VISVESVARAYATECHNOLOGICALUNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB RECORD

Computer Network Lab (23CS5PCCON)

Submitted by

Dhruv Rampuria (1BM22CS088)

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

<u>CERTIFICATE</u>

This is to certify that the Lab work entitled " Computer Network (23CS5PCCON)" carried out by **Dhruv Rampuria (1BM22CS088),** 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.

Sandhya A Kulkarni Associate Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE

Index

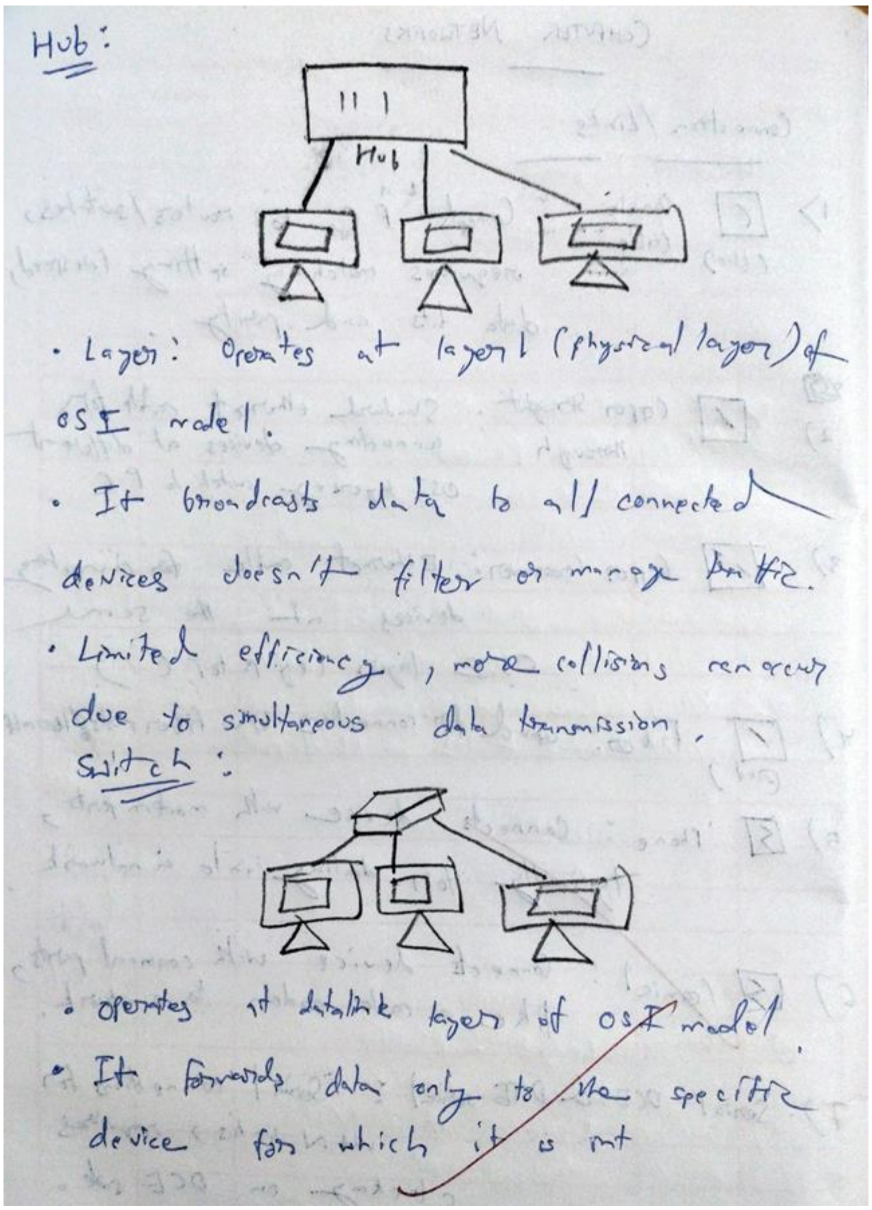
16	18-12-24	Using TCP/IP sockets, write a client-server program to make the client send the file name and the server to send back the contents of the requested file if present.	51-52
17	18-12-24	Using UDP sockets, write a client-server program to make client sending the file name and the server to send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

Github Link: DhruvR-2004/CN_Lab

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:



Hub:

Hub

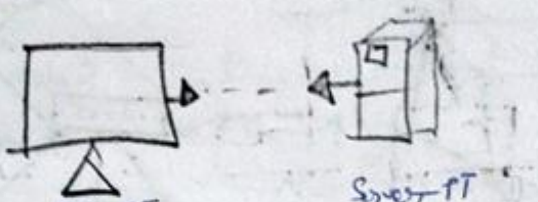
- Layer: Operates at layer 1 (physical layer) of OSI model.
- It broadcasts data to all connected devices doesn't filter or manage traffic.
- Limited efficiency, more collisions can occur due to simultaneous data transmission.

Switch:

- operates at datalink layer of OSI model.
- It forwards data only to the specific device for which it is meant

Experiment 1

1. PC to server.



Aim :- To set up a point network b/w a PC and a server, facilitating direct communication to observe data exchange.

Topology: A PC is connected to server using a crossover ethernet cable.

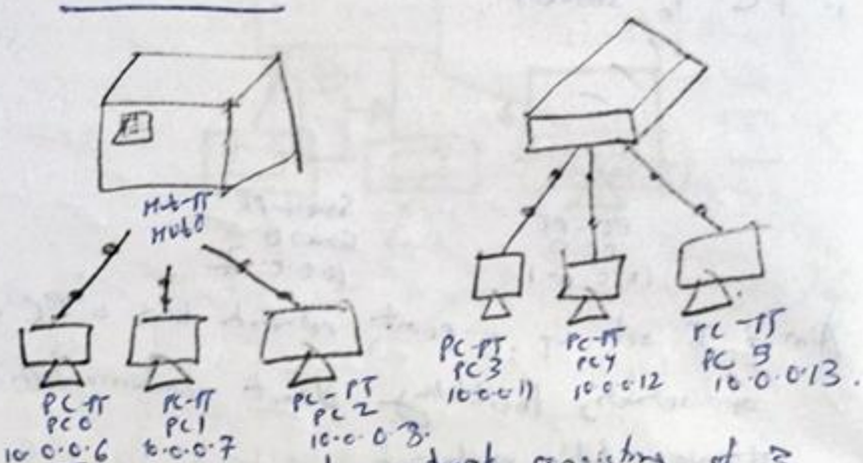
IP address of PC - 10.0.0.1 & server ~~using straight through ethernet cable~~.

Observation: ~~Hub~~

IP address of pc - 10.0.0.1, server - 10.0.0.2.

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

2. Hub and Switch



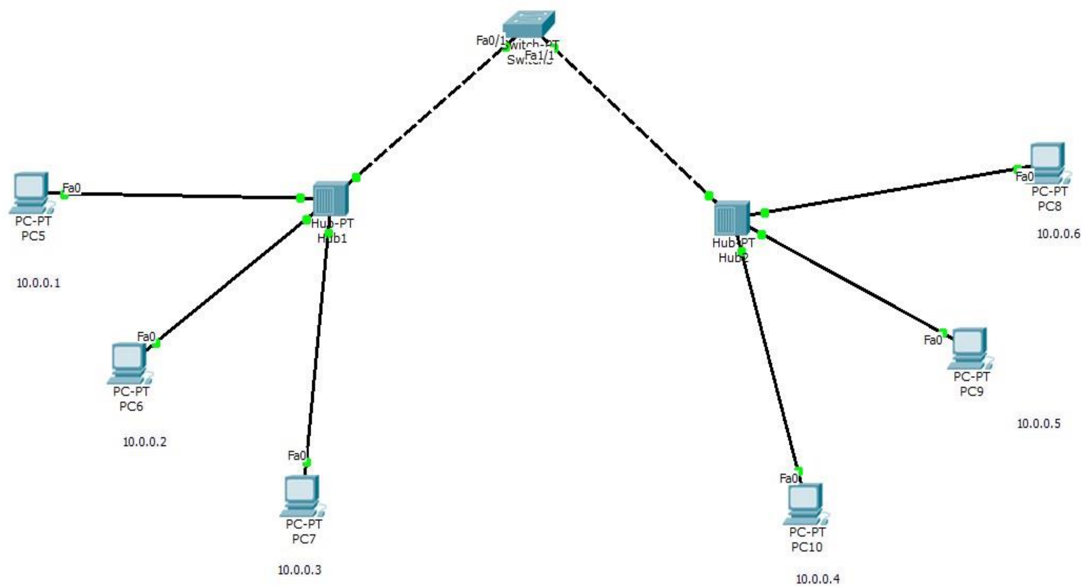
Aim – To create simple network consisting of 3 PCs connected to central hub and another network with 3 PC's connected to a switch, this connection will help observe the behaviours of data transmission using hub and switch devices.

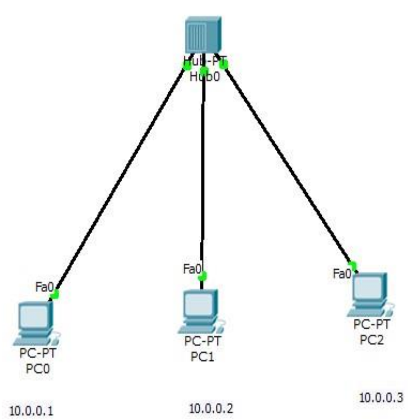
Topology: 3 PCs are connected to a hub and switch using straight through ethernet cables.

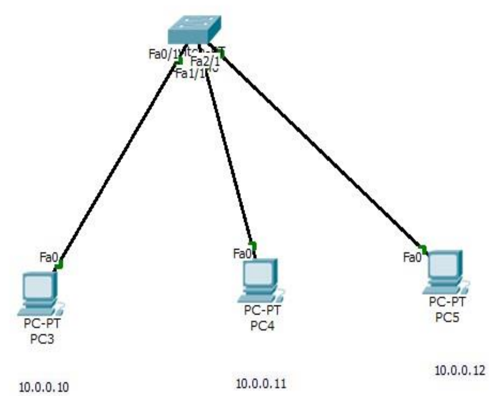
Observation: Hub broadcasts packets to all devices which may cause unnecessary traffic.

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

Screen Shots:


Fa0/1
Fa1/1
Fa0
PC-PT
PC5
10.0.0.1
PC-PT
PC6
10.0.0.2
PC-PT
PC7
10.0.0.3
Hub1
PC-PT
PC8
10.0.0.6
PC-PT
PC9
10.0.0.5
PC-PT
PC10
10.0.0.4


Hub0
Fa0
PC-PT
PC0
10.0.0.1
PC-PT
PC1
10.0.0.2
PC-PT
PC2
10.0.0.3


Fa0/1
Fa2/1
Fa1/1
Fa0
PC-PT
PC3
10.0.0.10
PC-PT
PC4
10.0.0.11
PC-PT
PC5
10.0.0.12

Program 2

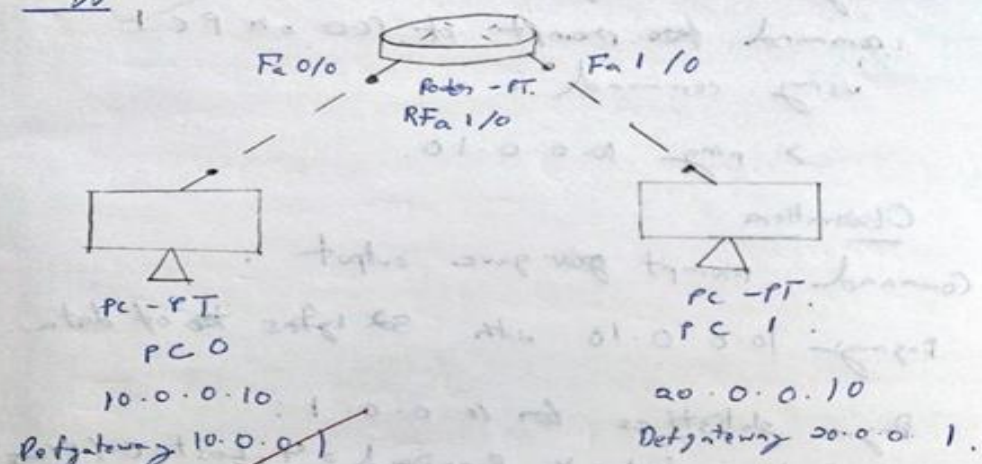
Aim: Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply.

Topology , Procedure and Observation:

LAB-2

AIM : To ~~convert~~ 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 and 20.0.0.1 respectively.

