VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

COMPUTER NETWORKS

Submitted by

DHRAVYA M (1BM21CS056)

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 **DHRAVYA M (1BM21CS056),** 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.

Proff Swathi Sridharan
Assistant Professor
Department of CSE
Bengaluru

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

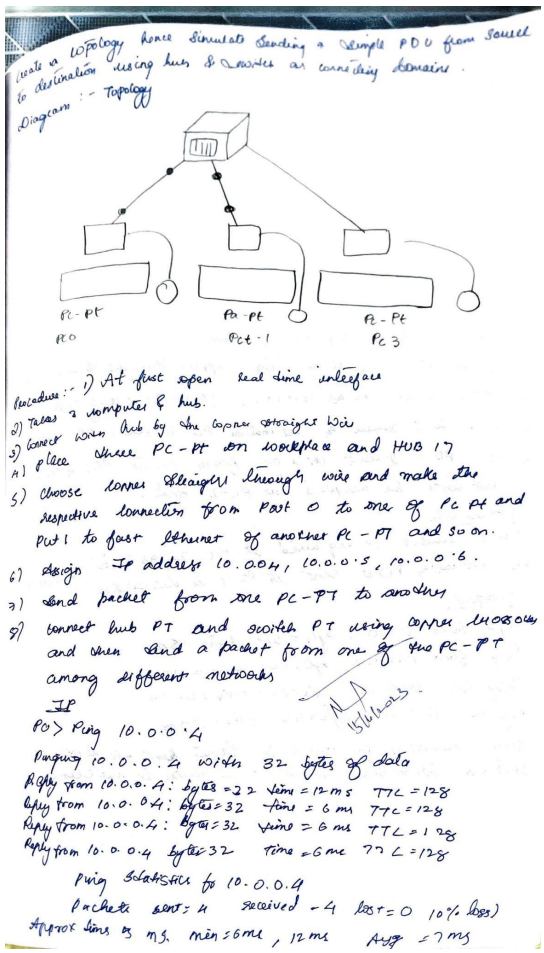
`

2

Index

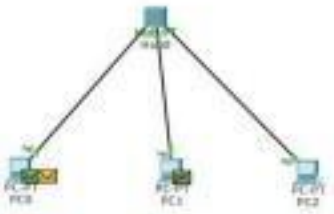
EXPERIMENT-1

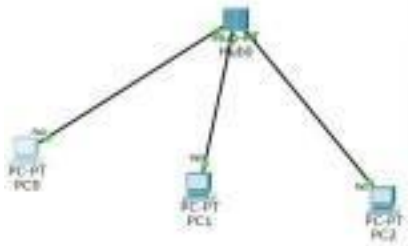
Q) Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message.

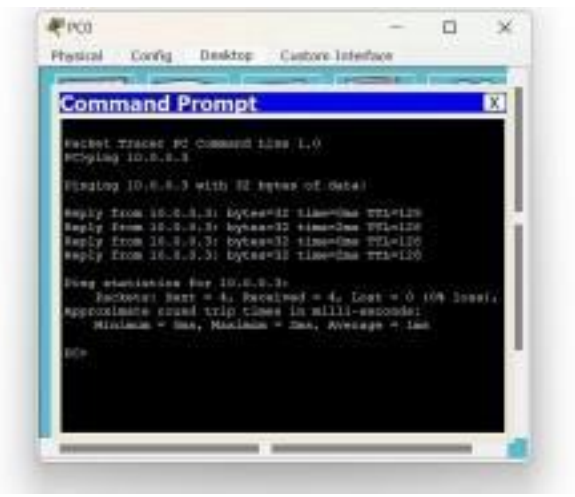


TOPOLOGY & OUTPUT

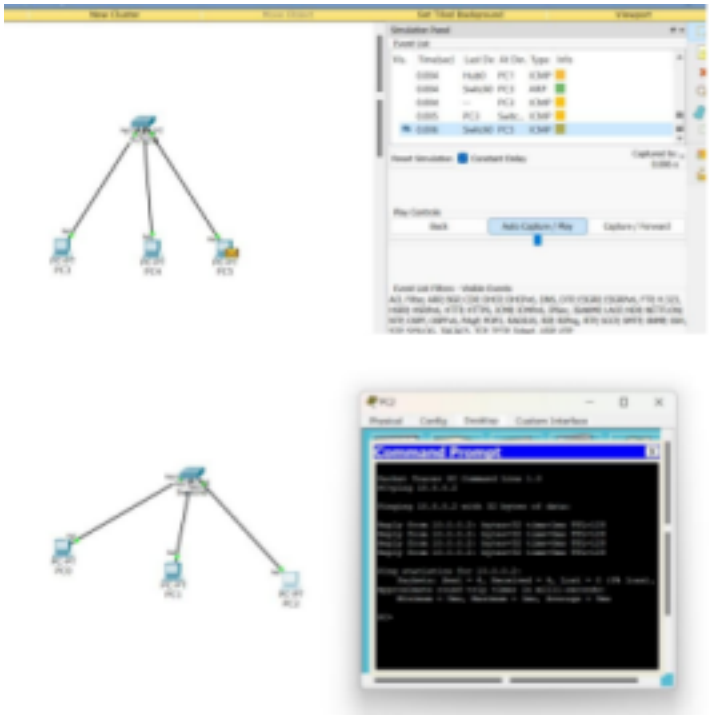
1. Hub and PCs



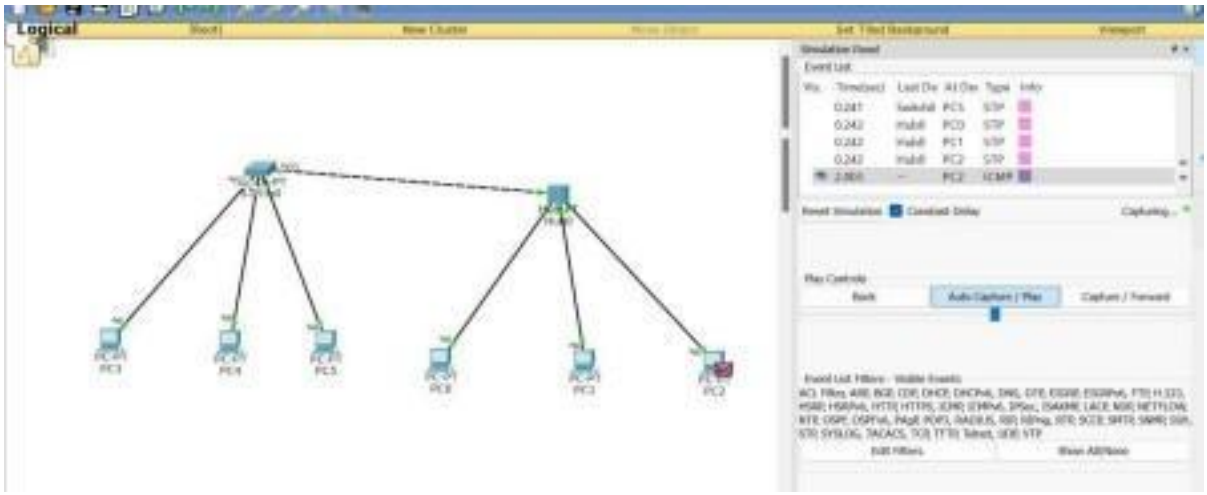


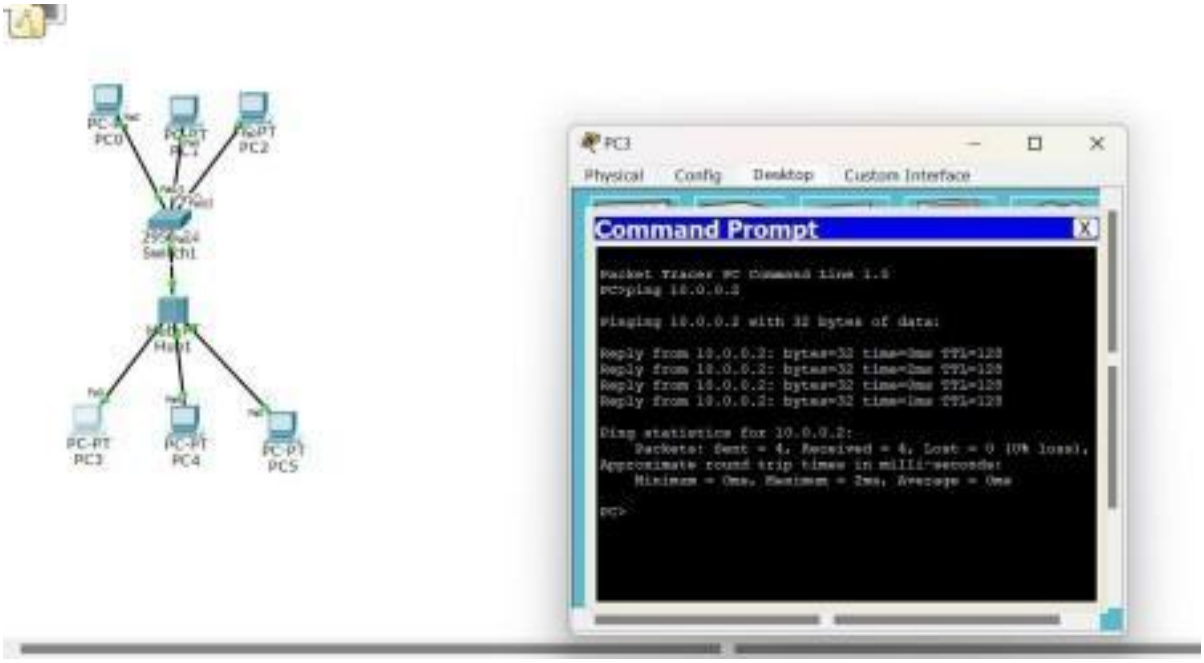


2. Switch and PCs



3. Hub,Switch and PCs

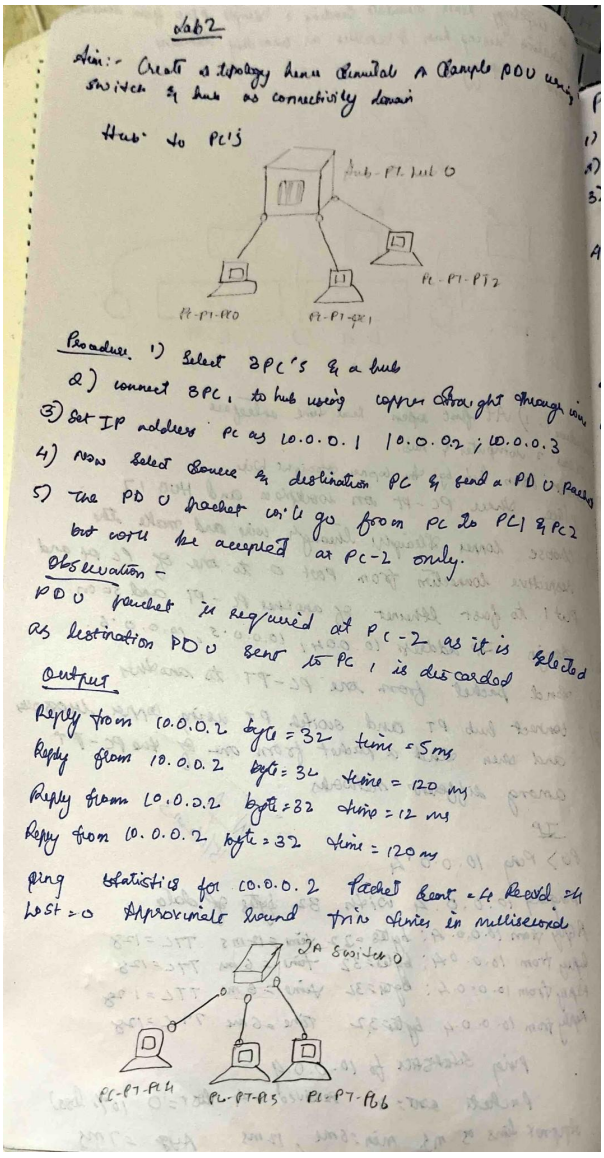



PC0
PC1
PC2
Switch1
Hub1
PC3
PC4
PC5
PC3
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.2
Pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
Reply from 10.0.0.2: bytes=32 time=1ms TTL=128
Ping statistics for 10.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 2ms, 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:



Procedure

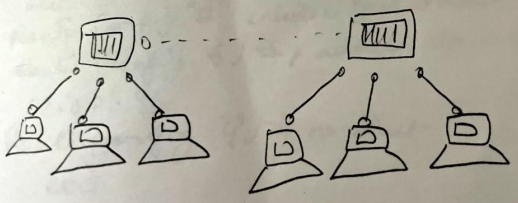
1) Select 3 PC's & switch
2) Connect them using copper straight through wire
3) Set IP address as 10.0.0.4, 10.0.0.5, 10.0.0.6 respectively for the PC's
4) Now select source & destination PC for instance PC-3 to PC5 & send a packet.

Output

Reply from 10.0.0.6 Byte =32 time TTL=120
Reply from 10.0.0.6 Bytes =0ms TTL=120
Reply from 10.0.0.6 Bytes=32 time=0ms TTL=120
Reply from 10.0.0.6 Byte=32 time 0ms TTL=120
ping statistics for 10.0.0.6
Package Sent = 4, Received = 4 Lost=0
Approx round trip times in milliseconds
Minimum = 0ms, Maximum = 3ms Average = 0ms

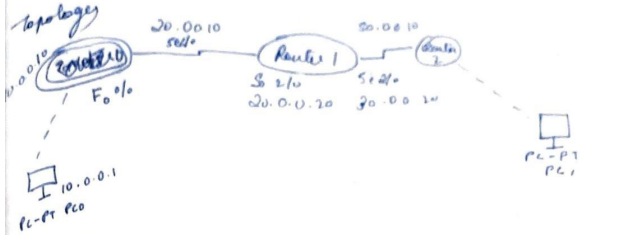
Observation:-

PC 3 sends packet to switch & it sends it both to PC & PC 5 in first round.
PC 5 rejects PC5 accepts & send acknowledgement packet to both PC 3 & PC 4
PC 4 discards PC3 accepts it. Now when PC 3 sends packets it sends only to PC5.



Aim configure using 3 routers & PCs

topology



Procedure

1) The network is started by selecting end devices PC0 to PC1 in genuine PC's and placing them in work
3) Select 3 routers PT and place them as router 0, Ro
4) PC0 & PC1 are connected to router 0 & router 2 respectiv using cross over.
4) Connect router 0 to router 1 router 1 to router 2.
5) Set up IP address of PC0 to 10.0.0.1 PT to 10.0.

Configure router by opening CLI

in router 0

Router > enable

Router # config t

Router (config) # interface fastethernet 0/0

Router (config if) # ip address 10.0.0.10
255.0.0.0

Router (config if) # no shut

exit

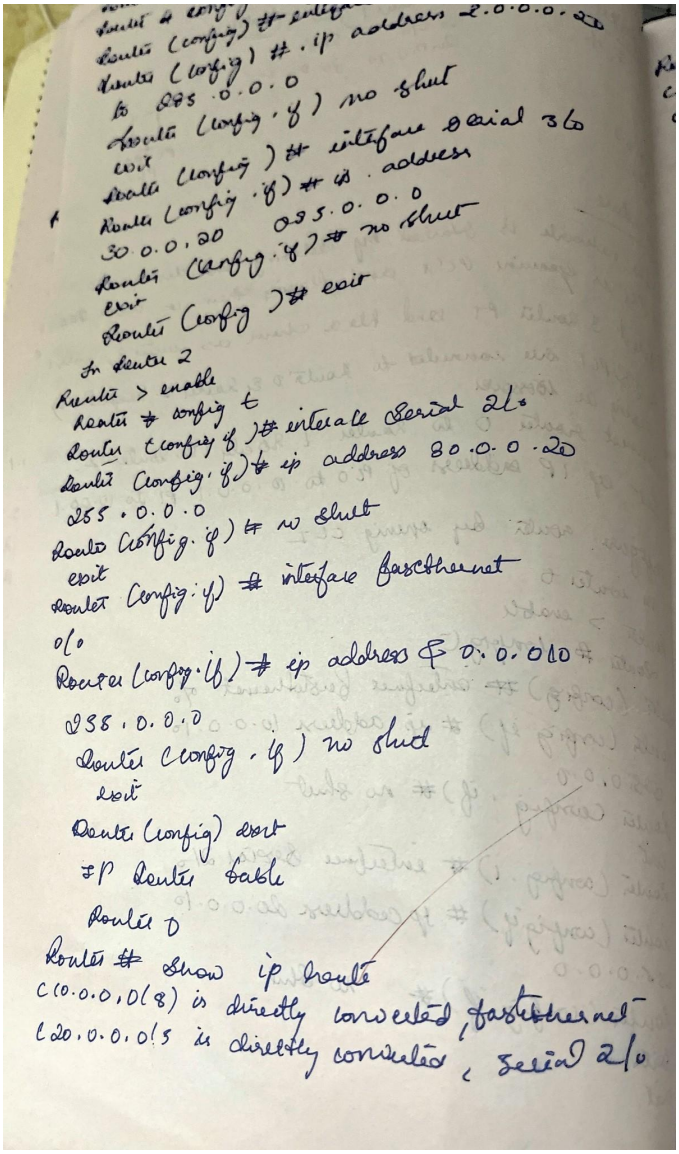
Router (config) # interface serial 2/0

Router (config if) # ip address 20.0.0.10
255.0.0.0

Router (config if) # no shut

exit

exit

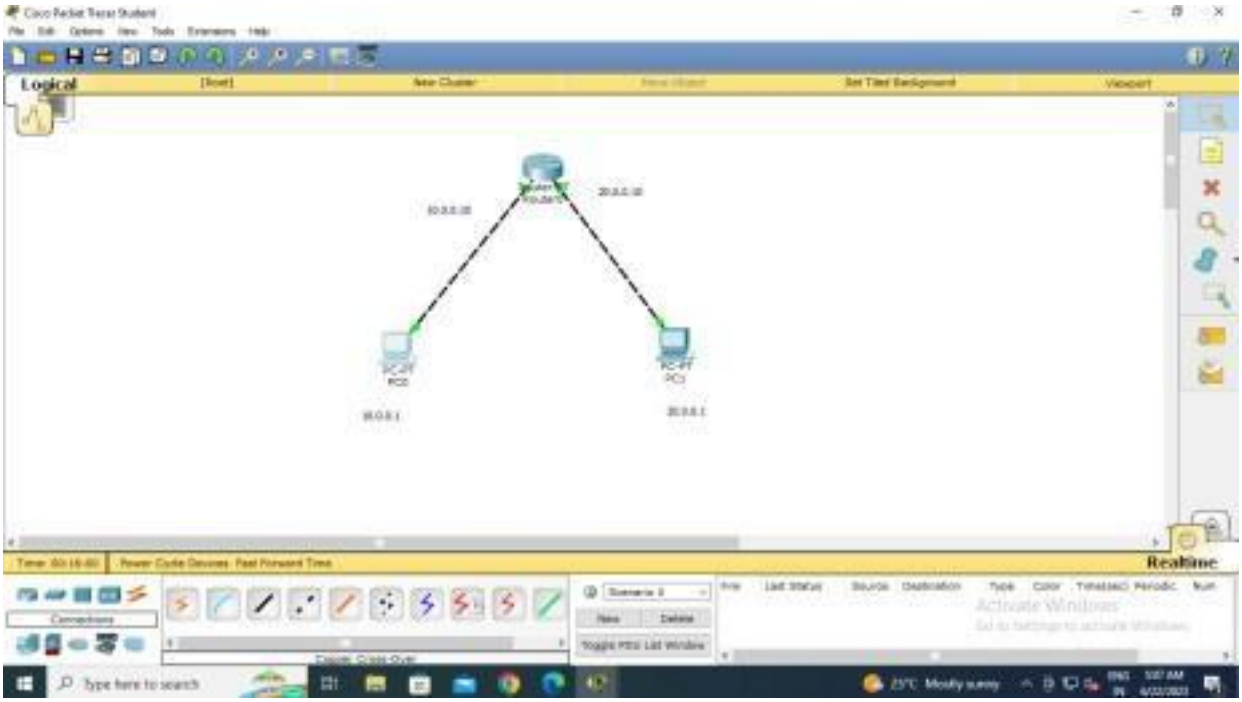


Router (config) # ip address 2.0.0.0 2
to 255.0.0.0
Router (config-if) no shut
exit
Router (config) # interface Serial 3/0
Router (config-if) # ip address
30.0.0.20 255.0.0.0
Router (config-if) # no shut
exit
Router (config) # exit

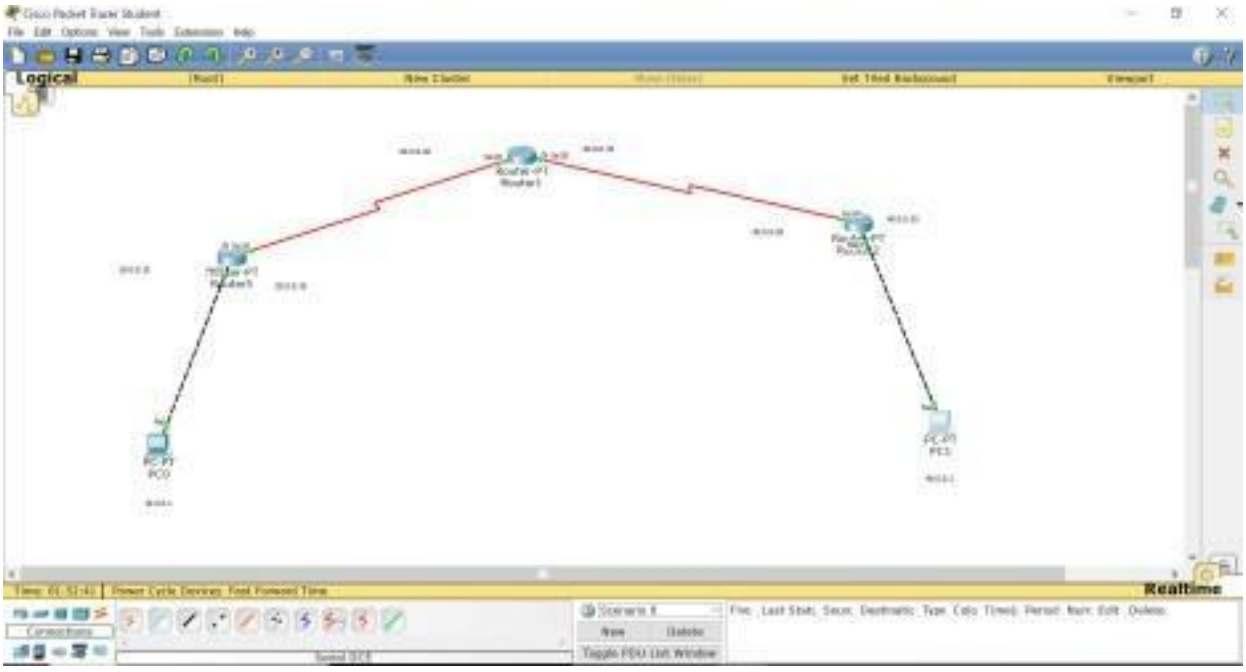
In Router 2
Router > enable
Router # config t
Router (config) # interface Serial 2/0
Router (config-if) # ip address 30.0.0.20
255.0.0.0
Router (config-if) # no shut
exit
Router (config) # interface fastethernet
0/0
Router (config-if) # ip address 9.0.0.10
255.0.0.0
Router (config-if) no shut
exit
Router (config) exit
#P Router table
Router D
Router # show ip route
C 10.0.0.0/8 is directly connected, fastethernet
C 20.0.0.0/8 is directly connected, Serial 2/0

TOPOLOGY:

