VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB RECORD

Computer Network Lab (23CS5PCCON)

Submitted by

Vatsal Amruthling Mural (1BM22CS323)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
Academic Year 2024-25 (odd)

B.M.S. College of Engineering

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled " Computer Network (23CS5PCCON)" carried out by **Vatsal Amruthling Mural (1BM22CS323),** 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. Shashikala Associate Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE

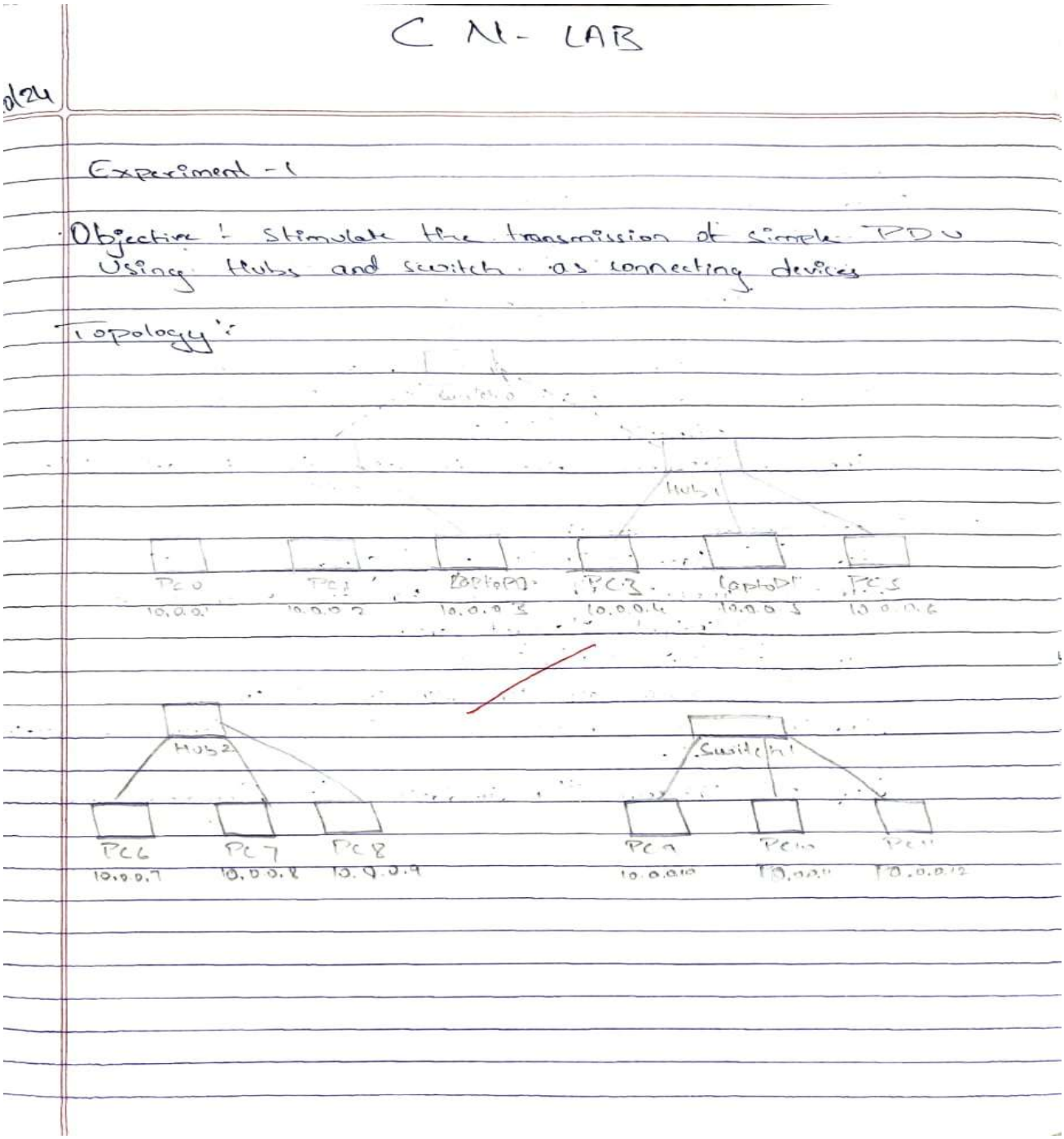
Index

GitHub Link;
https://github.com/VatsalMural18/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:



Procedure

1] 3 end devices & 1 Hub

Step 1: Place 3 generic PC's and 1 generic Hub

Step 2: Connect all the PC to the Hub using copper straight through wires

Step 3: By selecting PC → config → FastEthernet 0 set IP Address as 10.0.0.7, 10.0.0.8 & 10.0.0.9

Step 4: Select Add Sample PDU, select Source & destination PC's

Step 5: Observe the simulation in simulation Mode

2] 3 end devices & switch

Step 1: Place 3 PC's & 1 generic switch

Step 2: Connect all PC's to switch using copper Straight throught wires

Step 3: By selecting PC → config → FastEthernet 0 set IP address as 10.0.0.10, 10.0.0.11, 10.0.0.12

Step 4: Select Add Simple PDU & source & destination PC's

Step 5: Observe the simulation in simulation mode

3] 6 end devices & 2 Hubs & 1 switch

Step 1: Place 4 PC's, 2 laptops, 2 hubs & 1 switch

Step 2: Connect 2 PC's & 1 laptop to each hub using copper straight through wire & both hubs to the switch using copper cross-over wire

Step 3: Set IP address for all to end devices for 10.0.0.1 to 10.0.0.6

Step 4: Select Add simple PDU & select source & Pc

Step 5: Observe the simulation in simulation mode

Observation:

1] The hub sends the packet to all available devices the destination PC accepts the packet & sends the acknowledgement back all the remaining PC's reject the packet

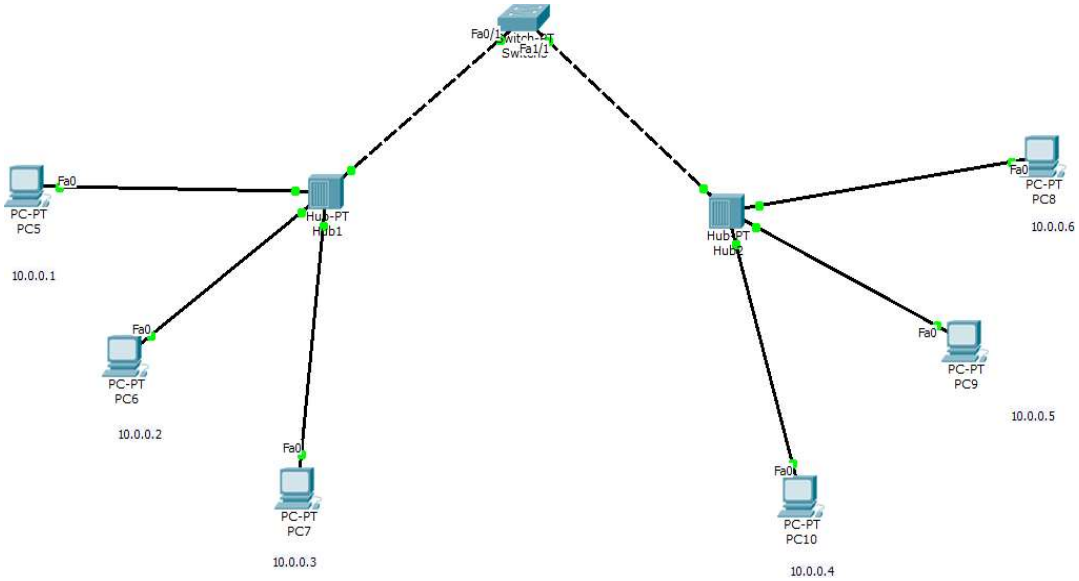
2] The switch sends the packet only to the destination PC which accepts & sends back acknowledgement back

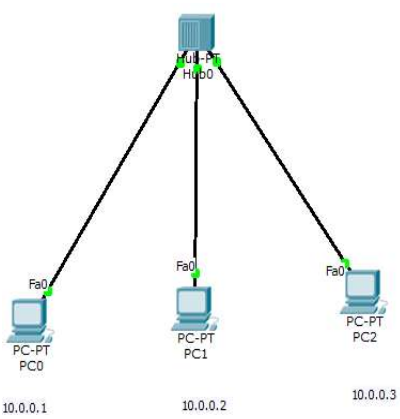
3] The hub0 receive packet & sends to other Pc & the switch, the switch sends it to hub1 which transmits it to all remaining PC's / the destination PC accepts the packet & acknowledgement by sending it back by hub1 -> switch -> hub0

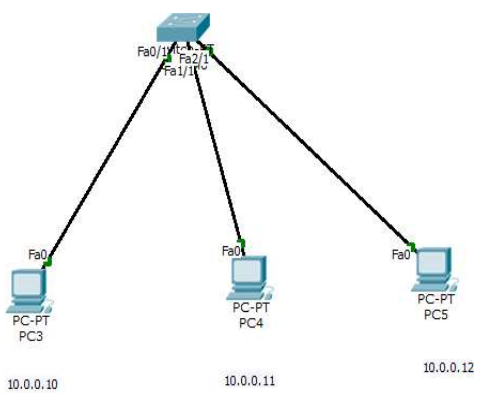
Difference between Hub & Switch

Hub	Switch
* Hub is operated in physical layer of OSI	* Switch is operated in data link layer of OSI
* Hub is broadcast type transmission	* Switch is unicast, multicast, broad cast type transmission
* Hub have 4/12 ports	* Switch can have 24 to 48 ports
* Hub is half duplex transmission model	* Switch is full duplex transmission model
* Hub is simple old not generally used	* Switch is sisophisticated & widely used.

Screen Shots:



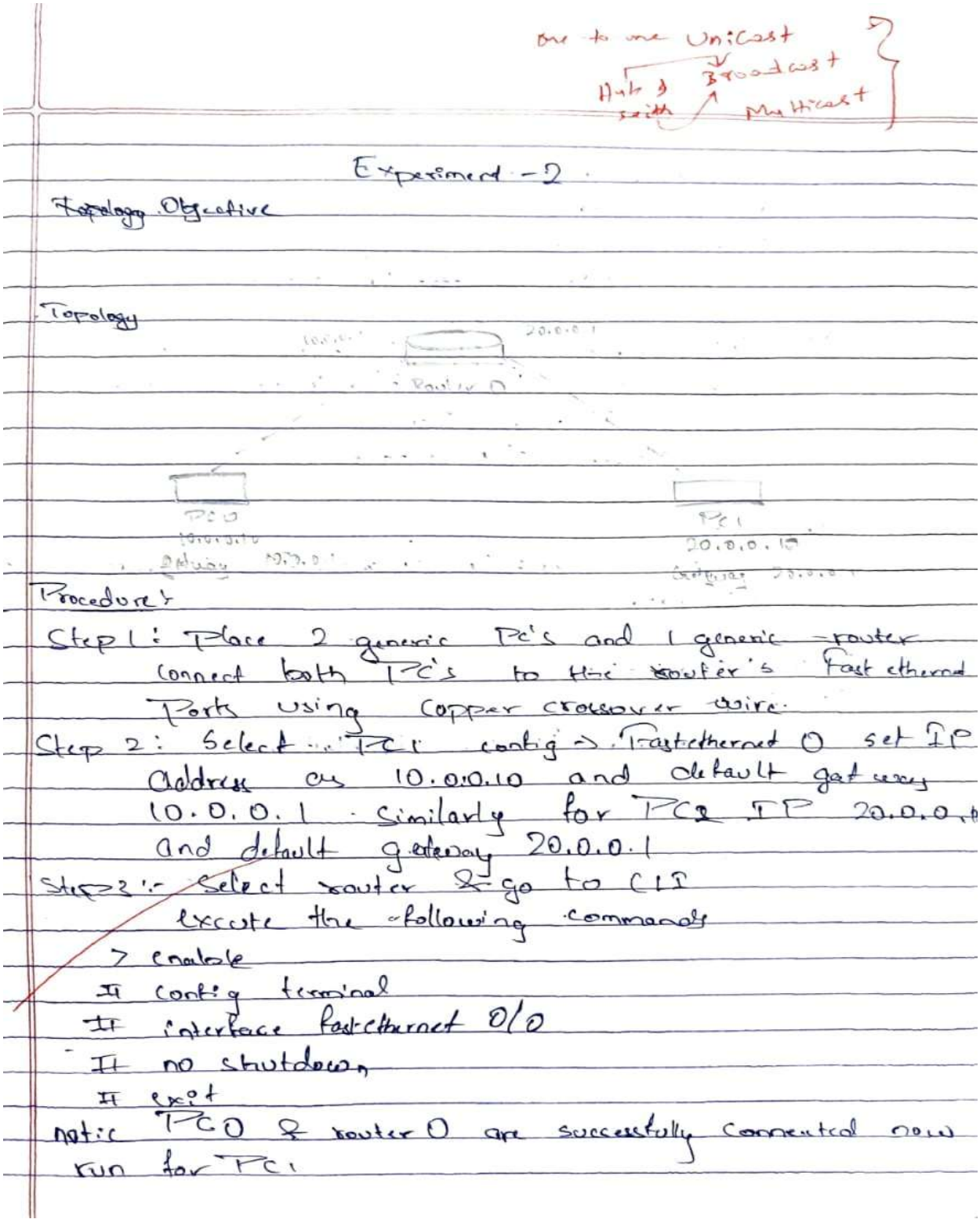




Program 2

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:



interface fastethernet 1/0
ip address 20.0.0.1 255.0.0.0
no shutdown. Successfully
PC1 is also ~~successfully~~ connected

Step 4:- Select PC0 -> Desktop -> Command prompt
~~Prin~~ Ping PC1 by running command
-> Ping 20.0.0.1
Observe the output

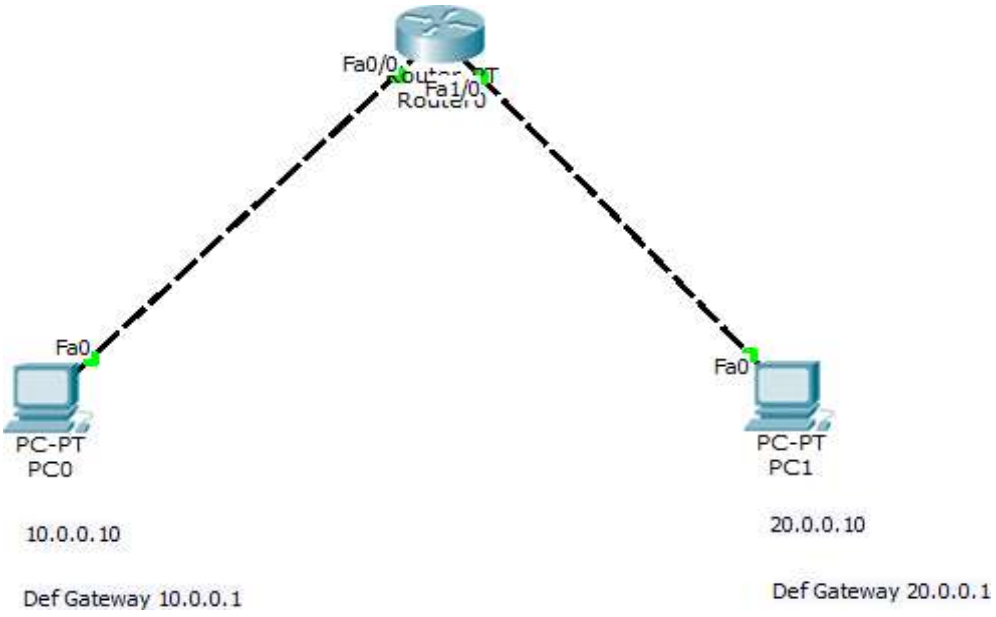
Observation:
PC0 successfully ~~Ping~~ PC1 with 32 bytes of data

