VISVESVARAYATECHNOLOGICALUNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB RECORD

Computer Network Lab (23CS5PCCON)

Submitted by

CHIRAG C S (1BM22CS079)

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 **Chirag C S (1BM22CS079),** who is a bonafide student of **B.M.S. College of Engineering.** It is in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements of the above-mentioned subject and the work prescribed for the said degree.

Dr. Nandhini Vineeth Associate Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE

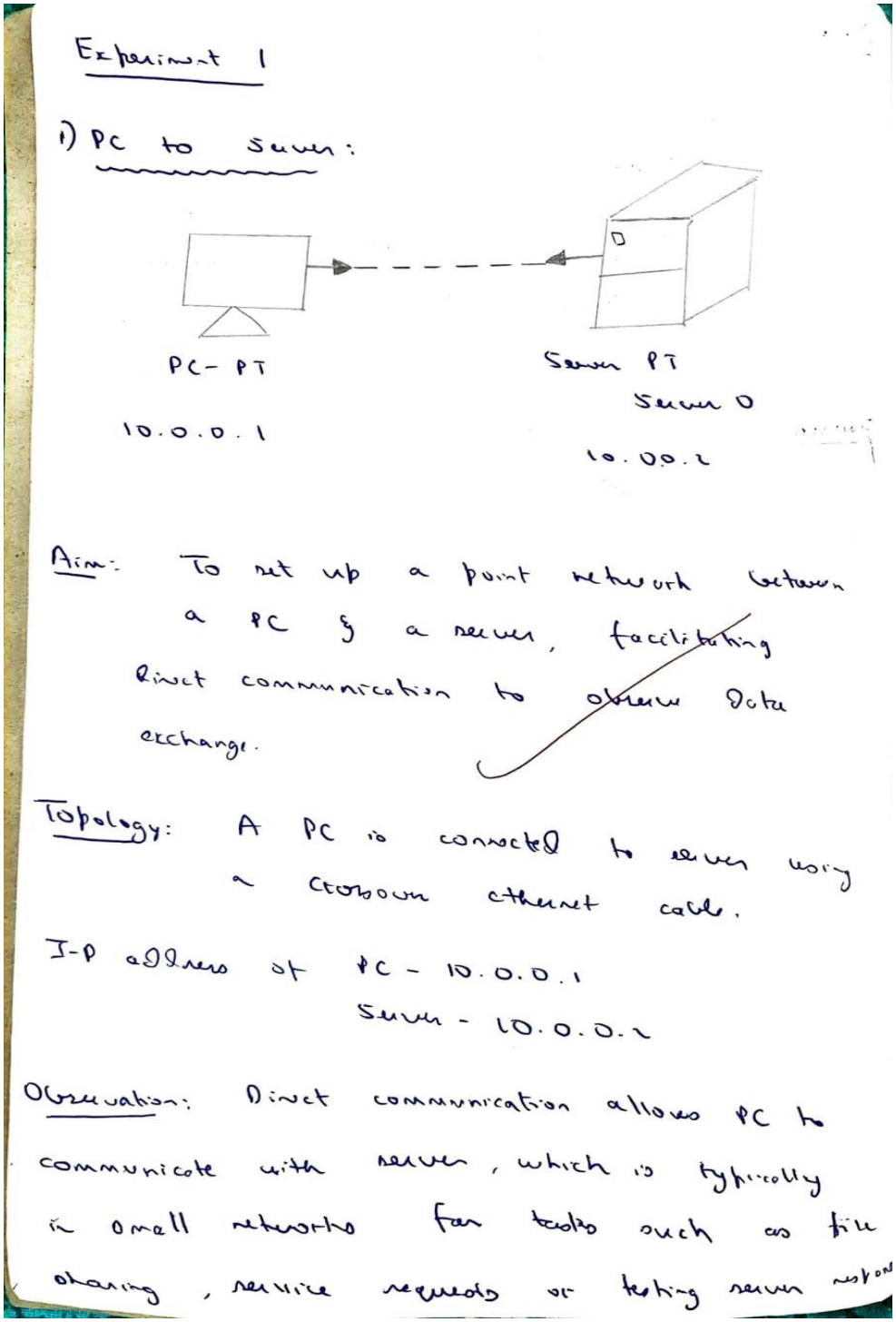
Index

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.	64-65
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.	66-67

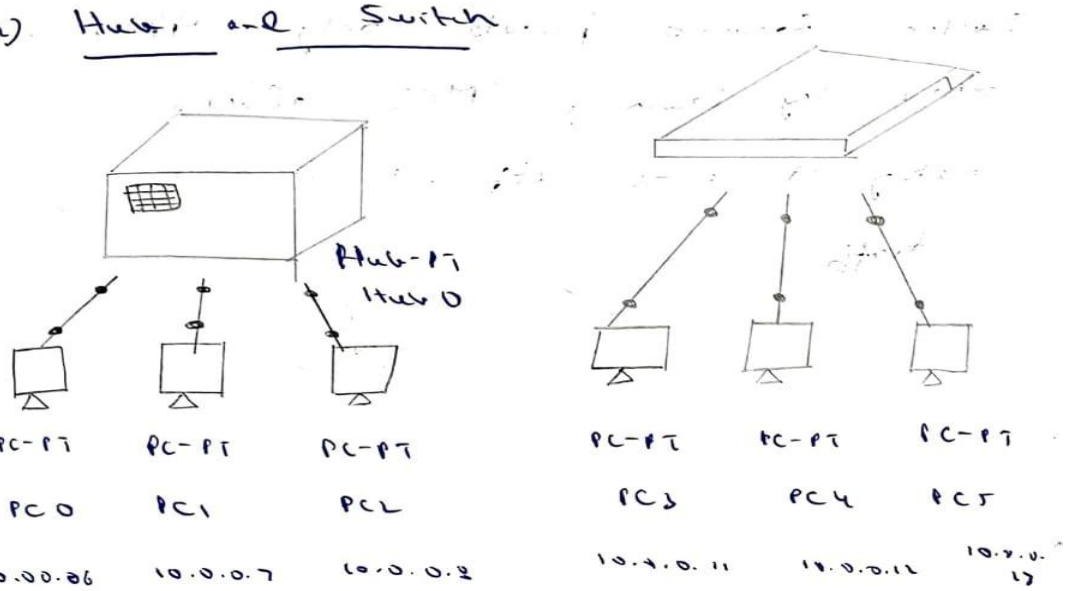
Program 1

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) Hubs and Switch.



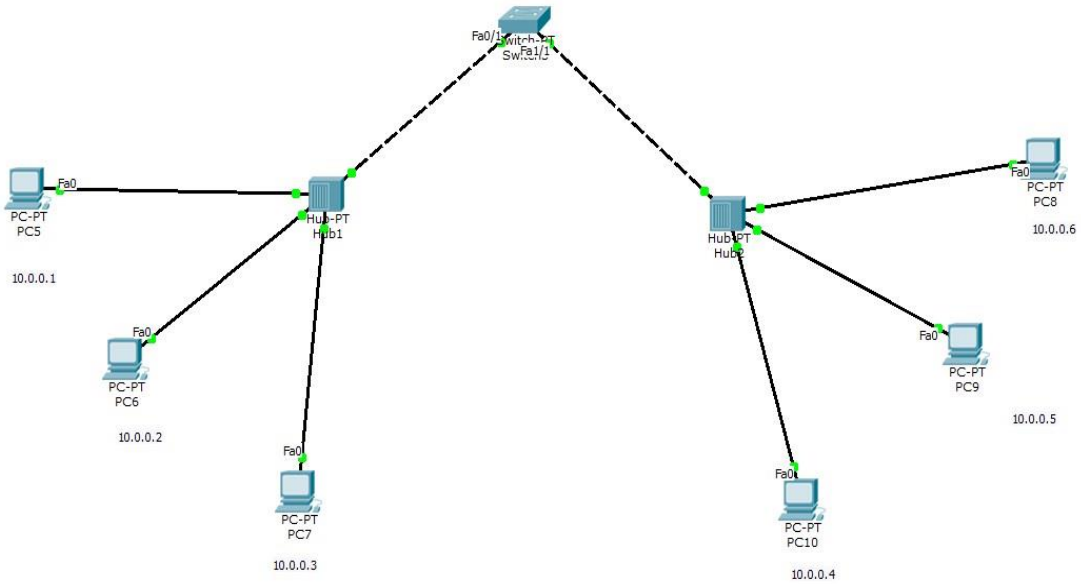
Aim: To create simple network consisting of 3 PCs connected to a central hub & another network with 3 PCs connected to a switch. This connection will help observe the behaviour of data transmission using hub & switch devices

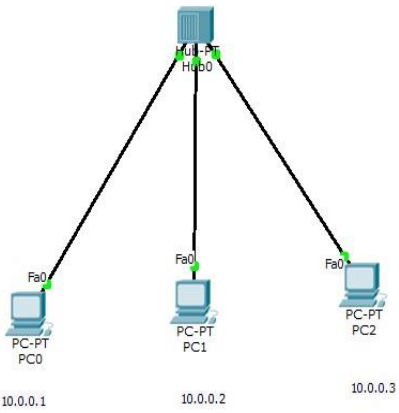
Topology: 3 PCs are connected to a hub & switch using straight through cables.

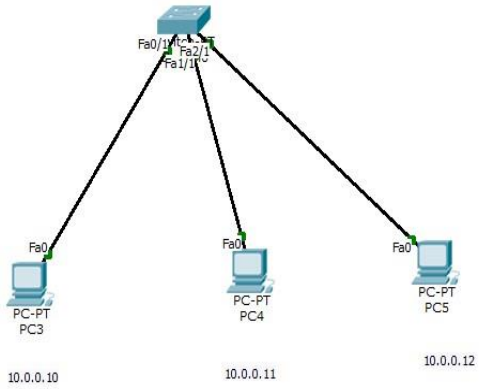
Observation: Hubs ~~transmit~~ broadcasts packets to all the devices which may cause unnecessary traffic.

Switch forwards packets only to appropriate device by learning MAC address, making it more efficient in routing 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.

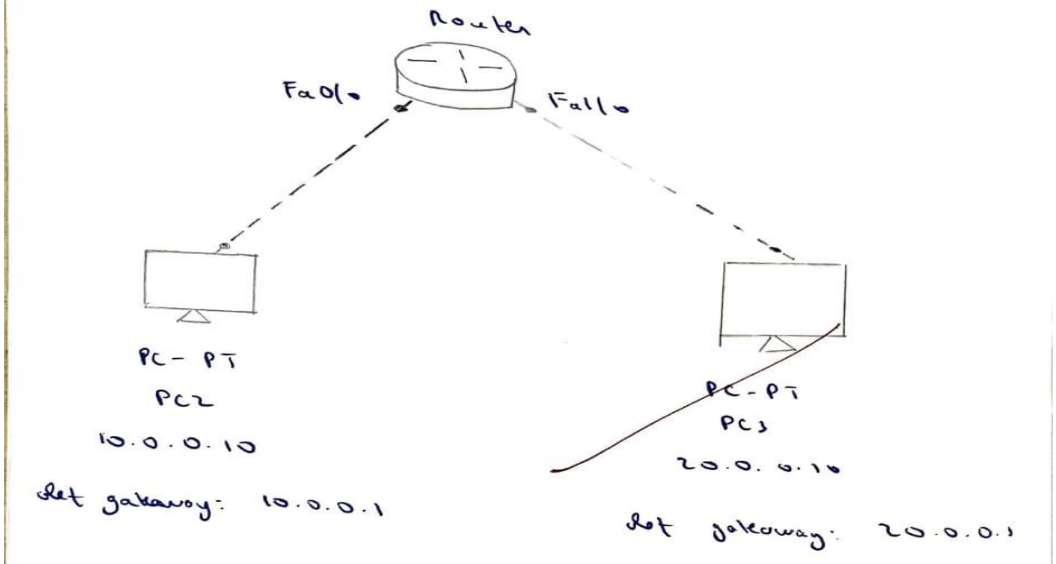
Topology , Procedure and Observation:

9/10/24

EXPERIMENT 2

Aim: To connect two different PCs on two different networks using a router

Topology:



Procedure:

1) Two PCs (PC0 & PC1) are assigned with IP 10.0.0.10 and 20.0.0.10 and gateway 10.0.0.1 & 20.0.0.1 respectively.

2) Open CLI in router & enter the following:

Router > enable

Router # config t

Router (config) # interface fastethernet 0/0

Router (config-if) # ip address 10.0.0.1 255.0.0.0

Router (config-if) # no shutdown

exit

Router (config) # interface fastethernet 1/0

Router (config-if) # ip address 20.0.0.1 255.0.0.0

Router (config-if) # no shutdown

exit.

3) Ping another system or interface from the command prompt of PC0 or PC1 using command

→ ping 10.0.0.10

Observation:

Command prompt gives output

Pinging 10.0.0.10 with 32 bytes of Data

Ping statistics for 10.0.0.10

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

Approximate round trip times in milli-seconds

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

The following type of ip route has been observed.

Router > enable

Router # show ip route

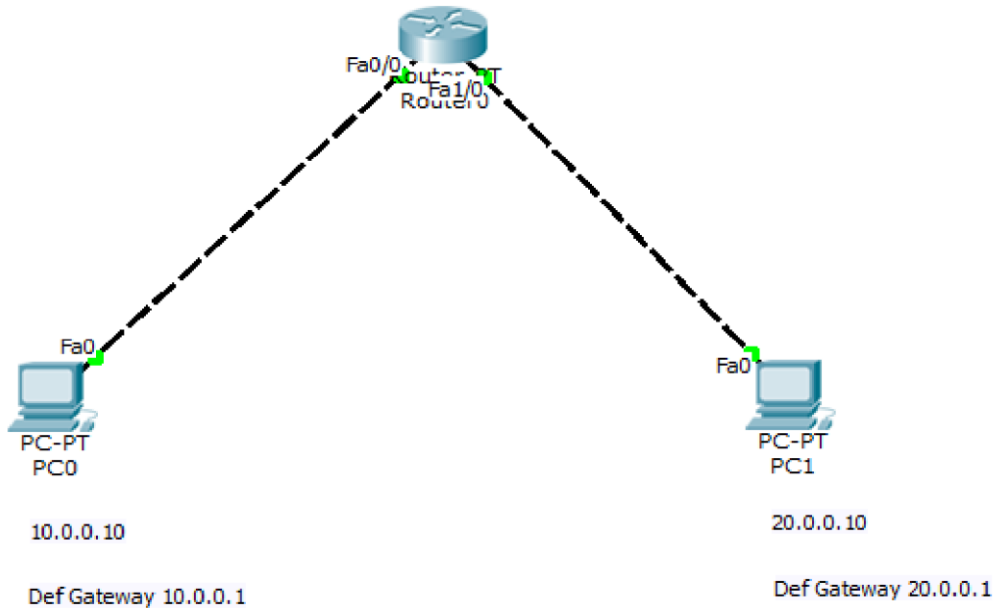
Codes: C - connected.

