BMS COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU

A Lab Report on

Computer Networks Lab

Submitted in partial fulfillment for the award of the degree of

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

RUSHIL
BINDROO
1BM21CS172

Under the Guidance of:

Prof. Lohith JJ
Assistant Professor
BMSCE

Department of Computer Science and Engineering
BMS College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560019

2022-2023

CERTIFICATE

This is to certify that the Lab work entitled "COMPUTER NETWORKS LAB" carried out by **Rushil Bindroo (1BM21CS172),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2023. The Lab report has been approved as it satisfies the academic requirements in respect of a **COMPUTER NETWORKS - (22CS4PCCON) work prescribed for the said degree.**

Prof. Lohith J J
Assistant Professor
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

INDEX

Sl. No.	Date	Experiment Title
CYCLE - 1		
1	15/06/2023	Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message.
2	22/06/2023	Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply.
3	13/07/2023	Configure default route, static route to the Router.
4	13/07/2023	Configure DHCP within a LAN and outside LAN.
5	20/07/2023	Configure Web Server, DNS within a LAN.
6	20/07/2023	Configure RIP routing Protocol in Routers
7.1	27/07/2023	Configure OSPF routing protocol
7.2	03/08/2023	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)
8.1	10/08/2023	To construct a VLAN and make the PC's communicate among a VLAN.
8.2	10/08/2023	To construct a WLAN and make the nodes communicate wirelessly.
9	10/08/2023	Demonstrate the TTL/ Life of a Packet.
10	10/08/2023	To understand the operation of TELNET by accessing the router in the server room from a PC in the IT office.
CYCLE - 2		
11	17/08/2023	Write a program for error detecting code using CRC CCITT (16-bits).
12	17/08/2023	Write a program for congestion control using Leaky bucket algorithm.
13	24/08/2023	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.
14	24/08/2023	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.

LAB 1

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

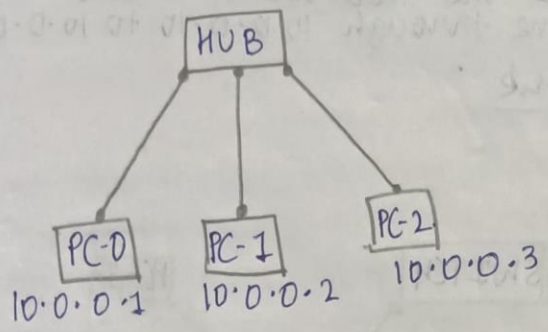
Observation

16/06/2023 Lab-01

Create a topology consisting of three or more devices connected with the help of a hub, switch and combined.

Aim: To understand the working of Hub, Switch and hybrid network.

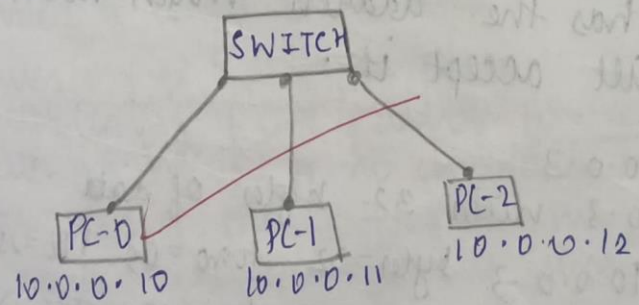
(i) Using Hub



Observation:

Whenever a source node sends data in the network, the hub recieves the data from the cource and brodcasts it over the network ie it sends the message to all the remaining nodes in the network and the node whose destination addres matches with the message send recieves & accepts that data. The communication was carried between 10.0.0.1 → 10.0.0.3.

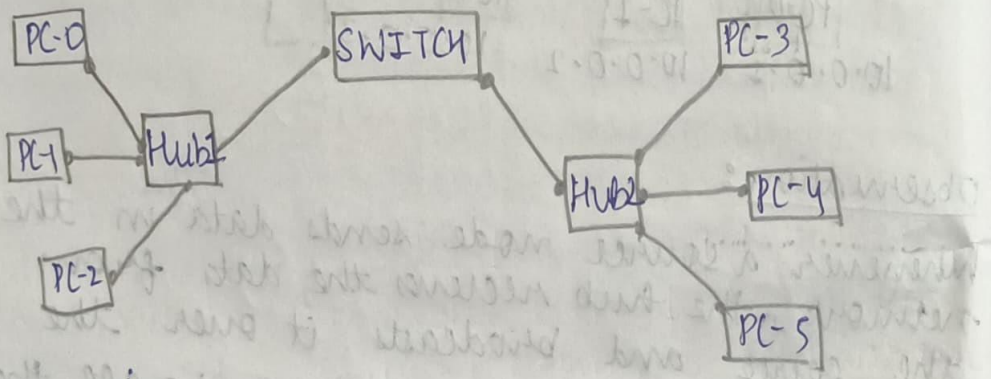
(ii) Using Switch



Observation:

Whenever a source node sends the data to other node, the switch first recieves the data. In the first time the switch sends it to all the other devices but when sent again it sends it to only the destination device which has the address match. The communication is done through 10.0.0.10 to 10.0.0.13

iii) Hybrid network



Observation:

The PC-0 sends the message to the Hub-1 which recieves it and then broadcasts to PC-1, PC-2 and the switch. The switch then sends it to Hub-2 which recieves it and broadcasts it to all the devices PC-3, PC-4 PC-5. The PC-4 which is the destination device and has the address match with the message will accept it.

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=0s TTL=255

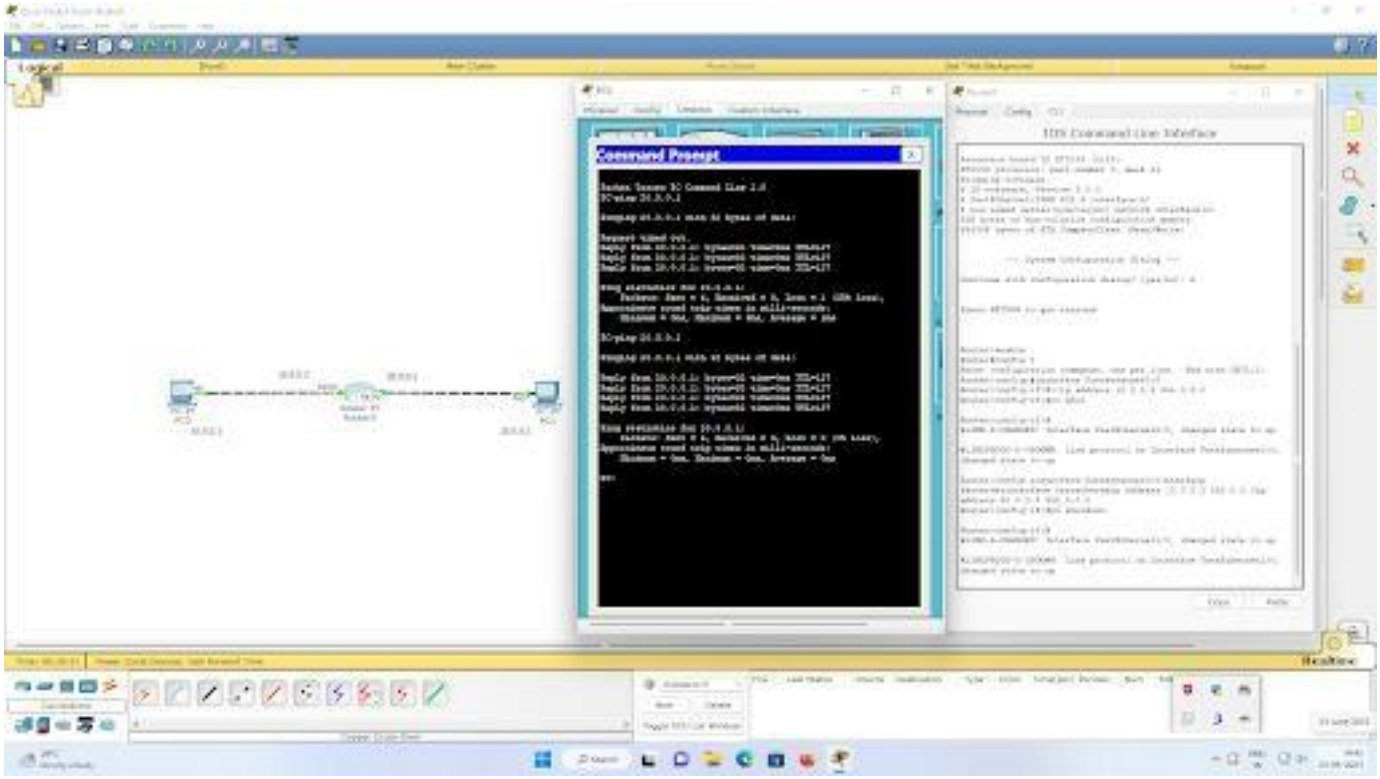
Reply from 10.0.0.3 bytes=32 time=0s TTL=255

Reply from 10.0.0.3 bytes=32 time=0s TTL=255

Reply from 10.0.0.3 bytes=32 time=0s TTL=255

Packets: Sent: 4 Recieved 4, Lost = 0

System Output-



LAB 2

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

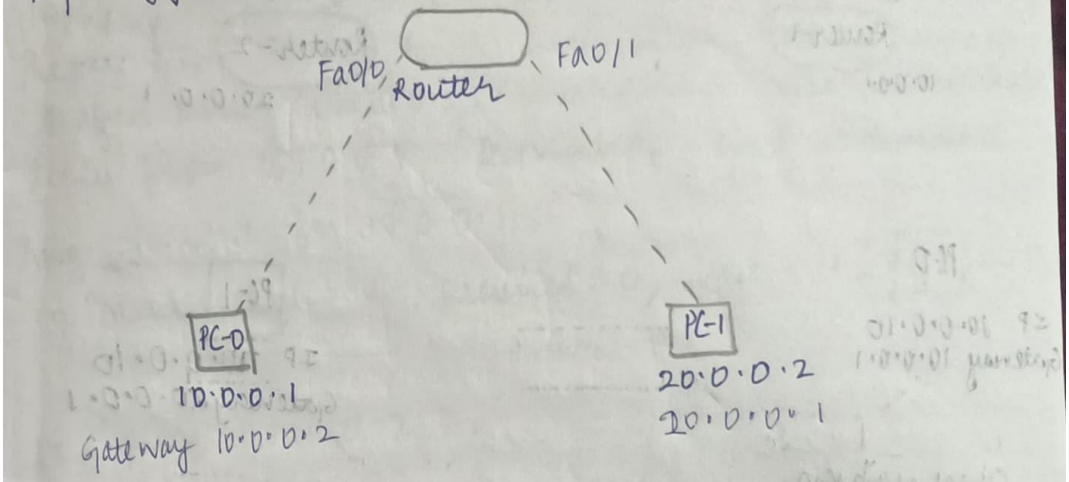
Observation-

23/06/2023 lab-02

Q Create a topology consisting of 2 devices connected with the help of a router

Aim : To check the functionality of a Router

Topology-1



Observation

The router Router-PT is connected to two end devices PC-0 & PC-1. First we configure the IP address of PC-0 & PC-1 as 10.0.0.1 and 20.0.0.2 since they belong to different network.

Then we configured the Router for each interface of PC-0 & PC-1 by writing a set of commands in the CLI window of the router

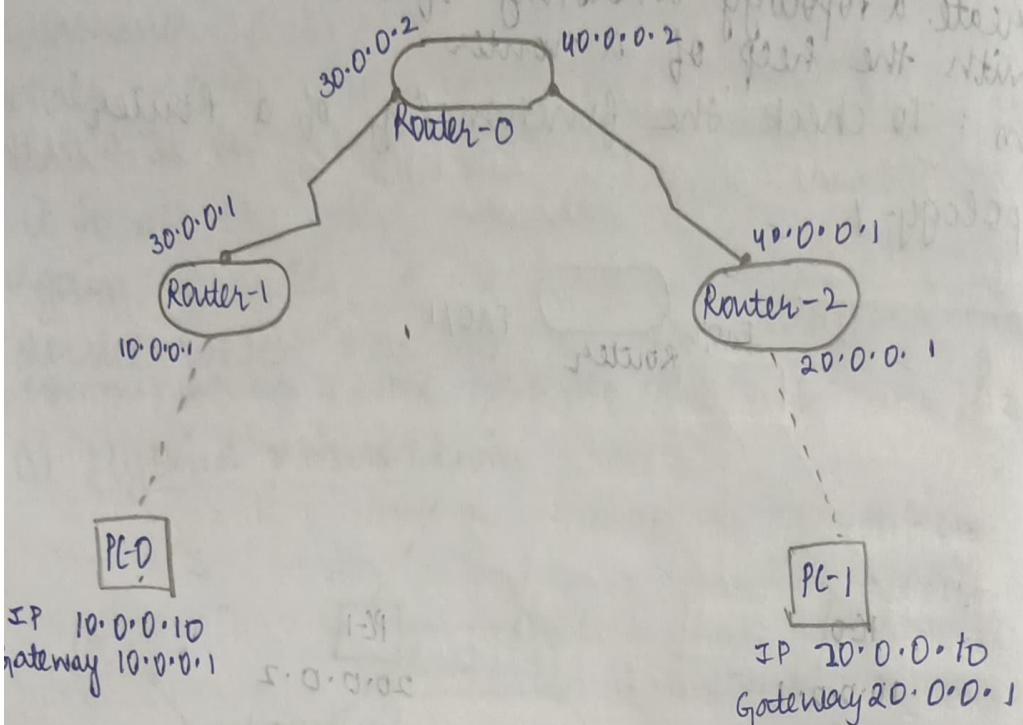
Router>enable
Router# configure terminal
Router(config)# interface Fa0/0
Router(config-if)# ip address 10.0.0.2 255.0.0.0
Router(config-if)# no shutdown

After setting the interface we ping but it shows request timed out since we have not set the gateway.

After setting the gateway for each device as 10.0.0.2 and 20.0.0.1 respectively, the connection is made.

The 10.0.0.2 will be the gateway for PC-0

Topology-2



Observation

In this topology Router-1 is connected to PC-0 and Router 2 is connected to PC-1. Router-0 is connected to Router-1 & Router-2 afterwards. For PC-0 IP address is set as 10.0.0.10 and the gateway for the connection to Router-1 is set as 10.0.0.1

For PC-2, IP address is set as 20.0.0.10 and the gateway for the connection is set as 20.0.0.1

Then for the Router-0 we set the gateway for Router-1 & Router-2 connection by the commands in the CLI window and setting as 30.0.0.2 & 40.0.0.1 respectively

Ping Requests

PC > ping 20.0.0.10

Pinging from 20.0.0.1: Destination with 32 bytes of data

Reply from 10.0.0.1 : Destination host unreachable
Reply from 10.0.0.1 : Destination host unreachable
Request timed out 10.0.0.1 : Destination host unreachable
Reply from 10.0.0.1 Destination host unreachable

Ping statistics for 20.0.0.10
Packets: sent = 4, received = 0, lost = 4 (100% loss)

PC > ping 30.0.0.2

Pinging 30.0.0.2 with 32 bytes of data:
Request timed out
Request timed out
Request timed out
Request timed out

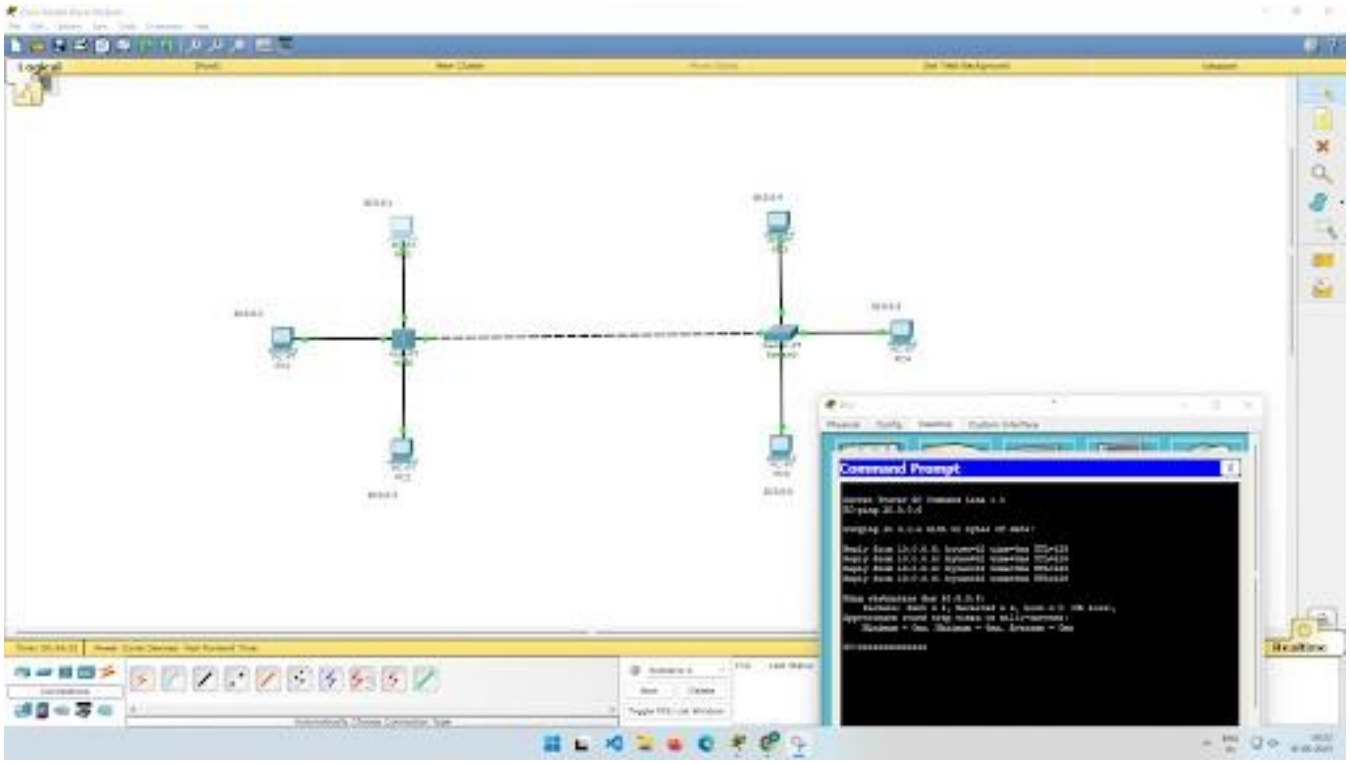
PC > ping 30.0.0.1
Pinging 30.0.0.1 with 32 bytes of data:
Reply from 30.0.0.1 bytes = 32 time = 0ms TTL = 255
Reply from 30.0.0.1 bytes = 32 time = 0ms TTL = 255
Reply from 30.0.0.1 bytes = 32 time = 0ms TTL = 255
Reply from 30.0.0.1 bytes = 32 time = 0ms TTL = 255

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

Page 36

(9)

System Output-



LAB 3

Configure default route, static route to the Router.

