

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**  
“JnanaSangama”, Belgaum -590014, Karnataka.



**LAB REPORT**  
on  
**COMPUTER NETWORKS**

*Submitted by*

**SREEJA KATTA (1BM21CS267)**

*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**  
**JUN-2023 to SEP-2023**

**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 NETWORKS” carried out by **SREEJA KATTA (1BM21CS267)** who is a 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.

**Dr. Latha N.R.**  
Assistant Professor  
Department of CSE  
BMSCE, Bengaluru

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

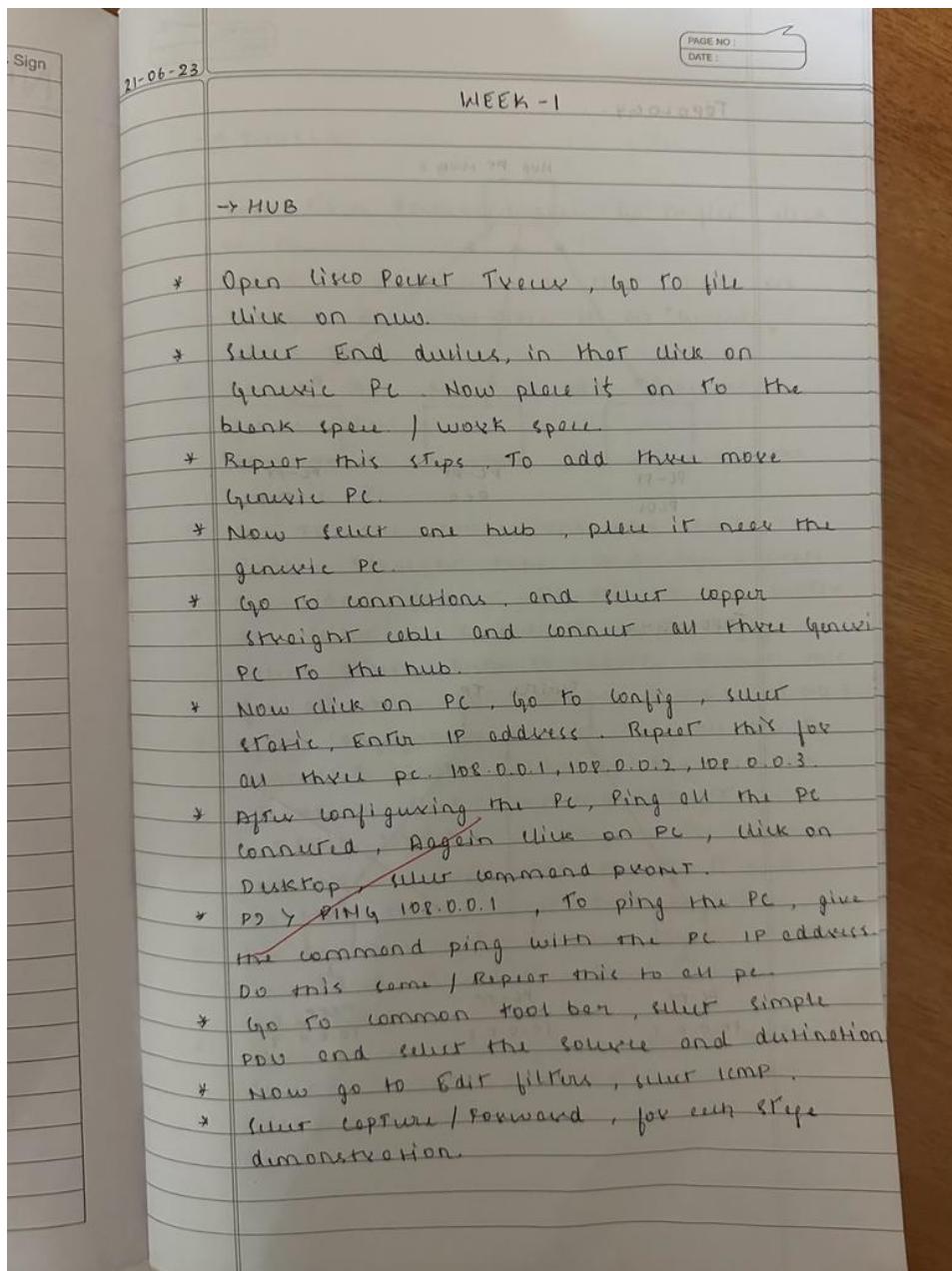
# Index

<b>Sl. No.</b>	<b>Date</b>	<b>Experiment Title</b>	<b>Page No.</b>
<b>CYCLE 1</b>			
1	15/6/23	Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrating ping messages.	4
2	22/6/23	Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply.	9
3	13/7/23	Configure default route, static route to the Router.	18
4	13/7/23	Configure DHCP within a LAN and outside LAN.	23
5	20/7/23	Configure Web Server, DNS within a LAN.	32
6	20/7/23	Configure RIP routing Protocol in Routers.	35
7	27/7/23	Configure OSPF routing protocol.	40
8	3/8/23	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP).	45
9	10/8/23	To construct a VLAN and make a pc communicate among VLAN.	49
10	10/8/23	Demonstrate the TTL/ Life of a Packet.	53
11	10/8/23	To construct a WLAN and make the nodes communicate wirelessly.	58
12	10/8/23	To understand the operation of TELNET by accessing the router in server room from a PC in IT office.	62
<b>CYCLE 2</b>			
13	17/8/23	Write a program for error detecting code using CRC CCITT (16-bits).	66
14	17/8/23	Write a program for congestion control using Leaky bucket algorithm.	72
15	24/8/23	Using TCP/IP 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.	76
16	24/8/23	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.	80
17	24/8/23	Tool Exploration -Wireshark	84

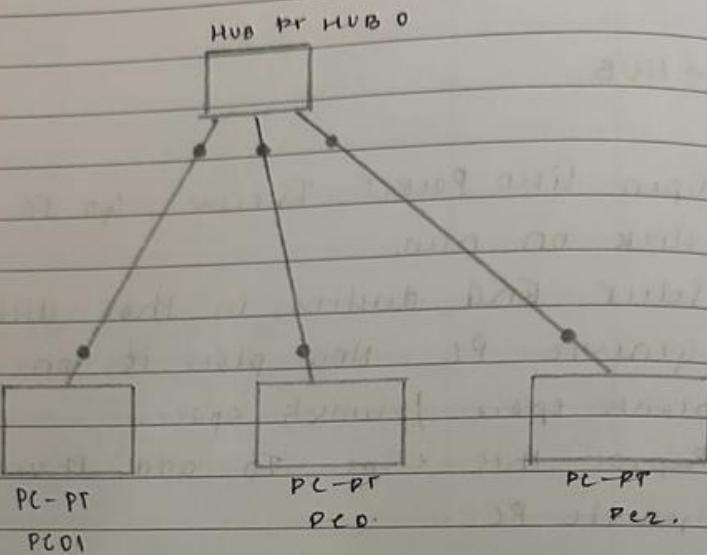
# WEEK 1

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

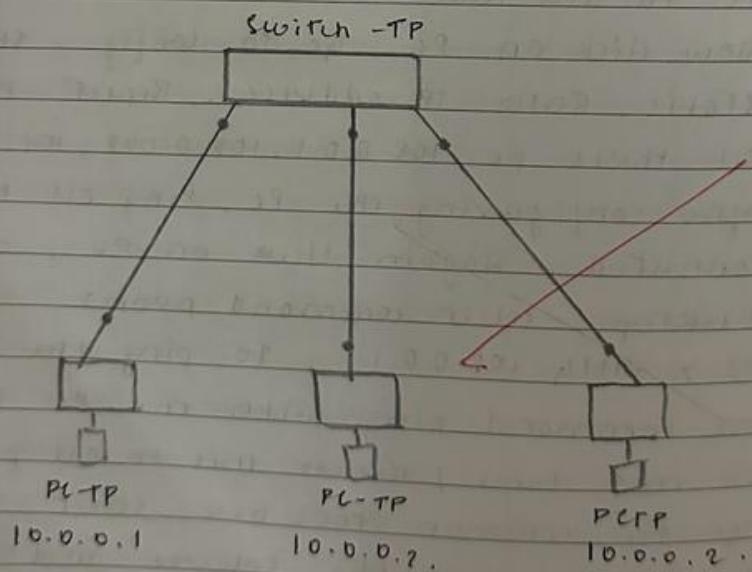
## OBSERVATION:



### TOPOLOGY.



### TOPOLOGY.



→ SWITCH

- \* Open Cisco Packet Tracer, go to file, click on new.
- \* Select End devices, In that click on generic PC. Now place it on "generic, work space".
- \* Repeat this steps, To add three more generic PC.
- \* Now select one hub, place it near the generic PC.
- \* Go to the connections, and select copper straight cable and copper cross over and connect all the pc and switch.
- \* Now click on generic PC, go to config, select static. Enter IP address, Repeat this for all three PC, 109.0.0.1, 109.0.0.2, 109.0.0.3
- \* After configuring the PC, ping all the PC connected, again click on PC, click on desktop, enter command prompt.
- \* PC> PING 109.0.0.1, To ping the PC, give the command ping with the PC IP address. Do this same) Repeat this to all pc.
- \* Go to common tool bar, select simple PDU and select the source and destination.
- \* Now go to Edit filters, select ICMP.
- \* Select capture | forward, for each stage demonstration.

- connecting two hubs with a switch.
- \* Double click on the Cisco packet tracer icon.
  - \* Build the network on the workspace.
  - \* Click on the end connection, select the required number of generic computer to build hub networks.
  - \* Give the IP address to each, Repeat this for all three 3 pc, 109.0.0.1, 109.0.0.2.
  - \* After configuring the pc, ping on the pc configured, again click on pc, click on desktop, enter command prompt
  - \* PC > PING 109.0.0.1, To ping the pc give command pc IP address. Do this same to all pc.

Transfer between PC<sub>0</sub> → PC<sub>3</sub>

PC<sub>0</sub> → hub.

hub → pc, & switch

PC<sub>1</sub> discards,

switch → hub

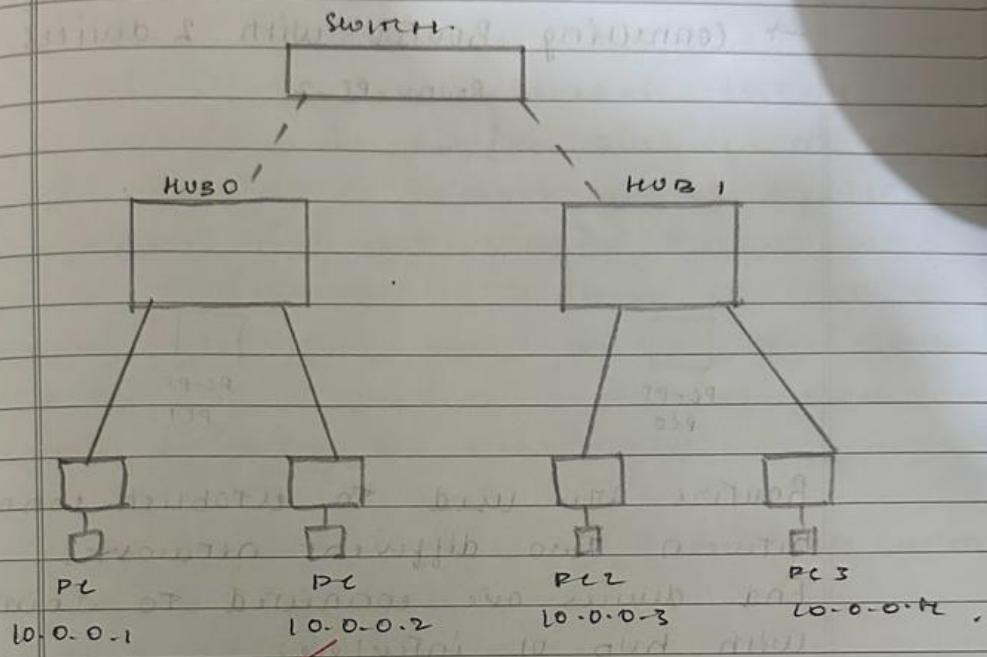
hub<sub>1</sub> → PC<sub>2</sub> & PC<sub>3</sub>

PC<sub>2</sub> discards, PC<sub>3</sub> finds out → hub<sub>1</sub>

hub<sub>1</sub> → PC<sub>1</sub> & switch.

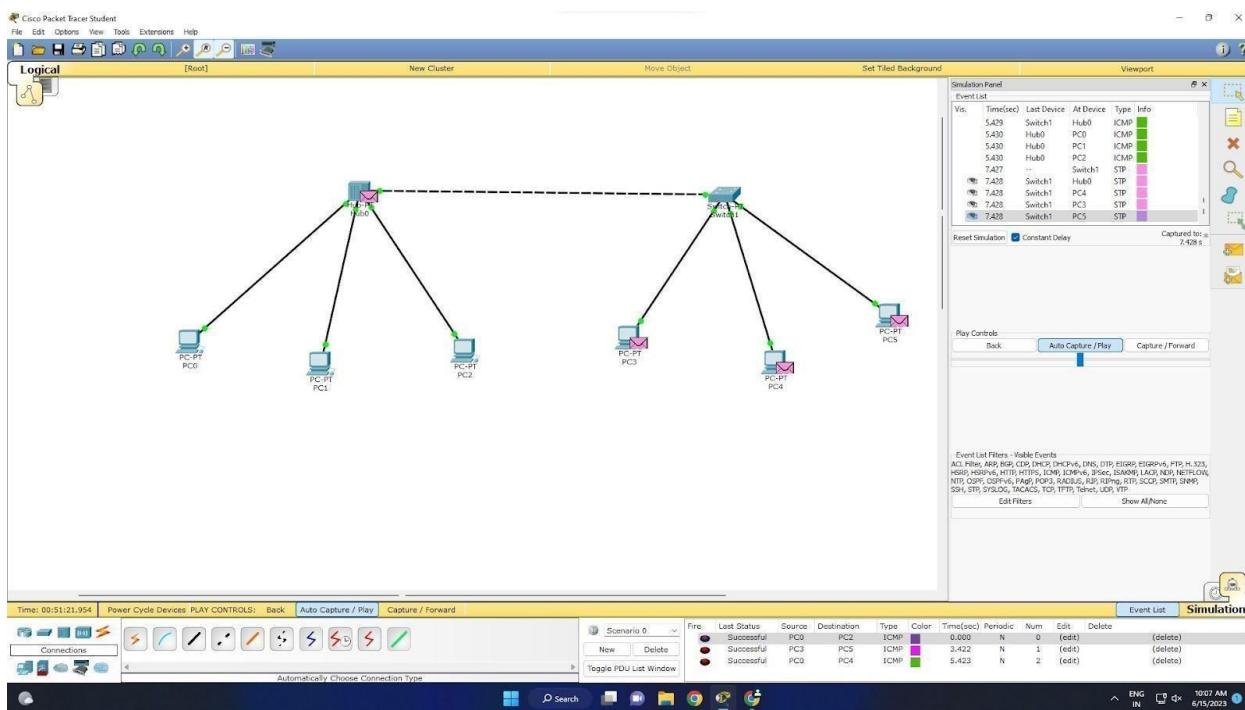
switch<sub>1</sub> → hub<sub>0</sub>

hub<sub>0</sub> → PC<sub>0</sub> & PC<sub>1</sub>



Wu

## TOPOLOGY:

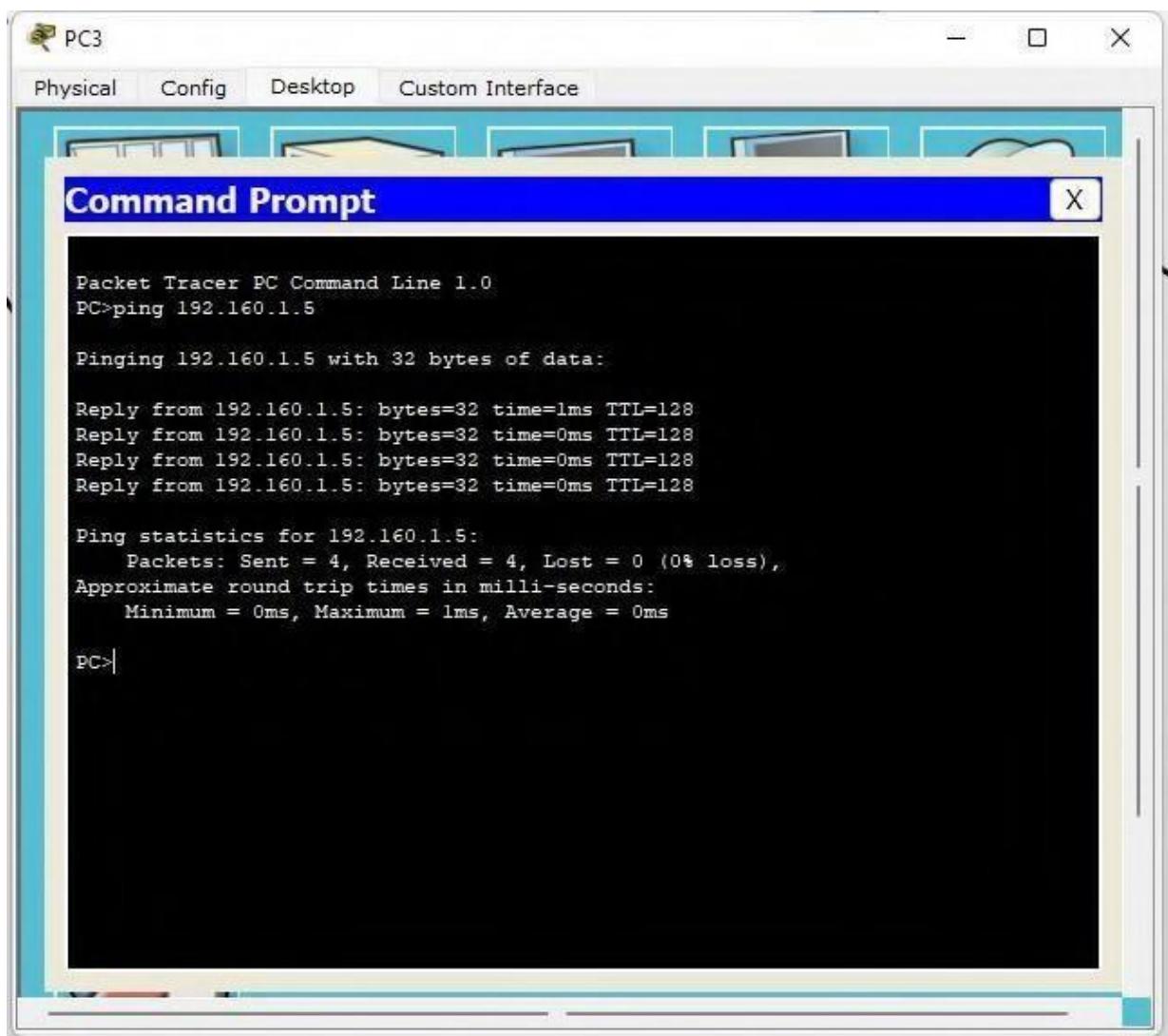


## OUTPUT:

```

PC0 Physical Config Desktop Custom Interface
Physical Config Desktop Custom Interface
Command Prompt
Packet Tracer FC Command Line 1.0
Pinging 192.160.1.5 with 32 bytes of data:
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
Ping statistics for 192.160.1.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
Pinging 192.160.1.5
Pinging 192.160.1.5 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 192.160.1.5:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>192.160.1.5
Invalid Command.
Pinging 192.160.1.2
Pinging 192.160.1.2 with 32 bytes of data:
Reply from 192.160.1.2: bytes=32 time=0ms TTL=128
Ping statistics for 192.160.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>

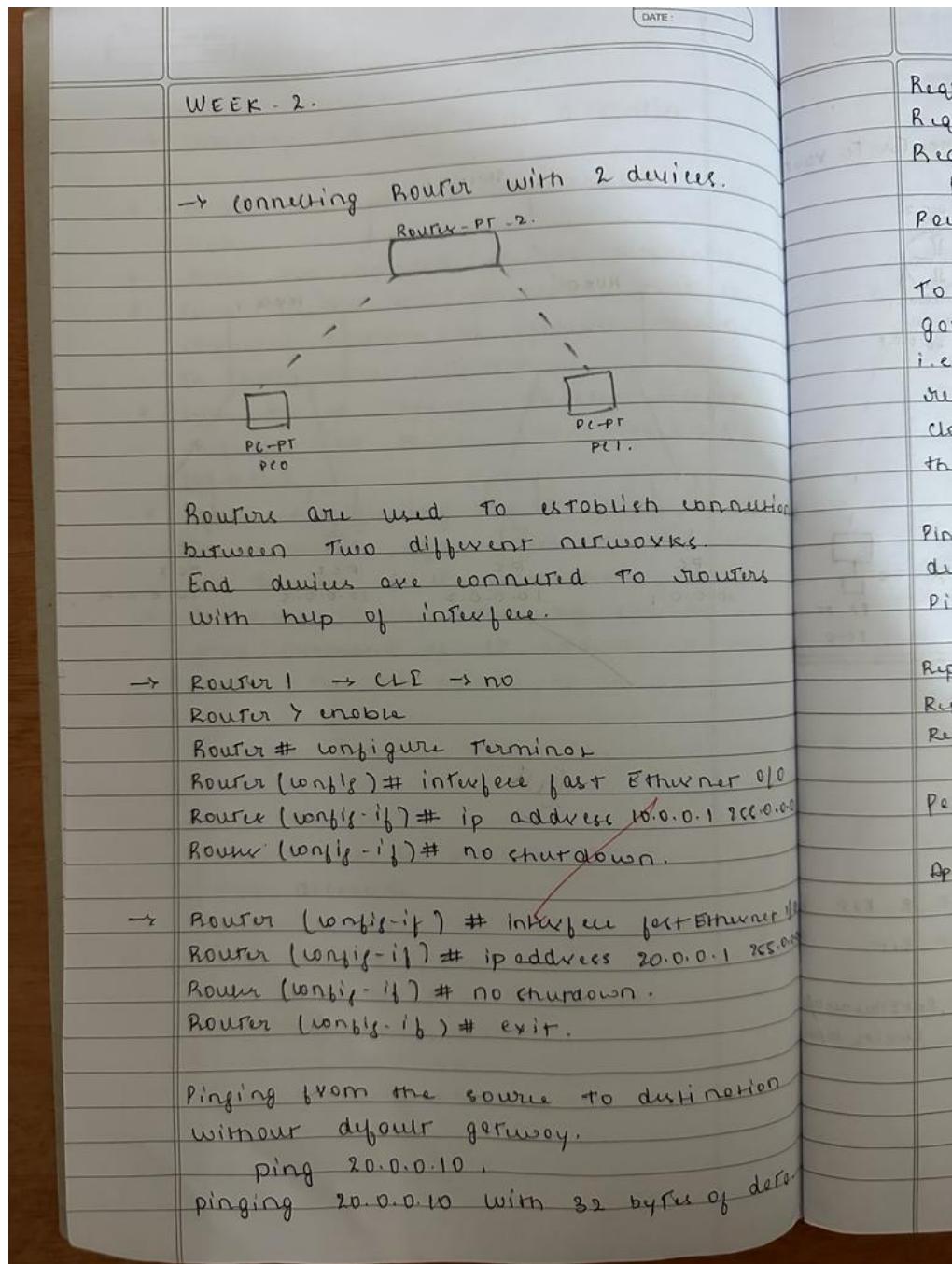
```



## WEEK 2

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

### OBSERVATION:



Request timed out.

