VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB RECORD

Computer Network Lab (23CS5PCCON)

Submitted by

SriKrishna Pejathaya P S (1BM22CS290)

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 **SriKrishna Pejathaya P S (1BM22CS290),** 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 Associate Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE

Index

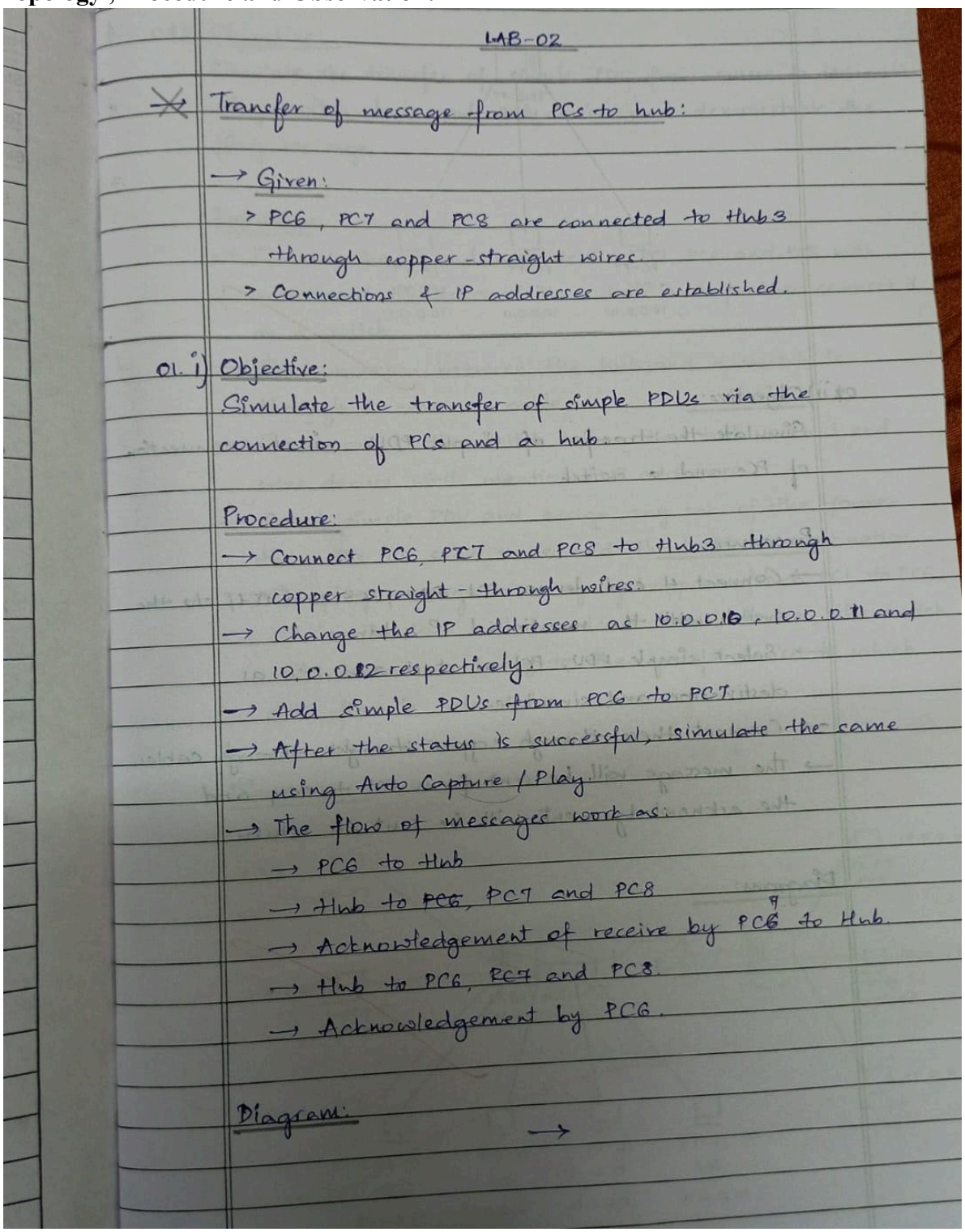
Sl. No.	Date	Experiment Title	Page No.
1	09-10-24	Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping messages.	4-8
2	09-10-24	Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply.	9-11
3	16-10-24	Configure default route, static route to the Router (Part 1).	12-15
4	23-10-24	Configure default route, static route to the Router (Part 2).	16-20
5	13-11-24	Configure DHCP within a LAN and outside LAN.	21-26
6	20-11-24	Configure RIP routing Protocol in Routers .	27-30
7	20-11-24	Demonstrate the TTL/ Life of a Packet.	31-33
8	27-11-24	Configure OSPF routing protocol.	34-37
9	18-12-24	Configure Web Server, DNS within a LAN.	38-39
10	18-12-24	To construct a simple LAN and understand the concept and operation of Address Resolution Protocol (ARP).	40-42
11	18-12-24	To understand the operation of TELNET by accessing the router in the server room from a PC in the IT office.	43-45
12	18-12-24	To construct a VLAN and make the PC's communicate among a VLAN.	46-49
13	18-12-24	To construct a WLAN and make the nodes communicate wirelessly.	50-52
14	18-12-24	Write a program for error detecting code using CRC-CCITT (16-bits).	53-54
15	18-12-24	Write a program for congestion control using Leaky bucket algorithm.	55-58
16	18-12-24	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.	59-61
17	18-12-24	Using UDP sockets, write a client-server program to make the client sending the file name and the server to send back the contents of the requested file if present.	62-64

Github Link: 1BM22CS290 - CN

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:



LAB-02

✳ Transfer of message from PCs to hub:

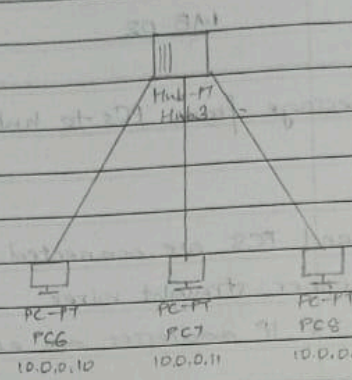
→ Given:
> PC6, PC7 and PC8 are connected to Hub3 through copper-straight wires.
> Connections & IP addresses are established.

01. i) Objective:
Simulate the transfer of simple PDUs via the connection of PCs and a hub.

Procedure:
→ Connect PC6, PC7 and PC8 to Hub3 through copper straight-through wires.
→ Change the IP addresses as 10.0.0.10, 10.0.0.11 and 10.0.0.12 respectively.
→ Add simple PDUs from PC6 to PC7.
→ After the status is successful, simulate the same using Auto Capture / Play.
→ The flow of messages work as:
  → PC6 to Hub
  → Hub to PC6, PC7 and PC8
  → Acknowledgement of receive by PC6 to Hub.
  → Hub to PC6, PC7 and PC8.
  → Acknowledgement by PC6.

Diagram:
→

PAGE NO :
DATE :



ii) Objective:

Simulate the transfer of simple PDUs via the connection of PCs and a switch

Procedure:

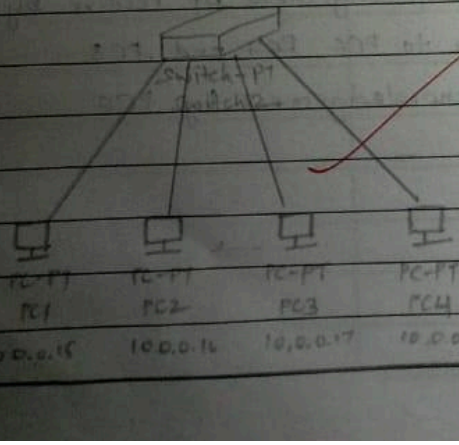
→ Connect 4 end devices PC1, PC2, PC3 and PC4 to the switch with the mentioned IP address

→ Select simple PDU, PC1 as start and PC4 as destination and simulate.

→ Connect these through copper straight-through cables.

→ The message will be sent from PC1 to PC4 and the acknowledgement will be sent in return.

Diagram:



PAGE NO : 03
DATE :

04(iii) Objective:

Simulate the transfer of simple PDU from source to destination using switch, hub and end devices and demonstrate the ping message.

Procedure:

→ Connect the 3 end user devices PC1, PC2 and PC3 with mentioned IP addresses to a hub and further connect it to a switch.

→ The connection between the hub and a switch is through a crossover cable.

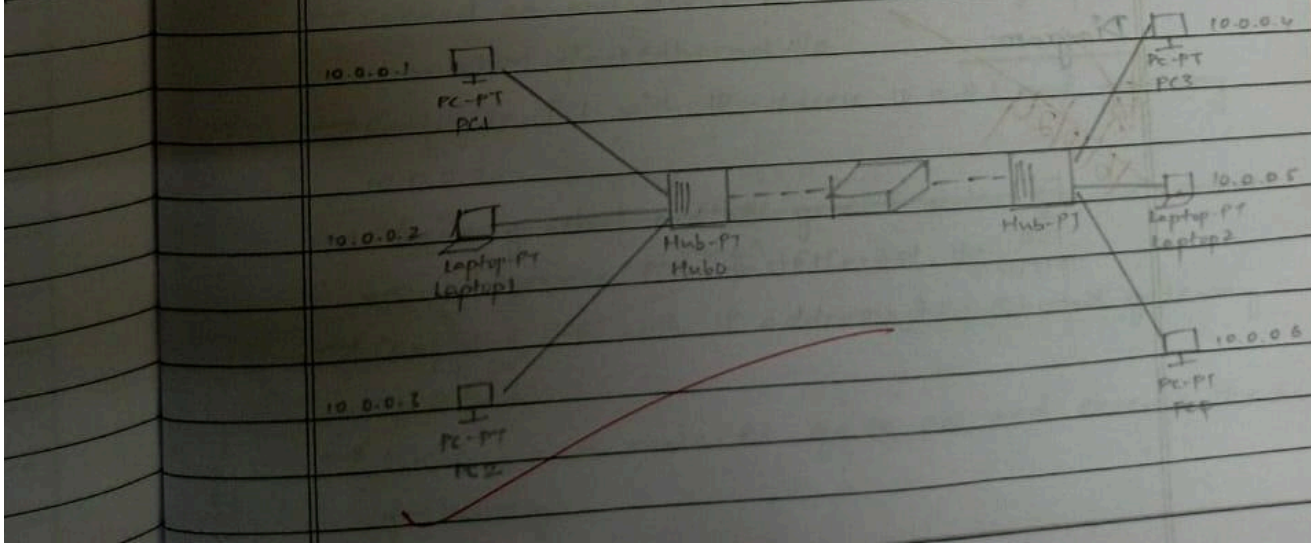
→ Then connect the switch to another hub with 3 end user devices with mentioned IP addresses.

→ Select simple PDU and assign any one of the former three PCs as start node and the latter as destination.

→ The ping message successfully happens from PC1 to PC6.

→ While the hub will broadcast to all devices, the switch will communicate only to the specific device to which data flow is intended.

Diagram:



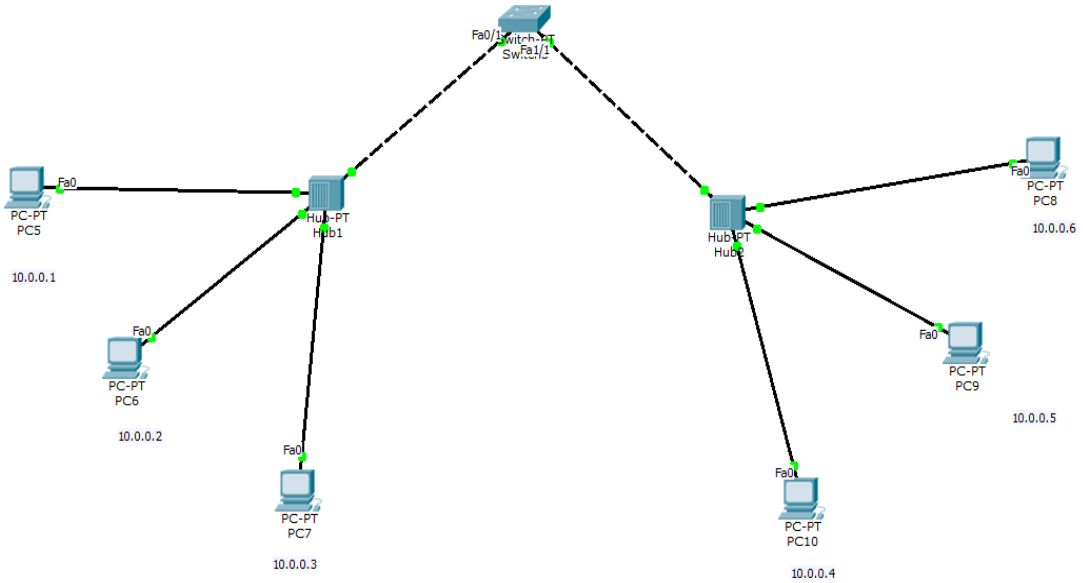
PAGE NO :
DATE :

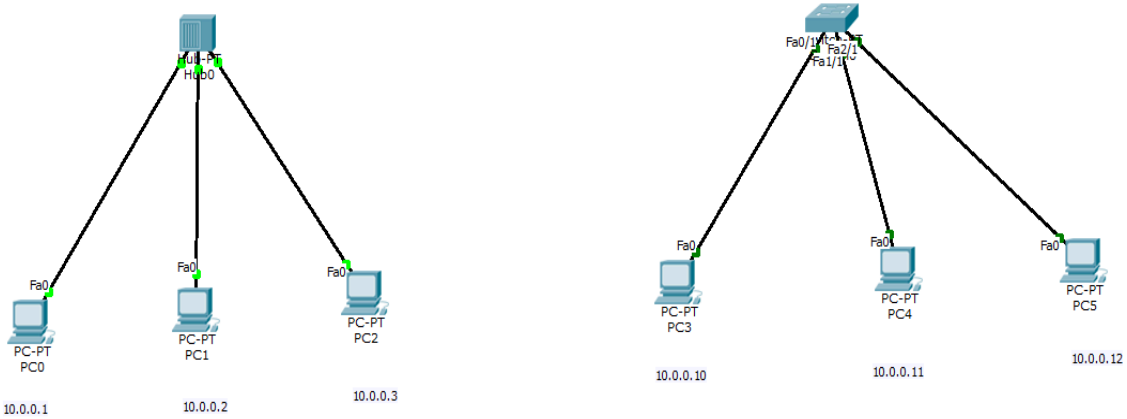
Router > enable
Router # configure terminal
Router (config) # interface fastEthernet 0/0
Router (config-if) # ip address 10.0.0.2 255.0.0.0
Router (config-if) # no shutdown
→ Interface fastEthernet 0/0, changed state to up
→ Similarly select router R2, go to CLI and execute.
Router > enable
Router # 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.

→ The connections between the routers and end devices is established.
→ Connect R1 with R2 using serial cable. To setup connection between routers again; go
→ Select router R1 and go to CLI
Router (config) # interface serial 2/0
Router (config-if) # ip address 30.0.0.1 255.0.0.0
Router (config-if) # no shutdown
→ Select router R2 and go to CLI and enable similar commands
→ Interface serial 2/0 changed state to up.
3/0

Observations:
→ After setting up the mentioned topology, an attempt was made to ping PC2 from PC1.
→ In the command prompt of PC1, type ping 20.0.0.1
→ Destination host unreachable.
Packets sent: 4 received: 0 lost: 4 loss: 100%.

Screen Shots:

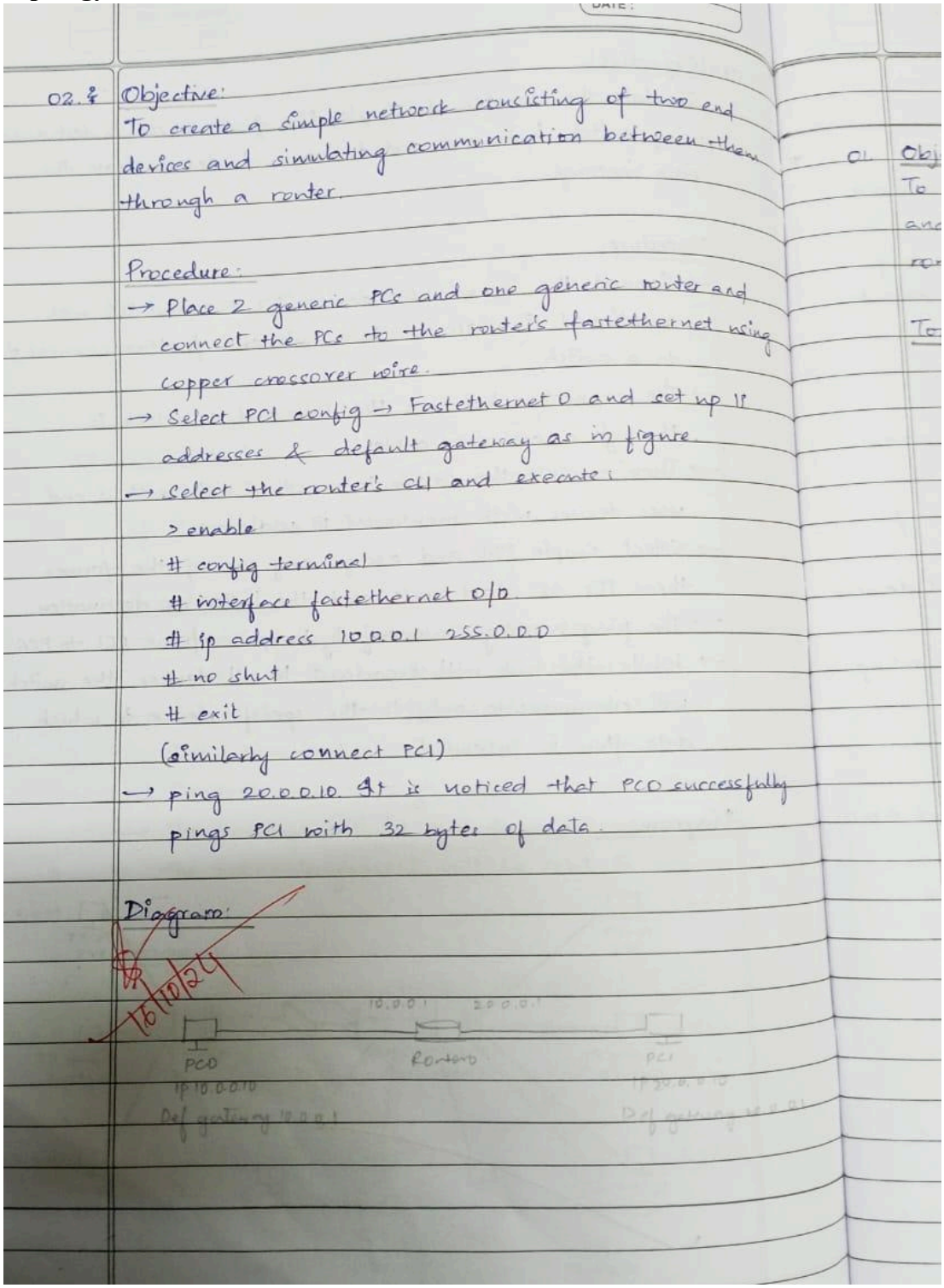




Program 2

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:



02. Objective:
To create a simple network consisting of two end devices and simulating communication between them through a router.

Procedure:
→ Place 2 generic PCs and one generic router and connect the PCs to the router's fastethernet using copper crossover wire.
→ Select PC1 config → Fastethernet 0 and set up IP addresses & default gateway as in figure
→ Select the router's CLI and execute:
> enable
config terminal
interface fastethernet 0/0
ip address 10.0.0.1 255.0.0.0
no shut
exit
(similarly connect PC1)
→ ping 20.0.0.10. It is noticed that PC0 successfully pings PC1 with 32 bytes of data.

Diagram:

16/10/24

PC0
Router0
PC1

PAGE NO : 07
DATE :

→ It is also observed that the end system PC1 was only pinged with the following IPs

ping 30.0.0.1

ping 10.0.0.2

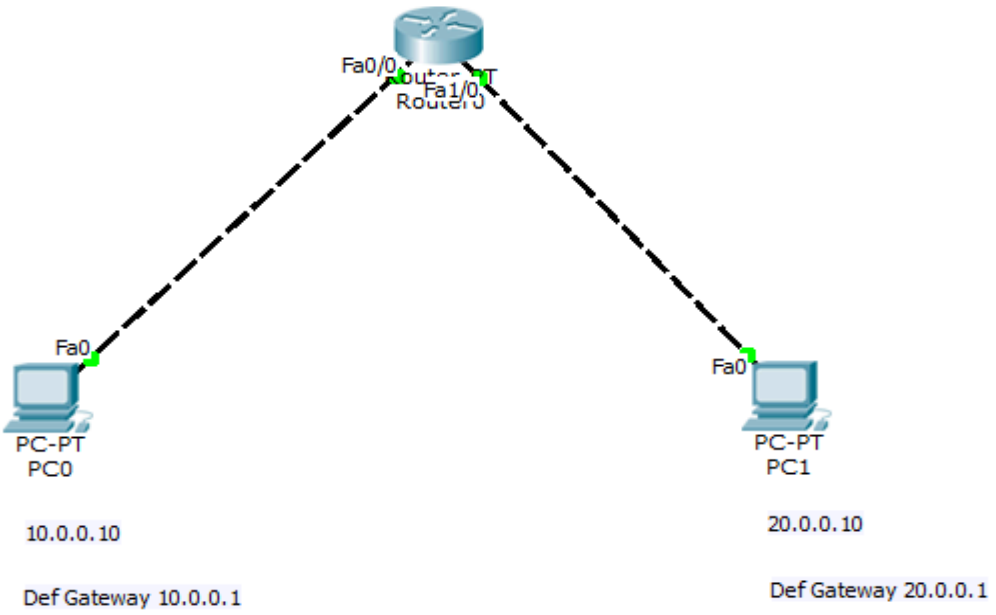
⇒ ping ~~30.1~~ successful

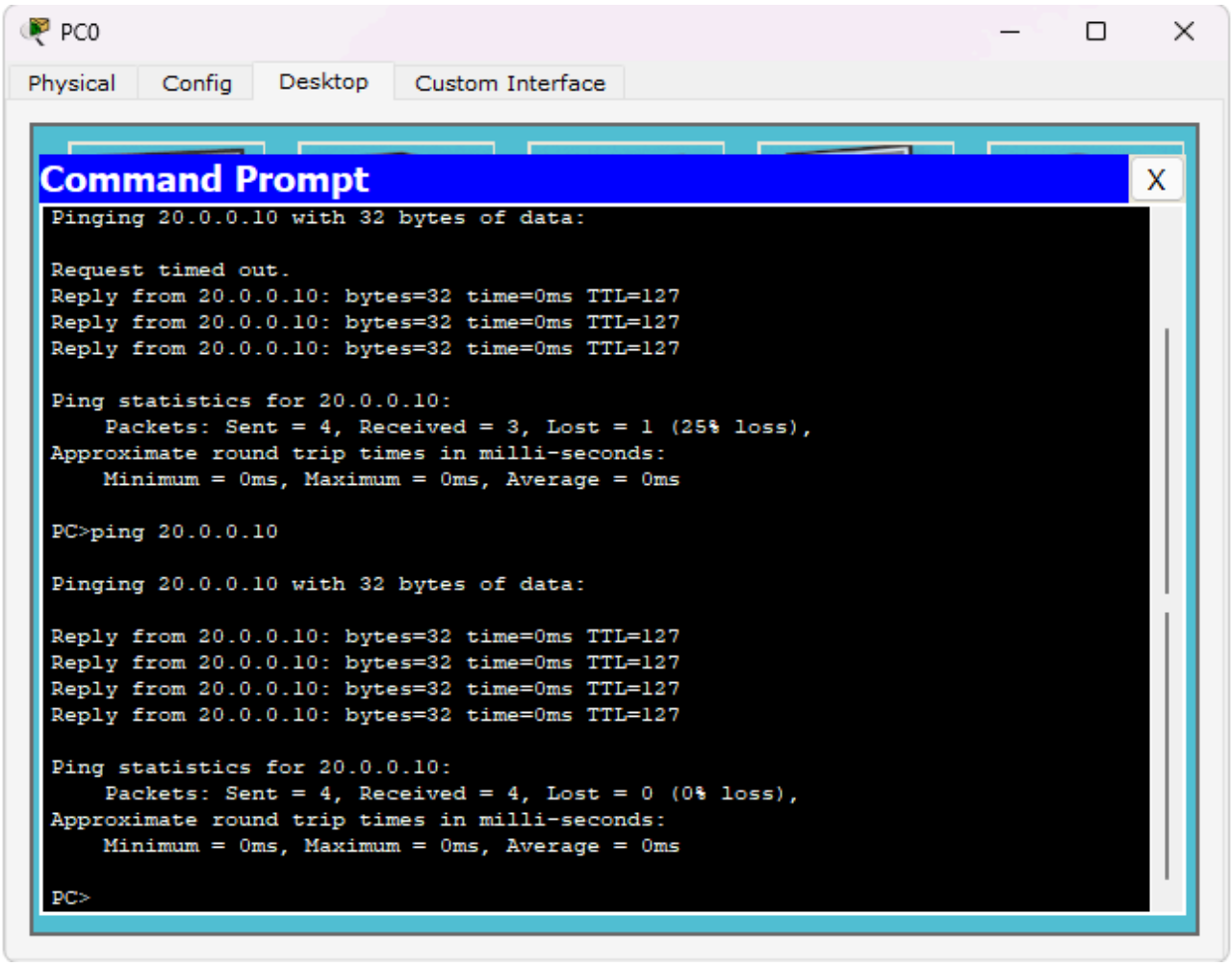
Packets sent:4 received:4 lost:0 loss = 0.0%.