Observation-

30/6/2023

Lab-03 (continued)

Aim: To depict static connection between 4 routers in a topology of routers

Procedure:

Go to CLI in Router configuration and enter the following commands "show ip route". This command is used to display the IP routing table of a router. The router provides additional route information, including how the route was learned, how long the route has been in the table and which specific interface to use to get to a predefined destination.

Setting of ip routes:

Router-1

```
Router > enable
Router # configure terminal
Router (config) # ip route 20.0.0.0 255.0.0.0 30.0.0.2
Router (config) # ip route 40.0.0.0 255.0.0.0 30.0.0.2
Router (config) # exit
```

Observation:

```
Router # show ip route
C  10.0.0.0/8 is directly connected, FastEthernet 0/0
S  20.0.0.0/8 [1/0] via 30.0.0.2
C  30.0.0.0/8 is directly connected, Serial 2/0.
S  40.0.0.0/8 [1/0] via 30.0.0.2
```

Router-0

```
Router > enable
Router # configure terminal
```

Observation

Router # show ip route

S 10.0.0.0/8 [1/0] via 30.0.0.1
S 20.0.0.0/8 [1/0] via 40.0.0.1
C 30.0.0.0/8 is directly connected, Serial 2/0
C 40.0.0.0/8 is directly connected, Serial 3/0

Router-2

Router >enable
Router # configure terminal
Router (config) # ip route 10.0.0.0 255.0.0.0 40.0.0.2
Router (config) # ip route 30.0.0.0 255.0.0.0 40.0.0.2
Router (config) # exit

Observation:

S 10.0.0.0/8 [1/0] via 40.0.0.2
C 20.0.0.0/8 is directly connected, FastEthernet 0/0
C 40.0.0.0/8 is directly connected, Serial 3/0
S 30.0.0.0/8 [1/0] via 40.0.0.2

Ping commands

Ping 20.0.0.10

Pinging 20.0.0.10 with 32 bytes of data.
Reply from 20.0.0.10 bytes=32 time=2 ms TTL=125
Reply from 20.0.0.10 bytes=32 time=9 ms TTL=125
Reply from 20.0.0.10 bytes=32 time=6 ms TTL=125
Reply from 20.0.0.10 bytes=32 time=11 ms TTL=125

Ping statistics for 20.0.0.10:
Packets sent = 4, Received = 4, Lost = 0 (0% Loss)

Ping 30.0.0.1

Pinging 30.0.0.1 with 32 bytes of data
Reply from 30.0.0.1 bytes=32 time=0 TTL=225
Reply from 30.0.0.1 bytes=32 time=0 TTL=225
Reply from 30.0.0.1 bytes=32 time=0 TTL=225
Reply from 30.0.0.1 bytes=32 time=2 TTL=225
Packets sent = 4, Received = 4 lost = 0 (0% Loss

Observation

Router # show ip route

S 10.0.0.0/8 [1/0] via 30.0.0.1
S 20.0.0.0/8 [1/0] via 40.0.0.1
C 30.0.0.0/8 is directly connected, Serial 2/0
C 40.0.0.0/8 is directly connected, Serial 3/0

Router-2

Router >enable
Router # configure terminal
Router (config) # ip route 10.0.0.0 255.0.0.0 40.0.0.2
Router (config) # ip route 30.0.0.0 255.0.0.0 40.0.0.2
Router (config) # exit

Observation:

S 10.0.0.0/8 [1/0] via 40.0.0.2
C 20.0.0.0/8 is directly connected, FastEthernet 0/0
C 40.0.0.0/8 is directly connected, Serial 3/0
S 30.0.0.0/8 [1/0] via 40.0.0.2

Ping commands

Ping 20.0.0.10

Pinging 20.0.0.10 with 32 bytes of data.
Reply from 20.0.0.10 bytes=32 time=2 ms TTL=125
Reply from 20.0.0.10 bytes=32 time=9 ms TTL=125
Reply from 20.0.0.10 bytes=32 time=6 ms TTL=125
Reply from 20.0.0.10 bytes=32 time=11 ms TTL=125

Ping statistics for 20.0.0.10:
Packets sent = 4, Received = 4, Lost = 0 (0% Loss)

Ping 30.0.0.1

Pinging 30.0.0.1 with 32 bytes of data
Reply from 30.0.0.1 bytes=32 time=0 TTL=225
Reply from 30.0.0.1 bytes=32 time=0 TTL=225
Reply from 30.0.0.1 bytes=32 time=0 TTL=225
Reply from 30.0.0.1 bytes=32 time=2 TTL=225
Packets sent = 4, Received = 4 lost = 0 (0% Loss

Ping 10.0.0.10

Pinging 10.0.0.10 with 32 bytes of data

Reply from 10.0.0.10 bytes =32 time = TTL = 255

Reply from 10.0.0.10 bytes = 32 time = TTL = 255

Reply from 10.0.0.10 bytes = 32 time = TTL = 255

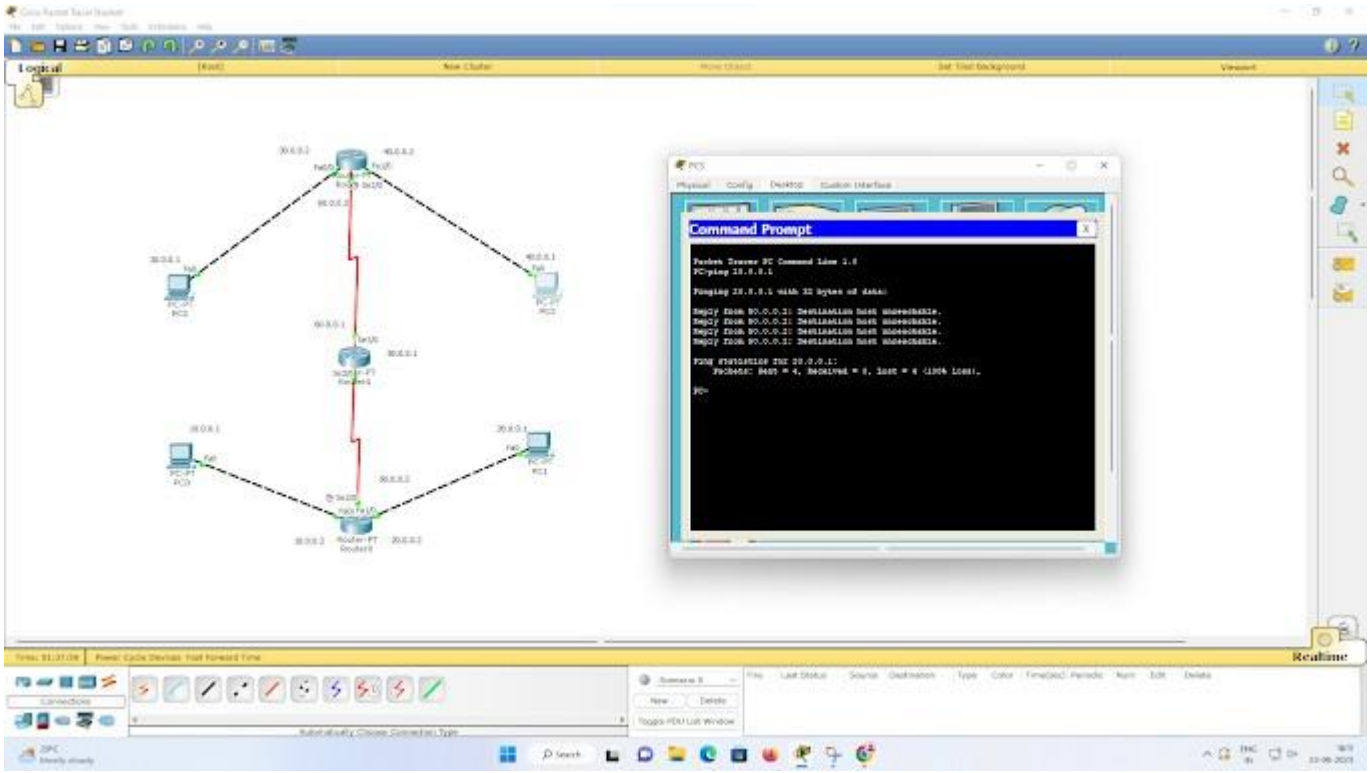
Reply from 10.0.0.10 bytes = 32 time = TTL = 255

Packets sent = 4 Recieved = 4 Lost = 0

Outcome

To understand how we can put static route to different routers in a connection of multiple routers.

System Output-



LAB 4
Configure DHCP within a LAN and outside LAN

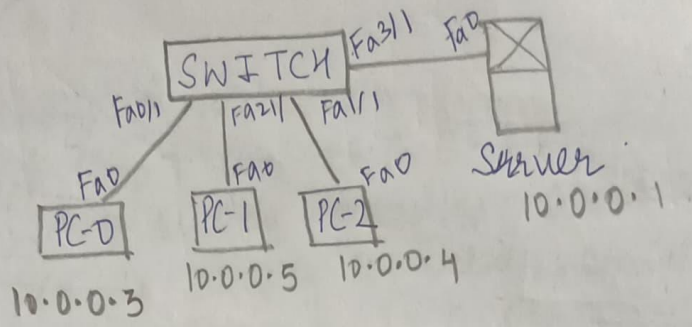
Observation-

14/7/2023

Lab-04

Aim: To configure DHCP within a lan, outside a LAN and three networks

Topology-1



Procedure:

1.) Three PC are connected to a switch and a general server is also connected.
2.) The server ip address is set to 10.0.0.1
3.) In the server services DHCP is switched on, now it serves as DHCP server.
4.) For the pool: server pool, starting ip address is set to 10.0.0.2.
5.) Now, for each ~~server~~ PC; set it as DHCP.
6.) All PC's are given IP Address Dynamically.

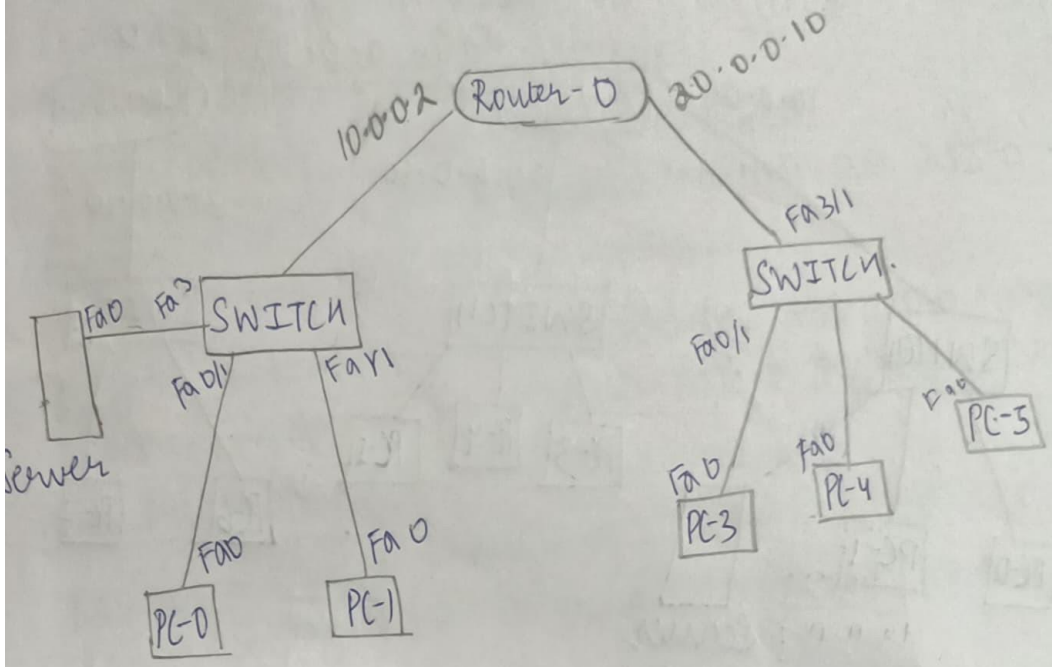
Observation:

IP address for all PC's is set automatically.

Result:

IP address of
PC-0 → 10.0.0.3
PC-1 → 10.0.0.5

Topology - 2



Procedure:

1.) The server ip address is set to 10.0.0.1
2.) Another server pool: Serverpool 1 and give gateway as 10.0.0.2.
3.) To the router set ip address for Fa0/1 & Fa3/1 as 10.0.0.2 and 20.0.0.10.
4.) Add the ip helper address command.
#(interface Fa 3/1
#(config) ip helper-address 10.0.0.1