Request timed out.

Request timed out.

Ping statistics for 20.0.0.10.

Packets sent=1 Received=0 Lost=1.

When we try to ping from source to destination without giving default gateway we get request timed out i.e. packets are sent back over nor received over local because the routers close nor know the path to send the packet to destination.

Pinging from source to destination with default gateway.

Ping 20.10.0.10.

Pinging 20.0.0.10 with 30 bytes of data.

Reply from 20.0.0.10 bytes=30 time=20ms T=

Reply from 20.0.0.10 bytes=32 time=0ms T=

Reply from 20.0.0.10 bytes=0ms T=127.

Packet sent=1 Received=1 Lost=0

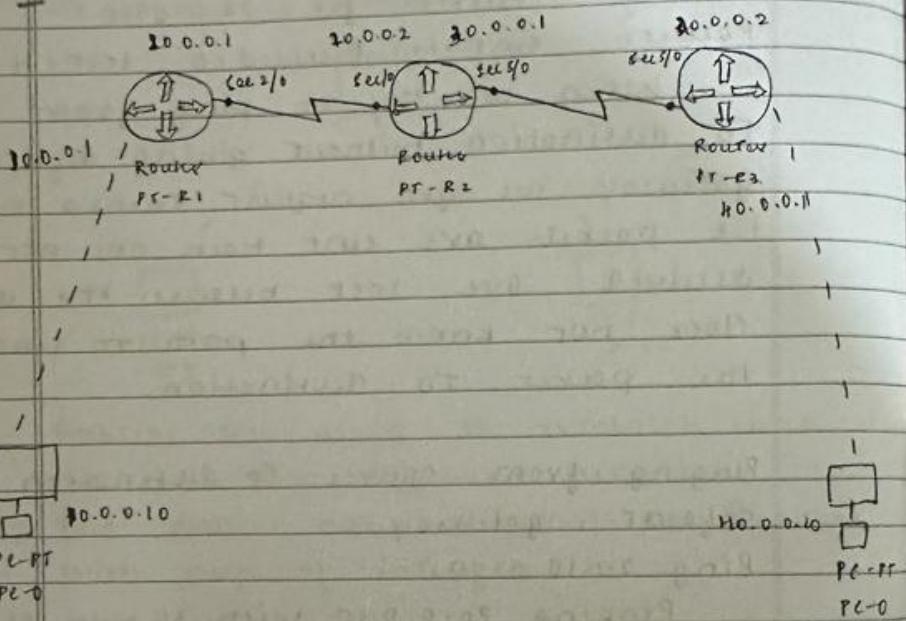
Approximate round trip time in ms.

Min=0ms Max=0ms Avg=0ms.

Ans

### WEEK 3

→ Configure default routes, static routes to your



### RESPONSE

Level 1.

- \* Enable.
- \* Show ip route.

Codes: C - Connected, S - Static, I - RIP/RP, R - RIP

Gateway of last resort is not set

- C 10.0.0.0/8 is directly connected, PortEthernet0
- C 20.0.0.0/8 is directly connected. Serial 1/0/1
- S 30.0.0.0/8 [1/0] via 20.0.0.2

## PROCEDURE

- \* Connect 3 Routers and 2 PCs using cross-over cable for the PC to router and a serial DCB cable to connect router to router.
- \* Set the IP address of both PC's and respective gateway number.
- \* For all 3 routers set the respective 2 IP address in CLI mode by using those commands.

→ Click on Router R1, then type IP route  
 $20.0.0.0 \text{ } 255.0.0.0 \text{ } 80.0.0.2$ .

→ Click on Router R2, then type IP route  
 $30.0.0.0 \text{ } 255.0.0.0 \text{ } 40.0.0.1$ .

Router RT-1.

Router & enable

Router # configure 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

Router (config-if)# exit

Router (config)# interface serial 0/0

Router (config)# ip address 20.0.0.1 255.0.0.0

Router (config-if)# no shutdown

# exit,

### Router PT-2.

Router > enable.

Router # configure terminal

Router (config) # interface serial 2/0

Router (config-if) # ip address 20.0.0.2 255.0.0.0

Router (config-if) # no shutdown.

Router (config-if) # exit.

Router (config) # interface serial 3/0

Router (config-if) # ip address 20.0.0.1 255.0.0.0

Router (config-if) # no shutdown.

### Router PT-3.

Router > enable.

Router # configure terminal

Router (config) # interface serial 2/0

Router (config-if) # ip address 30.0.0.2 255.0.0.0

Router (config-if) # no shutdown

Router (config-if) # exit.

Router (config) # interface serial 2/0

Router (config) # ip address 10.0.0.1 255.0.0.0

Router (config-if) # no shutdown.

For Router 1, set the IP route of other IP addresses statically by using following steps.

R1

\* Router > show ip route

C 10.0.0.0 is directly connected

C 20.0.0.0 is directly connected

Router > Enable

Router # configure terminal

# ip route 30.0.0.0 255.0.0.0 20.0.0.2

# ip route 10.0.0.0 255.0.0.0 20.0.0.2

R2

- \* Router # show ip route
  - C 20.0.0.0 is directly connected.
  - C 30.0.0.0 is directly connected.

Router # configure terminal

Router # ip route 10.0.0.0 255.0.0.0 20.0.0.1

# ip route 10.0.0.0 255.0.0.0 30.0.0.2

R3.

- \* Router # show ip route
  - C 30.0.0.0 is directly connected.
  - C 10.0.0.0 is directly connected.

Router # configure terminal

Router (config) # ip route 20.0.0.0 255.0.0.0  
30.0.0.1.

Router (config) # ip route 10.0.0.0 255.0.0.0  
30.0.0.2.

~~PC~~ PING OUTPUT.

PC # ping 10.0.0.1

Pinging 10.0.0.1 by PC - 32 bytes - 2ms TIC - 1s

Reply from 10.0.0.1: bytes=32 time=2ms TIC

Reply from 10.0.0.1: bytes=32 time=2ms TIC > 1s.

Ping statistics for 10.0.0.1

Packets: SENT=4 Received=3 loss=1.

## TOPOLOGY:

Approximate round trip times in milli seconds

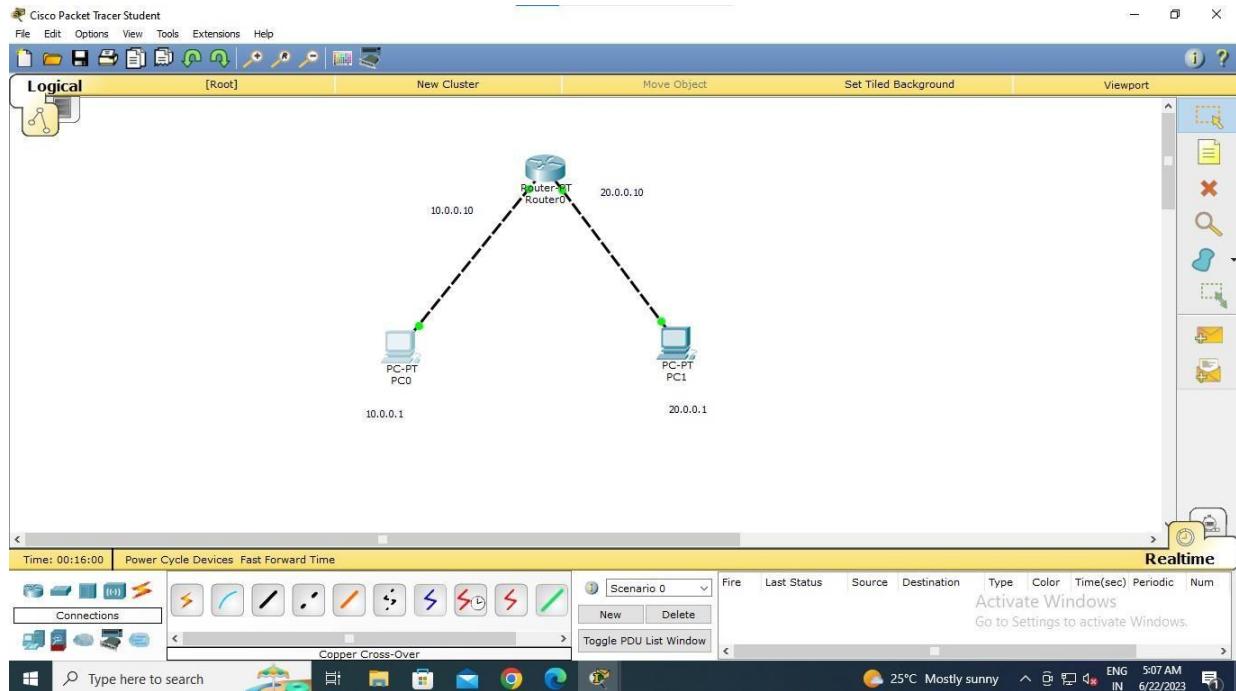
Minimum = 2 ms, Maximum = 16 ms, Average = 6 ms

#### OBSERVATION.

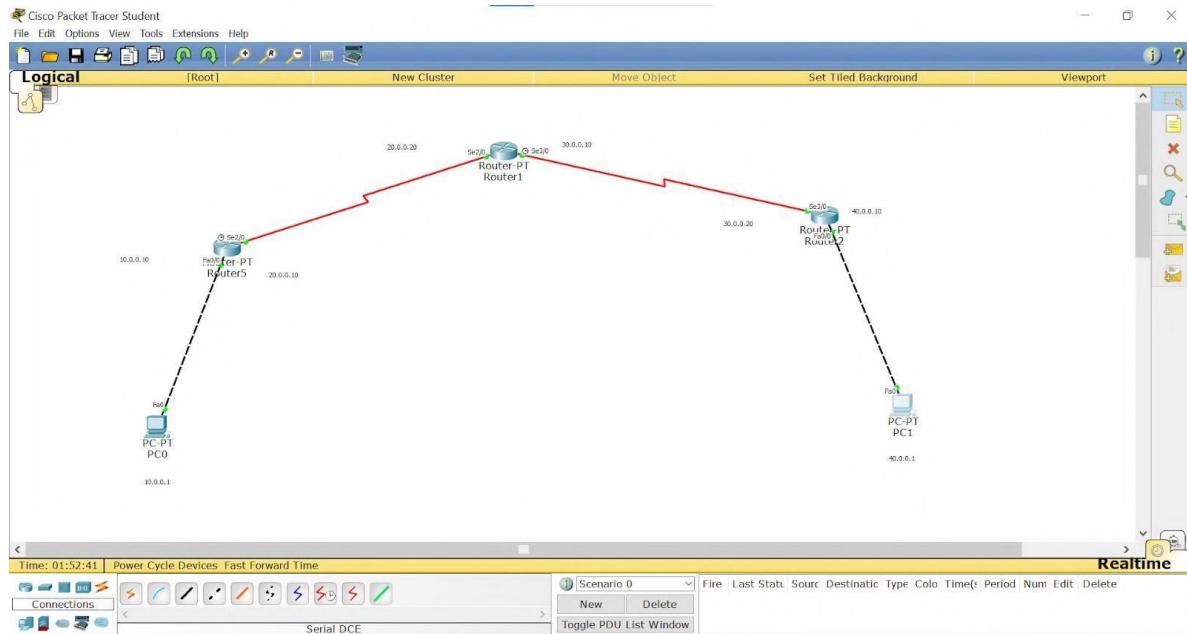
- \* A default route is the route which takes effect when no other route is available for an IP address destination.
- \* If a packet is received, the device first checks the IP destination address is not the device checks its routing table.
- \* If the remote destination subnet is not listed then the packet is forwarded to the next hop toward the destination using the default route.
- \* The process repeats until the packet is delivered.

full

## PROGRAM 2.1

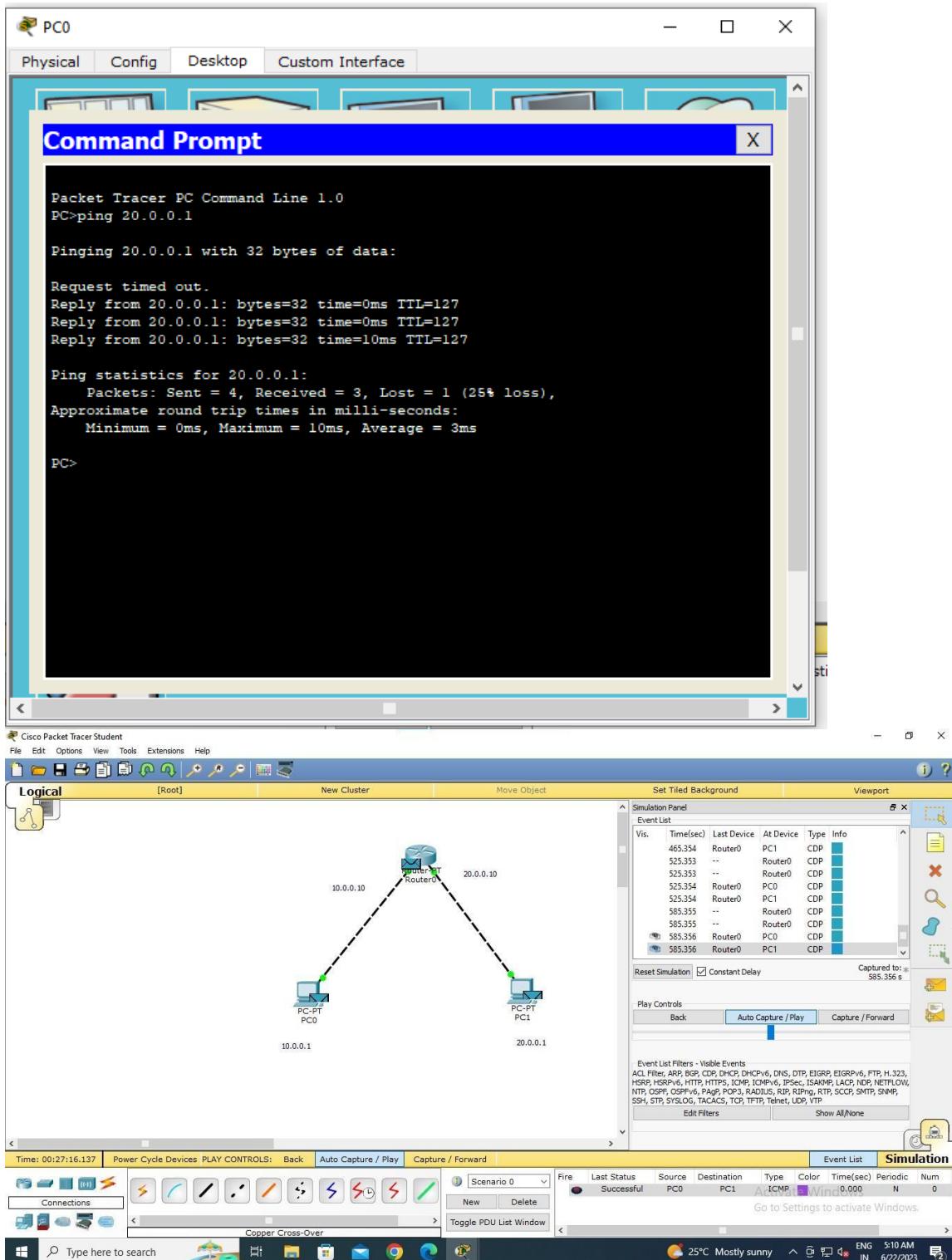


## PROGRAM 2.2



## OUTPUT:

### PROGRAM 2.1



## PROGRAM 2.2

The image shows two separate Command Prompt windows from the Packet Tracer software. Both windows have a blue title bar with the text "Command Prompt" and an "X" button. The window on the left is titled "PC0" and the window on the right is titled "PC1". Both windows have a toolbar at the top with tabs: "Physical", "Config", "Desktop", and "Custom Interface". The "Physical" tab is selected in both windows.

**PC0 Command Prompt Output:**

```
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Reply from 10.0.0.10: Destination host unreachable.
Reply from 10.0.0.10: Destination host unreachable.
Reply from 10.0.0.10: Destination host unreachable.
Request timed out.

Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>
```

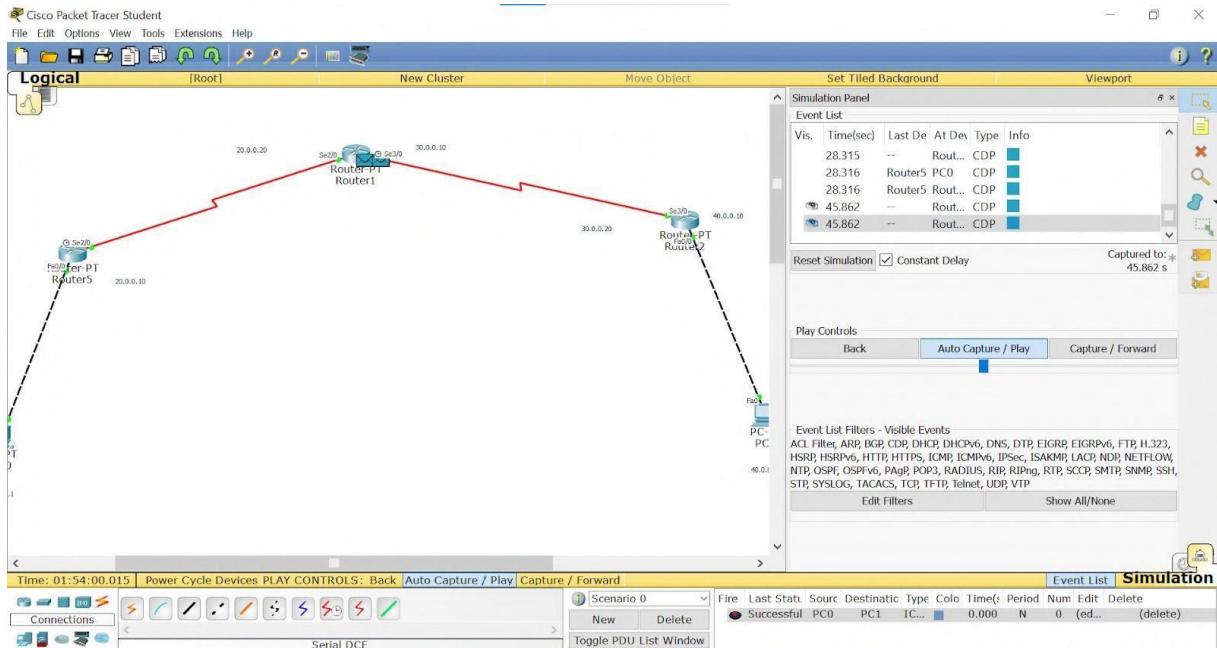
**PC1 Command Prompt Output:**

```
Packet Tracer PC Command Line 1.0
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=2ms TTL=125
Reply from 10.0.0.1: bytes=32 time=8ms TTL=125
Reply from 10.0.0.1: bytes=32 time=2ms TTL=125
Reply from 10.0.0.1: bytes=32 time=2ms TTL=125

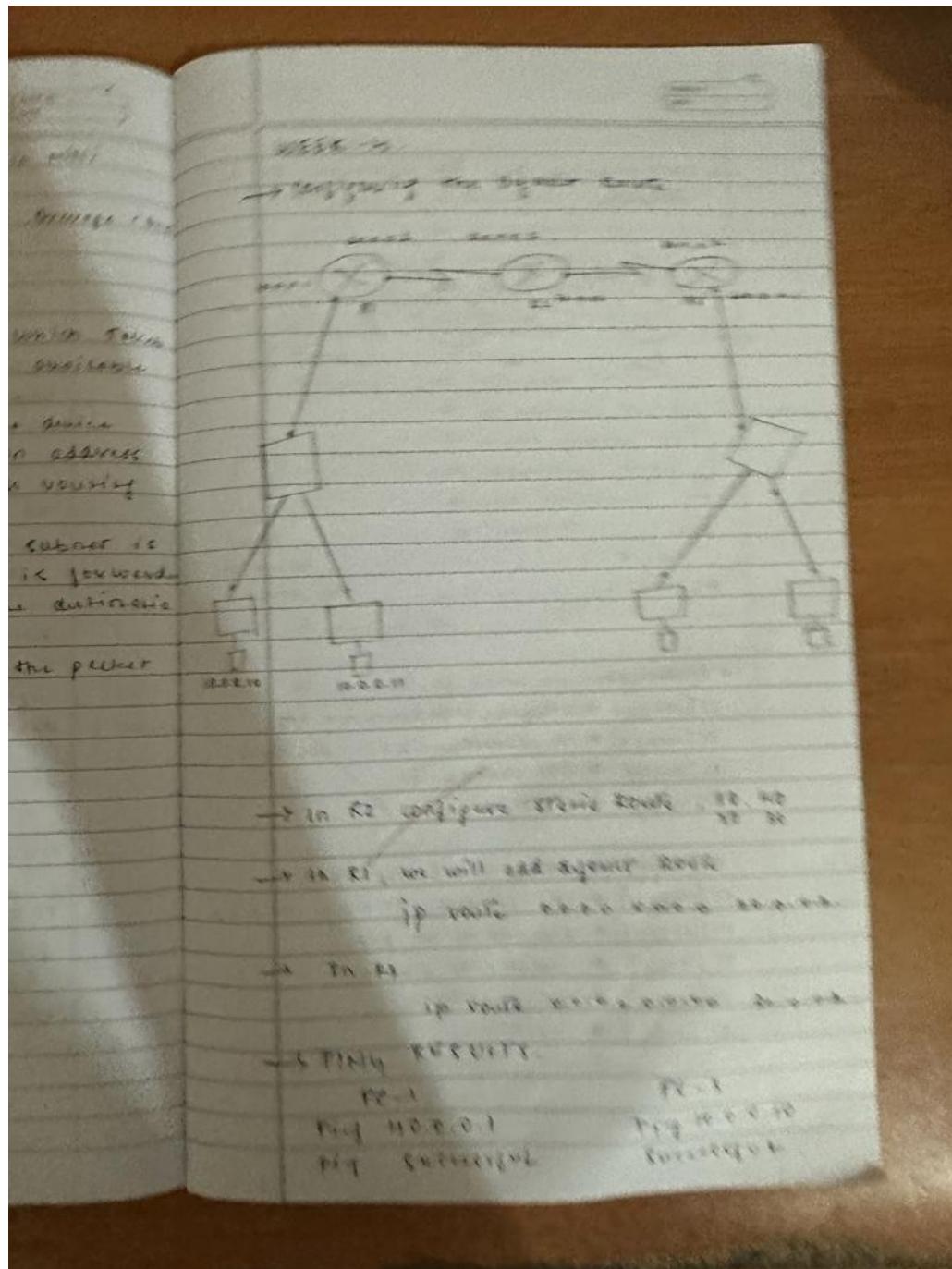
Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 8ms, Average = 3ms
PC>
```



## WEEK 3

Configure default route, static route to the Router.

OBSERVATION:



In Router R2.

```
R2#config # interface serial 1/0  
# ip address 30.0.0.2 255.0.0.0  
# encapsulation ppp  
# no shutdown.  
# exit
```

```
R2(config)# interface serial 1/1  
# ip address 30.0.0.1 255.0.0.0  
# encapsulation ppp  
# clockrate 64000  
# no shutdown  
# exit
```

In Router R3.

```
R3(config)# interface serial 0/0  
# ip address 30.0.0.2 255.0.0.0  
# encapsulation ppp  
# no shutdown.  
# exit
```

```
R3(config)# interface fastEthernet 0/0  
# ip address 40.0.0.1 255.0.0.0  
# no shutdown  
# exit
```

→ Configure RIP to the Router.

