VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

COMPUTER NETWORKS

Submitted by

AMSHU G M (1BM21CS019)

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 **AMSHU G M (1BM21CS019),** 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. SHYAMALA T.R.
Assistant Professor
Department of CSE
BMSCE, Bengaluru

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

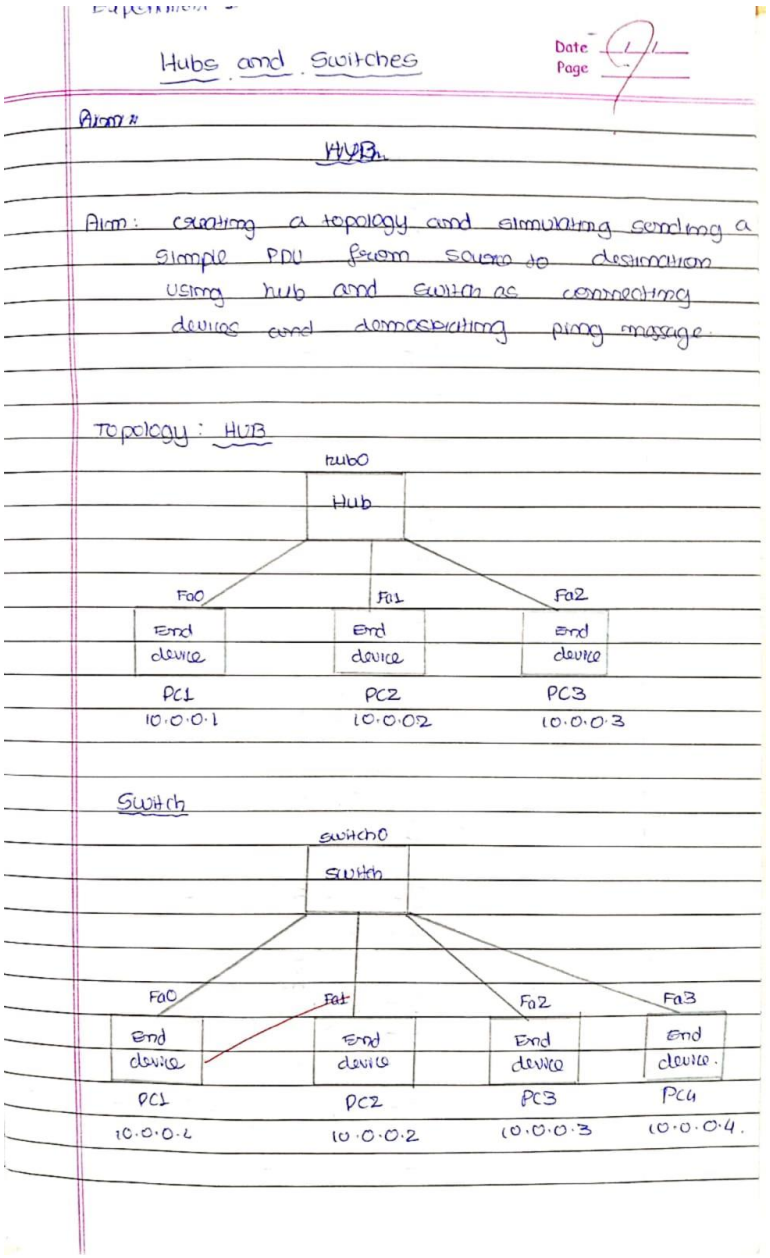
`

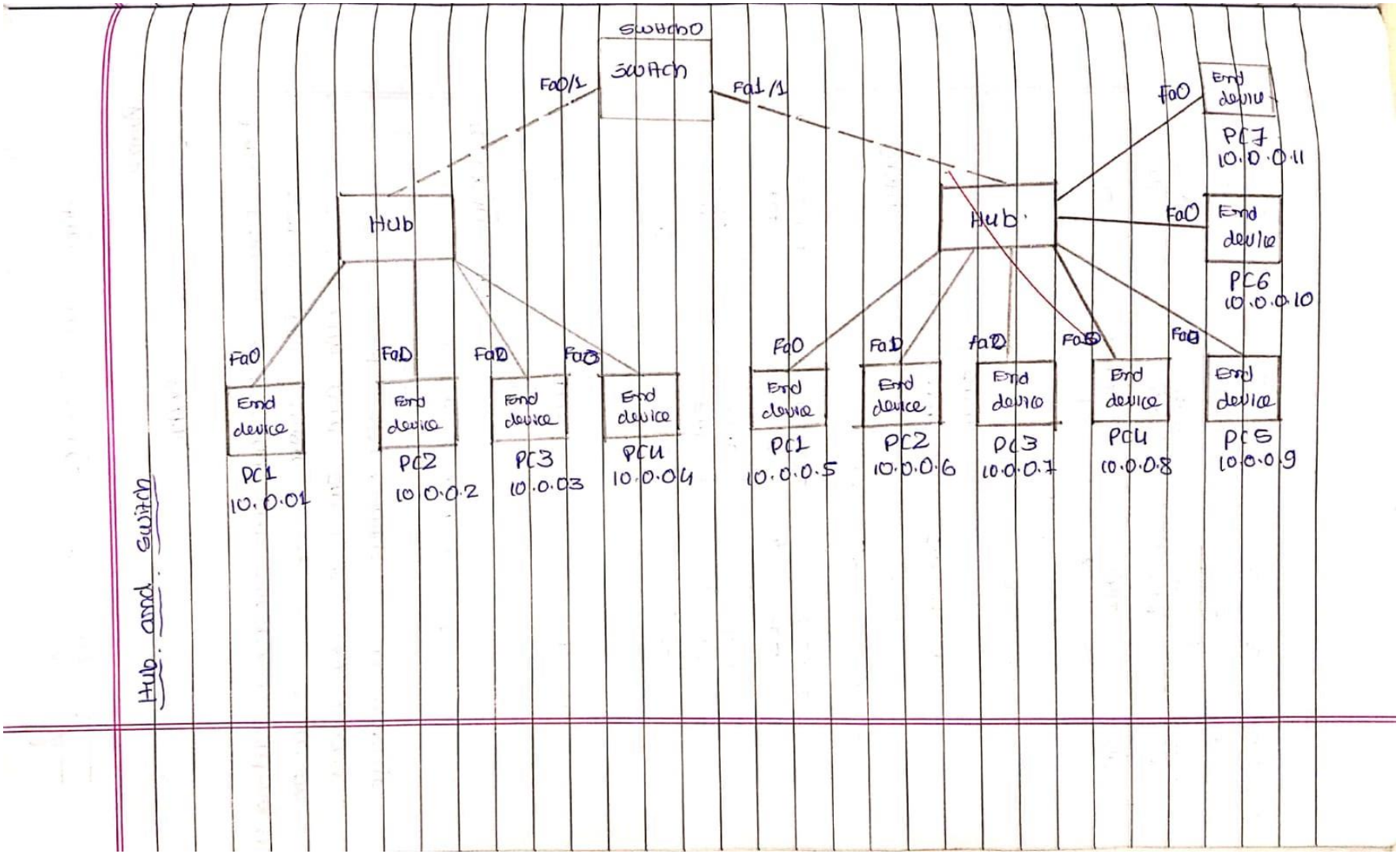
Index

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:




Hub and switch
Switch0
Switch
Fa0/1
Fa1/1
Hub
Hub
Fa0
Fa0
Fa0
Fa0
Fa0
Fa1
Fa2
Fa3
Fa0
Fa0
End device
PC1
10.0.0.1
End device
PC2
10.0.0.2
End device
PC3
10.0.0.3
End device
PC4
10.0.0.4
End device
PC1
10.0.0.5
End device
PC2
10.0.0.6
End device
PC3
10.0.0.7
End device
PC4
10.0.0.8
End device
PC5
10.0.0.9
End device
PC6
10.0.0.10
End device
PC7
10.0.0.11

Date __/__/__
Page _____

Procedure:

1] Hub:

* Add a hub and 3 end devices into logical workspace.
* connect the end devices to the hub using copper straight connectors.
* Assign a unique IP address to each end device
* Send the simple PDU from one device to other.
* Ping a PDU using command prompt in realtime mode

2] Switch:

* Add a switch and 4 end devices (PCs) into logical workspace.
* connect the end devices to the switch using copper straight connectors.
* Assign unique IP address to each of the end device.
* Send the simple PDU from one device to other.
* Ping a PDU using command prompt in realtime mode.

3] Switch and hub:

* Add a switch, 2 hubs and 11 end devices into logical workspace.
* connect 4 end devices to one hub and the remaining to the other.
* connect these two hubs to a switch, using copper cross-over connection.

* Assign unique IP address for each device.
* Send a simple PDU from the end device of one hub to the end device of other hub.
* In realtime mode ping a message from one end device from hub-1 to other end device in hub-2 using command prompt.

Result:

1] HUB:

command > ping 10.0.0.3.

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes = 32 time=1ms 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
Reply from 10.0.0.3: bytes = 32 time=1ms 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.

2] Switch:

command > ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:
Reply from 10.0.0.3: bytes = 32 time = 1ms TTL = 128
Reply from 10.0.0.3: bytes = 32 time = 0ms TTL = 128.

Date __/__/__
Page _____

Reply from 10.0.0.3 : bytes=32 time=0ms 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 time in milli-seconds:
Minimum =0ms, Maximum = 1ms, Average =0ms

3] Hub and switch

command > ping 10.0.0.8.

Pinging 10.0.0.8 with 32 bytes of data:

Reply from 10.0.0.8 : bytes=32 time=1ms TTL=128
Reply from 10.0.0.8 : bytes=32 time=1ms TTL=128
Reply from 10.0.0.8 : bytes=32 time=0ms TTL=128
Reply from 10.0.0.8 : bytes=32 time=0ms TTL=128.

Ping statistics for 10.0.0.8:
Packets : Sent=4, Received=4, Lost=0 (0% Loss)
Approximate round trip times in milli-seconds:
Minimum=0ms, Maximum=1ms, Average =3ms.

Observation:
Hub: Hub is a networking device which is used to transmit the signal to each port to respond from which the signal was received. It transmits signal to every port except port from which the signal is received.

- Switch: Switch is an intelligent device which sends message to selected destination device only. First it examines the destination address and send the message to the corresponding device(s).

