VISVESVARAYATECHNOLOGICALUNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB RECORD

Computer Network Lab (23CS5PCCON)

Submitted by

Shubham Maloo (1BM22CS343)

in partial fulfillment for the award of the degree of

BACHELOROFENGINEERING
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 **Shubham Maloo (1BM22CS343)** 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.

Spoorthi D M Assistant Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE

Index

16	18-12-24	Using TCP/IP sockets, write a client-server program to make the client send the file name and the server to send back the contents of the requested file if present.	51-52
17	18-12-24	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.	53-54

Github Link: https://github.com/ShubhamMaloo00/1BM22CS343_ShubhamMaloo_CNLab

Program 1

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

Topology , Procedure and Observation:

2

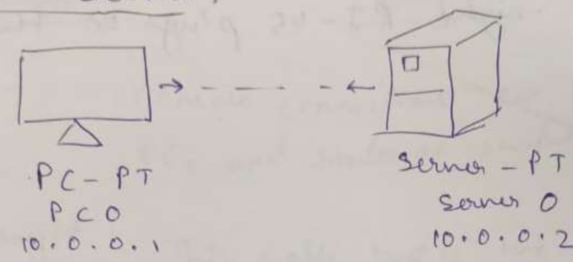
Connections/Links:

Cabel Type	Description
• Console :	Console connections can be made between PCs and routers or switches.
• Copper-Straight through :	This cable type is the standard ethernet media for connecting between devices that operate at different OSI layers.
• Copper Cross Over :	This cable type is the ethernet media for connecting between devices that operate at the same OSI layer.
• Fiber :	Fiber media is used to make connections between fiber parts.
• Phone :	Phone link connections can only be made between devices with modem parts.
• Coaxial :	It is used to make connections between coaxial parts.
• Serial DCE and DTE :	Serial connections, ~~often~~ after used for WLAN links, must be connected between serial parts.

• Octal

25/9/2 Experiment - 1 (4)

1.) PC to Server:

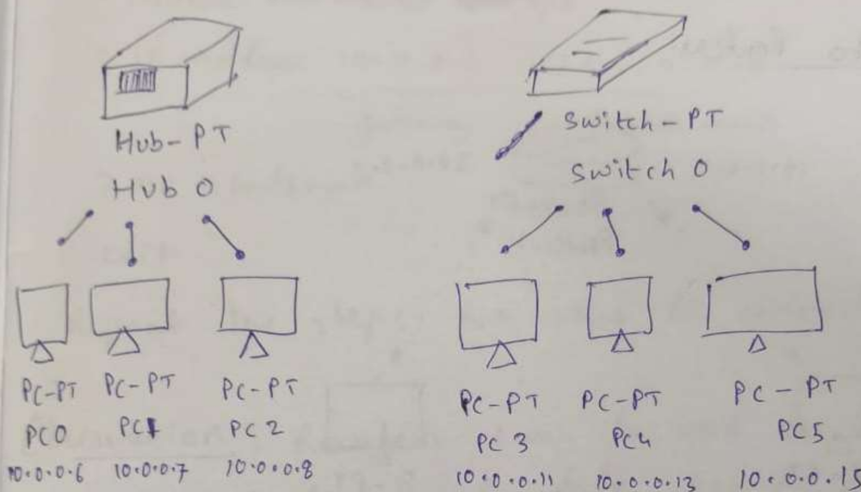


Aim: To set up a point-to-point network between a PC and a Server, facilitating direct communication to observe data exchange.

Topology: A PC is connected to server using a crossover ethernet cable.
IP address of PC - 10.0.0.1, Server - 10.0.0.2

Observation: Direct connection allows PC to communicate with server, which is typical in small networks for tasks such as file sharing. Service requests or testing server responses to client queries.

2) Hub to Switch :

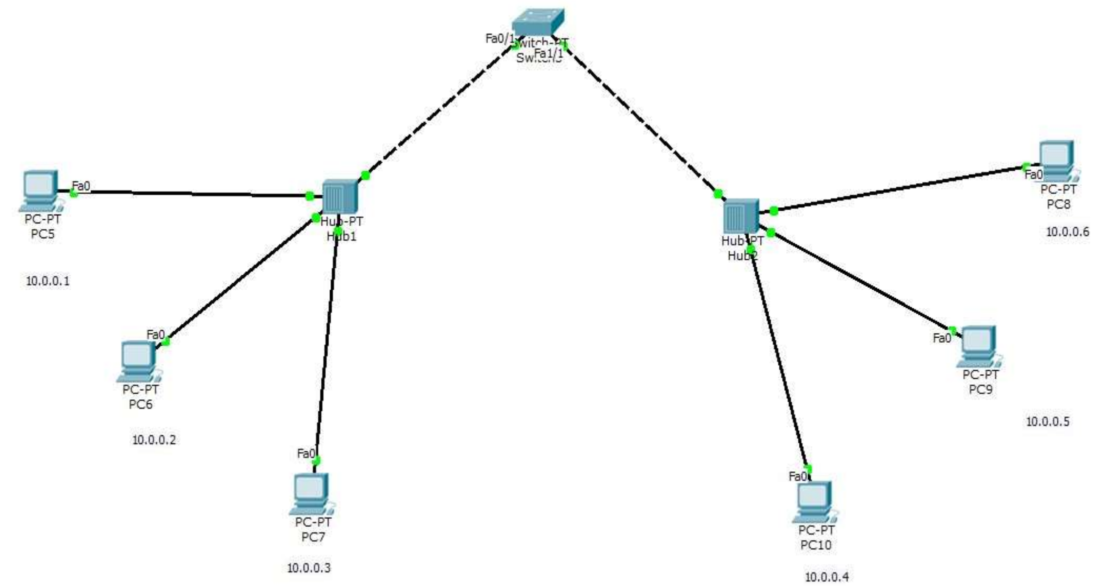


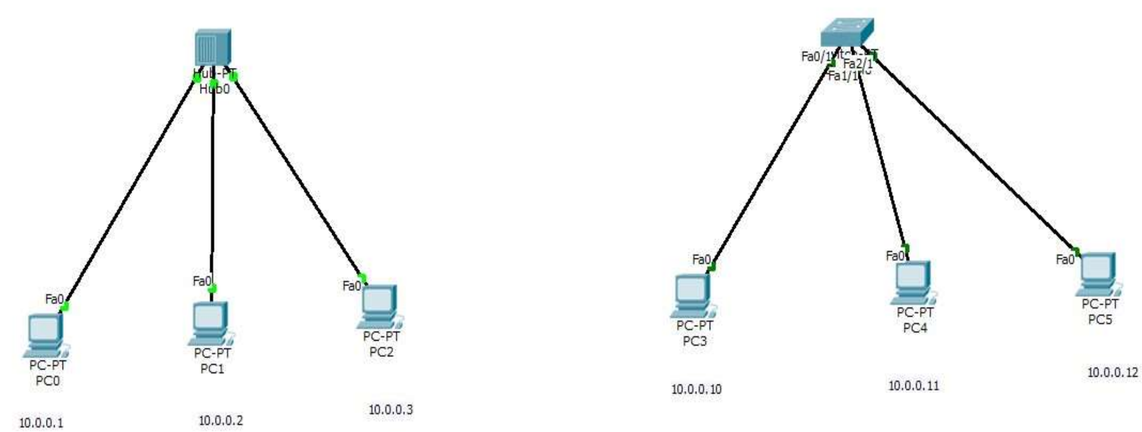
Aim: To create a simple network consisting of 3PCs connected to a Central Hub and another network with 3PCs connected to a switch. This connection will help observe the behaviour of data transmission using hub and switch devices.

Topology: 3 PCs are connected to a hub and switch using straight-through ethernet cables.

Observation: Hub broadcasts packets to all devices which may cause unnecessary traffic. Switch forwards packets only to appropriate device by learning MAC addresses making it more efficient in reducing traffic.

Screen Shots:

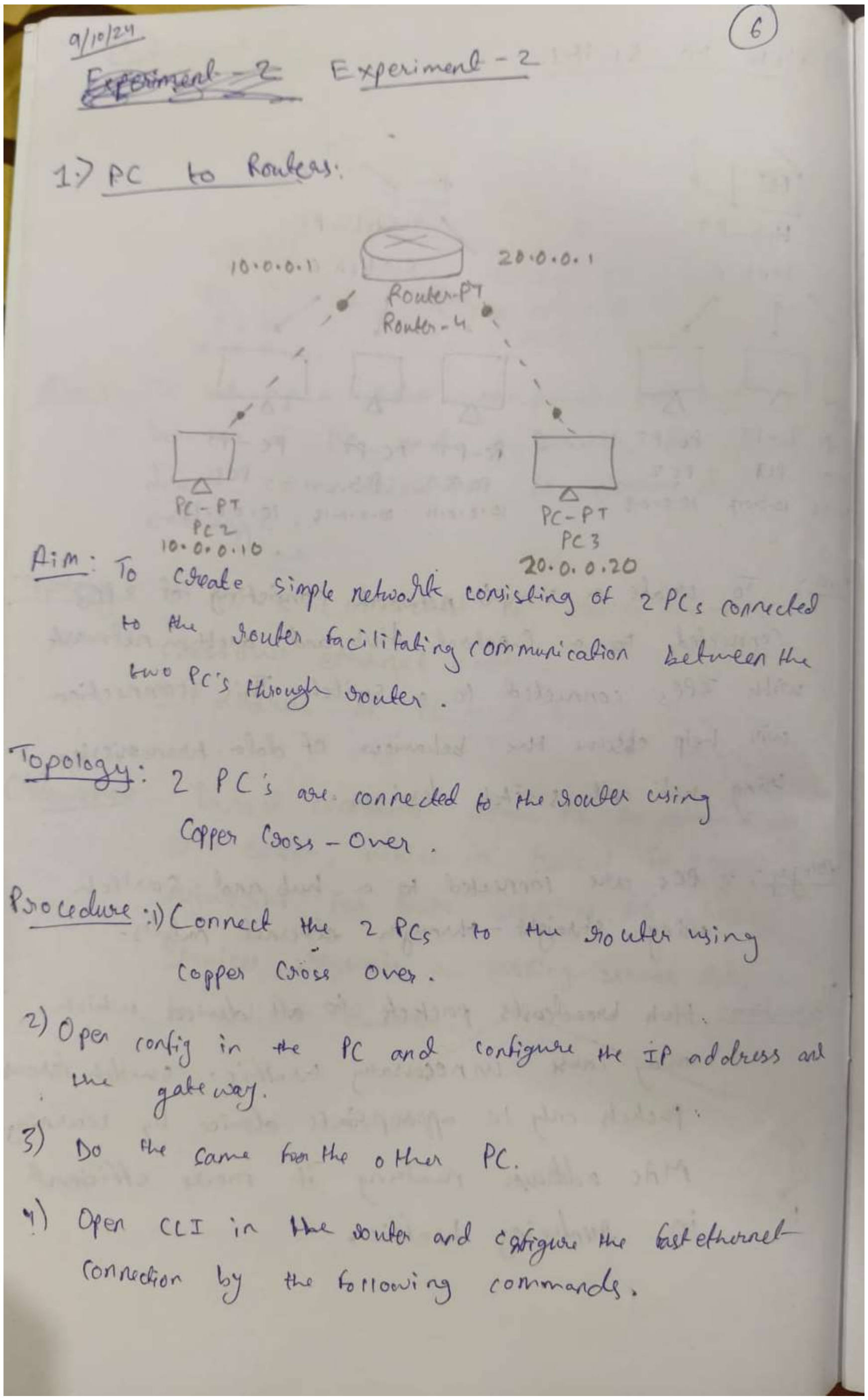




Program 2

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

Topology , Procedure and Observation:



9/10/24

6

~~Experiment - 2~~ Experiment - 2

1.) PC to Routers:

10.0.0.1

20.0.0.1

Router-PT

Router-4

PC-PT

PC2

10.0.0.10

PC-PT

PC3

20.0.0.20

Aim: To create simple network consisting of 2 PCs connected to the router facilitating communication between the two PC's through router.

Topology: 2 PC's are connected to the router using Copper Cross-Over.

Procedure :1) Connect the 2 PCs to the router using copper Cross Over.

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

3) Do the same for the other PC.

4) Open CLI in the router and configure the fast ethernet connection by the following commands.

> enable

> config terminal

> interface fastethernet 0/0

> ip address 10.0.0.1 255.0.0.0

$\underbrace{10.0.0.1}_{\text{gateway}}$ $\underbrace{255.0.0.0}_{\text{Subnet mask}}$

> no shutdown

exit.

Repeat the steps for other PC connection.

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

Output:

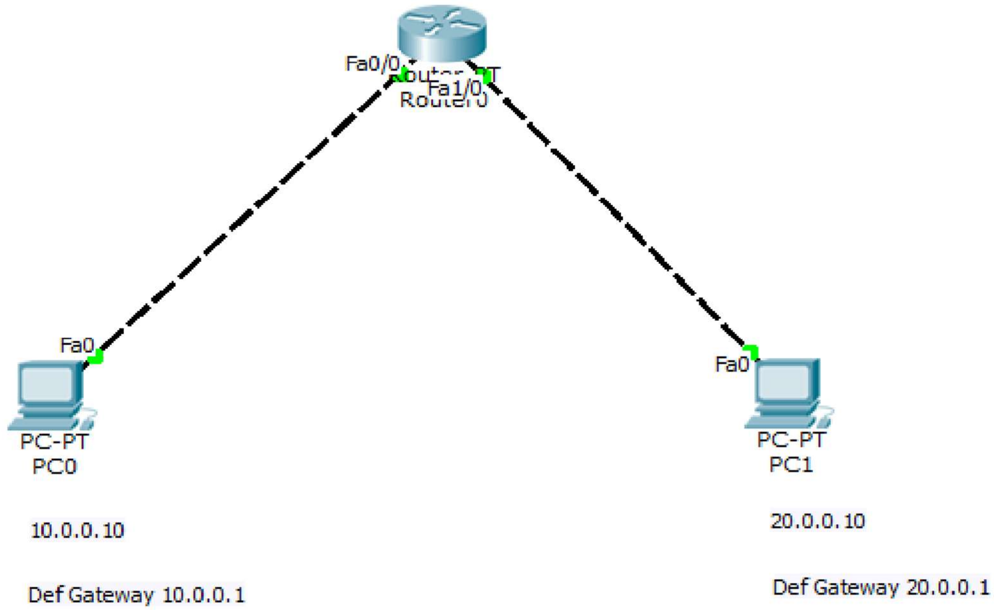
Router > show ip route

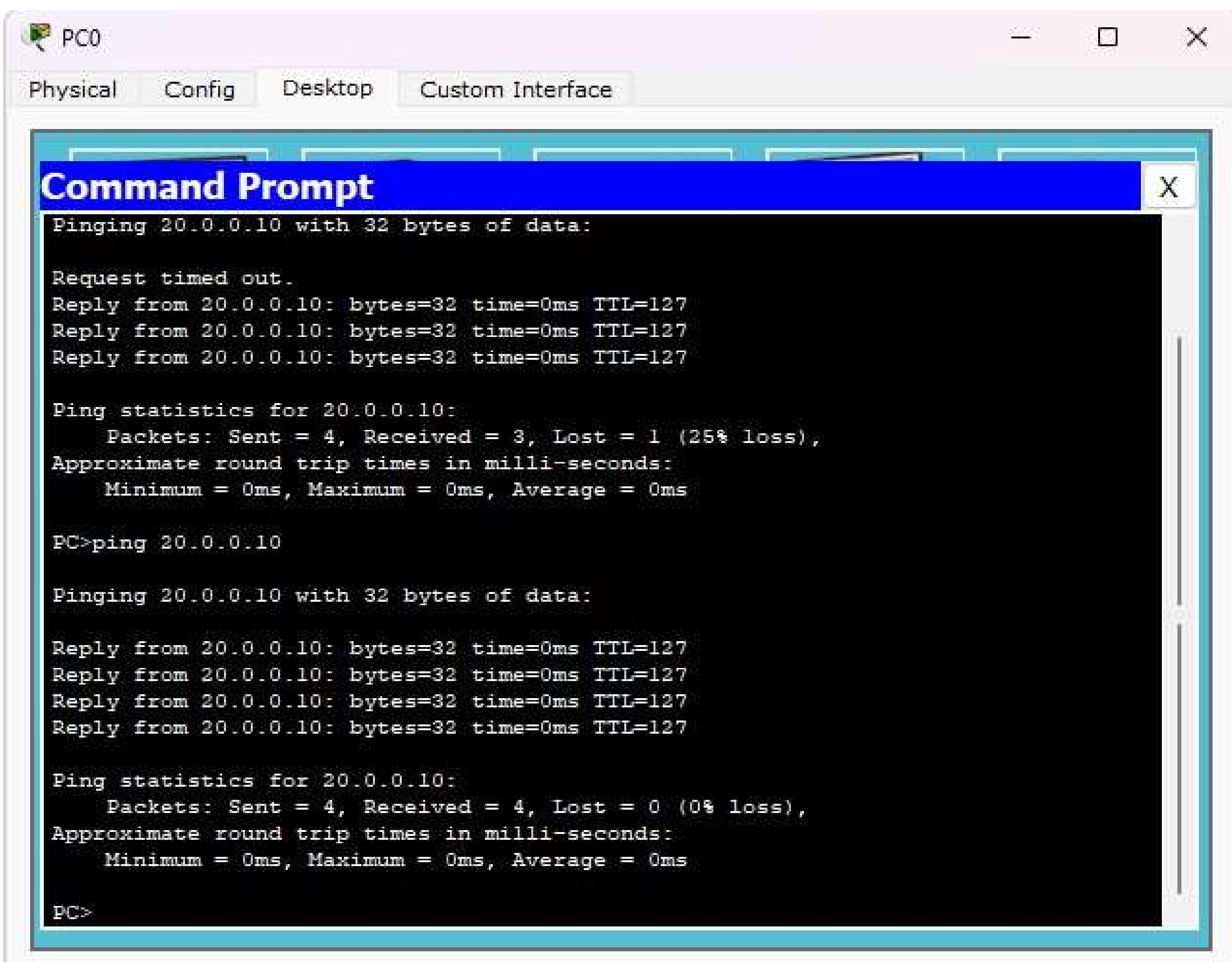
Gateway of last resort is not set

C 10.0.0.0/8 is directly connected FastEthernet 0/0

C 20.0.0.0/8 is directly connected, FastEthernet 1/0.