- \* Default gateway is the router address  
(10.0.0.0.2.7)
- \* DNS server is the server address.  
(10.0.0.1).
- \* TFTP address is the server address.  
(10.0.0.1)
- \* Change the start address to 10.0.0.1
- \* Now go to and click on desktop → IP configuration.
- \* In IP configuration select DHCP.
- \* The IP addresses will be set dynamically.

OUTPUT

PC-1

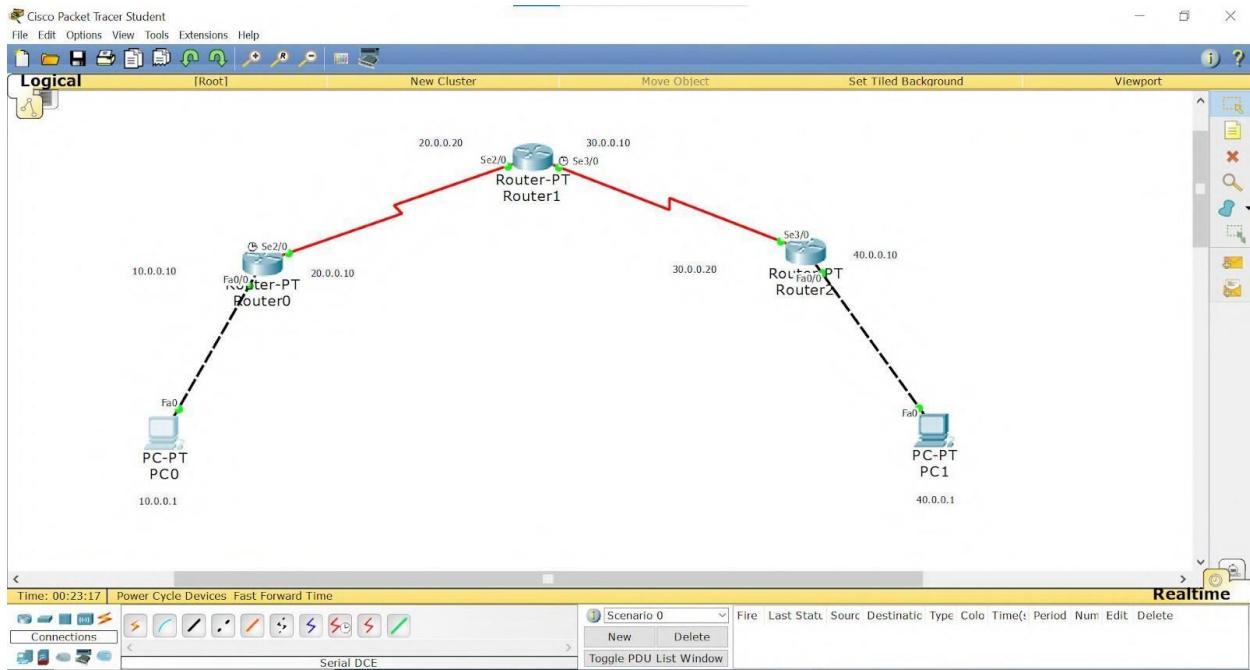
IP address 10.0.0.3.

PC-2

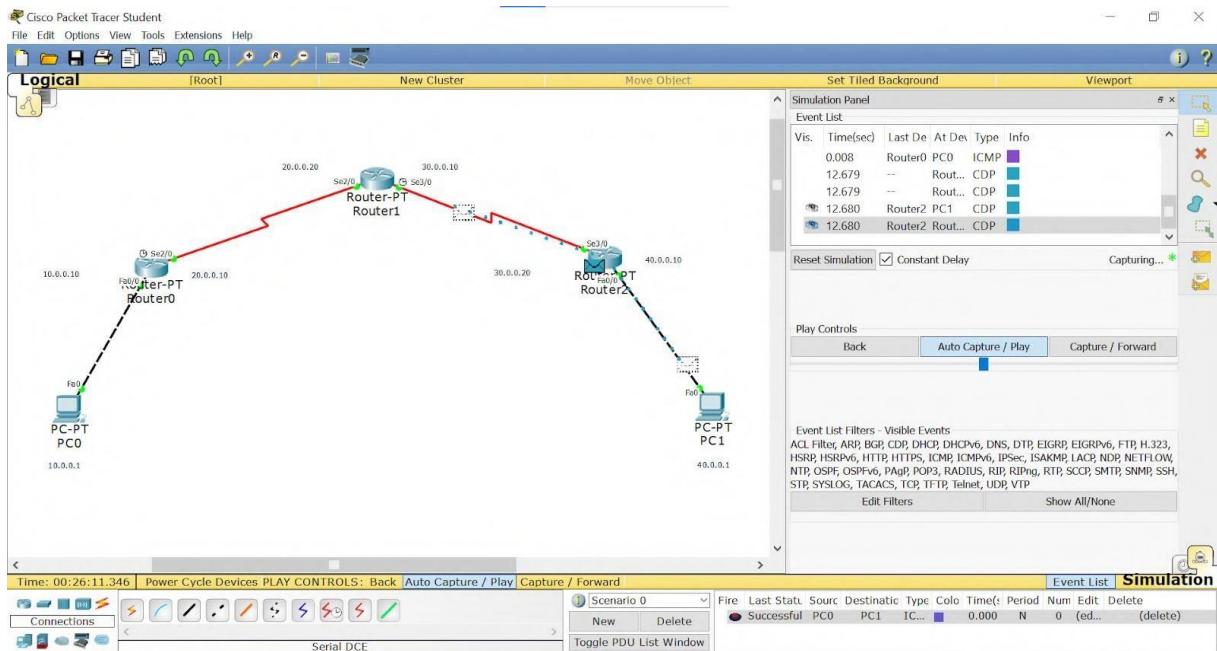
IP address 10.0.0.4

See /

## TOPOLOGY:



## OUTPUT:



PC0

Physical Config Desktop Custom Interface

## Command Prompt

```
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125
Reply from 40.0.0.1: bytes=32 time=16ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125

Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 16ms, Average = 6ms

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=21ms TTL=125
Reply from 40.0.0.1: bytes=32 time=9ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125
Reply from 40.0.0.1: bytes=32 time=4ms TTL=125

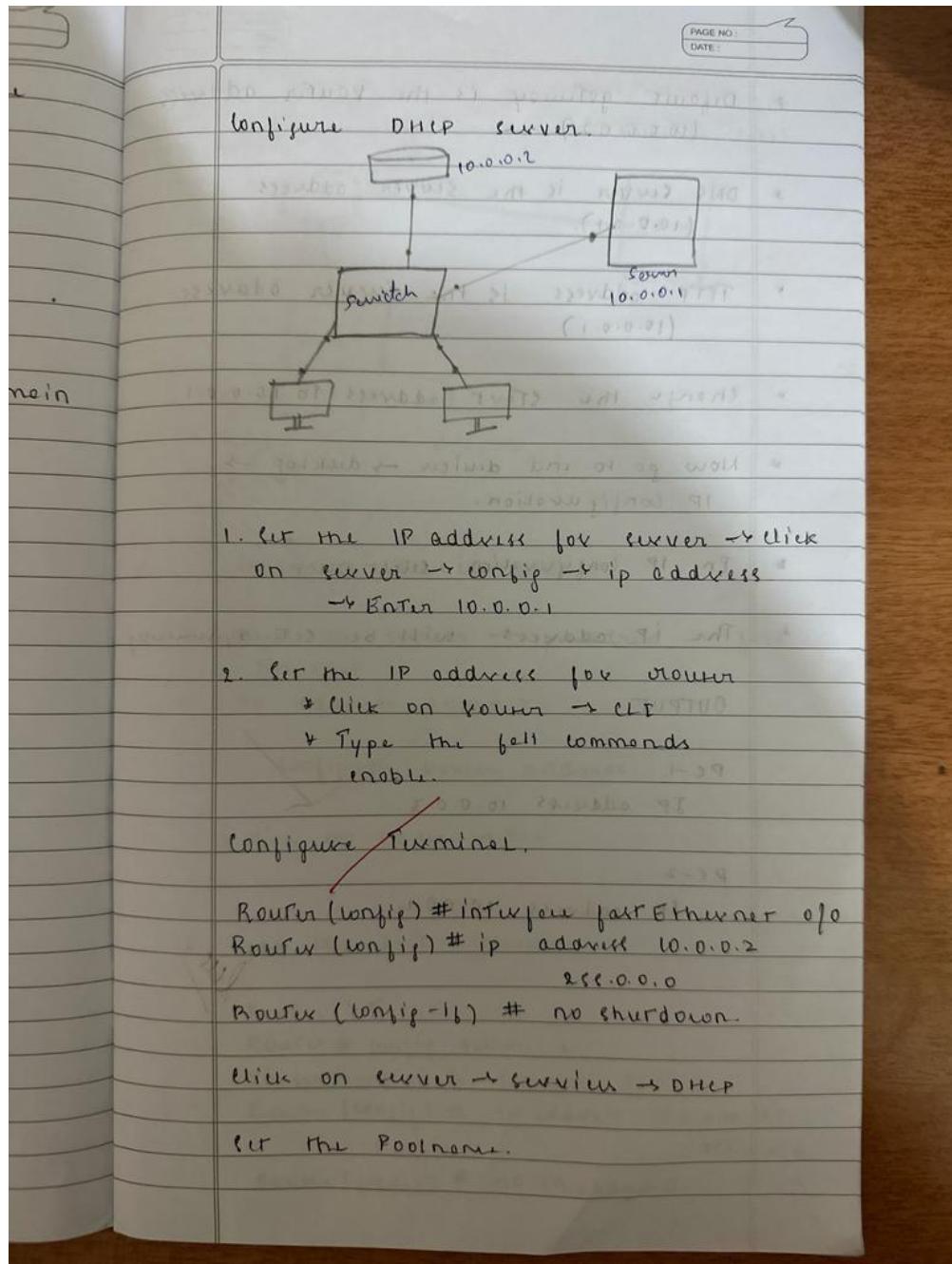
Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 21ms, Average = 9ms

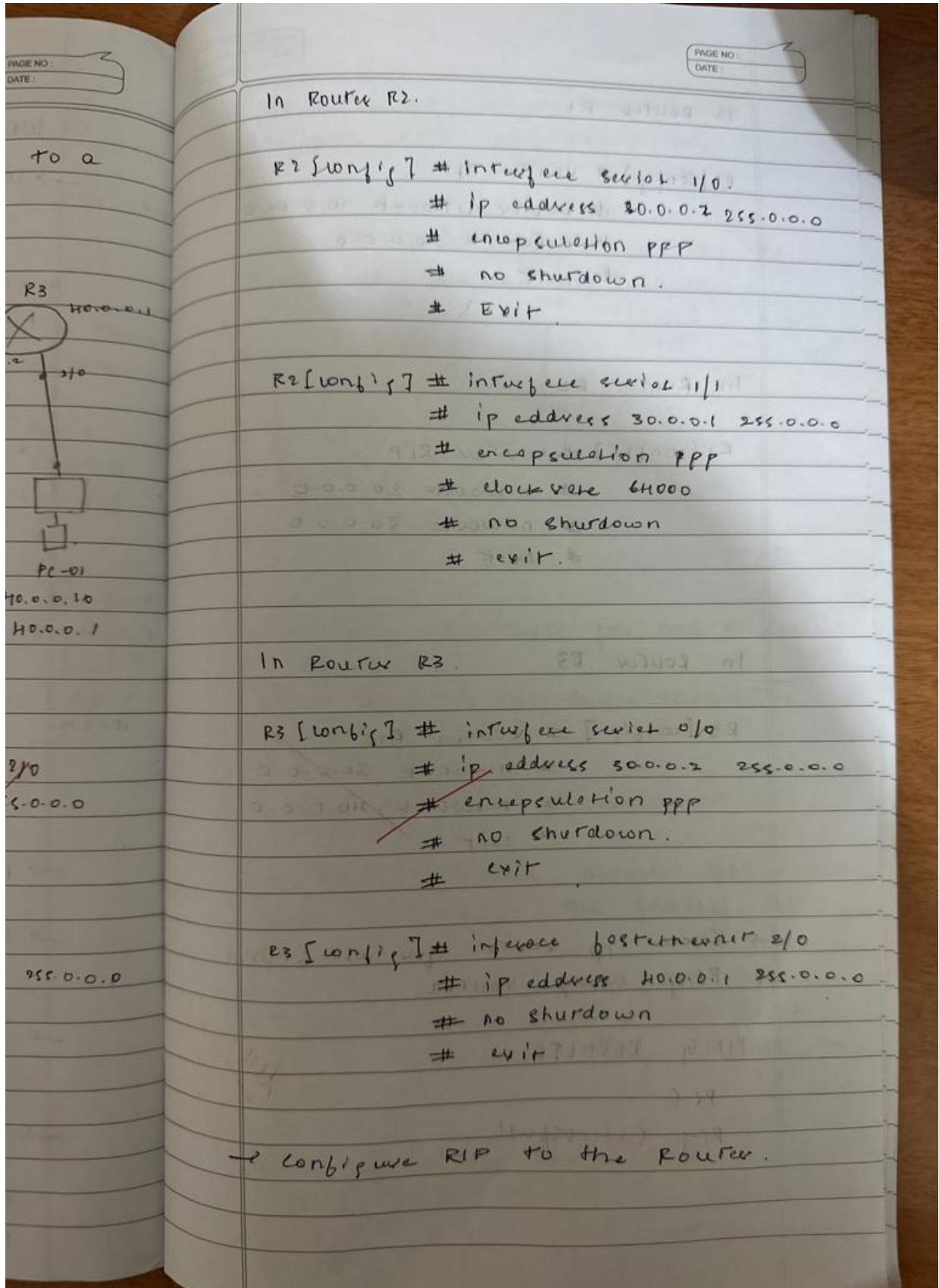
PC>
```

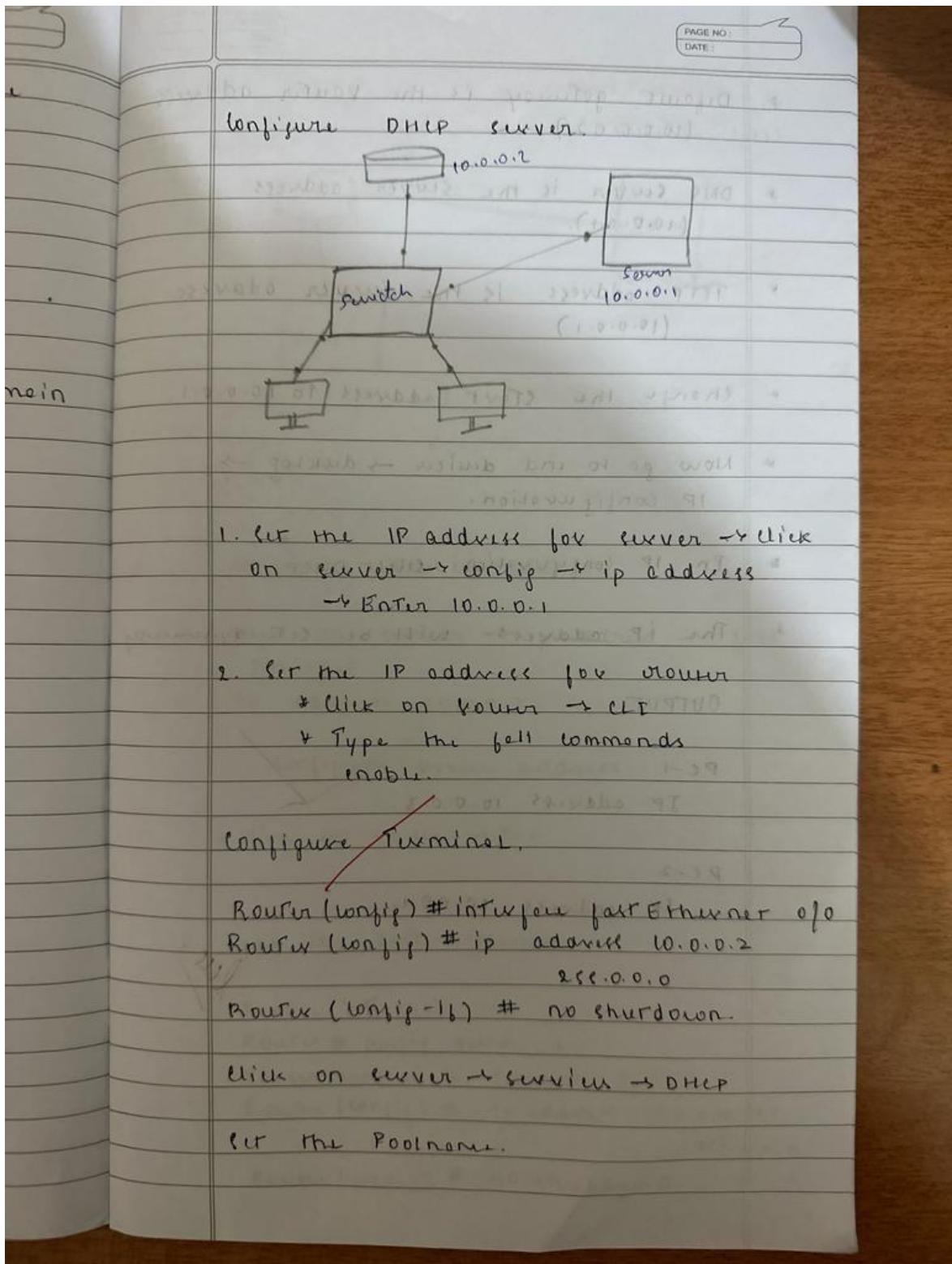
# WEEK 4

Configure DHCP within a LAN and outside LAN.

OBSERVATION:







- PAGE NO: \_\_\_\_\_  
DATE: \_\_\_\_\_
- \* Default gateway is the server address (10.0.0.0.2)
  - \* DNS server is the server address. (10.0.0.1)
  - \* TFTP address is the server address. (10.0.0.1)
  - \* Change the Start address to 10.0.0.1
  - \* Now go to and devices → desktop → IP configuration.
  - \* In IP configuration select DHCP.
  - \* The IP addresses will be set dynamically.

OUTPUT

PC-1

IP address 10.0.0.3.

PC-2

IP address 10.0.0.4.

See

DATE:

Router (config-if) # interface fasto

Router (config-if) # ip address 20.0.0.10  
255.0.0.0.

Router (config-if) # ip helper address  
10.0.0.2

Router (config-if) # no shutdown.