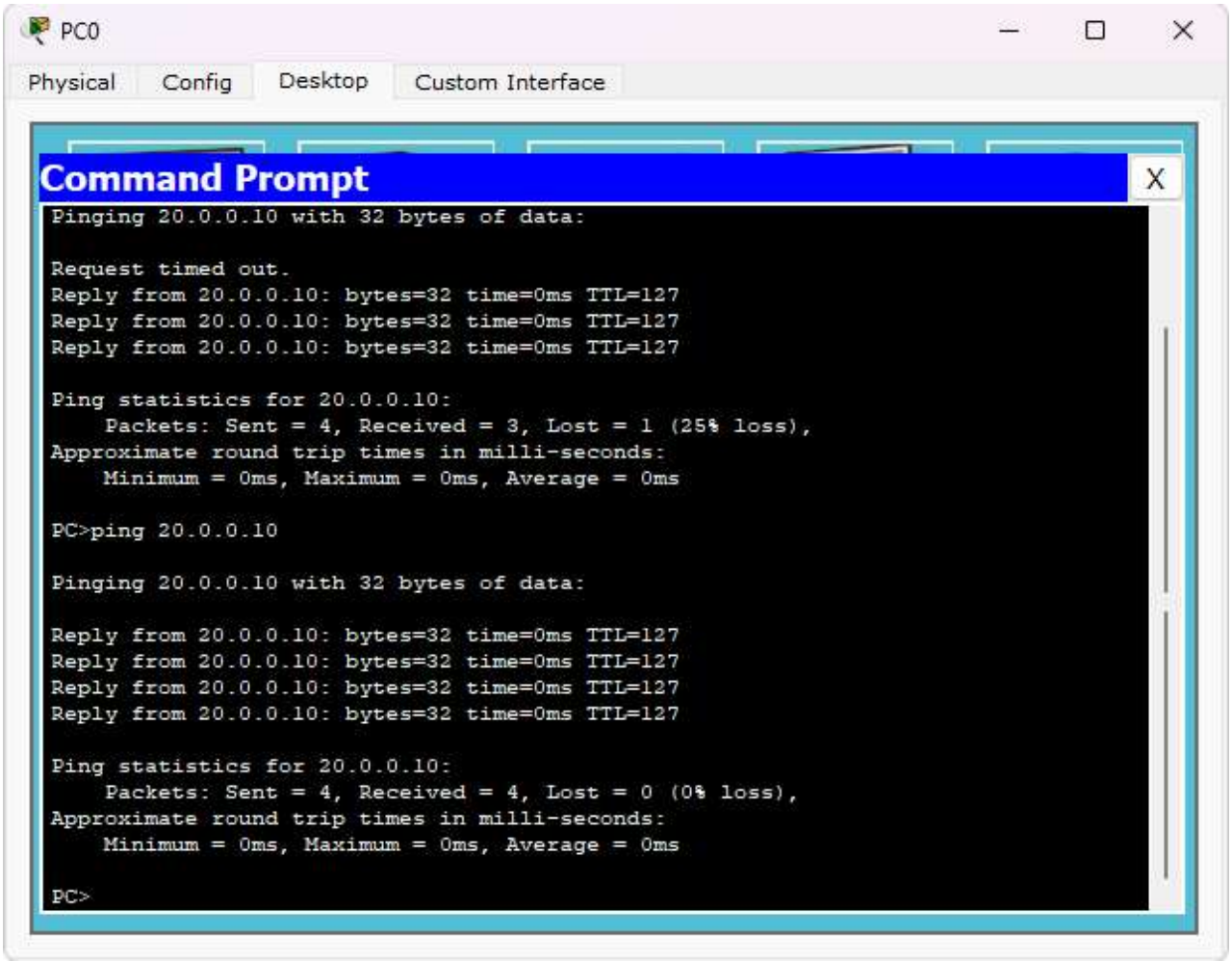
Ping Statistics
Packets
Sent:4, Received:4, Lost:0

TTL bytes

16/10

Screen Shots:





Program 3

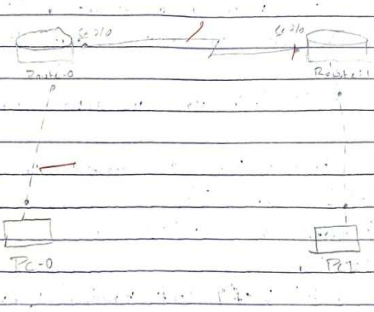
Aim: Configure default route, static route to the Router(Part 1).

Topology , Procedure and Observation:

Experiment - 3

Objective: To create 2 Network using 2 router with 2 end devices.

Topology



Procedure

Step 1: Place 2 generic PC and 2 Routers

Step 2: Connect PC-0 to Router-0 with copper Crossover cable and PC-1 to Router-1 with Copper crossover cable

Step 3: Connect Router 0 to Router 1 with Serial DCE cable

Step 4: Select PC-0 config -> Fastethernet 0 Set IP address 10.0.0.1 and set gateway as 10.0.0.2

Step 5: select PC1 config -> Fastethernet 0 set IP address 20.0.0.1 and set gateway as 20.0.0.2

Step 6: Select Router 0 & goto CFI

Execute the following commands

> enable

config terminal.

interface FastEthernet 0/0

ip address 10.0.0.1 255.0.0.0

no shut

notice PC0 & route 0 are successfully connected repeat the same Procedure for route 1 with ip address 20.0.0.1 255.0.0.0

notice PC1 & router1 are also successfully connected

Step 7: Go to router 0 -> CFI terminal & execute the following

interface serial 2/0

ipaddress 30.0.0.1 - 255.0.0.0

no shut

goto route 1 -> CFI · config terminal & execute

interface serial 2/0

ip address 30.0.0.2 255.0.0.0

no shut

notice route 0 & route 1 are successfully connected

Step 8: got to PC0 -> Desktop -> command prompt & run

Ping 20.0.0.1

Ping 30.0.0.2

Ping 30.0.0.1

& observe the Output

Observation

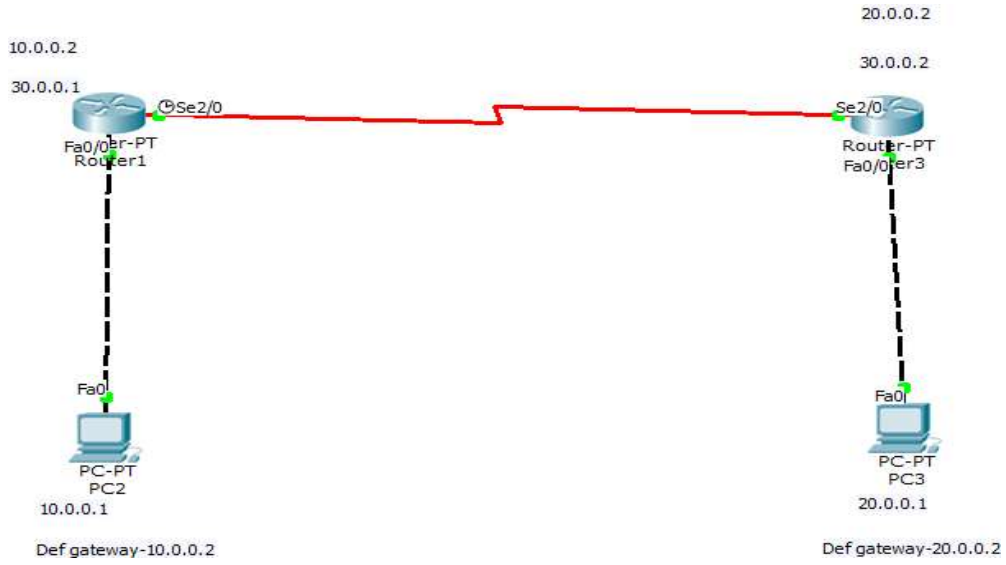
1] All PC's and router are connected successfully

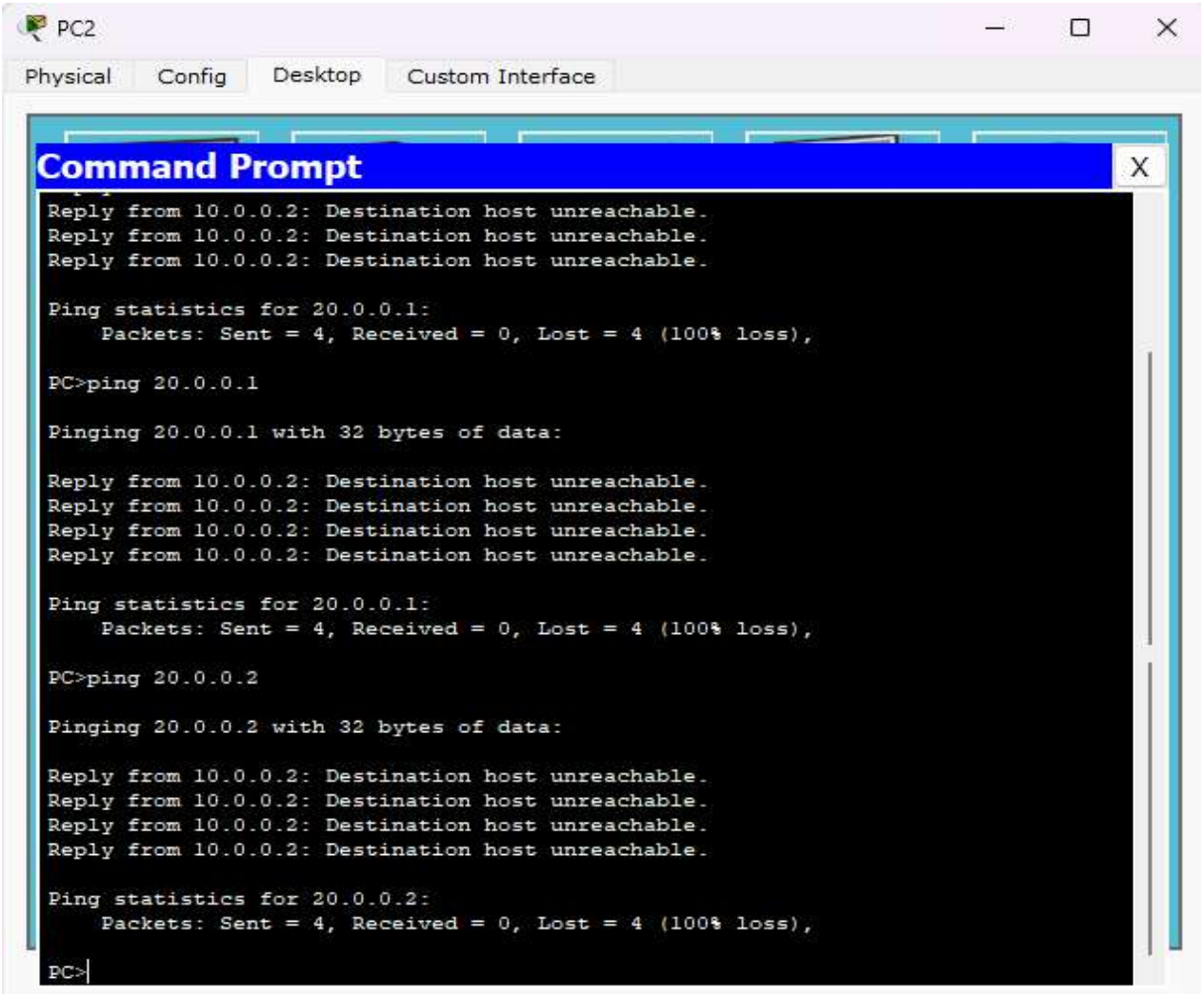
2] Ping 20.0.0.1 & Ping 30.0.0.2 are unsuccessfully & show destination host Unreachable. This happens because they are not neighbor networks with 10.0.0.1

3] Ping 30.0.0.1 will be successfully.

Router> ip route 30.0.0.0 255.0.0.0 20.0.0.2

Screen Shots:





Program 4

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

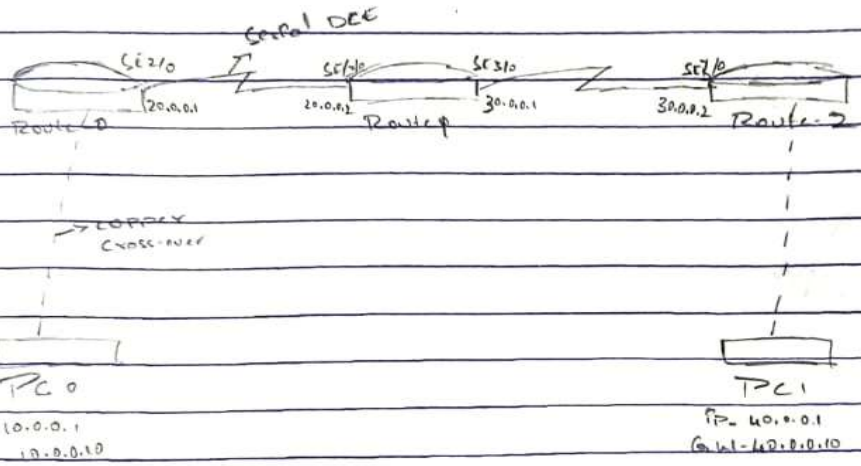
Topology , Procedure and Observation:

29/10/24

Experiment: 4

Objective : config the static & default routing using three routers & 2 PC's

Topology



Procedure

Step 1 : Place 3 Routers & 2 PC. Connect the PC to router with copper cross-over cable & Router to Router with Serial DCE

Step 2 : Select PC0 config -> Fastethernet 0/0 set IP address as 10.0.0.1 and config -> setting set gateway as 10.0.0.10

Step 3 : Select PC1 config -> Fastethernet 0/0 set IP address as 40.0.0.1 and config -> setting set gateway as 40.0.0.10

Step 4: Select Router 0 & goto to CLI
execute the following commands

> enable
config terminal
ip address interface FastEthernet 0/0
ip address 10.0.0.1 255.0.0.0
no shut
exit
interface serial 2/0
ip address 20.0.0.1 255.0.0.0
no shut

Router 2 -> CLI
> enable
config terminal
interface fastEthernet 0/0
ip address 40.0.0.1 255.0.0.0
no shut
exit
interface serial 3/0
ip address 30.0.0.2 255.0.0.0
no shut

Connection between PC1, R0 and PC2 R2 will turn green

Step 5 Router1 -> CLI

> enable
config terminal
interface serial 2/0
ip address 20.0.0.2 255.0.0.0
no shut
exit
interface serial 3/0
ip address 30.0.0.1 255.0.0.0
no shut
exit

Connect b/w R0, R1 & R2, R2 will turn green

Step 6:- Router -> CLI

config terminal
ip route 10.0.0.0 255.0.0.0 20.0.0.1
ip route 40.0.0.0 255.0.0.0 30.0.0.2
exit