5. For Server pool
Gateway: 10.0.0.2.
Start ip addr: 20.0.0.2.

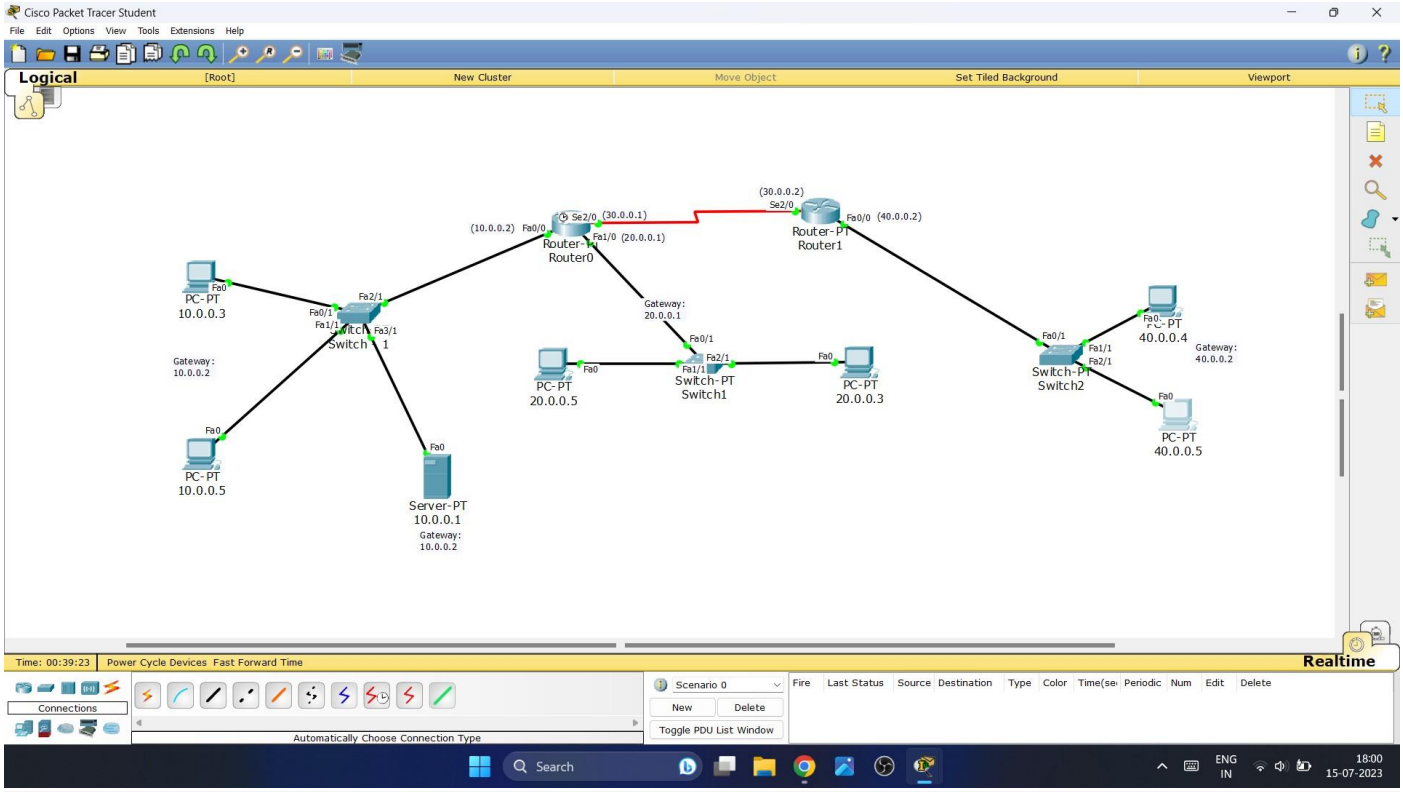
Observation

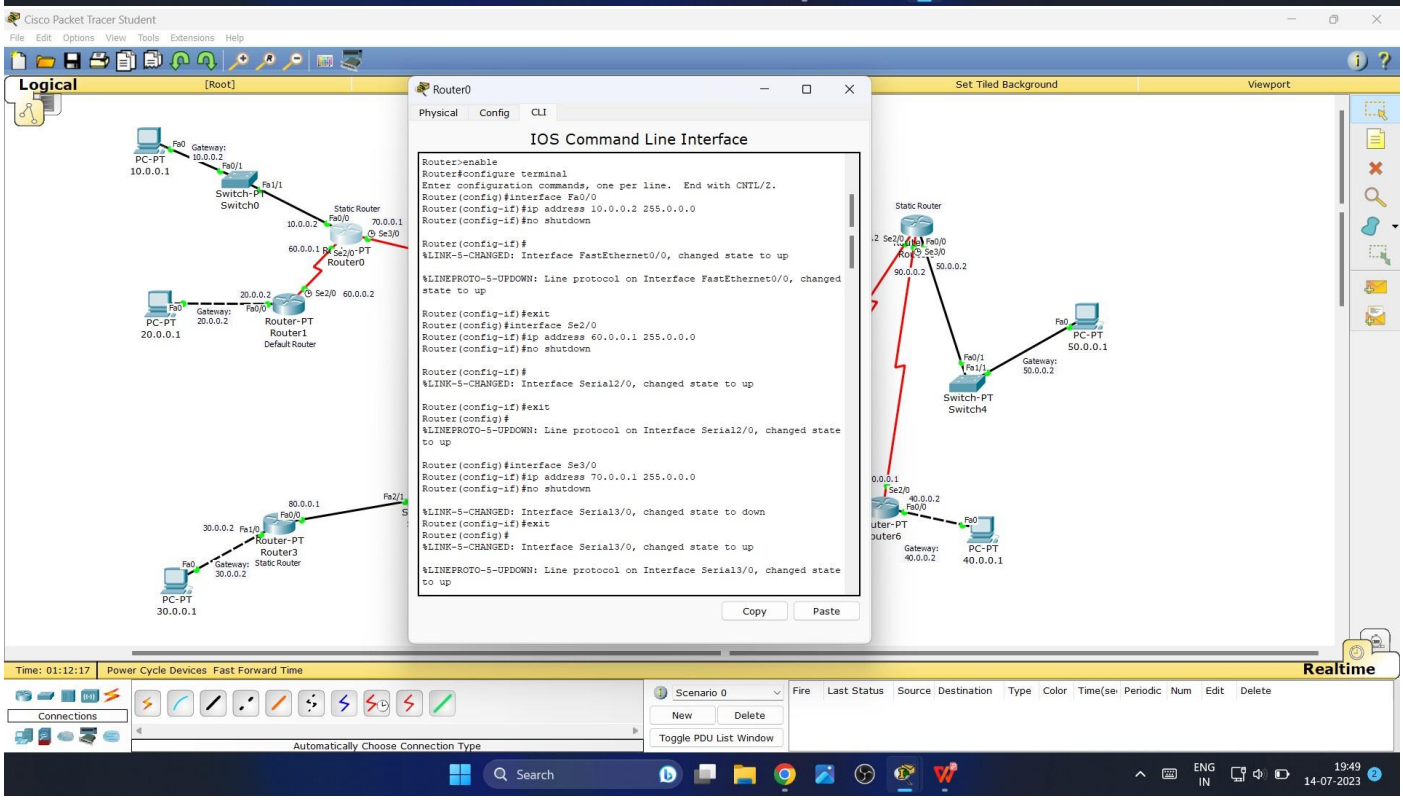
All PC's in LAN-2 will get ip address automatically.

Result:

PC-3 : 20.0.0.2
PC-4 : 20.0.0.4
PC-5 : 20.0.0.3

System Output-





LAB 5

Configure Web Server, DNS within a LAN.

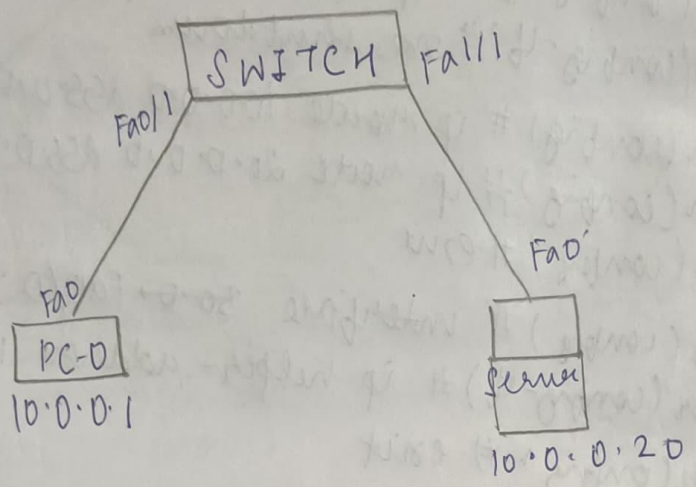
Observation-

21/7/2023

Lab-05

Aim:
Configure web server, DNS within a LAN.

Topology:



Procedure:

1.) Create a topology as shown using a PC, Server, and a switch.

2.) Set the ip address as 10.0.0.1 and 10.0.0.2 for PC and server respectively.

3.) In the server, under DNS service create new rushil.com website with URL 10.0.0.2 and add under HTTP, modify the index.html file and add name and usn as:
<h1> RUSHIL BINDROD </h1>
<h1>USN: 1BM21CS172</h1>

4.) In PC-0 go to desktop → web browser and type rushil.com. You'll be able to see the website with entered name & USN.

Result:

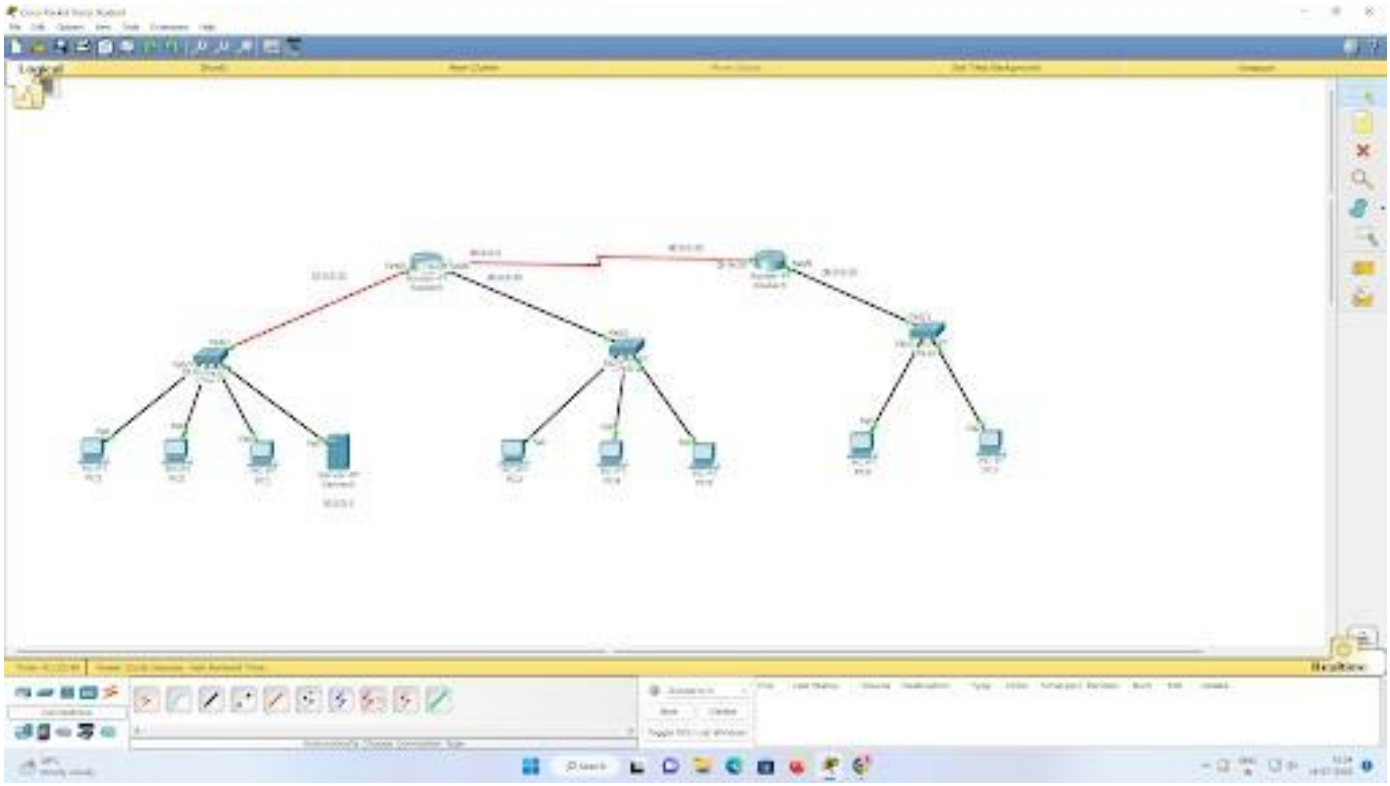
Web Browser
URL: http://rushil.com.

Cisco Packet Tracer
RUSHIL BINDROO
USN: 1BM21CS172

Outcome:

To learn how to configure a web server using the DNS protocol by following the given procedure.

System Output-

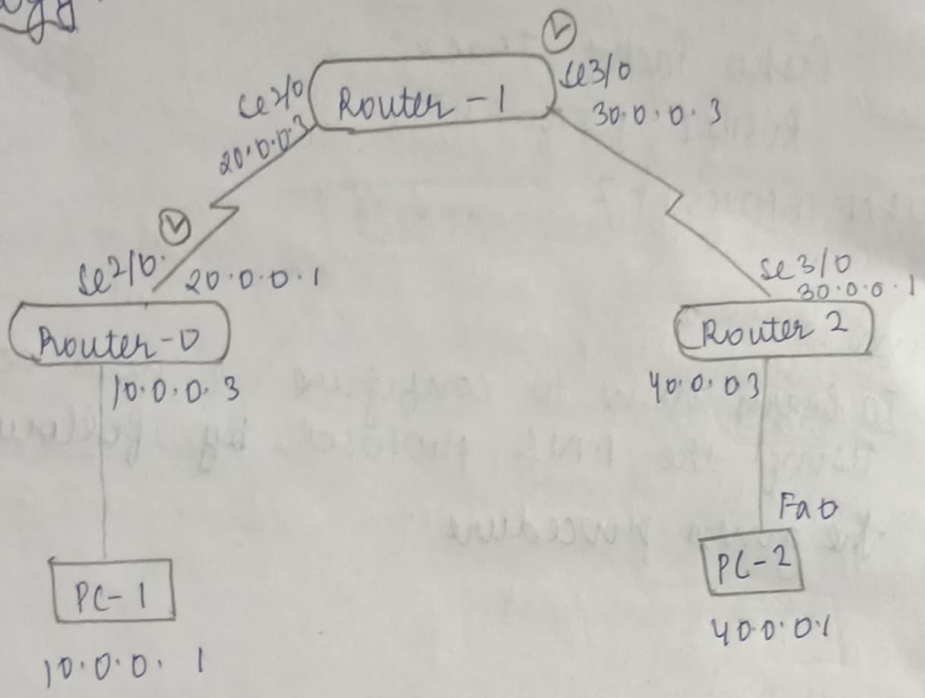


LAB 6
Configure RIP routing Protocol in Routers

Observation-

Aim: To configure RIP routing protocol in Routers.

Topology:



Procedure:

1.) Create a topology as shown above using 2PC's and 3 routers

2.) Configure the ip addresses of 2PC's as 10.0.0.1 and 40.0.0.1 for PC-1 and PC-2 respectively Set the gateways as 10.0.0.3 and 40.0.0.3

3.) Configure the routers (Router-0)

Router > enable
Router # config terminal
Router (config) # interface Fa 0/0
Router (config-if) # id addres 10.0.0.2 255.0.0.0
Router (config-if) # no shutdown
Router (config) # ip interface serial 2/0
Router (config-if) # ip address 20.0.0.1 255.0.0.0

Same - command for Router-1 & Router-2

→ For router 0,

Router(config)# interface se2/0
Router(config-if)# encapsulation ppp
Router(config-if)# no shutdown
Router(config-if)# exit

Repeat this for Router 1 interfaces se2/0 & se3/0
for router 2 se3/0

→ For router 0 (se2/0) and router 1 (se3/0)
Router(config)# interface se2/0
Router(config-if)# clock rate 64000
Router(config-if)# no shutdown
Router(config-if)# exit

For all the 3 routers, repeat this step:

> Router rip
> # network <network-1 ip>
> network <network 2 - ip>
> exit

Ping command

PC> ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data
Reply from 40.0.0.1 bytes=32 time = 9 ms TTL=125
Reply from 40.0.0.1 bytes=32 time = 2 ms TTL=125
Reply from 40.0.0.1 bytes=32 time = 3 ms TTL=125
Reply from 40.00.1 bytes=32 time = 12 ms TTL=125

Outcome: To learn to configure routers using RIP protocol using the commands learned and that RIP is an active routing protocol.

Same - command for Router-1 & Router-2

→ For router 0,

Router (config)# interface se2/0
Router (config-if)# encapsulation ppp
Router (config-if)# no shutdown
Router (config-if)# exit

Repeat this for Router 1 interfaces se2/0 & se3/0
for router 2 se3/0

→ For router 0 (se2/0) and router 1 (se3/0)

Router (config)# interface se2/0
Router (config-if)# clock rate 64000
Router (config-if)# no shutdown
Router (config-if)# exit

For all the 3 routers, repeat this step:

> Router rip
> # network <network-1 ip>
> network <network 2 - ip>
> exit

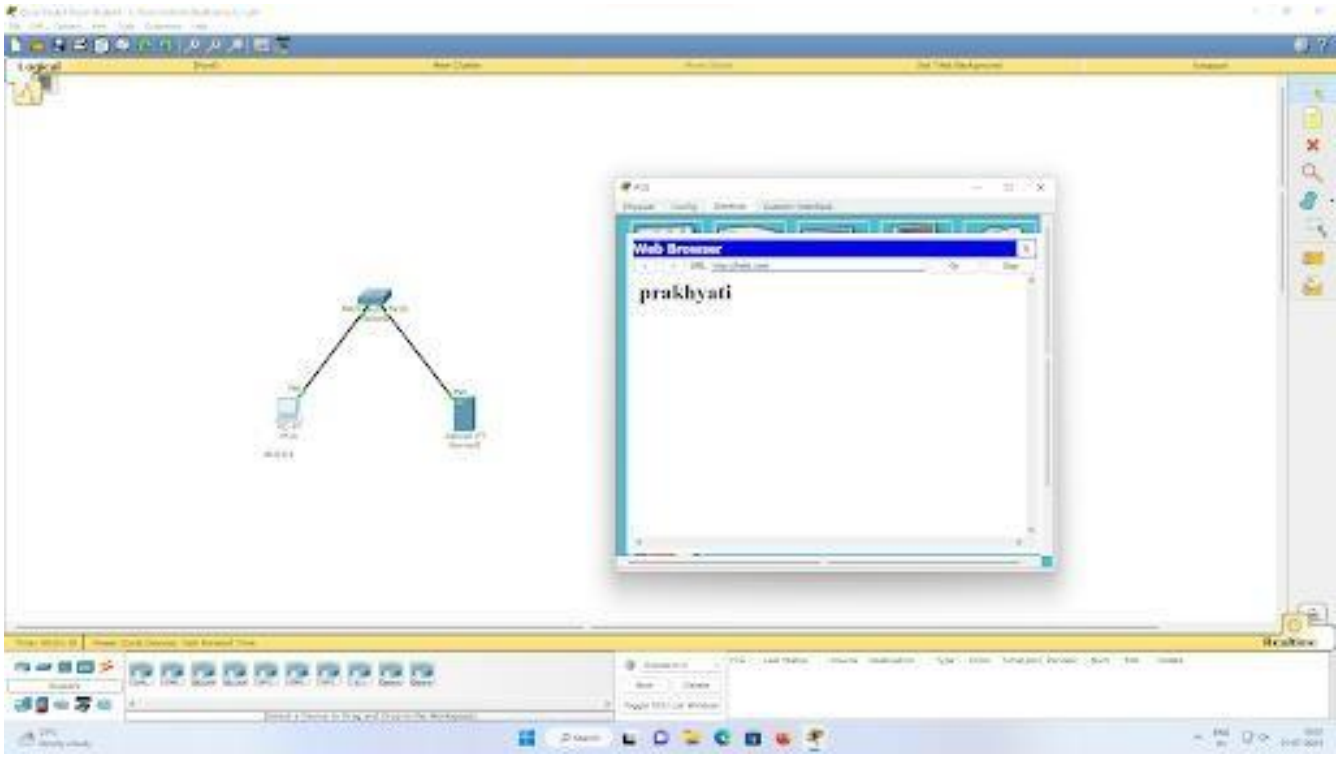
Ping command

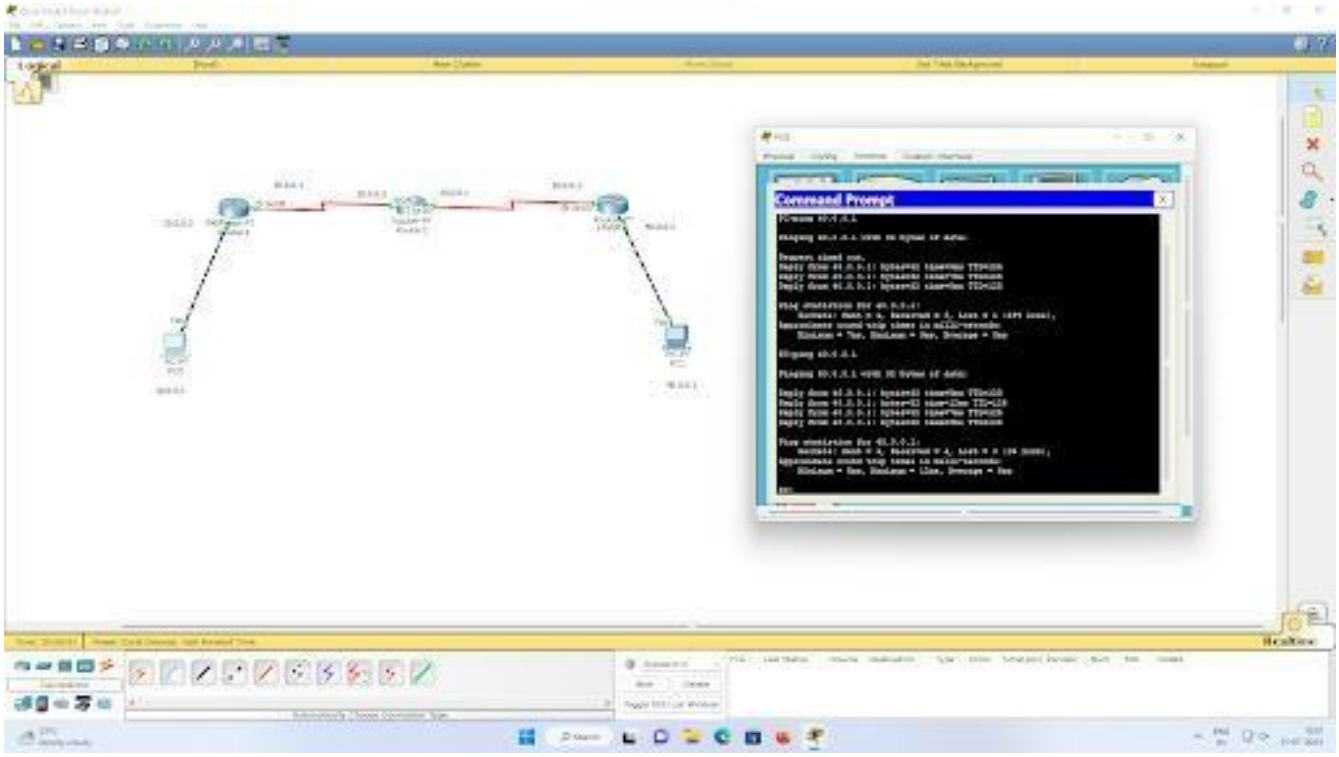
PC> ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data
Reply from 40.0.0.1 bytes = 32 time = 9 ms TTL=125
Reply from 40.0.0.1 bytes = 32 time = 2 ms TTL=125
Reply from 40.0.0.1 bytes = 32 time = 3 ms TTL=125
Reply from 40.00.1 bytes = 32 time = 12 ms TTL=125

Outcome: To learn to configure routers using RIP protocol using the commands learned and that RIP is an active routing protocol.

System Output-





LAB 7

Configure OSPF routing protocol

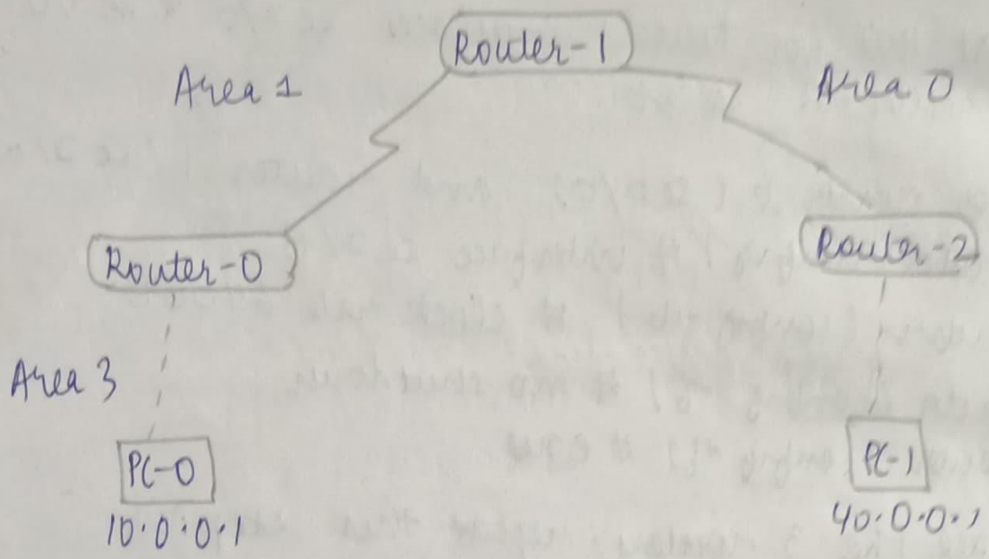
Observation-

04/08/2023

Lab-06

Aim: To demonstrate OSPF protocol in routers

Topology:



Procedure:

1.) Create the above topology with 3 routers and 2 PC's and set the ip address of the 2PC's

2.) Now provide the ip address of the router interfaces as well as the "encapsulation ppp" command at the router-router interface.

3.) For Router-0 & Router-1 in network 20 & 30 provide clock rate 64000.

4.) For each router enable ip routing by configuring OSPF in all routers and provide the router-id.

5.) Create dedicated loopback interface for all the routers.

6.) Create a virtual link to connect.

CLI commands

1) Configuring ip address to all router - interfaces

Router-1

Router > enable
Router # configure terminal
Router (config) # interface se2/0
Router (config-if) # ip address 20.0.0.2 255.0.0.0
Router (config-if) # encapsulation PPP
Router (config-if) # no shutdown
Router (config-if) # exit
Router (config) # interface se3/0
Router (config-if) # ip address 30.0.0.1 255.0.0.0
Router (config-if) # encapsulation PPP
Router (config-if) clock rate 64000
Router (config-if) # no shutdown

2.) Enabling ip routing by configuring ospf in all routers.

Router-0

Router (config) # router ospf 1
Router (config-router) # router-id 1.1.1.1
Router (config-router) # network 10.0.0.0 0.255.255.255 area 0
Router (config-router) # network 20.0.0.0 0.255.255.255 area 0
Router (config-router) # exit

Router -1

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

Router-2

Router(config)# router ospf 1
Router(config-router)# router-id 3.3.3.3
Router(config-router)# network 30.0.0.0 0.255.255.255 area0
Router(config-router)# network 40.0.0.0 0.255.255.255 area 2
Router(config-router)# exit

3.) Creating dedicated loopback interface for all routers.

Router-0

Router(config)# interface se2/0
Router(config-if)# interface loopback 0
Router(config-if)# ip add 172.16.1.252 255.255.0.0
Router(config-if)# no shutdown.

Router-1

Router(config)# interface se2/0
Router(config-if)# interface loopback 0
Router(config-if)# ip add 172.16.1.253 255.255.0.0
Router(config-if)# no shutdown

Router-2

Router(config)# interface se2/0
Router(config-if)# interface loopback 0
Router(config-if)# ip add 172.16.1.254 255.255.0.0
Router(config-if)# no shutdown.

4.) Create virtual link between Router-0 & Router-2.

Router-0

Router(config)# router ospf 1

Router (config-router)# area 1 virtual-link 2.2.2.2
Router(config-router)# exit

Router-1

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

Ping Command

pc > ping 40.0.0.10

Packets: Sent = 4, Received = 0, Lost = 4.

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 = 6ms TTL=125
Reply from 40.0.0.10 bytes = 32 time = 8ms TTL=125
Reply from 40.0.0.10 bytes = 32 time = 7ms TTL=125

Ping statistics for 40.0.0.10:
Packets: Sent = 4, Received = 4, Lost = 0.

Outcome:

OSPF stands for Open Shortest Path First is a link layer/ application layer protocol which is used to find the best route for packet transfer and universes creating a virtual link.

Router (config-router)# area 1 virtual-link 2.2.2.2
Router(config-route)# exit

Router-1

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

Ping Command

pc > ping 40.0.0.10

Packets: Sent = 4, Received = 0, Lost = 4.

pc > ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data.

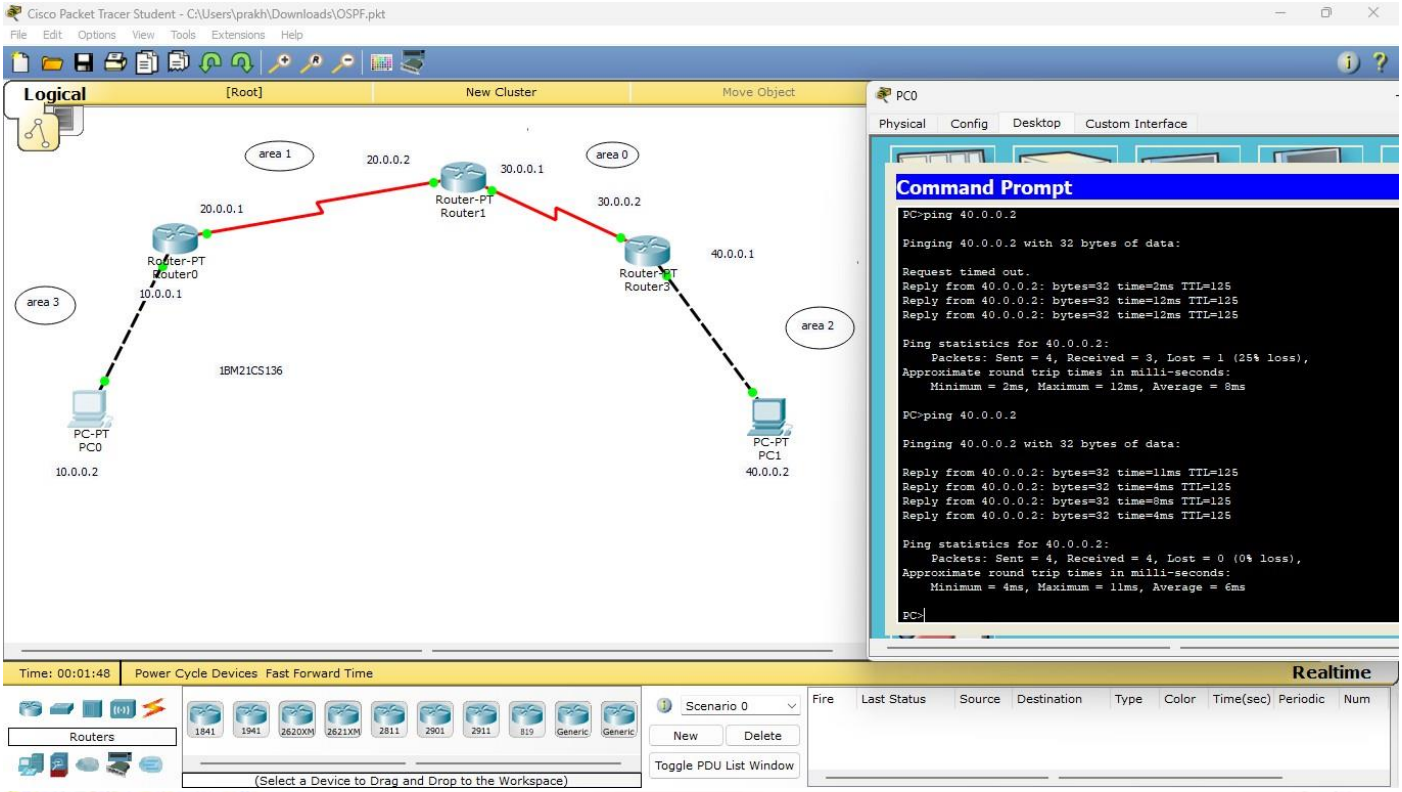
Reply from 40.0.0.10 bytes = 32 time = 9ms TTL=125
Reply from 40.0.0.10 bytes = 32 time = 6ms TTL=125
Reply from 40.0.0.10 bytes = 32 time = 8ms TTL=125
Reply from 40.0.0.10 bytes = 32 time = 7ms TTL=125

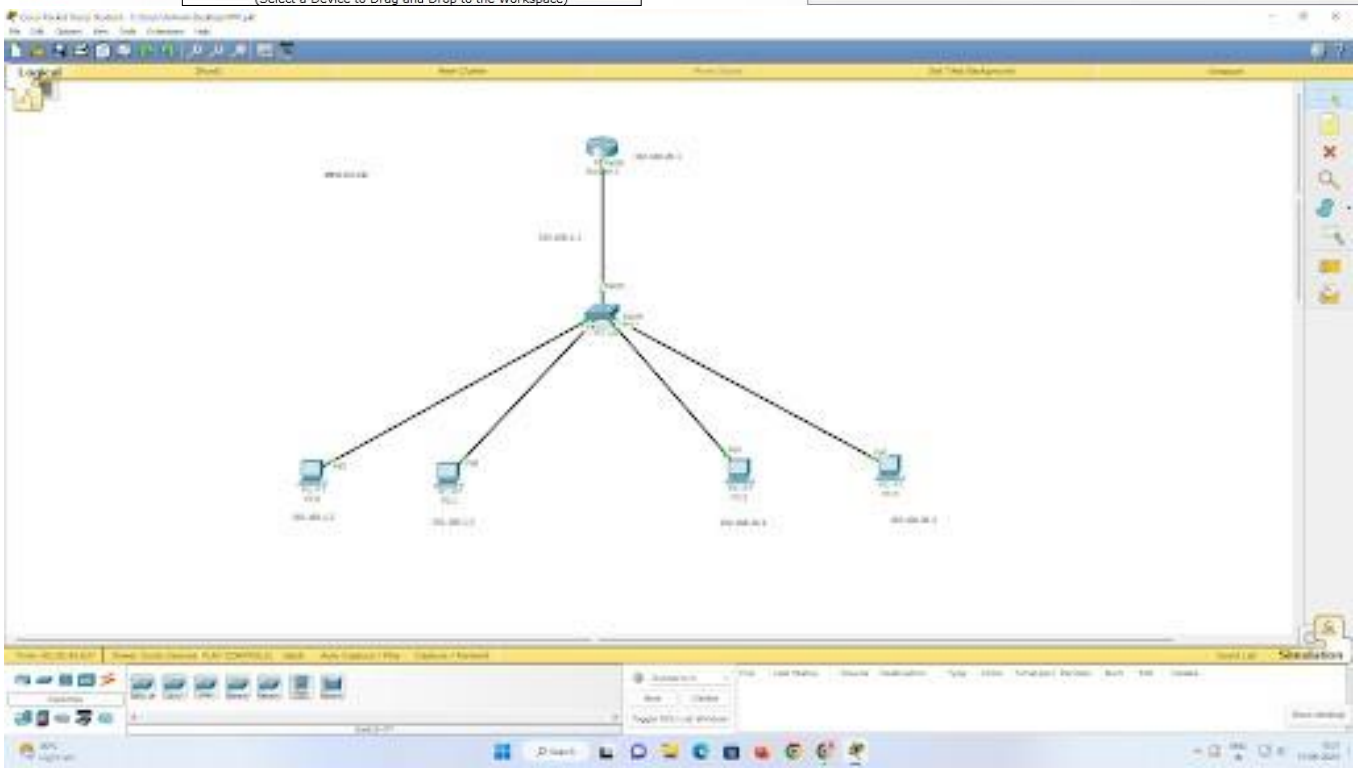
Ping statistics for 40.0.0.10:
Packets: Sent = 4, Received = 4, Lost = 0.