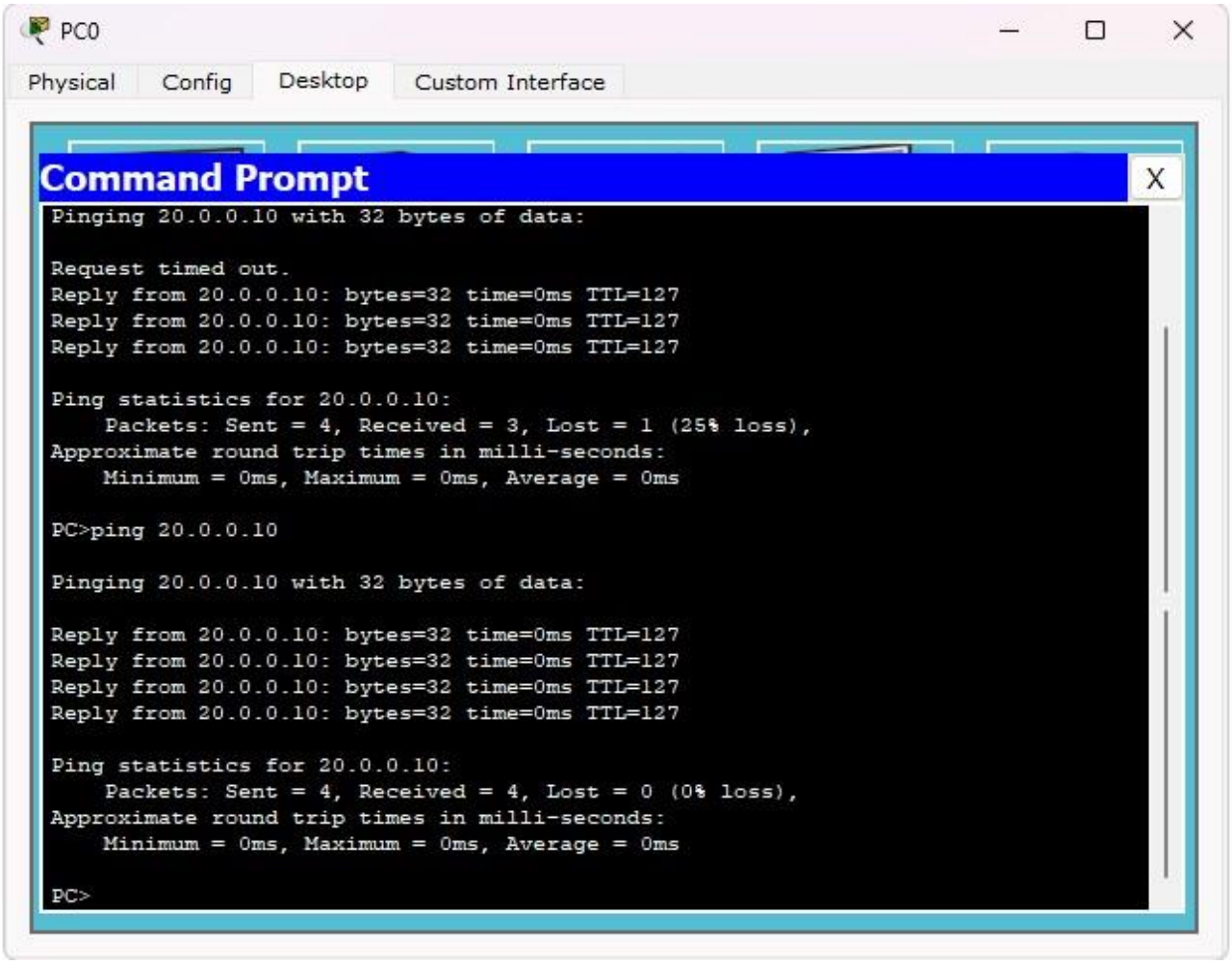
Gateway of last resort is not set

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

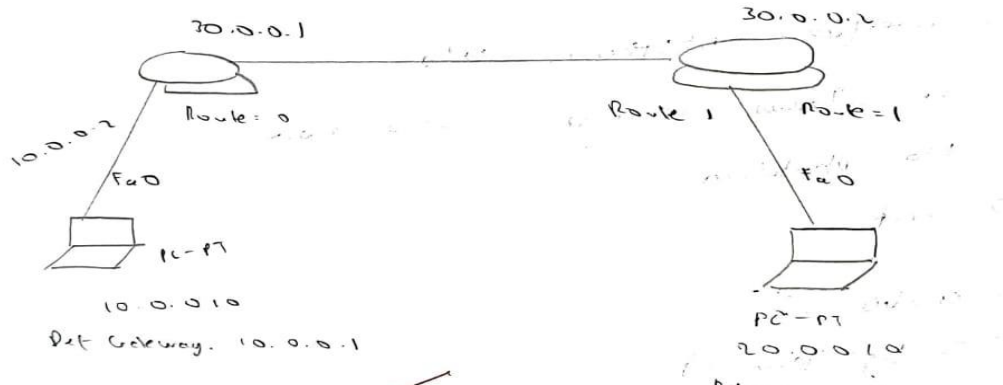
Aim: Configure default route, static route to the Router(Part 1).

Topology , Procedure and Observation:

LAB - 03

Aim: To configure IP Address to route in packet tracer. Explore following messages response destination unreachable

Topology:



Procedure:

1) Drag & drop 2 generic router and drag & drop 2 PC's

2) Connect the router and PC's using copper cross over cable

3) Connect the 2 routers using Serial DCE wire.

4) Set the gateway as 10.0.0.2 & IP address as 10.0.0.1 to PC1

5) Set the gateway as 20.0.0.2 S
IP address as 20.0.0.1 to PC 2

6) Execute the following commands in CLI
PC → (1)

→ enable
→ config terminal
→ interface fastethernet 0/0
→ IP address 10.0.0.1 255.0.0.0
→ No shutdown
→ exit.

→ enable
→ config terminal
→ interface fastethernet 1/0
→ IP address 20.0.0.1 255.0.0.0
→ no shut
→ exit.

9) Execute the following commands in CLI
exit till we get Router>
S then enable
will get →

C 10.0.0.0/8 is Directly connected
Fast ethernet 0/0.

C 30.0.0.0/8 is Directly connected
serial 2/0.

① Go to the Router → ①

§ cli § execute the following commands.

→ # interface serial 2/0

→ # ip address 30.0.0.1 255.0.0.0

→ no shut.

② Go to router → ②

§ cli § execute the following commands.

→ interface serial 2/0

→ ip address 30.0.0.2 255.0.0.0

→ no shut.

Output:

Pinging Different network

from PC (2)

Ping 10.0.0.1

Reply from 20.0.0.2 Destination host unreachable.

Ping statistic for 10.0.0.1

Packets: Sent = 4,

Received = 0

Lost = 4 (100% lost)

From PC (2)

Ping 30.0.0.1

Request time out

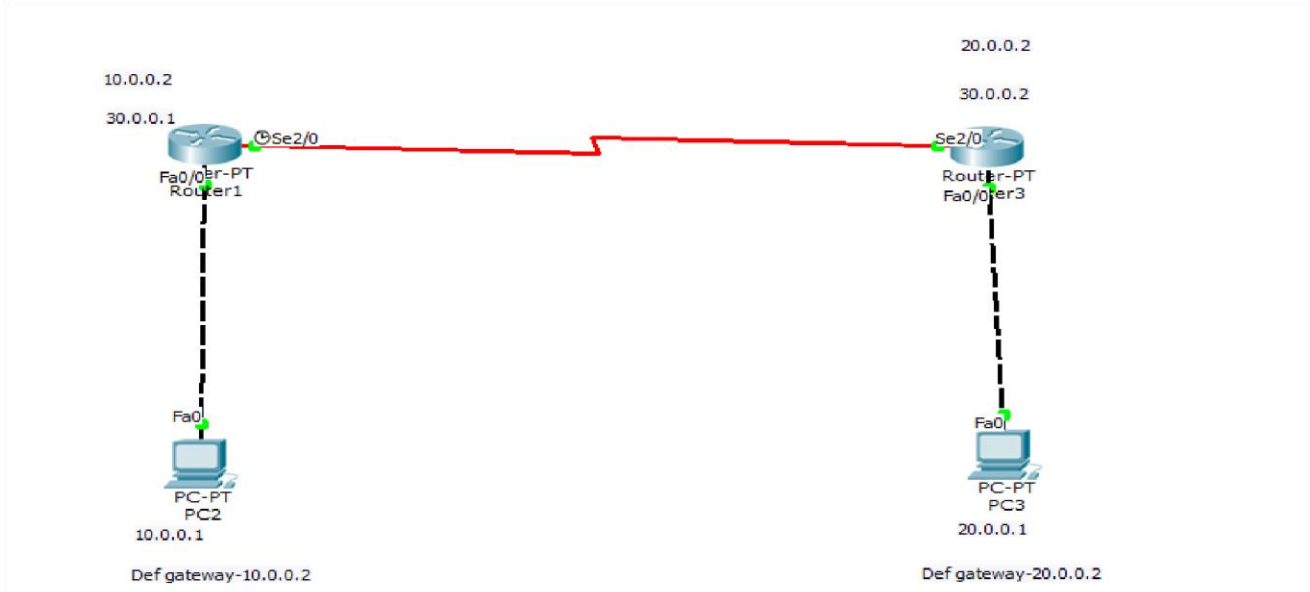
Ping status for 30.0.0.2

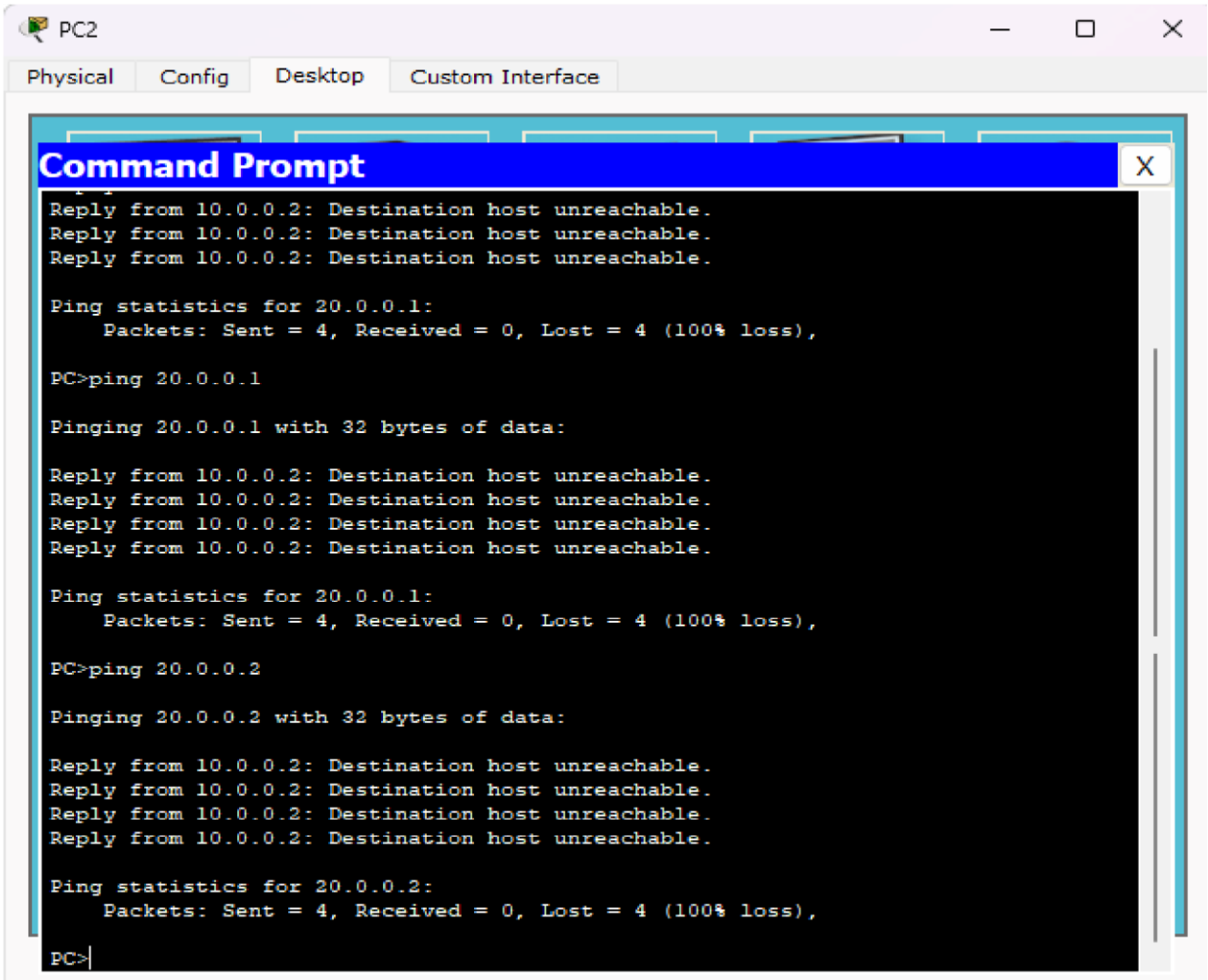
Packets: sent = 4, Received = 0, Lost = 4

(100% lost.

28/10

Screen Shots:





Program 4

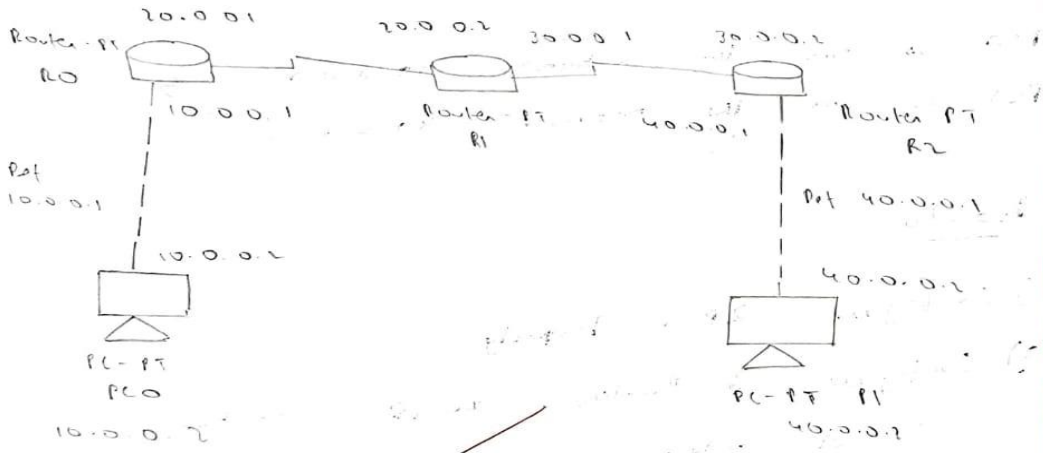
Aim: Configure default route, static route to the Router(Part 2).

Topology , Procedure and Observation:

Experiment 3

Aim: Configure Default Route, Static Route to the Router

Topology: 3 Routers with 2 PCs at opposite ends.



PC0: Fast ethernet 0/0

IP - 10.0.0.2, Def Gateway - 10.0.0.1

PC1: Fastethernet 0/0

IP - 40.0.0.2, Def Gateway - 40.0.0.1

R0:

Fastethernet 0/0 - 10.0.0.1

Serial 1/0 - 20.0.0.1

R2:

Fastethernet 0/0 : 40.0.0.1

Serial 2/0 : 30.0.0.2

R3:

Serial 2/0 : 20.0.0.2

Serial 3/0 : 30.0.0.1

PC0 is on the network 10.0.0.0 &

PC1 is on the network 40.0.0.0

Procedure:

1) Configure 2PCs properly.

2) Configure 3 Routers according to their correct interfaces - Fastethernet & Serial

3) Default route R1:

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

4) Static route R0:

ip route 0.0.0.0 0.0.0.0 20.0.0.2

5) Static route R2:

ip route 0.0.0.0 0.0.0.0 30.0.0.1

Observations:

R0:

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

R2:

show ip route

C 30.0.0.0/8 is Directly connected, Serial 2/0

C 40.0.0.0/8 is Directly connected, FastEthernet 0/0.