2). Open CLI in router and enter the following:-

```
Router>enable
Router#config terminal
Router(config)#interface fastethernet 0/0.
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shutdown.
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 on interface from the command ~~line~~ prompt of PC0 or PC1 using command.

> ping 10.0.0.10.

Observation

Command prompt ~~gav~~ gave output.

Pinging 10.0.0.10 with 32 bytes ~~to~~ of data.

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

IP route is as follows:
Codes: C - connected, S - static, I - IGRP, R - RIP,
M - Mobile, B - BGP
D - EIGRP,

The following are observed for IP route:
Codes: C - connected
Router > enable
Router # show ip route.

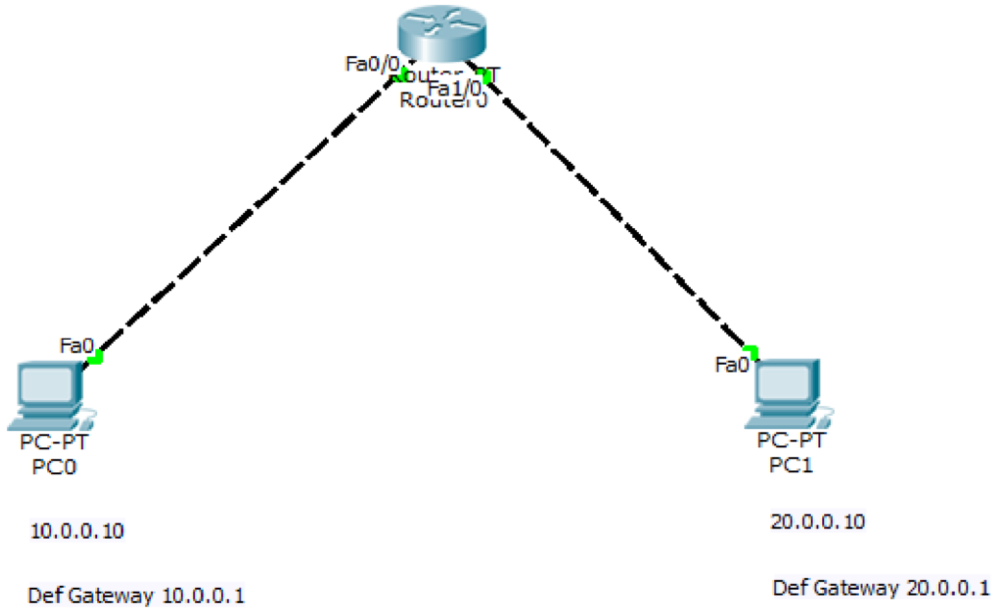
Codes: C - Connected

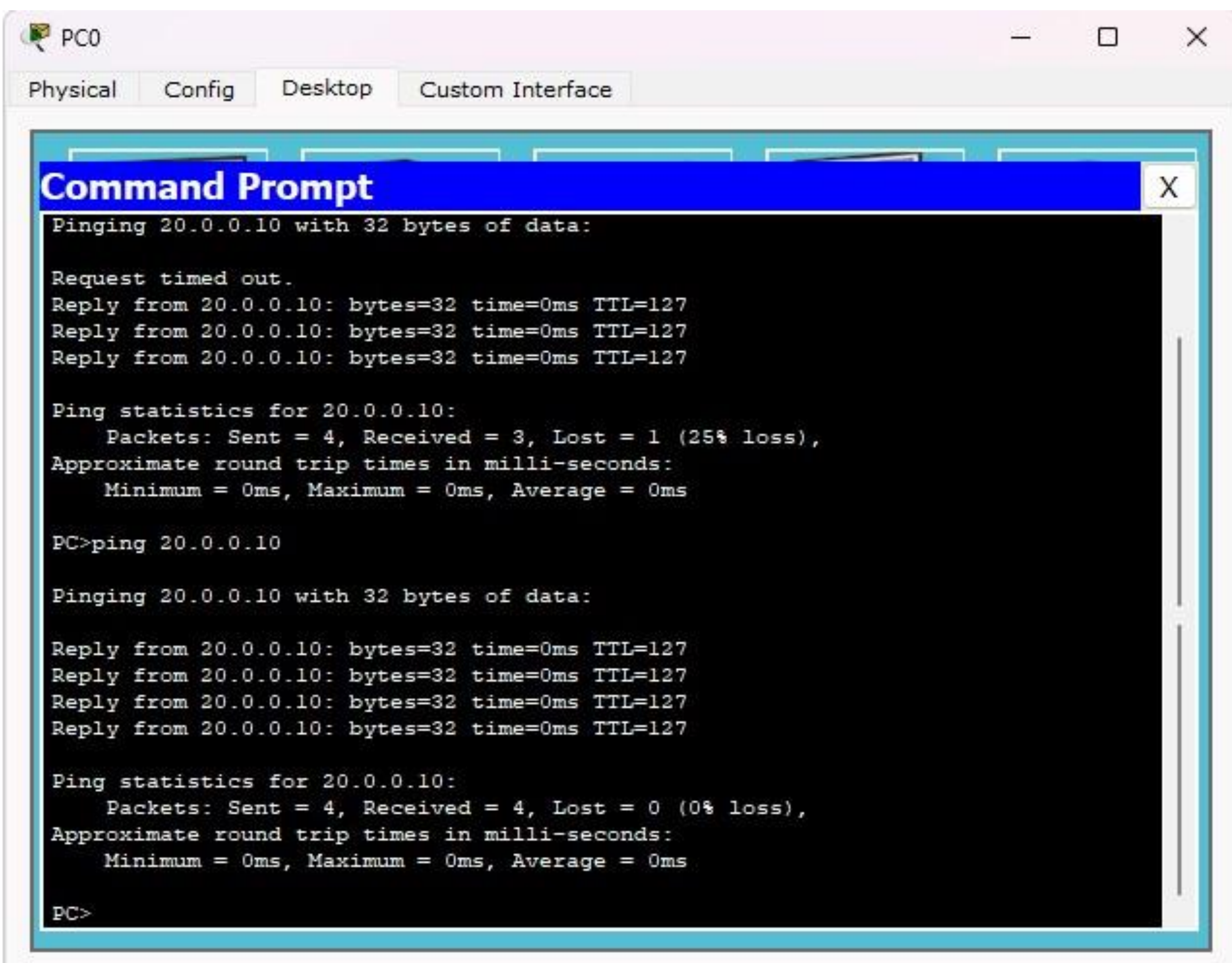
Gateway of last resort is not set

C 10.0.0.0/8 is directly connected, Fastethernet0/0

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

Screen Shots:





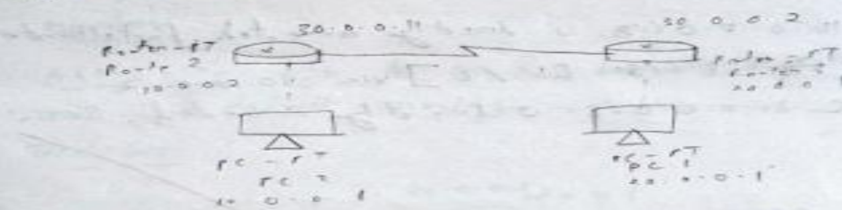
Program 3

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

Topology , Procedure and Observation:

LAB-3

Configure default route, static route to the router.



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

Topology :- Two PCs are connected to two different routers using copper cross over and these 2 routers 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 commands to configure static routing.~~

> ip route 20.0.0.0 255.0.0.0 30.0.0.2

open the CLI of other router,

> ip route 10.0.0.0 255.0.0.0 30.0.0.1

OBSERVATION :

After configuration of static routing, the 2 PCs of 2 different router networks are able to transfer / transmit the packets using ping command.

Output :

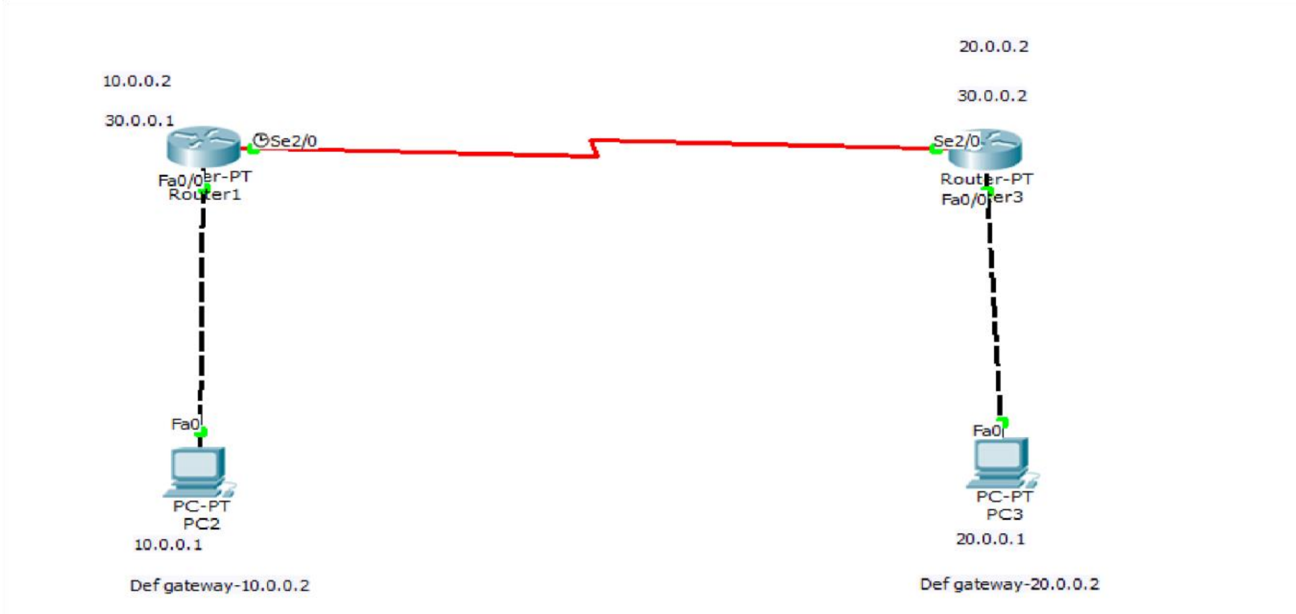
Router# show ip route.

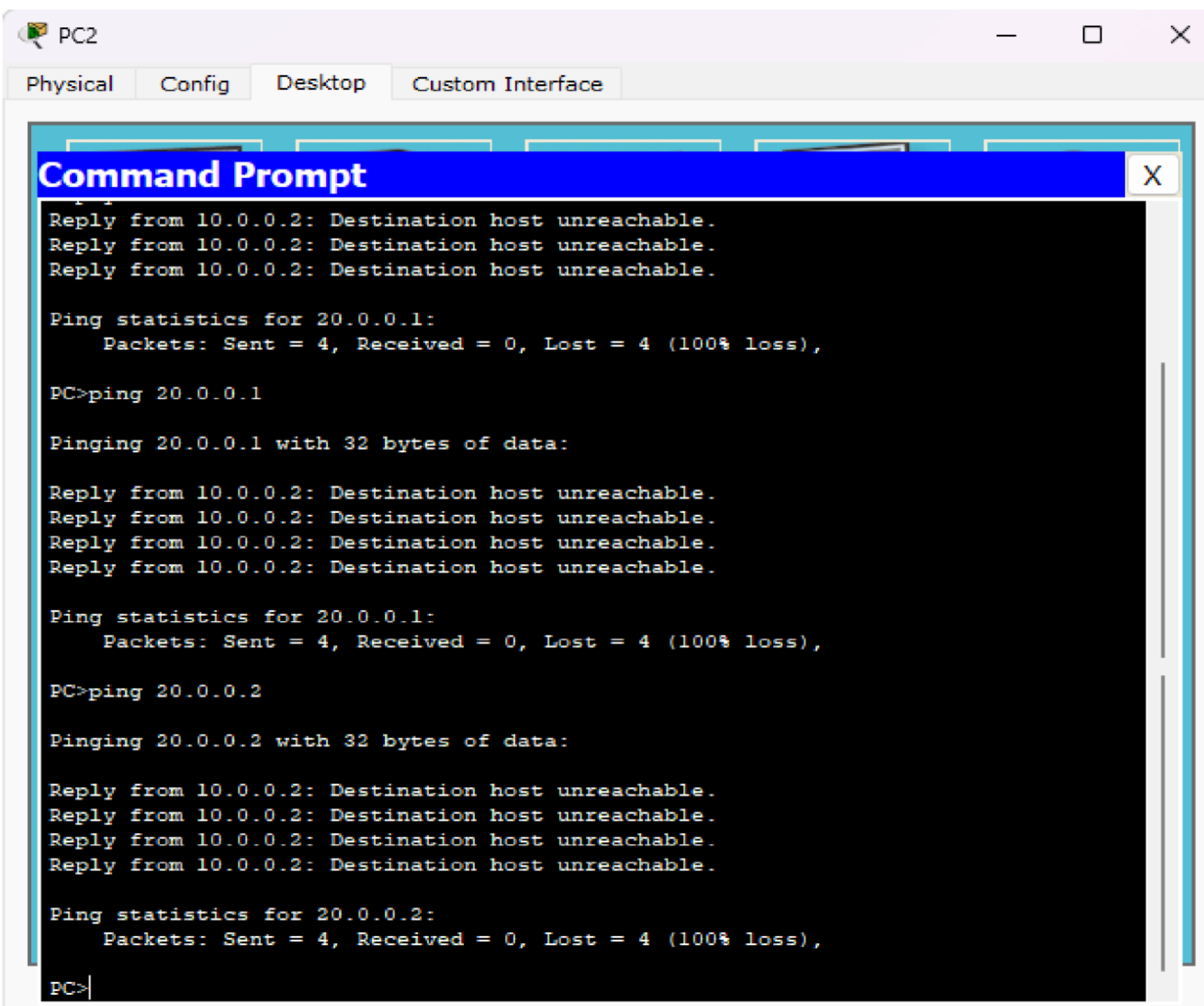
C 10.0.0.0/8 is directly connected, FastEthernet0/0

S 20.0.0.0/8 [1/0] via 30.0.0.2

C 30.0.0.0/8 is directly connected, Serial2/0

Screen Shots:

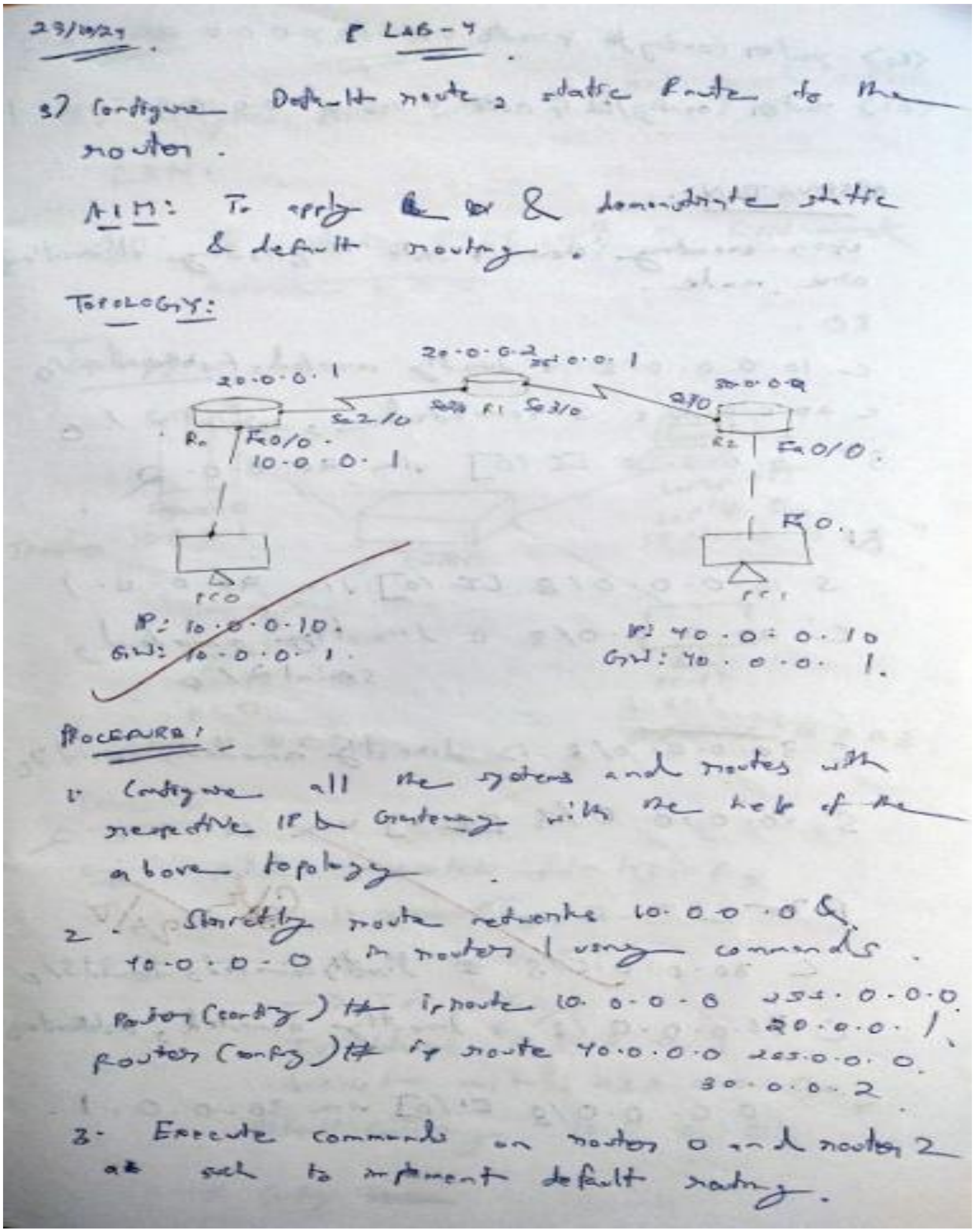




<u>Program 4</u>

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

Topology , Procedure and Observation:



23/10/24 LAB - 4.

3) Configure Default route, static Route to the router.

AIM: To apply & demonstrate static & default routing.

TOPOLOGY:

PROCEDURE:

1. Configure all the systems and routers with respective IP & Gateway with the help of the above topology.