Outcome:

OSPF stands for Open Shortest Path First is a link layer/ application layer protocol which is used to find the best route for packet transfer and universes creating a virtual link.

System Output-





LAB 8

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

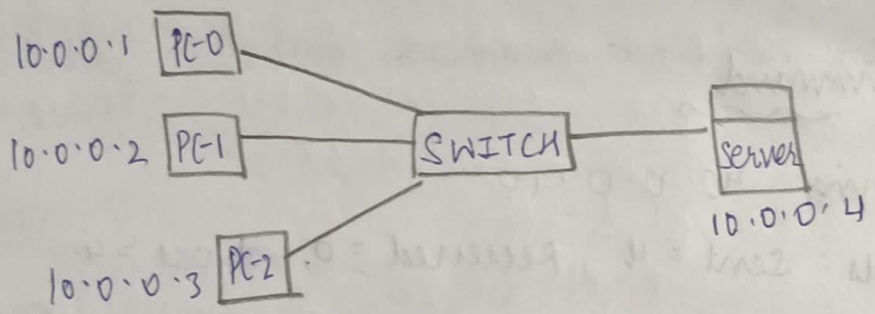
Observation-

11/08/2023

Lab - 07

Aim: To demonstrate ARP protocol in routers.

Topology:



Procedure:

1.) Create the topology as shown above with 3 PC's, 1 server and 1 switch.

2.) Assign ip address to all PC's and server in the network.

3.) GO to simulation mode and open the ARP tables for all 3 PC's and the server.

4.) Now send PDU from PC-0 to server via the switch and note the incoming and outgoing PDU details at the PC, switch and the server.

5. When the PDU reaches the server the ARP table of server is updated with the "mac" address of PC-0.

6. Now send the PDU back from server to the PC-0.

MAC address of the server.

8.) Similarly send PDU from all the PC's to the server and back in order to update the ARP table.

9.) Use the command "arp -a" in order to view the 'arp-table'.

CLI command & Observation

10.0.0.1

PC > arp -a.

Internet Address	Physical Address	Type
10.0.0.4	0001.0409.0936	Dynamic

10.0.0.4

PC > arp -a

Internet Address	Physical Address	Type
10.0.0.1	00e0.b04b.2ac8	Dynamic
10.0.0.2	0090.217e.17d6	Dynamic
10.0.0.3	0060.3e17.5ed6	Dynamic

Outcome:

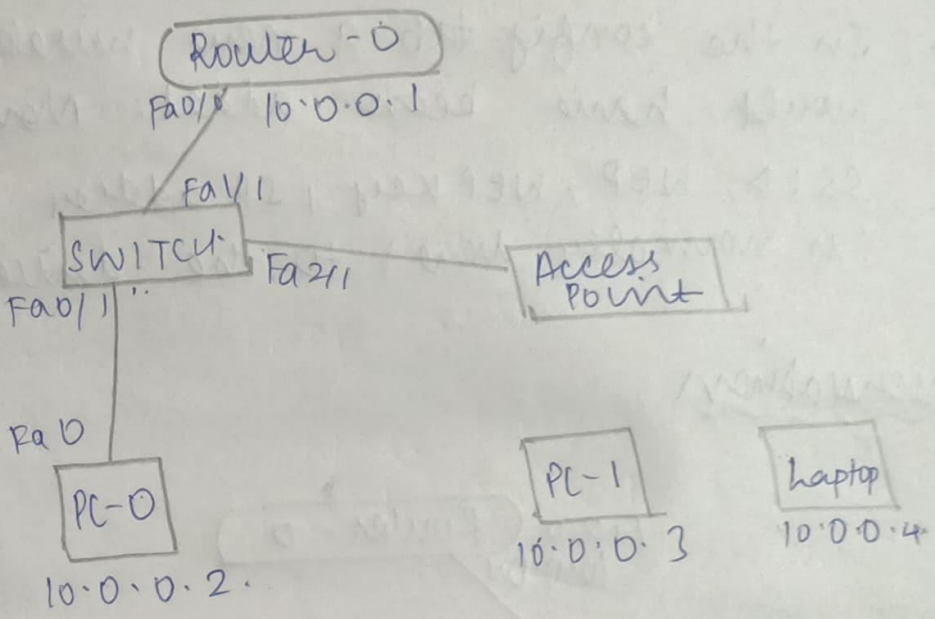
To learn about how the Address Resolution Protocol maps the IP address to the MAC address.

To construct a WLAN and make the nodes communicate wirelessly.

1) WLAN

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

Topology:



L=127

TTL=127

TTL=127

TTL=127

Procedure:

1.) Create the topology as shown above with PC's, switch, router, Access point and laptop.

2.) Configure PC0 and router.

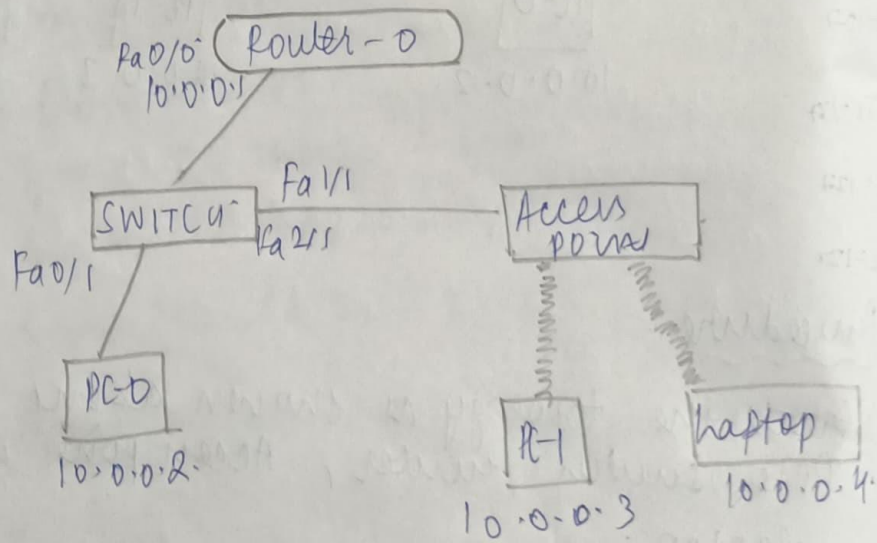
3.) Configure the Access point, go to port 1 and give the SSID name.

4.) Select WEP and give any 10 digit Hex key (1987654321O). Configure PC1 and laptop with wireless standards.

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

6.) In the config tab a new wireless interface would have been added. Now configure SSID, WEP, WEP key, IP address and Gateway (as normally done) to the device.

Observation

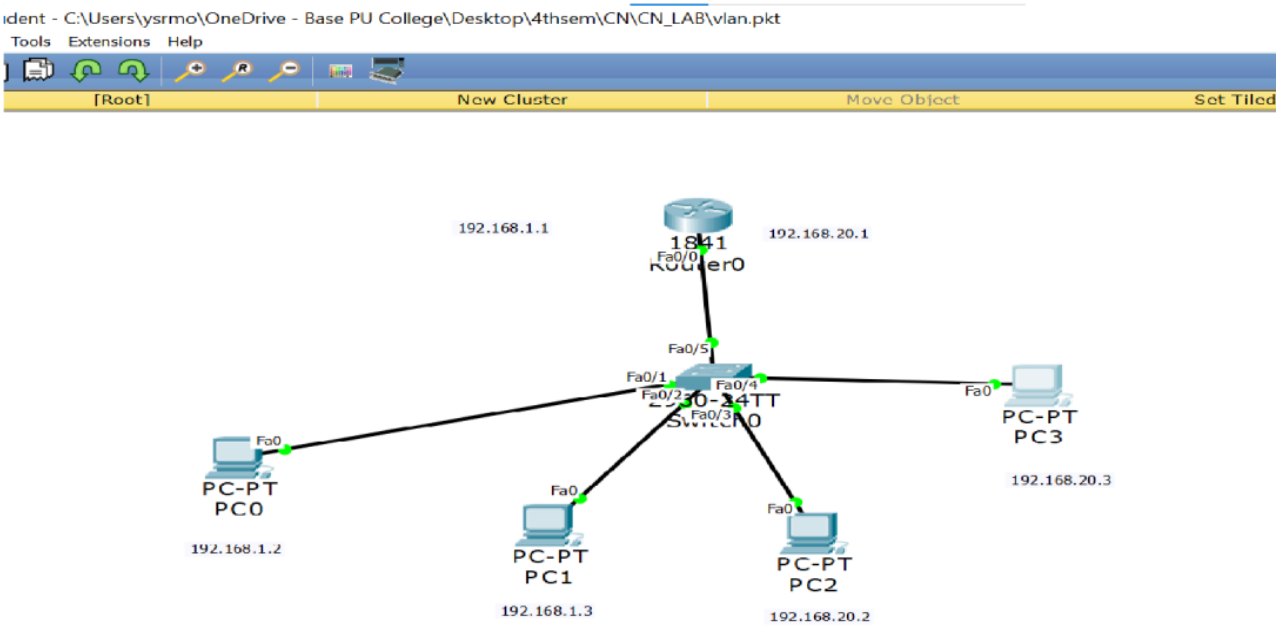


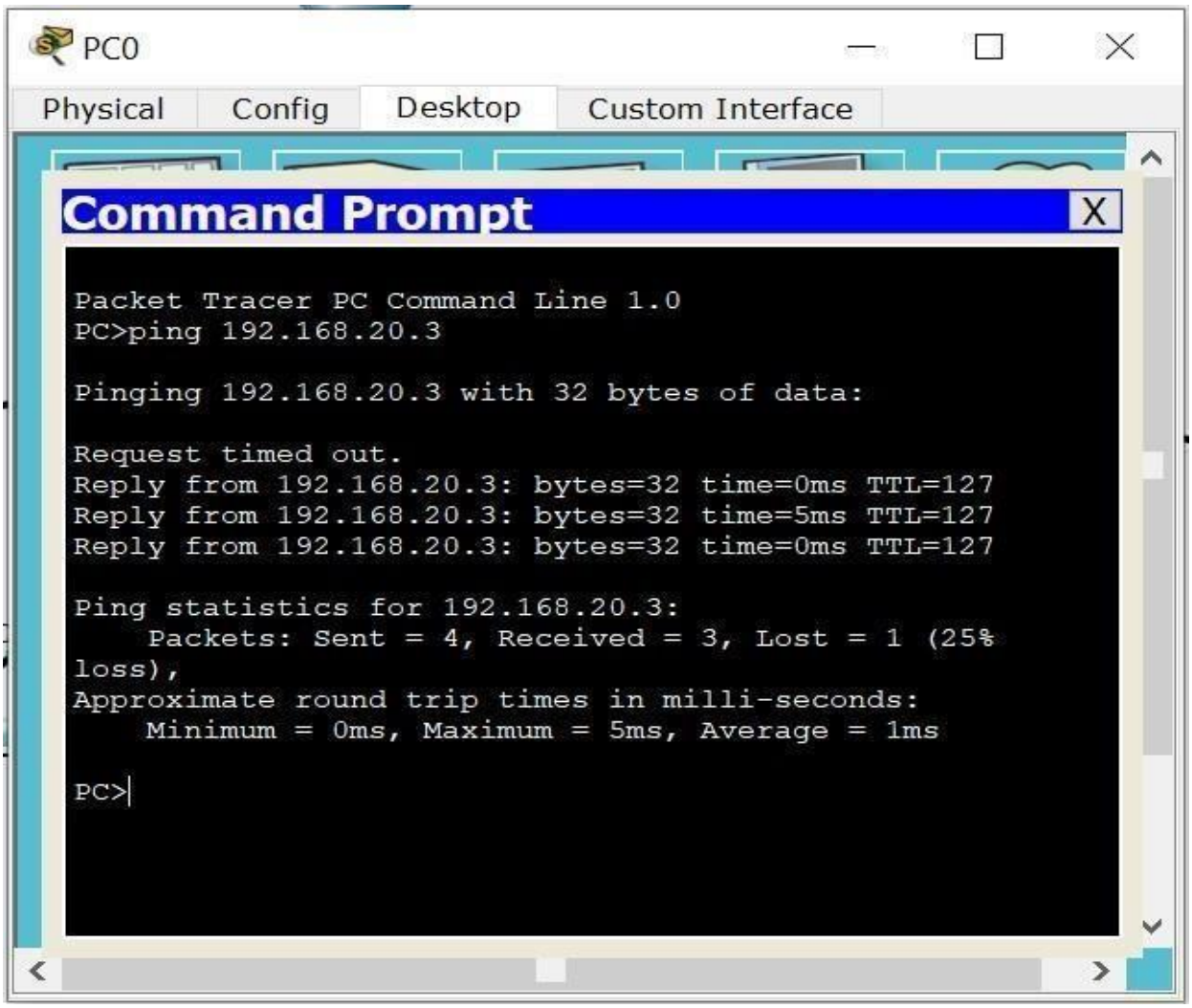
PC0

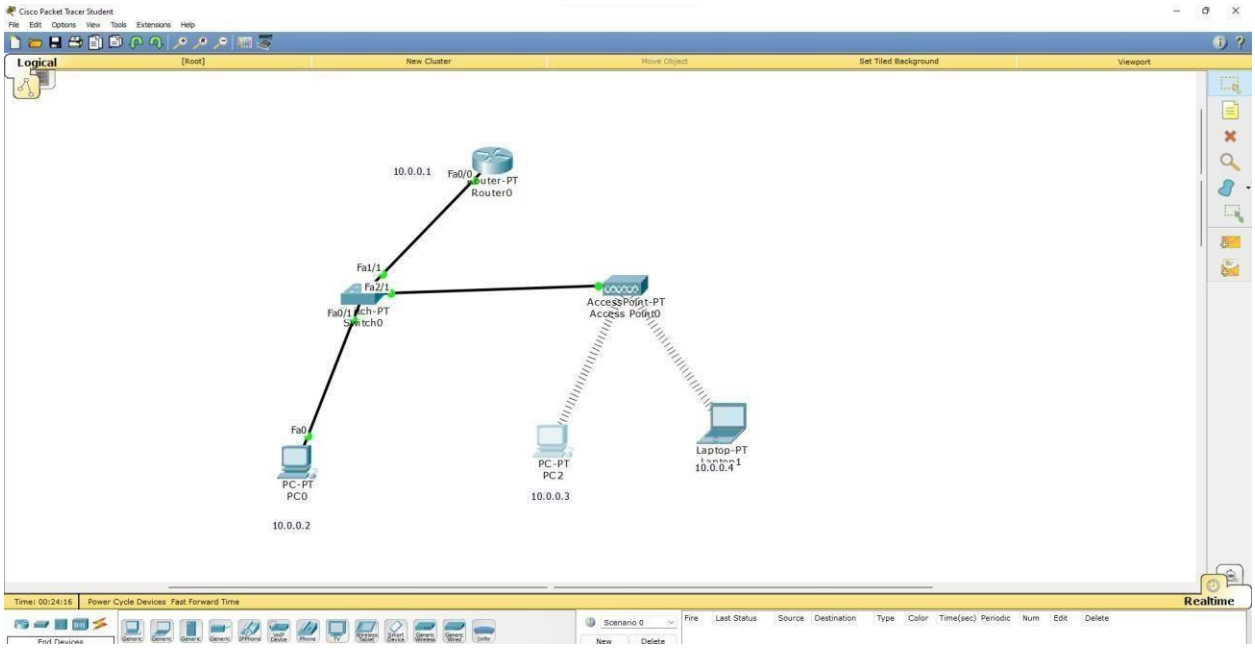
PC> ping 10.0.0.3

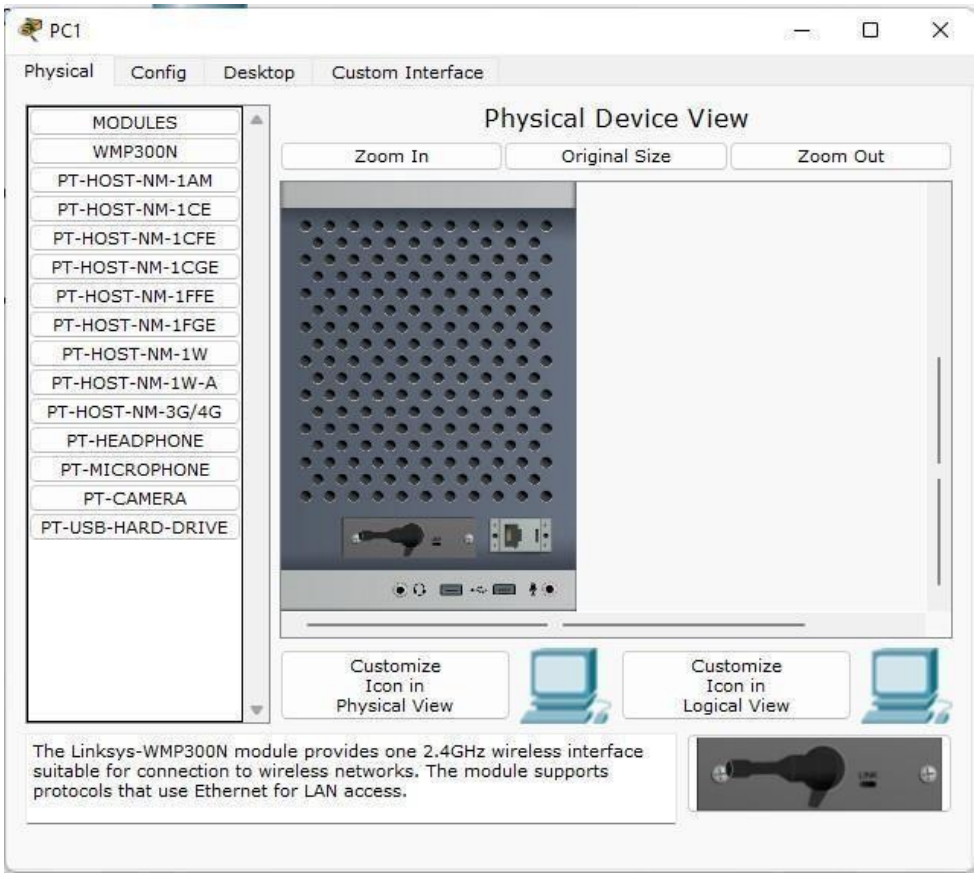
Pinging 10.0.0.3 with 32 bytes of data.
Reply from 10.0.0.3: bytes=32 time=4ms TTL=
Reply from 10.0.0.3: bytes=32 time=32ms TTL=
Reply from 10.0.0.3: bytes=32 time=35ms TTL=
Reply from 10.0.0.3, bytes=32 time=3ms TTL
Packets: Sent=4, Received=4 Lost=0