S* 0.0.0.0/0 (1/0) via 30.0.0.1

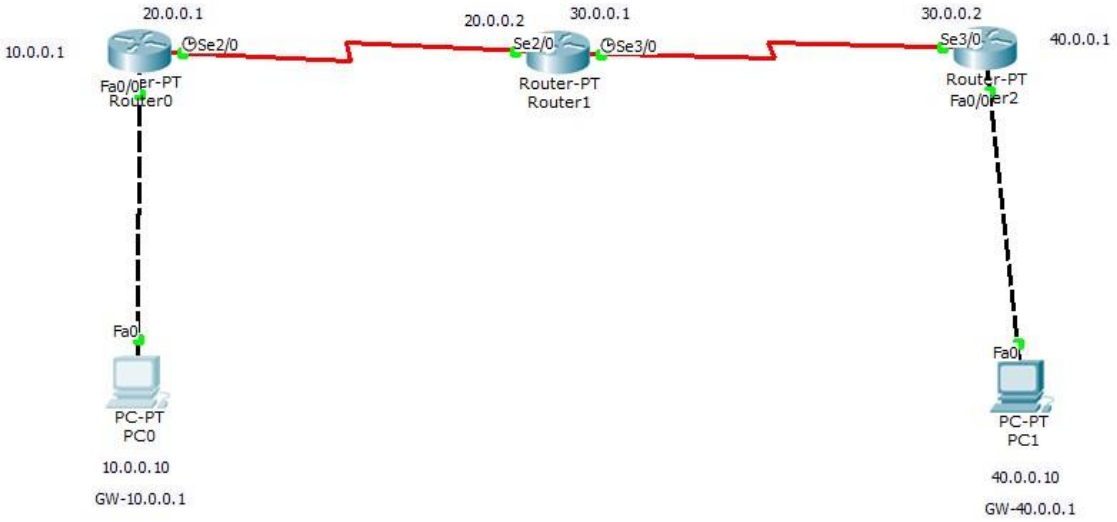
PC0

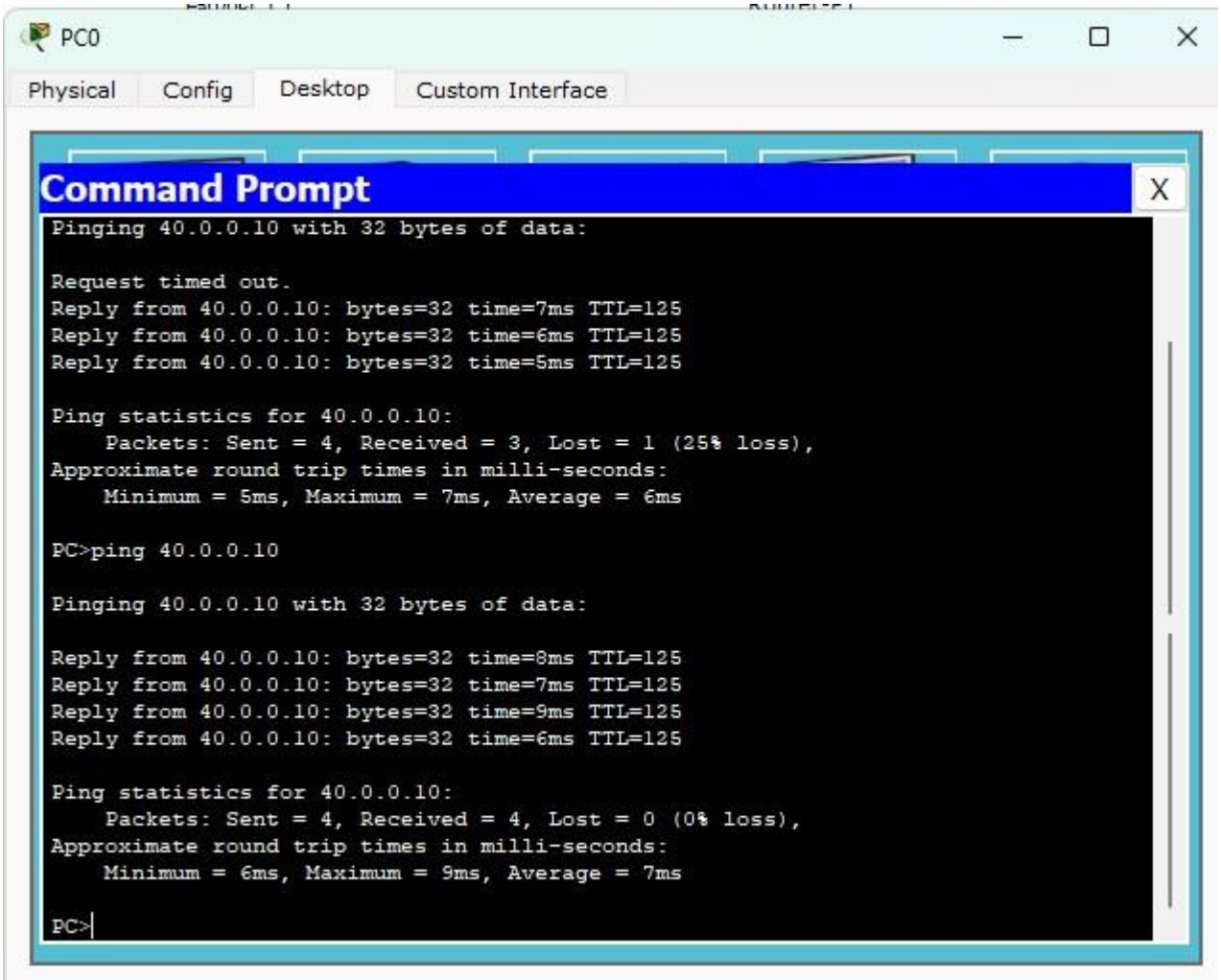
Ping 40.0.0.2

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

23/10

Screen Shots:

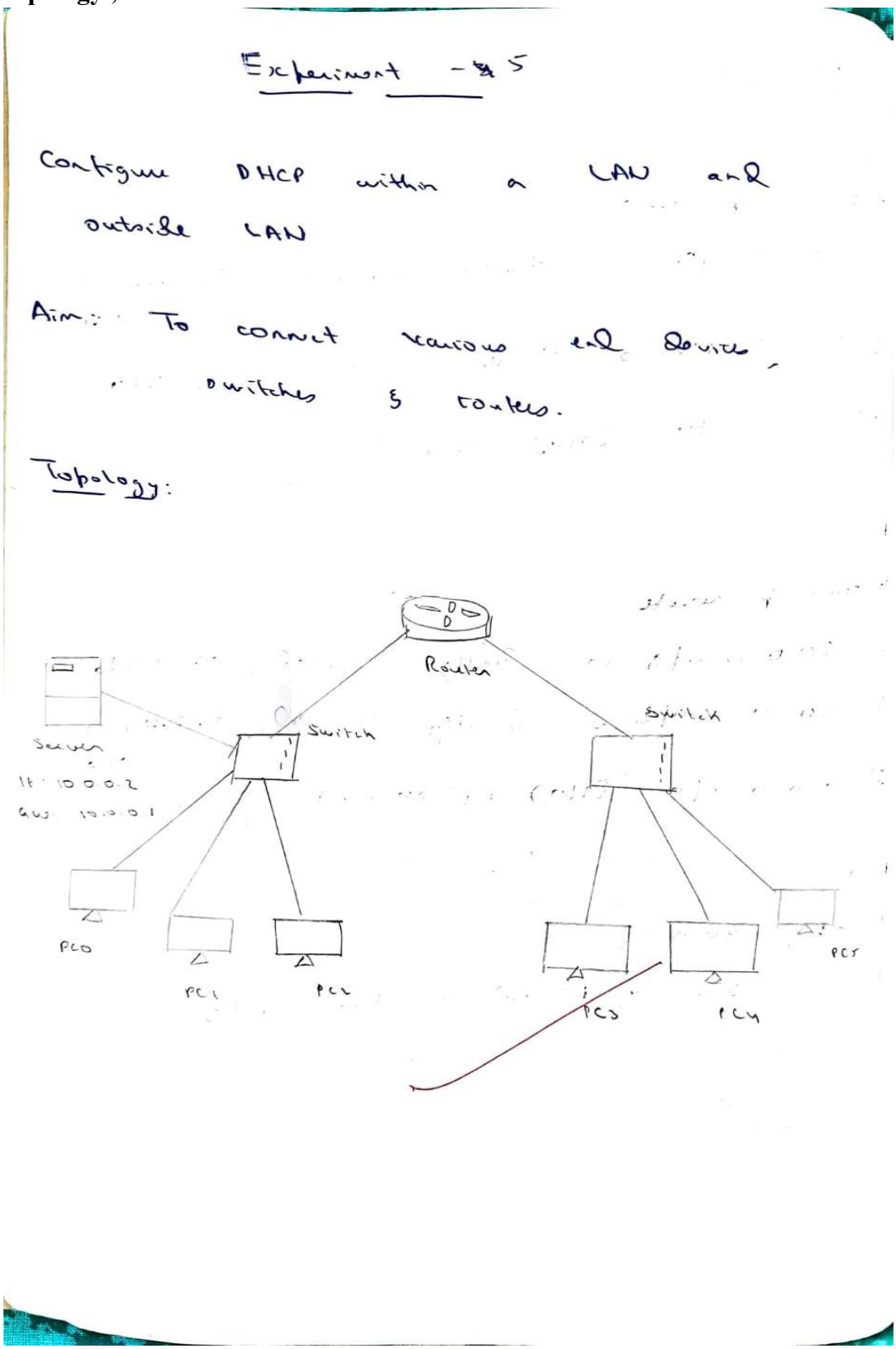




Program 5

Aim: Configure DHCP within a LAN and outside LAN.

Topology , Procedure and Observation:



Procedure:

1) Place a switch, a server, 3 PCs and configure them.

2) Construct another network similar to the above explained.

3) Under DHCP in config create a pool such that name is:

switch 1, Default Gw: 10.0.0.1 and start ip as 10.0.0.2

4) Click on DHCP under IP config in PC 0, 1, 2 & IP address will be set dynamically.

Observation:

DHCP request successful

IP address 10.0.0.2

Problem (2)

1) To enable DHCP for outside connection router is configured as such

Router Config fastEthernet 4/0

Router (config-if) # ip address 10.0.0.1
255.0.0.0

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

Router (config-if) no shut

exit

Router (config) fastEthernet 0/0

Router (config-if) address 20.0.0.1
255.0.0.0

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

Router (config-if) # no shut

exit.

2) Dynamically enable ip address for end devices

Observation:

1) Ip address dynamically set

2) On pinging network 2 devices

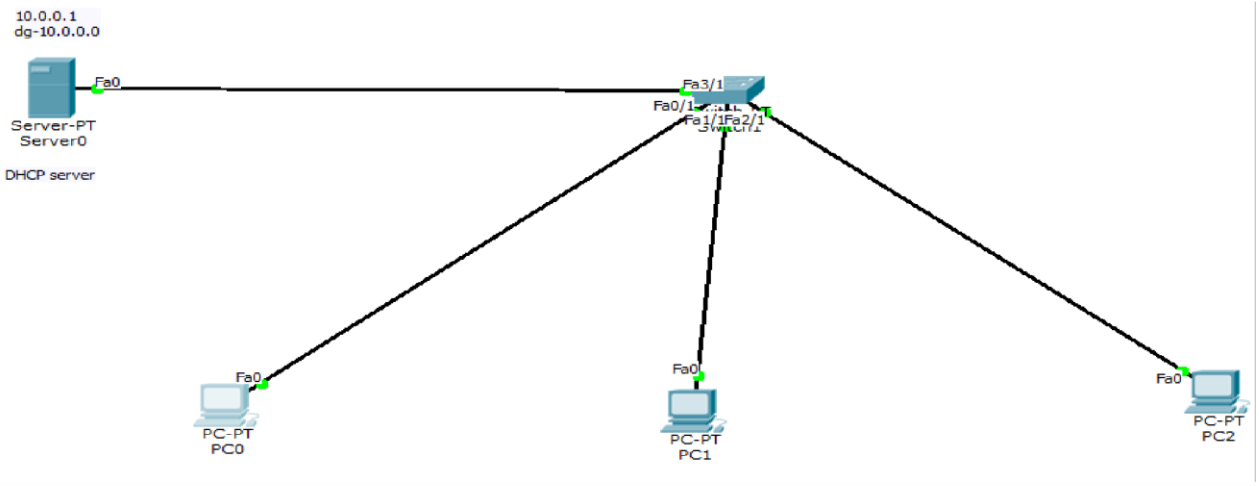
from network 1 :

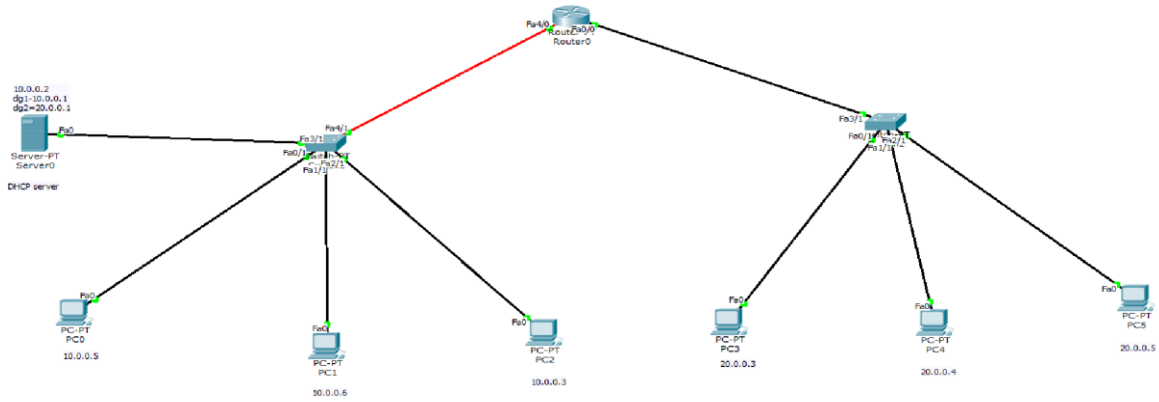
Pinging 20.0.0.9 with 32 bytes of data

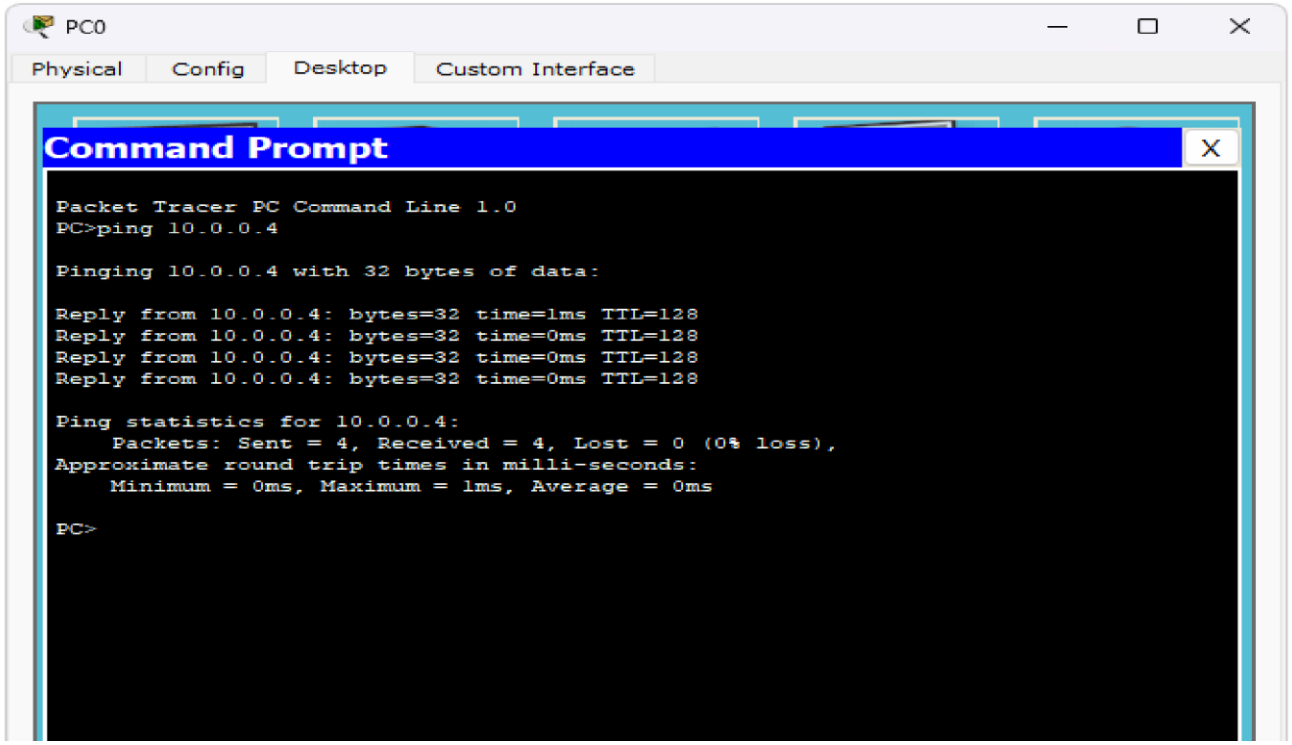
Packets : Sent = 4, Received = 4, Lost = 0 (0% lost).