2. Statically route networks 10.0.0.0 & 40.0.0.0 in router 1 using commands.

Router(config)# ip route 10.0.0.0 255.0.0.0 20.0.0.1

Router(config)# ip route 40.0.0.0 255.0.0.0 30.0.0.2.

3. Execute commands on router 0 and router 2 as such to implement default routing.

<R0> router (config)# ip route 0.0.0.0 0.0.0.0 20.0.0.2

<R2> router (config)# ip route 0.0.0.0 0.0.0.0 50.0.0.1

OBSERVATION:

Upon executing "show ip route" following observations are made.

R0.

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

C 20.0.0.0/8 is connected, serial 2/0

S* 0.0.0.0 [1/0] via 20.0.0.2.

R1

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

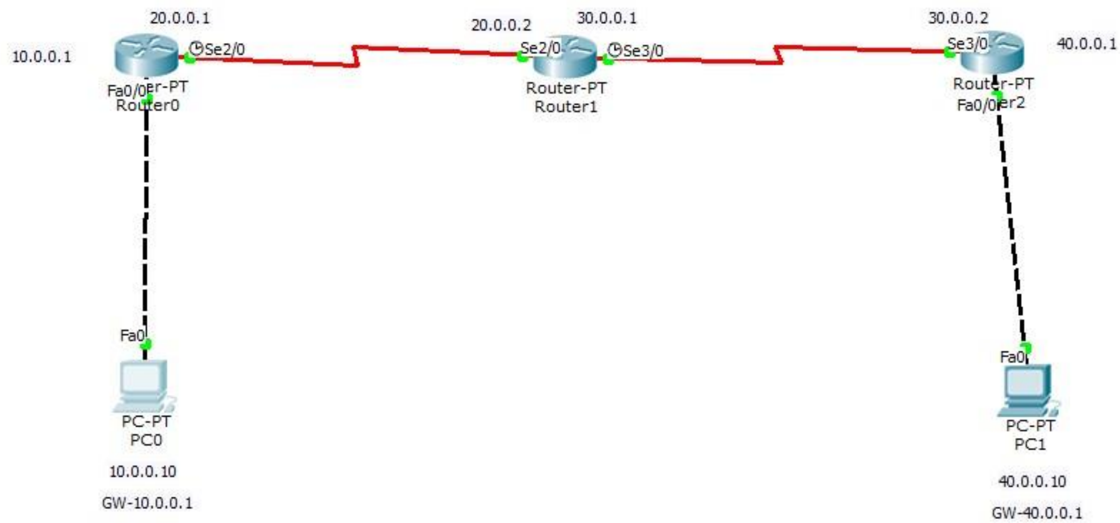
R2

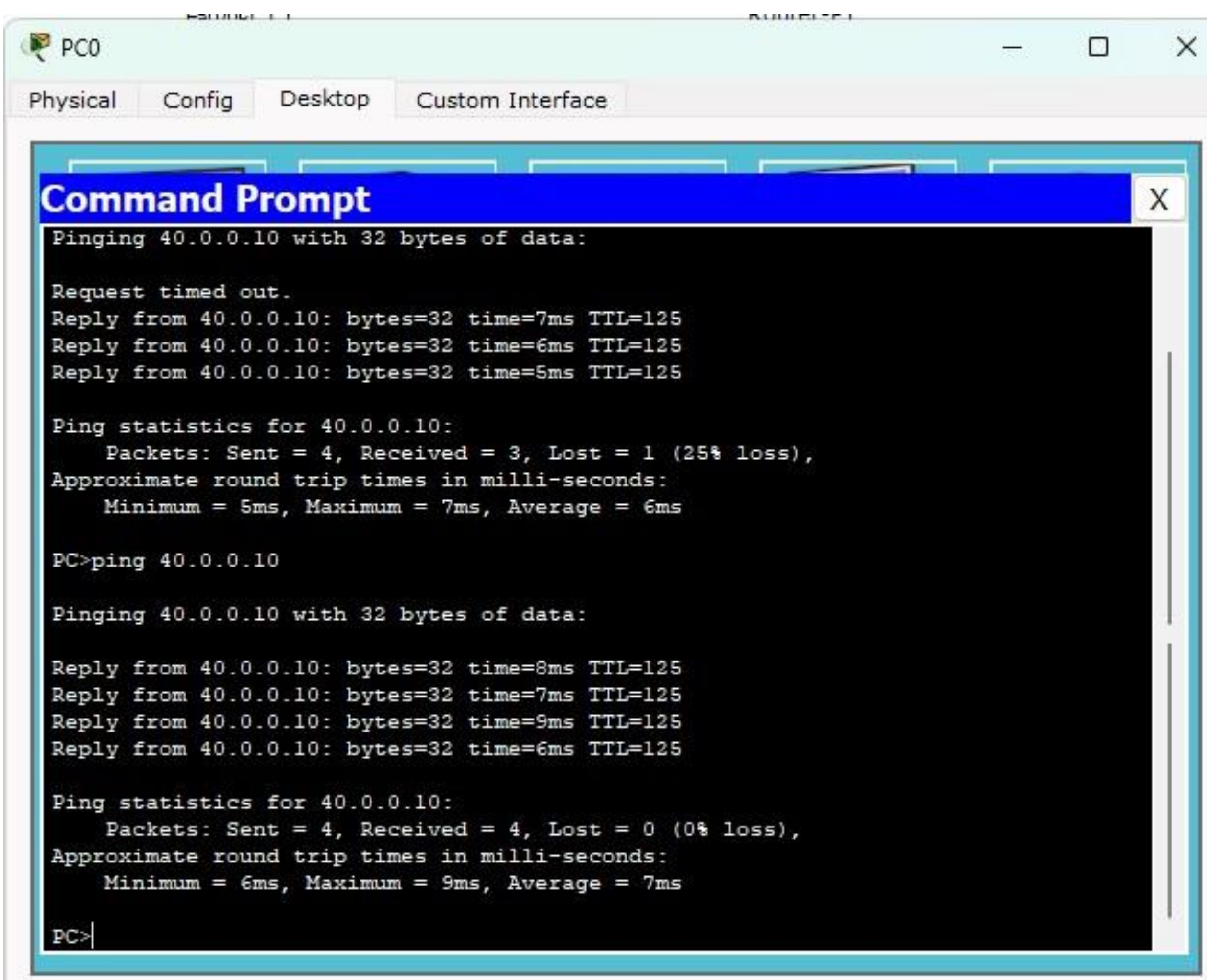
C 30.0.0.0/8 is directly connected, serial 3/0

C 40.0.0.0/8 is directly connected, fastethernet

S* 0.0.0.0/8 [1/0] via 30.0.0.1.

Screen Shots:

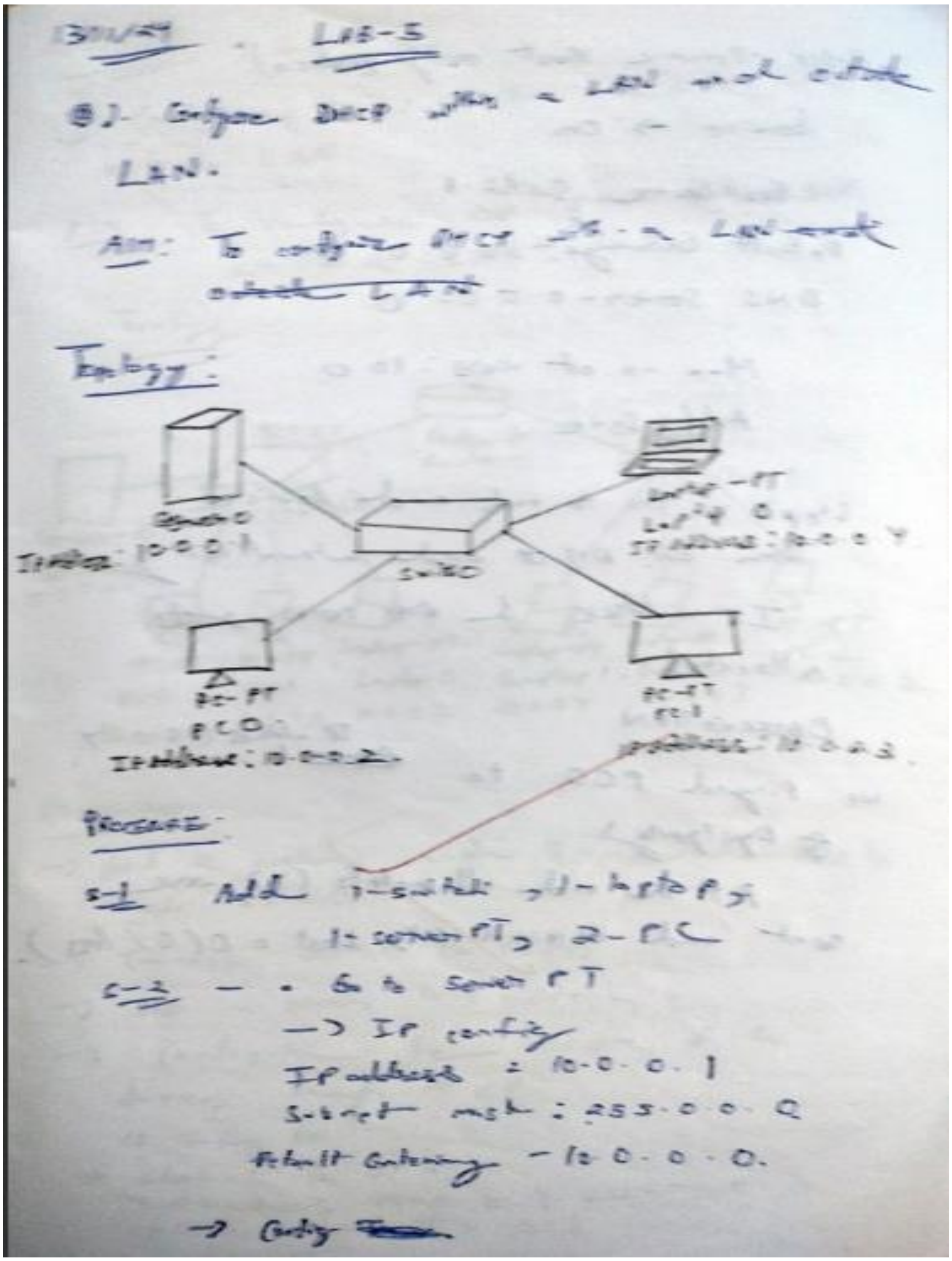




Program 5

Aim: Configure DHCP within a LAN and outside LAN.

Topology , Procedure and Observation:



13/11/24 LAB-5

Q) Configure DHCP within a LAN and outside LAN.

Aim: To configure DHCP within a LAN and outside LAN

Topology:

Procedure:

S-1 Add 1-switch, 1-laptop PT, 1-server PT, 2-PC

S-2 - Go to Server PT

→ IP config

IP address : 10.0.0.1

Subnet mask : 255.0.0.0

Default Gateway - 10.0.0.0.

→ Config

DHCP - Dynamic Host config protocol.

Service → On

Pool Name: Switch 1

Default Gateway - 10.0.0.0

DNS Server - 0.0.0.0

Max no of users : 100.

Add, Save.

Step 3 - Go to each end device
Turn on DHCP and automatically
IP address & DNS server gets
allocated.

OBSERVATION

We Pinged PC0 to ~~PC~~ 1. IP address 10.0.0.4,
By Typing)

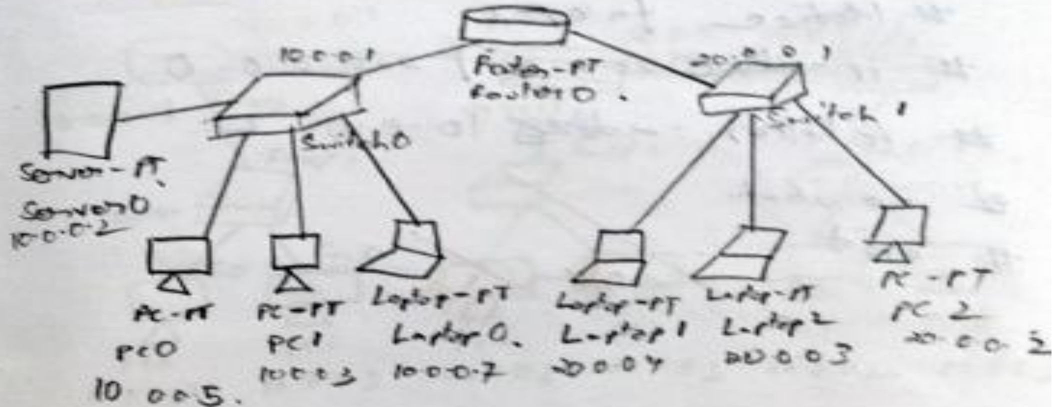
At the end all packets (4) were
sent and Received = 4, Lost = 0 (0% loss).

15/11/24 LAB-5

(5) Configure DHCP within a LAN and outside LAN.

AIM: To configure ~~on~~ DHCP ~~within and~~ outside a LAN.

Topology:



PROCEDURE

→ Add 2 switches and 3 end devices to each switch

→ Connect switch to router

→ Add a server to 1 switch

→ Configure the rest of the devices through DHCP.

\# Config t

\# interface fa 0/0
ip address 10.0.0.1 255.0.0.0
ip helper address 10.0.0.1
no shut

exit

Observed

The ping is sent and other assigning to all end devices.

2nd network CLI:

interface fa0/1

ip address 20.0.0.1 255.0.0.0

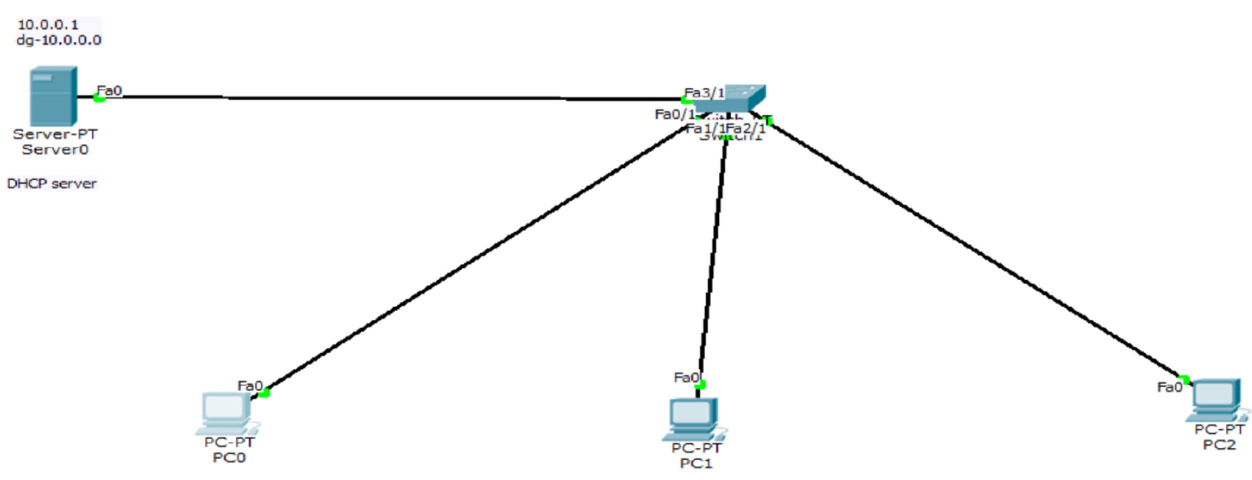
ip helper-address 10.0.0.2

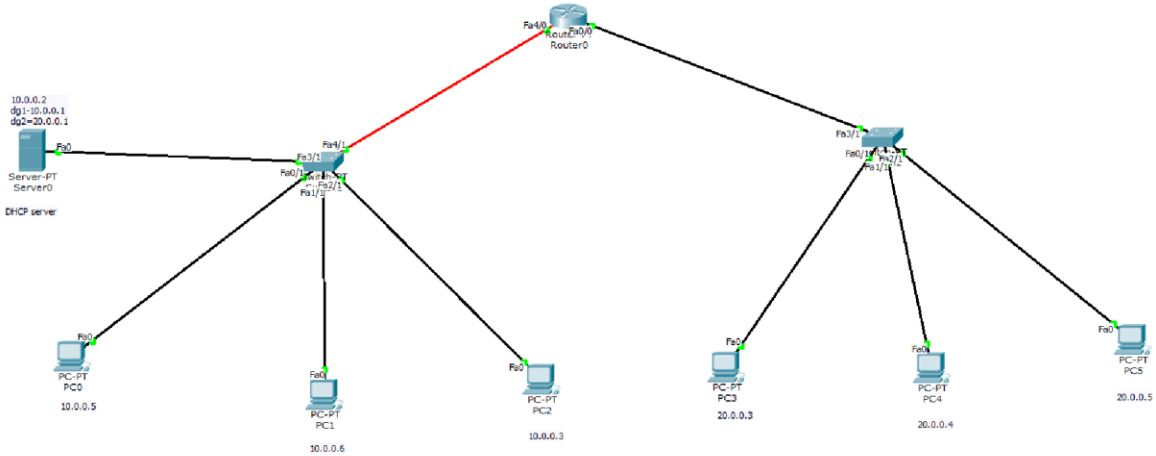
no shut