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

23/10/24

Screen Shots:

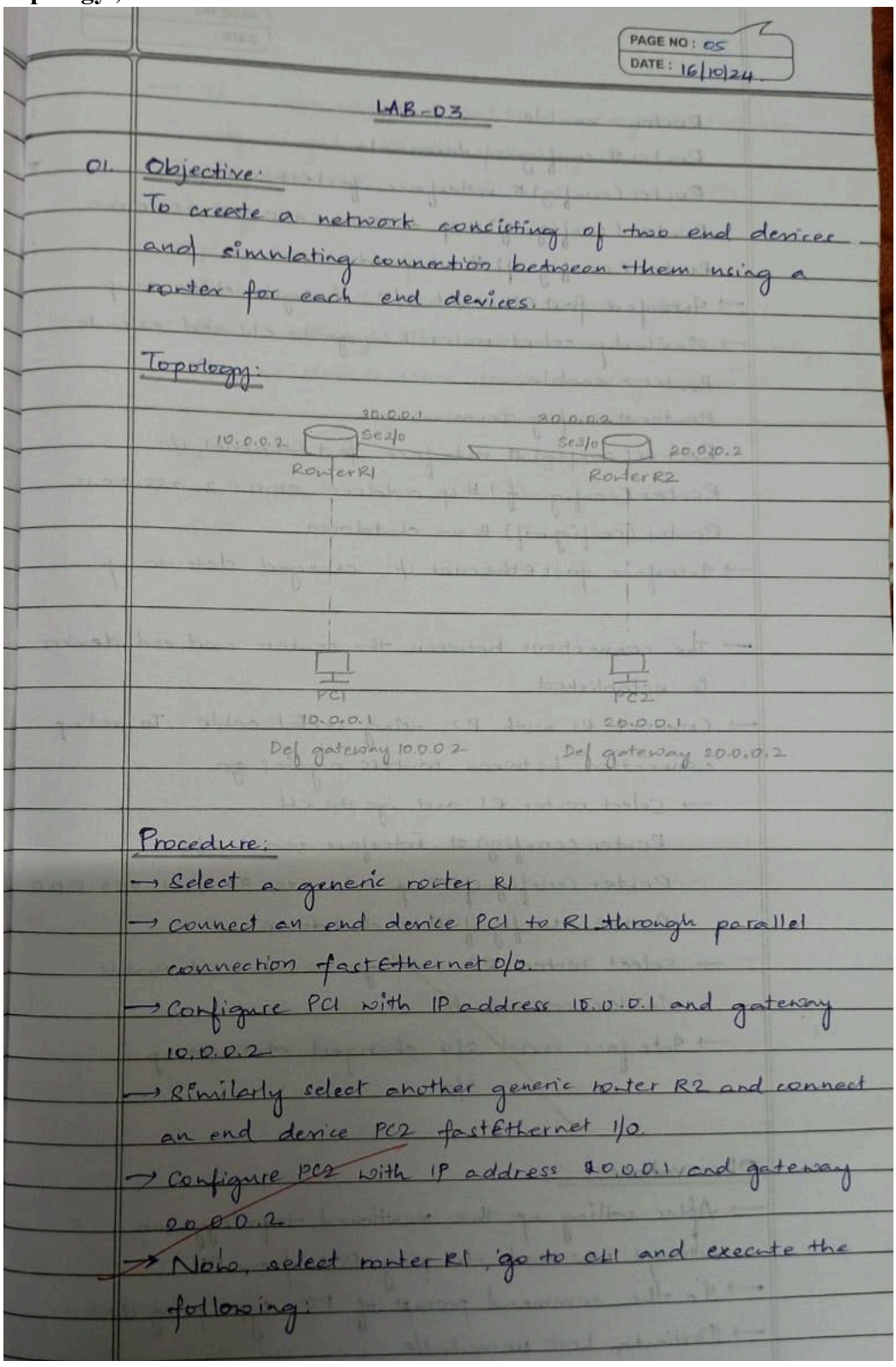



PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Pinging 20.0.0.10 with 32 bytes of data:
Request timed out.
Reply from 20.0.0.10: bytes=32 time=0ms TTL=127
Reply from 20.0.0.10: bytes=32 time=0ms TTL=127
Reply from 20.0.0.10: bytes=32 time=0ms TTL=127
Ping statistics for 20.0.0.10:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>ping 20.0.0.10
Pinging 20.0.0.10 with 32 bytes of data:
Reply from 20.0.0.10: bytes=32 time=0ms TTL=127
Reply from 20.0.0.10: bytes=32 time=0ms TTL=127
Reply from 20.0.0.10: bytes=32 time=0ms TTL=127
Reply from 20.0.0.10: bytes=32 time=0ms TTL=127
Ping statistics for 20.0.0.10:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>

Program 3

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

Topology , Procedure and Observation:



PAGE NO : 05
DATE : 16/10/24

LAB-03

01. Objective:
To create a network consisting of two end devices and simulating connection between them using a router for each end devices.

Topology:

Procedure:
→ Select a generic router R1
→ Connect an end device PC1 to R1 through parallel connection fastEthernet 0/0.
→ Configure PC1 with IP address 10.0.0.1 and gateway 10.0.0.2
→ Similarly select another generic router R2 and connect an end device PC2 fastEthernet 1/0.
→ Configure PC2 with IP address 20.0.0.1 and gateway 20.0.0.2
→ Now, select router R1, go to cli and execute the following:

PAGE NO :
DATE :

ip address 40.0.0.1 255.0.0.0
no shut
"Interface FastEthernet 1/0, changed state to up"
exit

→ To set up Static routing in router R1, do

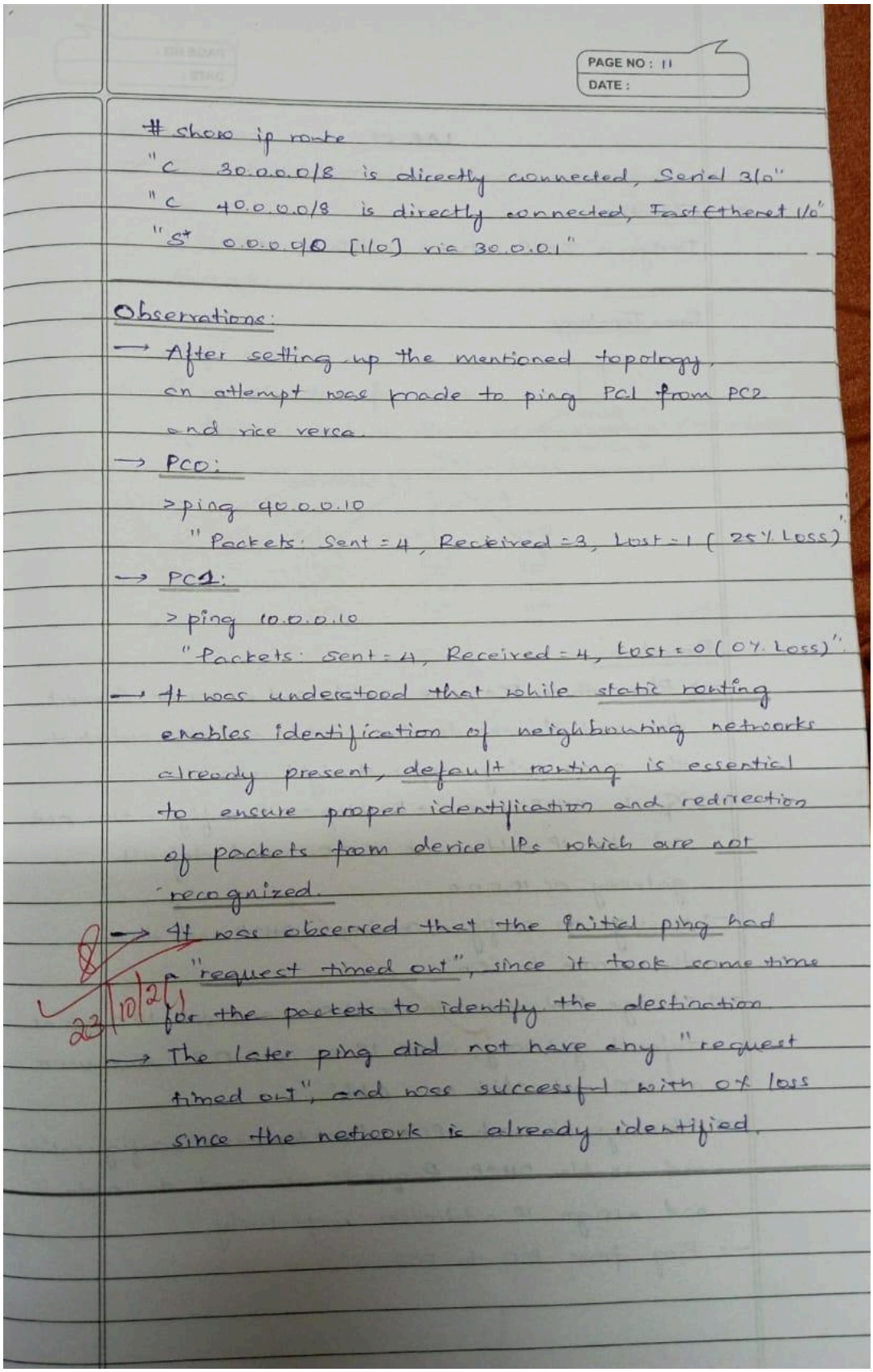
> enable
show ip route
"C 20.0.0.0/8 is directly connected, Serial 2/0"
"C 30.0.0.0/8 is directly connected, Serial 3/0"
config terminal
ip route 10.0.0.0 255.0.0.0 20.0.0.1
ip route 40.0.0.0 255.0.0.0 30.0.0.2
exit
show ip route
"S 10.0.0.0/8 [1/0] via 20.0.0.1"
"C 20.0.0.0/8 is directly connected, Serial 2/0"
"C 30.0.0.0/8 is directly connected, Serial 3/0"
"S 40.0.0.0/8 via 30.0.0.2"

→ To set up Default routing in R0, do:

> enable
config terminal
ip route 0.0.0.0 0.0.0.0 20.0.0.2
exit
show ip route
"C 10.0.0.0/8 is directly connected, FastEthernet 0/0"
"C 20.0.0.0/8 is directly connected, Serial 2/0"
"S* 0.0.0.0/0 [1/0] via 20.0.0.2"

→ To set up Default routing in R2, do:

> enable
config terminal
ip route 0.0.0.0 0.0.0.0 30.0.0.1
exit



PAGE NO : 11
DATE :

show ip route
"C 30.0.0.0/8 is directly connected, Serial 3/0"
"C 40.0.0.0/8 is directly connected, FastEthernet 1/0"
"S* 0.0.0.0/0 [1/0] via 30.0.0.1"

Observations:

→ After setting up the mentioned topology, an attempt was made to ping PC1 from PC2 and vice versa.

→ PC0:

>ping 40.0.0.10
"Packets: Sent = 4, Received = 3, Lost = 1 (25% Loss)"

→ PC1:

>ping 10.0.0.10
"Packets: Sent = 4, Received = 4, Lost = 0 (0% Loss)"

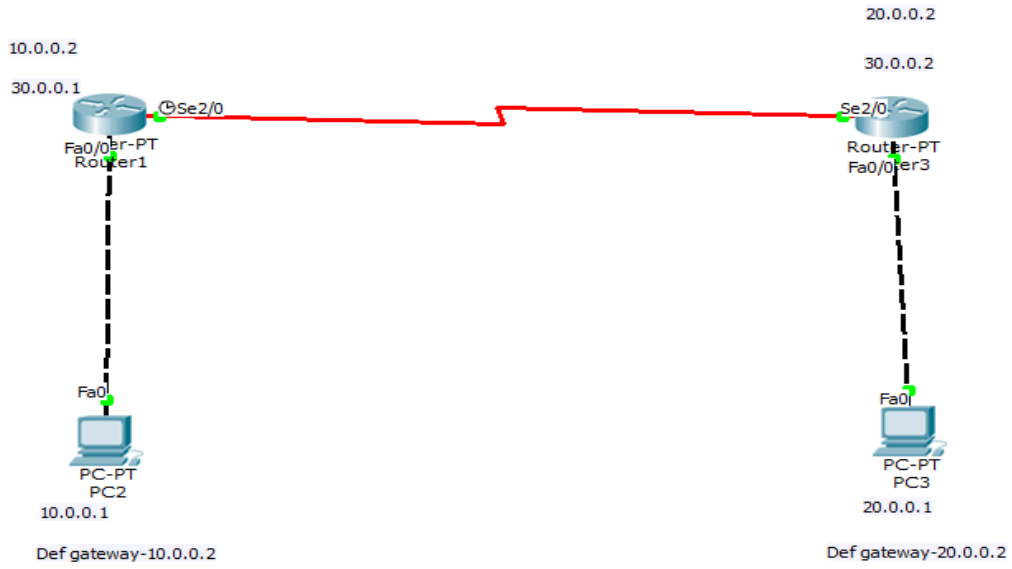
→ It was understood that while static routing enables identification of neighbouring networks already present, default routing is essential to ensure proper identification and redirection of packets from device IPs which are not recognized.

→ It was observed that the initial ping had "request timed out", since it took some time for the packets to identify the destination.

→ The later ping did not have any "request timed out", and was successful with 0% loss since the network is already identified.

23/10/21

Screen Shots:



PC2

Physical | Config | Desktop | Custom Interface

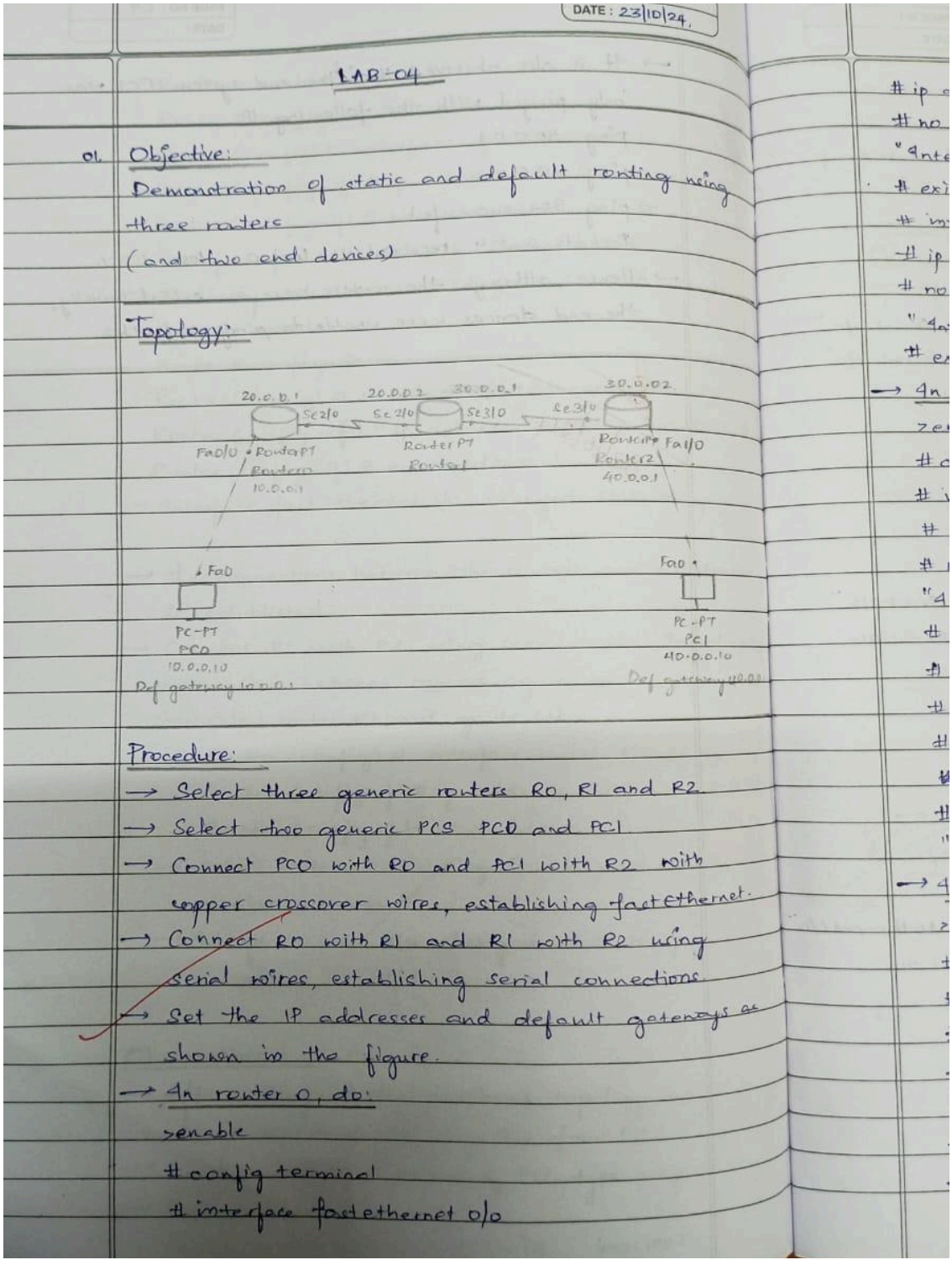
Command Prompt

```
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:



DATE : 23/10/24.

LAB-04

01. Objective:
Demonstration of static and default routing using three routers (and two end devices)

Topology:

Procedure:
→ Select three generic routers R0, R1 and R2.
→ Select two generic PCs PC0 and PC1
→ Connect PC0 with R0 and PC1 with R2 with copper crossover wires, establishing fastethernet.
→ Connect R0 with R1 and R1 with R2 using serial wires, establishing serial connections.
→ Set the IP addresses and default gateways as shown in the figure.
→ In router 0, do:
>enable
#config terminal
#interface fastethernet 0/0

PAGE NO : 09
DATE :

ip address 10.0.0.1 255.0.0.0
no shutdown
"Interface FastEthernet 0/0, changed state to up"
exit
interface serial 2/0
ip address 20.0.0.1 255.0.0.0
no shutdown
"Interface Serial 2/0, changed state to down"
exit

→ On router 1, do

> enable
config terminal
interface Serial 2/0
ip address 20.0.0.2 255.0.0.0
no shut
"Interface Serial 2/0, changed state to up"
exit
interface serial 3/0
ip address 30.0.0.1 255.0.0.0
no shut
"Interface Serial 3/0, changed state to down"
exit
"Interface Serial 3/0, changed state to up"

→ On router 2, do

> enable
config terminal
interface serial 3/0
ip address 30.0.0.2 255.0.0.0
no shut
"Interface Serial 3/0, changed state to up"
exit
interface fastethernet 1/0

PAGE NO :
DATE :

ip address 40.0.0.1 255.0.0.0
no shut
"Interface FastEthernet 1/0, changed state to up"
exit

→ To set up Static routing in router R1, do

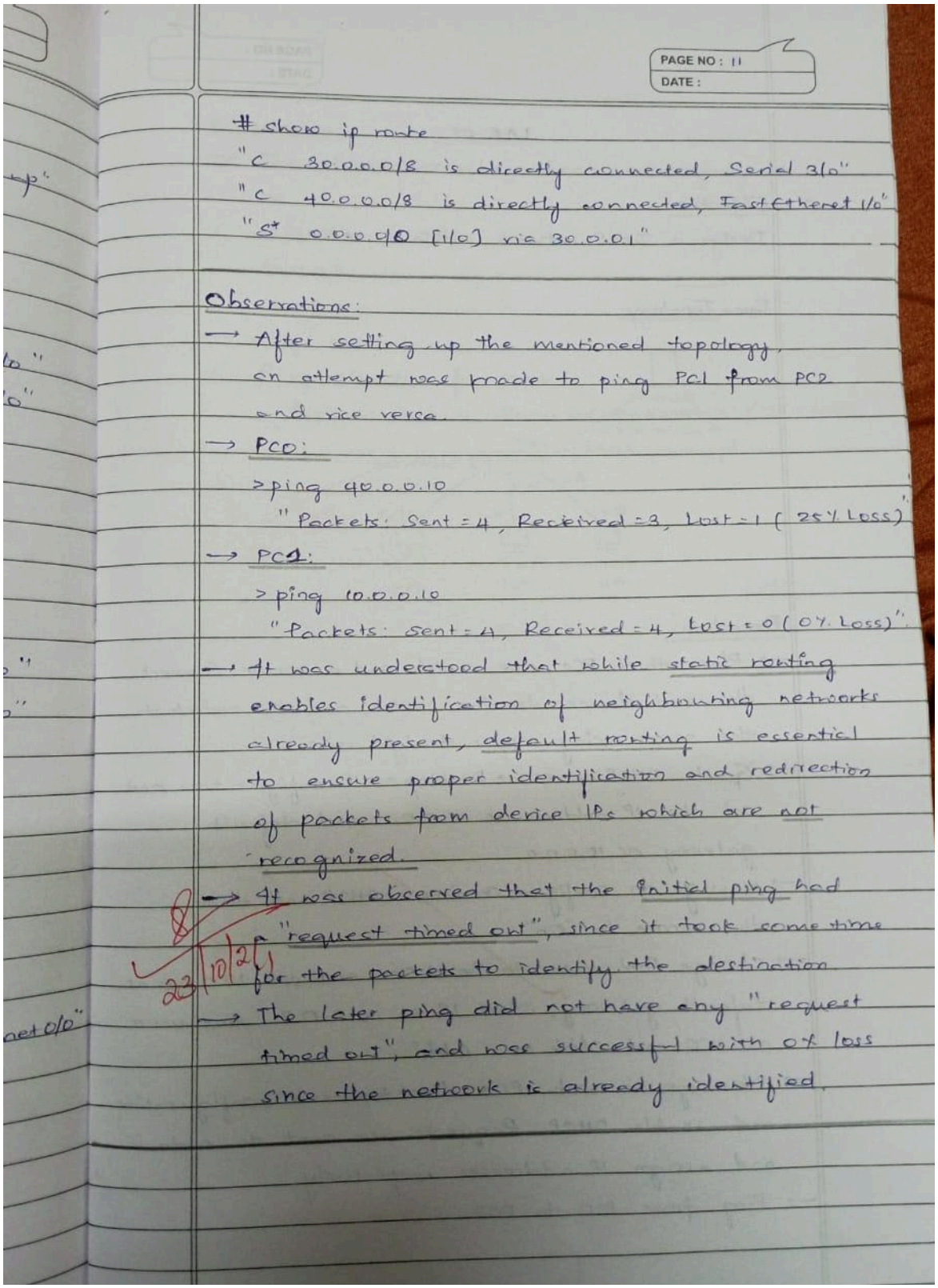
> enable
show ip route
"C 20.0.0.0/8 is directly connected, Serial 2/0"
"C 30.0.0.0/8 is directly connected, Serial 3/0"
config terminal
ip route 10.0.0.0 255.0.0.0 20.0.0.1
ip route 40.0.0.0 255.0.0.0 30.0.0.2
exit
show ip route
"S 10.0.0.0/8 [1/0] via 20.0.0.1"
"C 20.0.0.0/8 is directly connected, Serial 2/0"
"C 30.0.0.0/8 is directly connected, Serial 3/0"
"S 40.0.0.0/8 via 30.0.0.2"

→ To set up Default routing in R2, do:

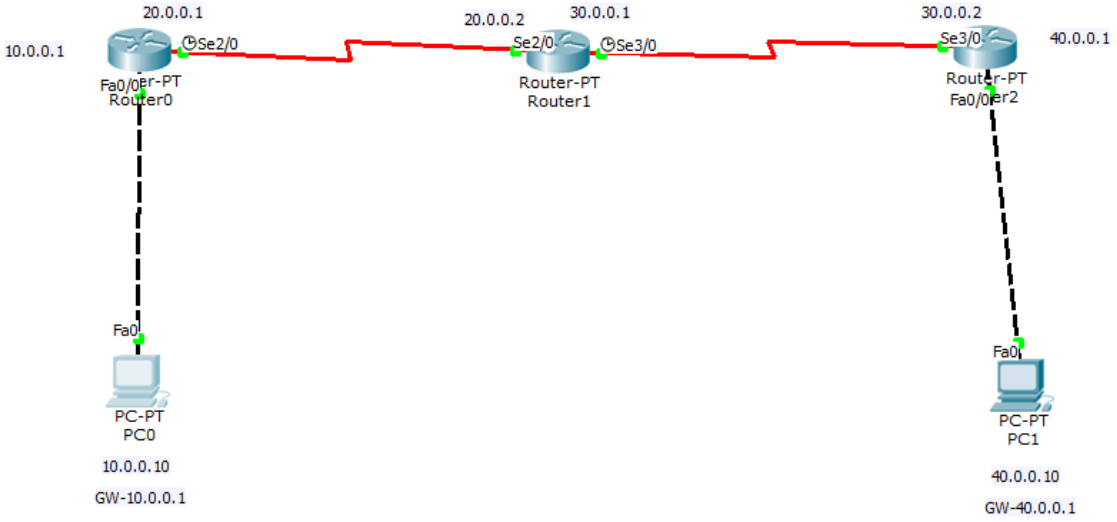
> enable
config terminal
ip route 0.0.0.0 0.0.0.0 20.0.0.2
exit
show ip route
"C 10.0.0.0/8 is directly connected, FastEthernet0/0"
"C 20.0.0.0/8 is directly connected, Serial 2/0"
"S* 0.0.0.0/0 [1/0] via 20.0.0.2"

→ To set up Default routing in R2, do:

> enable
config terminal
ip route 0.0.0.0 0.0.0.0 30.0.0.1
exit



Screen Shots:



PC0

Physical | Config | Desktop | Custom Interface

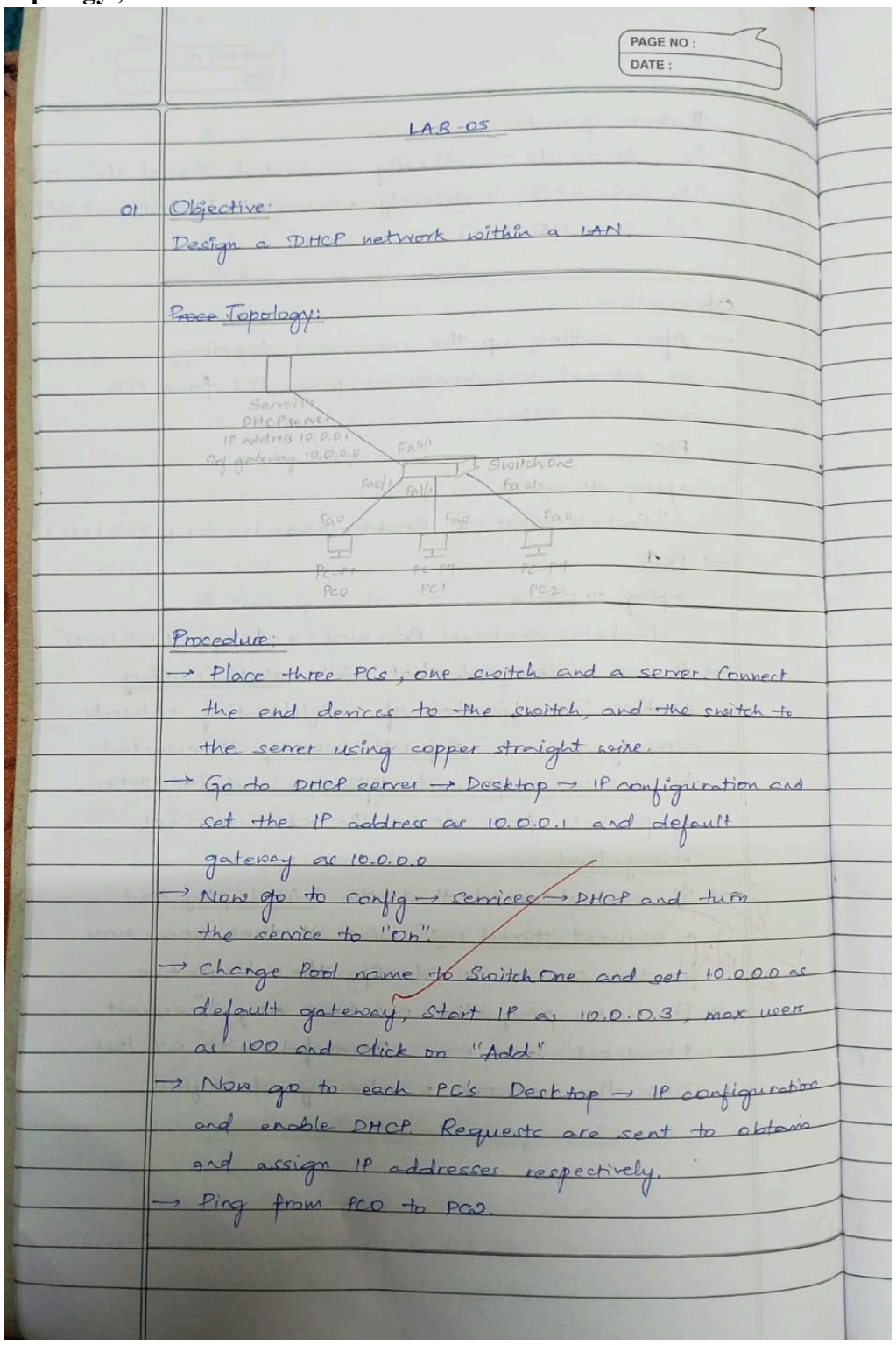
Command Prompt

```
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=6ms TTL=125
Reply from 40.0.0.10: bytes=32 time=5ms 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 = 5ms, Maximum = 7ms, Average = 6ms

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=8ms TTL=125
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Reply from 40.0.0.10: bytes=32 time=9ms 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 5

Aim: Configure DHCP within a LAN and outside LAN.