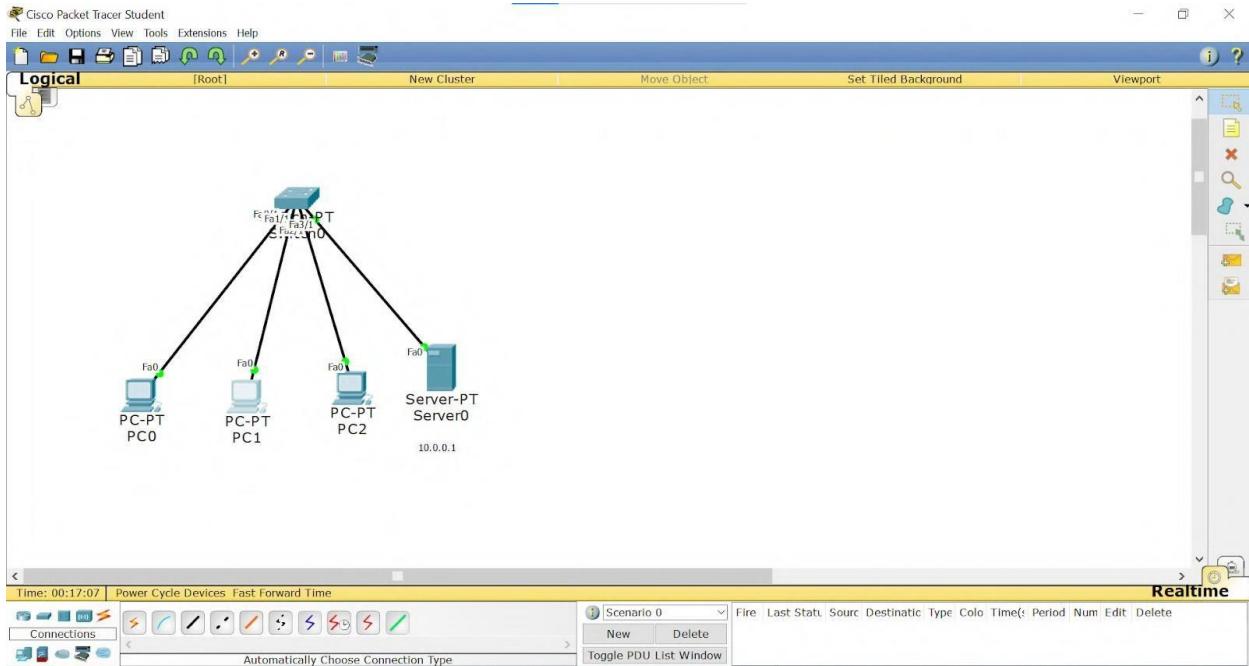
~~free  
2/8/23~~



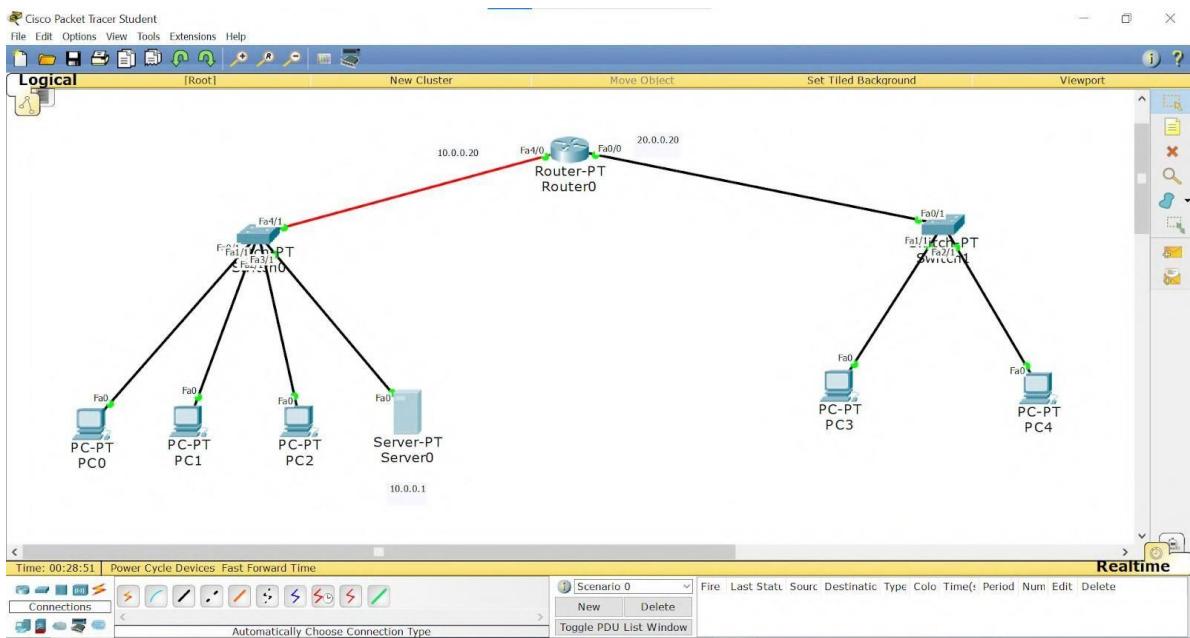


## TOPOLOGY:

### PROGRAM 4.1:

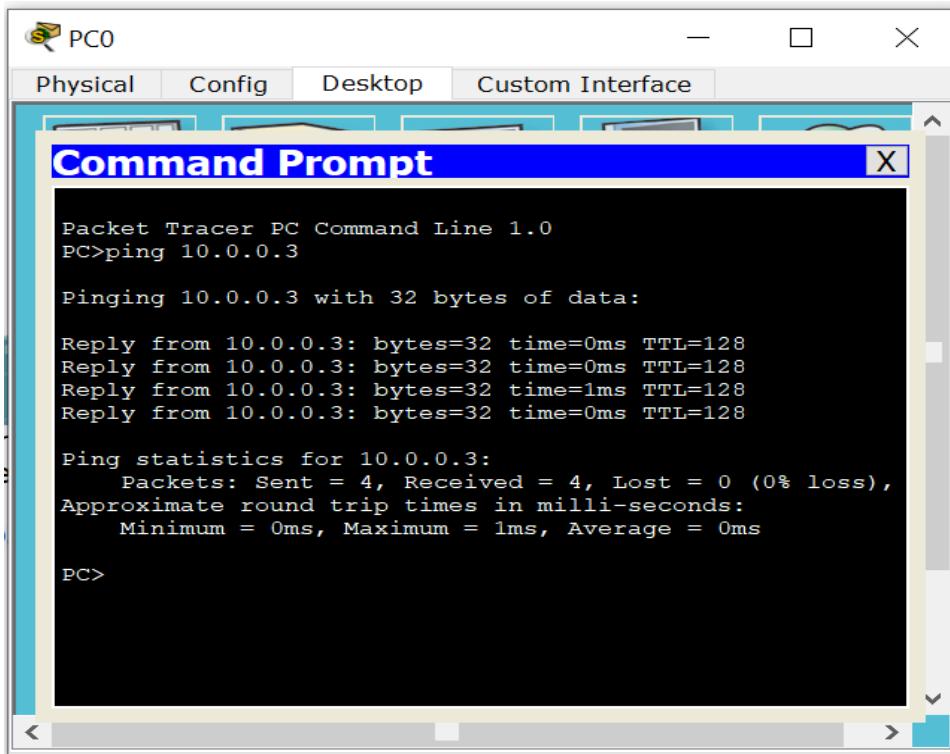


### PROGRAM 4.2:



## OUTPUT:

### PROGRAM 4.1:



PC0

Physical Config Desktop Custom Interface

**Command Prompt**

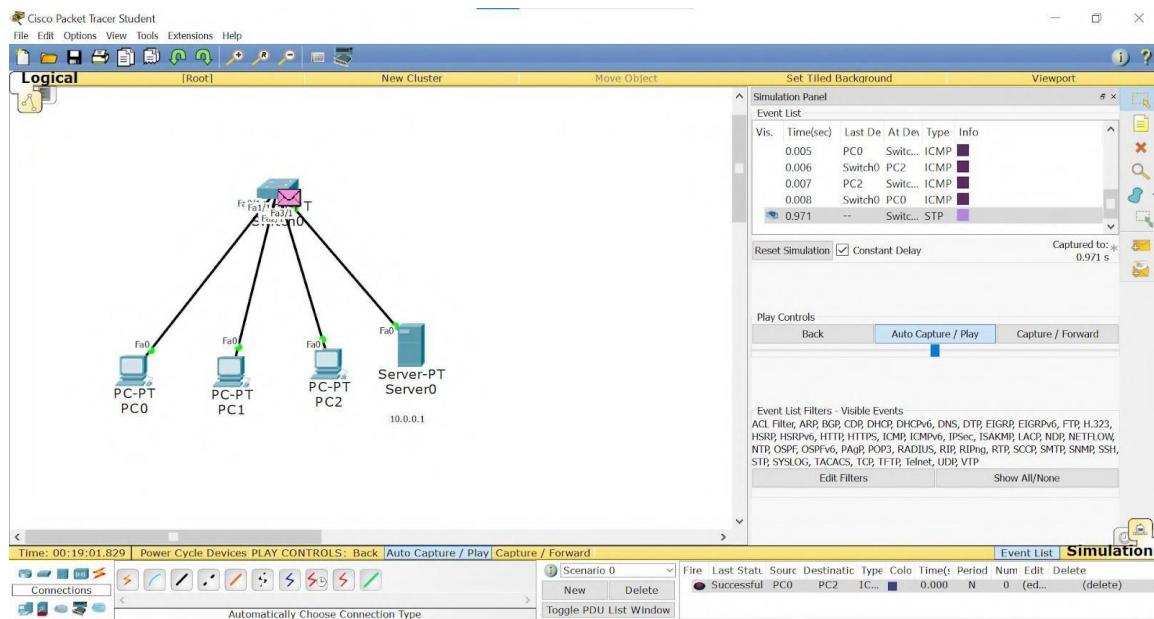
```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

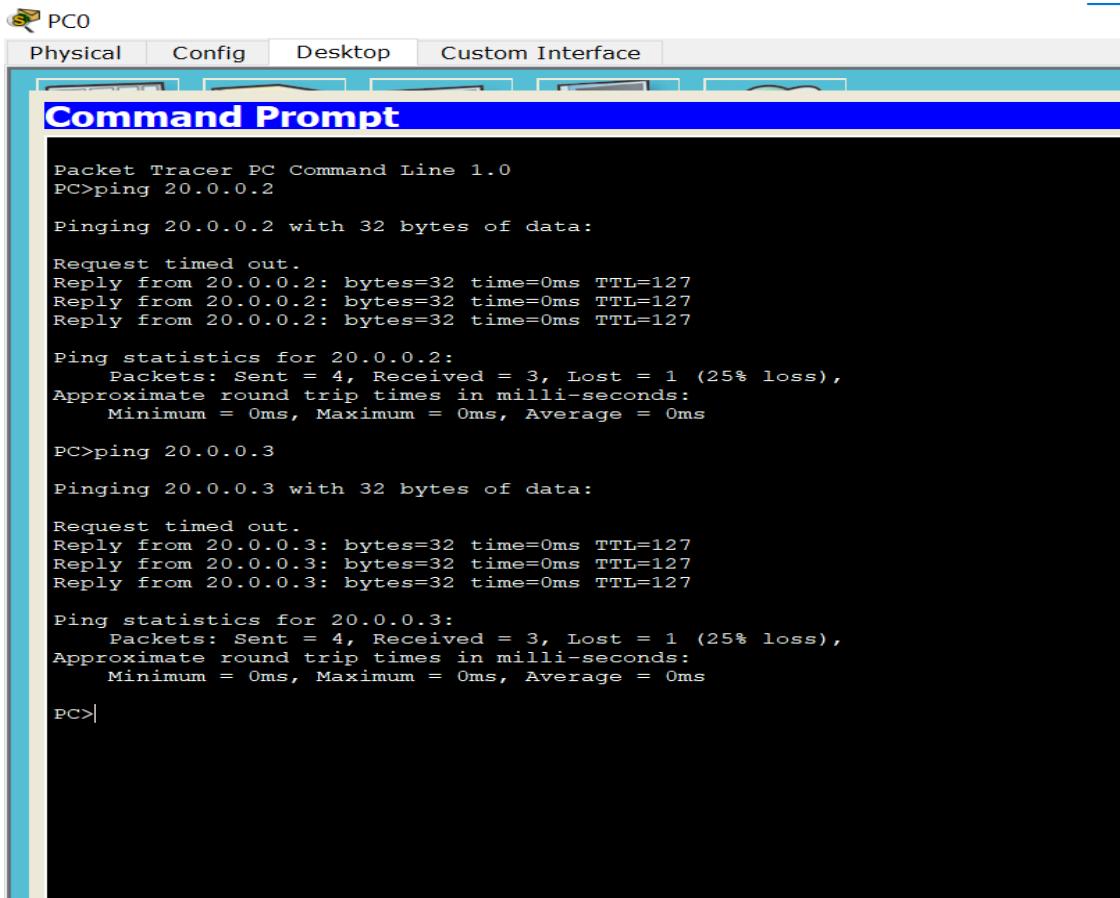
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=1ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128

Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

PC>
```



## PROGRAM 4.2:



The screenshot shows the "Command Prompt" window of the Packet Tracer PC Command Line interface. The window title is "Command Prompt". The content displays the output of several ping commands. The first command is "ping 20.0.0.2", which shows three replies from the target IP. The second command is "ping 20.0.0.3", which also shows three replies. Both commands include ping statistics at the end, indicating a 25% loss of packets. The interface has tabs at the top: Physical, Config, Desktop, and Custom Interface, with "Custom Interface" being the active tab.

```
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.2

Pinging 20.0.0.2 with 32 bytes of data:

Request timed out.
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127

Ping statistics for 20.0.0.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

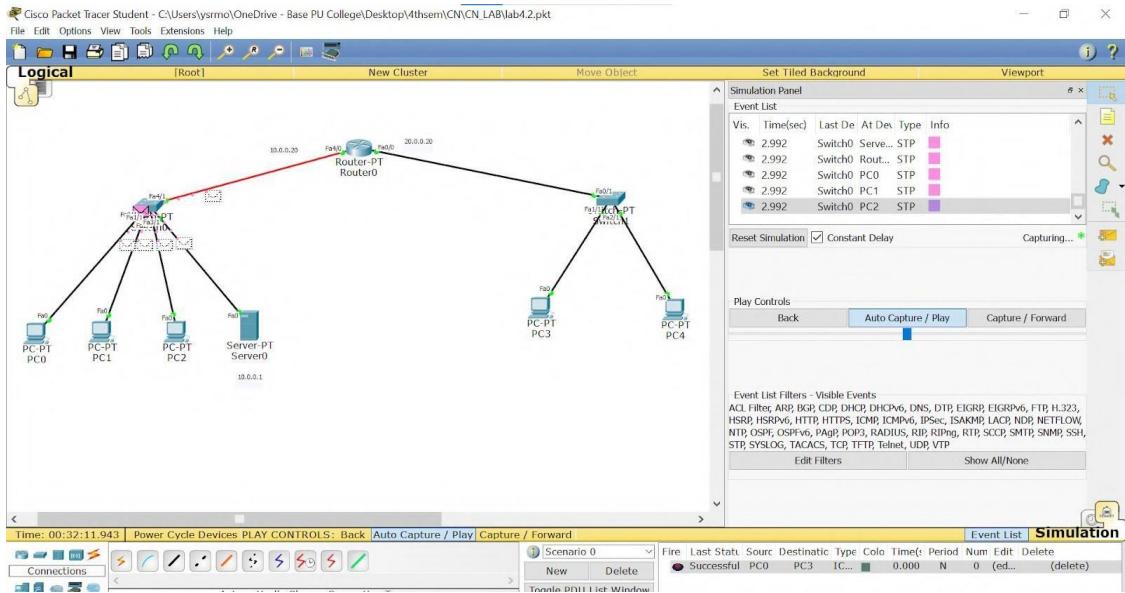
PC>ping 20.0.0.3

Pinging 20.0.0.3 with 32 bytes of data:

Request timed out.
Reply from 20.0.0.3: bytes=32 time=0ms TTL=127
Reply from 20.0.0.3: bytes=32 time=0ms TTL=127
Reply from 20.0.0.3: bytes=32 time=0ms TTL=127

Ping statistics for 20.0.0.3:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

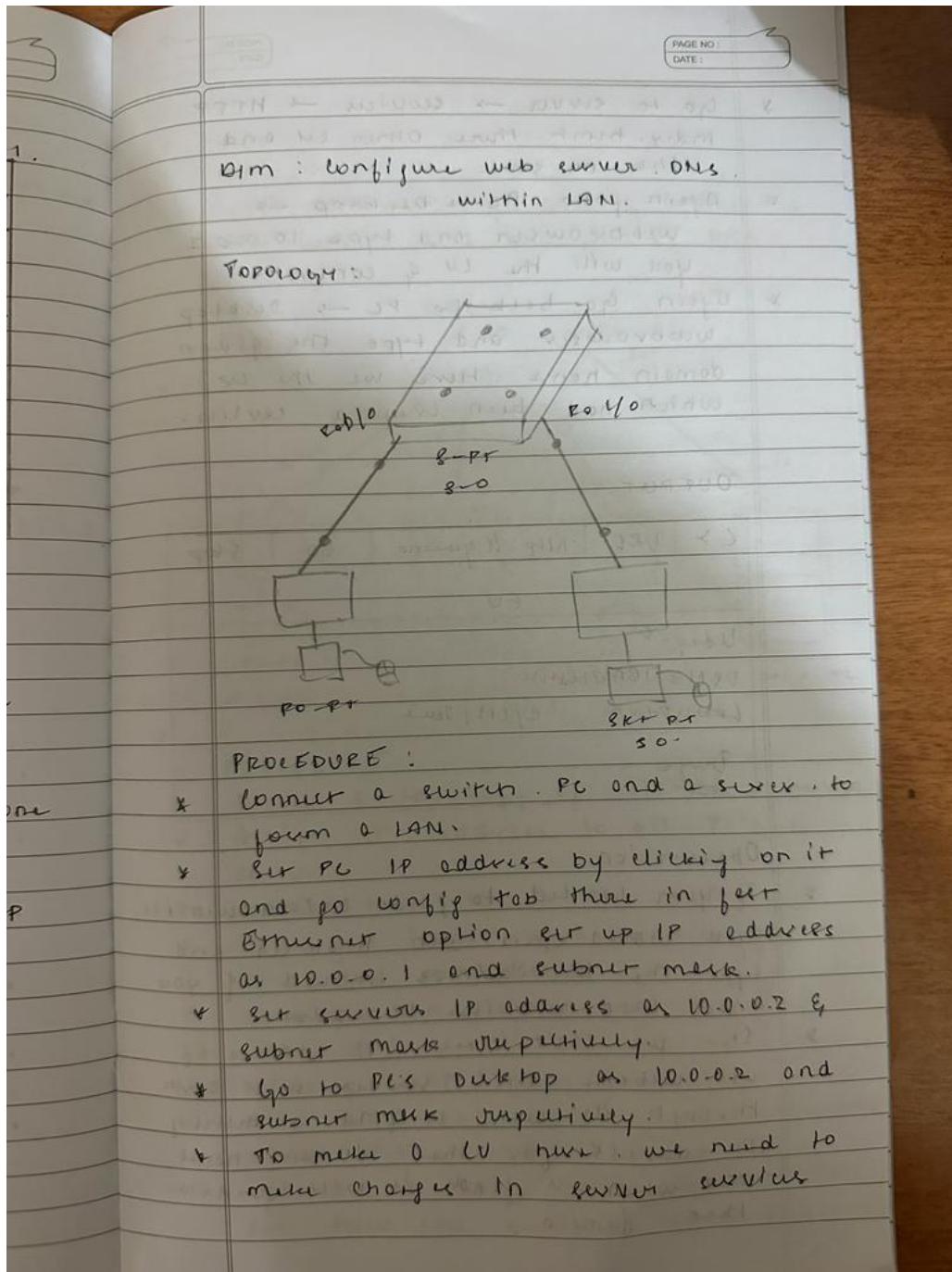
PC>|
```



# WEEK 5

Configure Web Server, DNS within a LAN.

OBSERVATION:



- \* Go to https://www.https://
- \* Many hosts have https on one  
place on same
- \* Open port 80 or 443 →  
https://domain.com https://domain.com  
you will see it is connect
- \* Open the file in PC → Clicked  
https://domain.com https://domain.com  
should know how to do it  
when has been created website.

#### QUESTION

Q & A for the question go to Q & A

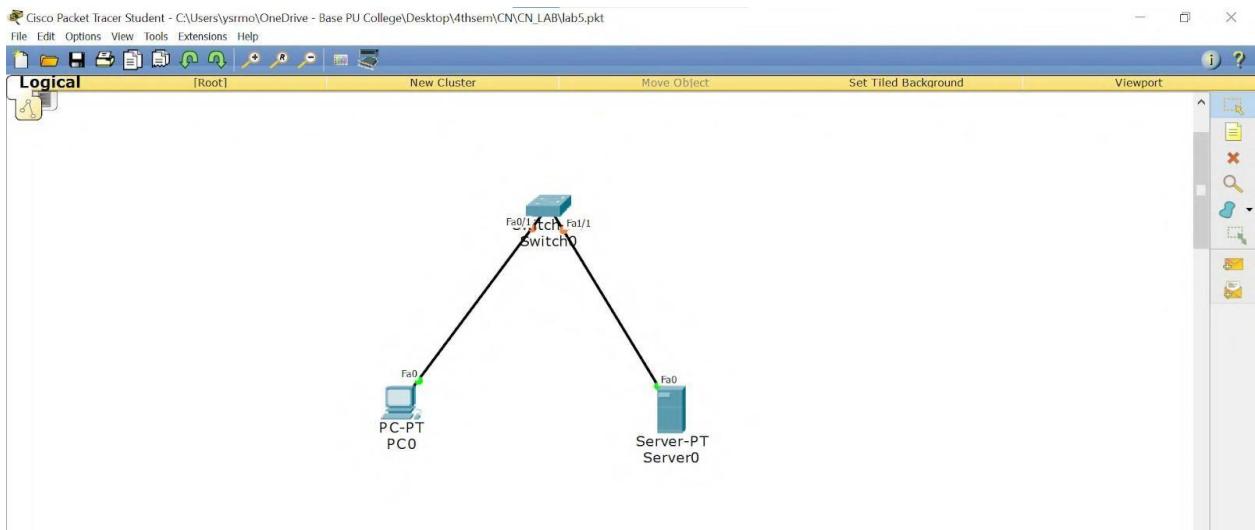
Topic 2 -  
SSL / TLS  
Protocol  
Encryption / Decryption

Source  
=

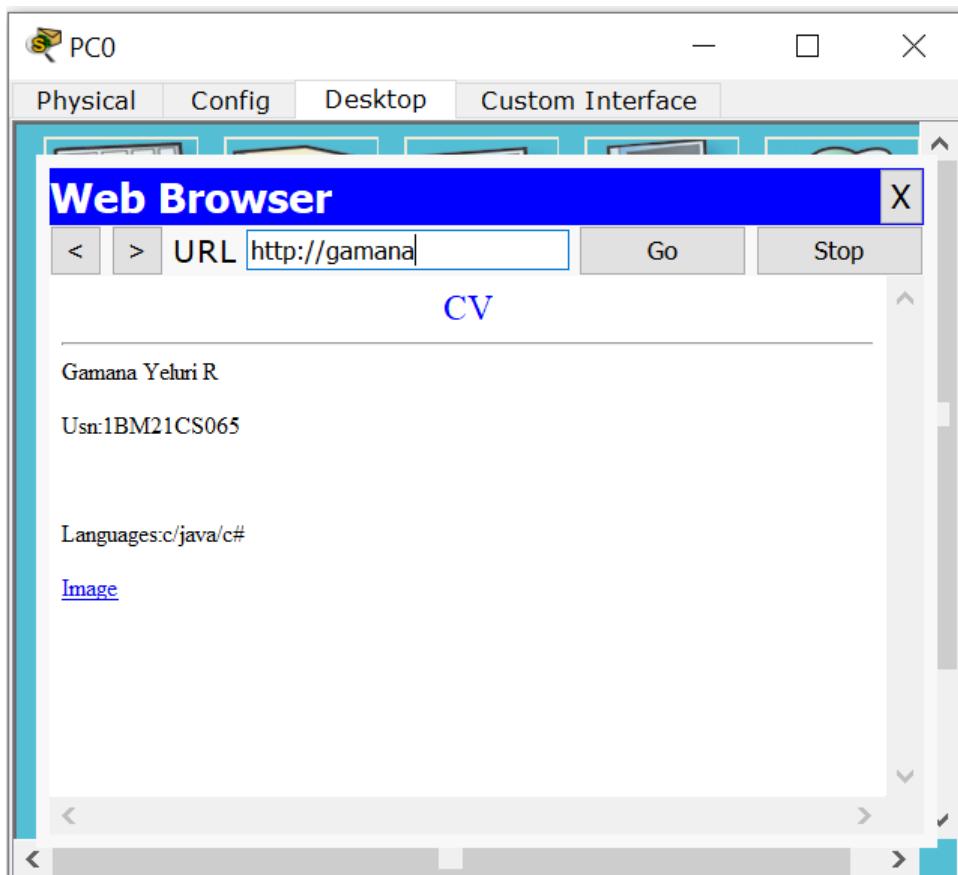
#### Observation

- \* If you wanted to go to website without  
the https part just type https and  
type in if website https if you  
know that something up website
- \* In the URL domain is written as  
http:// which does browser will run  
through the url to find a security  
if website for that domain https  
and when it finds it will connect  
that domain

## TOPOLOGY:



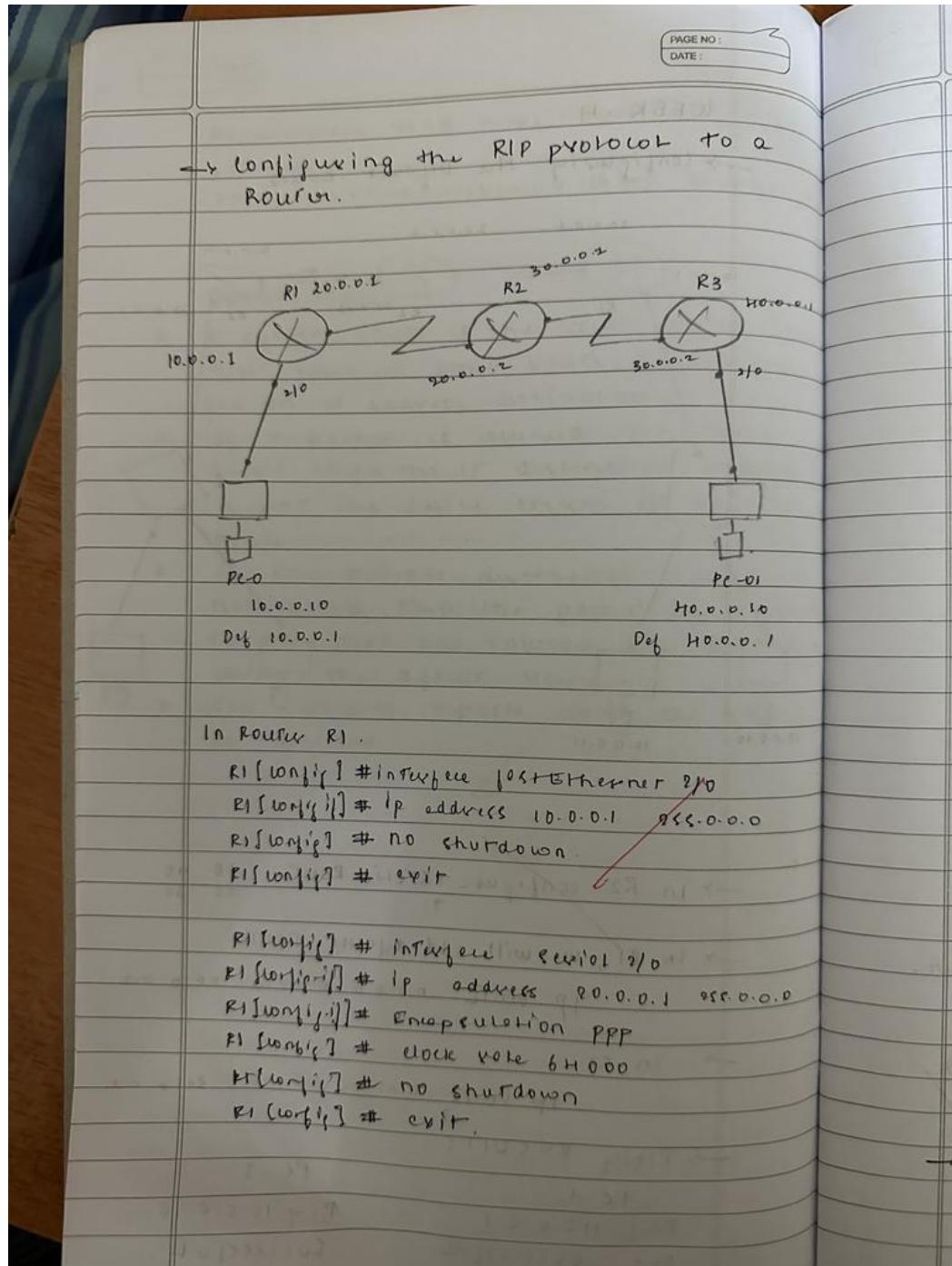
## OUTPUT:



# WEEK 6

Configure RIP routing Protocol in Routers.