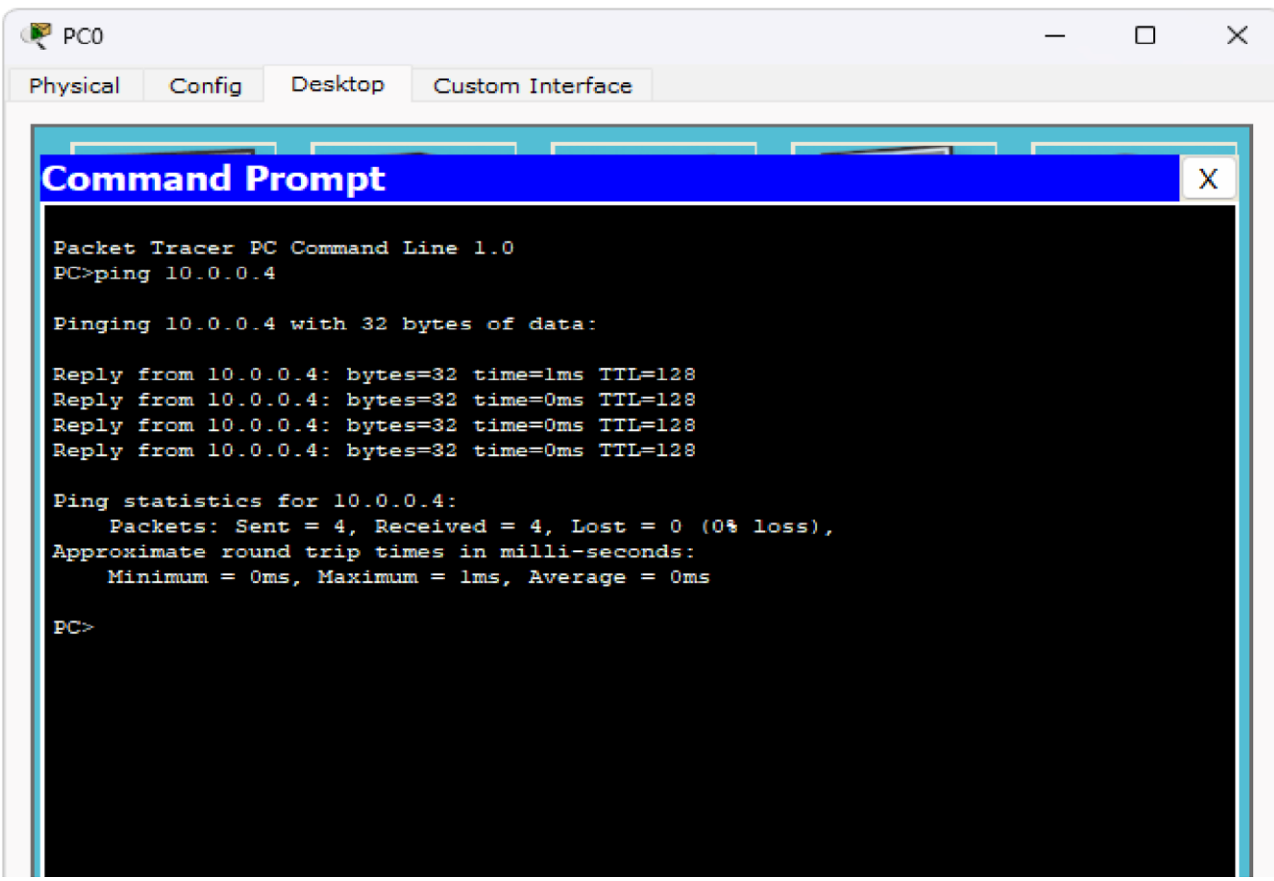
exit

13/11

Screen Shots:

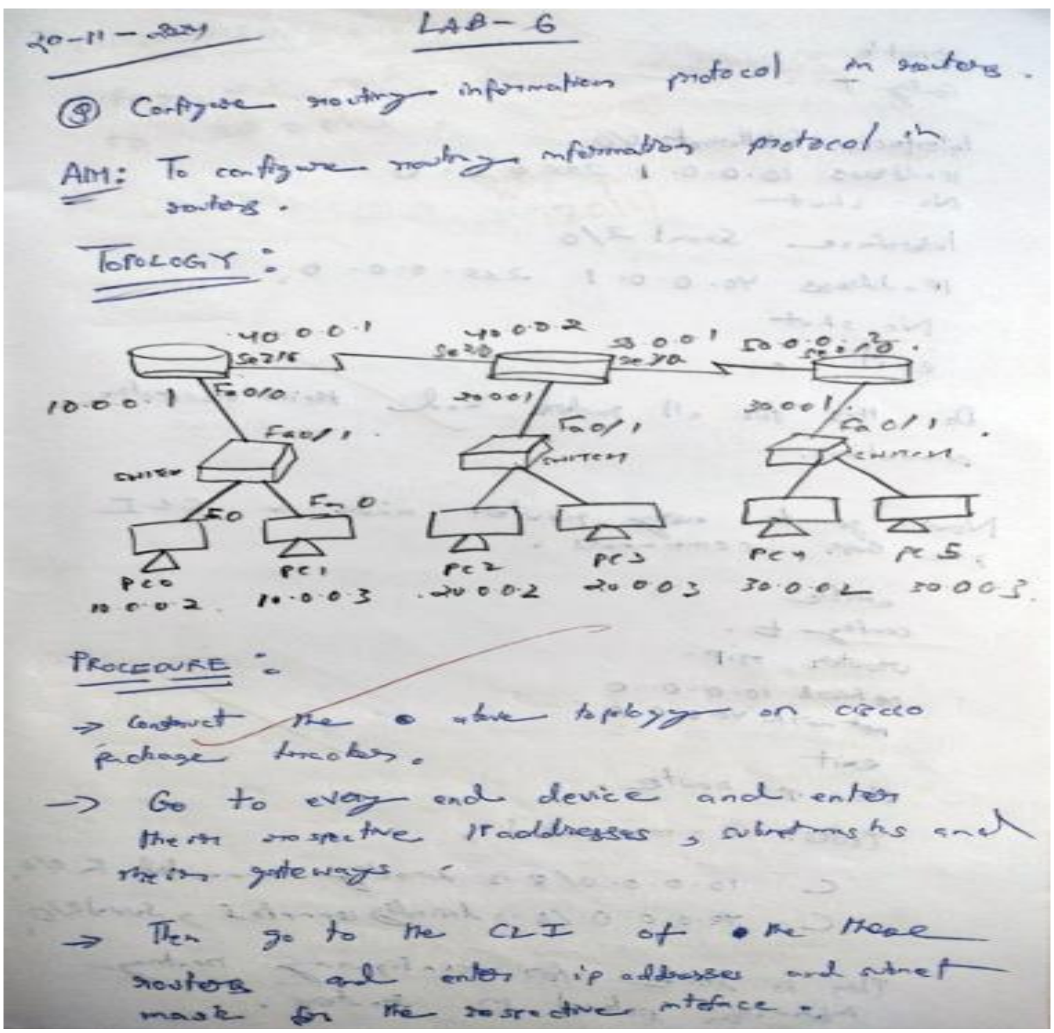



10.0.0.2
dg1=10.0.0.1
dg2=20.0.0.1
Server-PT
Server0
DHCP server
Router0
PC-PT
PC0
10.0.0.5
PC-PT
PC1
10.0.0.6
PC-PT
PC2
10.0.0.3
PC-PT
PC3
20.0.0.3
PC-PT
PC4
20.0.0.4
PC-PT
PC5
20.0.0.5


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.4
Pinging 10.0.0.4 with 32 bytes of data:
Reply from 10.0.0.4: bytes=32 time=1ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Ping statistics for 10.0.0.4:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
PC>

<u>Program 6</u>
Aim: Configure RIP routing Protocol in Routers .

Topology , Procedure and Observation:


20-11-2024

LAB-6

(5) Configure routing information protocol in routers.

AIM: To configure routing information protocol in routers.

TOPOLOGY:

PROCEDURE:

-> Construct the above topology on cisco packet tracer.

-> Go to every end device and enter their respective IP addresses, subnet masks and their gateways.

-> Then go to the CLI of the three routers and enter ip addresses and subnet mask for the respective interface.

enable

Config T.

Interface fastethernet 0/0
ip address 10.0.0.1 255.0.0.0
No. shut

Interface Serial 2/0

IP address 40.0.0.1 255.0.0.0,

No shut

exit.

Do this for all router and their respective connections.

Now go to every router and in CLI give these commands.

enable
config t.
router rip.
network 10.0.0.0
network 40.0.0.0.
exit
show ip route.
Codes: C - connected is

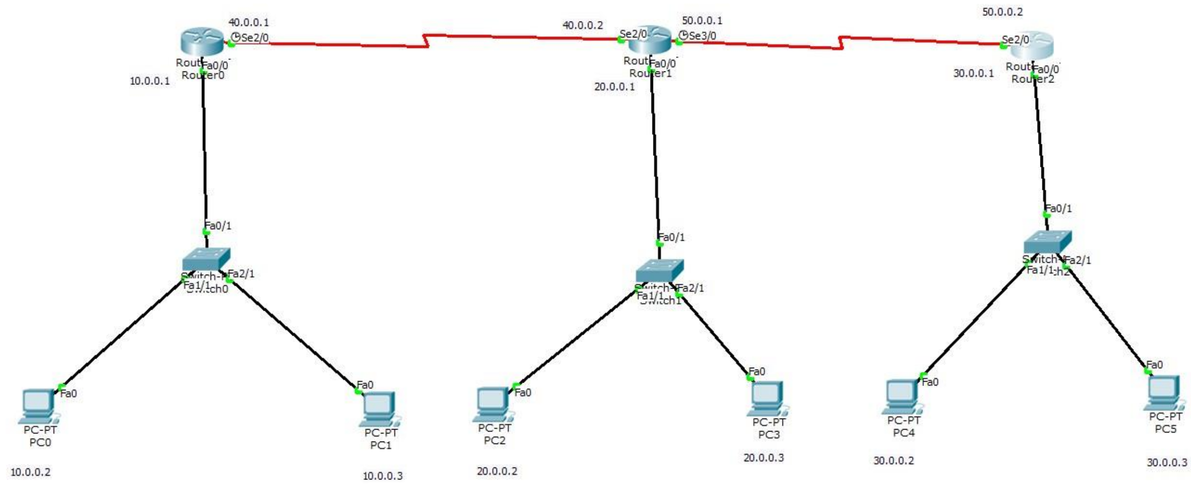
C 10.0.0.0/8 is directly connected, Fa 0/0,
C 40.0.0.0/8 is directly connected, Serial2/0

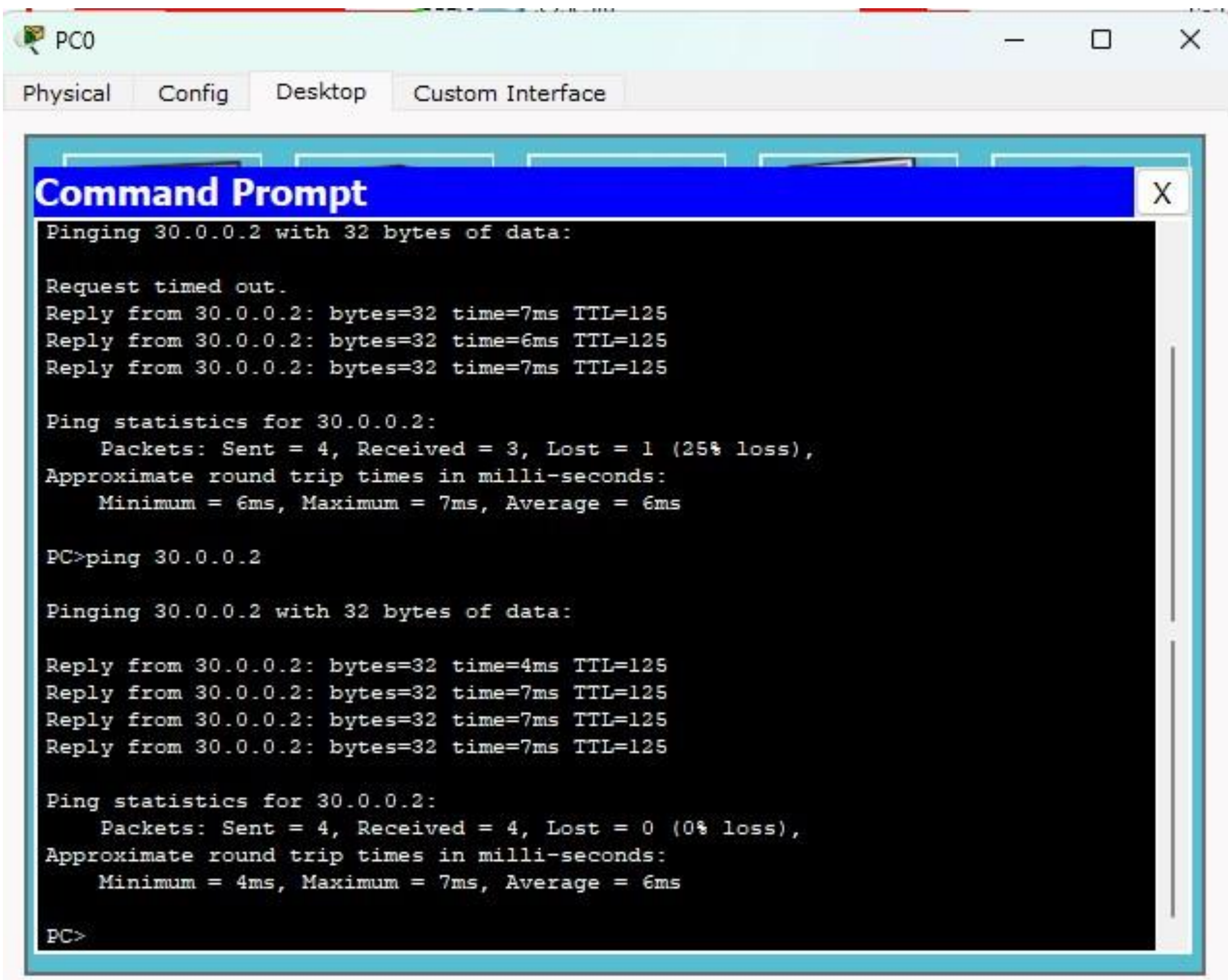
This is done for configuring routing information protocol in routers.

OBSERVATION

Ping was sent from IP address 10.0.0.2 to 30.0.0.3.

Screen Shots:


40.0.0.1
Se2/0
40.0.0.2
Se2/0
50.0.0.1
Se3/0
50.0.0.2
Se2/0
Fa0/0
Router0
10.0.0.1
Fa0/0
Router1
20.0.0.1
30.0.0.1
Fa0/0
Router2
Fa0/1
Fa1/1
Fa2/1
Switch0
Fa0/1
Fa1/1
Fa2/1
Switch1
Fa0/1
Fa1/1
Fa2/1
Switch2
Fa0
PC-PT
PC0
10.0.0.2
Fa0
PC-PT
PC1
10.0.0.3
Fa0
PC-PT
PC2
20.0.0.2
Fa0
PC-PT
PC3
20.0.0.3
Fa0
PC-PT
PC4
30.0.0.2
Fa0
PC-PT
PC5
30.0.0.3


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Pinging 30.0.0.2 with 32 bytes of data:
Request timed out.
Reply from 30.0.0.2: bytes=32 time=7ms TTL=125
Reply from 30.0.0.2: bytes=32 time=6ms TTL=125
Reply from 30.0.0.2: bytes=32 time=7ms TTL=125
