VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB RECORD

Computer Network Lab (23CS5PCCON)

Submitted by

VENUGOPALA C S (1BM23CS425)

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 **VENUGOPALA C S (1BM23CS425),** 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.

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

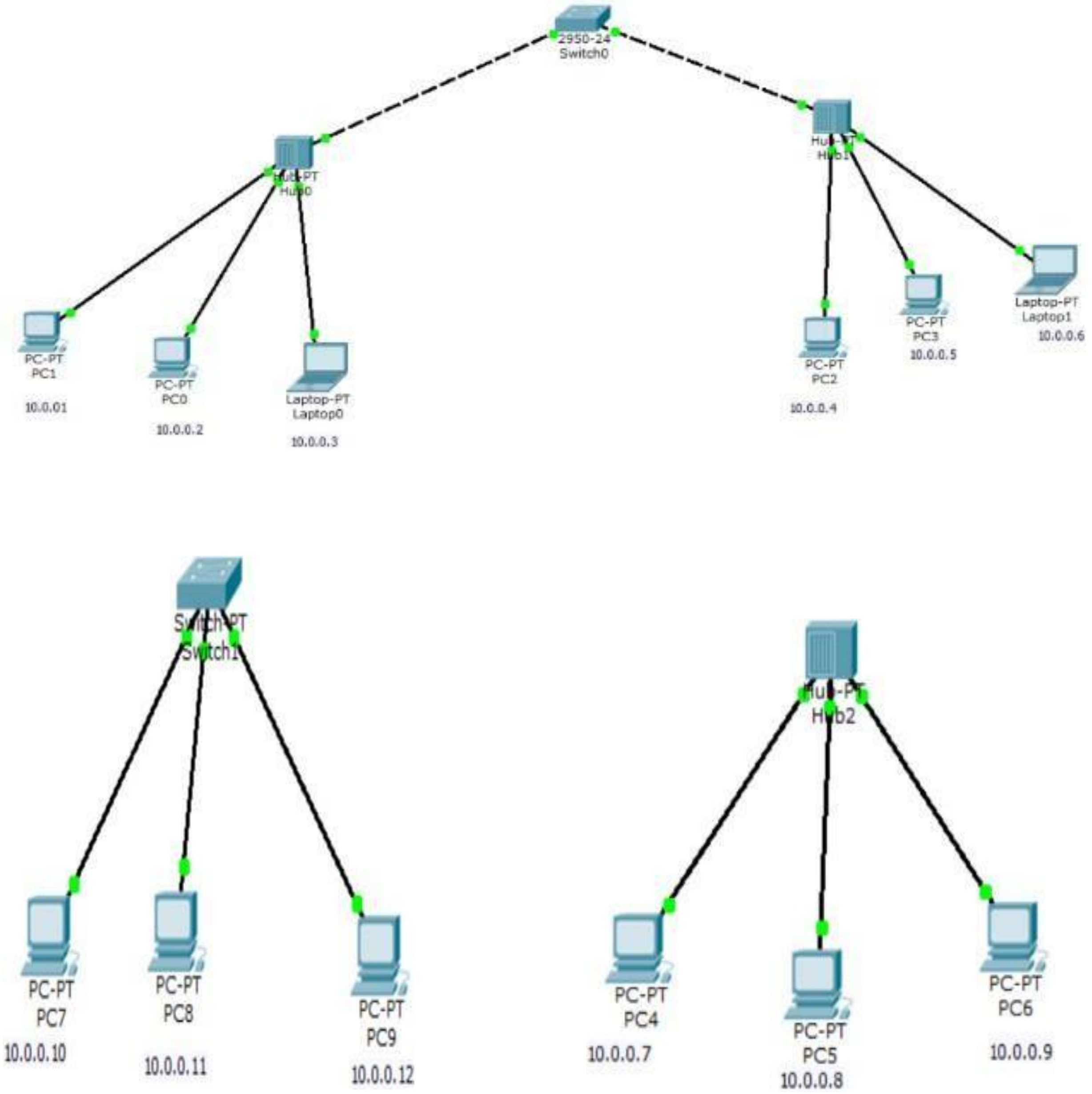
Index

Github Link : https://github.com/venug3727/computer-networks.git

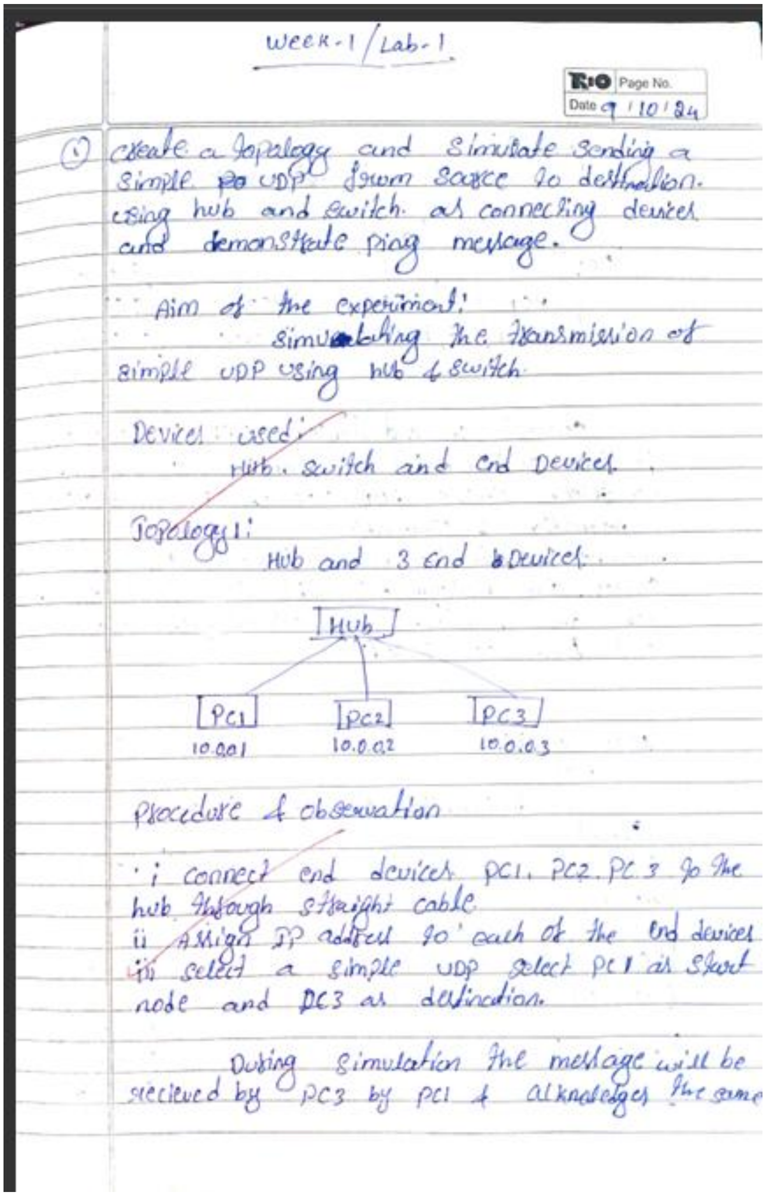
Program 1:

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 Observations:



week-1/Lab-1

Page No.
Date 9/10/24

(1) create a topology and simulate sending a simple ~~po~~ UDP from source to destination. using hub and switch as connecting devices and demonstrate ping message.

Aim of the experiment:
simulating the transmission of simple UDP using hub & switch

Devices used:
Hub. switch and end Devices .

Topology 1:
Hub and 3 end Devices.

Hub

PC1
10.0.0.1

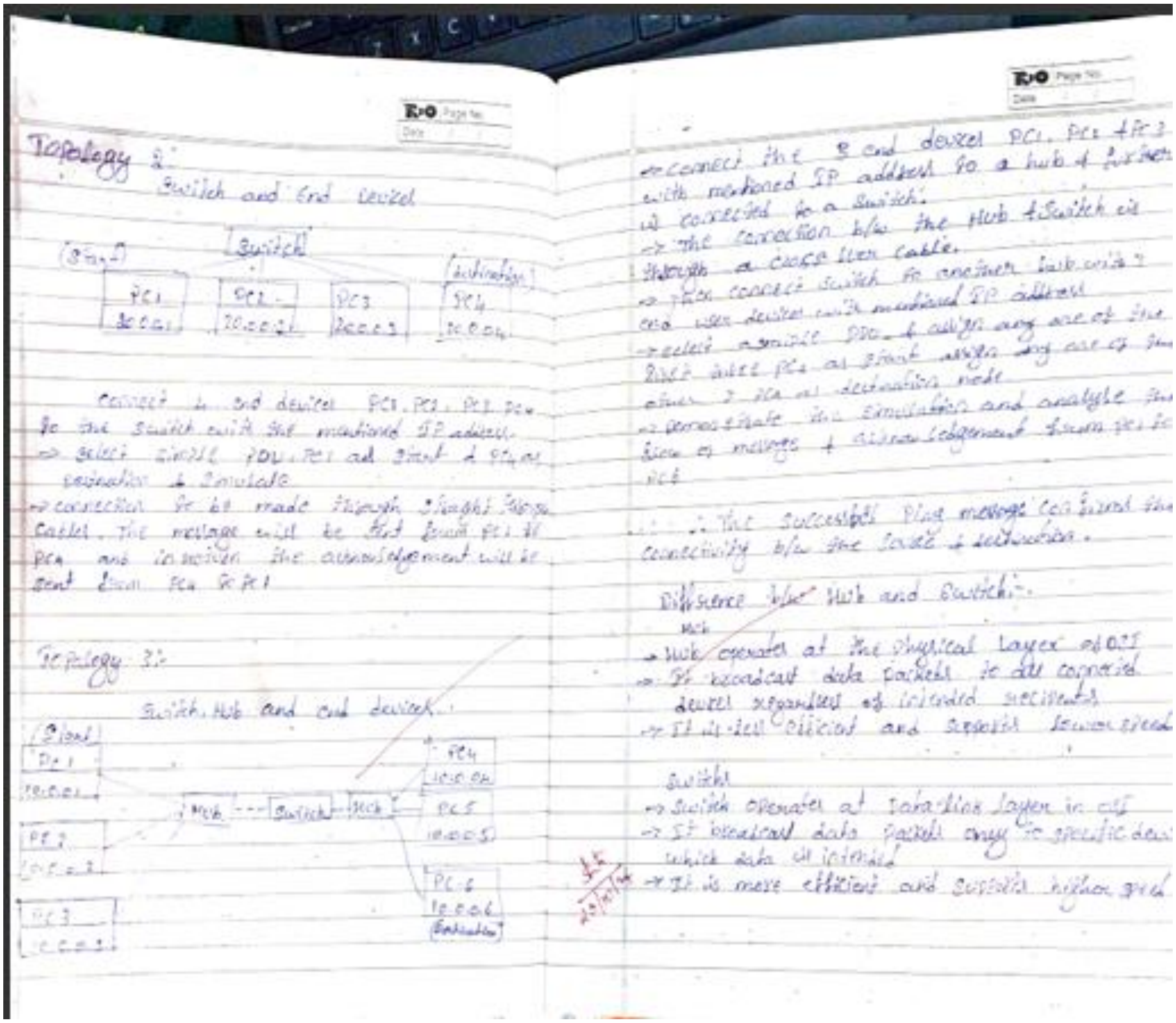
PC2
10.0.0.2

PC3
10.0.0.3

Procedure & observation

i connect end devices PC1, PC2, PC3 to the hub through straight cable
ii Assign IP address to each of the end devices
iii select a simple UDP select PC1 as start node and PC3 as destination.

During simulation the message will be recieved by PC3 by PC1 & acknowledges the same



Topology 2:

Switch and End devices

(Start)
Switch
(destination)
PC1
10.0.0.1
PC2
10.0.0.2
PC3
10.0.0.3
PC4
10.0.0.4

Connect 4 end devices PC1, PC2, PC3, PC4 to the switch with the mentioned IP address.
→ select simple PDU. PC1 as start & PC4 as destination & simulate
→ connection to be made through straight through cables. The message will be sent from PC1 to PC4 and in return the acknowledgement will be sent from PC4 to PC1

Topology 3:

Switch, Hub and end devices.

→ connect the 3 end devices PC1, PC2 & PC3 with mentioned IP address to a hub & further is connected to a switch.
→ The connection b/w the Hub & Switch is through a cross over cable.
→ then connect switch to another hub with 3 end user devices with mentioned IP address
→ select a simple PDU & assign any one of the first three PCs as start & assign any one of the other 3 PCs as destination node
→ demonstrate the simulation and analyse the flow of message & acknowledgement from PC1 to PC6

∴ The successful Ping message confirms the connectivity b/w the source & destination.

Difference b/w Hub and Switch:-

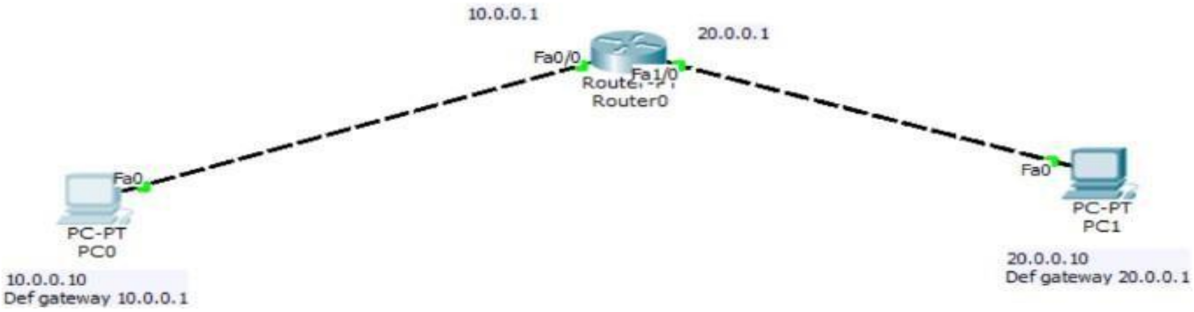
Hub
→ Hub operates at the physical layer of OSI
→ It broadcast data packets to all connected devices regardless of intended recipients
→ It is less efficient and supports lower speed

Switch
→ Switch operates at Data-link layer in OSI
→ It broadcast data packets only to specific devices which data is intended
→ It is more efficient and supports higher speed

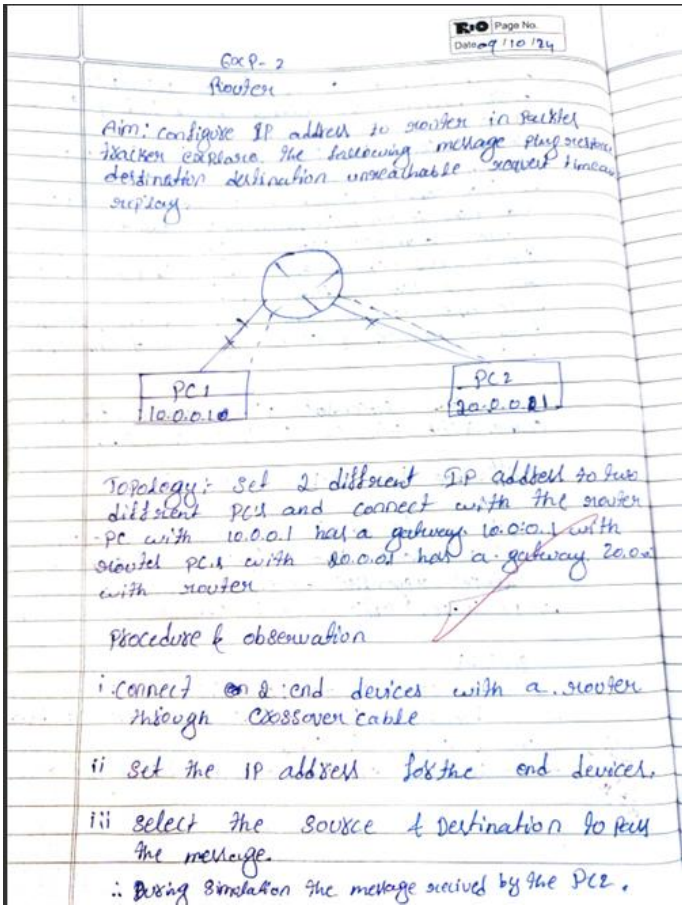
Program 2 :

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

Topology:



Procedure and Observations:



RIO Page No.
Date 09/10/24

Exp- 2
Router

Aim: configure IP address to router in packet tracer explore the following message ping response destination destination unreachable request timeout reply.

PC1
10.0.0.10

PC2
20.0.0.21

Topology:- set 2 different IP address to two different PC's and connect with the router -PC with 10.0.0.1 has a gateway 10.0.0.1 with router PC.2 with 20.0.0.1 has a gateway 20.0.0.1 with router

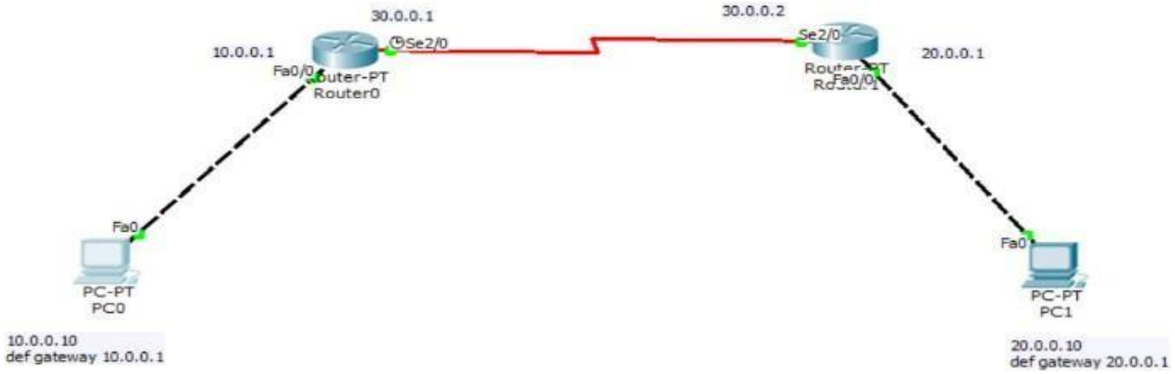
Procedure & observation

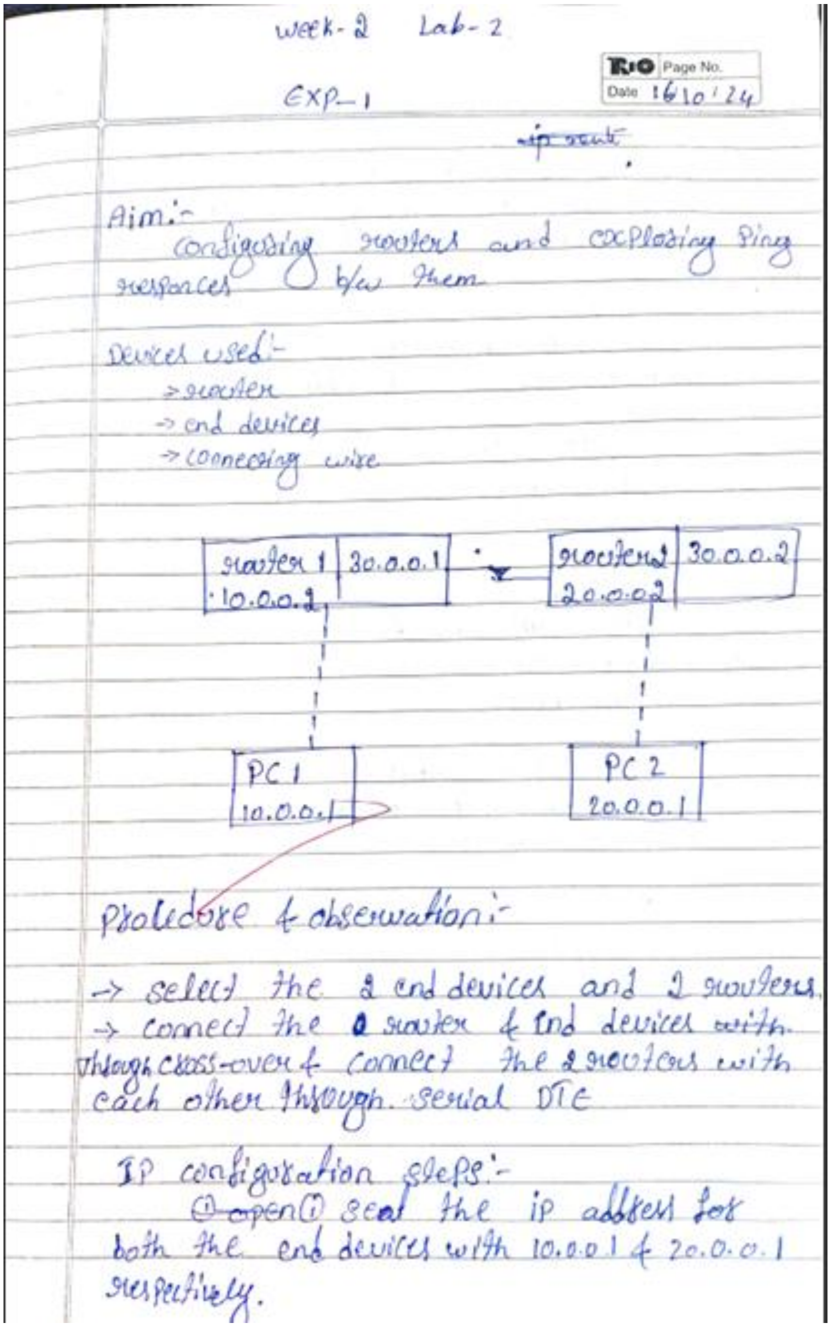
i. connect ~~on~~ 2 end devices with a router through crossover cable

ii set the IP address for the end devices.

iii select the source & Destination to pass the message.