OBSERVATION:



In Router R2.

```
R2 [config] # interface serial 1/0
# ip address 20.0.0.2 255.0.0.0
# encapsulation PPP
# no shutdown.
# exit.
```

```
R2 [config] # interface serial 1/1
# ip address 30.0.0.1 255.0.0.0
# encapsulation PPP
# clockrate 64000
# no shutdown
# exit.
```

In Router R3.

```
R3 [config] # interface serial 0/0
# ip address 50.0.0.2 255.0.0.0
# encapsulation PPP
# no shutdown.
# exit.
```

```
R3 [config] # interface fastethernet 0/0
# ip address 40.0.0.1 255.0.0.0
# no shutdown
# exit.
```

→ Configure RIP to the Router.

In ROUTER R1

```
R1 [config] # router RIP  
# network 10.0.0.0  
# network 90.0.0.0  
# exit.
```

In ROUTER R2

```
R2 [config] # router RIP  
# network 20.0.0.0  
# network 30.0.0.0  
# exit.
```

In ROUTER R3

```
R3 [config] # router RIP  
# network 30.0.0.0  
# network 40.0.0.0  
# exit.
```

- Show ip route for all router.
- Ping every router.

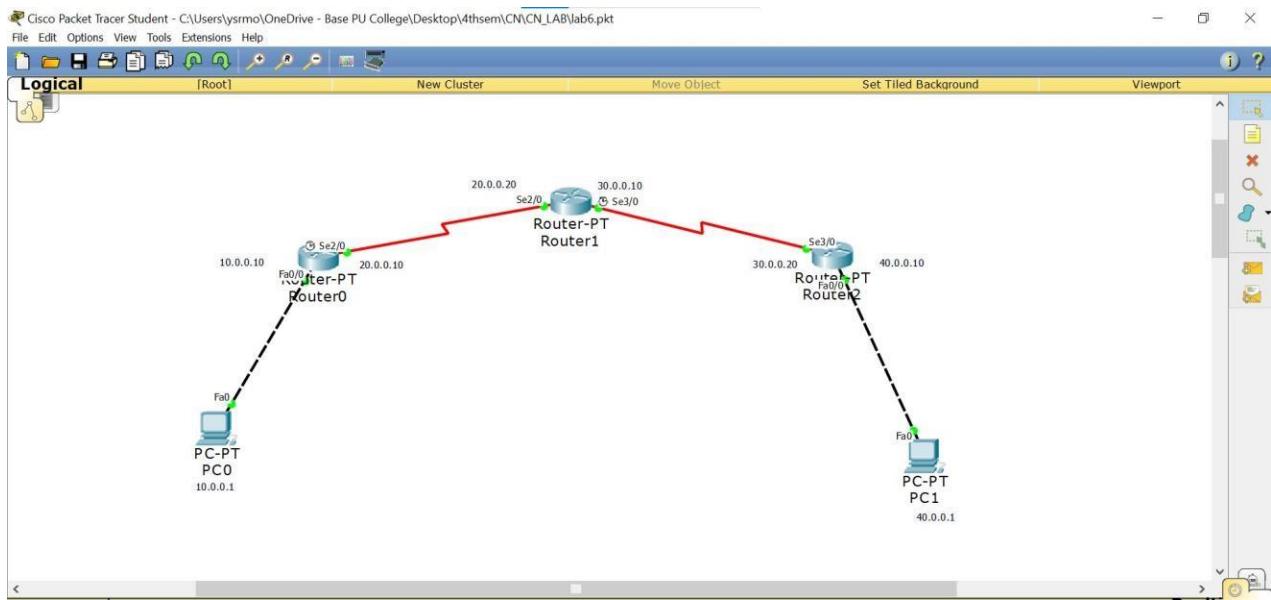
PING RESULTS.

PC 0

Ping successful.

tee

## TOPOLOGY:



## OUTPUT:

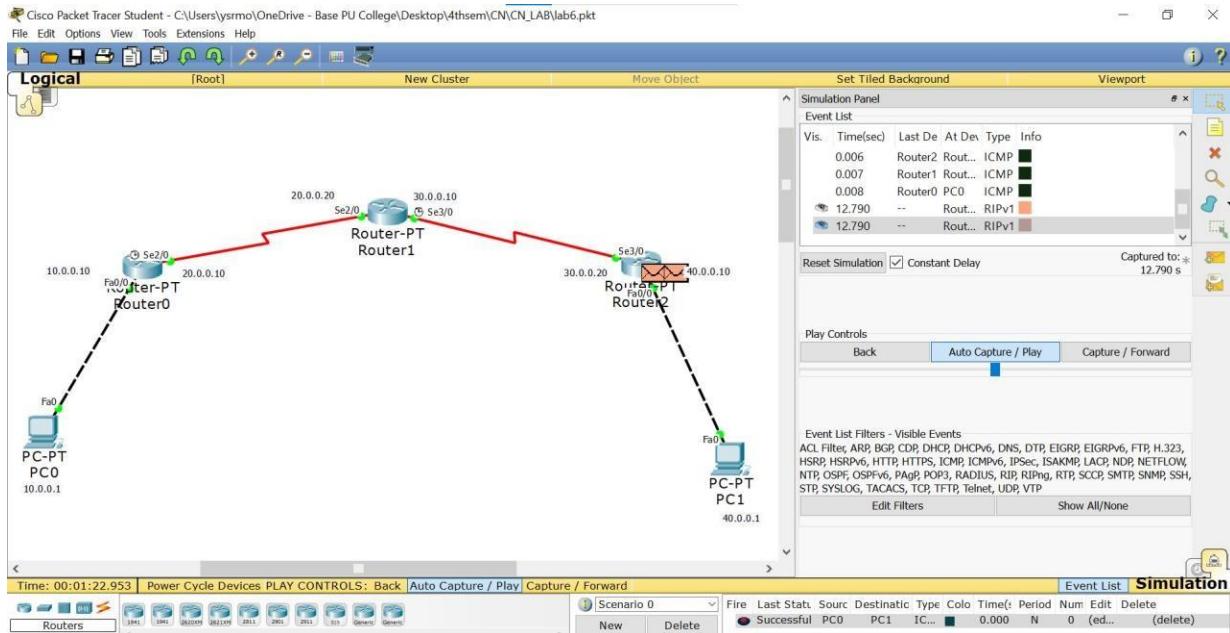
```
PC0
Physical Config Desktop Custom Interface
Command Prompt X
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.1: bytes=32 time=0ms TTL=125
Reply from 40.0.0.1: bytes=32 time=5ms TTL=125
Reply from 40.0.0.1: bytes=32 time=10ms TTL=125

Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 5ms, Maximum = 10ms, Average = 7ms

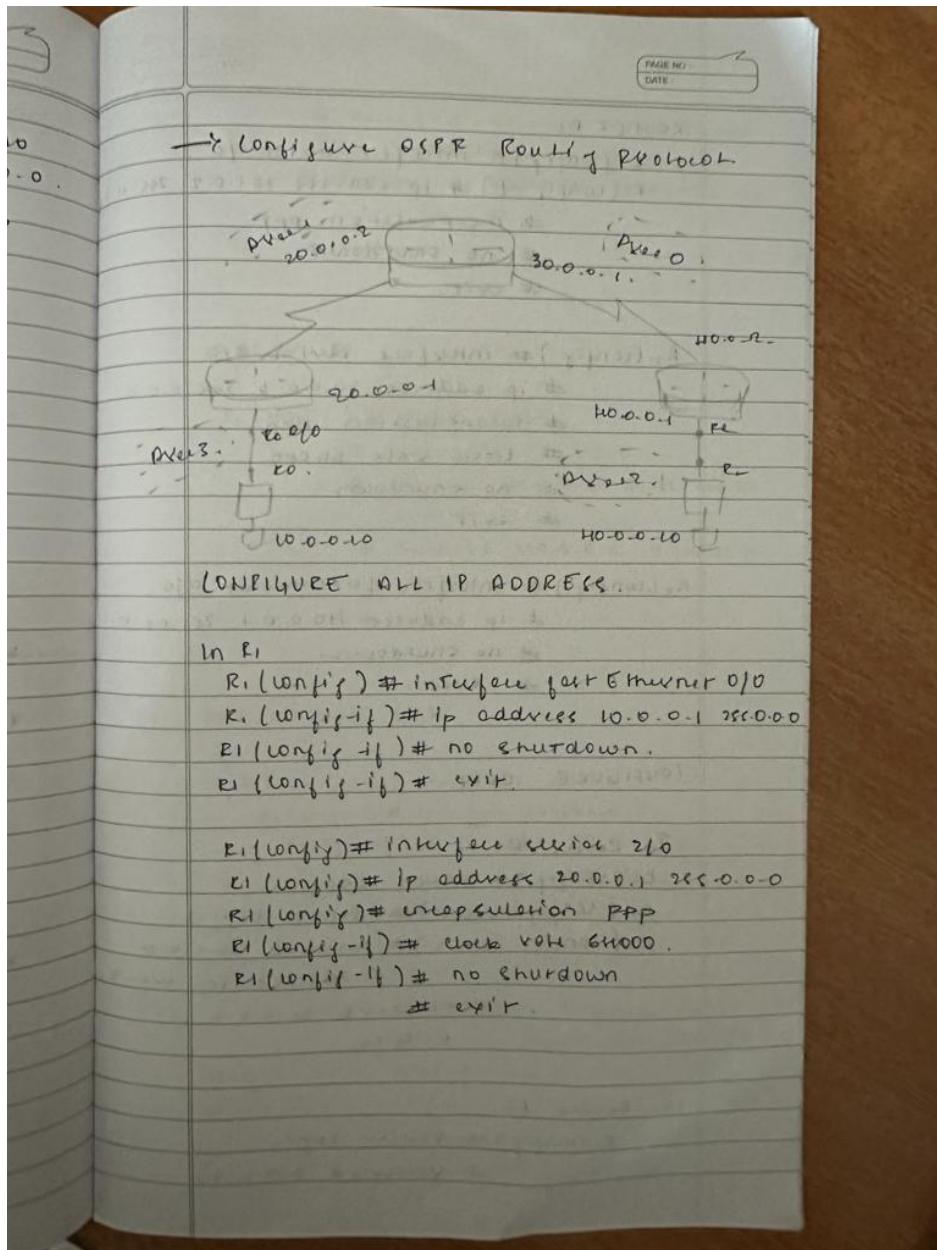
PC>
```



## WEEK 7

Configure OSPF routing protocol.

OBSERVATION:



### ROUTER R2

R2(config)# interface serial 2/0

R2(config-if)# ip address 20.0.0.2 255.0.0.0

# encapsulation PPP

# no shutdown

# exit.

R2(config)# interface serial 3/0

# ip address 30.0.0.1 255.0.0.0

# encapsulation PPP

# clock rate 64000

# no shutdown

# exit.

R2(config)# interface fastEthernet 0/0

# ip address 40.0.0.1 255.0.0.0

# no shutdown

# exit.

### CONFIGURE OSPF.

In Router R1.

R1(config)# router OSPF 1.

R1(config-router)# network 0.0.0.0

R1(config-router)# network 10.0.0.0

0.255.255.128.0.0.0.0

# network 20.0.0.0

0.255.255.0.0.0.0.0

In Router R2.

R2(config)# router OSPF 1.

# router-id 2.2.2.2

PAGE NO. \_\_\_\_\_  
 DATE: \_\_\_\_\_

2/0  
 0.0.0.2 255.0.0.0

# network 20.0.0.0 0.255.255.255  
 area 1.

# network 20.0.0.0 0.255.255.255  
 area 0.

# exit.

3/0  
 255.0.0.0

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

# exit.

Add virtual link.

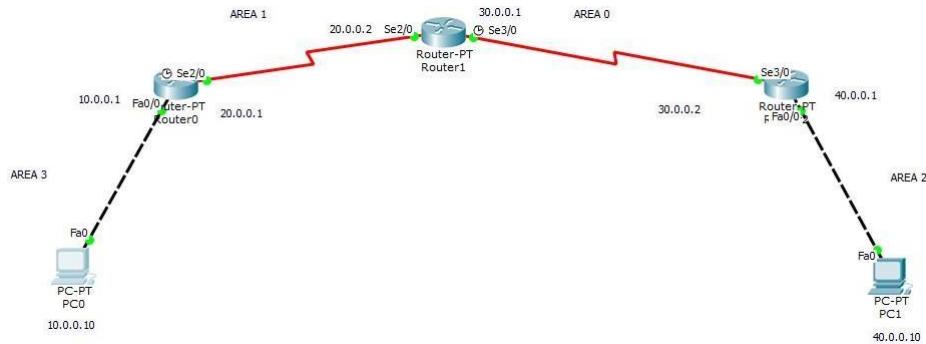
In R1:  
 R1 (config) # interface loopback 0  
 (config-if) # ip add 132.16.1.252  
 255.255.0.0.  
 # no shutdown

OSPF 1  
 10.0.0.0  
 .155. well 3.  
 .0.0

In R2:  
 R2 (config) # interface loopback 0  
 # ip add 12.16.1.254 255.255.0.0  
 # no shutdown

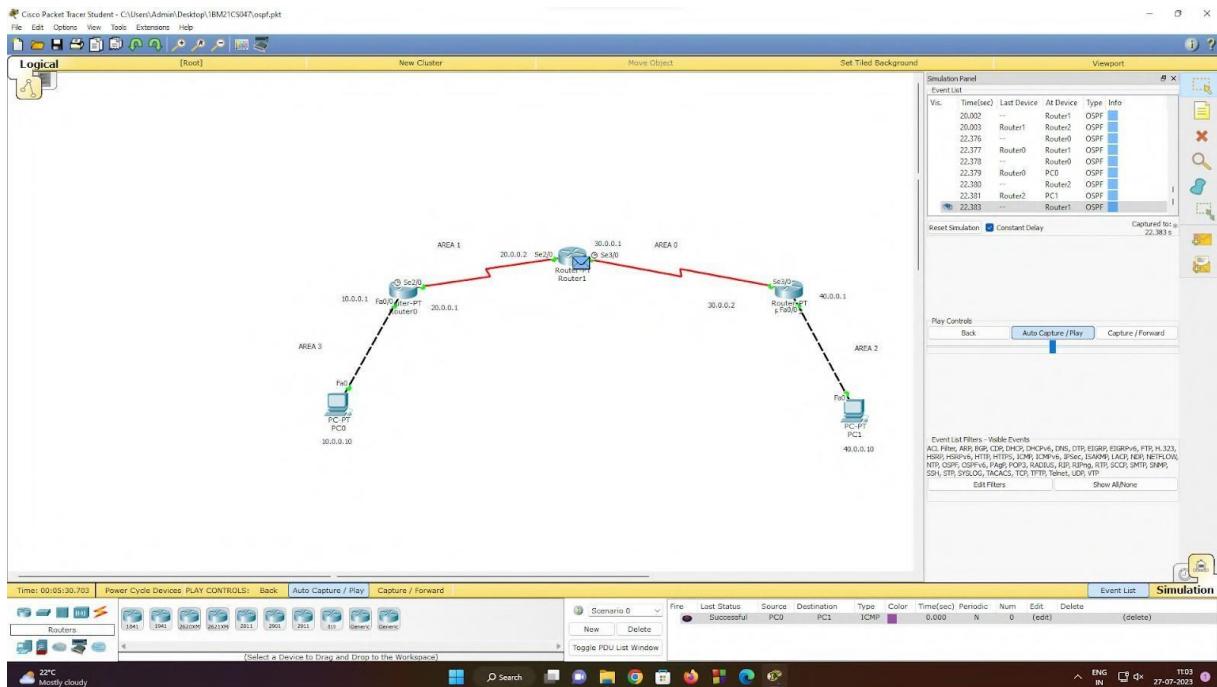
2.2,

## TOPOLOGY:



## OUTPUT:

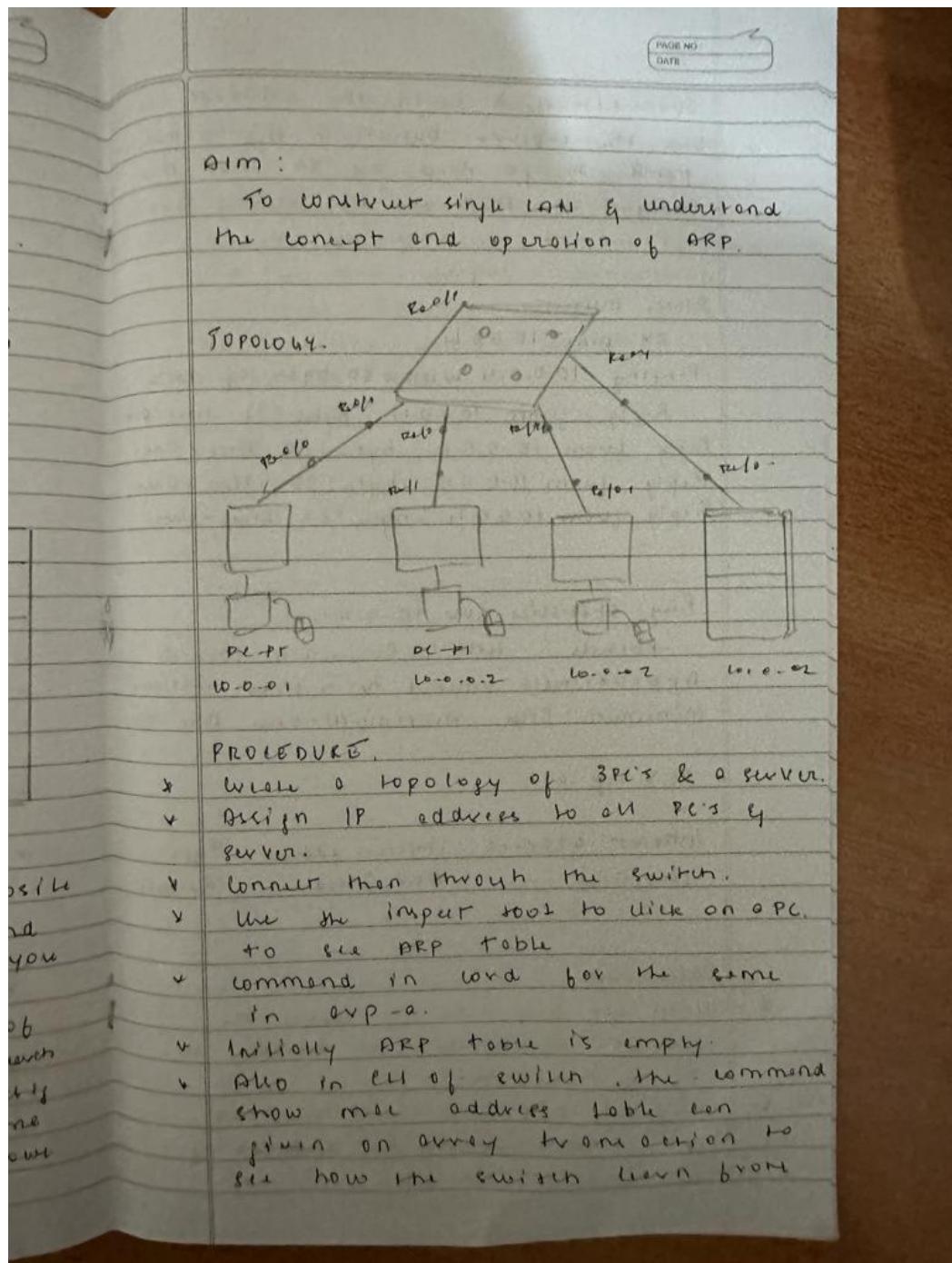
Packet Tracer PC Command Line 1.0  
PC>ping 40.0.0.10  
  
Pinging 40.0.0.10 with 32 bytes of data:  
  
Reply from 10.0.0.1: Destination host unreachable.  
  
Ping statistics for 40.0.0.10:  
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),  
  
PC>ping 40.0.0.10  
  
Pinging 40.0.0.10 with 32 bytes of data:  
  
Request timed out.  
Reply from 40.0.0.10: bytes=32 time=4ms TTL=125  
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125  
Reply from 40.0.0.10: bytes=32 time=12ms 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 = 4ms, Maximum = 12ms, Average = 7ms  
  
PC>



## WEEK 8

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

OBSERVATION:



transmission & build the address table  
use the capture button in the simulation  
panel to go step by step so that the  
changes in ARP can be clearly noted.

#### PING OUTPUT

py 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

Ping statistics for 10.0.0.4.

Packets: sent=4 received=4 loss=0%

Approximate round trip times in milliseconds:  
minimum=0ms maximum=0ms avg=0ms

py> exp-0.

Internet address	Physical address	Type
10.0.0.4	060.234a	dynamic

#### Observation -

\* When we ping & press F9 button the  
address of server is known to  
PC & vice versa.

\* When we ping between other two  
PC simultaneously the address of

PAGE NO:  
DATE:

dress Lease  
simulation  
that the  
very now.

PAGE NO:  
DATE:

with other the known.  
Every time as host need requires a  
mac address in order to send a  
packet to another host in LAN, it  
checks if ARP table to see if the  
IP to mac address translation  
address already exists. If the  
translation doesn't exist it  
performs ARP.

of dora.  
2 Time: Ons  
Time - Ons  
Time = Ons  
Time - Ons -

1001-0  
+ millions  
= Day : Ons.