System Output-





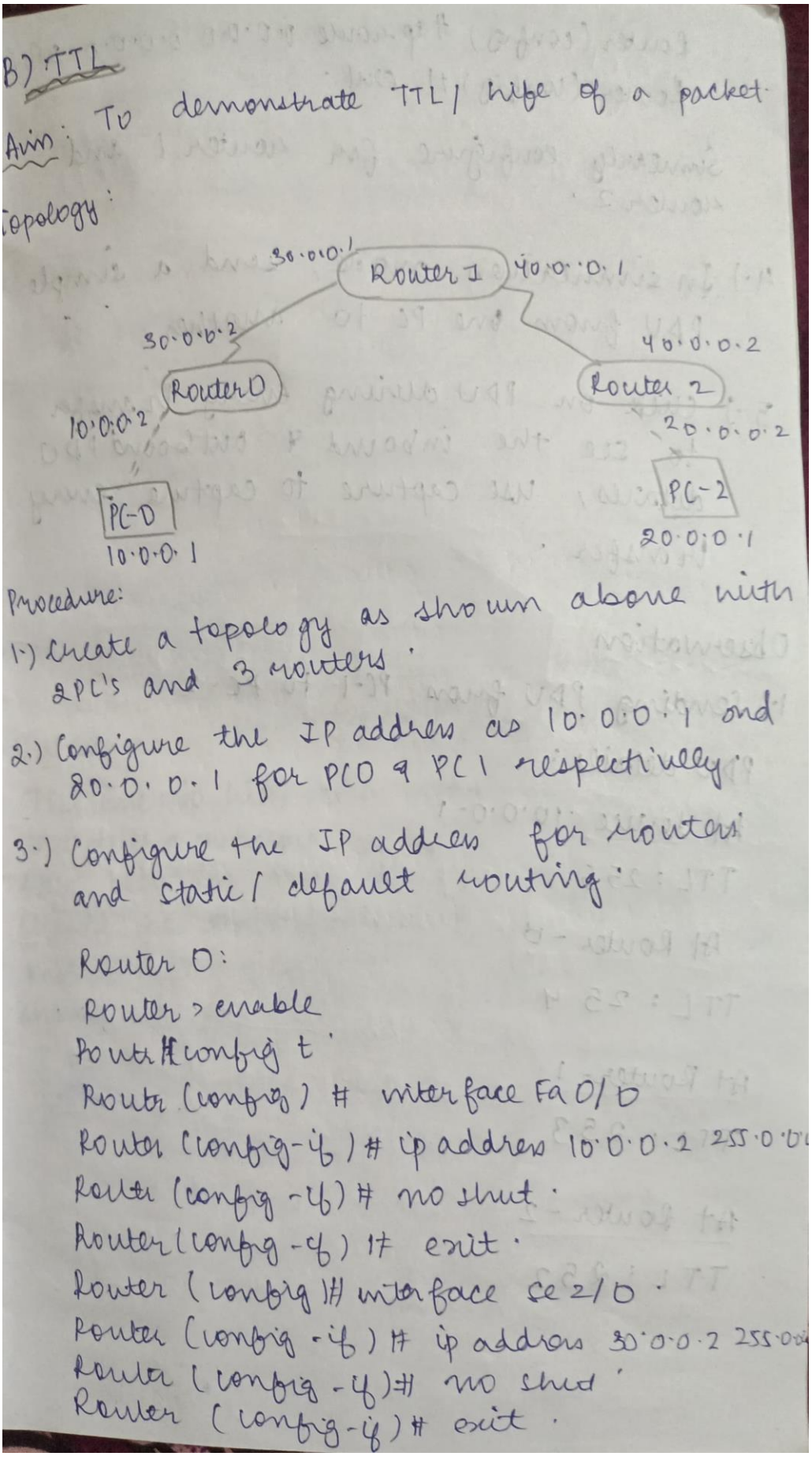

Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
10.0.0.1
Fa0/0
Router-PT
Router0
Fa1/1
Fa2/1
Fa0/1
Switch-PT
Switch0
Access Point-PT
Access Point0
Fa0
PC-PT
PC0
10.0.0.2
PC-PT
PC2
10.0.0.3
Laptop-PT
10.0.0.4
Time: 00:24:16
Power Cycle Devices Fast Forward Time
Realtime
Scenario 0
New
Delete
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Periodic
Num
Edit
Delete


PC1
Physical
Config
Desktop
Custom Interface
MODULES
WMP300N
PT-HOST-NM-1AM
PT-HOST-NM-1CE
PT-HOST-NM-1CFE
PT-HOST-NM-1CGE
PT-HOST-NM-1FFE
PT-HOST-NM-1FGE
PT-HOST-NM-1W
PT-HOST-NM-1W-A
PT-HOST-NM-3G/4G
PT-HEADPHONE
PT-MICROPHONE
PT-CAMERA
PT-USB-HARD-DRIVE
Physical Device View
Zoom In
Original Size
Zoom Out
Customize Icon in Physical View
Customize Icon in Logical View
The Linksys-WMP300N module provides one 2.4GHz wireless interface suitable for connection to wireless networks. The module supports protocols that use Ethernet for LAN access.

LAB 9

Demonstrate the TTL/ Life of a Packet.

Observation-



B) TTL

Aim: To demonstrate TTL / life of a packet.

Topology:

Procedure:

1.) Create a topology as shown above with 2 PC's and 3 routers.

2.) Configure the IP address as 10.0.0.1 and 20.0.0.1 for PC0 & PC1 respectively.

3.) Configure the IP address for routers and static / default routing.

Router 0:

Router > enable

Router # config t

Router (config) # interface Fa 0/0

Router (config-if) # ip address 10.0.0.2 255.0.0.

Router (config-if) # no shut.

Router (config-if) # exit.

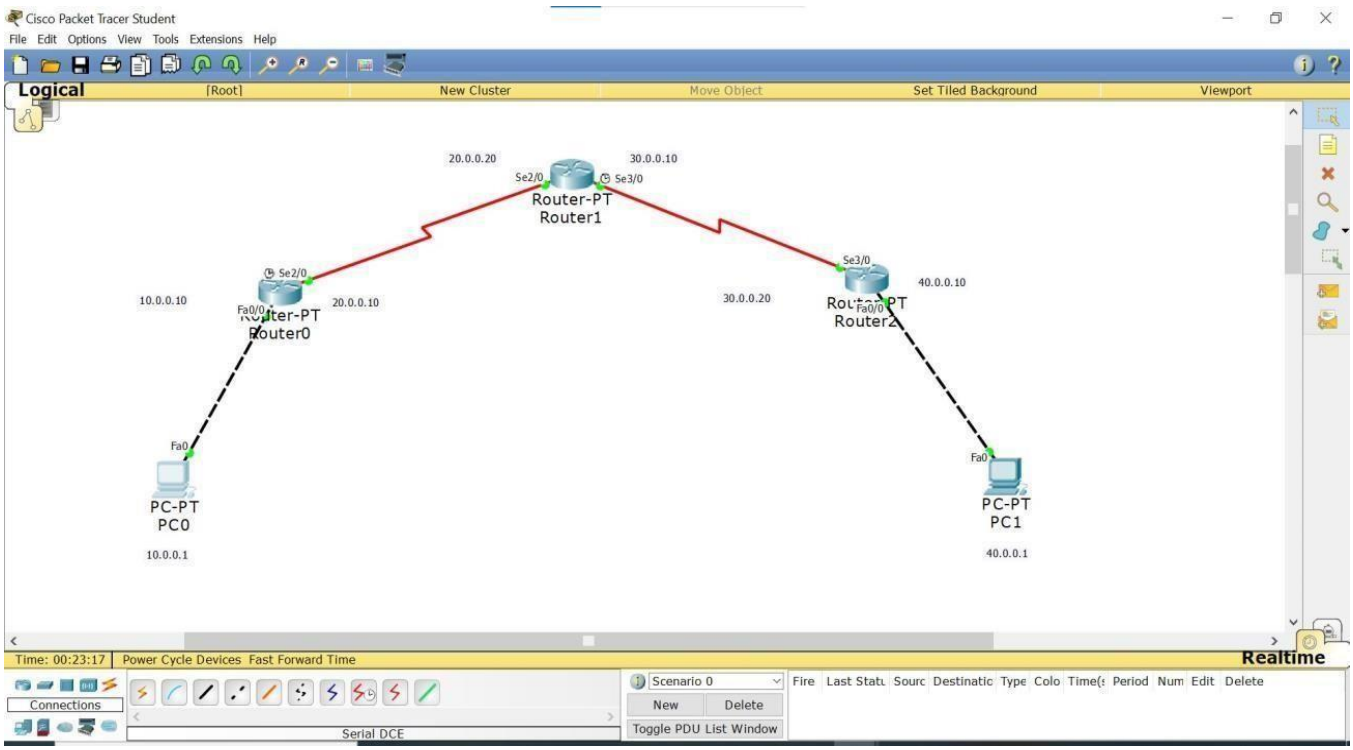
Router (config) # interface se 2/0.

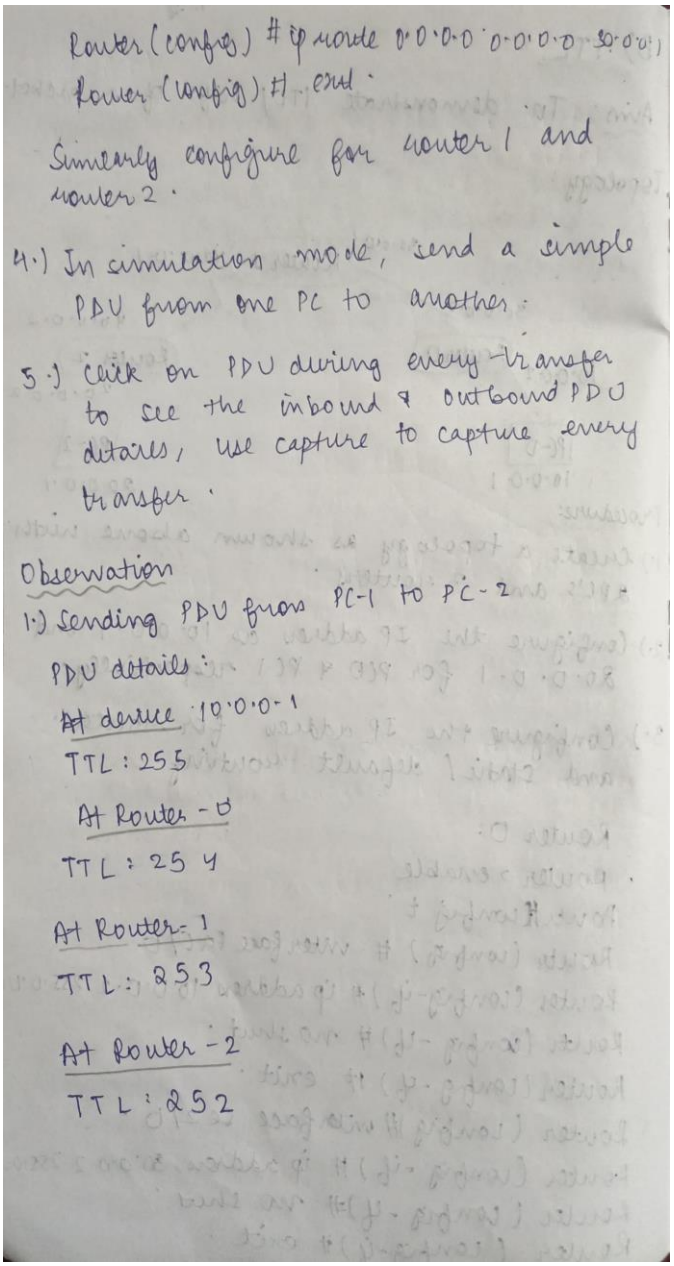
Router (config-if) # ip address 30.0.0.2 255.0.0.

Router (config-if) # no shut

Router (config-if) # exit.

System Output-





Router(config) # ip route 0.0.0.0 0.0.0.0 30.0.0.1
Router(config) # end

Similarly configure for router 1 and router 2.

4.) In simulation mode, send a simple PDU from one PC to another.

5.) Click on PDU during every transfer to see the inbound & outbound PDU details, use capture to capture every transfer.

Observation

1.) Sending PDU from PC-1 to PC-2

PDU details:

At device 10.0.0.1

TTL: 255

At Router-0

TTL: 254

At Router-1

TTL: 253

At Router-2

TTL: 252

LAB 10

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

Observation-

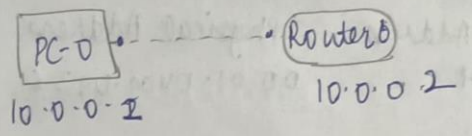
18/08/2023

Lab-08

A) Telnet

Aim: Create a topology in order to depict Telnet

Topology:



Procedure

1.) Create the topology as shown above with 1 PC and 1 router.

2.) In the CLI of the router process the following commands.