Observation about the experiment

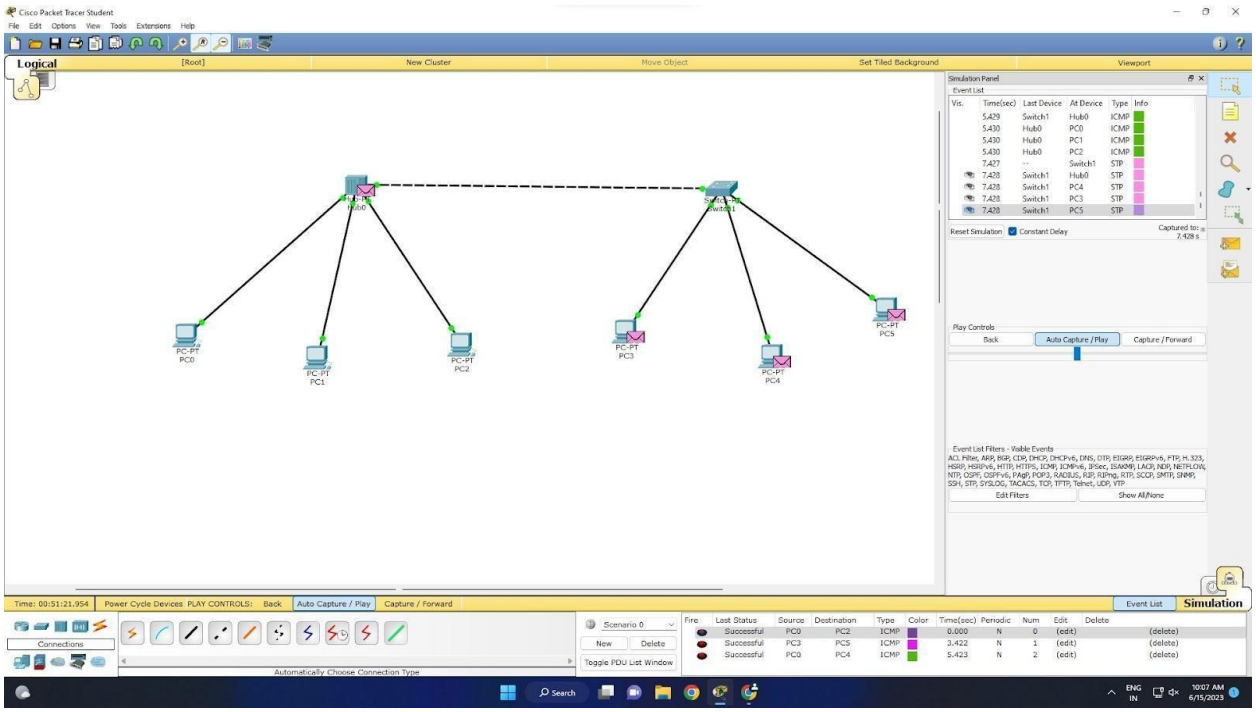
Hybrid - Pt to Pt WAN & switched WAN.

Internet → backbones
ISPs.

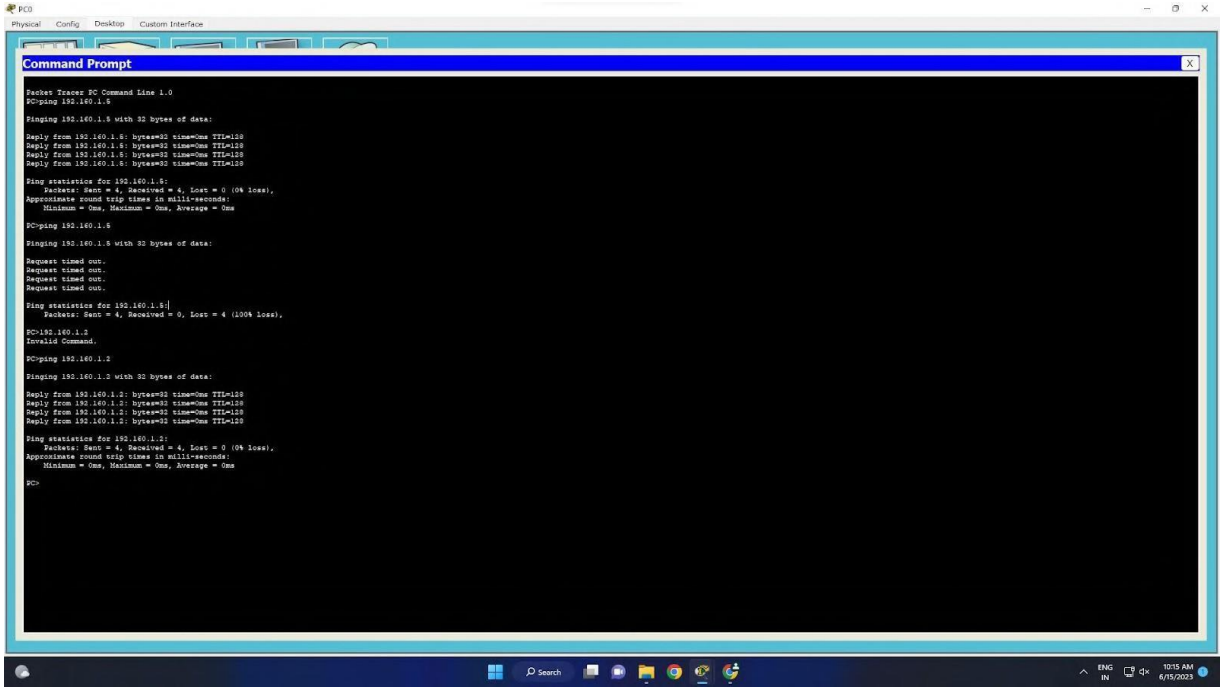
Lab 22/6/23

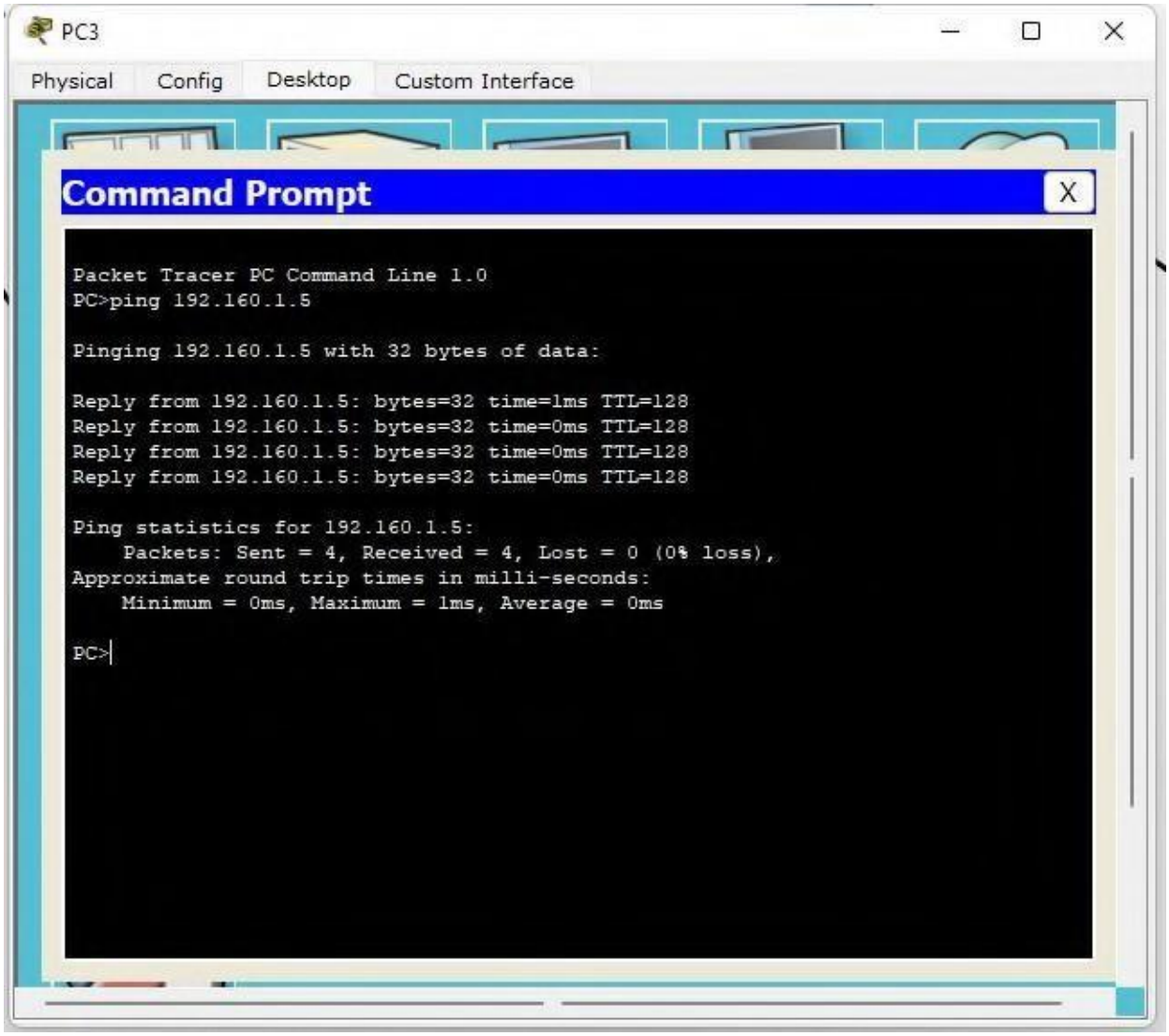
9

TOPOLOGY:



OUTPUT:

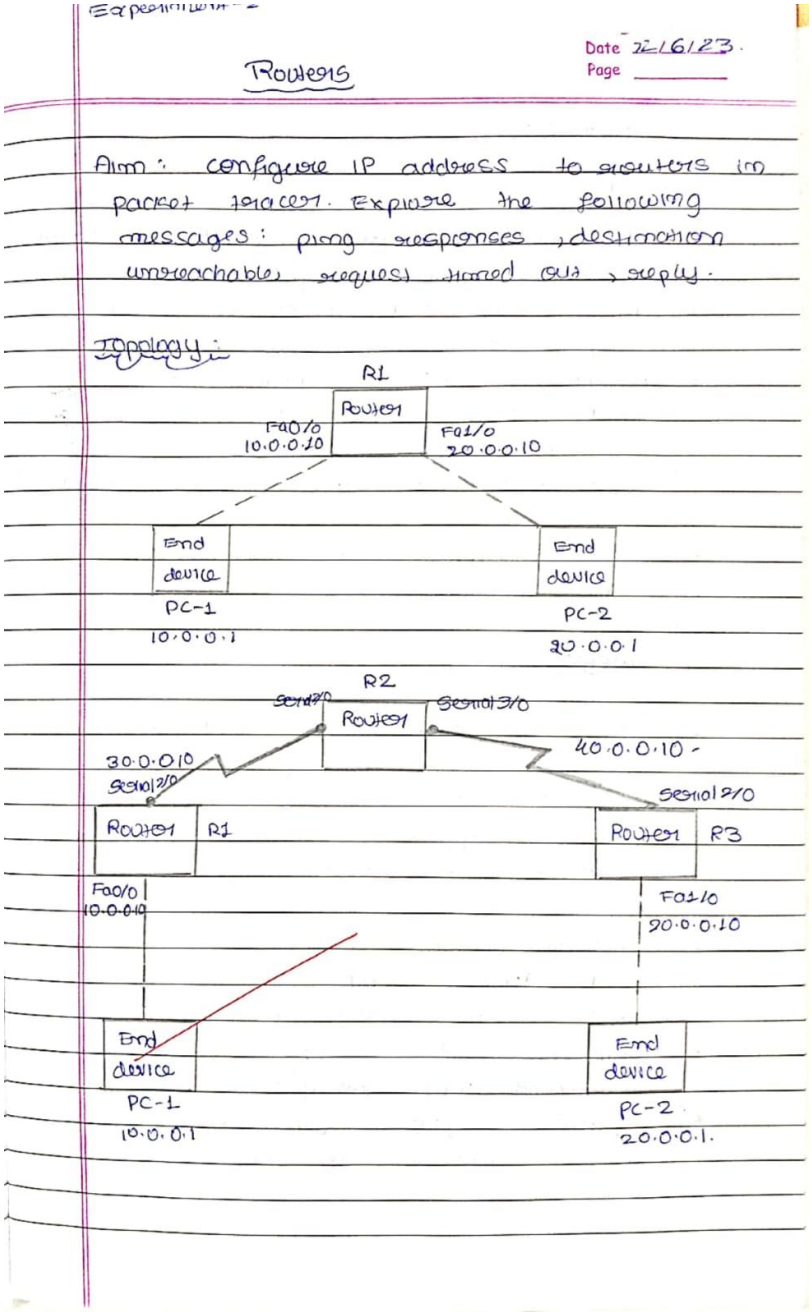



PC3
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 192.160.1.5
Pinging 192.160.1.5 with 32 bytes of data:
Reply from 192.160.1.5: bytes=32 time=1ms TTL=128
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
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 = 1ms, 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:

* Single router:

* Add two end devices and one router to the logical workspace.
* Connect the end devices to router using copper cross-over connector.
* Assign unique IP address to each end device and also assign IP to the router gateways.
* Configure the router using command line interface.
* Ping a PDU from end device of one network to other.

