VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB RECORD

Computer Network Lab (22CS4PCCON)

Submitted by

Sonal (1BM22CS286)

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 (22CS4PCCON)" carried out by **Sonal(1BM22CS286),** who is 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

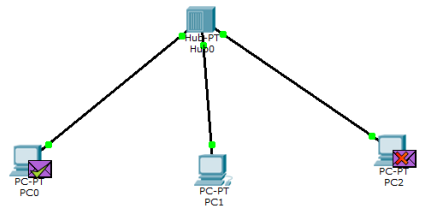
Index

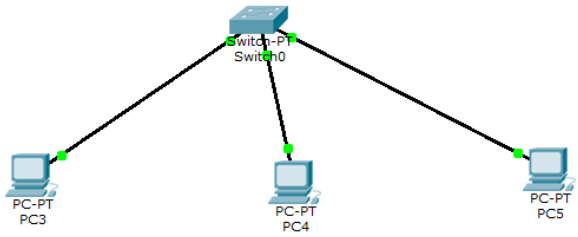
Github Link: sonalkolekar/CN

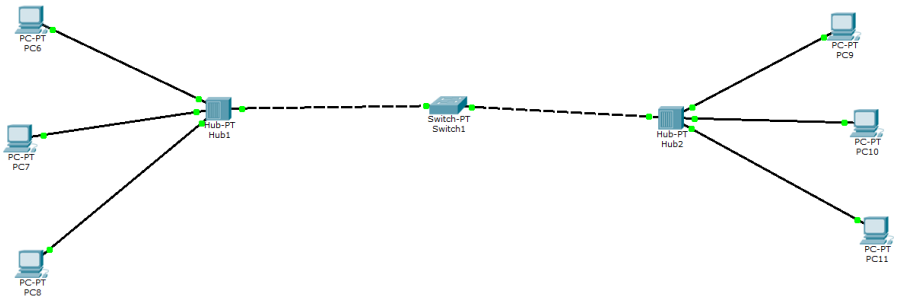
Program 1

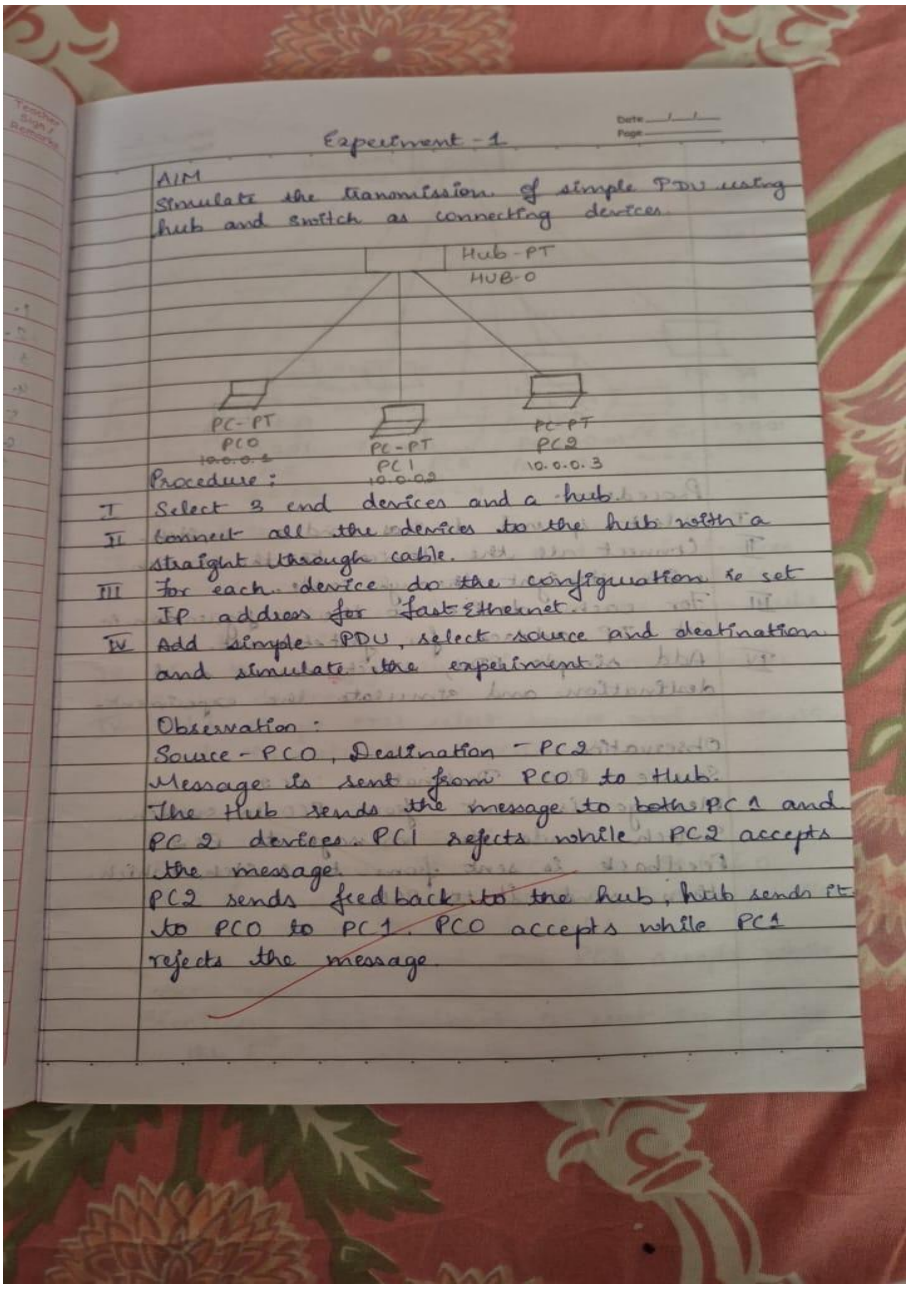
Aim of the program:
Simulate sending a simple PDU from source to destination using hub and switch as connecting devices

Procedure along with the topology and observation:









Experiment - 1

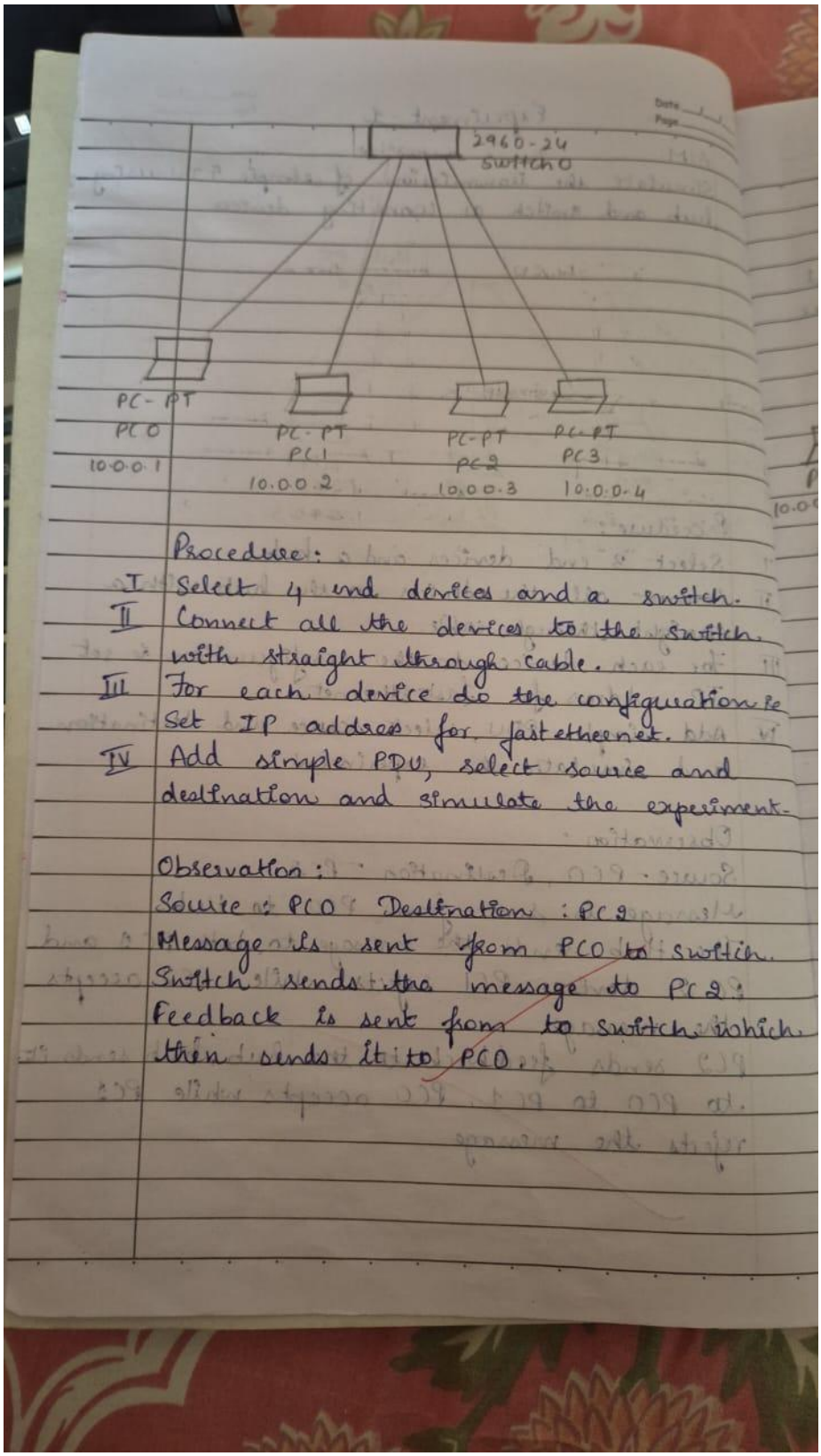
Date / /
Page

AIM
Simulate the transmission of simple PDU using hub and switch as connecting devices.

Procedure:

I Select 3 end devices and a hub.
II Connect all the devices to the hub with a straight through cable.
III For each device do the configuration to set IP address for fast Ethernet.
IV Add simple PDU, select source and destination and simulate the experiment.

Observation:
Source - PC0, Destination - PC2
Message is sent from PC0 to Hub.
The Hub sends the message to both PC1 and PC2 devices. PC1 rejects while PC2 accepts the message.
PC2 sends feedback to the hub, hub sends it to PC0 to PC1. PC0 accepts while PC1 rejects the message.



Procedure:

I Select 4 end devices and a switch.

II Connect all the devices to the switch with straight through cable.

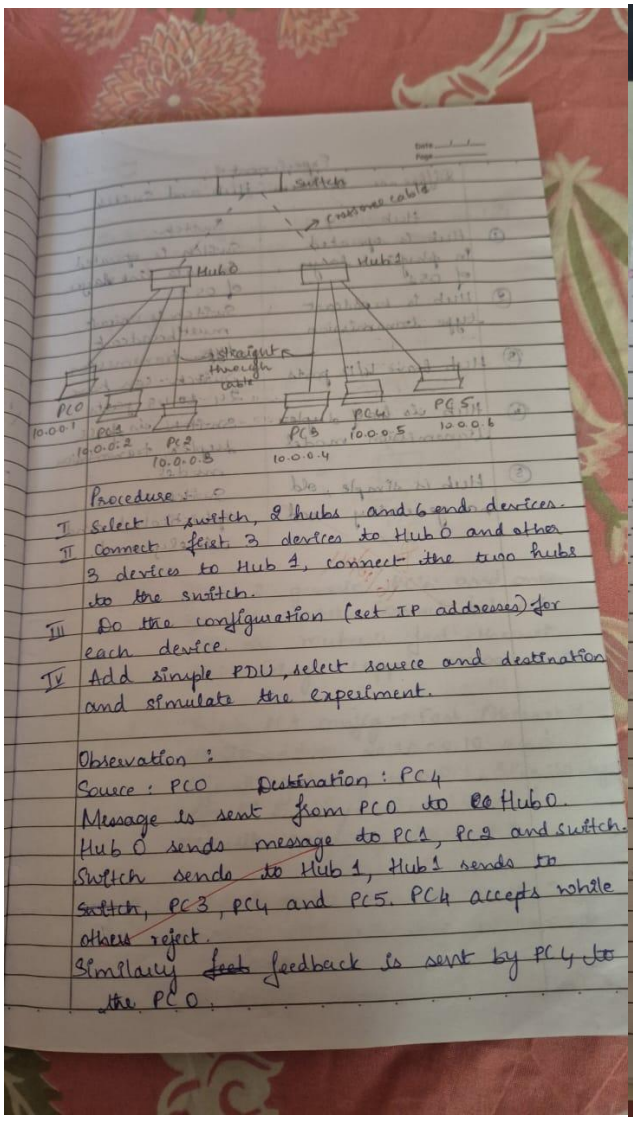
III For each device do the configuration ie Set IP address for fastethernet.

IV Add simple PDU, select source and destination and simulate the experiment.

Observation:

Source : PC0 Destination : PC2

Message is sent from PC0 to switch. Switch sends the message to PC2. Feedback is sent from to switch which then sends it to PC0.



Procedure :

I Select 1 switch, 2 hubs and 6 end devices.

II Connect first 3 devices to Hub 0 and other 3 devices to Hub 1, connect the two hubs to the switch.

III Do the configuration (set IP addresses) for each device.

IV Add simple PDU, select source and destination and simulate the experiment.

Observation :

Source : PC0 Destination : PC4

Message is sent from PC0 to Hub 0.

Hub 0 sends message to PC1, PC2 and switch.

Switch sends to Hub 1, Hub 1 sends to PC3, PC4 and PC5. PC4 accepts while others reject.

Similarly feedback is sent by PC4 to the PC0.

Experiment 2.

Difference between Hub and Switch

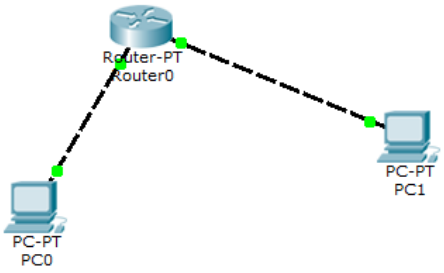
	Hub	Switch
1	Hub is operated in physical layer of OSI	Switch is operated in data link layer of OSI
2	Hub is broadcast type transmission	Switch is unicast-multibroadcast type transmission
3	Hub have 4/12 ports	Switch - can have 24 to 48 ports
4	Hub is half duplex transmission model	Switch is full duplex transmission model.
5	Hub is simple, old not generally used	Switch is so sophisticated and widely used.

16/10/24

Program 2

Aim of the program:
Configure IP address to routers in packet tracer.

Procedure along with the topology and observation:



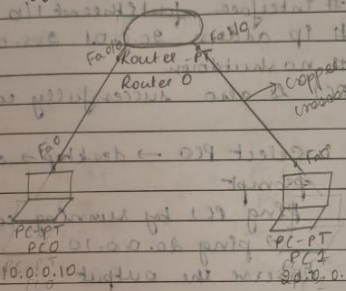
Command Prompt

```
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.10

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=1ms 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 = 1ms, Average = 0ms

PC>
```

Experiment - 2

Aim: Configure IP address to routers in packet tracer. Explore the

Topology:



Procedure:

Step 1: Place 2 generic PC's and one generic router. Connect both PC's to the router's fast ethernet ports using copper crossover wire.

Step 2: Select PC1 config → Fast Ethernet 0 Set IP address as 10.0.0.10 and default gateway 10.0.0.1. Similarly for PC 2 IP 20.0.0.10 and default gateway 20.0.0.1

Step 3: Select router and go to CLI execute the following commands
enable
config terminal
Interface fastEthernet 0/0
ip address 10.0.0.1 255.0.0.0
no shutdown
exit
notice PC0 and router 0 are successfully connected.
Now run for PC1
Interface fastEthernet 1/0
ip address 20.0.0.1 255.0.0.0
no shutdown
PC1 is also successfully connected

Step 4: Select PC0 → desktop → command prompt
Ping PC1 by running command
→ ping 20.0.0.10
Observe the output

Observation:

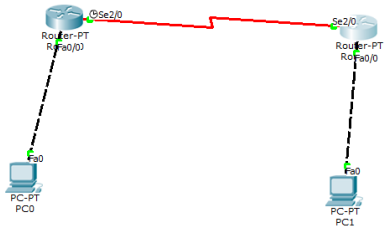
PC0 successfully pings PC1 with 32 bytes of data.

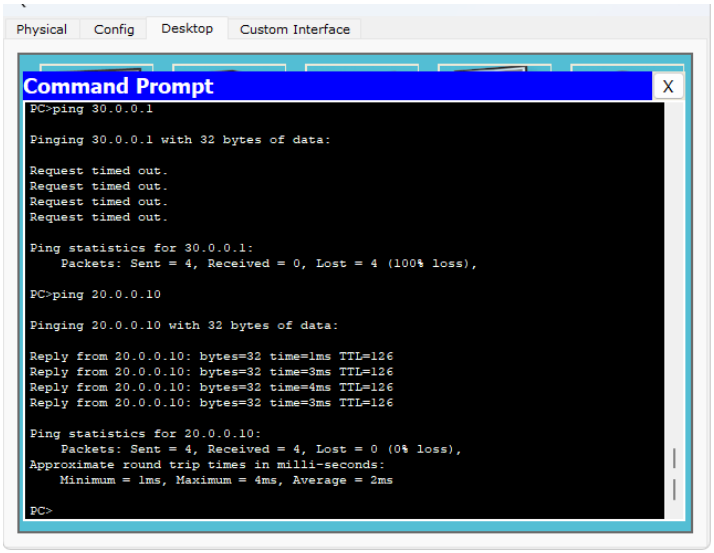
Ping Statistics
Packets:
sent = 4, received = 4, lost = 0

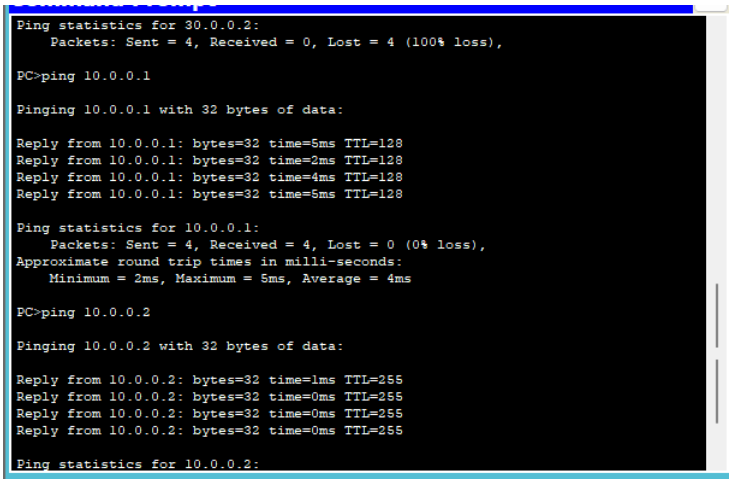
Program 3

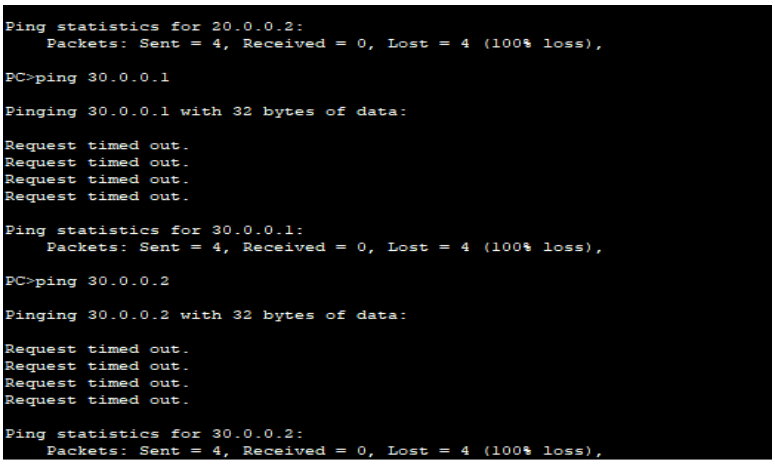
Aim of the program:
Configure default route, static route to the Router.(Part 1)

Procedure along with the topology and observation:





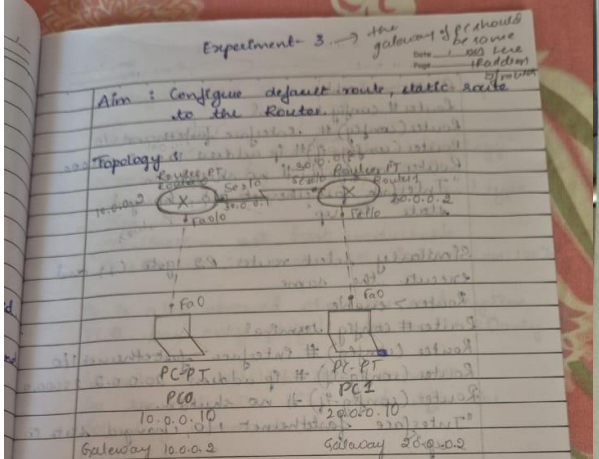

Ping statistics for 30.0.0.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=5ms TTL=128
Reply from 10.0.0.1: bytes=32 time=2ms TTL=128
Reply from 10.0.0.1: bytes=32 time=4ms TTL=128
Reply from 10.0.0.1: bytes=32 time=5ms TTL=128
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 5ms, Average = 4ms
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=1ms 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:


Ping statistics for 20.0.0.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 30.0.0.1
Pinging 30.0.0.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 30.0.0.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 30.0.0.2
Pinging 30.0.0.2 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 30.0.0.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

Experiment 3 → the gateway of PC should be same ip address

Aim : Configure default route, static route to the Router.