PROGRAM 2.1

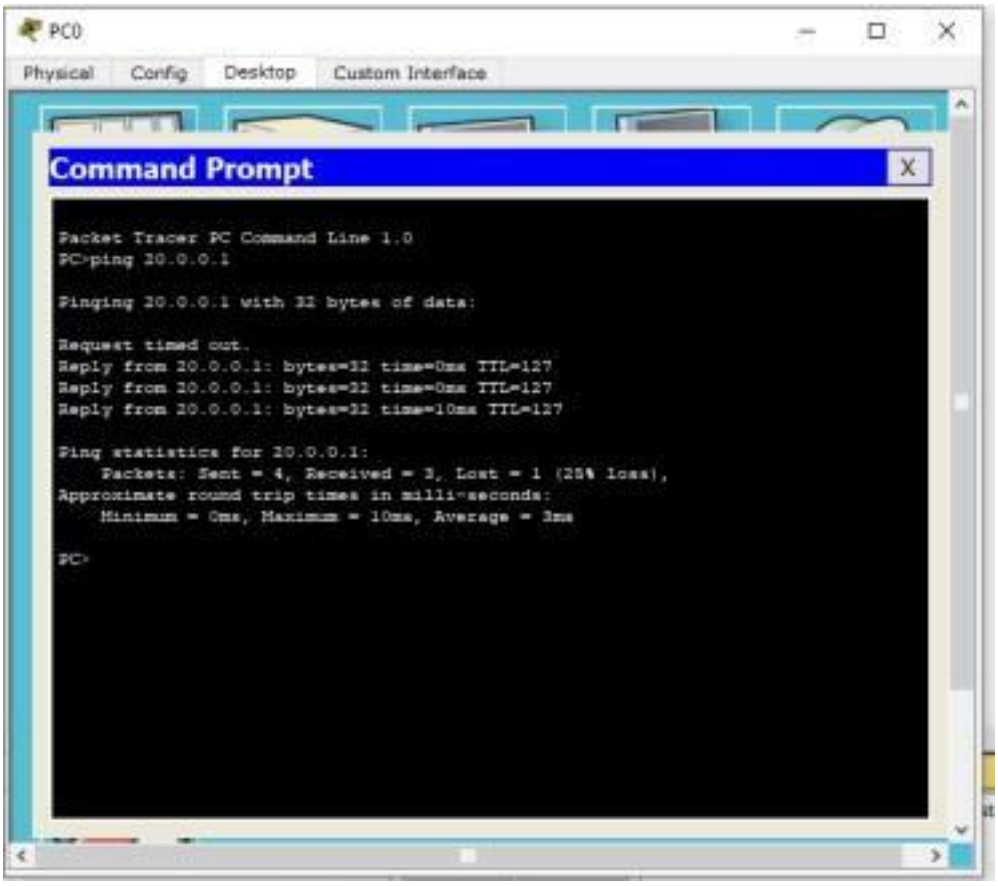


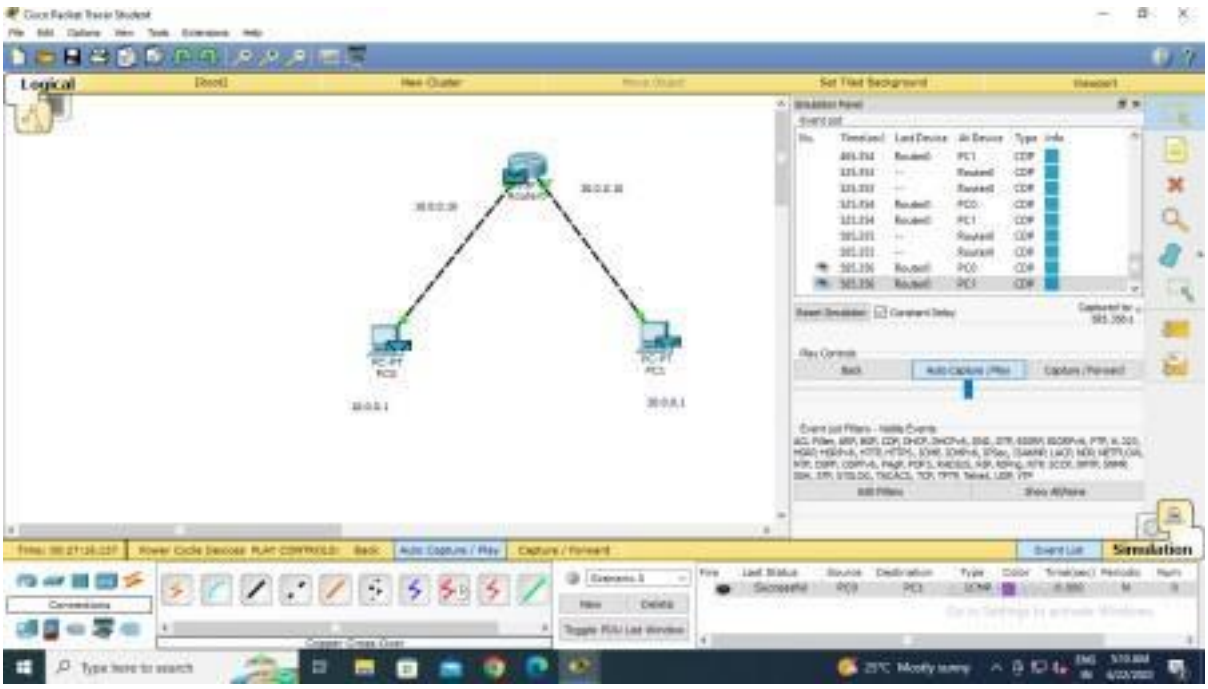
PROGRAM 2.2



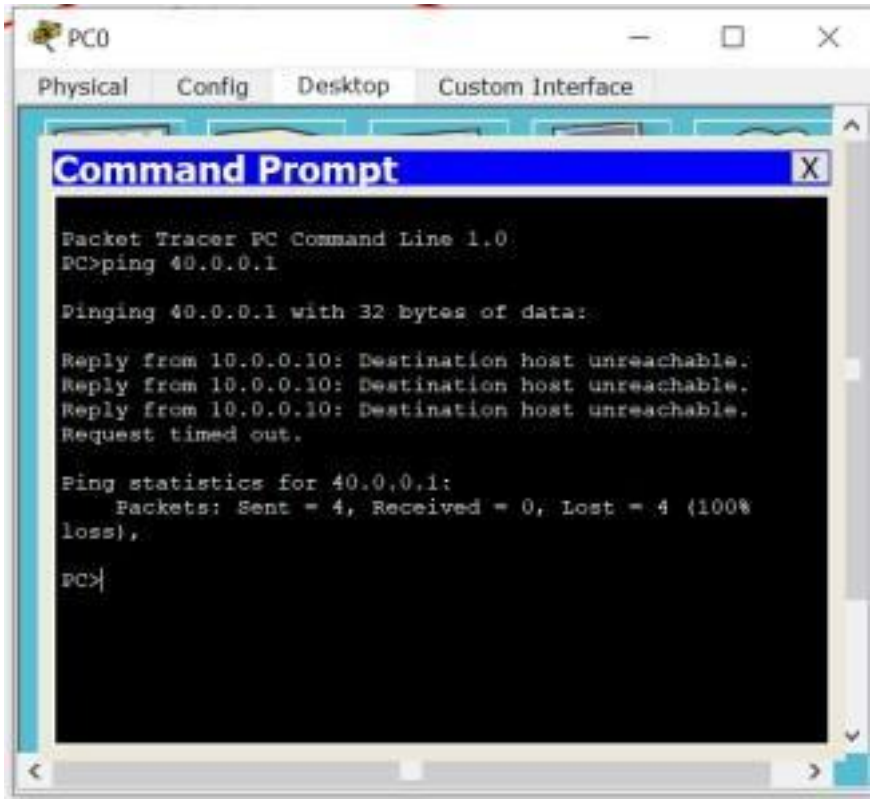
OUTPUT:

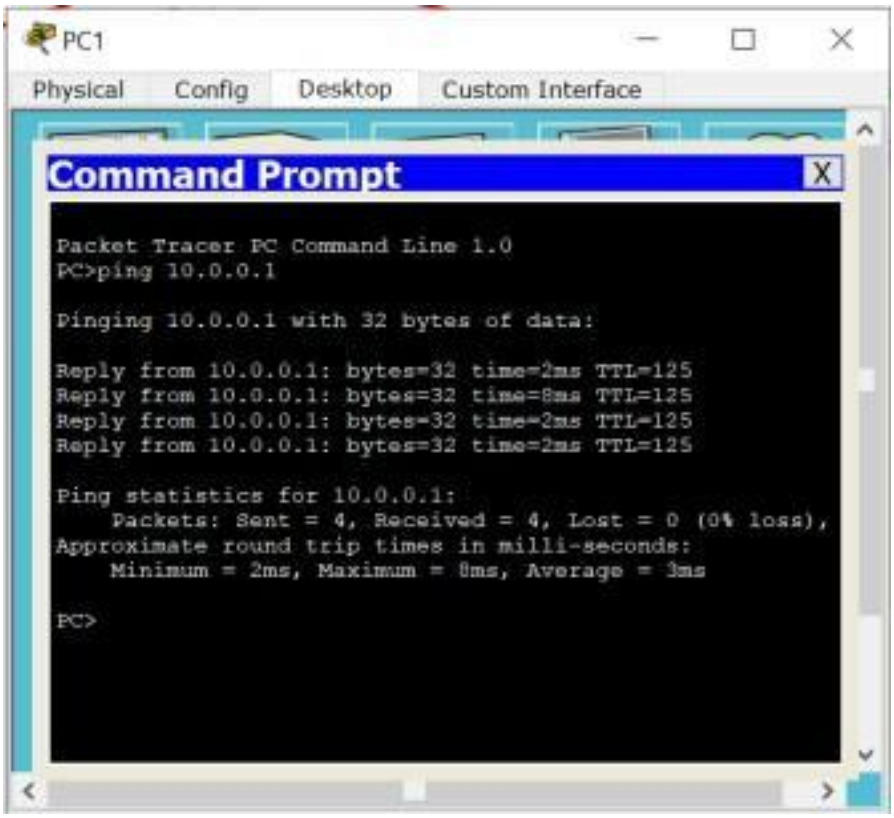
PROGRAM 2.1

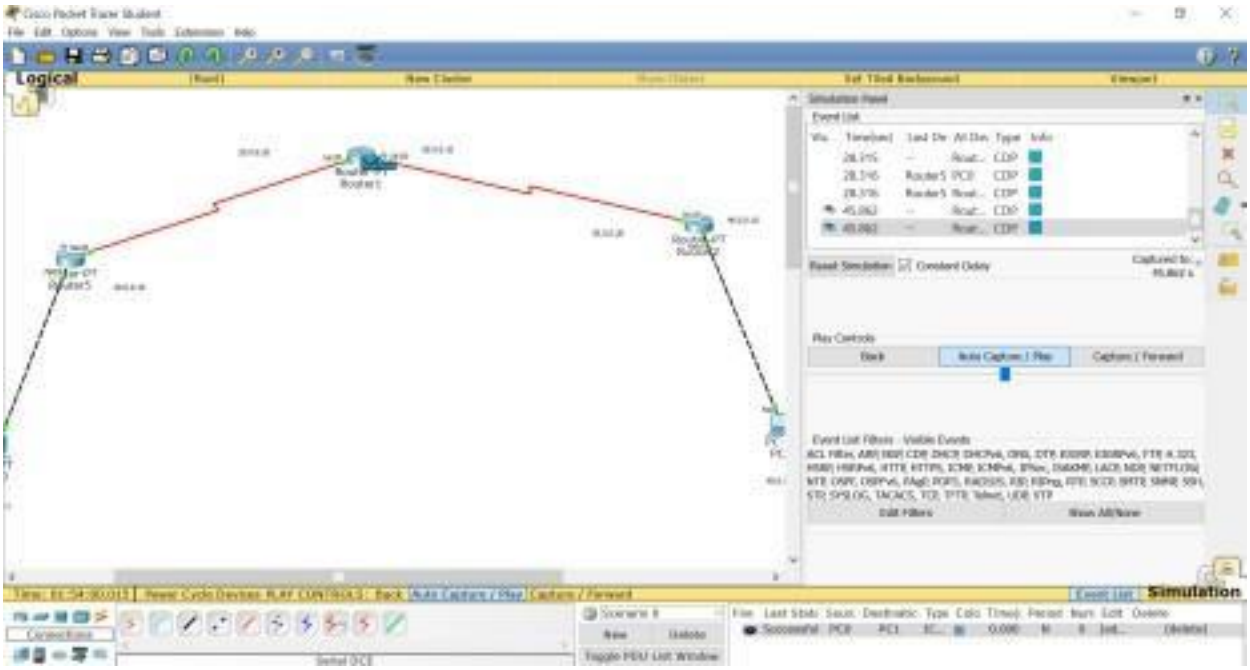




PROGRAM 2.2

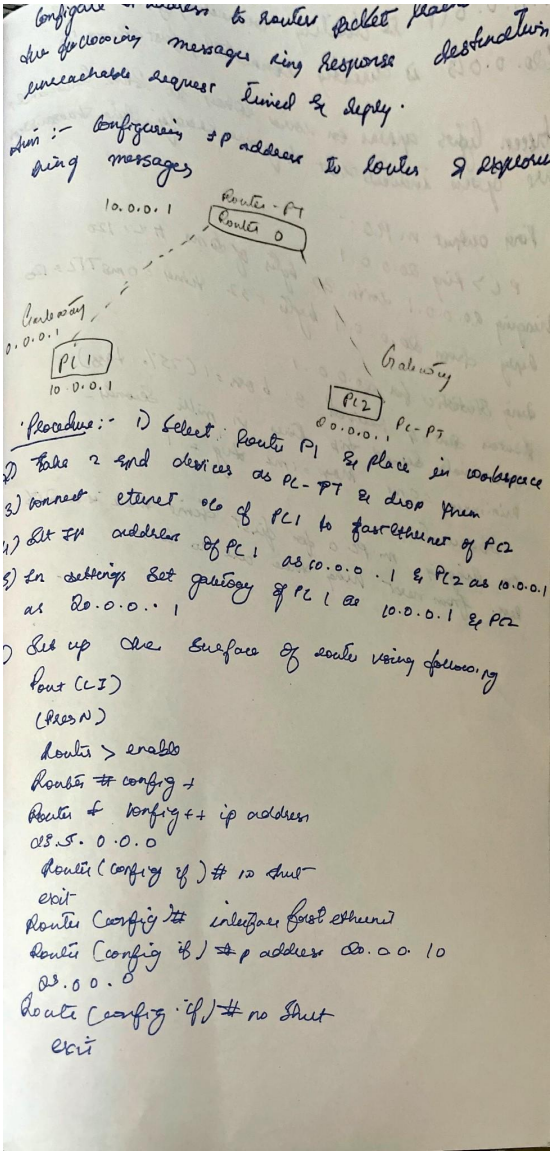

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Reply from 10.0.0.10: Destination host unreachable.
Reply from 10.0.0.10: Destination host unreachable.
Reply from 10.0.0.10: Destination host unreachable.
Request timed out.
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100%
loss),
PC>


PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
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>


Logical
Simulation

WEEK 3

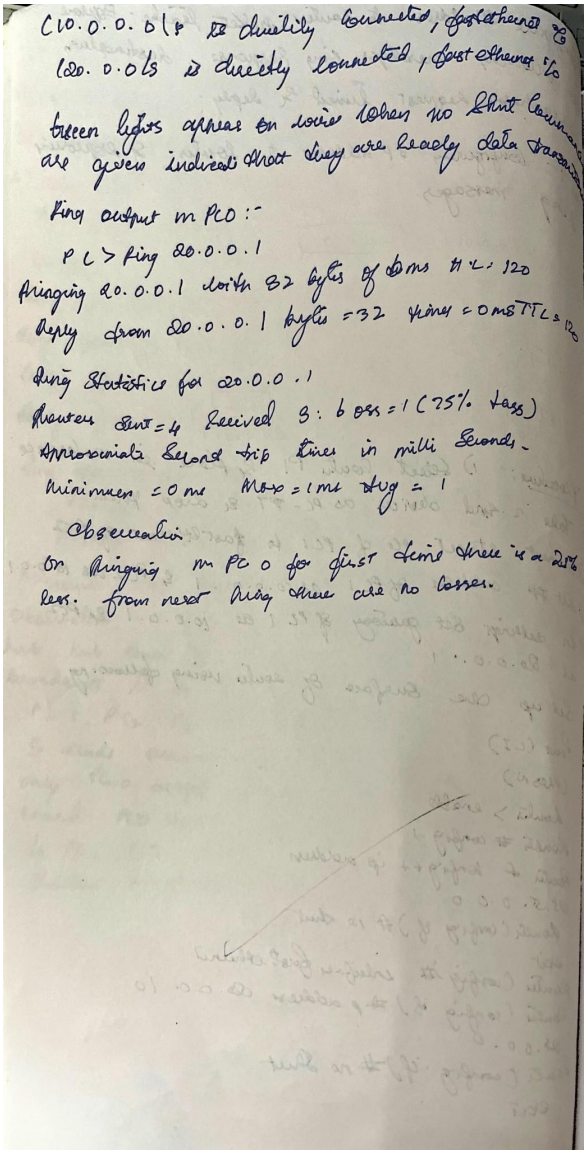
Configure default route, static route to the Router.
OBSERVATION:



Configure IP address to router packet tracer
Use following message ping Response destination unreachable request timed & reply.
Aim :- Configuring IP address to routes & Explore ping messages

Procedure :- 1) Select Router PT & place in workspace
2) Take 2 end devices as PC-PT & drop them
3) Connect ethernet o/c of PC1 to fastethernet of PC2
4) Set IP address of PC1 as 10.0.0.1 & PC2 as 10.0.0.1
5) In settings Set gateway of PC1 as 20.0.0.1 & PC2 as 10.0.0.1
6) Set up the Surface of router using following

Router (CLI)
(Press N)
Router > enable
Router # config t
Router # config t ip address
20.5.0.0.0
Router (config if) # no shut
exit
Router (config if) # interface fast ethernet
Router (config if) # ip address 20.0.0.10
20.0.0.0
Router (config if) # no shut
exit



C10.0.0.0/8 is directly connected, FastEthernet 0/0
L20.0.0.0/8 is directly connected, FastEthernet 1/0

Green lights appear on router when no shut command are given indicates that they are ready data transmission

Ping output in PC0 :-

PC > Ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data: TTL: 120

Reply from 20.0.0.1 bytes = 32 time = 0ms TTL = 120

Ping Statistics for 20.0.0.1

Packets Sent = 4 Received = 3 : 0 loss = 1 (75% loss)

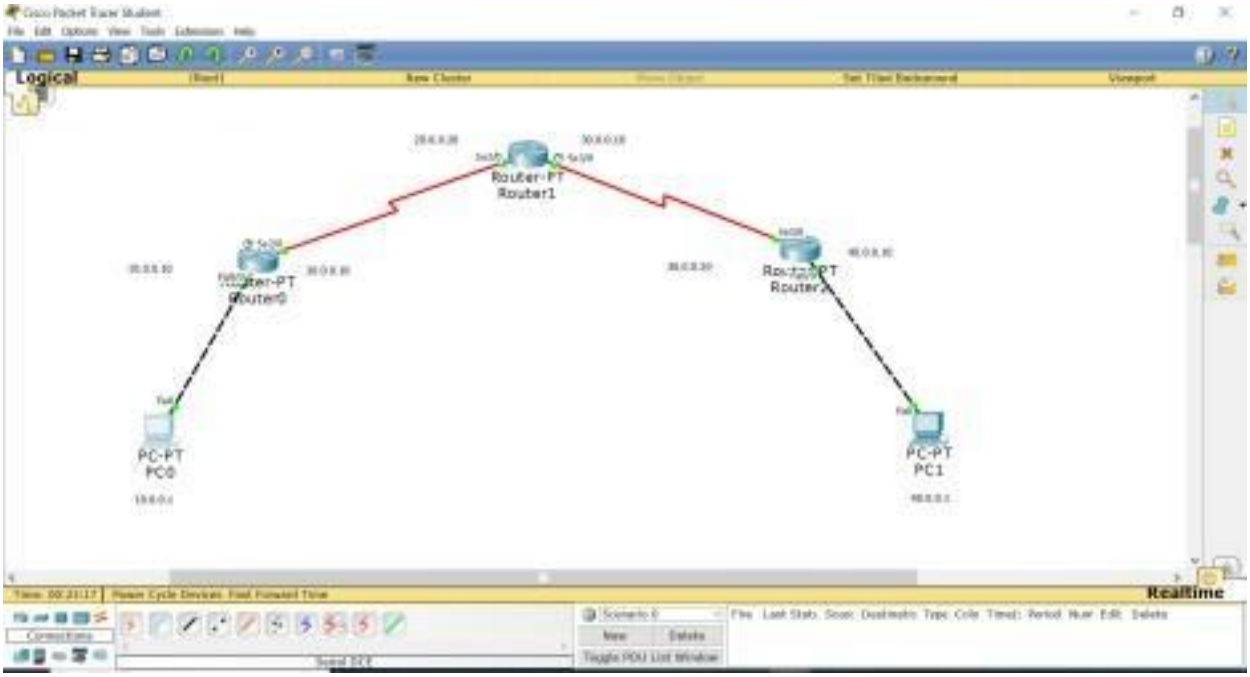
Approximate round trip times in milli seconds -

Minimum = 0 ms Max = 1 ms Avg = 1

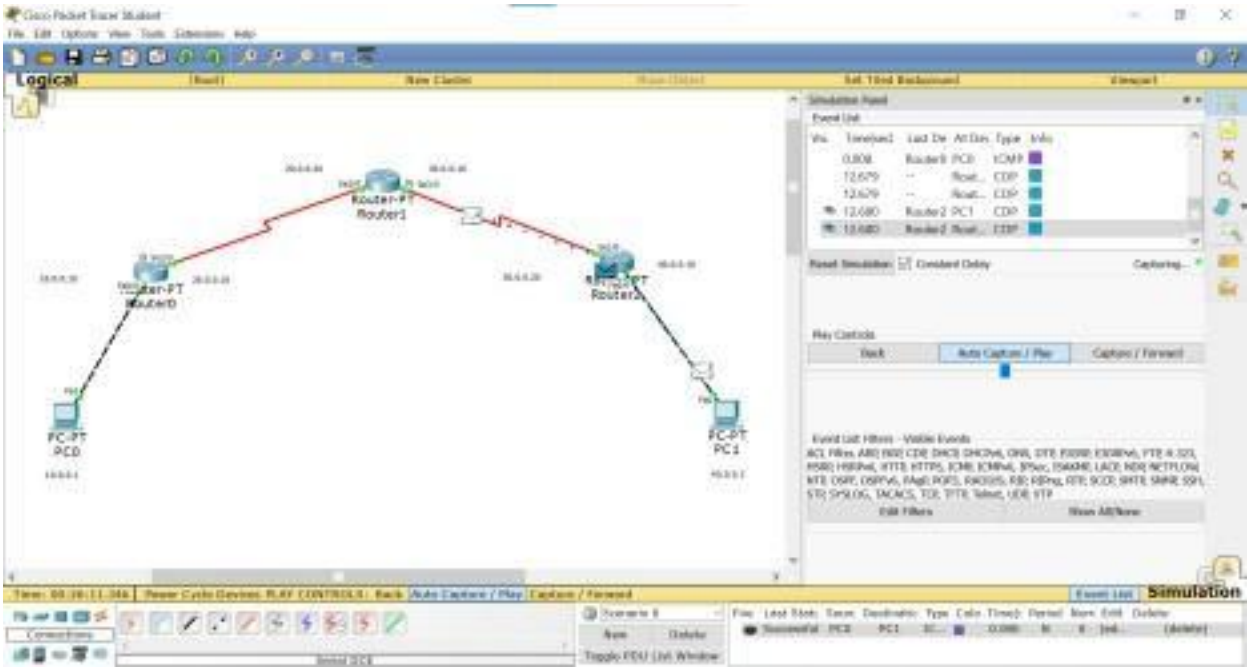
Observation

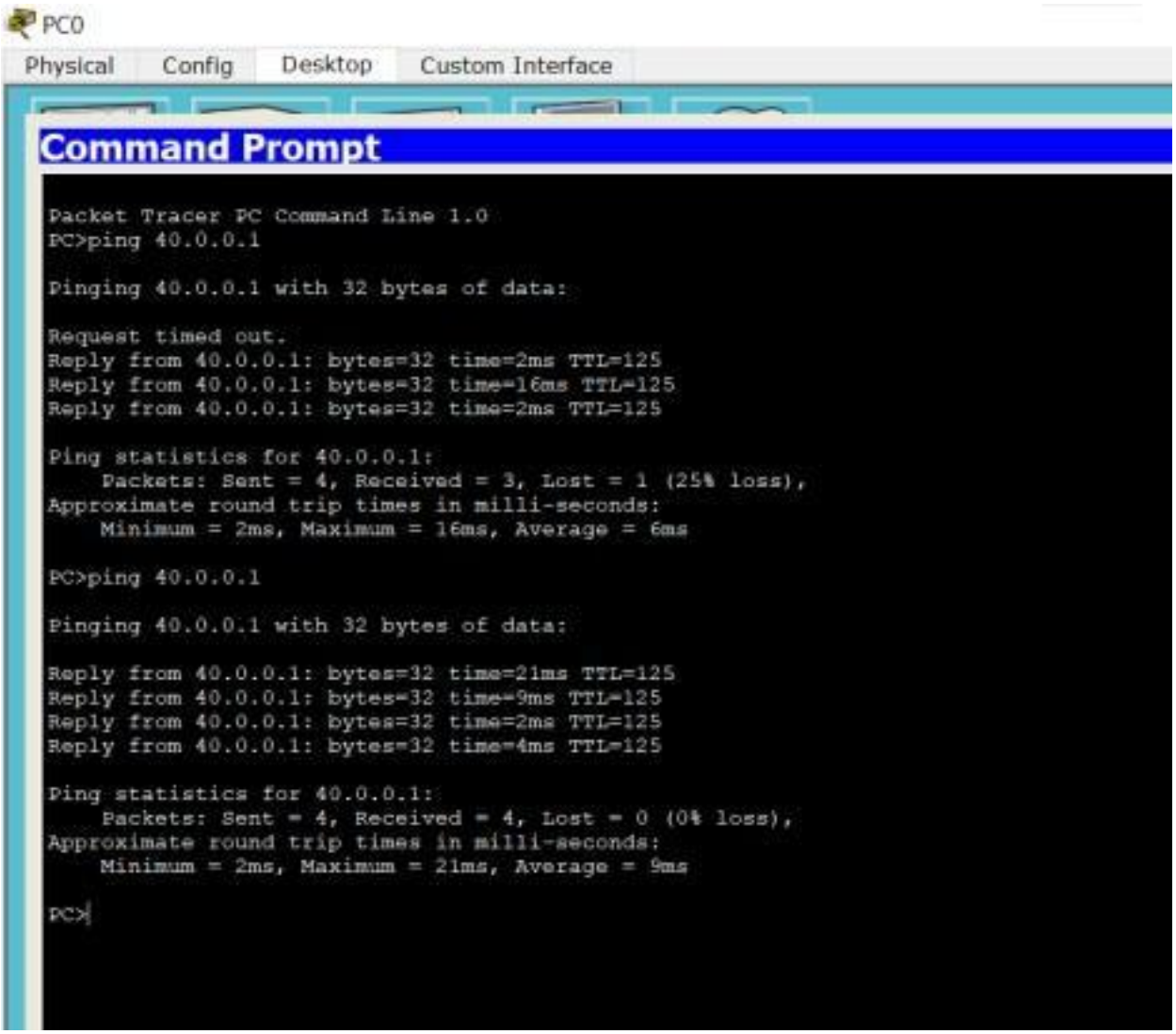
On Pinging in PC 0 for first time there is a 25% loss. from next ping there are no losses.