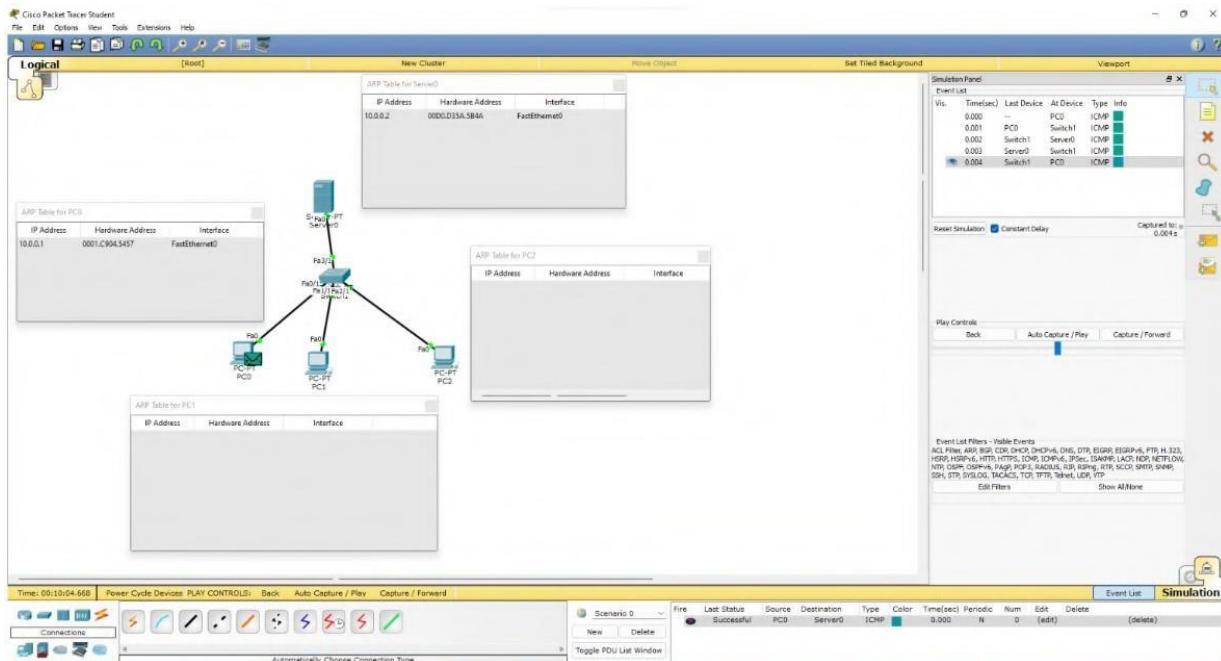
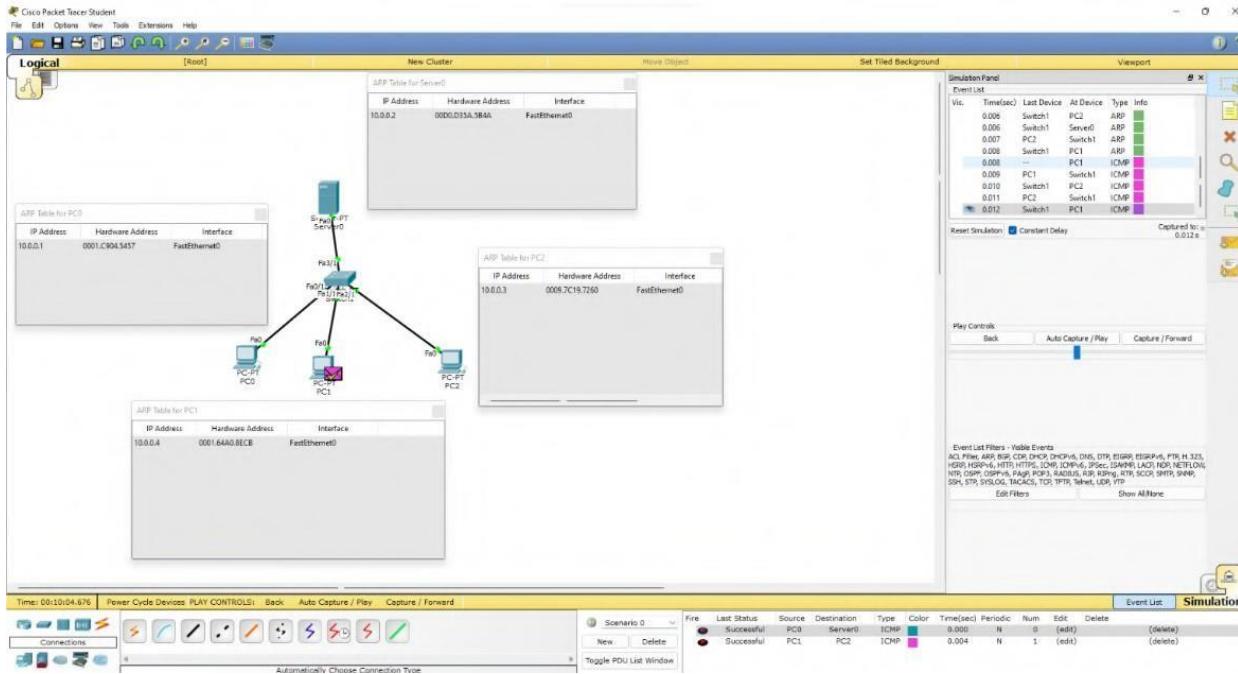
Type  
dynamic

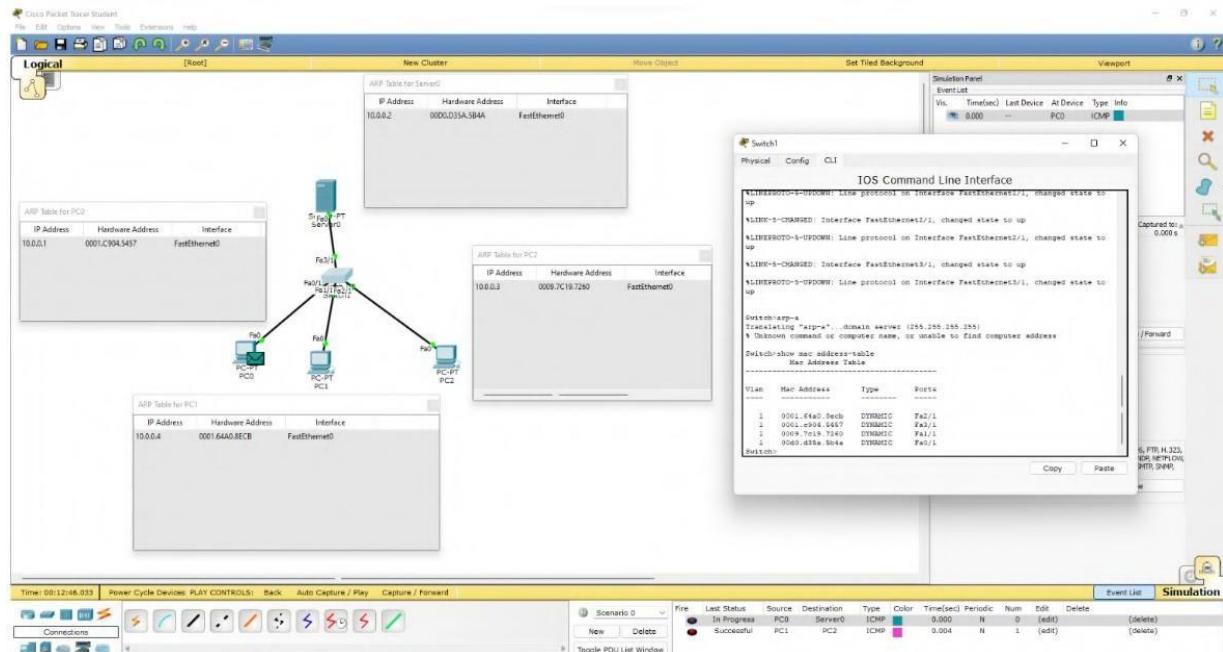
The  
on to

two  
in of

TOPOLOGY:

## OUTPUT:

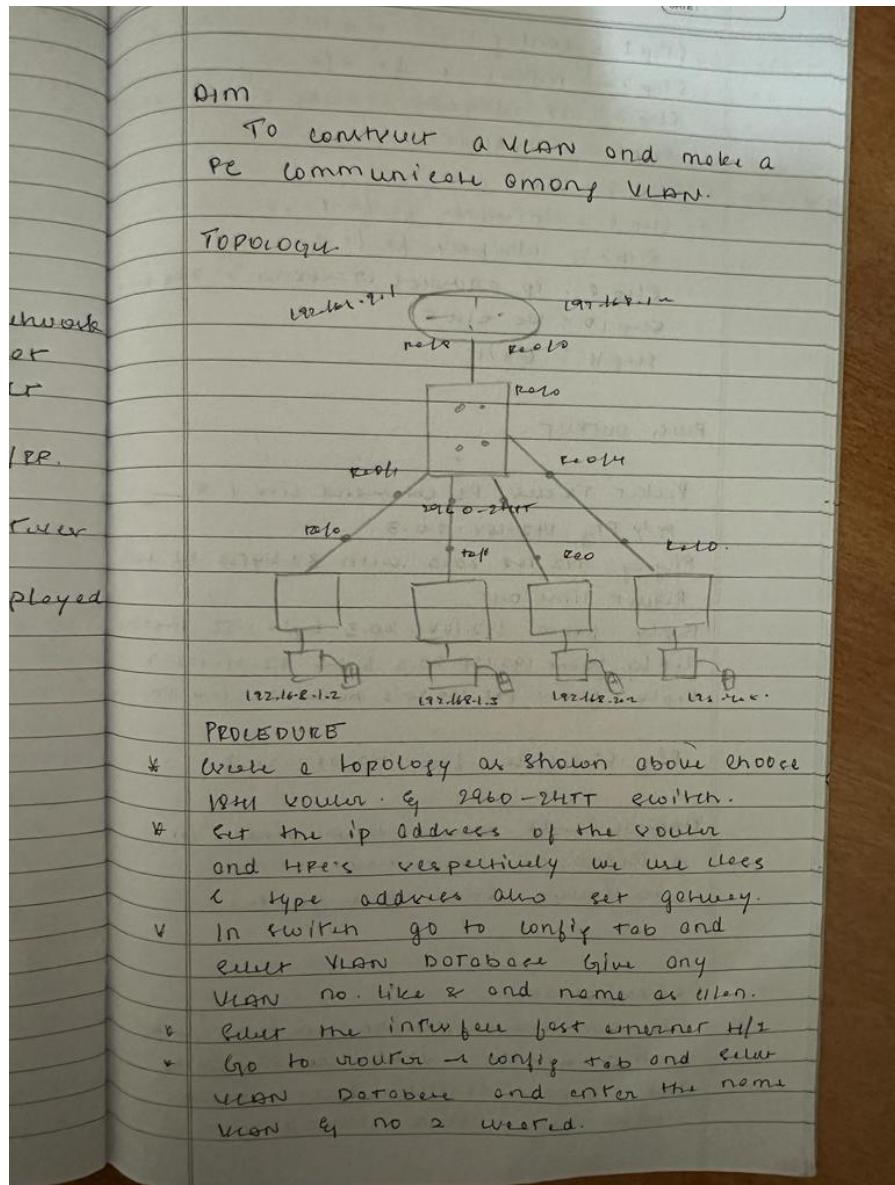




# WEEK 9

To construct a VLAN and make a pc communicate among VLAN.

## OBSERVATION:



Step 1 : config T  
Step 2 : Interface fa 0/0  
Step 3 : IP address 192.168.1.1 255.255.0  
Step 4 : NO error  
Step 5 : Exit.  
Step 6 : Interface fa 0/1  
Step 7 : Interface fa 0/1  
Step 8 : IP address 192.168.20.2 255.0.0.0  
Step 9 : NO error  
Step 10 : Exit.

#### PING OUTPUT.

Parker TVeller Pe command line 1-0  
PCY Plg 192.168.20.3.  
Plly 192.168.20.3 with 32 bytes of data.  
Request from out.  
Reply from 192.168.20.3 bytes: 32 time=82 us.  
Reply from 192.168.20.3 bytes: 32 time=82 us.  
Reply from 192.168.20.3 bytes: 32 time=82 us.

Plg statistics for 192.168.20.3

Approximate round trip time in  
milliseconds. minimum = 82  
maximum = 82 average = 82.

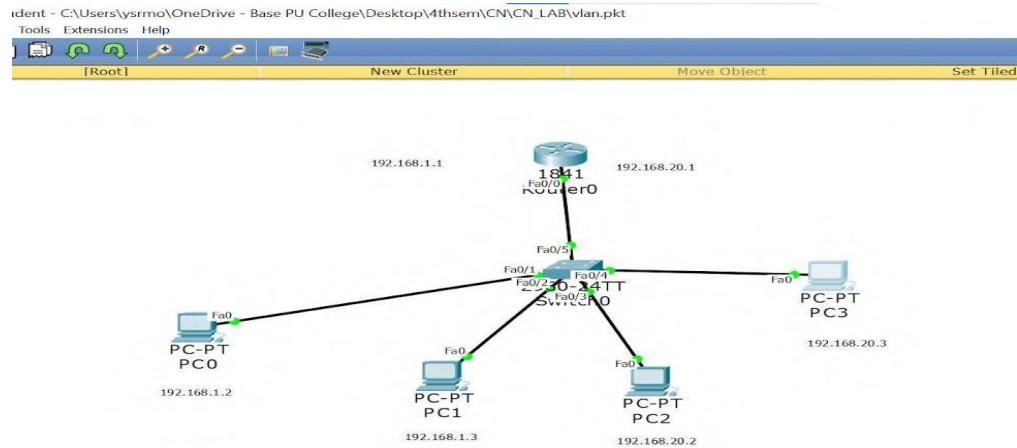
#### Observation.

- \* We can have device on one VLAN  
another another VLAN connected to  
the same switch they won't  
only hear other broadcast

- PAGE NO. \_\_\_\_\_  
DATE. \_\_\_\_\_
- ✓ Inter VLAN doesn't use IP address instead deal with subnet / uses C type address.
  - ✓ Inter VLAN routing give a flexible tool to logically subdivide their network that has potential to enhance security & performance.
- 2xx-0-0-0
- 0
- ob dars.
- 2 reasons.  
network  
telecom
- 3
- in
- 4
- 5
- 6
- 7 VLAN  
lead to



## TOPOLOGY:



## OUTPUT:

```

Packet Tracer PC Command Line 1.0
PC>ping 192.168.20.3

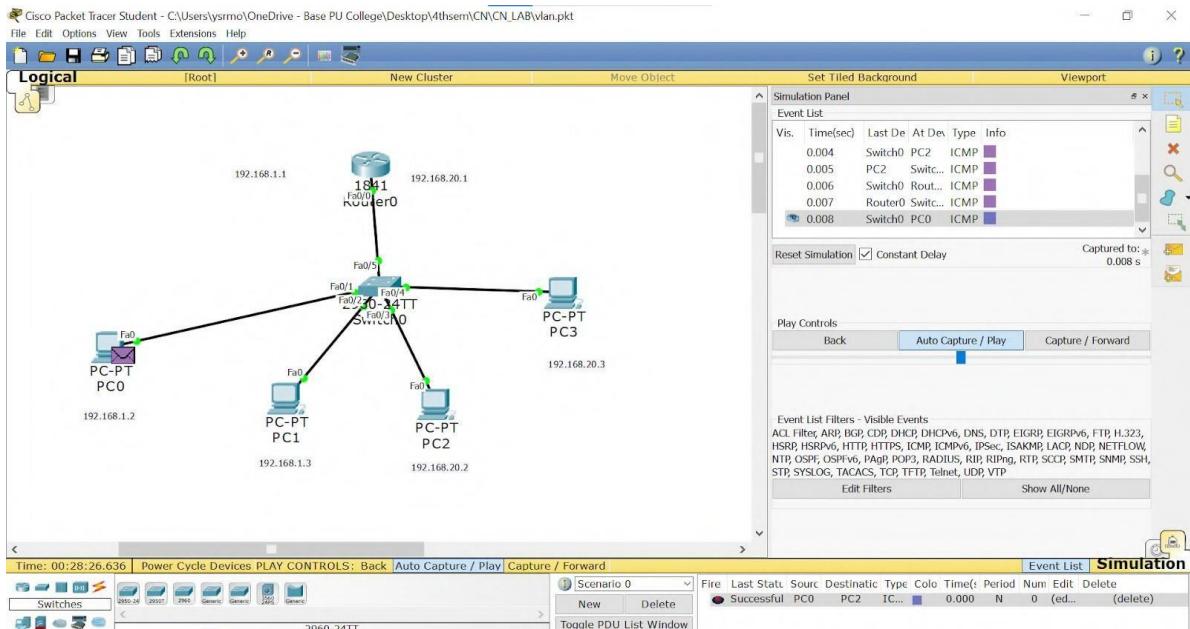
Pinging 192.168.20.3 with 32 bytes of data:

Request timed out.
Reply from 192.168.20.3: bytes=32 time=0ms TTL=127
Reply from 192.168.20.3: bytes=32 time=5ms TTL=127
Reply from 192.168.20.3: bytes=32 time=0ms TTL=127

Ping statistics for 192.168.20.3:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 5ms, Average = 1ms

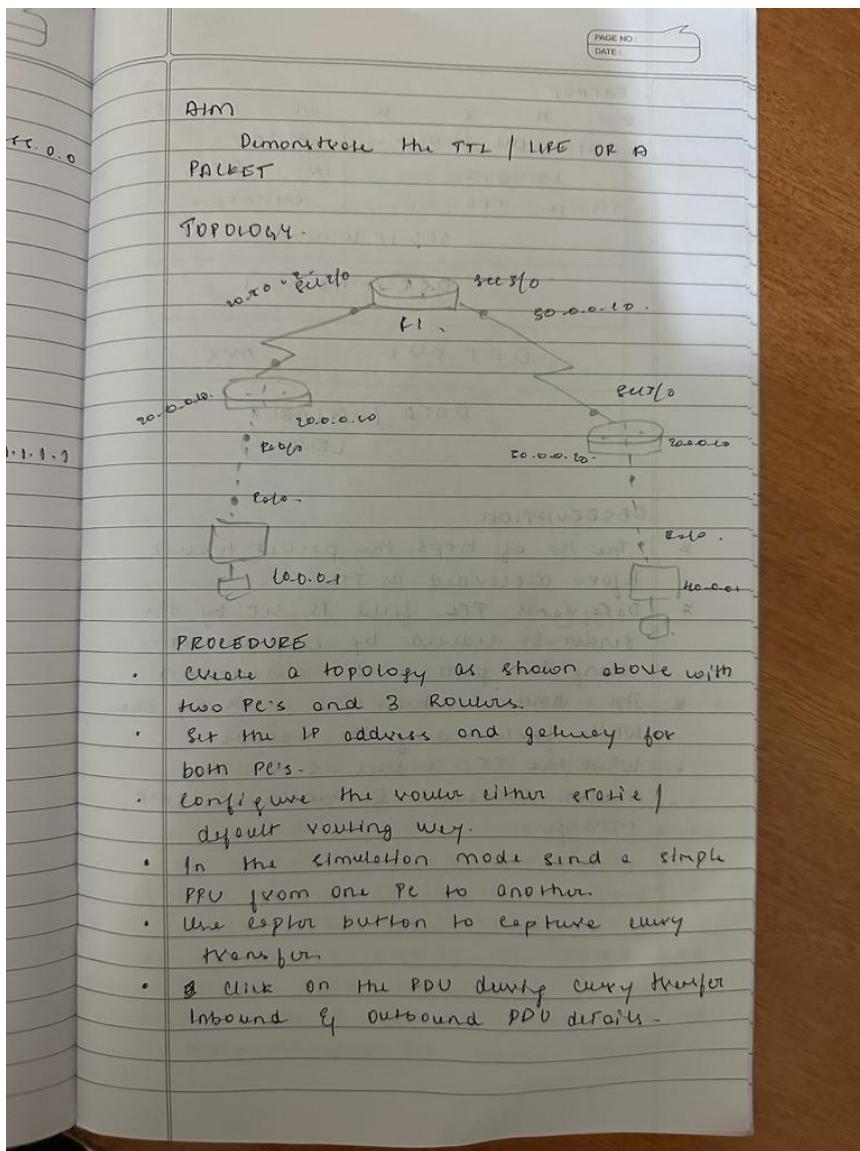
PC>

```



# WEEK 10

## Demonstrate the TTL/ Life of a Packet.OBSERVATION



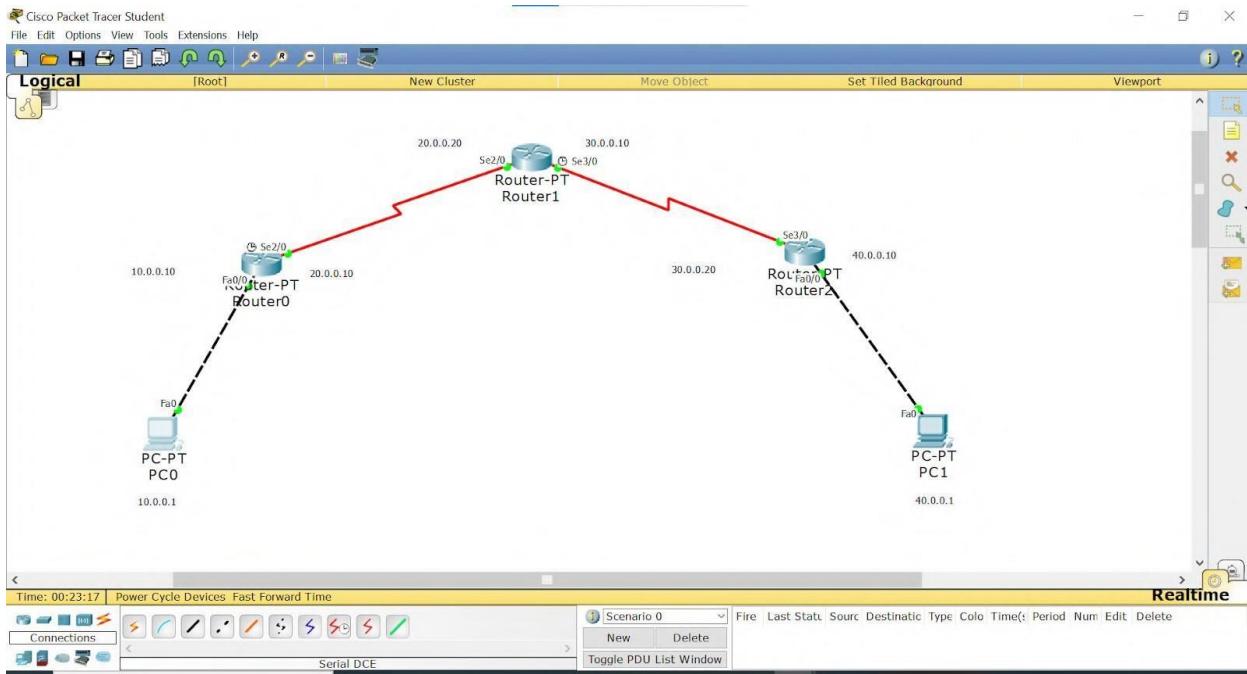
OUTPUT.

0	4	8	12	16	19	51.
	IHL	DSCP		TTL: 255	0x00	
	1D: 000				0x00	
TTL: 255	PRO: 0x1				encapsulation	
			SRC IP 10.0.0.1			
			DST IP 40.0.0.1			
				OPT 0x0	0x00	
				DATA (UPPLABLE LENGTH)		

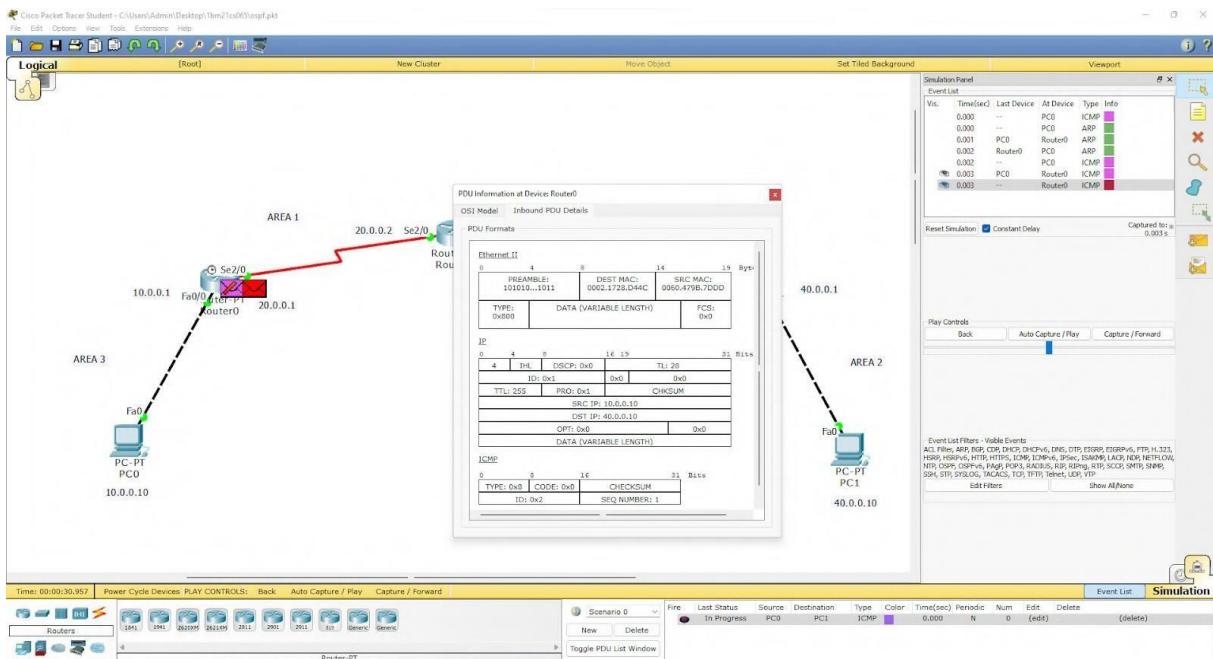
OBSERVATION.