Topology :



Procedure :

I Select a generic router R1.

II Connect an end device PC1 to router R1 through parallel connection. fastethernet 0/0.

III Configure PC 1 with ip address 10.0.0.10 and gateway 10.0.0.2

IV Similarly select another generic router R2 and connect an end device PC2 fastethernet 1/0.

V Configure PC 2 with ip address 20.0.0.10 and gateway 20.0.0.2.

Now select router R1 go to CLI and execute the following

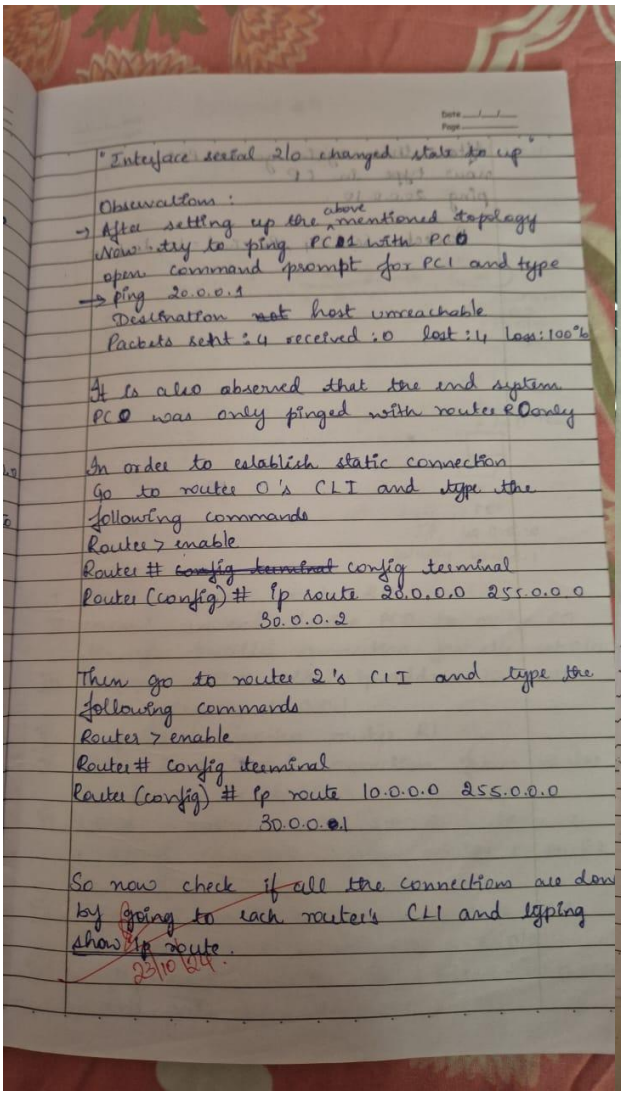
Router > enable
Router # config terminal
Router(config) # interface fastethernet 0/0
Router(config-if) # ip address 10.0.0.2 255.0.0.
Router(config-if) # no shutdown
"Interface fastethernet 0/0, changed state to up"

Similarly select router R2 goto CLI and execute the same
Router > enable
Router # config terminal
Router(config) # interface fastethernet 1/0
Router(config-if) # ip address 20.0.0.2 255.0.0.
Router(config-if) # no shutdown
"Interface fastethernet 1/0, changed state to up"

Hence the connection between Router and end devices is established.

Now connect router R1 with router R2 using serial cable (serially connected)
To setup connection btw routers again,
- 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.
Router(config-if) # no shutdown

Select router R2 and go to CLI
Router(config) # interface serial 2/0
Router(config-if) # ip address 30.0.0.2 255.0
Router(config-if) # no shutdown



"Interface serial 2/0 changed state to up"

Observations:
→ After setting up the above mentioned topology
Now try to ping PC1 with PC0
open command prompt for PC1 and type
→ ping 20.0.0.1
Destination ~~not~~ host unreachable
Packets sent: 4 received: 0 lost: 4 loss: 100%

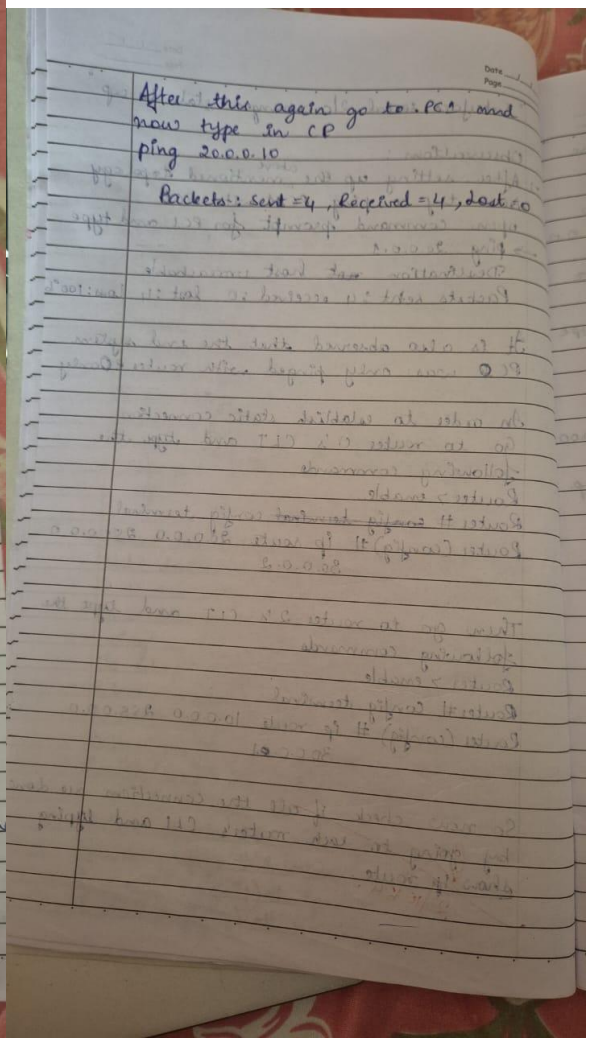
It is also observed that the end system PC0 was only pinged with router R0 only

In order to establish static connection
Go to router 0's CLI and type the following commands
Router > enable
Router # ~~config terminal~~ config terminal
Router (config) # ip route 20.0.0.0 255.0.0.0 30.0.0.2

Then go to router 2's CLI and type the following commands
Router > enable
Router # config terminal
Router (config) # ip route 10.0.0.0 255.0.0.0 30.0.0.1

So now check if all the connections are done by going to each router's CLI and typing show ip route.

23/10/24

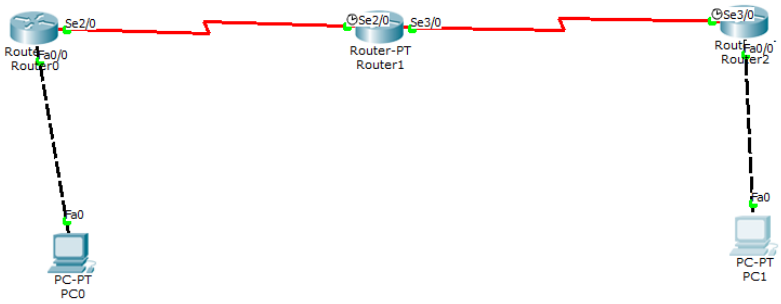


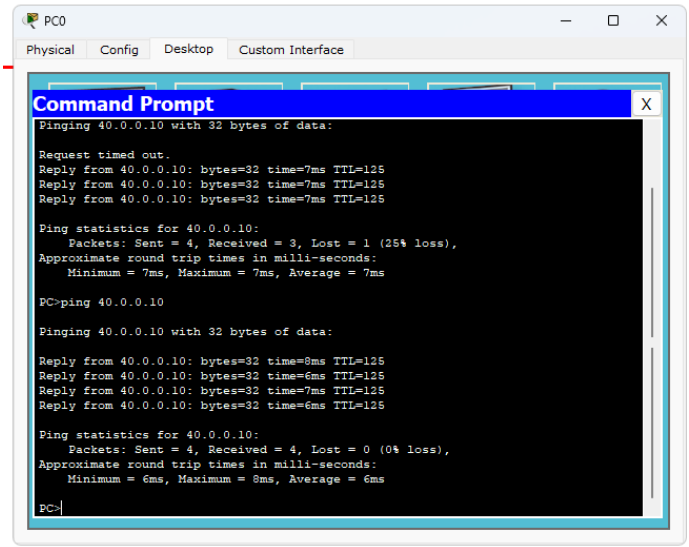
After this again go to PC1 and now type in CP
ping 20.0.0.10
Packets: Sent = 4, Received = 4, Lost = 0

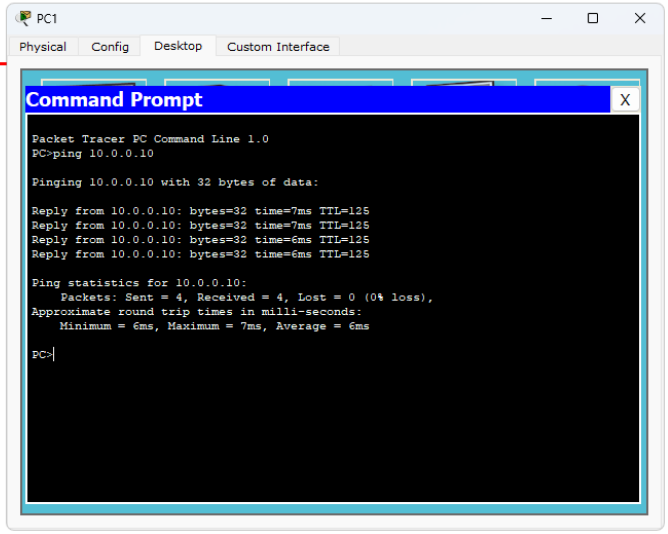
Program 4

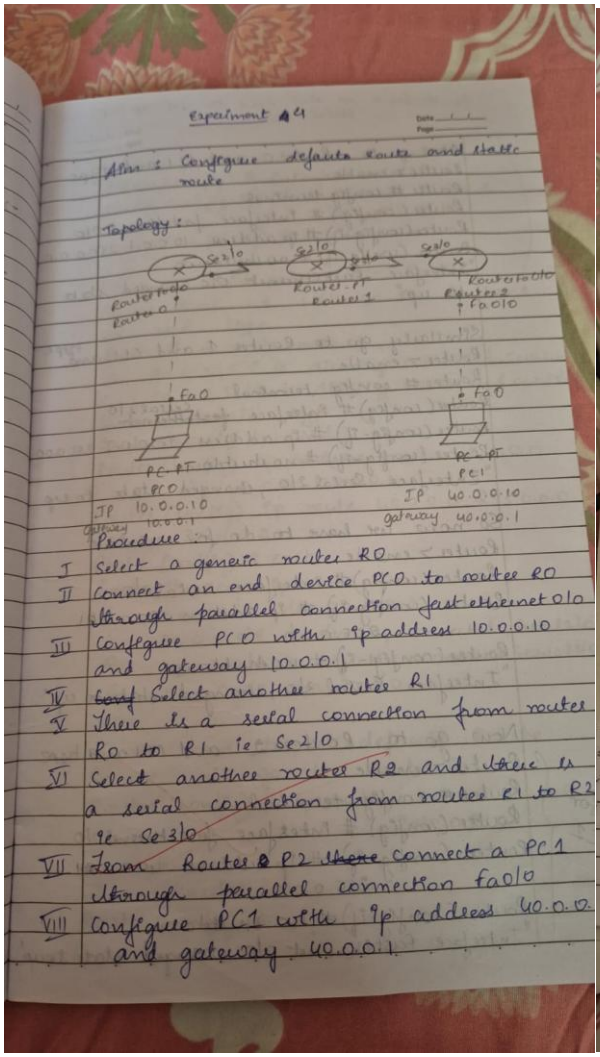
Aim of the program:
Configure default route, static route to the Router.(Part 2)

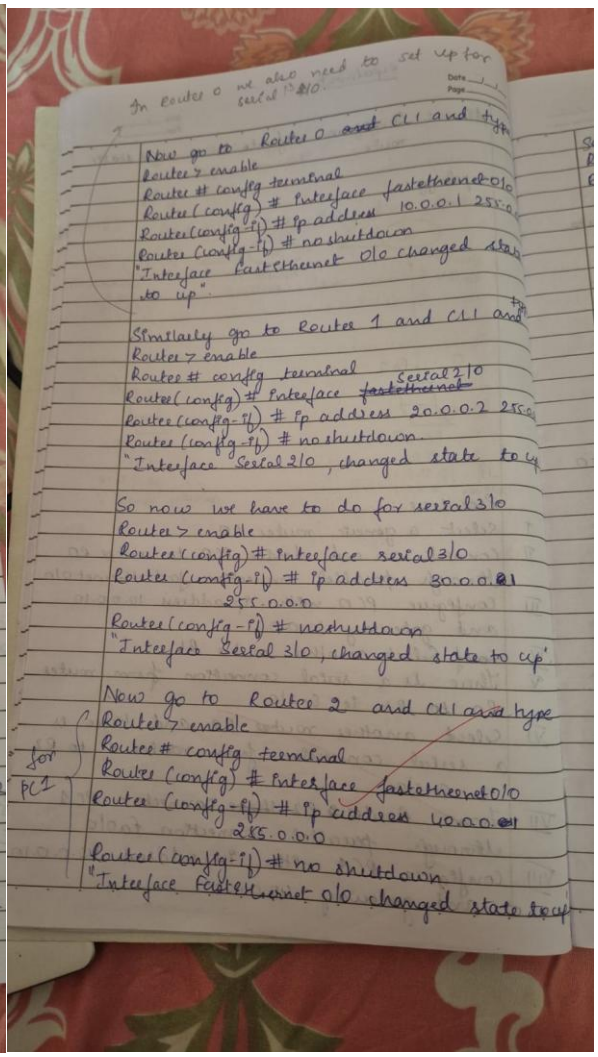
Procedure along with the topology:

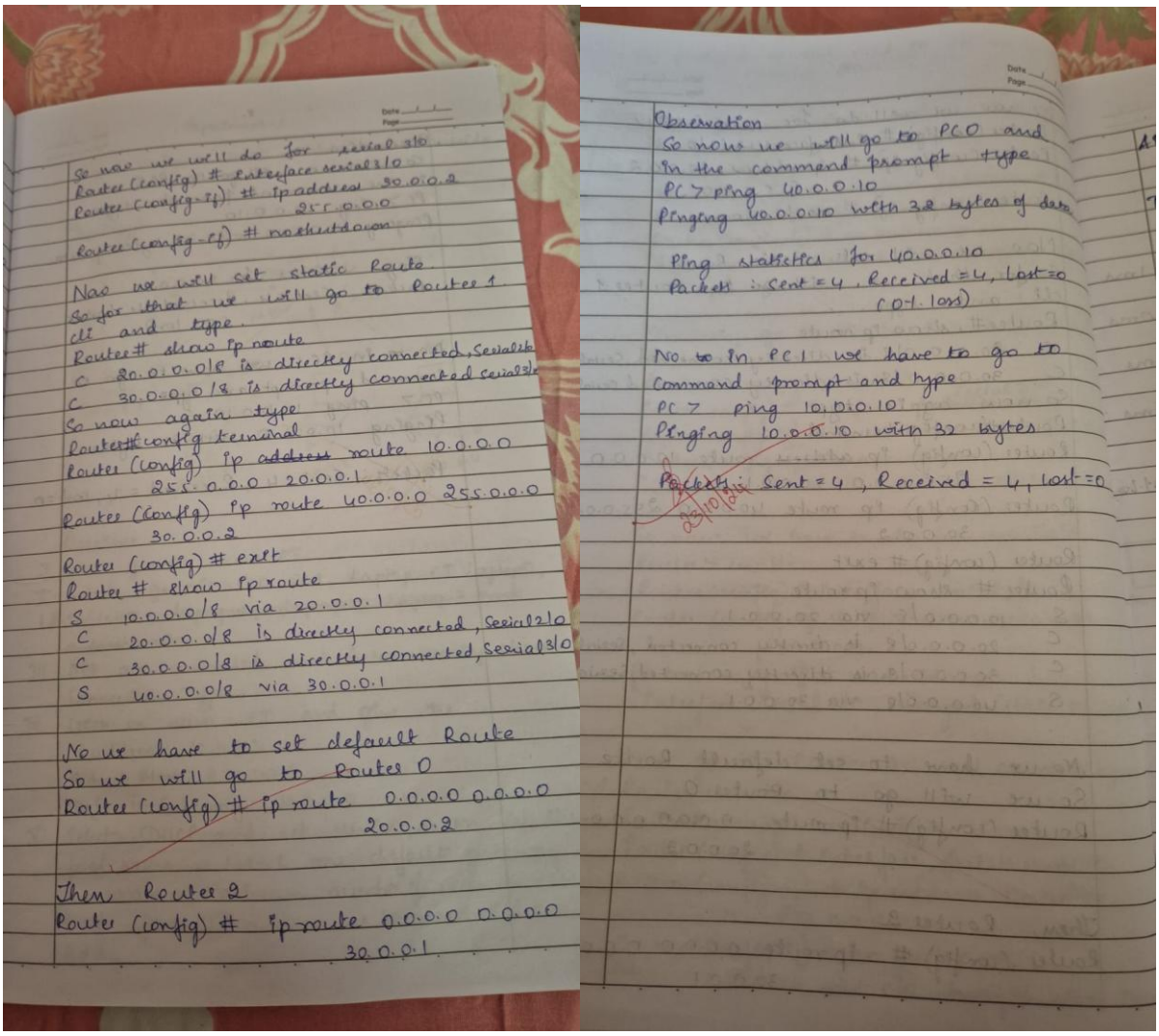

Se2/0
Se2/0
Se3/0
Se3/0
Fa0/0
Router-PT
Router1
Fa0/0
Fa0
Fa0
PC-PT
PC0
PC-PT
PC1


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=7ms 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 = 7ms, 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=8ms TTL=125
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Reply from 40.0.0.10: bytes=32 time=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 = 8ms, Average = 6ms
PC>


PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.10
Pinging 10.0.0.10 with 32 bytes of data:
Reply from 10.0.0.10: bytes=32 time=7ms TTL=125
Reply from 10.0.0.10: bytes=32 time=7ms TTL=125
Reply from 10.0.0.10: bytes=32 time=6ms TTL=125
Reply from 10.0.0.10: bytes=32 time=6ms TTL=125
Ping statistics for 10.0.0.10:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 6ms, Maximum = 7ms, Average = 6ms
PC>


Experiment 4 C1
Aim : Configure default route and static route
Topology :
Se2/0
Se2/0
Se2/0
Router PT
Router 0
Router 1
Router 2
Fa0/0
Fa0
Fa0
PC-PT
PC0
PC1
IP 10.0.0.10
gateway 10.0.0.1
IP 40.0.0.10
gateway 40.0.0.1
Procedure :
I Select a generic router R0
II Connect an end device PC0 to router R0 through parallel connection fastethernet 0/0
III Configure PC0 with ip address 10.0.0.10 and gateway 10.0.0.1
IV Select another router R1
V There is a serial connection from router R0 to R1 ie Se2/0
VI Select another router R2 and there is a serial connection from router R1 to R2 ie Se 3/0
VII From Router 2 R2 connect a PC1 through parallel connection fa0/0
VIII Configure PC1 with ip address 40.0.0.10 and gateway 40.0.0.1