TOPOLOGY:



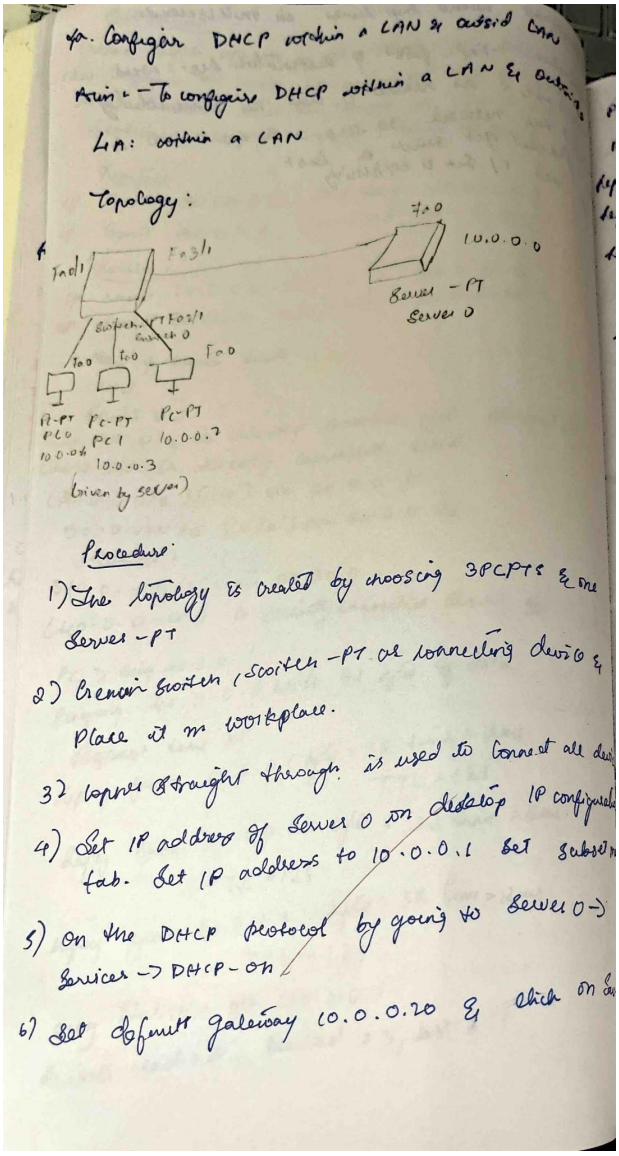
OUTPUT:

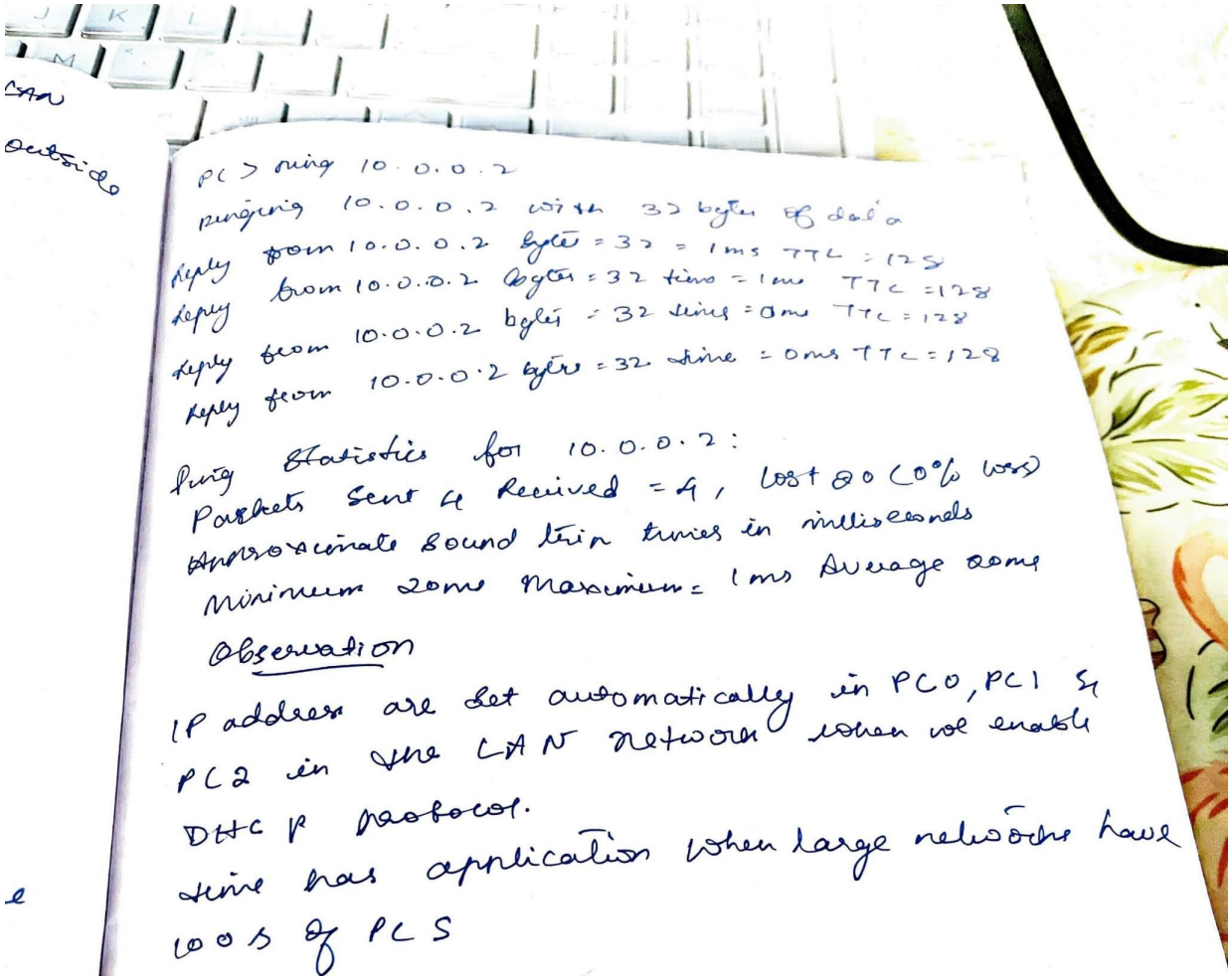



PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Request timed out.
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125
Reply from 40.0.0.1: bytes=32 time=16ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms 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:





PC > ping 10.0.0.2

pinging 10.0.0.2 with 32 bytes of data

Reply from 10.0.0.2 bytes = 32 = 1ms TTL = 128

Reply from 10.0.0.2 bytes = 32 time = 1ms TTL = 128

Reply from 10.0.0.2 bytes = 32 time = 0ms TTL = 128

Reply from 10.0.0.2 bytes = 32 time = 0ms TTL = 128

Ping Statistics for 10.0.0.2:

Packets Sent 4 Received = 4, Lost 00 (0% loss)

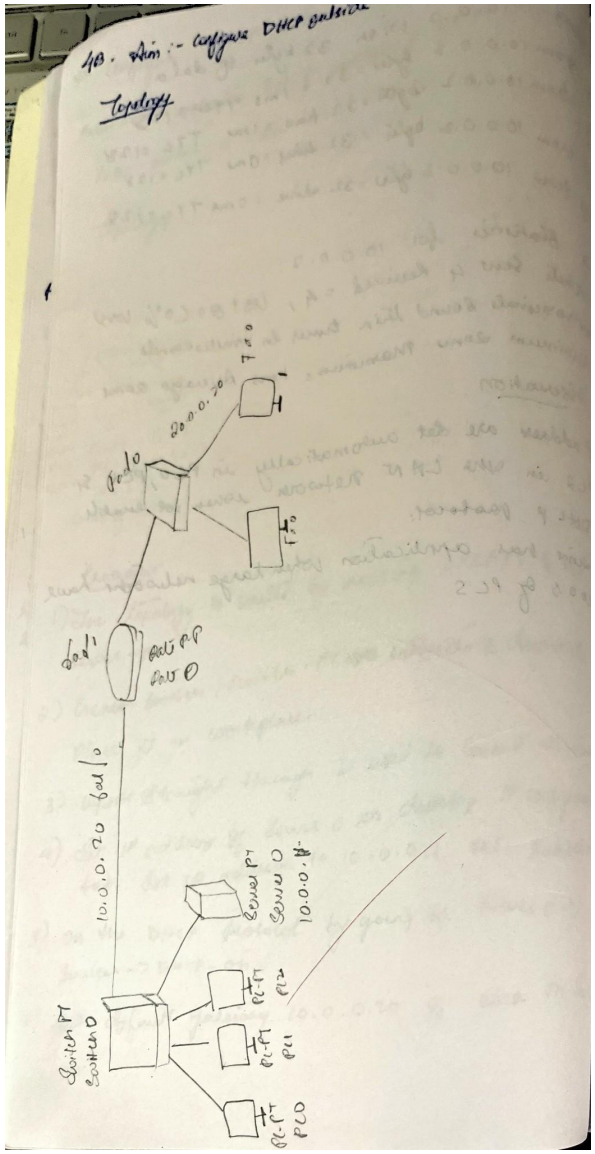
Approximate round trip times in milliseconds

Minimum 20ms Maximum = 1ms Average 20ms

Observation

IP address are set automatically in PC0, PC1 & PC2 in the LAN network when we enable DHCP protocol.

time has application when large networks have loos of PCS


Topology
Fa0/0
Fa0/1
Fa0/0
Fa0/1
10.0.0.20
Switch PT
Switch0
Server PT
Server0
PC-PT
PC2
PC-PT
PC1
PC-PT
PC0

Procedure :-

1.) To the topology created in NP, connect a router (Router-PT) using a straight through wire.

2.) Through Switch-PT Switch 1 connect 2 PCs, PC3 & PC4 & connect Switch 1 to Router 0

3.) In Router 0, Set IP address Fa4/0 to 10.0.0.20 & Fa0/0 to 20.0.0.20

4.) In router 0,
interface fastethernet 0/0
Router (config-if)# ip helper address 10.0.0.1
Router (config-if)# no shut.

5.) In Server 0 goto config > setting > gateway & set gateway 10.0.0.20

6.) Set services to DHCP.
Start IP address to 20.0.0.2

7.) In desktop mode of PC3 & PC4 select DHCP & they will automatically be assigned IP address as 20.0.0.2 & 20.0.0.3

Project output

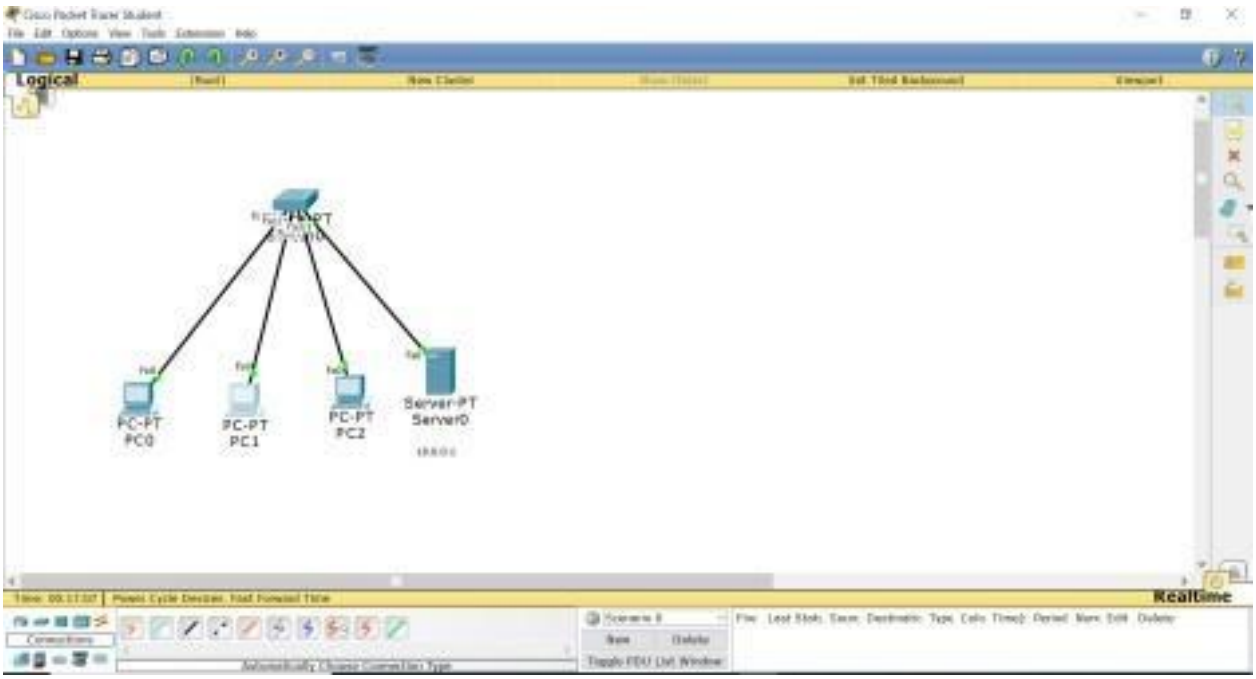
PC0
PC > Ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data

Reply from 20.0.0.2 bytes=32 time=1ms TTL=127
Reply from 20.0.0.2 bytes=32 time=20ms TTL=127

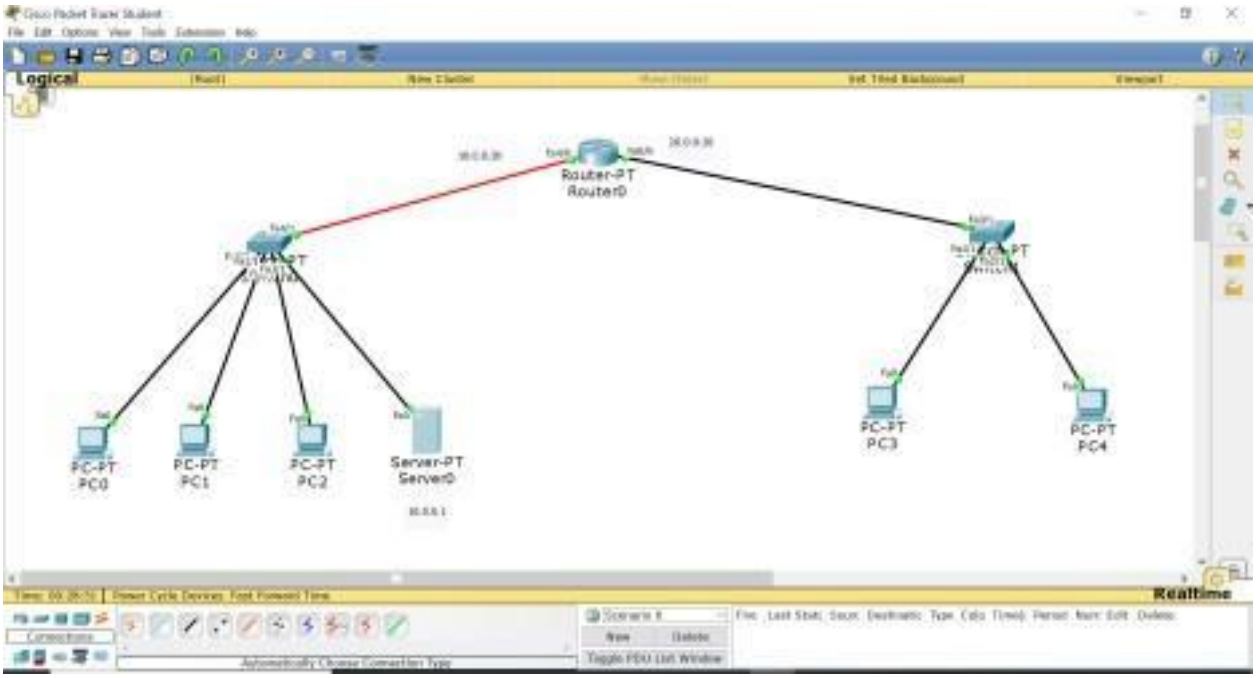
Pinging Statistics for 2

TOPOLOGY:

PROGRAM 4.1:

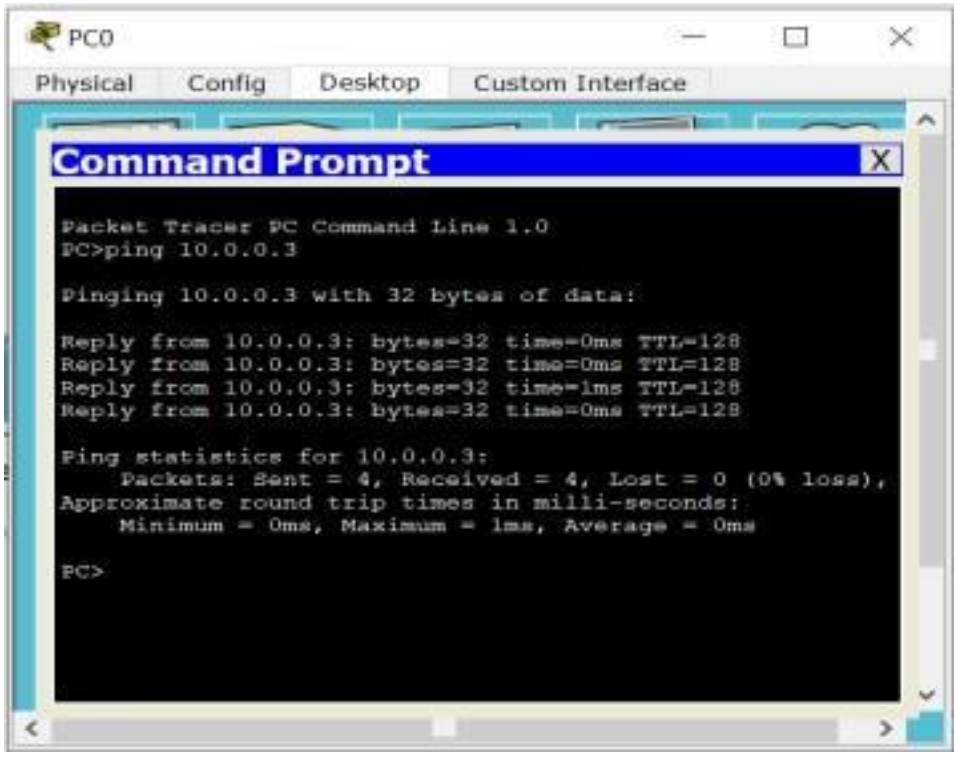


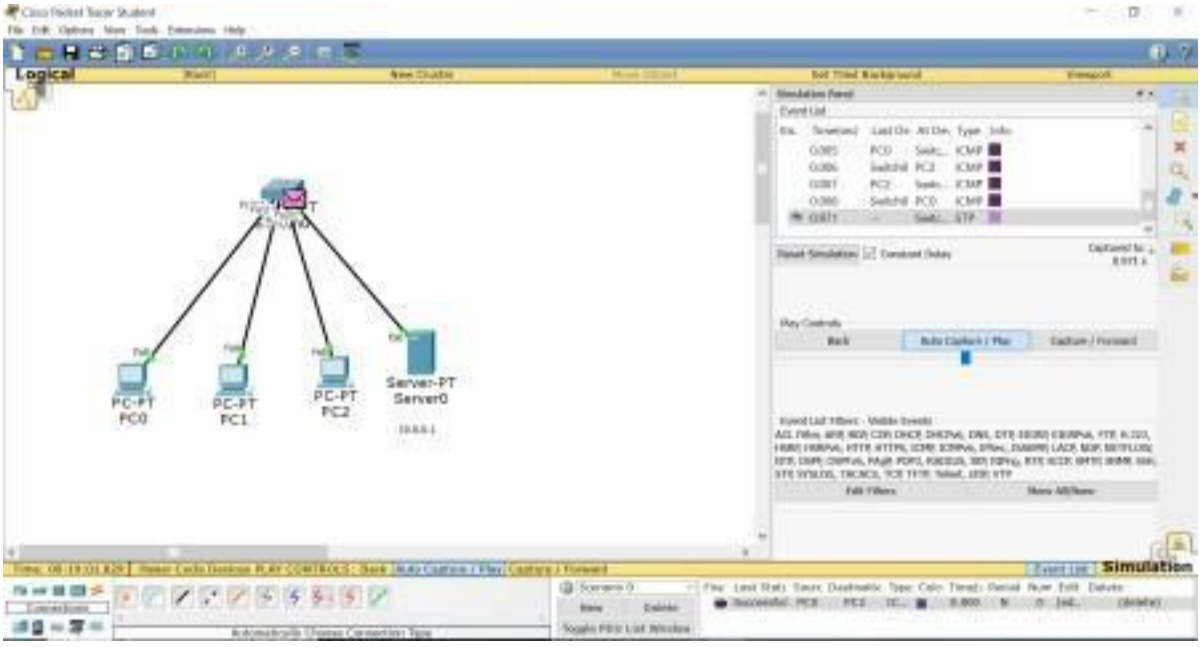
PROGRAM 4.2:



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>


Logical
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
Server-PT
Server0
Simulation

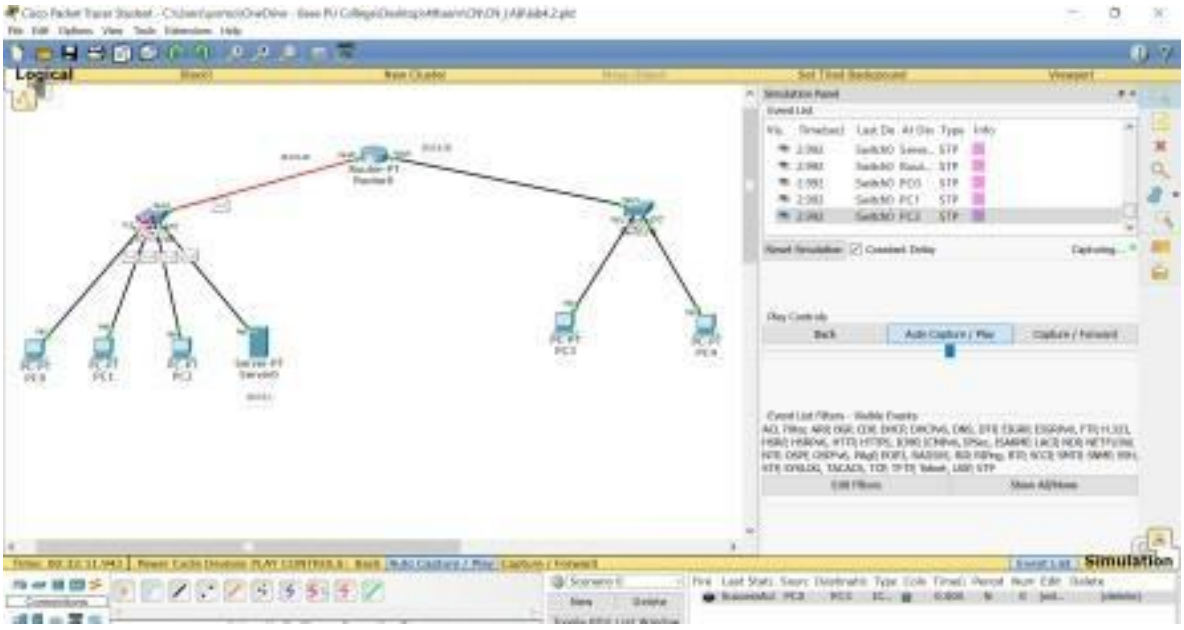
PROGRAM 4.2:

PC0

Physical | Config | Desktop | Custom Interface

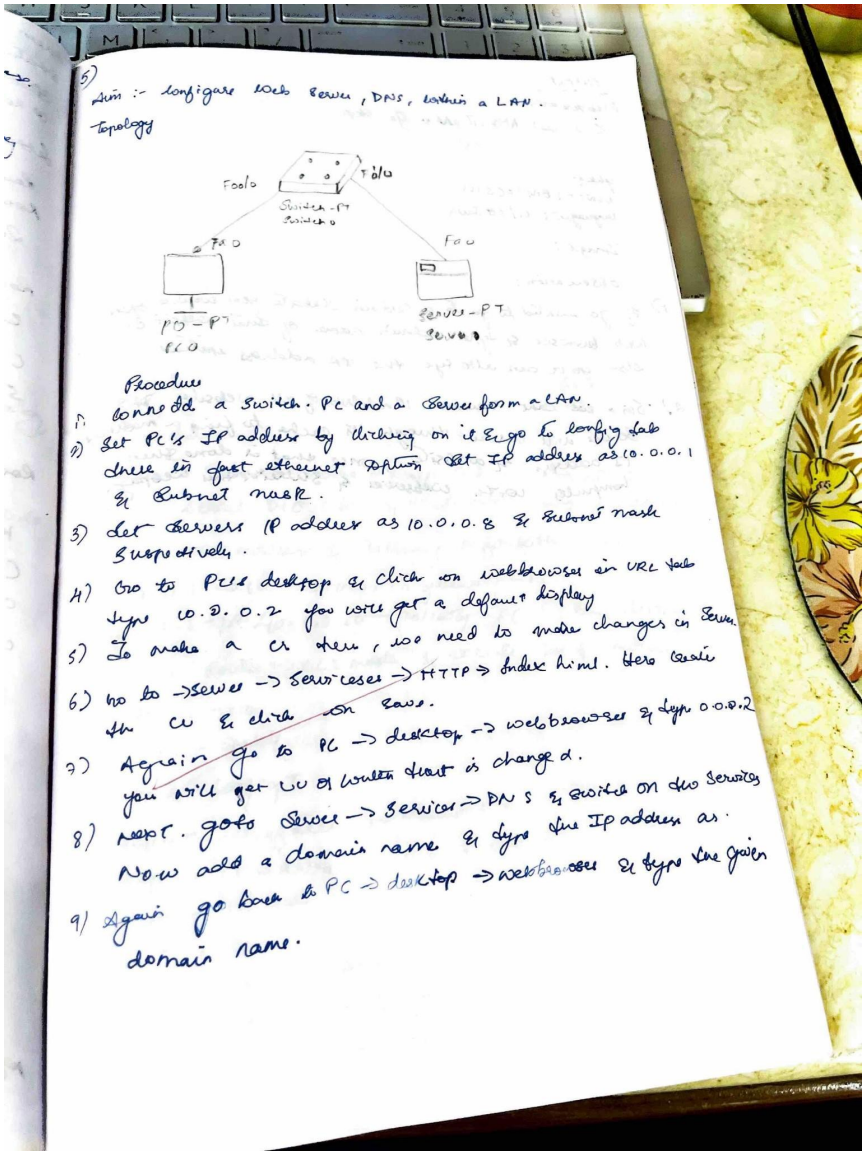
Command Prompt

```
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:



Output

Web browser

< > cat http://github.com Go Stop

CV

github

USN: - 1BM20CS111

Languages: C / C++ Java

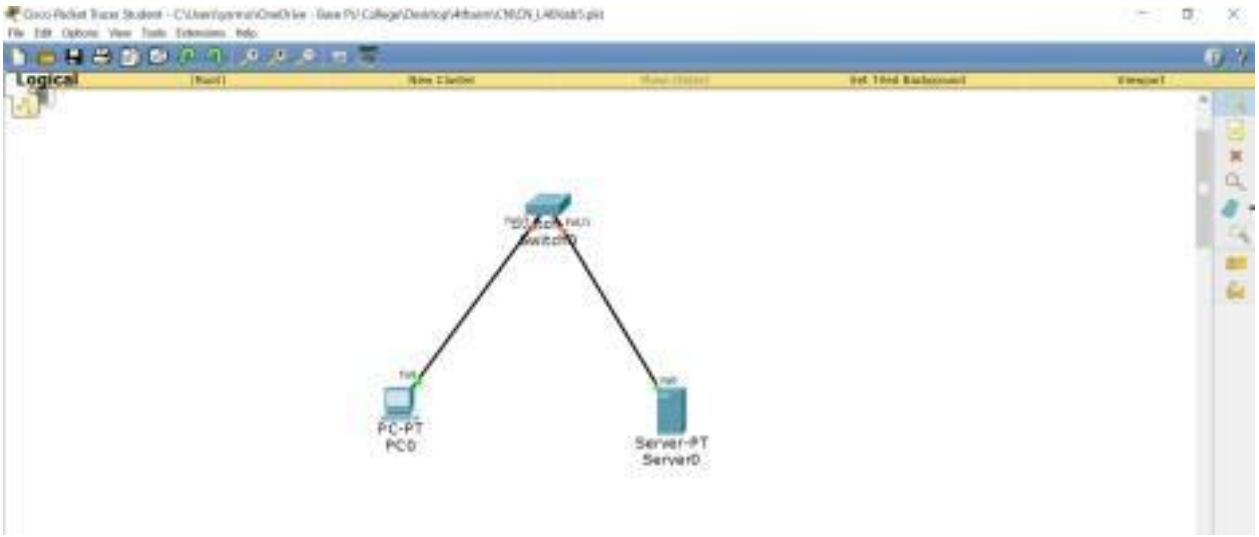
Image:

Observation:

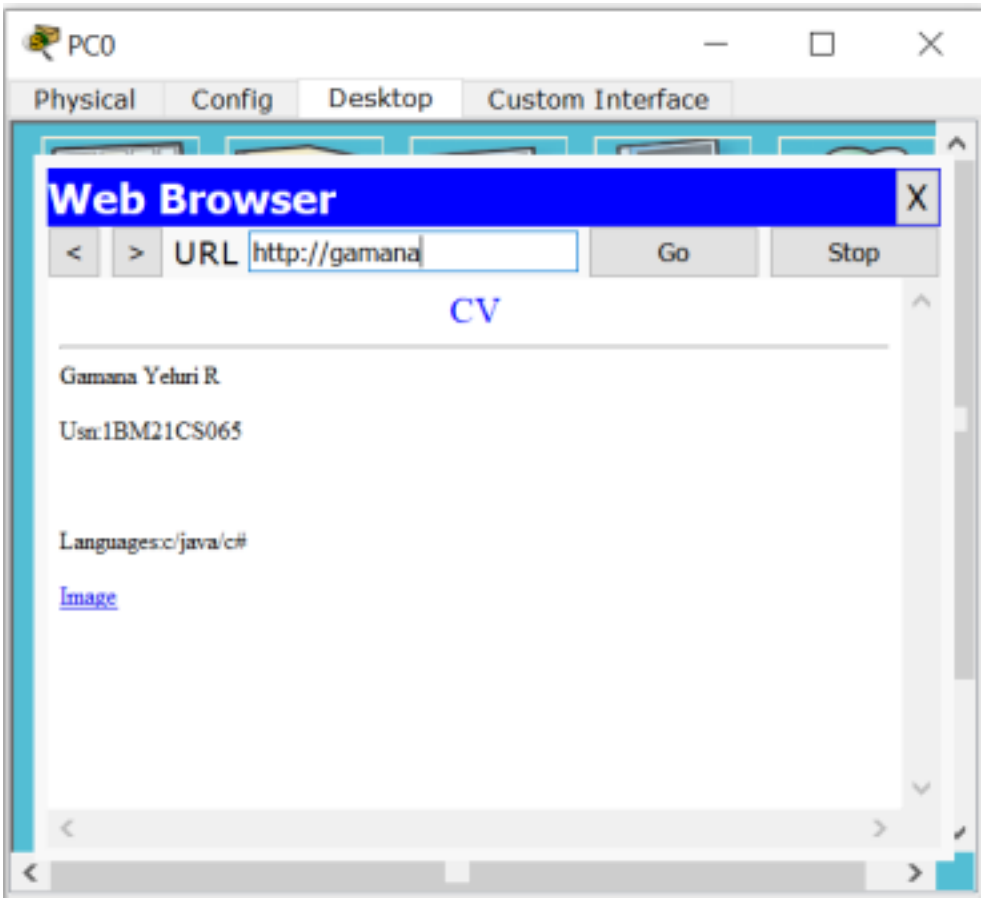
1) If you wanted to go to a certain website you could g
web browser & type address name of that website
else you can also type the IP address instead

2) Since we cant remember IP address of all websites DN
Server will server through its cache to find a ma
IP address of website, once that is done then
computer with webserver & skibidi this webpag

TOPOLOGY:

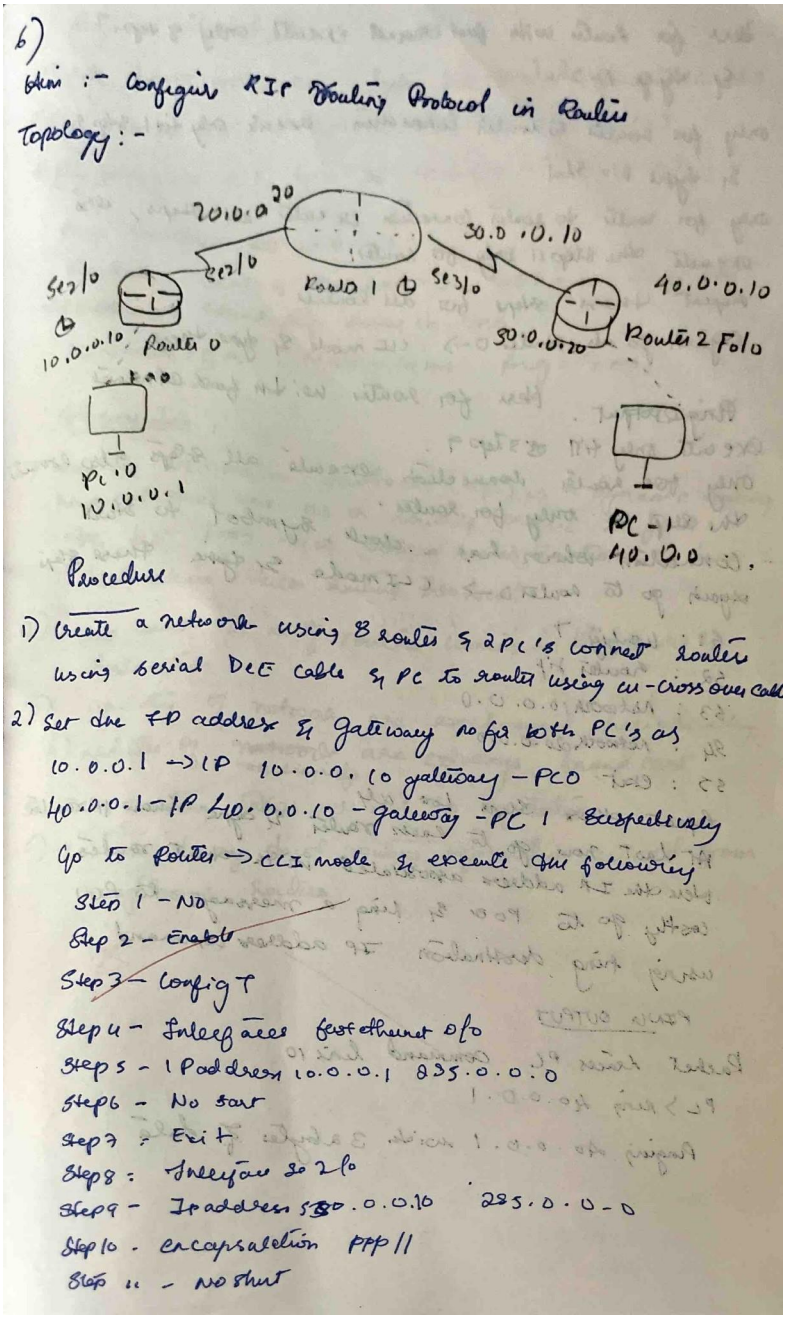


OUTPUT:



WEEK 6

Configure RIP routing Protocol in Routers.
OBSERVATION:



Here for Router with fast ethernet execute only 8 step 7
Eg type No shut

only for router to router connection execute only till 8 step 9
Eg type No shut

only for router to router connection execute all steps, else execute the step 11 only for router

Repeat these steps for all routers.

Again go to Router 0 → CLI mode Eg type these

Ping output. Here for router with fast ethernet execute only till step 9.

only for router connection execute all steps also execute the step 11 only for router.

Connection where has a clock symbol to start

Again go to router 0 → CLI mode Eg type these steps

S1 : config t
S2 : router rip
S3 : network 10.0.0.0
S4 : network 20.0.0.0
S5 : exit

Repeat these steps for all

At last now go to each router Eg type show ip route

Here the IP address associated with that routers

Lastly go to PC0 Eg ping a message to PC1

using ping destination IP address command

PING OUTPUT

Packet tracer PC command line 10

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=8ms TTL=127

Reply from 40.0.0.1 : bytes 32 time=5ms TTL=125

Reply from 40.0.0.1 : bytes=32 time=10 ms TTL=125

Ping Statistics for 40.0.0.1 :

Packets : Sent=4 Received=3 , Lost=1 (25% loss)

Approximate round trip times in ms.

Minimum=5ms Max=10ms Avg=7ms

Observation

1) Routing information protocol (rip) is a dynamic routing protocol that uses as a routing metric to find the best path b/w source & distination. It is a destination Vector routing protocol

2) Hop count is no of routers coming in b/w source & destination.

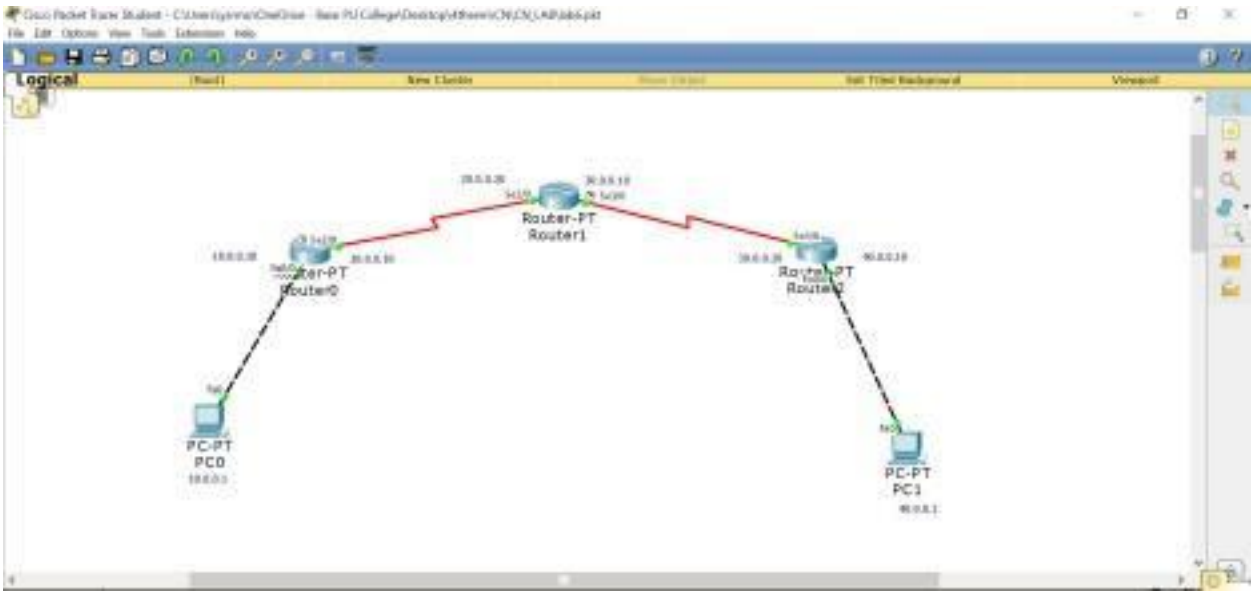
3) updates of network are exchanged periodically

4) updates of ~~network~~ routing info are always broad cast

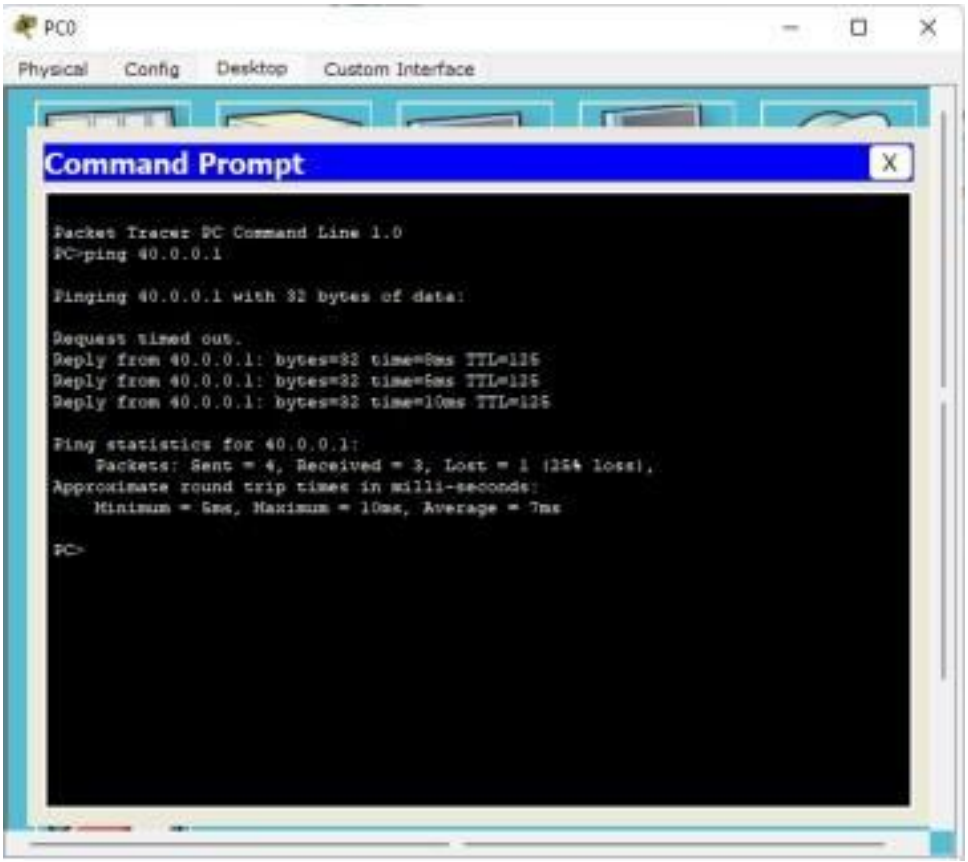
5) Full routing tables as sent in updates

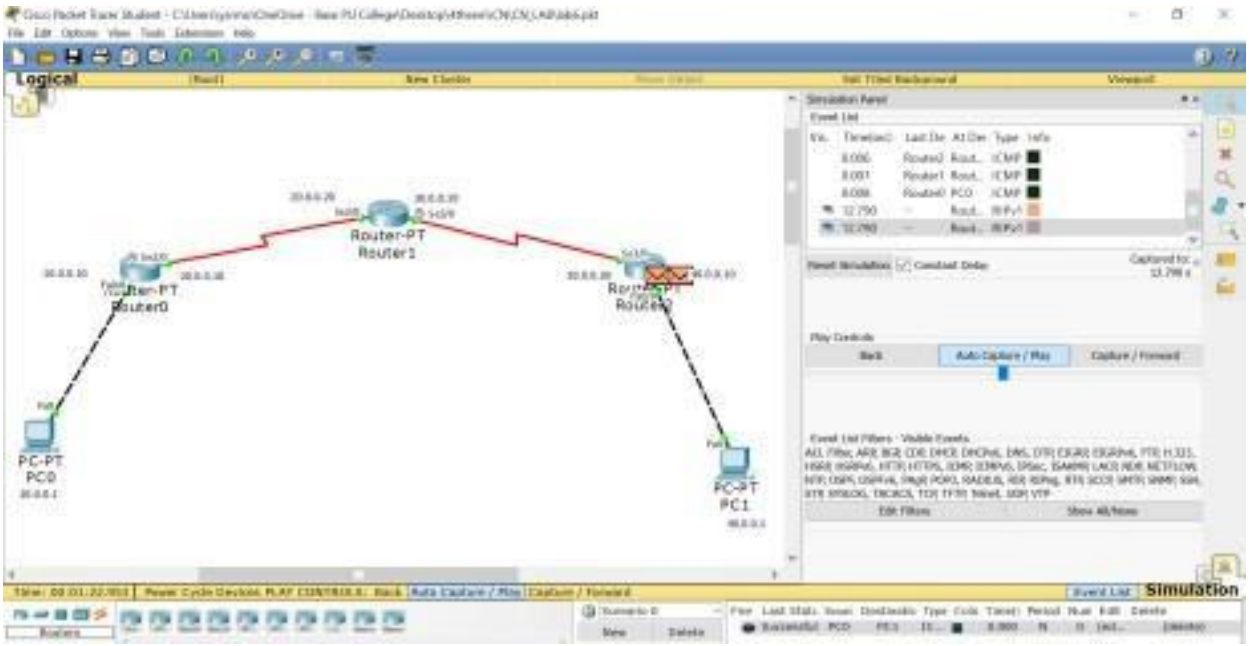
6) Routers always trust routing information received from neighbours routers.

TOPOLOGY:



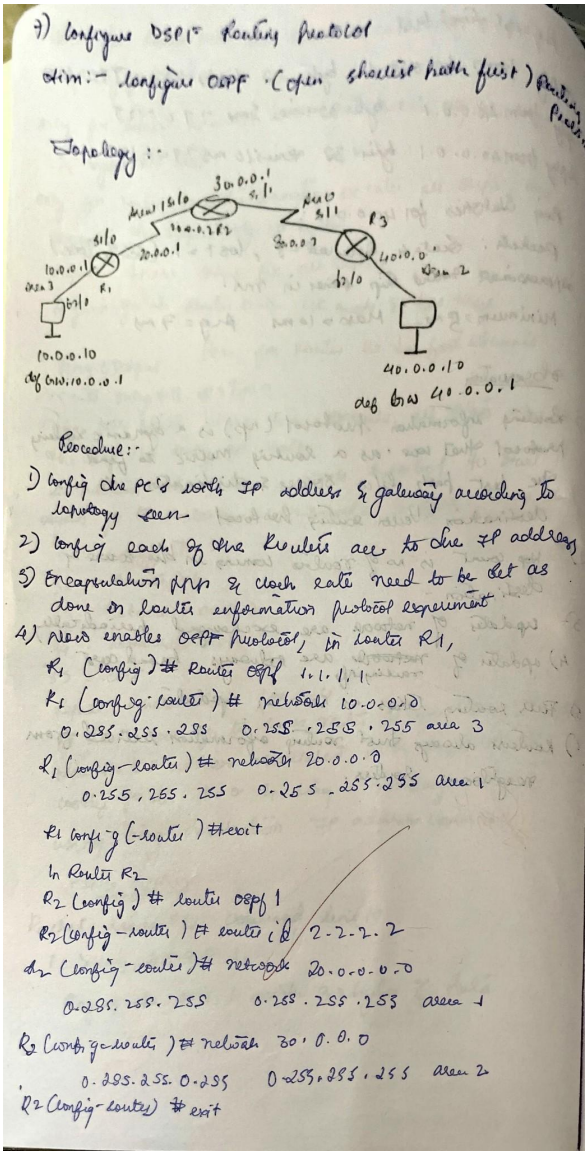
OUTPUT:

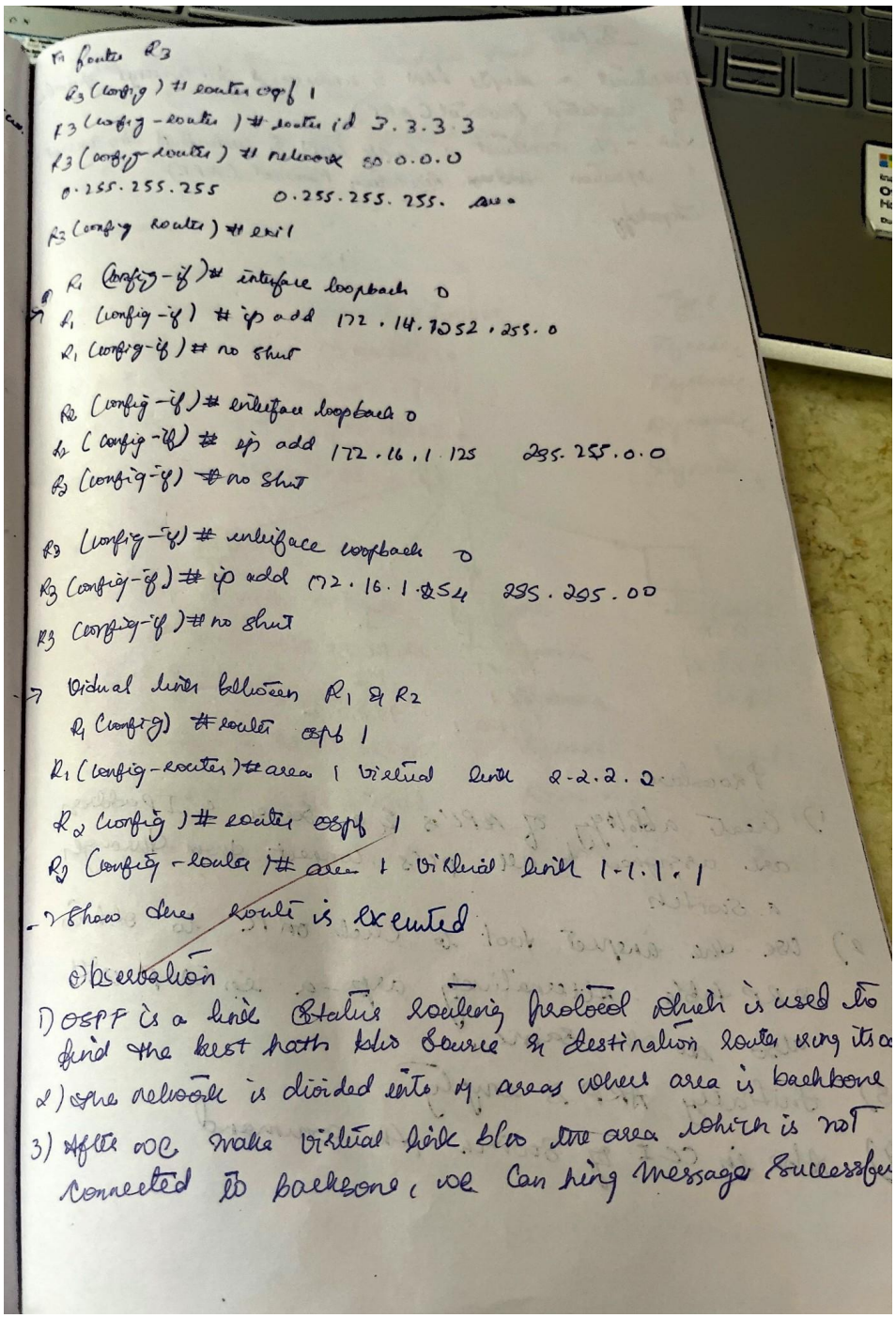



Logical
Router-PT
Router1
Router-PT
Router0
Router-PT
Router2
PC-PT
PC0
PC-PT
PC1
Simulation

WEEK 7

Configure OSPF routing protocol.
OBSERVATION:





In router R3

R3(config)# router ospf 1

R3(config-router)# router id 3.3.3.3

R3(config-router)# network 0.0.0.0 0.255.255.255 0.255.255.255. area 0

R3(config-router)# exit

R1(config-if)# interface loopback 0

R1(config-if)# ip add 172.14.7052.255.0

R1(config-if)# no shut

R2(config-if)# interface loopback 0

R2(config-if)# ip add 172.16.1.125 255.255.0.0

R2(config-if)# no shut