∴ During simulation the message received by the PC2.





week-2 Lab-2

RIO Page No.

Date 16/10/24

EXP-1

~~ip route~~.

Aim:-

configuring routers and exploring ping responces b/w them

Devices used:-

>router

-> end devices

-> connecting wire

router 1 | 30.0.0.1
10.0.0.1

router2 | 30.0.0.2
20.0.0.2

PC 1
10.0.0.1

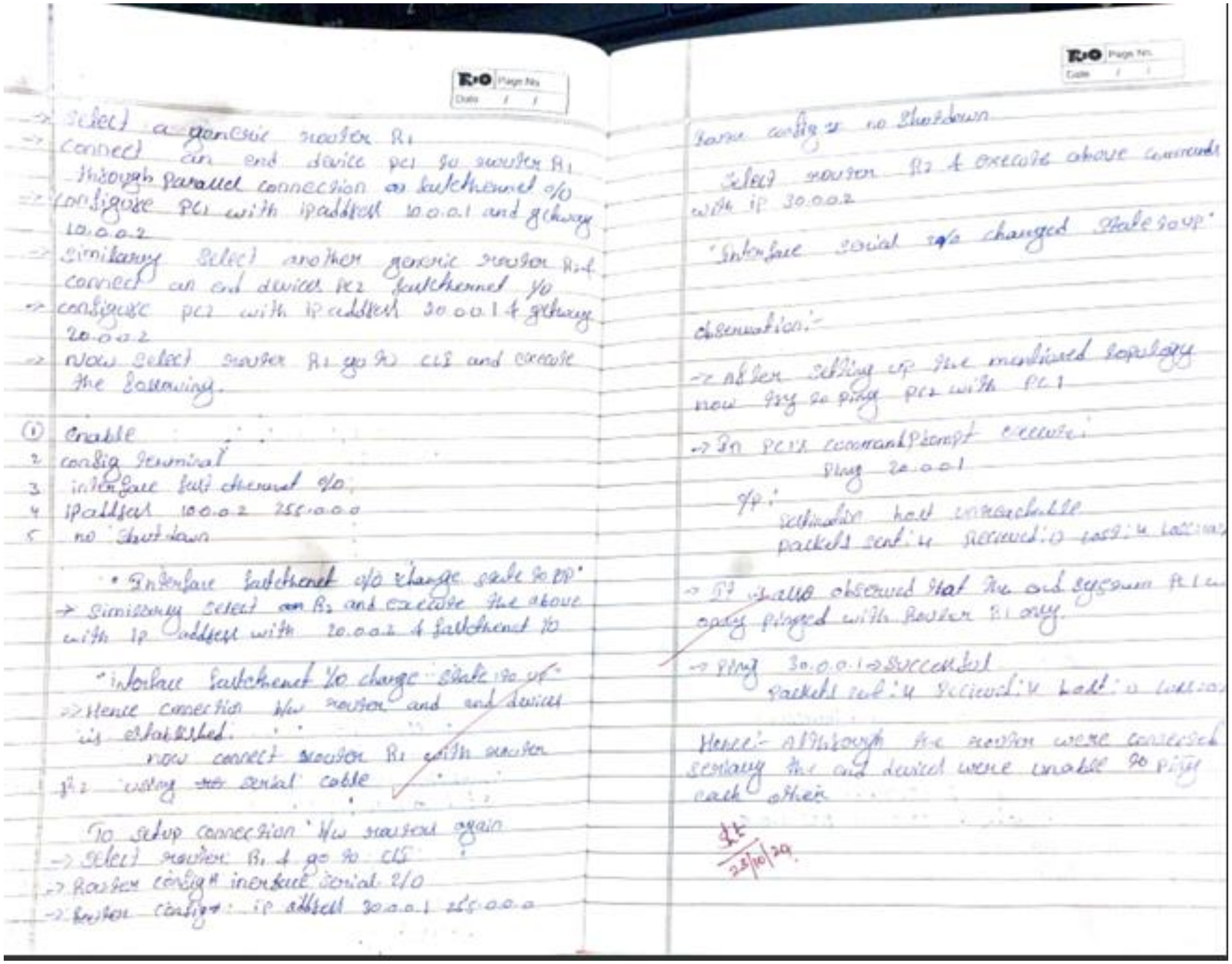
PC 2
20.0.0.1

Procedure & observation:-

-> select the 2 end devices and 2 routers.
-> connect the 2 router & end devices with through cross-over & connect the 2 routers with each other through serial DTE

IP configuration steps:-

~~① open~~ ① seal the ip address for both the end devices with 10.0.0.1 & 20.0.0.1 respectively.



-> Select a generic router R1
-> Connect an end device pc1 to router R1 through parallel connection ~~as~~ fastethernet 0/0
-> Configure PC1 with ipaddress 10.0.0.1 and gateway 10.0.0.2
-> Similarly select another generic router R2 & connect an end device PC2 fastethernet 0/0
-> configure pc2 with ipaddress 20.0.0.1 & gateway 20.0.0.2
-> Now select router R1 go to CLI and execute the following.

① enable
2 config terminal
3 interface fast ethernet 0/0
4 ipaddress 10.0.0.2 255.0.0.0
5 no shutdown

• Interface fastethernet 0/0 change state to up"
-> Similarly select ~~on~~ R2 and execute the above with ip address with 20.0.0.2 & fastethernet 0/0

"interface fastethernet 0/0 change state to up"
=> Hence connection b/w router and end devices is established.
now connect router R1 with router R2 using ~~the~~ serial cable

To setup connection b/w routers again
-> Select router R1 & go to CLI
-> Router config# interface serial 2/0
-> Router config#: ip address 30.0.0.1 255.0.0.0

Same config # no shutdown
Select router R2 & execute above commands with ip 30.0.0.2
"Interface serial ~~2/0~~ changed state to up"

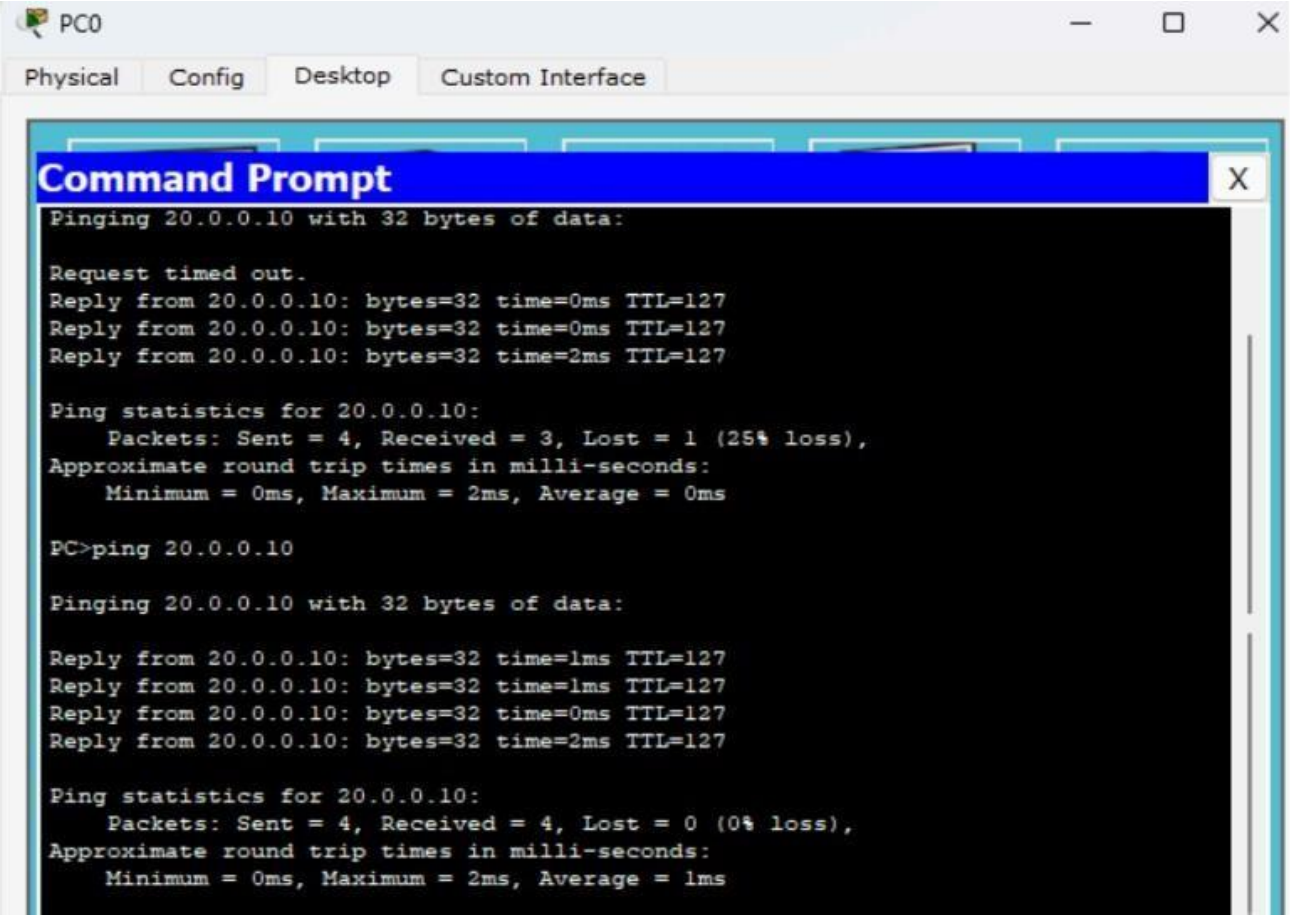
observation:-
-> After setting up the mentioned topology now try to ping pc2 with PC1
-> In PC1's command prompt execute:
ping 20.0.0.1
o/p:
destination host unreachable
packets sent: 4 received: 0 lost: 4 loss: [illegible]

-> It is also observed that the end systems PC1 can only pinged with Router R1 only.

-> Ping 30.0.0.1 => successful
packets sent: 4 received: 4 lost: 0 loss: [illegible]

Hence:- Although the routers were connected serially the end devices were unable to ping each other

23/10/24.

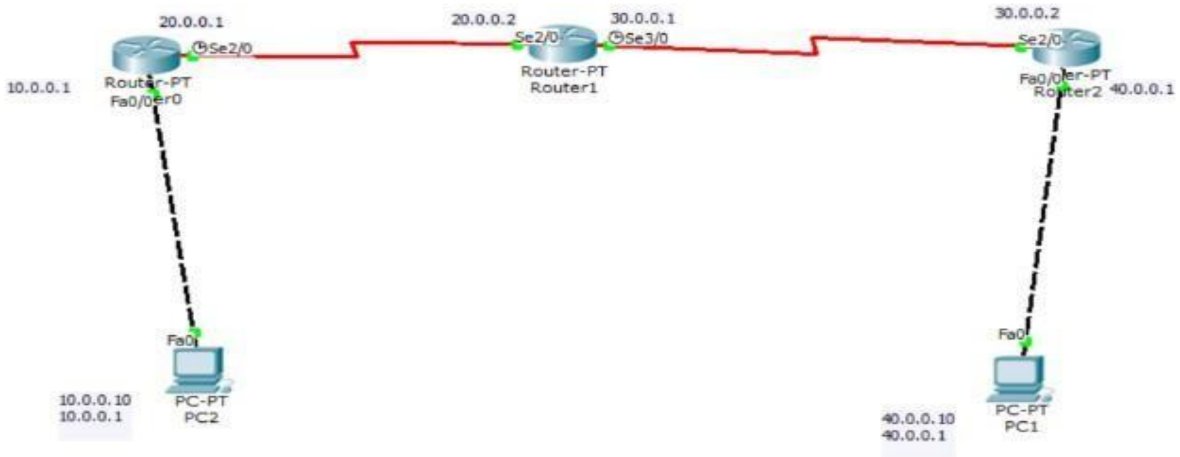


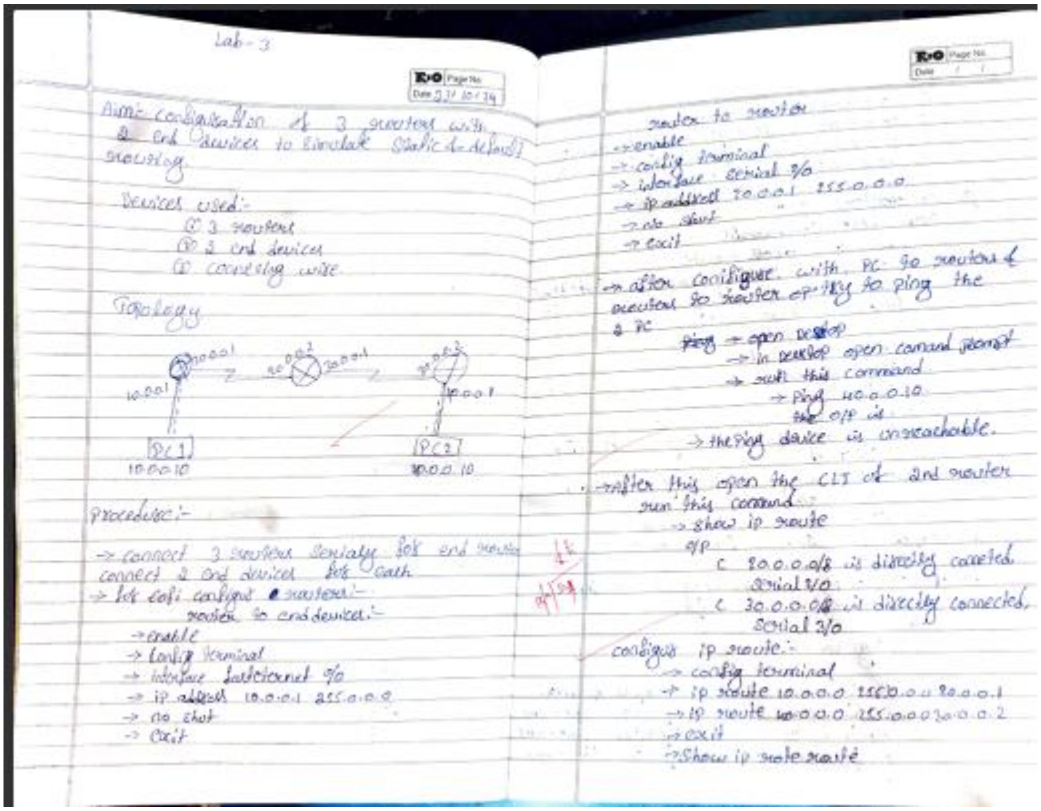
```
Pinging 20.0.0.10 with 32 bytes of data:

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=2ms 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 = 2ms, 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=1ms 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=2ms 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 = 2ms, Average = 1ms
```

Program 3:

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

Topology, Procedure and Observations:





Lab-3

Date 21/10/24

Aim:- configuration of 3 routers with 2 end devices to simulate static & default routing.

Devices used:-
- ① 3 routers
- ② 2 end devices
- ③ connecting wire

Topology

20.0.0.1
20.0.0.2
30.0.0.1
30.0.0.2
10.0.0.1
40.0.0.1
PC1
10.0.0.10
PC2
40.0.0.10

Procedure:-

-> connect 3 routers serially for end router connect 2 end devices for each
-> for confi configure routers:-
router to end devices:-

```
-> enable
-> config terminal
-> interface fastethernet 0/0
-> ip address 10.0.0.1 255.0.0.0
-> no shut
-> exit
```

router to router

```
-> enable
-> config terminal
-> interface serial 2/0
-> ip address 20.0.0.1 255.0.0.0
-> no shut
-> exit
```

-> after configure with PC to routers & routers to router, try to ping the 2 PC

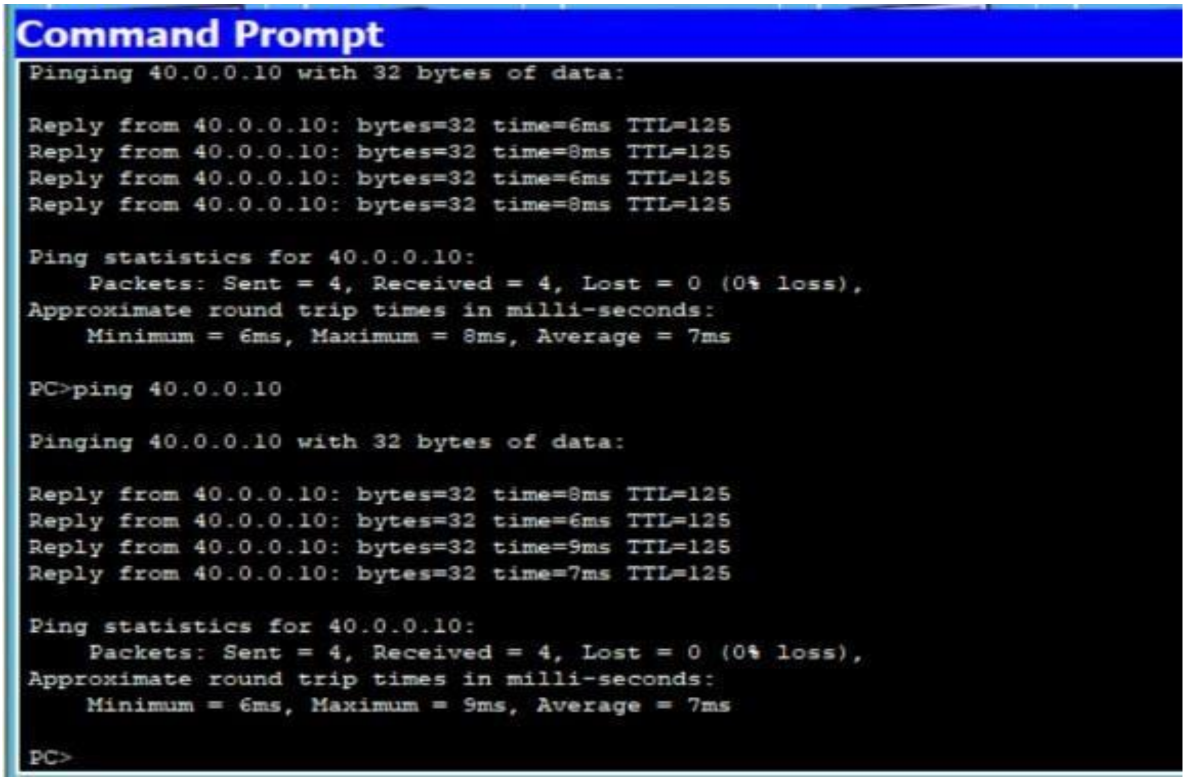
ping -> open Desktop
-> in Desktop open command prompt
-> run this command
-> ping 40.0.0.10
the o/p is
-> the ping device is unreachable.