In router 0 we also need to set up for serial 2/0.
Now go to Router 0 and CLI and type
Router> enable
Router# config terminal
Router(config)# interface fastethernet 0/0
Router(config-if)# ip address 10.0.0.1 255.0.0.0
Router(config-if)# no shutdown
"Interface FastEthernet 0/0 changed state to up"
Similarly go to Router 1 and CLI and type
Router> enable
Router# config terminal
Router(config)# interface serial 2/0
Router(config-if)# ip address 20.0.0.2 255.0.0.0
Router(config-if)# no shutdown
"Interface Serial 2/0, changed state to up"
So now we have to do for serial 3/0
Router> enable
Router(config)# interface serial 3/0
Router(config-if)# ip address 30.0.0.1 255.0.0.0
Router(config-if)# no shutdown
"Interface Serial 3/0, changed state to up"
Now go to Router 2 and CLI and type
for PC1
Router> enable
Router# config terminal
Router(config)# interface fastethernet 0/0
Router(config-if)# ip address 40.0.0.1 255.0.0.0
Router(config-if)# no shutdown
"Interface FastEthernet 0/0 changed state to up"



So now we will do for serial 3/0
Router (config) # interface serial 3/0
Router (config-if) # ip address 30.0.0.2
255.0.0.0
Router (config-if) # no shutdown

Now we will set static Route.
So for that we will go to Router 1 cli and type.
Router # show ip route
C 20.0.0.0/8 is directly connected, Serial2/0
C 30.0.0.0/8 is directly connected serial3/0
So now again type
Router # config terminal
Router (config) ip ~~address~~ route 10.0.0.0
255.0.0.0 20.0.0.1
Router (config) ip route 40.0.0.0 255.0.0.0
30.0.0.2
Router (config) # exit
Router # show ip route
S 10.0.0.0/8 via 20.0.0.1
C 20.0.0.0/8 is directly connected, Serial2/0
C 30.0.0.0/8 is directly connected, Serial3/0
S 40.0.0.0/8 via 30.0.0.1

No we have to set default Route
So we will go to Routes 0
Router (config) # ip route 0.0.0.0 0.0.0.0
20.0.0.2

Then Router 2
Router (config) # ip route 0.0.0.0 0.0.0.0
30.0.0.1

Observation
So now we will go to PC 0 and in the command prompt type
PC > ping 40.0.0.10
Pinging 40.0.0.10 with 32 bytes of data

Ping statistics for 40.0.0.10
Packets : Sent = 4, Received = 4, Lost = 0
(0% loss)

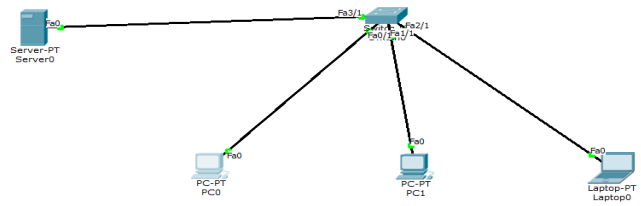
No in PC 1 we have to go to command prompt and type
PC > ping 10.0.0.10
Pinging 10.0.0.10 with 32 bytes

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

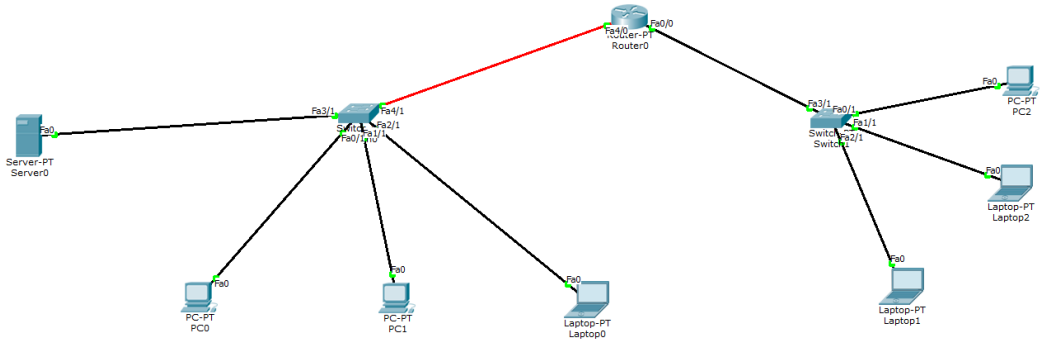
Program 5

Aim of the program:
Configure DHCP within a LAN and outside LAN

Procedure along with the topology and observation:




```
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=2ms 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 = 2ms, Average = 0ms

PC>
```



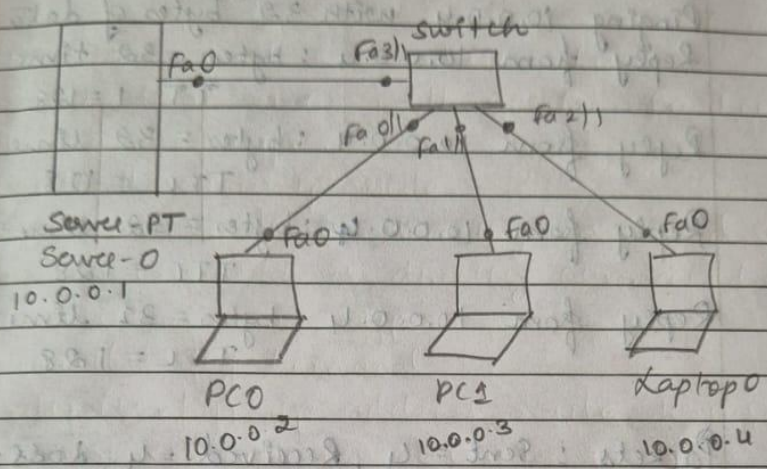

```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms 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 = 0ms, Maximum = 0ms, Average = 0ms

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=2ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
```

Experiment-5.

Date __/__/__
Page

Aim : Configure DHCP within LAN and outside LAN

Topology :



Procedure :

I Select generic server-PT

II Select generic PC0, PC1, Laptop-PT Laptop0,

III Switch0, ~~Laptop-1, laptop2, Laptop-3 and a router-0.~~

III All are connected used Automatically choose Connection Type.

IV Click on server-PT and give the
IP address : 10.0.0.1
Subnet Mask: 255.0.0.0
Default Gateway : 10.0.0.0

V Select DHCP and set service as on. Set the pool Name as switch one, default gateway as 10.0.0.0, maximum number of users as 100 and start IP address as 10.0.0.3. Click Add.

VI Click on each PC, navigate to IP configuration and select DHCP.

VII Ping from one device to another

Observation :-

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

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

Procedure :-

I Select server, set IP Address as 10.0.0.2, default gateway as 10.0.0.0, DNS server as 0.0.0.0.

II Navigate to config, select DHCP modify default gateway as 10.0.0.1, start IP address as 10.0.0.0 (switch 1)

III Name a new pool as switch two, Set startIP as 20.0.0.3, default gateway as 20.0.0.1
Maximum number of user as 100
Set gateway in setting as 10.0.0.1

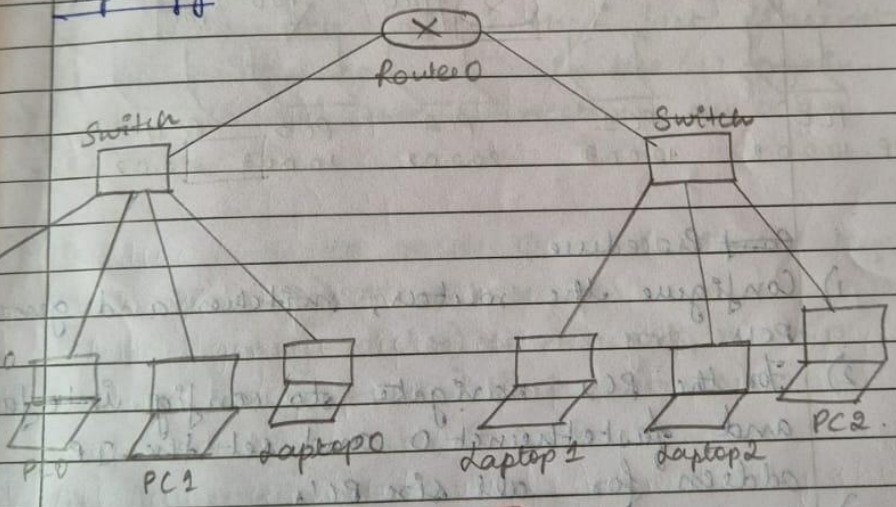
IV Select router, navigate to CLI and enter the following commands.

no

enable
config terminal
interface fastethernet 1/0
ip address 10.0.0.1 255.0.0.0
ip helper address 10.0.0.2
no shutdown
exit
Repeat the same

Observation
~~The IP address is set for all end devices.~~

Topology



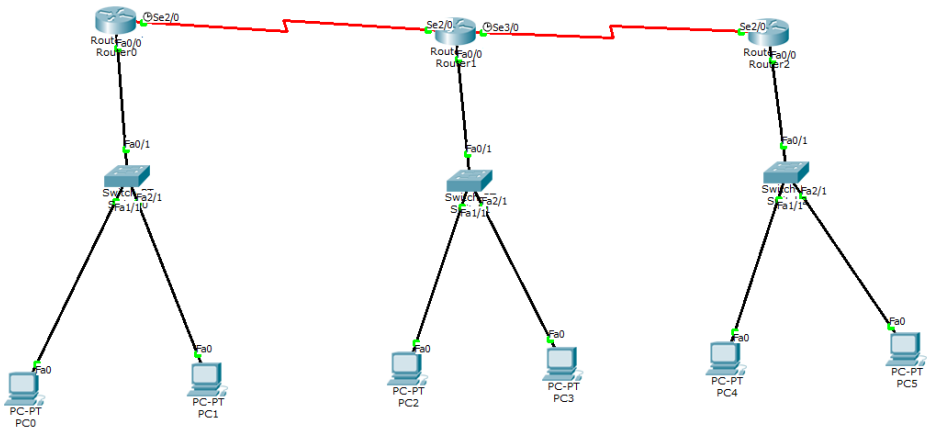
Observation
The IP address is set for all end devices.

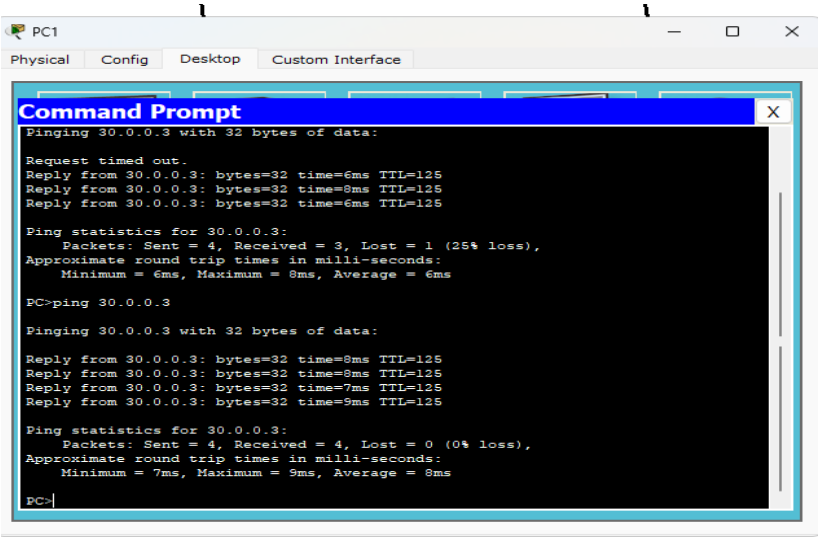
20/11/24

Program 6

Aim of the program:
Configure RIP routing Protocol in Routers

Procedure along with the topology and observation:

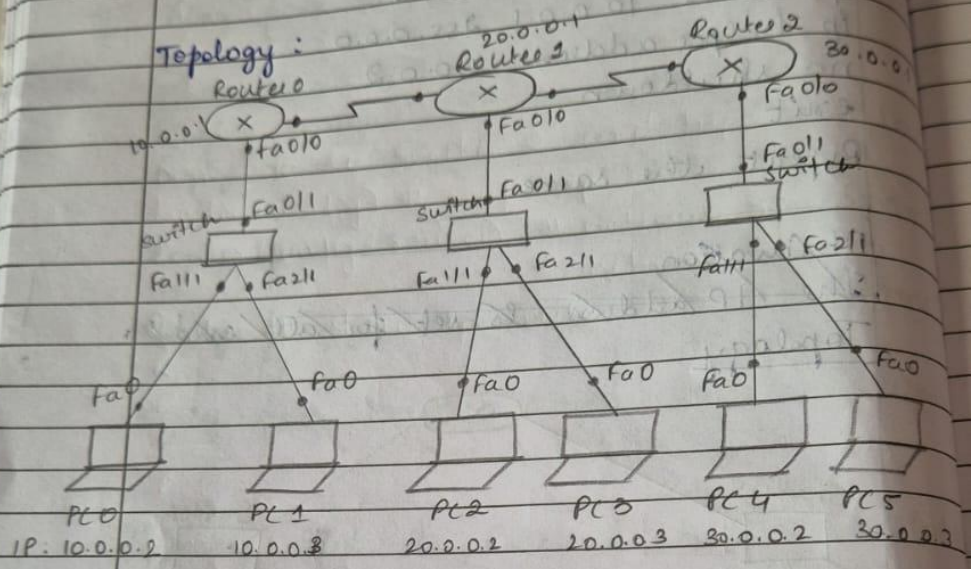




Experiment - 6.

Aim : Configure RIP routing protocol in Routers.

Topology :



~~Cont~~ Procedure :

1) Configure the routers, switches and generic PC's
2) For the PC's navigate to config interface and fastethernet 0 and set the IP address for all six PC's
3) Similarly for all six PC's, navigate to config, global, settings and set gateway as the IP address of the router it is connected to.
4) In Router 0, navigate to config, fast ethernet 0/0, and add ip address (10.0.
5) Then go to serial 3/0 and update ip address as 40.0.0.1 Go to CLI and enter command no shut.

Search

Date __/__/__
Page

6) In Router 1, Go to Interface, fast ethernet 0/0 set ip address as 20.0.0.1. Go to serial 2/0 and set IP address 40.0.0.2 In CLI enter no shut.
Go to serial 3/0 and set ip address as 50.0.0.1
In CLI enter no shut.

7) In Router 2,
Go to interface, fast ethernet 0/0, set IP address as 30.0.0.1, Go to serial 2/0 and set IP address as 50.0.0.2.

8) If connections are not active after this, in all routers check the fast ethernet and enter no shut in the CLI again.

9) Now all the ~~are~~ connections are active (green).

10) So now go to router 0 and in CLI type
Router (config) # router rip
Router (config-router) # network 10.0.0.0
Router (config-router) # network 40.0.0.0
Router (config-router) # exit
Router (config) # exit
Router # show ip route
C . . .
R
C
C . . .

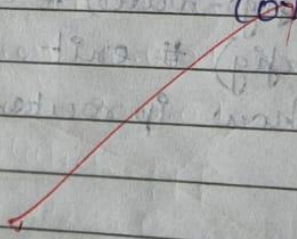
Now go to router 1. and in CLI type
Router (config) ~~-no~~ # router rip
Router (config-router) # network 40.0.0.0
network 50.0.0.0
network 20.0.0.0

Router# show ip route
R...
C...
C...
C...

Now go to router 2 and type
Router(config)# router rip
Router(config-router)# network 50.0.0.0
network 30.0.0.0

Router# show ip route
R...
R...
C...
R...
C...

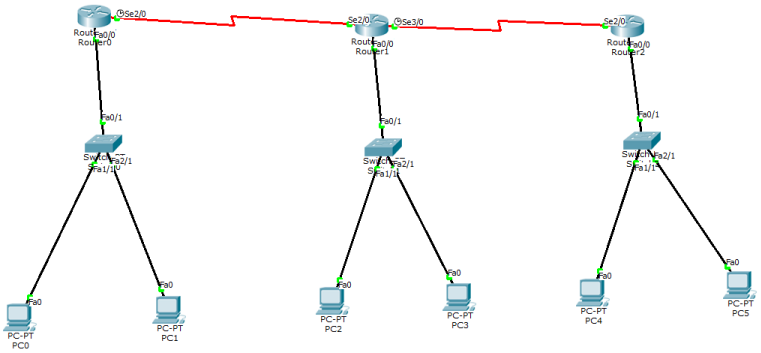
Observation:
From PC 1 we can ping to PC 5
PC> ping 30.0.0.3
Packets: Sent = 4, Received = 4, Lost = 0
(0% Loss).

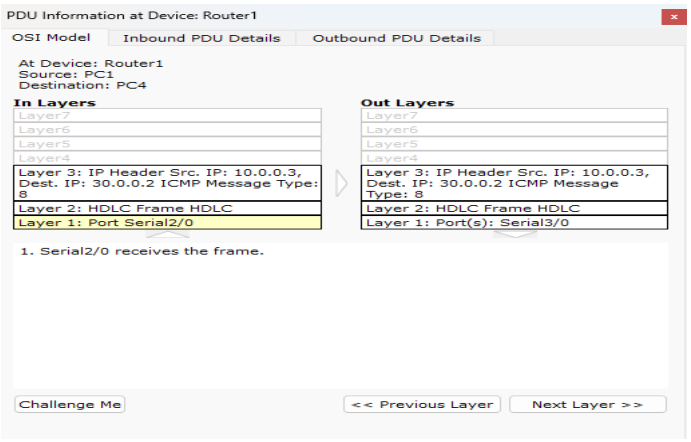


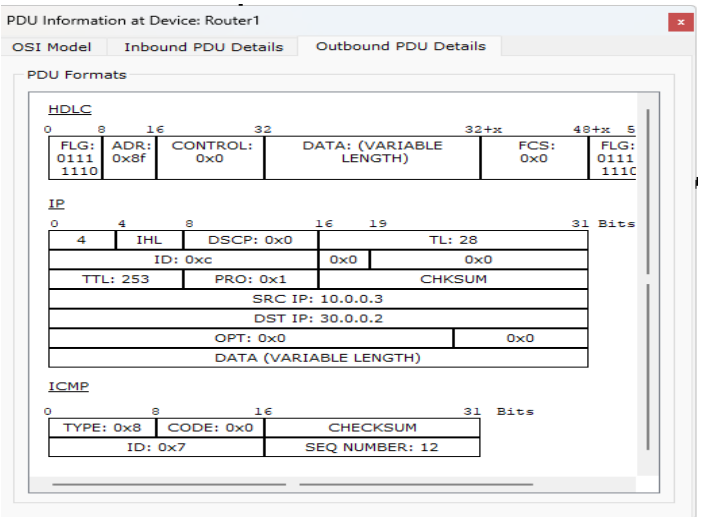
Program 7

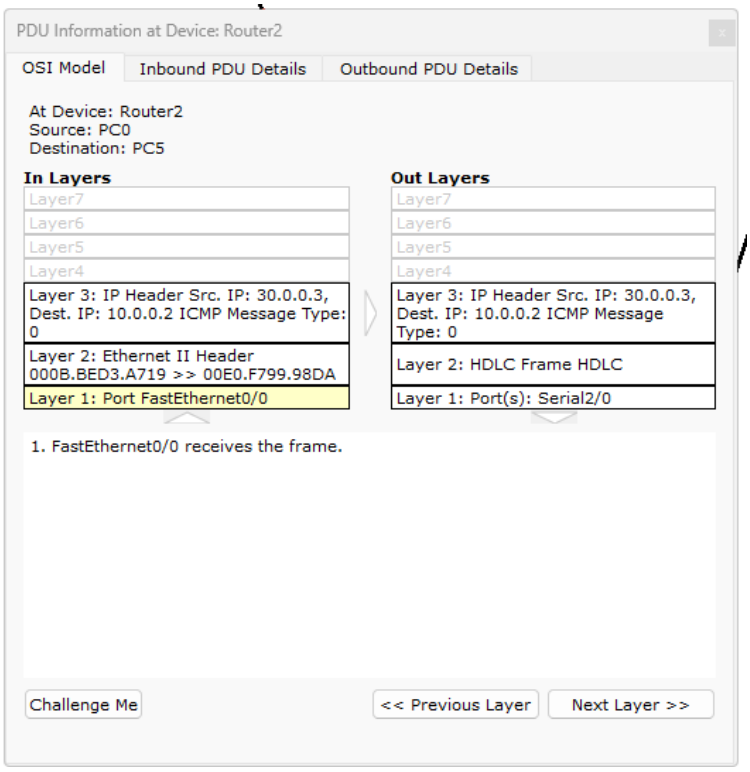
Aim of the program:
Demonstrate the TTL/ Life of a Packet:

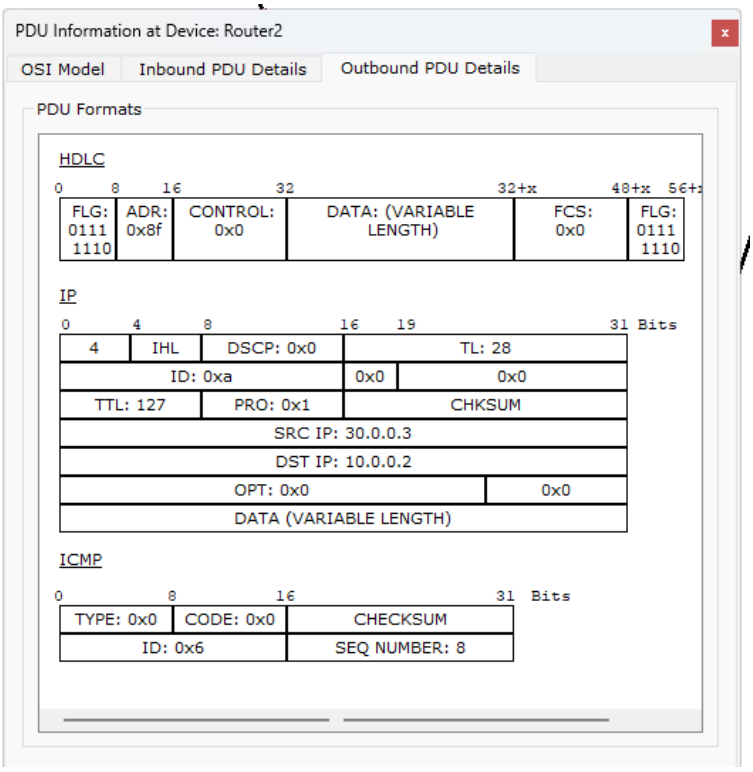
Procedure along with the topology and observation:

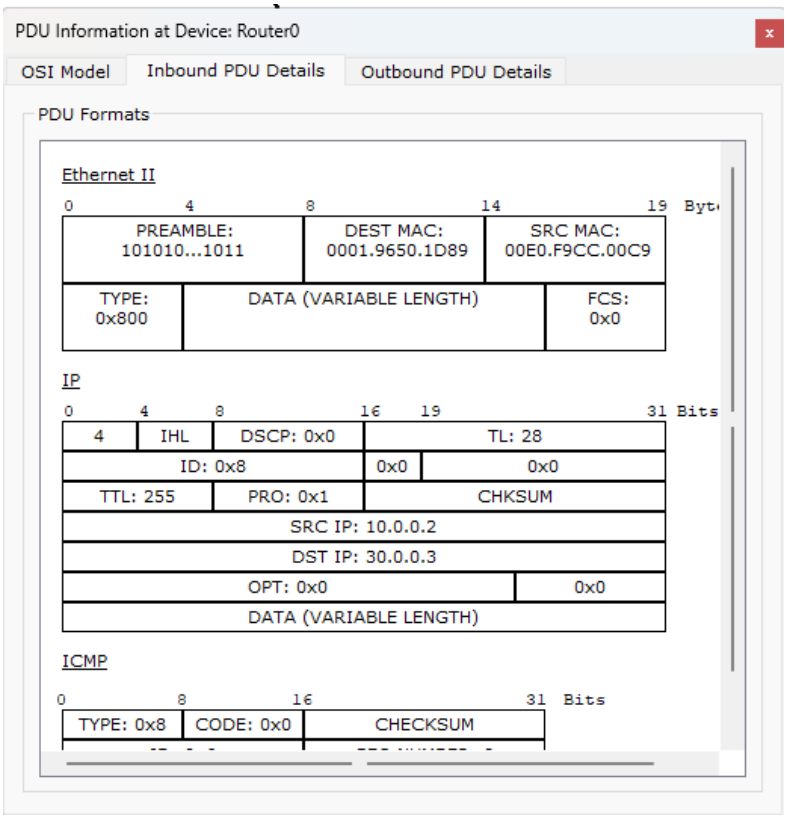


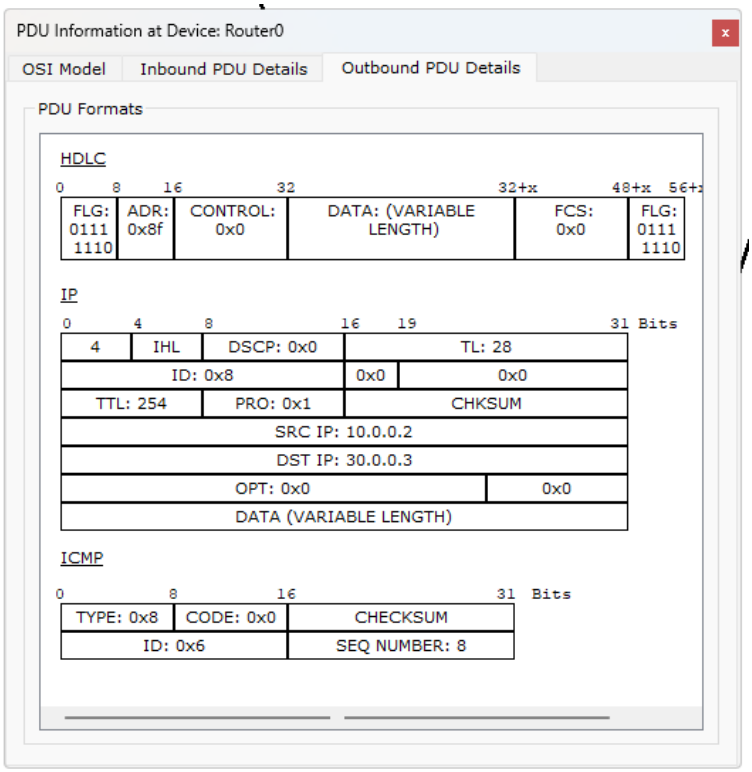





PDU Information at Device: Router2
OSI Model
Inbound PDU Details
Outbound PDU Details
At Device: Router2
Source: PC0
Destination: PC5
In Layers
Out Layers
Layer7
Layer6
Layer5
Layer4
Layer 3: IP Header Src. IP: 30.0.0.3, Dest. IP: 10.0.0.2 ICMP Message Type: 0
Layer 2: Ethernet II Header 000B.BED3.A719 >> 00E0.F799.98DA
Layer 1: Port FastEthernet0/0
Layer 3: IP Header Src. IP: 30.0.0.3, Dest. IP: 10.0.0.2 ICMP Message Type: 0
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: Router2
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: 127
PRO: 0x1
CHKSUM
SRC IP: 30.0.0.3
DST IP: 10.0.0.2
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
0 8 16 31 Bits
TYPE: 0x0
CODE: 0x0
CHECKSUM
ID: 0x6
SEQ NUMBER: 8


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: 0001.9650.1D89
SRC MAC: 00E0.F9CC.00C9
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
0 4 8 16 19 31 Bits
4
IHL
DSCP: 0x0
TL: 28
ID: 0x8
0x0
0x0
TTL: 255
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.2
DST IP: 30.0.0.3
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: 0x8
0x0
0x0
TTL: 254
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.2
DST IP: 30.0.0.3
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
0 8 16 31 Bits
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x6
SEQ NUMBER: 8

Experiment 7

Date __/__/__
Page

Aim :-
Demonstrate the TTL / Life of a package

Procedure
- Add a simple PDU across the PC's of different network.
- Consider PC0 to PC5

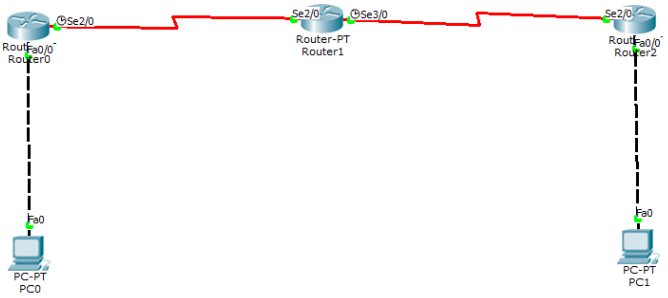
Observation
While Auto capture and observing the TTL across each PC, It was observed as follows:
PDU information at Device : PC1 TTL: 225
PDU information at Device : Router1 TTL: 254
PDU information at Device : Router2 TTL: 253
- Cisco packet tracer has the maximum TTL as 225
- It is observed that the TTL document on the message is being passed stop by stop (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 packet are received.

20/11/24

Program 8

Aim of the program:
Configure OSPF routing protocol

Procedure along with the topology and observation:




```
Pinging 40.0.0.10 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.10: bytes=32 time=8ms 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 = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 6ms, Maximum = 9ms, 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=2ms TTL=125
Reply from 40.0.0.10: bytes=32 time=5ms TTL=125
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125

Ping statistics for 40.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 7ms, Average = 5ms

PC>
```

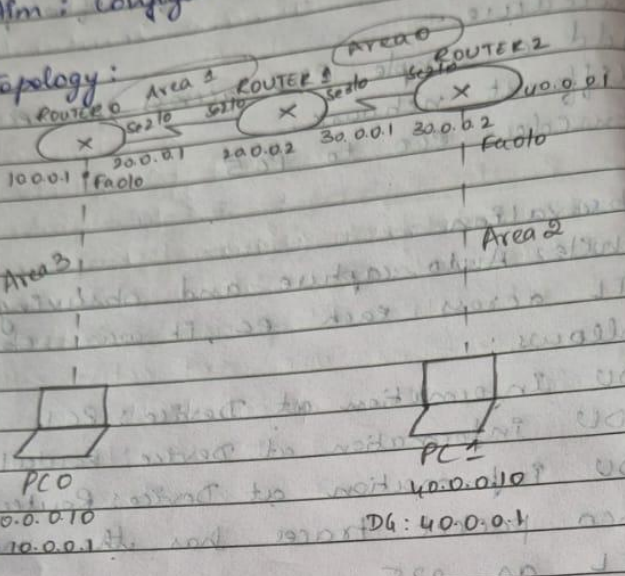
```
Command Prompt

Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.10

Pinging 10.0.0.10 with 32 bytes of data:

Reply from 10.0.0.10: bytes=32 time=7ms TTL=125
Reply from 10.0.0.10: bytes=32 time=7ms TTL=125
Reply from 10.0.0.10: bytes=32 time=5ms TTL=125
Reply from 10.0.0.10: bytes=32 time=6ms TTL=125

Ping statistics for 10.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 5ms, Maximum = 7ms, Average = 6ms

PC>
```

Experiment-8

27.11.24

Aim: Configure OSPF routing protocol

Topology:



Procedure:

Step1: Create topology like the above ~~pict~~ diagram with IP address and default gateway for Fa0/0 and se2/0 and se 3/0

Step2: Configure the IP address to all interfaces.

R0: ip address 10.0.0.1 255.0.0.0 (fa 0/0)

R0: ip address 20.0.0.1 255.0.0.0 (se 2/0)

encapsulation ppp.

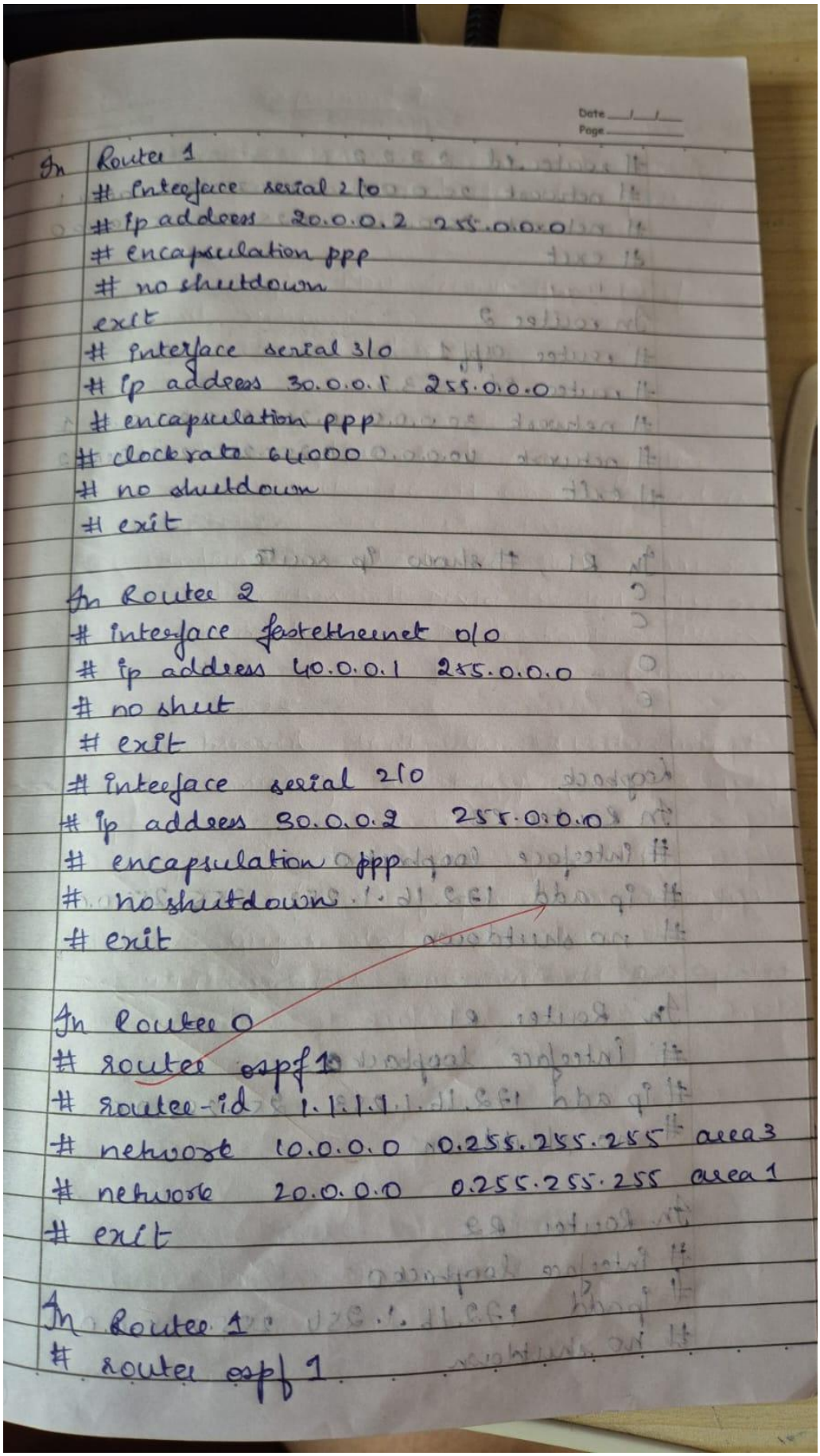
In router R1

R1: ip address 30.0.0.1/2 ~~5~~.0.0.0

clock rate 64000

no shutdown

exit

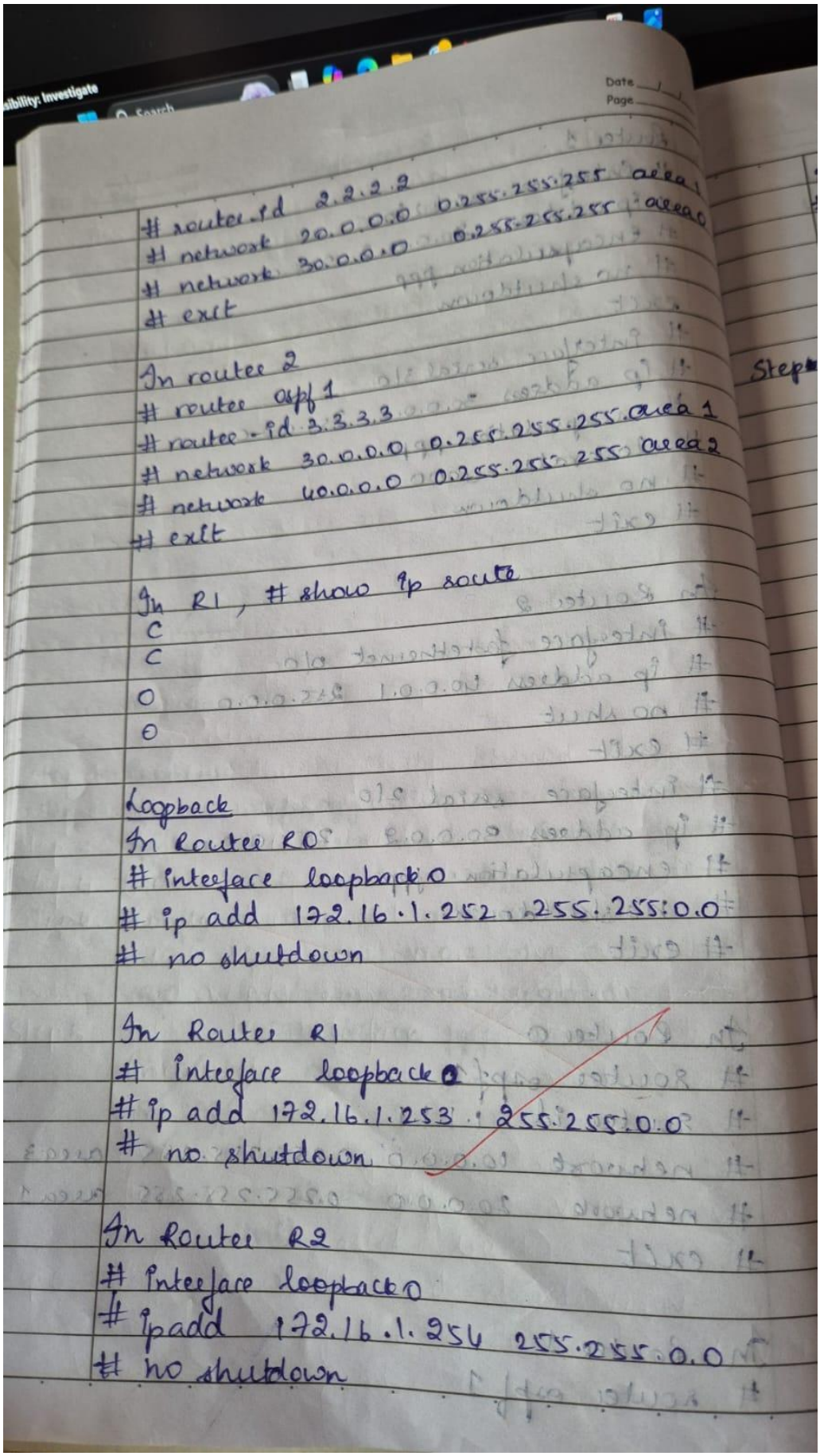


In Router 1
Interface serial 2/0
ip address 20.0.0.2 255.0.0.0
encapsulation ppp
no shutdown
exit
interface serial 3/0
ip address 30.0.0.1 255.0.0.0
encapsulation ppp
clock rate 64000
no shutdown
exit

In Router 2
interface fastethernet 0/0
ip address 40.0.0.1 255.0.0.0
no shut
exit
interface serial 2/0
ip address 30.0.0.2 255.0.0.0
encapsulation ppp
no shutdown
exit

In Router 0
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

In Router 1
router ospf 1



router id 2.2.2.2
network 20.0.0.0 0.255.255.255 area 1
network 30.0.0.0 0.255.255.255 area 0
exit

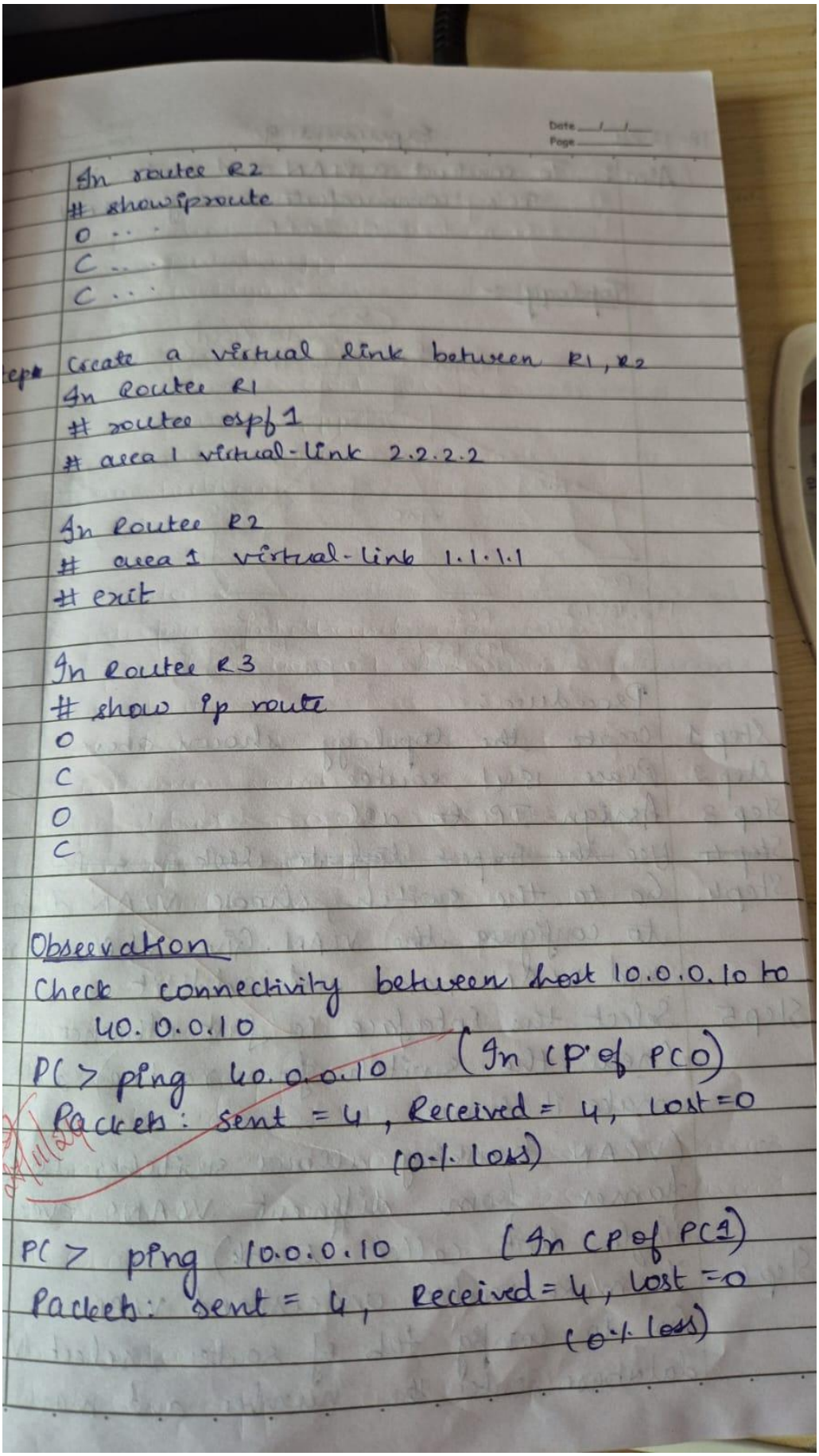
In router 2
router ospf 1
router-id 3.3.3.3
network 30.0.0.0 0.255.255.255 area 1
network 40.0.0.0 0.255.255.255 area 2
exit

In R1, # show ip route
C
C
O
O

Loopback
In Router R0
interface loopback 0
ip add 172.16.1.252 255.255.0.0
no shutdown

In Router R1
interface loopback 0
ip add 172.16.1.253 255.255.0.0
no shutdown

In Router R2
interface loopback 0
ipadd 172.16.1.254 255.255.0.0
no shutdown



In router R2
show ip route
O ...
C ...
C ...