R3(config-if)# interface loopback 0

R3(config-if)# ip add 172.16.1.254 255.255.00

R3(config-if)# no shut

→ Virtual link between R1 & R2

R1(config)# router ospf 1

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

R2(config)# router ospf 1

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

→ Show ospf route is executed

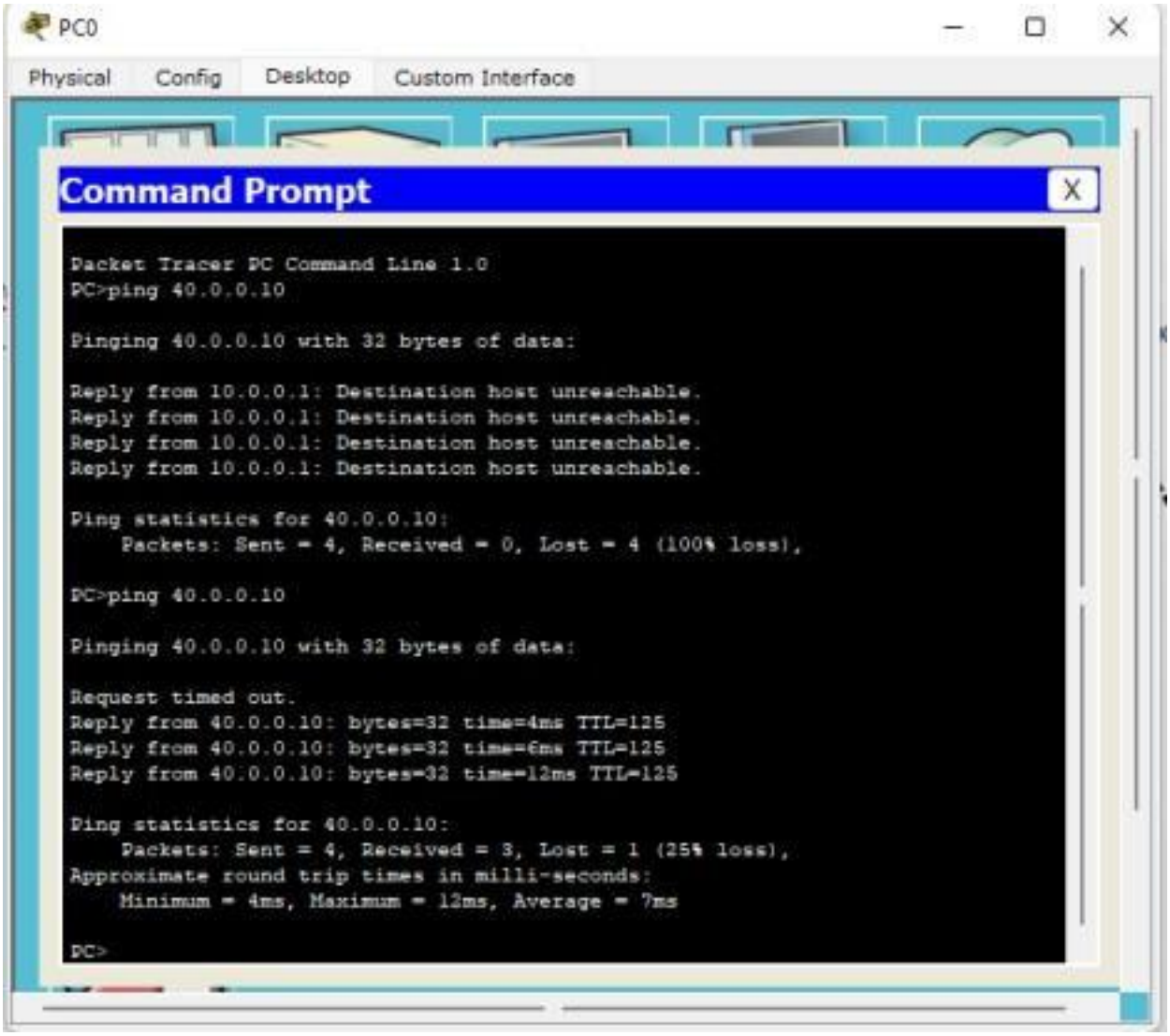
Observation

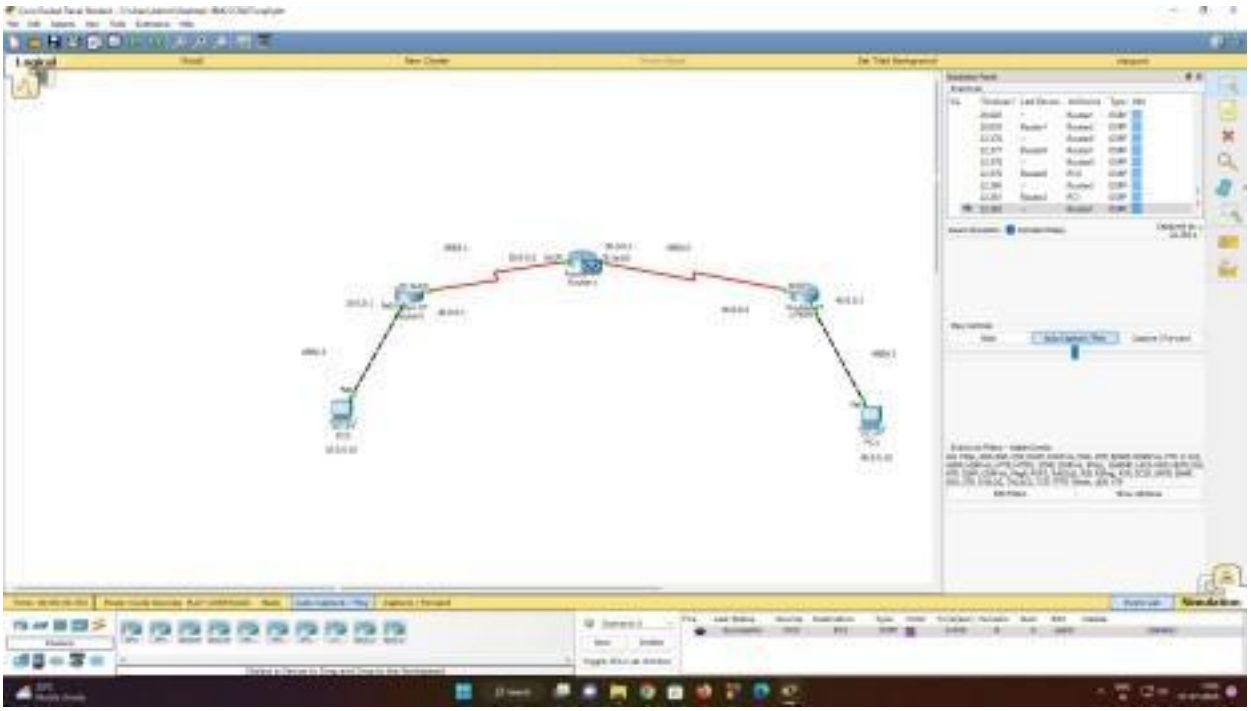
1) OSPF is a link State routing protocol which is used to find the best path b/w source & destination router using its a
2) The network is divided into 4 areas where area is backbone
3) After we make virtual link b/w the area which is not connected to backbone, we can ping message successfully

TOPOLOGY:


AREA 1
30.0.0.2
Se3/0
30.0.0.1
Se3/0
Router-PT
Router1
AREA 0
10.0.0.1
20.0.0.1
30.0.0.2
40.0.0.1
AREA 3
AREA 2
Fa0
PC-PT
PC0
10.0.0.10
PC-PT
PC1
40.0.0.10

OUTPUT:


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
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.
Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.
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).

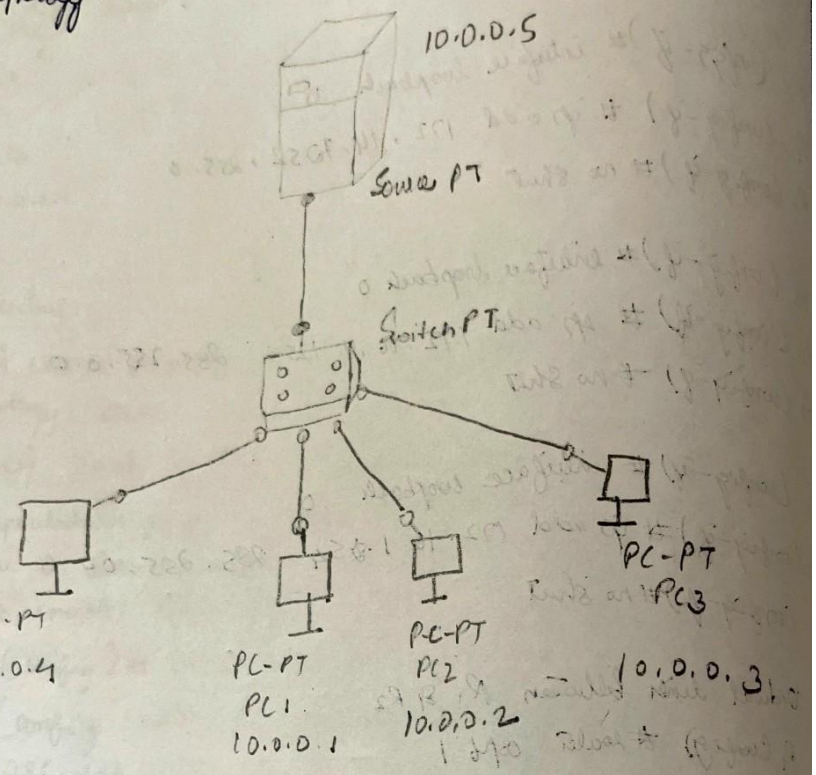
OBSERVATION:

8. lab

Construct a simple LAN & understand the concepts of Address Resolution Protocol (ARP)

Aim:- To construct a simple LAN & understand the concept operation Address Resolution Protocol (ARP)

Topology



Procedure

1) Create a topology of 4 PC's & a Server & IP address are assigned to all PC's, connect them through a switch
2) Use the inspect tool to click on PC to obtain ARP table. Alternatively arp -a in CMD will also do the same.
3) Initially ARP is empty
4) Also in CLI of switch the command

5) Use capture / forward in Simulation panel to go step by step to changes

6) Nodes & switches get updated in ARP table & new connections start

Command Prompt

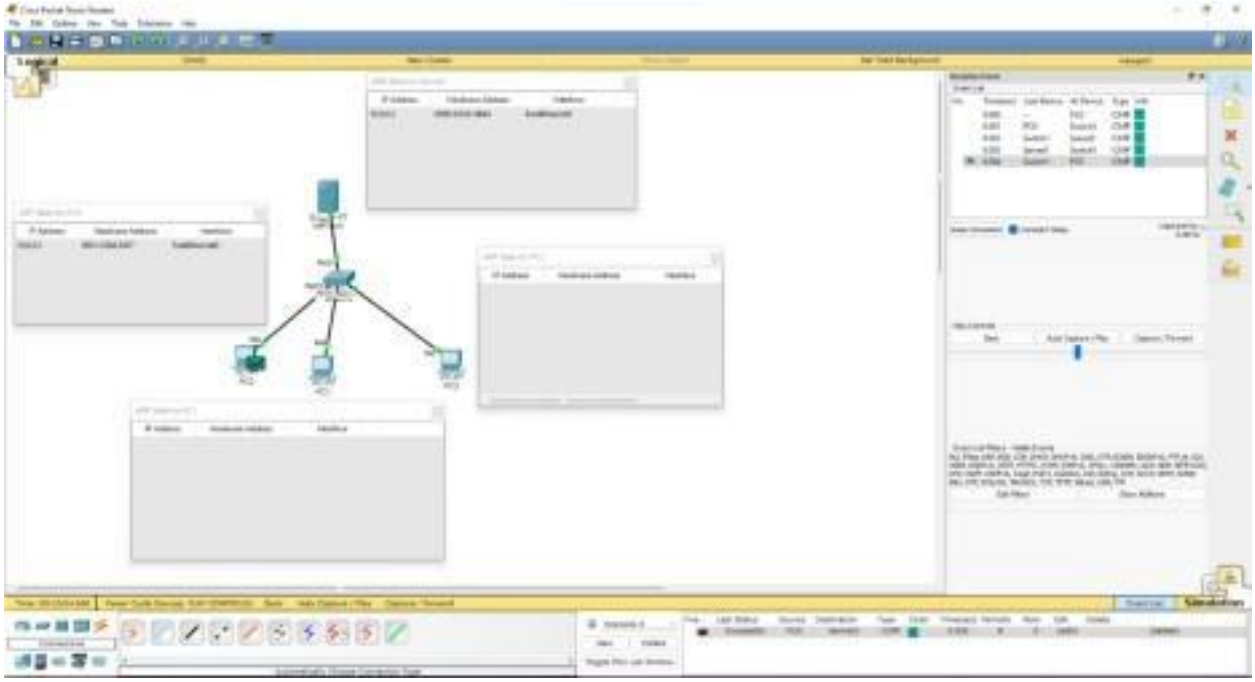
PC > arp -a

internet add	Physical address	Type
10.0.0.2	0005.5e6a.7da2	Dynamic
10.0.0.3	0030.6265.7a19	Dynamic
10.0.0.4	0001.6383.dd62	Dynamic
10.0.0.5	0004.9a42.616c	Dynamic

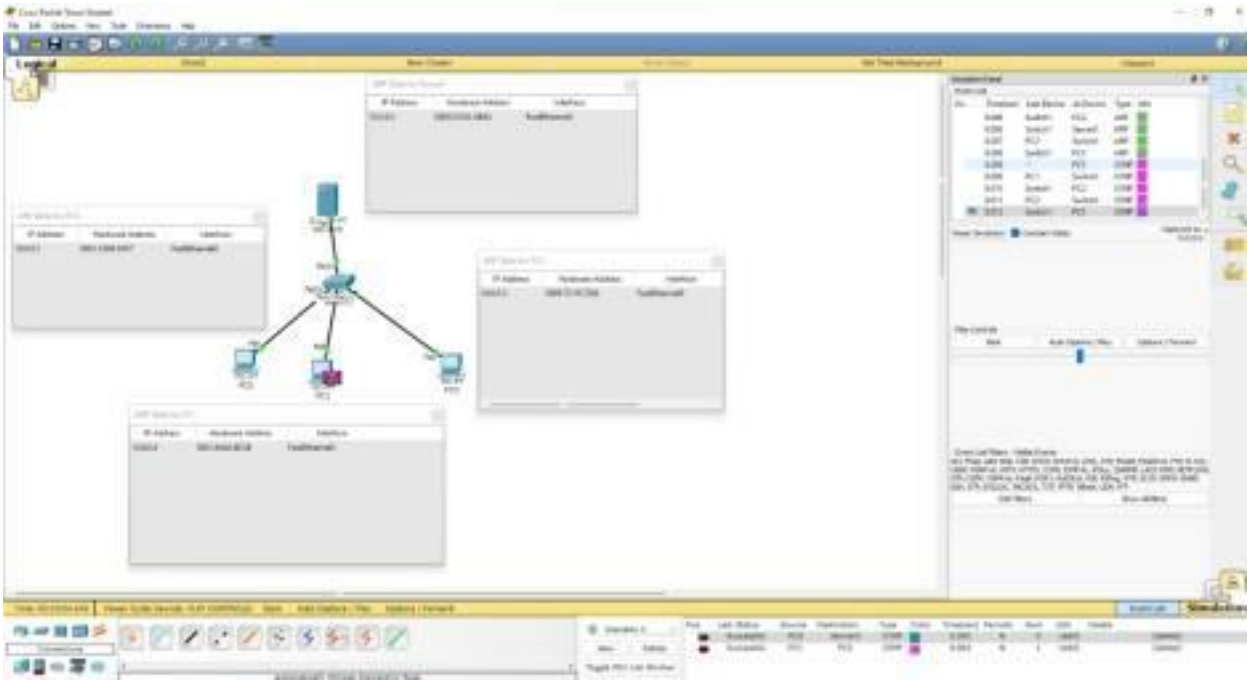
MAC address Table

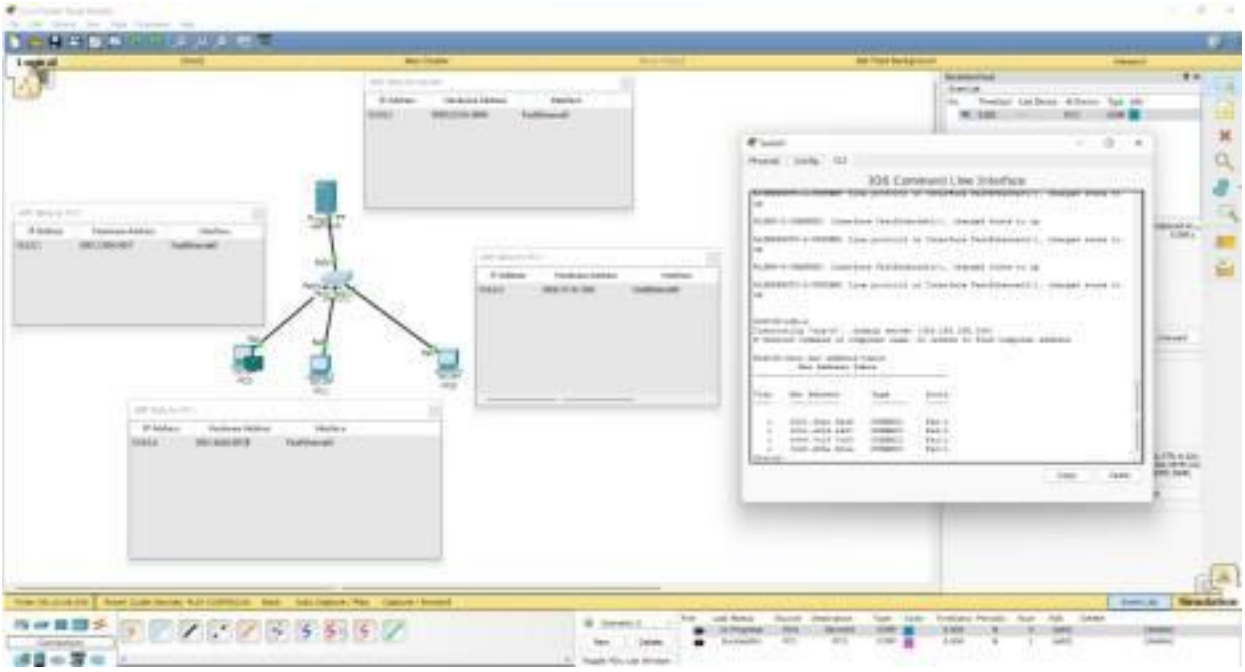
VLAN	MAC Address	Type	Ports
1	0001.683.6dd2	Dynamic	Fa0/1
1	0003.e49d.b2d9	Dynamic	Fa0/1
1	0004.9a42.616c	Dynamic	Fa2/1
1	0005.6e6a.7da2	Dynamic	Eth 6/1
1	0003.6286.7a1a	Dynamic	Fa 3/1

TOPOLOGY:



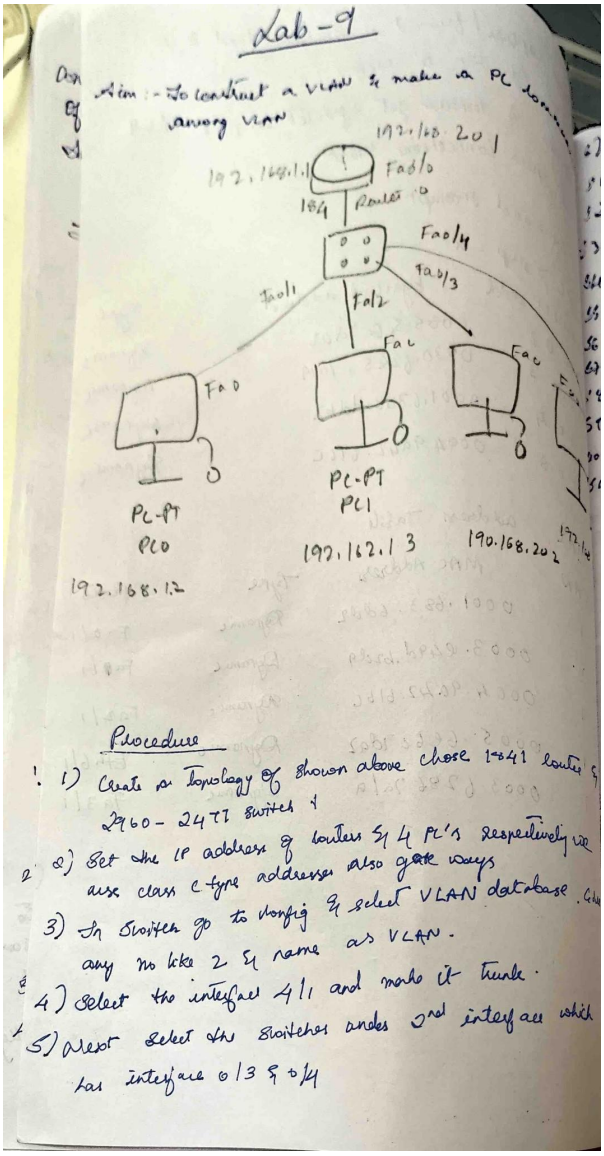
OUTPUT:

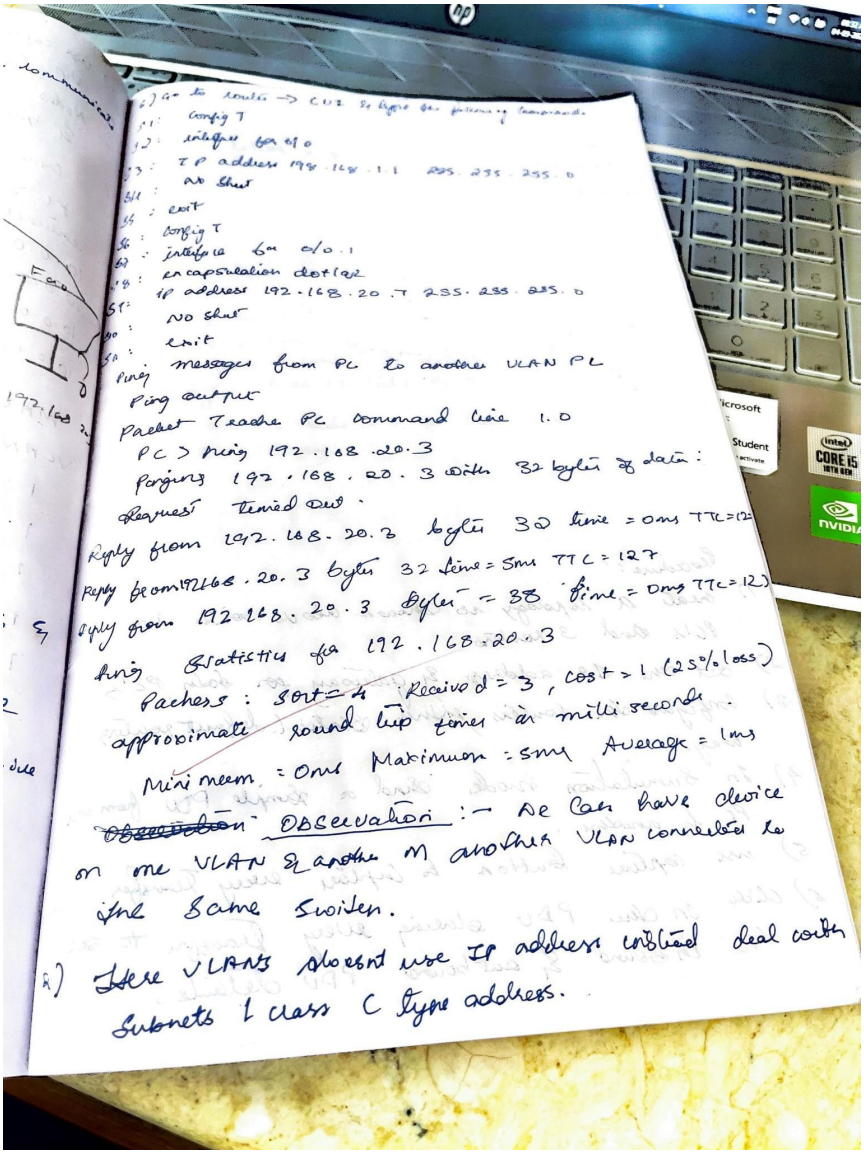




WEEK 9

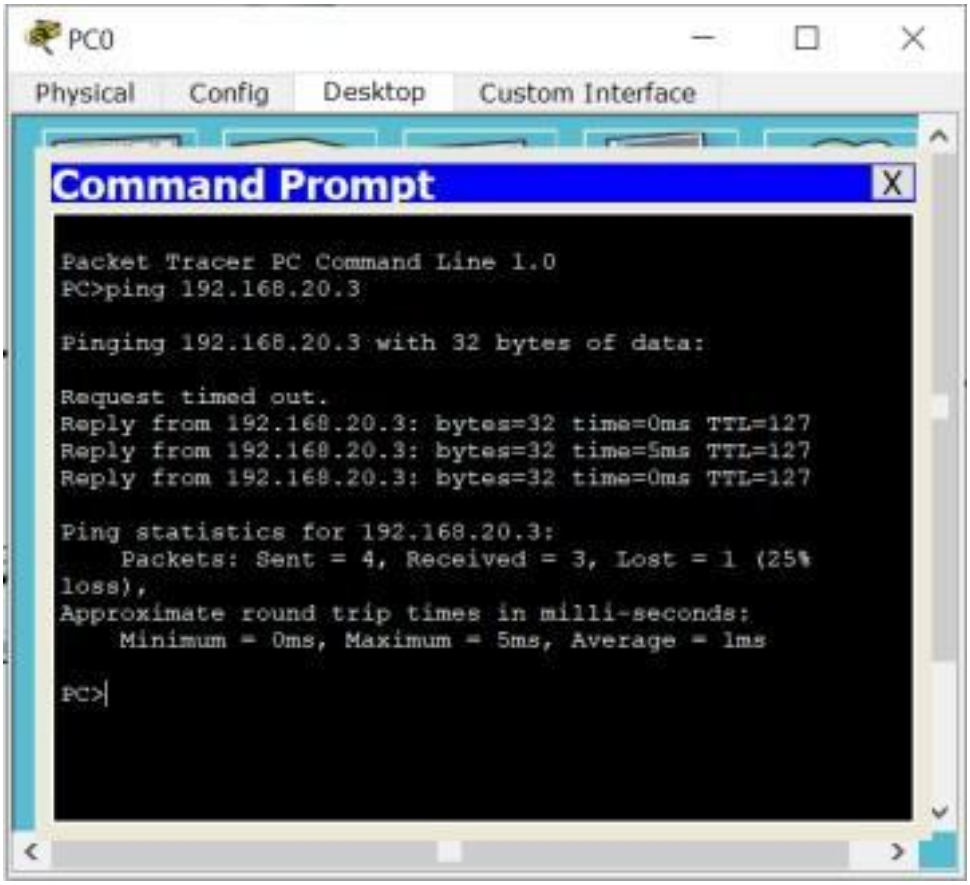
To construct a VLAN and make a pc communicate among VLAN.
OBSERVATION:

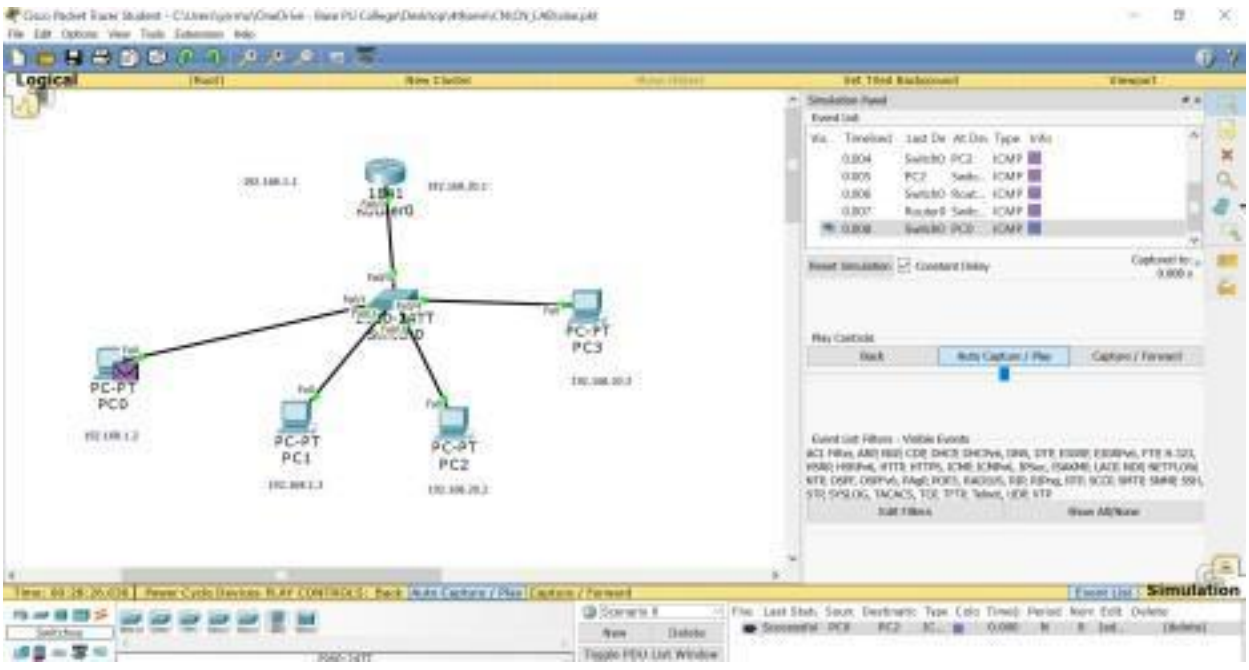




Go to router -> CLI is there for giving commands.
Config T
interface fa 0/0
IP address 192.168.1.1 255.255.255.0
No shut
exit
Config T
interface fa 0/1
encapsulation dot1q
ip address 192.168.20.1 255.255.255.0
No shut
exit
Pinging messages from PC to another VLAN PC
Ping output
Packet Tracer PC command 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=12
Reply from 192.168.20.3 bytes 32 time = 5ms TTL = 127
Reply from 192.168.20.3 bytes = 32 time = 0ms TTL=12
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
Observation :- We can have device on one VLAN & another on another VLAN connected to the same switch.
Q) Here VLANs doesn't use IP address instead deal with Subnets & class C type address.

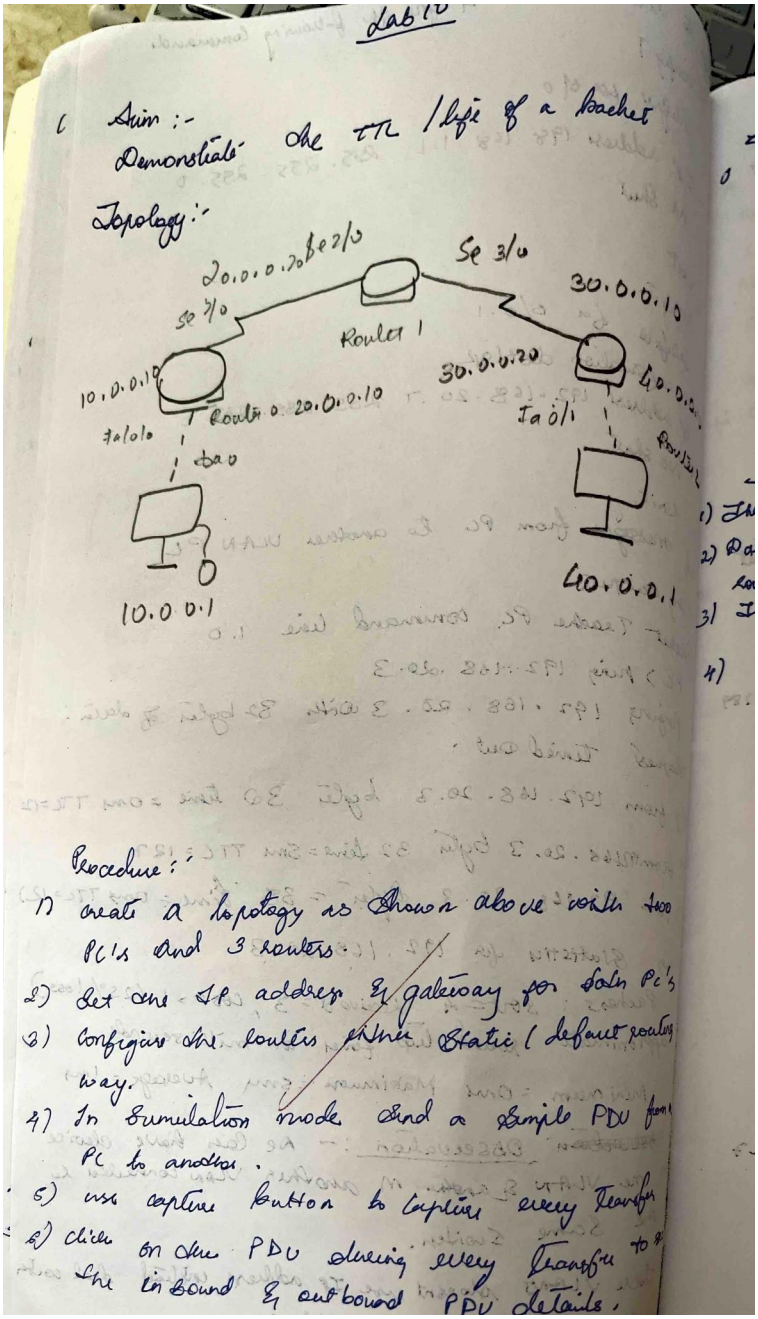
TOPOLOGY and OUTPUT:



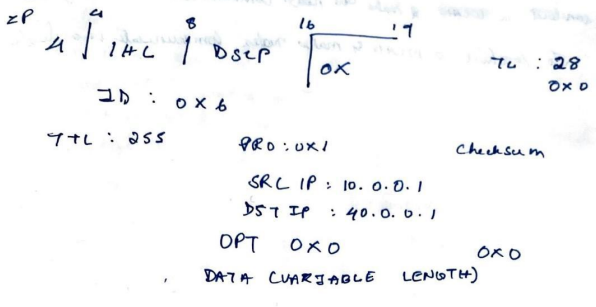


WEEK 10

Demonstrate the TTL/ Life of a Packet.
OBSERVATION:



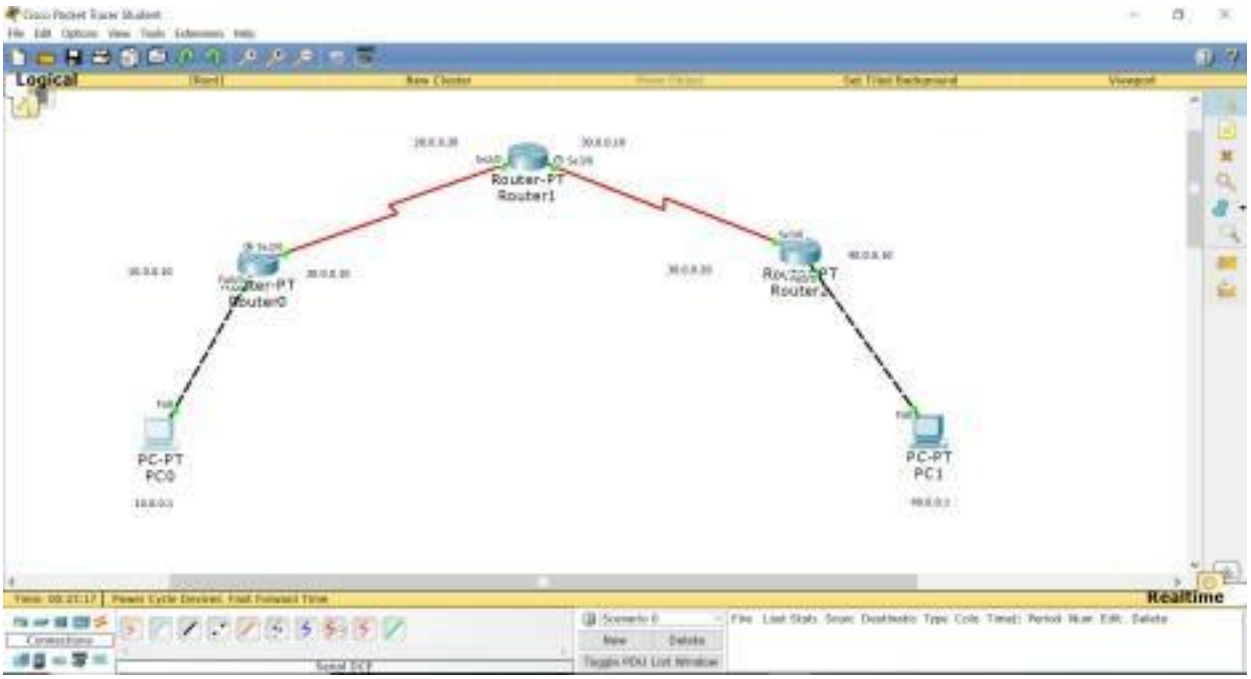
Output :-



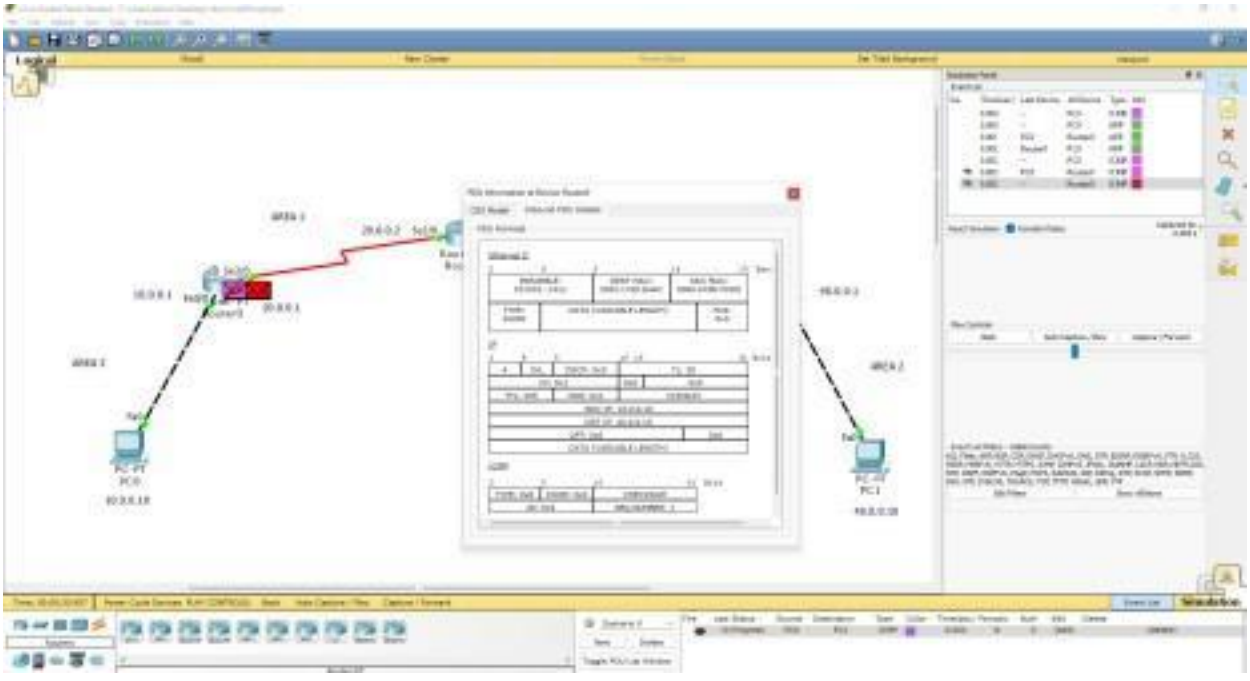
Observation

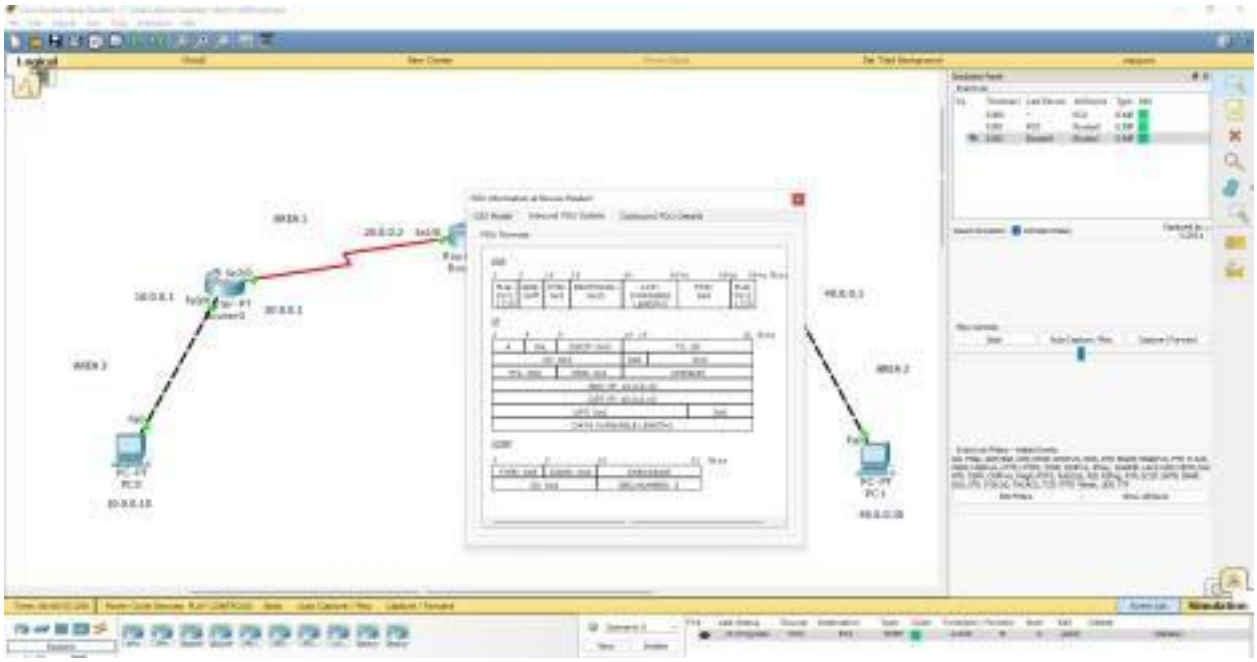
1) It no of hops the packet travel before being discarded as TTL
2) Datagram TTL field is set by the sender & reduced by each router along the path to its destination.
3) IP router reduces TTC value by one while forwarding the packets
4) When the ttc value is 0, the router discards & sends an ICMP message.

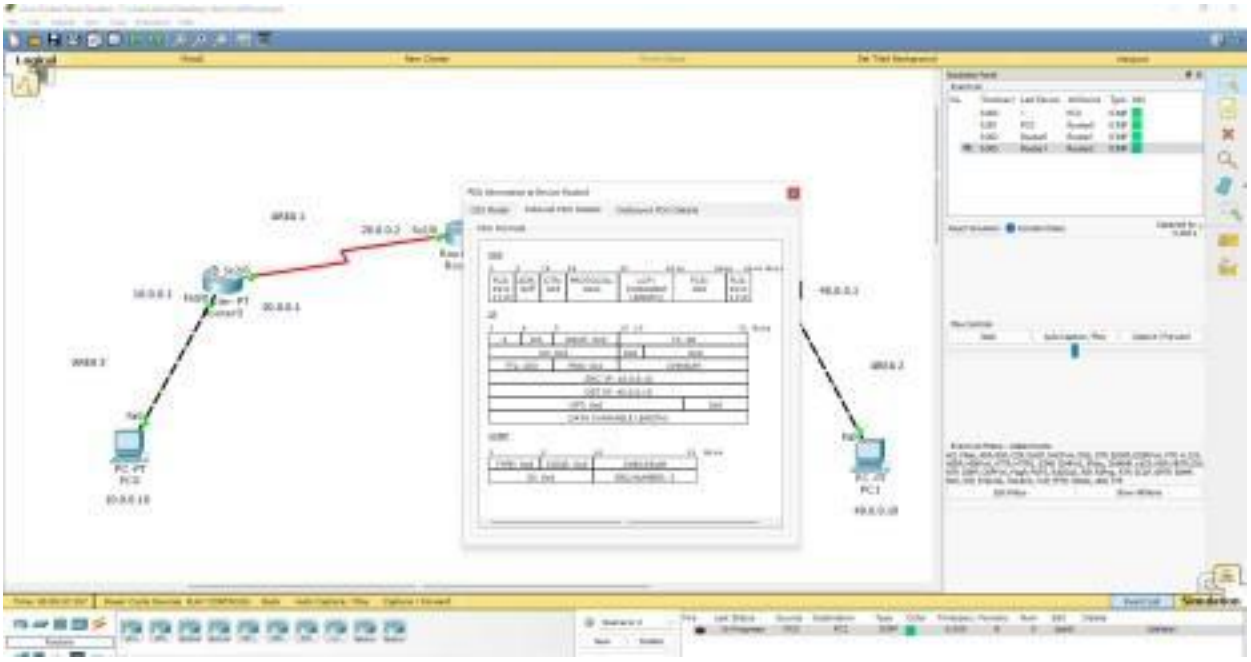
TOPOLOGY:

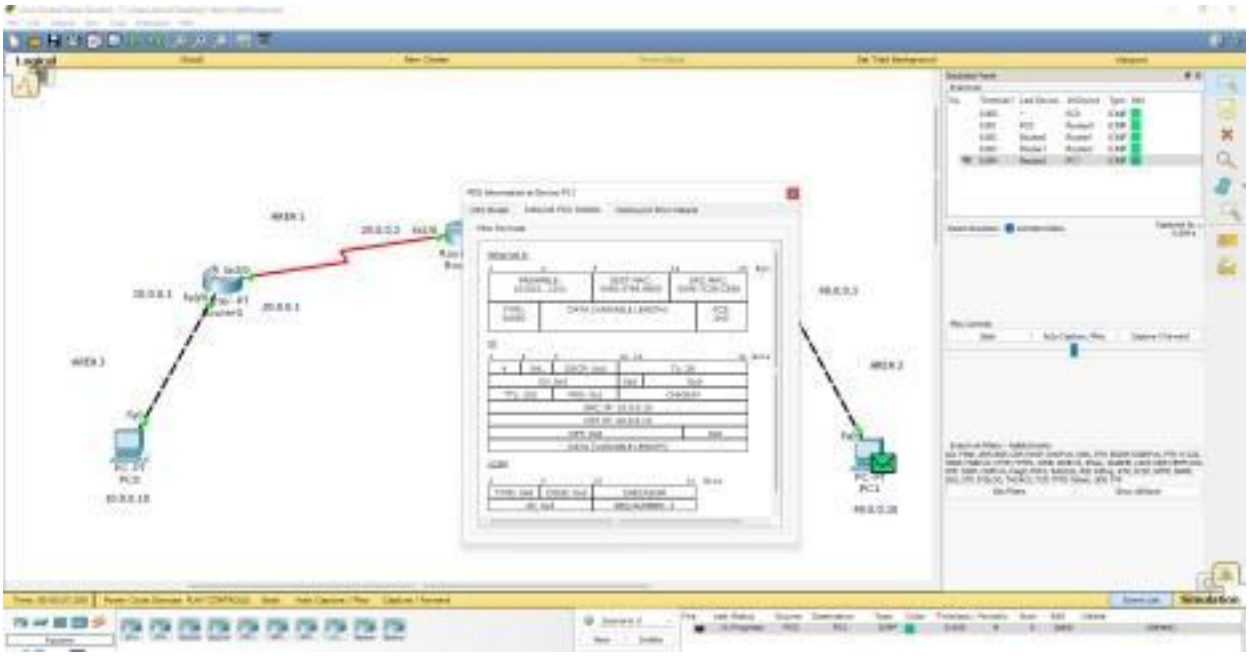


OUTPUT:



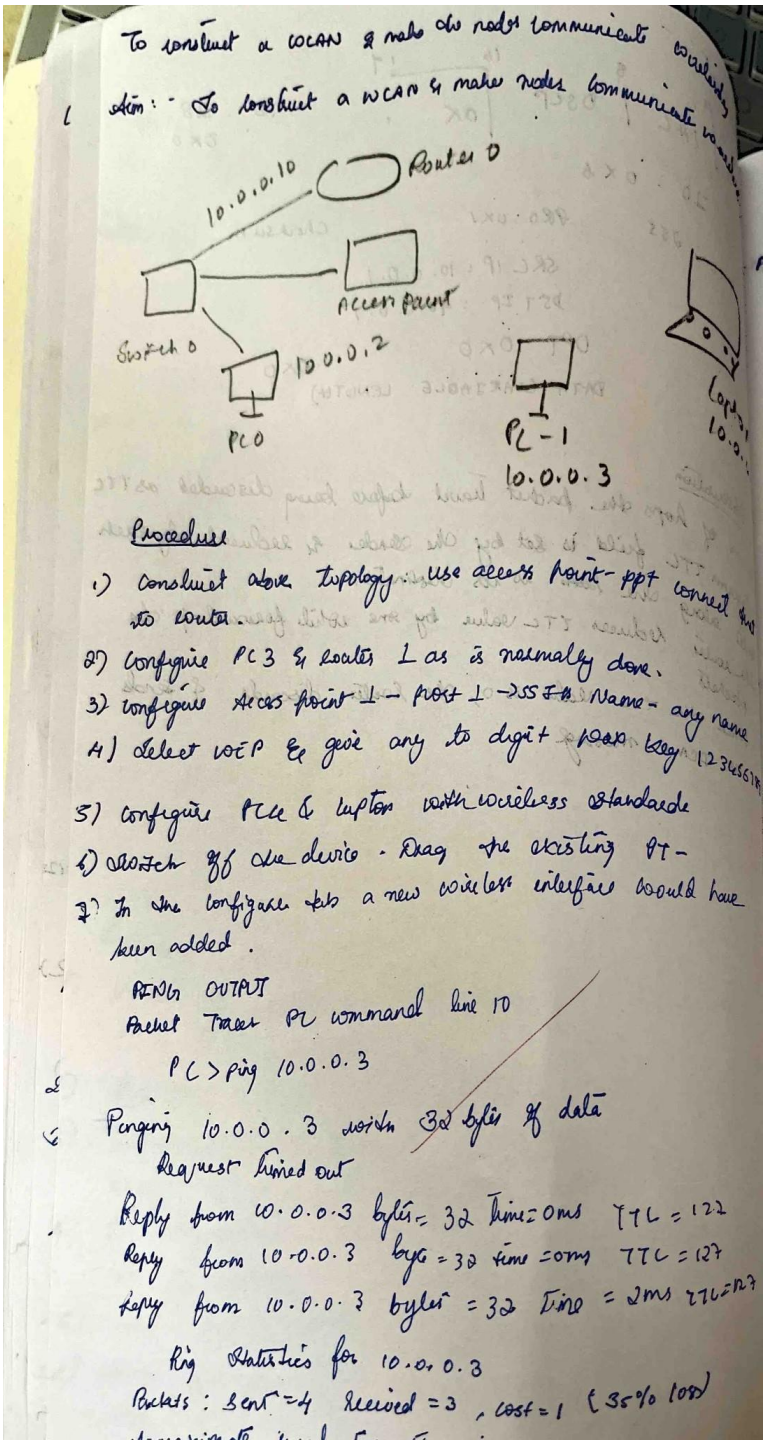






WEEK 11

To construct a WLAN and make the nodes communicate wirelessly
OBSERVATION:



Minimum = 0 ms, Maximum = 1 ms Average = 0 ms

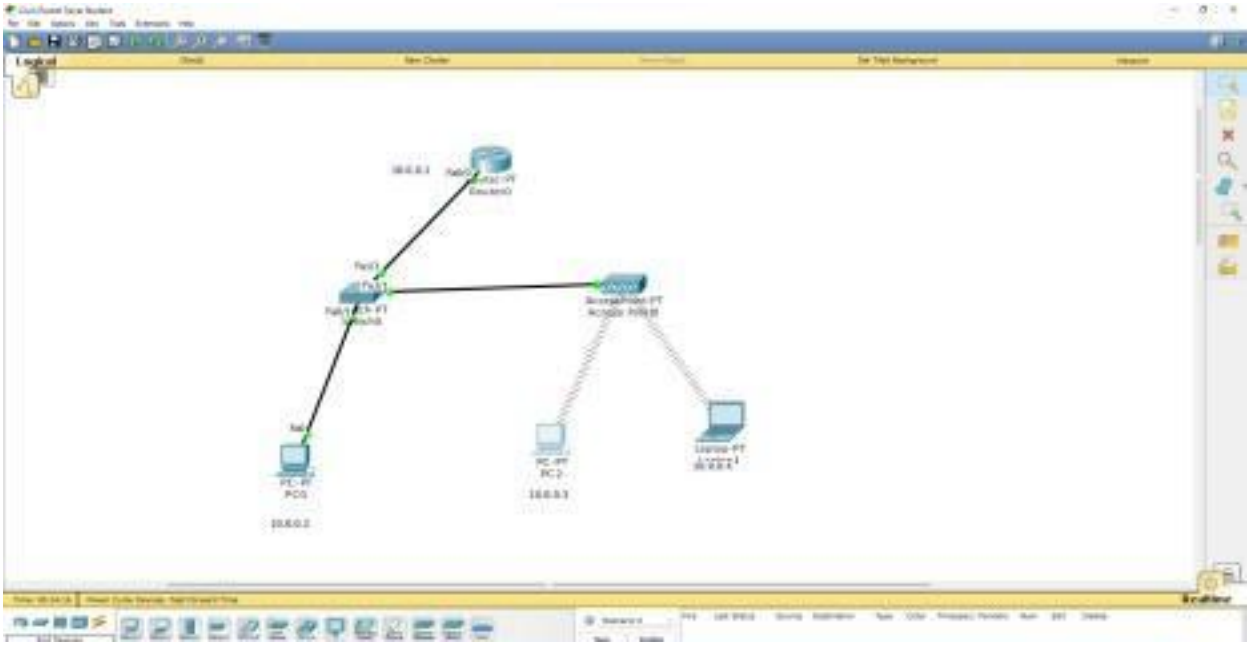
Observation

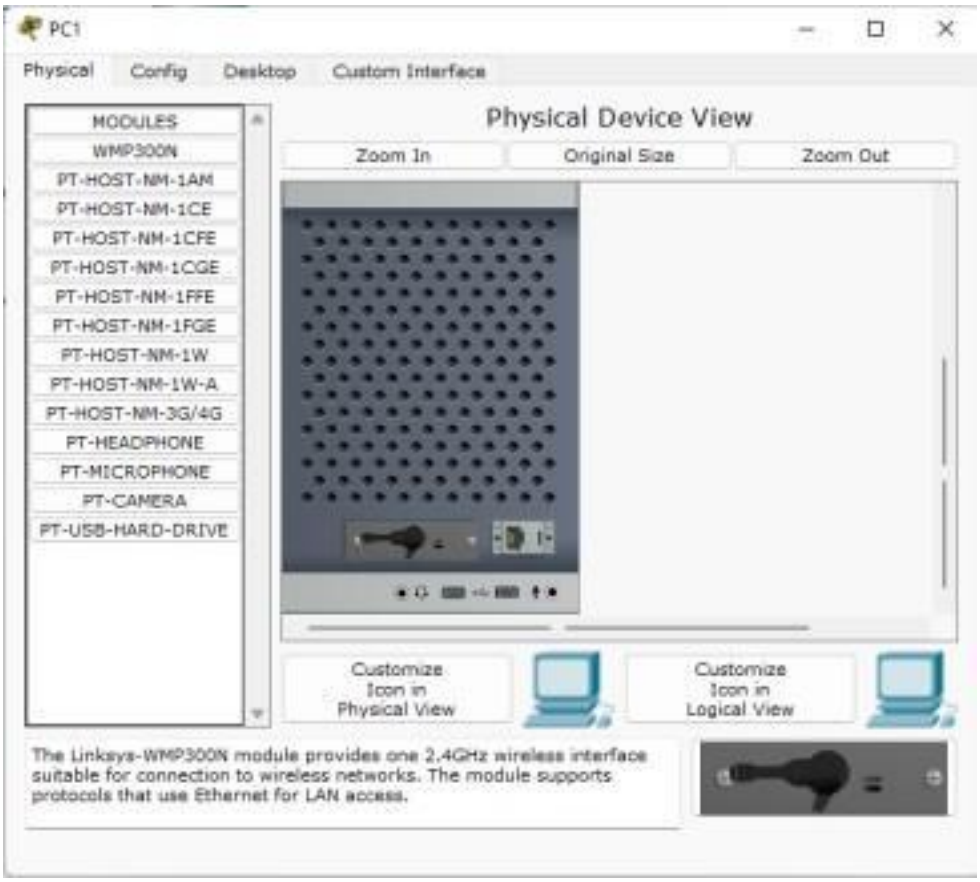
WLAN is a group of colocated devices that form a network based on radio transmissions

Data sent in packets contain layer with labels & instructions - MAC address to end points for routing.

The access point is the base station that serves as a hub to which the users can connect to multiple devices wirelessly & transmit data.

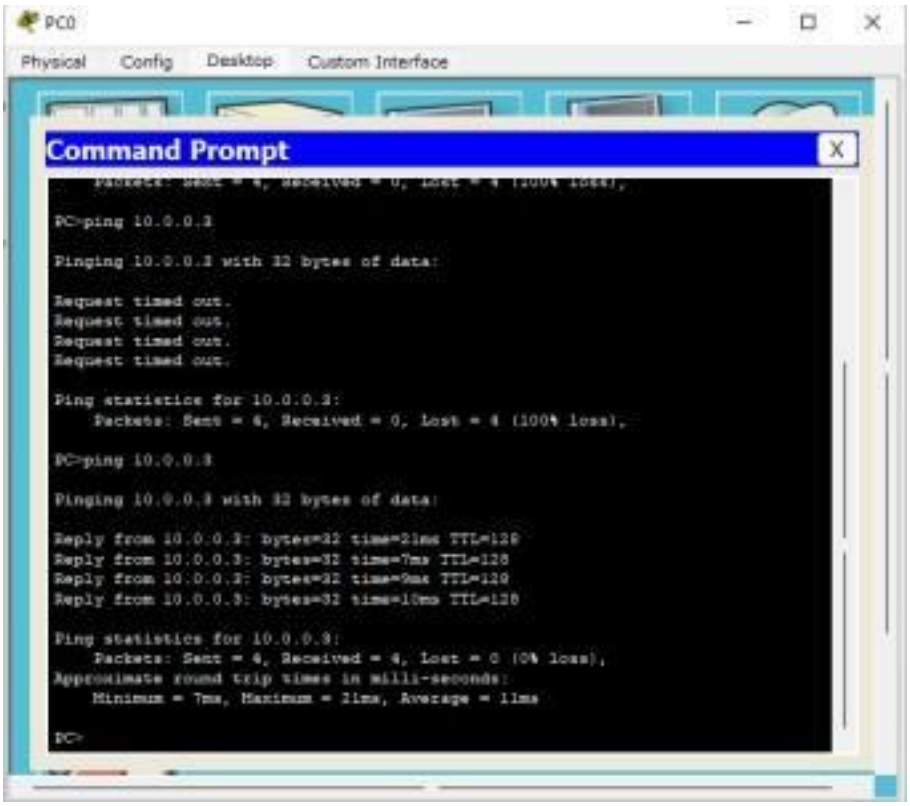
TOPOLOGY:




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.


Laptop0
Physical
Config
Desktop
Custom Interface
MODULES
WPC300N
PT-LAPTOP-NM-1AM
PT-LAPTOP-NM-1CE
PT-LAPTOP-NM-1CFE
PT-LAPTOP-NM-1CGE
PT-LAPTOP-NM-1FFE
PT-LAPTOP-NM-1FGE
PT-LAPTOP-NM-1W
PT-LAPTOP-NM-1W-A
PT-LAPTOP-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-WPC300N module provides one 2.4GHz wireless interface suitable for connection to wireless networks. The module supports protocols that use Ethernet for LAN access.

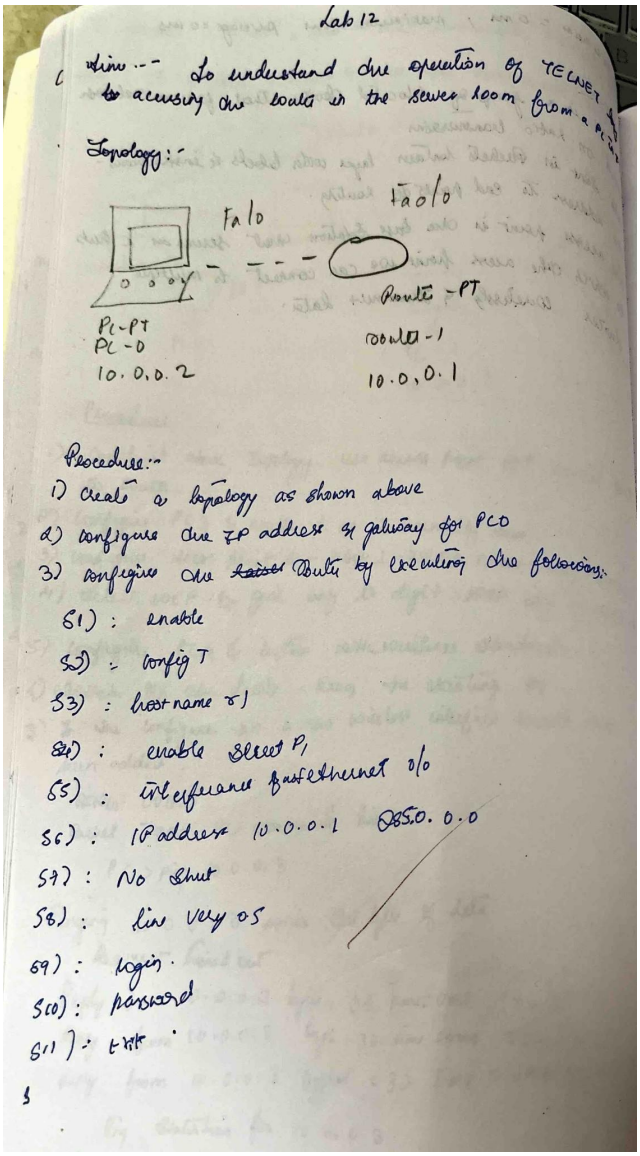
OUTPUT:

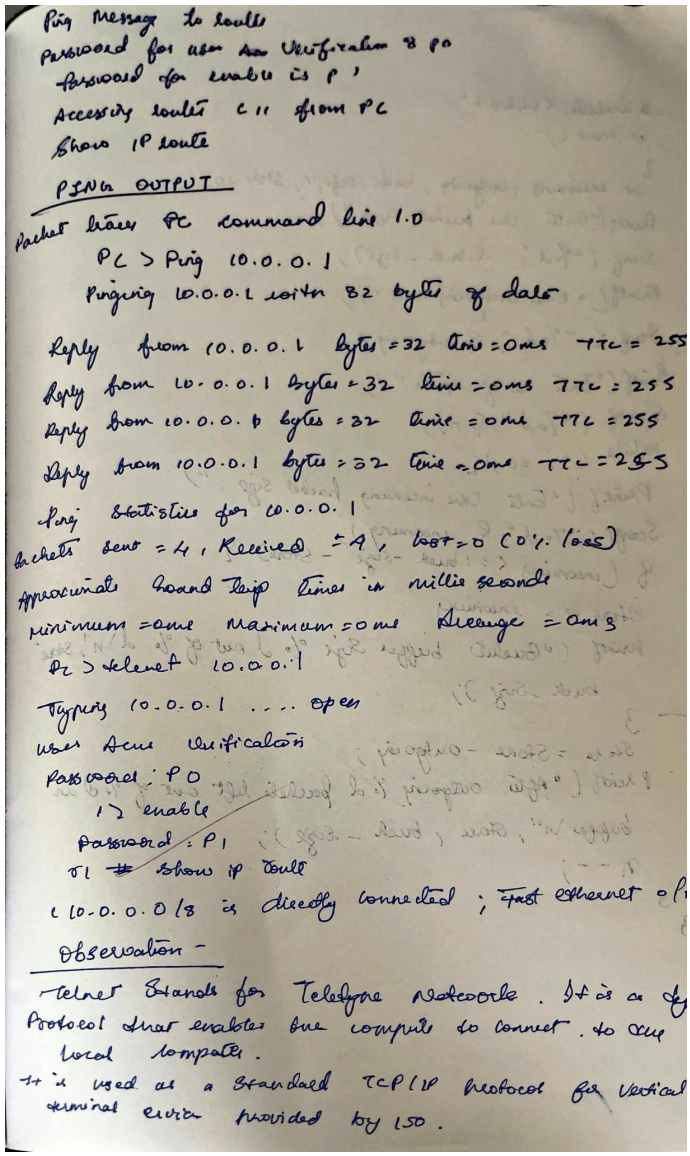

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
PC>ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 10.0.0.3:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data:
Reply from 10.0.0.3: bytes=32 time=2ms 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:





Ping Message to router
Password for user Access Verification is P0
Password for enable is P1
Accessing router CLI from PC
Show IP route

PING 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=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
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 = 0ms Average = 0ms

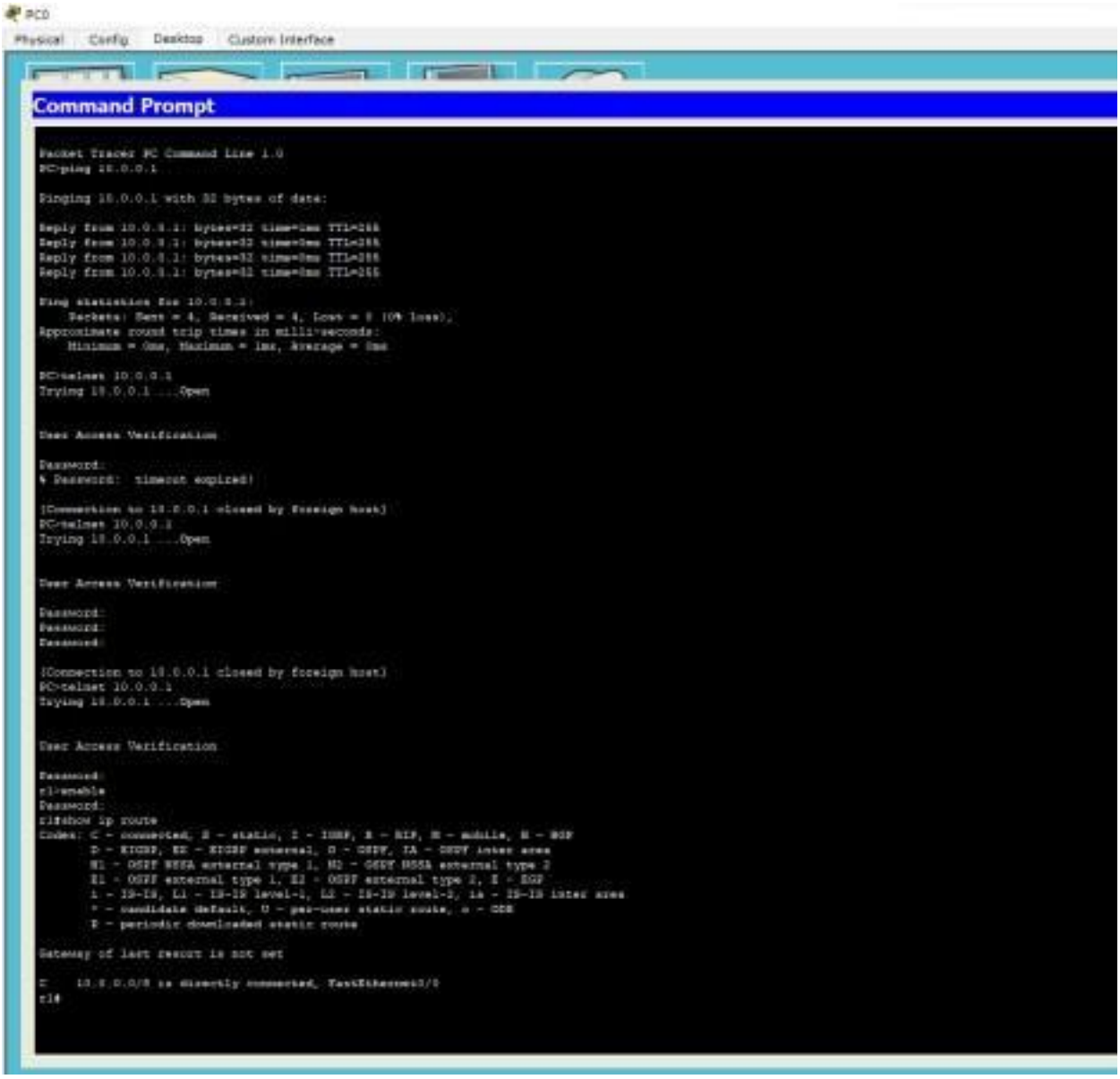
PC > telnet 10.0.0.1

Trying 10.0.0.1 ... Open
User Access Verification
Password: P0
1> enable
Password: P1
r1 # show ip route
C 10.0.0.0/8 is directly connected, FastEthernet 0/1

Observation -

Telnet stands for Teletype Network. It is a type of protocol that enables one computer to connect to and local computer.
It is used as a standard TCP/IP protocol for virtual terminal service provided by ISO.

OUTPUT:


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
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:
r1>enable
Password:
r1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route
Gateway of last resort is not set
C 10.0.0.0/8 is directly connected, FastEthernet0/0
r1#

WEEK 13

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

CODE:

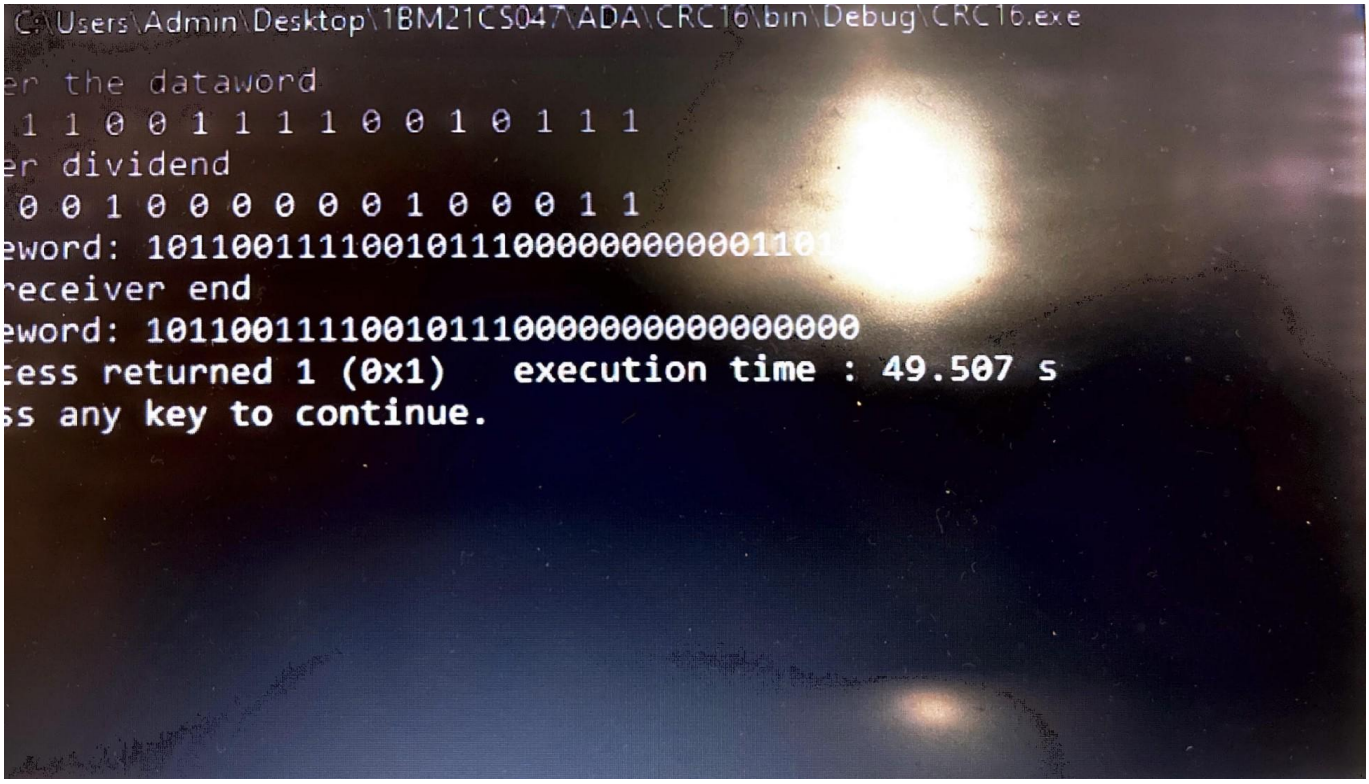
```
#include<stdio.h>
int arr[17];