Screen Shots:





Program 3

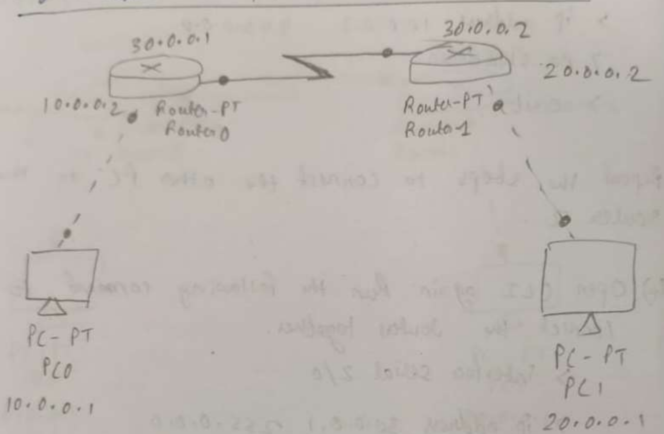
Aim: Configure default route, static route to the Router(Part 1).

Topology , Procedure and Observation:

16/10/24 (10)

Experiment - 3

1.> Configure default route, static route to the Router.



Aim: Successful transmission of packets from one PC to other of different router.

Topology: Two PCs are connected to two different routers using copper cross over and those two routes are connected to each other using serial DCE.

Procedure:

1) After successful configuration of the two routers. Open CLI of one router and follow the command to configure static routing.

> ip route 20.0.0.0 255.0.0.0 30.0.0.2

Open the CLI of other route

> ip route 10.0.0.0 255.0.0.0 30.0.0.1

Observation:

11

After Configuration of static routing two PCs of two different router ~~are~~ networks are able to transfer/ ~~to~~ transmits the packet using the ping command.

~~>ping~~

Output.

> show ip route

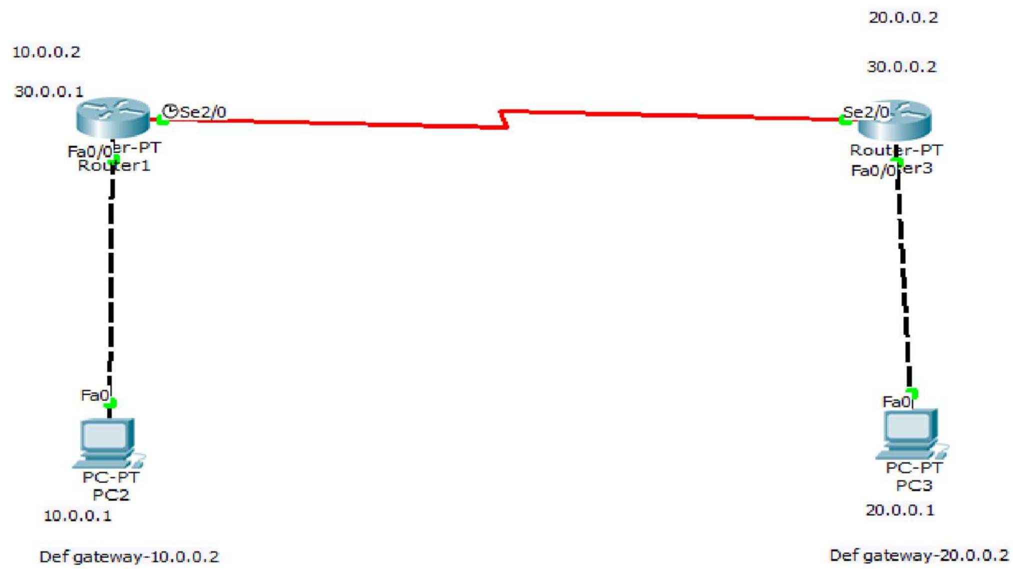
Gateway of last resort is not set.

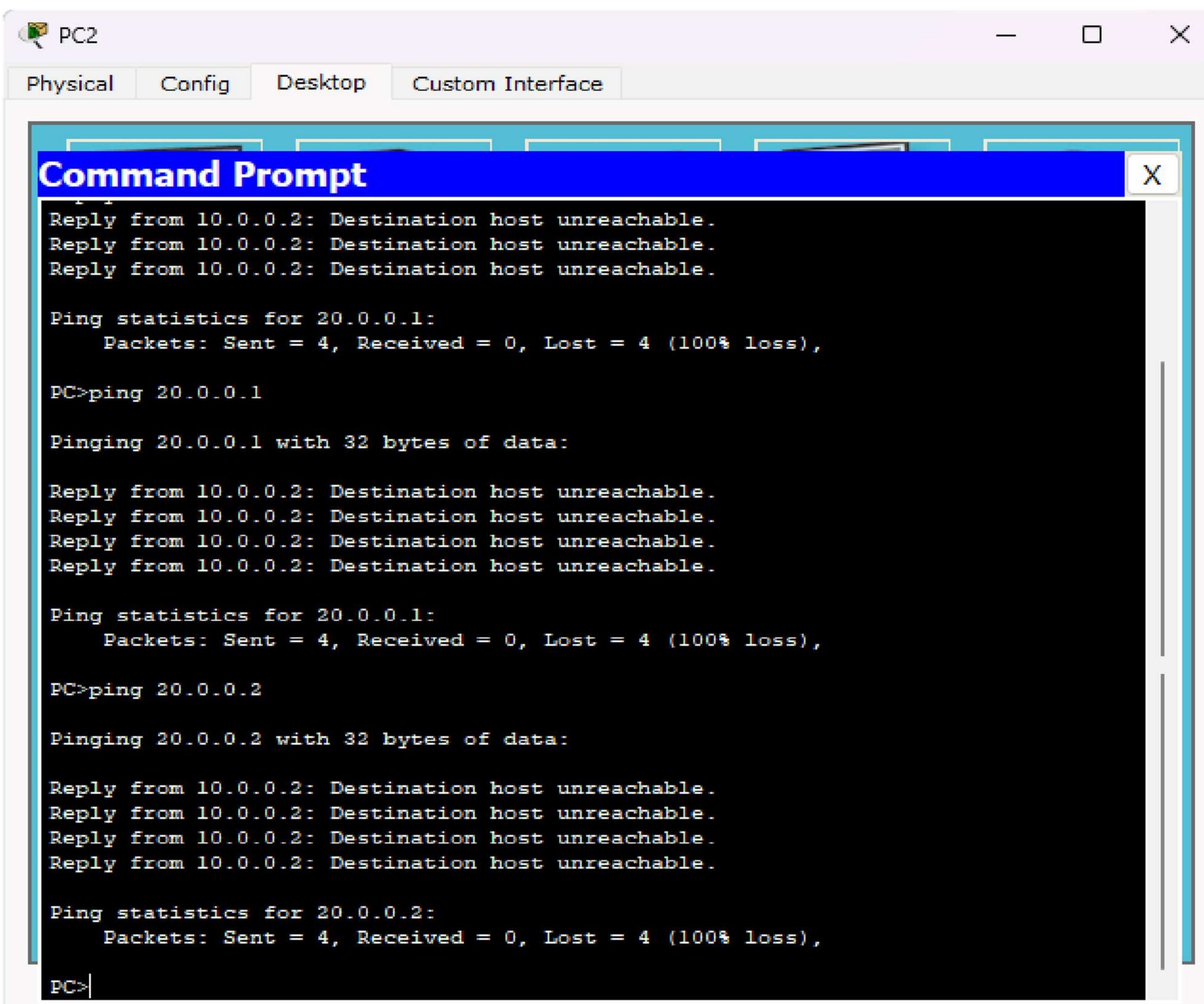
C 10.0.0.0/8 is directly connected, FastEthernet 0/0

S 20.0.0.0/8 [1/0] via 30.0.0.2

C 30.0.0.0/8 is directly connected, serial 2/0.

Screen Shots:





Program 4

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

Topology , Procedure and Observation:

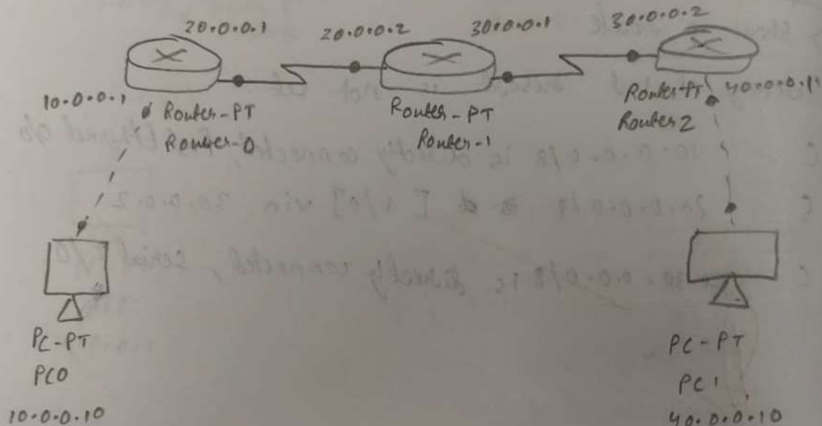
23/10/24

(12)

Experiment - 3

1) Configure default route, static route to the Router.

Aim: To configure a default route and static route to a router in a network and ~~to~~ verify connectivity.



Topology: Router 0 and Router 2 are connected serially via Router 1 use DCE cable. PC0 is connected to Router0 using copper cross-over, similarly PC-1 is connected to Router 2.

Procedure:

1) Configure the fast Ethernet connection between the PC and the router with appropriate IP addresses.

2) For Router 1 set the static routing connectivity by following commands.

```
Router # config terminal
Router(config) # ip route 40.0.0.0 255.0.0.0 30.0.0.2
Router(config) # ip route 10.0.0.0 255.0.0.0 20.0.0.1
```

These commands will set the static Routing of the Router 1. (13)

3.) set the default Routing if Router 0 and Router 2.

- Open CLI on Router 0.

  Router # config terminal.

  Router(config) # ip route 0.0.0.0 0.0.0.0 20.0.0.2

- Open CLI on Router 2

  Router # config terminal.

  Router(config)# ip route 0.0.0.0 0.0.0.0 30.0.0.1

when the network and the subnet mask are set as 0.0.0.0, Any ~~*~~ unknown network will be transferred to the next hop address mentioned.

Observation: After configuring the default route, PC0 could communicate with external networks, including PC1. The static route ensured that packets are followed the specified path as given in the static routing.

Output.

Router 1: # show ip route

S 10.0.0.0/8 [1/0] via 20.0.0.1

C 20.0.0.0/8 is directly connected, serial 2/0

C 30.0.0.0/8 is directly connected, serial 3/0

S 40.0.0.0/8 [1/0] via 30.0.0.2

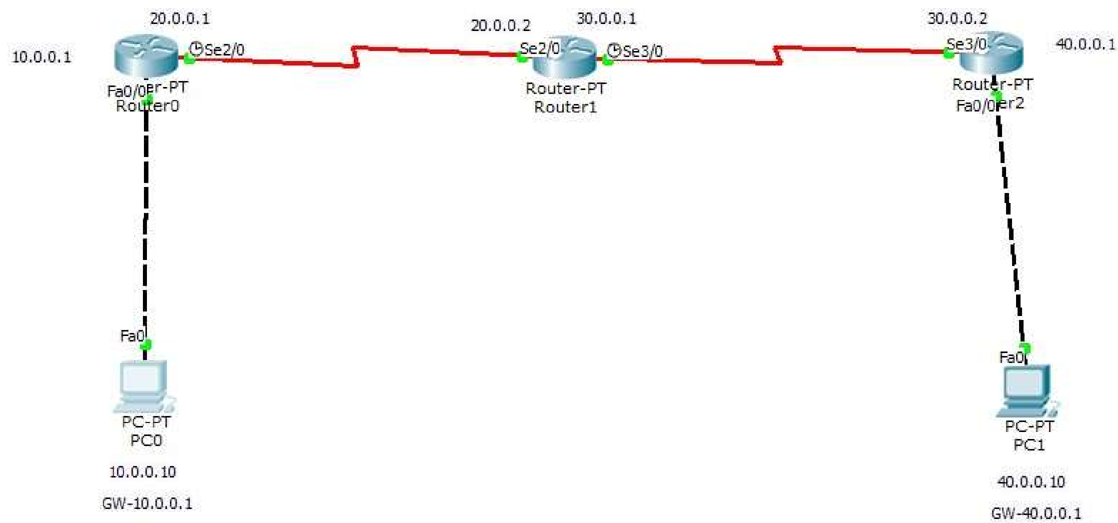
Router 0 : # show ip route

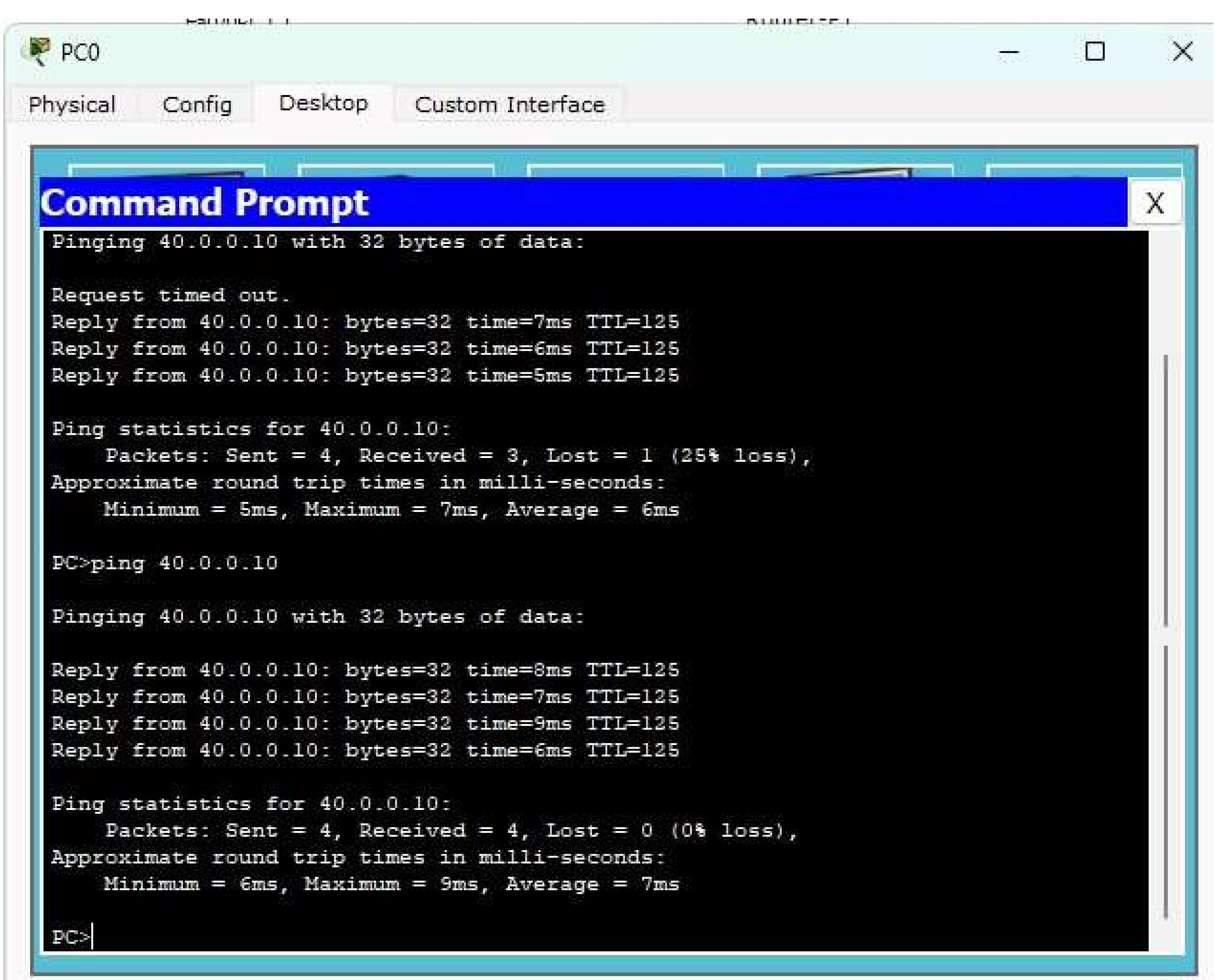
C 10.0.0.0/8 is directly connected, FastEthernet 0/0
C 20.0.0.0/8 is directly connected serial 2/0
S* 0.0.0.0/0 [1/0] via 20.0.0.2.