CLI Commands:

```
Router>enable
Router# config t
Router(config)# hostname R1
R1(config)# enable secret P1
R1(config)# interface Fa0/0
R1(config-if)# ip address 10.0.0.1 255.0.0.0
R1(config-if)# no shut
```

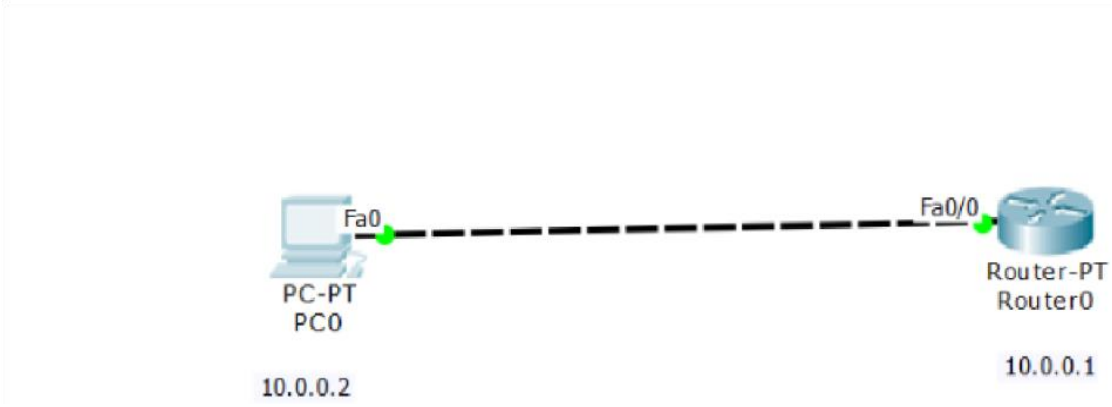
R1(config-vty) # line vty 0 5
R1(config-line) # login
% login disabled on line 132, until
'password' is set
R1(config-line) # password PO
R1(config-line) # exit
R1(config) # exit
R1 # wr

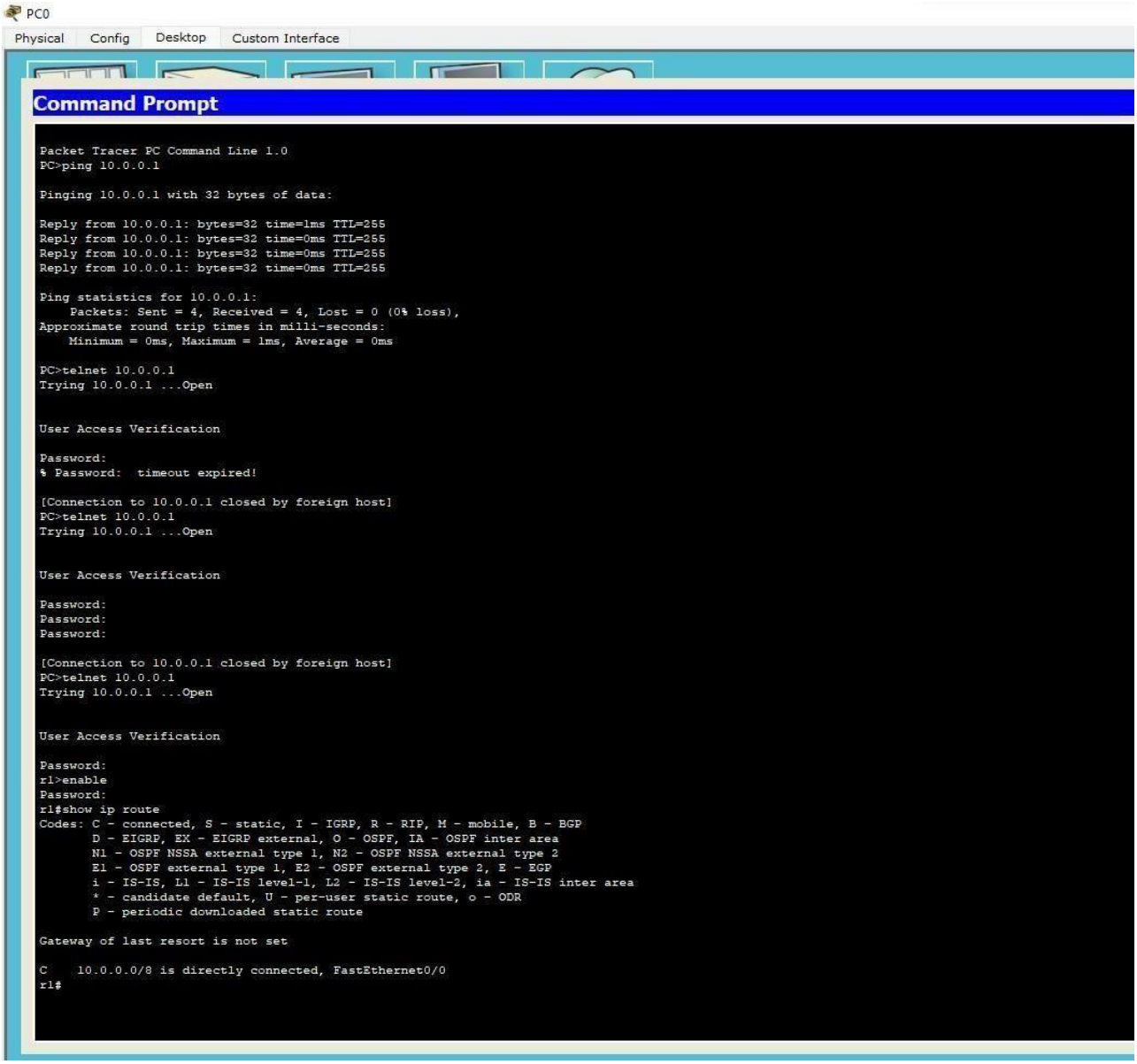
Observation

PC> ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1 bytes=32 time=0ms TTL=25
Reply from 10.0.0.1 bytes=32 time=0ms TTL=25
Reply from 10.0.0.1 bytes=32 time=0ms TTL=25
Reply from 10.0.0.1 bytes=32 time=0ms TTL=25
Ping statistics for 10.0.0.1
Packets: sent=4, Recieved=4, lost 0

PC> telnet 10.0.0.1
Trying 10.0.0.1 ... Open
User Access Verification
Password:
R1>enable
Password:
R1# show ip route
Codes: C
D
N1
E1

System Output-

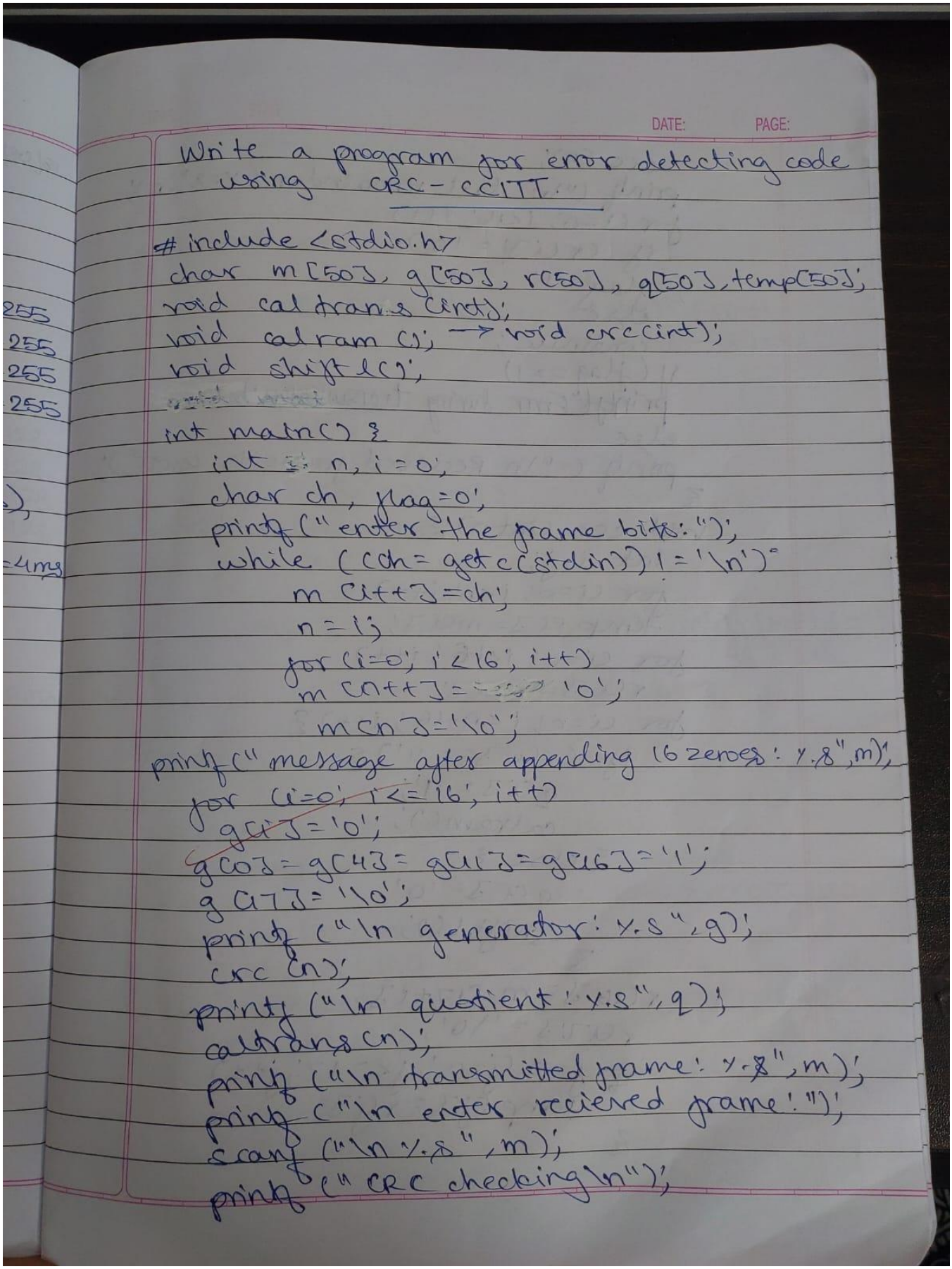




LAB 11

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

Observation-



DATE: PAGE:

Write a program for error detecting code using CRC - CCITT.

```
#include<stdio.h>
char m[50], g[50], r[50], q[50], temp[50];
void caltrans(int);
void calram();  -> void crc(int);
void shift1();

int main(){
    int n, i=0;
    char ch, flag=0;
    printf("enter the frame bits:");
    while((ch=getc(stdin))!='\n')
        m[i++]=ch;
        n=i;
        for(i=0; i<16; i++)
        m[n++]='0';
        m[n]='\0';
printf("message after appending 16 zeroes: %s",m);
    for(i=0; i<=16; i++)
      g[i]='0';
g[0]=g[4]=g[11]=g[16]='1';
g[17]='\0';
printf("\n generator: %s",g);
crc(n);
printf("\n quotient: %s",q);
caltrans(n);
printf("\n transmitted frame: %s",m);
printf("\n enter recieved frame:");
scanf("\n%s",m);
printf("CRC checking\n");
```

```
    crc(n);
    printf("\n last remainder : %.8s", r);
    for(i=0; i<16; i++)
      if(r[i] != '0')
      flag = 1;
      else
      continue;
    if(flag == 1)
    printf("Error during transmission");
    else
    printf("\n recieved frame is correct");
}
void crc(int n) {
    int i, j;
    for(i=0; i<n; i++)
    temp[i] = m[i];
  for(i=0; i<16; i++)
    r[i] = m[i];
  for(i=0; i<n-16; i++) {
      if(r[0] == '1') {
          q[i] = '1';
          caltran();
      } else {
          q[i] = '0';
          shiftl();
      }
  r[16] = m[17+i];
  r[17] = '\0';
  for(j=0; j<=17; j++)
    temp[j] = r[j];
}
```

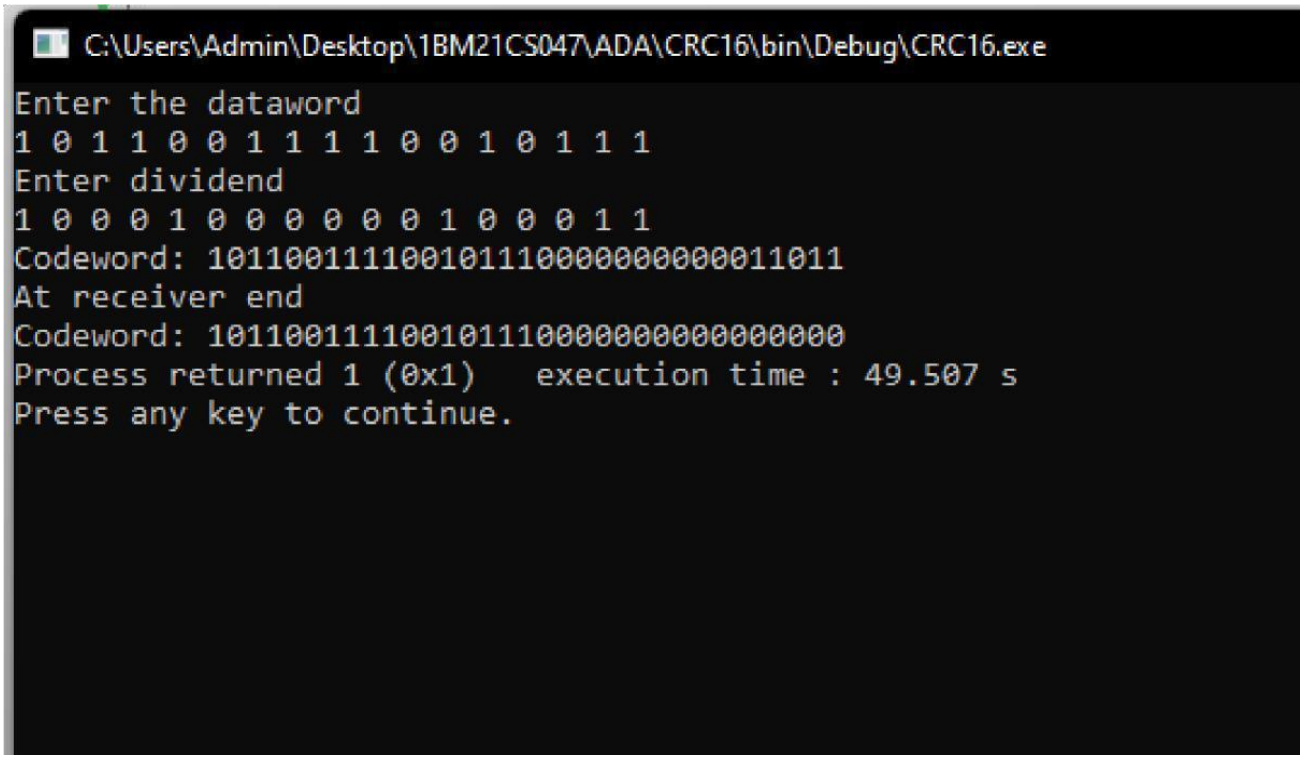
DATE: PAGE:

```
    g[n-16]='1';
}
void calram(){
    int i,j;
    for(i=1;i<=16;i++)
    r[i-1]=((int)temp[i]-48)^((int)
                g[i]-48)+48;
}
void shift(){
    int i;
    for(i=1;i<=16;i++)
    r[i-1]=r[i];}
void caltrans(int n){
    int i,k=0;
    for(i=n-16;i<n;i++)
    m[i]=((int)m[i]-48)^((int)r[k++]-
                48)+48;
    m[i]='0'; return 0;}
```

OUTPUT-

```
enter the frame bits: 1011
message after appending 16 zeroes:
                1011000000000000
generator: 1000100000010001
quotient: 1011
transmitted frame: 1011011000001011
crc checking
last mechanism: 0000000000000000
recieved frame is correct.
```

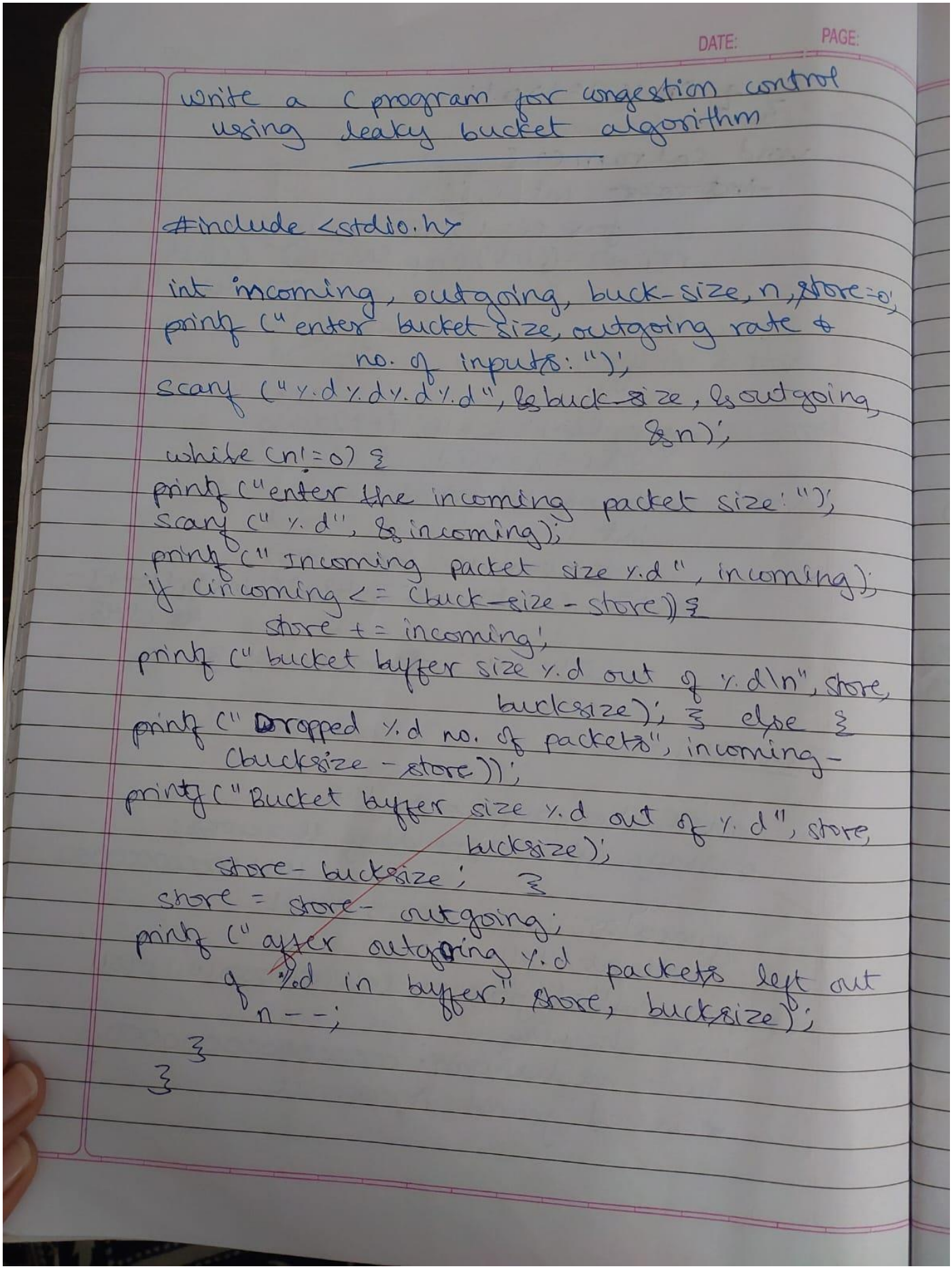
System Output-



LAB 12

Write a program for congestion control using Leaky bucket algorithm.

Observation-



write a C program for congestion control using leaky bucket algorithm

```
#include <stdio.h>

int incoming, outgoing, buck_size, n, store=0;
printf("enter bucket size, outgoing rate & no. of inputs: ");
scanf("%d %d %d %d", &buck_size, &outgoing, &n);
while(n!=0) {
printf("enter the incoming packet size: ");
scanf("%d", &incoming);
printf("Incoming packet size %d", incoming);
if(incoming <= (buck_size - store)) {
        store += incoming;
printf("bucket buffer size %d out of %d\n", store, bucksize); } else {
printf("Dropped %d no. of packets", incoming-(bucksize - store));
printf("Bucket buffer size %d out of %d", store, bucksize);
    store = bucksize; }
store = store - outgoing;
printf("after outgoing %d packets left out of %d in buffer", store, bucksize);
    n--;
}
}
```

DATE: PAGE:

- OUTPUT -

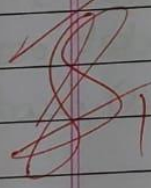
enter bucket size, outgoing rate & no. of inputs: 20 10 2

→ enter the incoming packet size: 30

⇒ incoming packet size: 30
dropped 10 no. of packets
buffer size 0 out of 20
after outgoing 10 packets left out of 20 in buffer

→ enter the incoming packet size: 10

⇒ incoming packet size: 10
buffer size 20 out of 20
after outgoing 10 packets out of 20 in buffer.



System Output-

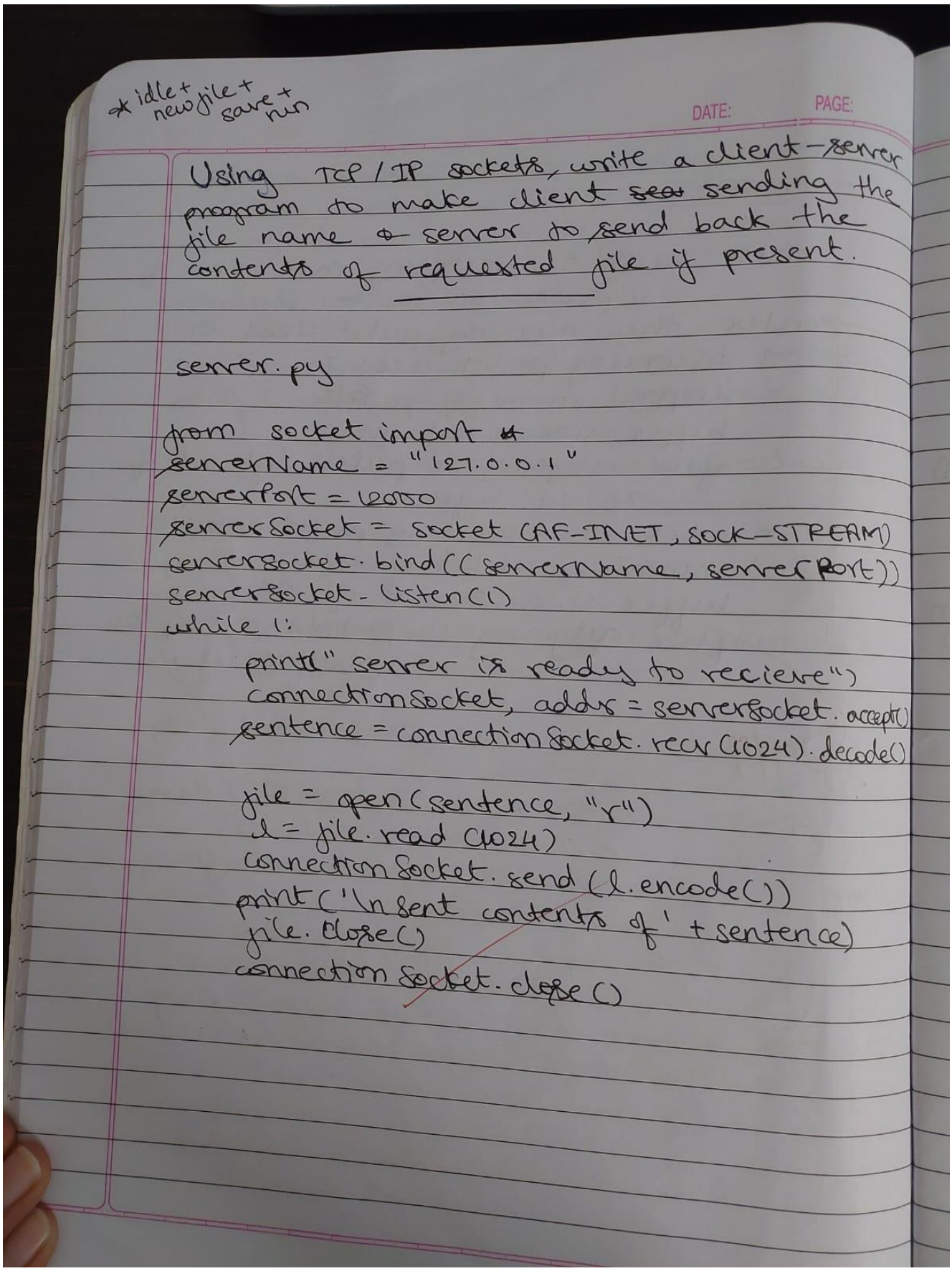
```
PS D:\VS Code> cd "d:\VS Code\OS\" ; if ($?) { gcc bucket.c -o bucket } ; if ($?) { .\bucket }
Enter Bucket size and outstream size
2000
100
Packet of 41 bytes accepted
Remaining bytes: 2000
If you want to stop input, press 0, otherwise, press 1
1
Packet of 467 bytes accepted
Remaining bytes: 1633
If you want to stop input, press 0, otherwise, press 1
1
Packet of 334 bytes accepted
Remaining bytes: 1399
If you want to stop input, press 0, otherwise, press 1
1
Packet of 500 bytes accepted
Remaining bytes: 999
If you want to stop input, press 0, otherwise, press 1
1
Packet of 169 bytes accepted
Remaining bytes: 930
If you want to stop input, press 0, otherwise, press 1
1
Packet of 724 bytes accepted
Remaining bytes: 306
If you want to stop input, press 0, otherwise, press 1
1
Packet of 478 bytes is discarded
Remaining bytes: 406
If you want to stop input, press 0, otherwise, press 1
1
Packet of 358 bytes accepted
Remaining bytes: 148
If you want to stop input, press 0, otherwise, press 1
1
Packet of 962 bytes is discarded
Remaining bytes: 248
If you want to stop input, press 0, otherwise, press 1
0
Remaining bytes: 348
Remaining bytes: 448
Remaining bytes: 548
Remaining bytes: 648
Remaining bytes: 748
```

```
Remaining bytes: 348
Remaining bytes: 448
Remaining bytes: 548
Remaining bytes: 648
Remaining bytes: 748
Remaining bytes: 848
Remaining bytes: 948
Remaining bytes: 1048
Remaining bytes: 1148
Remaining bytes: 1248
Remaining bytes: 1348
Remaining bytes: 1448
Remaining bytes: 1548
Remaining bytes: 1648
Remaining bytes: 1748
Remaining bytes: 1848
Remaining bytes: 1948
Remaining bytes: 2000
PS D:\VS Code\OS>
```

LAB 13

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.

Observation-



* idle + new file + save + run

DATE: PAGE:

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

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)
while 1:
    print(" server is ready to recieve")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()

    file = open(sentence, "r")
    l = file.read(1024)
    connectionSocket.send(l.encode())
    print('\n sent contents of' + sentence)
    file.close()
    connectionSocket.close()
```

DATE: PAGE:

client.py

```
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()
```

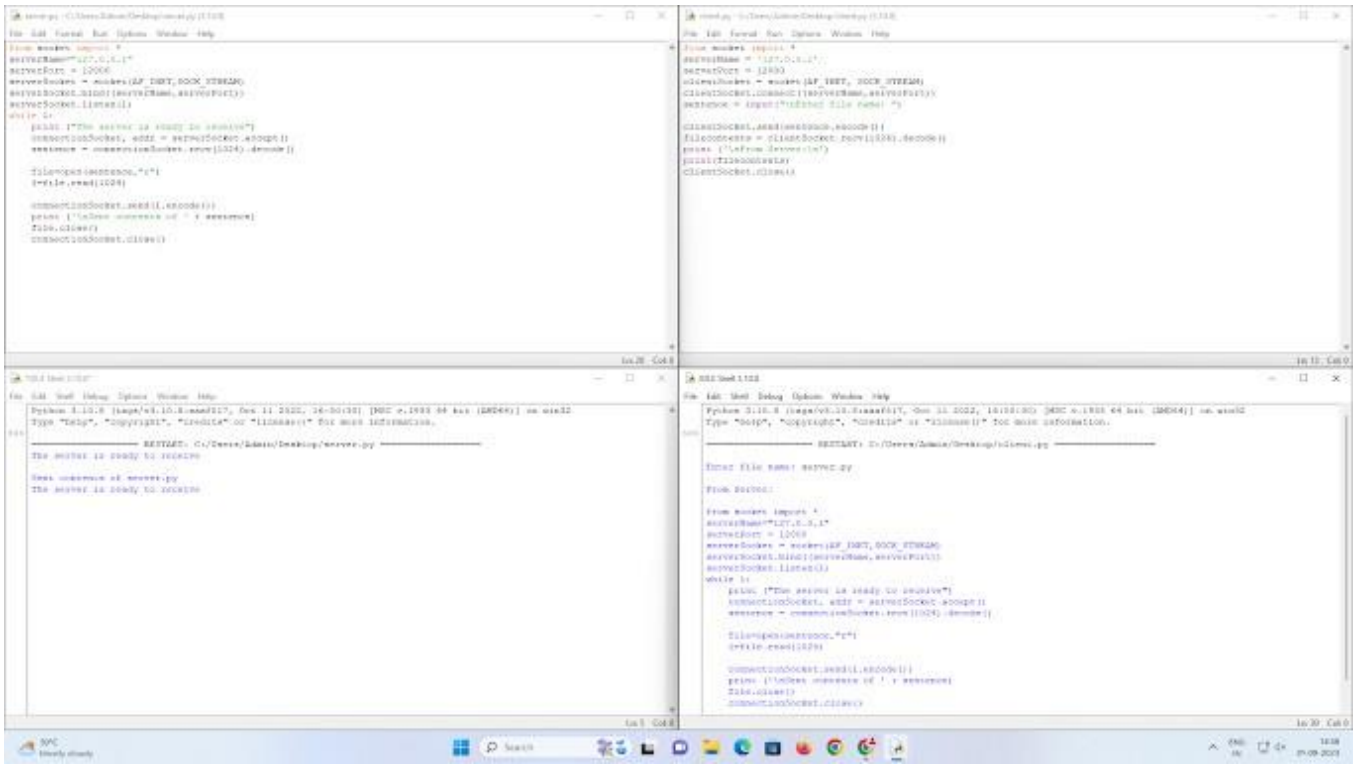
-OUTPUT-

1) server is ready to recieve
sent contents to server.py
server is ready to recieve.

2) enter file name: server.py
~~from server:~~
~~from socket import *~~
~~serverName = "127.0.0.1"~~
~~serverPort = 12000~~
~~serverSocket = socket(AF_IN~~

(contents of server.py, o/p).

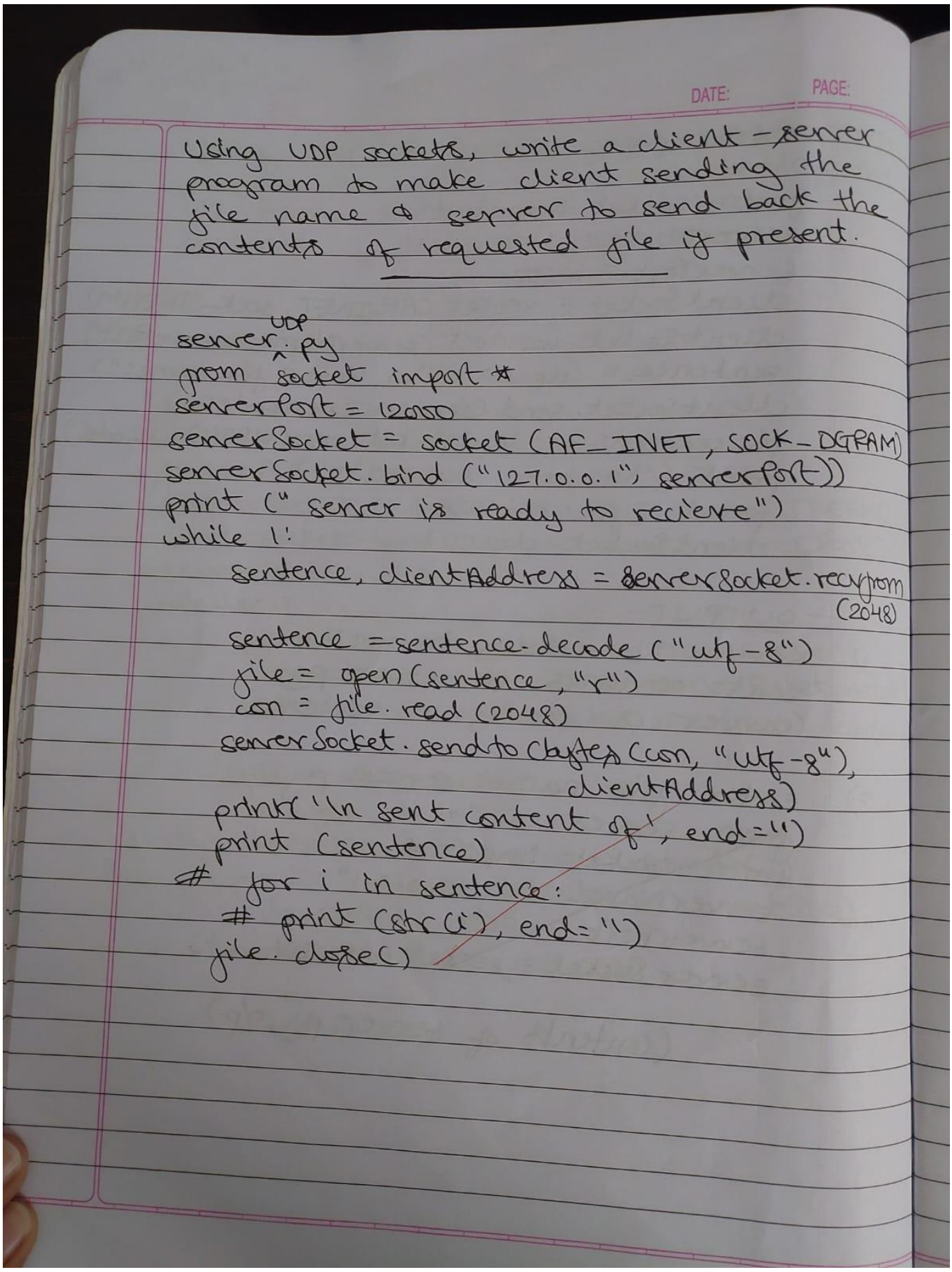
System Output-



LAB 14

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.

Observation-



DATE: PAGE:

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

server.py (UDP)

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print("server is ready to recieve")
while 1:
    sentence, clientAddress = serverSocket.recvfrom
                                        (2048)
    sentence = sentence.decode("utf-8")
    file = open(sentence, "r")
    con = file.read(2048)
    serverSocket.sendto(bytes(con, "utf-8"),
                        clientAddress)
    print('\n sent content of ', end='')
    print(sentence)
#   for i in sentence:
    #   print(str(i), end='')
    file.close()
```

DATE: PAGE:

clientUDP.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("\n enter file name:")
clientSocket.sendto(bytes(sentence, "utf-8"),
                    (serverName, serverPort))
filecontents, serverAddress = clientSocket.
print("\n Reply fro                    recvfrom(2048)
print('\n Reply from Server:\n')
print(filecontents.decode("utf-8"))
# for i in filecontents:
    # print(str(i), end="")
clientSocket.close()
clientSocket.close()
```

- OUTPUT -

1) Server is ready to recieve
   Sent contents of serverUDP.py

2) enter file name: serverUDP.py
   (contents of serverUDP.py O/P)

System Output-