Step Create a virtual link between R1, R2
In Router R1
router ospf 1
area 1 virtual-link 2.2.2.2

In Router R2
area 1 virtual-link 1.1.1.1
exit

In Router R3
show ip route
O
C
O
C

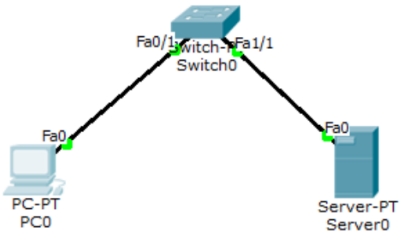
Observation
Check connectivity between host 10.0.0.10 to 40.0.0.10
PC> ping 40.0.0.10 (In CP of PC0)
Packets: Sent = 4, Received = 4, Lost = 0
(0% loss)

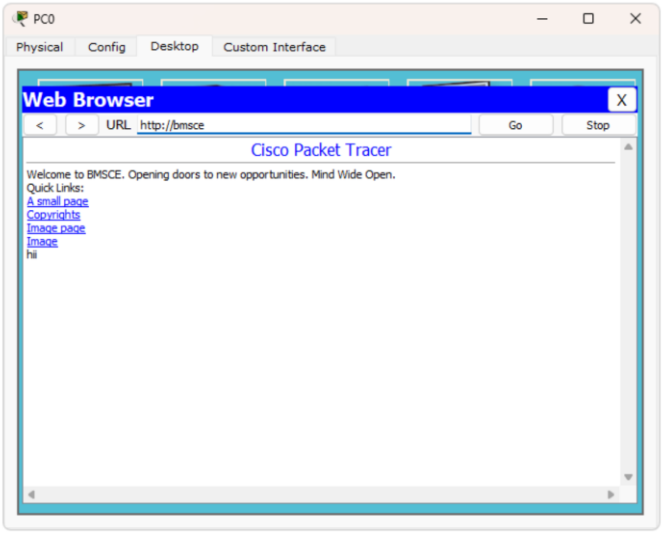
PC> ping 10.0.0.10 (In CP of PC1)
Packets: Sent = 4, Received = 4, Lost = 0
(0% loss)

Program 9

Aim of the program:
Configure Web Server, DNS within a LAN.

Procedure along with the topology and observations:





Date
Page

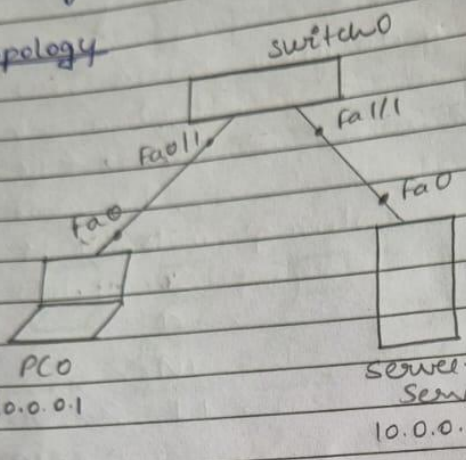
Experiment 11

18-12-24

DNS

Aim: To demonstrate WEB Server and DNS using packet tracer

Topology



Procedure

i) Create the above topology using PC, server and switch
ii) Set the IP address of PC normally
iii) To set IP of server, go to config, then select static method, set IP and make sure port status is on
iv) Ping server from PC
(Ping is successful, 0% loss)
v) Now go to PC → Desktop → webserver
vi) In the virtual browser, type in the ip address of server and click
vii) Now web page is visible

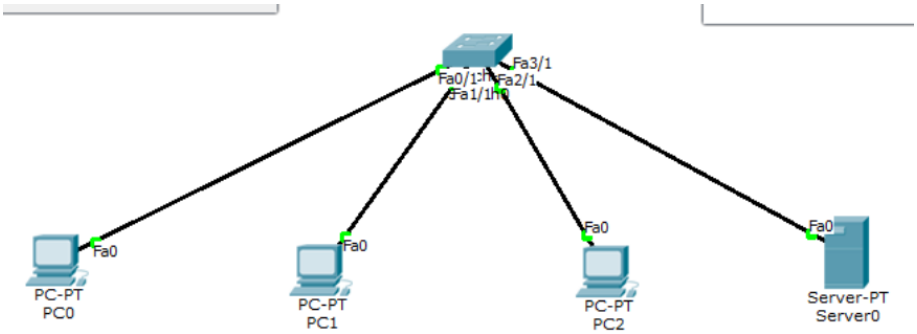
Observation:

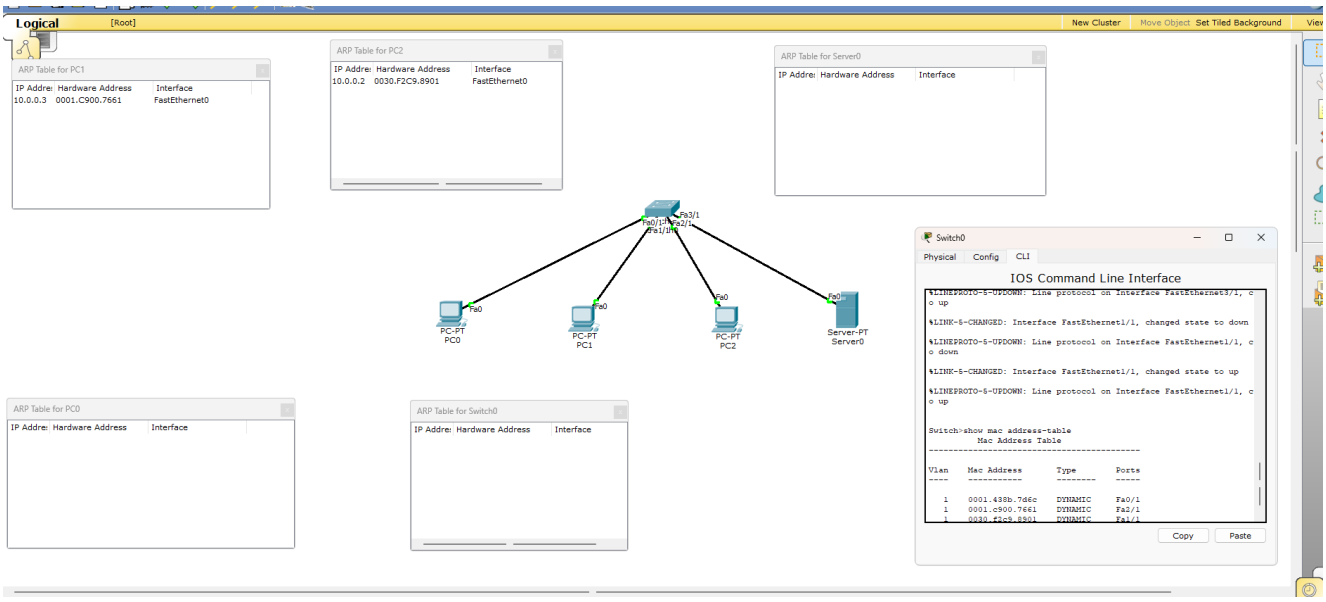
- Successfully accessed the servers web page from PC by entering IP address
- Created simple LAN with PC, switch & server.

Program 10

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

Procedure along with the topology and observations:





iii. Connect them through a switch
iv. Use the inspect tool to click on a PC to see the ARP table

English (India) Accessibility: Investigate

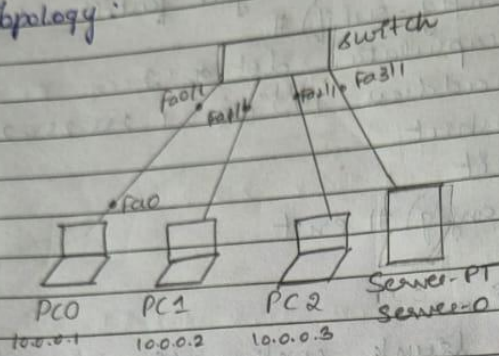
Date
Page

Experiment 10

18-12-24

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

Topology:



Procedure:

Step 1: Create a topology of 3 PC's and server and switch

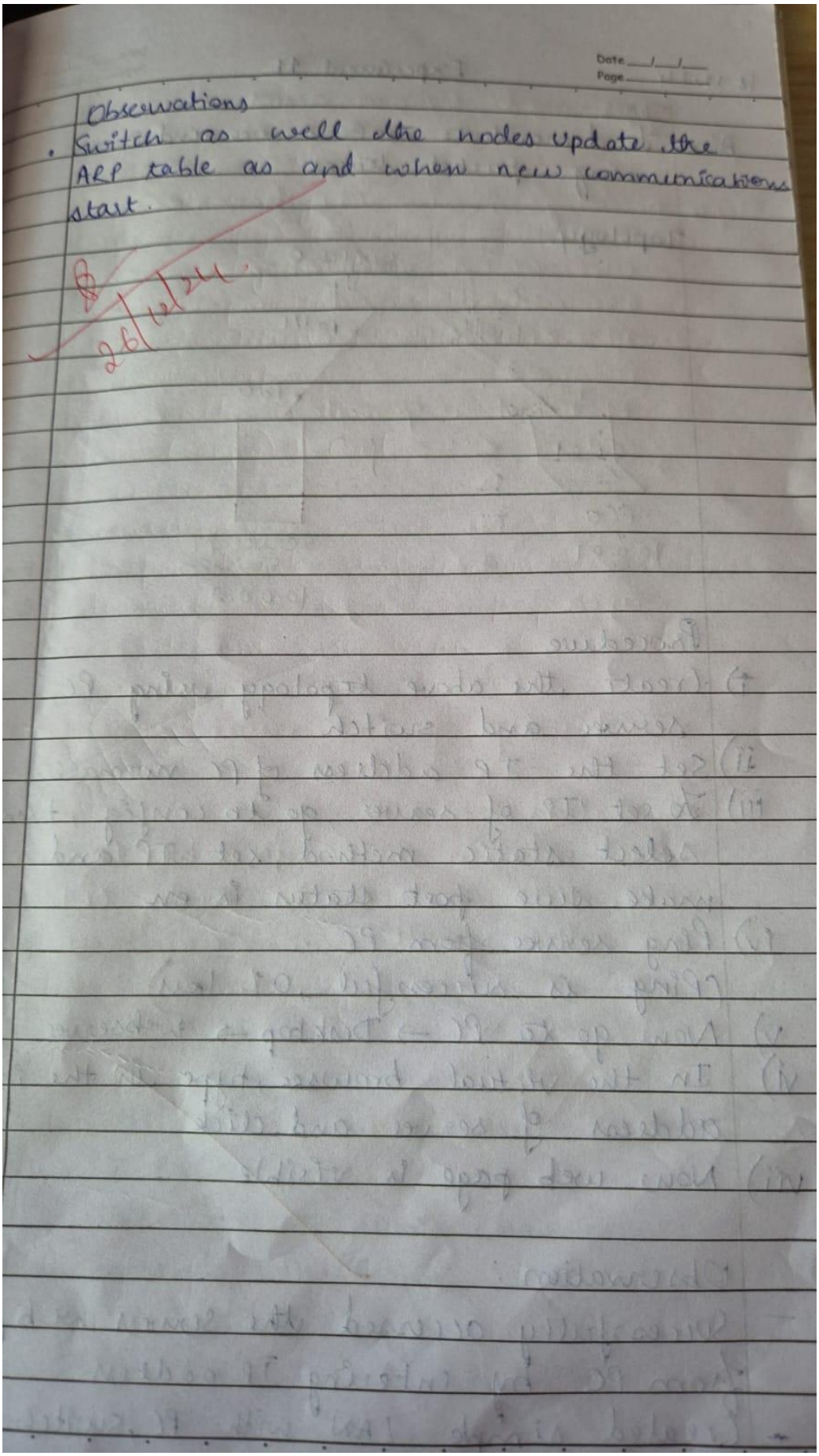
Step 2: Assign IP to all

Step 3: Connect them through a switch

Step 4: Use the Inspect tool to click on a PC to see the ARP table (command in CLI for same if arp -a)

Step 5: Initially ARP table if empty (Also in CLI of switch, the command - show mac address - table can be given on every transaction to see how the switch learn from transaction and build the address table

Step 6: Use the capture in the simulation panel to go step so that changes in ARP can be clearly no


Date __/__/__
Page

Observations

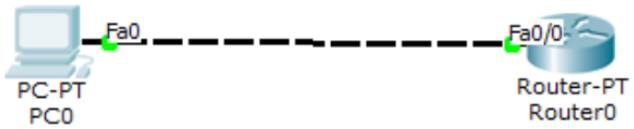
- Switch as well the nodes update the ARP table as and whenn new communications start.

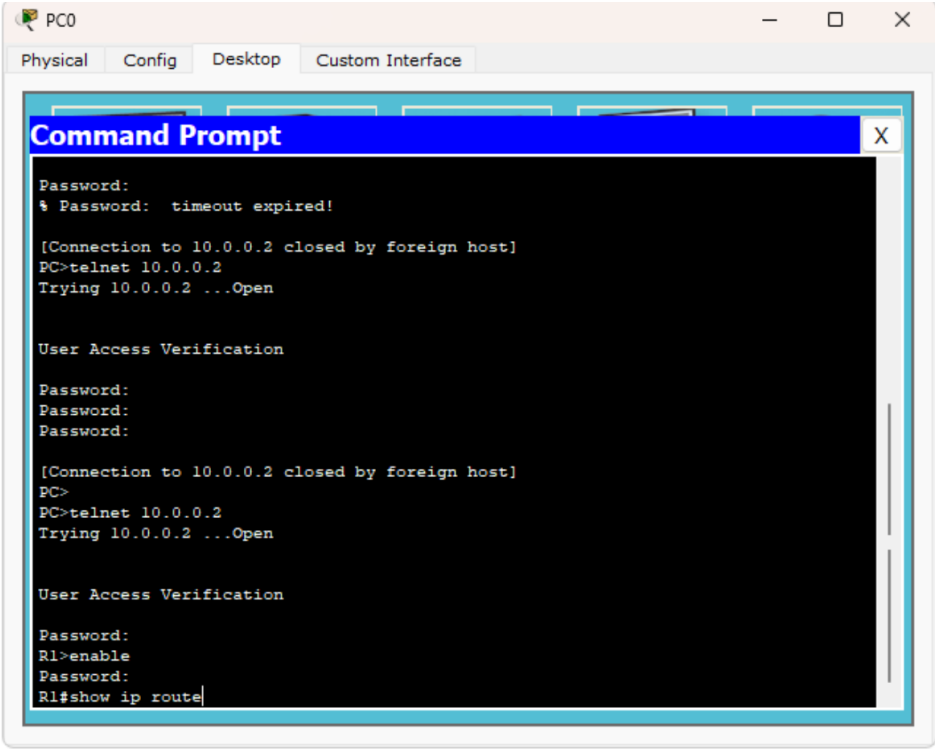
26/12/24.

Program 11

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

Procedure along with the topology:





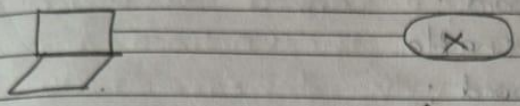
Q Search

Date __/__/__
Page

8.12.24 Experiment 12

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

Topology



PC0 — Router0

10.0.0.2 — 10.0.0.1

Procedure

Step 1: Make simple topology shown above

Step 2: Commands in router

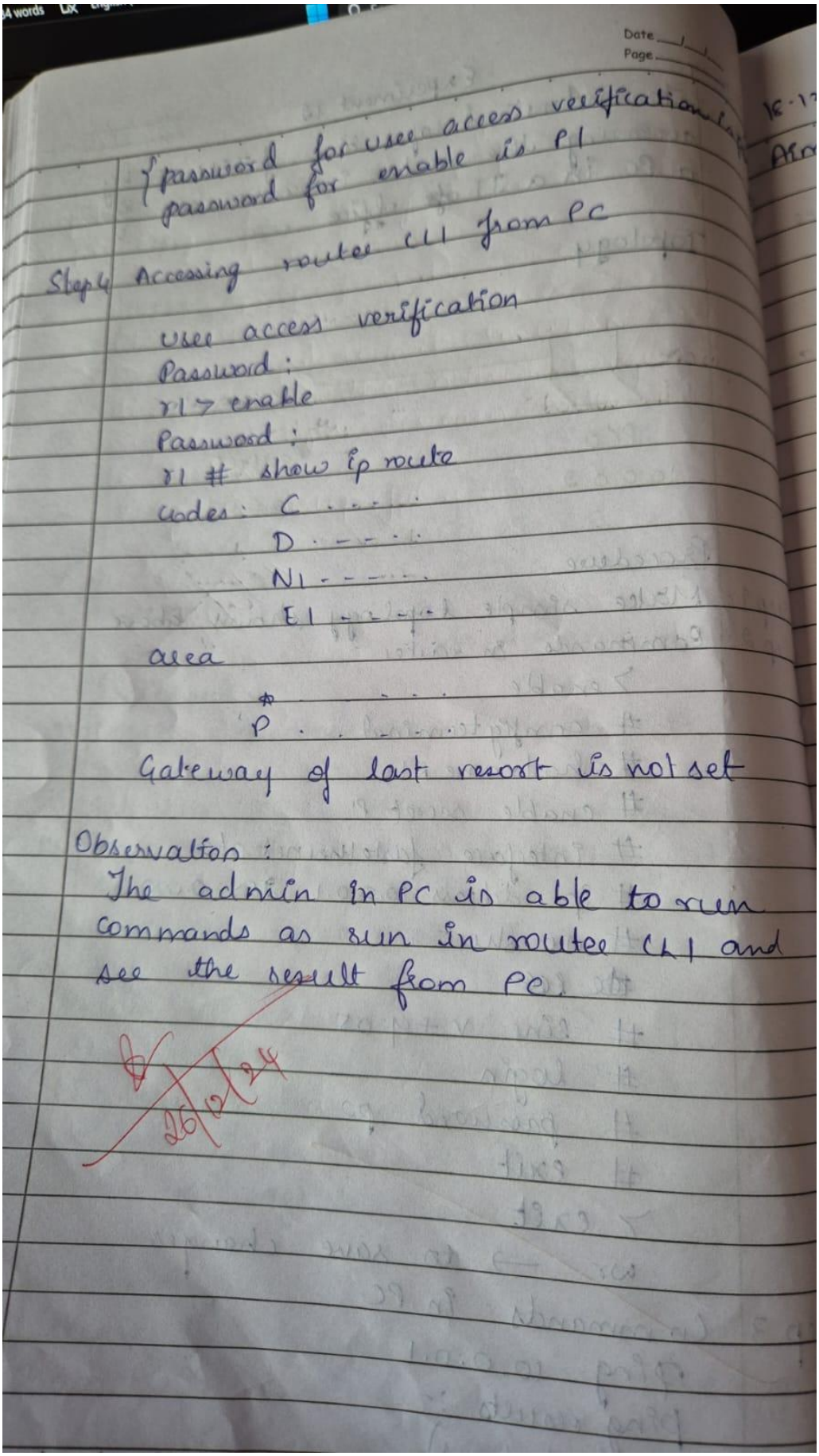
```
> enable
# configterminal
# hostname R1
# enable secret P1
# interface fastethernet 0/0
# ip address 10.0.0.1 255.0.0.0
# no shut
# line vty 05
# login
# password p0
# exit
> exit
wr → to save changes
```

3 Commands in PC

```
ping 10.0.0.1
```

ping results :-

(0% loss)



Password for user access verification is P1
password for enable is P1

Step 4 Accessing router CLI from PC

User access verification
Password:
R1> enable
Password:
R1 # show ip route
Codes: C - - - -
D - - - -
N1 - - - -
E1 - - - -
area
* - - -
P - - - -
Gateway of last resort is not set

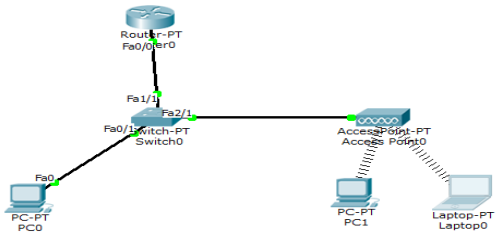
Observation:
The admin in PC is able to run commands as run in router CLI and see the result from PC.