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);
```

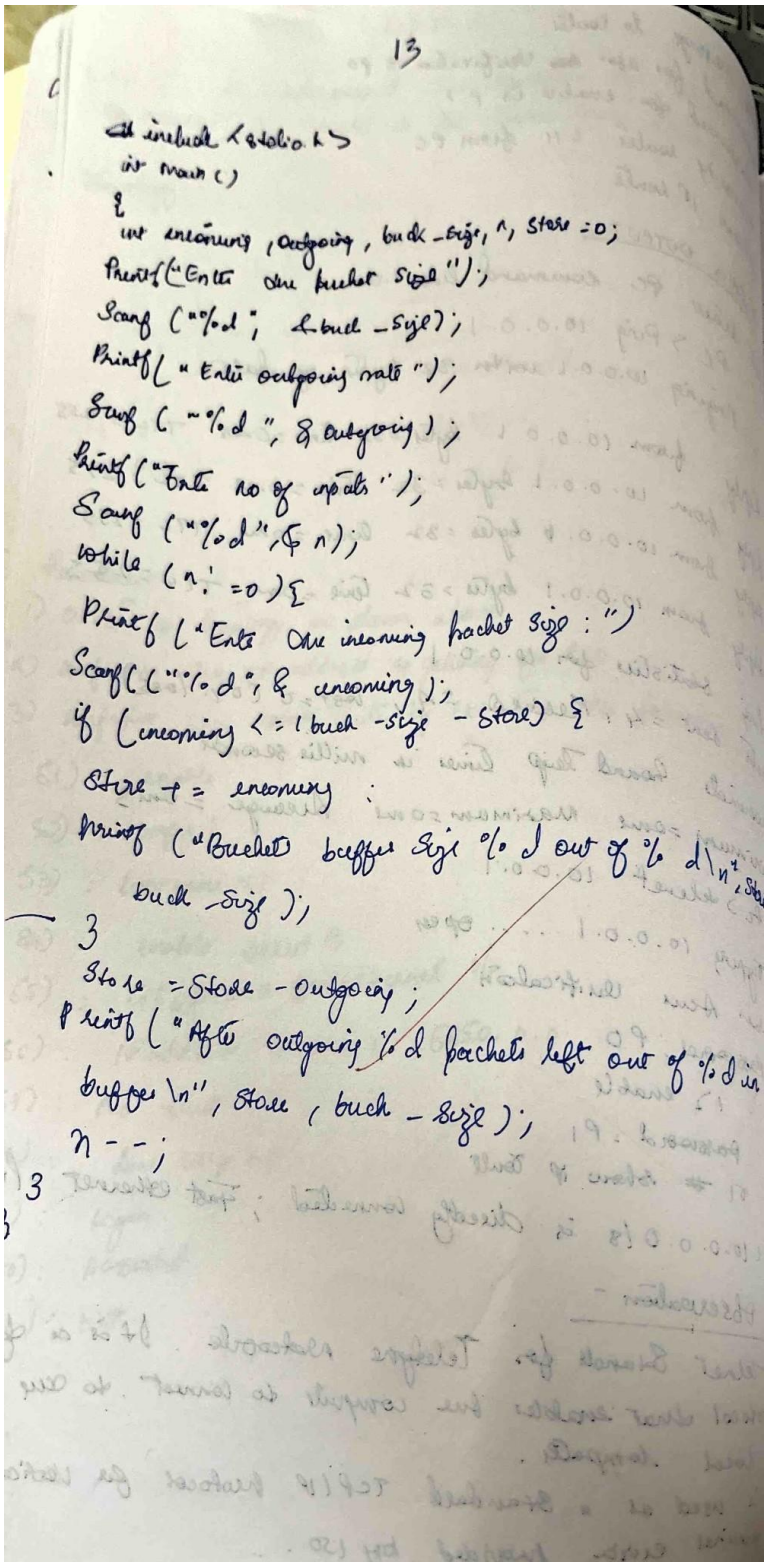
```
    arr[16]=div[i++];