Route 0 -> CLI

config terminal
ip route 0.0.0.0 0.0.0.0 20.0.0.2

Route 2 -> CLI

>enable
config terminal
ip route 0.0.0.0 0.0.0.0 30.0.0.1

Step 6:- Show IP route in the router, Ping 40.0.0.10 from 10.0.0.1 & vice versa

Observation

i). All the connections are successfull

ii) PC1 -> Desktop -> command prompt

Ping 40.0.0.10 is successfull

Ping Statistics

Sent=4, Recived=4, lost 0 (0% Loss)

iii] ip route in R1

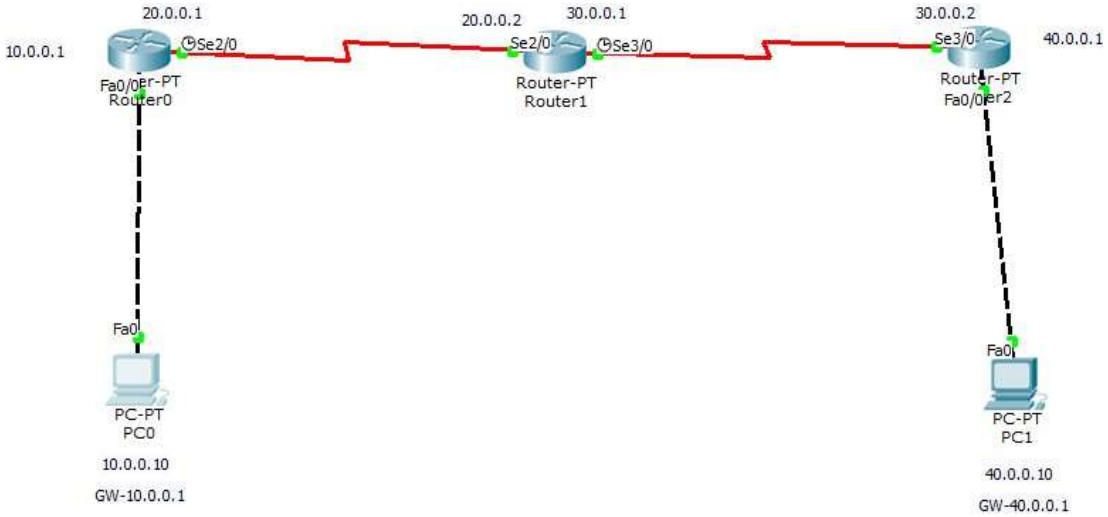
S 10.0.0.0 via 20.0.0.1
C 20.0.0.0 directly
C 30.0.0.0 directly
S 40.0.0.0 via 30.0.0.2

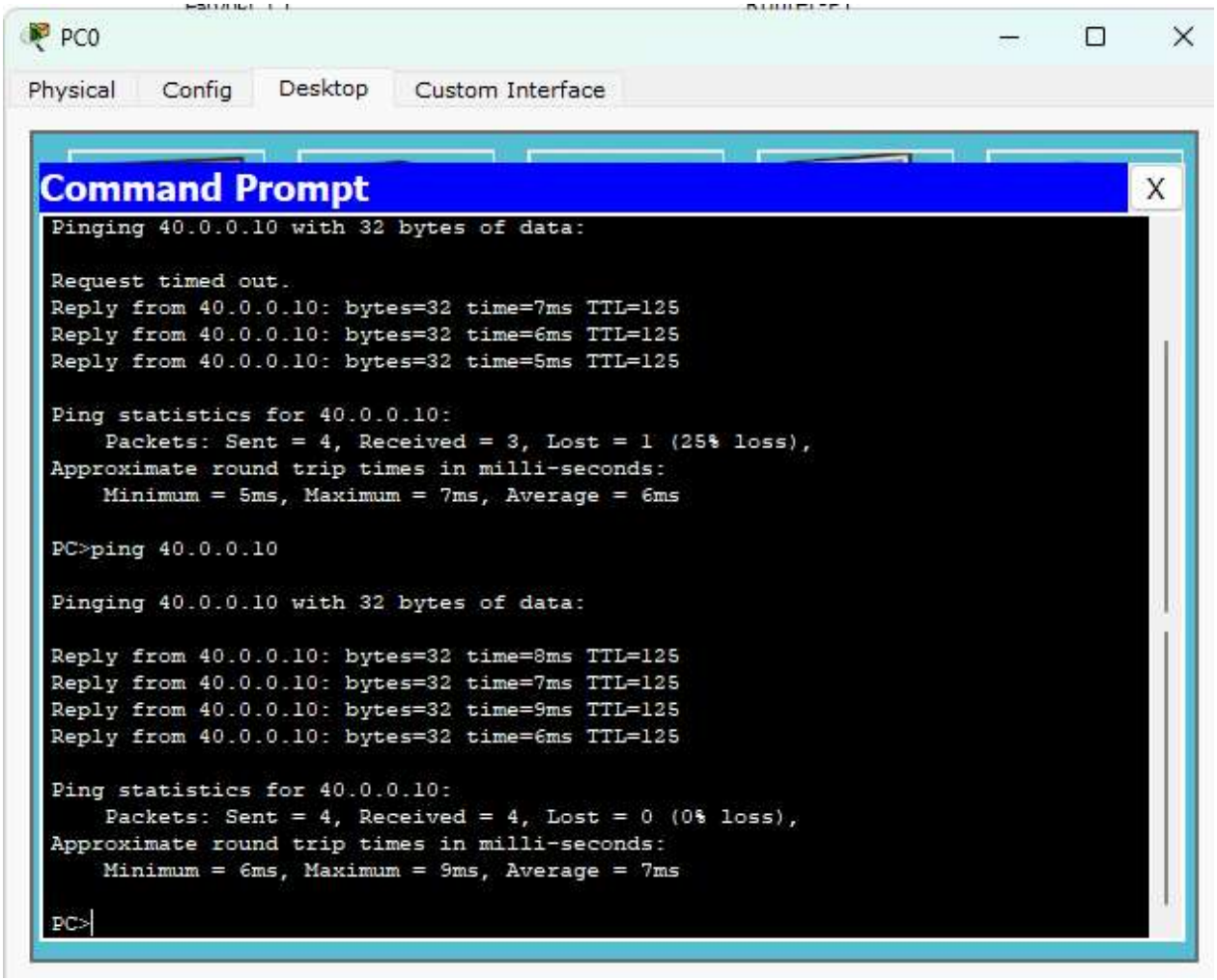
iv] Ping 40.0.0.1

Replay from 40.0.0.10 byte 32 time 0ms TTL=125
Replay from 40.0.0.1 bytes 32 time=7ms TTL=125
Replay from 40.0.0.1 bytes 32 time 8ms TTL=125
Replay from 40.0.0.1 bytes 32 time 7ms TTL=125

13/11

Screen Shots:





Program 5

Aim: Configure DHCP within a LAN and outside LAN.

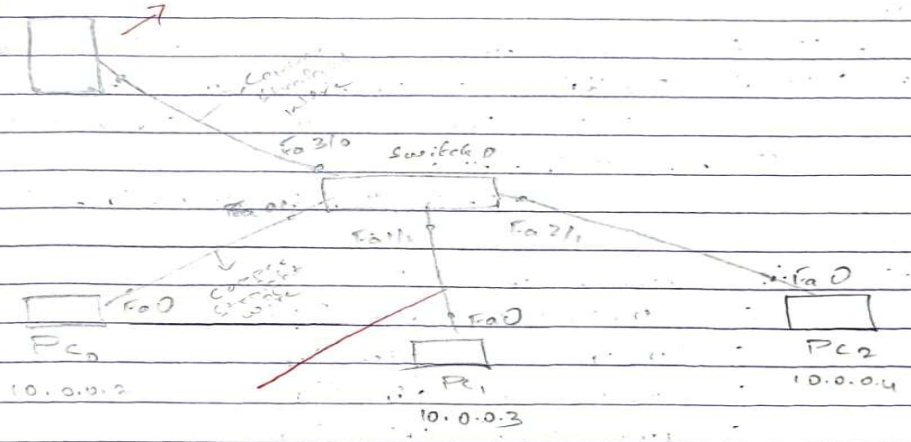
Topology , Procedure and Observation:

13/11/24

Experiment - 5

3 PC's, 1 Switch & 1 Server using DHCP

Topology



Procedure

Step 1 : Place 3 PC's, 1 Switch & 1 Server PT.
Connect PC to switch these using Copper straight wire & Switch to Server using Copper straight wire

Step 2 : Select Server -> Desktop -> IP configuration
IP address - 10.0.0.1
Default Gateway - 10.0.0.0

Step 3 : In server PT go to Config -> SERVICES -> DHCP
turn Services to on
Pool name : switch 1
Default gateway : 10.0.0.0

Start IP : 10.0.0.3
Max number of users : 100
Click on ADD

Step 4 :- Go to each PC Desktop -> IP configuration &
Change IP configuration from static to DHCP
The IP address will be assigned automatically

Step 5 :- Ping from PC_0 to PC_3

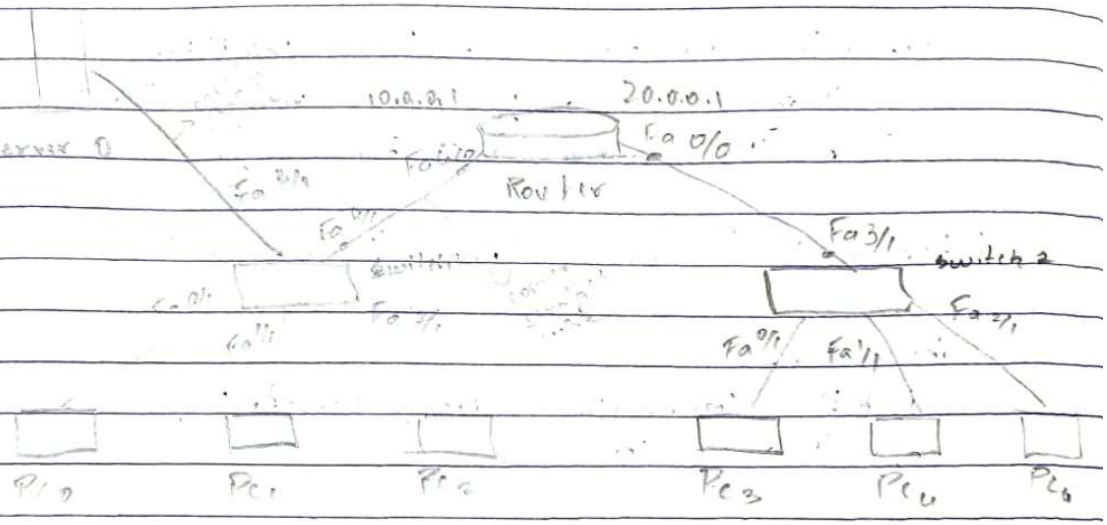
Observation
1] All connections are successfull
2] Ping 10.0.0.4

Pinging 10.0.0.4 with 30 bytes of data
reply from 10.0.0.4 bytes = 32 time = 0ms TTL = 128
reply from 10.0.0.4 bytes = 32 time = 0ms TTL = 128
reply from 10.0.0.4 bytes = 32 time = 0ms TTL = 128
reply from 10.0.0.4 bytes = 32 time = 0ms TTL = 128

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

Objective: Design a DHCP network outside LAN using Router

Topology



Procedure:

Step 1:- Place 6 Pc's, 2 switch, 1 server, 1 router & connect them as shown in the figure

Step 2:- Server → Desktop → IP configuration
IP address - 10.0.0.2
Default gateway - 10.0.0.1

Step 3:- Config → services → DHCP, turn service to ON
Pool Name : switch 1
Default gateway : 10.0.0.1
Start IP : 10.0.0.3
Max user : 100
Click Add

Pool name : Switch 2
Default Gateway :- 20.0.0.1
Start IP : 20.0.0.3
Max User : 100
Click on Add

Step-4 :- Go to router CLI
> enable
config terminal
interface fa 4/0
ip address 10.0.0.1 255.0.0.0
no shut
exit

ip interface fa 0/0
ip address 20.0.0.1 255.0.0.0
ip helper-address 10.0.0.2
no shut

All router switch connections go UP

Step 5 :- Go to all 6 Pc's & change IP config from static to DHCP address will be automatically assigned

Observation

1) All connections are successfull

* Pinging from 10.0.0.4 to 20.0.0.6

Pinging 20.0.0.6 with 32 bytes of data
Requested timed Out
Requested timed Out
Requested timed Out
Requested timed Out

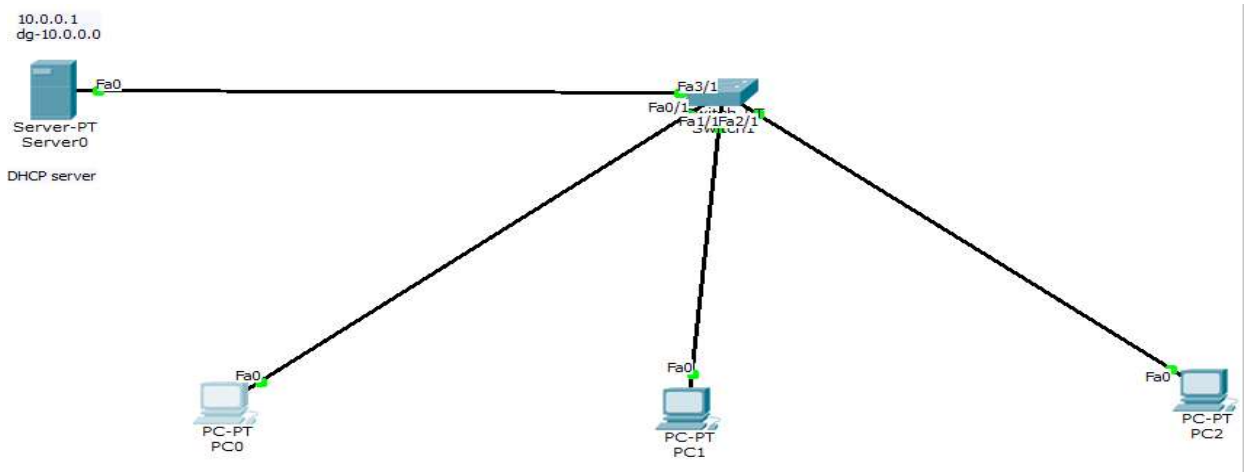
Sent : 4 Received : 0 Lost : 4 (100% loss)