Ping statistics for 30.0.0.2:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 6ms, Maximum = 7ms, Average = 6ms
PC>ping 30.0.0.2
Pinging 30.0.0.2 with 32 bytes of data:
Reply from 30.0.0.2: bytes=32 time=4ms TTL=125
Reply from 30.0.0.2: bytes=32 time=7ms TTL=125
Reply from 30.0.0.2: bytes=32 time=7ms TTL=125
Reply from 30.0.0.2: bytes=32 time=7ms TTL=125
Ping statistics for 30.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 4ms, Maximum = 7ms, Average = 6ms
PC>

Program 7
Aim: Demonstrate the TTL/ Life of a Packet .

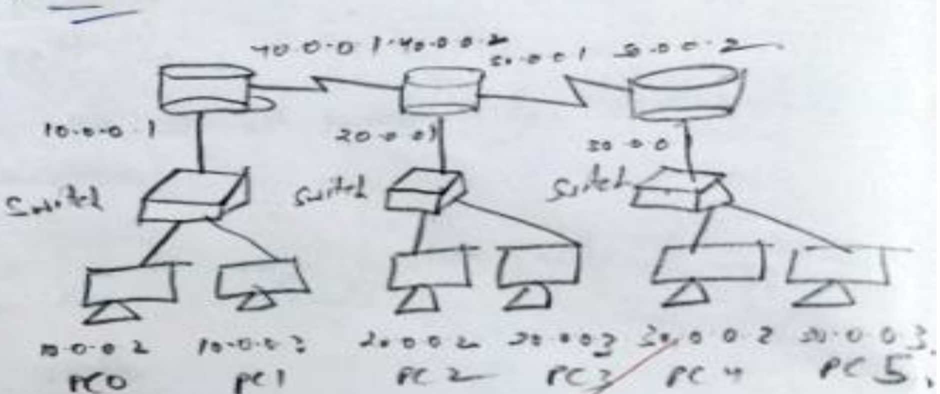
Topology , Procedure and Observation:

20-11-2024 LAB-6

Q. Demonstrate TTL or life of a packet.

AIM: To demonstrate TTL or life of a packet

TOPOLOGY::



PROCEDURE:

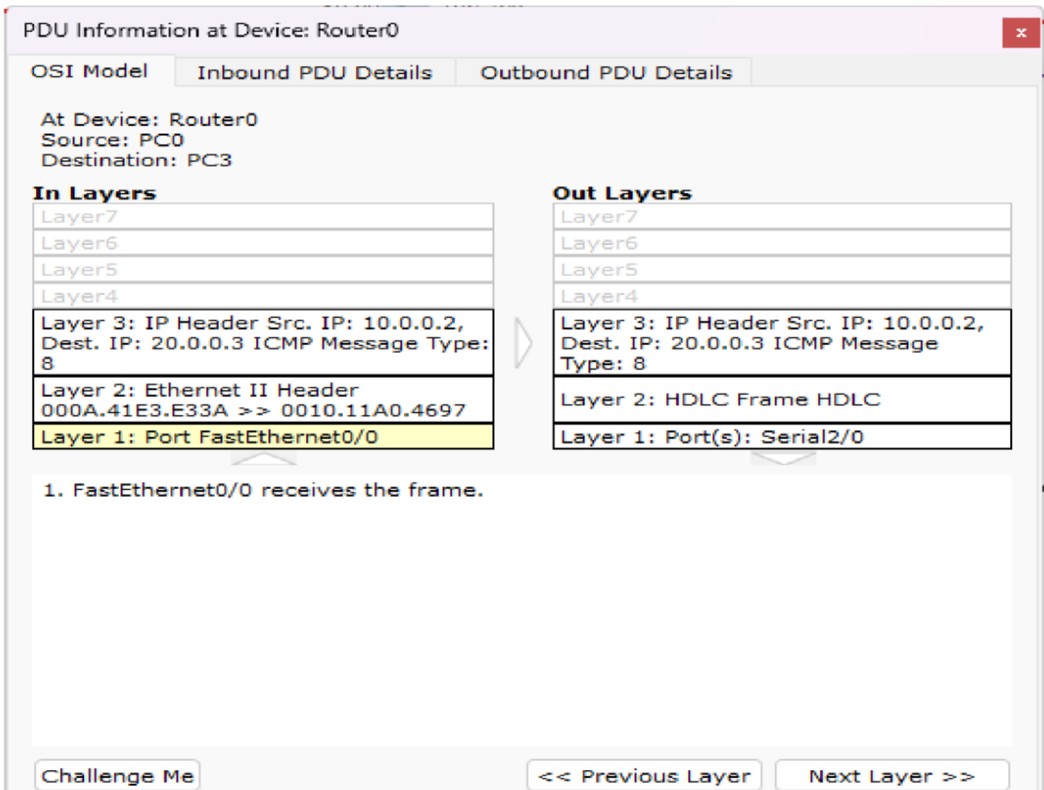
The configuration has been configured and established.

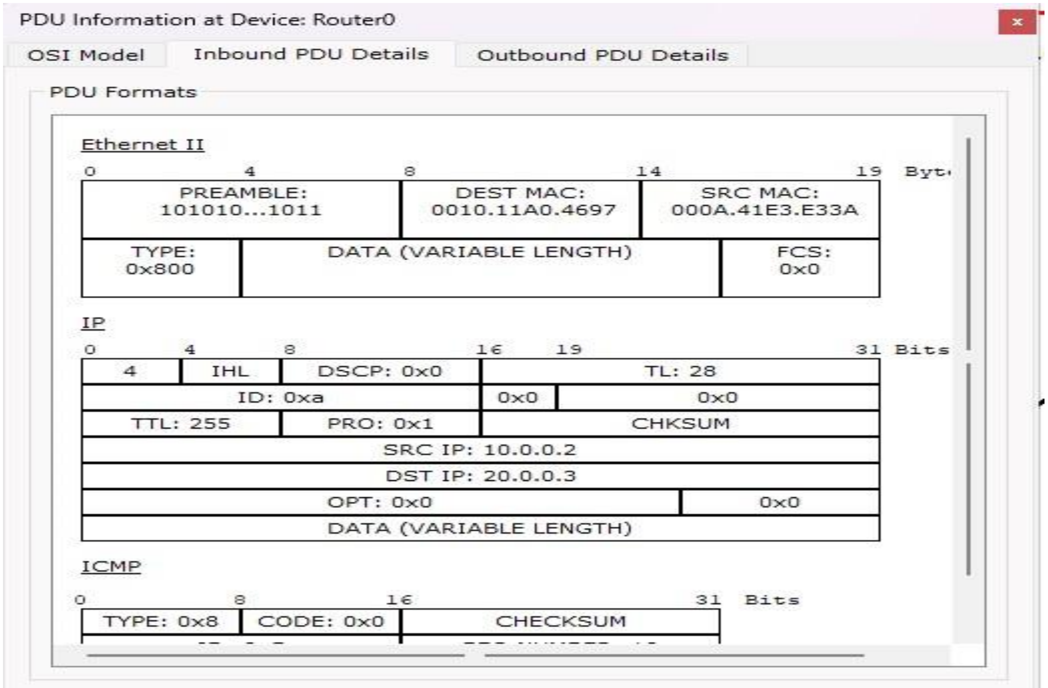
Now start the simulation and send a package from the system with IP address 10.0.0.2 to 30.0.0.3 and click on auto capture/play button, the simulation would start and it would end when TTL becomes 0 and the packet would be removed .

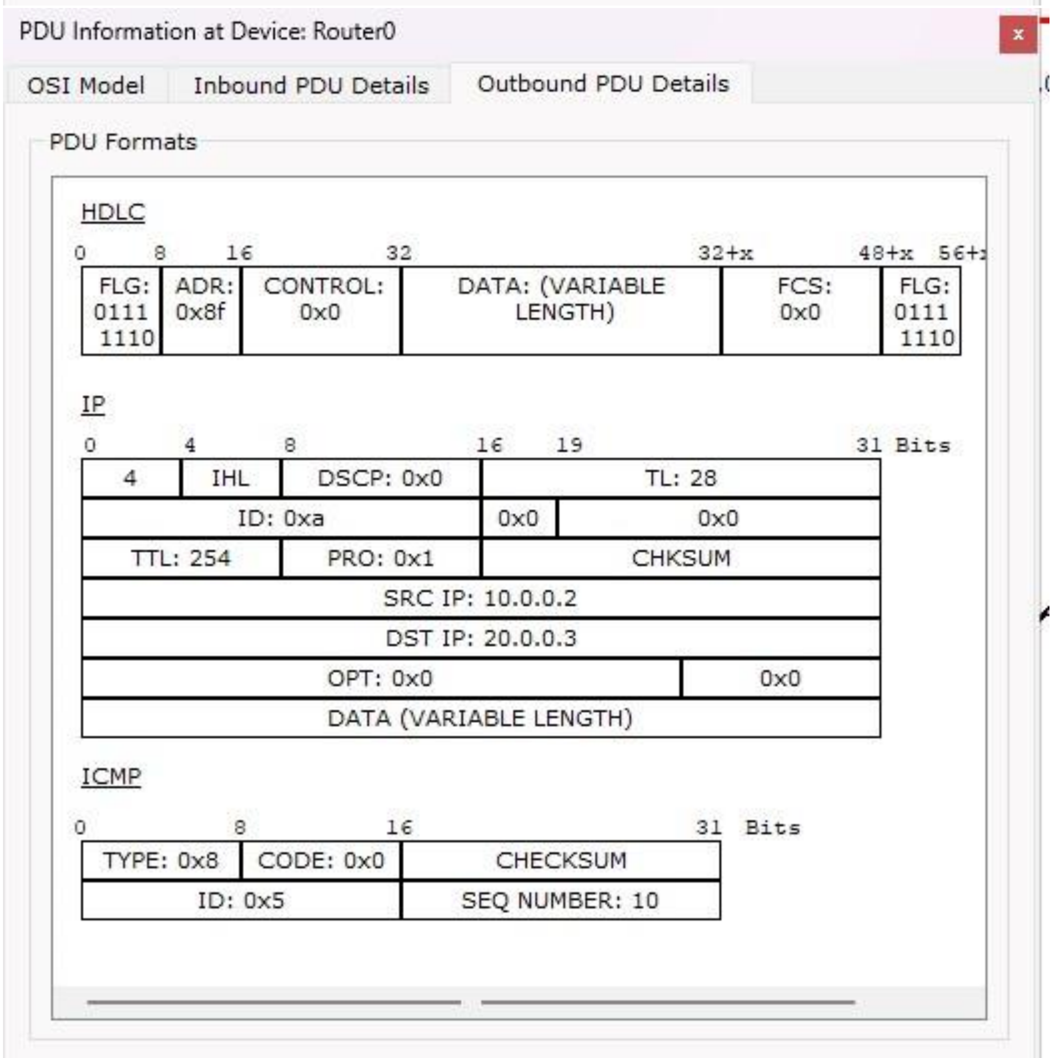
OBSERVATION :

We notice that the TTL decrease when the message is sent from one router to other.

Screen Shots:




PDU Information at Device: Router0
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
0 4 8 14 19 Byt
PREAMBLE: 101010...1011
DEST MAC: 0010.11A0.4697
SRC MAC: 000A.41E3.E33A
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
0 4 8 16 19 31 Bits
4
IHL
DSCP: 0x0
TL: 28
ID: 0xa
0x0
0x0
TTL: 255
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.2
DST IP: 20.0.0.3
OPT: 0x0
0x0
DATA (VARIABLE LENGTH)
ICMP
0 8 16 31 Bits
TYPE: 0x8
CODE: 0x0
CHECKSUM


PDU Information at Device: Router0
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
0 8 16 32 32+x 48+x
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 .

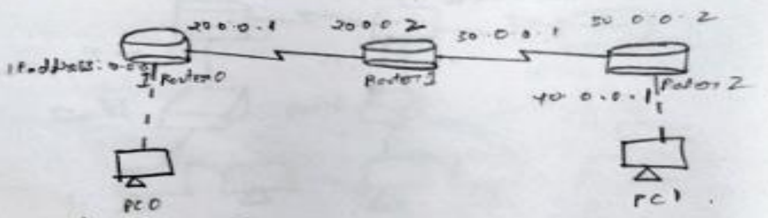
Topology , Procedure and Observation:

LAB-7

Q. To Demonstrate OSPF routing protocol using different routers and networks.

AIM : To demonstrate OSPF routing protocol using different routers and networks.

Topology:



IP address: 10.0.0.10

IP address : 40.0.0.10.

PROCEDURE:

S1: Create the above topology.

S2: Configure ip address to all interfaces. to Routers

For all connection to end devices just enter the respective ip address and subnetmask.

For one end of serial interface enter commands