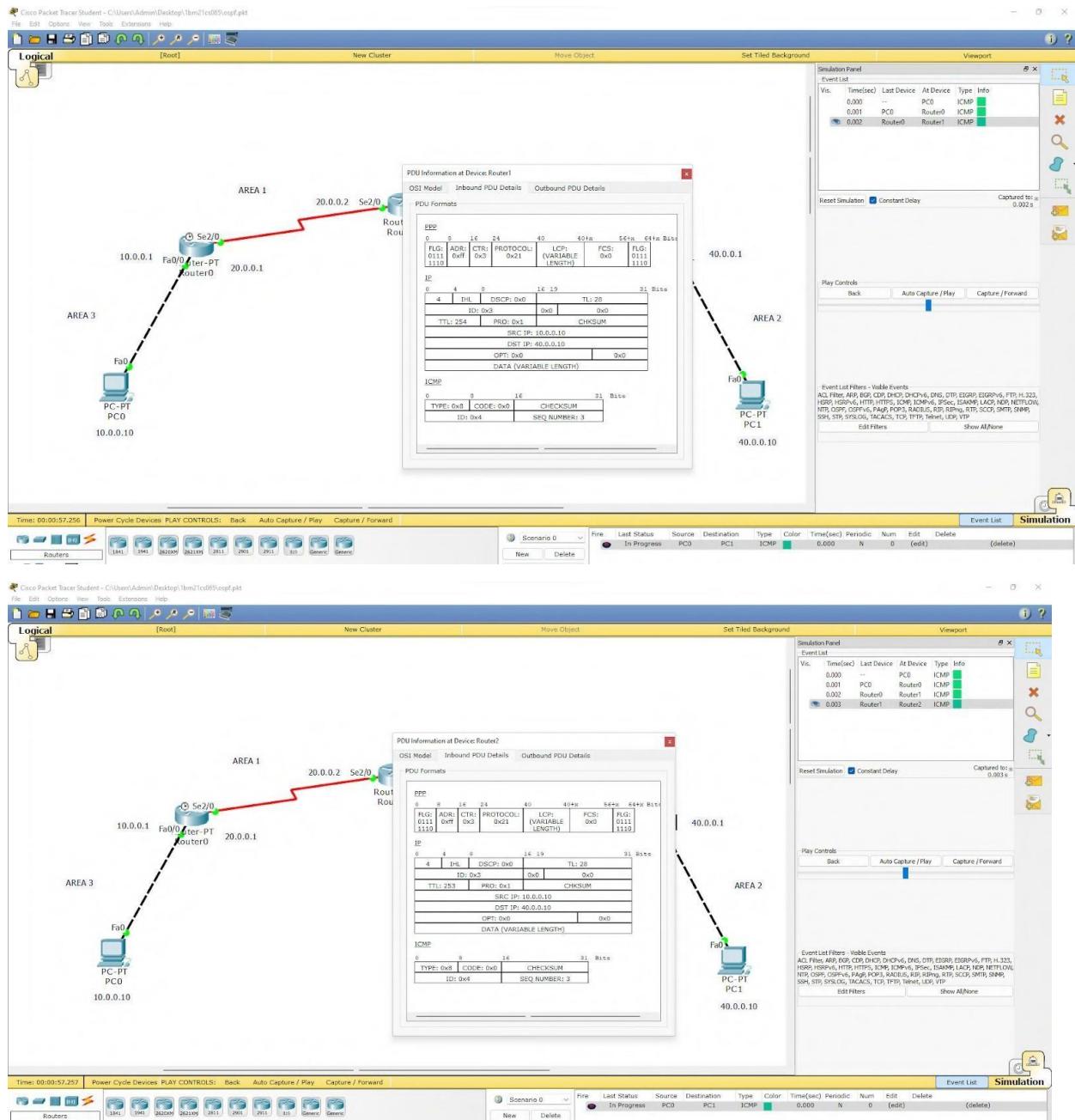
- \* The no of hops the packet travel before discarded as TTL.
- \* Datagram TTL field is set by the sender & reduced by each router along the path to its destination.
- \* The routers reduces TTL value by one while forwarding the packets.
- \* When the TTL value is 0, the router discards & sink on temp memory.

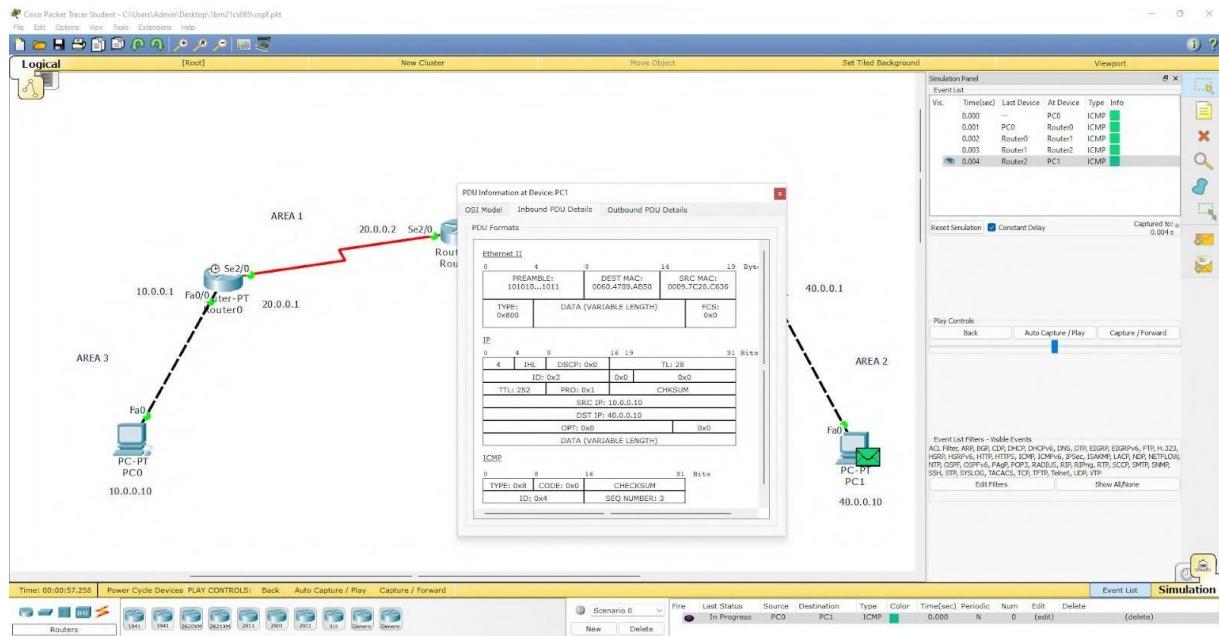
## TOPOLOGY:



## OUTPUT:



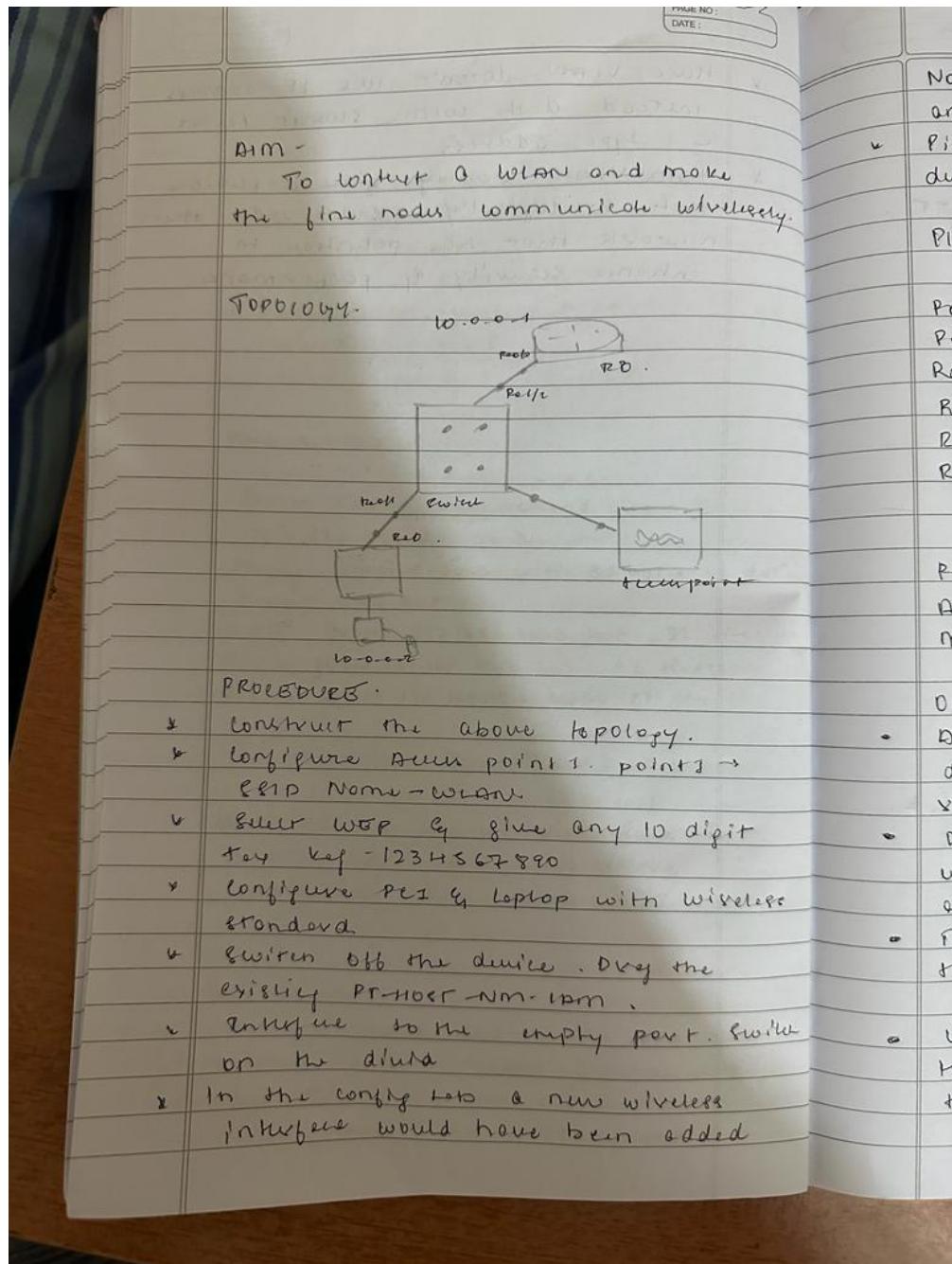




# WEEK 11

To construct a WLAN and make the nodes communicate wirelessly

## OBSERVATION:



Now configure SSID, WEP key, IP address and gateway to the device.

- Ping from every device to every other device.

#### PING OUTPUT.

Power Tether Pe command line 10.

Pe y Ping 10.0.0.3.

Pinging 10.0.0.3 with 32 bytes of data.

Request time out.

Reply from 10.0.0.3 bytes 32 Time=0ms

Reply from 10.0.0.3 bytes 32 Time=0ms

Reply from 10.0.0.3 bytes 32 Time=0ms

Ping statistics for 10.0.0.3.

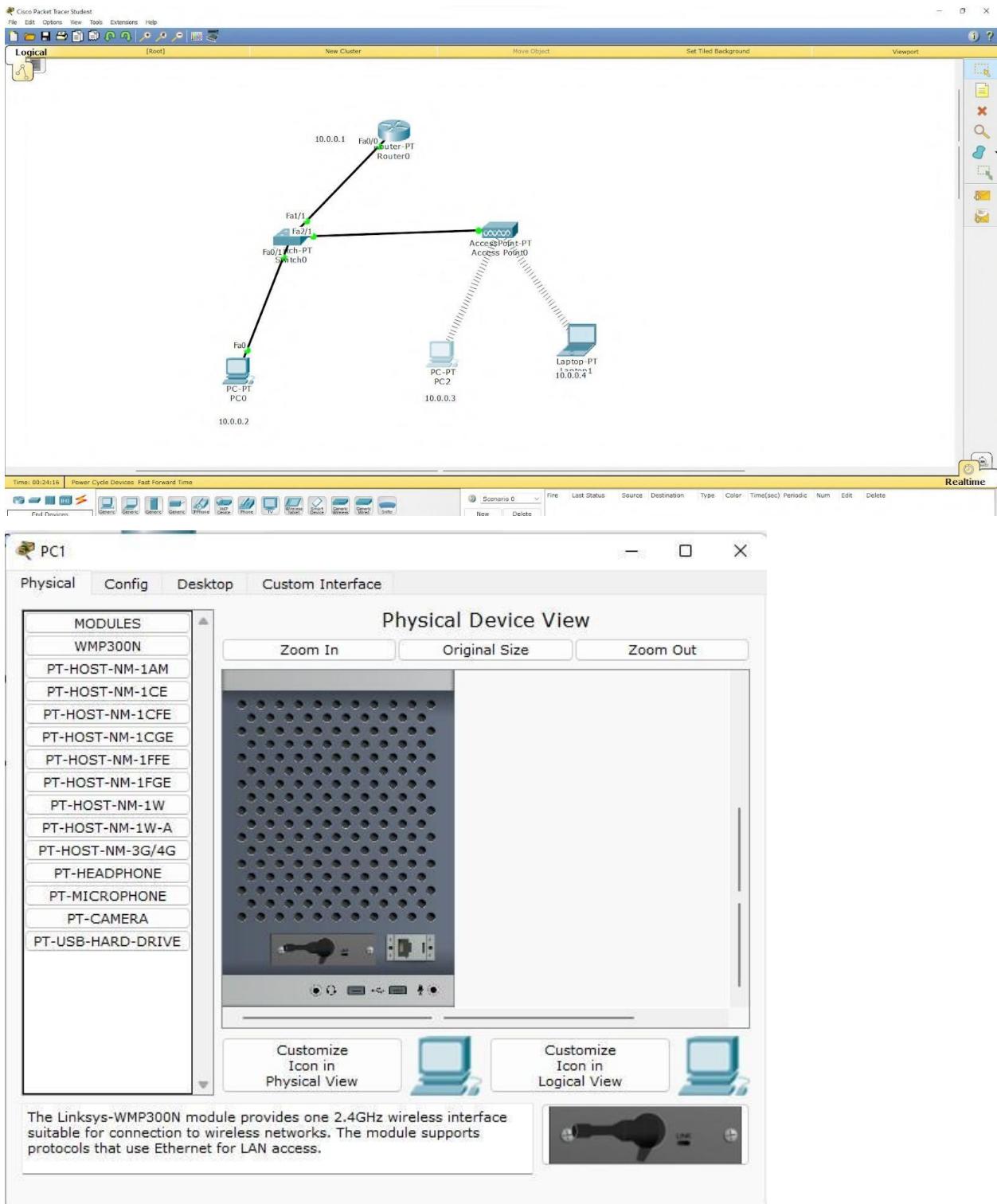
Packets: sent = 4 Received = 3 Lost = 1

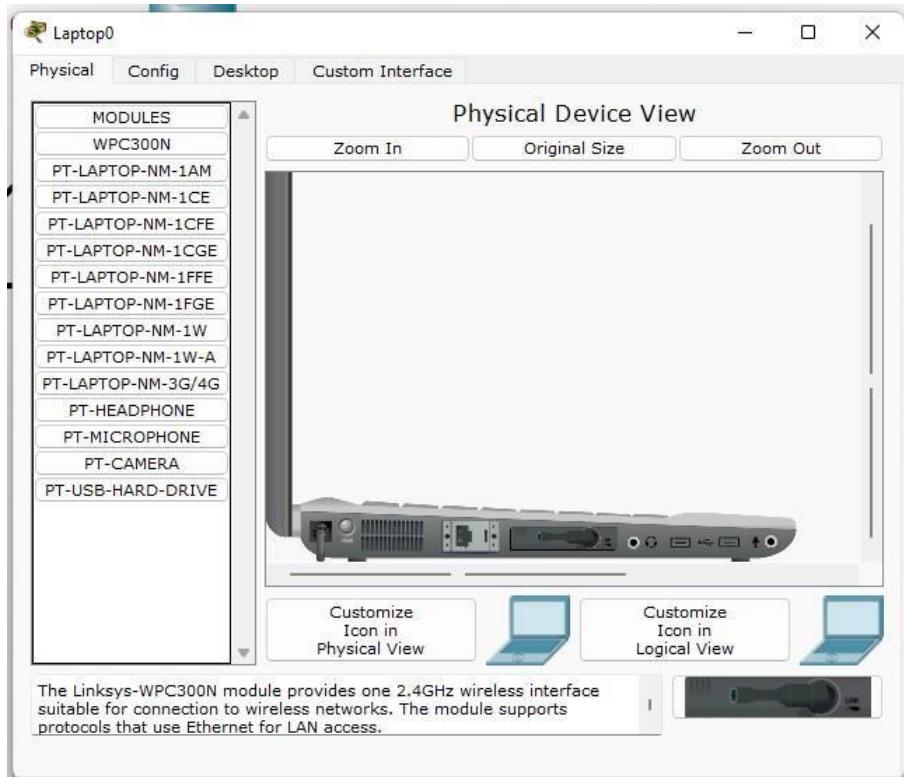
Approximate round trip times in ms  
minimum = 0ms Maximum = 1ms Average = 0ms.

#### Observation.

- WLAN is a group of connected devices that form network based on radio transmissions.
- Data sent in packets contain layer with labels & instructions. MAC address to endpoint for routing.
- The access point is the base station that serves as a hub at which other stations connect.
- With one access point we can connect to multiple devices wirelessly & transmit data.

## TOPOLOGY:





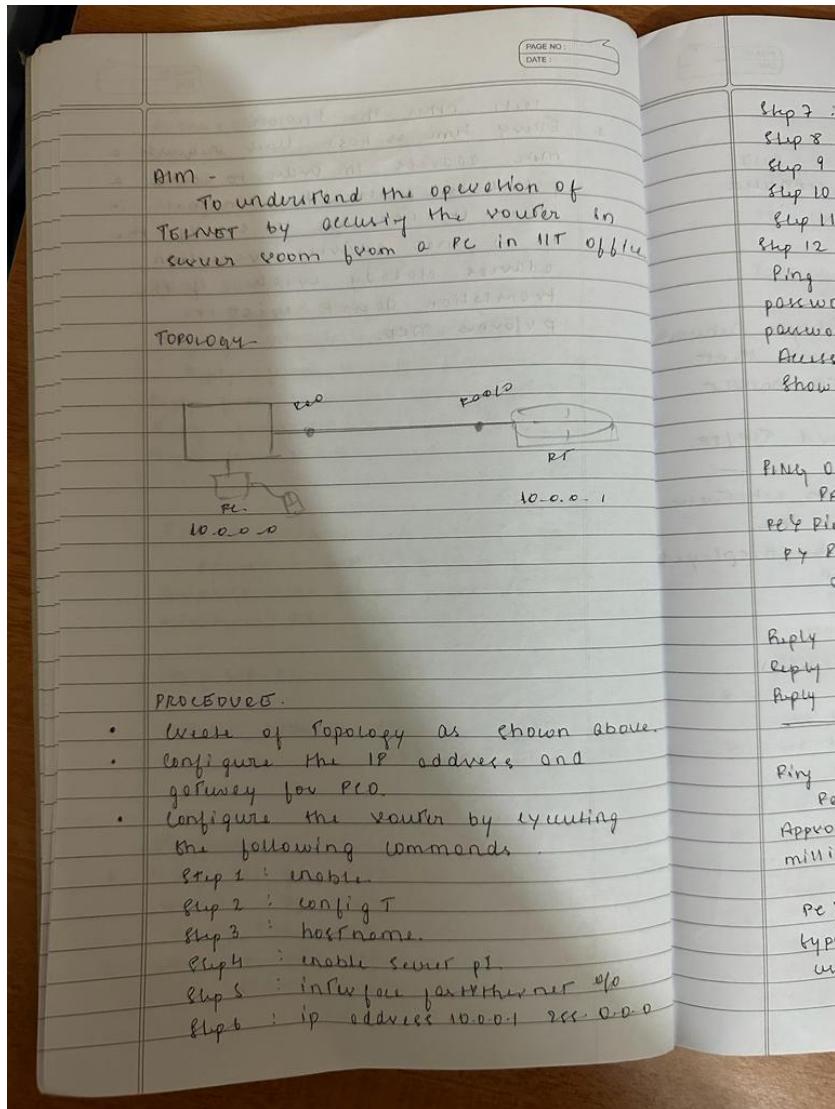
## OUTPUT:

```
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),  
PC>ping 10.0.0.3  
Pinging 10.0.0.3 with 32 bytes of data:  
Request timed out.  
Request timed out.  
Request timed out.  
Request timed out.  
Ping statistics for 10.0.0.3:  
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),  
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=21ms TTL=128  
Reply from 10.0.0.3: bytes=32 time=7ms TTL=128  
Reply from 10.0.0.3: bytes=32 time=9ms TTL=128  
Reply from 10.0.0.3: bytes=32 time=10ms 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 = 7ms, Maximum = 21ms, Average = 11ms  
PC>
```

## WEEK 12

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

## OBSERVATION:



Step 7 : No enter

Step 8 : Line VTY 0 S.

Step 9 : login

Step 10 : password p0.

Step 11 : Exit Exit

Step 12 : wr.

Ping message to Router.

password for user verification is p0

password for enable is p1.

Accessing Router eth from PC.

Show ip route

#### PING OUTPUT

PACKET from PC command line 1.0.

PC> ping 10.0.0.1

PC> ping 10.0.0.1 with 32 bytes of  
data.

Reply from 10.0.0.1 bytes=32 time=0ms.

Reply from 10.0.0.1 bytes=32 time=0ms.

Reply from 10.0.0.1 bytes=32 time=0ms.

Reply statistics for 10.0.0.1.

Packets: sent 4 received 4 lost 0

Approximate round trip times in  
milliseconds, one may = one.

PC> telnet 10.0.0.1

typing 10.0.0.1 open

user Access Verification

Password : P1.

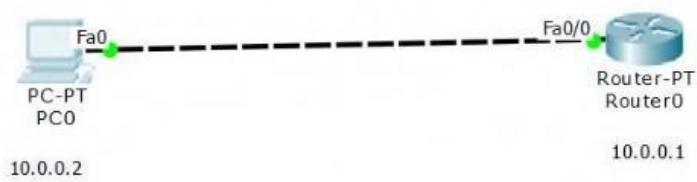
U # show ip route.

e. 10.0.0.18 directly connected  
from Ethernet 0/0.