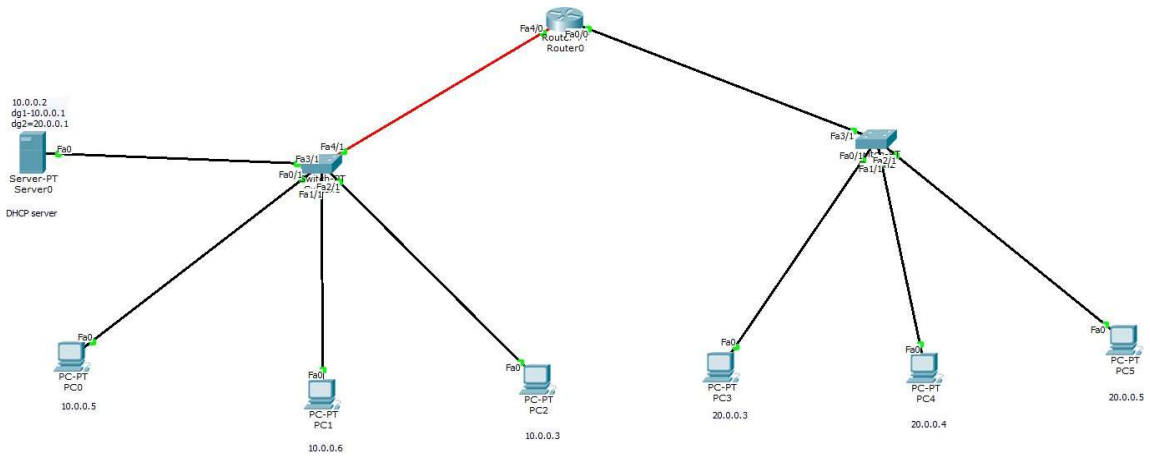
* Ping 10.0.0.2 from 10.0.0.4

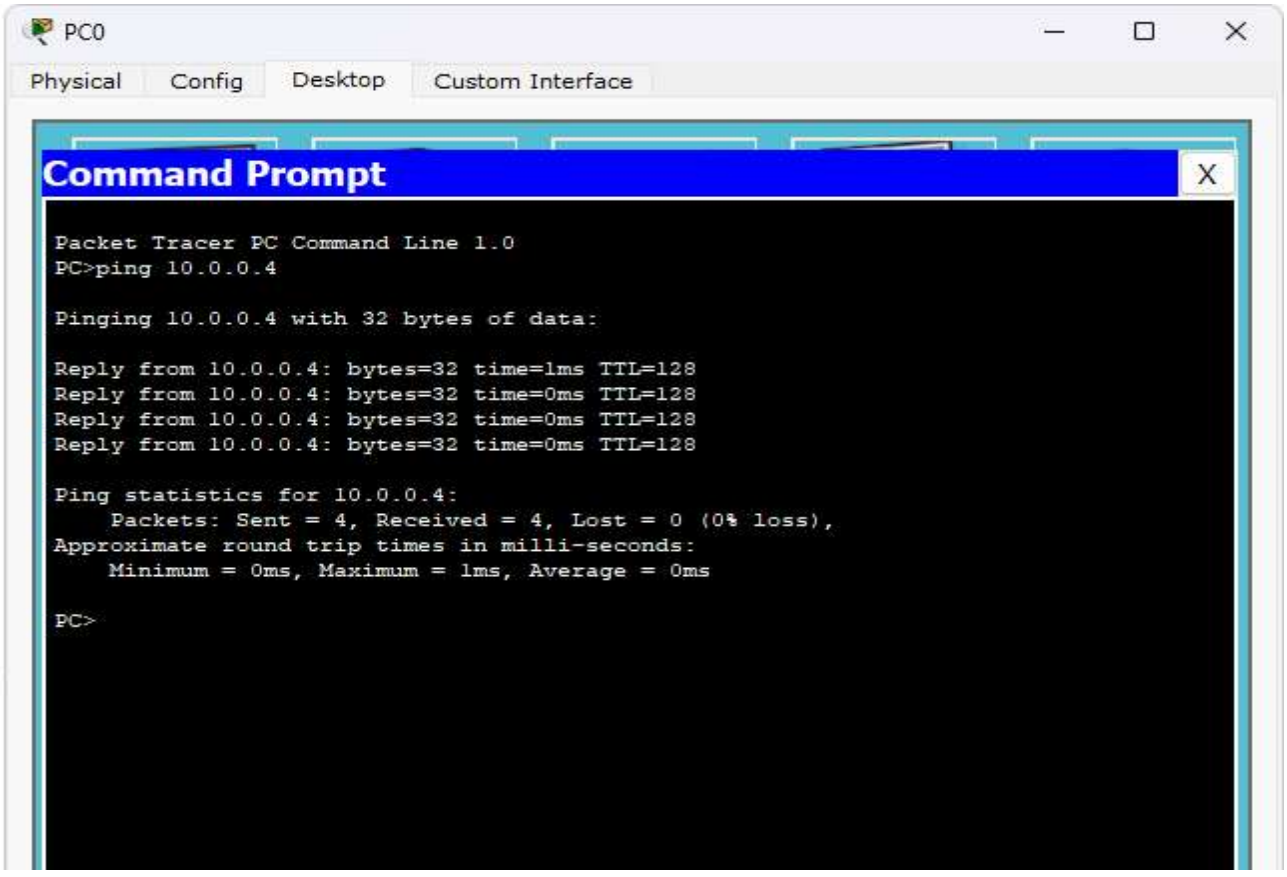
Pinging 10.0.0.2 with 32 byte of data
Replay from 10.0.0.2 : bytes = 32 time = 0ms TTL = 128
Replay from 10.0.0.2 : bytes = 32 time = 0ms TTL = 128
Replay from 10.0.0.2 : bytes = 32 time = 0ms TTL = 128
Replay from 10.0.0.2 : bytes = 32 time = 0ms TTL = 128

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

Screen Shots:




Router0
Fa4/0
Fa0/0
10.0.0.2
dg1-10.0.0.1
dg2=20.0.0.1
Server-PT
Server0
DHCP server
Fa0
Fa4/1
Fa3/1
Fa2/1
Fa1/1
Fa3/1
Fa2/2
Fa1/1
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>

Program 6

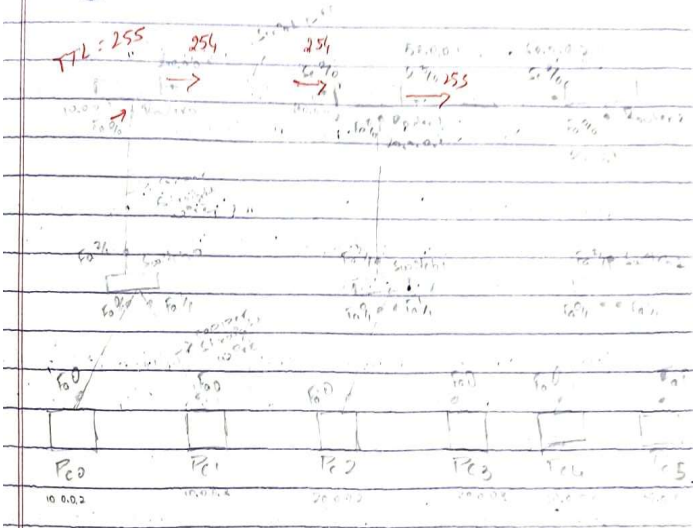
Aim: Configure RIP routing Protocol in Routers .

Topology , Procedure and Observation:

Experiment 6

Objective : Configure routing information protocol in Routers

Topology :



Procedure :

Step 1 :- Place 3 Routers, 3 switches & 6 PC's and connect them. Router to switches using copper straight wire and switches to PC's using copper straight wire & Router to Router using Serial DCE wire

Step 2 :- Go to Router0 config -> Fastethernet 0/0 set IP address as 10.0.0.1 & config -> Serial 2/0 set IP address as 40.0.0.1

Step 3 :- Go to PC0 & PC1 config -> setting & set gateway as 10.0.0.1 & go to fastethernet 0 set IP address as 10.0.0.2 & 10.0.0.3 respectively

Step 4 :- Go to Router1 config -> Fastethernet 0/0 set IP address 20.0.0.1 & Serial 2/0 set IP address as 40.0.0.2 Serial 3/0 IP address as 50.0.0.1

Step 5 :- Go to PC2 & PC3 config -> setting set gateway as 20.0.0.1 & in fastethernet 0 set IP address as 20.0.0.2 & 20.0.0.3 respectively

Step 6 :- Go to Router2 config -> Fastethernet 0/0 set IP address as 30.0.0.1 & Serial 2/0 set IP address as 50.0.0.2

Step 7 :- Go to PC4 & PC5 config -> setting set gateway as 30.0.0.1 & in fastethernet 0 set IP address as 30.0.0.2 & 30.0.0.3 respectively

Step 8 :- In Both Router & PC's [End Devices] set the Port Status to 'ON' in fastethernet 0/0, Serial 2/0 & Serial 3/0

Step 9:- In Router0 CLI ->

Router>enable
Config terminal
router rip
network 10.0.0.0
network 40.0.0.0
exit
exit
show IP route

Step 10:- In Router1 CLI

>enable
config terminal
router rip
network 40.0.0.0
network 50.0.0.0
network 20.0.0.0
exit
exit
show IP route

Step 11:- In Router2 CLI ->

>enable
config terminal
router rip
network 50.0.0.0
network 30.0.0.0
exit
exit
show IP route.

Observation

1) All connections are successfully

Ping from 10.0.0.2 to 30.0.0.2

Pinging 30.0.0.2 with 32 bytes of data
Requested timed out
Reply from 30.0.0.2 bytes=32 time=6ms TTL=125
Reply from 30.0.0.2 bytes=32 time=6ms TTL=125
Reply from 30.0.0.2 bytes=32 time=6ms TTL=125

Ping statistics for 30.0.0.2:
Packet: sent=4, Received=3, lost=1 (25% loss)

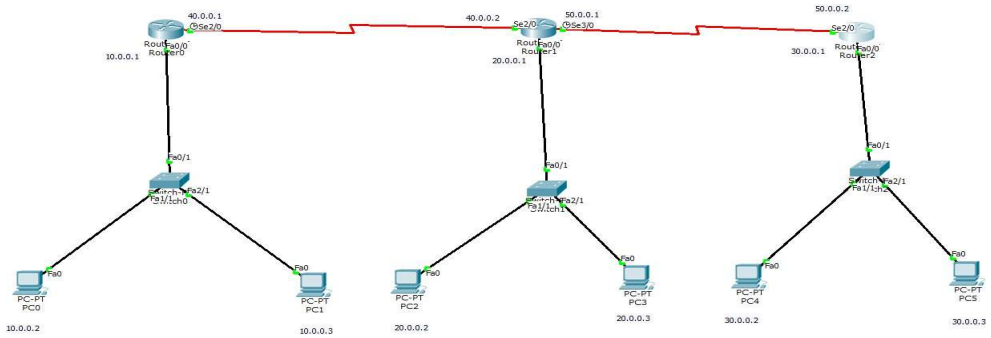
>Ping 30.0.0.2

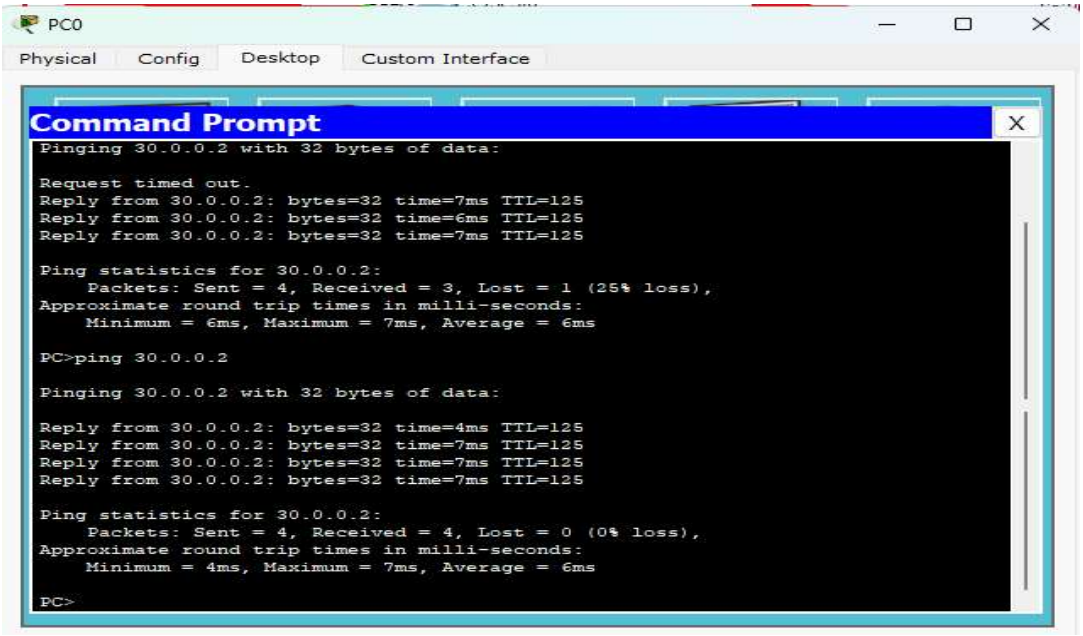
Pinging 30.0.0.2 with bytes of data
Reply from 30.0.0.2 bytes=32 time=8ms TTL=125
Reply from 30.0.0.2 bytes=32 time=7ms TTL=125
Reply from 30.0.0.2 bytes=32 time=7ms TTL=125
Reply from 30.0.0.2 bytes=32 time=7ms TTL=125

Ping statics for 30.0.0.2:

Packet: Sent=4, Received=4, lost=0 (0% loss

Screen Shots:





Program 7

Aim: Demonstrate the TTL/ Life of a Packet .

Topology , Procedure and Observation:

Objective :- Demonstrate TTL / Life of a Packet

Procedure

Step 1 :- Construct the previous topology

Step 2 :- Create a simple PDU source : PC0 and destination PC5 & go to simulation mode

Step 3 :- Click Auto capture / Play & select the Packet when it reaches router 0, check the value of TTL in bound PDU, Out bound PDU

Step 4 :- Continue to press Auto-capture / Play & check the same when Packet reaches all routes

Observation

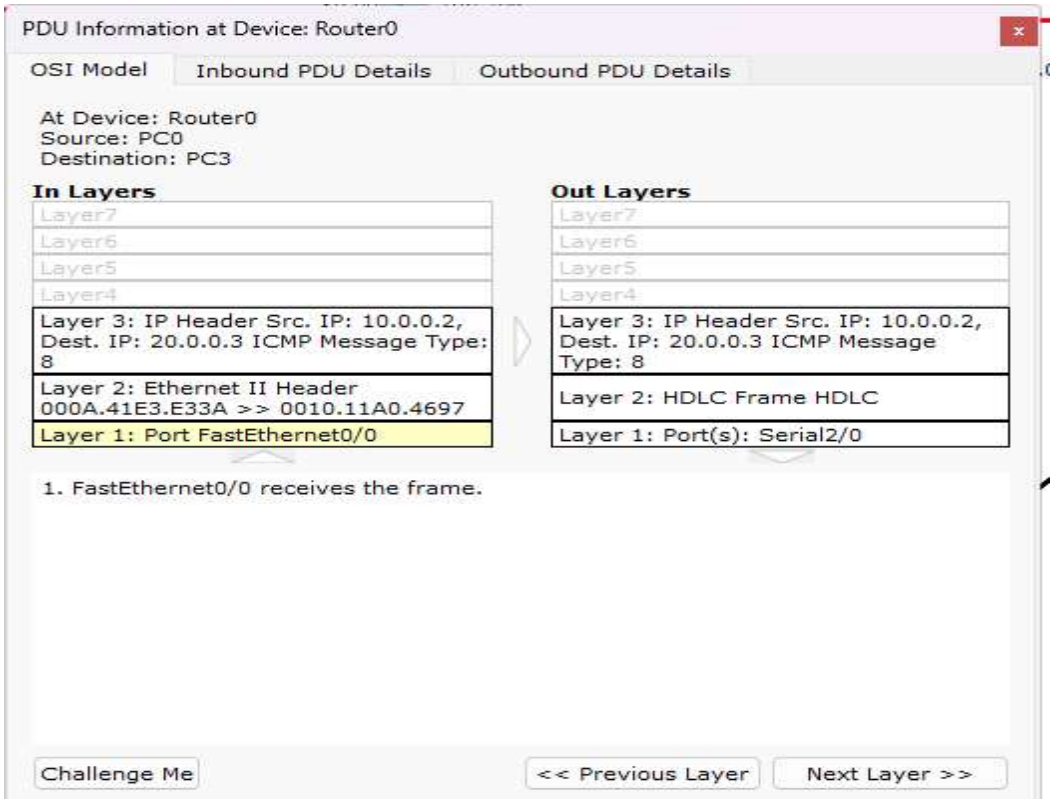
1. Router Contain information on packet fill level 3
2. TTL value decreases as it passes each router
3. Router 0