```
# interface serial 2/0
# ip address 20.0.0.1 255.0.0.0
# encapsulation PPP
# clock rate 64000
# no shut
# exit.
```

And for other end of a serial connection just enter encapsulation ppp and not the clock rate.

S-3 Next, Enable ip routing by configuring ospf routing protocol in all routers,

In Router R1,

```
R1(config) # router ospf 1
R1(config-router) # router-id 1.1.1.1
R1(config-router) # network 10.0.0.0 0.255.255.255 area 3
R1(config-router) # network 20.0.0.0 0.255.255.255 area 1
R1(config-router) # exit
```

Give similar commands for all routers.

S-4 Check routing table of R1,

```
Router# show ip route
```

Codes: C - connected

:

:

S-5: Check routing table of R3,

```
R3# show ip route
```

Codes: C - connected

:

S-6: Create virtual link between R1, R2, by this we create a virtual link to connect area 3 to area 0.

R1,

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

Do similar for R2.

S7: R2 & R3 get updates about Area 3. Now, check routing table of R3.

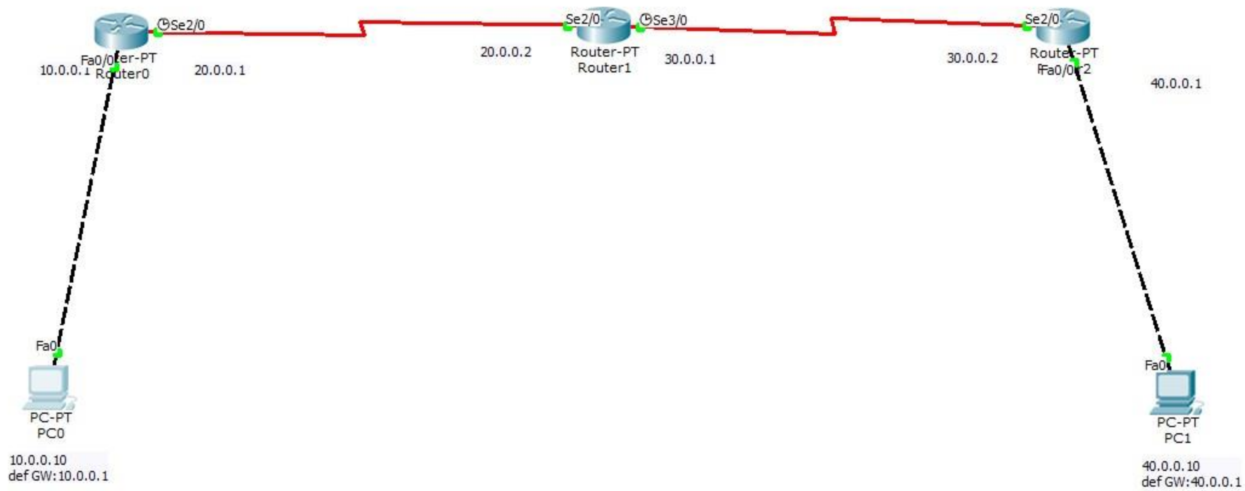
R3# show ip route.

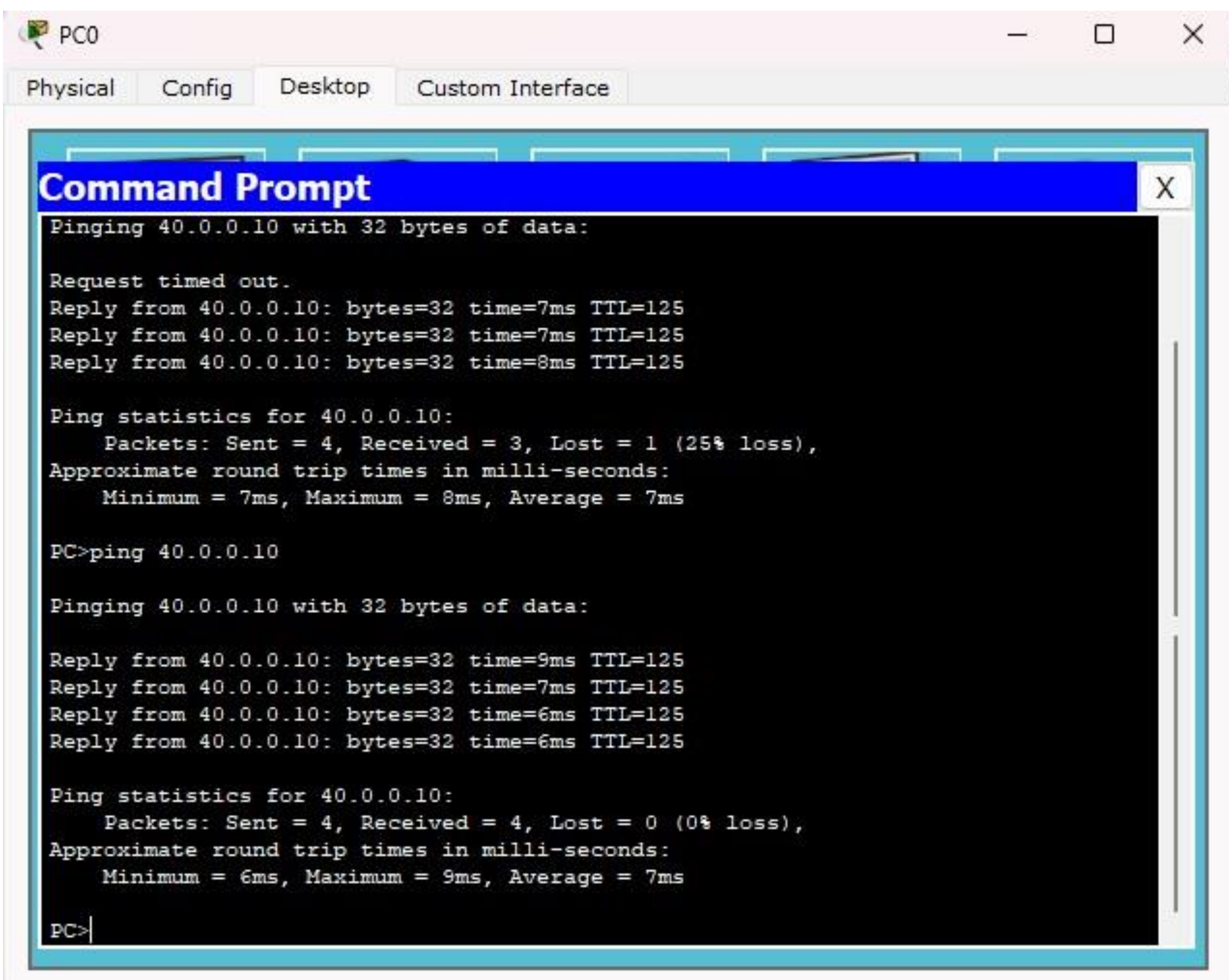
S8: Check connectivity between host 10.0.0.10 to 40.0.0.10

Observation:

We can successfully send ping from 10.0.0.10 to 40.0.0.10.

Screen Shots:


Se2/0
Router0
10.0.0.1
20.0.0.1
20.0.0.2
Se2/0
Router-PT
Router1
Se3/0
30.0.0.1
30.0.0.2
Se2/0
40.0.0.1
Fa0
PC-PT
PC0
10.0.0.10
def GW:10.0.0.1
Fa0
PC-PT
PC1
40.0.0.10
def GW:40.0.0.1


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Pinging 40.0.0.10 with 32 bytes of data:
Request timed out.
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Reply from 40.0.0.10: bytes=32 time=8ms TTL=125
Ping statistics for 40.0.0.10:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 7ms, Maximum = 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=9ms 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 = 9ms, Average = 7ms
PC>

Program 9

Aim: Configure Web Server, DNS within a LAN.

Topology , Procedure and Observation:

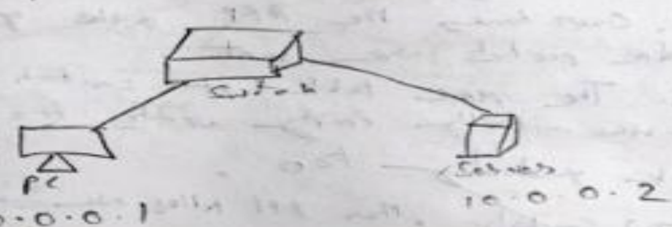
18/12/23

LAB-11

DNS

AIM: Configure Web Server, DNS within a LAN.

Topology:



PROCEDURE:

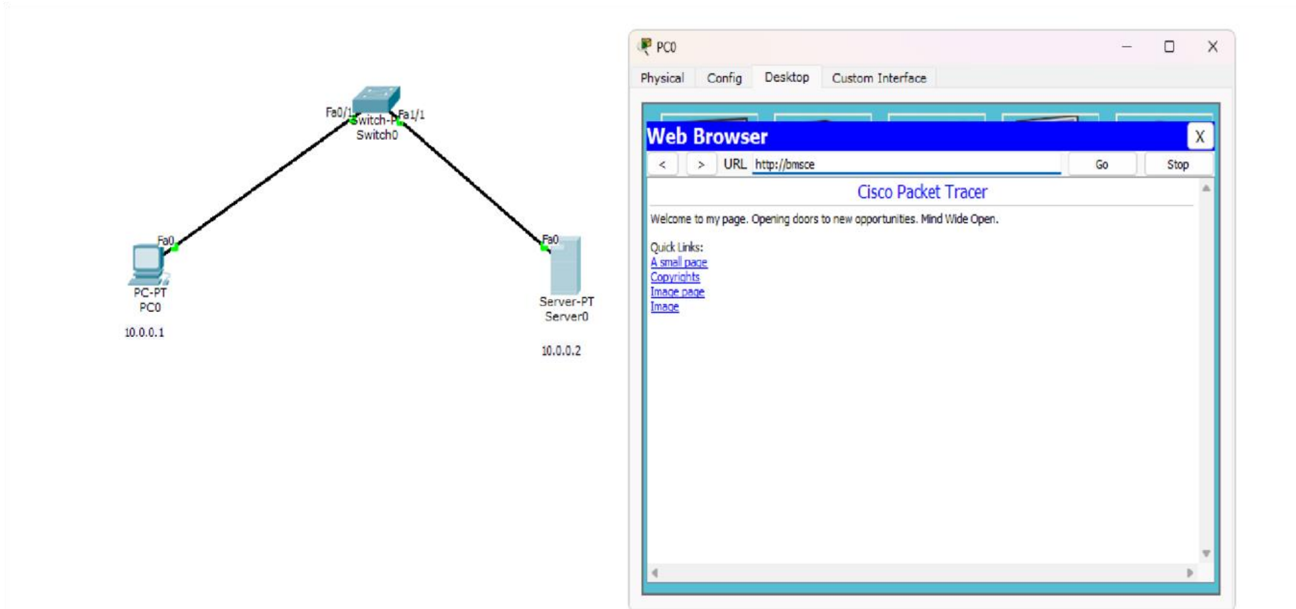
1) Set up the LAN as per topology mentioned above & configure devices.
2) Go to Server -> Services -> DNS:
   Name: bmsce [ur name]
   Address: 10.0.0.2
   Add the mapping of domain name to address
3) Go to PC -> Config -> Global -> Settings ->
   DNS server: 10.0.0.2
4) Go to PC -> Desktop -> Web Browser
   Type URL: http://bmsce

Observations:

1) The webpages hosted by server were visible on the browser.

2) DNS was ...
domain name to the IP address.
3) DNS server is a server that contains
a domain name : IP address mapping
to which the end devices send requests
to get the Name to IP addresses.

Screen Shots:



Program 10

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

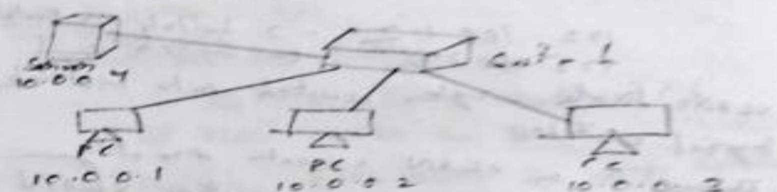
Topology , Procedure and Observation:

18/12/24

LAB - 10

AIM : To construct simple LAN & understand the concept and operation of ARP.

TOPOLOGY :



PROCEDURE :

1) Create the topology as shown above.
2) Configure the PCs and the servers.
3) Click on Inspect mode, then click on the end devices and open ARP tables.
4) Send a data packet from any end device say server to say 10.0.0.3 PC.
5) Grow Simulation mode to capture each step of data transfers.

OBSERVATIONS :

1) The arp tables of all end devices are initially empty.

2) When the data packet from server arrives at the switch, since the source MAC address is unknown, it sends a broadcast message to all devices.

3) The device with IP address present in the destination address of the data packet responds to message.

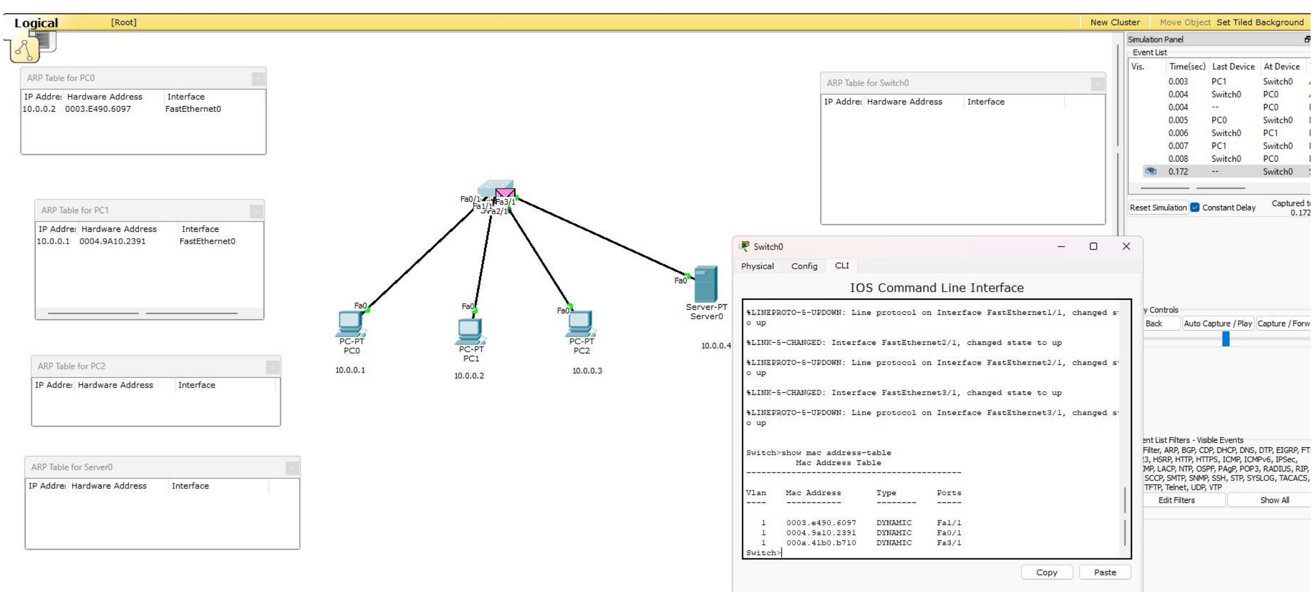
4) The server and PC update their ARP tables matching IP addresses to MAC address.

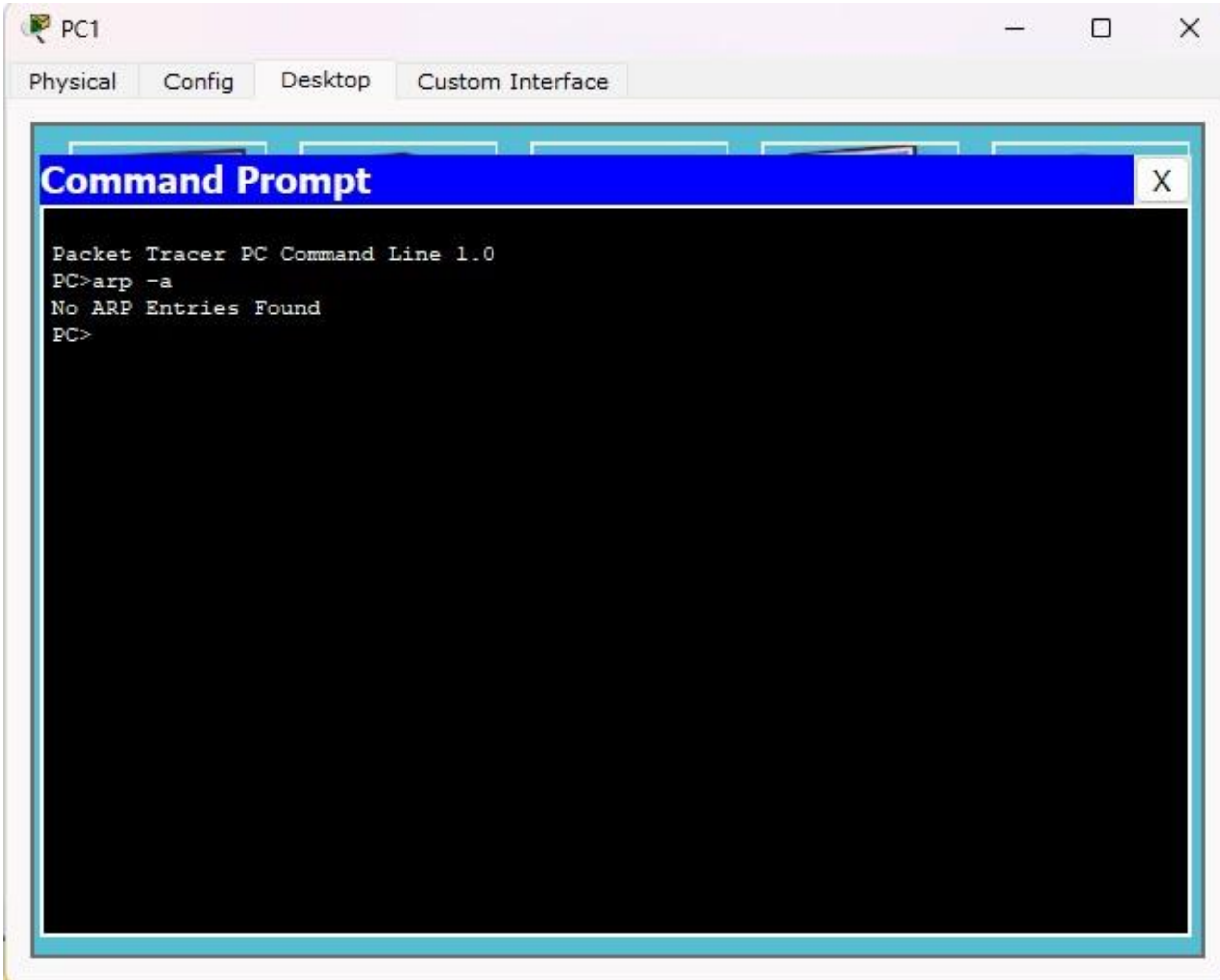
5). Over time the ARP tables grows as data packets are sent.

6). The MAC table of switch which was initially empty updates its MAC table gradually too.

7) Similarly other ARP tables are updated.

Screen Shots:

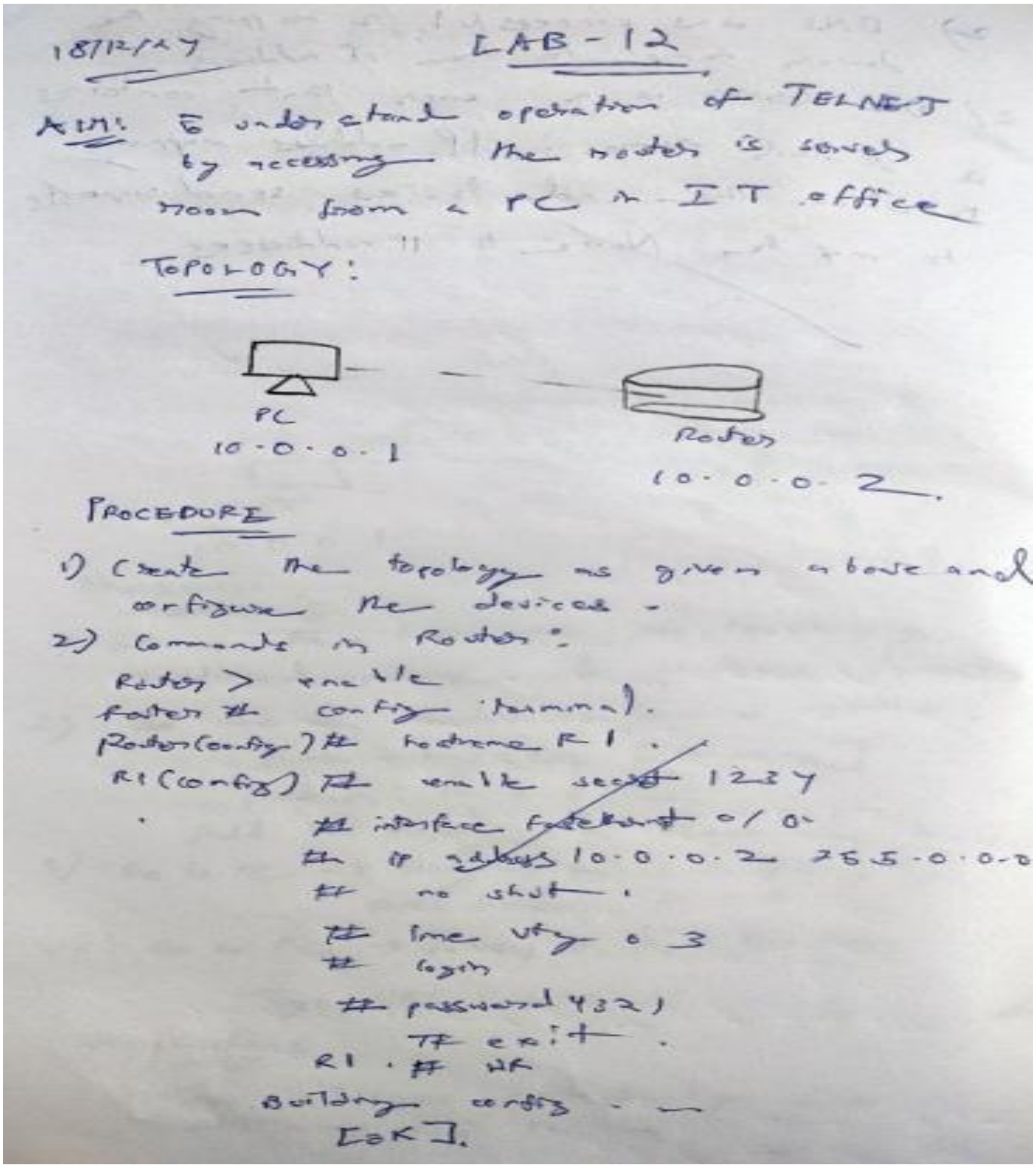



PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>arp -a
No ARP Entries Found
PC>

Program 11

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

Topology , Procedure and Observation:



18/12/24

LAB - 12

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

TOPOLOGY:

PC
10.0.0.1

Router
10.0.0.2.

PROCEDURE

1) Create the topology as given above and configure the devices.

2) Commands in Router:

Router > enable

Router # config terminal.

Router(config)# hostname R1.

R1(config)# enable ~~secret~~ 1234

\# interface fastethernet 0/0

\# ip address 10.0.0.2 255.0.0.0

\# no shut.

\# line vty 0 3

\# login

\# password 4321

\# exit.

R1.# wr

Building config ...

[OK].

Note : vty 0 2 : First few virtual terminal lines for Telnet access.

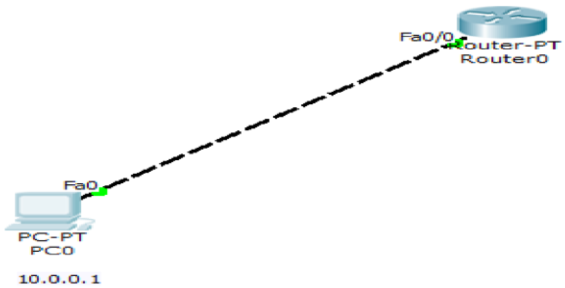
3) In PC : command Prompt :
- First try pinging to see if devices are connected

PC > telnet 10.0.0.2
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, fastethernet 0/0
~~R1 #~~

Observations :

1) Admin in PC is able to run commands as if in router's CLI and see results on from PC.
2) Telnet allows users to establish a remote session with another device like routers over TCP/IP n/w.
3) Using Telnet, we can access & control remote device's CLI as if you were physically connected to it.

Screen Shots:



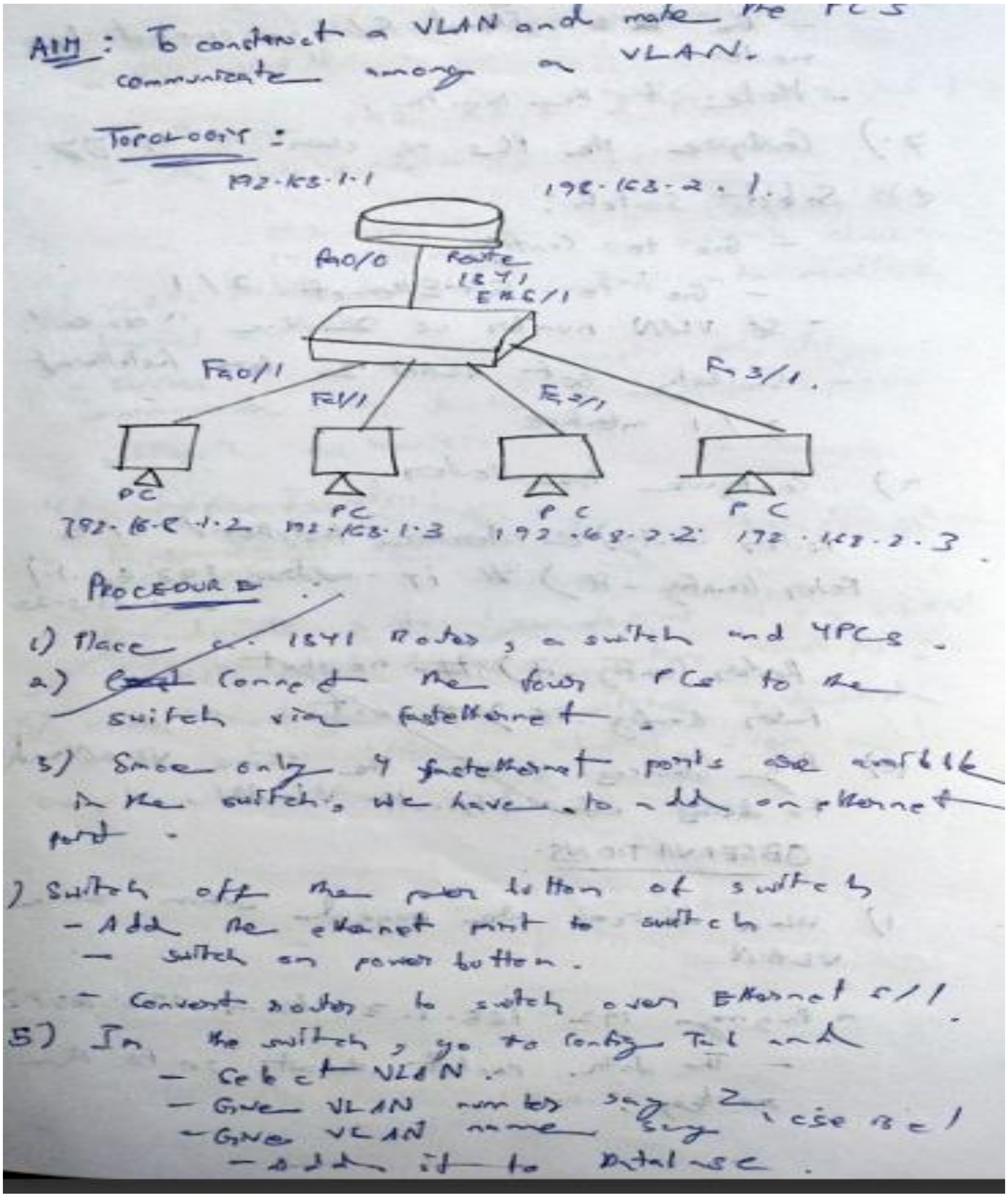
Command Prompt