Router 2 : # show ip route

C 40.0.0.0/8 is directly connected, FastEthernet 0/0
C 30.0.0.0/8 is directly connected, serial 3/0
S* 0.0.0.0/0 [1/0] via 30.0.0.1

Screen Shots:





Program 5

Aim: Configure DHCP within a LAN and outside LAN.

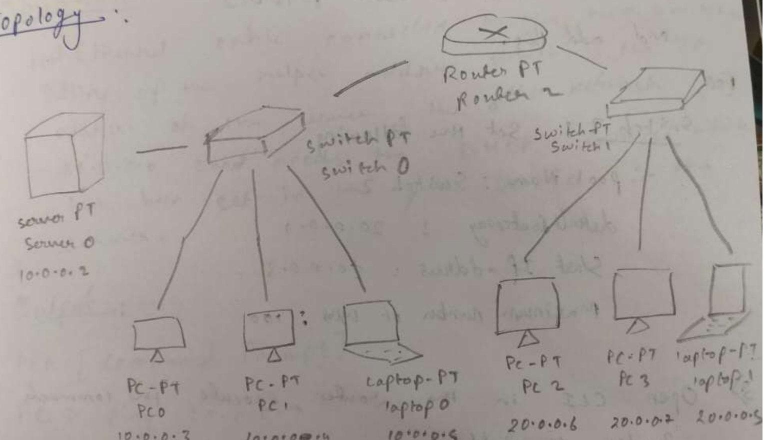
Topology , Procedure and Observation:

13/11/24 Experiment - 4 (15)

Configure DHCP within a LAN and outside LAN.

Aim: To configure and verify the functioning of DHCP both within a LAN and in a network that extends beyond the local LAN.

Topology:



Procedure:

1.) Connect the end devices and server with the routers using copper straight through as shown in the topology.

2.) In server,
go to Desktop > IP configuration and set the static address as
IP address : 10.0.0.2
Subnet Mask : 255.0.0.0
Default Gateway : 10.0.0.1

16

Then go to Services > DHCP

configure the pools like below.

Switch 1 : Set the following

pool name : switch 1

default Gateway : 10.0.0.1

Start IP address : 10.0.0.3

maximum number of users : 100

and add this.

For,

Switch 2 : Set the following

pool Name : Switch 2

default Gateway : 20.0.0.1

Start IP address : 20.0.0.3

Maximum number of users : 100

3) Open CLI in the router, execute the commands

Router # enable,

Router # config terminal

Router (config) # interface FastEthernet 0/0

Router (config-if) # ip ~~helper~~ address 10.0.0.1 255.0.0.0

Router (config-if) # ip helper-address 10.0.0.2

Router (config-if) # - no shutdown.

exit.

Router (config) # interface fastEthernet 1/0. (17)

Router (config-if) # ip address 20.0.0.1 255.0.0.0

Router (config-if) # ip helper-address 10.0.0.2

Router (config-if) # no shutdown

exit.

Observation: Set up the Router with the fastEthernet cable connected to the two switches Setting up the helper address which is the IP address of the server, the other network 20.0.0.0 cand access the DHCP service which has been set in the pool servic in the server.

Output:

PC0: [Command Prompt]

PC > ping 20.0.0.7

Pinging 20.0.0.7 with 32 bytes of data.

Request timed out.

Reply from 20.0.0.7 : byte = 32 time = 0ms TTL = 127

Reply from 20.0.0.7 : byte = 32 time = 0 ms TTL = 127

Reply from 20.0.0.7 byte = 32 time = 2 ms TTL = 12

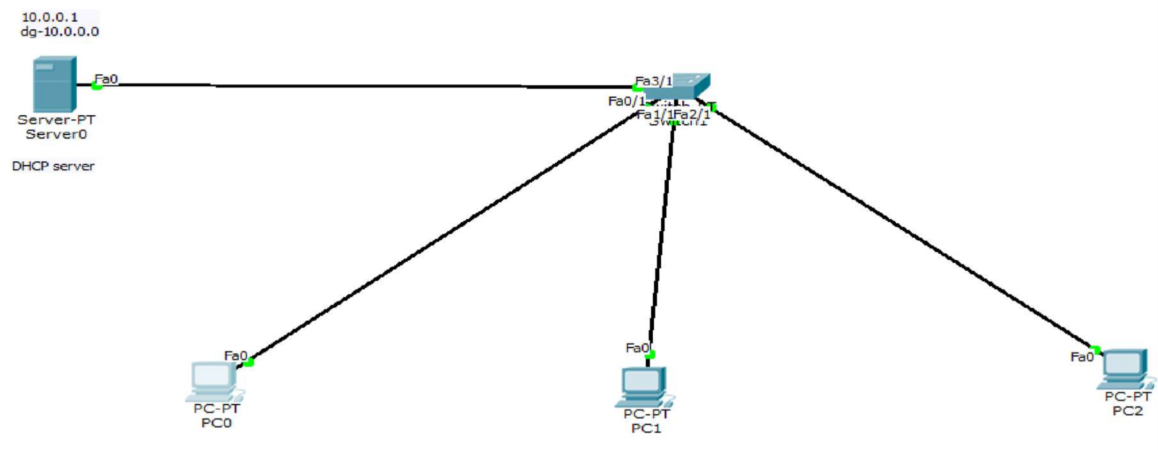
Ping Statistics for 20.0.0.7

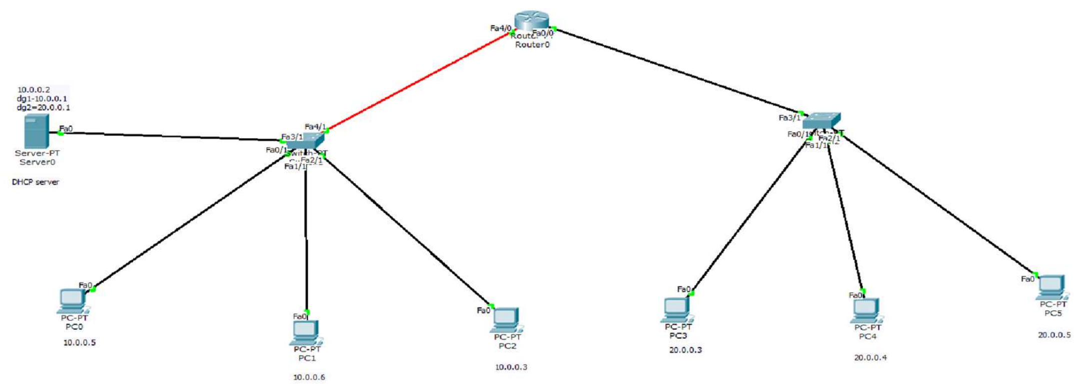
Packets : sent = 4, Received = 3 lost = 1 (25% loss)

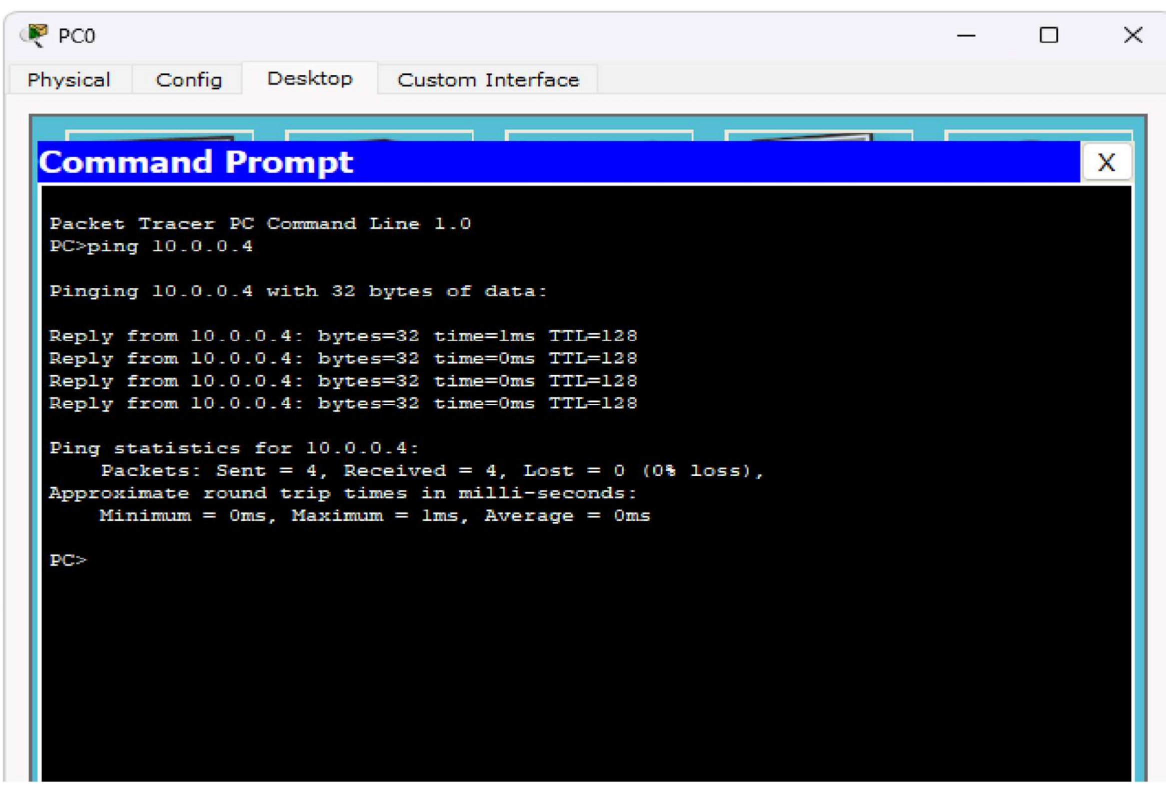
27/11/24

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

Screen Shots:







Program 6

Aim: Configure RIP routing Protocol in Routers .

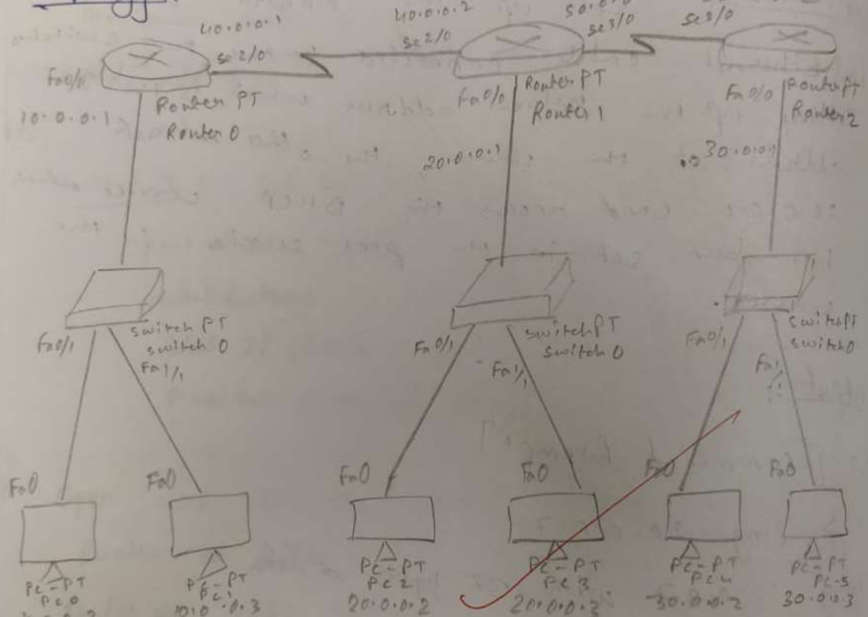
Topology , Procedure and Observation:

20/11/24 Experiment - 5 (18)

1) Configure the routing Protocol in Routers. RIP

Aim: To configure RIP routing Protocol in Routers and verify connectivity.

Topology:



Procedure.

1.) Configure connect the end devices with the switches which are in ~~to~~ turn connected to the routers as shown in the topology.

2.) Now open CLI in Router 0.

Router > enable

Router # config terminal

(iii) Router (config) # router rip
Router (config - router) # network 10.0.0.0
Router (config - router) # network 40.0.0.0

Open CLI in Router 1.

Router > enable
Router # config terminal
Router (config) # router rip
Router (config - router) # network 40.0.0.0
Router (config - router) # network 20.0.0.0
Router (config - router) # network 50.0.0.0

Open CLI in Router 2.

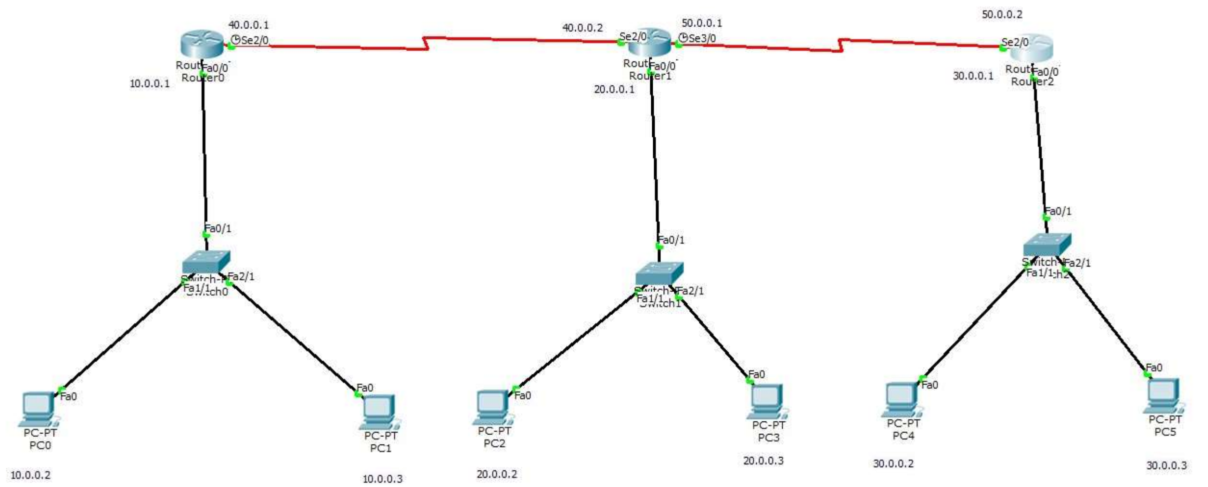
Router > enable
Router # config terminal
Router (config) # router rip
Router (config - router) # network 50.0.0.0
Router (config - router) # network 30.0.0.0