13/11/23

Screen Shots:



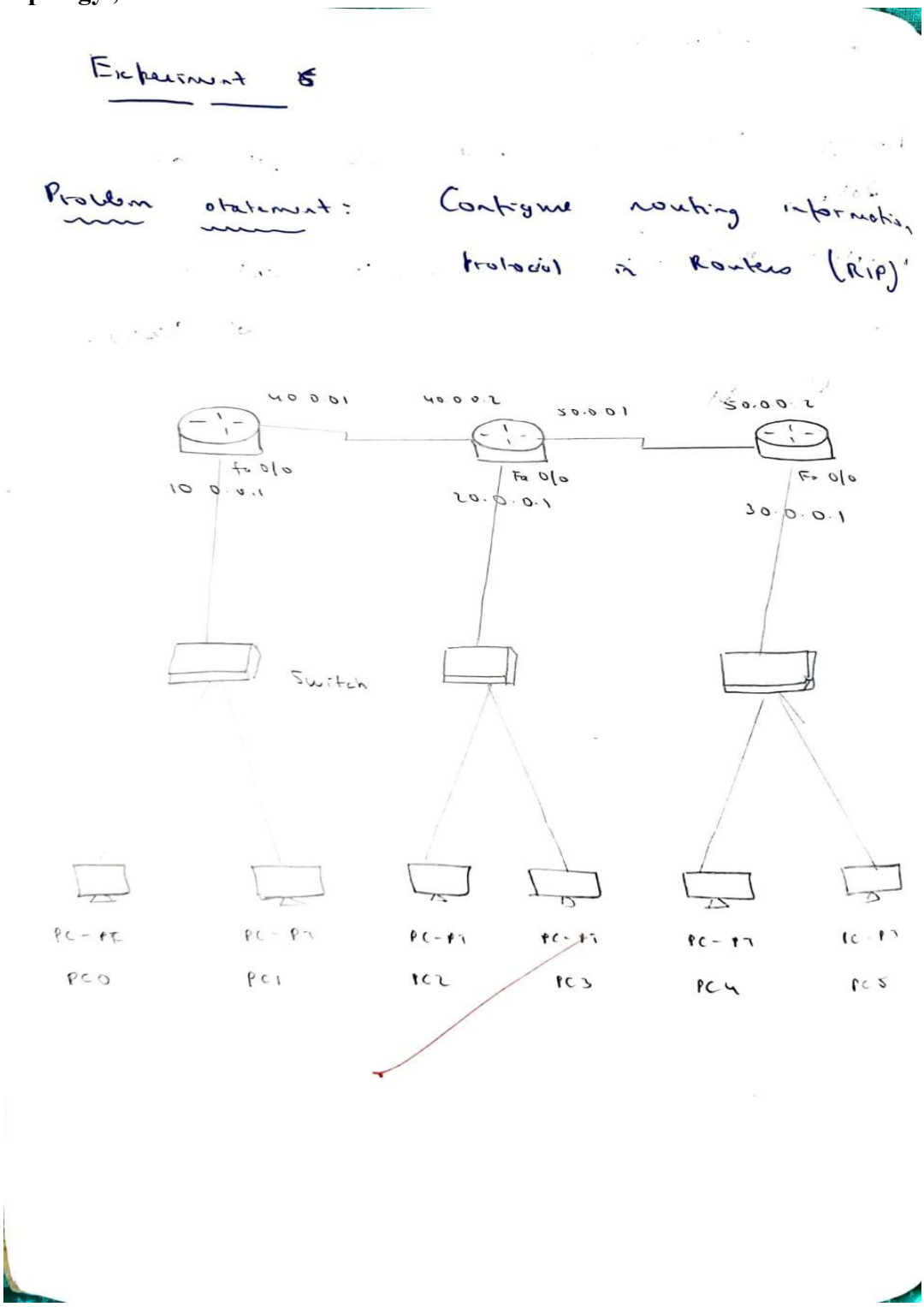




Program 6

Aim: Configure RIP routing Protocol in Routers .

Topology , Procedure and Observation:



Procedure:

1) Place 3 Routers (Generic), 3 Generic Switches to 6 PCs

2) Connect the Routers to the corresponding switches & connect 2 PCs to one switch.

3) Use Automatically choose Connection Type

4) Configure the End Device and Define gateway.

5) Configure the routers using CLI and check for Green Green Light .. for all connections.

For Router 0:

Router > enable

Router # config terminal

Router (config)# router rip

Router (config-router)# network 10.0.0.0

Router (config-router)# network 40.0.0.0

Router (config)# exit.

For Router 1:

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

For 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
Router (config)# exit
```

Demonstrate the TTL life of a packet :

Observation :

① Before routing information Protocol :

In Router 2 :

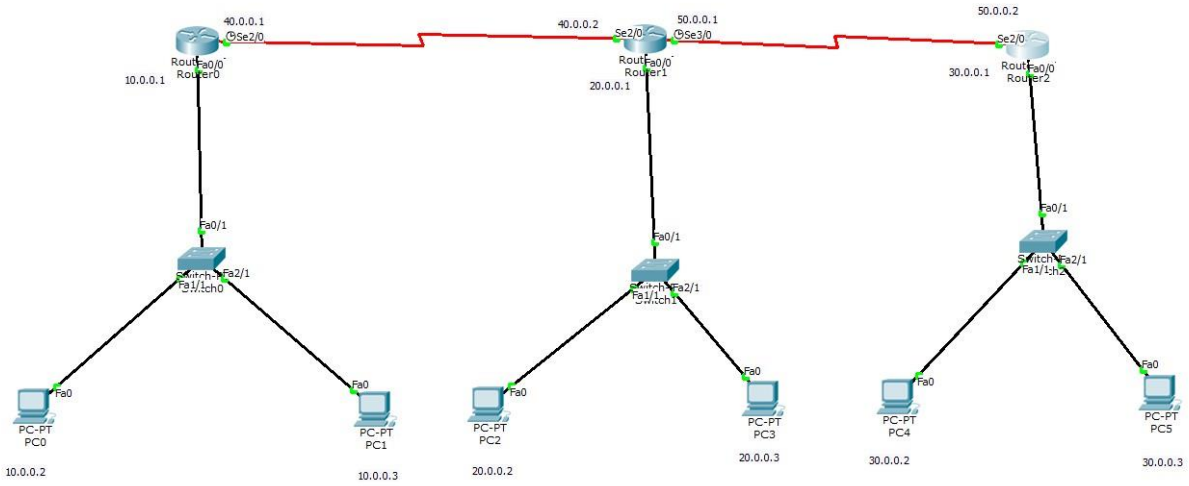
Show ip route :

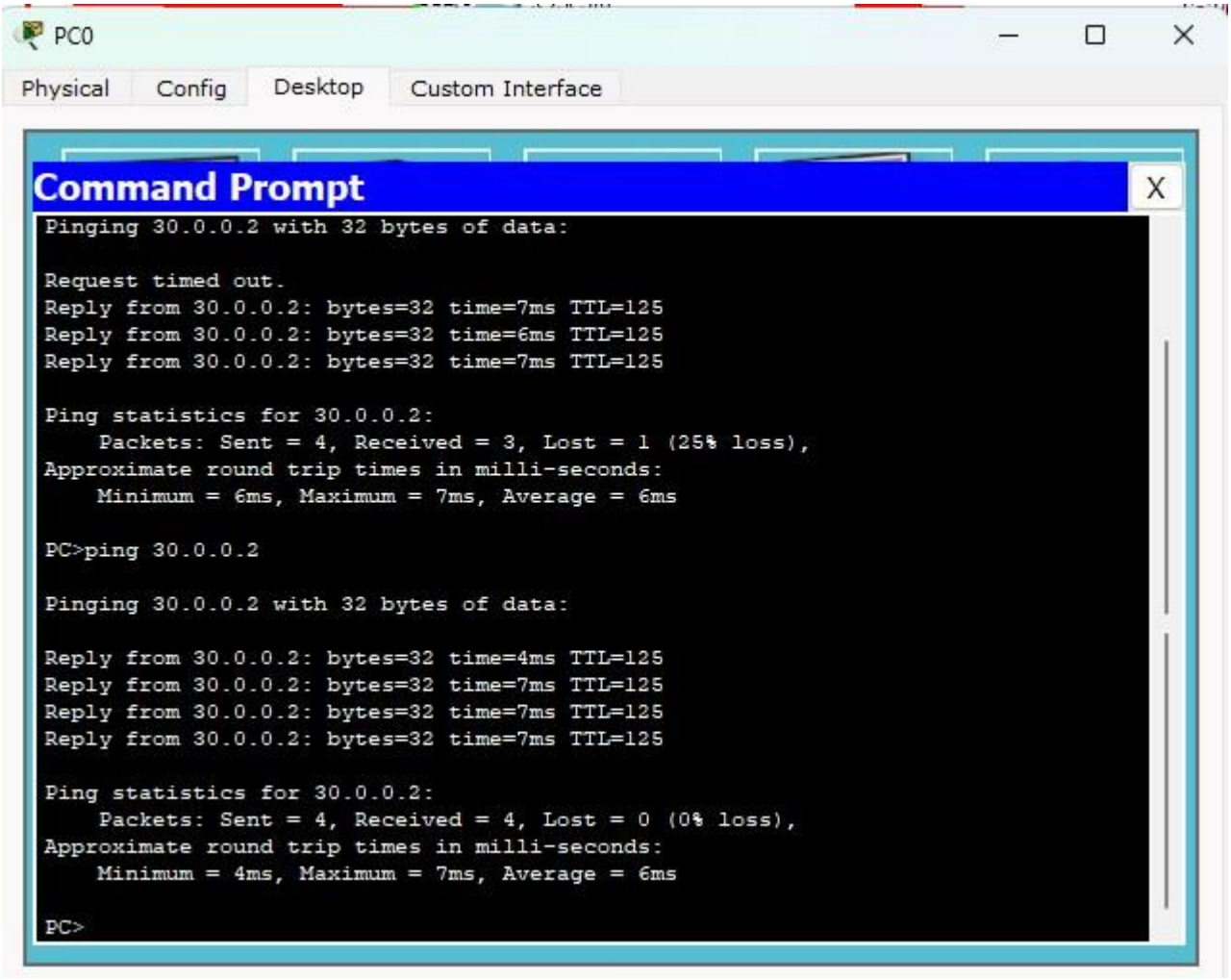
C 30.0.0.0/8 is Directly connected, Fast Ethernet 0/

C 50.0.0.0/8 is Directly connected, Serial 2/0

R 10.0.0.0/8 [120/1] via 50.0.0.1, 0

Screen Shots:

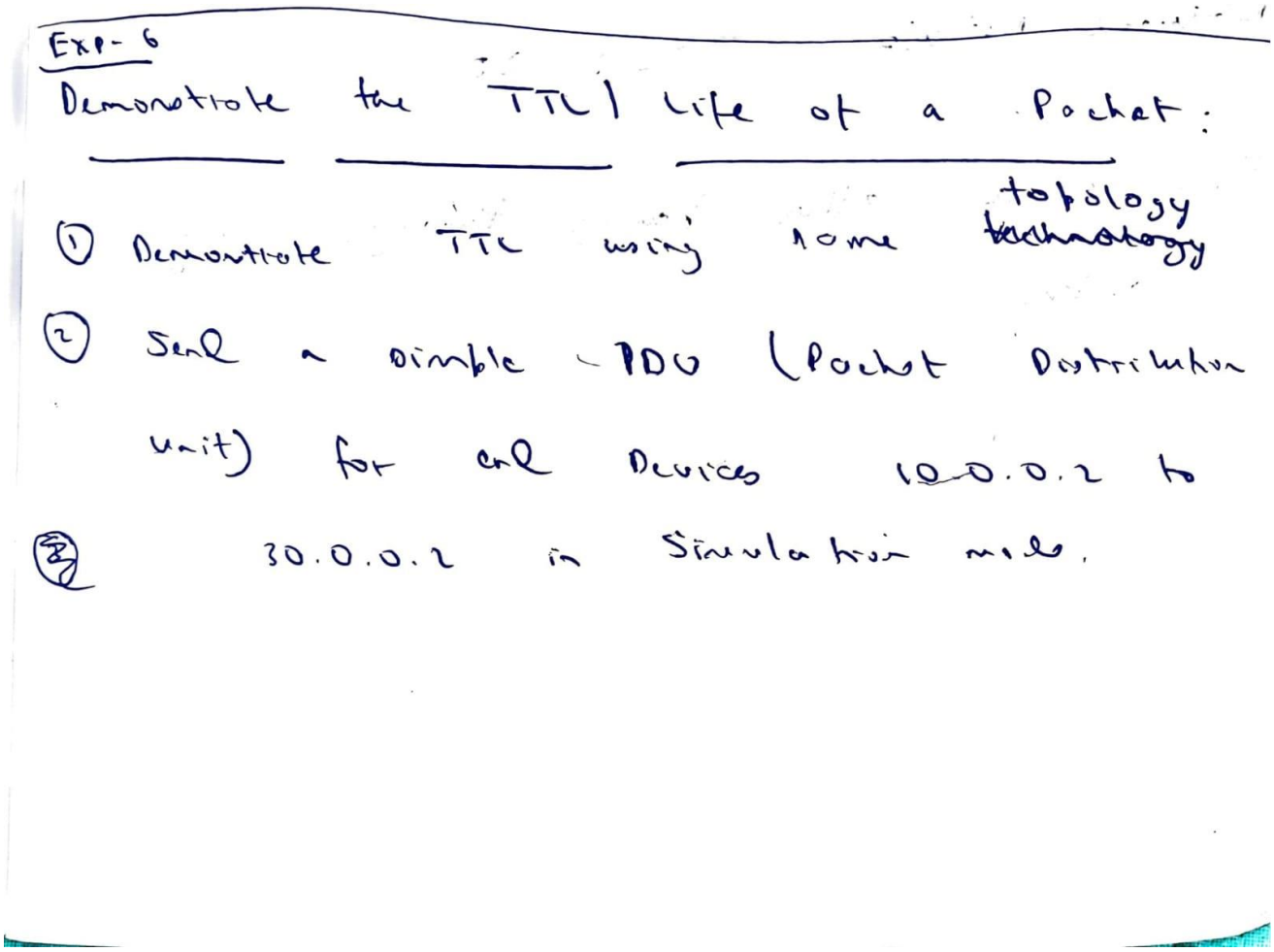




Program 7

Aim: Demonstrate the TTL/ Life of a Packet .

Topology , Procedure and Observation:



Observation:

① Router 0

Inbound Details : TTL: 255

Outbound Details : TTL: 254

② Router 1

Inbound Details : TTL : 254

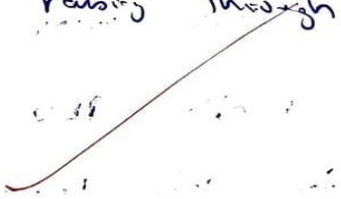
Outbound Details : TTL : 253

③ Router 2

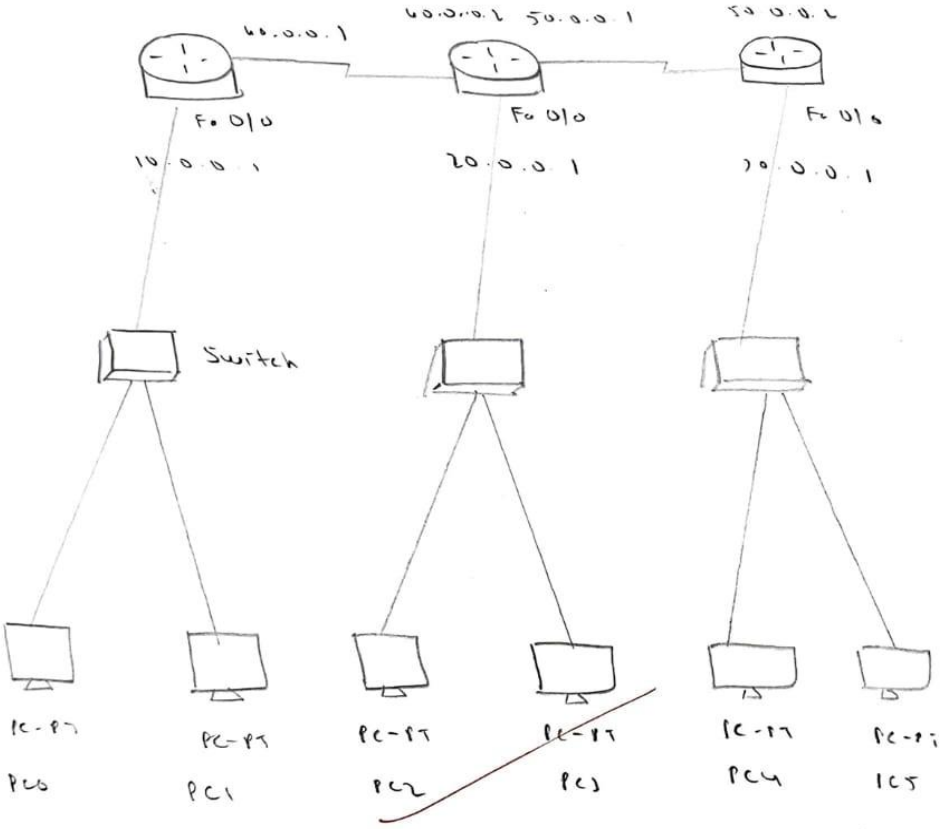
Inbound Details : TTL : 253

Outbound Details : TTL : 252

TTL values after passing through every router.



Topology:



Practice (continue)

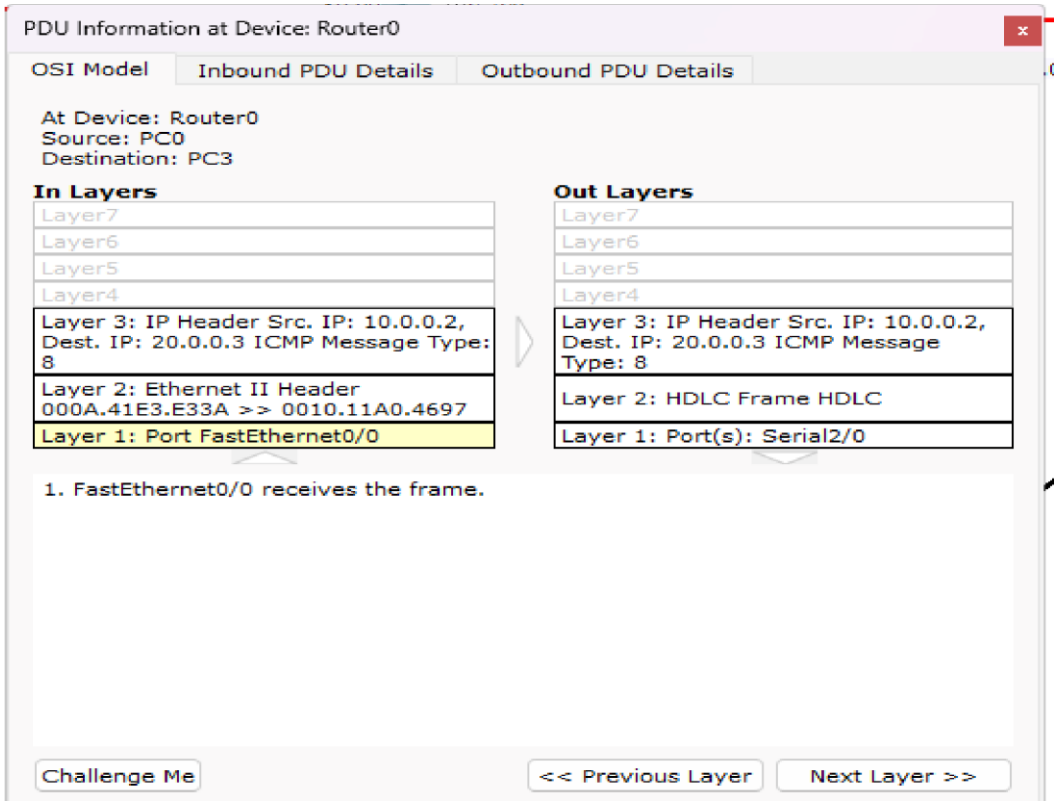
2014

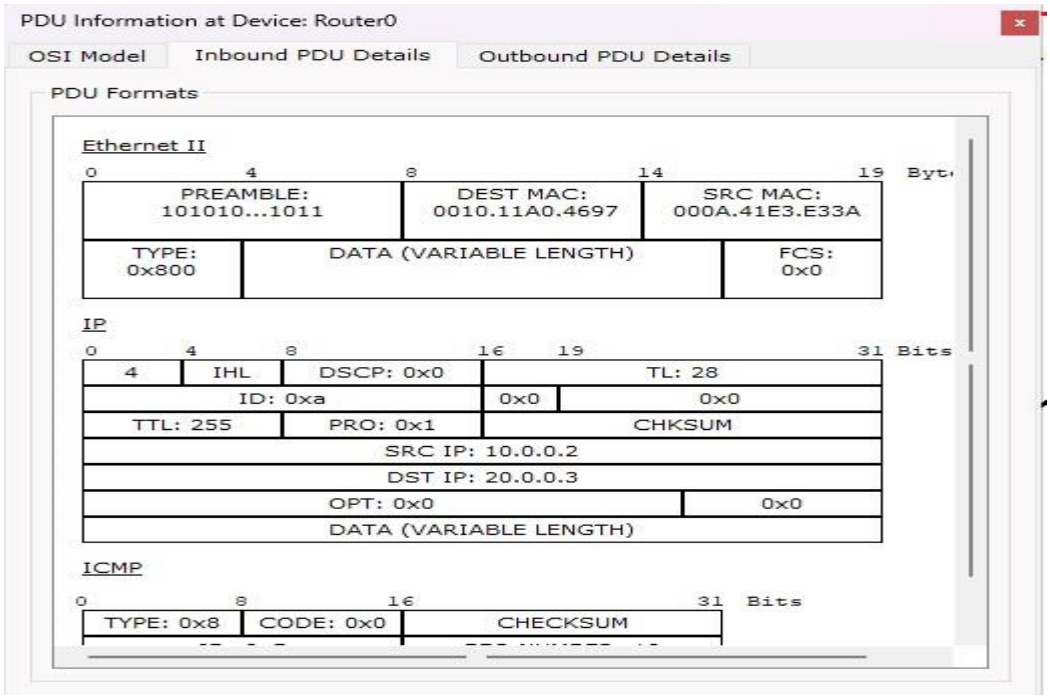
Sol

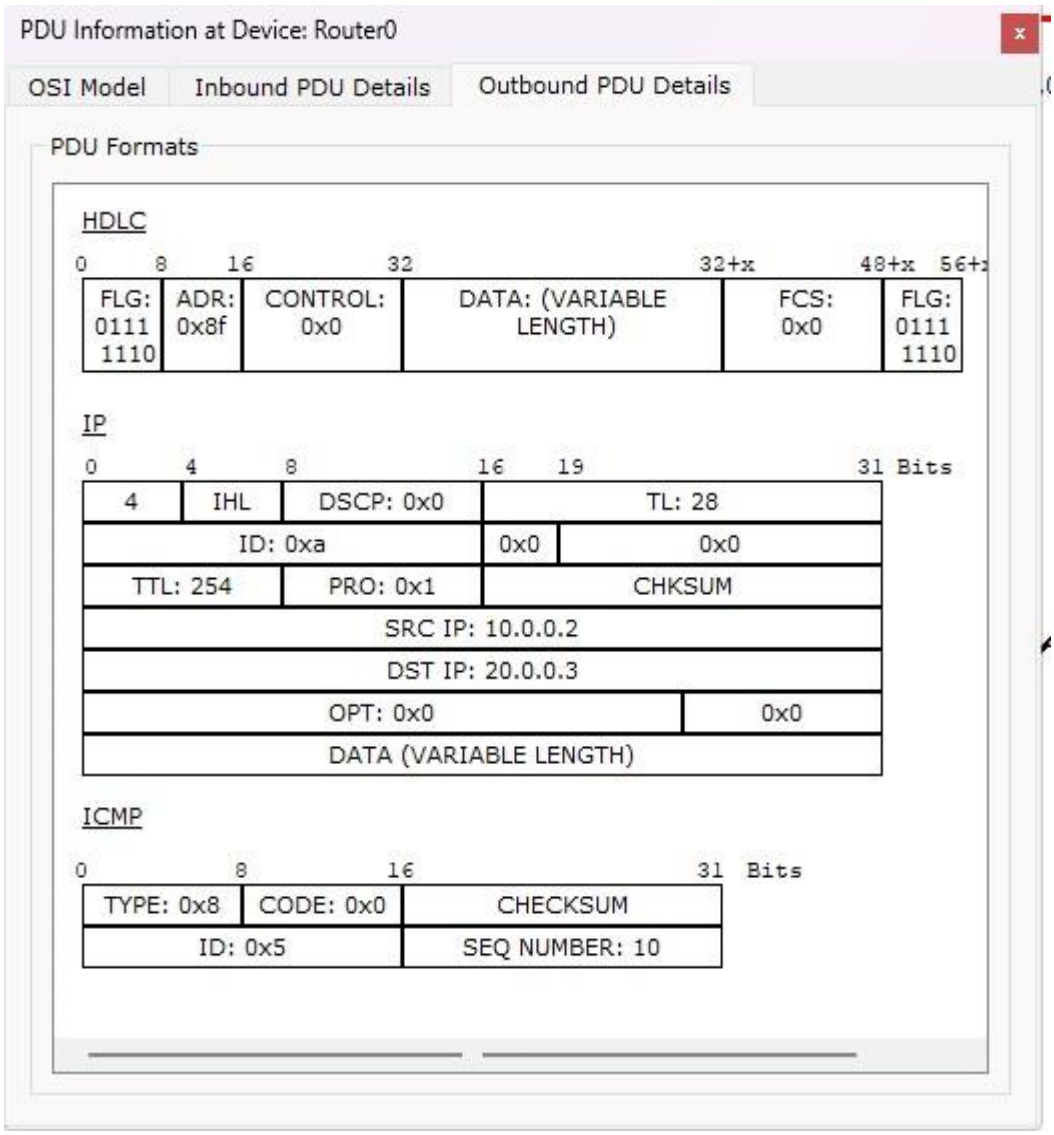
→ Configure the devices as the above topology.

→ Configure the routers as well

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 .

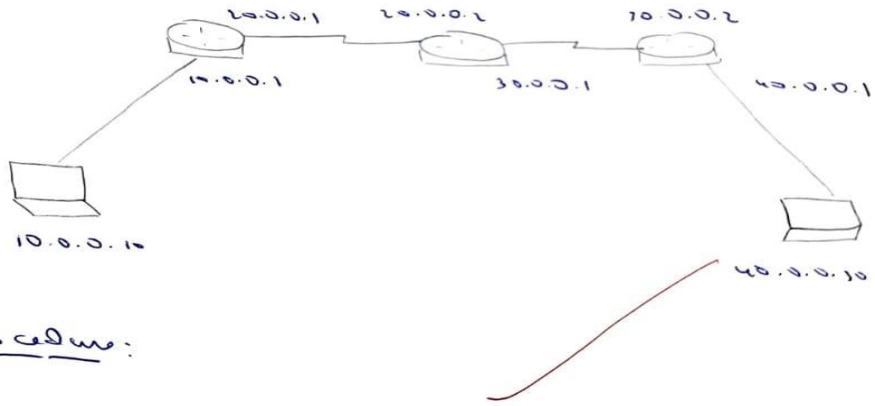
Topology , Procedure and Observation:

Experiment 8

Configure OSPF routing protocol

Aim: To demonstrate OSPF, routing protocol with several routers & networks

Topology:



Procedure:

1) Configure all the devices as given clock in the topology

2) Enter the following commands:

) Router (config-if) # encapsulation ppp (To all the ports with clock 0 symbol):

) Router (config-if) # clock rate 64000

2) Enter the OSPF commands:

-) Router (config) # router ospf 1

-) Router (config-router) # router-id 1.1.1.1
(or any arbitrary id)

-) Router (config-router) # network
x.0.0.0 0.255.255.255 area y

3) Enter the loopback commands:

-) Router (config) # interface loopback 0

-) Router (config-if) # ip add 172.16.1.252
255.255.0.0

-) Router (config-if) # no shut.

4) Create Virtual link

-) R1 (config) # router ospf 1

-) R1 (config-router) # area 1 virtual-link
2.2.2.1

:) R2(config)# router ospf 1

:) R2(config-router)# area 1 virtual-link 1.1.1.1.

Observation:

Upon entering show ip route in R3

O IA 10.0.0.0/8 via 30.0.0.1, serial 2/0

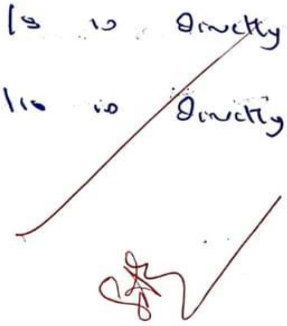
O IA 20.0.0.0/8 via 30.0.0.1, serial 2/0

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

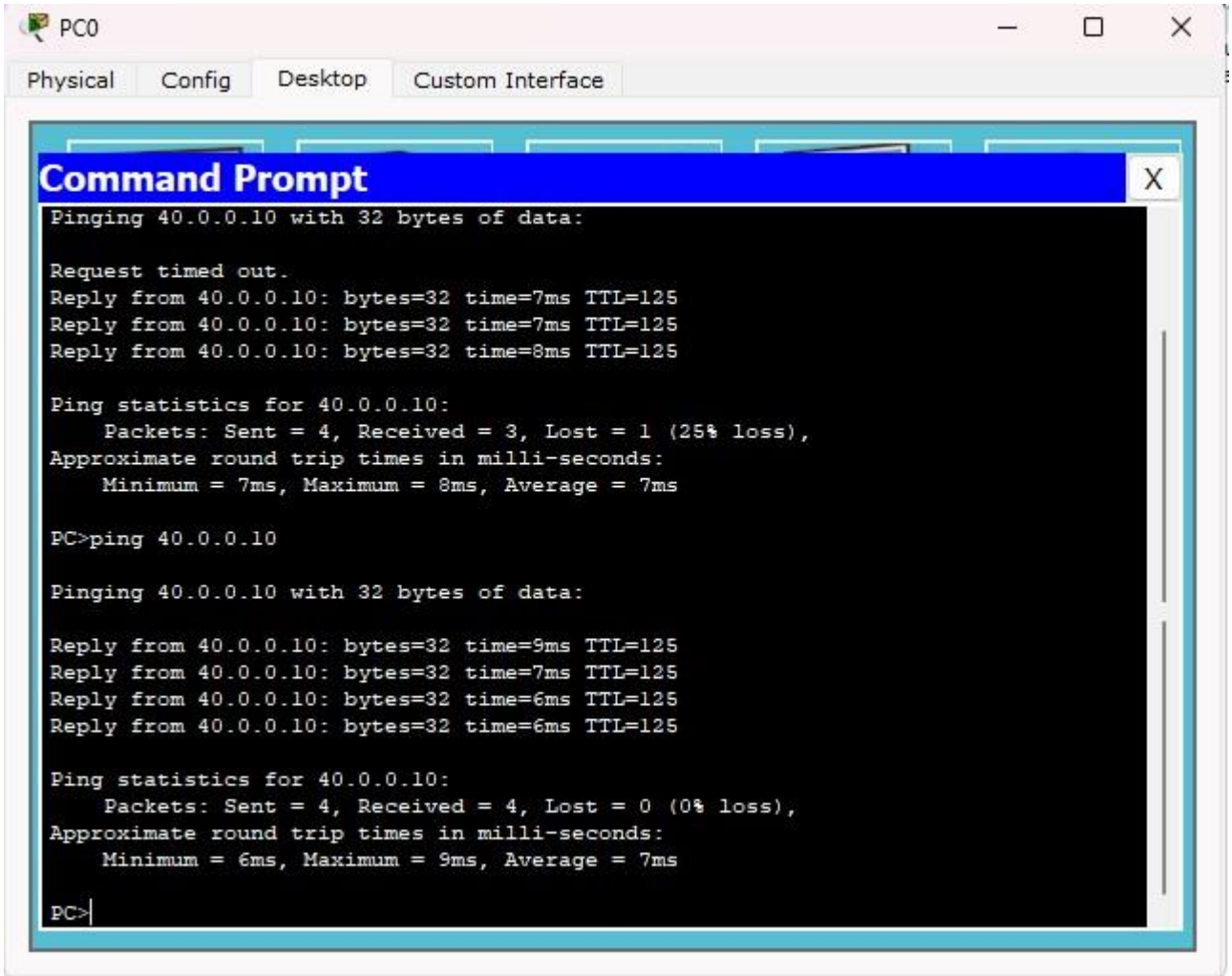
C 30.0.0.1/32 is directly connected, serial 2/0

C 40.0.0.0/8 is directly connected, fastethernet

C 192.16.0.0/16 is directly connected, loopback 0



Screen Shots:



Program 9

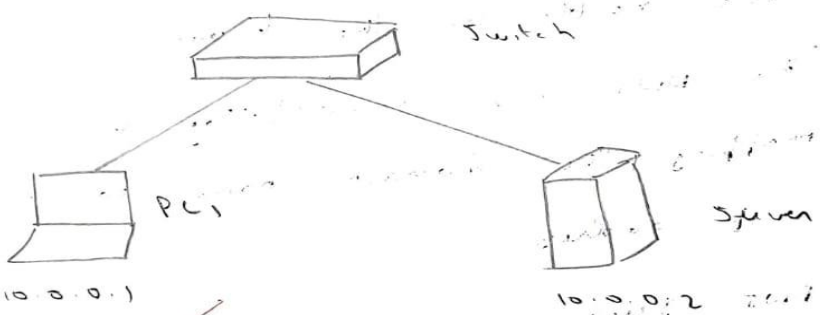
Aim: Configure Web Server, DNS within a LAN.

Topology , Procedure and Observation:

DNS

Configure Web Server, DNS within a LAN

Topology



Procedure:

1) Set up the LAN as shown in the topology
2) Go to Server → Service → DNS
   Name: BMS
   Address: 10.0.0.2
3) Add the mapping of domain name by address
4) Go to PC → config → global → Setting → DNS server: 10.0.0.2
   (The server that provides the DNS mapping)

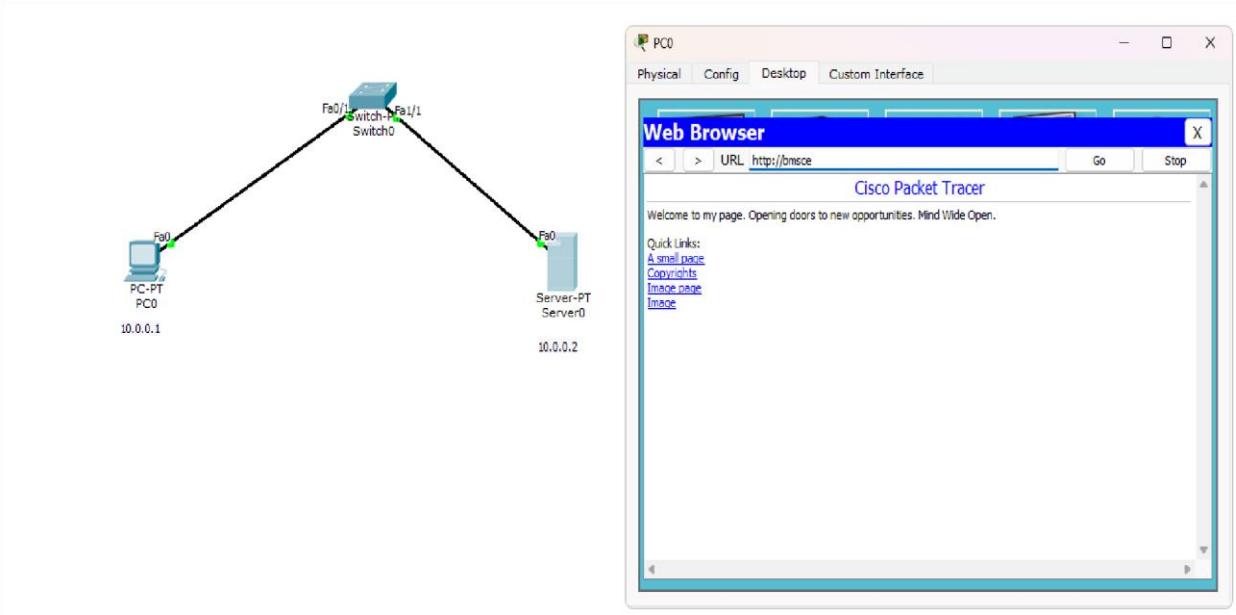
Go to PC → Desktop → Web Browser

Type the URL: https://bms..

Observation:

- The Webpage hosted by the server were visible on the browser
- The DNS was successfully in mapping the domain name to the ip address
- DNS Server is a server that contains a Domain name: IP address mapping to which the end devices sent request to map the name to IP Address.

Screen Shots:



Program 10

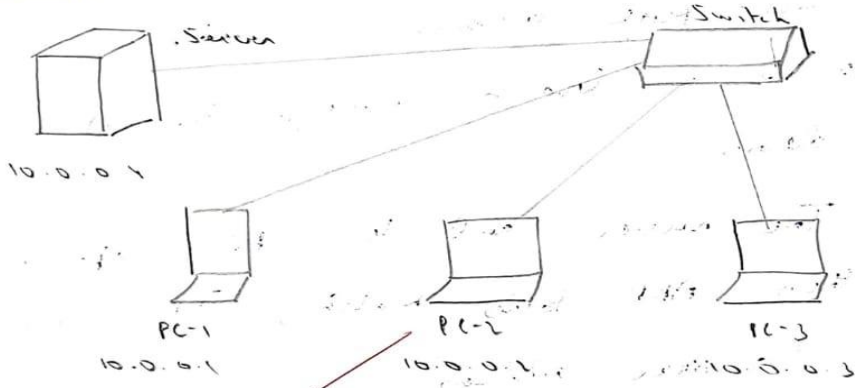
Aim: To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)

Topology , Procedure and Observation:

ARP

To construct a simple LAN & understand the concept & operation of address resolution protocol.

Topology



Procedure:

1) Create the topology as shown above
2) Configure the PC's & the server.