-> after this open the CLI of 2nd router run this command
-> show ip route
o/p

```
C 20.0.0.0/8 is directly conneted.
  serial 2/0
C 30.0.0.0/8 is directly connected.
  serial 3/0
```

configure ip route:-

```
-> 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
```



Command Prompt

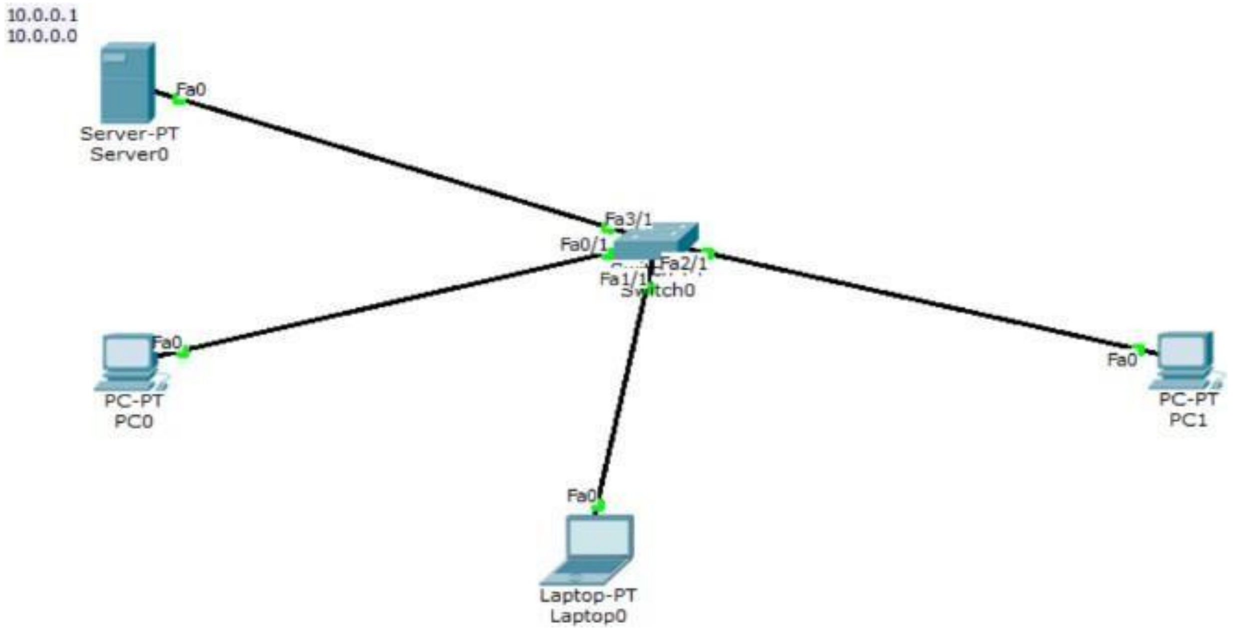
```
Pinging 40.0.0.10 with 32 bytes of data:

Reply from 40.0.0.10: bytes=32 time=6ms TTL=125
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=8ms 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 = 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=9ms 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 = 6ms, Maximum = 9ms, Average = 7ms

PC>
```

Program 4:

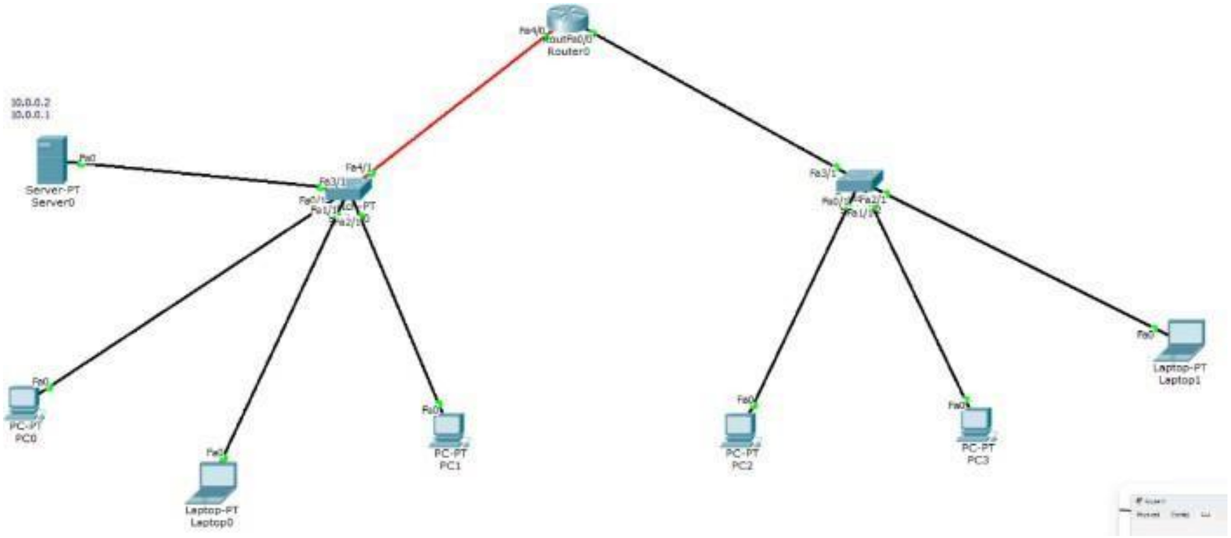
Aim: Configure DHCP within a LAN and outside LAN.

Topology:

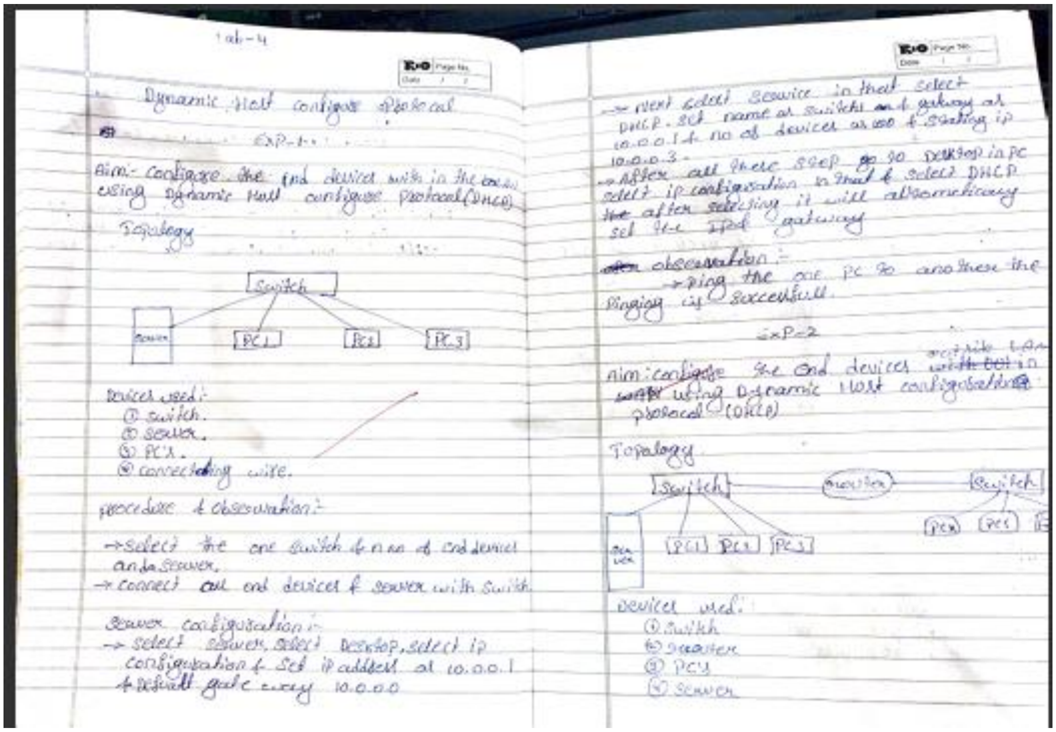
Within LAN



Outside LAN



Procedure and Observation:



Lab-4

Page No.
Date

Dynamic Host configure Protocol

EXP-1

Aim:- Configure the end devices with in the ~~LAN~~ using Dynamic Host configure Protocol (DHCP)

Topology

Switch
Server
PC1
PC2
PC3

Devices used:-

1. switch.
2. server.
3. PC's.
4. connecting wire.

procedure & observation:-

→ select the one switch & n no of end devices and a server.

→ connect all end devices & server with switch.

server configuration:-

→ select server, select Desktop, select ip configuration & set ip address at 10.0.0.1 & default gate way 10.0.0.0

Page No.
Date

→ next select service in that select DHCP. set name as switch1 ~~and~~ & gateway as 10.0.0.1 & no of devices as 100 & starting ip 10.0.0.3.

→ After all these step go to Desktop in PC select ip configuration in that & select DHCP ~~the~~ after selecting it will automatically set the IP & gateway

~~obs~~ observation:-

→ ping the one PC to another the pinging is successfull.

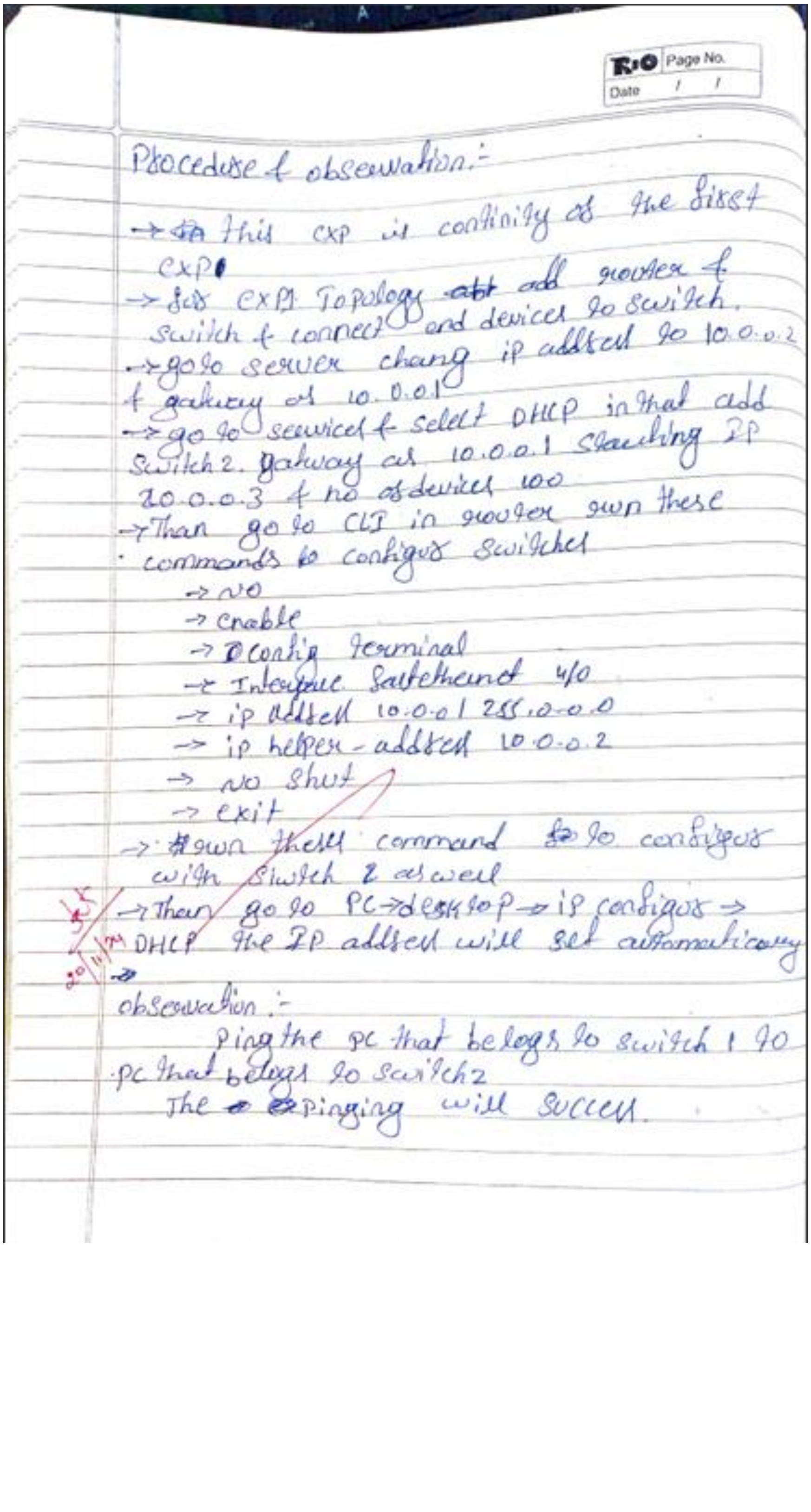
EXP-2

Aim: configure the end devices ~~with the LAN~~ in multiple LAN ~~switch~~ using Dynamic Host configuration protocol (DHCP)

Topology

Device used:

1. Switch
2. Router
3. PC's
4. Server



Page No.
Date / /

Procedure & observation:-

→ This cxp is continity of the first cxp

→ for cxp1 Topology ~~abt~~ add router & switch & connect and devices to switch.

→ go to server chang ip address to 10.0.0.2 & gateway of 10.0.0.1

→ go to services & select DHCP in that add Switch2. gateway as 10.0.0.1 Starting IP 10.0.0.3 & no of devices 100

→ Than go to CLI in router run these commands to configur switches

→ no
→ enable
→ config terminal
→ Interface fastethernet 4/0
→ ip address 10.0.0.1 255.0.0.0
→ ip helper - address 10.0.0.2
→ no shut
→ exit

→ run these command to to configur with Switch 2 as well

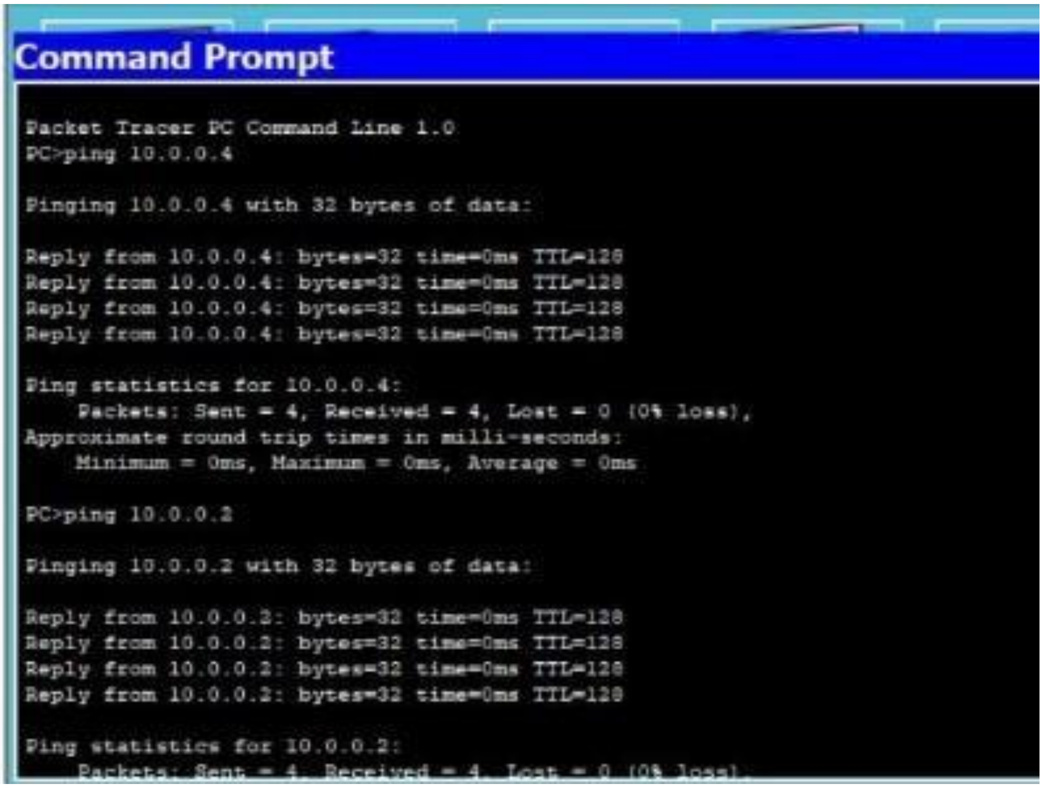
→ Than go to PC → desktop → ip configur → DHCP the IP address will set automaticany

5/5 20/11/24

observation :-

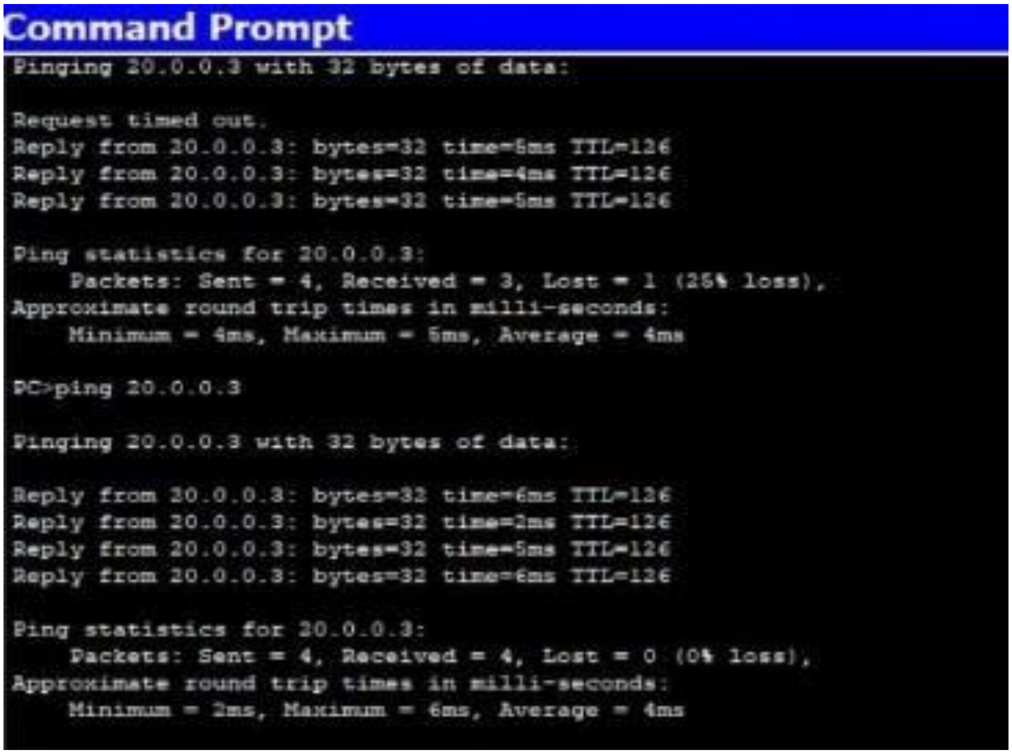
Ping the pc that belogs to switch 1 to pc that belogs to switch2

The pinging will succes.



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

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=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

Within LAN



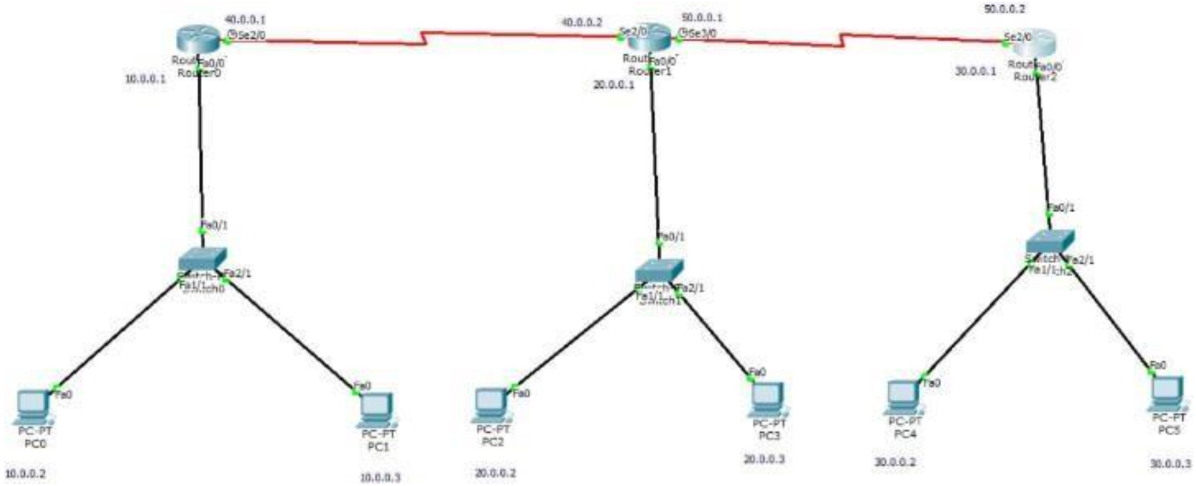
```
Pinging 20.0.0.3 with 32 bytes of data:

Request timed out.
Reply from 20.0.0.3: bytes=32 time=5ms TTL=126
Reply from 20.0.0.3: bytes=32 time=4ms TTL=126
Reply from 20.0.0.3: bytes=32 time=5ms TTL=126

Ping statistics for 20.0.0.3:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 4ms, Maximum = 5ms, Average = 4ms

PC>ping 20.0.0.3

Pinging 20.0.0.3 with 32 bytes of data:

Reply from 20.0.0.3: bytes=32 time=6ms TTL=126
Reply from 20.0.0.3: bytes=32 time=2ms TTL=126
Reply from 20.0.0.3: bytes=32 time=5ms TTL=126
Reply from 20.0.0.3: bytes=32 time=6ms TTL=126

Ping statistics for 20.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 6ms, Average = 4ms
```

Outside LAN

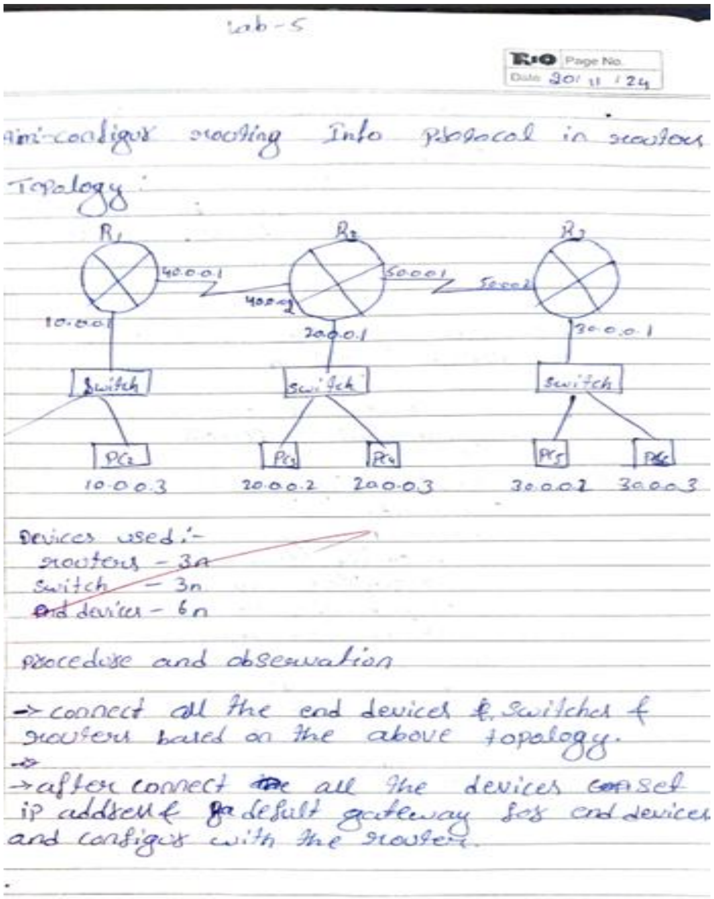
Program 5:

Aim: Configure RIP routing Protocol in Routers.

Topology:



Procedure and Observation:


lab-5
Page No.
Date 30/11/24
R1
R2
R3
40.0.0.1
40.0.0.2
50.0.0.1
50.0.0.2
10.0.0.1
20.0.0.1
30.0.0.1
Switch
Switch
Switch
PC2
PC3
PC4
PC5
PC6
10.0.0.3
20.0.0.2
20.0.0.3
30.0.0.2
30.0.0.3

aim-configur routing Info protocol in routers

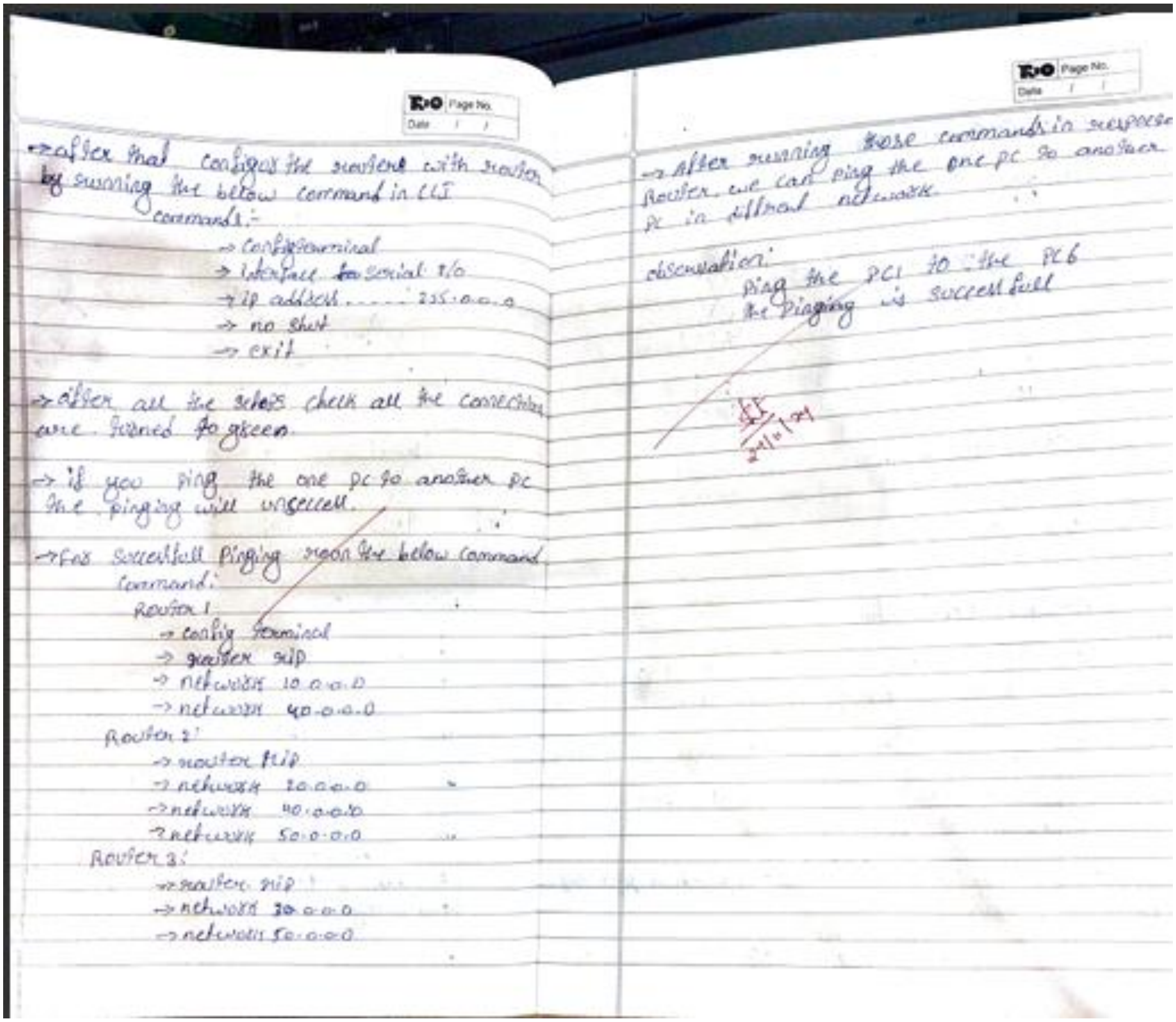
Topology:

Devices used:-
routers - 3n
switch - 3n
~~end~~ devices - 6n

procedure and observation

-> connect all the end devices & switches & routers based on the above topology.

-> after connect ~~the~~ all the devices ~~con~~ set ip address & ~~ga~~ defult gateway for end devices and configur with the router.



Page No.
Date / /

→ after that configur the routers with router by running the below command in CLI

commands:-

```
→ config terminal
→ interface ~~to~~ serial 2/0
→ ip address ..... 255.0.0.0
→ no shut
→ exit
```

→ after all the steps check all the connection are turned to green.

→ if you ping the one pc to another pc the pinging will unsuccess.

→ For successfull pinging run the below command

Command:

Router 1

```
→ config terminal
→ router rip
→ network 10.0.0.0
→ network 40.0.0.0
```

Router 2:

```
→ router rip
→ network 20.0.0.0
→ network 40.0.0.0
→ network 50.0.0.0
```

Router 3:

```
→ router rip
→ network 30.0.0.0
→ network 50.0.0.0
```

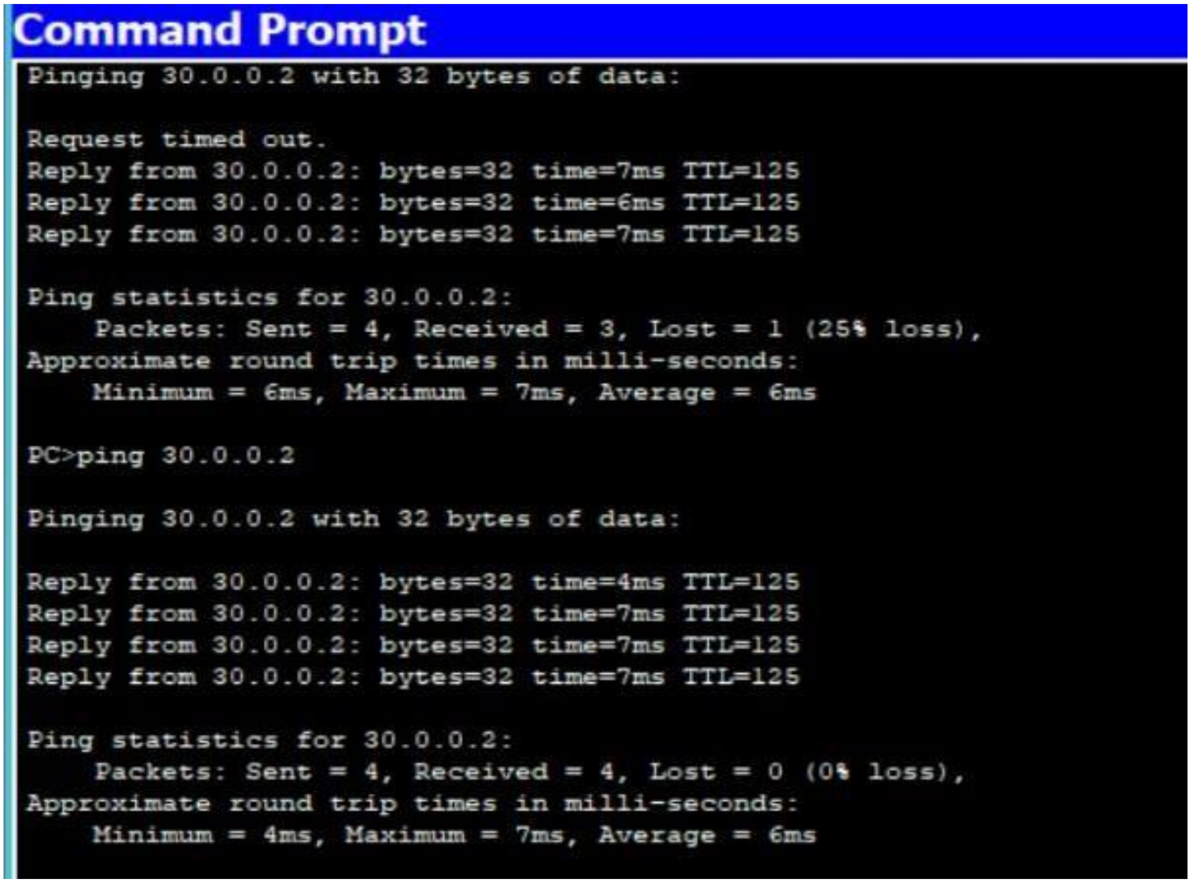
Page No.
Date / /

→ After running those commands in respective Router we can ping the one pc to another pc in differnt network.

observation:

Ping the PC1 to the PC6
the Pinging is successfull

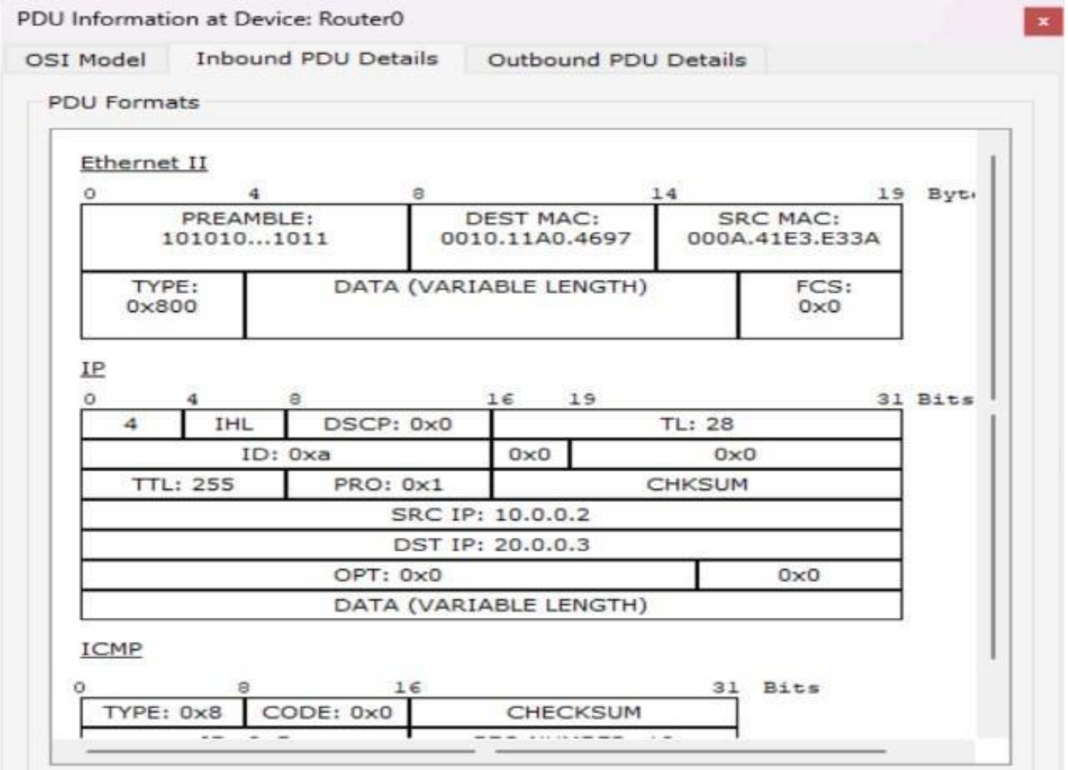
24/10/24



```
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
```

Program 6:

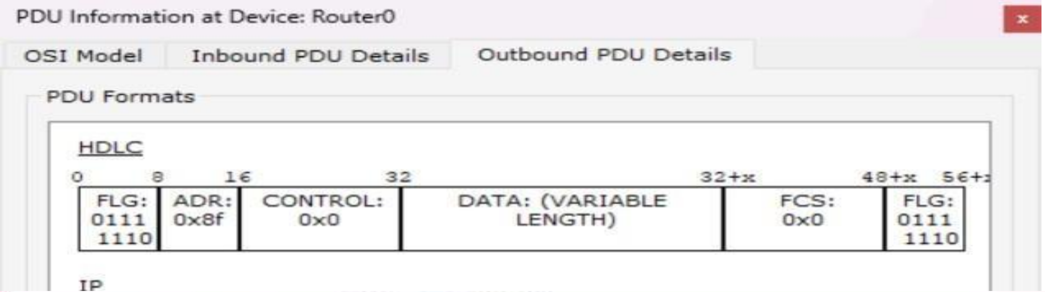
Aim: Demonstrate the TTL/ Life of a Packet.



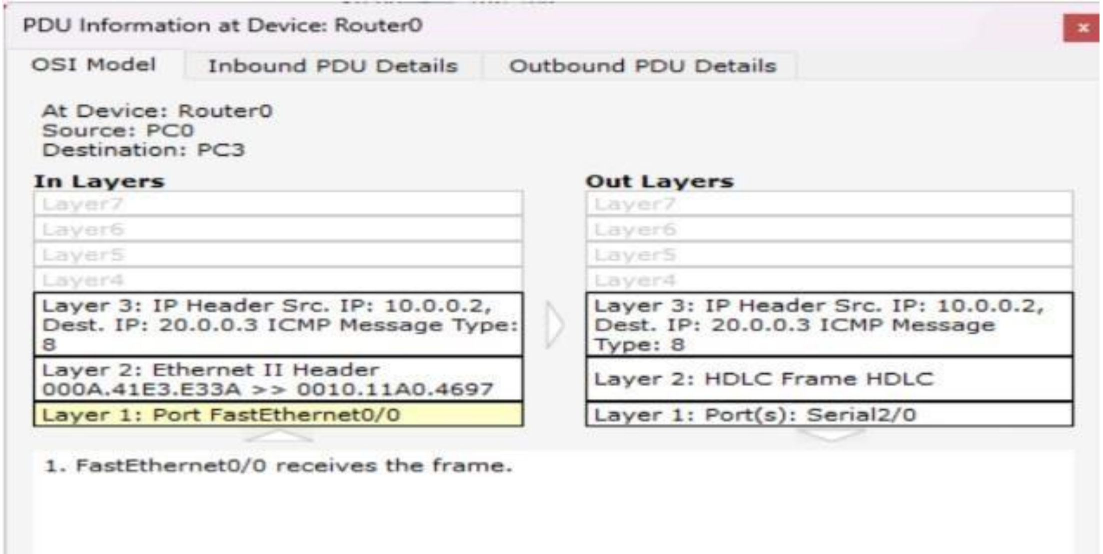
PREAMBLE: 101010...1011
DEST MAC: 0010.11A0.4697
SRC MAC: 000A.41E3.E33A
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0

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)

TYPE: 0x8
CODE: 0x0
CHECKSUM



FLG: 0111 1110
ADR: 0x8f
CONTROL: 0x0
DATA: (VARIABLE LENGTH)
FCS: 0x0
FLG: 0111 1110



At Device: Router0
Source: PC0
Destination: PC3

In 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

Out Layers

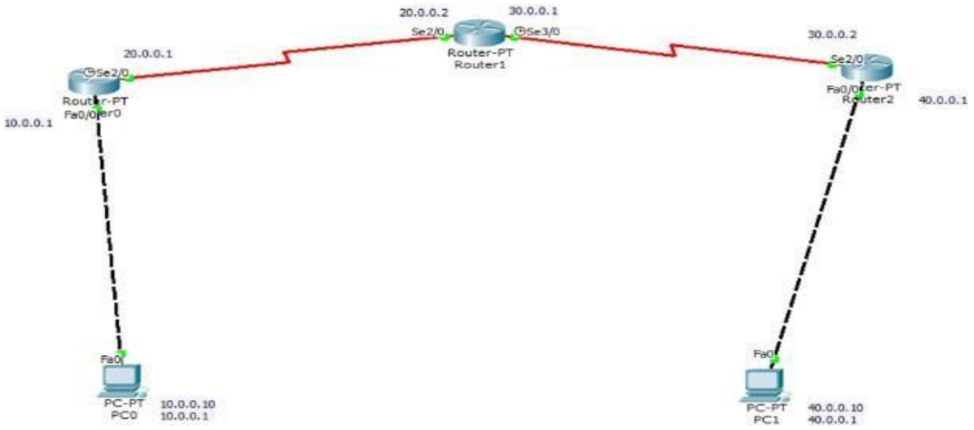
Layer7
Layer6
Layer5
Layer4
Layer 3: IP Header Src. IP: 10.0.0.2, Dest. IP: 20.0.0.3 ICMP Message Type: 8
Layer 2: HDLC Frame HDLC
Layer 1: Port(s): Serial2/0

1. FastEthernet0/0 receives the frame.

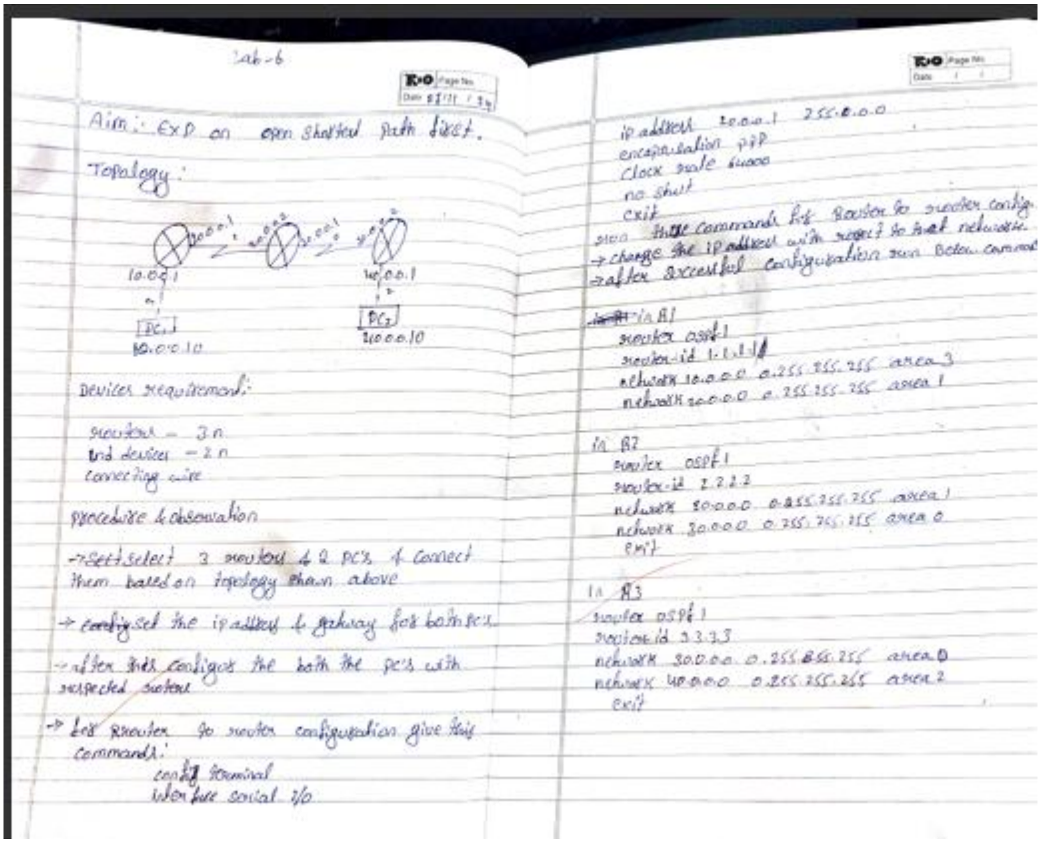
Program 7:

Aim: Configure OSPF routing protocol**.**

Topology:



Procedure and Observation:



lab-6

Date 08/11/24

Aim: Exp on open shortest path first.