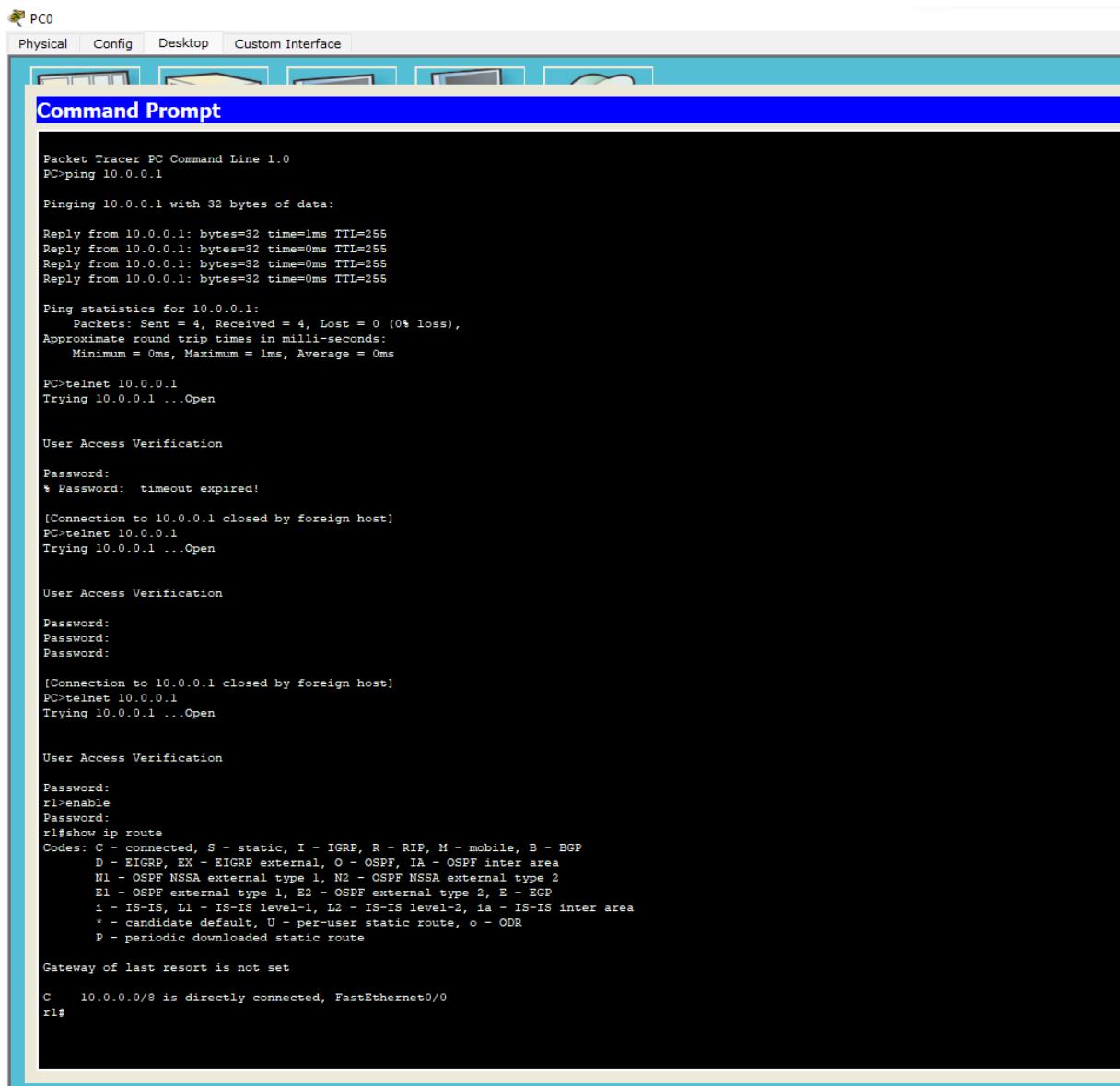
#### OBSERVATION

- \* TELNET stands for Teletype Network.  
It is a type of protocol that enables one computer to connect to the local computer.
- \* It is used a standard TCP/IP protocol by ISO.  
During TELNET operation, whatever is being performed on the remote computer will be displayed by the local computer.

## TOPOLOGY:



## OUTPUT:



The screenshot shows a "Command Prompt" window within the Packet Tracer interface. The window title is "Command Prompt". The content of the window is as follows:

```
Packet Tracer PC Command Line 1.0
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=1ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255

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 = 1ms, Average = 0ms

PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open

User Access Verification

Password:
* Password: timeout expired!

[Connection to 10.0.0.1 closed by foreign host]
PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open

User Access Verification

Password:
Password:
Password:

[Connection to 10.0.0.1 closed by foreign host]
PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open

User Access Verification

Password:
rl>enable
Password:
rl#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
rl#
```

## WEEK 13

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

CODE:

```
#include<stdio.h>
int arr[17];

void xor(int x[], int y[])
{
    int k=0;
    for(int i=1;i<16;i++)
    {
        if(x[i]==y[i])
            arr[k++]=0;
        else
            arr[i]=1;
    }
}

void main()
{
    int dd[17],div[33],ze[17],i,k;

    printf("Enter the dataword \n");
    for(i=0;i<17;i++)
        scanf("%d",&div[i]);

    for(i=i;i<33;i++)
        div[i]=0;

    for(i=0;i<17;i++)
        ze[i]=0;
    printf("Enter dividend \n");
```

```

for(i=0;i<17;i++)
    scanf("%d",&dd[i]);

i=0;
k=0;
for(i=i;i<17;i++)
    arr[k++]=div[i];
while(i<33)
{
    if(arr[0]==0)
        xor(arr,ze);
    else
        xor(arr,dd);

    arr[16]=div[i++];

}
k=0;
for(i=17;i<33;i++)
    div[i]=arr[k++];
printf("Codeword: ");
for(i=0;i<33;i++)
    printf("%d",div[i]);

for(i=0;i<17;i++)
    arr[i]=0;

printf("\nAt receiver end \n");

k=0;
for(i=i;i<17;i++)
    arr[k++]=div[i];
while(i<33)
{

```

```

if(arr[0]==0)
    xor(arr,ze);
else
    xor(arr,dd);

arr[16]=div[i++];

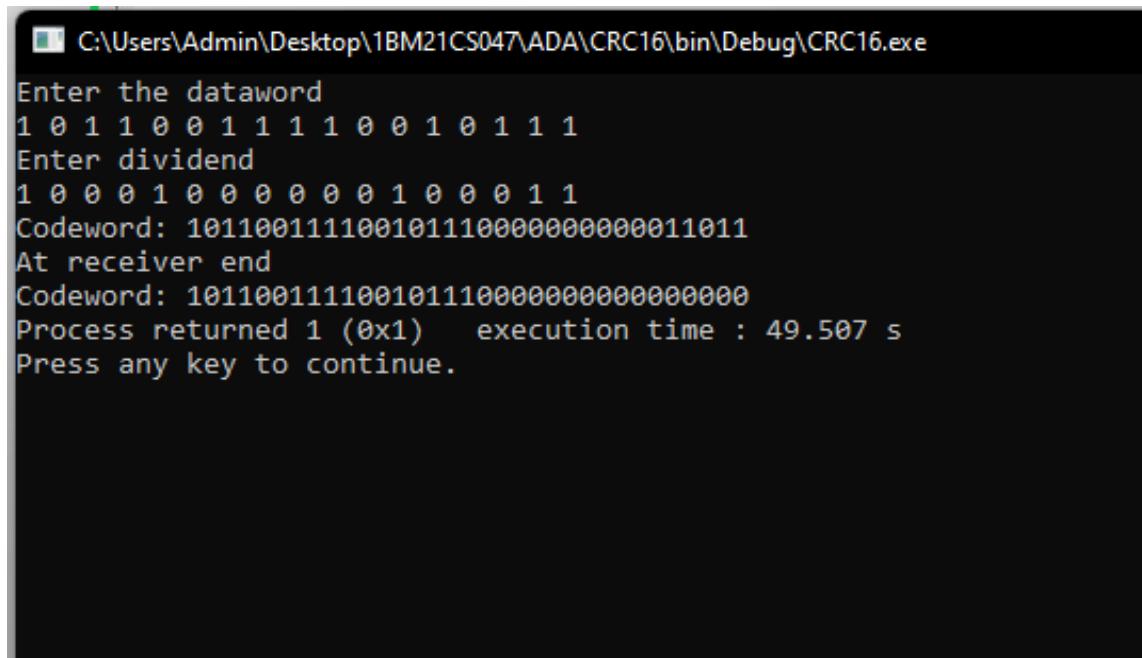
}

k=0;
for(i=17;i<33;i++)
    div[i]=arr[k++];

printf("Codeword: ");
for(i=0;i<33;i++)
    printf("%d",div[i]);
}

```

## OUTPUT:



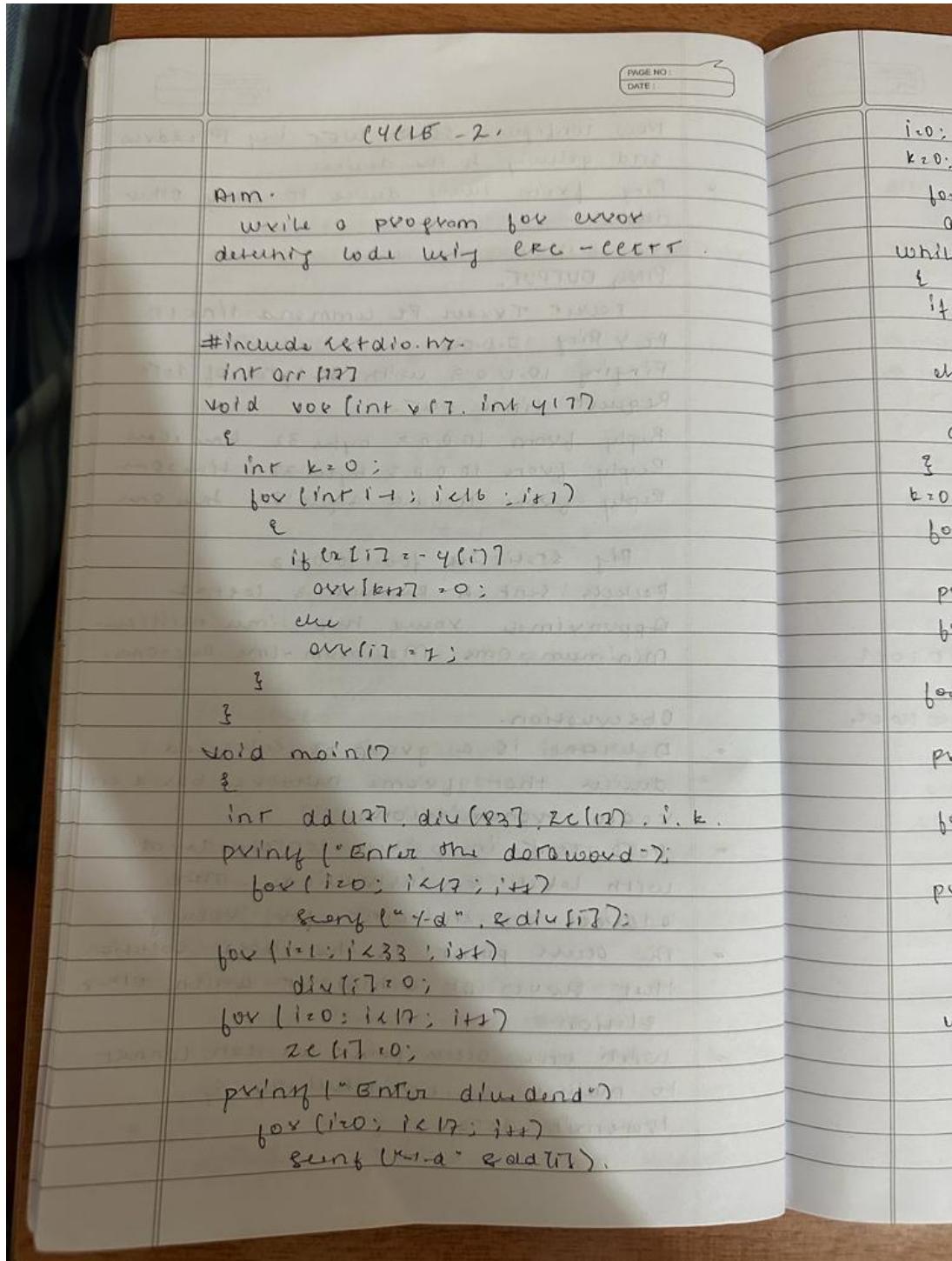
```

C:\Users\Admin\Desktop\1BM21CS047\ADA\CRC16\bin\Debug\CRC16.exe

Enter the dataword
1 0 1 1 0 0 1 1 1 1 0 0 1 0 1 1 1
Enter dividend
1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 1
Codeword: 101100111100101110000000000011011
At receiver end
Codeword: 10110011110010111000000000000000
Process returned 1 (0x1)   execution time : 49.507 s
Press any key to continue.

```

## OBSERVATION:



PAGE NO. \_\_\_\_\_  
 DATE: \_\_\_\_\_

```

i=0;
k=0;
for (i=0; i<17; i++)
  arr[k+i] = div[i];
while (i<33)
{
  if (arr[0] == 0)
    XOR (arr, 2);
  else
    XOR (arr, dd);
  arr[16] = div[i];
}
k=0;
for (i=0; i<33; i++)
  div[i] = arr[k+i];
printf (" codeword");
for (i=0; i<33; i++)
  printf (" -d ", div[i]);
for (i=0; i<17; i++)
  arr[i] = 0;
printf (" AT receiver end ");
k=0;
for (i=17; i<17; i++)
  arr[k+i] = div[i];
while (i<33)
{
  if (arr[0] == 0)
    XOR (arr, 2);
  else
}
  
```

arr(107 .div [114])

9

$$k=0$$

```
for(i<10 ; i<33 ; i+1)
```

Q: You're over [ing]

```
print("wordword");
```

for(i=0; i<33; i++)

privately for a. div (17);

乙

## OUTPUT.

Enter the doorway.

101100 Moon, 0-11

Enter the auditor.

16 n m o o m i

## Wadsworth

... no idea how many more.

Dr. Rutherford (University of Alberta)

10 000 000 000 000 000

## WEEK 14

Write a program for congestion control using Leaky bucket algorithm.

CODE:

```
#include <stdio.h>
#include <stdlib.h> // Include this for the rand() function
int main()
{
    int buckets, outlets, k = 1, num, remaining;
    printf("Enter Bucket size and outstream size\n");
    scanf("%d %d", &buckets, &outlets);
    remaining = buckets;
    while (k)
    {
        num = rand() % 1000; // Generate a random number between 0 and 999
        if (num < remaining)
        {
            remaining = remaining - num;
            printf("Packet of %d bytes accepted\n", num); // Added missing variable
        }
        else
        {
            printf("Packet of %d bytes is discarded\n", num);
        }
        if (buckets - remaining > outlets)
        {
            remaining += outlets; // Fixed the calculation
        }
        else
            remaining = buckets;
        printf("Remaining bytes: %d \n", remaining);
        printf("If you want to stop input, press 0, otherwise, press 1\n");
        scanf("%d", &k);
    }
}
```

```

}

while (remaining < buckets) // Fixed the condition
{
    if (buckets - remaining > outlets)
    {
        remaining += outlets; // Fixed the calculation
    }
    else
        remaining = buckets;
    printf("Remaining bytes: %d \n", remaining);
}
return 0; // Added a return statement to indicate successful completion
}

```

## 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

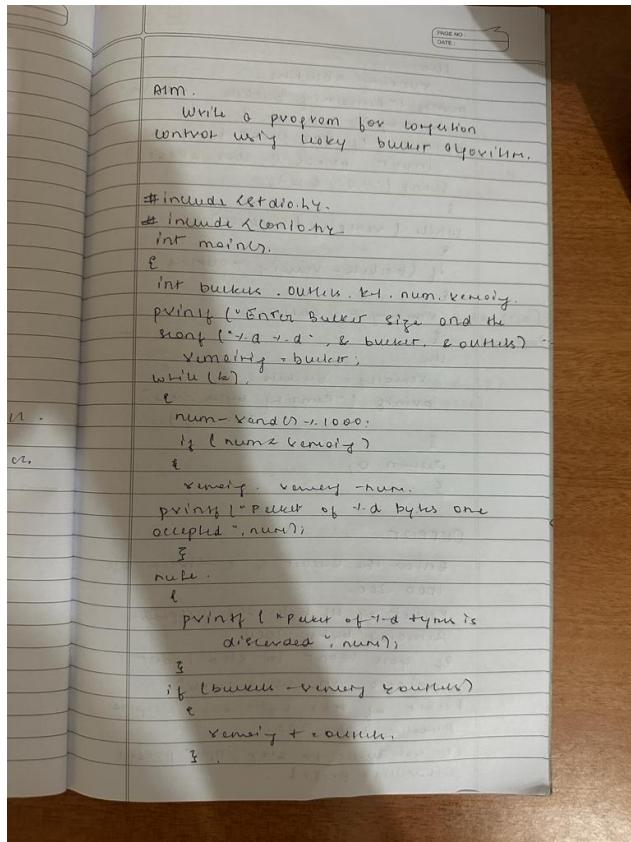
```

```

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> █

```

## OBSERVATION:



```
the  
removing = buckets;  
pointy + removing buckets; -fd in  
removing );  
printf (" If you want to stop  
input press 0, else press 1)  
scanf ("%d", &k);  
{  
while (removing < buckets)  
{  
if (buckets - removing > outlets)  
{  
removing + = outlets;  
}  
else  
removing = buckets;  
pointy (" Removing bytes : -d );  
removing );  
scanf (0);  
}
```

## OUTPUT.

Enter the Bucket of Outstream size  
1000 . 200.

Packet of 101 bytes are accepted.  
Remaining bytes : 1000.

If you want to stop input  
press 0, otherwise press 1  
Packet of 100 bytes are accepted  
Remaining bytes 783.

If you want to stop input press 0  
otherwise press 1 .

## WEEK 15

Using TCP/IP 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.

CODE:

ClientTCP.py

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

ServerTCP.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 ("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    file = open(sentence, "r")
    l = file.read(1024)
    connectionSocket.send(l.encode())
```

```

print ("\nSent contents of " + sentence)
file.close()
connectionSocket.close()

```

## OUTPUT:

The image shows two separate Python IDLE shells running simultaneously. Both windows have the title "IDLE Shell 3.11.4". The left window contains the code for the ClientTCP.py program, which includes imports for socket, defines a serverName and serverPort, creates a serverSocket, and enters a loop to receive data from a connectionSocket, encode it, and then send it back. The right window contains the code for the ServerTCP.py program, which defines a serverName and serverPort, creates a serverSocket, binds it to the address, and listens for connections. It then enters a loop to receive data from a connectionSocket, print the received sentence, and then close both the file and the connectionSocket.

```

File Edit Shell Debug Options Window Help
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> ===== RESTART: C:\Users\Admin\Desktop\lbtm2lcs065\ClientTCP.py =====
Enter file name:ServerTCP.py

From server:

from socket import *
serverName="127.0.0.1"
serverPort=12000
serverSocket=socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print("The server is ready to receive")
    connectionSocket,addr=serverSocket.accept()
    sentence=connectionSocket.recv(1024).decode()
    file=open(sentence,"r")
    l=file.read(1024)
    connectionSocket.send(l.encode())
    print('\'nSent contents of' + sentence)
    file.close()
    connectionSocket.close()

>>>

File Edit Shell Debug Options Window Help
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> ===== RESTART: C:\Users\Admin\Desktop\lbtm2lcs065\ServerTCP.py =====
The server is ready to receive
Sent contents ofServerTCP.py
The server is ready to receive

```

## OBSERVATION:

PAGE NO.: \_\_\_\_\_  
DATE: \_\_\_\_\_

AIM: Use TCP / IP sockets. Write a client server program to make client send the filename and the server to send back the contents of the required file if present.

Client TCP.py.

```
from socket import *
server_name = "127.0.0.1"
server_port = 12000
client_socket = socket(AF_INET, SOCK_STREAM)
client_socket.connect((server_name, server_port))
sentence = input("Enter file name")
client_socket.send(sentence.encode())
file_contents = client_socket.recv(1024)
print(file_contents)
print(file_contents)
client_socket.close()
```

ServerTCP.py-

```
from socket import *
server_name = "127.0.0.1"
server_port = 12000
server_socket = socket(AF_INET, SOCK_STREAM)
server_socket.bind(("", server_port))
server_socket.listen(1)
while True:
    print("The server is ready to receive")
    connection_socket, addr = server_socket.accept()
    sentence = connection_socket.recv(1024)
    file = open(sentence, "r")
    l = file.read(1024)
    connection_socket.send(l)
    file.close()
    connection_socket.close()
```





## **WEEK 16**

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:

ClientUDP.py

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

```

con=file.read(2048)
serverSocket.sendto(bytes(con,"utf-8"),clientAddress)
print ("\nSent contents of ", end = " ")
print (sentence)
# for i in sentence:
# print (str(i), end = " ")
file.close()

```

## OUTPUT:

The image shows two windows of the IDLE Shell 3.11.4 interface. Both windows have the title "IDLE Shell 3.11.4".

**Left Window (Client Side):**

```

File Edit Shell Debug Options Window Help
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> = RESTART: C:/Users/Admin/Desktop/lbm2lcs065/ClientUDP.py
Enter file name: ServerUDP.py
Reply from Server:

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)
    sentence = sentence.decode("utf-8")
    file=open(sentence, "r")
    con=file.read(2048)
    serverSocket.sendto(bytes(con,"utf-8"),clientAddress)
    print ("\nSent contents of ", end = " ")
    print (sentence)
    # for i in sentence:
    # print (str(i), end = ' ')
    file.close()

>>>

```

**Right Window (Server Side):**

```

File Edit Shell Debug Options Window Help
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> = RESTART: C:/Users/Admin/Desktop/lbm2lcs065/ServerUDP.py
The server is ready to receive
Sent contents of ServerUDP.py
|

```

## OBSERVATION

(n)  
new)

Aim:  
Using UDP socket . write a client server program to make client sending the file name and the server send back the contents of the requested file if present.

Client UPP.py.  
from socket import \*  
ServerName = "122.0.0.1"  
ServerPort = 12000.  
ClientSocket = socket(AF\_INET, SOCK\_DGRAM)  
ServerAddress = input("Enter the file name").  
ClientSocket.sendto (bytes(ServerAddress, "UTF-8"),  
file contents, ServerAddress, ClientSocket).  
recvfrom (2048)  
print ("Reply from server\n")  
print (filecontents.decode("utf-8"))  
for i in filecontents:  
# print (str(i)-end".  
ClientSocket.close().  
ClientSocket = socket()  
ServerPort = 12000.  
from socket import \*  
ServerPort = 12000.  
ServerSocket = socket(AF\_INET, SOCK\_DGRAM)  
ServerSocket.bind ((ServerName, ServerPort))  
print ("The server is ready to receive")  
while 1:  
Request, ClientAddress = ServerSocket.  
recvfrom (2048)

```
sentence = sentence . mode [ "UTF-8" ]  
file = open ( sentence , "r" )  
con . file . read ( 8000 )  
server . socket . sendto ( bytes ( con . "HTTP/  
client address )  
print ( "In the content of " , end = " " )  
print ( sentence )  
# for i in sentence  
# print ( str ( i ) + end )  
file . close ( )
```

# WEEK 17

## Tool Exploration -Wireshark

### OBSERVATION:

