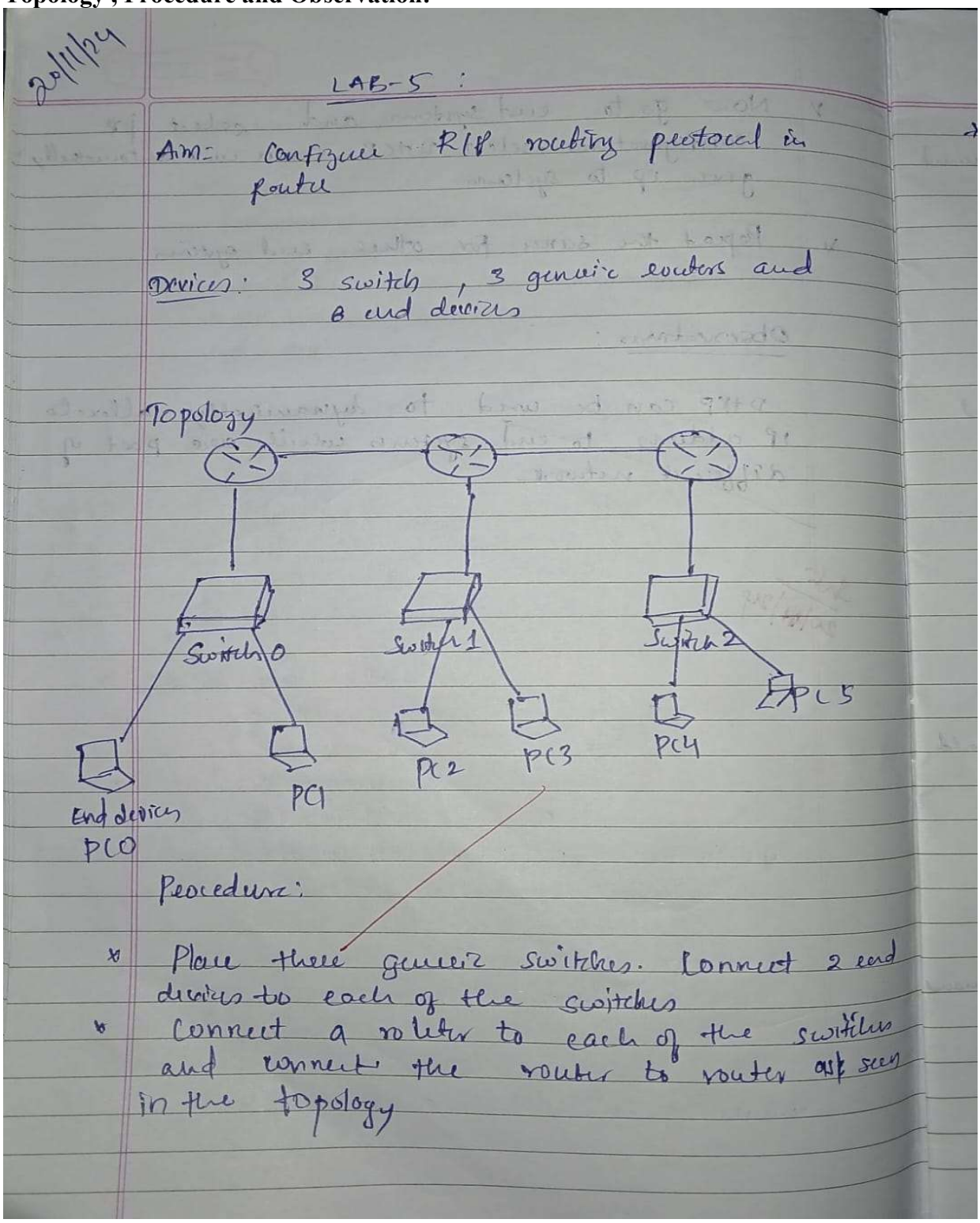



PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.4
Pinging 10.0.0.4 with 32 bytes of data:
Reply from 10.0.0.4: bytes=32 time=1ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from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



* Set Configuration of PC0 and PC1
set IP address of PC0 as 10.0.0.2
set default gateway - 10.0.0.1

set IP address of PC1 as 10.0.0.3
set default gateway - 10.0.0.1

* configure PC2 and PC3
set IP of PC2 : 20.0.0.2
default gateway : 20.0.0.1

Set IP of PC3 : 20.0.0.3
default gateway : 20.0.0.1

* Configure PC4 and PC5
Sett IP of PC4 : 30.0.0.2
Default gateway : 30.0.0.1

Sett IP of PC5 : 30.0.0.3
default gateway : 30.0.0.1

* Configure Router 0
Go to CLI and ~~write following~~ Command
> enable
config terminal
Interface fa 0/0
ip address 10.0.0.1 255.0.0.0
no shut
exit

Similarly configure Router1 with 20.0.0.1 255.0.0.0
and Router2 with 30.0.0.1 255.0.0.0

* Configure R0, R1

Go to CLI of R0 and enter following command

```
> enable
# config terminal
# interface Se2/0
# ip address 40.0.0.1 255.0.0.0
# no shut
# exit
```

Go to CLI of R1 and enter following command

```
> enable
# config terminal
# interface Se2/0
# ip address 40.0.0.2 255.0.0.0
# no shut
# exit
```

* Configure R1, R2

~~> int~~

Go to CLI of R1 and enter following command

~~> enable~~

```
# interface Se3/0
# ip address 50.0.0.1 255.0.0.0
# no shut
# exit
```

Go to CLI of R2 and enter following command

```
# config terminal
# interface Se2/0
# ip address 50.0.0.2 255.0.0.0
# no shut
# exit
```

Router RIP configuration

For Router 0

```
# config terminal
# router rip
# Network 10.0.0.0
# Network 40.0.0.0
```

for Router 1

```
# config terminal
# router rip
# Network 20.0.0.0
# Network 40.0.0.0
# Network 50.0.0.0
```

For Router 2

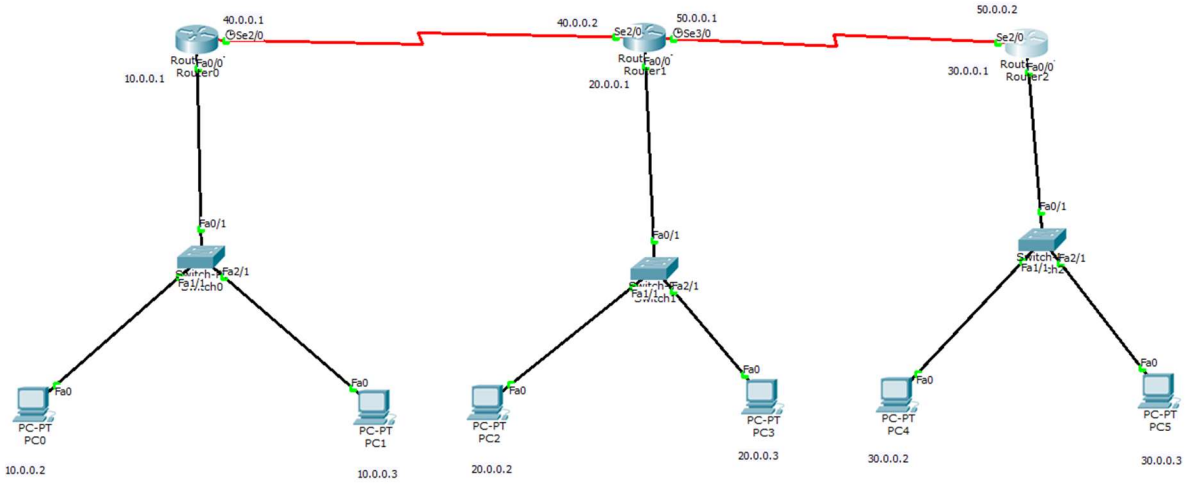
```
# config terminal
# router rip
# Network 50.0.0.0
# Network 30.0.0.0
```

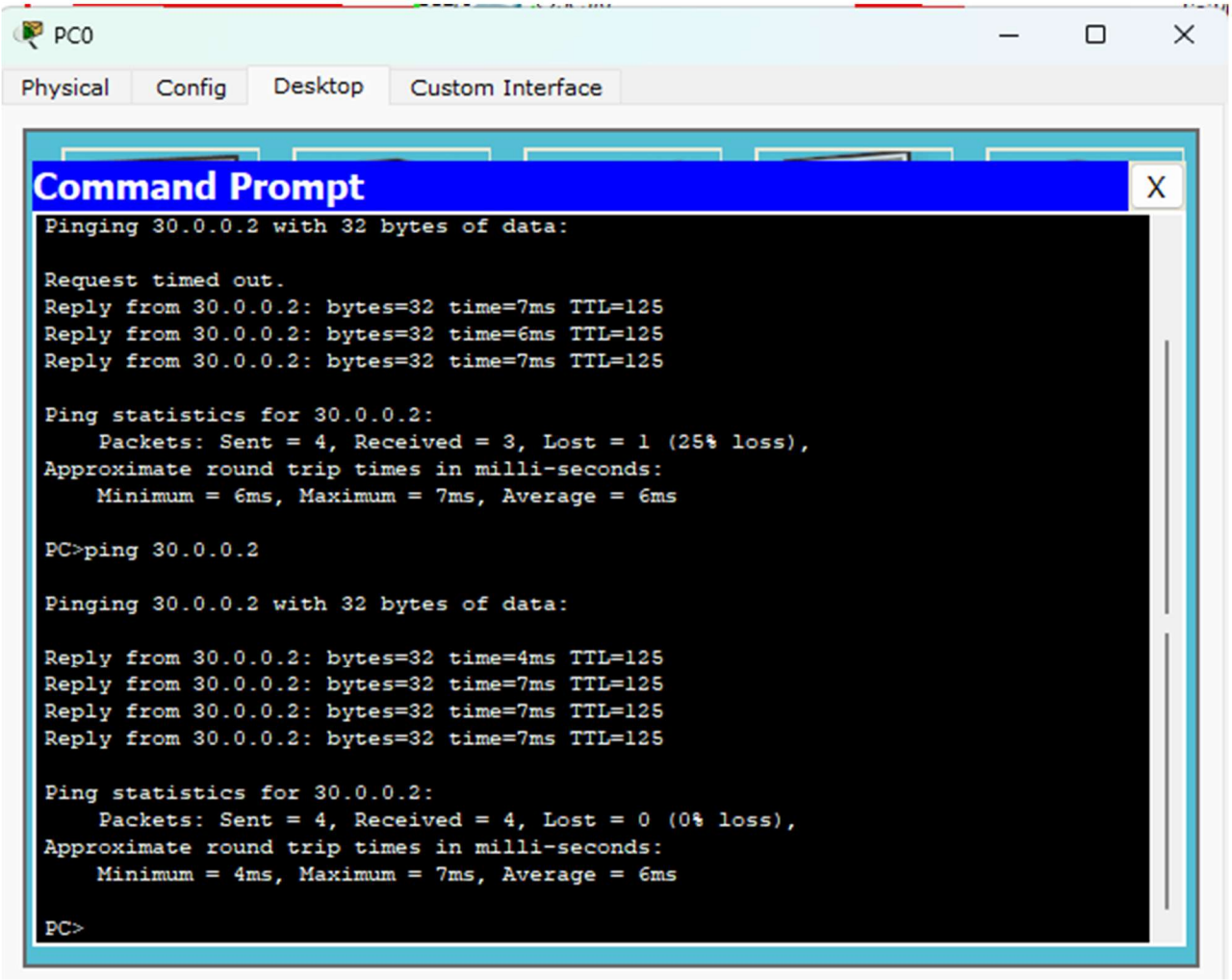
Observation:

After setting RIP routing protocol, we successfully ping between the end devices

SK
20/11/24

Screen Shots:



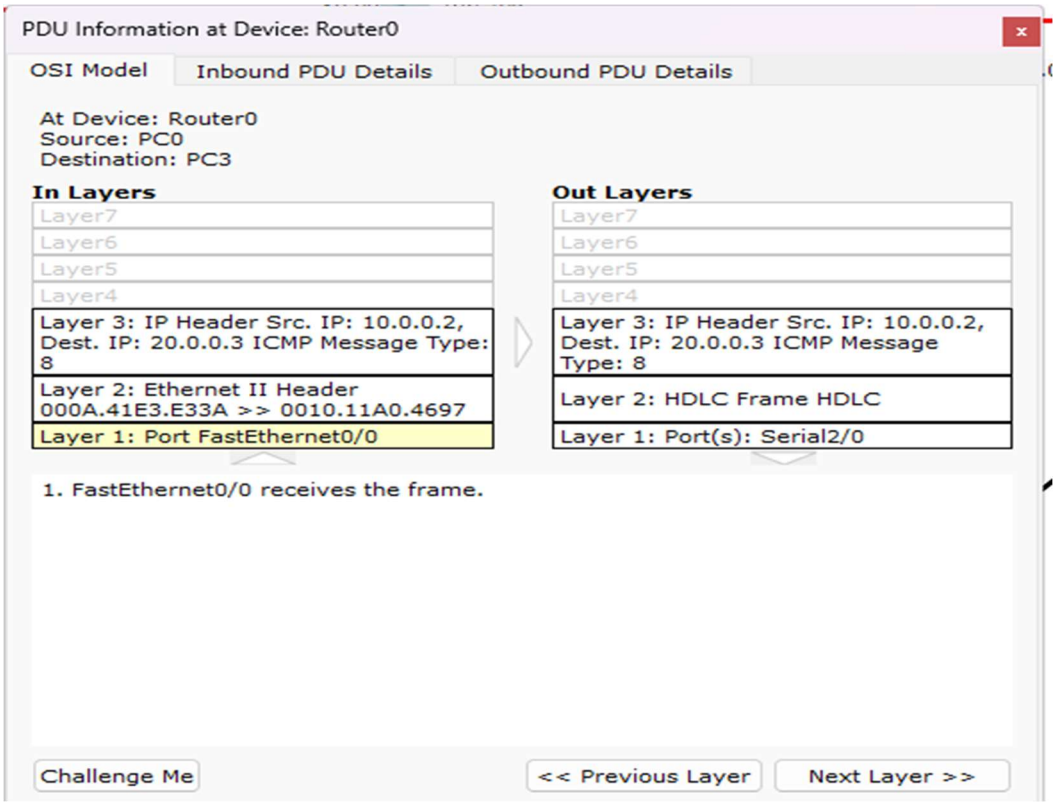


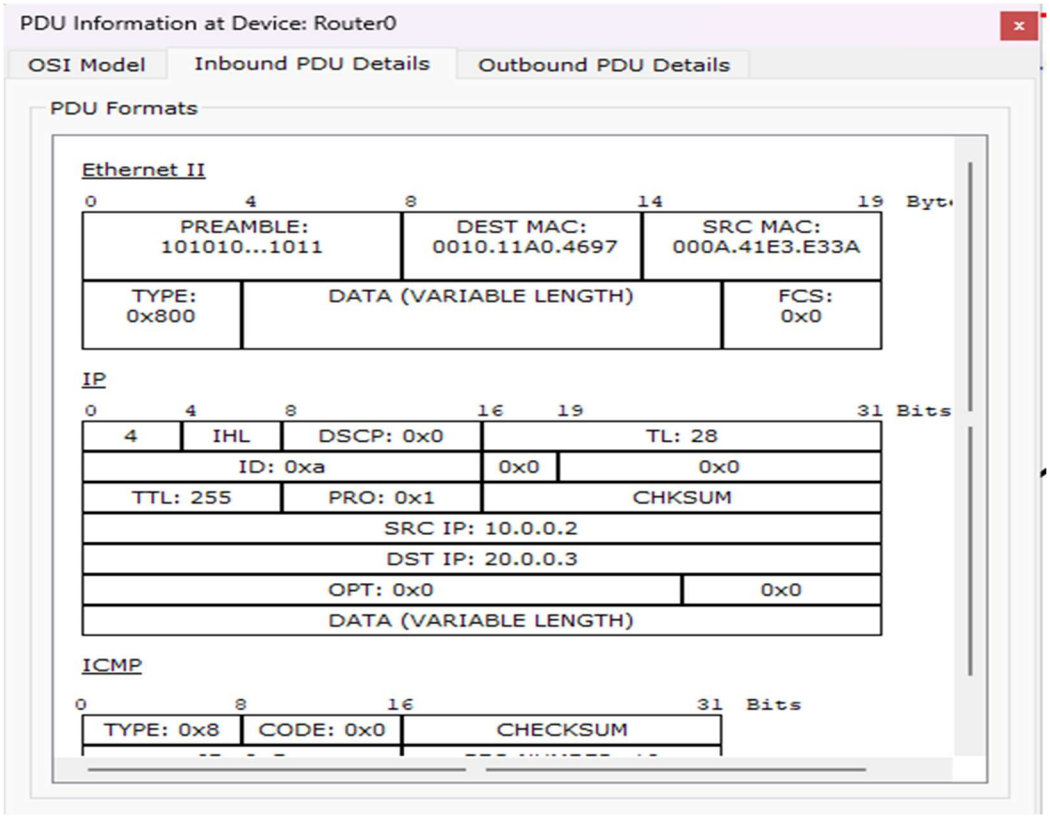
Program 7

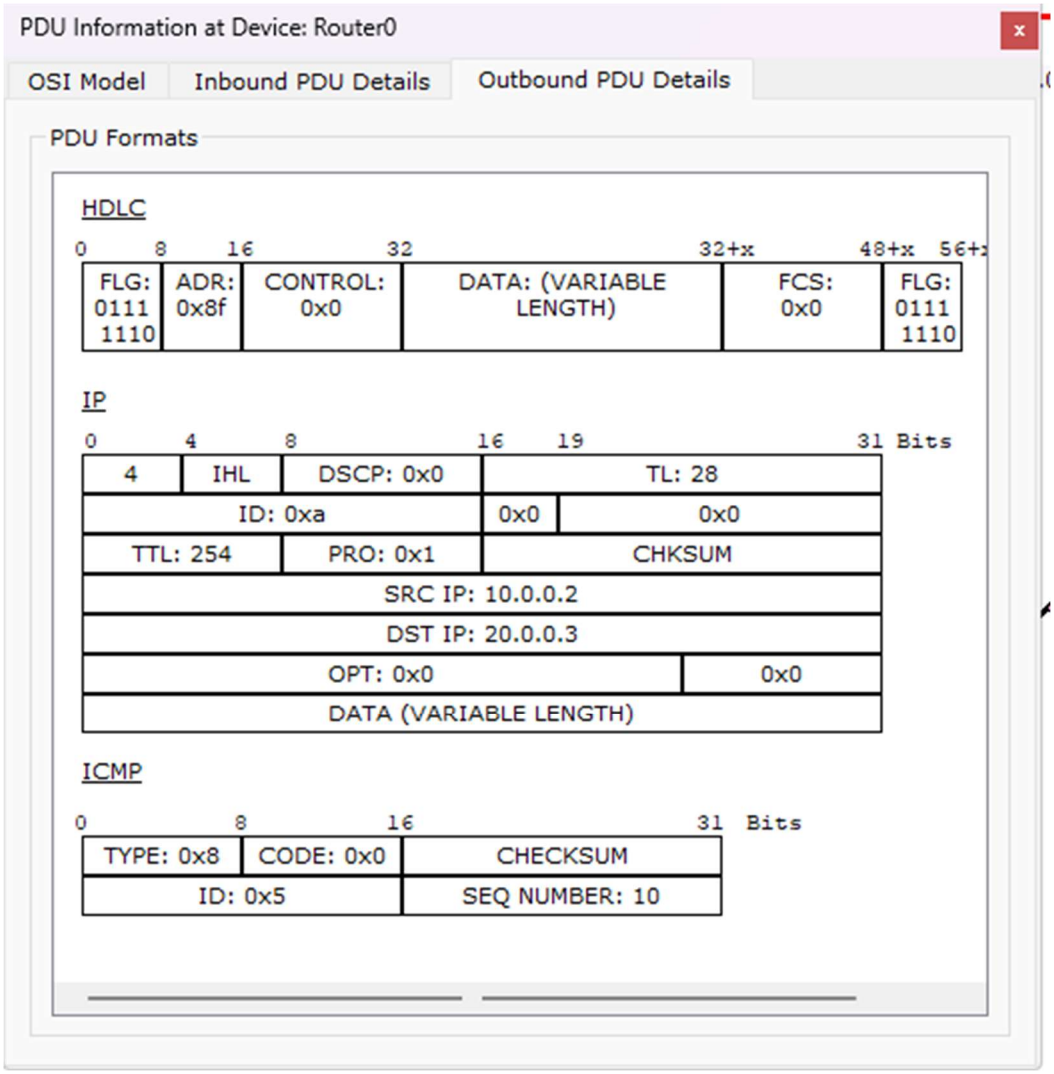
Aim: Demonstrate the TTL/ Life of a Packet .

Topology , Procedure and Observation:

Screen Shots:



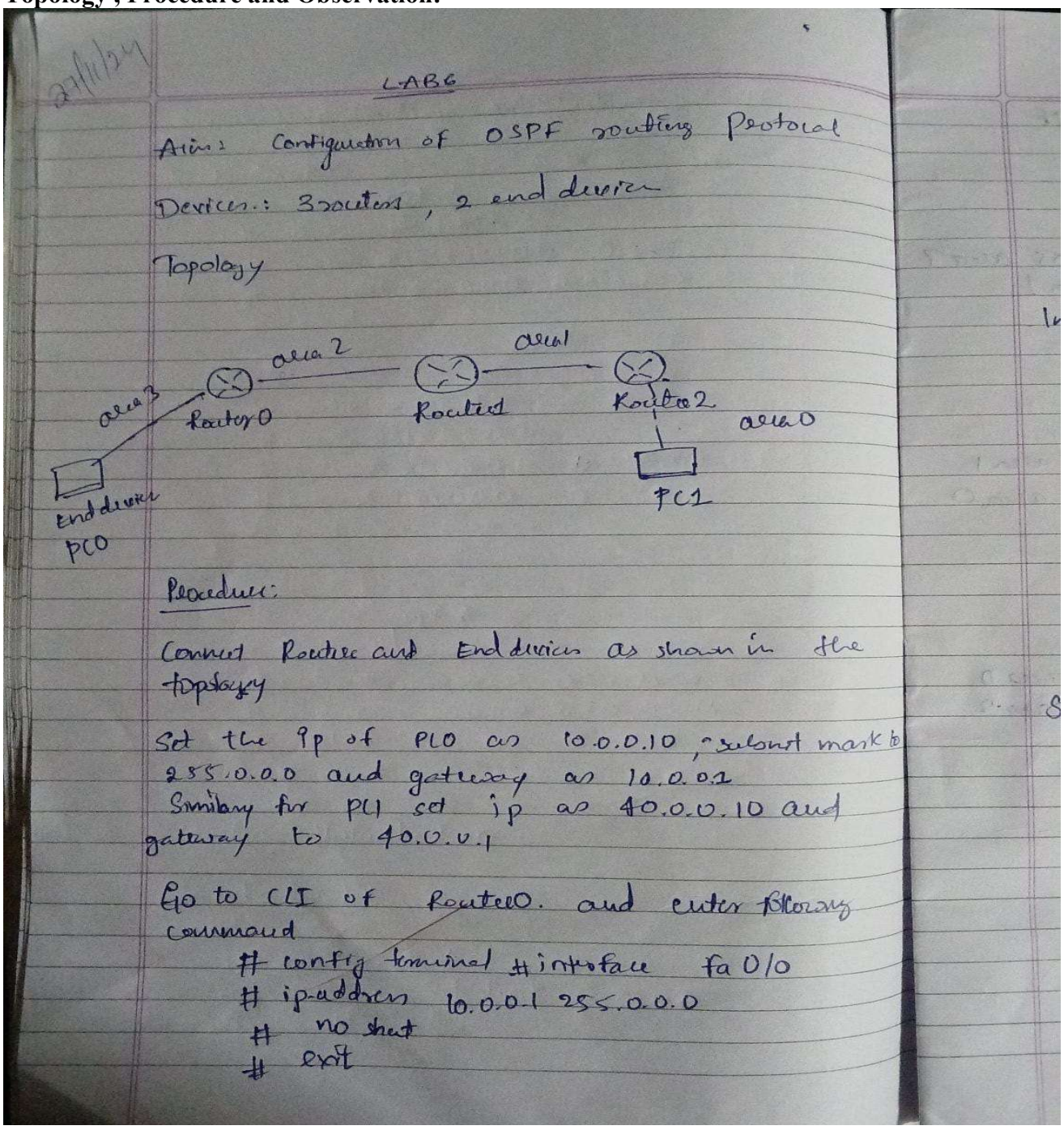

PDU Information at Device: Router0
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
0 4 8 14 19 Byt
PREAMBLE: 101010...1011
DEST MAC: 0010.11A0.4697
SRC MAC: 000A.41E3.E33A
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
0 4 8 16 19 31 Bits
4
IHL
DSCP: 0x0
TL: 28
ID: 0xa
0x0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



Interface se1/0
ip address 20.0.0.1 255.0.0.0
encapsulation ppp
clock rate 64000
no shut
exit

In Router R2

interface se1/0
ip address 20.0.0.2 255.0.0.0
encapsulation ppp
no shut
exit

interface se2/0
ip address 30.0.0.1 255.0.0.0
encapsulation ppp
clock rate 64000
exit

Similarly in R3

interface se1/0
ip address 30.0.0.2 255.0.0.0
encapsulation ppp
no shut
exit

interface fa2/0
ip address 40.0.0.1 255.0.0.0
no shut
exit

* Enabling ip routing by configuring Ospf

In Router R1
router ospf 1
router-id 1.1.1.1
network 10.0.0.0 0.255.255.255 area 1
network 20.0.0.0 0.255.255.255 area 1
exit

In Router R2
router ospf 1
router-id 2.2.2.2
network 10.0.0.0 0.255.255.255 area 1
network 30.0.0.0 0.255.255.255 area 0
exit

In Router R3
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
exit

* Creating Virtual-Link

In Router R1
router ospf 1
area 1 virtual-link 2.2.2.2
exit

In Router R2
area 1 virtual-link 1.1.1.1
exit

for Router R3

show ip route

IA 20.0.0.0 /8
IA 20.0.0.0/8
30.0.0.0/8
30.0.0.1/8
40.0.0.0/8
172.16.0.0/16

Observation:

~~Pc1 can successfully ping Pc1 from Pc0~~

Loopback

Router 1:
interface loopback 0
ip add 172.16.1.252 255.255.0.0
no shut
exit

Router 2
interface loopback 0
ip add 172.16.1.253 255.255.0.0

Router 3 :

interface loopback 0
ip add 172.16.1.252 255.255.0.0
no shut
end

Observation : All connections are successfull

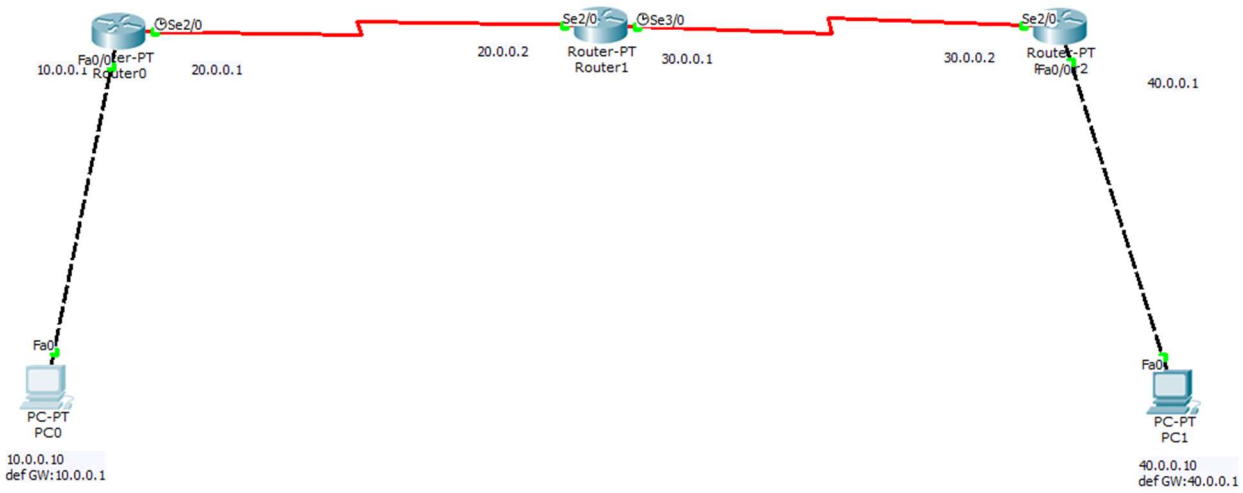
ping 40.0.0.10

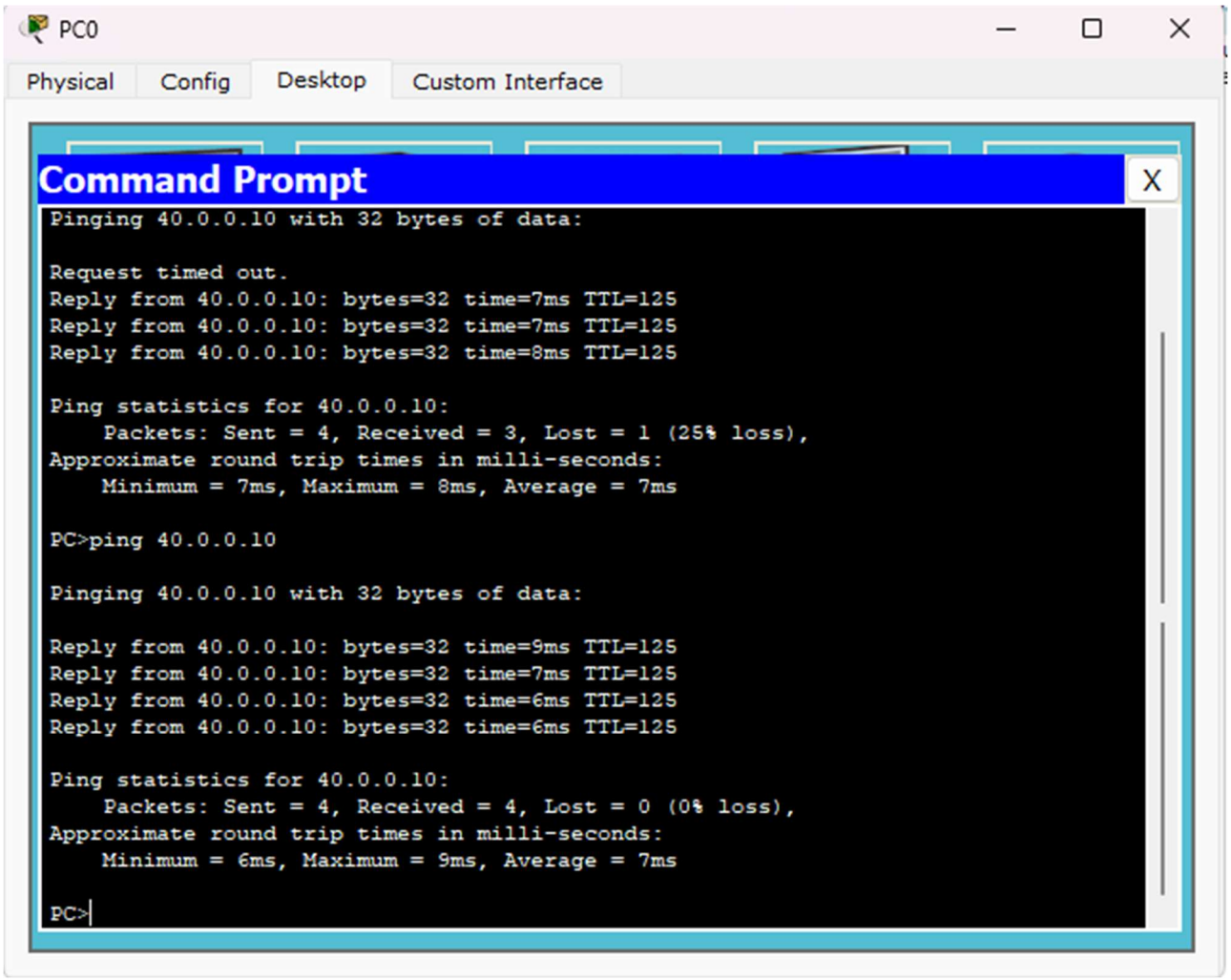
reply from 40.0.0.10 bytes=32 time=7ms TTL=125
reply from 40.0.0.10 bytes=32 time=7ms TTL=125
reply from 40.0.0.10 bytes=32 time=7ms TTL=125
reply from 40.0.0.10 bytes=32 time=7ms TTL=125

Ping Statistics

sent=4, Received=4, Loss=0 (0% loss)

Screen Shots:




PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Pinging 40.0.0.10 with 32 bytes of data:
Request timed out.
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Reply from 40.0.0.10: bytes=32 time=8ms TTL=125
Ping statistics for 40.0.0.10:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 7ms, Maximum = 8ms, Average = 7ms
PC>ping 40.0.0.10
Pinging 40.0.0.10 with 32 bytes of data:
Reply from 40.0.0.10: bytes=32 time=9ms TTL=125
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=6ms TTL=125
Ping statistics for 40.0.0.10:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 6ms, Maximum = 9ms, Average = 7ms
PC>

Program 9

Aim: Configure Web Server, DNS within a LAN.

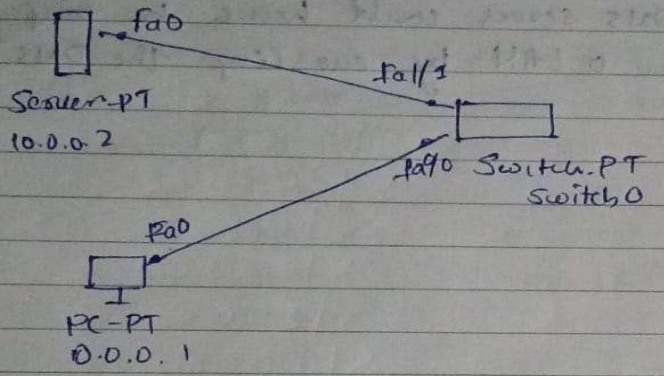
Topology , Procedure and Observation:

LAB-8

Objective

Configure Web server, DNS with a LAN

Topology



Procedure :

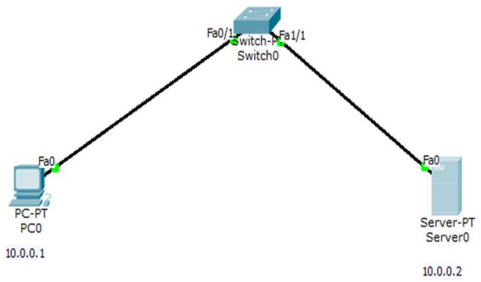
* Place an end device, a server and a switch. Connect them using copper straight wires
* Assign IP addresses as demostrated in topology
* To set IP of server
  - go to config
  - select DNS method, turn it on & add a resource
  - set IP, make sure port is on
  - select HTTP
  - Turn the services to on
  - Select PC0 → Desktop → Web browser
  - Enter URL specified in DNS resources