Topology , Procedure and Observation:



PAGE NO :
DATE :

LAB-05

01 Objective:
Design a DHCP network within a LAN

Topology:

Procedure:

→ Place three PCs, one switch and a server. Connect the end devices to the switch, and the switch to the server using copper straight wire.

→ Go to DHCP server → Desktop → IP configuration and set the IP address as 10.0.0.1 and default gateway as 10.0.0.0

→ Now go to Config → Services → DHCP and turn the service to "On".

→ Change Pool name to Switch One and set 10.0.0.0 as default gateway, start IP as 10.0.0.3, max users as 100 and click on "Add".

→ Now go to each PC's Desktop → IP configuration and enable DHCP. Requests are sent to obtain and assign IP addresses respectively.

→ Ping from PC0 to PC2.

PAGE NO : 13
DATE :

Observation:

→ The IP addresses have been successfully set as

10.0.0.2

10.0.0.3

10.0.0.4

→ ping 10.0.0.2

Ping statistics for 10.0.0.3:

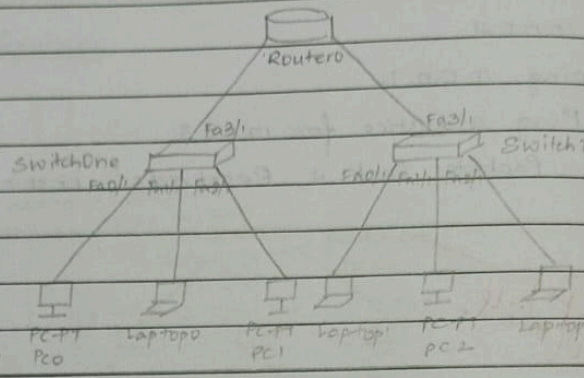
Packets: Sent = 4, Received = 1, Lost = 3 (75 % Loss)

20/11/24.

PAGE NO :
DATE :

02. Objective:
Design a DHCP network outside LAN using router

Topology:



Procedure:

→ Place PCs and Laptops to a total of six, one server, two switches and a router. Connect the end devices to switches to router & server to switch using copper straight wire.

→ Go to the DHCP server's IP configuration and set IP address as 10.0.0.2 and default gateway as 10.0.0.1

→ Go to the DHCP services in Config, turn the services to "ON" and do the following:
Set the poolname to SwitchOne, Default gateway to 10.0.0.1, Start IP to 10.0.0.3 and 100 users

→ Do the same for switch two with default gateway as 20.0.0.1, start IP to 20.0.0.3 and 100 users.

→ Now go to the CLI of the router & do the following:
>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
interface fa 0/0
ip address 20.0.0.1 255.0.0.0
ip helper address 10.0.0.2
no shut.

→ Now go to all end devices and change the IP configuration from static to DHCP. This enables the IP addresses to be automatically requested and assigned based on the server settings.

Observations:

→ The IP addresses have been successfully set

10.0.0.3 — 20.0.0.3
10.0.0.4 — 20.0.0.4
10.0.0.5 — 20.0.0.5

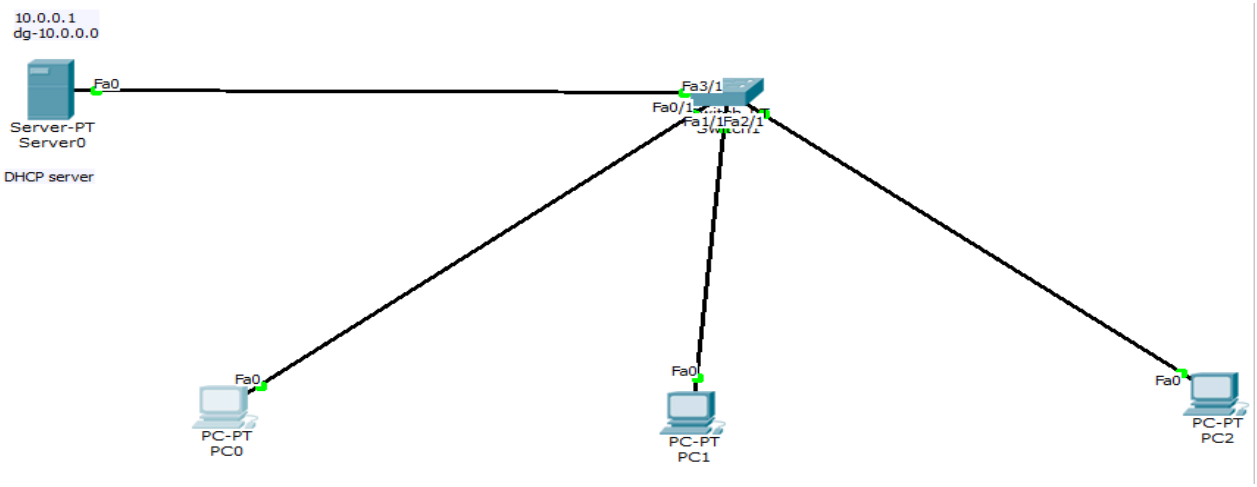
→ ping 10.0.0.3

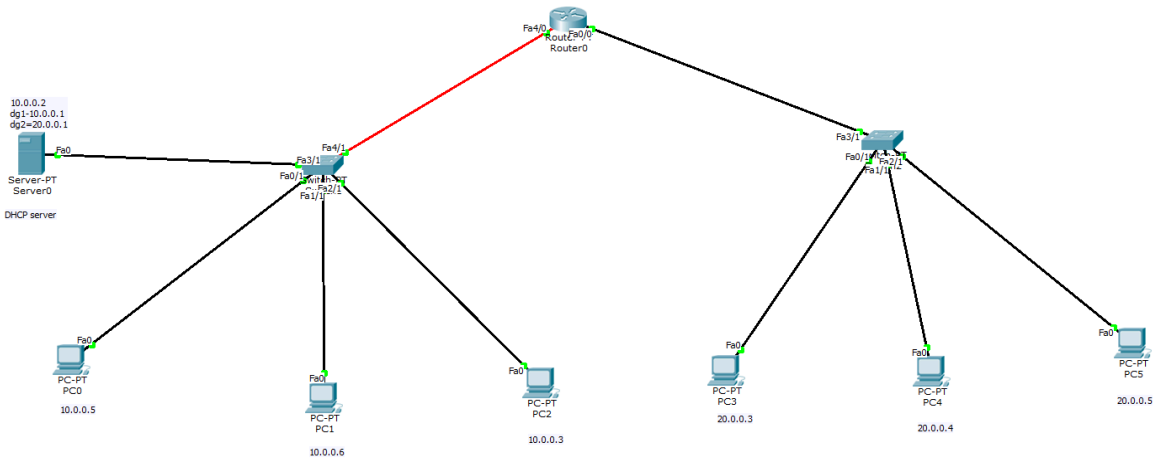
Ping statistics for 10.0.0.3:
Packets: Sent=4, Received=0, Lost=4 (100% Loss)

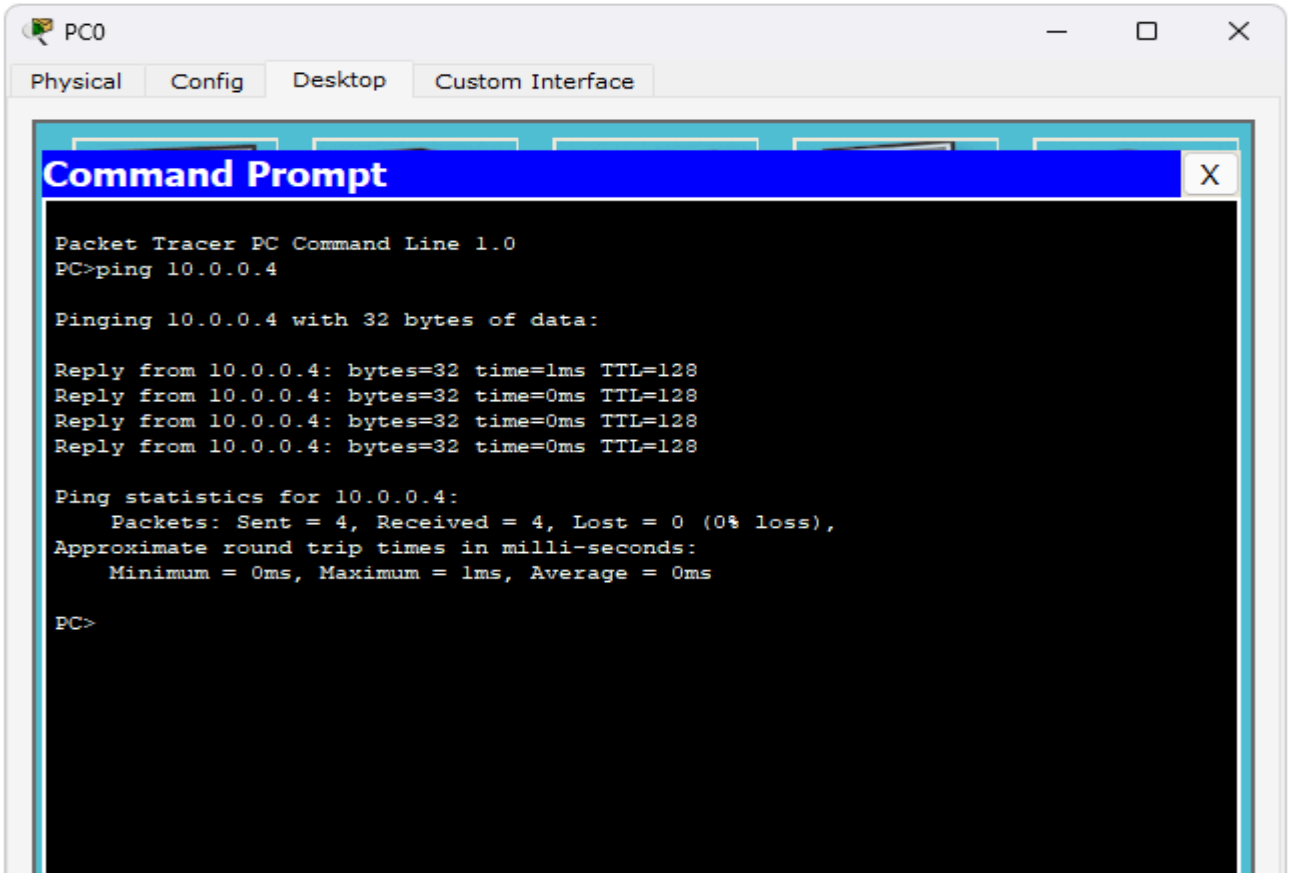
→ ping 20.0.0.3

Ping statistics for 20.0.0.3:
Packets: Sent=4, Received=4, Lost=0 (0% Loss)

Screen Shots:



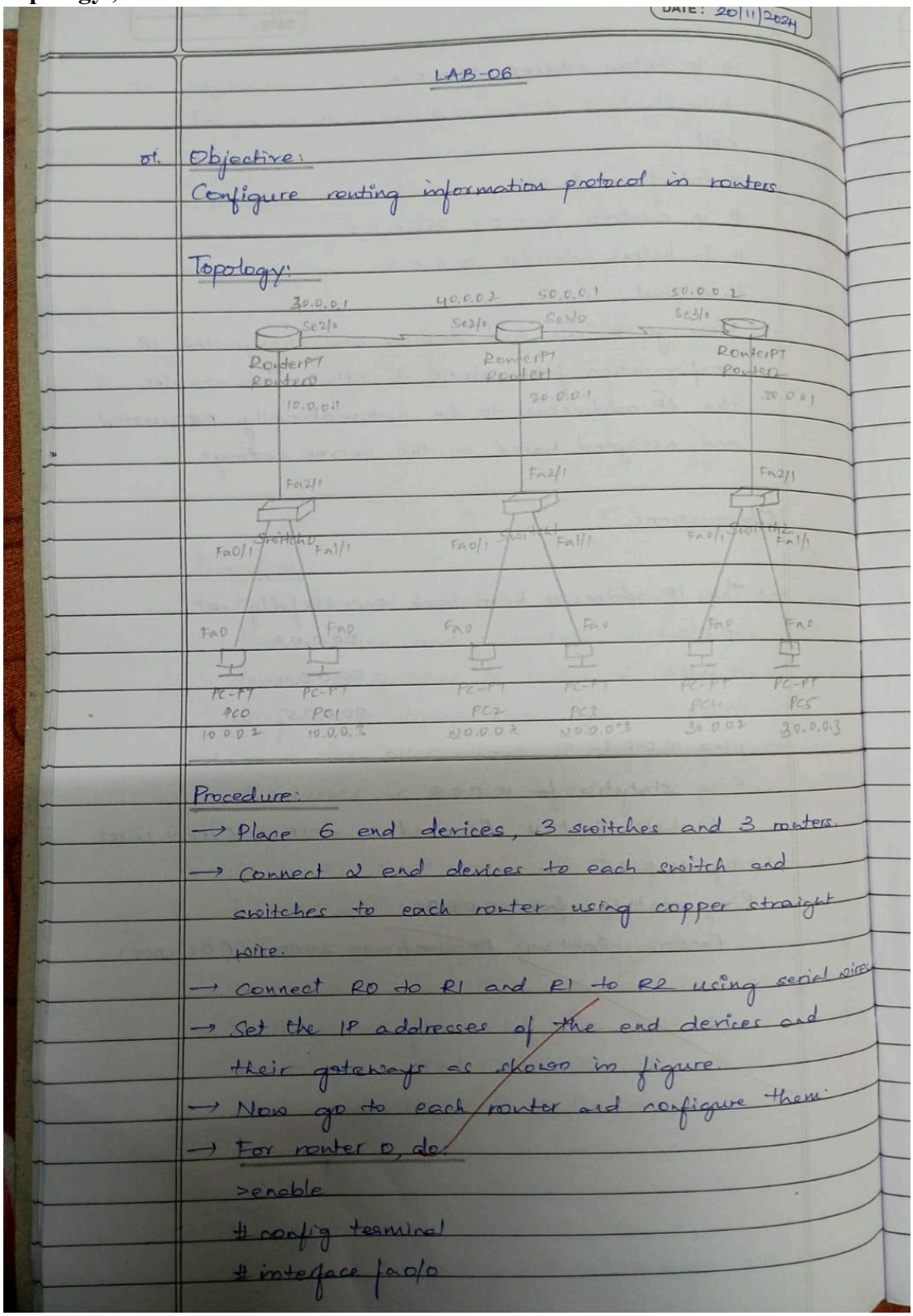



PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.4
Pinging 10.0.0.4 with 32 bytes of data:
Reply from 10.0.0.4: bytes=32 time=1ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from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



DATE: 20/11/2024

LAB-06

01. Objective:

Configure routing information protocol in routers

Topology:

Procedure:

→ Place 6 end devices, 3 switches and 3 routers.

→ Connect 2 end devices to each switch and switches to each router using copper straight wire.

→ Connect R0 to R1 and R1 to R2 using serial wire.

→ Set the IP addresses of the end devices and their gateways as shown in figure.

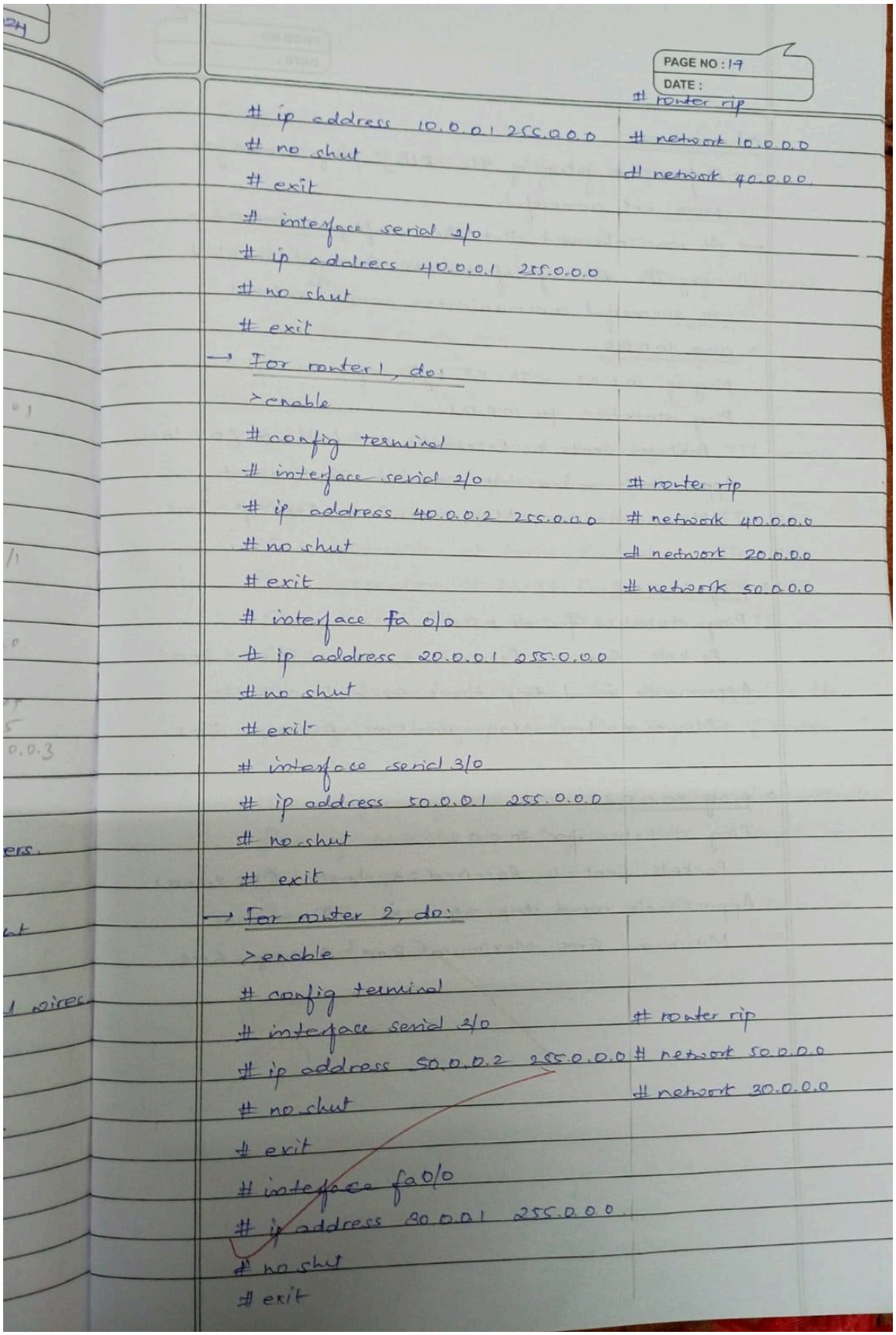
→ Now go to each router and configure them.

→ For router 0, do:

>enable

config terminal

interface fa0/0



PAGE NO : 17
DATE :

ip address 10.0.0.1 255.0.0.0
no shut
exit
interface serial 2/0
ip address 40.0.0.1 255.0.0.0
no shut
exit

→ For router 1, do:

>enable
config terminal
interface serial 2/0
ip address 40.0.0.2 255.0.0.0
no shut
exit
interface fa 0/0
ip address 20.0.0.1 255.0.0.0
no shut
exit
interface serial 3/0
ip address 50.0.0.1 255.0.0.0
no shut
exit

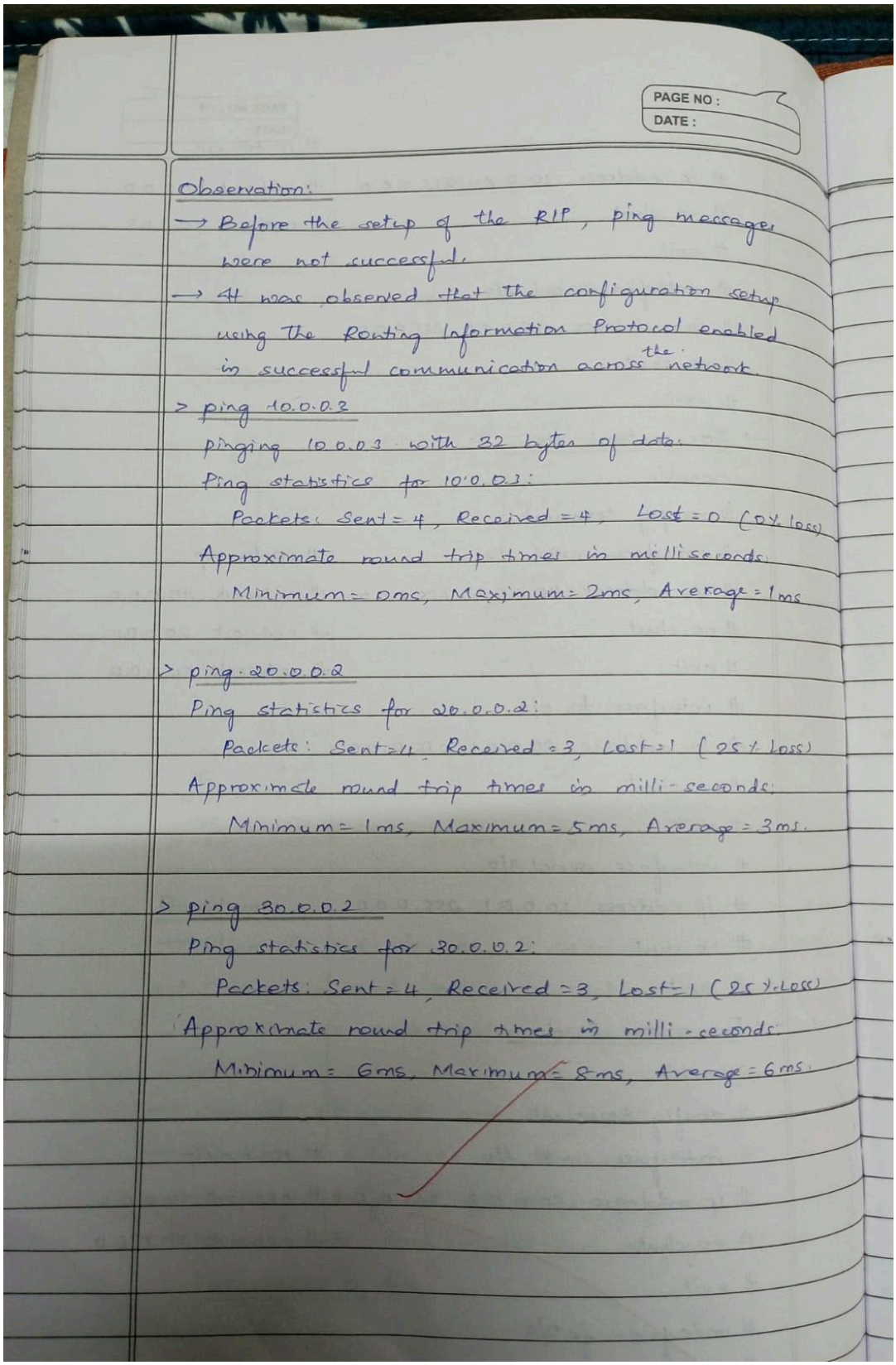
→ For router 2, do:

>enable
config terminal
interface serial 3/0
ip address 50.0.0.2 255.0.0.0
no shut
exit
interface fa0/0
ip address 30.0.0.1 255.0.0.0
no shut
exit

router rip
network 10.0.0.0
network 40.0.0.0

router rip
network 40.0.0.0
network 20.0.0.0
network 50.0.0.0

router rip
network 50.0.0.0
network 30.0.0.0



PAGE NO :
DATE :

Observation:

→ Before the setup of the RIP, ping messages were not successful.

→ It was observed that the configuration setup using the Routing Information Protocol enabled in successful communication across the network.

> ping 10.0.0.3

pinging 10.0.0.3 with 32 bytes of data:

Ping statistics for 10.0.0.3:

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

Approximate round trip times in milliseconds:

Minimum = 0ms, Maximum = 2ms, Average = 1ms

> ping 20.0.0.2

Ping statistics for 20.0.0.2:

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

Approximate round trip times in milli-seconds:

Minimum = 1ms, Maximum = 5ms, Average = 3ms.

> ping 30.0.0.2

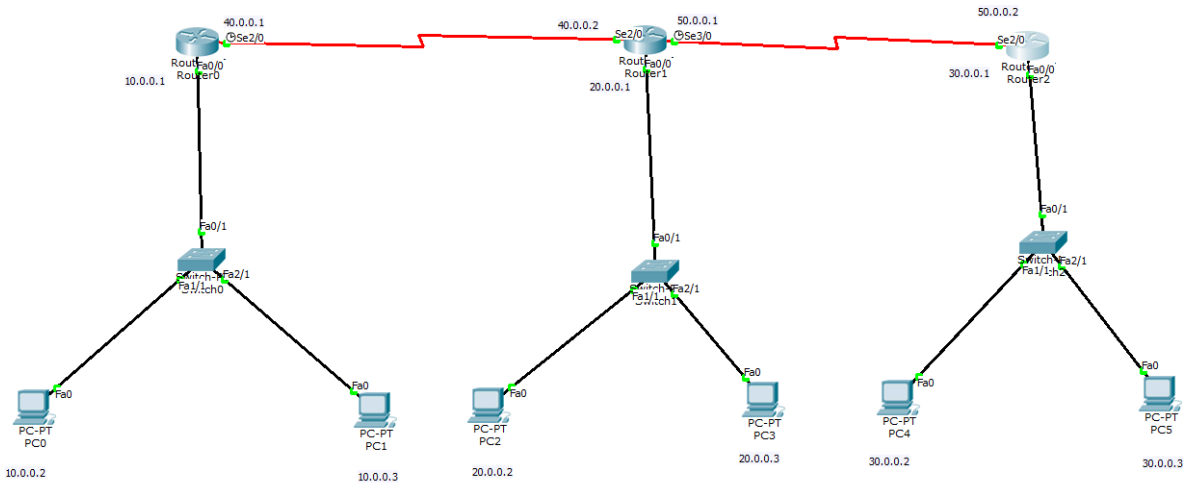
Ping statistics for 30.0.0.2:

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

Approximate round trip times in milli-seconds:

Minimum = 6ms, Maximum = 8ms, Average = 6ms.

Screen Shots:



PC0

Physical | Config | Desktop | Custom Interface

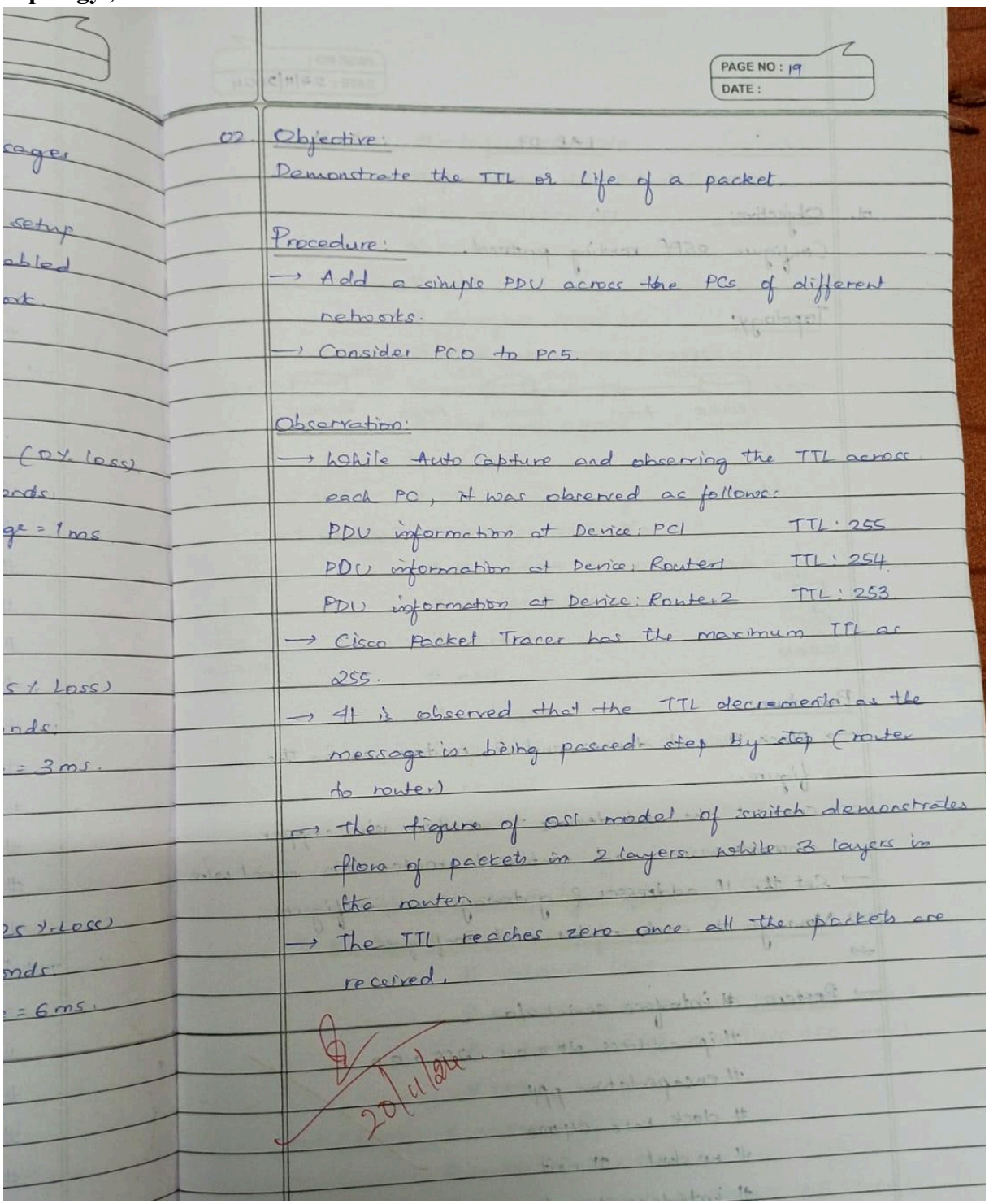
Command Prompt

```
Pinging 30.0.0.2 with 32 bytes of data:

Request timed out.
Reply from 30.0.0.2: bytes=32 time=7ms TTL=125
Reply from 30.0.0.2: bytes=32 time=6ms TTL=125
Reply from 30.0.0.2: bytes=32 time=7ms TTL=125

Ping statistics for 30.0.0.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 6ms, Maximum = 7ms, Average = 6ms

PC>ping 30.0.0.2

Pinging 30.0.0.2 with 32 bytes of data:

Reply from 30.0.0.2: bytes=32 time=4ms TTL=125
Reply from 30.0.0.2: bytes=32 time=7ms TTL=125
Reply from 30.0.0.2: bytes=32 time=7ms TTL=125
Reply from 30.0.0.2: bytes=32 time=7ms TTL=125

Ping statistics for 30.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 4ms, Maximum = 7ms, Average = 6ms

PC>
```

Program 7

Aim: Demonstrate the TTL/ Life of a Packet .

Topology , Procedure and Observation:



PAGE NO : 19
DATE :

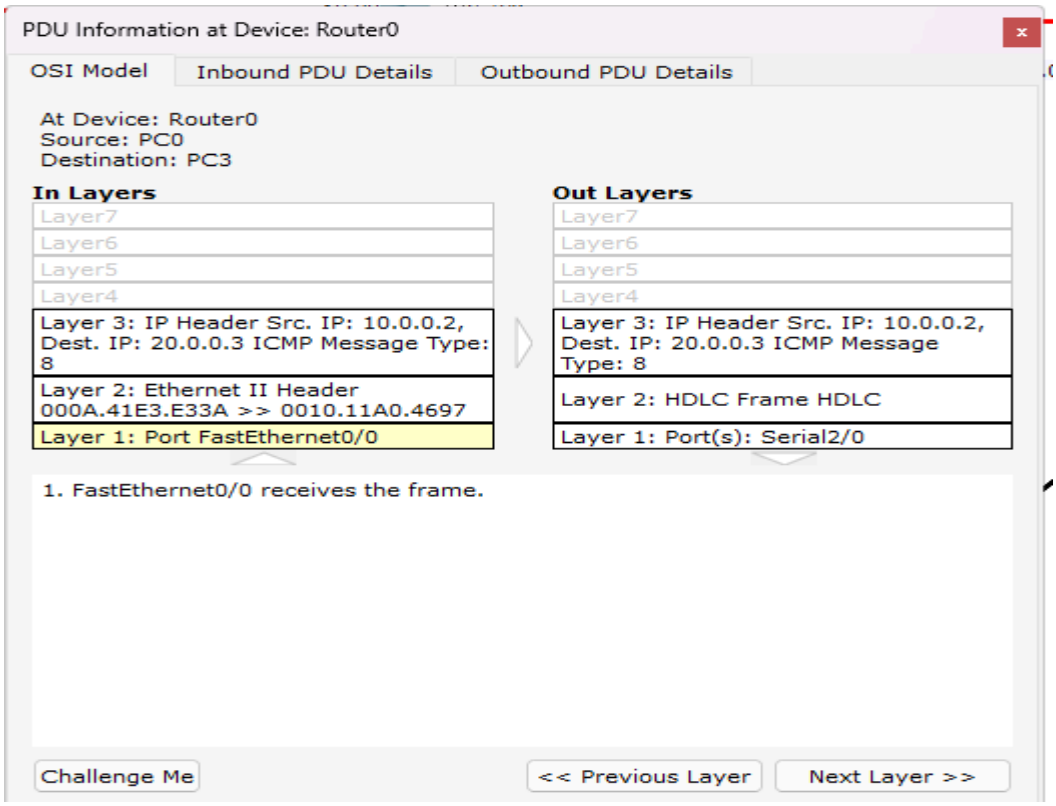
02. Objective:
Demonstrate the TTL or life of a packet.

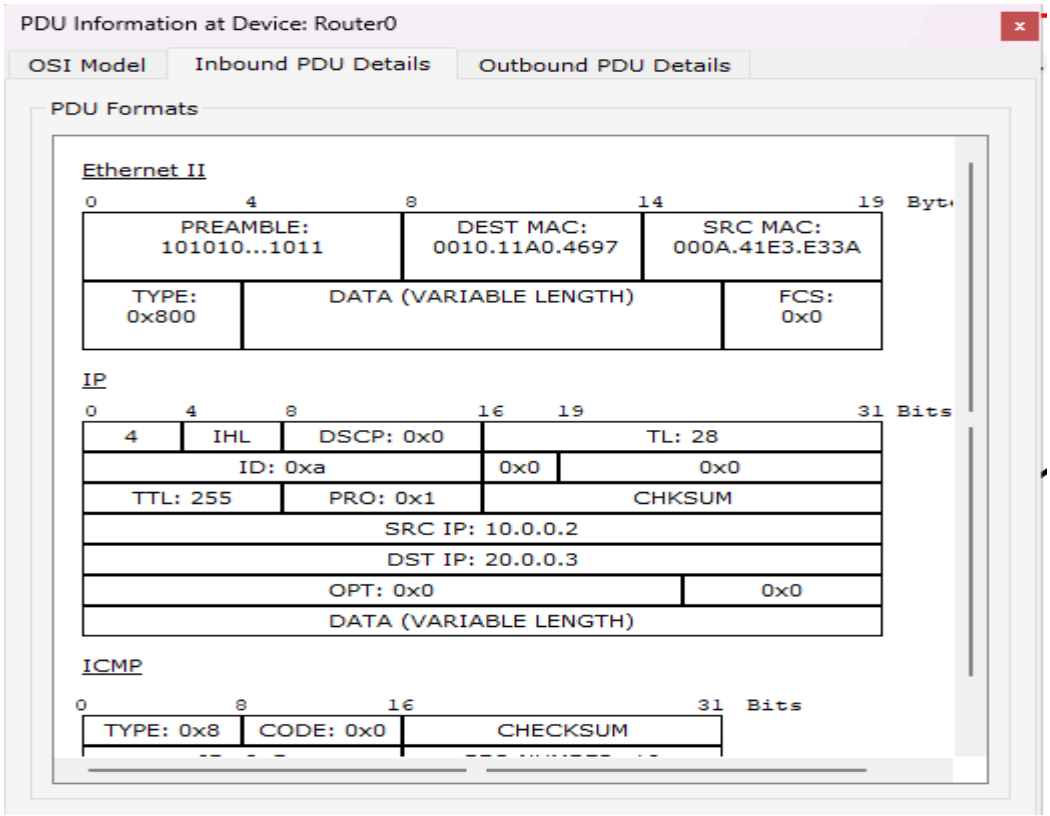
Procedure:
→ Add a simple PDU across the PCs of different networks.
→ Consider PC0 to PC5.

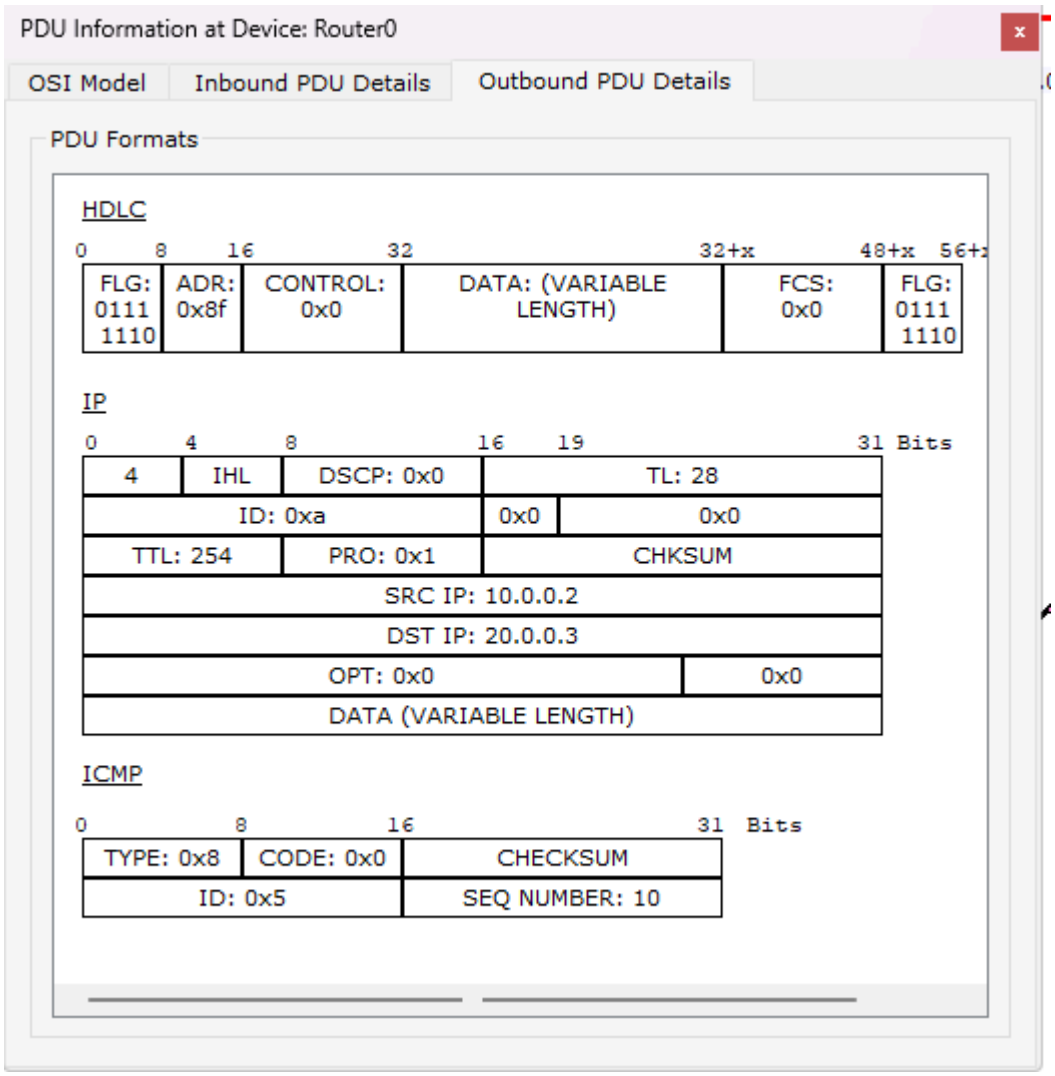
Observation:
→ While Auto Capture and observing the TTL across each PC, it was observed as follows:
PDU information at Device: PC1 TTL: 255
PDU information at Device: Router1 TTL: 254
PDU information at Device: Router2 TTL: 253
→ Cisco Packet Tracer has the maximum TTL as 255.
→ It is observed that the TTL decrements as the message is being passed step by step (router to router)
→ The figure of OSI model of switch demonstrates flow of packets in 2 layers while 3 layers in the router.
→ The TTL reaches zero once all the packets are received.

20/11/24

Screen Shots:


PDU Information at Device: Router0
OSI Model
Inbound PDU Details
Outbound PDU Details
At Device: Router0
Source: PC0
Destination: PC3
In Layers
Out Layers
Layer7
Layer6
Layer5
Layer4
Layer 3: IP Header Src. IP: 10.0.0.2, Dest. IP: 20.0.0.3 ICMP Message Type: 8
Layer 2: Ethernet II Header 000A.41E3.E33A >> 0010.11A0.4697
Layer 1: Port FastEthernet0/0
Layer 3: IP Header Src. IP: 10.0.0.2, Dest. IP: 20.0.0.3 ICMP Message Type: 8
Layer 2: HDLC Frame HDLC
Layer 1: Port(s): Serial2/0
1. FastEthernet0/0 receives the frame.
Challenge Me
<< Previous Layer
Next Layer >>

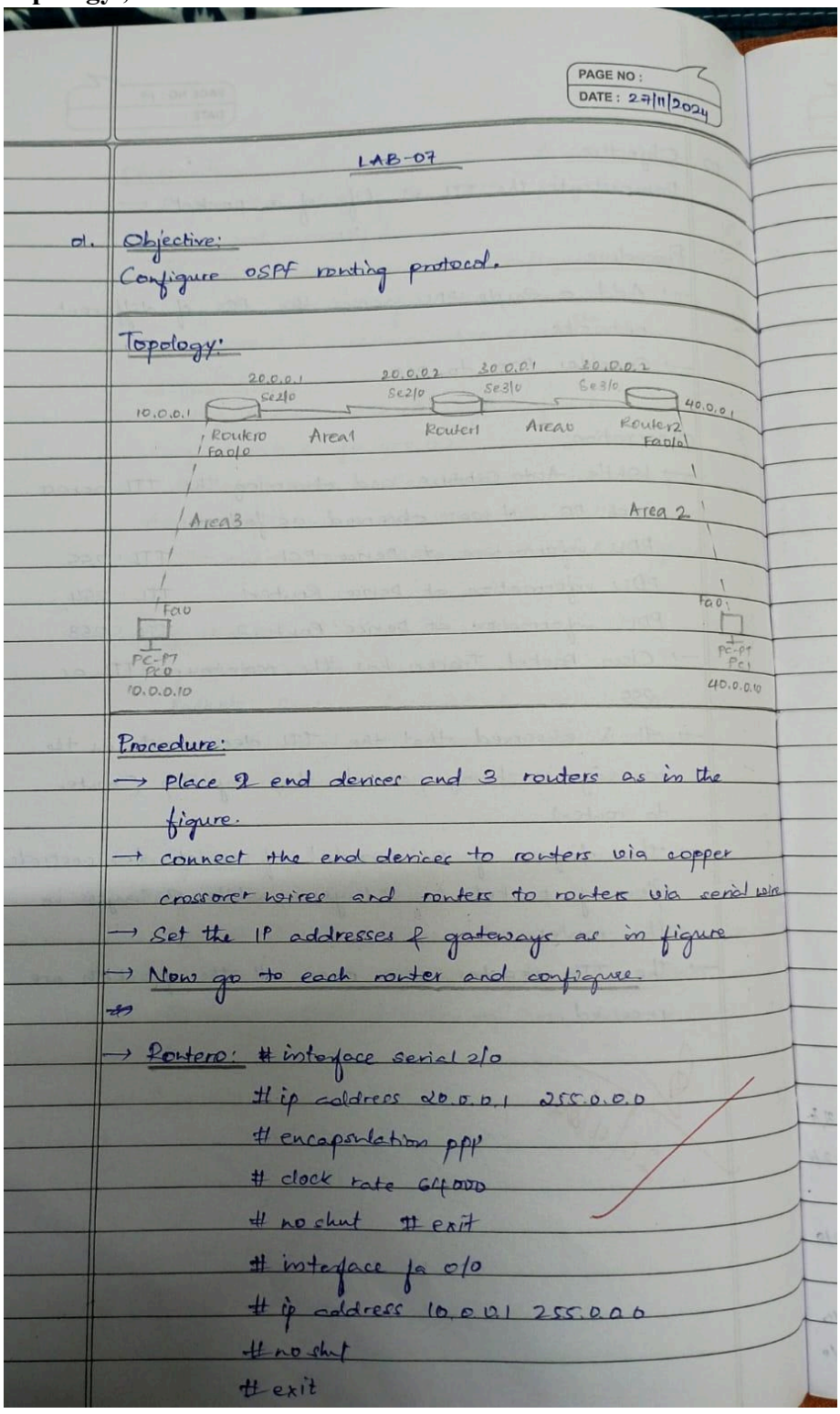

PDU Information at Device: Router0
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
0 4 8 14 19 Byt
PREAMBLE: 101010...1011
DEST MAC: 0010.11A0.4697
SRC MAC: 000A.41E3.E33A
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
0 4 8 16 19 31 Bits
4
IHL
DSCP: 0x0
TL: 28
ID: 0xa
0x0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



PAGE NO :
DATE : 27/11/2024

LAB-07

01. Objective:
Configure OSPF routing protocol.

Topology:

Procedure:

→ Place 2 end devices and 3 routers as in the figure.

→ connect the end devices to routers via copper crossover wires and routers to routers via serial wire

→ Set the IP addresses & gateways as in figure

→ Now go to each router and configure.

→ Router0: # interface serial 2/0
ip address 20.0.0.1 255.0.0.0
encapsulation ppp
clock rate 64000
no shut # exit
interface fa 0/0
ip address 10.0.0.1 255.0.0.0
no shut
exit

PAGE NO : 21
DATE :

→ Router1: # interface serial 2/0
ip address 20.0.0.2 255.0.0.0
encapsulation ppp
no shut
exit
interface serial 3/0
ip address 30.0.0.1 255.0.0.0
encapsulation ppp
clock rate 64000
no shut
exit

→ Router2: # interface serial 3/0
ip address 30.0.0.2 255.0.0.0
encapsulation ppp
no shut
exit
interface fa 0/0
ip address 40.0.0.1 255.0.0.0
no shut
exit

→ Enable IP routing by configuring OSPF.

→ Router R0: # 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
exit

→ Router1: # 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
exit

PAGE NO :
DATE :

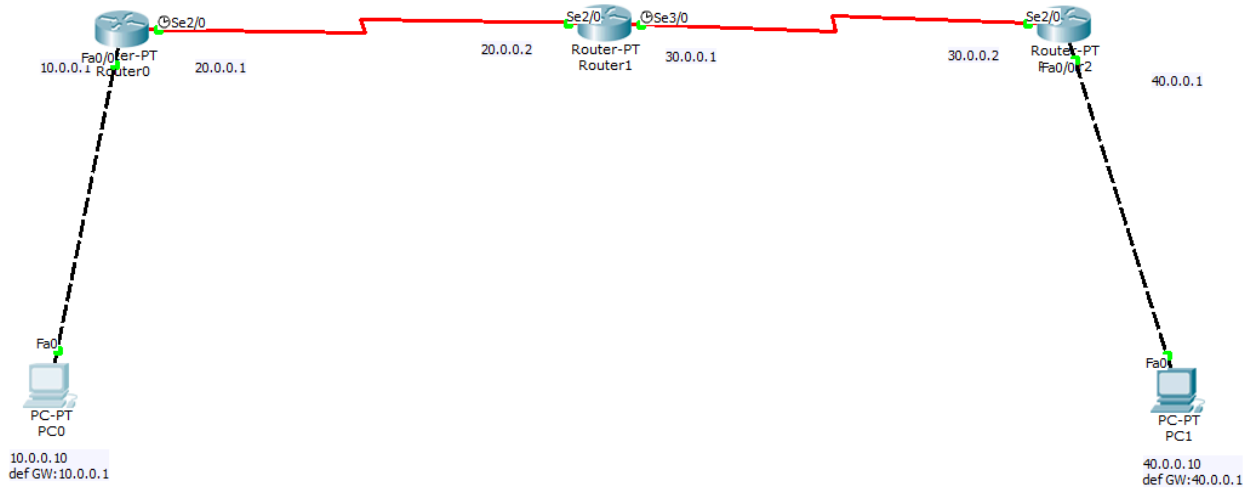
→ Router2: # router ospf 1
router-id 3.3.3.3
network 30.0.0.0 0.255.255.255 area 0
network 40.0.0.0 0.255.255.255 area 2
exit

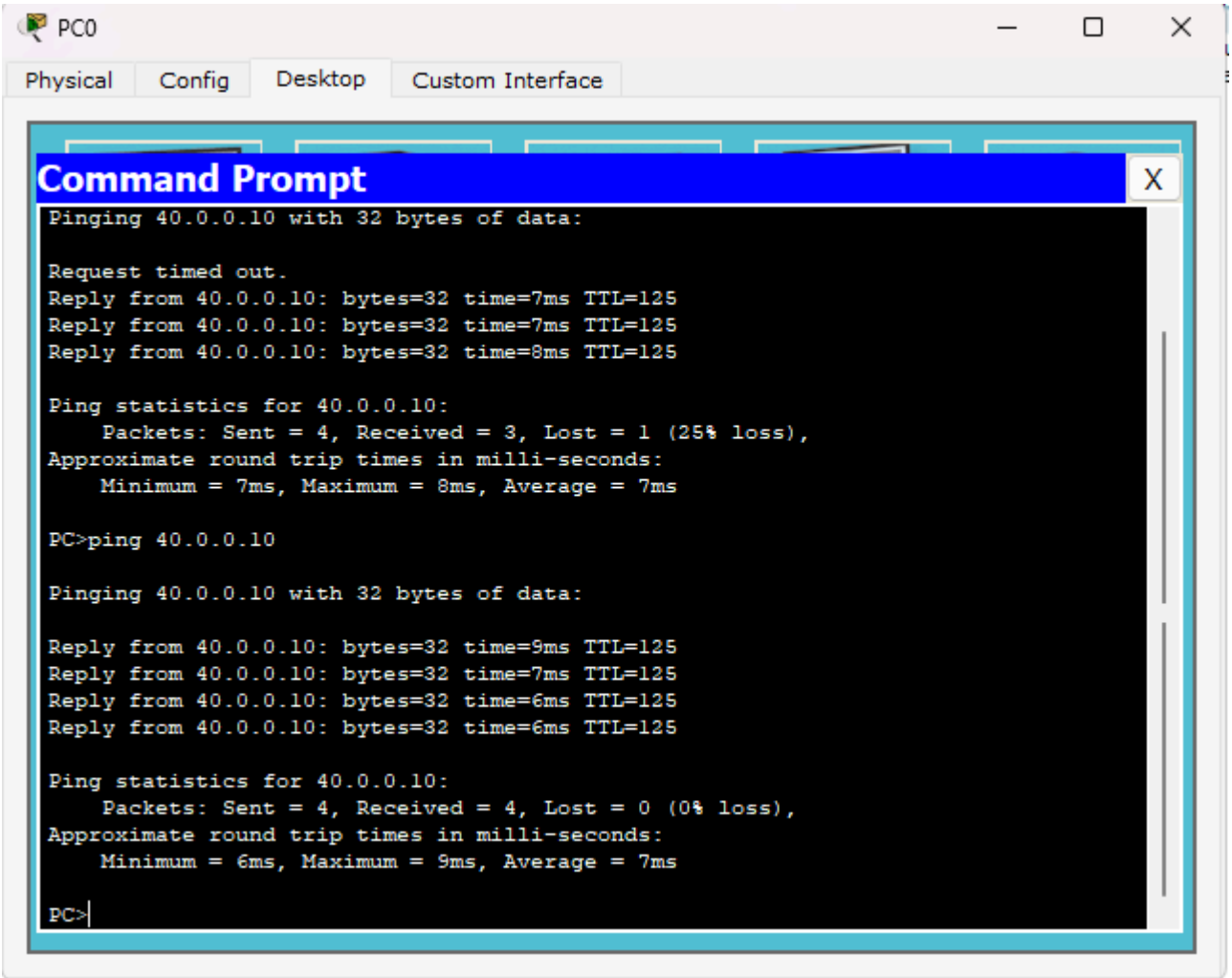
→ Configure loopback address to routers.
→ Router0: # interface loopback 0
ip add 172.16.1.252 255.255.0.0
no shut
→ Router1: # interface loopback 0
ip add 172.16.1.253 255.255.0.0
no shut
→ Router2: # interface loopback 0
ip add 172.16.1.254 255.255.0.0
no shut

→ Create virtual link between Router0 and router1
→ Router0: # router ospf 1
area 1 virtual-link 2.2.2.2
→ Router1: # router ospf 1
area 1 virtual-link 1.1.1.1

Observations:
→ Once the IP routing was enabled by OSPF.
#show ip route indicated:
O IA 40.0.0.0/8 [110/129] via 20.0.0.2, 00:04:23, Serial2/1
O IA 30.0.0.0/8 [110/128] via 20.0.0.2, 00:07:28, Serial2/1
→ After loopback, # show ip route indicated
O IA 20.0.0.0/8 [110/128] via 30.0.0.1 00:18:58, Serial2/0
→ After virtual link, # show ip route indicated
O IA 20.0.0.0/8 [110/128] via 30.0.0.1, 00:01:58, Serial2/0
O IA 10.0.0.0/8 [110/129] via 30.0.0.1, 00:01:56, Serial2/0

Screen Shots:

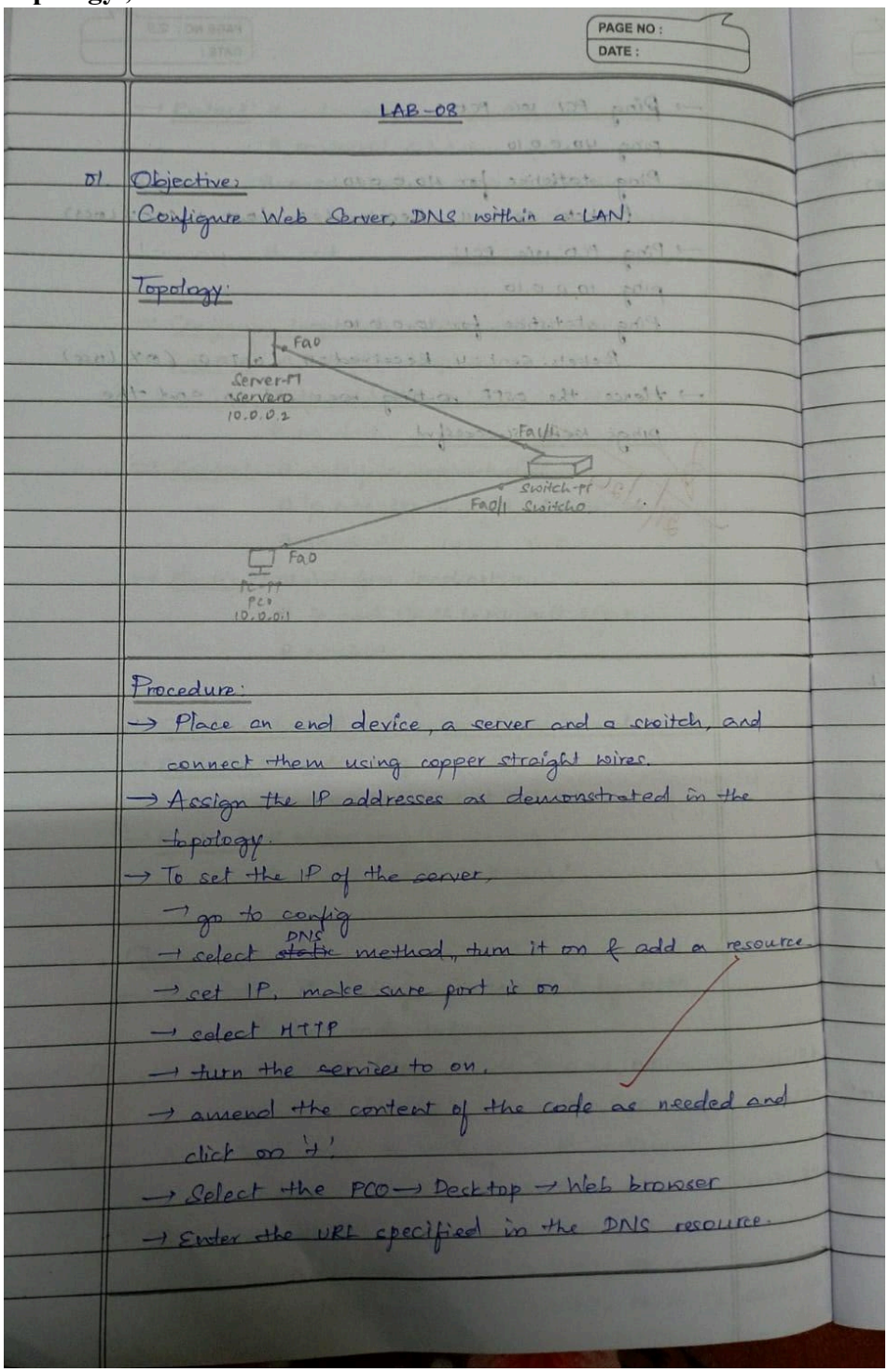




Program 9

Aim: Configure Web Server, DNS within a LAN.

Topology , Procedure and Observation:



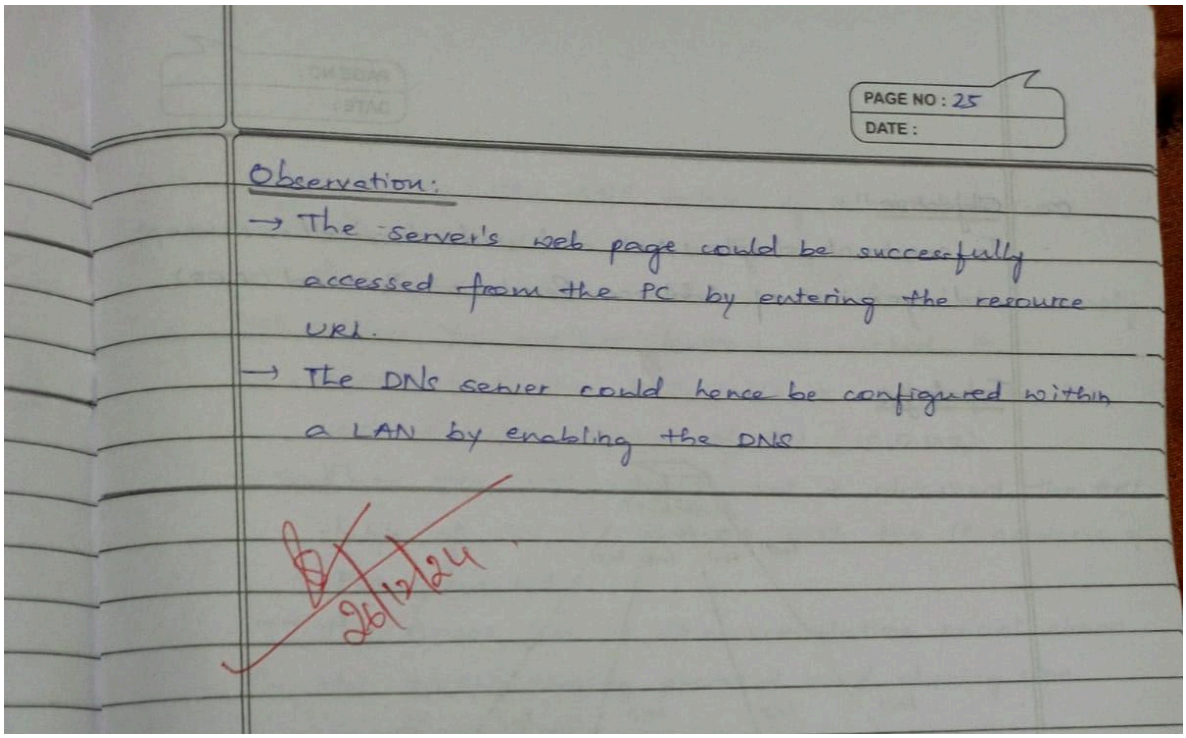
PAGE NO :
DATE :

LAB-08

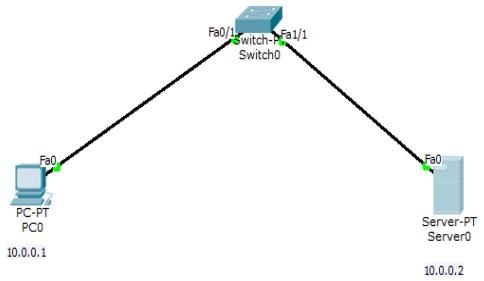
01 Objective:
Configure Web Server, DNS within a LAN.

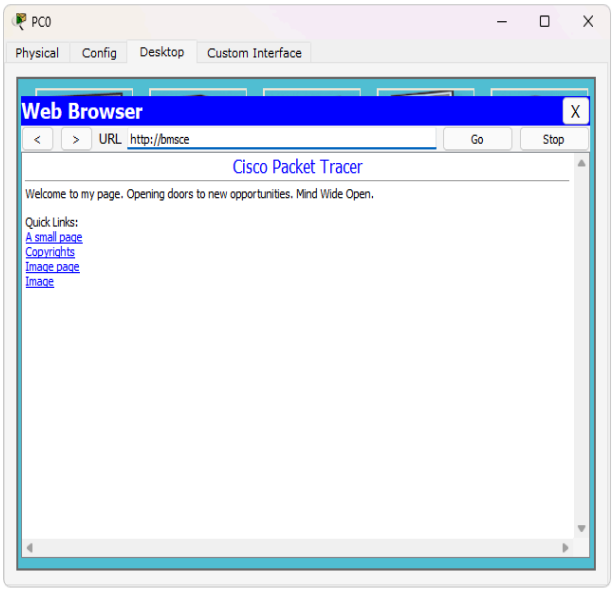
Topology:

Procedure:
→ Place an end device, a server and a switch, and connect them using copper straight wires.
→ Assign the IP addresses as demonstrated in the topology.
→ To set the IP of the server,
→ go to config
→ select ~~static~~ DNS method, turn it on & add a resource.
→ set IP, make sure port is on
→ select HTTP
→ turn the services to on.
→ amend the content of the code as needed and click on 'a'
→ Select the PC0 → Desktop → Web browser
→ Enter the URL specified in the DNS resource.



Screen Shots:

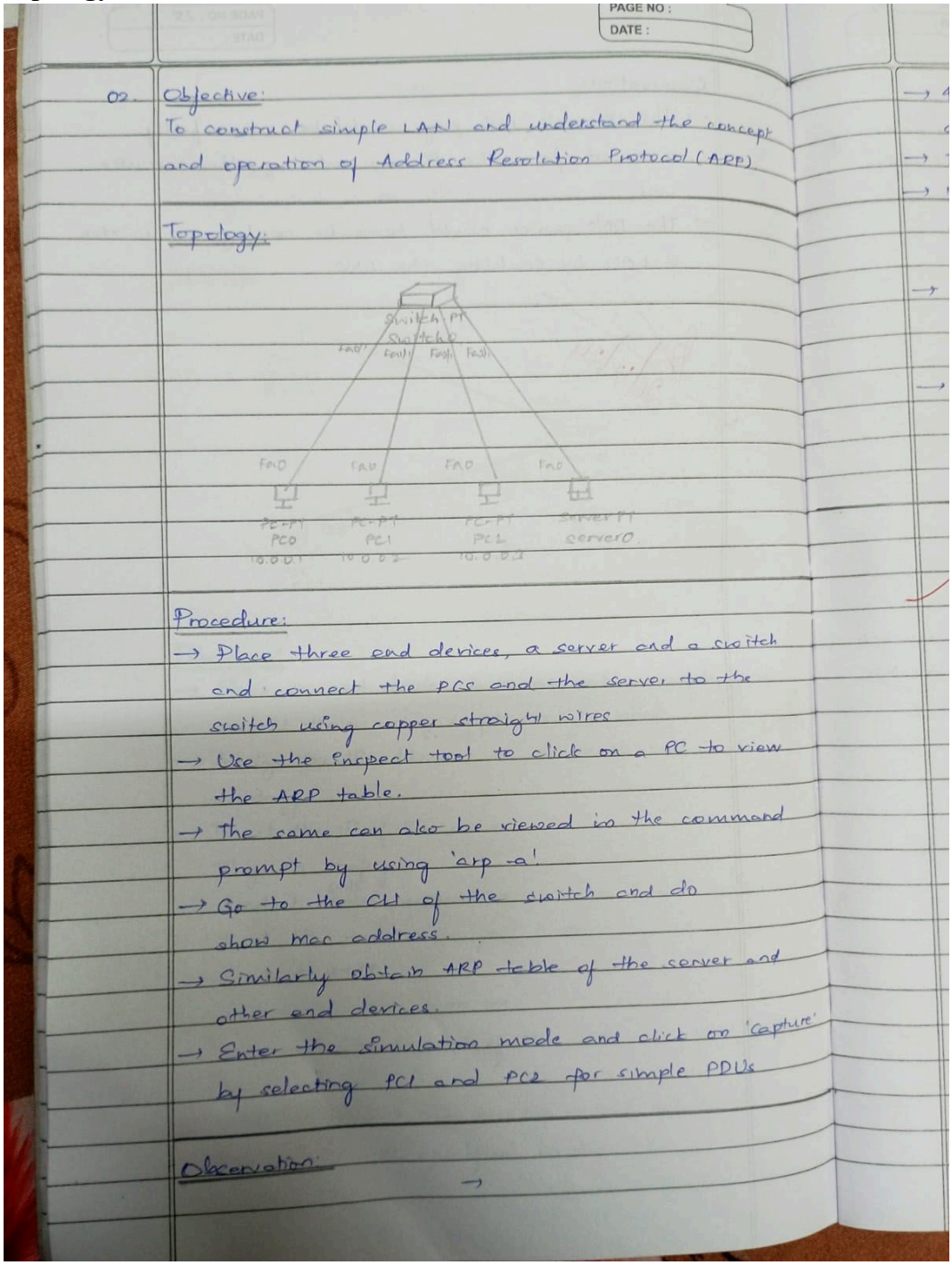




Program 10

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

Topology , Procedure and Observation:



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

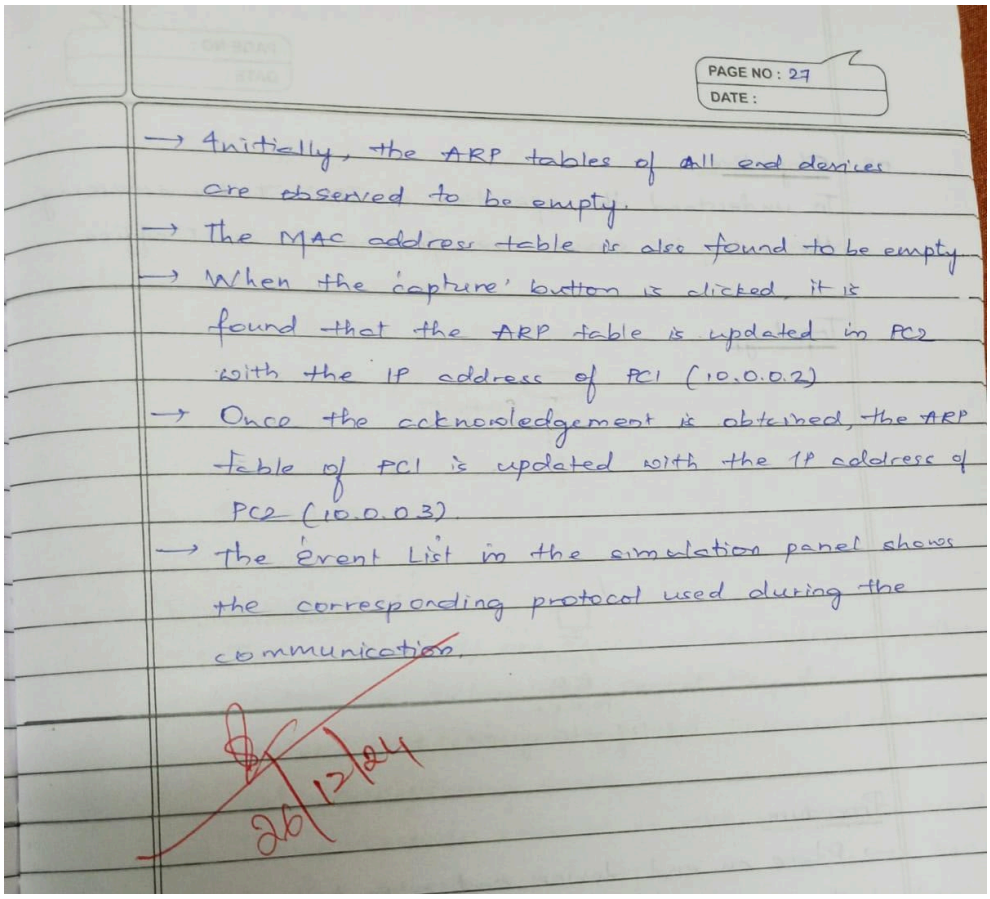
Topology:

Procedure:

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

Observation:

→

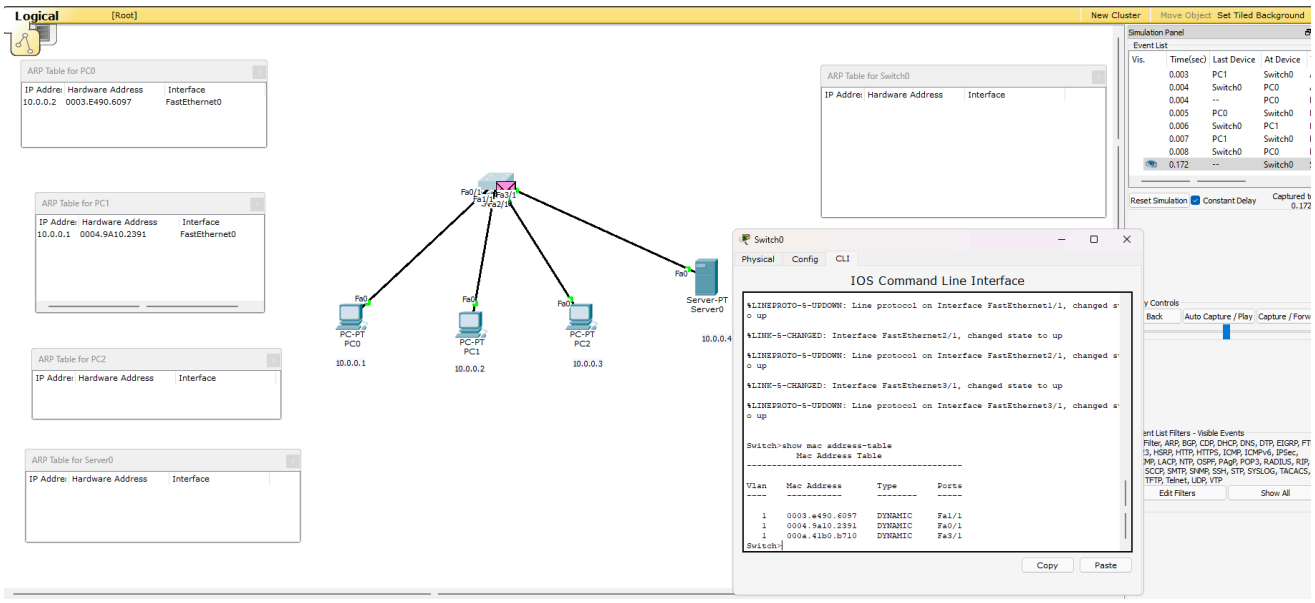


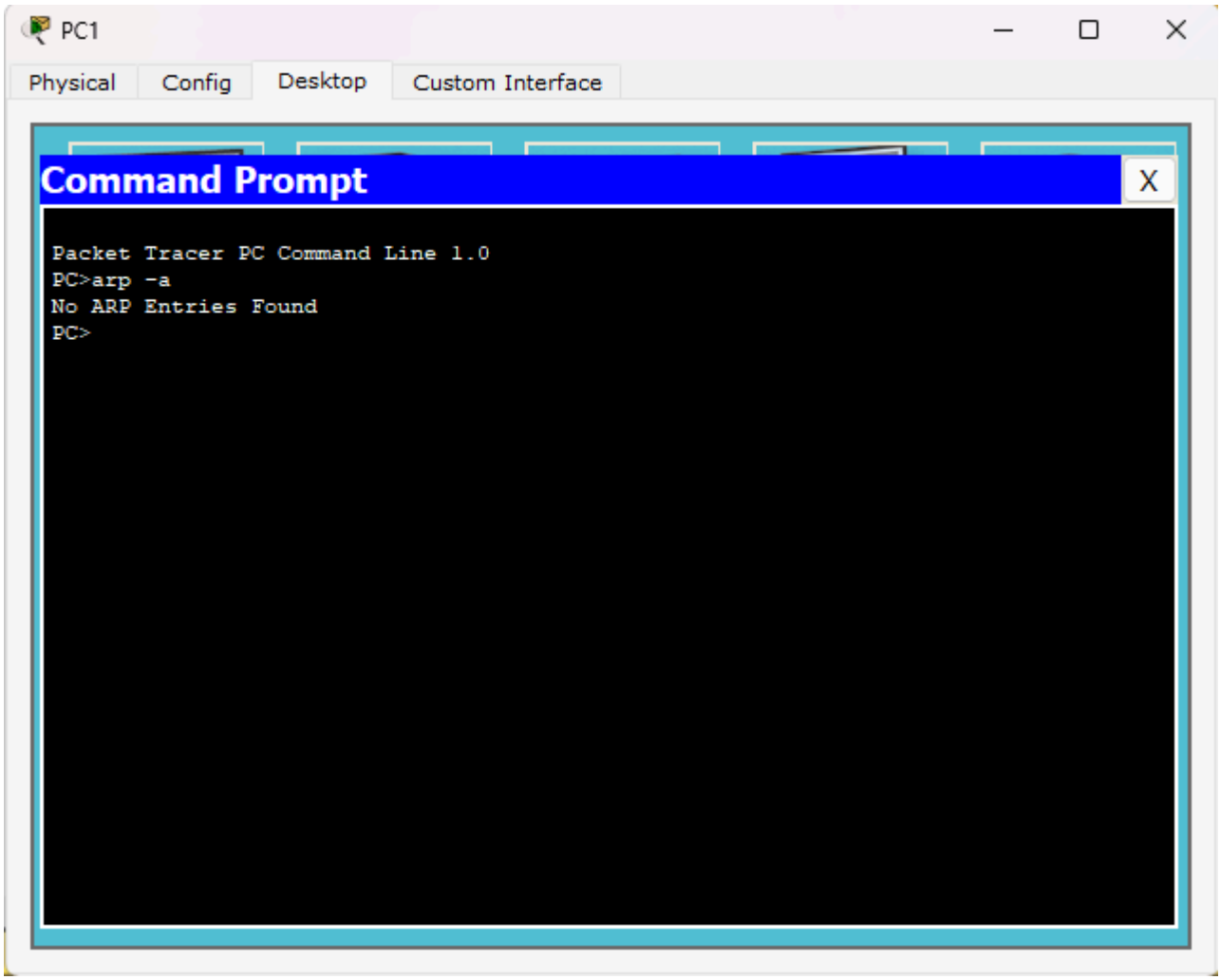
PAGE NO : 27
DATE :