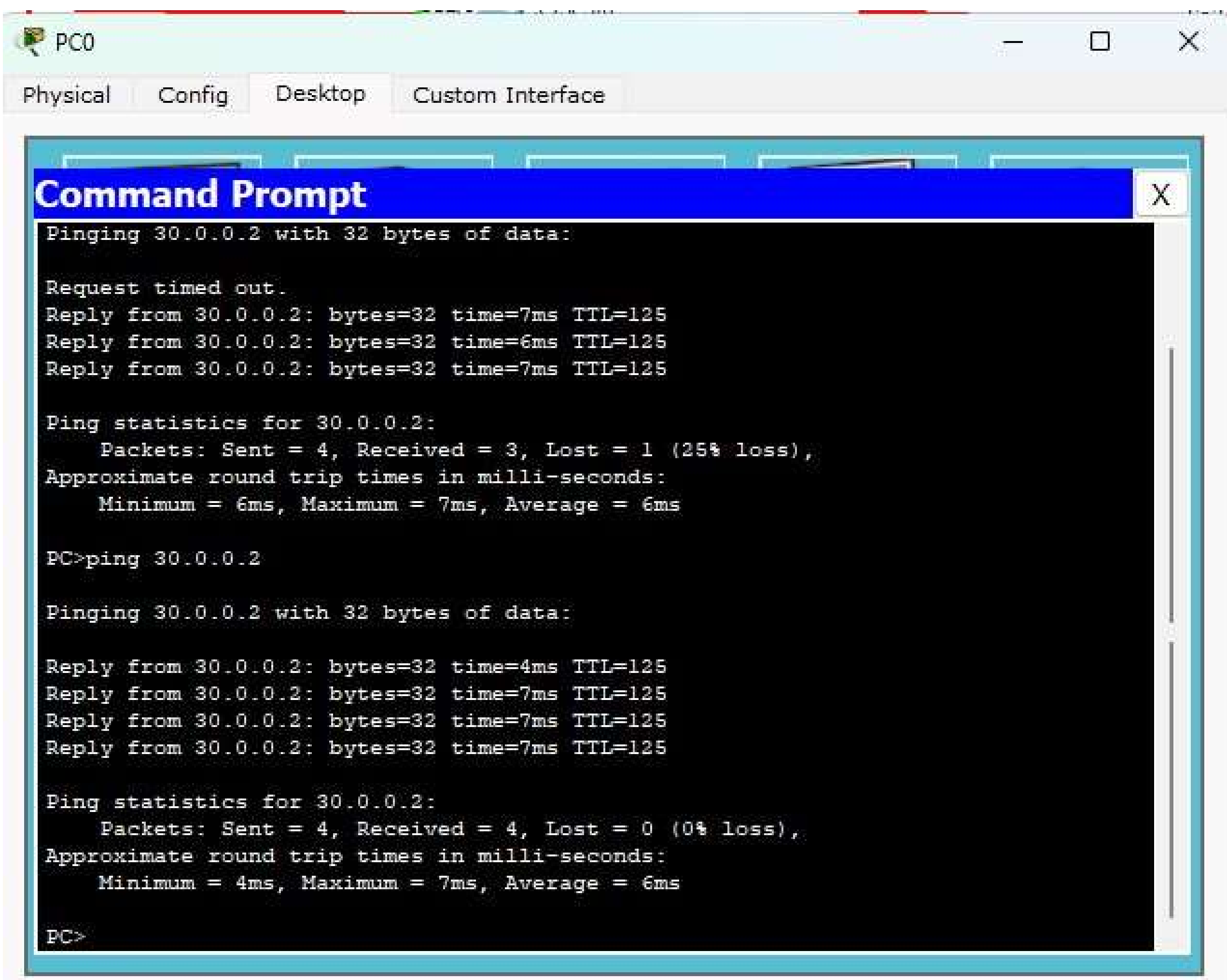
Give the IP address of the root to establish the connection.

Observation :- After configuring the RIP router protocol the end devices are able to communicate properly.

Output : Router 0
Router > enable
Router # show ip route.
C 10.0.0.0/8 is directly connected, fastEthernet 0/0
R 20.0.0.0/8 [120/1] via 40.0.0.2, 00:00:24, serial 2/0
R 30.0.0.0/8 [120/1] via 40.0.0.2, 00:00:24, serial 2/0
C 40.0.0.0/8 is directly connected, Serial 2/0.
R 50.0.0.0/8 [120/1] via 40.0.0.2, 00:00:24, serial 2/0

Screen Shots:

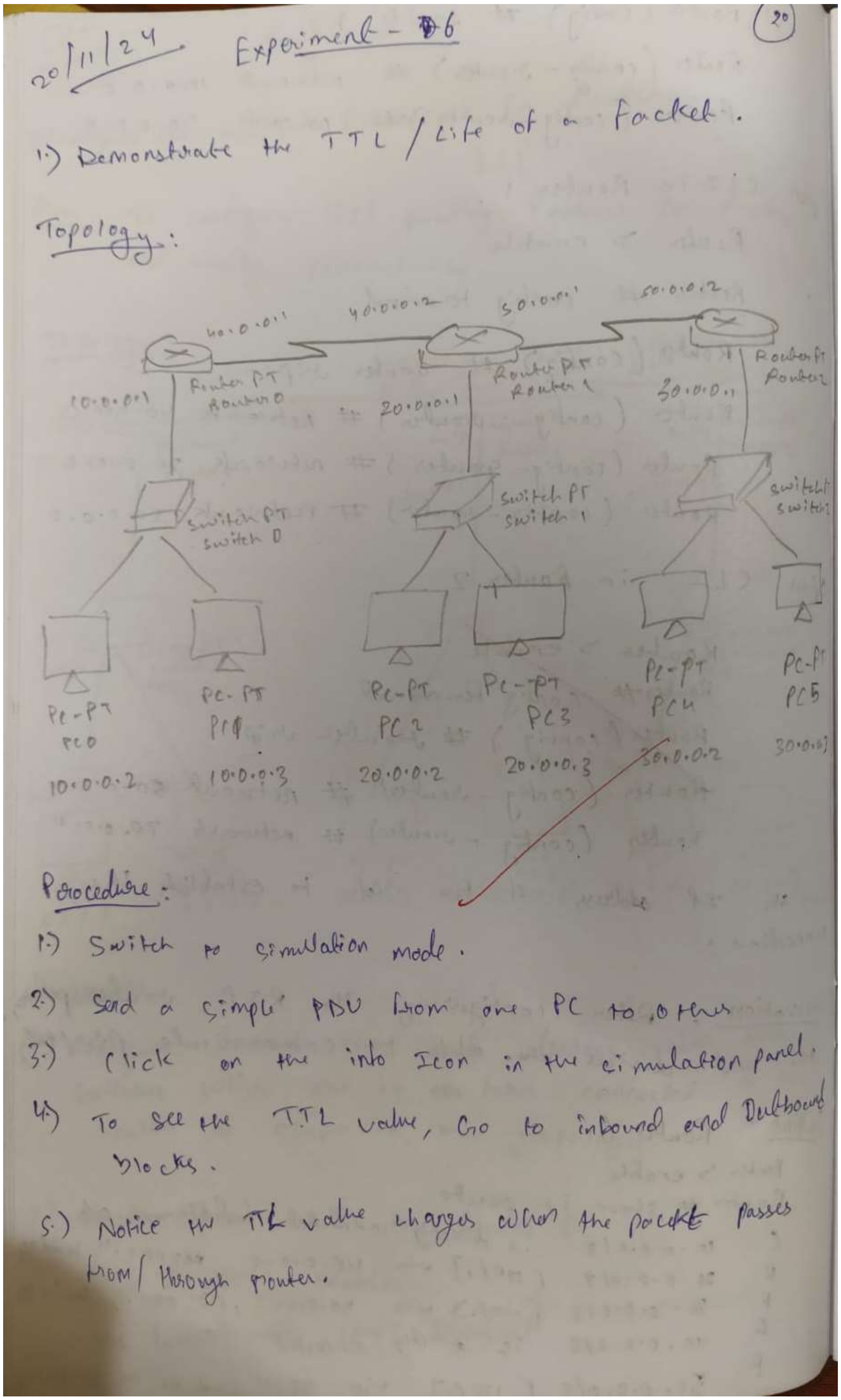




Program 7

Aim: Demonstrate the TTL/ Life of a Packet .

Topology , Procedure and Observation:



20/11/24 Experiment - 6

(20)

1.) Demonstrate the TTL / Life of a Packet.

Topology:

Procedure:

1.) Switch to simulation mode.

2.) Send a simple PDU from one PC to others.

3.) Click on the info Icon in the simulation panel.

4.) To see the TTL value, Go to inbound and Outbound blocks.

5.) Notice the TTL value changes when the packet passes from/through router.

(21)

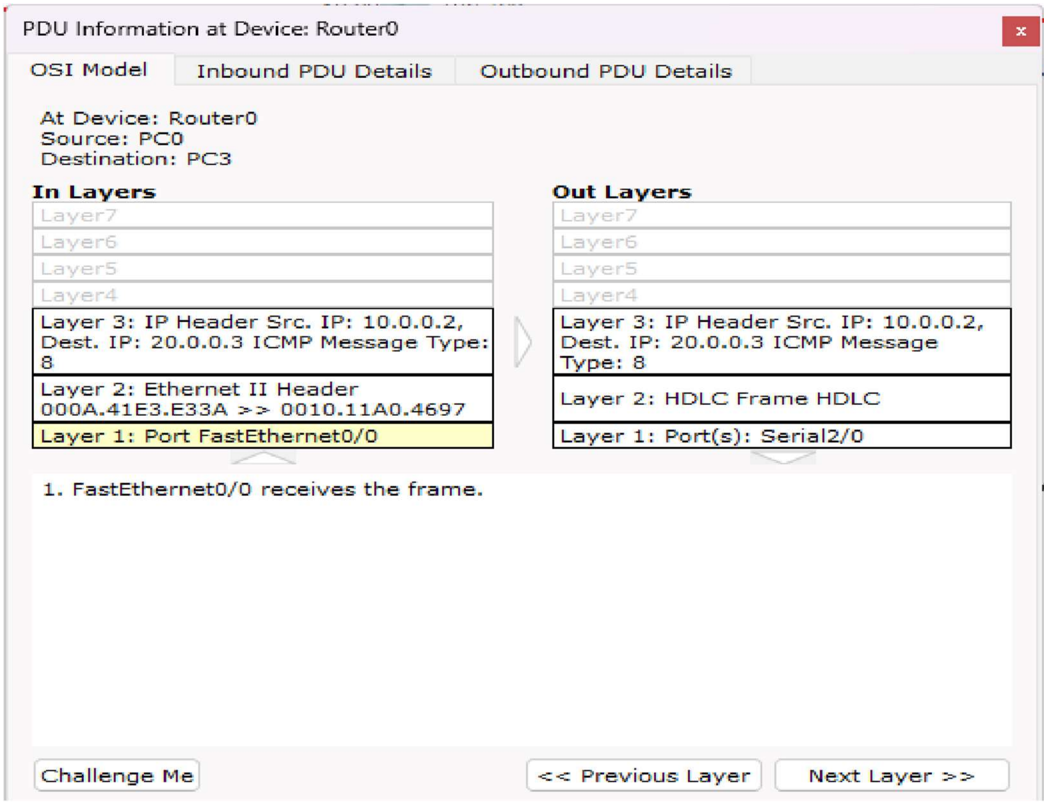
Observation :

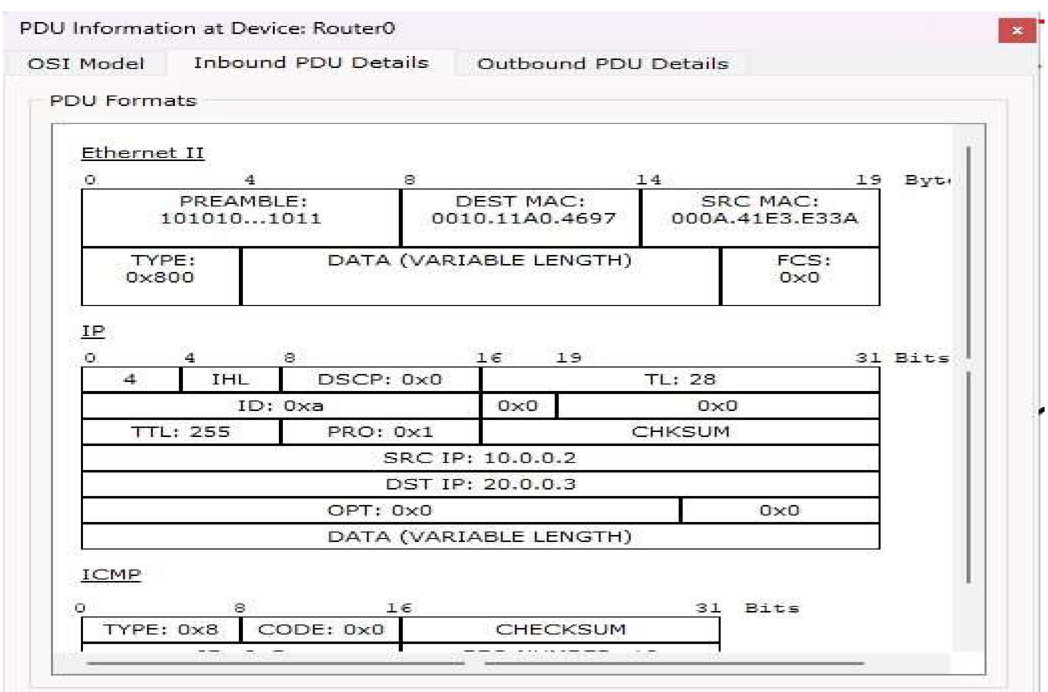
As a packet travels through a network, its TTL (Time to live) value decreases by 1 at each router it passes. Initially, the packet might start with a TTL of 255. when it reaches the first router the TTL reduced to 254. As the packet continues ~~the TTL reduced to 25~~ its journey through cash successive router, the TTL continues to decrement - 253 after the second router 252 after the third and so on.

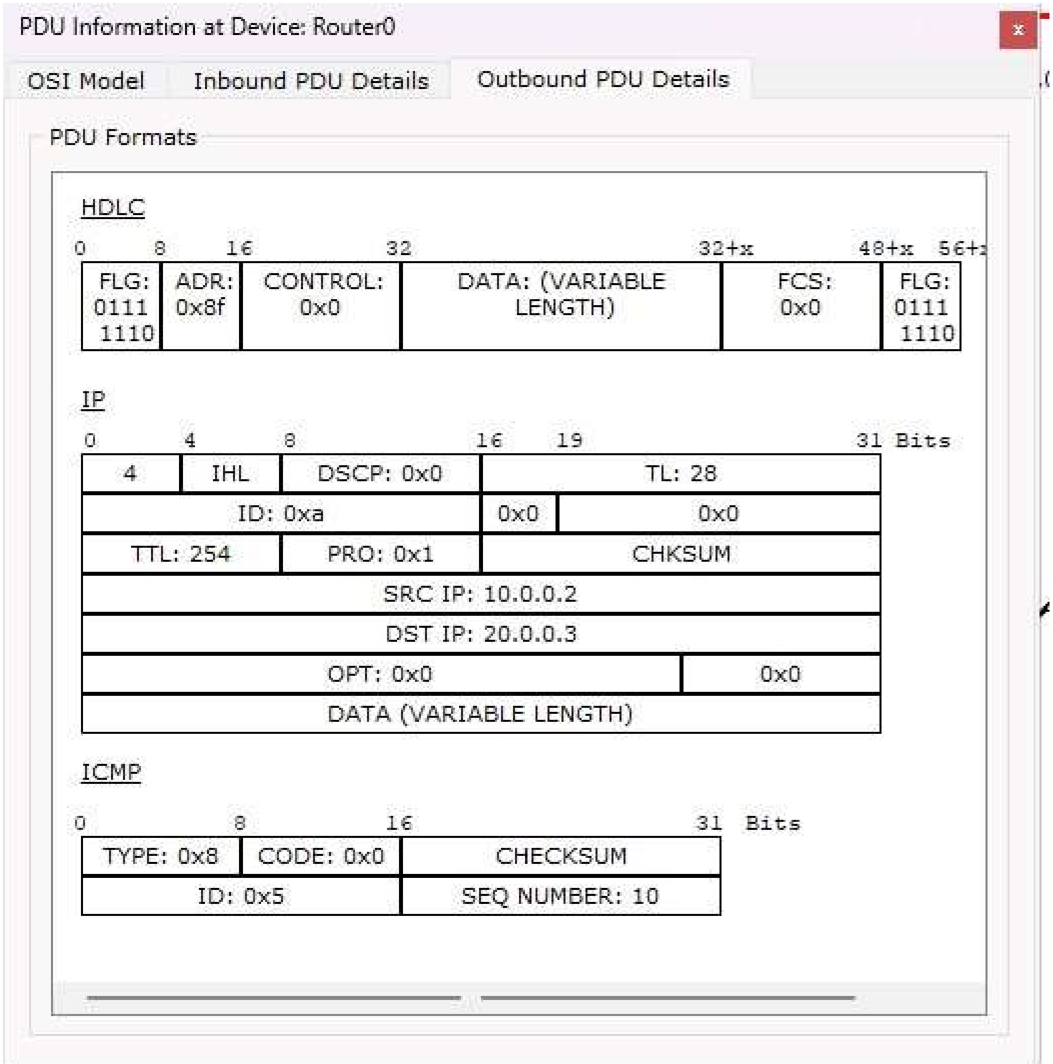
if the TTL reaches 0 before reaching its destination the packet is discarded and an ICMP "Time Exceeded" message is sent back to the sender.

28/11

Screen Shots:






PDU Information at Device: Router0
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
0 8 16 32 32+x 48+x 56+x
FLG: 0111 1110
ADR: 0x8f
CONTROL: 0x0
DATA: (VARIABLE LENGTH)
FCS: 0x0
FLG: 0111 1110
IP
0 4 8 16 19 31 Bits
4
IHL
DSCP: 0x0
TL: 28
ID: 0xa
0x0
0x0
TTL: 254
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.2
DST IP: 20.0.0.3
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
0 8 16 31 Bits
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x5
SEQ NUMBER: 10

Program 8

Aim: Configure OSPF routing protocol .