* Multiple routers:

* Add two end devices and 3 routers to the logical workspace.
* Connect the end devices to router using copper cross-over and connect routers to routers using serial connectors.
* Assign unique IP address to each end device and also assign IP to the router gateways.
* Configure all of the routers using command line interface.
* Ping a PDU from end device of one network to other.

Date 22/6/23
Page

Result:

* Single router:

PC> ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data:

Request timed out.
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=5ms TTL=127

Ping statistics for 20.0.0.1:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss).
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 5ms, Average = 1ms.

* Multiple routers:

PC> ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data:
Reply from 40.0.0.10: bytes=32 time=7ms TTL=254
Reply from 40.0.0.10: bytes=32 time=5ms TTL=254
Reply from 40.0.0.10: bytes=32 time=6ms TTL=254
Reply from 40.0.0.10: bytes=32 time=6ms TTL=254

Ping statistics for 40.0.0.10:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)
Approximate round trip time in milli-seconds:
Minimum = 5ms, Maximum = 7ms, Average = 6ms.

Commands used to configure Router:

1] Enable
2] config t
3] interface <gateway port>
4] ip address <ip address> <subnet mask>
5] no shut
6] ip route <network ID> <subnet mask> <gateway ip>
7] no shut.

Observation on the equipment:

* A router is a hardware device which is used to connect a LAN with an internet connection. It is used to receive, analyze and forward the incoming packets to another network.
* A router forwards the packet based on the information available in the routing table.
* It determines the best path from the available paths for transmission of the packet.

* Result:

Before Routes have knowledge about entire network:

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
Reply from 10.0.0.10: Destination host unreachable

Date __/__/__
Page _____

observation :

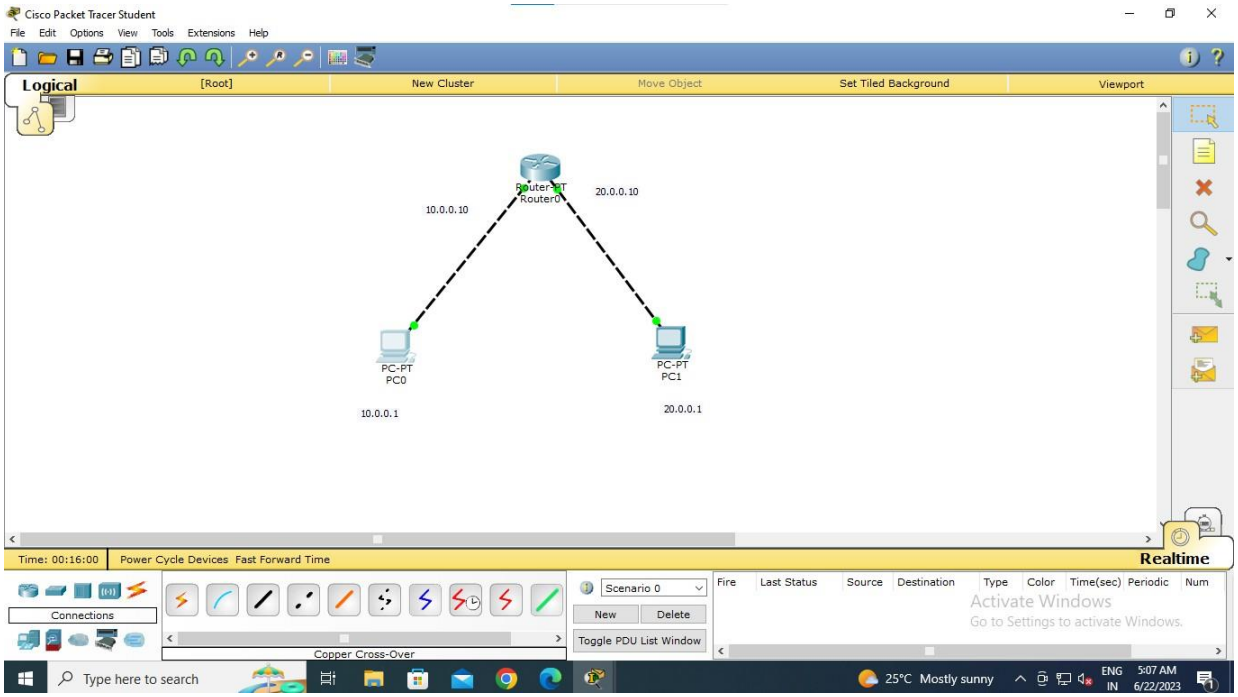
* PC-0 sends the data packet to router. Once the router receives a packet, it looks at its IP address. The destination IP address tells the router the destination of the packet. The router has multiple ports it could send a packet along, and its goal is to send the packet to a router that's closer to its final destination. The packet then arrives at destination i.e. PC-1.

* Destination host unreachable : Implies that host we are trying to ping is down. It is happened because the router has no information about the destination end device or it's network. Router needs to be routed with the information of IP's of destination and it's network host.

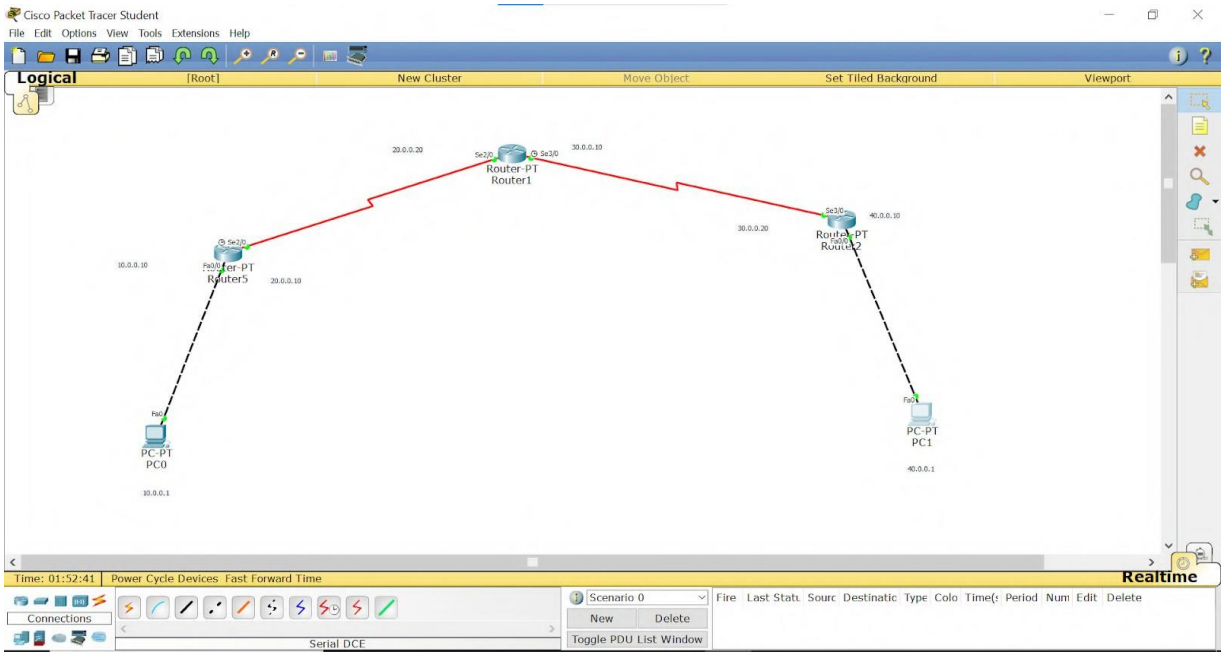
* Request timed out : Implies that ICMP packets reached from one host to other host but the reply could not reach the requesting host. This is due to the improper assignments of the gateways. There may be more packet loss or some physical issue.

TOPOLOGY:

PROGRAM 2.1

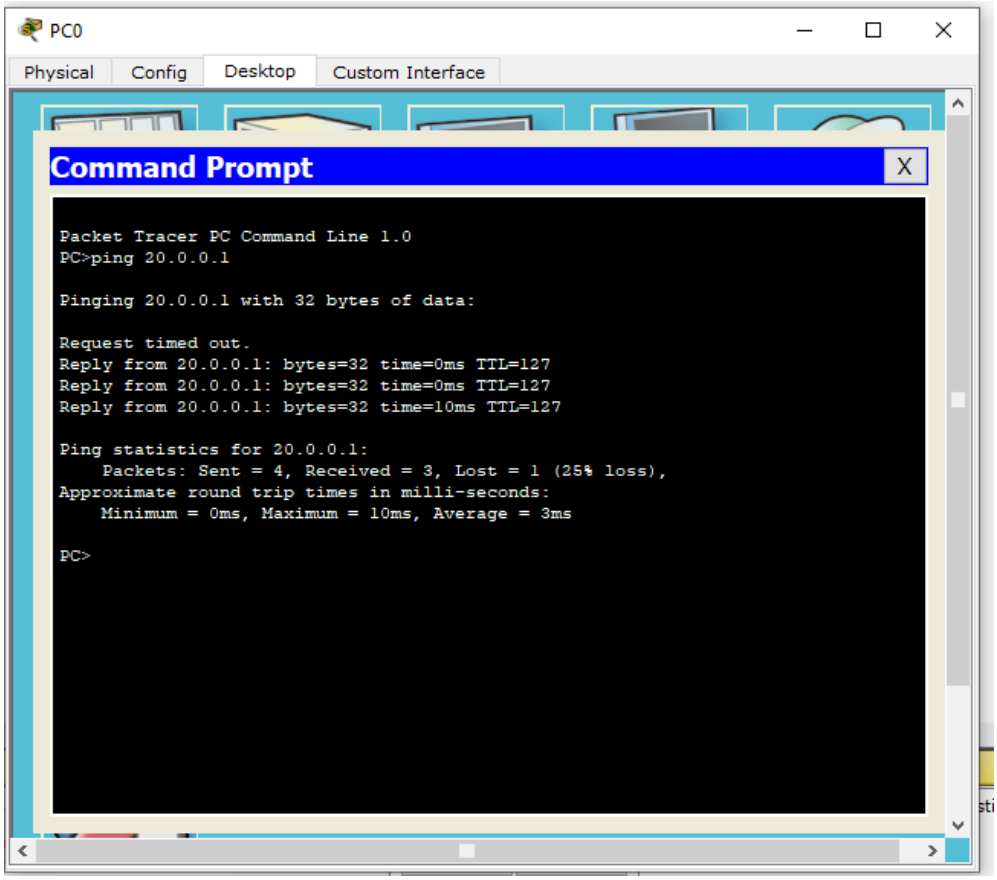


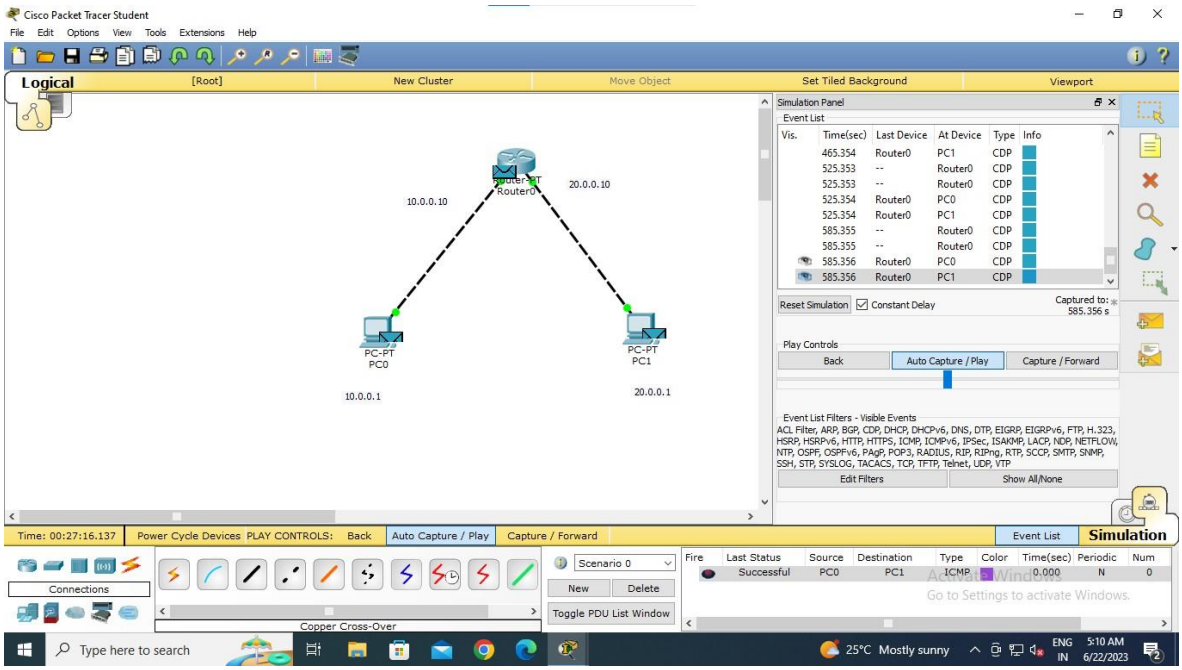
PROGRAM 2.2



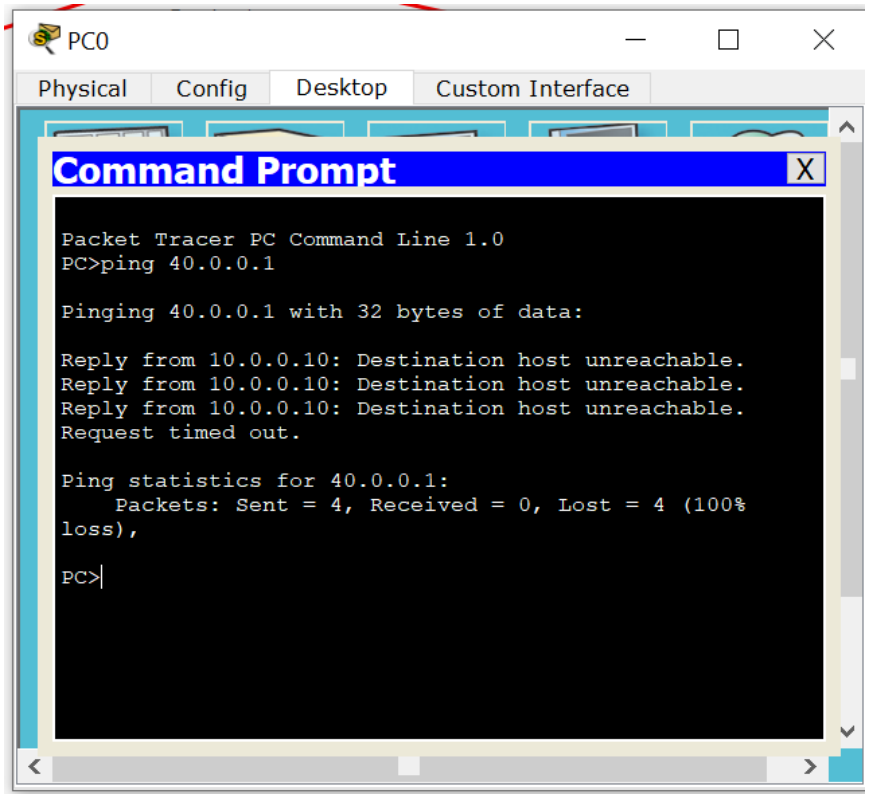
OUTPUT:

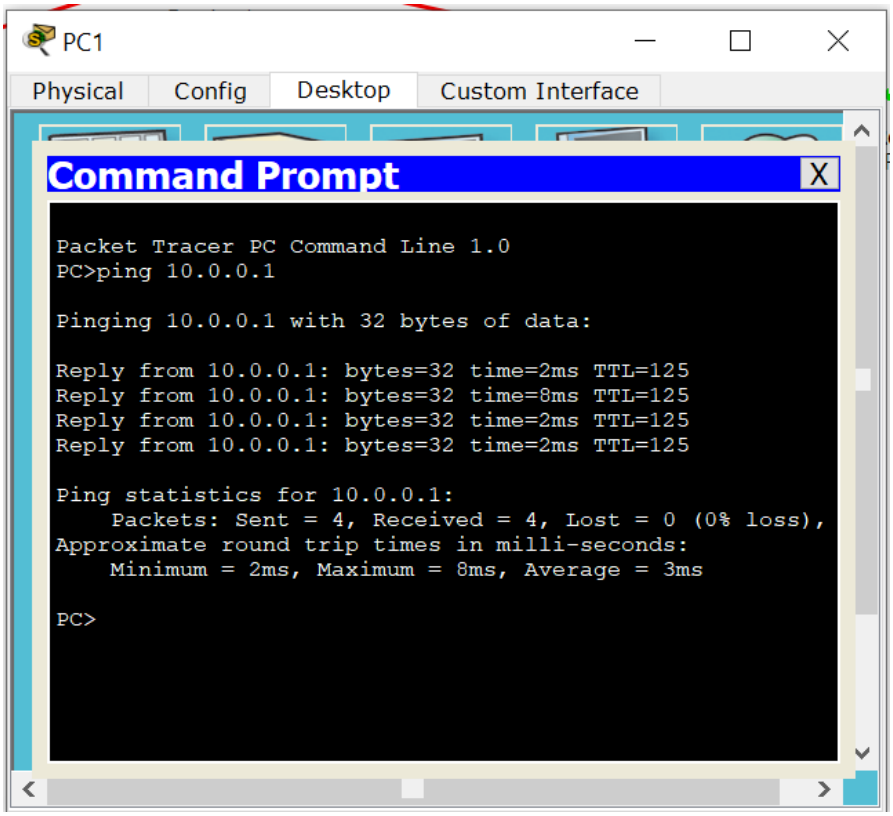
PROGRAM 2.1

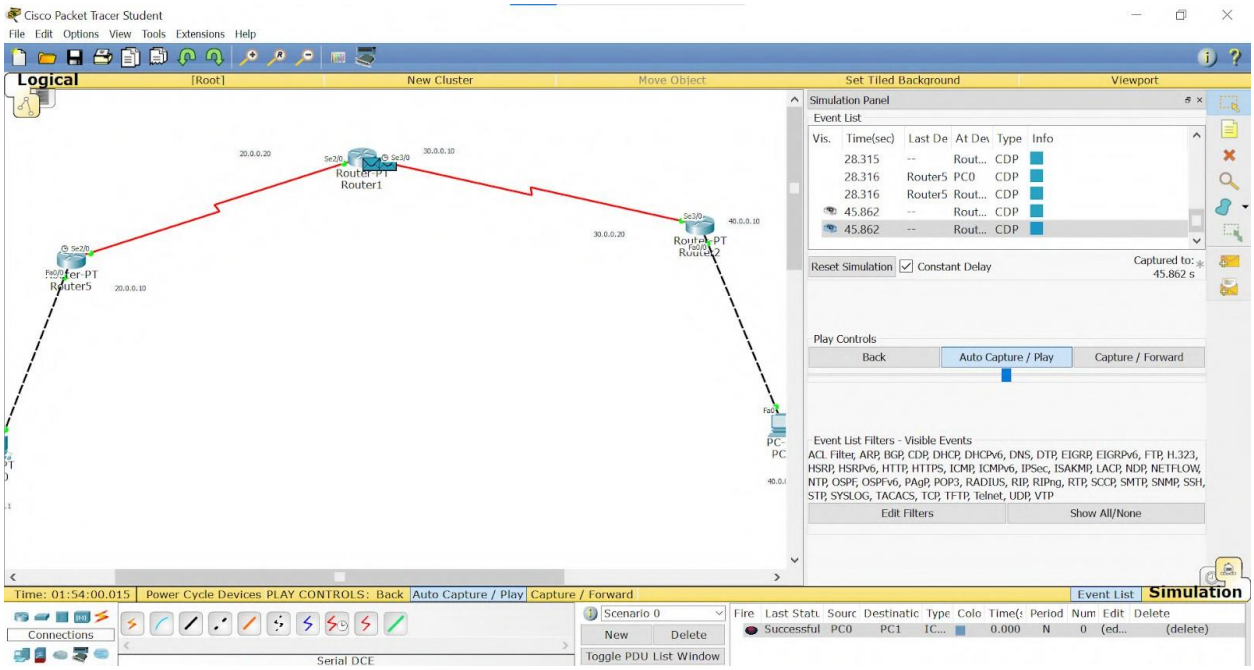




PROGRAM 2.2


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>

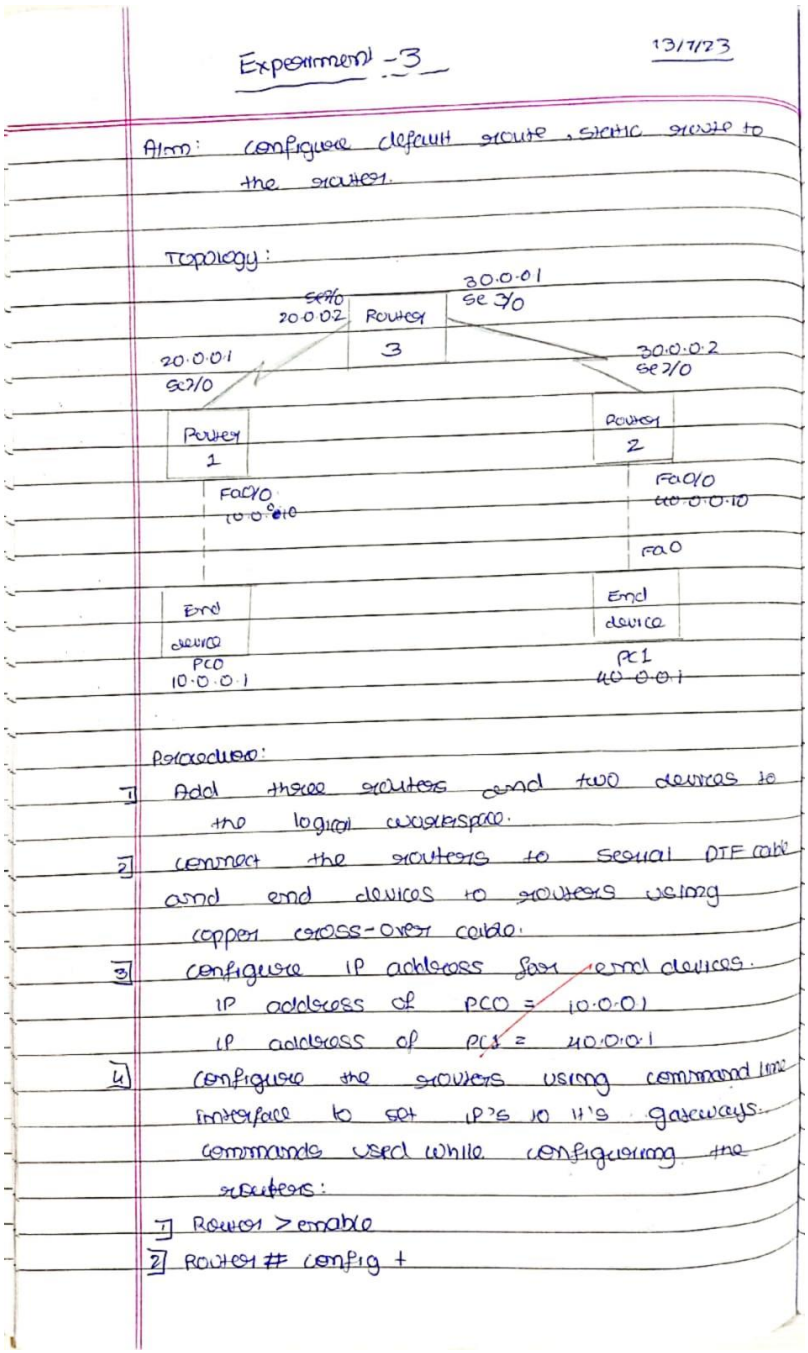

PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=2ms TTL=125
Reply from 10.0.0.1: bytes=32 time=8ms TTL=125
Reply from 10.0.0.1: bytes=32 time=2ms TTL=125
Reply from 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>


Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router1
Router5
Router2
PC-PC
Simulation Panel
Event List
Vis. Time(sec) Last De At Dev Type Info
28.315 -- Rout... CDP
28.316 Router5 PC0 CDP
28.316 Router5 Rout... CDP
45.862 -- Rout... CDP
45.862 -- Rout... CDP
Reset Simulation
Constant Delay
Captured to: 45.862 s
Play Controls
Back
Auto Capture / Play
Capture / Forward
Event List Filters - Visible Events
ACL Filter, ARP, BGP, CDP, DHCP, DHCPv6, DNS, DTP, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, LACP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, RADIUS, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, VTP
Edit Filters
Show All/None
Time: 01:54:00.015
Power Cycle Devices PLAY CONTROLS: Back
Auto Capture / Play
Capture / Forward
Event List
Simulation
Connections
Serial DCE
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Statu Sourc Destinatic Type Colo Time(s Period Num Edit Delete
Successful PC0 PC1 IC... 0.000 N 0 (ed... (delete)

WEEK 3

Configure default route, static route to the Router.

OBSERVATION:



Experiment -3

13/7/23

Aim: Configure default route, static route to the routers.

Topology:

Procedure:

1] Add three routers and two devices to the logical workspace.

2] Connect the routers to serial DTE cable and end devices to routers using copper cross-over cable.

3] Configure IP address for end devices.
IP address of PC0 = 10.0.0.1
IP address of PC1 = 40.0.0.1

4] Configure the routers using command line interface to set IP's to it's gateways.
Commands used while configuring the routers:

1] Router > enable

2] Router # config t

Date 13/7/23
Page

3] Router(config) # ~~ip address~~ interface <port>
4] Router(config-if) # ip address <ip address> <subnet mask>
5] Router(config-if) # ~~no shut~~ no shut
6] Router(config)# exit.

* set the gateways for end devices.
gateway for PC0 : 10.0.0.10
gateway for PC1 : 40.0.0.10.

7] Default routing is possible for networks 10.0.0.0 and 40.0.0.0. Following commands are used to set default routers for the networks 10.0.0.0 and 40.0.0.0
ip route 0.0.0.0 0.0.0.0 20.0.0.2.
ip route 0.0.0.0 0.0.0.0 30.0.0.2.

8] For router 2, we need to mention the ip route
ip route 40.0.0.0 255.0.0.0 30.0.0.1
ip route 10.0.0.0 255.0.0.0 20.0.0.1

9] ping from one end device to another.

Result :

Pinging 40.0.0.1 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.1 : bytes=32 time=15ms TTL=125
Reply from 40.0.0.1 : bytes=32 time=11ms 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 = 15ms, Average = 9ms

IP route of Router 1
Gateway of last resort is 20.0.0.1 to
networks 0.0.0.0

C 10.0.0.0/8 is directly connected, FastEthernet 0/0
C 20.0.0.0/8 is directly connected, Serial 2/0
S* 0.0.0.0 [1/0] via 20.0.0.1
[1/0] via 20.0.0.2.

Observation:

* A default route is the route that takes effect when no other route is available for an IP destination address.
* If the destination address is not local, the device checks its routing table. If the remote destination subnet is not present in routing table, the packet is forwarded to the next hop towards destination using the default route.
* The default route generally has a next hop address of another routing device, which performs the same process.
* The process repeats until a packet is delivered to the destination.
* whereas static routing refers to specifying

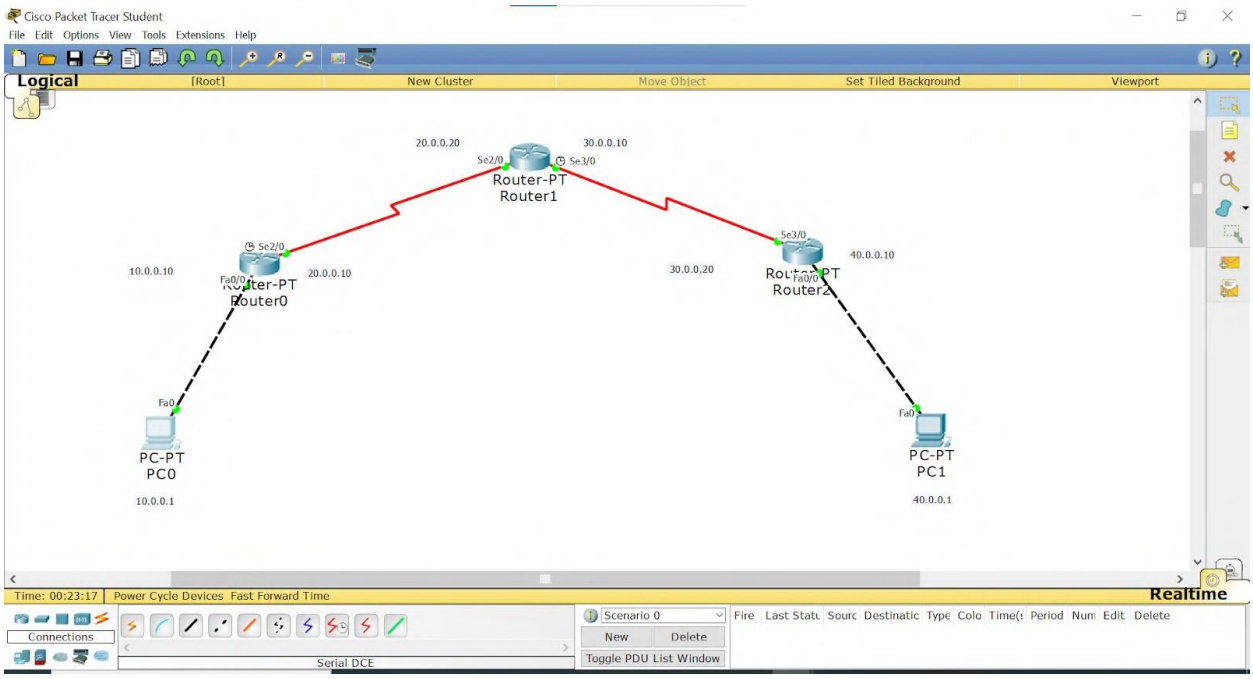
Date __/__/__
Page _____

a route for a given destination IP address.

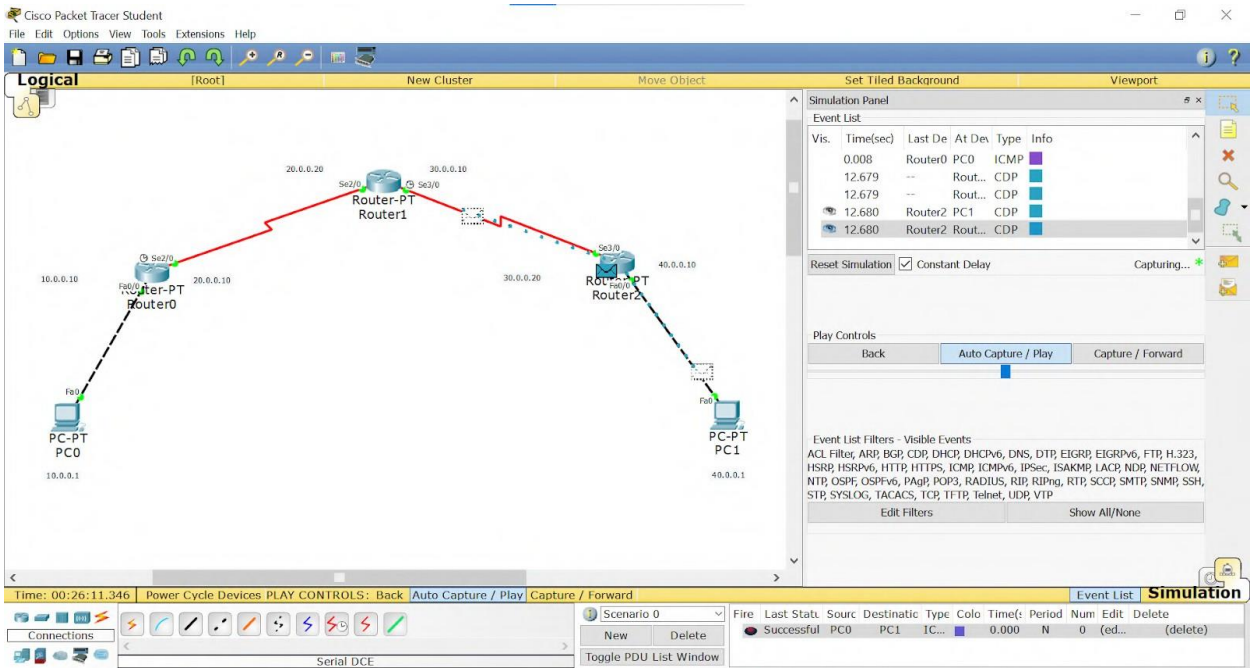
* the destination IP address of a packet is checked. For a given destination IP address the next hop is stored in routing table. And packets are sent to that destination IP.