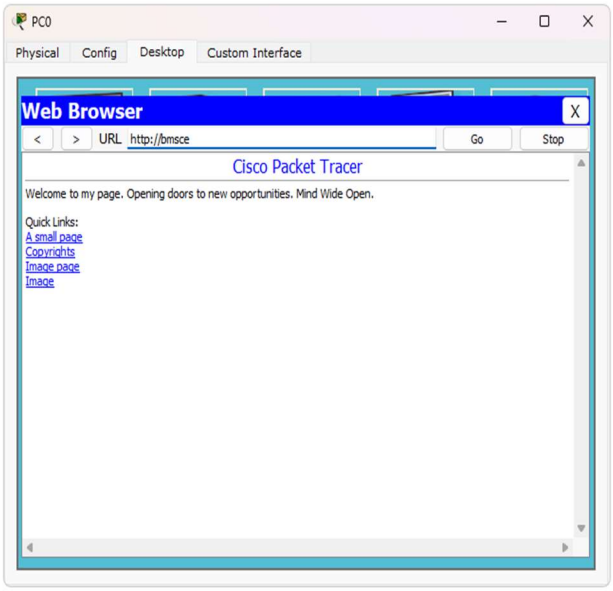
Observation:

- The server's web page could be successfully accessed from the PC by entering the resolved URL
- The DNS server could hence be configured within a LAN by enabling the DNS

30/12/19

Screen Shots:

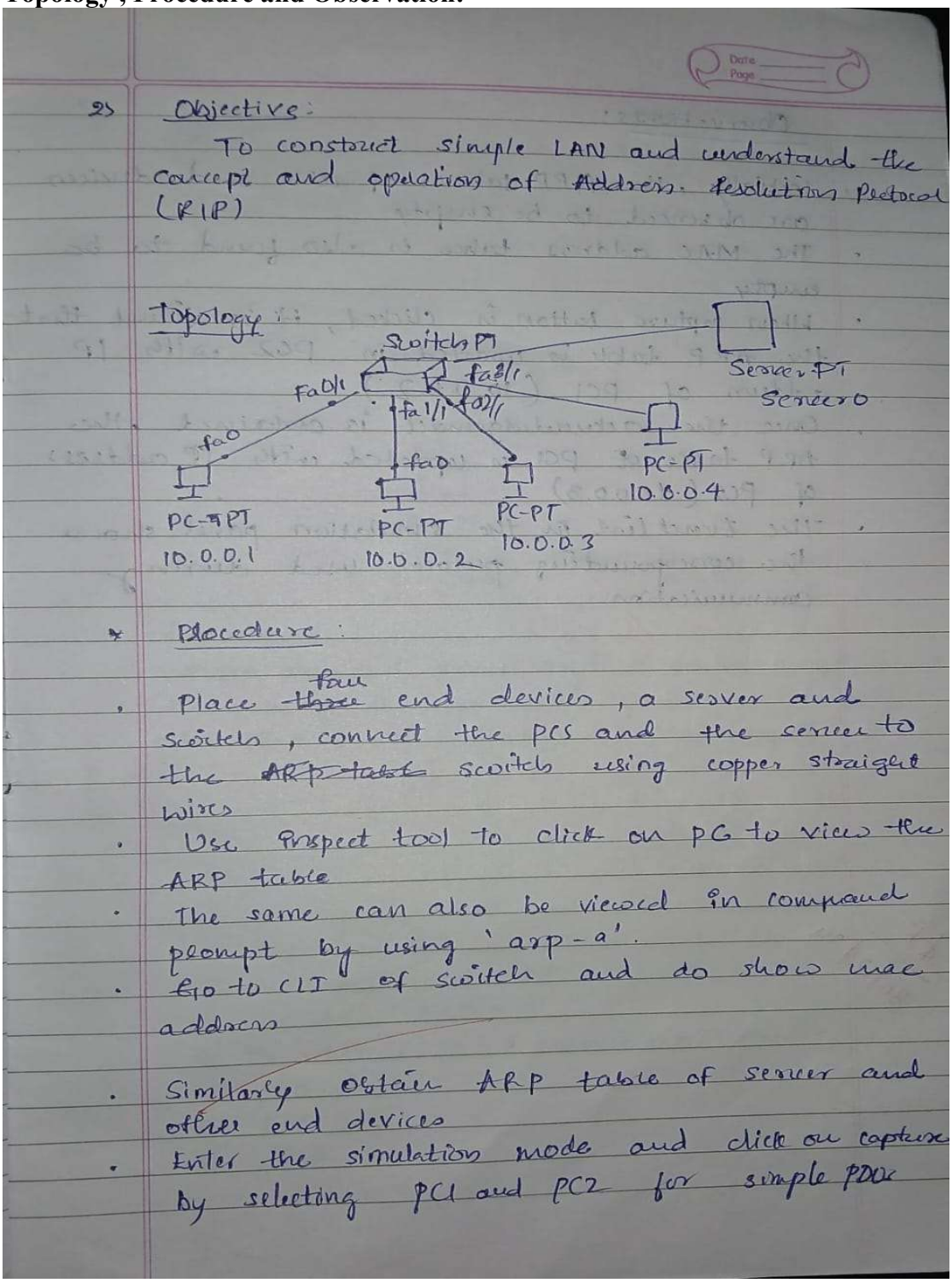




Program 10

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

Topology , Procedure and Observation:



25 Objective:

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

Topology:

Procedure:

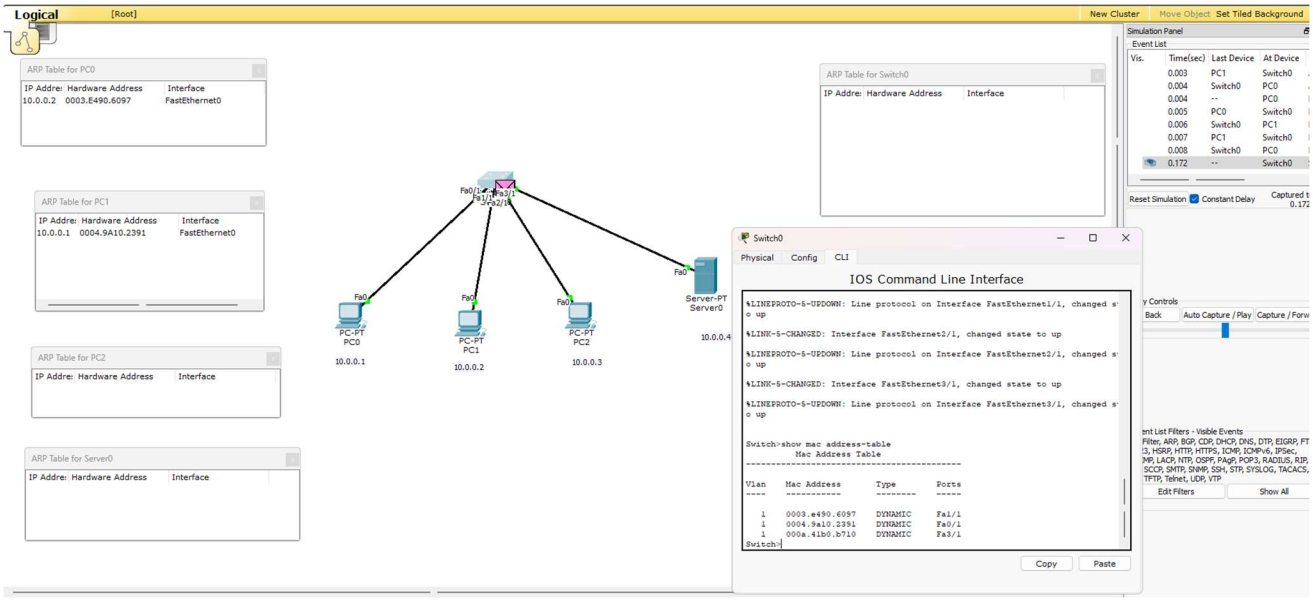
- Place ~~three~~ four end devices, a server and switch, connect the PCs and the server to the ~~ARP table~~ switch using copper straight wires
- Use Inspect tool to click on PC to view the ARP table
- The same can also be viewed in command prompt by using 'arp -a'.
- Go to CLI of switch and do show mac address
- Similarly obtain ARP table of server and other end devices
- Enter the simulation mode and click on capture by selecting PC1 and PC2 for simple PDUs

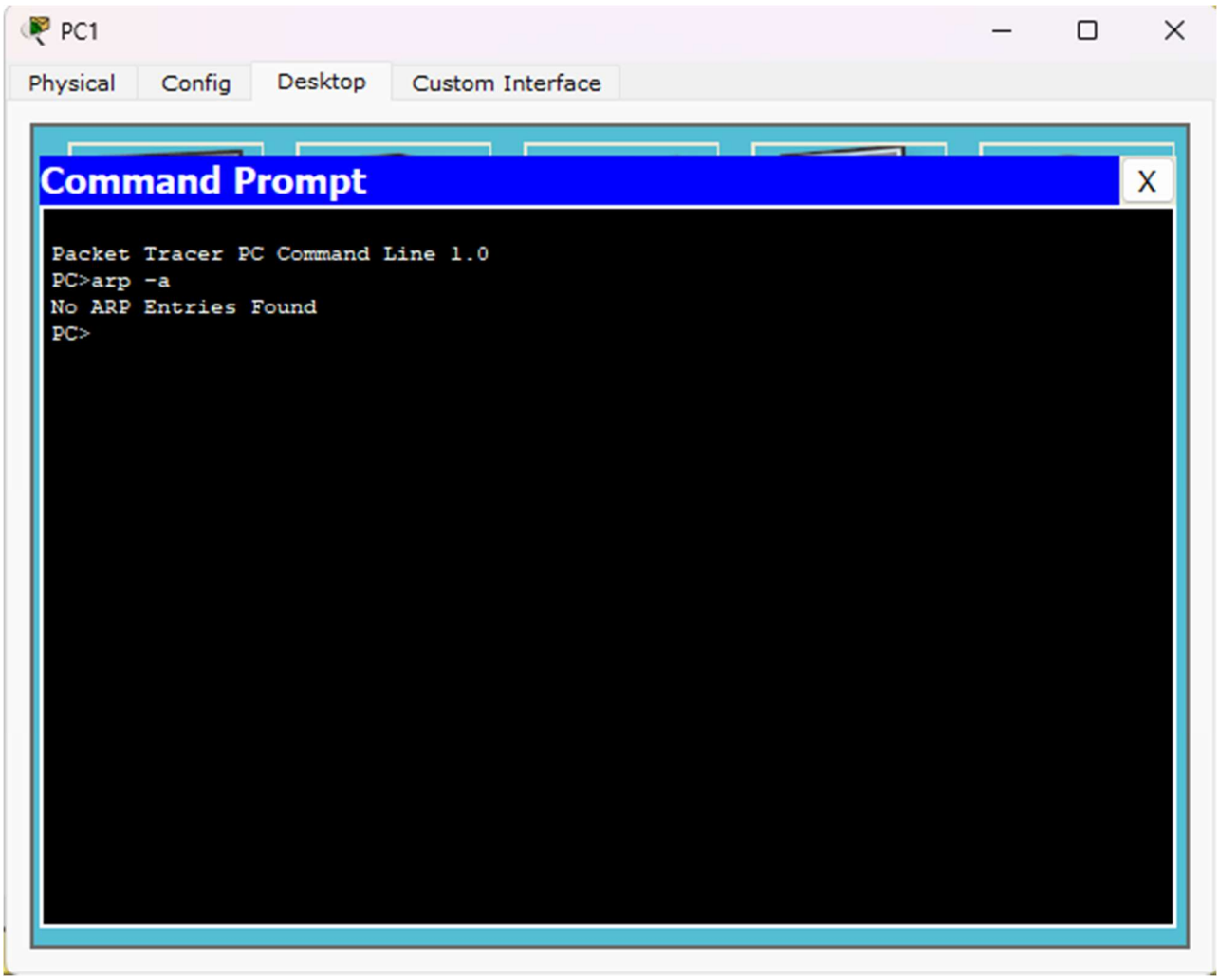
Observations:

- Initially, the ARP tables of all end devices are observed to be empty
- The MAC address table is also found to be empty
- When capture button is clicked, it is found that the ARP table is updated in PC2 with IP address of PC1 (10.0.0.2)
- Once the acknowledgement is obtained, the ARP table of PC1 is updated with IP address of PC2 (10.0.0.3)
- The Event list in the simulation panel shows the corresponding protocol used during communication

30/12/24

Screen Shots:

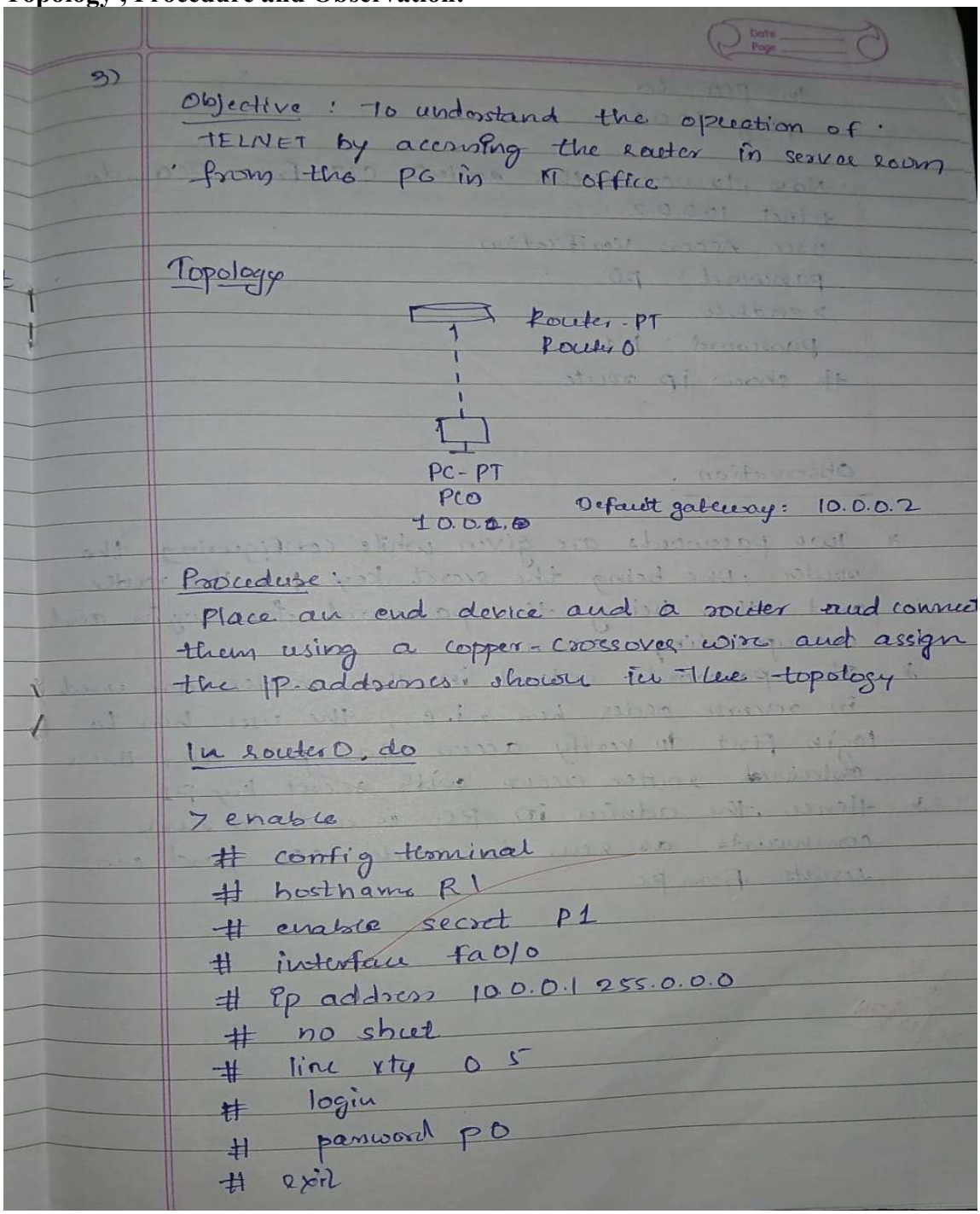



PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>arp -a
No ARP Entries Found
PC>

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:



In PC0, do
ping 10.0.0.1

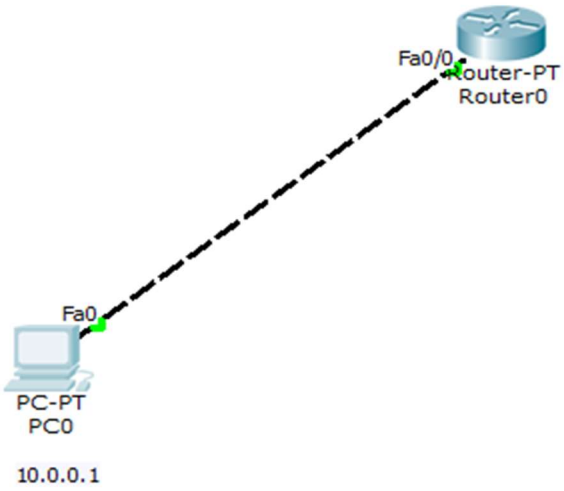
Now, to access the router0 CLI from PC0, do:
telnet 10.0.0.2
User Access Verification
password: p0
> enable
password: p1
show ip route

Observation:

* Two passwords are given while configuring the router, one being the secret key for the router and other being the password for login and corresponding access
* The passwords entered in the router are used in reverse order here, i.e., the user has to login first to verify access via p0 and then obtained router access with secret key p1
* Hence, the admin in PC is able to run commands as run in router CLI and see results from pc

30/12/24

Screen Shots:



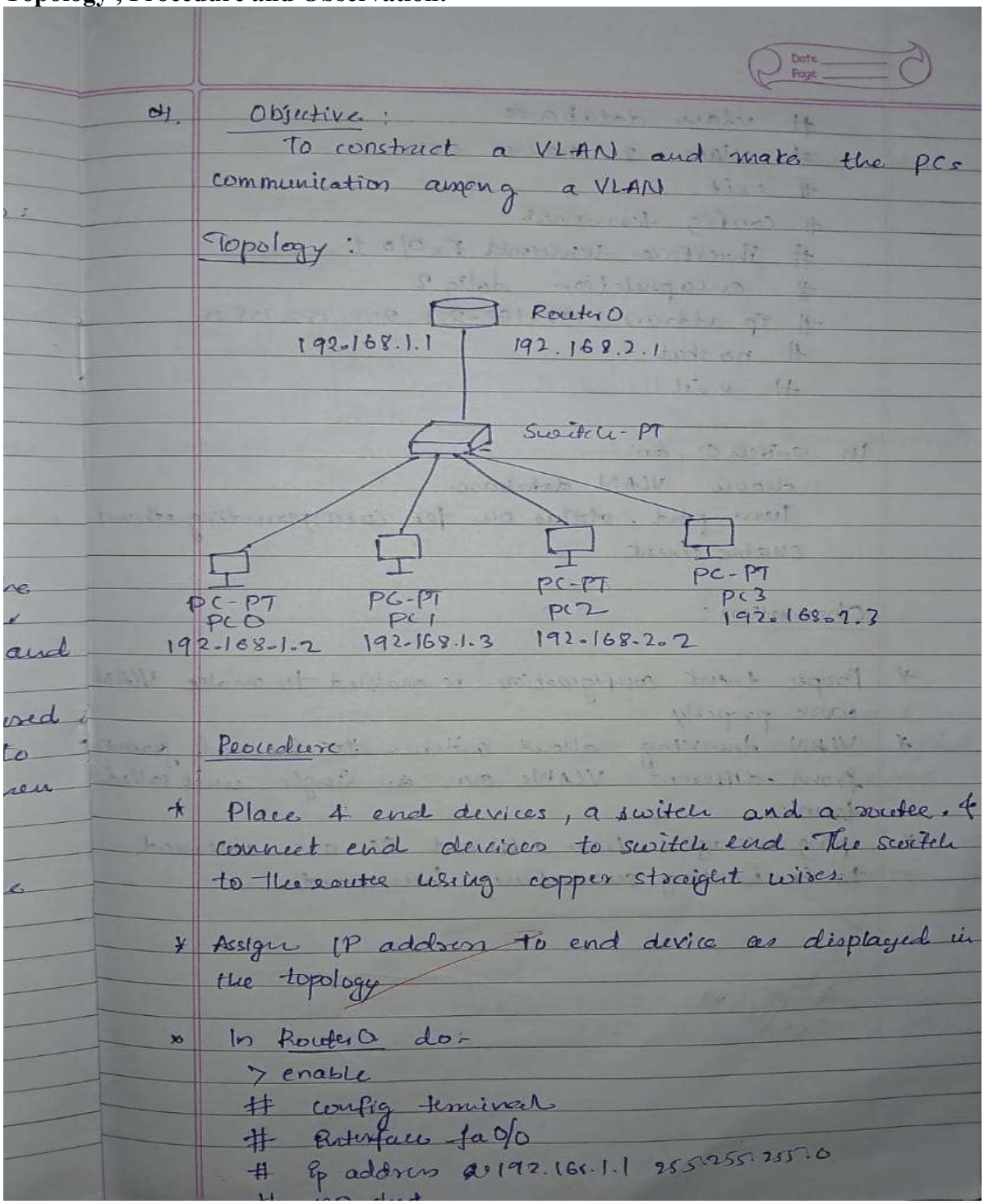
Command Prompt

```
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:



vlan database
vlan 2 name cseise
exit
config terminal
interface fa 0/0.1
encapsulation dotlq 2
ip address 192.168.2.1 255.255.255.0
no shut
exit

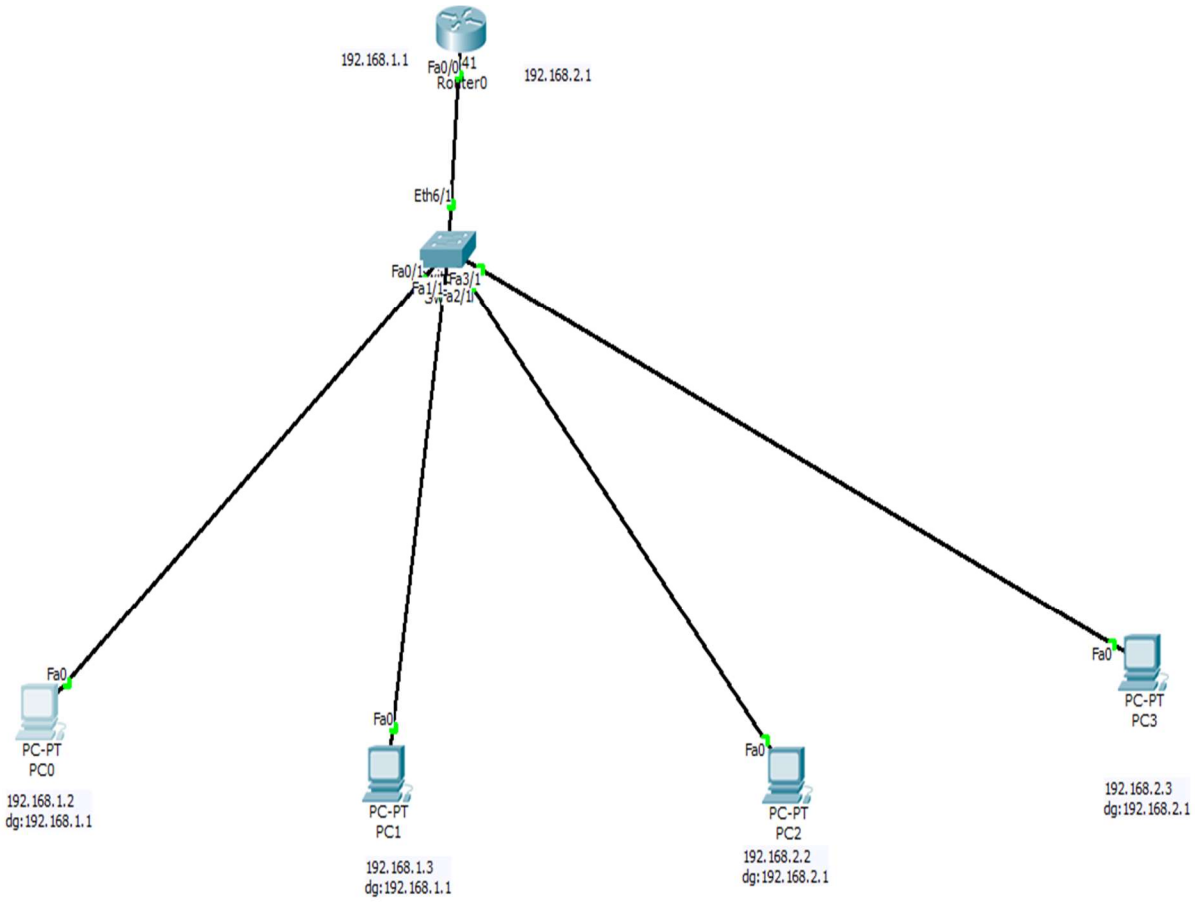
In switch 0, do:
choose VLAN database
Turn port, status on for corresponding ethernet
enable trunk

Observation:

* Proper trunk configuration is enabled to make VLAN work properly
* VLAN trunking allows switches to forward frames from different VLANs over a single unified trunk
* ping messages from different PCs are observed to be working successfully henceforth

30/12/24

Screen Shots:



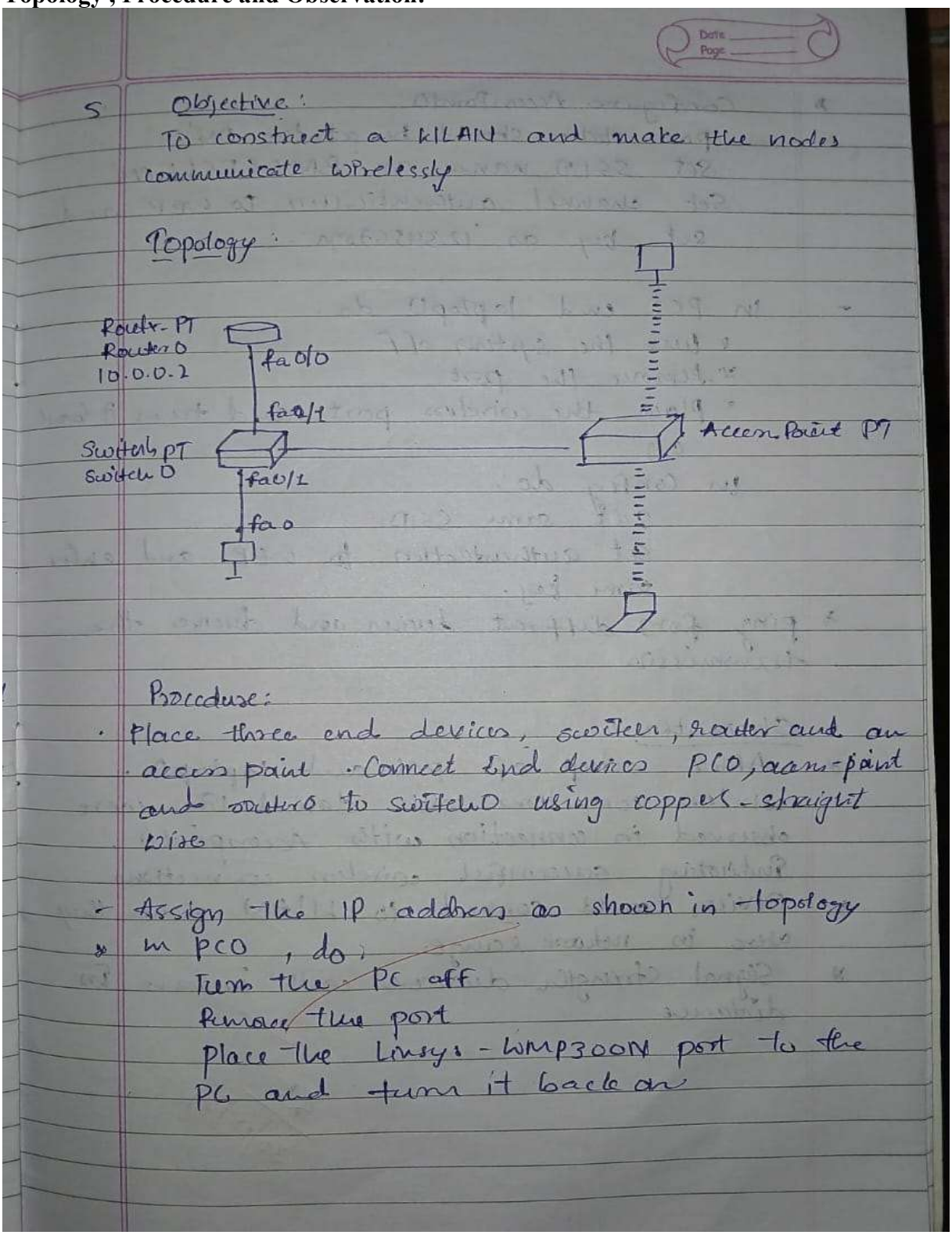
Command Prompt