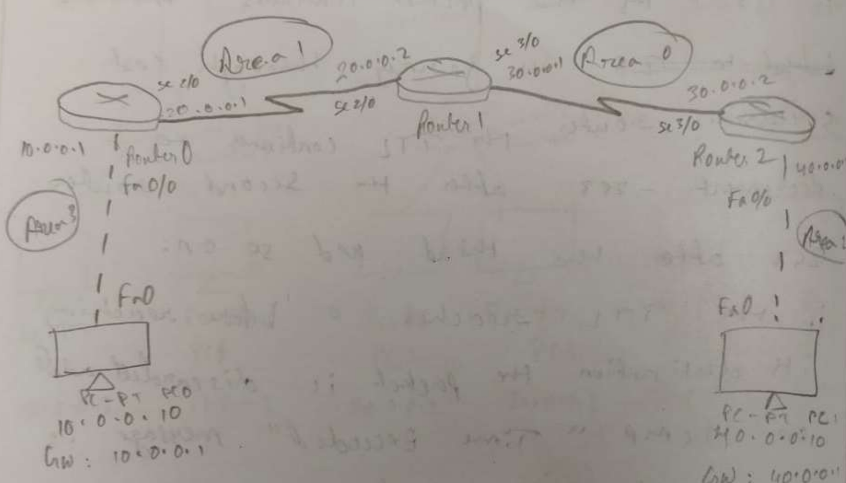
Topology , Procedure and Observation:

27/11/24 Experiment ~~6~~7 (22)

1.) Configure OSPF routing Protocol.

Aim: To configure OSPF routing Protocol in Routers and verify connectivity.

Topology:



Procedure:

1.) Establish the network ~~as~~ connection as shown in the topology.

2.) Set the IP address of the PC and the gateway address respectively.

3.) Open CLI in Router 0.

(config) # interface fastEthernet 0/0

(config-if)# ~~interface~~ ip address 10.0.0.1 255.0.0.0

(config-if)# no shutdown

(config-if)# exit

(config) # interface serial 2/0 (23)

(config-if) # ip address 20.0.0.1 255.0.0.0

(config-if) # encapsulation ppp

(config-if) # clock rate 64000.

Open CLI in router 1

(config) # interface serial 2/0

(config-if) # ip address 20.0.0.2 255.0.0.0

(config-if) # encapsulation PPP

(config-if) # no shutdown

(config-if) # exit

(config) # interface serial 3/0

(config-if) # ip address 30.0.0.1 255.0.0.0

(config-if) # encapsulation ppp

(config-if) # clock rate 64000.

(config-if) # no shutdown

(config-if) # exit.

Open CLI in router 2.

(config) # interface FastEthernet 0/0

(config-if) # ipaddress 40.0.0.1 255.0.0.0

(config-if) # no shutdown

(config-if) # exit

(24)

(config)# interface serial 3/0
(config-if)# ip address 30.0.0.2 255.0.0.0
(config-if)# encapsulation ppp
(config-if)# no shut down
(config-if)# exit.

4.) Enable ip routing by configuring ospf routing protocol.

In Router R0

(config) # router ospf 1
(config-router)# router-id 1.1.1.1
network 10.0.0.0 0.255.255.255 area3
network 20.0.0.0 0.255.255.255 area1
exit.

In Router R1

(config) # router ospf 1
(config-router) # router-id 3.3.3.3
network 30.0.0.0 0.255.255.255 area
network 40.0.0.0 0.255.255.255 area1
exit.

5.) check the routing table.

Router # show ip route.

You can see the code O which stands for the OSPF connection.

25

There must be one interface up to keep ospf process up. so it is better to configure loopback address to router.

R0 (config) # interface loopback0
(config-if) # ip add 172.16.1.252 255.255.0.0

no shutdown.

R1 (config) # interface loop back 0
R(config-if) # ip add 172.16.1.253 255.255.0.0
no shutdown.

R2 (config) # interface loop back 0.
(config-if) # ip add 172.16.1.254 255.255.0.0
no shutdown

6.) Check Routing table of R3.
You can see the code O here now,

7.) Create virtual link between R1 and R2.

R1 (config) # router ospf 1
(config-router) # area 1 virtual-link 2.2.2.2

R2 (config) # router ospf 1
(config-router) # area 1 virtual-link 1.1.1.1
area 0 virtual-link 3.3.3.3

(26)

R3(config) # router ospf 1

(config-router) # area 0 virtual-link 2.2.2.2

8.) Check connectivity between host 10.0.0.10 to 40.0.0.10.

Observation: After config the ospf routing protocol and establishing the virtual link, the two end hosts can communicate seamlessly.

Output:

PC0:

PC > ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data.

Reply from 40.0.0.10 byte = 32 time = 9ms TTL

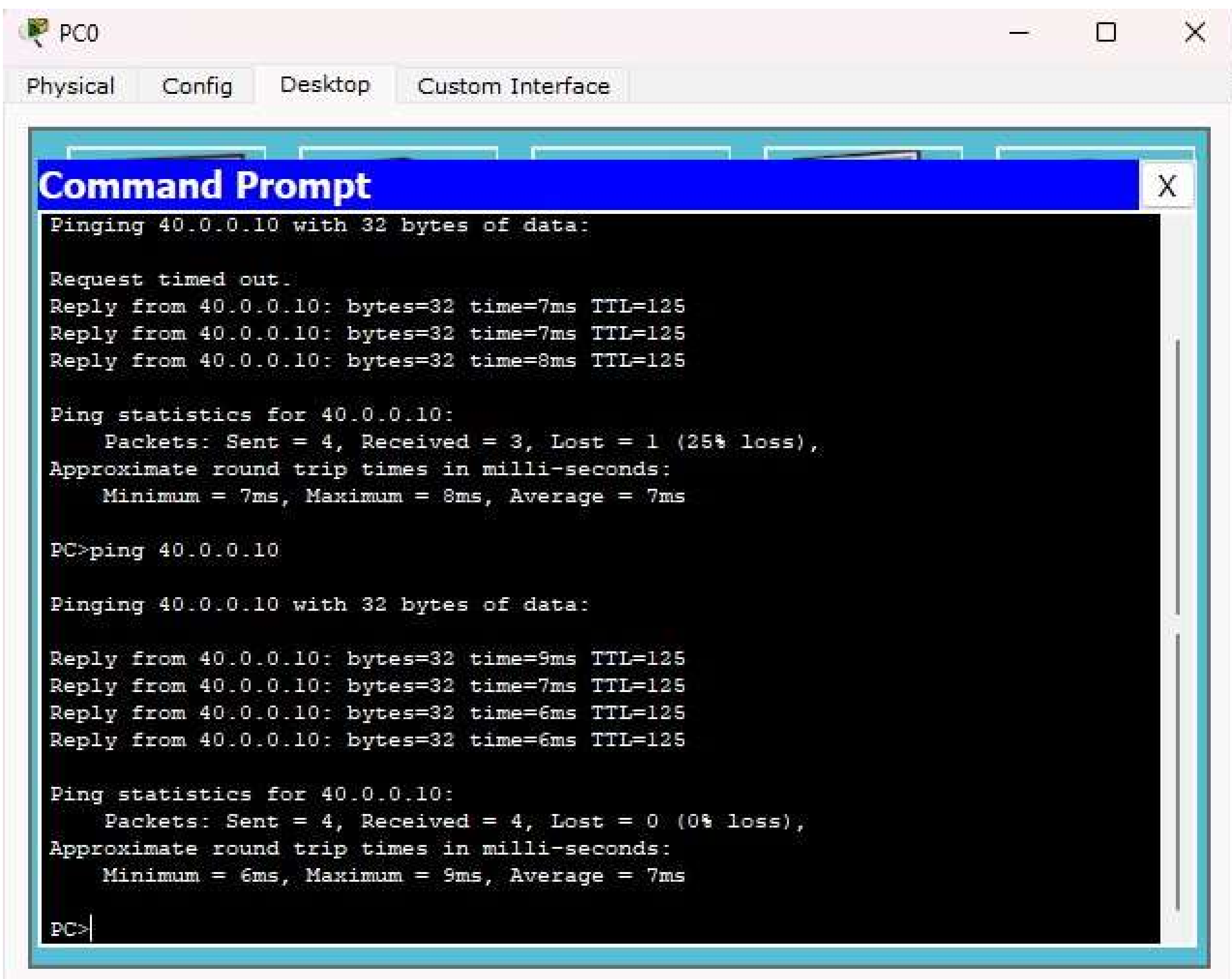
(4 times)

Ping statistics for

Packets Sent = 4 Received = 3 lost = 1

18/12/24

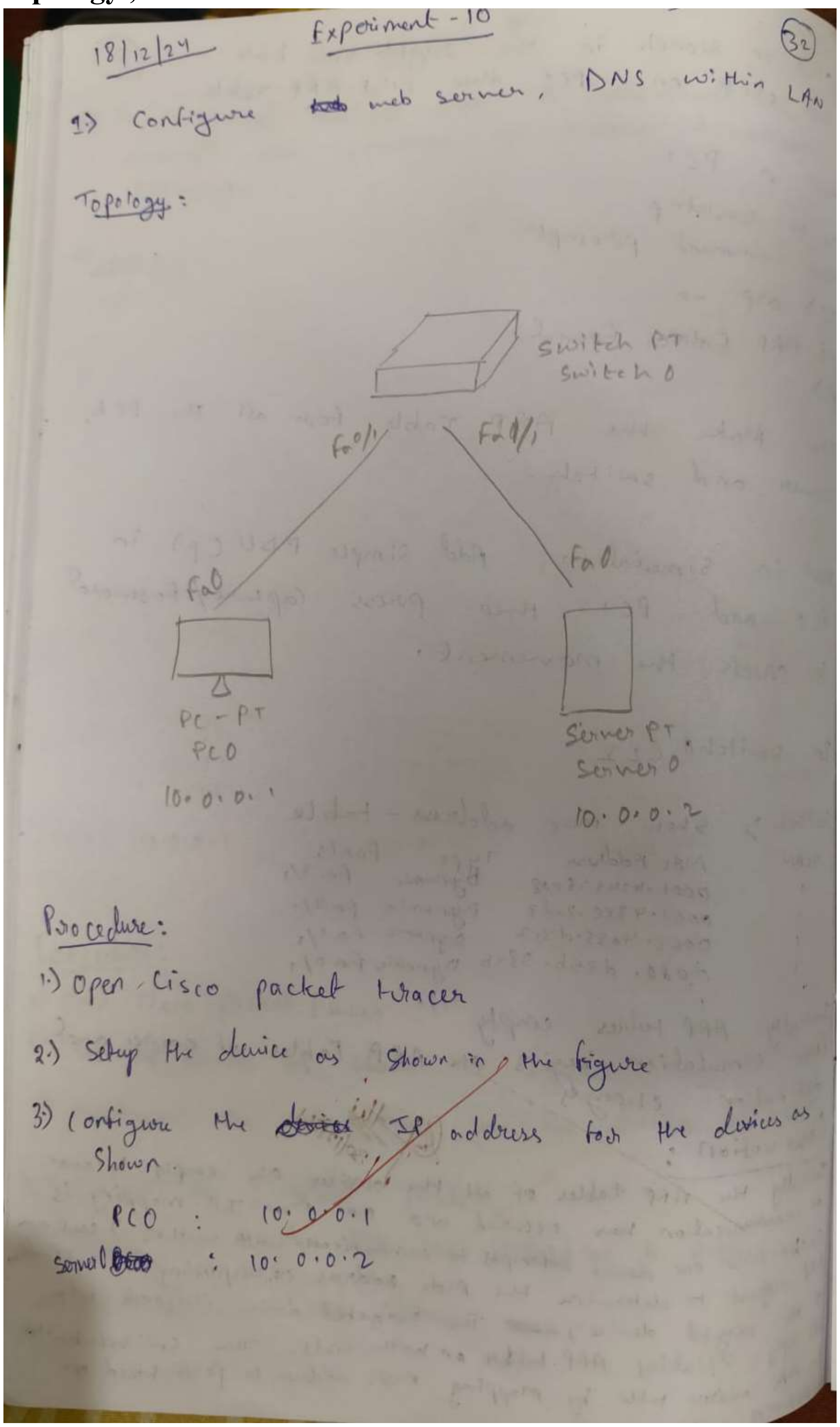
Screen Shots:



Program 9

Aim: Configure Web Server, DNS within a LAN.

Topology , Procedure and Observation:



18/12/24 Experiment - 10 (32)

1) Configure ~~web~~ web server, DNS within LAN

Topology:

Procedure:

1.) Open Cisco packet tracer

2.) Setup the device as shown in the figure

3.) Configure the ~~device~~ IP address for the devices as shown.

PC0 : 10.0.0.1

server0 : 10.0.0.2

4.) Now go to the Server
then config
then DNS
Name : bmsce
address : 10.0.0.2
Press Add.
Then press ON.
5.) Go to PC0 → config → global → setting → DNS server: 10.0.0.2 (The server that provide DNS mapping)
6.) Now go to PC0
then Desktop the Web browser
In URL : https://bmsce then press Go

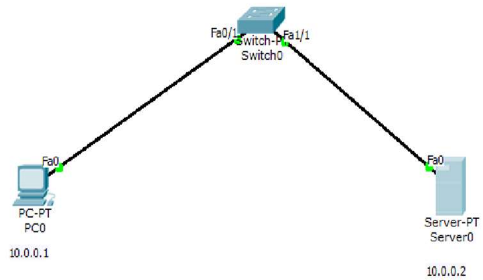
Observation:

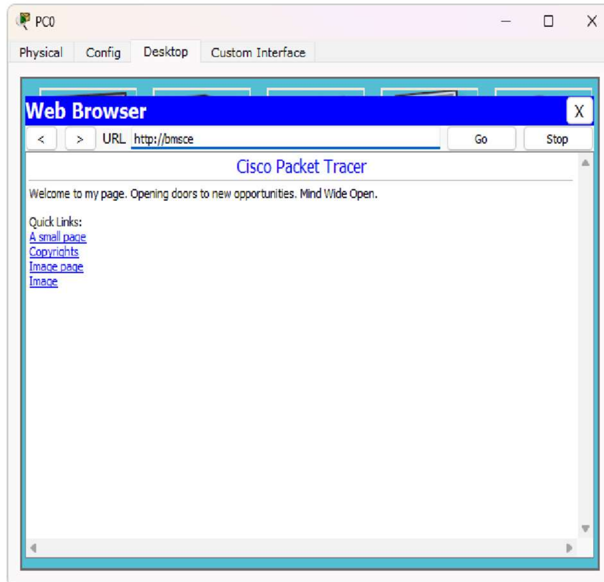
1) The webpage hosted by the server visible on the browser
2.) The DNS was successfully in mapping the domain, name to IP address.
3.) DNS server is a server that contains a domain name: IP address mapping to which the end device send requests to map the name of IP address.

Welcome to CN lab:
Quick link:
A small page
Copyright :
image page :
image.