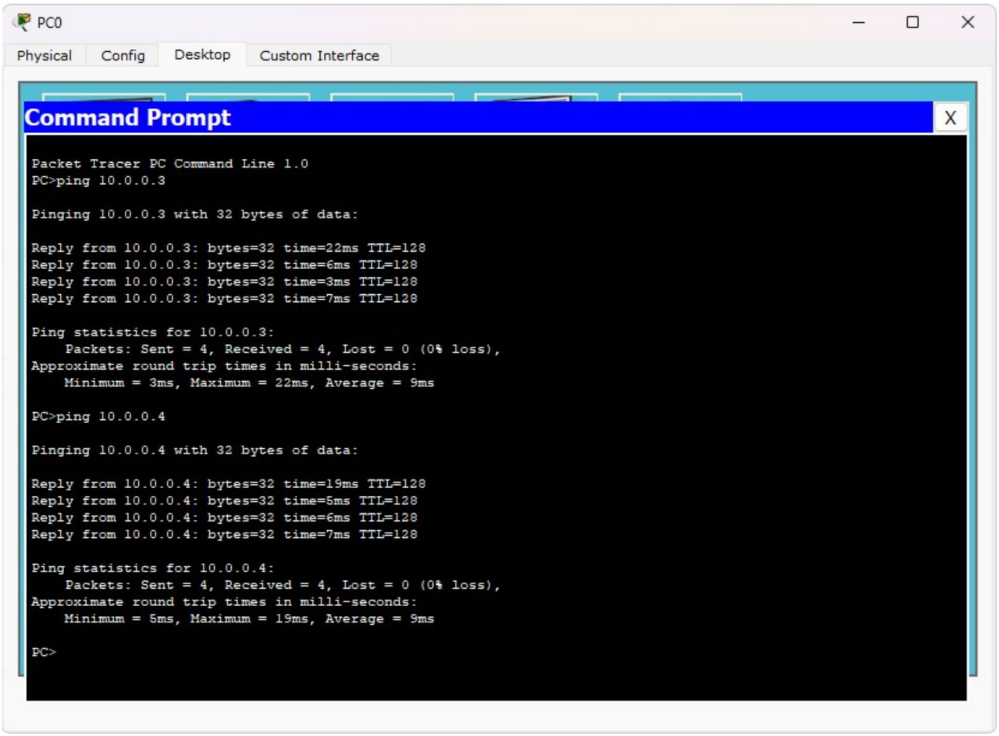
26/12/24

Program 12

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

Procedure along with the topology and observation:





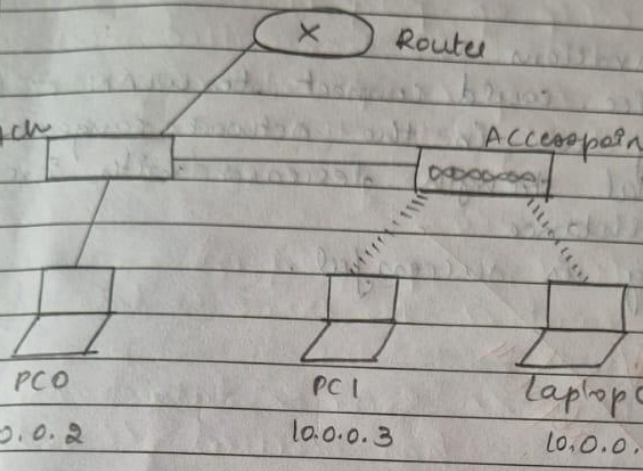
16.12.24

Date __/__/__
Page

Experiment 13

Aim: WLAN Connection and make the nodes communicate wirelessly

Topology



Procedure

Step 1 Construct the above topology
Configure PC0 and router as normally done

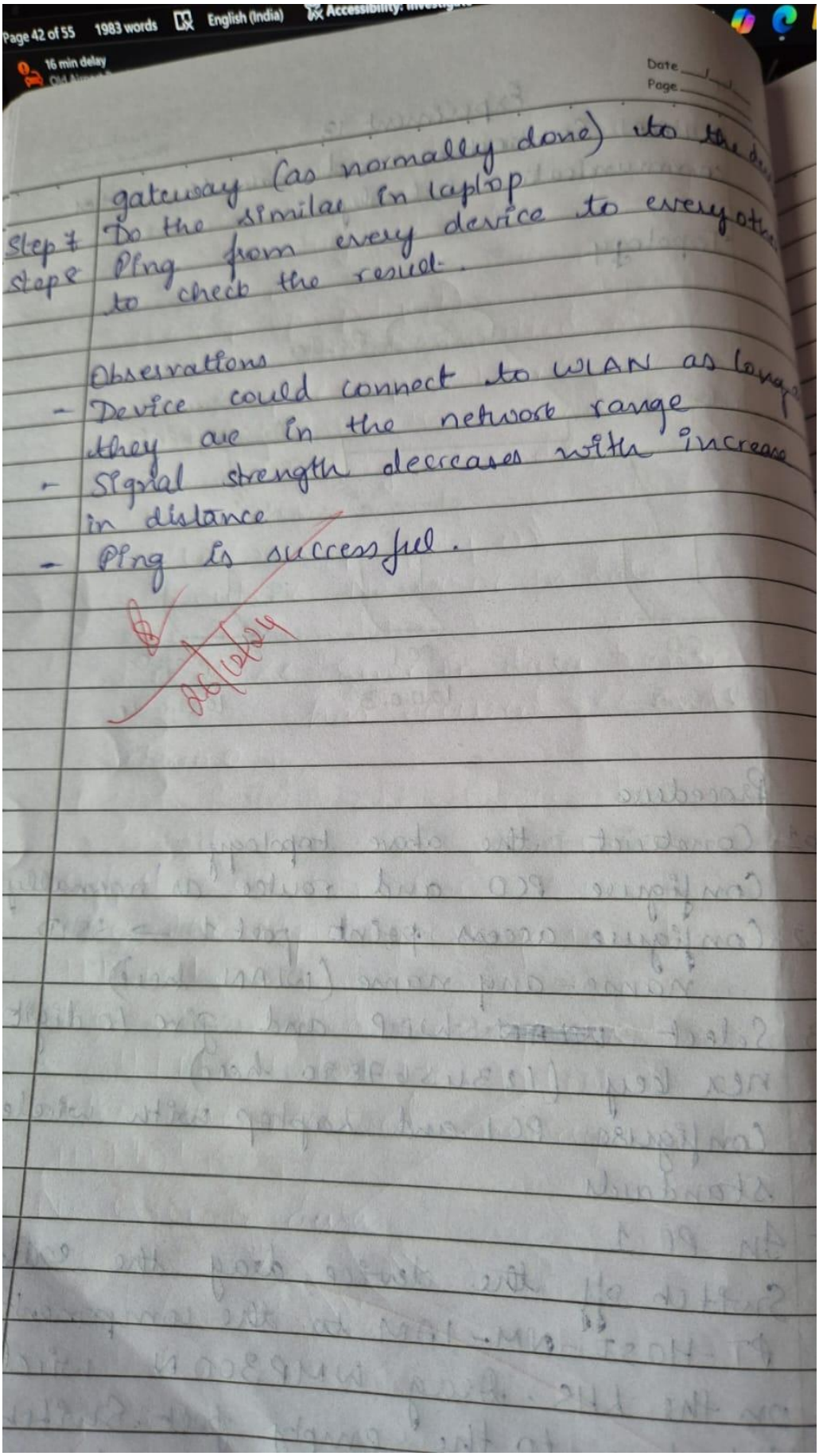
Step 2 Configure access point port 1 → SSID name - any name (WLAN here)

Step 3 Select ~~WLAN~~ WEP and give 10 digit hex key (1234567890 here)

Step 4 Configure PC1 and Laptop with wireless standards

Step 5 In PC 1
Switch off the device, drag the existing PT-HOST-NM-1AM to the component listed on the LHS. Drag WMP300N wireless interface to the empty port. Switch on PC

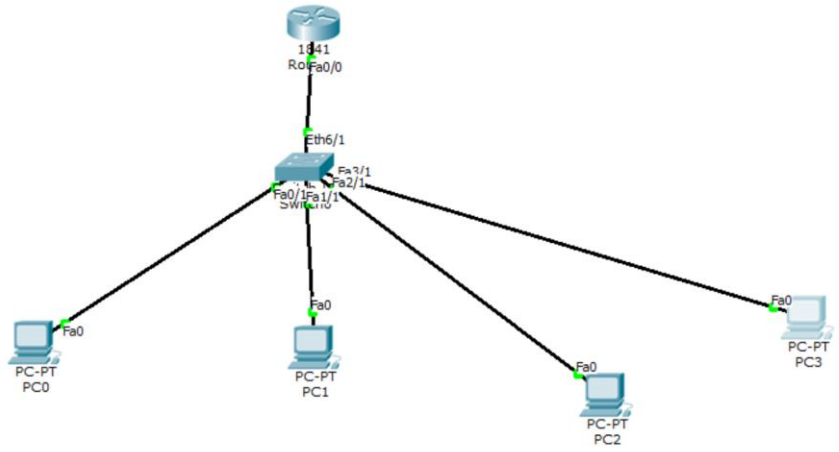
Step 6 In the config tab a new wireless interface would have been added. Now configure SSID, WEP, WEP key, IP address and

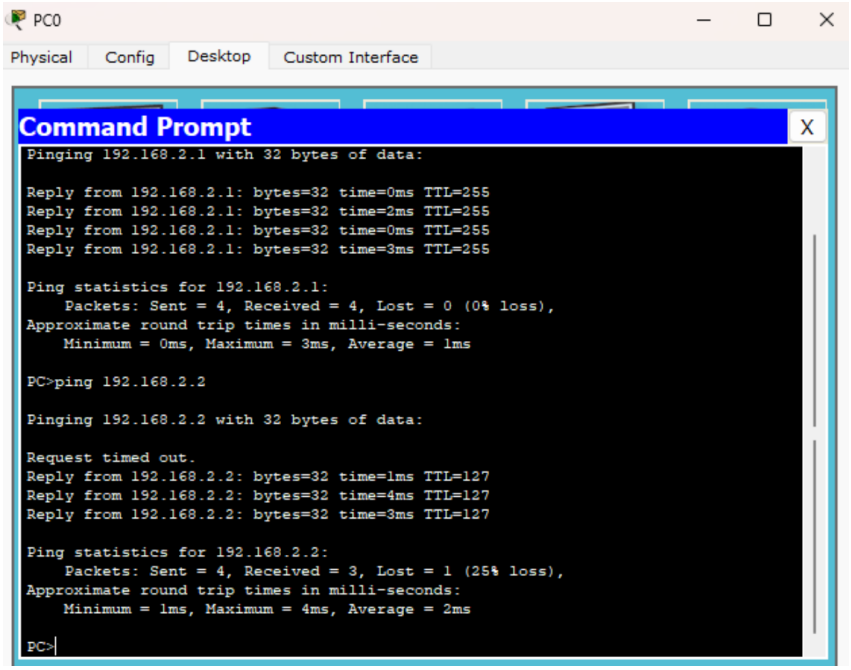

Page 42 of 55 1983 words English (India)
16 min delay
Date
Page
gateway (as normally done) to the
Step 7 Do the similar in laptop
steps Ping from every device to every othe
to check the result.
Observations
- Device could connect to WLAN as long
they are in the network range
- Signal strength decreases with increase
in distance
- Ping is successful.
8
26/12/24

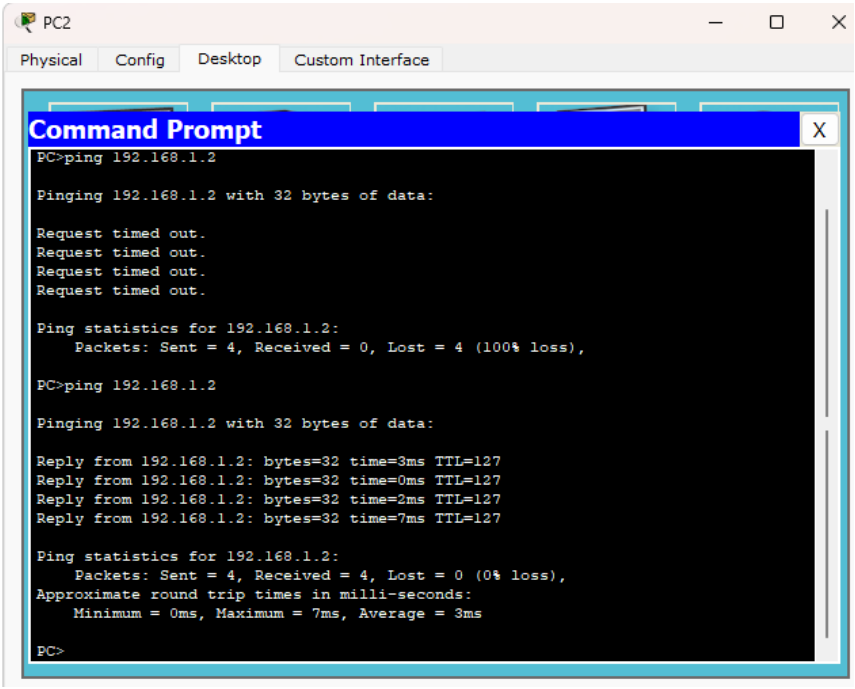
Program 13

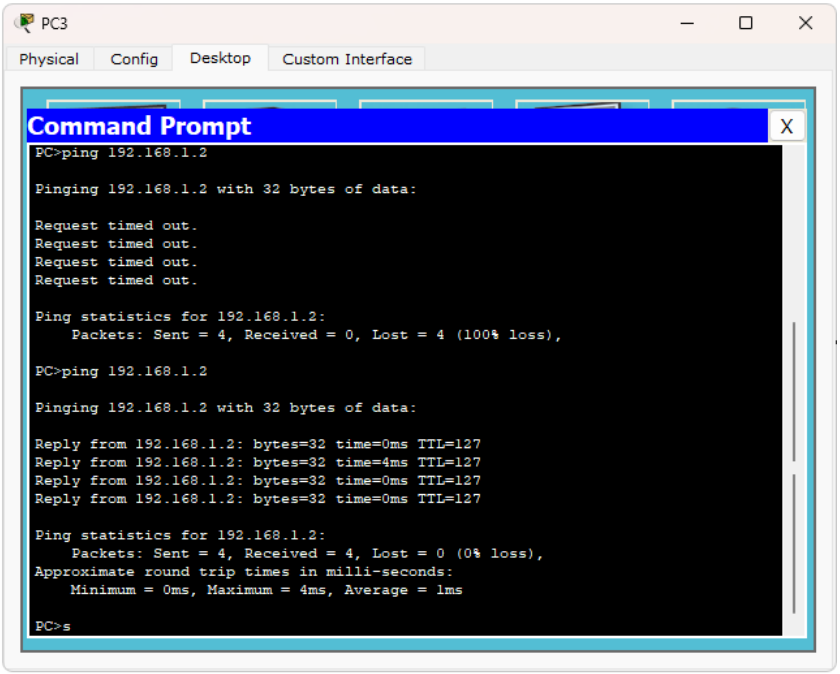
Aim of the program:
To construct a VLAN and make the PC's communicate among a VLAN

Procedure along with the topology and observations:






PC2
Physical
Config
Desktop
Custom Interface
Command Prompt
X
PC>ping 192.168.1.2
Pinging 192.168.1.2 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 192.168.1.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 192.168.1.2
Pinging 192.168.1.2 with 32 bytes of data:
Reply from 192.168.1.2: bytes=32 time=3ms TTL=127
Reply from 192.168.1.2: bytes=32 time=0ms TTL=127
Reply from 192.168.1.2: bytes=32 time=2ms TTL=127
Reply from 192.168.1.2: bytes=32 time=7ms TTL=127
Ping statistics for 192.168.1.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 7ms, Average = 3ms
PC>


PC3
Physical
Config
Desktop
Custom Interface
Command Prompt
X
PC>ping 192.168.1.2
Pinging 192.168.1.2 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 192.168.1.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 192.168.1.2
Pinging 192.168.1.2 with 32 bytes of data:
Reply from 192.168.1.2: bytes=32 time=0ms TTL=127
Reply from 192.168.1.2: bytes=32 time=4ms TTL=127
Reply from 192.168.1.2: bytes=32 time=0ms TTL=127
Reply from 192.168.1.2: bytes=32 time=0ms TTL=127
Ping statistics for 192.168.1.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 4ms, Average = 1ms
PC>s

Page 47 of 56 1838 words English (India) Accessibility: Investigate

25°C

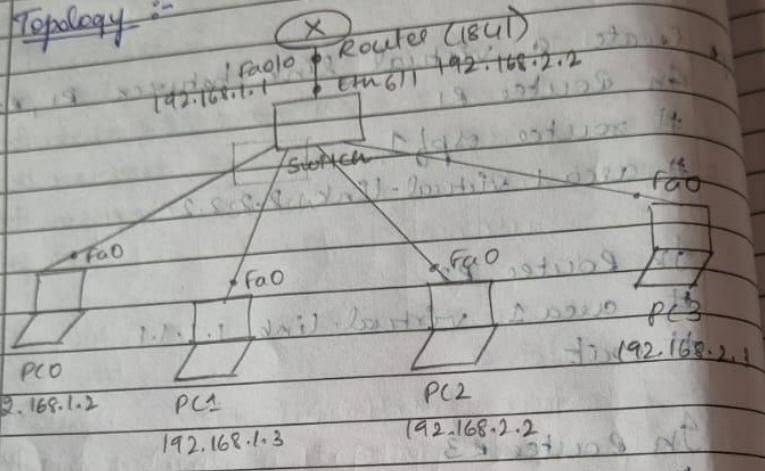
Date / /
Page

Experiment 9

18.12.24

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

Topology :-



Procedure :

Step 1 Create the topology shown above.