Topology:

10.0.0.1 20.0.0.1 20.0.0.2 30.0.0.1 30.0.0.2 40.0.0.1

PC1 10.0.0.10

PC2 40.0.0.10

Devices requirement:

routers - 3 n
end devices - 2 n
connecting wire

procedure & observation

→ ~~set~~ select 3 routers & 2 PC's & connect them based on topology shown above

→ ~~config~~ set the ip address & gateway for both pc's

→ after this configure the both the pc's with respected router

→ for Router go router configuration give this commands:

```
config terminal
interface serial 2/0
ip address 20.0.0.1 255.0.0.0
encapsulation ppp
clock rate 64000
no shut
exit
```

~~use~~ these commands for Router to router config.

→ change the ip address with respect to that network.

→ after successful configuration run Below command

~~[illegible]~~ in R1

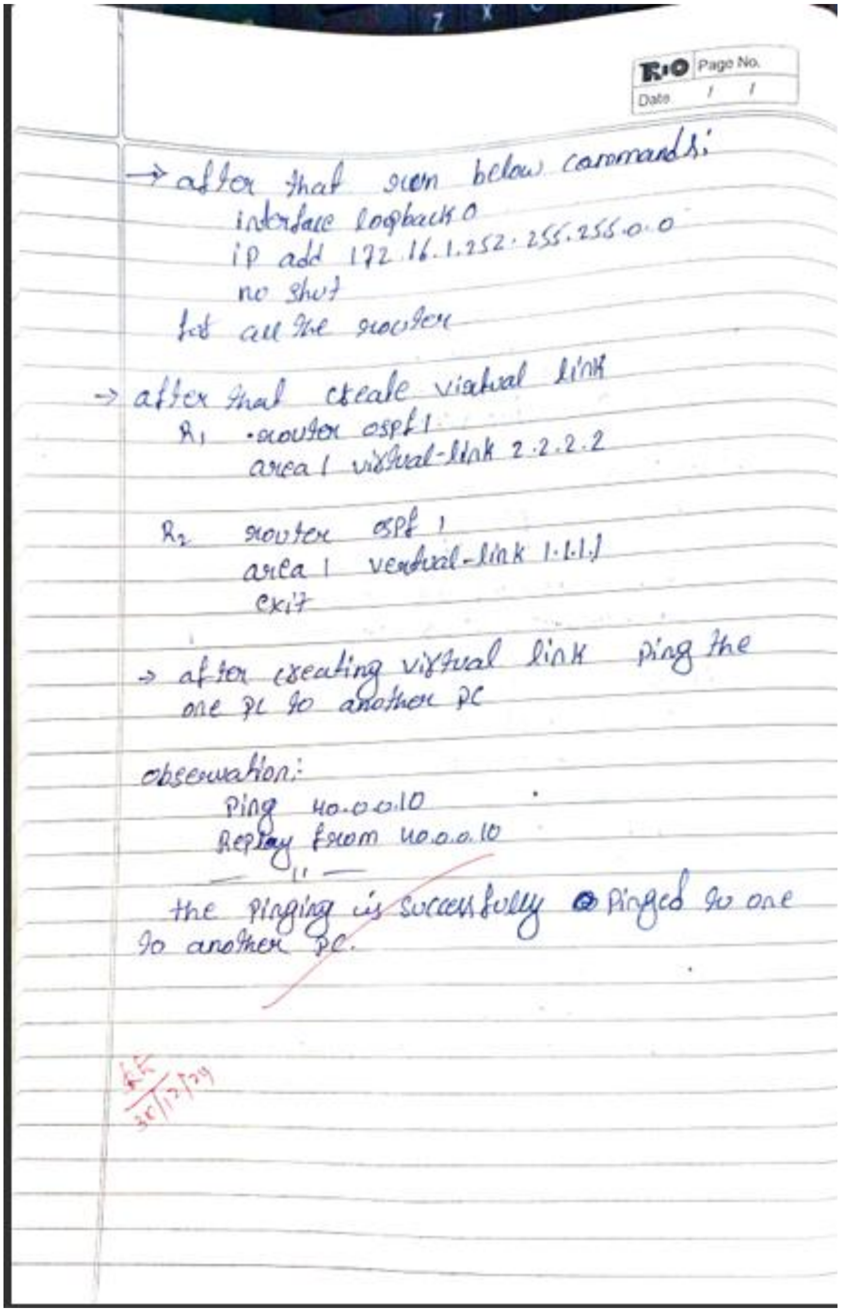
```
router ospf 1
router-id 1.1.1.1
network 10.0.0.0 0.255.255.255 area 3
network 20.0.0.0 0.255.255.255 area 1
```

in R2

```
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 R3

```
router ospf 1
router-id 3.3.3.3
network 30.0.0.0 0.255.255.255 area 0
network 40.0.0.0 0.255.255.255 area 2
exit
```



Page No.
Date / /

→ after that run below commands:

```
interface loopback 0
ip add 172.16.1.252 · 255.255.0.0
no shut
```

for all the router

→ after that create virtual link

```
R1  · router ospf 1
      area 1 virtual-link 2.2.2.2

R2    router ospf 1
      area 1 vertual-link 1.1.1.1
      exit
```

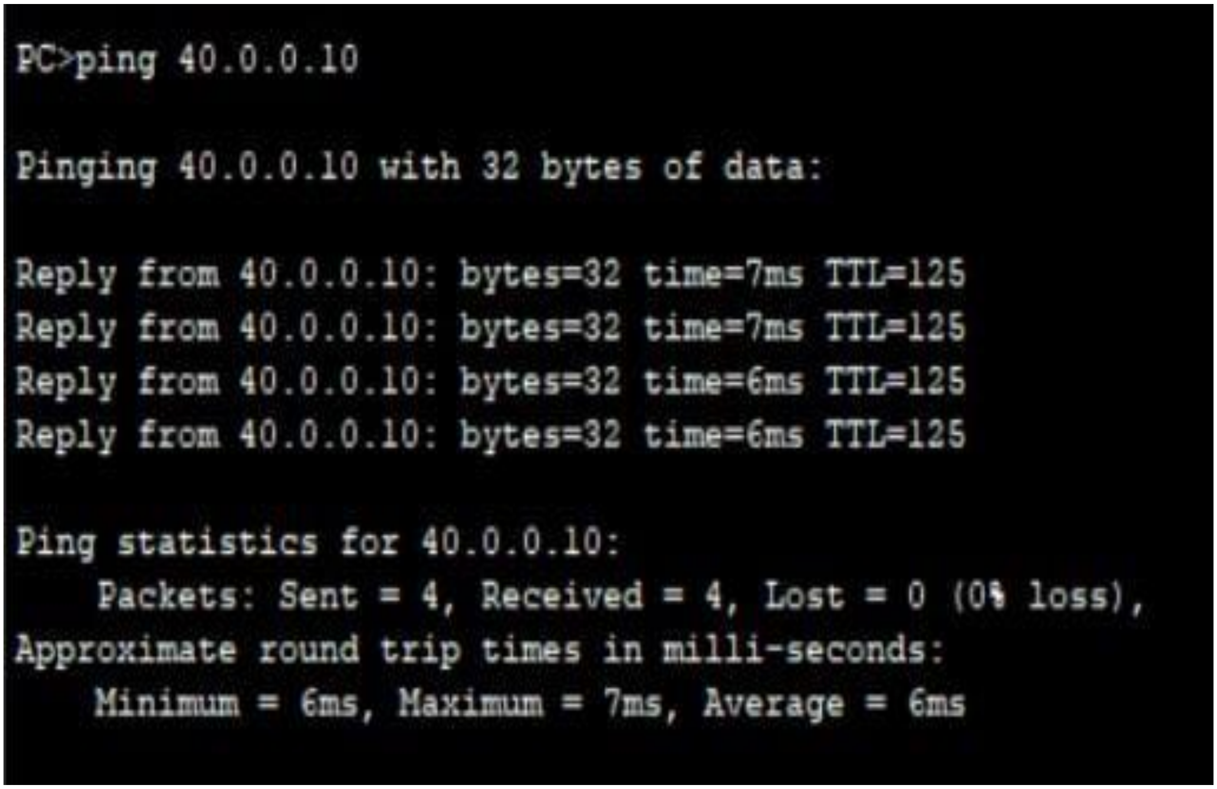
→ after creating virtual link ping the one pc to another pc

observation:

```
Ping 40.0.0.10
Replay from 40.0.0.10
— " —
```

the pinging is succesfully pinged to one to another pc.

30/12/19

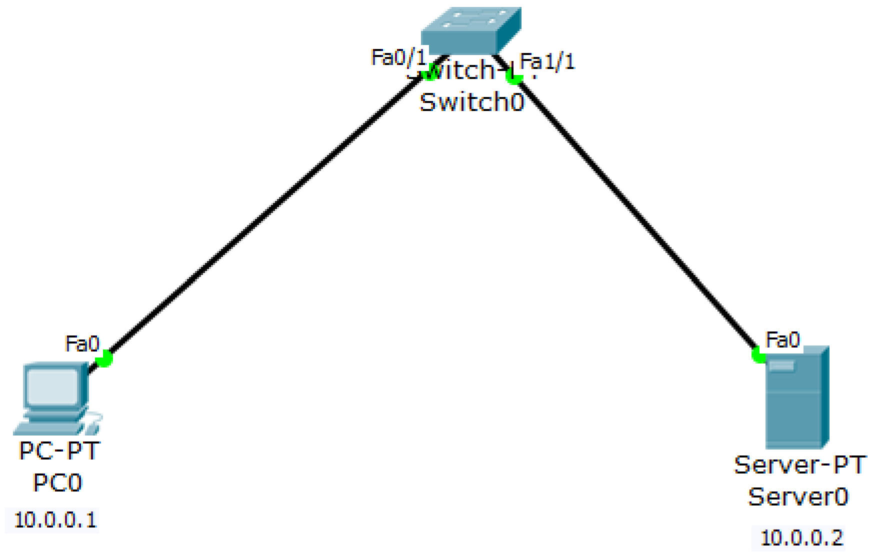


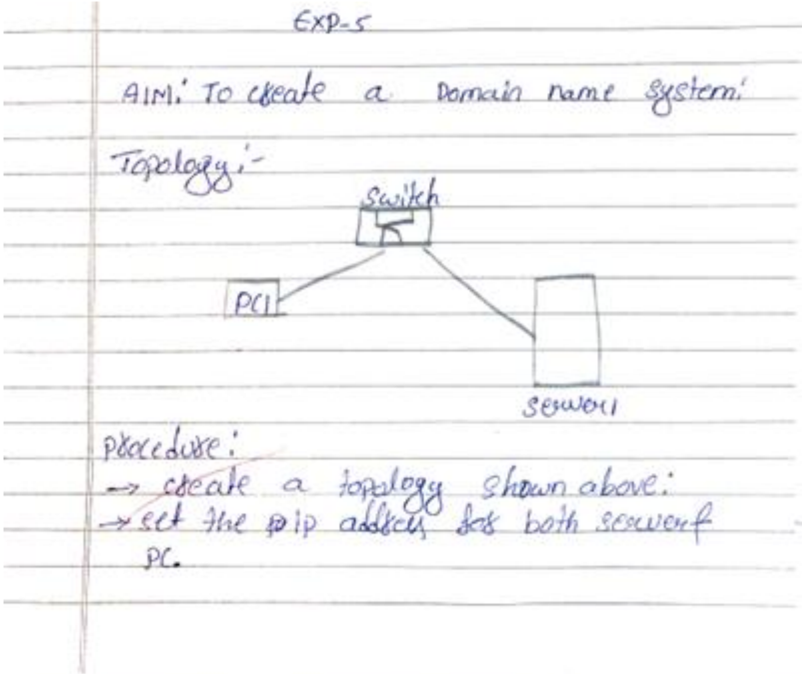
```
PC>ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data:

Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125

Ping statistics for 40.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 6ms, Maximum = 7ms, Average = 6ms
```

Program 8:

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

Topology:





EXP-5

AIM: To create a Domain name system:

Topology:-

Switch

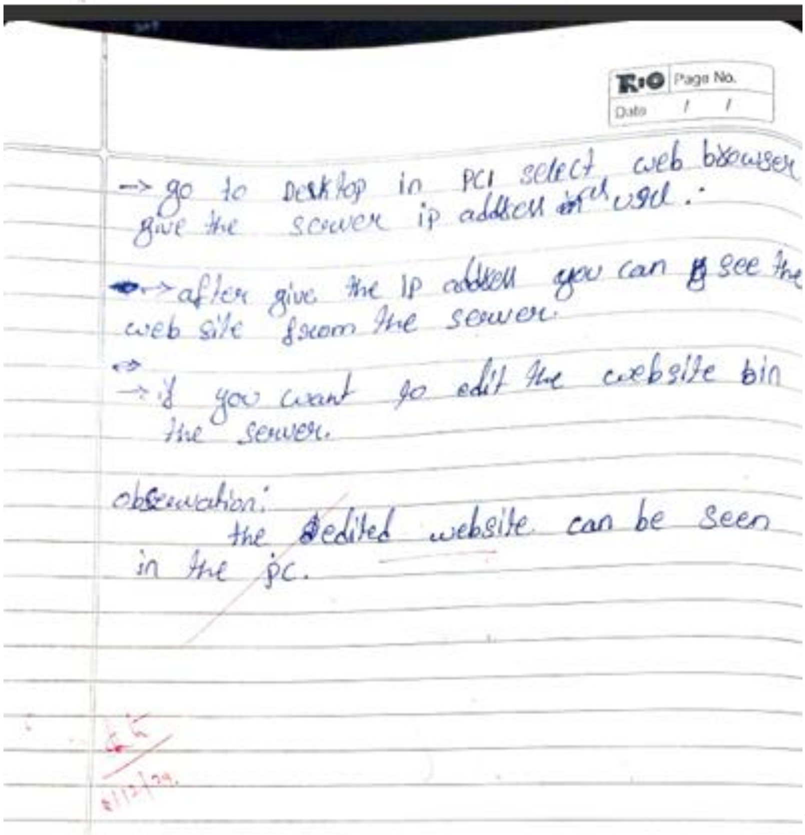
PC1

server1

procedure:

→ create a topology shown above:

→ set the ~~p~~ ip address for both server & PC.



R:O Page No. Date / /

→ go to Desktop in PC1 select web browser give the server ip address ~~in~~ url.:

→ after give the ip address you can ~~g~~ see the web site from the server.

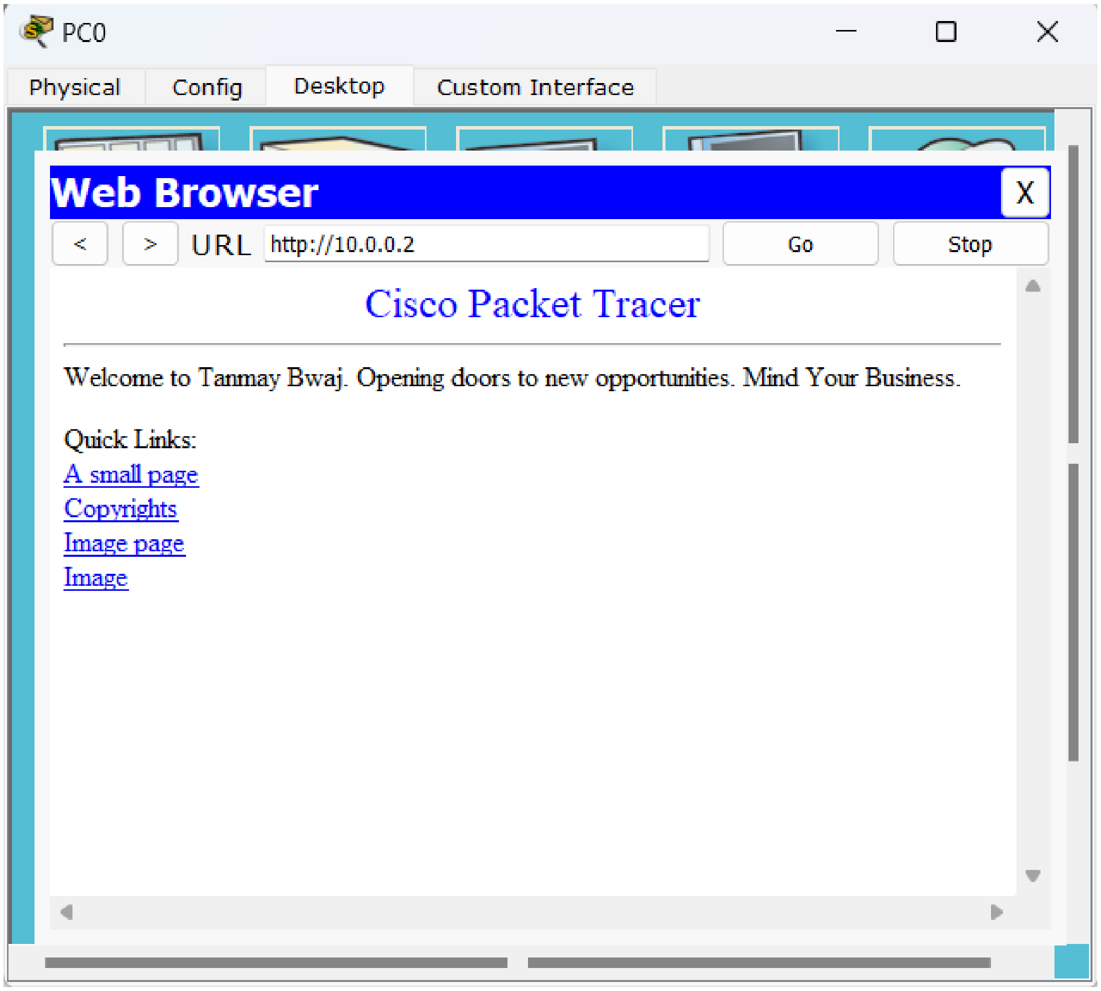
→ if you want to edit the website bin the server.

observation:

the ~~s~~edited website can be seen in the pc.

6/12/24

Procedure and Observations:



http://10.0.0.2

Cisco Packet Tracer

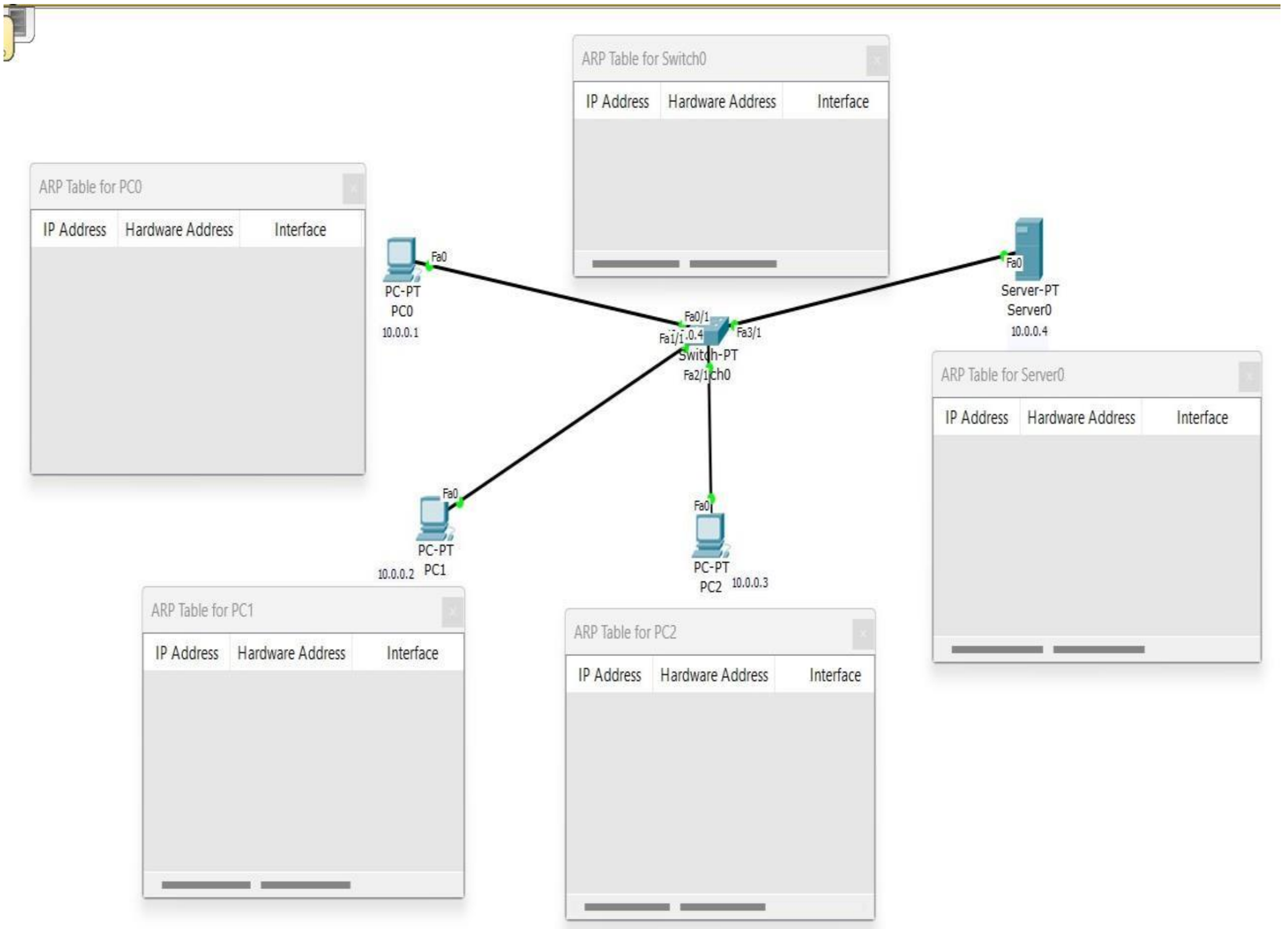
Welcome to Tanmay Bwaj. Opening doors to new opportunities. Mind Your Business.

Quick Links:
A small page
Copyrights
Image page
Image

Program 9:

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

Topology:



ARP Table for Switch0

IP Address	Hardware Address	Interface

ARP Table for PC0

IP Address	Hardware Address	Interface

ARP Table for Server0

IP Address	Hardware Address	Interface

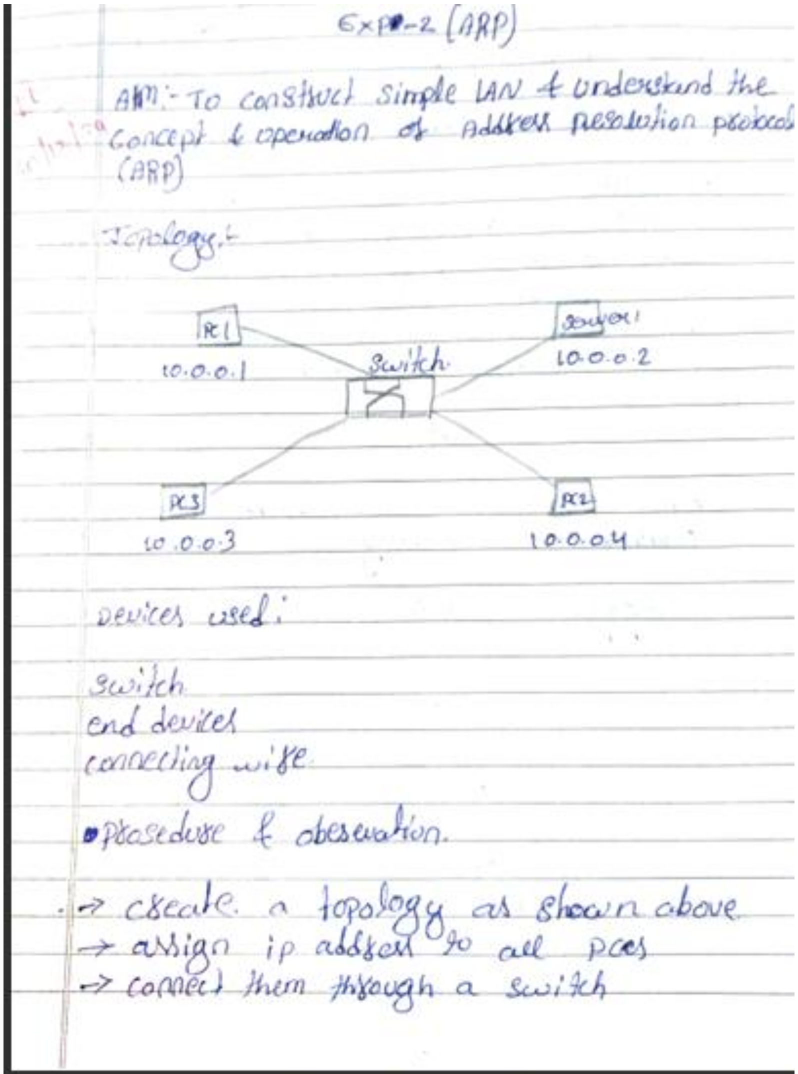
ARP Table for PC1

IP Address	Hardware Address	Interface

ARP Table for PC2

IP Address	Hardware Address	Interface

Procedure and Observations:



EXP-2 (ARP)

AIM:- To construct simple LAN & understand the concept & operation of Address resolution protocol (ARP)

Topology:-

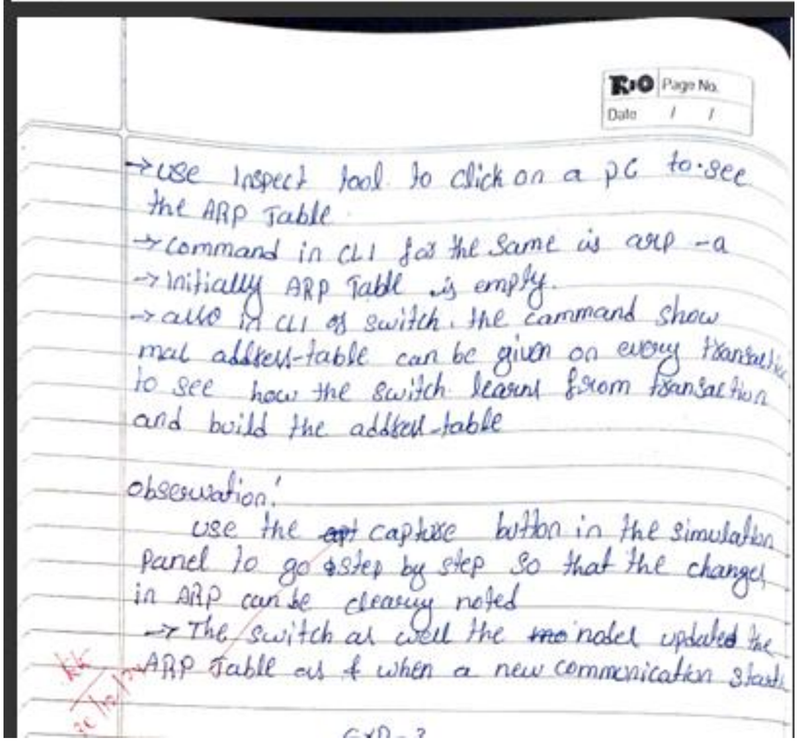
PC1
10.0.0.1
Switch
Server1
10.0.0.2
PC3
10.0.0.3
PC2
10.0.0.4

Devices used:

Switch
end devices
connecting wire

Procedure & obeservation.

→ create a topology as shown above
→ assign ip address to all pcs
→ connect them through a switch



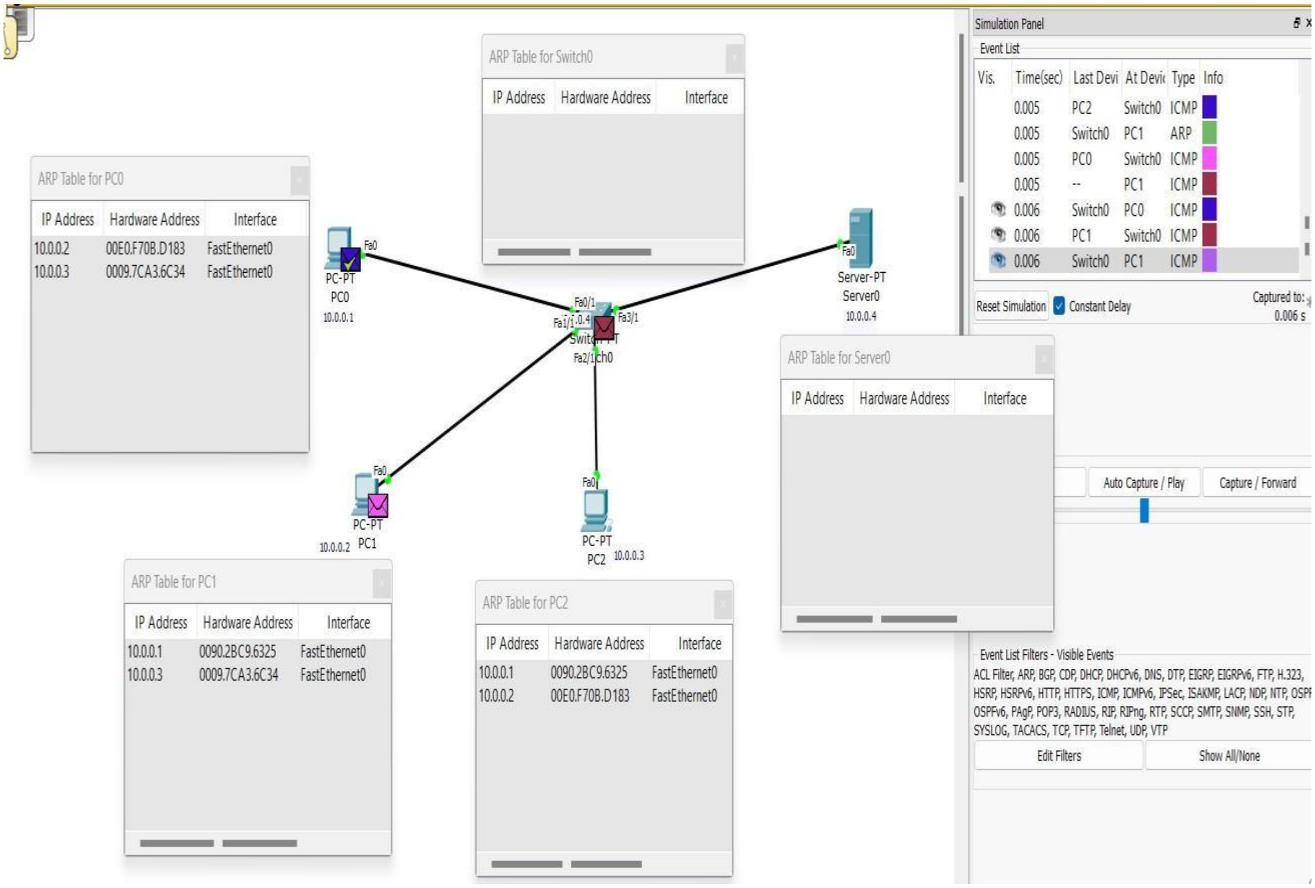
RIO Page No.
Date / /

→ use Inspect tool to click on a pc to see the ARP Table
→ command in CLI for the Same is arp -a
→ Initially ARP Table is empty.
→ also in CLI of switch. the command show mac address-table can be given on every transaction to see how the switch learns from transaction and build the address-table

observation:

use the ~~arp~~ capture button in the simulation panel to go step by step so that the changes in ARP can be clearly noted
→ The switch as well the ~~mo~~ nodes updated the ARP Table as & when a new communication starts

EXP-3



ARP Table for Switch0

IP Address	Hardware Address	Interface

ARP Table for PC0

IP Address	Hardware Address	Interface
10.0.0.2	00E0.F70B.D183	FastEthernet0
10.0.0.3	0009.7CA3.6C34	FastEthernet0

ARP Table for Server0

IP Address	Hardware Address	Interface

ARP Table for PC1

IP Address	Hardware Address	Interface
10.0.0.1	0090.2BC9.6325	FastEthernet0
10.0.0.3	0009.7CA3.6C34	FastEthernet0

ARP Table for PC2

IP Address	Hardware Address	Interface
10.0.0.1	0090.2BC9.6325	FastEthernet0
10.0.0.2	00E0.F70B.D183	FastEthernet0

Simulation Panel

Event List

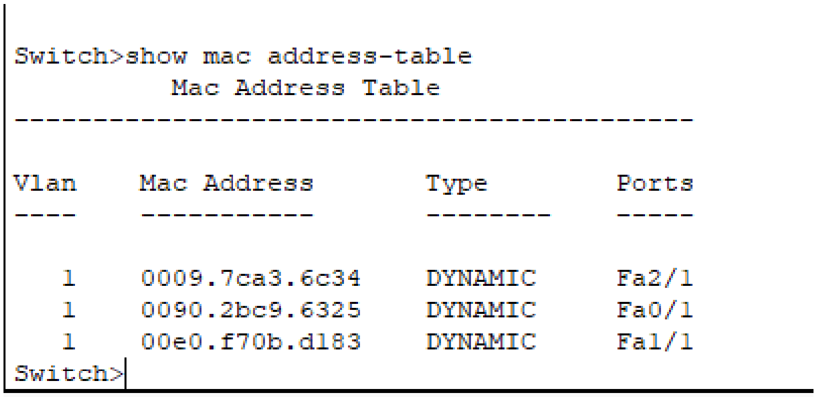
Vis.	Time(sec)	Last Devi	At Devic	Type	Info
	0.005	PC2	Switch0	ICMP	🟦
	0.005	Switch0	PC1	ARP	🟩
	0.005	PC0	Switch0	ICMP	🟪
	0.005	--	PC1	ICMP	🟥
👁	0.006	Switch0	PC0	ICMP	🟦
👁	0.006	PC1	Switch0	ICMP	🟥
👁	0.006	Switch0	PC1	ICMP	🟪

Reset Simulation ☑ Constant Delay

Captured to:
0.006 s

Event List Filters - Visible Events
ACL Filter, ARP, BGP, CDP, DHCP, DHCPv6, DNS, DTP, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, LACP, NDP, NTP, OSPF
OSPFv6, PAgP, POP3, RADIUS, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, VTP

Edit Filters | Show All/None

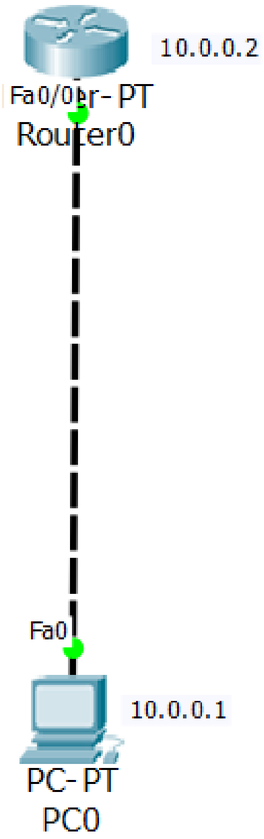


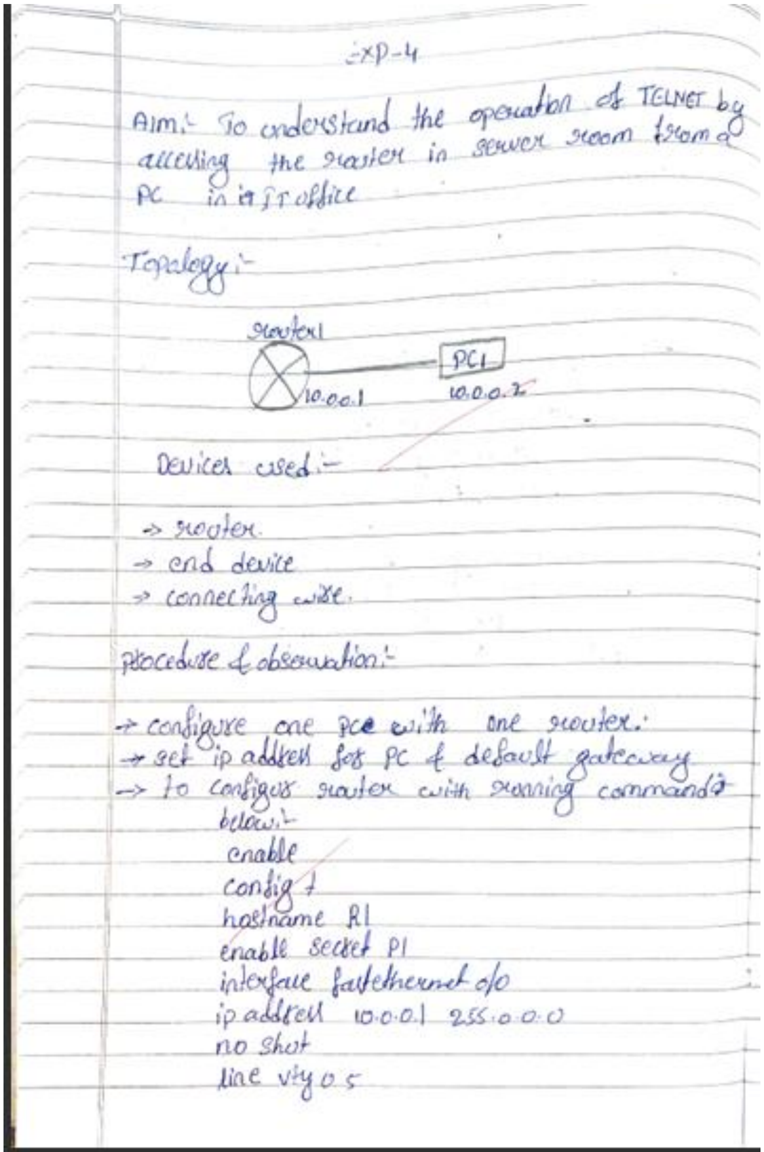
```
Switch>show mac address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    0009.7ca3.6c34    DYNAMIC     Fa2/1
   1    0090.2bc9.6325    DYNAMIC     Fa0/1
   1    00e0.f70b.d183    DYNAMIC     Fa1/1
Switch>
```

Program 10:

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

Topology :





EXP-4

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

Topology:-

router1

PC1

10.0.0.1

10.0.0.2

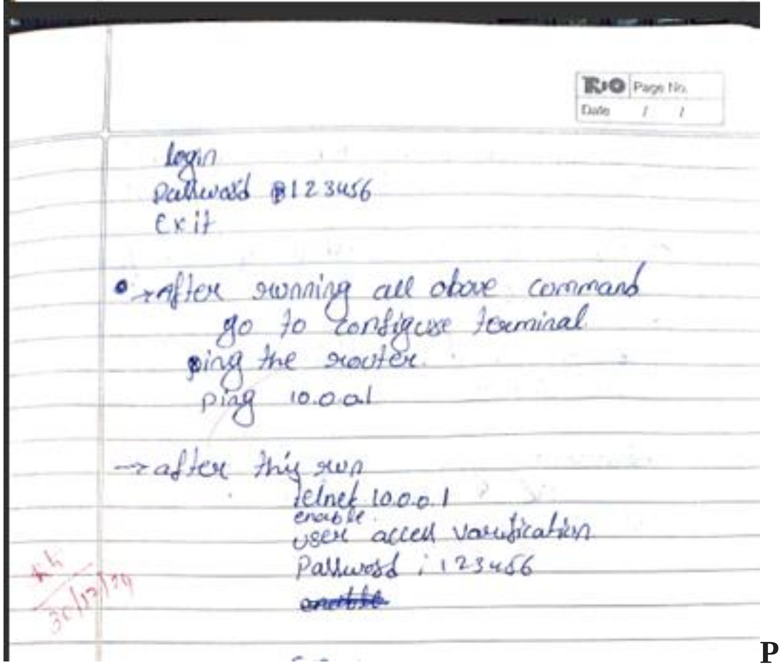
Devices used:-

-> router.
-> end device
-> connecting wire.

Procedure & observation:-

-> configure one pc with one router.
-> set ip address for PC & default gateway
-> to configure router with running commands below:-

```
enable
config t
hostname R1
enable secret P1
interface fastethernet 0/0
ip address 10.0.0.1 255.0.0.0
no shut
line vty 0 5
```



Page No.
Date / /

```
login
Password 123456
exit
```

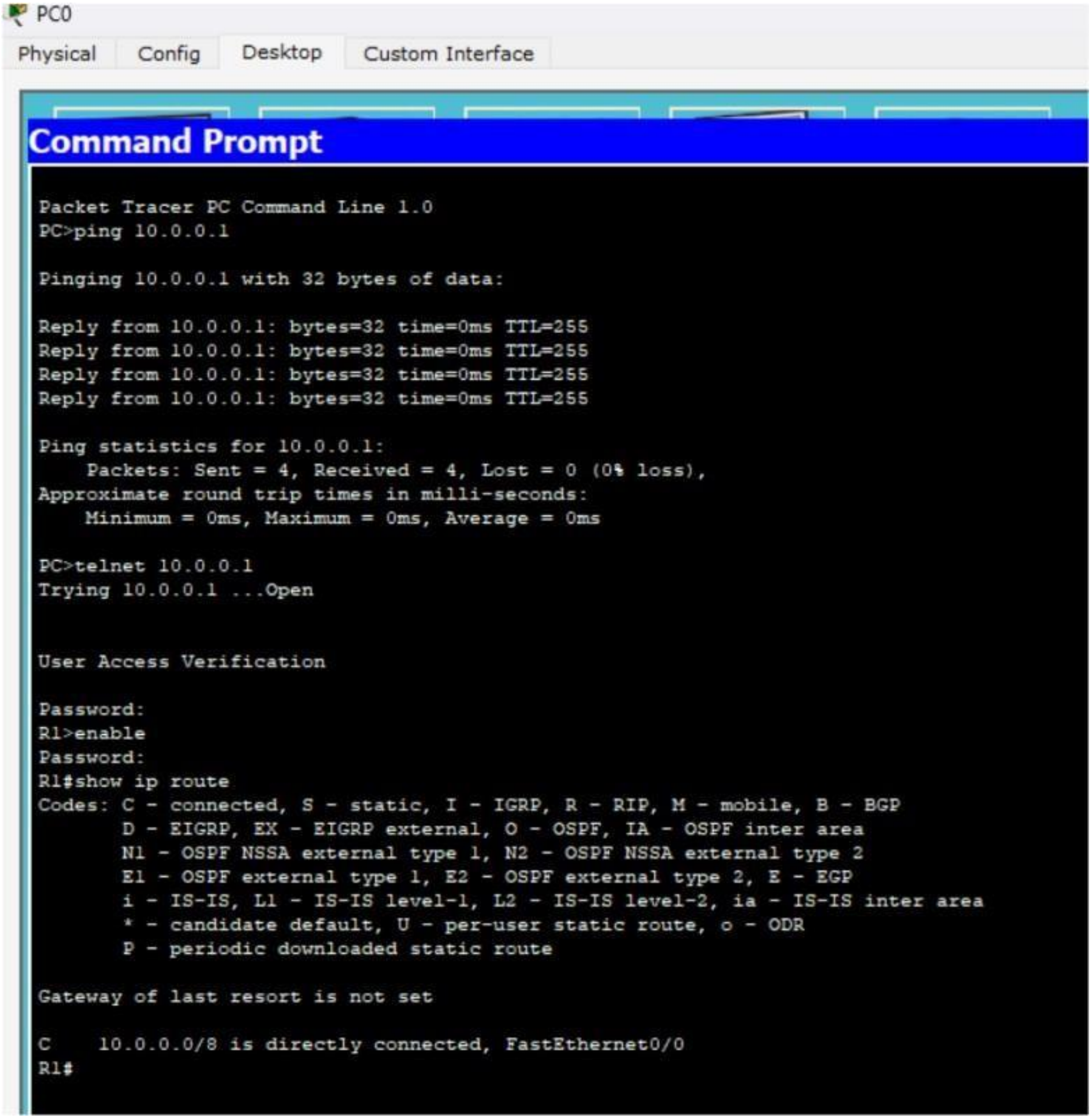
- after running all above command go to configure terminal ping the router.
  ping 10.0.0.1

-> after this run

```
telnet 10.0.0.1
enable
user access verification
Password : 123456
~~enable~~
```

30/12/24

Procedure and Observations:

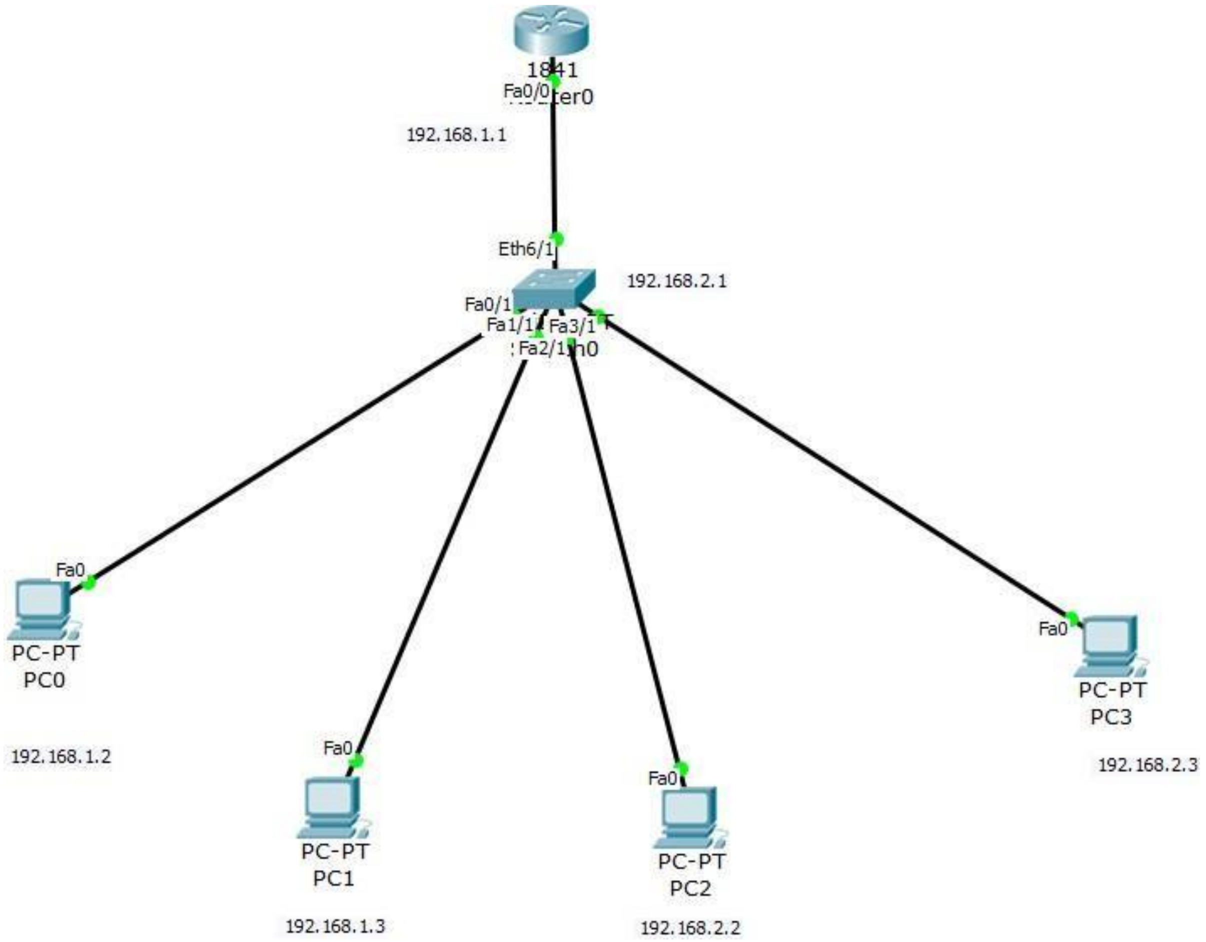


```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>telnet 10.0.0.1
Trying 10.0.0.1 ...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#
```

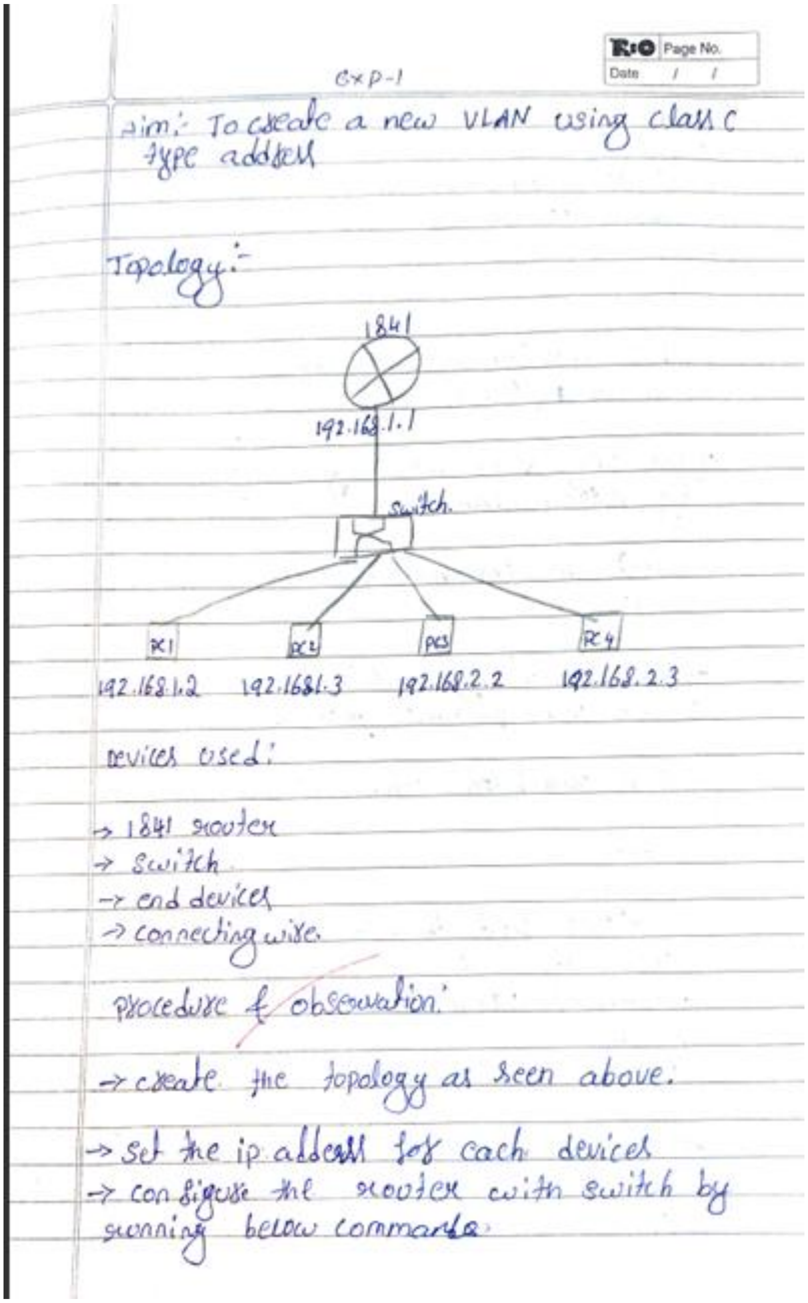
<u>Program 11</u>:

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

Topology:



Procedure and Observations:



Page No.

Date / /

exp-1

Aim:- To create a new VLAN using class C type address

Topology:-

1841
192.168.1.1
switch.
PC1
PC2
PC3
PC4
192.168.1.2
192.168.1.3
192.168.2.2
192.168.2.3

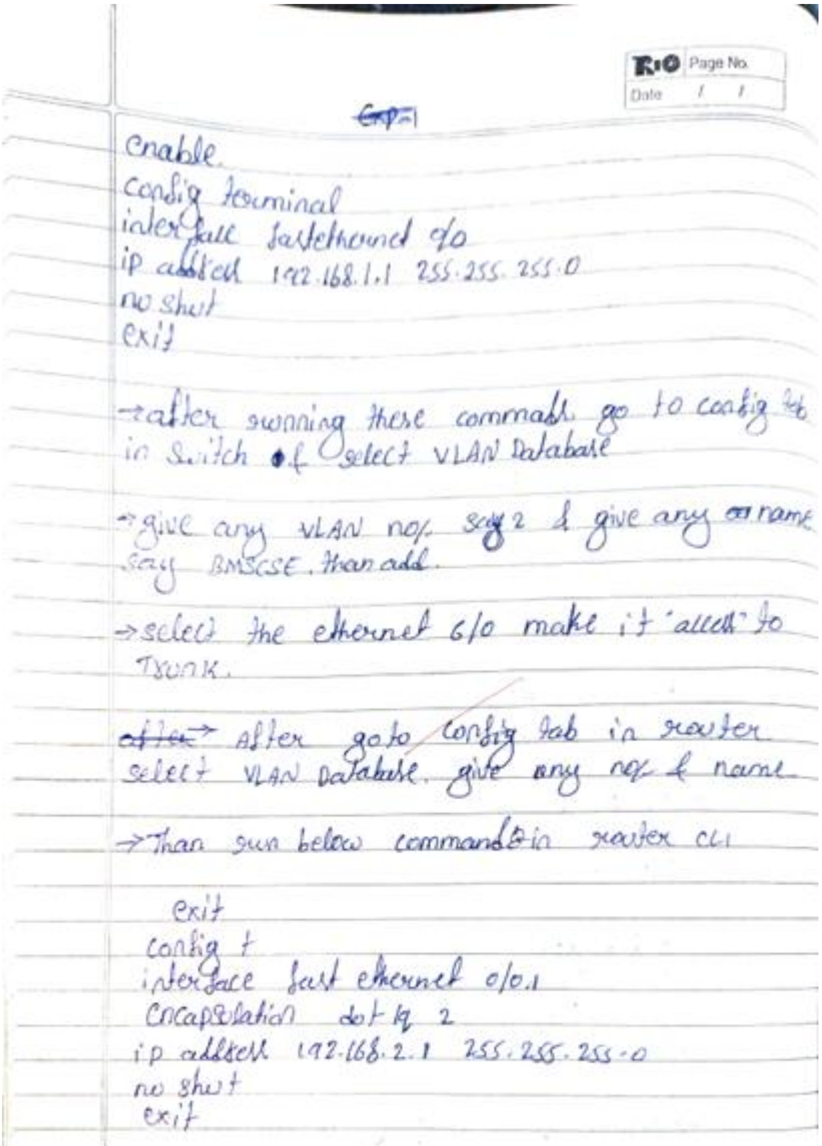
Devices used:

-> 1841 router
-> switch
-> end devices
-> connecting wire.

procedure & observation:

-> create the topology as seen above.

-> set the ip address for each devices
-> configure the router with switch by running below commands



Page No.
Date / /

~~Exp-1~~

```
enable
config terminal
interface fastethernet 0/0
ip address 192.168.1.1 255.255.255.0
no shut
exit
```

→after running these commands go to config tab in Switch ~~of~~ select VLAN Database

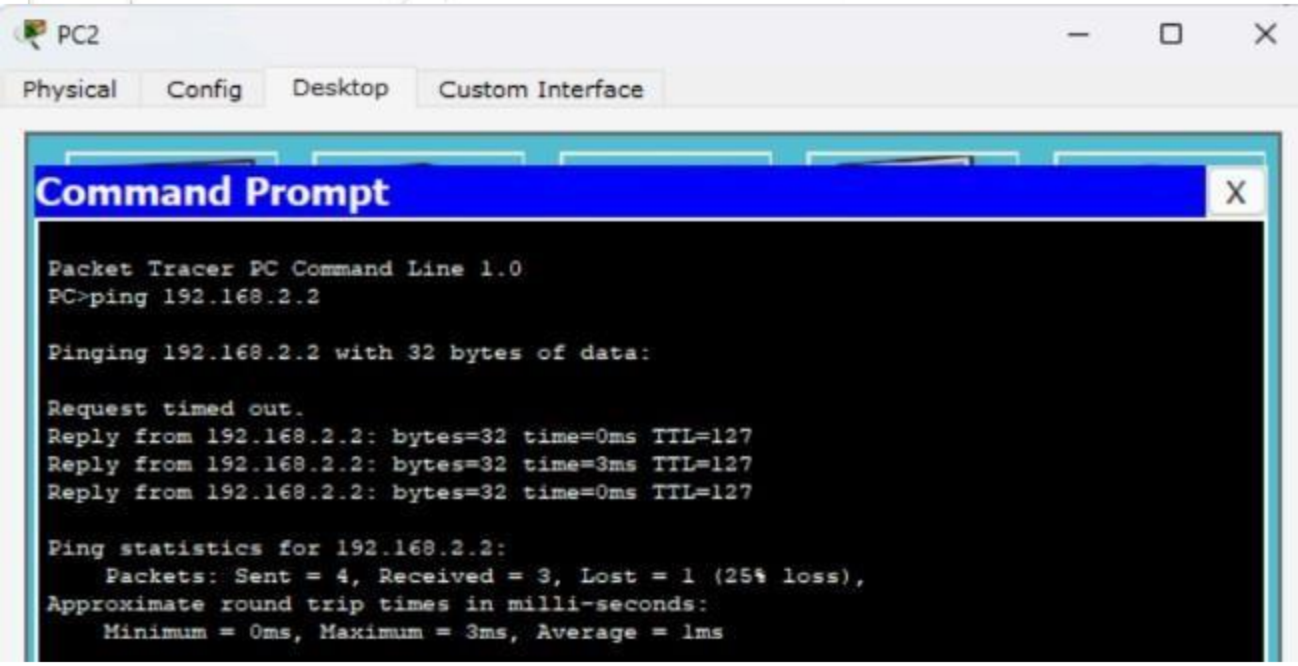
→give any VLAN no. say 2 & give any ~~or~~ name say BMSCSE. than add.

→select the ethernet 6/0 make it 'access' to Trunk.

~~after~~→ After goto config tab in router select VLAN Database. give any no. & name

→Than run below commands in router CLI

```
  exit
 config t
 interface fast ethernet 0/0.1
 encapsulation dot1q 2
ip address 192.168.2.1 255.255.255.0
no shut
 exit
```

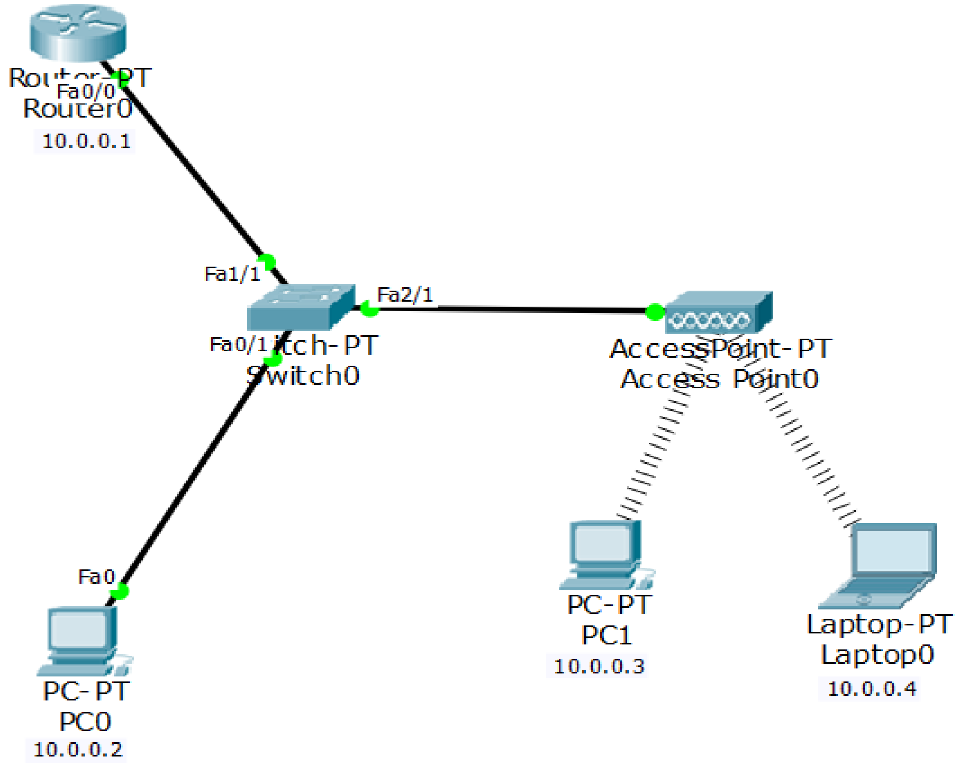


```
Packet Tracer PC Command Line 1.0
PC>ping 192.168.2.2

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=3ms 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 = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 3ms, Average = 1ms
```

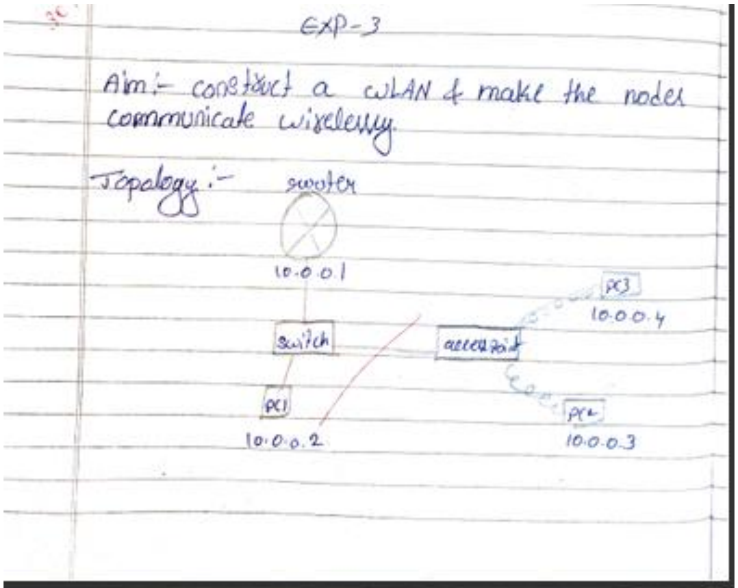
Program 12:

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

Topology:



Procedure and Observations:

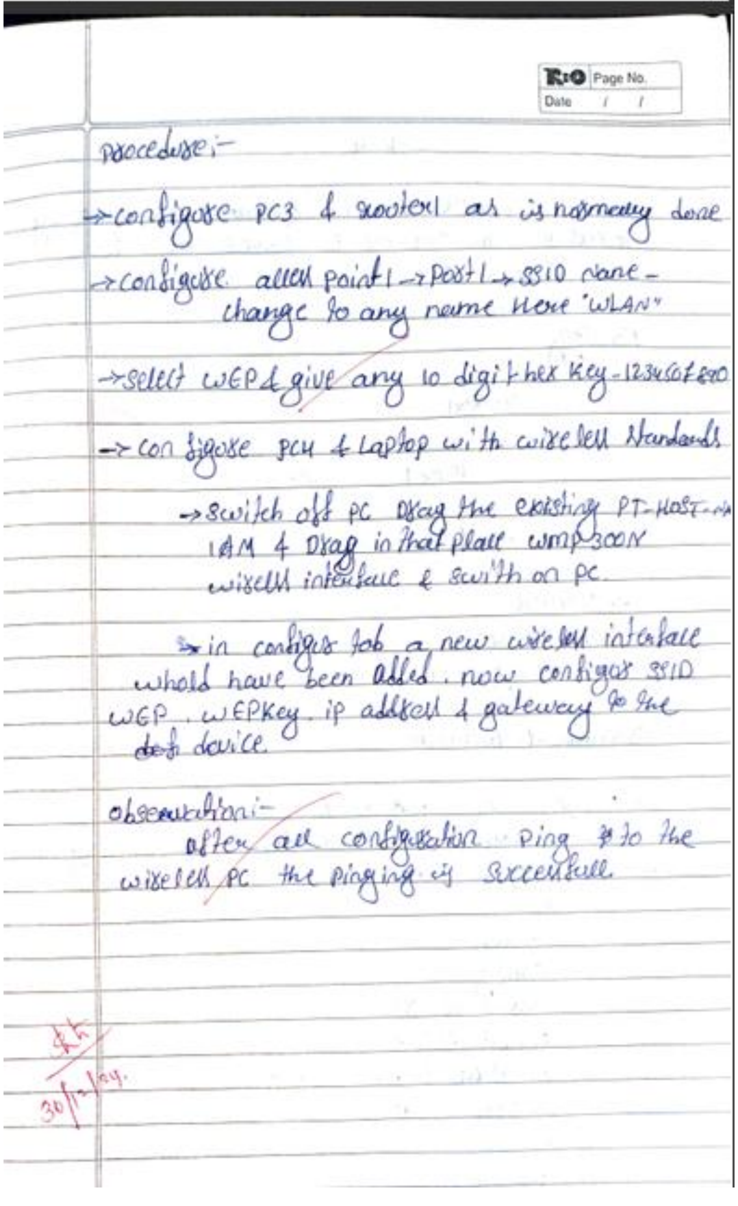


EXP-3

Aim:- construct a WLAN & make the nodes communicate wirelessly.

Topology:-

router
10.0.0.1
switch
PC1
10.0.0.2
access point
PC3
10.0.0.4
PC2
10.0.0.3



RiO Page No.
Date / /

procedure:-

→configure PC3 & router1 as is normally done

→configure access point1 → Port1 → SSID name -
change to any name Here "WLAN"

→select WEP & give any 10 digit hex key-1234567890

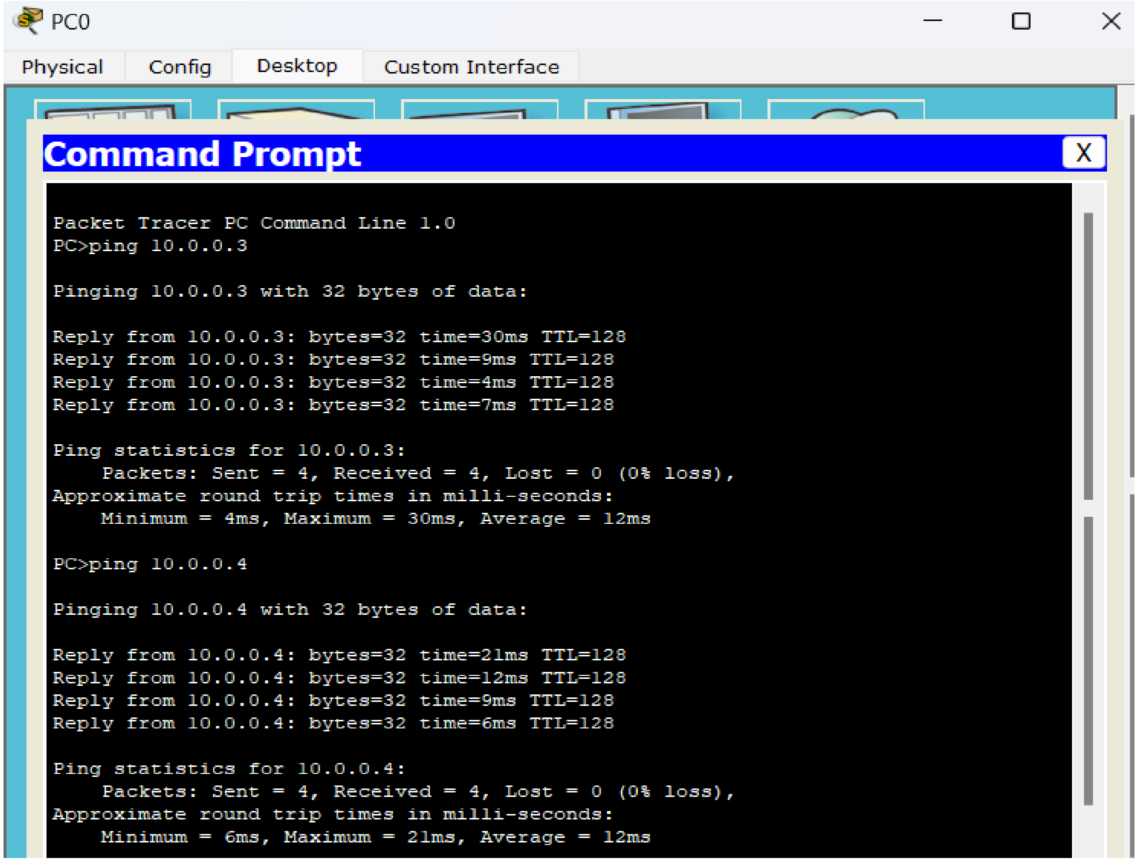
→configure PC4 & Laptop with wireless standards

→switch off PC Drag the existing PT-HOST-NM-1AM & Drag in that place WMP300N wireless interface & switch on PC.

→in config tab a new wireless interface whould have been added. now configure SSID WEP, WEPKey, ip address & gateway to the ~~def~~ device.

observation:-
after all configuration ping ~~#~~ to the wireless PC the pinging is successfull

30/12/24.



```
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=30ms TTL=128
Reply from 10.0.0.3: bytes=32 time=9ms TTL=128
Reply from 10.0.0.3: bytes=32 time=4ms 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 = 4ms, Maximum = 30ms, Average = 12ms

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=21ms TTL=128
Reply from 10.0.0.4: bytes=32 time=12ms TTL=128
Reply from 10.0.0.4: bytes=32 time=9ms TTL=128
Reply from 10.0.0.4: bytes=32 time=6ms 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 = 6ms, Maximum = 21ms, Average = 12ms
```

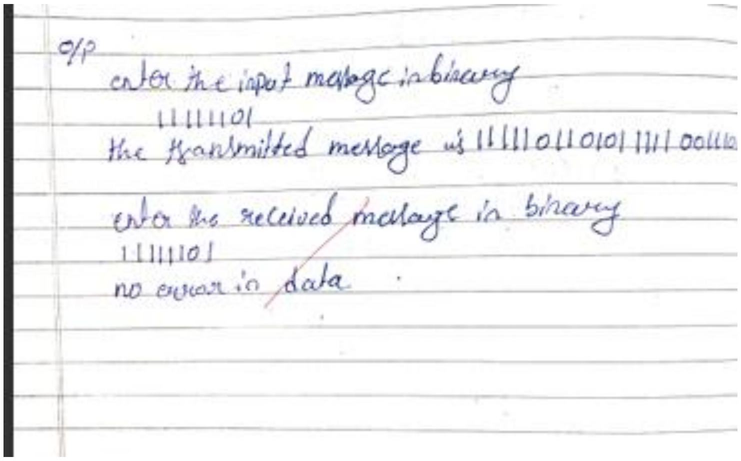
CYCLE - 2

Program 13:

Aim: 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");
}
/* Perform XOR on the msg with the selected polynomial */
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';
}}}
/* check for errors. return 0 if error detected */
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];
     /* x 16 + x12 + x5 + 1 */
      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 occurred" << endl;
     return 0;
}
```

Observations:



o/p
enter the input message in binary
11111101
the transmitted message is 11111 0110101 1111 00110
enter the received message in binary
11111101
no error in data

Program 14:

Aim: Write a program for congestion control using Leaky bucket algorithm.

Algorithm:

```
1. Start
2. Set the bucket size or the buffer size.
3. Set the output rate.
4. Transmit the packets such that there is no overflow.
5. Repeat the process of transmission until all packets are transmitted.
       (Reject packets whosesize is greater than the bucket size.)
6. Stop
```

Code:

```
#include <iostream>
#include <string.h>
using namespace std;

#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#define NOF_PACKETS 10
int rand(int a){
      int rn = (random() % 10) % a;
      return rn == 0 ? 1 : rn;
}
int main() {
    int packet_sz[NOF_PACKETS], i, clk, b_size, o_rate, p_sz_rm=0, p_sz, p_time, op;
    for(i = 0; i<NOF_PACKETS; ++i)
          packet_sz[i] = rand(6) * 10;
    for(i = 0; i<NOF_PACKETS; ++i)
         printf("\npacket[%d]:%d bytes\t", i, packet_sz[i]);
    printf("\nEnter the Output rate:");
    scanf("%d", &o_rate);
    printf("Enter the Bucket Size:");
    scanf("%d", &b_size);
    for(i = 0; i<NOF_PACKETS; ++i){
         if( (packet_sz[i] + p_sz_rm) > b_size)
              if(packet_sz[i] > b_size)/*compare the packet size with bucket size*/
                    printf("\n\nIncoming packet size (%dbytes) is Greater than bucket capacity
                                   (%dbytes)-PACKET REJECTED", packet_sz[i], b_size);
              else
                     printf("\n\nBucket capacity exceeded-PACKETS REJECTED!!");
        else {
             p_sz_rm += packet_sz[i];
             printf("\n\nIncoming Packet size: %d", packet_sz[i]);
             printf("\nBytes remaining to Transmit: %d", p_sz_rm);
             p_time = rand(4) * 10;
             printf("\nTime left for transmission: %d units", p_time);
            for(clk = 10; clk <= p_time; clk += 10) {
                 sleep(1);
                  if(p_sz_rm) {
                        if(p_sz_rm <= o_rate)/*packet size remaining comparing with output rate*/
                             op = p_sz_rm, p_sz_rm = 0;
                        else
                             op = o_rate, p_sz_rm -= o_rate;
                         printf("\nPacket of size %d Transmitted", op);
                         printf(" --- Bytes Remaining to Transmit: %d", p_sz_rm);
                  }
                  else {
                     printf("\nTime left for transmission: %d units", p_time-clk);
                     printf("\nNo packets to transmit!!");
}}}}
return 0;
}
```

OUTPUT:

```
packet[0]:30 bytes
packet[1]:10 bytes
packet[2]:10 bytes
packet[3]:50 bytes
packet[4]:30 bytes
packet[5]:50 bytes
packet[6]:10 bytes
packet[7]:20 bytes
packet[8]:30 bytes
packet[9]:10 bytes
Enter the Output rate:100
Enter the Bucket Size:50
Incoming Packet size: 30
Bytes remaining to Transmit: 30
Time left for transmission: 20 units
Packet of size 30 Transmitted --- Bytes Remaining to Transmit: 0
Time left for transmission: 0 units
No packets to transmit!!

Incoming Packet size: 10
Bytes remaining to Transmit: 10
Time left for transmission: 30 units
Packet of size 10 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!!
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!!

Incoming Packet size: 50
```

```
Bytes remaining to Transmit: 50
Time left for transmission: 20 units
Packet of size 50 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: 10 units
Packet of size 10 Transmitted --- Bytes Remaining to Transmit: 0
Incoming Packet size: 20
Bytes remaining to Transmit: 20
Time left for transmission: 20 units
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 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: 20 units
Packet of size 10 Transmitted --- Bytes Remaining to Transmit: 0
Time left for transmission: 0 units
No packets to transmit!!
```

<u>Program 15:</u>

Aim: 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.

Algorithm:

Client Side

1. Start.
2. Create a socket using the socket() system call.
3. Connect the socket to the server's address using the connect() system call.
4. Send the filename of the required file using the send() system call.
5. Read the contents of the file sent by the server using the recv() system call.
6. Stop.

Code:

```
#include <unistd.h>
int main()
{
    int soc, n;
    char buffer[1024], fname[50];
    struct sockaddr_in addr;
    /* socket creates an endpoint for communication and returns a file descriptor */
    soc = socket(PF_INET, SOCK_STREAM, 0);
    /*
    * sockaddr_in is used for ip manipulation
    * we define the port and IP for the connection.
    */
    addr.sin_family = AF_INET;
    addr.sin_port = htons(7891);
    addr.sin_addr.s_addr = inet_addr("127.0.0.1");
    /* keep trying to establish connection with server */
    while(connect(soc, (struct sockaddr *) &addr, sizeof(addr))) ;
        printf("\nClient is connected to Server");
    printf("\nEnter file name: ");
    scanf("%s", fname);
     /* send the filename to the server */
    send(soc, fname, sizeof(fname), 0);
    printf("\nRecieved response\n");0
     /* keep printing any data received from the server */
     while ((n = recv(soc, buffer, sizeof(buffer), 0)) > 0)
         printf("%s", buffer);
     return 0;
}
```

Algorithm:

Server Side

1. Start.
2. Create a socket using socket() system call.
3. Bind the socket to an address using bind() system call.
4. Listen to the connection using listen() system call.
5. accept connection using accept()
6. Receive filename and transfer contents of file with client.
7. Stop.

Code:

```
#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("\nServer is Online");
    /* listen for connections from the socket */
    listen(welcome, 5);
    /* accept a connection, we get a file descriptor */
   new_soc = accept(welcome, NULL, NULL);
   /* receive the filename */
   recv(new_soc, fname, 50, 0);
   printf("\nRequesting for file: %s\n", fname);
   /* open the file and send its contents */
   fd = open(fname, O_RDONLY);
   if (fd < 0)
       send(new_soc, "\nFile not found\n", 15, 0);
   else
      while ((n = read(fd, buffer, sizeof(buffer))) > 0)
  send(new_soc, buffer, n, 0);
  printf("\nRequest sent\n");
  close(fd);
  return 0;
}
```

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 16:

Aim: 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.

Code:

```
// server program for udp connection
#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
// Driver code
int main()
{
    char buffer[100];
    char *message = "Hello Client";
    int listenfd, len;
    struct sockaddr_in servaddr, cliaddr;
    bzero(&servaddr, sizeof(servaddr));
    // Create a UDP Socket
    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 server address to socket descriptor
    bind(listenfd, (struct sockaddr*)&servaddr, sizeof(servaddr));
    //receive the datagram
    len = sizeof(cliaddr);
    int n = recvfrom(listenfd, buffer, sizeof(buffer), 0, (struct sockaddr*)&cliaddr,&len);
    //receive message from server
    buffer[n] = '\0';
    puts(buffer);
    // send the response
    sendto(listenfd, message, MAXLINE, 0,(struct sockaddr*)&cliaddr, sizeof(cliaddr));
}


// udp 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
// Driver code
int main()
{
    char buffer[100];
    char *message = "Hello Server";
    int sockfd, n;
    struct sockaddr_in servaddr;
    // clear 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;
    // create datagram socket
    sockfd = socket(AF_INET, SOCK_DGRAM, 0);
    // connect to server
   if(connect(sockfd, (struct sockaddr *)&servaddr, sizeof(servaddr)) < 0) {
        printf("\n Error : Connect Failed \n");
        exit(0);
    }
    // request to send datagram
    // no need to specify server address in sendto
    // connect stores the peers IP and port
    sendto(sockfd, message, MAXLINE, 0, (struct sockaddr*)NULL, sizeof(servaddr));
    // waiting for response
    recvfrom(sockfd, buffer, sizeof(buffer), 0, (struct sockaddr*)NULL, NULL);
    puts(buffer);
    // close the descriptor
    close(sockfd);
}
```

Output:

```
//Server output
Server is Online.
Hello Server

//Client Output
Hello Client
```