30/12/24

Screen Shots:

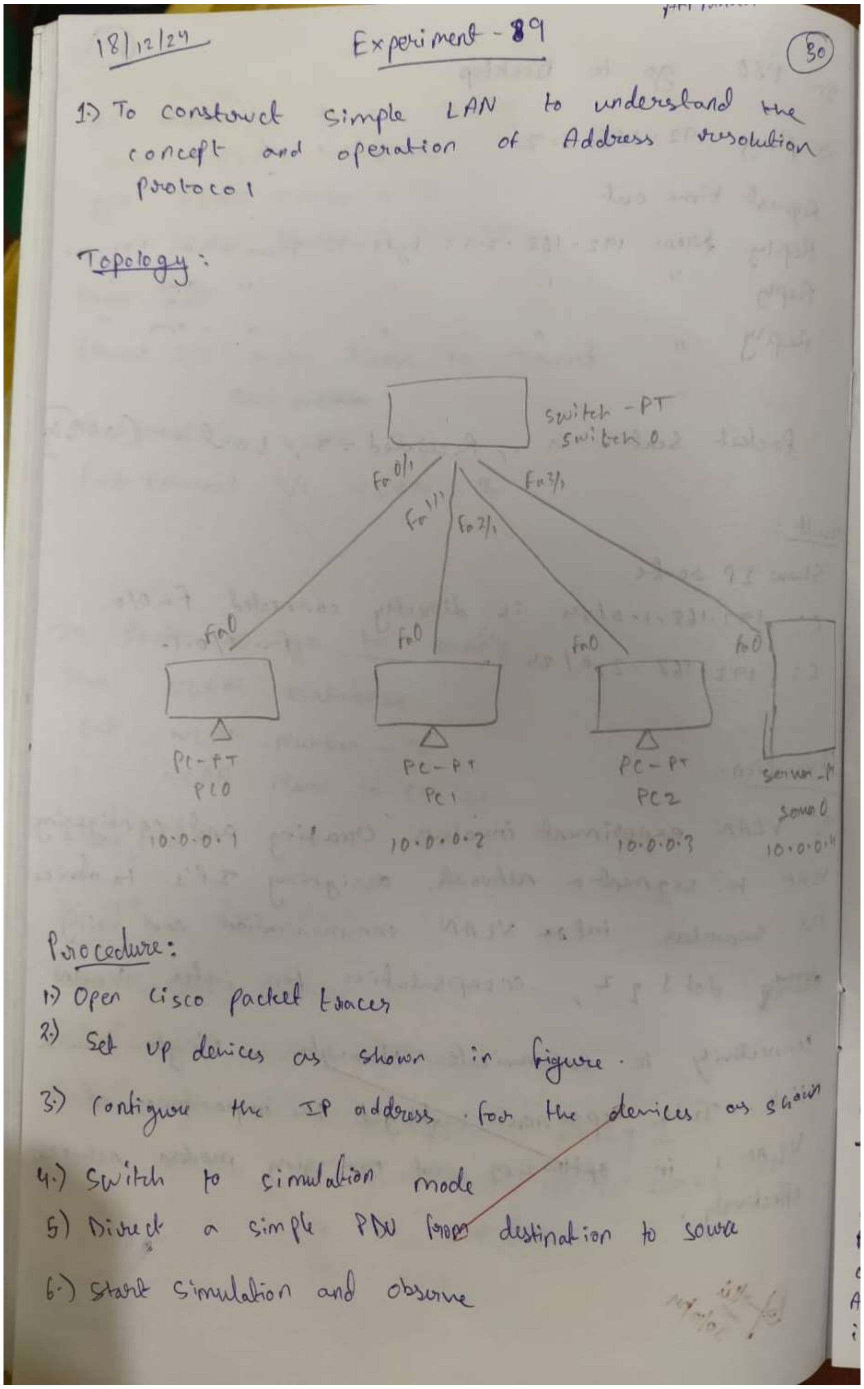




Program 10

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

Topology , Procedure and Observation:


18/12/24

Experiment - 9

30

1.) To construct simple LAN to understand the concept and operation of Address resolution protocol

Topology:

Procedure:

1.) Open Cisco packet tracer

2.) Set up devices as shown in figure.

3.) Configure the IP address for the devices as shown

4.) Switch to simulation mode

5.) Direct a simple PDU from destination to source

6.) Start simulation and observe

4) Click on search in the right side bar (31)
then click on PC0 then click ARP Table.

Now in PC1
Go to Desktop
Then command prompt
PC> arp -a
No ARP Entries Found.
PC>

Now Make the ARP Table for all the PCs, server and switch.

Now in simulation Add simple PDU (p) in PC0 and PC1 then press capture/Forward to check the movement.

In switch0 CLI

Switch > show mac address-table

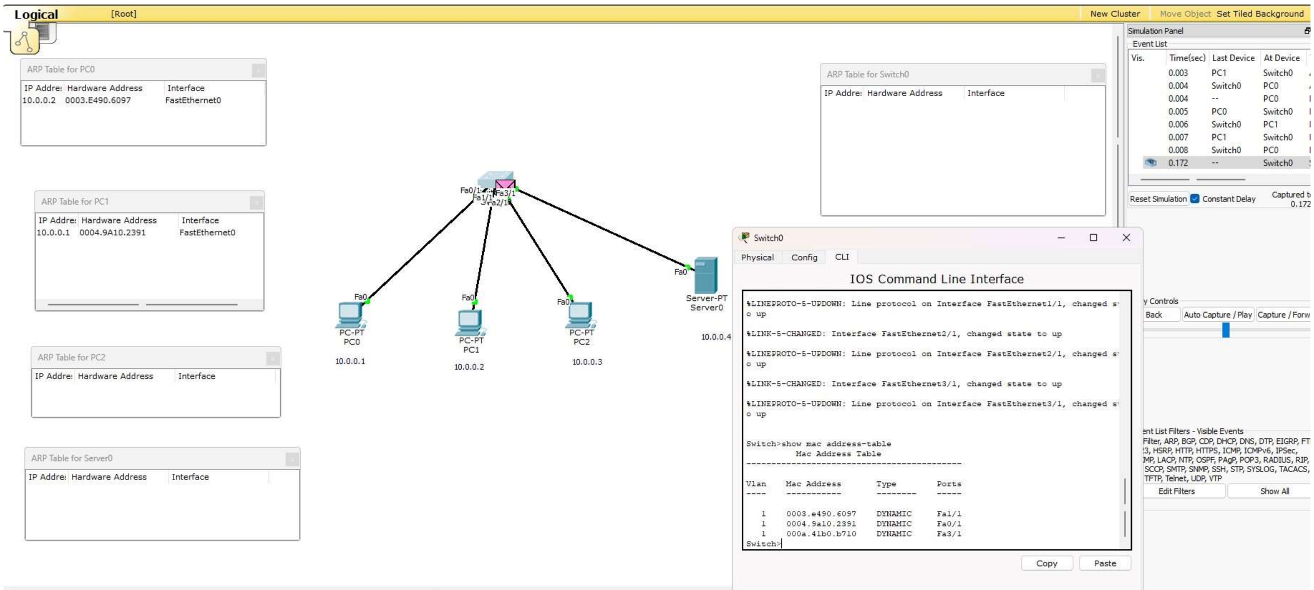
VLAN	MAC Address	Type	Ports
1	0001.4245.6e08	Dynamic	Fa 3/1
1	0001.43ee.2ad3	Dynamic	Fa 2/1
1	0002.4a35.d3c2	Dynamic	Fa 1/1
1	00d0.d35b.384b	Dynamic	Fa 0/1

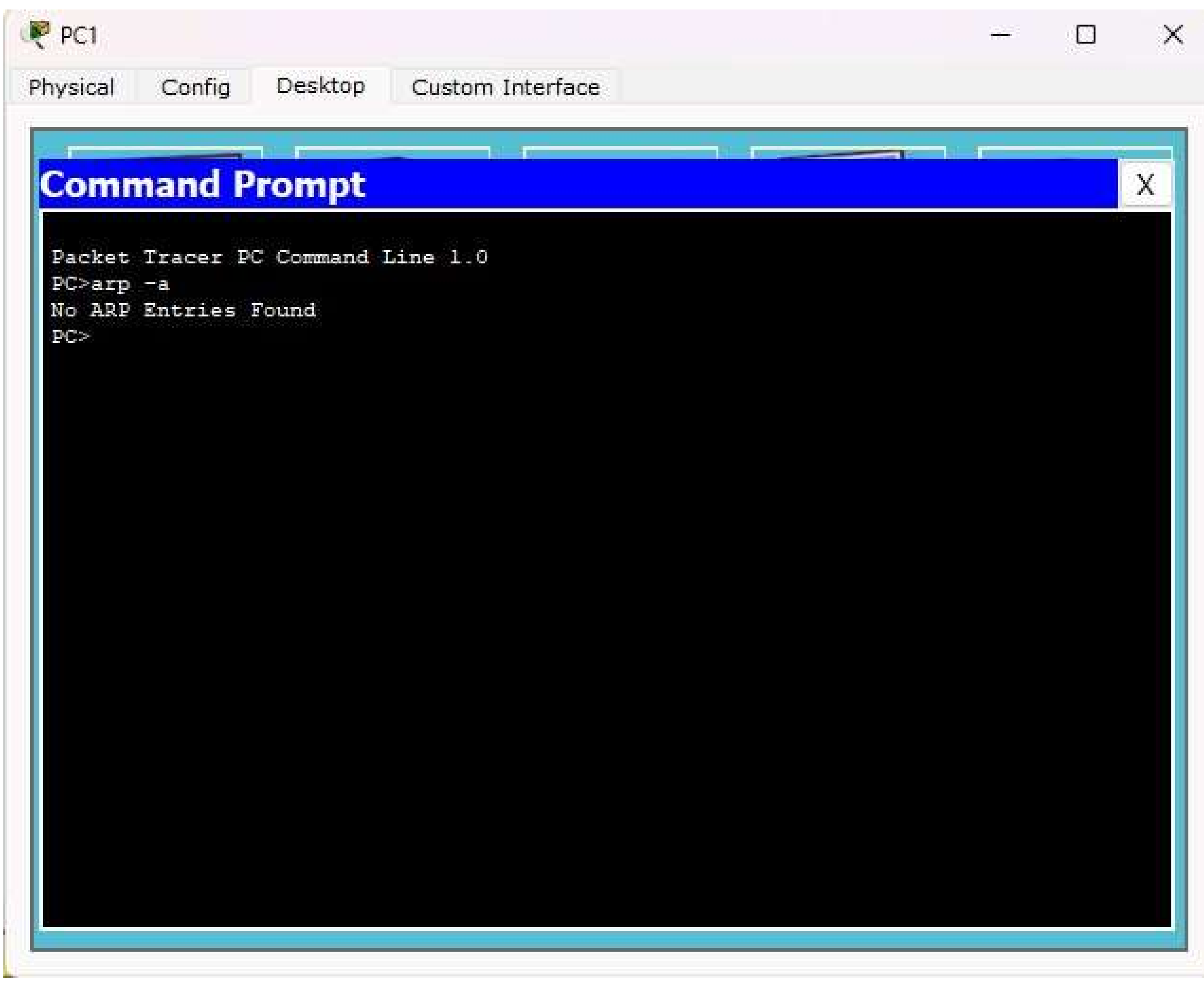
Initially ARP tables empty
After simulation begins the ARP Tables of source and destination changes.

Observation:

Initially the ARP tables of all the devices are empty because no communication has occurred and no Mac-IP mapping is cached. When one device attempts to communicate with another, it sends an ARP request to determine the MAC address corresponding to the IP address of the targeted device; The targeted device responds with ARP reply, updating ARP tables on both ends. The switch build its MAC address table by mapping MAC address to ports based on receiving frame.

Screen Shots:

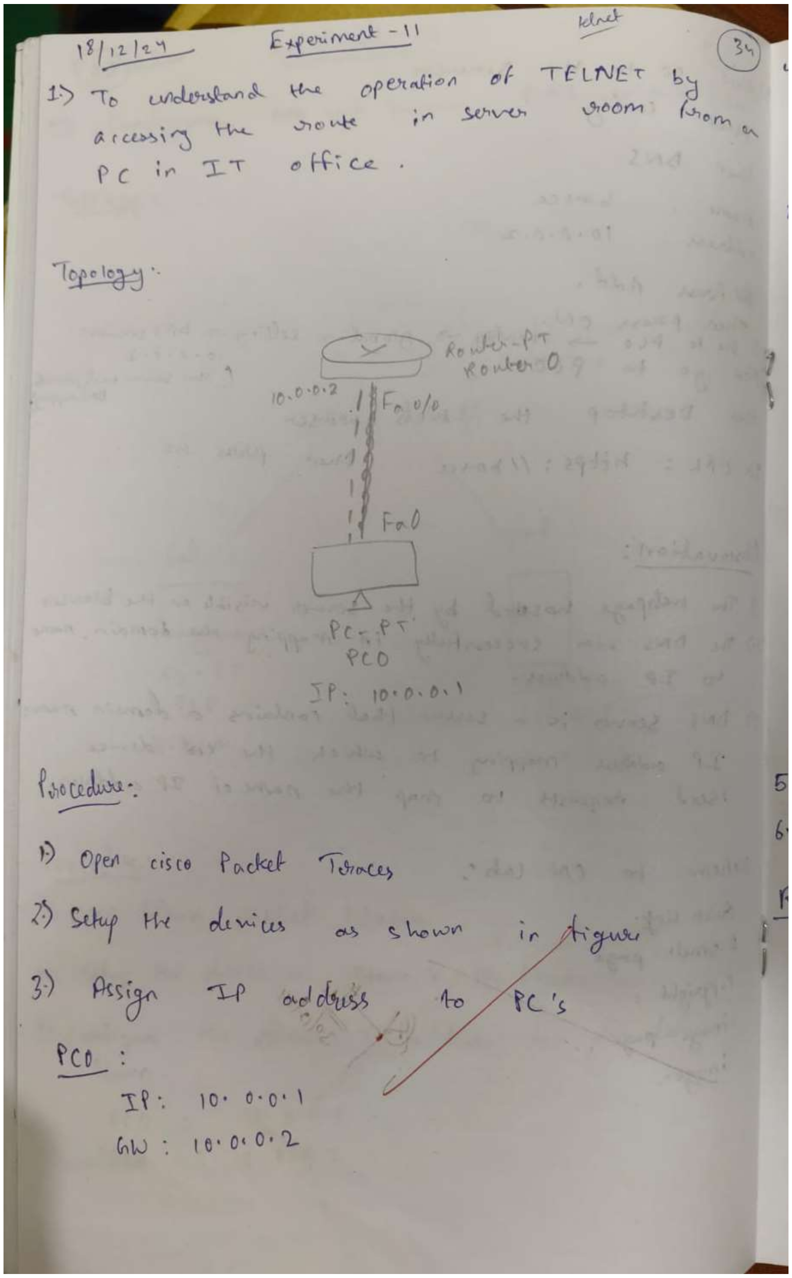




Program 11

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

Topology , Procedure and Observation:



18/12/24 Experiment - 11 telnet (34)

1.) To understand the operation of TELNET by accessing the route in server room from a PC in IT office.

Topology:

Router-PT
Router0
10.0.0.2
Fa0/0
Fa0
PC-PT
PC0
IP: 10.0.0.1

Procedure:

1.) Open cisco Packet Teraces

2.) Setup the devices as shown in figure

3.) Assign IP address to PC's

PC0 :

IP: 10.0.0.1

GW: 10.0.0.2

4) In Router 0

CLI

Router > enable

Router # config t

Router (config) # hostname R1

R1 (config) # enable secret BMSCE

R1 (config) # interface FastEthernet 0/0

IP address 10.0.0.2 255.0.0.0

No shutdown

link vty 0 3

login

password < password 2>

exit

exit

wr (To save changes in router)

5.) Go to CMD in PC0 and ping 10.0.0.2

6.) After 1st ping now type Telnet 10.0.0.2

Result:

[ping]

Pinging 10.0.0.2 with 32 bytes of data

Reply from 10.0.0.2 with bytes 32 time = 0ms TTL=

:

Ping Statistic:

36

[Telnet]

Trying 10.0.0.2 - - - . open

User access verification

<Enter password 1>

R1 > enable

Password : <password 2>

R1 # show IP route

Gateway of last route is not set.

C: 10.0.0.0/8 is directly connected, Fa0/0

R1 #

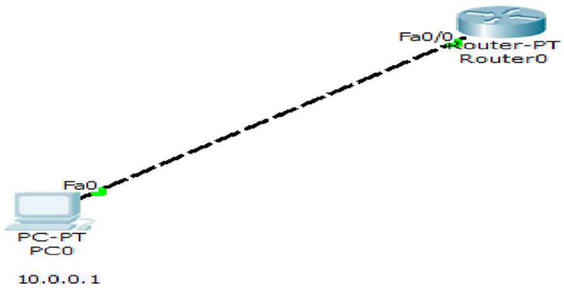
Observation:

Telnet is a text based protocol that enables remote communication over TCP/IP networks. It allows the execution of commands on a remote device, often used for initial setup ~~or~~ your management.

In the experiment, we see that all configs and commands executed via Telnet mirrored those done directly on the router but from PC interface instead. Disadvantage is that Telnet lacks encryption making it less secure compared to SSH.