Step 2 Place 1841 routers

Step 3 Assign IP to all

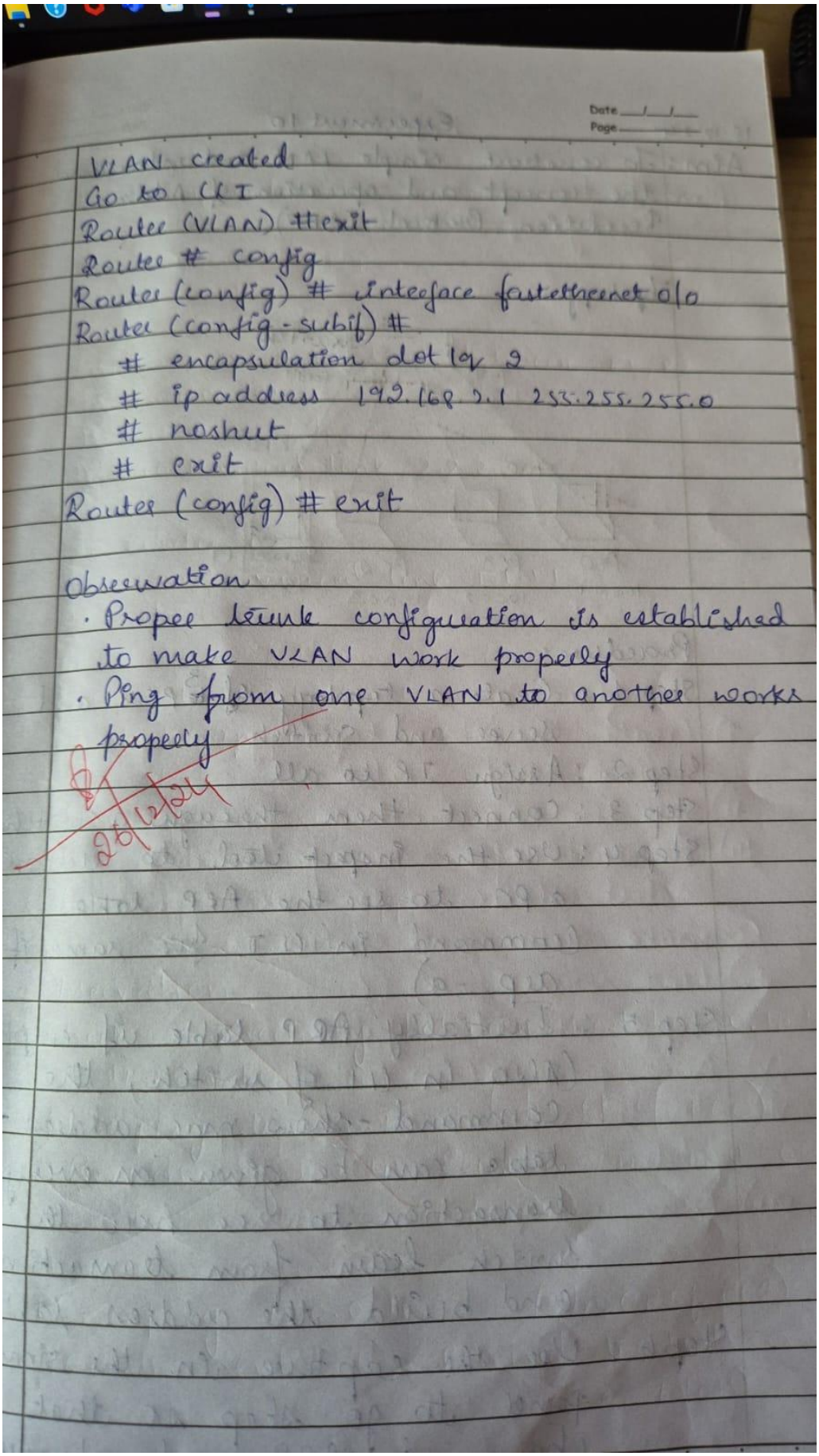
~~Step 4 Use the inspect tool to click on a PC.~~

Step 4 Go to the switch, choose VLAN database to configure the VLAN. Give VLAN number and VLAN name and add it

Step 5 Select the Interface ie fast ethernet (near the switch from router) and make it trunk

(VLAN trunking allows switches to forward frames from different VLANS over a single link called trunk)

Step 6 To make routers understand VLAN Go to config tab of router, select VLAN database, enter the number and name of



VLAN created
Go to CLI
Router (VLAN) #exit
Router # config
Router (config) # interface fastethernet 0/0
Router (config-subif) #
encapsulation dot1q 2
ip address 192.168.2.1 255.255.255.0
noshut
exit
Router (config) # exit

Observation
- Proper trunk configuration is established to make VLAN work properly
- Ping from one VLAN to another works properly

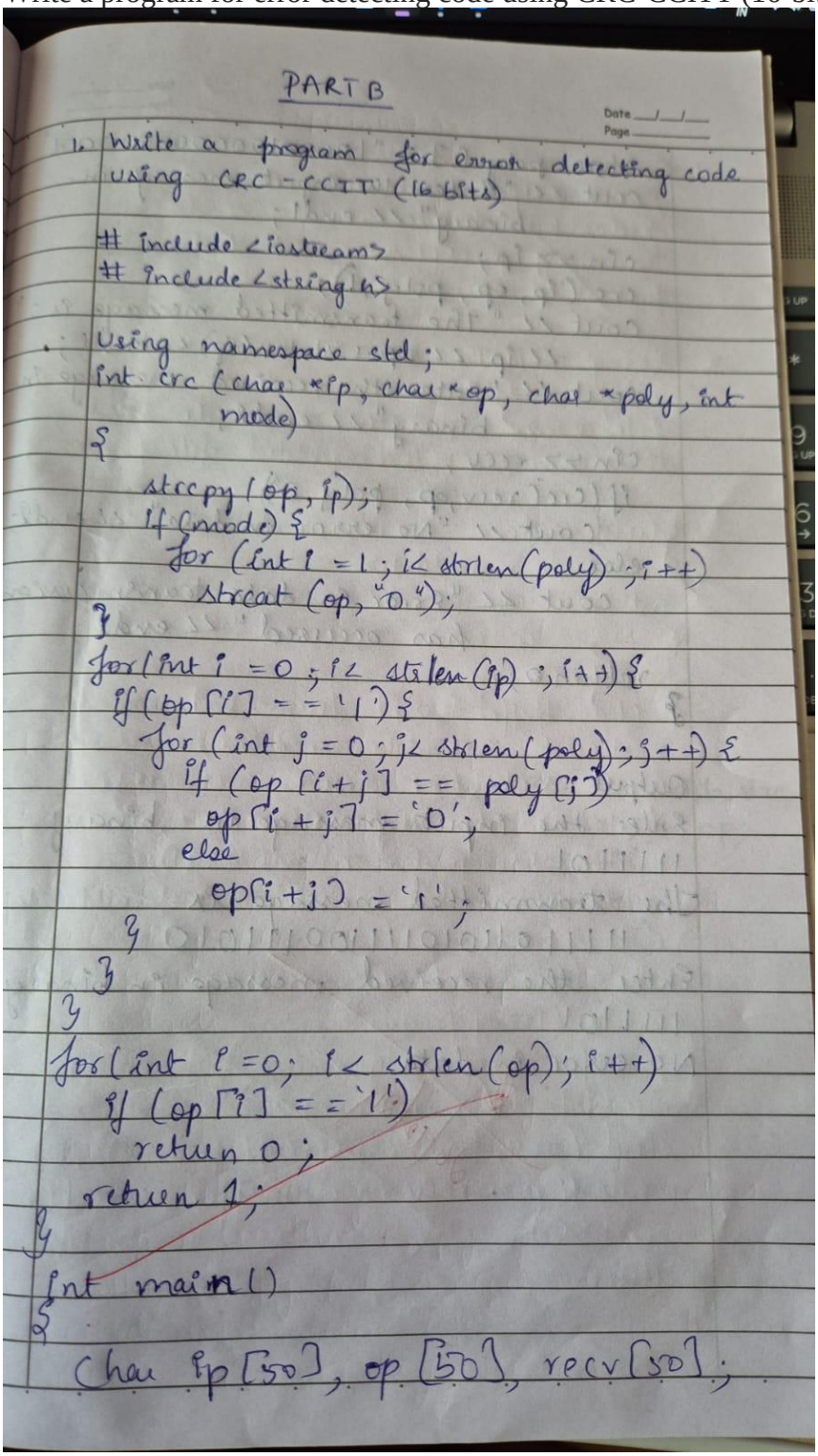
26/12/24

PART B PROGRAMS

Program 14

Aim of the program

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



PART B

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

```
#include<iostream>
#include<string.h>
using namespace std;
int crc(char *ip, char *op, char *poly, int mode)
{
  strcpy(op, ip);
  if(mode){
    for(int i=1; i<strlen(poly); i++)
      strcat(op, "0");
  }
  for(int i=0; i<strlen(ip); i++){
    if(op[i]=='1'){
      for(int j=0; j<strlen(poly); j++){
        if(op[i+j]==poly[j])
          op[i+j]='0';
        else
          op[i+j]='1';
      }
    }
  }
  for(int i=0; i<strlen(op); i++)
    if(op[i]=='1')
      return 0;
  return 1;
}
int main()
{
  char ip[50], op[50], recv[50];
```

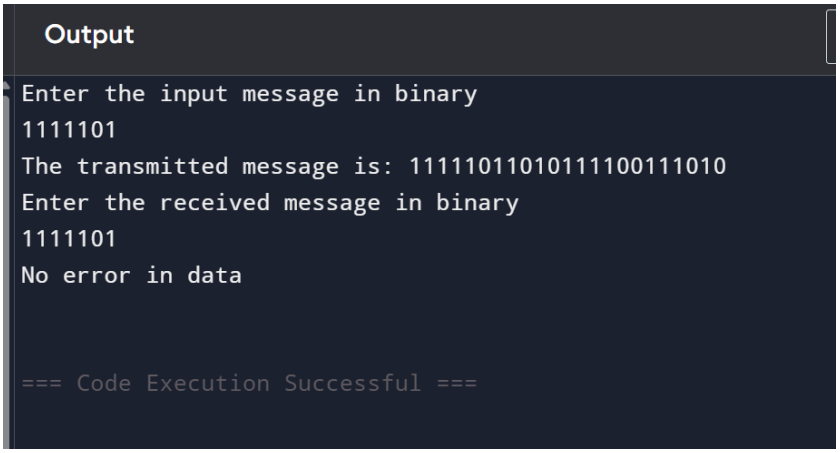
```
char poly[] = "10001000000100001";
cout << "Enter the input message in
        binary" << endl;
cin >> ip;
crc(ip, op, poly, 1);
cout << "The transmitted message is:
     << ip << op+strlen(ip) << endl;
cout << "Enter the received message in
        binary" << endl;
cin >> recv;
if(crc(recv, op, poly, 0))
  cout << "No error in data" << endl;
else
  cout << "Error in data transmission
          has occured" << endl;
return 0;
}
```

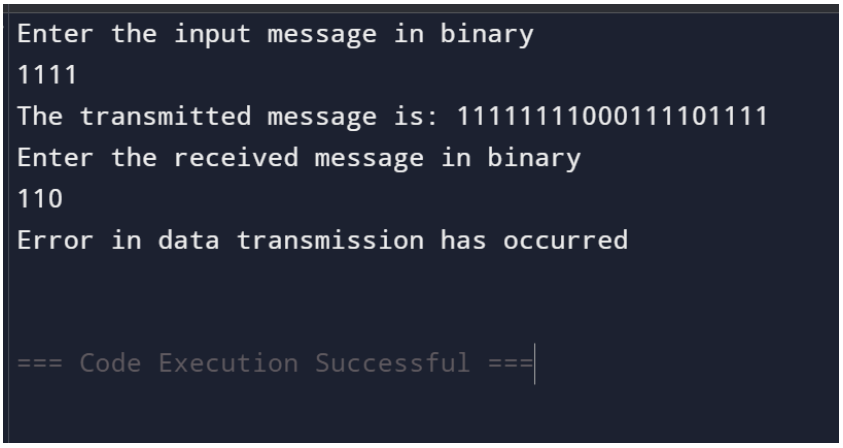
Output:

Enter the input message in binary
11111101
The transmitted message is
1111101101011110011010
Enter the received message in binary
111101
No error in data

26/12/24

Output

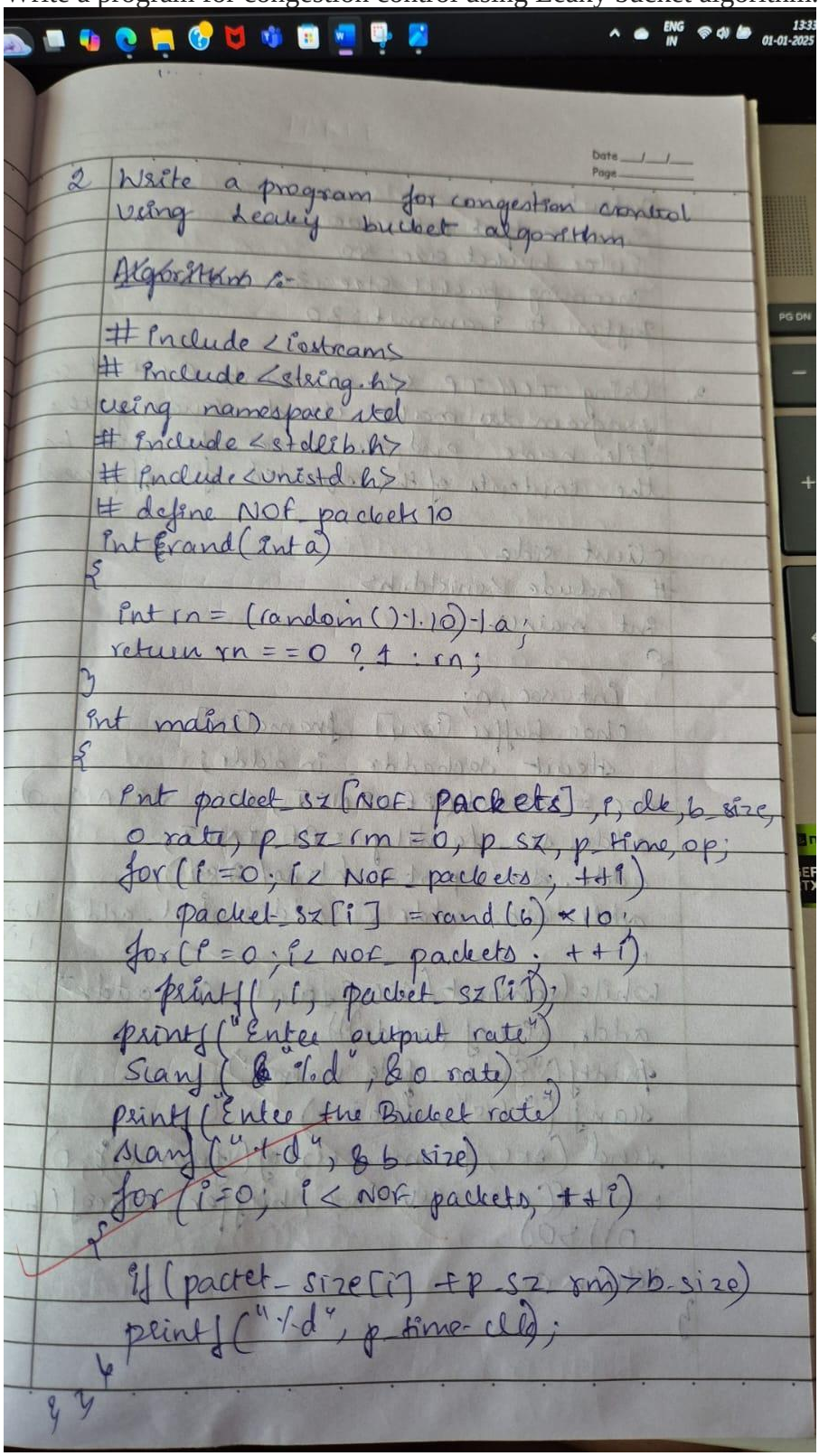

Output
Enter the input message in binary
1111101
The transmitted message is: 11111011010111100111010
Enter the received message in binary
1111101
No error in data
=== Code Execution Successful ===


Enter the input message in binary
1111
The transmitted message is: 11111111000111101111
Enter the received message in binary
110
Error in data transmission has occurred
=== Code Execution Successful ===

Program 15

Aim of the program

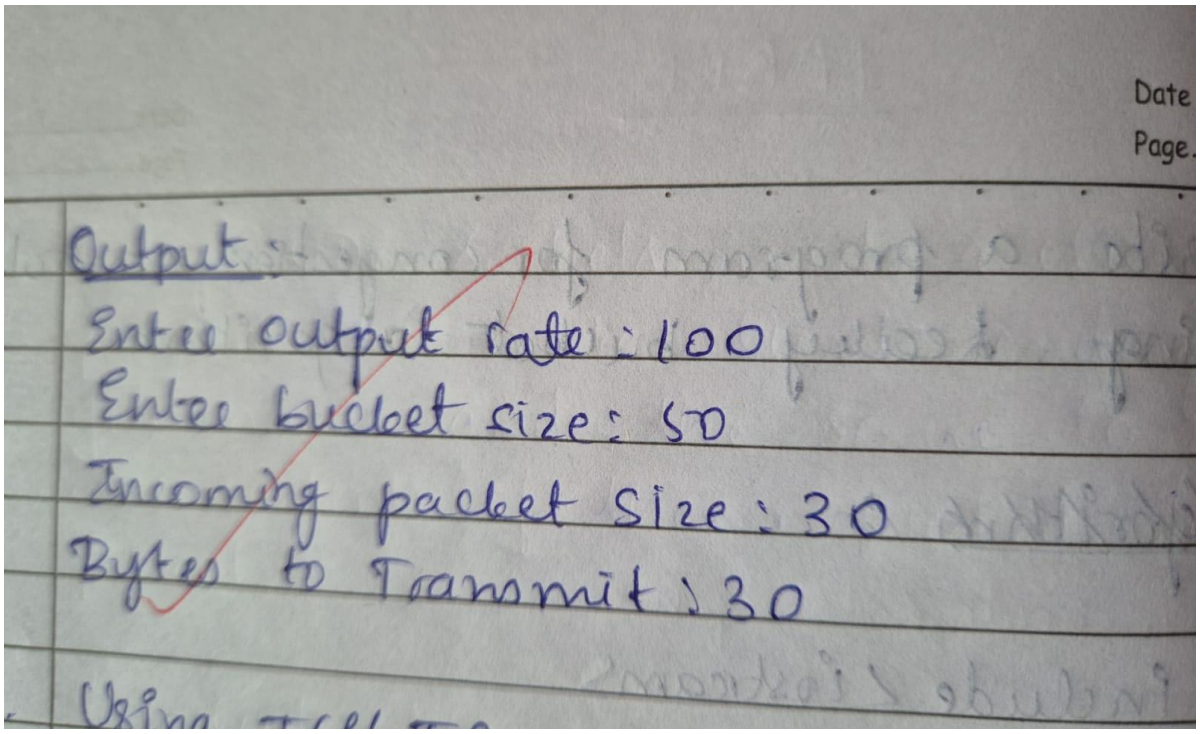
Write a program for congestion control using Leaky bucket algorithm.