```
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:



* Configure AccessPoint0
  Port status should be set to 'ON'
  Set SSID name as 'BMSCECSECN'
  Set channel authentication to WEP and
  set key as '1234567890'

- In PC1 and laptop0 do:
  * turn the system off
  * Remove the port
  * Place the wireless port and turn it back on

  In Config do:
  Set same SSID
  Set authentication to WEP and enter same key.

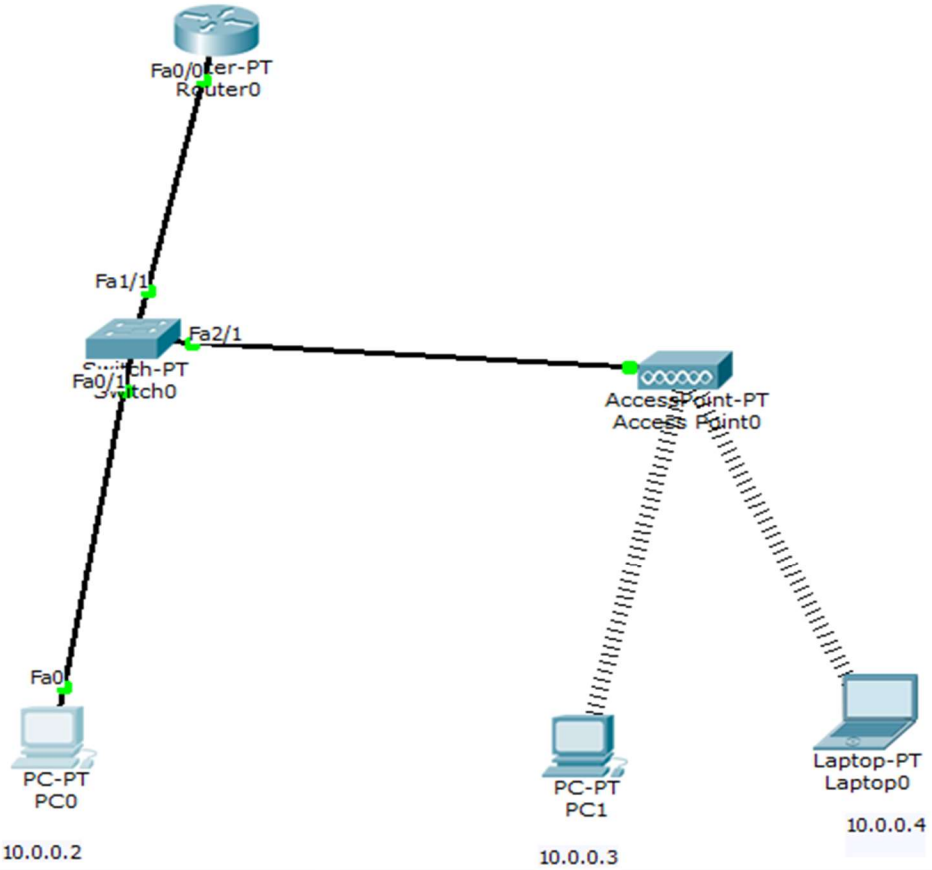
* ping from different devices and observe the transmission

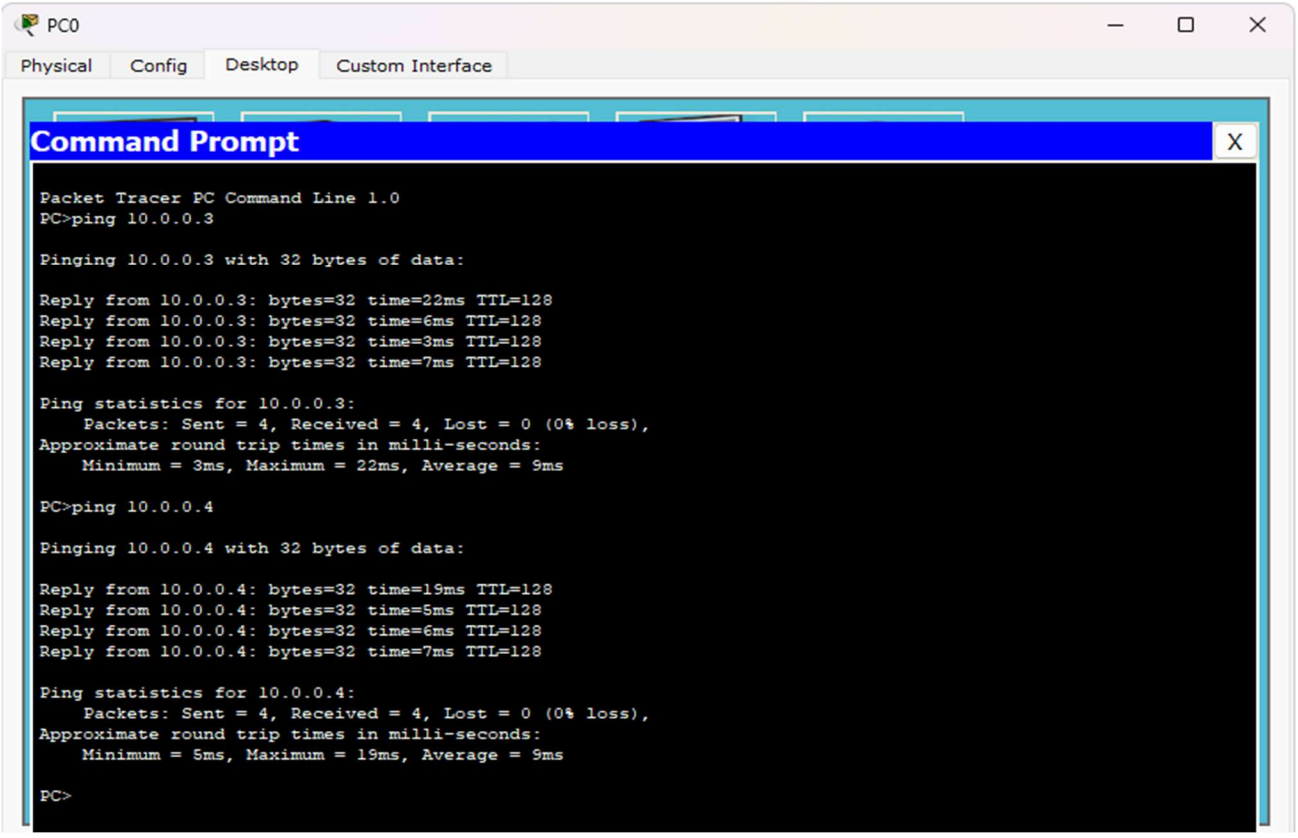
Observations:

* After setup of PC1 and laptop0, wireless connections with dashed line were observed in connection with Accesspoint0, indicating succesful wireless connections
* Devices could connect to WLAN since they were in network range
* Signal strength decreases with increase in distance

30/12/24

Screen Shots:



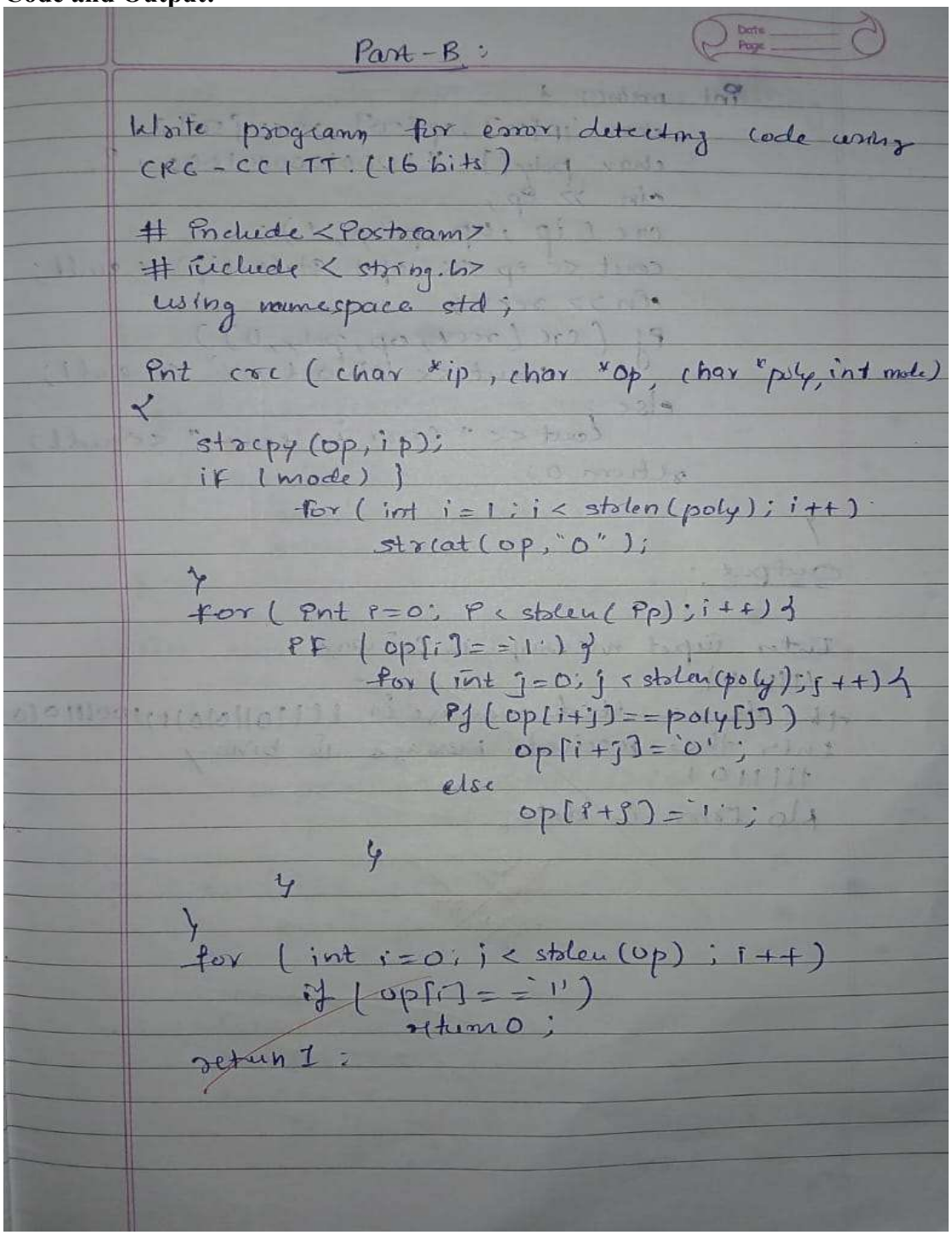


PART-B

Program 14

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

Code and Output:



Part-B :

Write 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(op); 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;
```

```
int main() {
    char ip[50], op[50], recv[50];
    char poly[] = "10001000000100001";
    cin >> ip;
    crc(ip, op, poly, 1);
    cout << ip << op+strlen(ip) << endl;
    cin >> recv;
    if (crc(recv, op, poly, 0))
        cout << "No error" << endl;
    else
        cout << "error in trans." << endl;
    return 0;
}
```

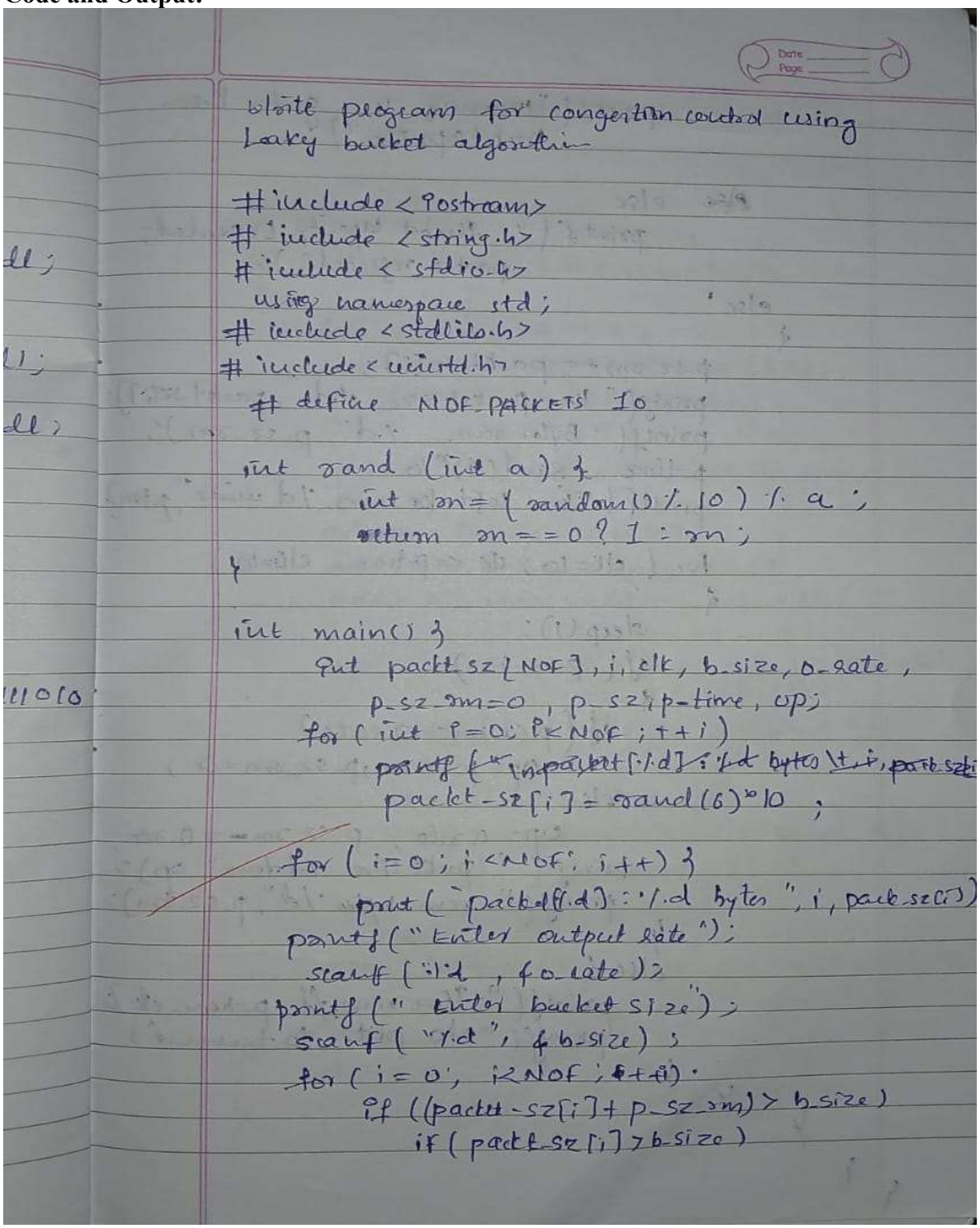
Output:

Enter input message in binary:
1111101
The transmitted message is 11111011010111100111010
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:



```
write program for congestion control using
Leaky bucket algorithm

#include <iostream>
#include <string.h>
#include <stdio.h>
using namespace std;
#include <stdlib.h>
#include <time.h>
#define NOF_PACKETS 10

int rand (int a) {
    int rn = (random() % 10) % a;
    return rn == 0 ? 1 : rn;
}

int main() {
    int packt_sz[NOF], i, clk, b_size, o_rate,
        p_sz_rm=0, p_sz, p_time, op;
    for (int i=0; i<NOF; ++i)
        packet_sz[i] = rand(6)*10;

    for (i=0; i<NOF; i++) {
        printf("packet[%d]: %d bytes ", i, pack_sz[i]);
    printf("Enter output rate");
    scanf("%d", &o_rate);
    printf("Enter bucket size");
    scanf("%d", &b_size);
    for (i=0; i<NOF; i++)
        if ((packt_sz[i] + p_sz_rm) > b_size)
            if (packt_sz[i] > b_size)
```

```
        printf("Incomming is greater than
                bucket capacity");