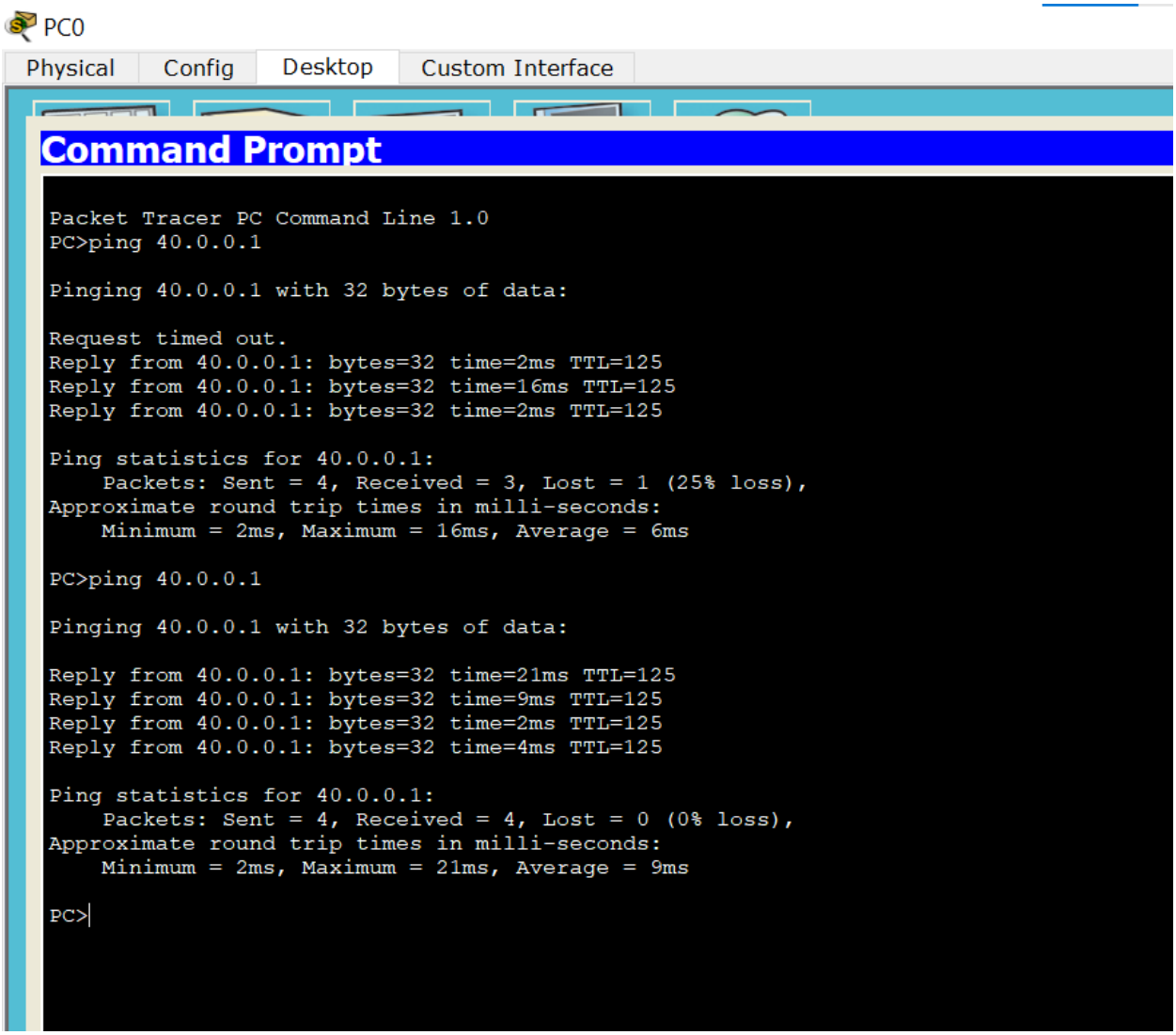
20/7/23

TOPOLOGY:



OUTPUT:

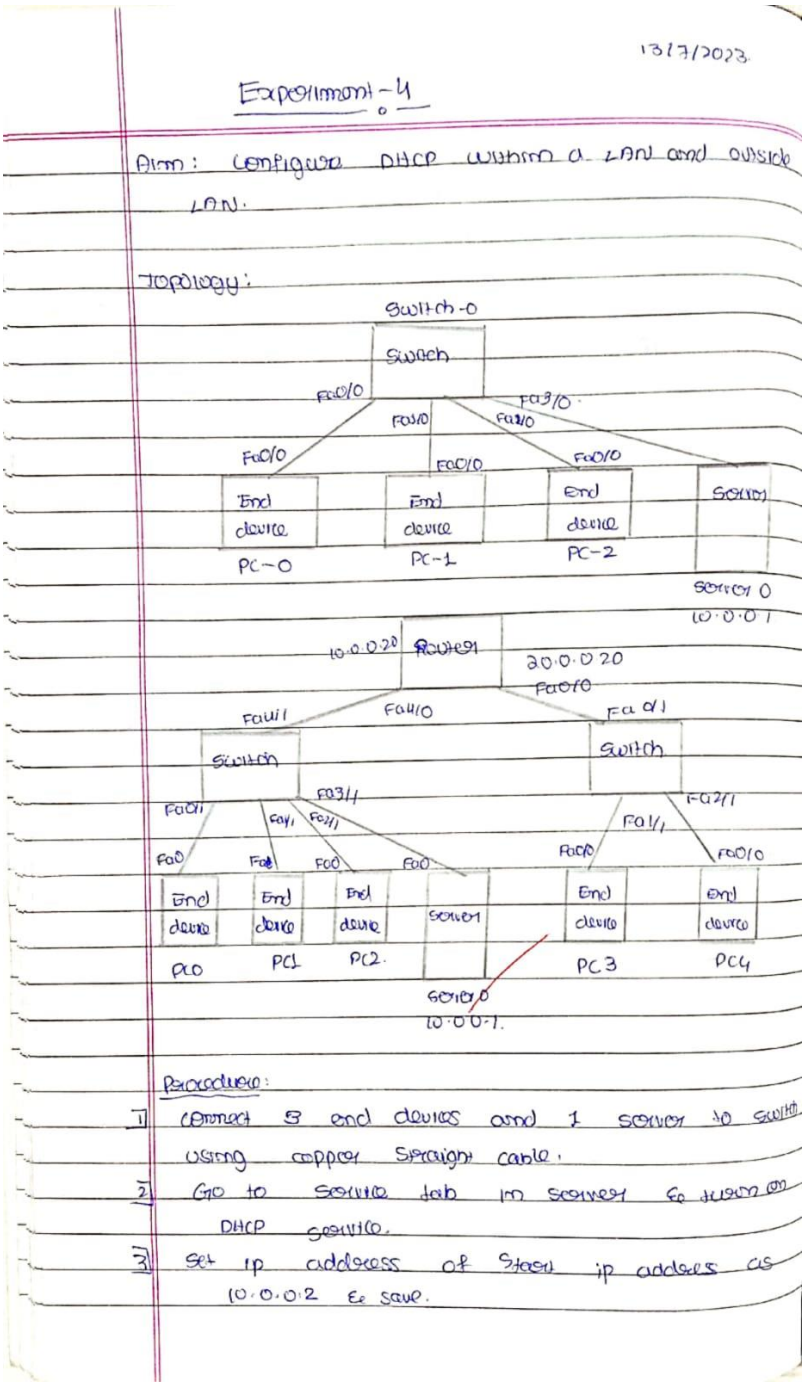



PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Request timed out.
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125
Reply from 40.0.0.1: bytes=32 time=16ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms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



Date 13/7/2023.
Page

4] Before which set ip address of server to 10.0.0.1 under fastethernet in config t. mode.

5] Click on PC0 and go to desktop tab, click ip configuration select DHCP, it will request for ip address and successfully get DHCP request. Also set the ip address.

6] Repeat same process to other two PC's.

7] Go to PC's command prompt and ping the message.

8] Add a router, a switch and 2 PC's to the logical workspace and connect the router to both switches. Connect the PC's to the newly added switch, using copper straight cable.

9] Set the server ip address of server using the following commands:

i) enable
ii) config t
iii) interface fastethernet 4/0
iv) ip address 10.0.0.20 255.0.0.0
v) no shut
vi) exit
vii) interface fastethernet 0/0.
viii) ip address 20.0.0.20 255.0.0.0.
ix) no shut
x) exit.

10] Now, goto the server and set gateway as 10.0.0.1

11] Go to router CLI and follow the below commands to enable DHCP in the new network.

i) config t
ii) interface fastethernet - 0/0

iii) ip helper-address 10.0.0.1
iv) no shut.

12] Change server pool and set Start IP address to 200.0.1.

13] Repeat step 2 for other two PC's.

Output:

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 from 10.0.0.3:

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

Approximate round trip times in milli-seconds

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

Output : -2

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 = 128 ms
Reply from 20.0.0.2 : bytes=32 time=0ms TTL = 128 ms
Reply from 20.0.0.2 : bytes=32 time=0ms TTL = 128 ms

ping statistics for 20.0.0.2 :

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

Date __/__/__
Page _____

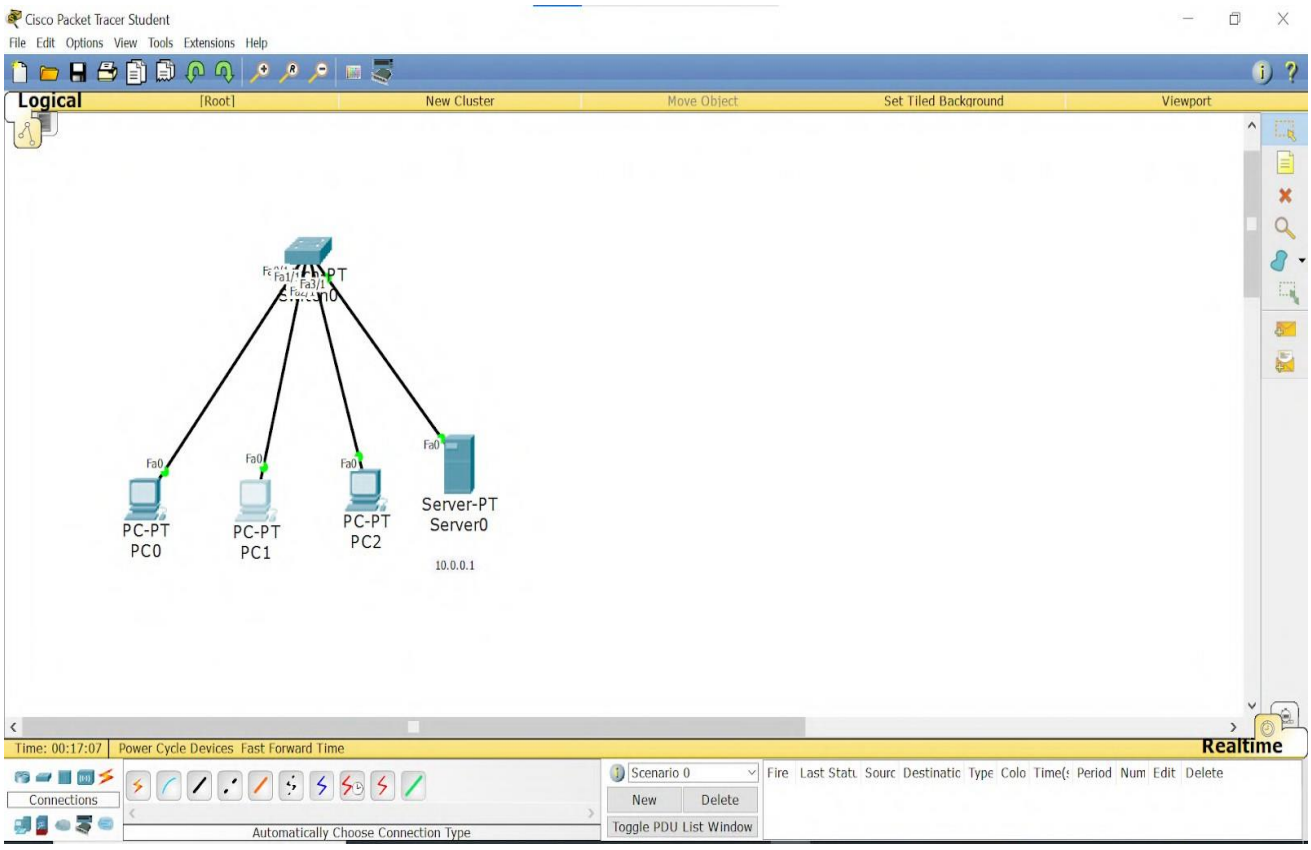
Approximate round trip time in milli-second
Minimum = 0ms, Maximum = 0ms, Average = 0ms.