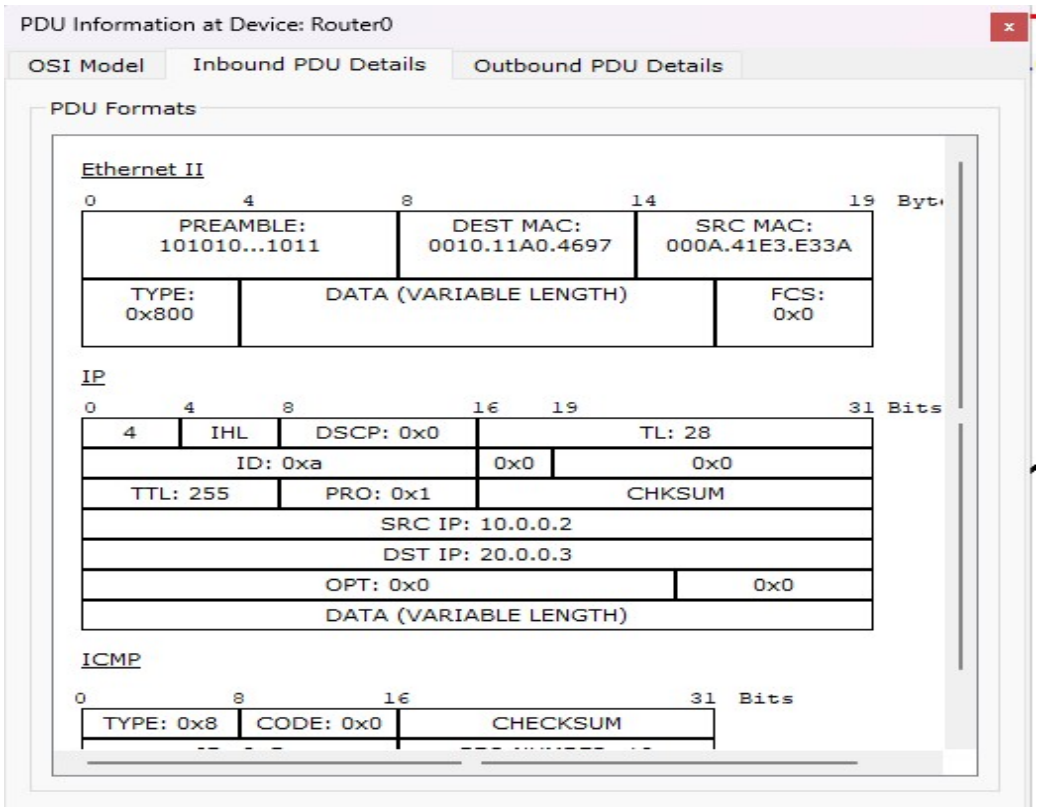
Inbound PDU — TTL : 255

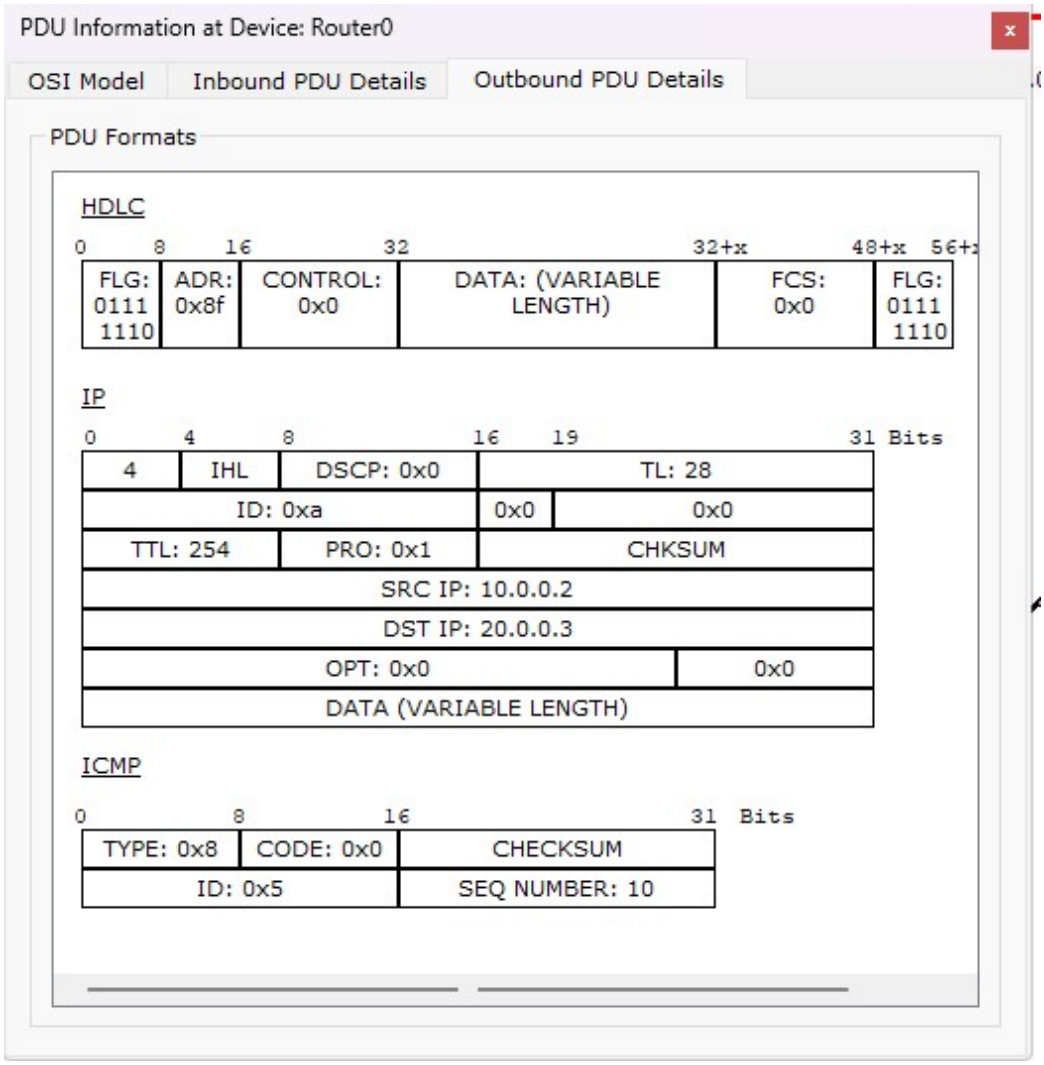
Outbound PDU — TTL :- 254

27/11

Screen Shots:



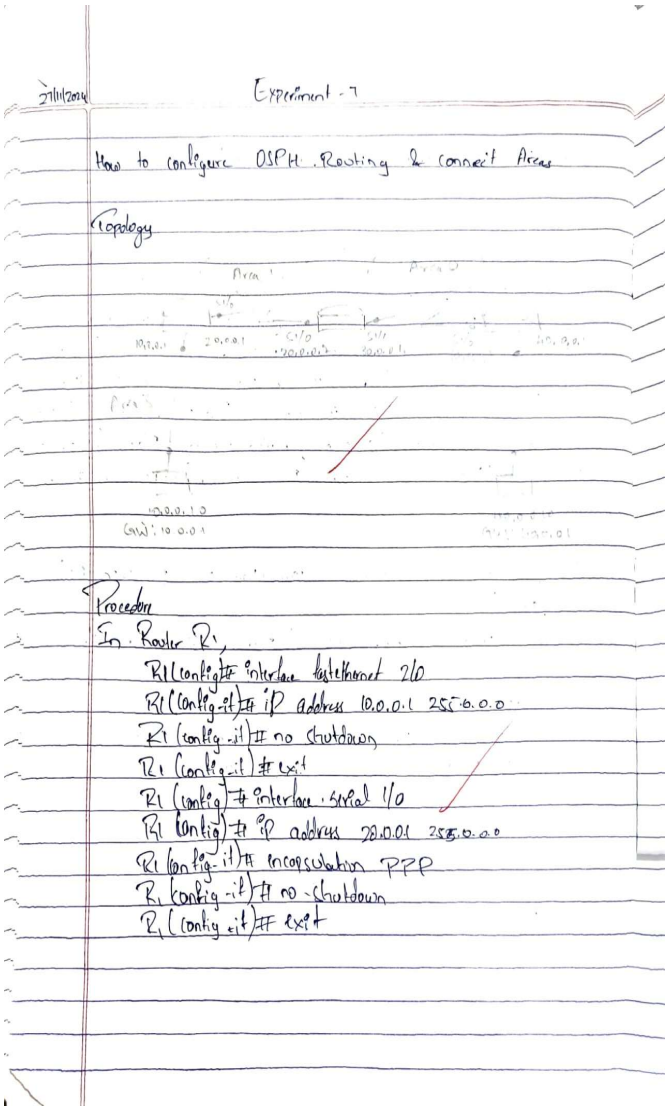


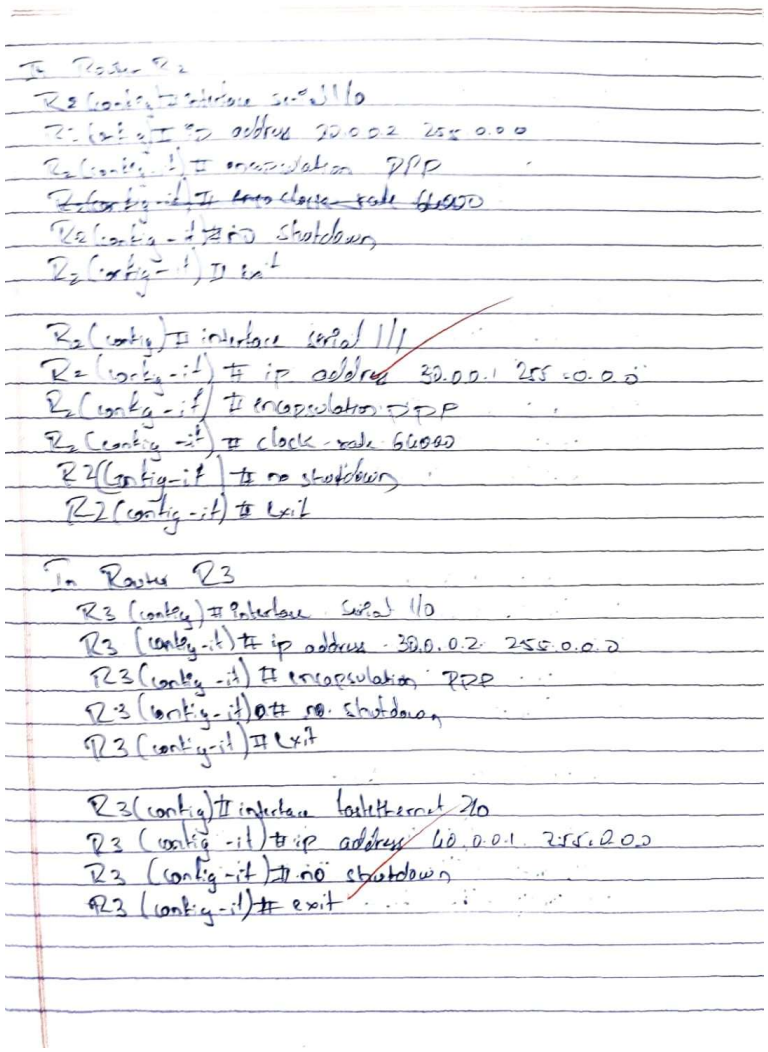

PDU Information at Device: Router0
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
0 8 16 32 32+x 48+x 56+x
FLG: 0111 1110
ADR: 0x8f
CONTROL: 0x0
DATA: (VARIABLE LENGTH)
FCS: 0x0
FLG: 0111 1110
IP
0 4 8 16 19 31 Bits
4
IHL
DSCP: 0x0
TL: 28
ID: 0xa
0x0
0x0
TTL: 254
PRO: 0x1
CHKSUM
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:





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 1
R1(config-router)# network 20.0.0.0 0.255.255.255 area 1
R1(config-router)# exit

In Router R2

R2(config)# router ospf 1
R2(config-router)# router-id 2.2.2.2
R2(config-router)# network 20.0.0.0 0.255.255.255 area 1
R2(config-router)# network 30.0.0.0 0.255.255.255 area 0
R2(config-router)# exit

In Router R3

R3(config)# router ospf 1
R3(config-router)# router-id 3.3.3.3
R3(config-router)# network 30.0.0.0 0.255.255.255 area 0
R3(config-router)# network 40.0.0.0 0.255.255.255 area 2
R3(config-router)# exit

Router# show IP route

R1(config-if)# interface loopback 0
R1(config-if)# ip add 172.16.1.252 255.255.0.0
R1(config-if)# no shutdown

R2(config-if)# interface loopback 0
R2(config-if)# ip add 172.16.1.253 255.255.0.0
R2(config-if)# no shutdown

R3(config-if)# interface loopback 0
R3(config-if)# ip add 172.16.1.254 255.255.0.0
R3(config-if)# no shutdown

R3# show IP route

In Router R1

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

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

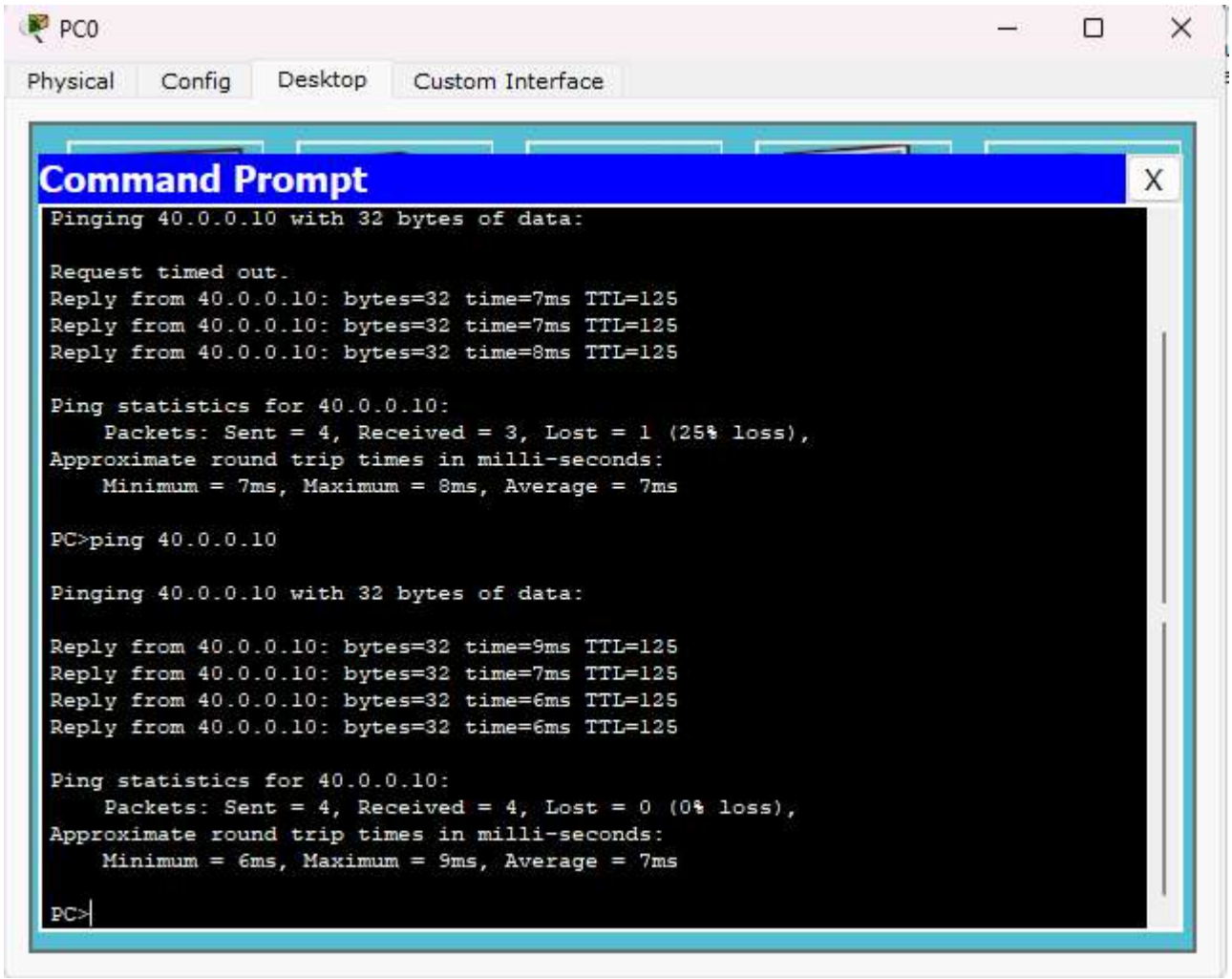
R3# show IP route

Observation

Ping 40.0.0.10 : 56 bytes
64 bytes from 40.0.0.10: seq 1 ttl = 61 time = 173.753 ms
64 bytes from 40.0.0.10: seq 2 ttl = 61 time = 64.794 ms
64 bytes from 40.0.0.10: seq 3 ttl = 61 time = 66.708 ms
64 bytes from 40.0.0.10: seq 4 ttl = 61 time = 60.824 ms
64 bytes from 40.0.0.10: seq = ttl = 61 time = 96.807 ms

--- 40.0.0.10 Ping statistics ---
6 Packets trasmitted, 5 Packets recived, 16% Packet loss
round-trip min/avg/max = 60.824/92.438/173.753 ms

Screen Shots:



Program 9

Aim: Configure Web Server, DNS within a LAN.