3) Click on Inspect mode (p) → and device → open ARP Table.
4) Send a Data Packet from any end device say server to PC → 10.0.0.3
5) Open simulation mode to capture both step of Data transfer.

Observation

→ The ARP Table of all end devices are initially empty

→ When the data packet from device arrives at the switch since the source MAC address is unknown, it sends a broadcast message to all devices

→ The server and the PC updates their ARP tables matching IP address to MAC address

→ Once time the ARP tables grows by data packet are sent.

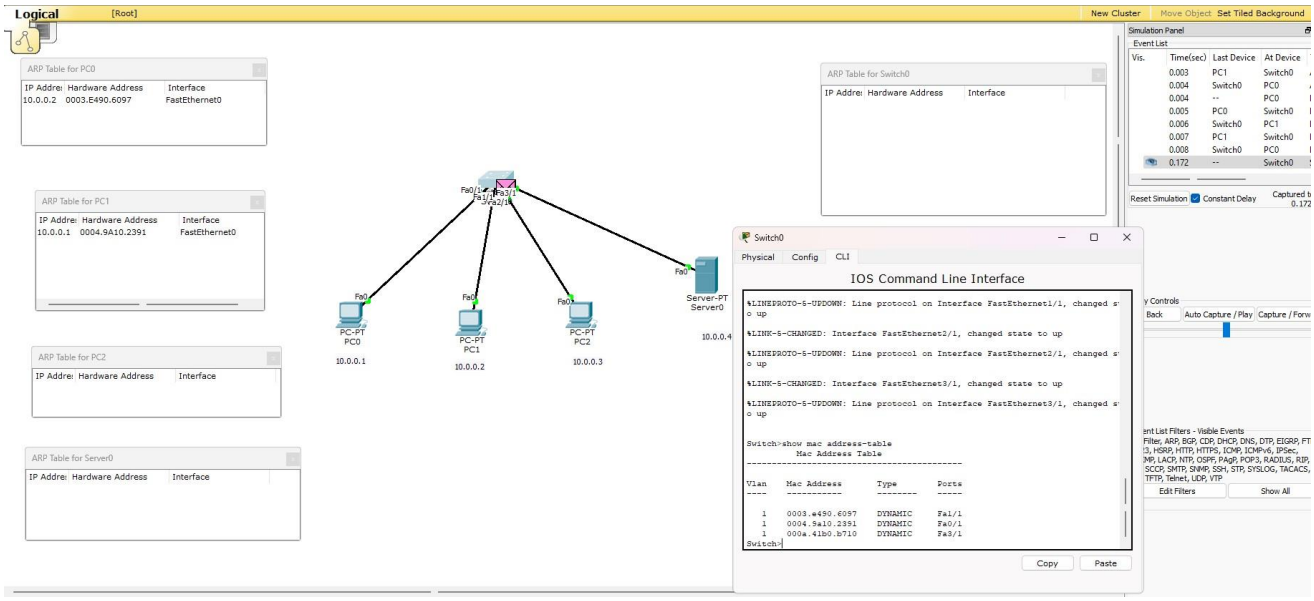
→ ~~Similarly the ARP table~~

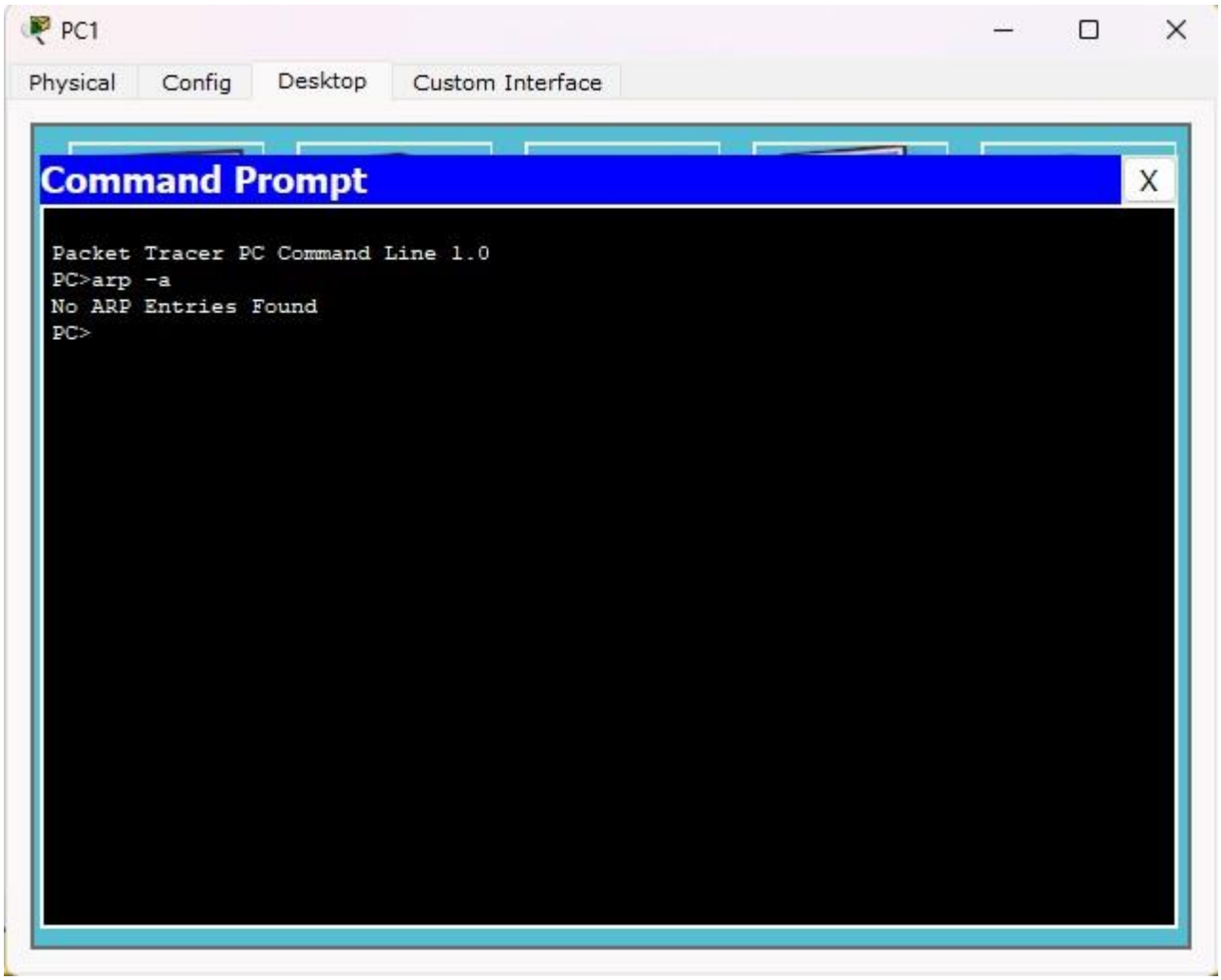
Similarly other ARP tables are updated.

ARP table for 10.0.0.4

IP address	Hardware Address	Interface
10.0.0.3	0001.C726.47E5	FastEthernet 0

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:

TELNET

To understand the operation of telnet, by accessing the router in server room from a PC in office.

Topology :

Procedure:

1) Create the topology as given as above & configure the devices

2) In Router:

router enable
router config t
router # host name R9
R1 # enable secret 1234
interface fast ethernet 0/0
ip address 10.0.0.2 255.0.0.0
no shut
line vty 0 5

login
password 4321 } 2nd lw
next to

:) In PC → Command prompt

:) First try pinging to see if devices are connected

PC> telnet (0.0.0.2)

Trying 10.0.0.2 open

User access verification

Password: 4321

Password: 4321

R1 > enable

Password: 1234

R1 # show ip route

C 10.0.0.0/8 is directly connected

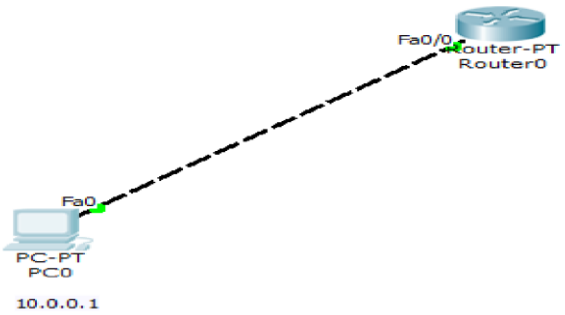
fast ethernet 0/1.0.

Observation:

:) The admin in P.C is able to run command as run in router CLI's are sent from PC

:) Telnet allows user to establish a remote session with another device

Screen Shots:



Command Prompt

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

Pinging 10.0.0.2 with 32 bytes of data:

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

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

PC>telnet 10.0.0.2
Trying 10.0.0.2 ...Open


User Access Verification

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

Gateway of last resort is not set

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

Program 12

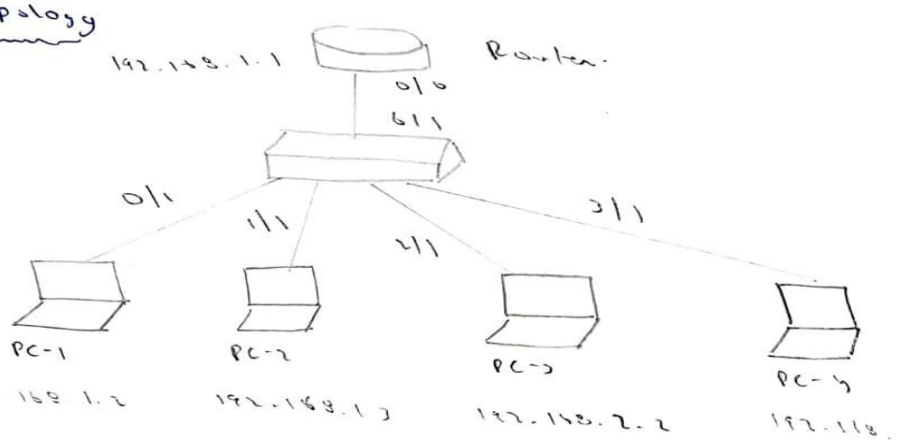
Aim: To construct a VLAN and make the PC's communicate among a VLAN .

Topology , Procedure and Observation:

VLAN

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

Topology



Procedure:

1) Place a router, a switch and 4 PCs and make connections as shown in topology
2) Connect the four PCs to the switch via fastethernet
3) Since only 4 fast ethernet ports are available in a switch, we have to add an ethernet port.
4) Switch off the power button of switch & add ethernet port to switch.

⇒ Switch on the power both

⇒ Connect the router to the switch via ethernet

⇒ In the switch, go to config tab and:

→ Select VLAN Database

→ Give VLAN no say 2

→ Give VLAN name say "cse"

→ Add it to the database (now)

⇒ Select the switch & go to config

→ go to ethernet

→ make it the trunk.

⇒ Configure the PCs as shown in topology

⇒ Select Switch:

→ Go to config

→ Go to fastethernet 2/1

→ Set VLAN no as 2 & name 2 as "cse"

→ Similarity set VLAN 2 for fastethernet 3/1 interface

→ Configure the Router.

) interface fastethernet 0/0
) IP address 192.168.1.1 255.255.255.0
) no shut
) exit

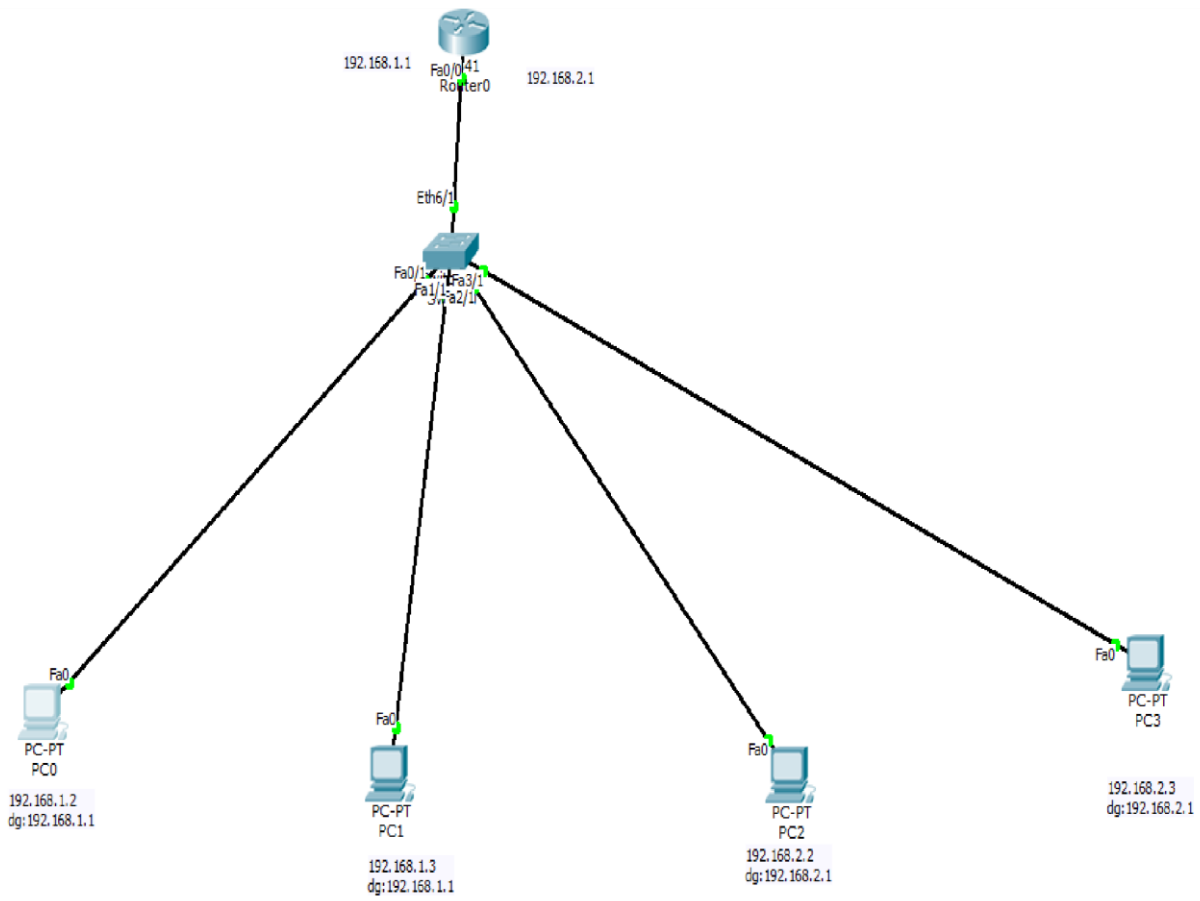
→ Now to configure the router VLAN sub

) interface fastethernet 0/0
) encapsulation dot 1q
) ip address 192.168.2.1 255.255.255.0
) no shut
) exit.

Observation:

) VLAN divide a single switch into multiple logical switches.
) Broadcast sent by device in one VLAN do not reach device in another VLAN
) When the device are pinged within same VLAN
) Pinging 192.168.1.3 from 192.168.1.2
) The data packet should go to the router
) The switch forwards the packet without the

Screen Shots:



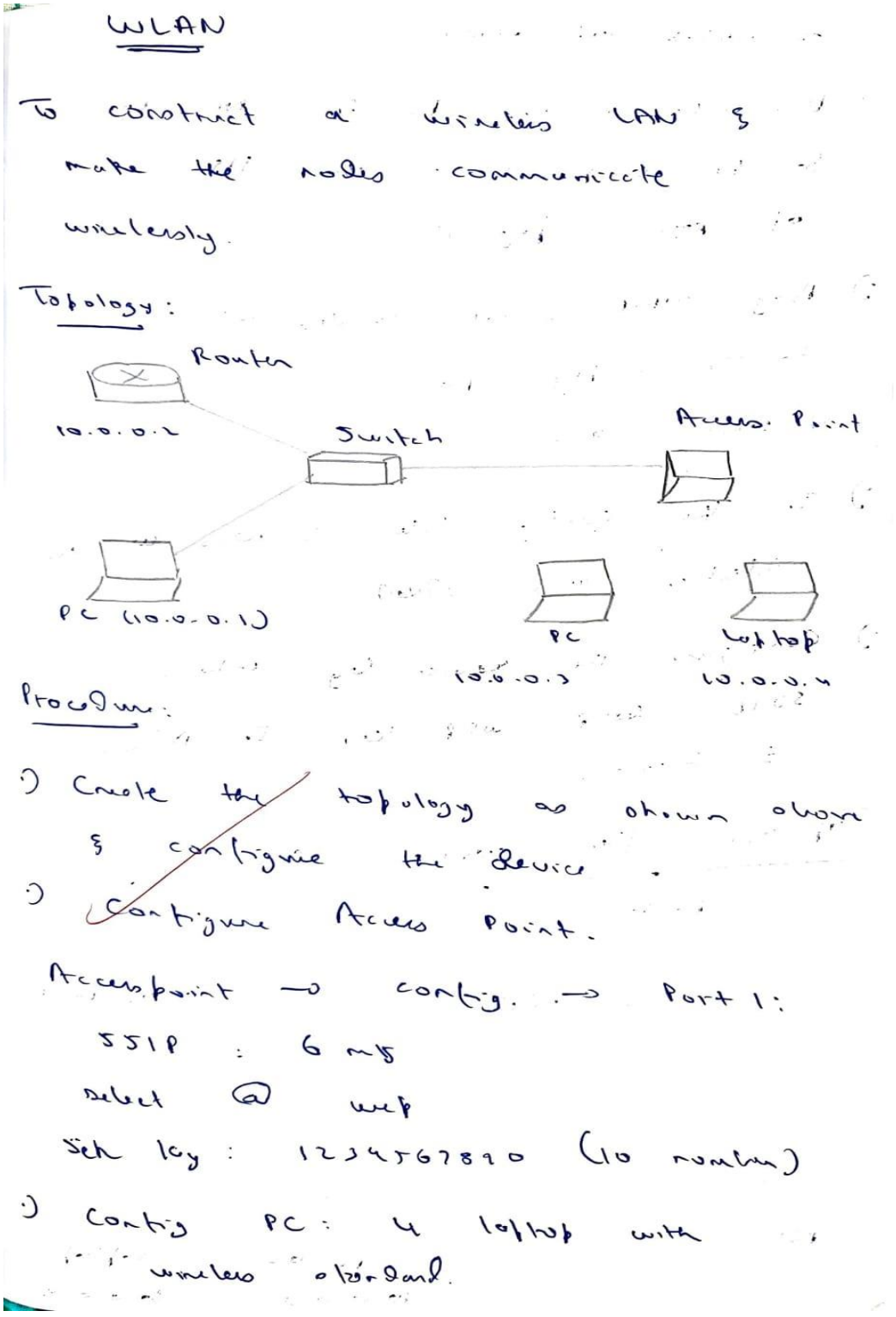
Command Prompt