    else else
        printf("Bucket capacity Exceeded,
                packet rejected");
else
{
    p_sz_rm += packt_sz[i];
    printf("In. packet size %d", packt_sz[i]);
    printf("Bytes rem. %d", p_sz_rm);
    p_time = rand(4) * 10
    printf("Time left for trans. %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 = = o_rate;
            printf("packt size %d trans", op);
            printf("rem. bytes %d", p_sz_rm);
        }
        else {
            printf("Time rem: %d p_time_clk");
            printf("No pkts to transmit")
        }
    }
}
```

9. Using TCP/IP sockets, write client-server program

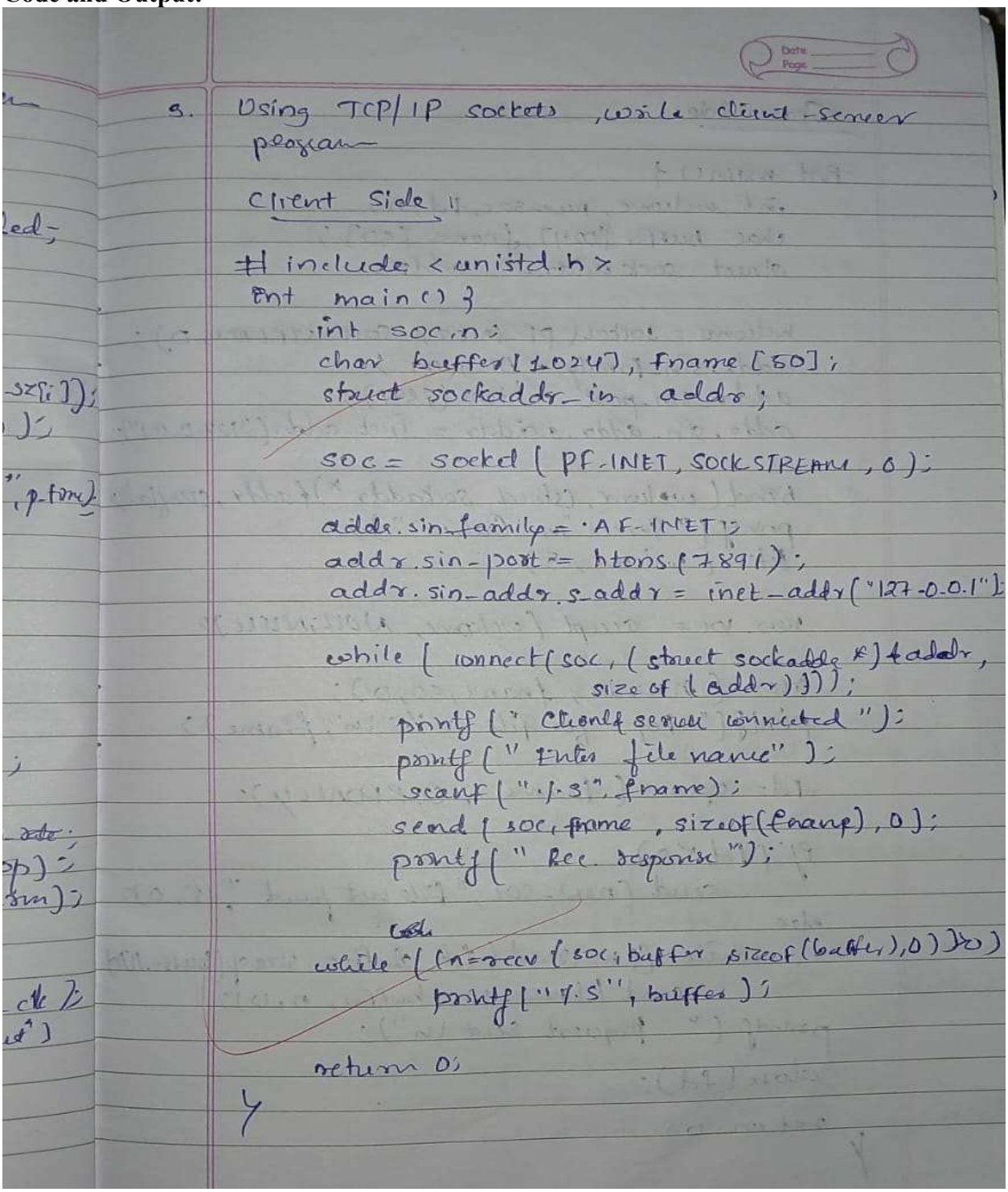
Client Side :

```
#include <unistd.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("Client server connected");
    printf("Enter file name");
    scanf("%s", fname);
    send(soc, fname, sizeof(fname), 0);
    printf("Rec. response");

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

    return 0;
}
```

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:



9. Using TCP/IP sockets, write client-server program

Client Side :

```
#include <unistd.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("Client socket connected");
        printf("Enter file name");
        scanf("%s", fname);
        send(soc, fname, sizeof(fname), 0);
        printf("Rec. response");

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

    return 0;
}
```

Server Side:

```
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("Server in Online");
    listen(welcome, 5);

    new_soc = accept(welcome, NULL, NULL);

    recv(new_soc, fname, 50, 0);
    printf("Req. for file : %s \n", fname);

    fd = open(fname, O_RDONLY);

    if(fd < 0)
        send(new_soc, "file not found", 15, 0);
    else
        while((n = read(fd, &buffer, sizeof(buffer))) > 0)
            send(new_soc, buffer, n, 0);
    printf("Request sent \n");
    close(fd);

    return 0;
}
```

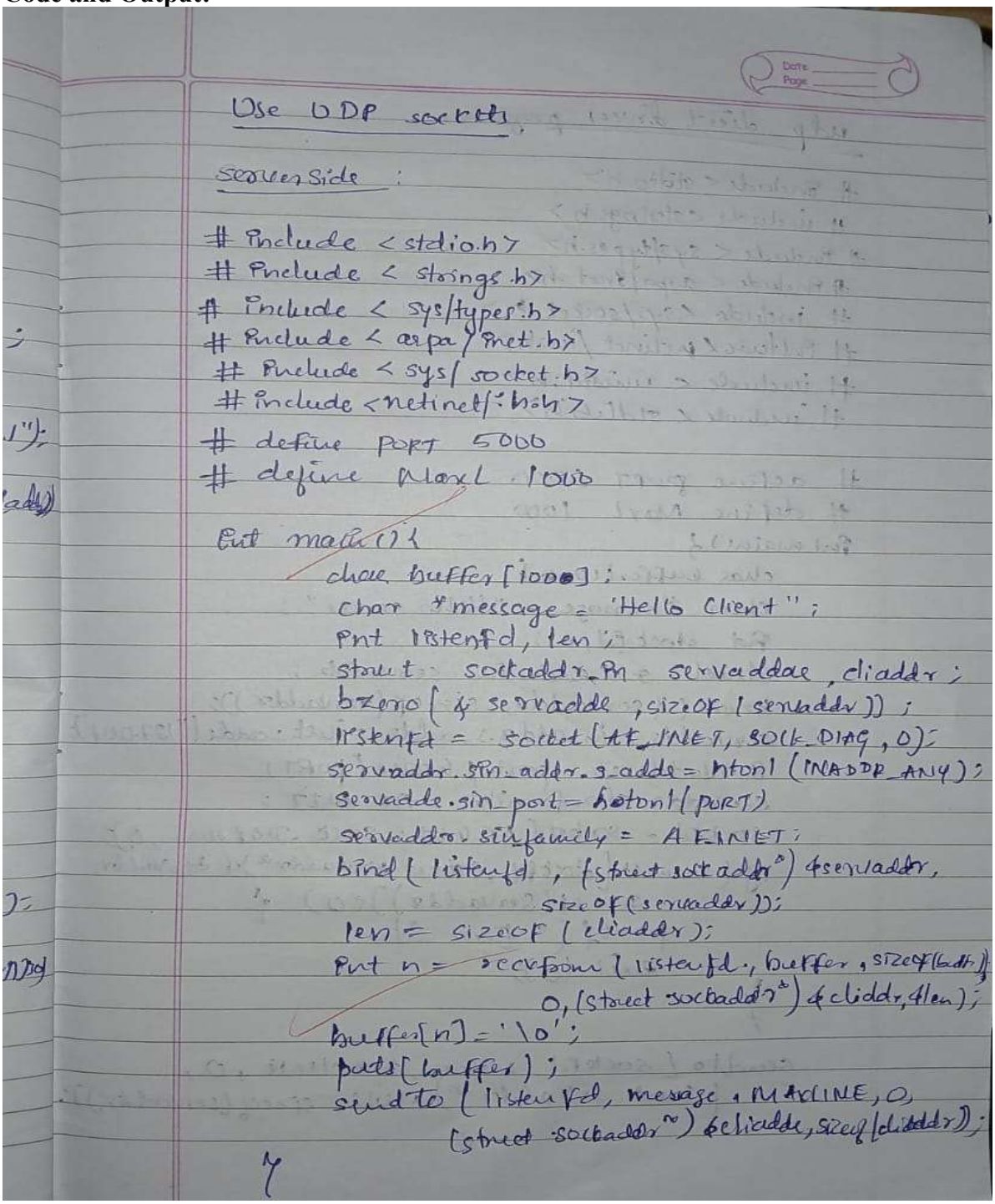
Output:

```
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.

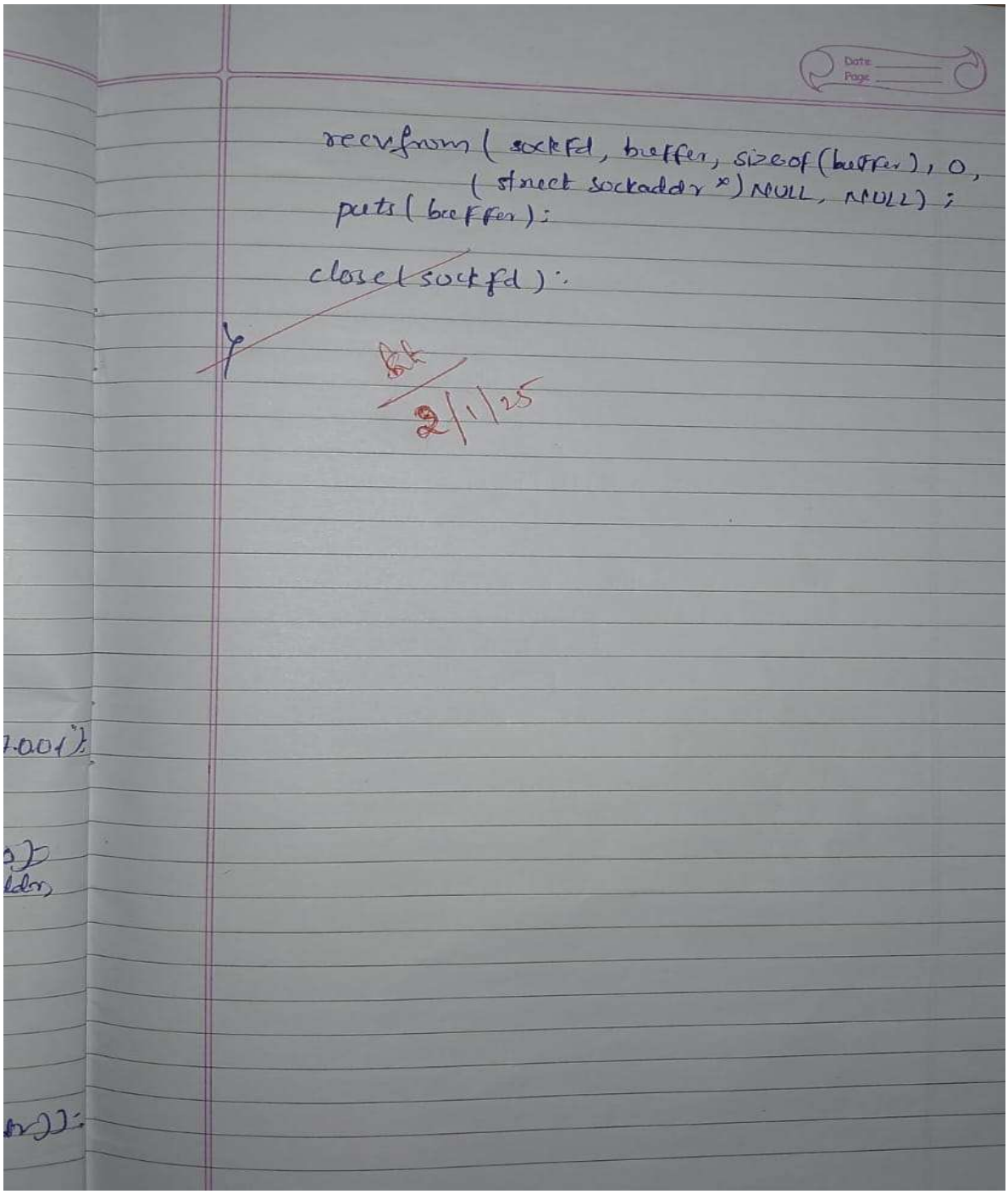
Code and Output:



```
Use UDP sockets.

Server Side :

#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 MAXL 1000

int main(){
    char buffer[1000];
    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);
    int n = recvfrom(listenfd, buffer, sizeof(buffer),
                     0, (struct sockaddr*)&cliaddr, &len);
    buffer[n] = '\0';
    puts(buffer);
    sendto(listenfd, message, MAXLINE, 0,
           (struct sockaddr*)&cliaddr, sizeof(cliaddr));
}
```

udp direct driver program

```
#include <stdio.h>
#include <strings.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 MAXL 1000
int main(){
    char buffer[1000];
    char *message = "Hello server";
    int sockfd, n;
    struct sockaddr_in servaddr;
    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");
    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("\nError");
        exit(0);
    }

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



Server Output:
Server is online
Hello Server

Client Output:
Hello Client