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

26/12/24

Screen Shots:

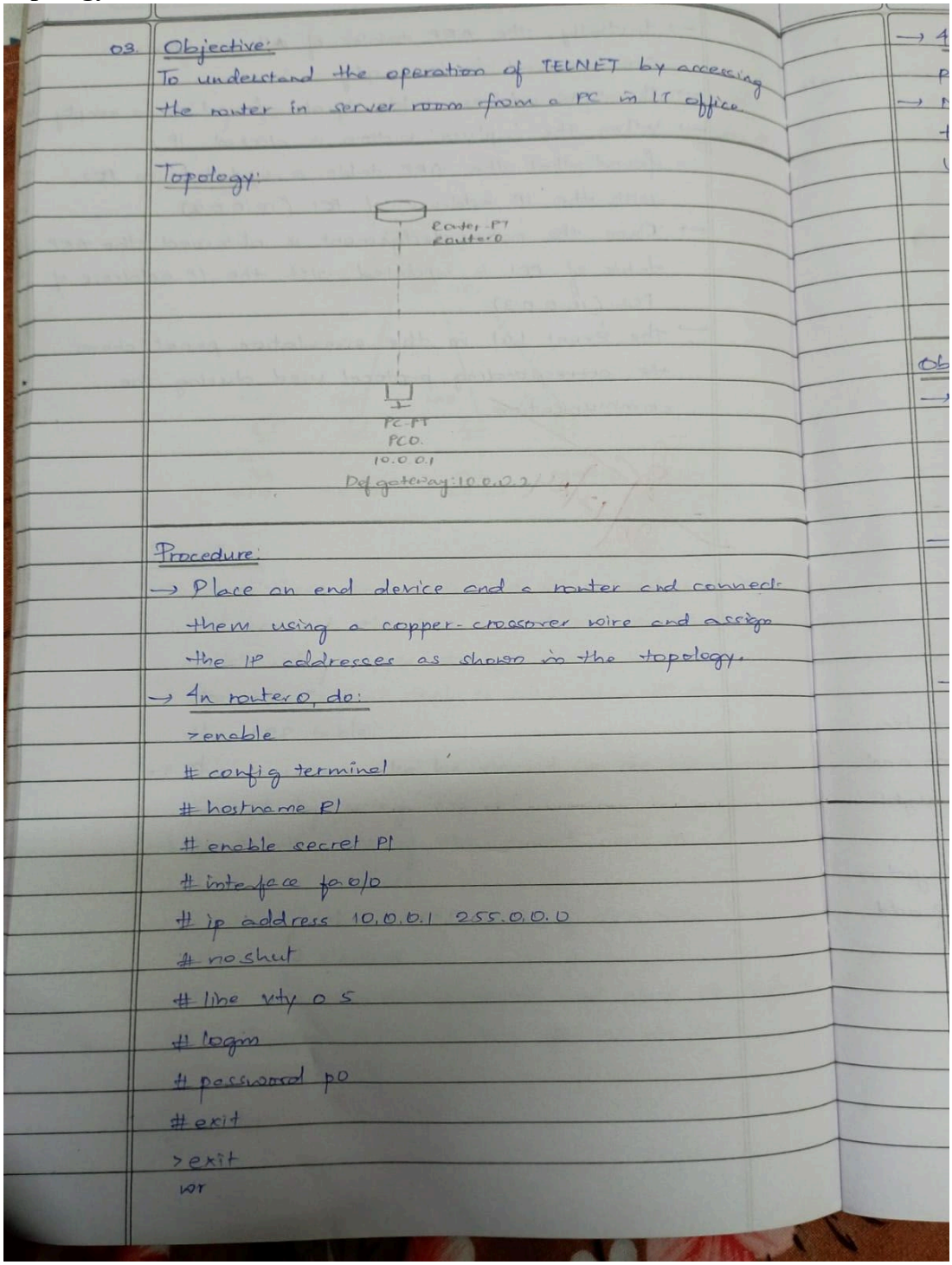


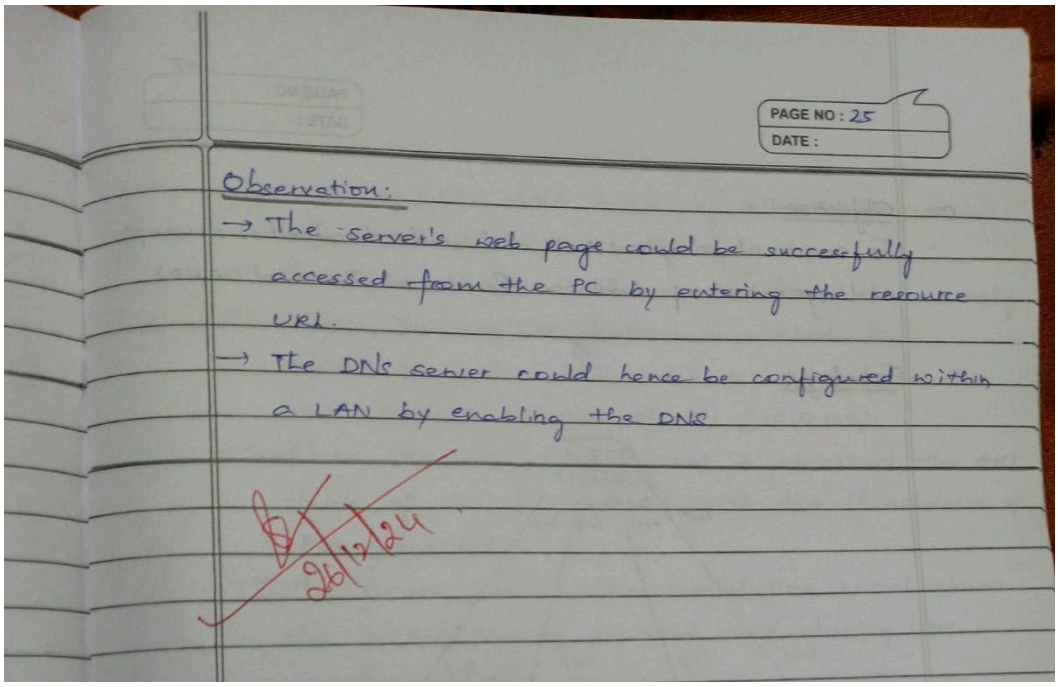

PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>arp -a
No ARP Entries Found
PC>

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:





PAGE NO : 25
DATE :

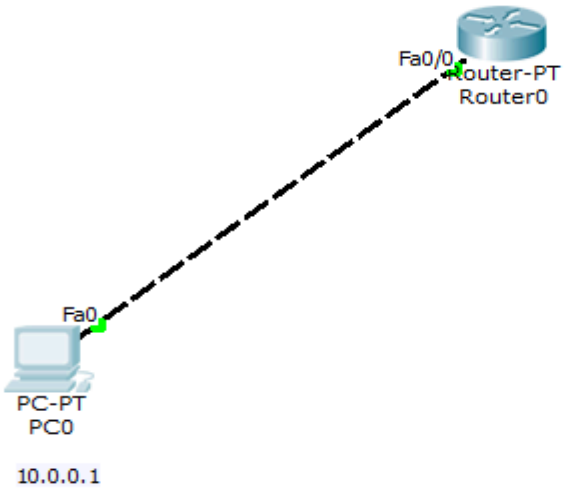
Observation:

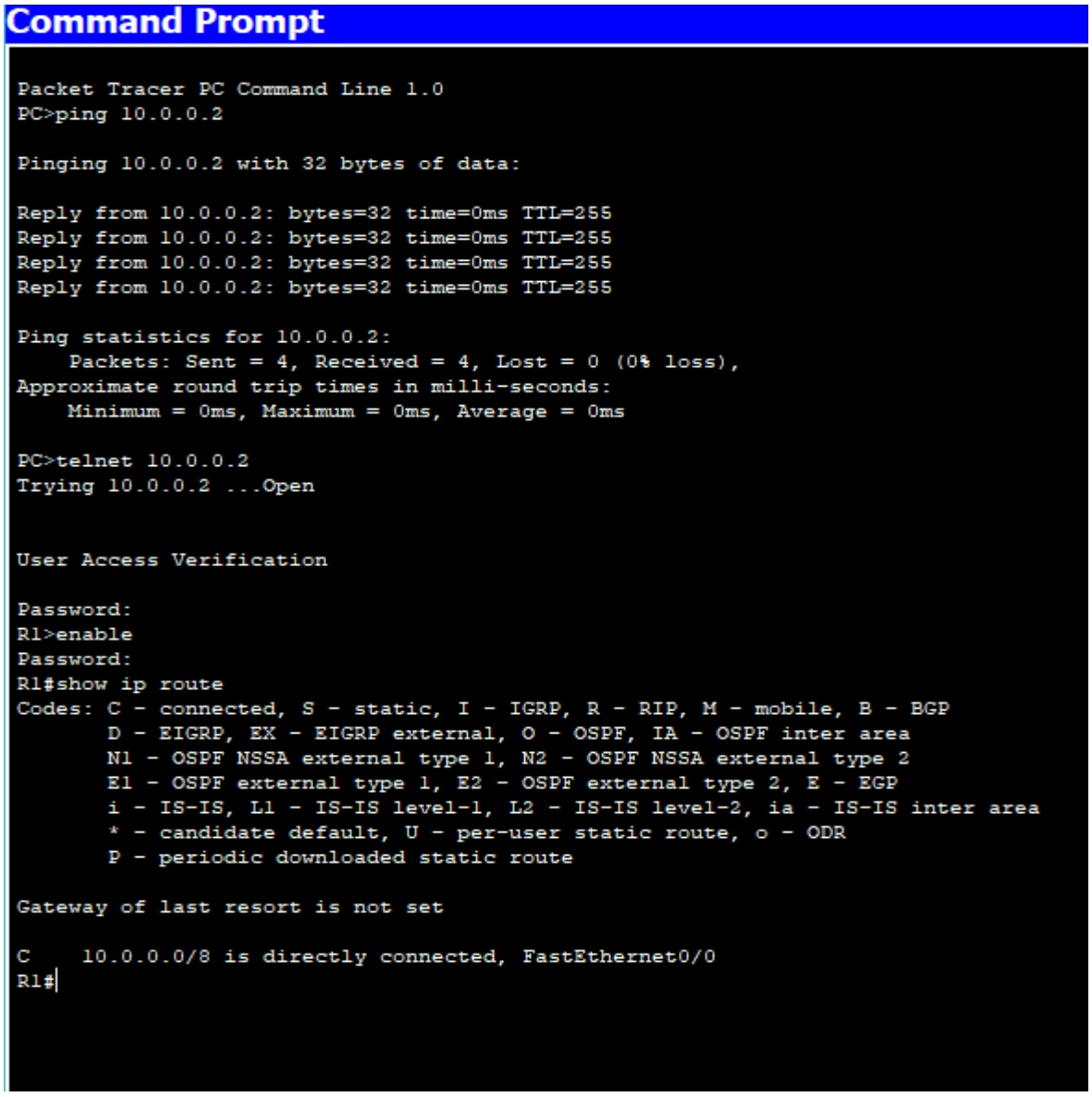
→ The server's web page could be successfully accessed from the PC by entering the resource URL.

→ The DNS server could hence be configured within a LAN by enabling the DNS

26/12/24

Screen Shots:

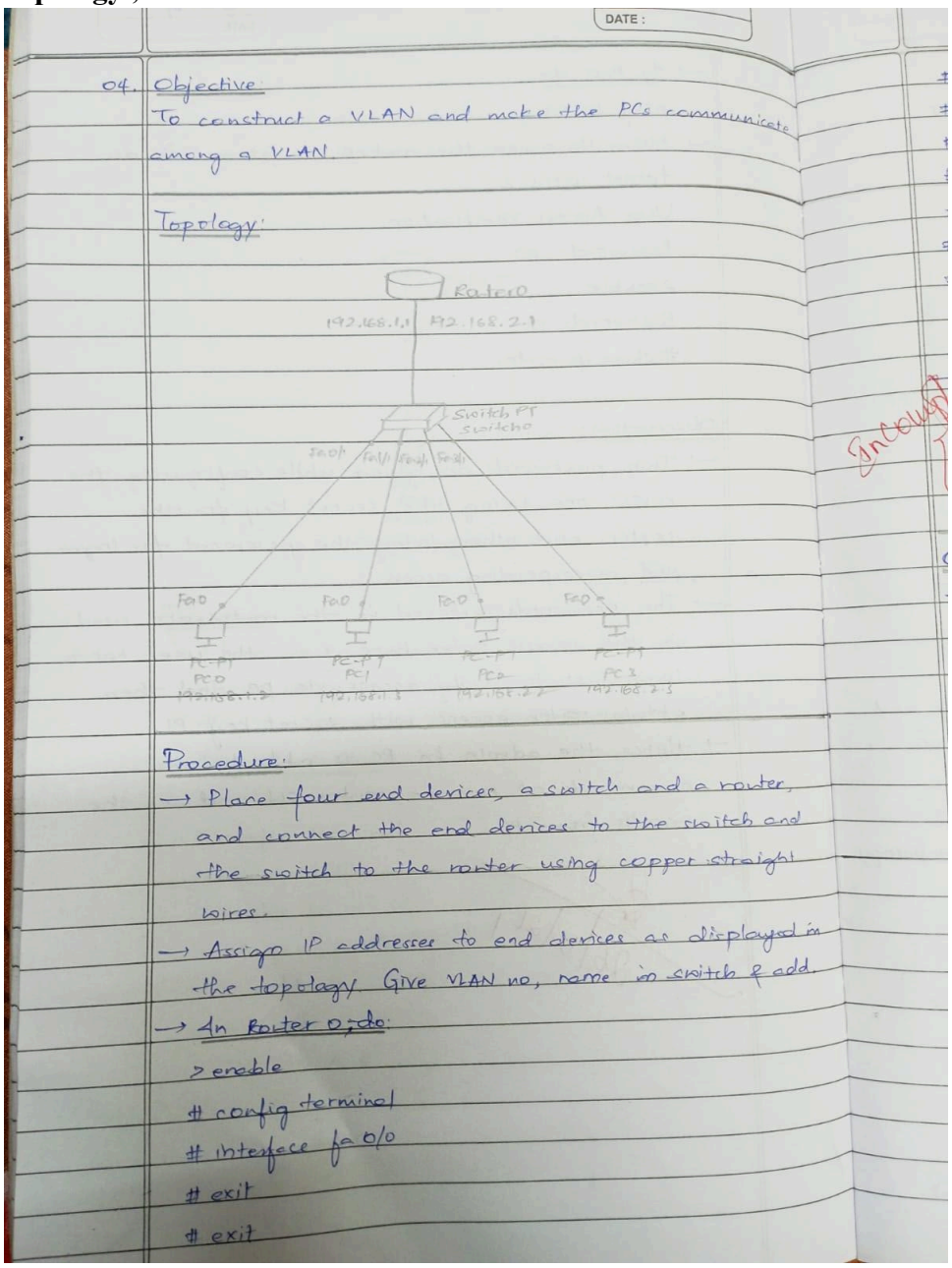



Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.2
Pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Ping statistics 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:



DATE :

04. Objective:
To construct a VLAN and make the PCs communicate among a VLAN.

Topology:

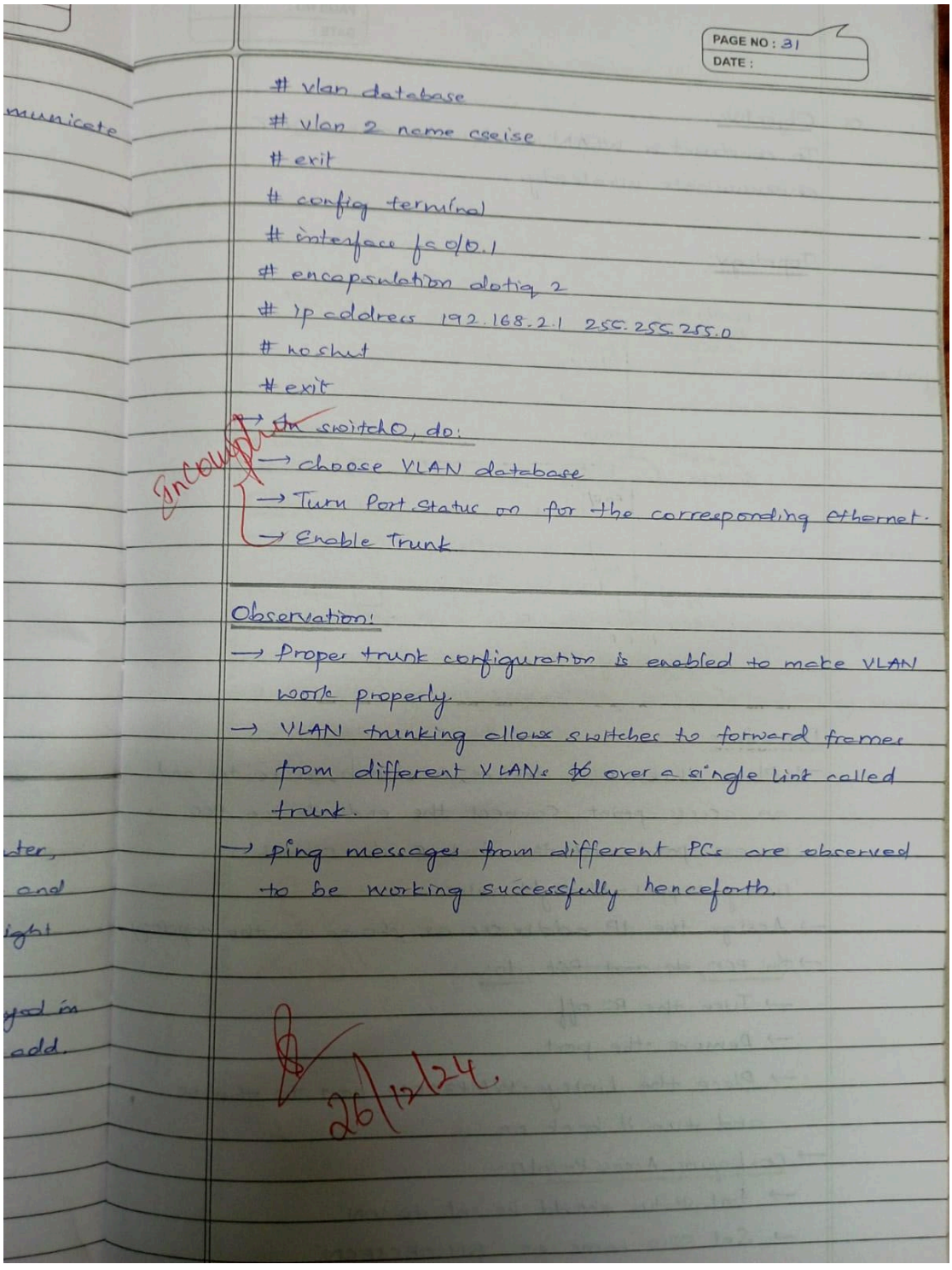
Procedure:

→ Place four end devices, a switch and a router, and connect the end devices to the switch and the switch to the router using copper straight wires.

→ Assign IP addresses to end devices as displayed in the topology. Give VLAN no, name in switch & add.

→ In Router 0 cli:

> enable
config terminal
interface fa 0/0
exit
exit



PAGE NO : 31
DATE :

vlan database
vlan 2 name ccelse
exit
config terminal
interface fa 0/0.1
encapsulation dot1q 2
ip address 192.168.2.1 255.255.255.0
no shut
exit

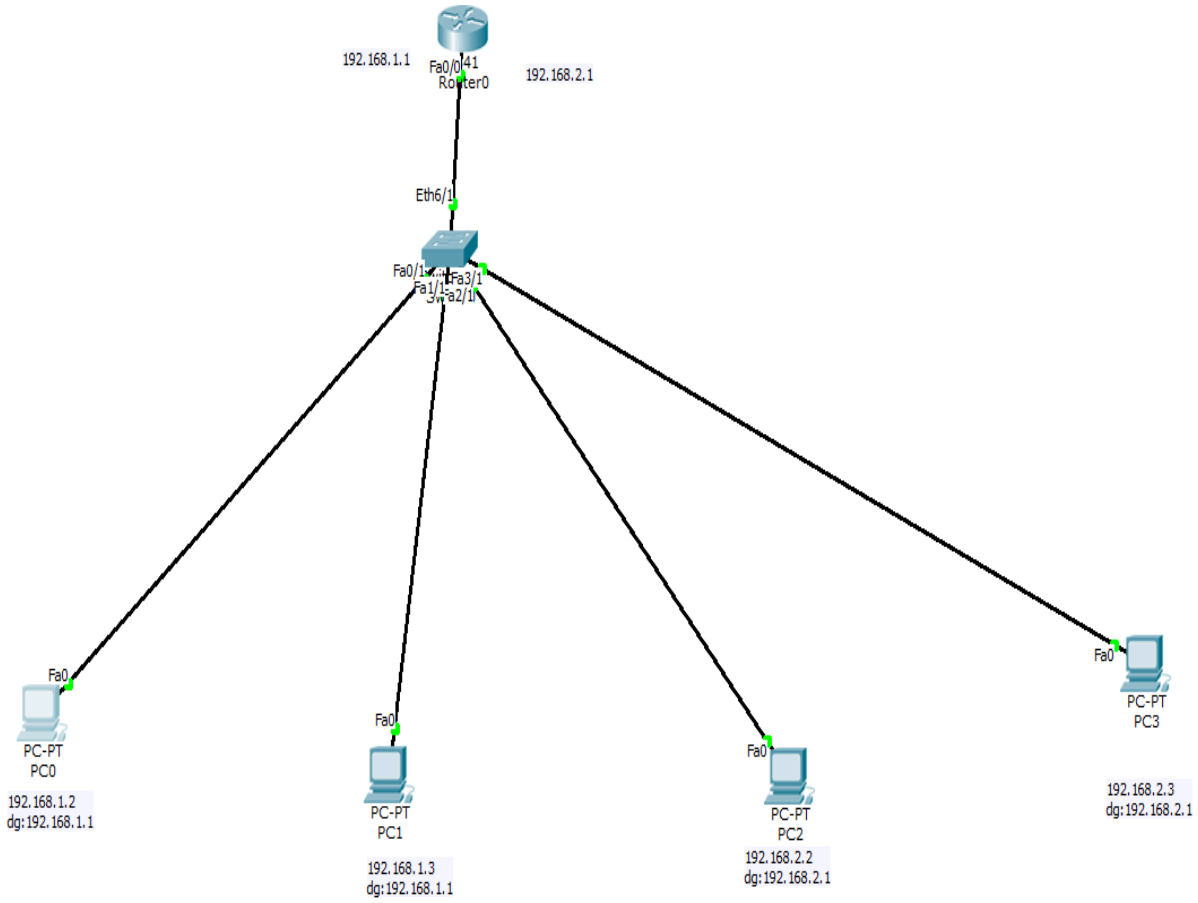
→ switch0, do:
→ choose VLAN database
→ Turn Port Status on for the corresponding Ethernet.
→ Enable Trunk

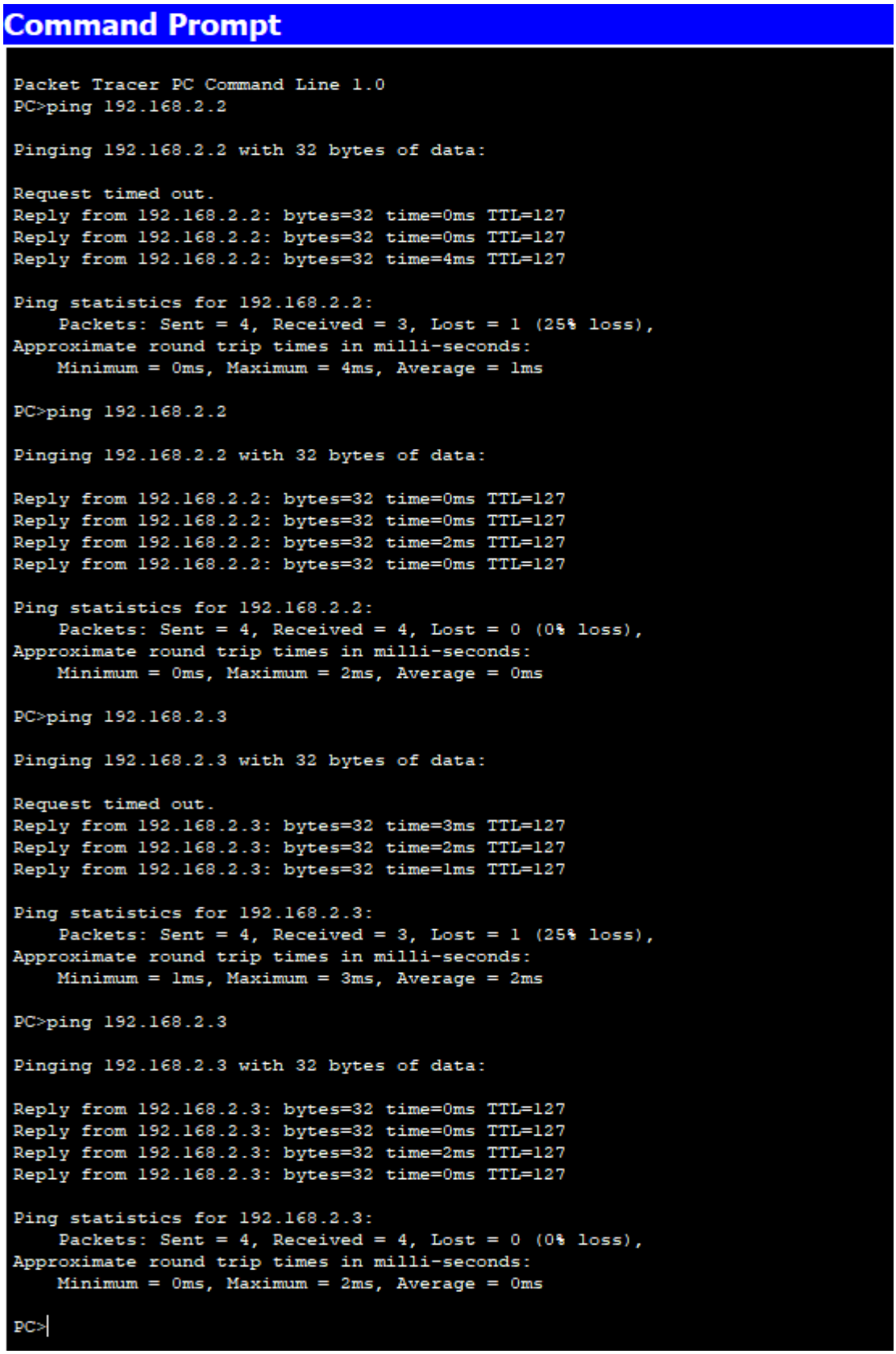
Observation:

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

26/12/24.