```
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:



→ switch off device

:) Drag the existing IT - Res NTM - LAB to the component in the LHS of physical

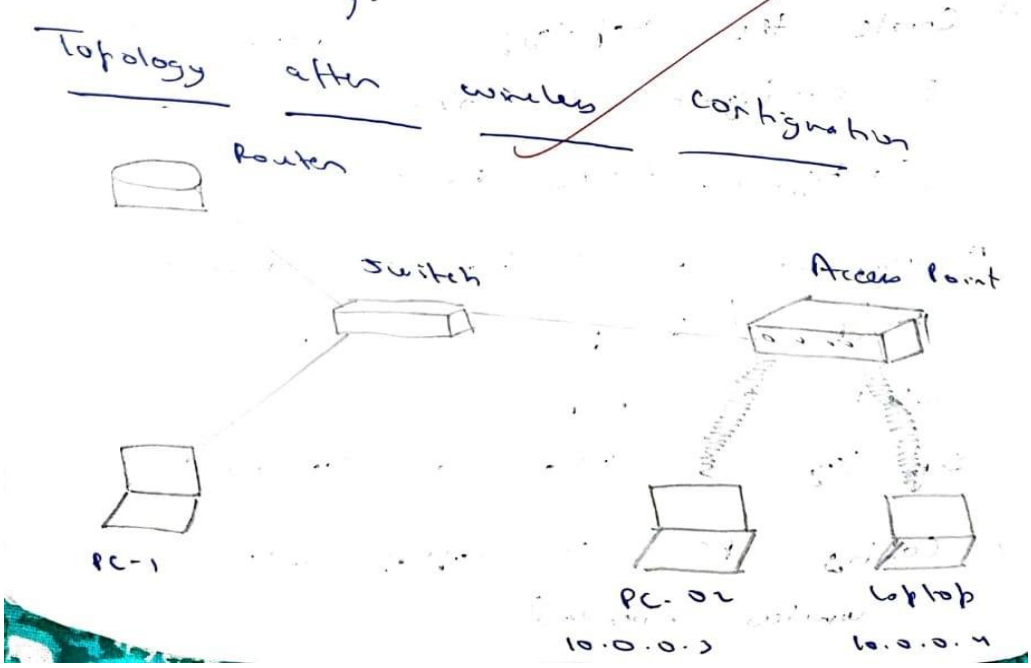
:) Drag WMP 300N wireless interface to the empty point

:) switch on.

:) In the config tab, a new wireless interface was added.

:) Configure the device by entering SSID, WEP, WEP key, IP Address, & Gateway.

Topology after wireless configuration



Ping from every device to every other device to check for connection.