2 Write a program for congestion control using Leaky bucket algorithm

Algorithm :-

```
#include <iostream>
#include <stdlib.h>
using namespace std;
#include <stdlib.h>
#include <unistd.h>
#define NOF_packets 10
int rand(int a)
{
  int rn = (random()%10)%a;
  return rn == 0 ? 1 : rn;
}
int main()
{
  int packet_sz[NOF_packets], i, clk, b_size,
  o_rate, p_sz_rm = 0, p_sz, p_time, op;
  for(i=0; i< NOF_packets; ++i)
    packet_sz[i] = rand(6) * 10;
  for(i=0; i< NOF_packets; ++i)
    printf("%d", packet_sz[i]);
  printf("Enter output rate");
  scanf("%d", &o_rate);
  printf("Enter the Bucket rate");
  scanf("%d", &b_size);
  for(i=0; i< NOF_packets; ++i)
  {
    if(packet_size[i] + p_sz_rm > b_size)
      printf("%d", p_time-clk);
  }
}
```



Output

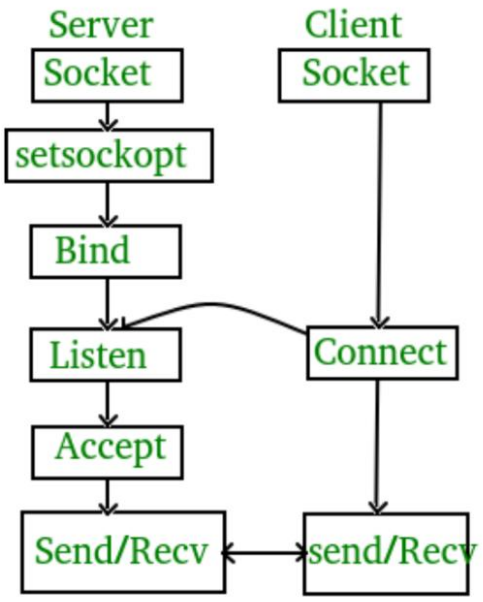
```
Packet[0]: 10 bytes
Packet[1]: 40 bytes
Packet[2]: 30 bytes
Packet[3]: 10 bytes
Packet[4]: 50 bytes
Packet[5]: 10 bytes
Packet[6]: 40 bytes
Packet[7]: 10 bytes
Packet[8]: 30 bytes
Packet[9]: 10 bytes
Enter the Output rate: 20
Enter the Bucket Size: 50


Incoming Packet size: 10
Bytes remaining to transmit: 10
Time allocated for transmission: 20 units
Packet of size 10 transmitted. ---- Bytes remaining to transmit: 0
Time left for transmission: 0 units
No packets to transmit!

Incoming Packet size: 40
Bytes remaining to transmit: 40
Time allocated for transmission: 30 units
Packet of size 20 transmitted. ---- Bytes remaining to transmit: 20
Packet of size 20 transmitted. ---- Bytes remaining to transmit: 0
Time left for transmission: 0 units
No packets to transmit!

Incoming Packet size: 30
Bytes remaining to transmit: 30
Time allocated for transmission: 20 units
Packet of size 20 transmitted. ---- Bytes remaining to transmit: 10
Packet of size 10 transmitted. ---- Bytes remaining to transmit: 0
```

```
Incoming Packet size: 10
Bytes remaining to transmit: 10
Time allocated for transmission: 20 units
Packet of size 10 transmitted. ---- Bytes remaining to transmit: 0
Time left for transmission: 0 units
No packets to transmit!

Incoming Packet size: 40
Bytes remaining to transmit: 40
Time allocated for transmission: 10 units
Packet of size 20 transmitted. ---- Bytes remaining to transmit: 20

Incoming Packet size: 10
Bytes remaining to transmit: 30
Time allocated for transmission: 20 units
Packet of size 20 transmitted. ---- Bytes remaining to transmit: 10
Packet of size 10 transmitted. ---- Bytes remaining to transmit: 0

Incoming Packet size: 30
Bytes remaining to transmit: 30
Time allocated for transmission: 10 units
Packet of size 20 transmitted. ---- Bytes remaining to transmit: 10

Incoming Packet size: 10
Bytes remaining to transmit: 20
Time allocated for transmission: 10 units
Packet of size 20 transmitted. ---- Bytes remaining to transmit: 0

=== Code Execution Successful ===
```

Program 16

Aim of the program

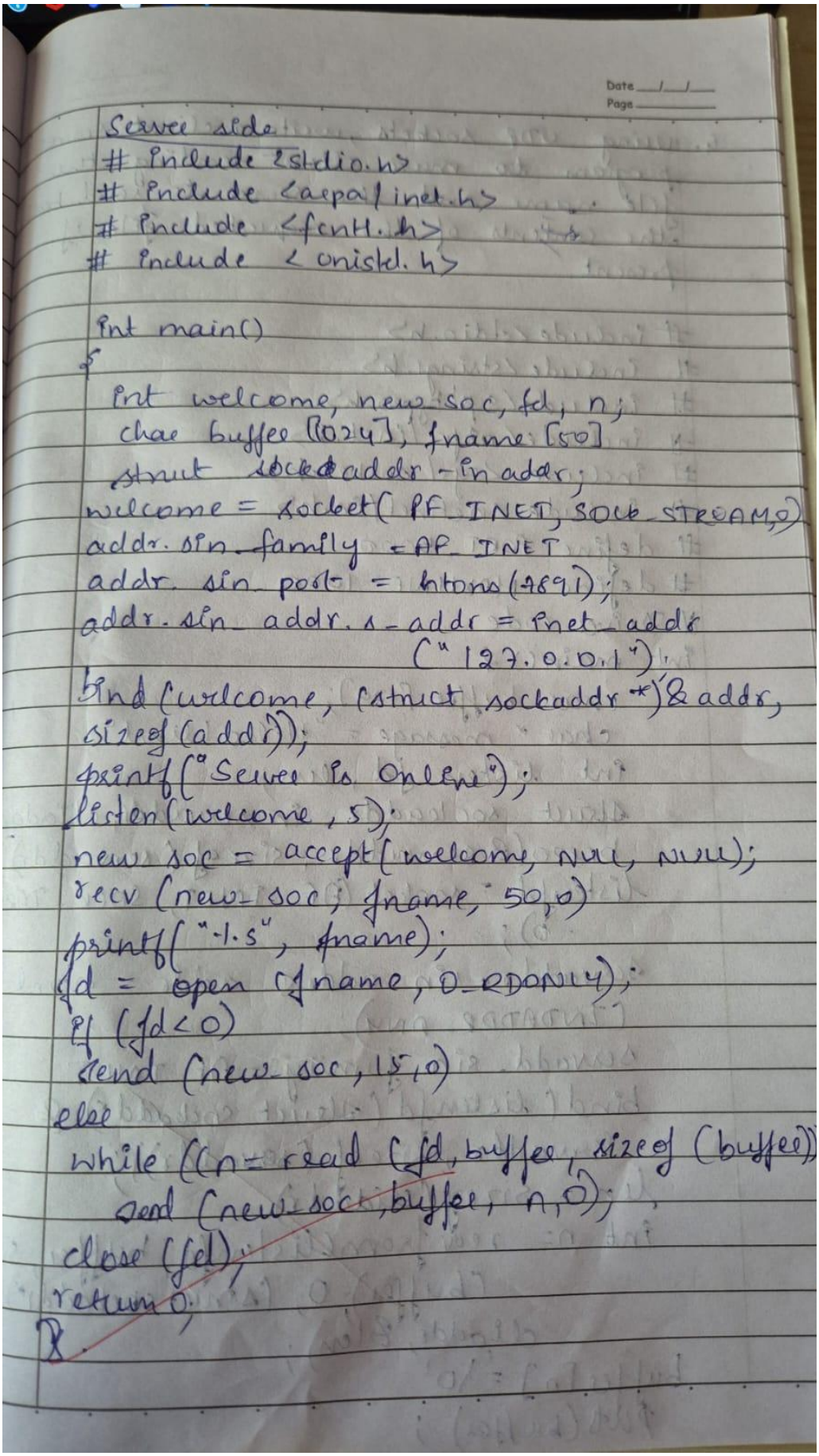
Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.



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

Client side

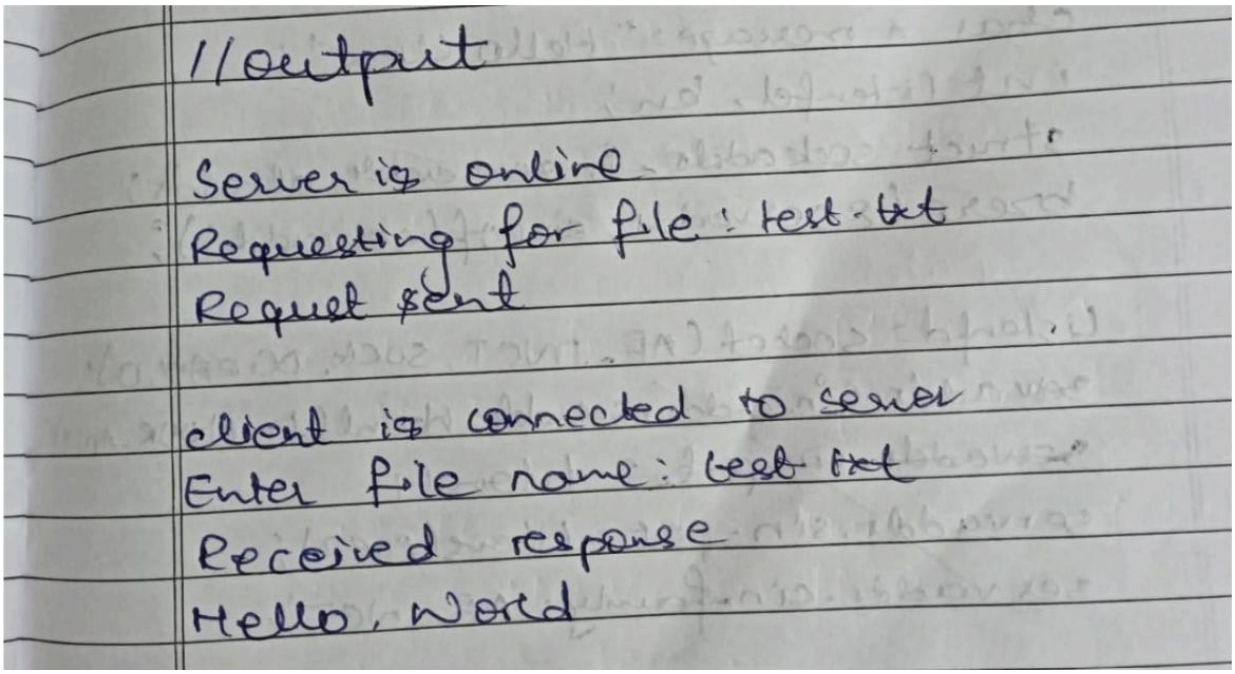
```
# include <unistd.h>
int main()
{
   int soc, n;
   char buffer[1024], fname[50]
    struct sockaddr_in addr;
soc = socket(PF_INET, SOCK_STREAM, 0);
addr.sin_family = AF_INET;
addr.sin_port = htons(7891);
addr.sin_addr.s_addr = inet_addr("127.0.0.1");
while(connect(soc,(struct sockaddr*)&
addr, sizeof(addr)));
printf("Enter file name")
scanf("%s", fname)
send(soc, fname, sizeof(fname), 0)
while((n = recv(soc, buffer, sizeof(buffer),
0)) > 0)
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(addr));
printf("Server is Online");
listen(welcome, 5);
new_soc = accept(welcome, NULL, NULL);
recv(new_soc, fname, 50,0)
printf("%s", fname);
fd = open(fname, O_RDONLY);
if (fd<0)
 send(new_soc, 15,0)
else
 while ((n= read(fd, buffer, sizeof(buffer))
   send(new_soc, buffer, n,0);
close(fd);
return 0;
}
```

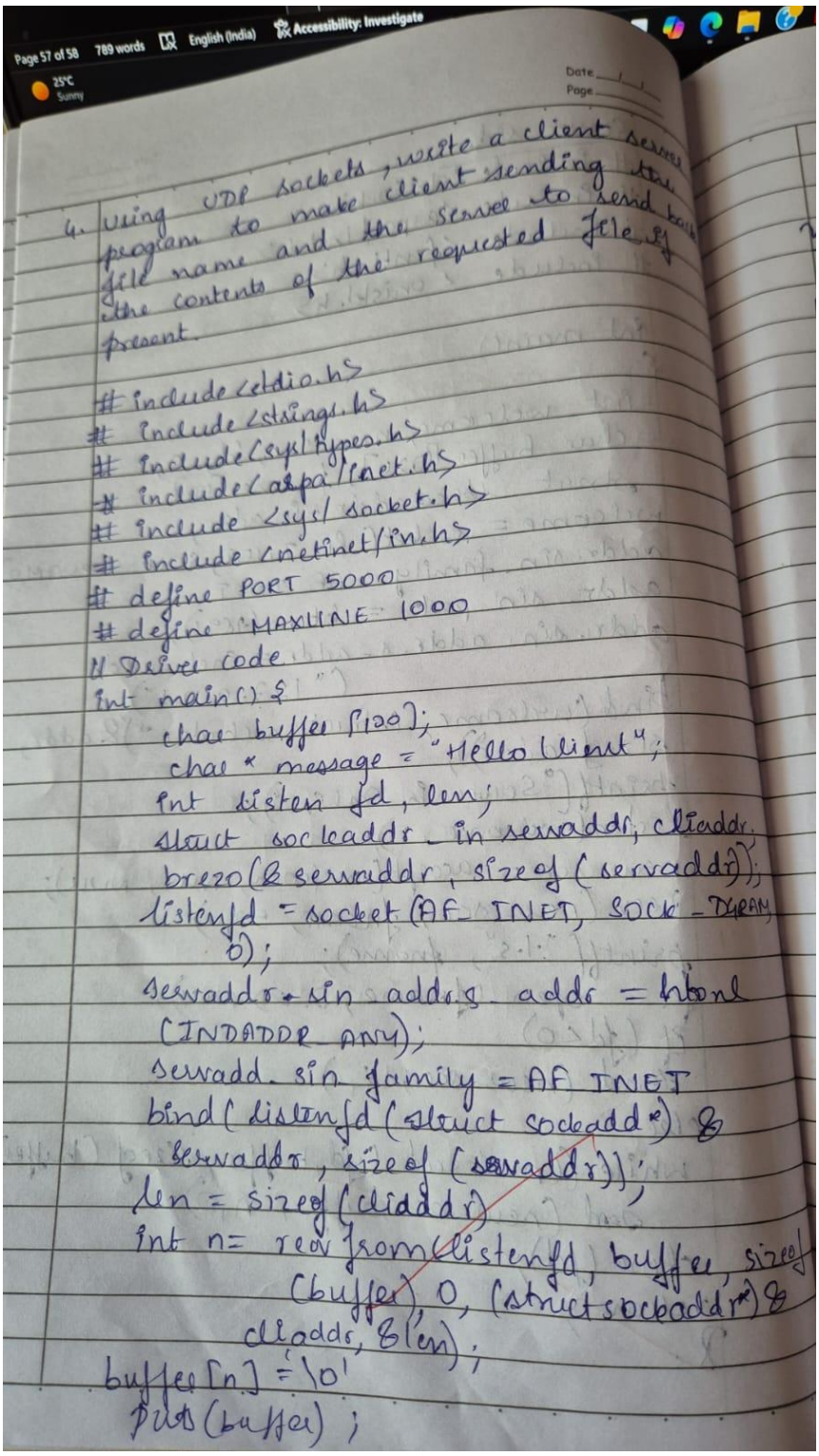
Output


//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

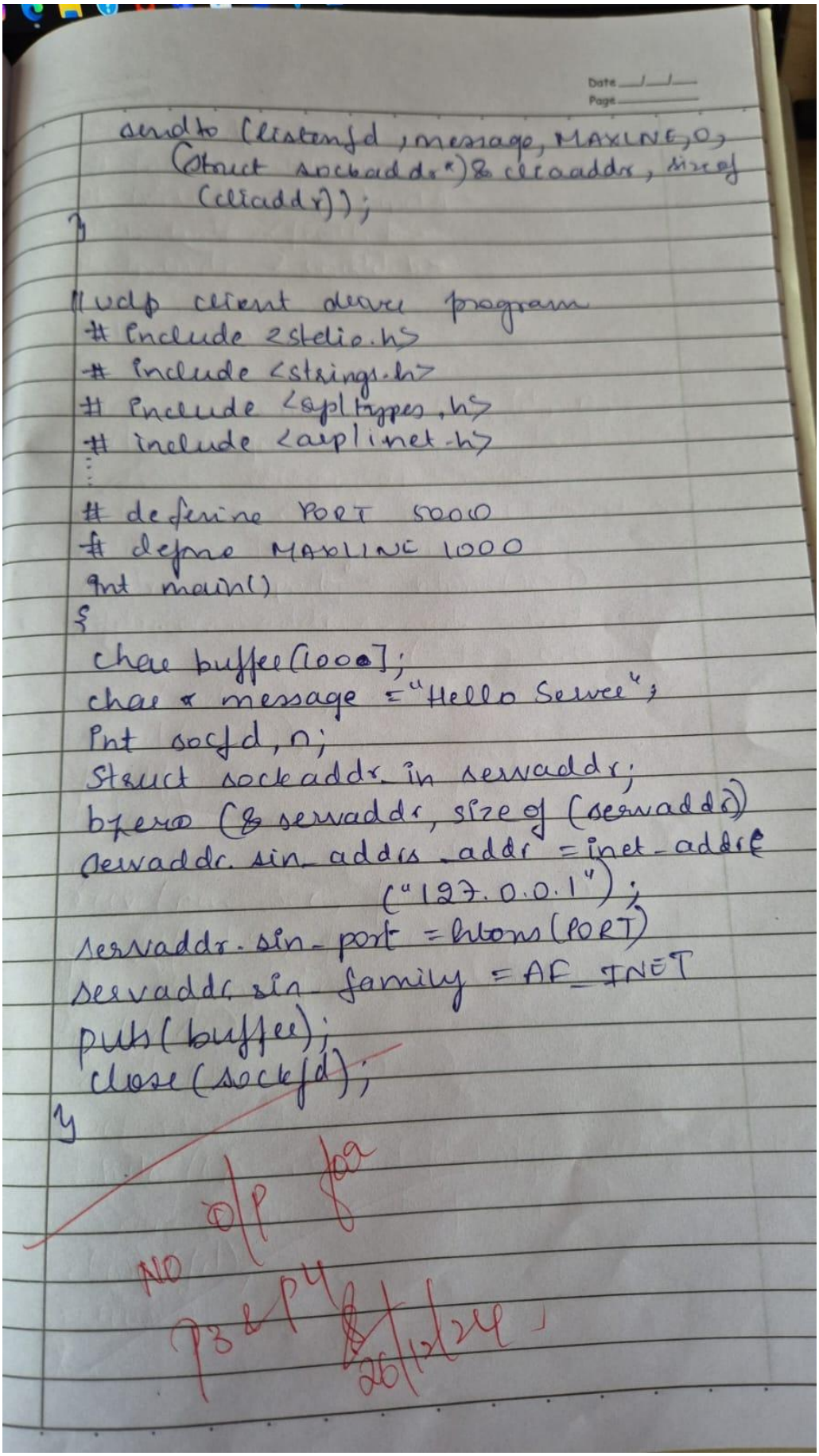
Aim of the program

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



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

```
#include<stdio.h>
#include<string.h>
#include<sys/types.h>
#include<arpa/inet.h>
#include<sys/socket.h>
#include<netinet/in.h>
#define PORT 5000
#define MAXLINE 1000
// Driver code
int main() {
    char buffer[100];
    char *message = "Hello Client";
    int listenfd, len;
    struct sockaddr_in servaddr, cliaddr;
    bzero(&servaddr, sizeof(servaddr));
    listenfd = socket(AF_INET, SOCK_DGRAM,
        0);
    servaddr.sin_addr.s_addr = htonl
    (INDADDR_ANY);
    servadd.sin_family = AF_INET
    bind(listenfd(struct sockadd*) &
        servaddr, sizeof(servaddr));
    len = sizeof(cliaddr)
    int n= recvfrom(listenfd, buffer, sizeof
        (buffer), 0, (struct sockaddr*) &
        cliaddr, &len);
    buffer[n] = '\0'
    puts(buffer);
```

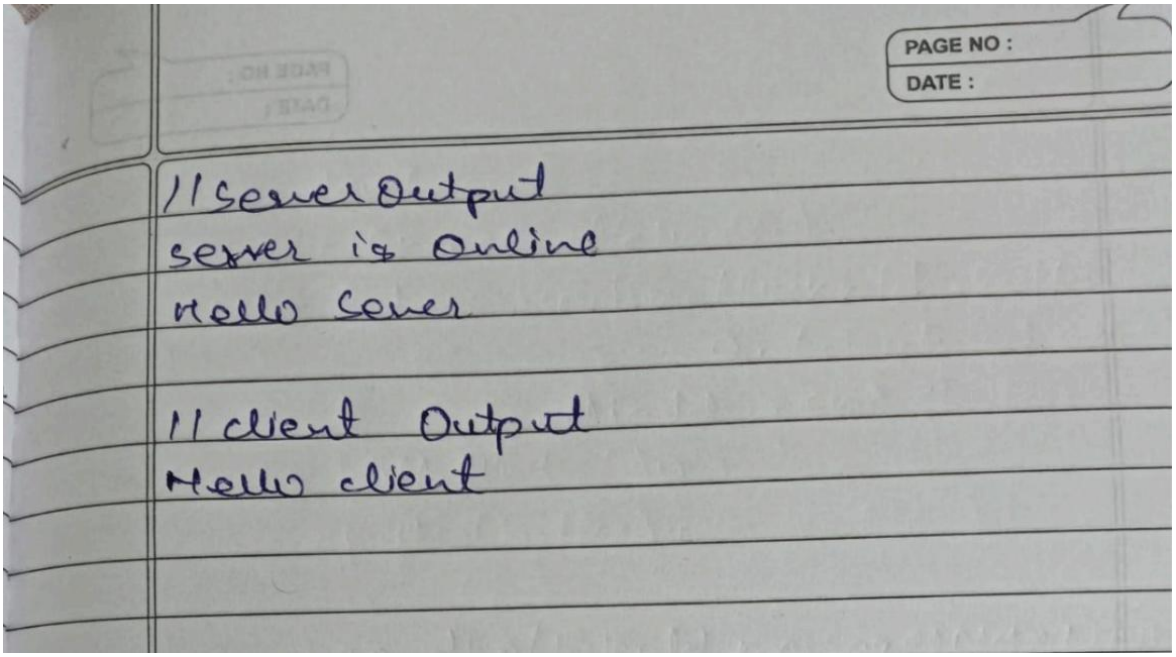


```
sendto (listenfd, message, MAXLINE, 0,
    (struct sockaddr *)& cliaddr, sizeof
    (cliaddr));
}

// udp client server program
#include <stdio.h>
#include <strings.h>
#include <sys/types.h>
#include <arpa/inet.h>
.
.
#define PORT 5000
#define MAXLINE 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
 puts(buffer);
 close(sockfd);
}
```

NO o/p for

73 & P4 & 26/12/24

Output


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