```
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:



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

TOPOLOGY :

192.168.1.1 192.168.2.1

Fa0/0 Router 1841 Eth6/1

Fa0/1 Fa1/1 Fa2/1 Fa3/1.

PC PC PC PC

192.168.1.2 192.168.1.3 192.168.2.2 192.168.2.3 .

PROCEDURE :

1) Place a 1841 Router, a switch and 4PCs .
2) ~~Conn~~ Connect the four PCs to the switch via fastethernet .
3) Since only 4 fastethernet ports are available in the switch, we have to add an ethernet port .
4) Switch off the power button of switch
   - Add the ethernet port to switch
   - switch on power button .
   - Connect router to switch over Ethernet 6/1 .
5) In the switch, go to Config Tab and
   - Select VLAN .
   - Give VLAN number say 2
   - Give VLAN name say [illegible]
   - Add it to Database .

6) Select the switch:

— Go to Config

— Go to Ethernet 1/1 i.e. connect to router.

— Make it the trunk.

7.) Configure the PCs as shown in topology.

8) Select switch:

— Go to Config

— Go to FastEthernet 2/1

- Set VLAN number as 2 i.e. access.

-- Similarly set VLAN 2 for FastEthernet 3/1 interface.

9). Configure the router,

Router (config) # interface fastethernet 0/0

Router (config-if) # ip-address 192.168.1.1 255.255.255.0

Router (config-if) # no shut

Router (config-if) # exit

10) Ping devices within the same VLAN and to devices of a different VLAN.

OBSERVATIONS.

1) When devices are pinged within same VLAN.

→ Pinging 192.168.1.3 from 192.168.1.2

— The data packet doesn't go to the router,

- The switch forwards the packet without a need of the router.

2) When a device pings a device of another VLAN.

- Pinging 192.168.2.3 from 192.168.1.2
- The data packet is as follows:

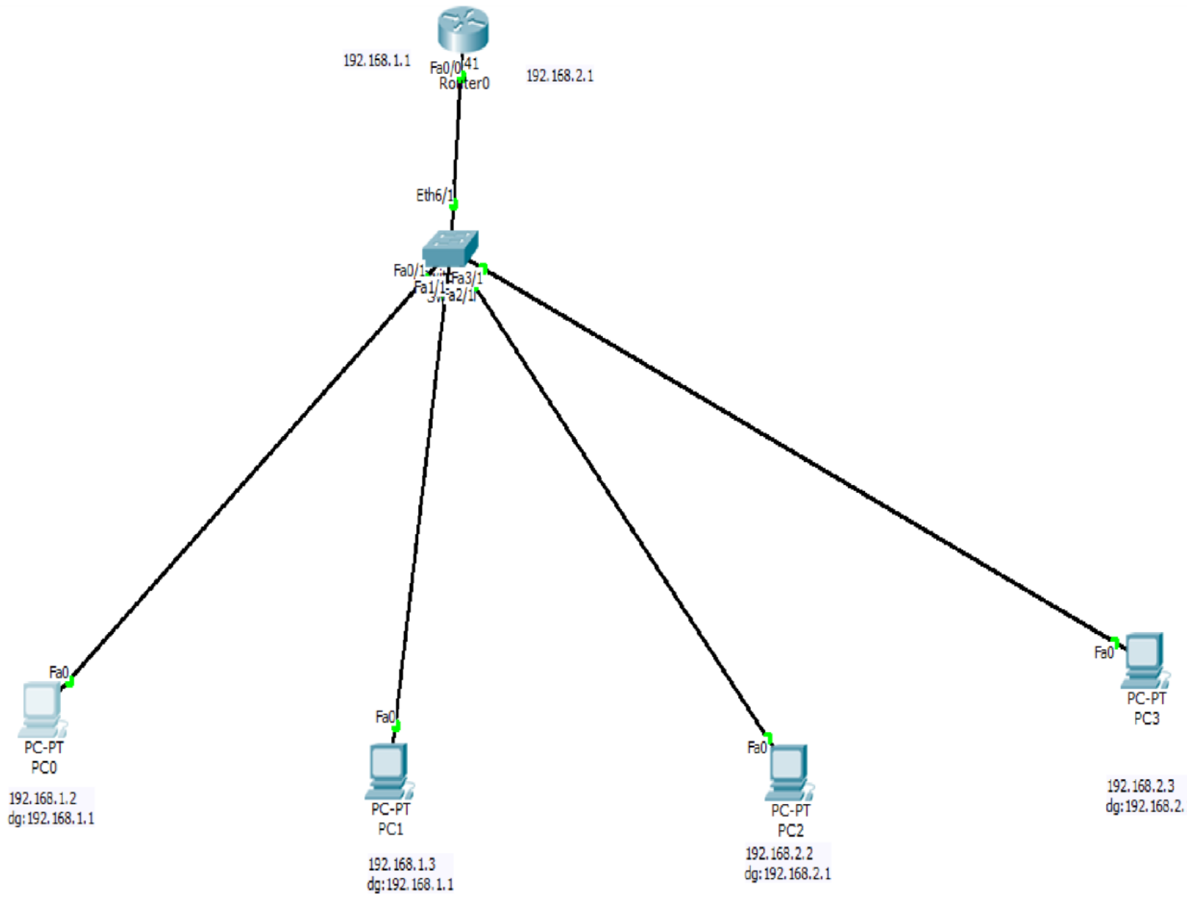
192.168.1.2 -> switch -> Router
192.168.2.3 <- switch and

3) VLANs divide a single switch into multiple logical switches.
- Devices in one VLAN cannot directly communicate with devices in another VLAN without a router.