Observation

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

:) Pinging 10.0.0.3 to 10.0.0.1

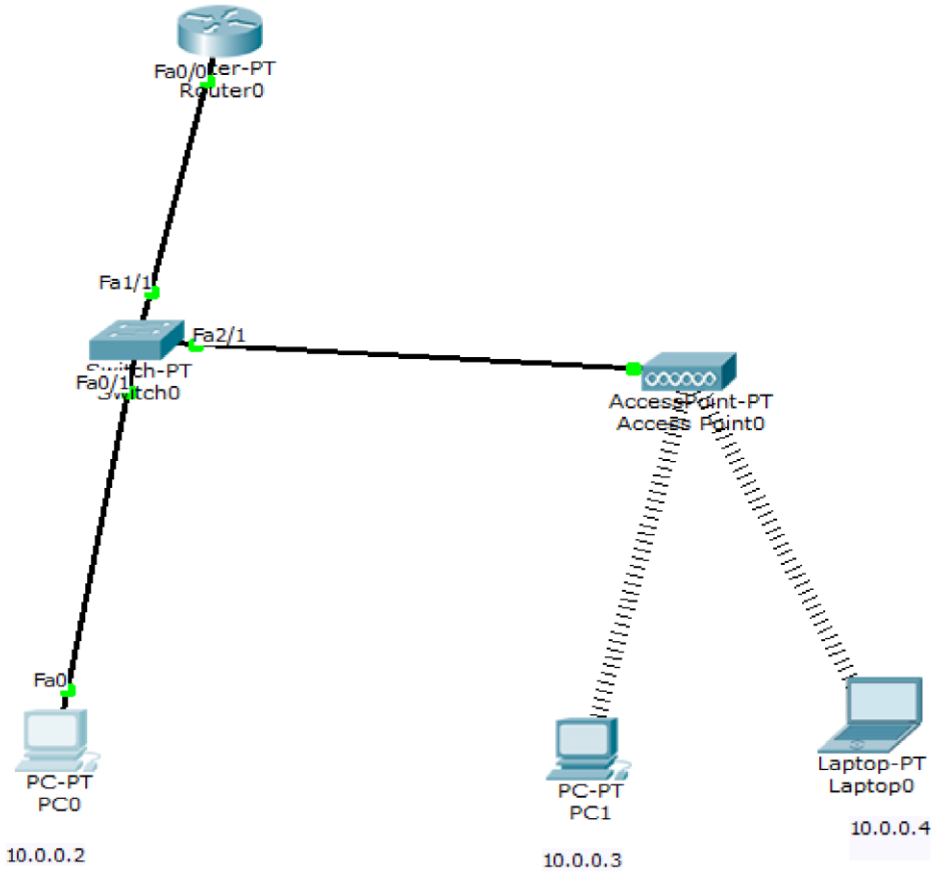
10.0.0.3 → Access Point

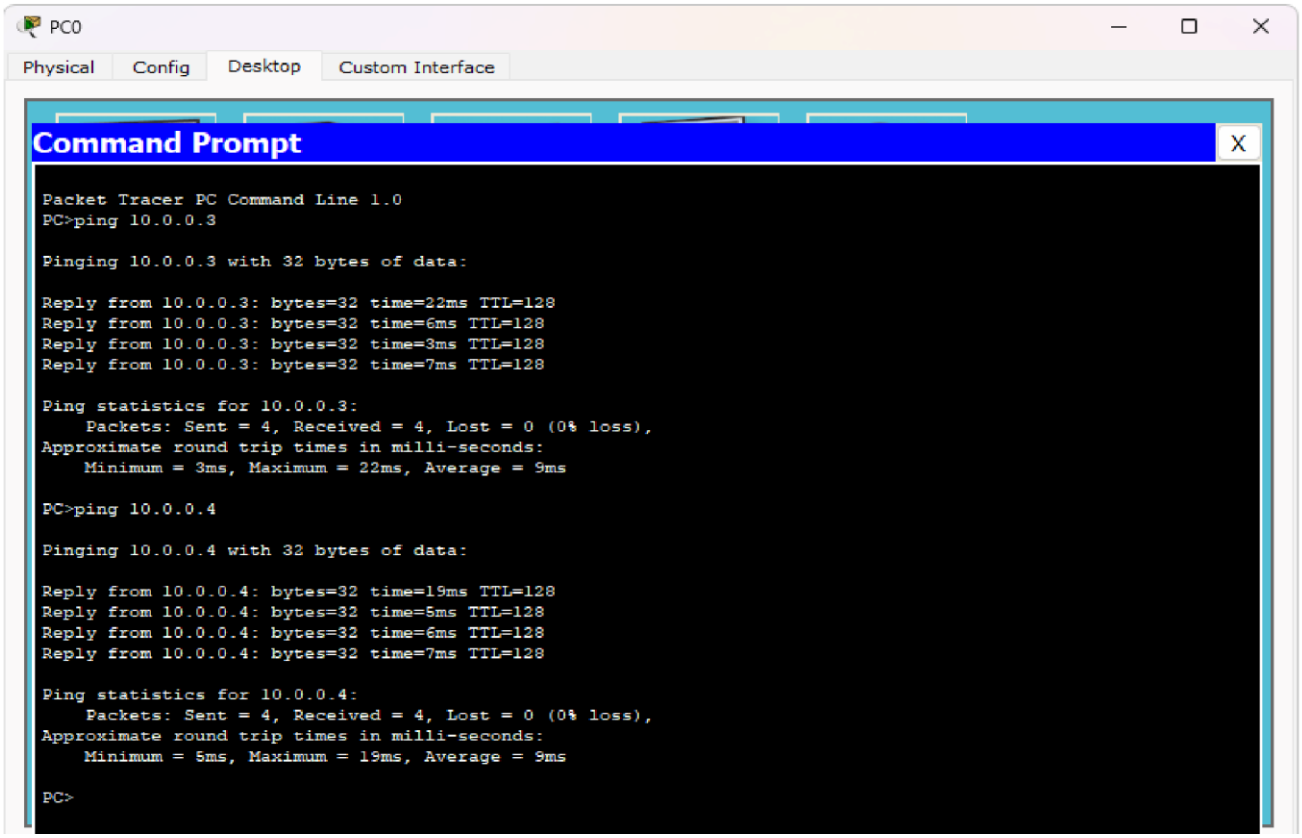
Switch 10.0.0.1

:) WMP 300N wireless interfaces

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

Screen Shots:



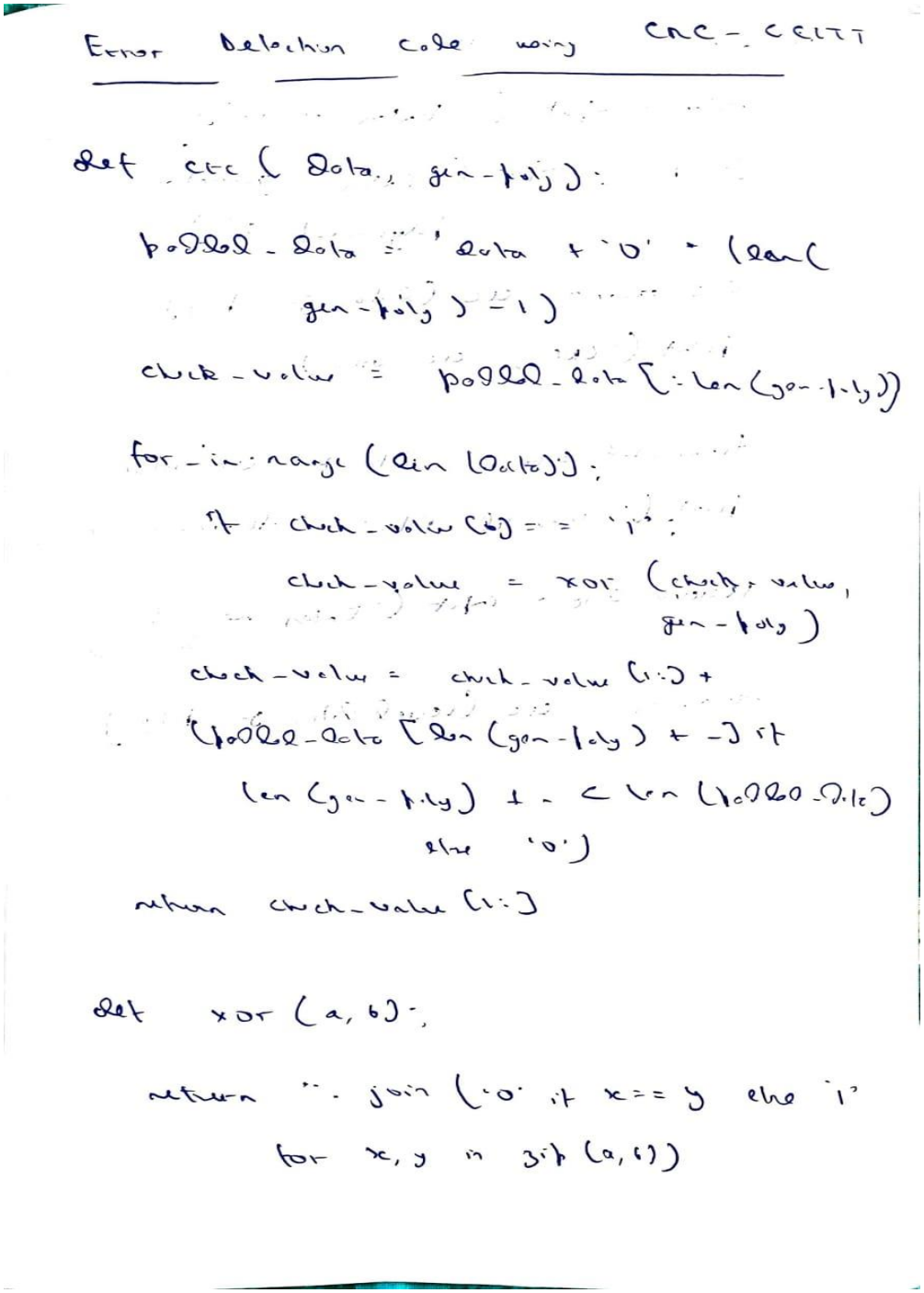


PART-B

Program 14

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

Code and Output:

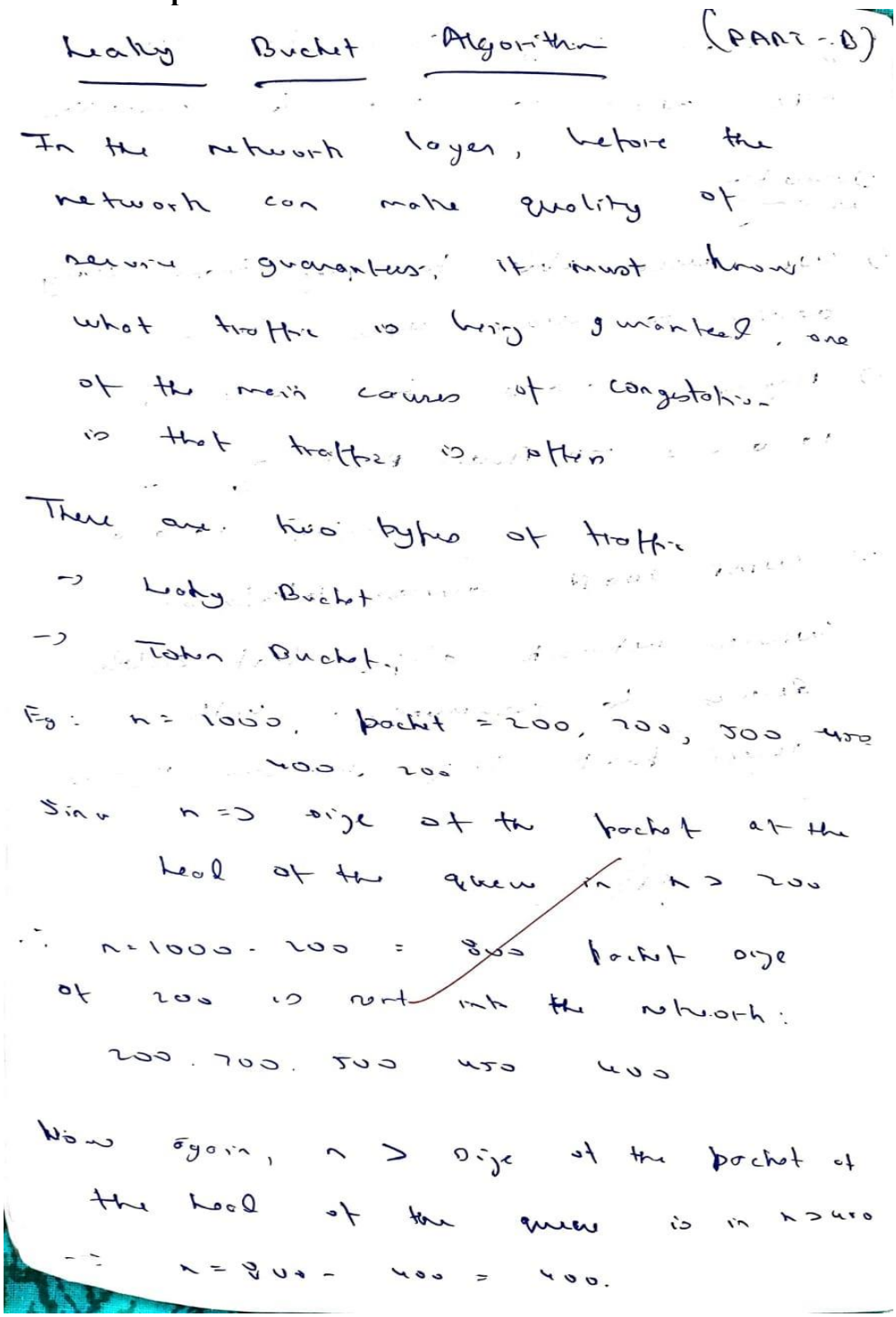


```
if __name__ == "__main__":
    data = input("Enter Data: ")
    gen_poly = input("Enter generator polynomial: ")
    crc_value = crc(data, gen_poly)
    print("CRC", crc_value)

    transmitted_data = data + crc_value
    print("Transmitted Data:", transmitted_data)

    received_data = input("Enter received data: ")
    remainder = crc(received_data, gen_poly)
```

Program 15

Write a program for congestion control using Leaky bucket algorithm.

Code and Output:



Leaky Bucket Algorithm (PART-B)

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 bursty.

There are two types of traffic
→ Leaky Bucket
→ Token Bucket.

Eg: n = 1000, packet = 200, 700, 500, 450, 400, 200

Since n => size of the packet at the head of the queue in n > 200

∴ n = 1000 - 200 = 800 packet size of 200 is sent in the network:

200, 700, 500, 450, 400

Now again, n > size of the packet at the head of the queue is in n > 450

∴ n = 800 - 400 = 400.

Code:

```
import time
import random

def leaky_bucket(packets, bucket_size,
                 output_rate):

    remaining = 0

    for packet in packets:

        if packet > bucket_size:

            print(f"Packet of size {packet} bytes
            exceeds bucket capacity {bucket_size}
            bytes) - REJECTED").

        elif packet + remaining > bucket_size:

            print(f"Bucket capacity exceeded
            with packet size {packet} bytes -
                                REJECTED")

        else:

            remaining = packet

            print(f"In Packet of size {packet}
                  bytes added to bucket")

            print(f"Bytes in bucket: {remaining}")
```

```
while remaining > 0:
    time.sleep(1)
    if remaining <= output_rate:
        print(f"Transmitting {remaining} bytes")
        remaining = 0
    else:
        print(f"Transmitting {output_rate} bytes")
        remaining -= output_rate
print(f"Bytes remaining in bucket: {remaining}")
```

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:

CLIENT - TCP

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = input("\n Enter file name :")
clientSocket.send(sentence.encode())
fileContents = clientSocket.recv(1024).decode()
print("\n From Server: \n")
print(fileContents)
clientSocket.close()
```

SERVER TCP

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
```

```
while 1:
    print ('The server is ready to receive')
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    file = open(sentence, 'r')
    l = file.read(1024)
    connectionSocket.send(l.encode())
    file.close()

    connectionSocket.close()
```

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:

CLIENT UDP

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input('Enter file name:')
clientSocket.sendto(sentence.encode(),
                    (serverName, serverPort))
fileContents, serverAddress = clientSocket.recvfrom
                              (2048)
print('From Server:', fileContents.decode())
clientSocket.close()
```

SERVER UDP

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(('127.0.0.1', serverPort))
print('The server is ready to receive')
```

```
While True:
    message, clientAddress = serverSocket.recvfrom(2048)
    try:
        with open(message.decode(), 'r') as file:
            fileContents = file.read(2048)
        serverSocket.sendto(fileContents.encode(),
                            clientAddress)
        print('Sent contents to client:')
        print(fileContents)
    except FileNotFoundError:
        errorMessage = 'File not found!'
        serverSocket.sendto(errorMessage.encode(),
                            clientAddress)
        print('File not found error
              sent to client.')
```