}
```

```
    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:



OBSERVATION:



```
13

#include <stdio.h>
int main()
{
int incoming, outgoing, buck_size, n, store = 0;
Printf("Enter the bucket size");
Scanf("%d", &buck_size);
Printf("Enter outgoing rate");
Scanf("%d", &outgoing);
Printf("Enter no of inputs");
Scanf("%d", &n);
while (n != 0){
Printf("Enter the incoming packet size:")
Scanf("%d", &incoming);
if (incoming <= (buck-size - store){
store += incoming;
Printf("Bucket buffer size %d out of %d\n", st
    buck_size);
}
store = store - outgoing;
Printf("After outgoing %d packets left out of %d in
buffer\n", store, buck_size);
n--;
}
```

Output

Enter no of Queues, buffer size, input & output Packet Size

4
7
4
1

Packet is accepted rem. space = 3

Packet is accepted rem. space = 0

Packet not accepted rem space = 1

Packet not accepted rem space = 2

WEEK 14

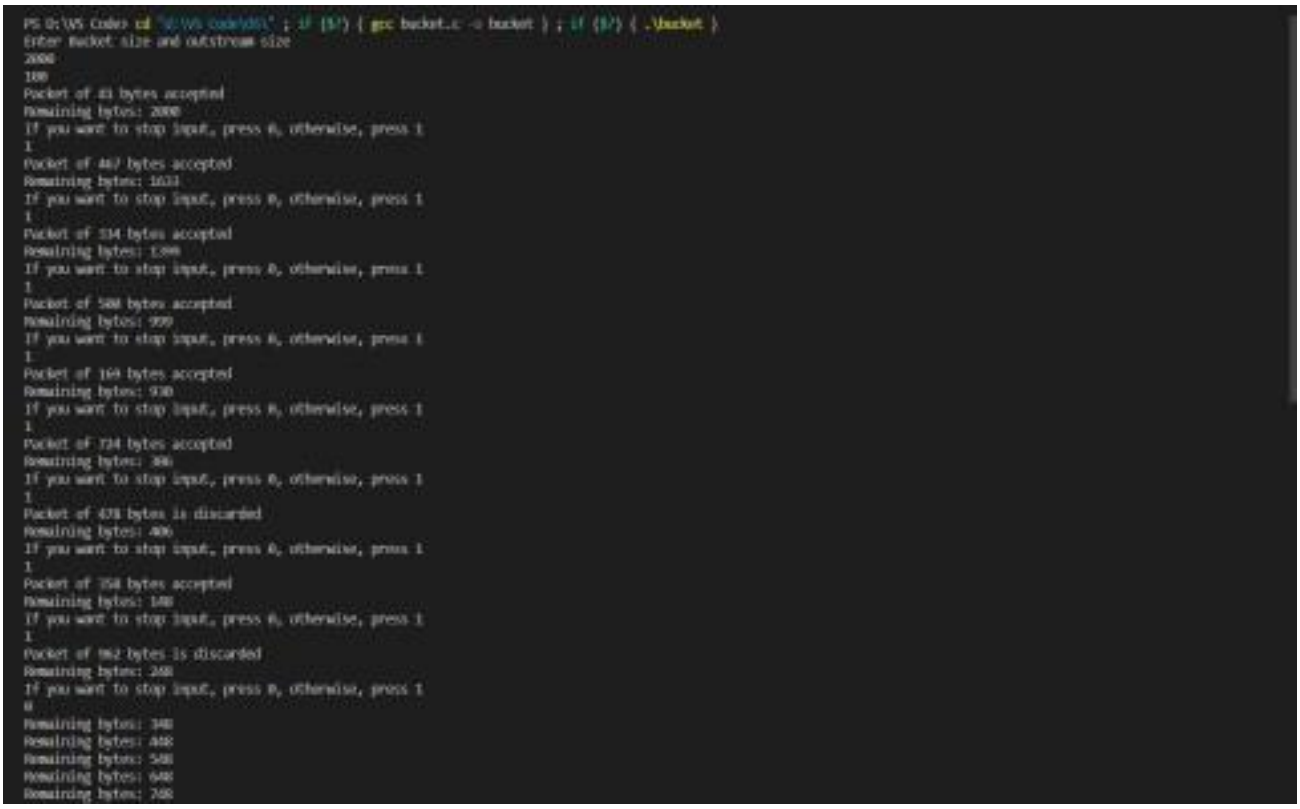
Write a program for congestion control using Leaky bucket algorithm.

CODE:

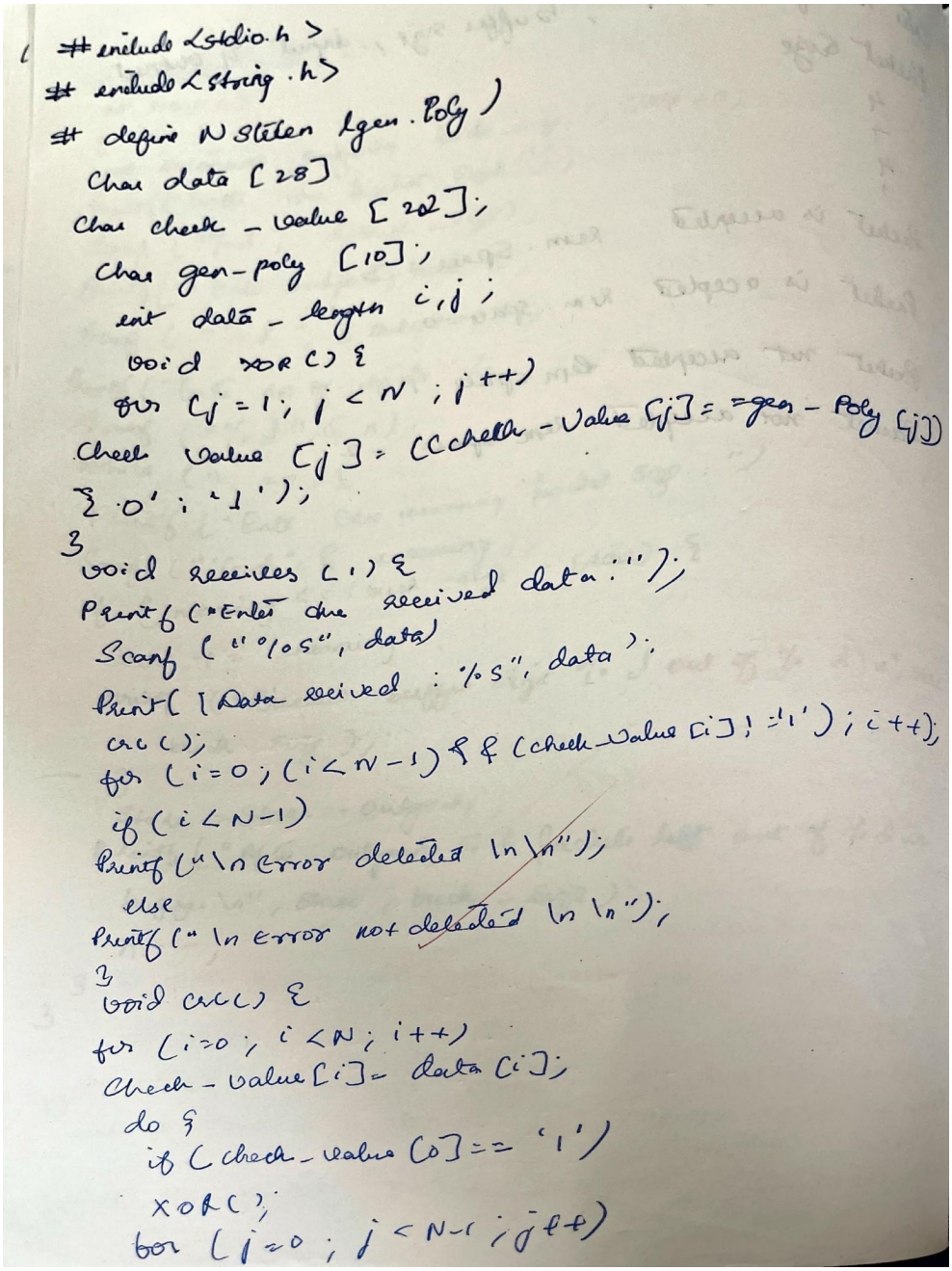
```
#include <stdio.h>
#include <stdlib.h> // Include this for the rand() function
int main()
{
   int buckets, outlets, k = 1, num, remaining;
   printf("Enter Bucket size and outstream size\n");
   scanf("%d %d", &buckets, &outlets);
   remaining = buckets;
   while (k)
   {
      num = rand() % 1000; // 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:



OBSERVATION:



```
#include <stdio.h>
#include <string.h>
#define N strlen(gen_poly)
  char data[28]
  char check_value[28];
  char gen_poly[10];
  int data_length,i,j;
  void XOR() {
  for(j=1; j<N; j++)
  check_value[j] = (check_value[j] == gen_poly[j])
  ? '0' : '1');
}
void receive() {
Printf("Enter the received data:");
 Scanf("%s", data)
Printf("Data received : %s", data);
 crc();
for(i=0;(i<N-1) && (check_value[i]!='1'); i++);
 if(i<N-1)
Printf("\n Error detected \n\n");
  else
Printf("\n Error not detected \n\n");
}
void crc() {
for(i=0; i<N; i++)
 check_value[i] = data[i];
  do {
   if(check_value[0] == '1')
   XOR();
   for(j=0; j<N-1; j++)
```

```
check_value[j] = check_value[j+1]
check_value[j] = data[i++]
} while (i <= data_length + N-1)
}
int main()
{
printf("\n Enter data to be transmitted \n");
scanf("%s", &data);
printf("\n Enter the divisor polynomial \n");
scanf("%s", &gen_poly);
data_length = strlen(data);
for (i = data_length; i < data_length + N-1; i++)
data[i] = '0';
printf("\n Data Padded with n-1 zeros : %s", data);
crc();
printf("\n CRC value is : %s", check_value);
for (i = data_length; i < data_length + N-1; i++)
data[i] = check_value[i - data_length];
printf("\n check_value to be sent : %s", data);
printf("\n ..... \n");
receive();
return 0;
}
```

Output

Enter the data word : 1100101011100100

Calculated CRC : 1110100101110001

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()
```

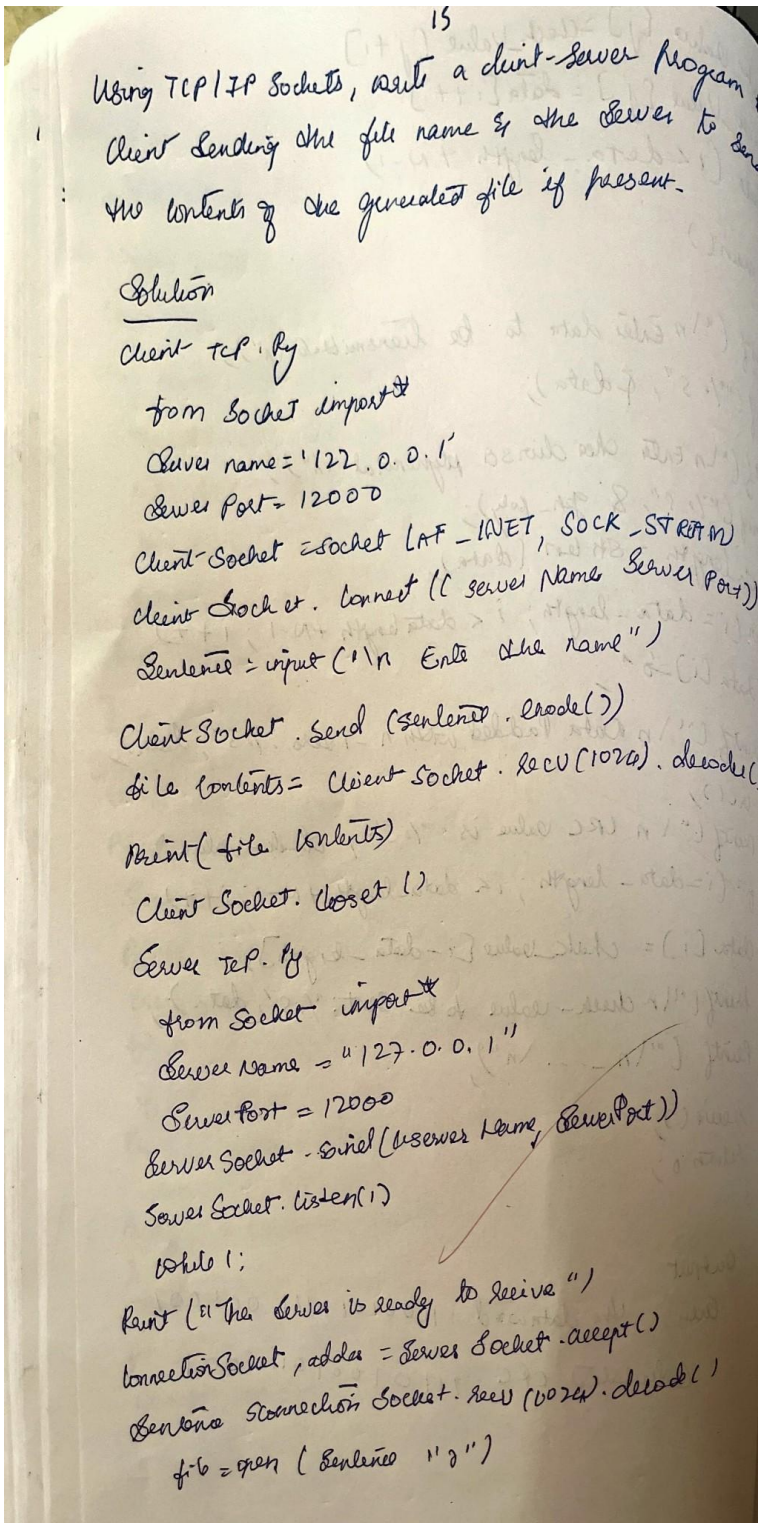
ServerTCP.py

```
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:

OBSERVATION:



15

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

Solution

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 ("\n Enter the name")
Client Socket. Send (Sentence. encode())
file contents = Client Socket. recv (1024). decode()
Print (file contents)
Client Socket. Close ()
```

Server TCP. Py

```
from Socket import *
Server name = "127.0.0.1"
Server Port = 12000
Server Socket. Bind ((Server Name, Server Port))
Server Socket. listen (1)
while 1:
Print ("The Server is ready to receive")
connection Socket, addr = Server Socket. accept()
Sentence = connection Socket. recv (1024). decode()
    f = open (Sentence, "r")
```

l = file.read(1024)
connectionSocket.send(l.encode())
Print("\n Sent contents of " + sentence)
file.close()
connectionSocket.close()

Output

Restart: C:\Users\Admin\AppData\Local\Programs\Python\Python310\Server_tcp.py

The server is ready to receive
Sent contents of the server_tcp.py
The server is ready to receive

Restart: C:\Users\Admin\AppData\Local\Program\Python\Python310\client_tcp.py

Enter file name: Server_tcp.py
from socket import
Server name = "127.0.0.1"
Server port = 12000
Server Socket = socket(AF_INET, Sock_STREAM)
-> (contents sent by the server displayed here

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:

OBSERVATION:

2) using UDP Sockets write a Client-Server program to make client sending the file name & the server to send back the contents of the requested file if present.

→ client UDP.py

```
from Socket import *
ServerName = "127.0.0.1"
ServerPort = 12000
ClientSocket = socket(AF_INET, SOCK_DGRAM)
Sentence = input("\n Enter the file name:");
ClientSocket.sendto(bytes(Sentence,"utf-8"),(ServerName,
Sentence port))
file contents, Server address = clientSocket.recvfrom(2048)
Print("\n Reply from server: \n")
Print(file contents, decode("utf-8"))
# for i in file contents:
# print(str(i), end="")
    ClientSocket.close()
    ClientSocket.close()
```

ServerUDP.py

```
from Socket import *
ServerPort = 12000
ServerSocket = socket(AF_INET, SOCK_DGRAM)

while 1:
Sentence, clientAddress = ServerSocket.recvfrom(2048)
Sentence = Sentence.decode("utf-8")
    file = open(sentence, "r")
```

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

Output

Server

The Server is ready to receive

Client

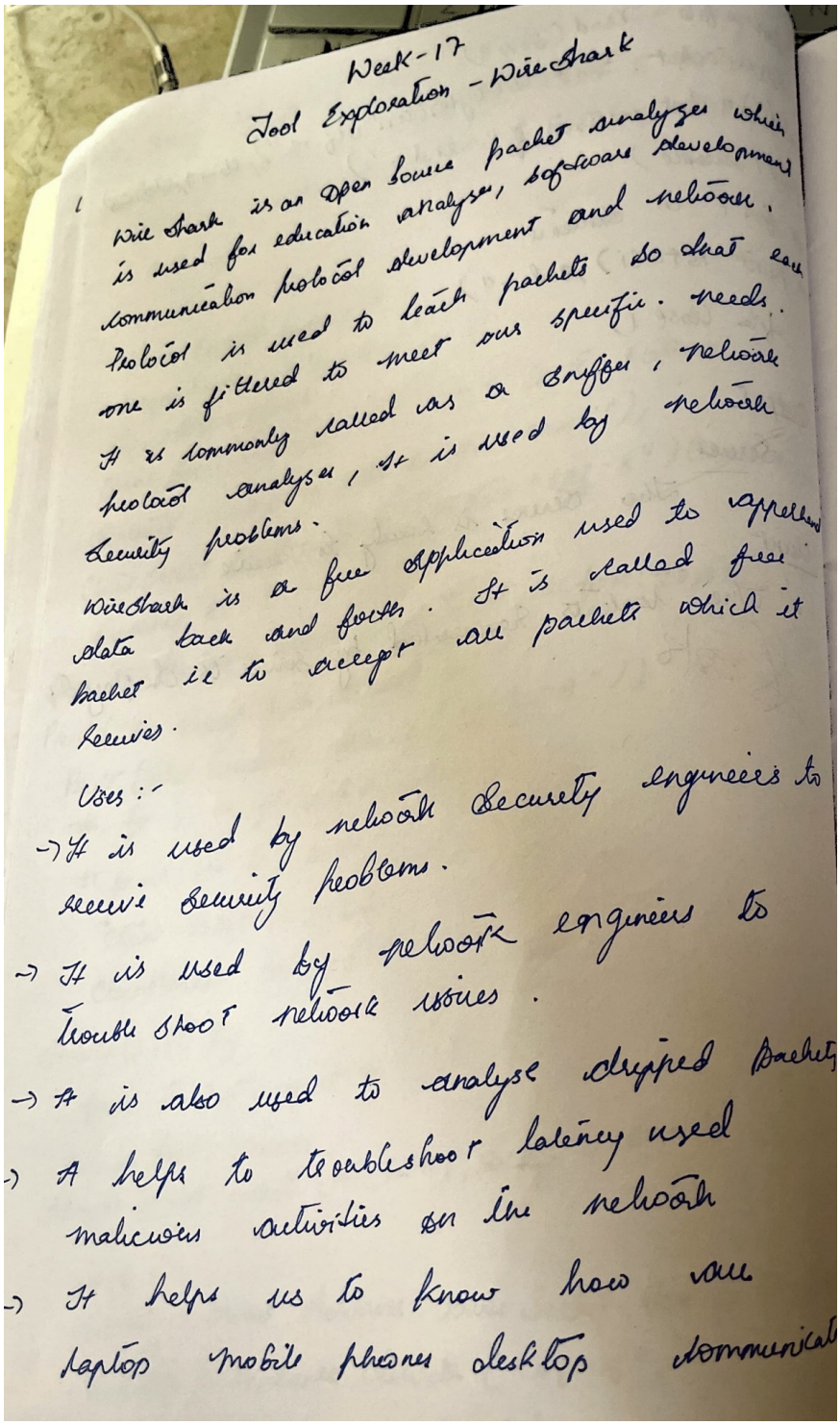
The contents requested by server is displayed

22/8

WEEK 17

Tool Exploration -Wireshark

OBSERVATION:



Week-17

Tool Exploration - Wireshark

Wireshark is an open source packet analyzer which is used for education analysis, software development communication protocol development and network protocol is used to each packets so that each one is filtered to meet our specific needs. It is commonly called as a sniffer, network protocol analyser, it is used by network security problems.

Wireshark is a free application used to capture data back and forth. It is called free packet i.e to accept all packets which it receives.

Uses :-

-) It is used by network security engineers to access security problems.

-) It is used by network engineers to trouble shoot network issues.

-) It is also used to analyse dropped packets

-) It helps to troubleshoot latency used malicious activities on the network

-) It helps us to know how your laptop mobile phones desktop communicate

Functionality of Wireshark

It is similar to a TCP dump networking. It has a graphic end filtering functions. It also unicast traffic which is not to networks mac address interface host networking method to network traffic. When it is enabled switch sends copies of all network packet present at port to another port.

Features of Wireshark

It is a multi platform software ie it can run on linux, Windows OSX, Free BSD etc:- It is a standard three pane, packet It performs deep inspection of protocols. It performs best & filter option which makes easy to user to view. the data

- It can capture raw USB traffic
- It is useful in IP address.
- It also evolves live analysis, ie from different types of network like ethernet loopback etc through which we can read live data.