30/10/24

Screen Shots:



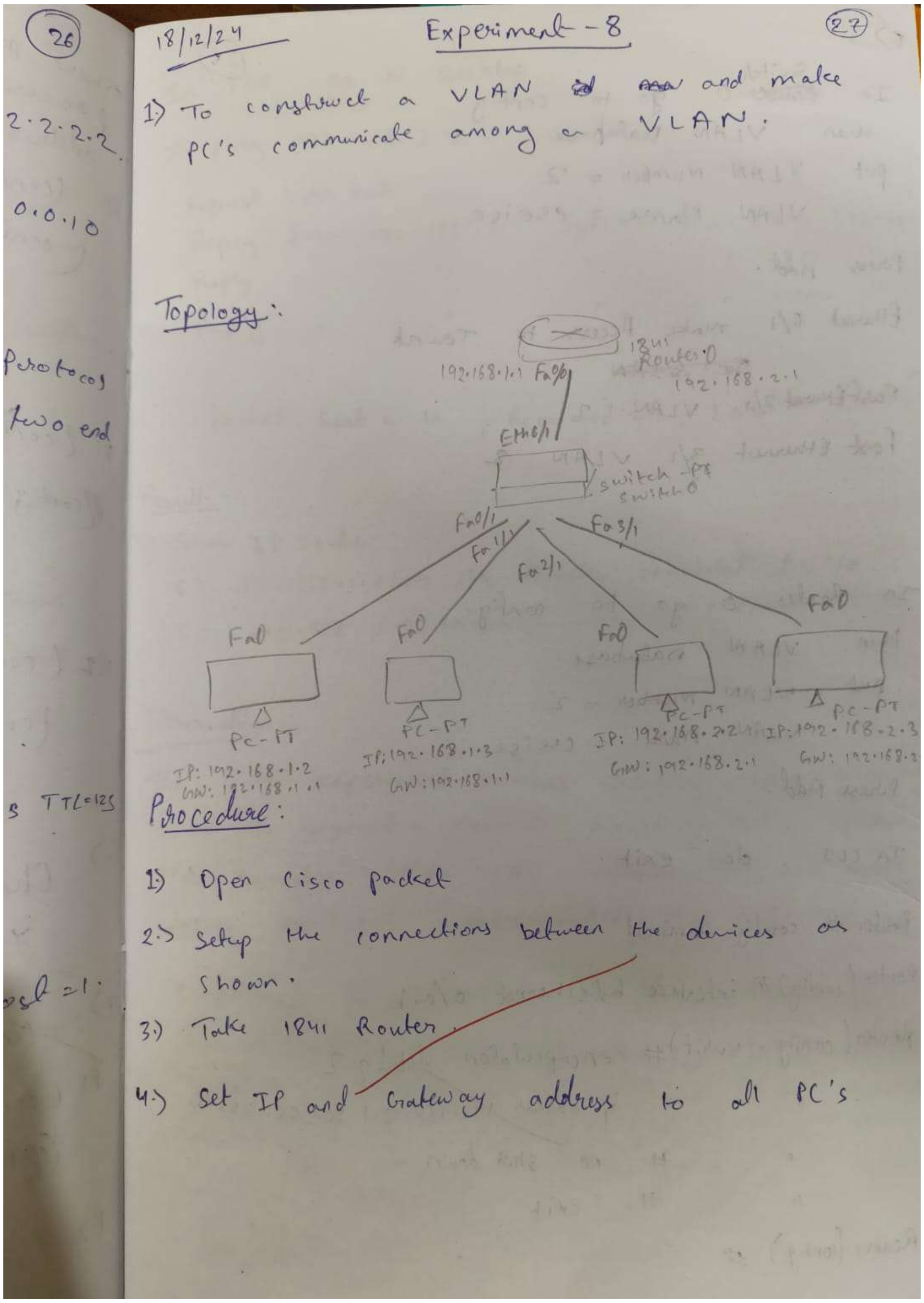
Command Prompt

```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255
Reply from 10.0.0.2: bytes=32 time=0ms TTL=255

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>telnet 10.0.0.2
Trying 10.0.0.2 ...Open


User Access Verification

Password:
R1>enable
Password:
R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    10.0.0.0/8 is directly connected, FastEthernet0/0
R1#
```

Program 12

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

Topology , Procedure and Observation:


18/12/24

Experiment - 8

1.) To construct a VLAN and make PC's communicate among a VLAN.

Topology:

Procedure:

1.) Open Cisco packet

2.) Setup the connections between the devices as shown.

3.) Take 1841 Router

4.) Set IP and Gateway address to all PC's

5.)

In ~~Router~~ Switch 0 go to config
then VLAN Database
put VLAN Number = 2
VLAN Name = cseise

Press Add.

Ethernet 6/1 make Access to Trunk

~~So VLAN~~

Fast Ethernet 2/1 VLAN 2

Fast Ethernet 3/1 VLAN 3

In Router 0 go to config
then VLAN Database
put VLAN Number = 2
VLAN Name = cseise

Press Add.

In CLI, do exit.

Router # config terminal

Router(config)# interface fastethernet 0/0.1

Router(config-subif)# encapsulation dot1q 2

" " # ip address 192.168.2.1 255.255.255.0

" # no shut down

" # exit

Router(config) #

29

In PC0 go to Desktop

>> ping 192.168.2.2

Request time out

Reply from 192.168.2.2: byte = 32 time = 3ms TTL = 127

Reply " " " " = 0ms "

Reply " " " " = 0ms "

Packet Sent = 4, Received = 3, Loss = 1 (25% loss)

Result:

Show IP route

C: 192.168.1.0/24 is directly connected Fa0/0

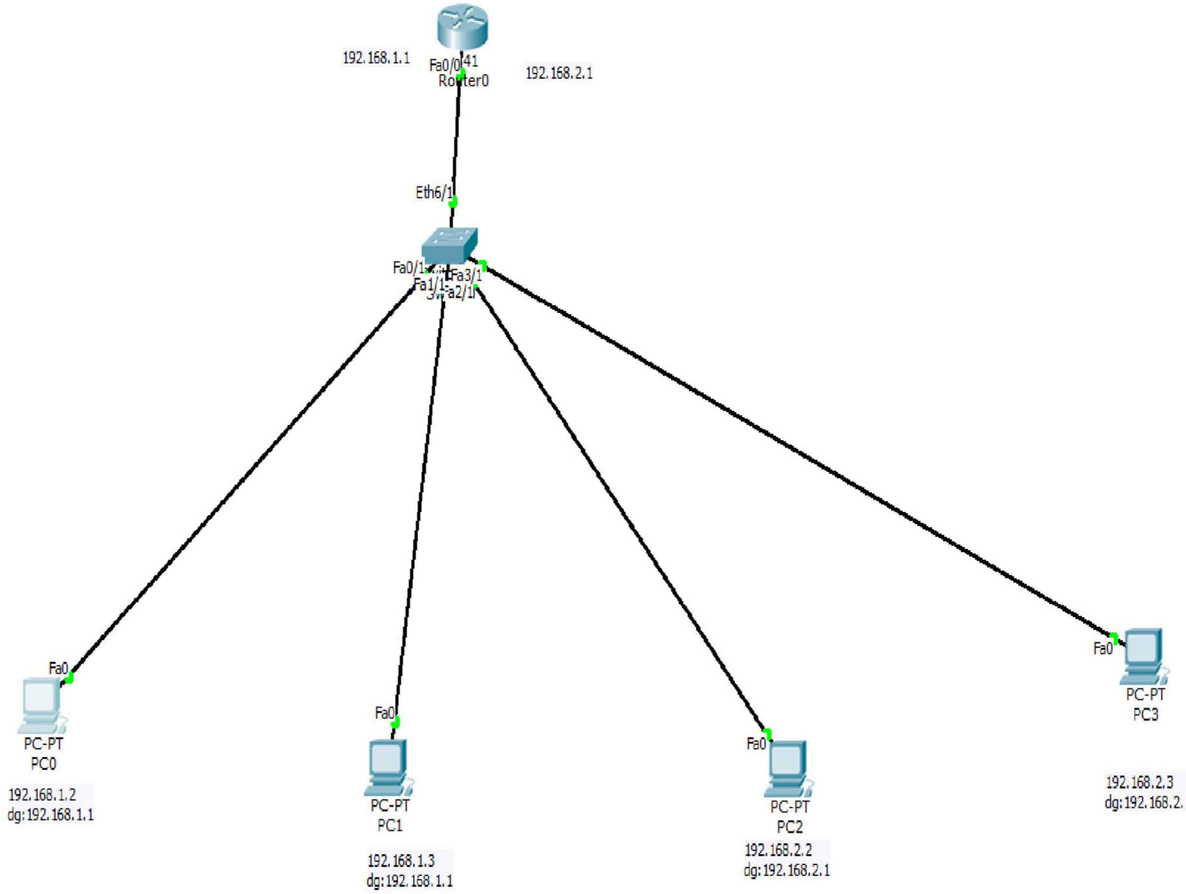
C: 192.168.2.0/24 ——— Fa 0/0.1.

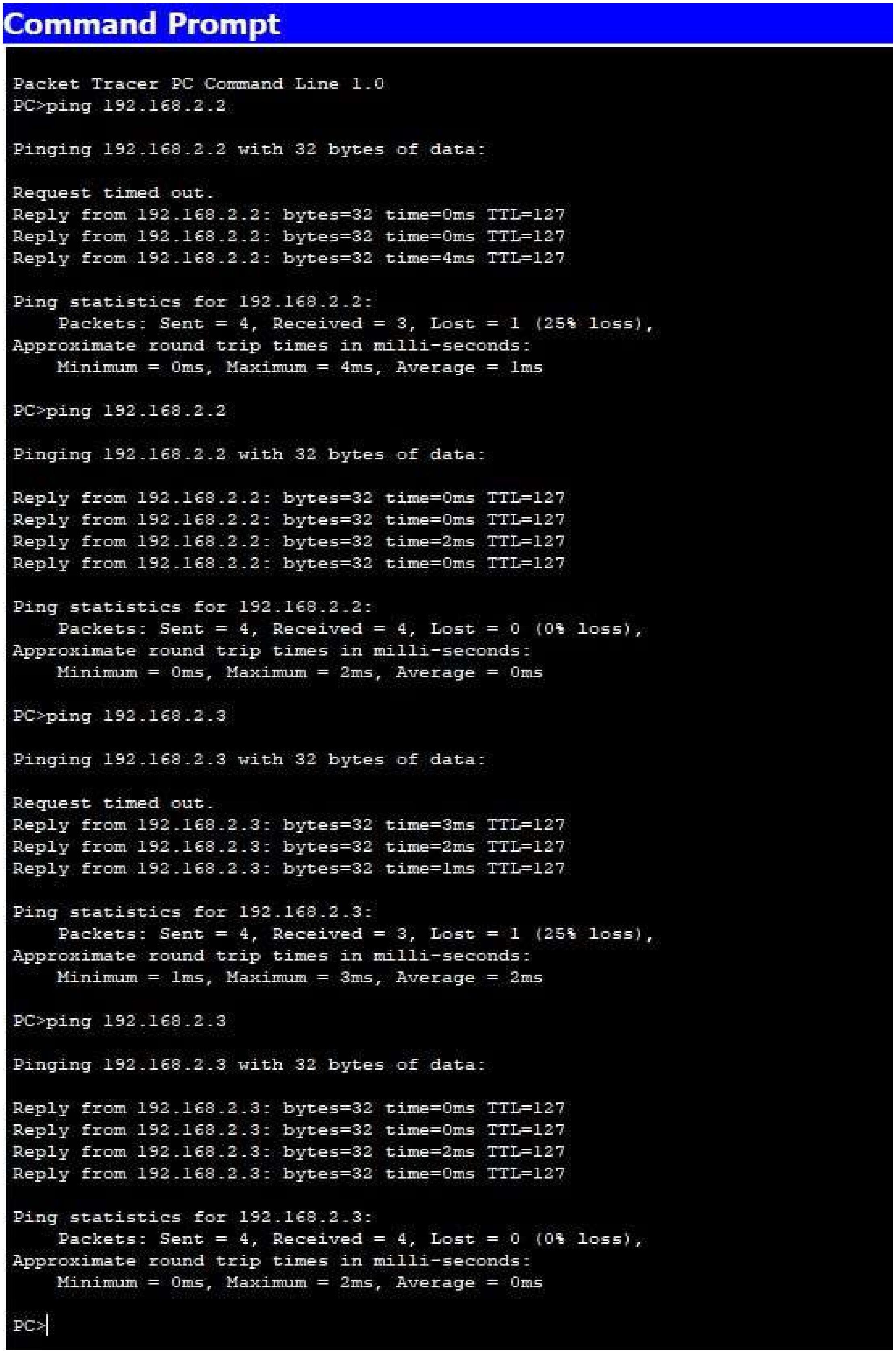
Observation:

The VLAN experiment involves creating and configuring VLAN to segment a network, assigning IP's to device for seamless intra VLAN communication and using ~~dot1q~~ dot 1 q 2, encapsulation for inter VLAN connectivity to communicate through a single trunk link. This experiment highlights the importance of VLAN's in optimizing and managing modern network effectively.

30/10/24

Screen Shots:





Command Prompt

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

Pinging 192.168.2.2 with 32 bytes of data:

Request timed out.
Reply from 192.168.2.2: bytes=32 time=0ms TTL=127
Reply from 192.168.2.2: bytes=32 time=0ms TTL=127
Reply from 192.168.2.2: bytes=32 time=4ms TTL=127

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 4ms, Average = 1ms

PC>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Reply from 192.168.2.2: bytes=32 time=0ms TTL=127
Reply from 192.168.2.2: bytes=32 time=0ms TTL=127
Reply from 192.168.2.2: bytes=32 time=2ms TTL=127
Reply from 192.168.2.2: bytes=32 time=0ms TTL=127

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms

PC>ping 192.168.2.3

Pinging 192.168.2.3 with 32 bytes of data:

Request timed out.
Reply from 192.168.2.3: bytes=32 time=3ms TTL=127
Reply from 192.168.2.3: bytes=32 time=2ms TTL=127
Reply from 192.168.2.3: bytes=32 time=1ms TTL=127

Ping statistics for 192.168.2.3:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 3ms, Average = 2ms

PC>ping 192.168.2.3

Pinging 192.168.2.3 with 32 bytes of data:

Reply from 192.168.2.3: bytes=32 time=0ms TTL=127
Reply from 192.168.2.3: bytes=32 time=0ms TTL=127
Reply from 192.168.2.3: bytes=32 time=2ms TTL=127
Reply from 192.168.2.3: bytes=32 time=0ms TTL=127

Ping statistics for 192.168.2.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms

PC>
```

Program 13

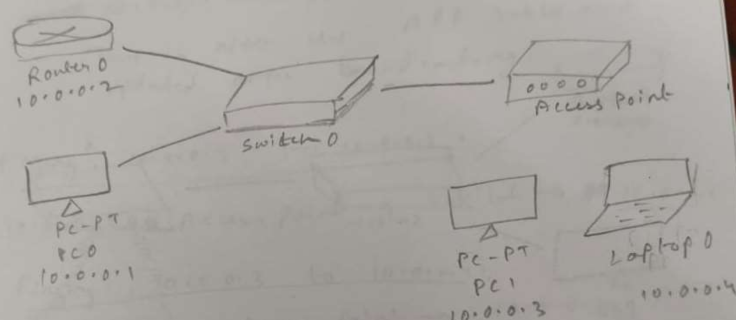
Aim: To construct a WLAN and make the nodes communicate wirelessly.

Topology , Procedure and Observation:

18/12/24 Experiment -12 WLAN (37)

1.) To conduct a wireless LAN and make nodes communicate wirelessly.

Topology :



Procedure :

1.) Create a topology as given above and configure the device

2.) Configure the access point.

Click Access Point → config → Port 1 :

SSID : bmsce

Select : WEP

Set key : 0123456789 0

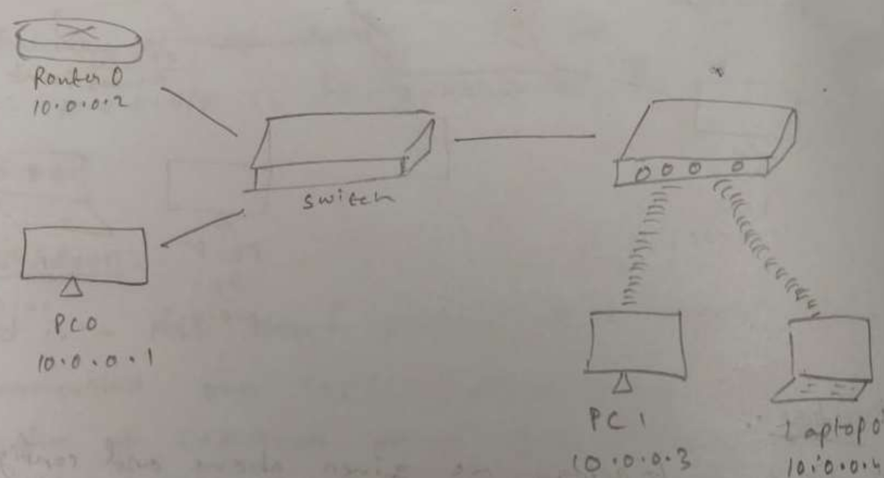
3.) Configuring PC1 and laptop with wireless standard

- Switch off device
- Drag the existing PT-HOST-NM-1AM to component listed in the LHS
- Drag WMP300N wireless interface to the empty port
- Switch on the device.

4.) In the configure take a new wireless interface was added.

5.) configure the device by entry SSID, WEP, WEP Key, IP address and Gateway.

Topology after wireless configuration.



6.) Ping from every device to every other device to check for connetion.

Observation:

1.) We were able to ping from every device to every other device

2.) Access Point:

- Creates bridge between wired and wireless devices
- SSID Broadcasting: annances the wireless network's name (SSID) to allow devices to connect using WEP, WPA or WPA2

(39)

3.) WMP 300N wireless interface :

- Wireless network adapter that enables devices to communicate with access point using wireless signals.

4.) Pinging : 10.0.0.1 to 10.0.0.3 :

~~10.0.0.1~~ 10.0.0.1 ⟶ switch → Access Point → 10.0.0.3

- This is after the ARP tables and updated after broadcasting.

5.) Pinging : 10.0.0.3 to 10.0.0.1 :

10.0.0.3 ⟶ Access point → switch → 10.0.0.1

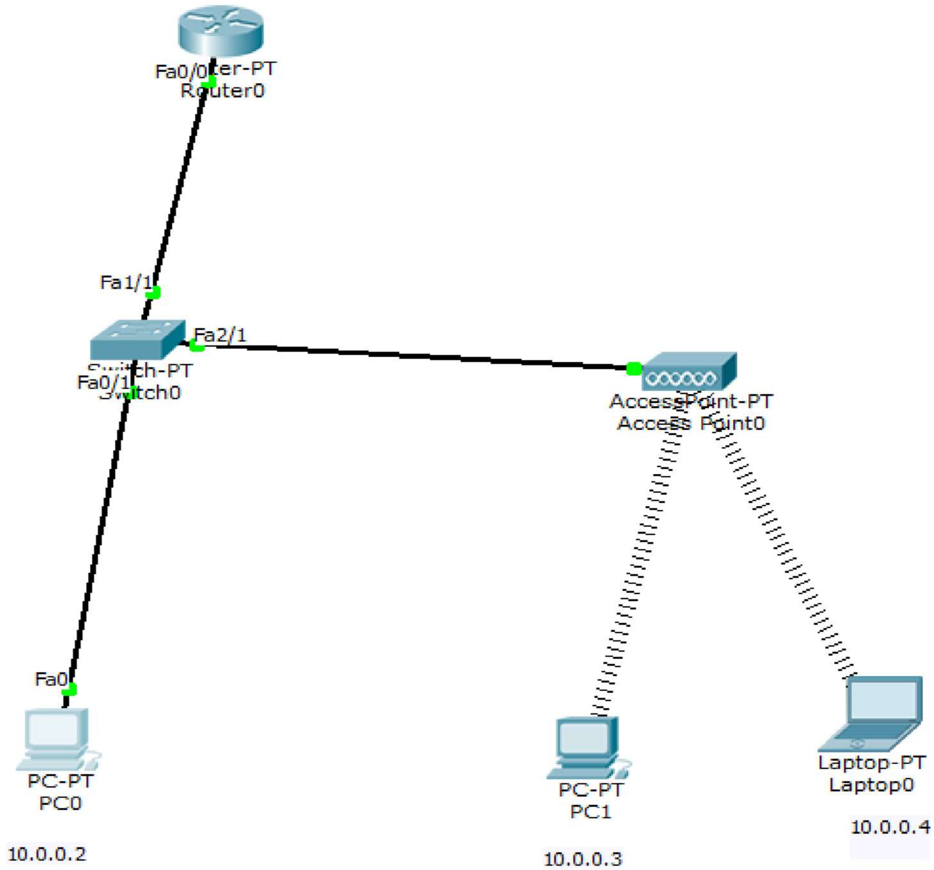
6.) Pinging : 10.0.0.3 to 10.0.0.4 :

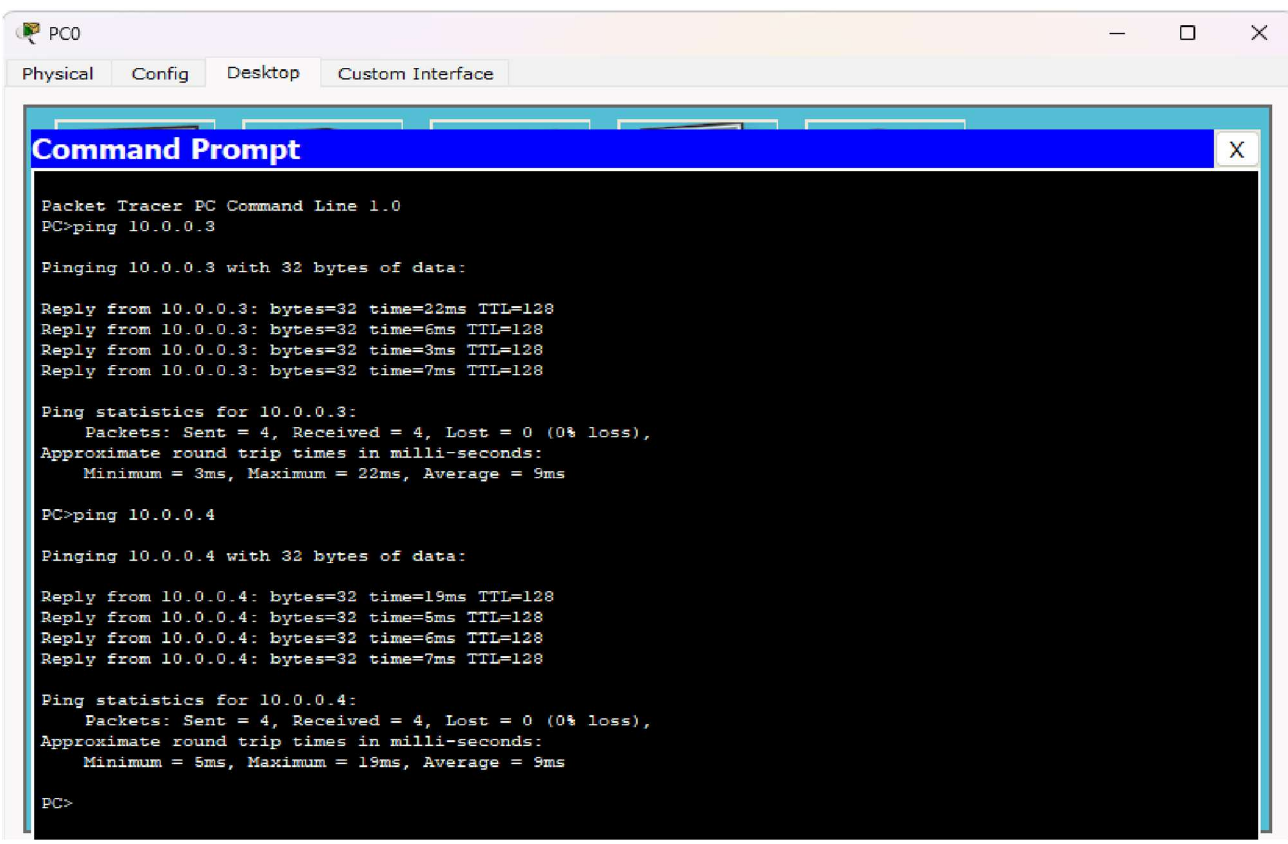
10.0.0.3 → Access point → 10.0.0.4

7.) Every device is now connected to every other device in the WLAN.

30/12/24

Screen Shots:





PART-B

Program 14

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

Code :

1/1/24

(43)

Aim : Implementation of CRC.

Code :

```
def xor(a, b):
    result = [ ]
    for i in range (1, len(b)):
        if (a[i] == b[i]):
            result.append('0')
        else:
            result.append('1')
    return ''.join(result)

def moddiv (dividend, divisor):
    pick = len[divisor]
    temp = dividend [0: pick]
    while pick < len(dividend):
    if temp[0] == '1':
        temp = xor (divisor, temp) + dividend [pick]
    else:
        temp = xor('0' * pick, temp) + dividend [pick].
    pick += 1
    if temp[0] == '1':
        temp = xor (divisor, temp)
    else:
        temp = xor ('0'* pick, temp)
    checkword = temp ~~checkword = temp~~
    return ~~...~~ checkword.
```

(44)

```
def encodeData(data, key):
    l-key = len(key)
    append-data = data + '0' * (l-key - 1)
    remainder = mod2div(append-data, key)
    codeword = data + remainder.
    print("Remainder", remainder)
    print("EncodeData (Data + remainder):",
                        codeword)

data = "100100"
key = "1101"
encodeData(data, key).
```

Output:

Sender site - - -

Remainder : 001

Encode Data (Data + Remainder) : 100100001

Receiver side - - -

Correct message received.

Output

```
Enter data: 1100110
Enter generator polynomial: 1101
CRC: 100
Transmitted Data: 1100110100
Enter received data: 1100110100
No Error


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

<u>Program 15</u>

Write a program for congestion control using Leaky bucket algorithm.

Code :

1~~0~~/12/24

Part - B

(40)

1.) Leaky Bucket Algorithm :

In the network layer before the network can make Quality of service guarantees, it must know what traffic is being guaranteed, one of the main causes of congestion is that traffic is often bursty.

There are 2 types of traffic shapping :

1.) Leaky Bucket

2) Token Bucket :

Ex: Let n = 1000

Packet = 200, 700, 500, 450, 400, 200

Since n > size of the packet at the head of the Queue i.e. n > 200.

Therefore, n = 1000 - 200 = 800

Packet size of 200 is sent into the network

200 700 500 450 400

Now again n > size of packet at the head of queue i.e. n > 400

Therefore n = 800 - 400 = 400.

Code :

```
#include <stdio.h>
int main(){
    int incoming, outgoing, bucket-size, n, store=0;
    pf(" Enter bucket size, outgoing rate and no of i/p);
    Scanf(" %d %d %d ", & buck-sz, & outgoing, &n);
}
```

```
while (n != 0 {
printf ( " Enter the incoming packet size: );
scanf ( " %d, & incoming);
printf ( " Incoming packet size %d \n", incoming);
if (incoming <= ( buck-size - store)){
    store + = incoming;
    printf (" Bucket buffer size %d out of
            %d \n", store, buck-size);
}
else {
    printf (" Dropped %d no of packets\n",
            incoming - (buck-size) - store));
    printf (" Bucket buffer size %d out
            of %d \n",
            store, buck-size);
    store = buck - size;
}
store = store - outgoing;
printf (" After outgoing %d bytes left out of %d
    in buffer \n", store, buck-size);
n -- ;
}
```

42

Output:

Enter bucket size, outgoing rate and no of inputs: 100 20 3

Enter the incoming packet size: 30

Incoming packet size: 30

Bucket buffer size 30 out of 100

After outgoing 10 bytes leftout of 100 in buffer

Enter incoming packet size : 50

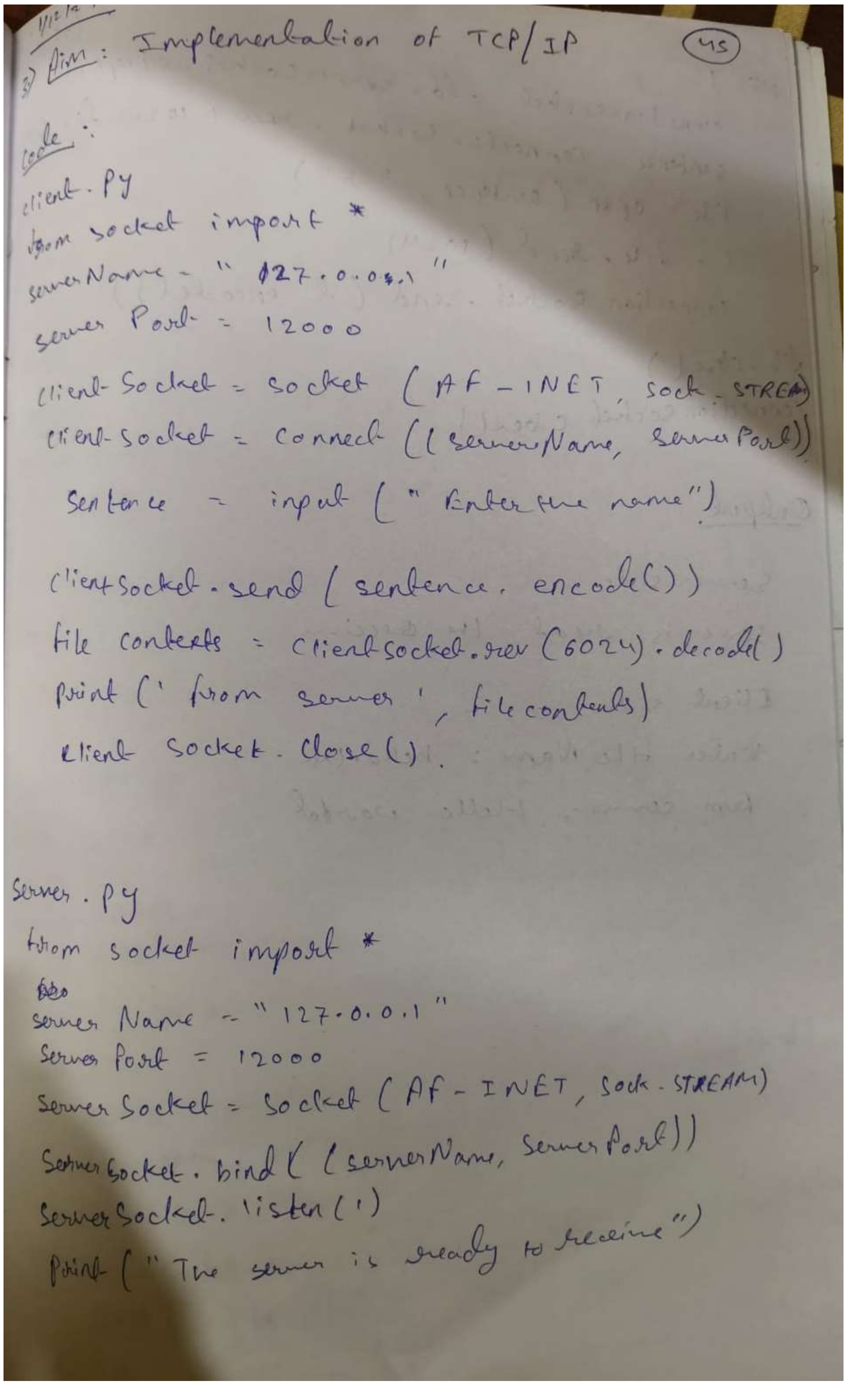
Output Clear

```
Generated packets: [80, 63, 57, 12, 69]
Enter bucket size: 60
Enter output rate: 30
Packet of size 80 bytes exceeds bucket capacity (60 bytes) - REJECTED
Packet of size 63 bytes exceeds bucket capacity (60 bytes) - REJECTED

Packet of size 57 bytes added to bucket
Bytes in bucket: 57
Transmitting 30 bytes
Bytes remaining in bucket: 27
Transmitting 27 bytes
Bytes remaining in bucket: 0

Packet of size 12 bytes added to bucket
Bytes in bucket: 12
Transmitting 12 bytes
Bytes remaining in bucket: 0
Packet of size 69 bytes exceeds bucket capacity (60 bytes) - REJECTED

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

Program 16

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

Code and Output:



1/12/14

3) Aim: Implementation of TCP/IP (45)

Code:

client.py

```
from socket import *
serverName = "127.0.0.1"
server Port = 12000
client Socket = socket (AF_INET, sock_STREAM)
client Socket = connect ((serverName, serverPort))
Sentence = input (" Enter the name")
clientSocket.send (sentence.encode())
file contents = clientsocket.recv (6024).decode()
print (' from server ', file contents)
client socket.close()
```

Server.py

```
from socket import *
server Name = "127.0.0.1"
Server Port = 12000
Server Socket = socket (AF_INET, sock_STREAM)
Server Socket.bind ((serverName, ServerPort))
ServerSocket.listen(1)
print (" The server is ready to receive")
```

(46)

```
while 1:
    connectionsocket, add = serversocket.accept()
    sentence = connection socket . recv ( 1024).decode()
    file = open ( sentence , "r")
    l = file . read (1024)
    connection socket . send ( l. encode())
file . close( )
connection socket.close()..
```

Output :

Server side ———
server is read the receive.

Client side ———
Enter file Name : hello.txt
from server : Hello world.

Program 17

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

Code and Output:

47) Aim : Implement UDP (47)

code:

```
clientUDP.py
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("Enter files name")
clientSocket.sendto(bytes(sentence, "utf-8"),
                    (serverName, serverPort))
fileContents, serverAddress = clientSocket.recvfrom(2048)
Print("From server" fileContents)
clientSocket.close().
```

```
serverUDP.py
from socket import *
serverPort = 12000
serverSockets = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
Print("The server is ready to receive")
While 1:
    sentence, clientAddress = serverSockets.recvfrom(2048)
    file = open(sentence, "r")
    l = file.read(2048)
    serverSocket.sendto(bytes(l "utf-8"), clientAddress
    Print(" sent back to client", l)
file.close()
```

(48)

Output :

Server side - - -

The server side is read to receive.

Sent back to client : hello world

Client side - - -

Enter file Name : hello.txt

from server : hello world.