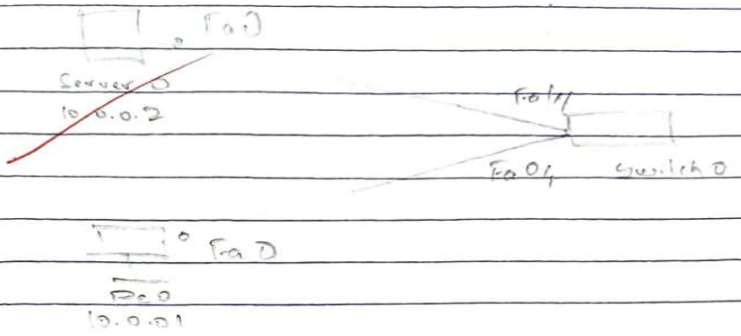
Topology , Procedure and Observation:

Experiment - 8

Objective :-

Configure Web Server, DNS, within a LAN

Topology



Procedure :-

-> Place an end device a server & a switch & connect them using copper straight wires

-> Assign the IP address as demonstrated in topology

-> To set the IP of the server

-> goto config

-> select DNS method, turn it on & add a resource

-> set IP address ; make sure port is ON

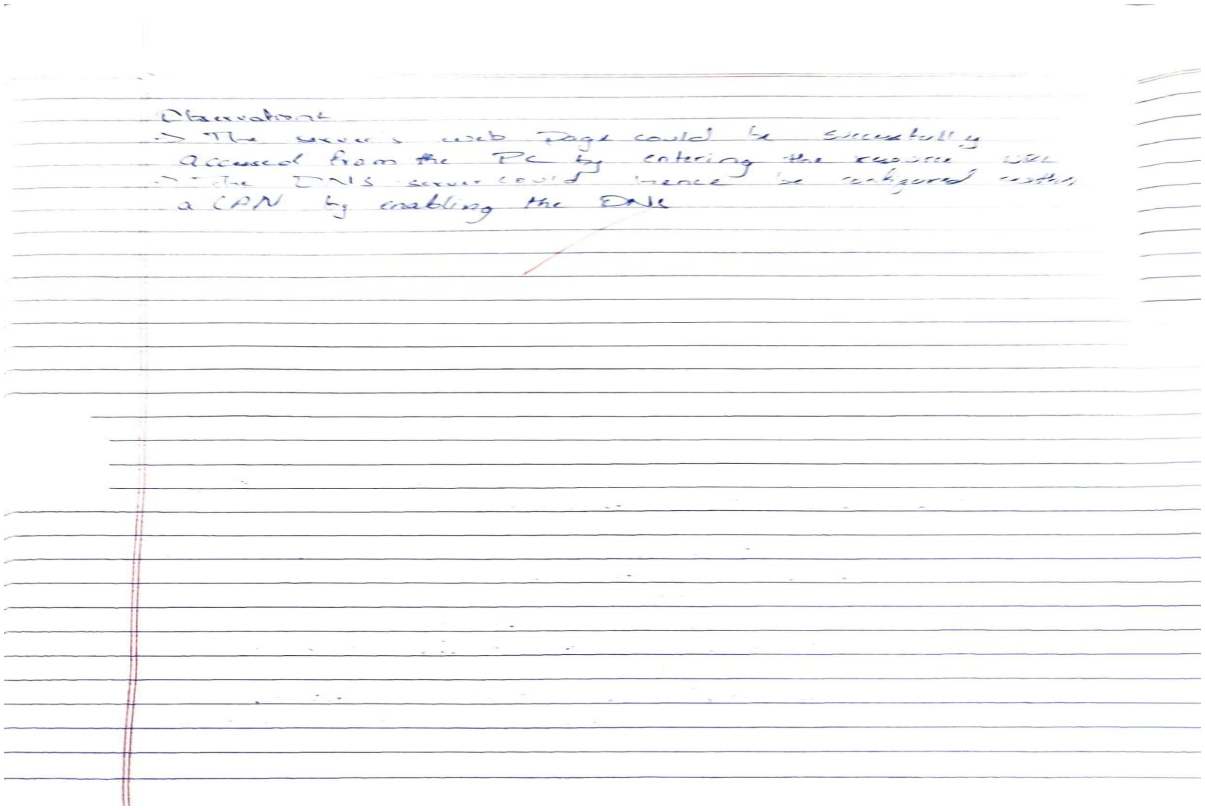
-> Select HTTP.

-> Turn the services to on

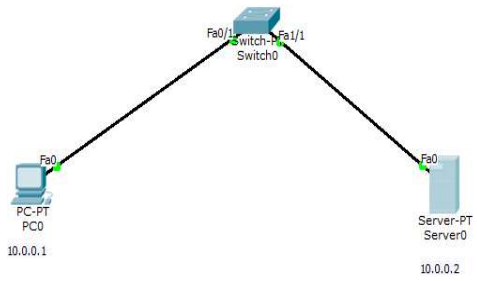
-> comment the content of the code as needed & click on it

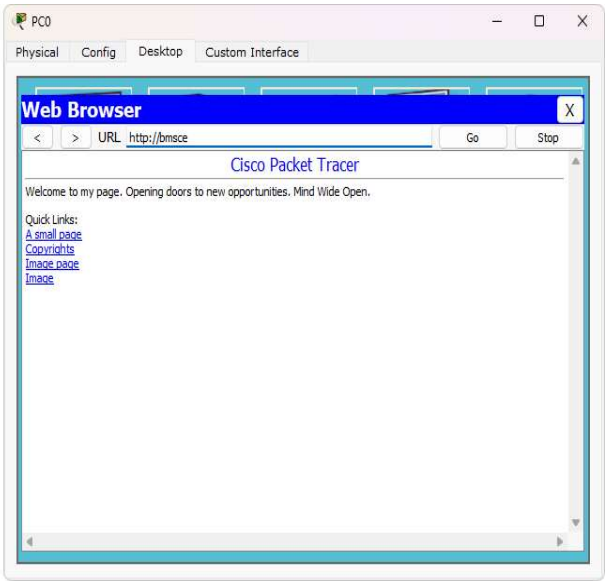
-> select the PC0 -> Desktop -> web browser

-> Enter the URL specified in the DNS resource



Screen Shots:





Program 10

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

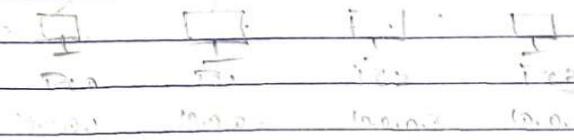
Topology , Procedure and Observation:

Experiment - 10

Objective

To construct simple LAN & understand the concept & Operation of Address Resolution Protocol (ARP)

Topology



Procedure :

-> Place 3 end devices, a server and a switch and connect the PC and the server to the switch using copper straight wires.

-> Use the inspect tool to click on a PC to view the ARP table

-> The same can also be viewed in the command Prompt by using 'arp -a'

-> Go to CLI of the switch and do 'show mac address'

-> Similarly Obtain ARP table of the server & other end devices

-> Enter the simulating mode & click on capture/ by selecting PC1 & PC2 for simple PDU's

Observation

-> Initially, the ARP tables of all end devices are observed to be empty

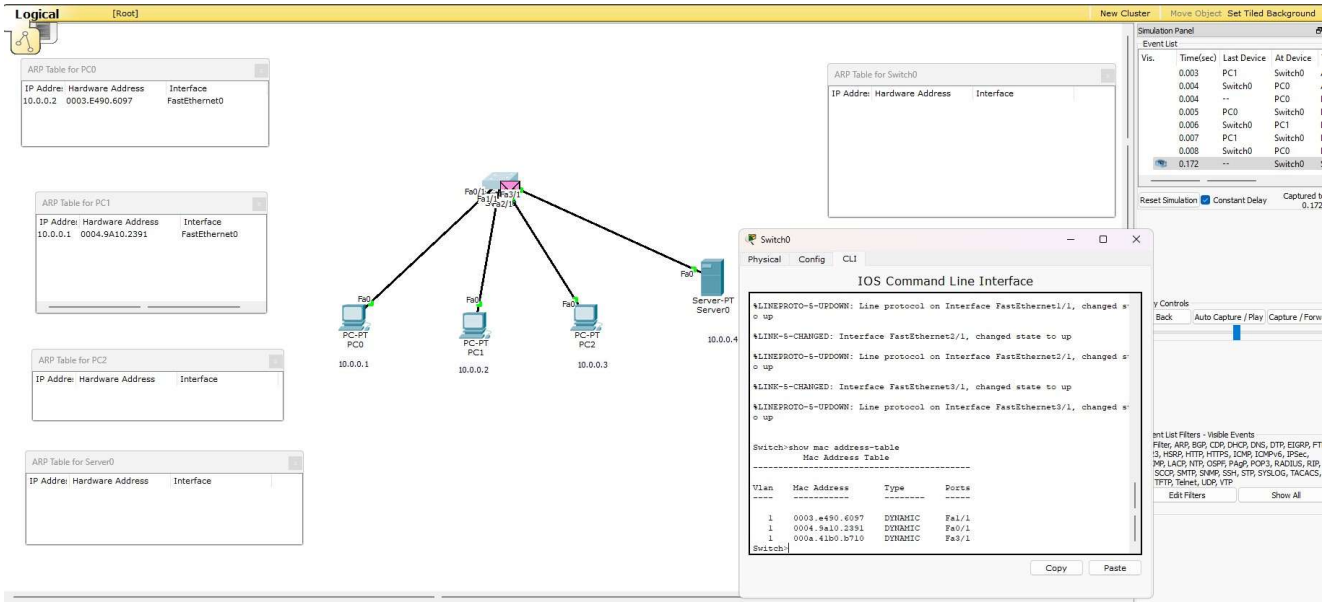
-> The MAC address table is also found to be empty

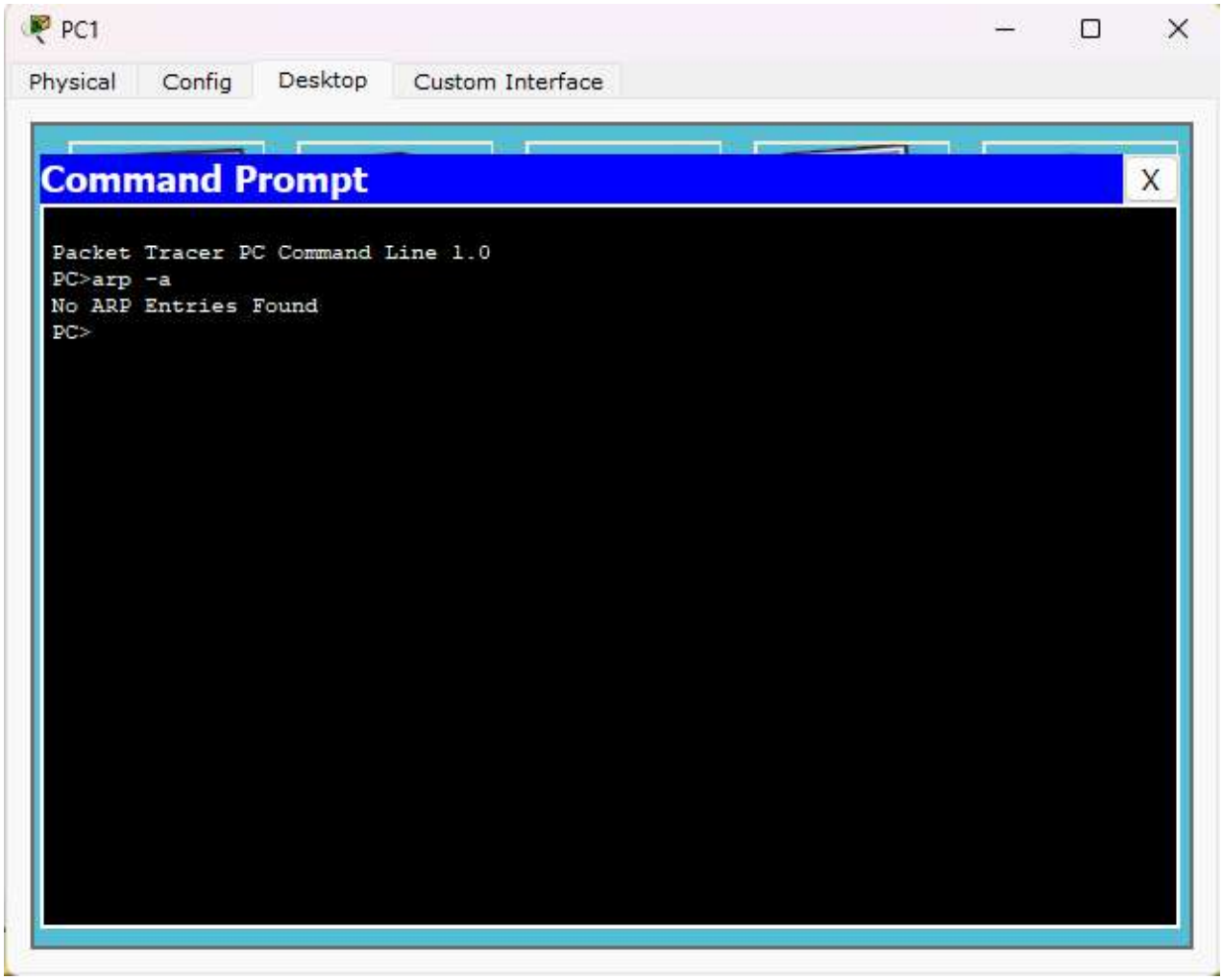
-> When the capture button is clicked it is found that ARP table is updated in PC2 with the IP address of PC1 (10.0.0.2)

-> Once the acknowledgement is obtained the ARP table of PC1 is updated with the IP address of PC2 (10.0.0.3)

-> The event list in the simulation Panel shows the corresponding Protocol used during the communication

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.