Screen Shots:

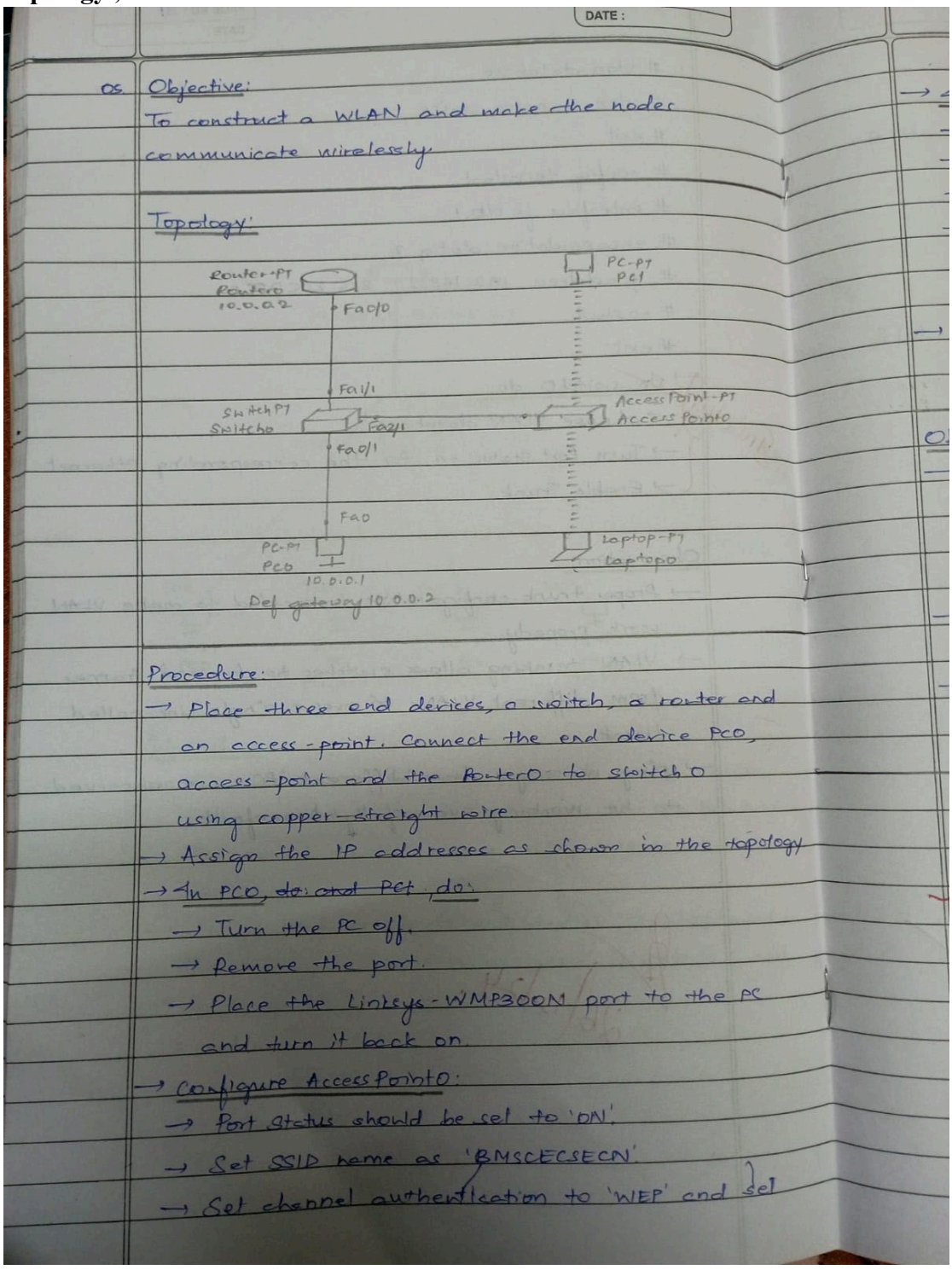

192.168.1.1
Fa0/0
Router0
192.168.2.1
Eth6/1
Fa0/1
Fa1/1
Fa2/1
Fa3/1
Fa0
PC-PT
PC0
192.168.1.2
dg:192.168.1.1
Fa0
PC-PT
PC1
192.168.1.3
dg:192.168.1.1
Fa0
PC-PT
PC2
192.168.2.2
dg:192.168.2.1
Fa0
PC-PT
PC3
192.168.2.3
dg:192.168.2.1


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 :

OS

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

Topology:

Procedure:

→ Place three end devices, a switch, a router and an access-point. Connect the end device PC0, access-point and the Router0 to switch 0 using copper-straight wire.

→ Assign the IP addresses as shown in the topology

→ In PC0, do: and PC1, do:

→ Turn the PC off.

→ Remove the port.

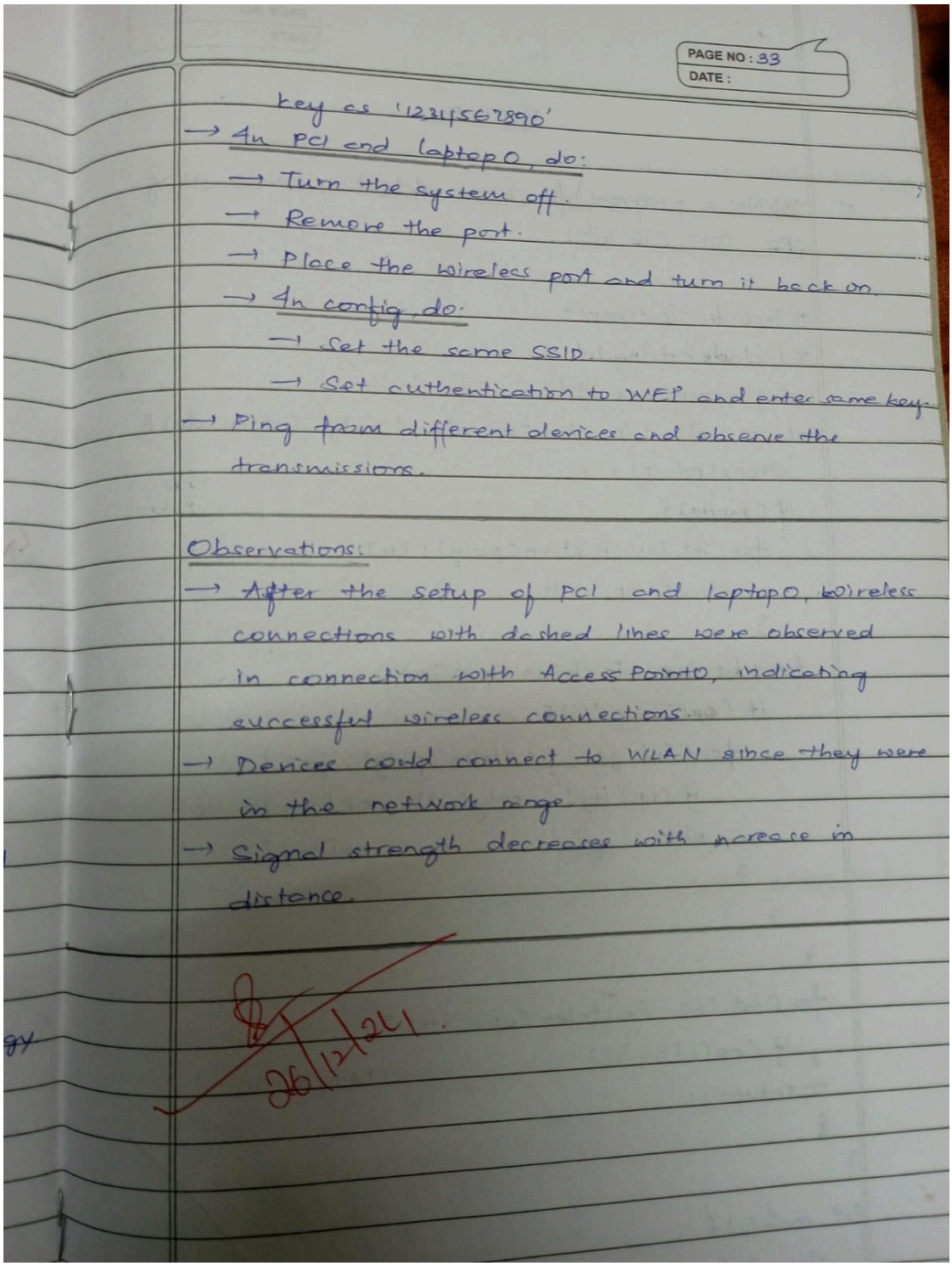
→ Place the Linksys-WMP300N port to the PC and turn it back on.

→ Configure Access Point0:

→ Port status should be set to 'ON'.

→ Set SSID name as 'BMSCECSECN'

→ Set channel authentication to 'WEP' and set



PAGE NO : 33
DATE :

key as '1234567890'

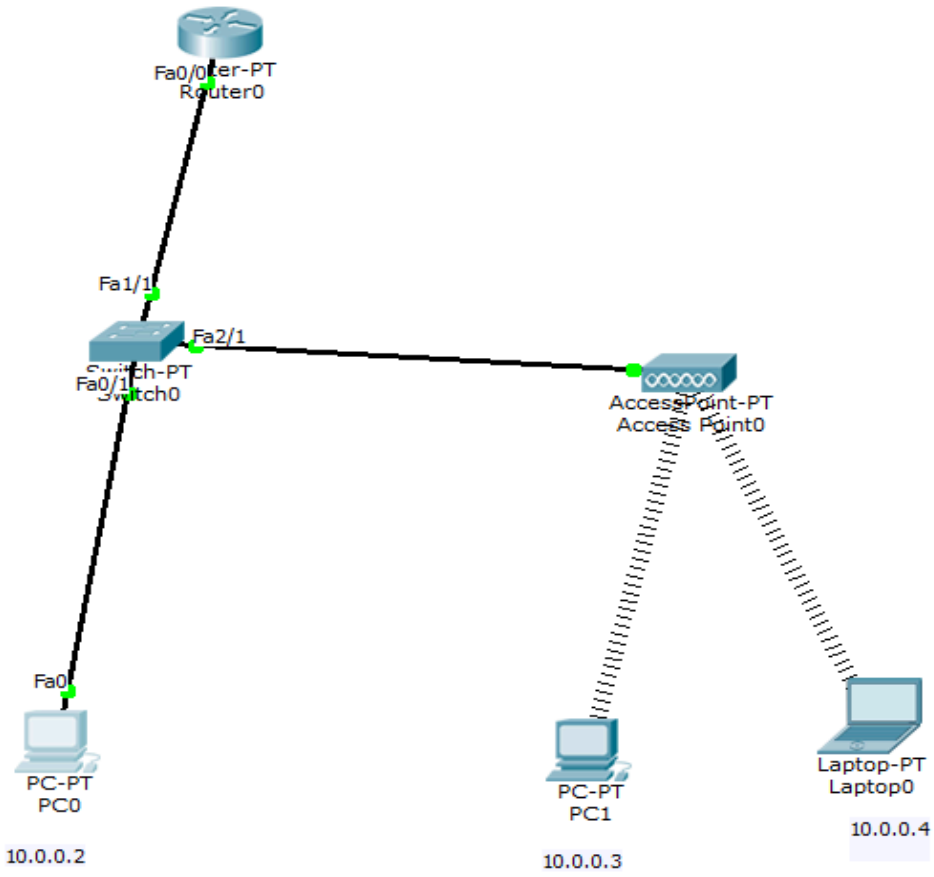
→ In PC1 and laptop0, do:
  → Turn the system off.
  → Remove the port.
  → Place the wireless port and turn it back on.
  → In config, do:
    → Set the same SSID.
    → Set authentication to WEP and enter same key.
→ Ping from different devices and observe the transmissions.

Observations:

→ After the setup of PC1 and laptop0, wireless connections with dashed lines were observed in connection with Access Point0, indicating successful wireless connections.
→ Devices could connect to WLAN since they were in the network range.
→ Signal strength decreases with increase in distance.

26/12/24.

Screen Shots:



PC0

Physical | Config | Desktop | Custom Interface

Command Prompt

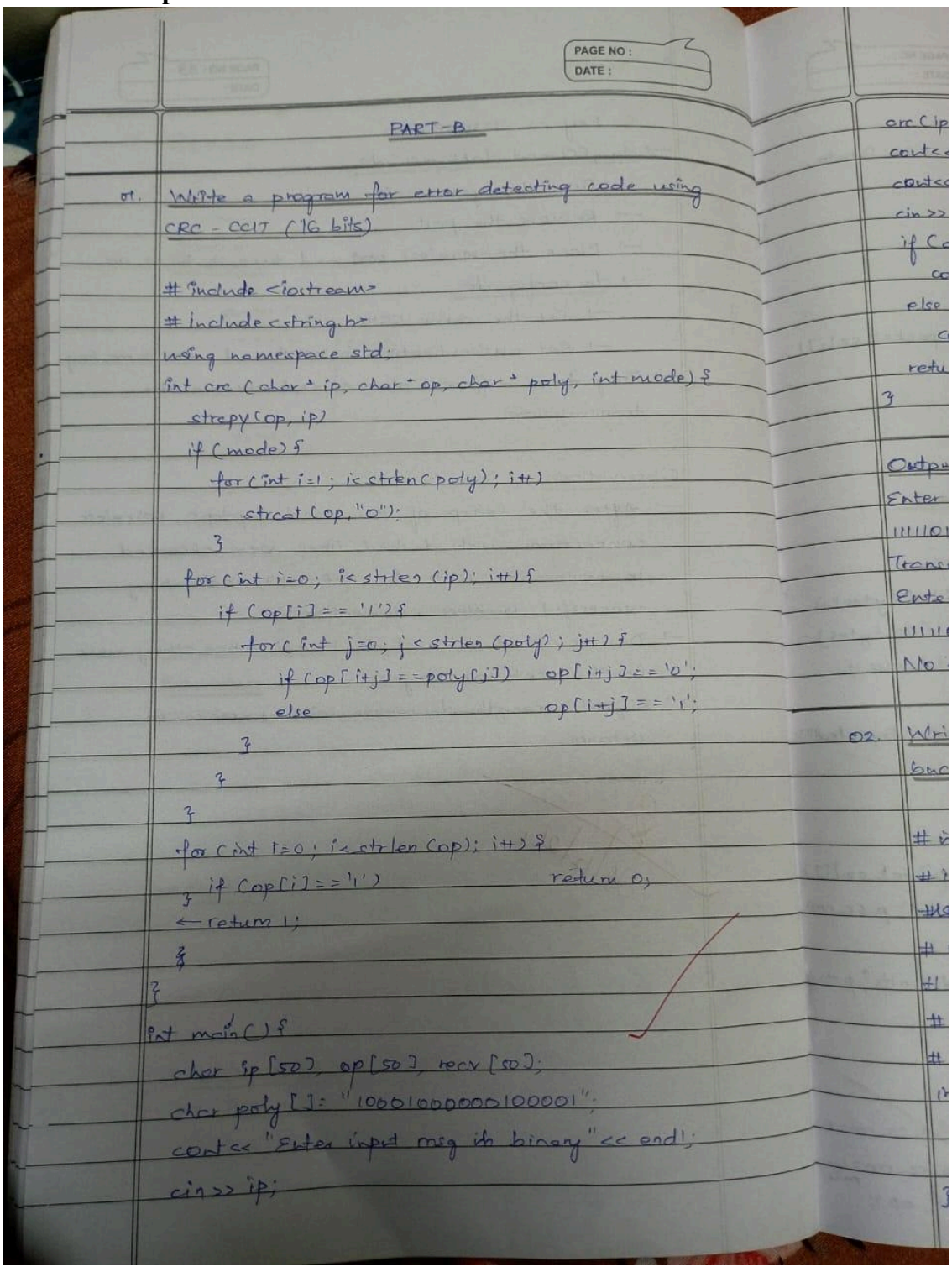
```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time=22ms TTL=128
Reply from 10.0.0.3: bytes=32 time=6ms TTL=128
Reply from 10.0.0.3: bytes=32 time=3ms TTL=128
Reply from 10.0.0.3: bytes=32 time=7ms TTL=128

Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 22ms, Average = 9ms

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=19ms TTL=128
Reply from 10.0.0.4: bytes=32 time=5ms TTL=128
Reply from 10.0.0.4: bytes=32 time=6ms TTL=128
Reply from 10.0.0.4: bytes=32 time=7ms 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 = 5ms, Maximum = 19ms, Average = 9ms

PC>
```

PART-B

Program 14

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

Code and Output:

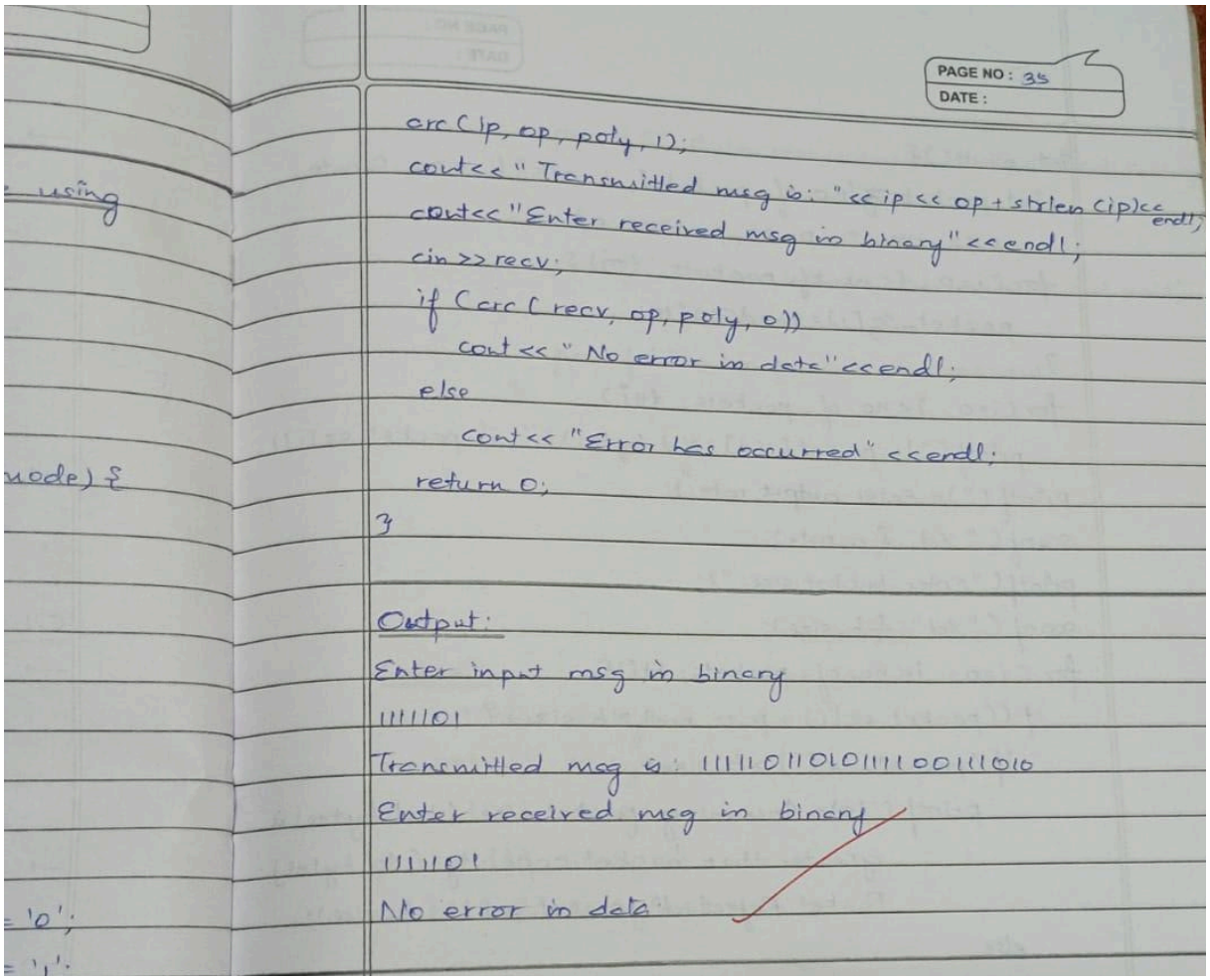


PART-B

01. Write a program for error detecting code using CRC - CCIT (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;
}
int main(){
  char ip[50], op[50], recv[50];
  char poly[]: "10001000000100001";
  cout<< "Enter input msg in binary"<< endl;
  cin>> ip;
```

02.



PAGE NO : 35
DATE :

```
crc(ip, op, poly, 1);
cout<< "Transmitted msg is: "<<ip<<op+strlen(ip)<<endl;
cout<<"Enter received msg in binary"<<endl;
cin>>recv;
if(crc(recv, op, poly, 0))
    cout<<"No error in data"<<endl;
else
    cout<<"Error has occurred"<<endl;
return 0;
}
```

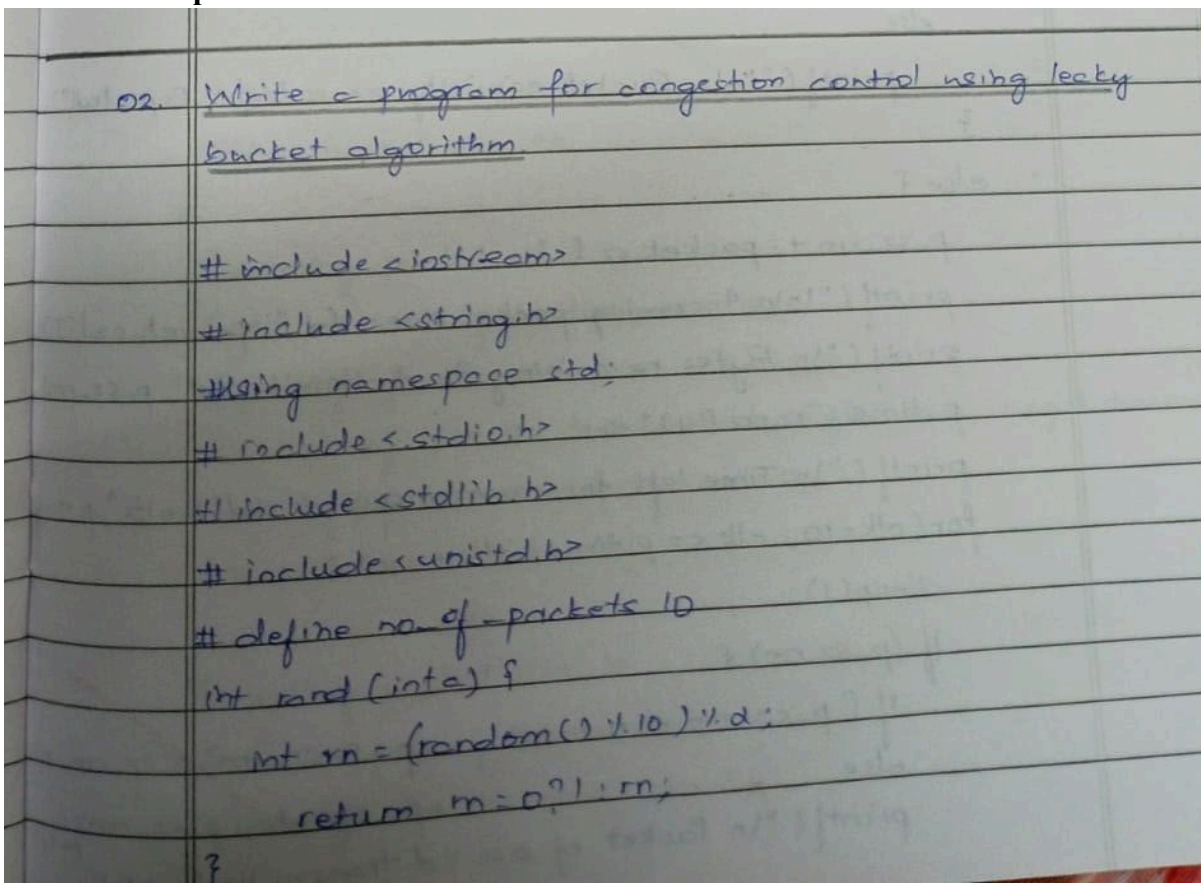
Output:

```
Enter input msg in binary
111101
Transmitted msg is: 111101101011110011110
Enter received msg in binary
111101
No error in data
```

Program 15

Write a program for congestion control using Leaky bucket algorithm.

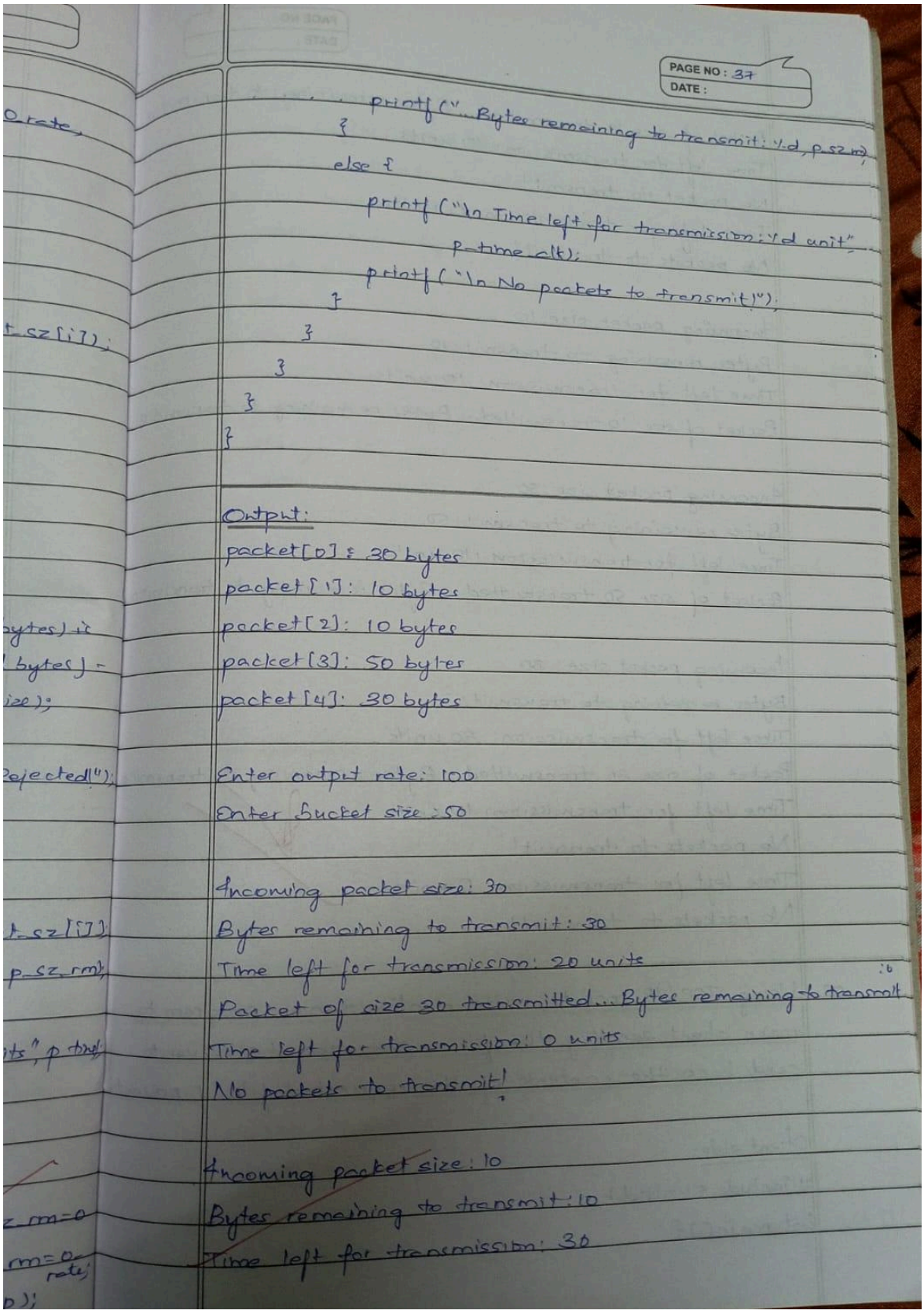
Code and Output:



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

```
int main(){
  int packets[no_of_packets], i, clk, b_size, o_rate,
      p_sz_rm=0, p_sz, p_time, op;
  for(i=0; i<no_of_packets; ++i){
    packet_sz[i] = rand(6)*10;
  }
  for(i=0; i<no_of_packets; ++i)
    printf("\n packet[%d]: %d bytes\t", i, packet_sz[i]);
  printf("\n Enter output rate:");
  scanf("%d", &o_rate);
  printf("Enter bucket size:");
  scanf("%d", &b_size);
  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\n Incoming packet size (%d bytes) is
               greater than bucket capacity (%d bytes) -
               Packet Rejected", packet_sz[i], b_size);
      else
        printf("\n\n Bucket capacity exceeded - Rejected!!");
    }
    else{
      p_sz_rm += packet_sz[i];
      printf("\n\n Incoming packet size: %d", packet_sz[i]);
      printf("\n Bytes remaining to transmit: %d", p_sz_rm);
      p_time = rand(4)*10;
      printf("\n Time 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=o_rate;
          printf("\n Packet of size %d transmitted", op);
```



```
        printf("... Bytes remaining to transmit: %d", p_sz_rm);
      }
      else {
        printf("\n Time left for transmission: %d unit",
                p_time_clk);
        printf("\n No 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 output rate: 100
Enter 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 transmit!

Incoming packet size: 10
Bytes remaining to transmit: 10
Time left for transmission: 30
```

PAGE NO :
DATE :

Packet of size 10 transmitted.. Bytes remaining to transmit:0
Time left for transmission: 10 units
No packet 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

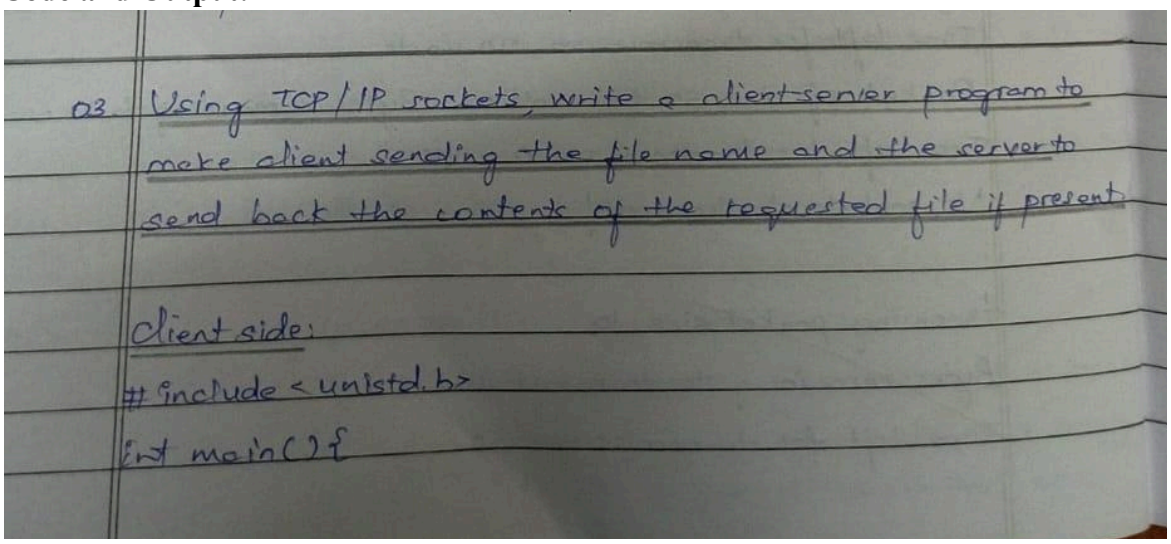
Incoming packet size: 50
Bytes remaining to transmit: 50
Time left for transmission: 10 units
Packet of size 50 transmitted.. Bytes remaining to transmit:0

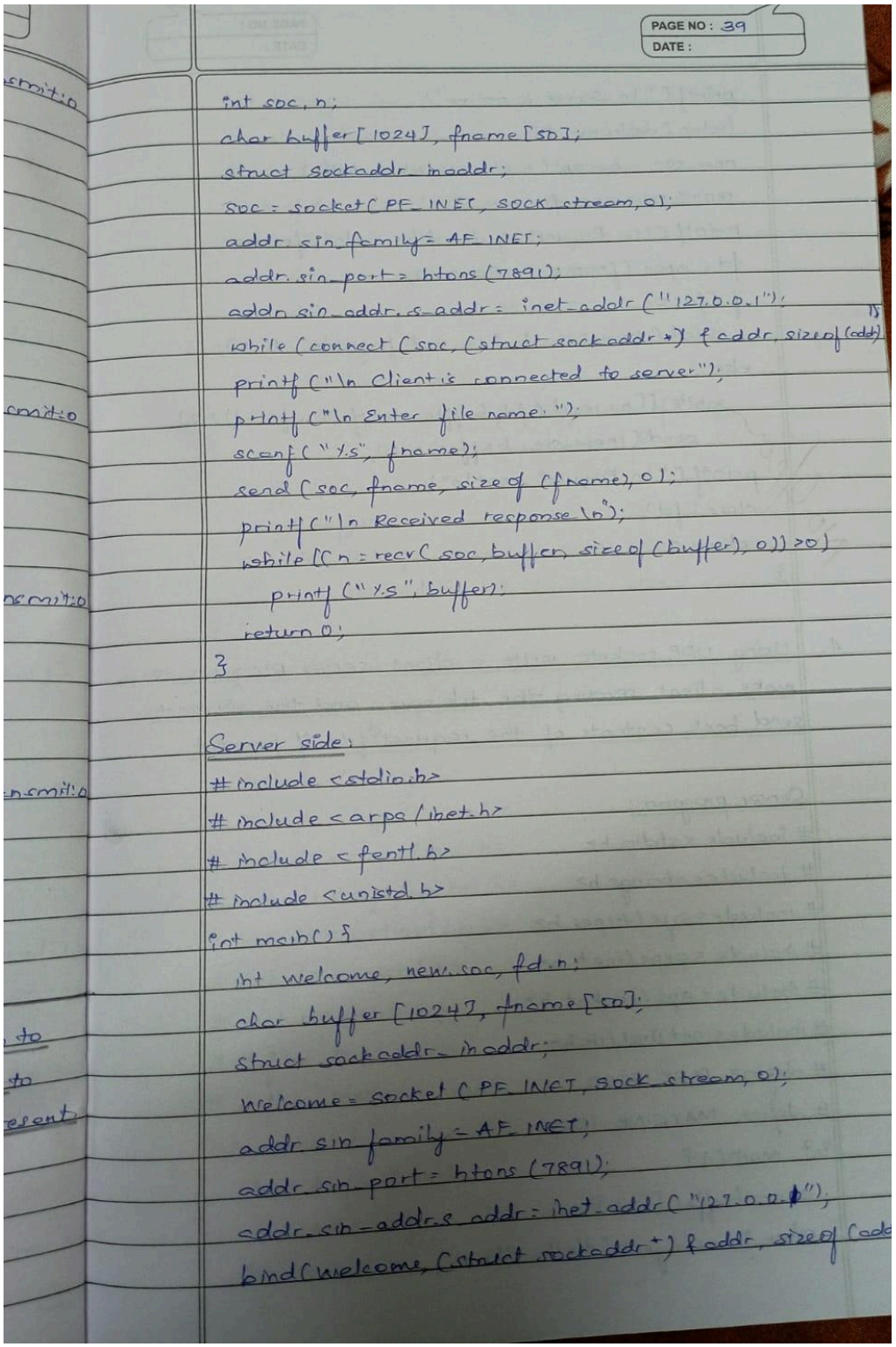
Incoming packet size: 30
Bytes remaining to transmit: 30
Time left for transmission: 30 units
Packet of size 30 transmitted.. Bytes remaining to transmit:0
Time left for transmission: 10 units
No packets to transmit!
Time left for transmission: 0 units
No packets to transmit!

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:



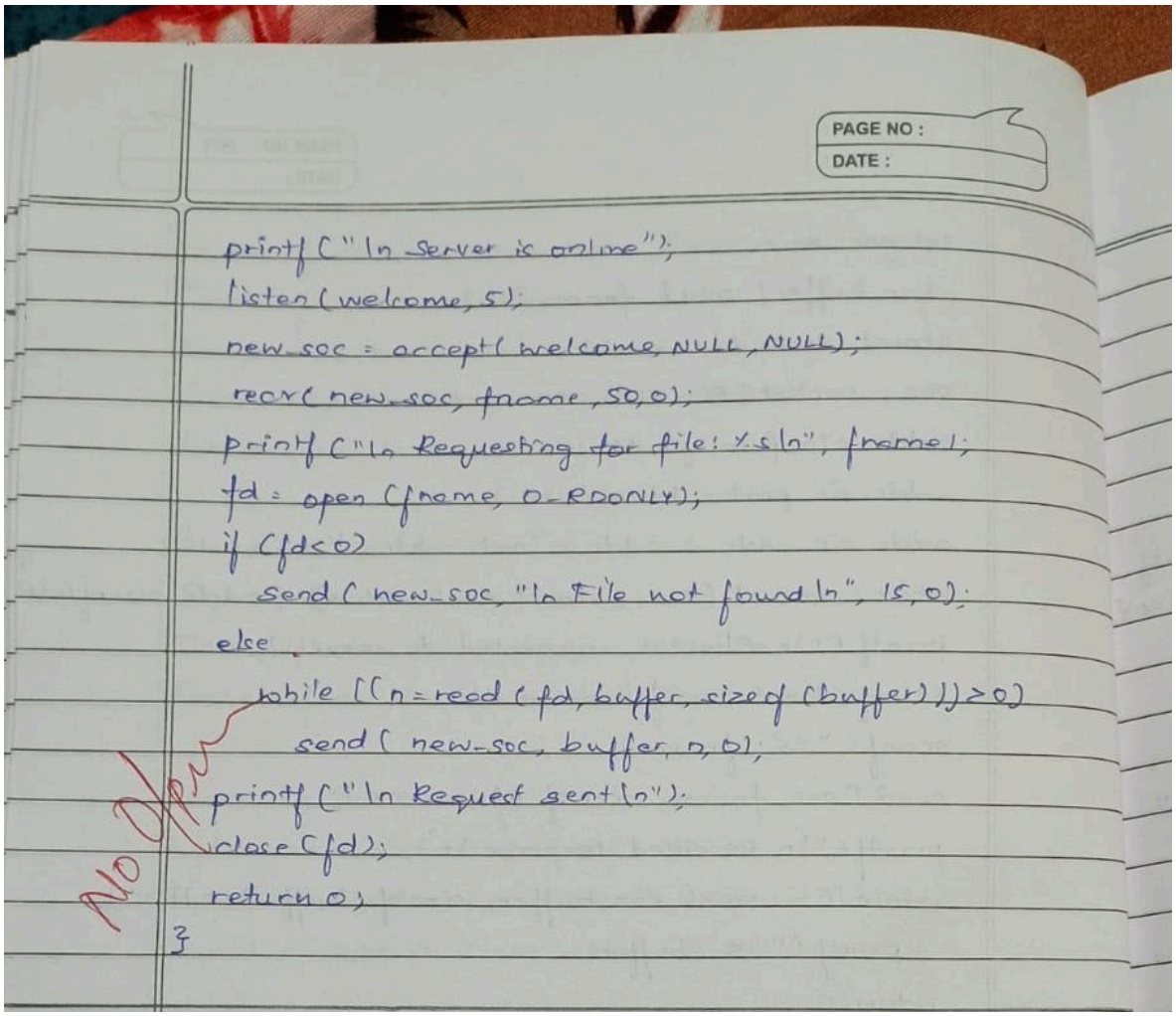


PAGE NO : 39
DATE :

```
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("\n Client is connected to server");
printf("\n Enter file name: ");
scanf("%s", fname);
send(soc, fname, sizeof(fname), 0);
printf("\n Received response \n");
while((n = recv(soc, buffer, sizeof(buffer), 0)) > 0)
    printf("%s", buffer);
return 0;
}
```

Server side:

```
#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(add
```



```
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, "\n File not found\n", 15, 0);
else
  while ((n=read(fd, buffer, sizeof(buffer)))>0)
    send(new_soc, buffer, n, 0);
printf("\n Request sent\n");
close(fd);
return 0;
}
```

Output:

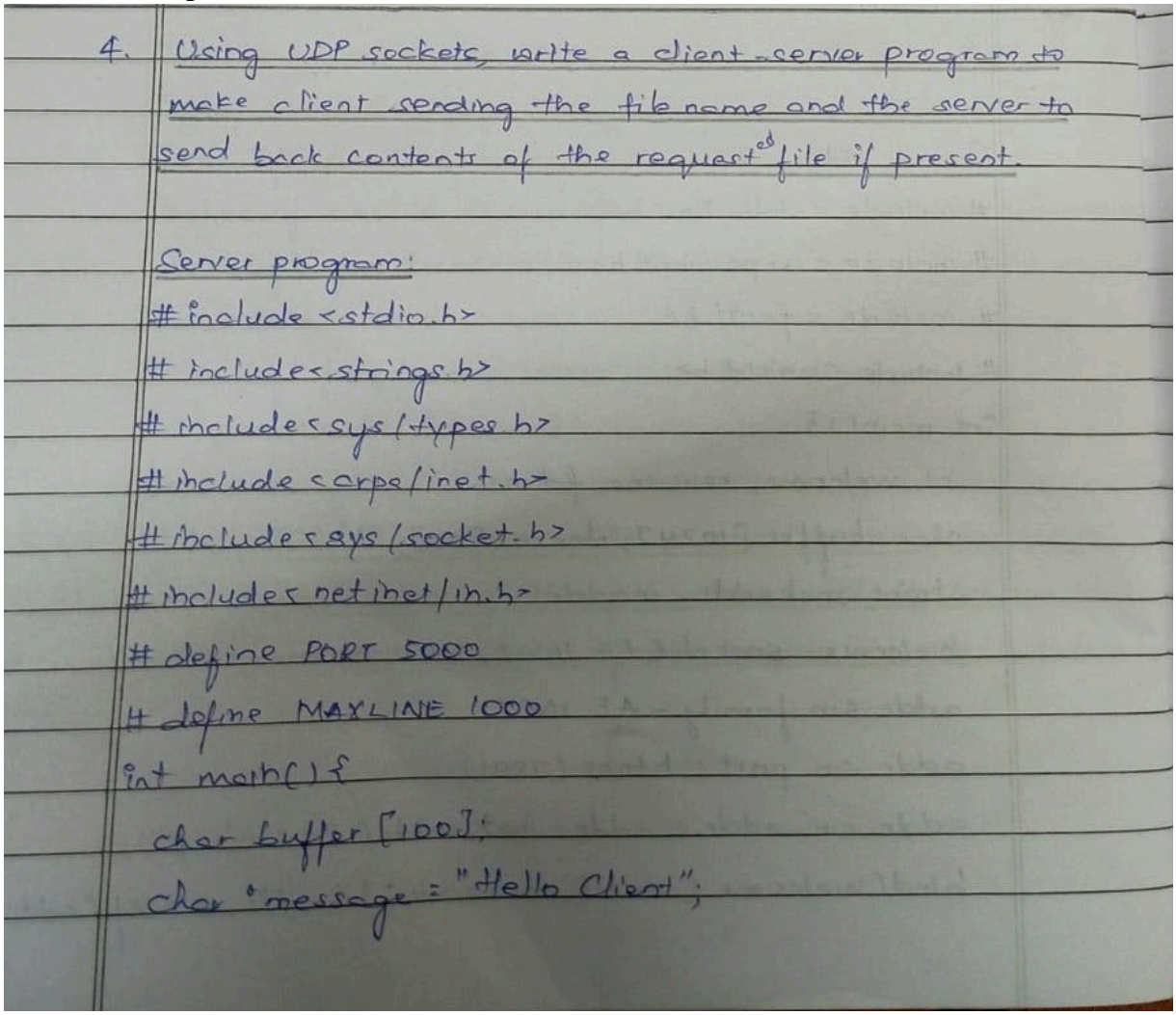
Server is online
Requesting for file: test.txt
Request sent

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

Server program:

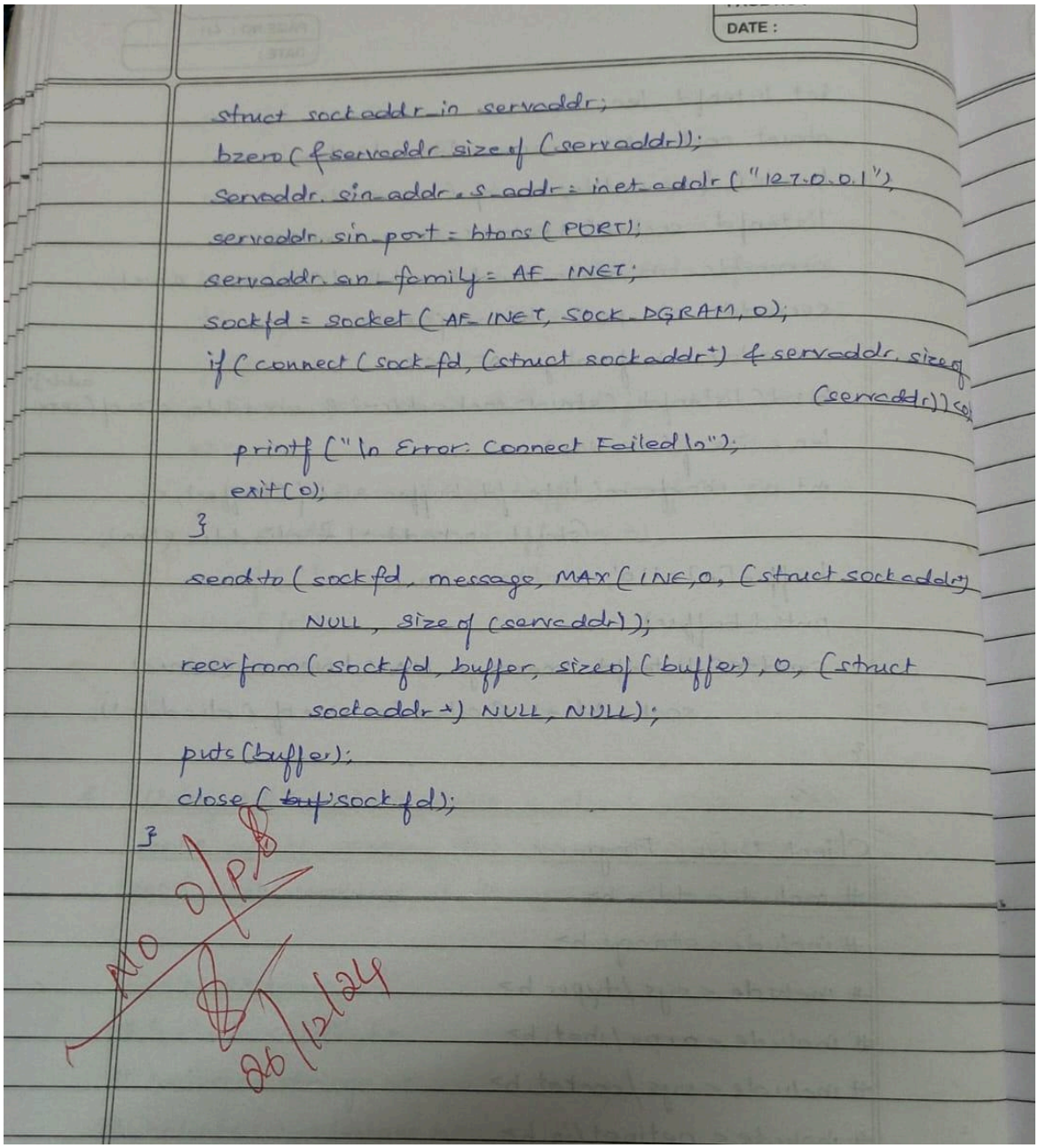
```
#include <stdio.h>
#include<strings.h>
#include<sys/types.h>
#include<arpa/inet.h>
#include<sys/socket.h>
#include<netinet/in.h>
#define PORT 5000
#define MAXLINE 1000
int main(){
  char buffer[100];
  char *message = "Hello Client";
```

PAGE NO : 41
DATE :

```
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));
}
```

Client 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 MAXLINE 1000
int main() {
  char buffer[100];
  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("\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);
}
```

A10 O/P
B
B 26/12/24

Server Output:
Server is online
Hello Server

Client Output:
Hello Client