4) Traffic Isolation:
- Each VLAN maintains its own broadcast domain.
- Broadcasts sent by devices in one VLAN do not reach devices in another VLAN.

5) VLAN trunking allows switches to forward frames from different VLANs over a single ~~~~ link called trunk.

Screen Shots:



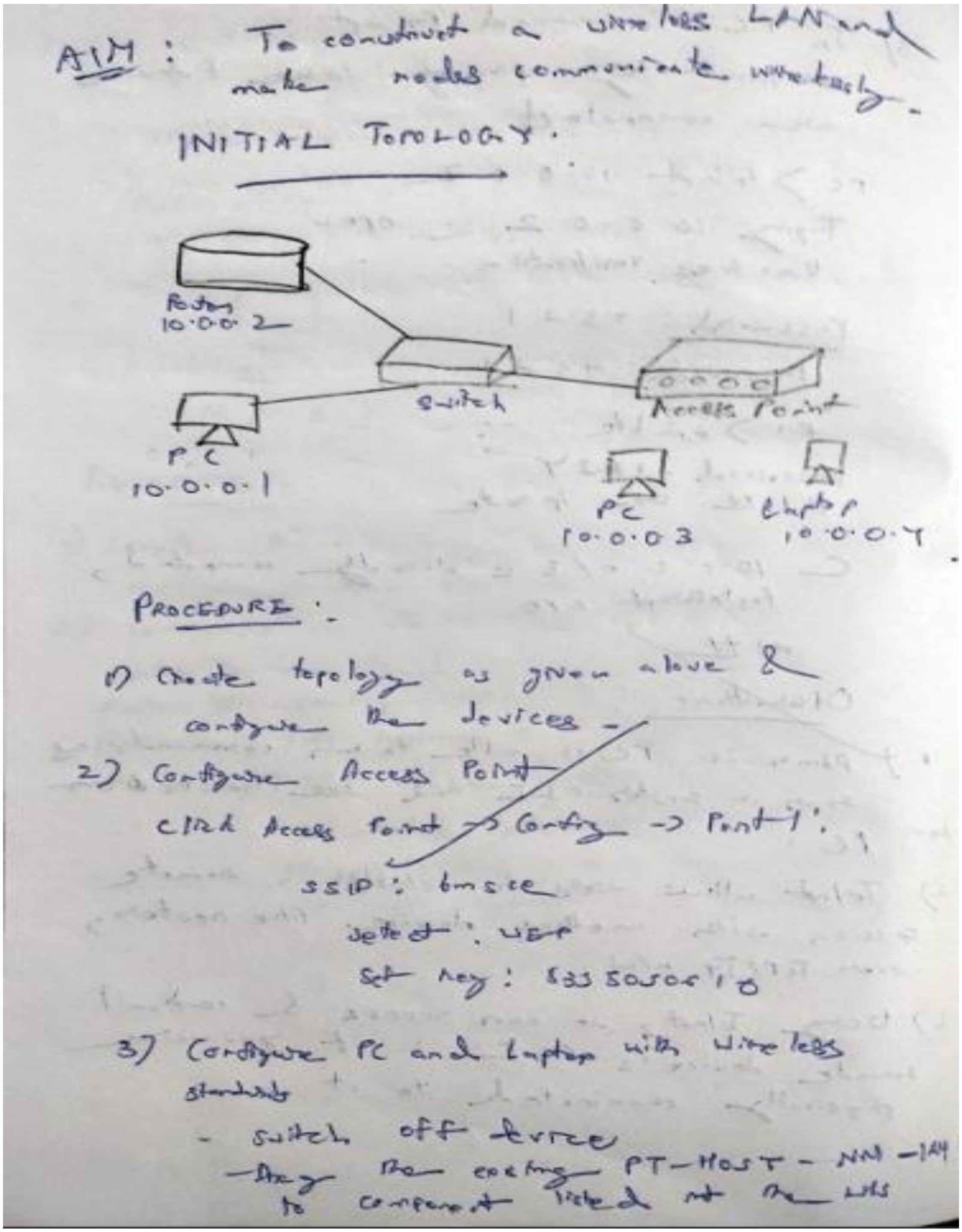
Command Prompt

```
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:



AIM : To construct a wireless LAN and make nodes communicate wirelessly.

INITIAL TOPOLOGY.

PROCEDURE :-

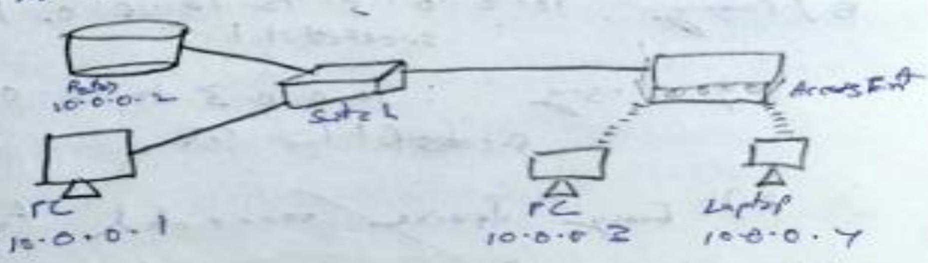
1) Create topology as given above & configure the devices.

2) Configure Access Point
click Access Point -> Config -> Port 1.
SSID : bmsce
select : WEP
Set key : 5335050510

3) Configure PC and Laptop with wireless standards
- switch off device
- drag the existing PT-HOST-NM-1AM to component listed at the LHS

of Physical.
- Drag WMP 300N wireless interface to the empty port.
- Switch on the device.
4). In the config tab, a new wireless interface was added.
5) Configure the device by entering SSID, WEP, WEP key, IP address and gateway.

After wireless connection:



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

Observations:

1) We were able to ping from every device to every other device.
2) Access Point:
Creates bridge between wired & wireless devices.
- SSID broadcasting: announces the wireless network's name (SSID) to allow devices to connect using WEP, WPA or WPA2

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 → Switch → Access Point → 10.0.0.3

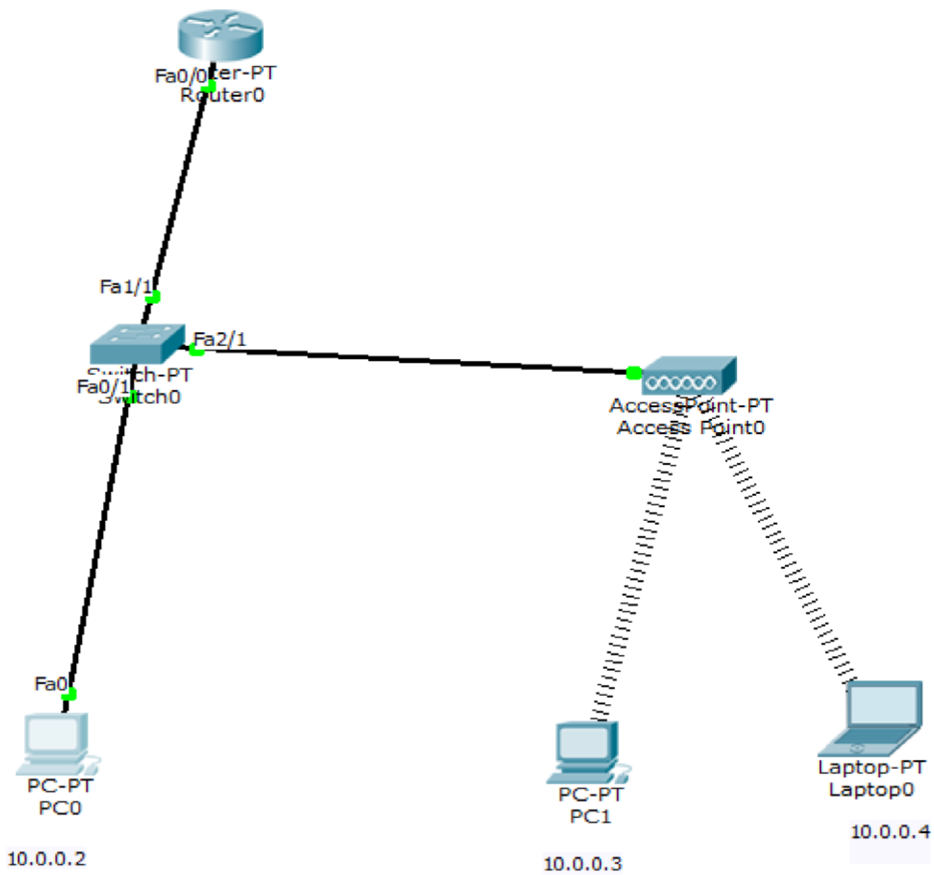
— This is after ARP tables are updated.

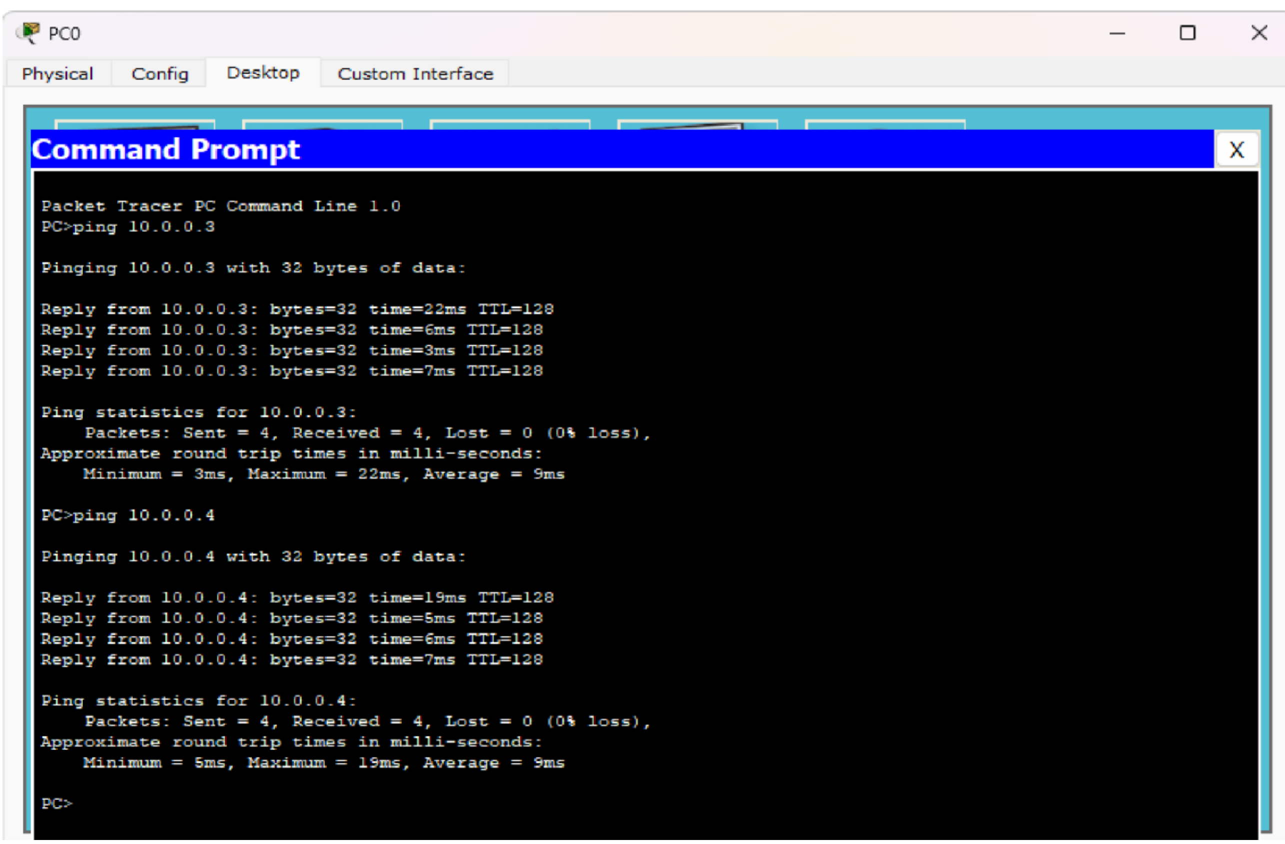
5) Pinging : 10.0.0.3 to 10.0.0.1
successful

6) Pinging : 10.0.0.3 to 10.0.0.7:
successful.

7) Every device connected to every other using WLAN

Screen Shots:





PART-B

Program 14

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

Code :

```
# 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 mod2div (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
    return checkword.
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("Encoded Data (Data+Remainder) :", [illegible])

data = "100100"
key = "1101"
encodedata (data, key)
```

Output

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

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

Program 15

Write a program for congestion control using Leaky bucket algorithm.

Code :

Leaky Bucket

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 shaping :—

1. Leaky bucket
2. Token bucket

Ex: Let n = 1000.

Packets = 200 700 500 450 400 200

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

Therefore, n = 1000 − 200 = 800.

Packet size of 200 is sent into network. 200 700 500 450 400

Now, again n > size of packet at 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 inputs");
scanf ("%d %d %d", &bucket_size, &outgoing, &n);
while (n != 0) {
    pf ("Enter incoming packet size");
    scanf ("%d", &incoming);
```

```
pf ("Incoming packet size %d\n", incoming)
                                                    ;
  if ( incoming <= (bucket-size - store)) {
          store += incoming;
  pf ("Bucket buffer size %d out of %d\n",
                                                    ;
        store, buck-size ));
    }
     else {
pf ("Dropped %d no. of packets \n", incoming.
        (buck-size ) - store )) ;
pf (" Bucket buffer size %d out of %d\n",
        store, buck-size );
            Store = buck - size ;
      }
    Store = store - outgoing ;
    pf (" After outgoing %d bytes left out
        %d in buffer \n", store, buck-size);
              n - - ;
     }
 }
```

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:

(i) AIM : Implementation of TCP/IP :

(ii) Code

```
Client.py
from socket import *
serverName = '127.0.0.1'
serverPort = 12000.
clientSocket = socket (AF-INET , SOCK-STREAM )
clientSocket . connect (( serverName , serverPort )).
sentence = input ("Enter the name ").
clientSocket . send (sentence . encode ( ))
filecontents = clientSocket . recv(1024) . decode()
print ('from server', filecontents )
clientSocket . close ( ).

Server.py .
from socket import *.
serverName = "127.0.0.1"
serverPort = 12000.
serverSocket = socket (AF-INET , SOCK-STREAM)
serverSocket . bind (( serverName , serverPort ))
serverSocket . listen ( 1 )
print ("The server is ready to receive").
while 1:
        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()
```

(#) <u>Output</u>

Server side - —

server is ready to 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:

AIM:- Implement UDP.

Code:

```
ClientUDP.py
from socket import *
ServerName = "127.0.0.1"
ServerPort = 20000
ClientSocket = Socket (AF_INET, SOCK_DGRAM)
Sentence = input ("Enter file 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
serverSocket = socket (AF_INET, SOCK_DGRAM)
serverSocket.bind (("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    file = open (sentence, "r")
    l = file.read (2048)
    serverSocket.sendto (bytes (l, "utf-8"), clientAddress)
    print ("Send back to client", l)
    file.close ().
```

(#) Output

Server Side - —

The Server side is ready to receive

Sent back to client : Hello World

Client side :- —

Enter file name : hello.txt

From server : Hello World.