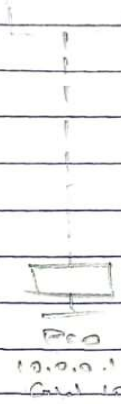
Topology , Procedure and Observation:

Experiment - 9

Objective:-

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

Topology



Experiment

-> Place an end device and a router end connect them using a copper crossover wire and assign the IP address as shown in fig

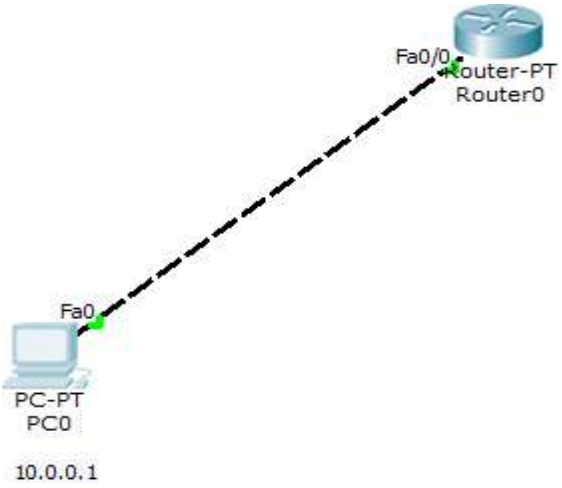
-> In router 0:

```
# config terminal
# Hostname R1
# enable secret P1
# Inter face fa 0/0
# ip address 10.0.0.1 255.0.0.0
# no shut
# line vty 0 5
# login
# Password P0
# exit
> exit
W r
```

In "i c n":
ping 10.0.0.1
Allows to access the router from PC0
telnet 10.0.0.2
User Access verification
Password P0
Enable
Password P1
show IP route

Observations:
-> Two Passwords are given while configuring the router one being the secret key for the router and other being the Password for login & corresponding access
-> The Password entered in the router are used in reverse order here i.e the user has to login first to verify access via P0 & then obtain router access with secrete key P1
-> Hence the admin in PC is able to run command as run in router CLI and see the result from PC

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

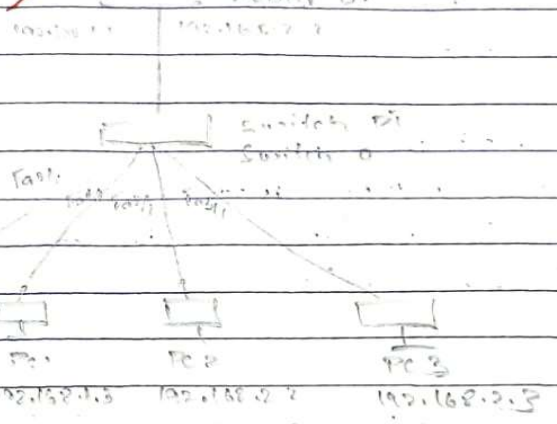
Aim: To construct a VLAN and make the PC's communicate among a VLAN .

Topology , Procedure and Observation:

Experiment - 11

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

Topology :



Procedure:

-> Place 4 end devices, a switch and a router & connect the end devices to switch & switch to router using copper-straight wiring

-> Assign IP to end devices as displayed in the topology Give VLAN no. name in switch & add

-> In Router 0

> enable

config terminal

Interface fa0/0

exit

exit

vlan database

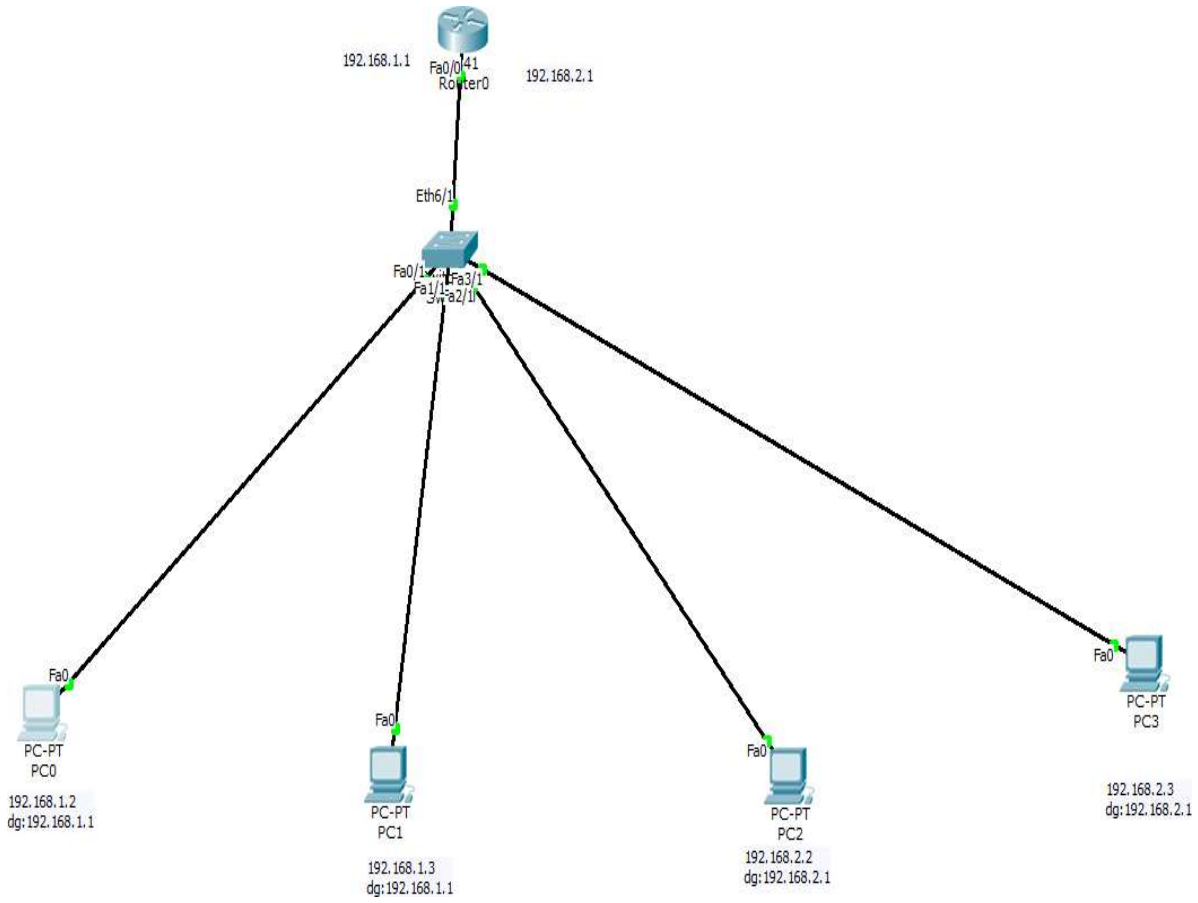
vlan 2 name csc ise

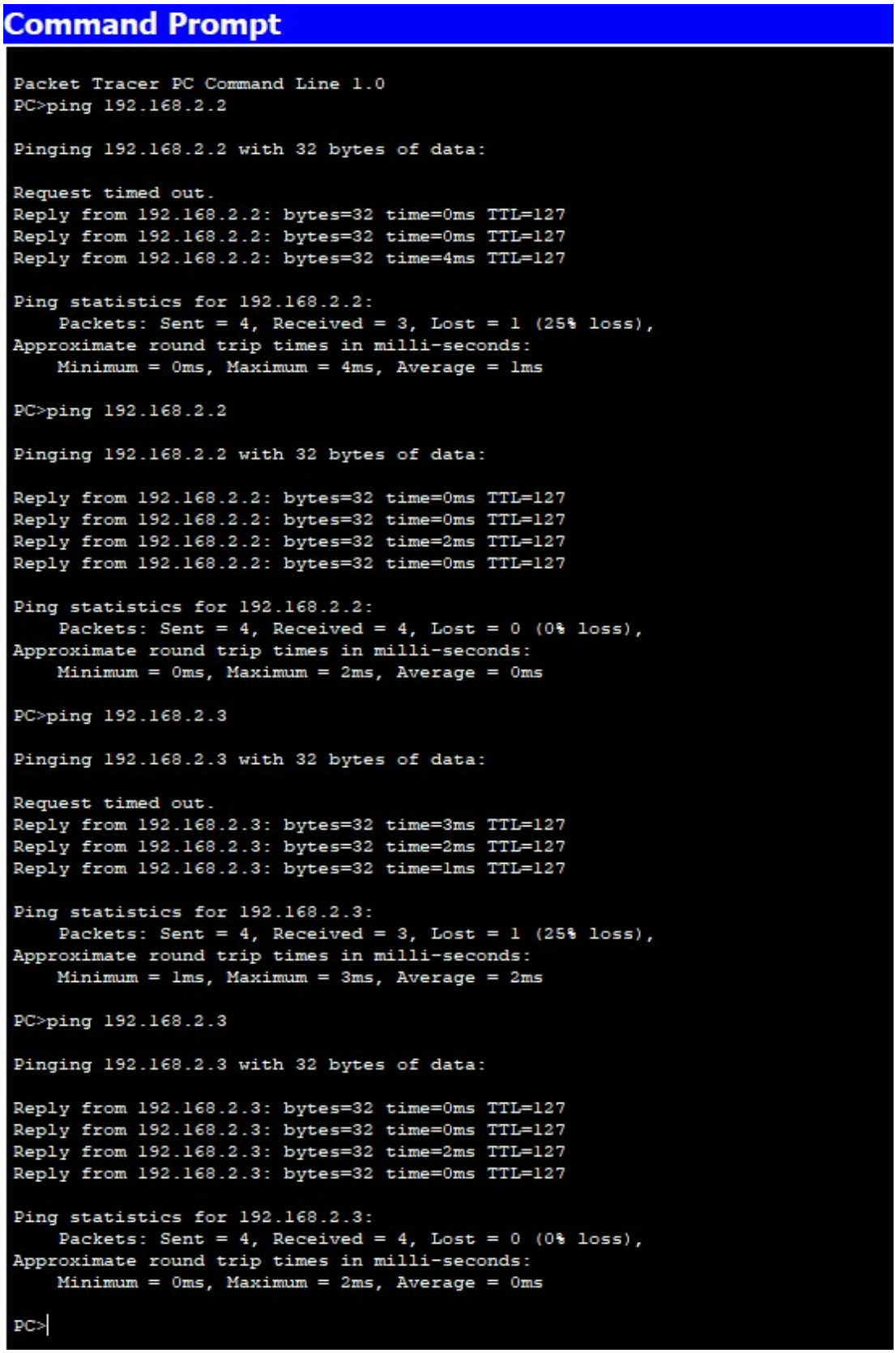
exit
config terminal
interface fa 0/0.1
encapsulation dot1q 2
ip address 192.168.2.1 255.255.255.0
no shut
exit

-> In switch 0
-> choose VLAN database
-> Turn Port status on for the corresponding ethernet
-> Enable Trunk

Observations
-> Proper Trunk configuration is enabled to make VLAN work Properly
-> VLAN trunking allows switches to forward frames from different VLAN over single link called trunk
-> Ping messages from different PC are observed to be reaching successfully hence forth

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

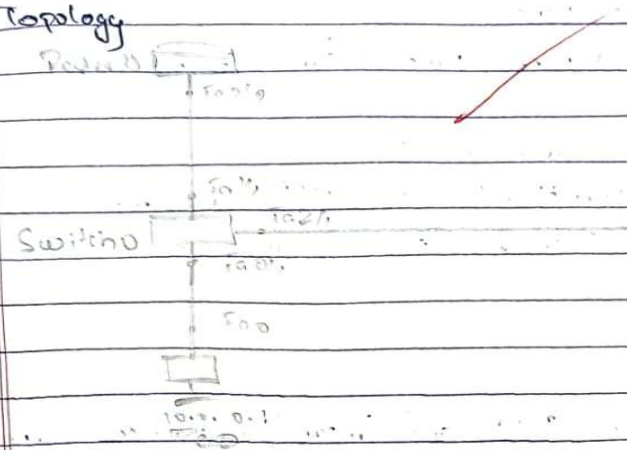
Aim: To construct a WLAN and make the nodes communicate wirelessly.

Topology , Procedure and Observation:

Experiment - 10

Objective: Construct a WLAN and make the nodes communicate wirelessly

Topology



Procedure:

-> Place 3 end devices, a switch, a router, and an access-point. Connect the end devices PC0, access-point and the router 0 to switch 0 using copper-straight wire

-> Assign the IP address as shown in the topology

-> In PC0

-> Turn the Pc off

-> Remove the port

-> Place the LinkSys-WMP300N port to the Pc and turn it back on

-> Configure Access Point 0:

-> Port status should be set to 'ON'

-> Set SSID as name as "BMSCECSECN"

-> Set channel authentication to 'WEP' and set

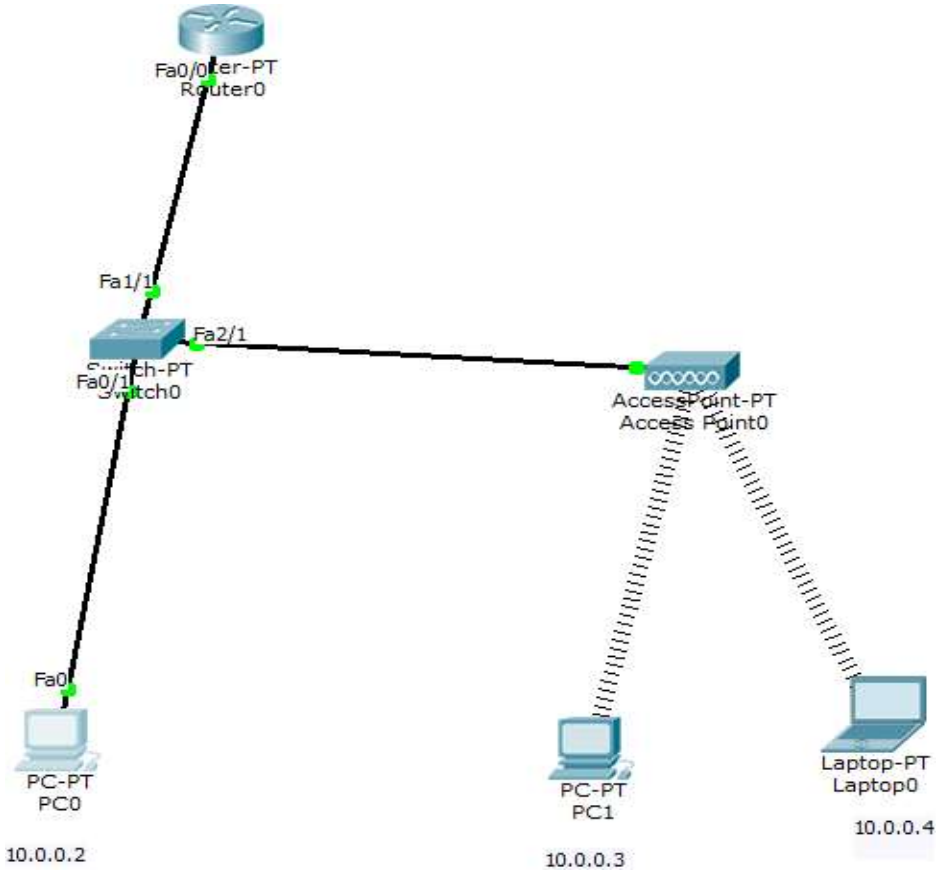
key as '1234567890'

-> In PC1 and laptop0
  -> Turn the system off
  -> Remove the port
  -> Place the wireless port & turn it back on
  -> In config
    -> Set the same SSID
    -> Set authentication to WEP and enter same key
-> Ping from different devices and observe the transmission

Observations:-

-> After the setup of PC1 and laptop0, wireless connecting with Access Point0 indicating successful wireless connection
-> Devices could connect to WLAN since they were in the network range
-> Signal strength decrease with increase in distance

Screen Shots:



PC0

Physical | Config | Desktop | Custom Interface

Command Prompt

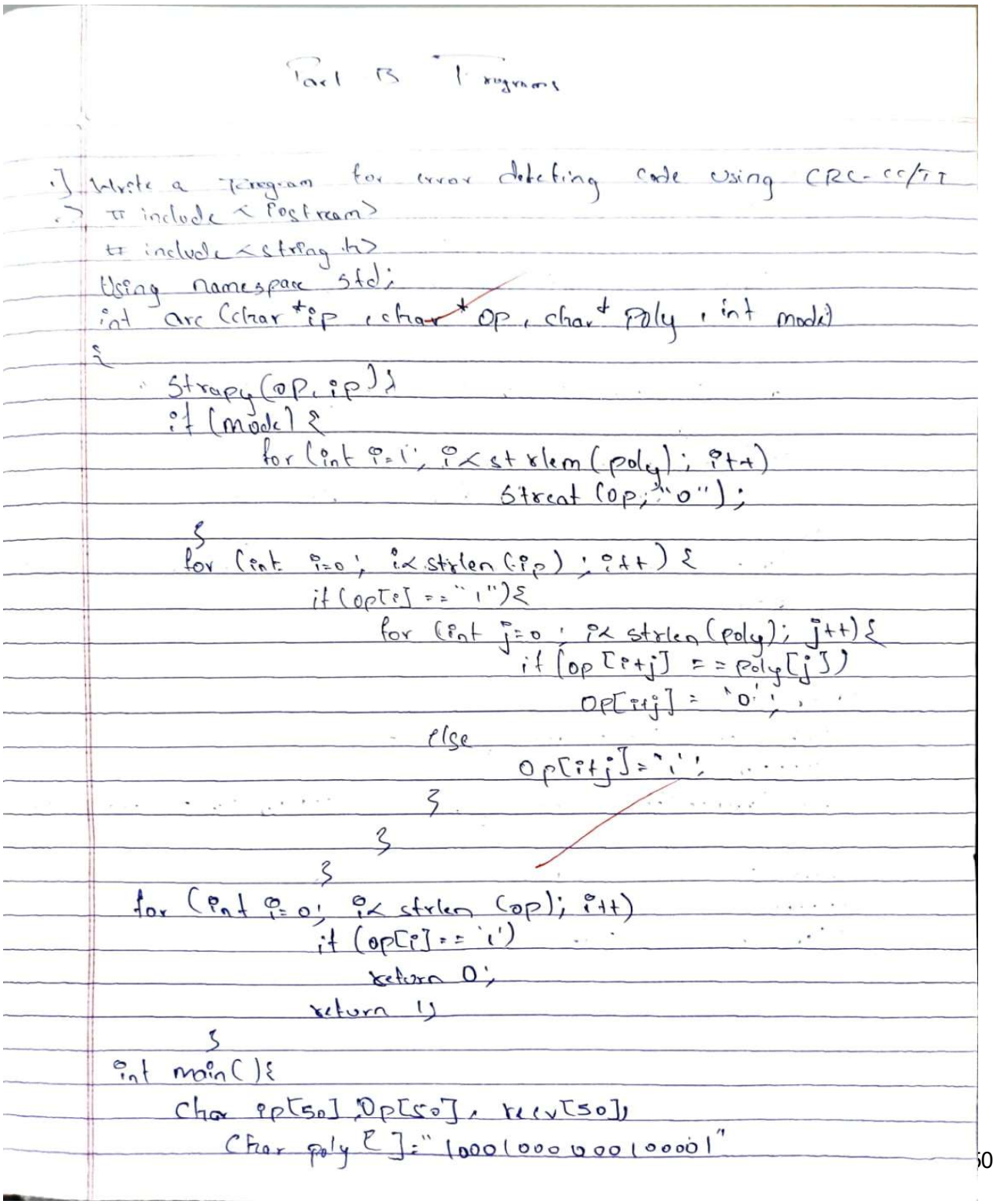
```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time=22ms TTL=128
Reply from 10.0.0.3: bytes=32 time=6ms TTL=128
Reply from 10.0.0.3: bytes=32 time=3ms TTL=128
Reply from 10.0.0.3: bytes=32 time=7ms TTL=128

Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 22ms, Average = 9ms

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=19ms TTL=128
Reply from 10.0.0.4: bytes=32 time=5ms TTL=128
Reply from 10.0.0.4: bytes=32 time=6ms TTL=128
Reply from 10.0.0.4: bytes=32 time=7ms 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 = 5ms, Maximum = 19ms, Average = 9ms

PC>
```

PART-B

Program 14

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

Code and Output:



```
Part B Programs

1) Write a program for error detecting code using CRC-CCITT
#include <iostream>
#include <string.h>
Using namespace std;
int arc (char *ip, char *op, char * poly, int mode)
{
    strcpy(op, ip);
    if (mode) {
        for (int i=1; i<strlen(poly); i++)
            strcat(op, "0");
    }
    for (int i=0; i<strlen(ip); i++) {
        if (op[i] == '1') {
            for (int j=0; i<strlen(poly); j++) {
                if (op[i+j] == poly[j])
                    op[i+j] = '0';
                else
                    op[i+j] = '1';
            }
        }
    }
    for (int i=0; i<strlen(op); i++)
        if (op[i] == '1')
            return 0;
        return 1;
}
int main(){
    char ip[50], op[50], recv[50];
    char poly[] = "10001000000100001";
```

```
cout << "Enter the input message in binary"
cin >> ip;
crc(ip, op, poly, 1);
cout << "The transmitted message is:" << ip
    << op + strlen(ip) << endl;

cout << "Enter the received message in binary :" << endl;
cin >> recv;
if (crc(recv, op, poly, 0))
        cout << "No error in data" << endl;
else
    cout << "Error in data transmission has occurred" << endl;
    return 0;
}
```

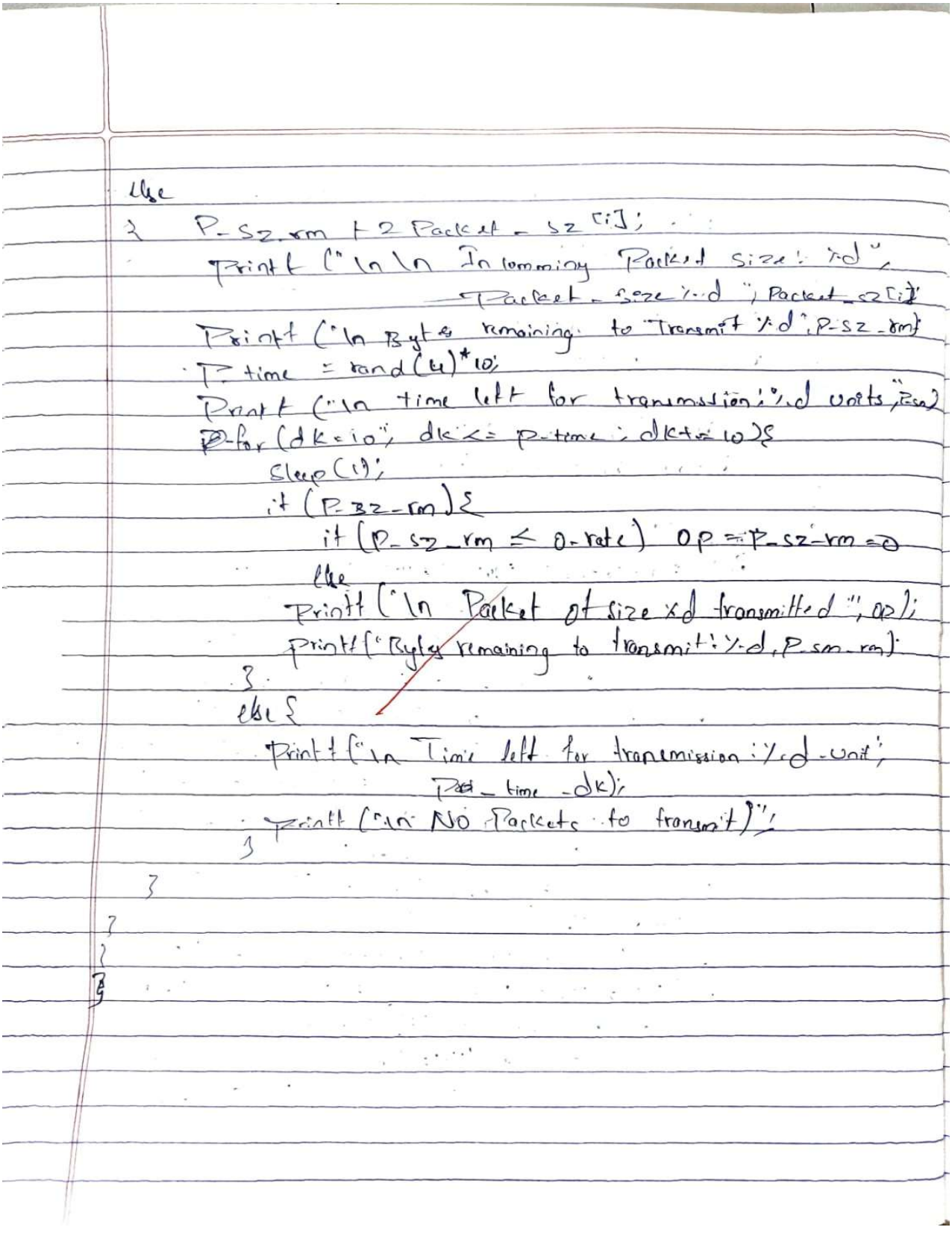
Output

```
Enter the input message in binary
111101
The transmitted message is : 111101011010111100111010

Enter the received message in binary
111101
No error in data
```

Program 15

Write a program for congestion control using Leaky bucket algorithm.

Code and Output:



```
else
{   P_sz_rm = Packet_sz[i];
    Printf("\n\n Incomming Packet size: %d",
                Packet_size %d ", Packet_sz[i]);
    Printf("\n Bytes remaining to Transmit %d", P_sz_rm);
    P_time = rand(4)*10;
    Printf("\n time left for transmission: %d units", P_time);
    for(dk=10; dk<= P_time; dk+=10){
        sleep(1);
        if(P_sz_rm){
            if(P_sz_rm <= o_rate) op = P_sz_rm = 0
            else
            Printf("\n Packet of size %d transmitted", op);
            Printf("Bytes remaining to transmit: %d", P_sm_rm);
        }
        else{
            Printf("\n Time left for transmission: %d unit",
                        P_time_dk);
            Printf("\n No Packets to transmit");
        }
    }
}
```

```
else
} P_sz_rm = Packet_sz[i];
  Printf("\n\n Incomming Packet size: %d",
                Packet_size %d", Packet_sz[i]);
  Printf("\n Bytes remaining to Transmit %d", P_sz_rm);
  p_time = rand(4)*10;
  Printf("\n time left for transmission: %d units", p_time);
  for(dk=10; dk<= p_time; dk+=10){
     Sleep(1);
     if(P_sz_rm){
        if(P_sz_rm <= o_rate) op = P_sz_rm = 0
        else
        Printf("\n Packet of size %d transmitted", op);
        printf("Bytes remaining to transmit: %d", P_sm_rm);
     }
     else{
        Printf("\n Time left for transmission: %d unit",
                     P_time_dk);
        Printf("\n No Packets to transmit)";
     }
  }
}
```

Output :

Packet[0] : 30 bytes
Packet[1] : 10 bytes
Packet[2] : 10 bytes
Packet[3] : 50 bytes
Packets[4] : 30 bytes

Enter Output rate : 100
Enter bucket size : 50 :

Incomming packet size : 30
Bytes remaining to transmit : 30
Time left for transmission : 20 units
Packet of size 30 transmitted .. Bytes remaining to transmit :
Time left for transmission : 0 units
No packet to transmit

Incoming Packets size : 10
Bytes Remaining to transmit : 100
Time left for transmission : 30
Packet of size 10 transmitted .. Byte remaining to transmit : 0
Time left for transmission : 10
No packet to transmit :
Time left for transmission : 0 units
No packet to transmit :

Incoming packet size : 10
Byte remaining to transmit : 10
Time left for transmission : 10 units
Packet of size 10 transmitted... Byte remaining to transmit : 0

Incoming Packet size : 30
Byte remaining to transmit : 30
Time left for transmission : 30 units
Packet of size 30 transmitted... Byte remaining to transmit : 0
Time left for transmission : 10 units
No Packets to transmit
Time left for transmission : 0 units
No Packets to transmit.

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:

3] Using TCP/IP sockets, write a client-server program to make client sending the filename & the server to send back the contents of requested file if present

Client side :-

```
# include <unistd>
int main() {
int soc, n;
char buffer[1024], fname[50];
struct sockaddr_in addr;
soc = socket(PF_INET, SOCK_STREAM, 0);
addr.sin_family = AF_INET;
addr.sin_port = htons(7891);
addr.sin_addr.s_addr = inet_addr("127.0.0.1");
while (connect(soc, (struct sockaddr *) &addr, sizeof(addr))
Printf("\n Enter file name");
Scanf("Enter file name");
Scanf("%s", fname);
Send(soc, fname, sizeof(fname), 0);
Printf("\n Received response \n");
while ((n = recv(soc, buffer, sizeof(buffer), 0)) > 0)
        Printf("%s", buffer);
        return 0;
}
```

Server side:

```
#include <stdio.h>
#include <fcntl.h>
#include<unistd.h>
#include<sys/socket.h>
int main(){
    int welcome, new_sock, fd=0;
    char buffer[1024], fname[50];
    struct sockaddr_in addr;
    welcome = socket(PF_INET, SOCK_STREAM, 0);
    addr.sin_family = AF_INET;
    addr.sin_port = htons(7891);
    addr.sin_addr.s_addr = inet_addr("127.0.0.1");
    bind(welcome,(struct sockaddr*)&addr,sizeof(addr));
    printf("\n Server is online");
    listen(welcome,5);
    new_sock = accept(welcome, NULL, NULL);
    read(new_sock, fname, 50, 0);
    Printf("\n Requesting for file : %s\n", fname);
    fd = open(fname, O_RDONLY);
    if(fd<0)
        send(new_sock,"\n File not found \n", 50, 0);
    else
    while((n= read(fd, buffer, sizeof(buffer))>0
        send(new_sock, buffer, 2n, 0);
    Printf("\n Request sent \n")
    close(fd)
    return 0;
}
```

Output

Server is Online
Requesting for file : test.txt
Request Send
Client is connected to server
Enter file name : test.txt
Received response
Hello Helloworld.

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:

4] Using UDP sockets, write a client - server Program to make client sending the filename & the server to send back contents of the requested file if present

Server Program:

```
#include <stdio.h>
#include <strings.h>
#include <sys/types.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#define PORT 5000
#define MAXLINE 1000
int main()
    char buffer[100];
    char message = "Hello client";
int listenfd, len;
struct sockaddr_in servaddr, cliaddr;
bzero(&servaddr, sizeof(servaddr));
listenfd = socket(AF_INET, SOCK_DGRAM, 0);
servaddr.sin_port = htons(PORT);
bind(listenfd, (struct sockaddr*)&servaddr, sizeof(addr)
len = sizeof(cliaddr)
int n = recvfrom(listenfd, buffer, sizeof(buffer), 0,
        (struct sockaddr*)&cliaddr, &len);
buffer[n] = '\0';
puts(buffer)
sendto(listenfd, message, MAXLINE, 0, (struct sockaddr*)
    &clientaddr, sizeof(cliaddr));
}
```

Client driven Program

```
#include <stdio.h>
#include <strings.h>
#include <sys/types.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <unistd.h>
#include <stdlib.h>
#define PORT 5000
#define MAXLINE 1000
int main() {
    char buffer[1000];
    char *message = "Hello server";
    int sockfd, n;
```

?

```
struct sockaddr_in servaddr;
bzero (& servaddr, sizeof (servaddr));
servaddr.sin_addr.s_addr = inet_addr ("127.0.0.1");
servaddr.sin_port = htons (PORT)
sockfd = socket (AF_INET, SOCK_DGRAM, 0)
if (connect (sock_fd, (struct sockaddr*) & servaddr.
                                    sizeof (servaddr))
    printf (" In Error: Connect Failed \n");
    exit (0);
}
send to (sock fd, message, MAX (INET, 0, (struct
                                    sockaddr*) NULL)
recfrom (sock_fd, buffer, sizeof (buffer), 0, (struct
                    sockaddr*) NULL, NULL);
    puts (buffer
    close (sock_fd);
}
```

3.0 //

Output

Server Output

Server is online

HELLO SERVER

Client Output

Hello Client.