Observation:

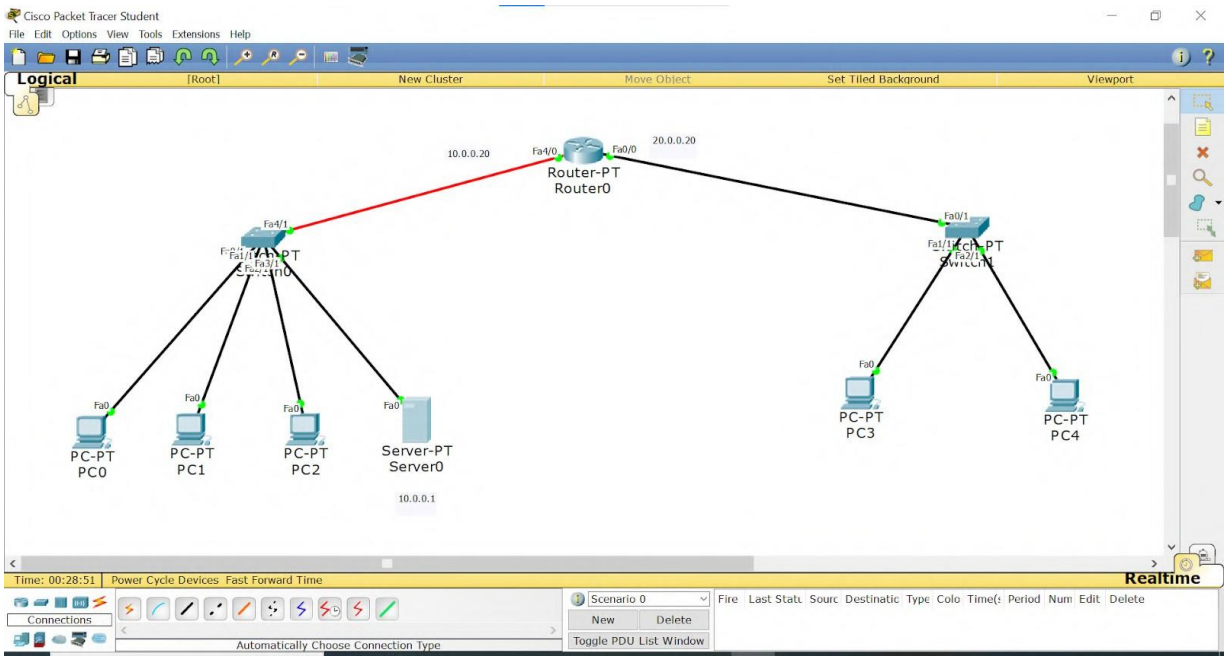
* The Dynamic Host Configuration Protocol (DHCP) is a networking management protocol used on IP networks for automatically assigning IP addresses and other communication to device connected to the network.
* DHCP automatically assign an IP address to any device or node.
* It is a client server protocol where the server manages a pool of unique IP address
* It responds to all client request by providing IP configuration information from address pools, previously specified by a network administrator.

TOPOLOGY:

PROGRAM 4.1:

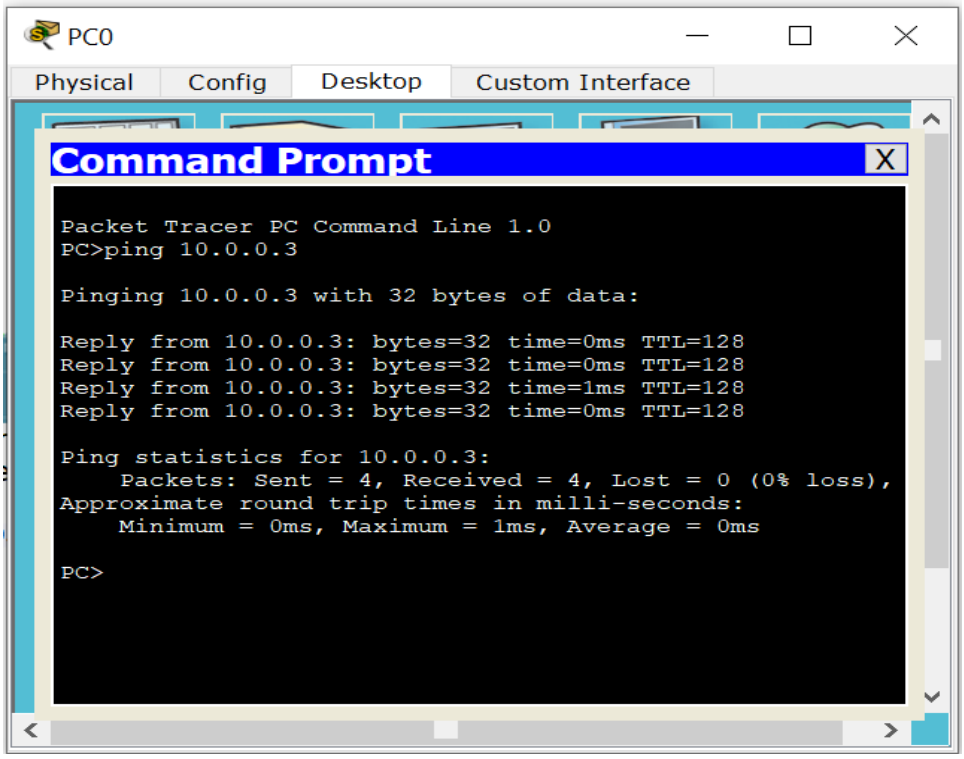


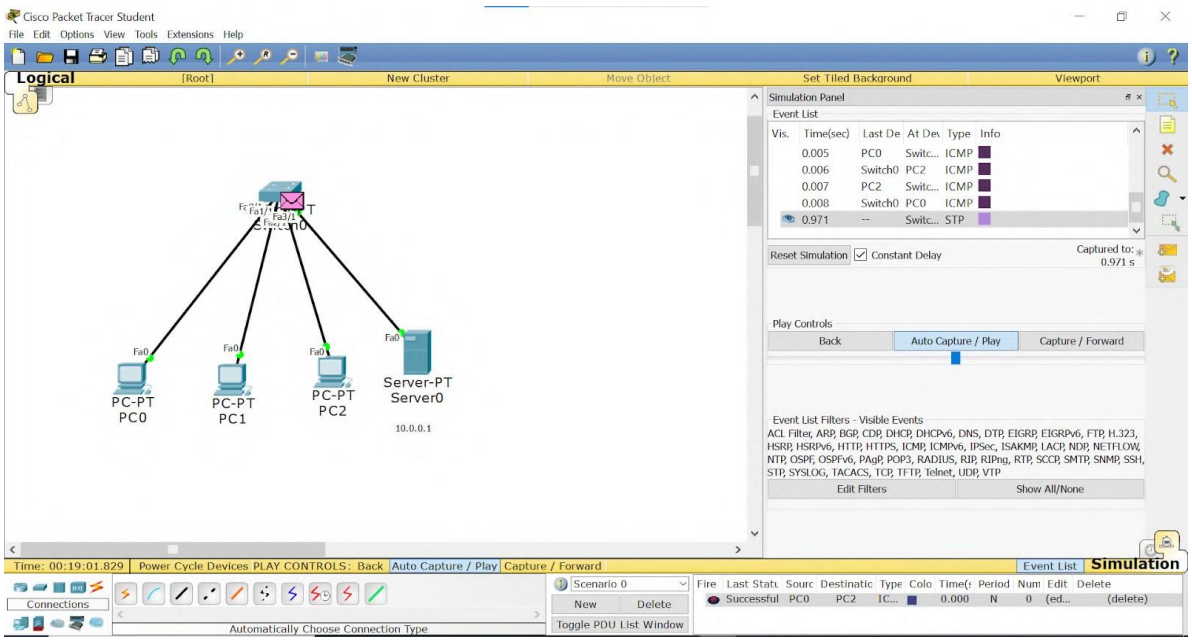
PROGRAM 4.2:


Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
10.0.0.20
20.0.0.20
Fa4/0
Fa0/0
Router-PT
Router0
Fa4/1
Fa0/1
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
Server-PT
Server0
10.0.0.1
PC-PT
PC3
PC-PT
PC4
Time: 00:28:51
Power Cycle Devices Fast Forward Time
Realtime
Connections
Automatically Choose Connection Type
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Statu
Sourc
Destinatic
Type
Colo
Time(s
Period
Num
Edit
Delete

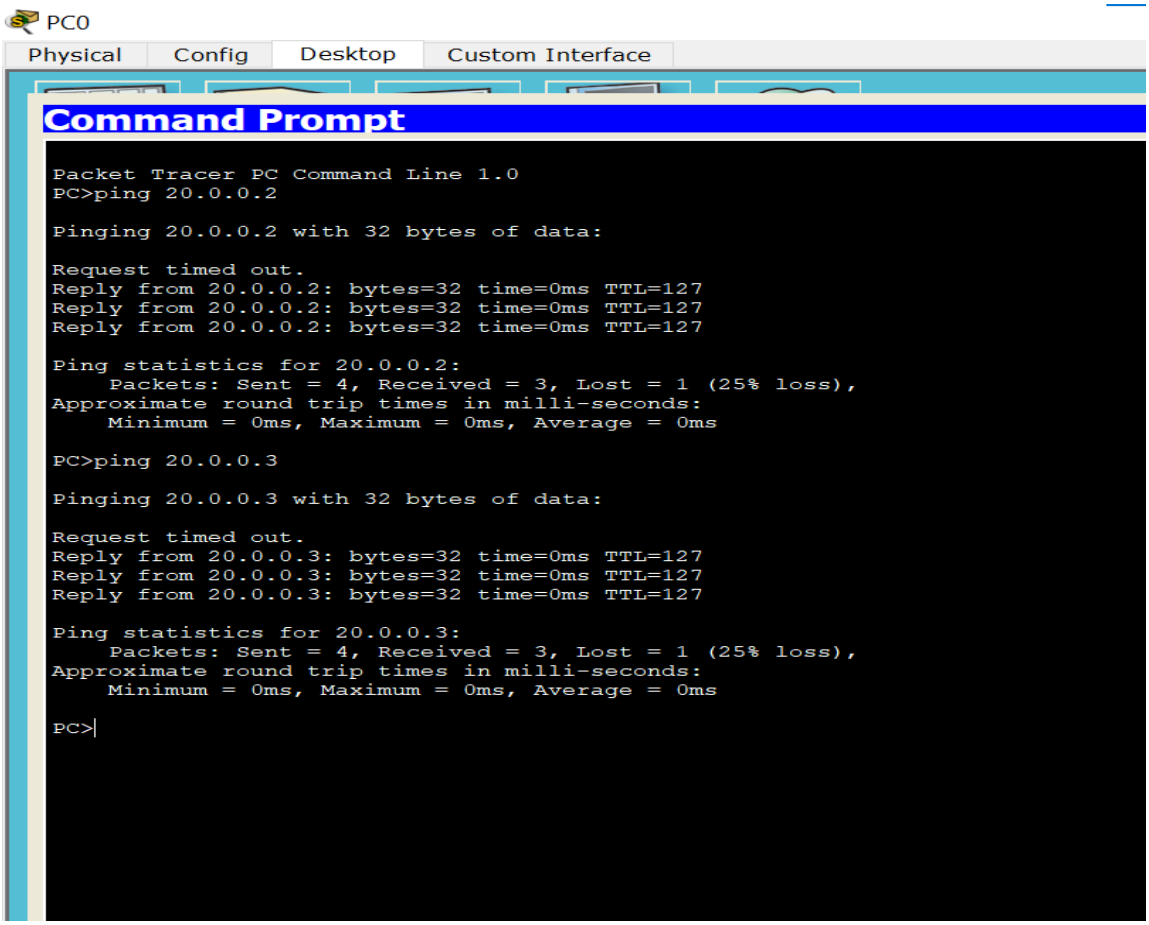
OUTPUT:

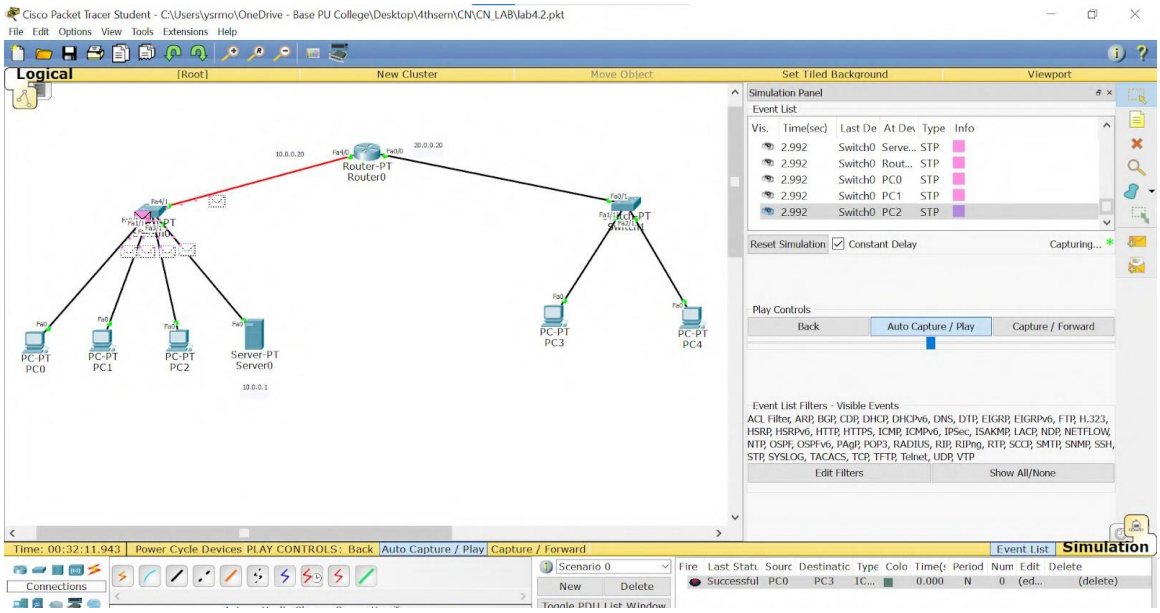
PROGRAM 4.1:





PROGRAM 4.2:

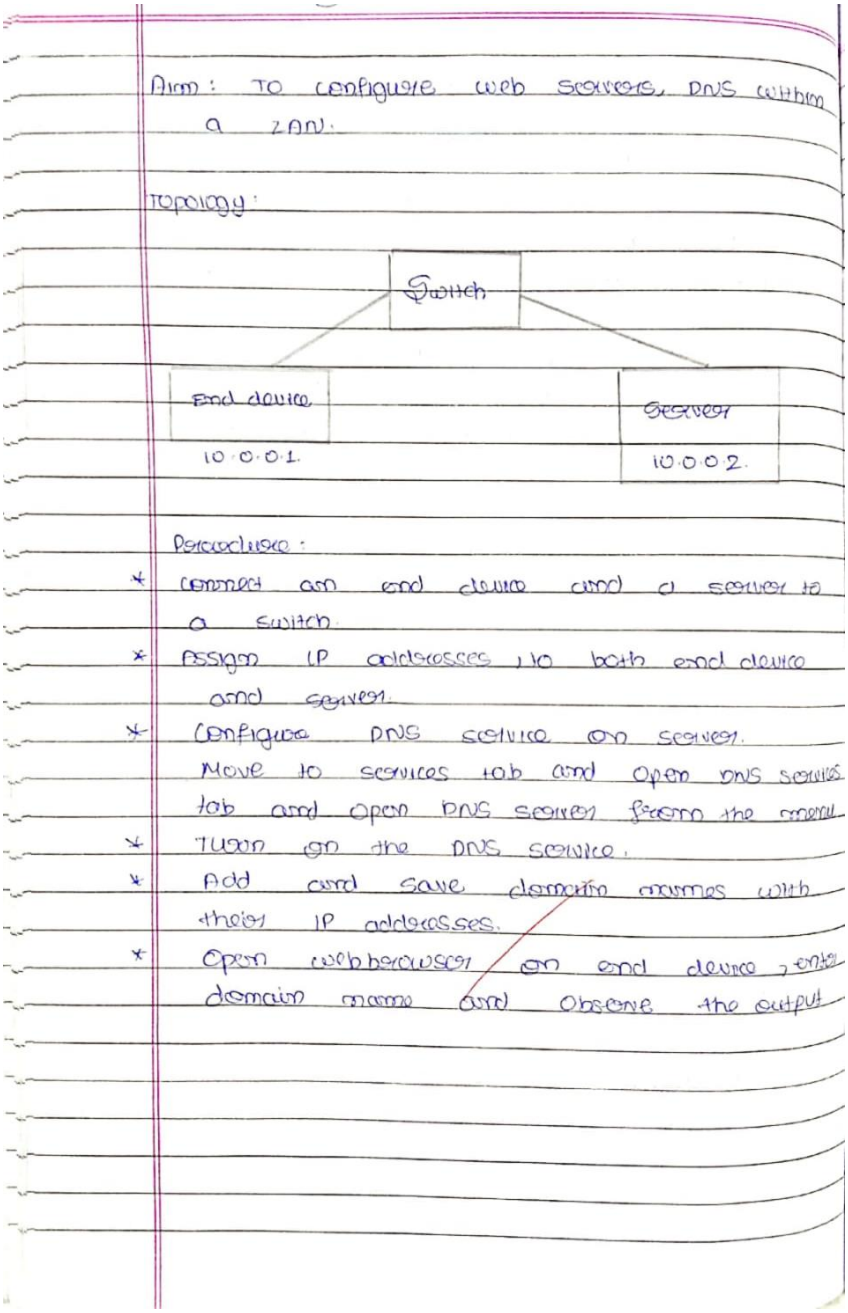

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data:
Request timed out.
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Reply from 20.0.0.2: bytes=32 time=0ms 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>


Cisco Packet Tracer Student - C:\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\CN\CN_LAB\lab4.2.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT
Router0
10.0.0.20
20.0.0.20
Server-PT
Server0
10.0.0.1
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
Server-PT
Server0
PC-PT
PC3
PC-PT
PC4
Simulation Panel
Event List
Vis. Time(sec) Last De At Dev Type Info
2.992 Switch0 Serve... STP
2.992 Switch0 Rout... STP
2.992 Switch0 PC0 STP
2.992 Switch0 PC1 STP
2.992 Switch0 PC2 STP
Reset Simulation
Constant Delay
Capturing...
Play Controls
Back
Auto Capture / Play
Capture / Forward
Event List Filters - Visible Events
ACL Filter, ARP, BGP, CDP, DHCP, DHCPv6, DNS, DTP, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, LACP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, RADIUS, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, VTP
Edit Filters
Show All/None
Time: 00:32:11.943
Power Cycle Devices PLAY CONTROLS: Back
Auto Capture / Play
Capture / Forward
Event List
Simulation
Connections
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Statu Sourc Destinatic Type Colo Time(s Period Num Edit Delete
Successful PC0 PC3 IC... 0.000 N 0 (ed... (delete)

WEEK 5

Configure Web Server, DNS within a LAN.

OBSERVATION:



Date 20/7/23
Page

commands:

Router(config)# router rip
Router(config-router)# network <network address>
Router(config-router)# network <network address>
Router(config-router)# no shut.

For router 1:

Router(config)# router rip
Router(config-router)# network 10.0.0.0
Router(config-router)# network 20.0.0.0

* Ping end devices to test connection.

Result:

>ping 10.0.0.1

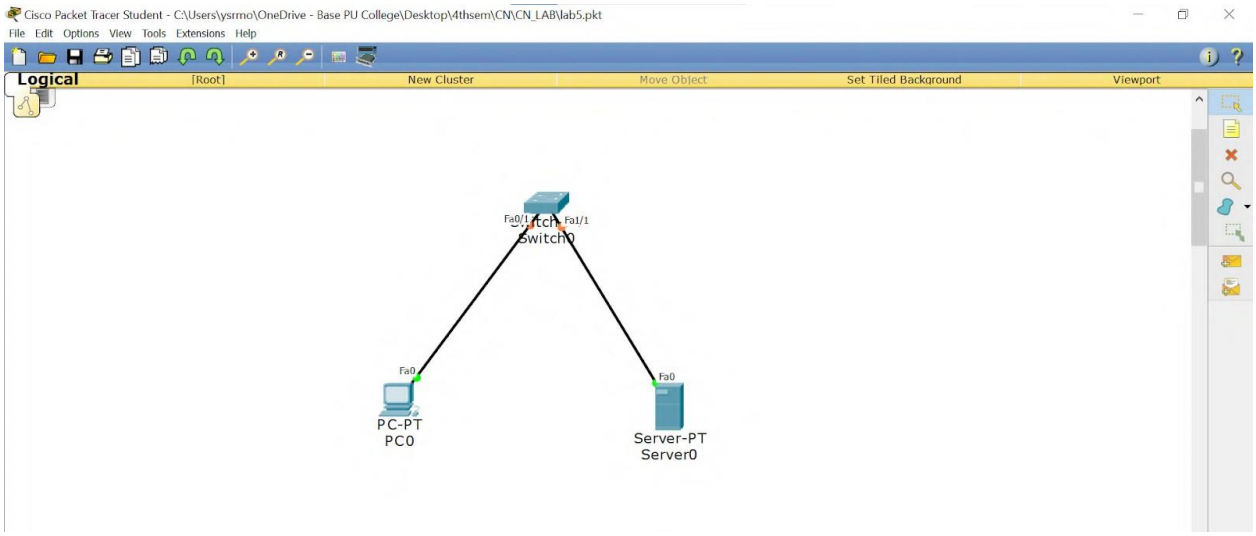
Pinging 10.0.0.1 with 32 bytes of data
Request timed out.
Reply from 10.0.0.1: bytes=32 time=6ms TTL=125
Reply from 10.0.0.1: bytes=32 time=20ms TTL=125
Reply from 10.0.0.1: bytes=32 time=2ms TTL=125
Ping statistics for 10.0.0.1:
Packets: Sent=4 Received=3 Lost=1 (25% Loss)

Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 20ms, Average = 10ms

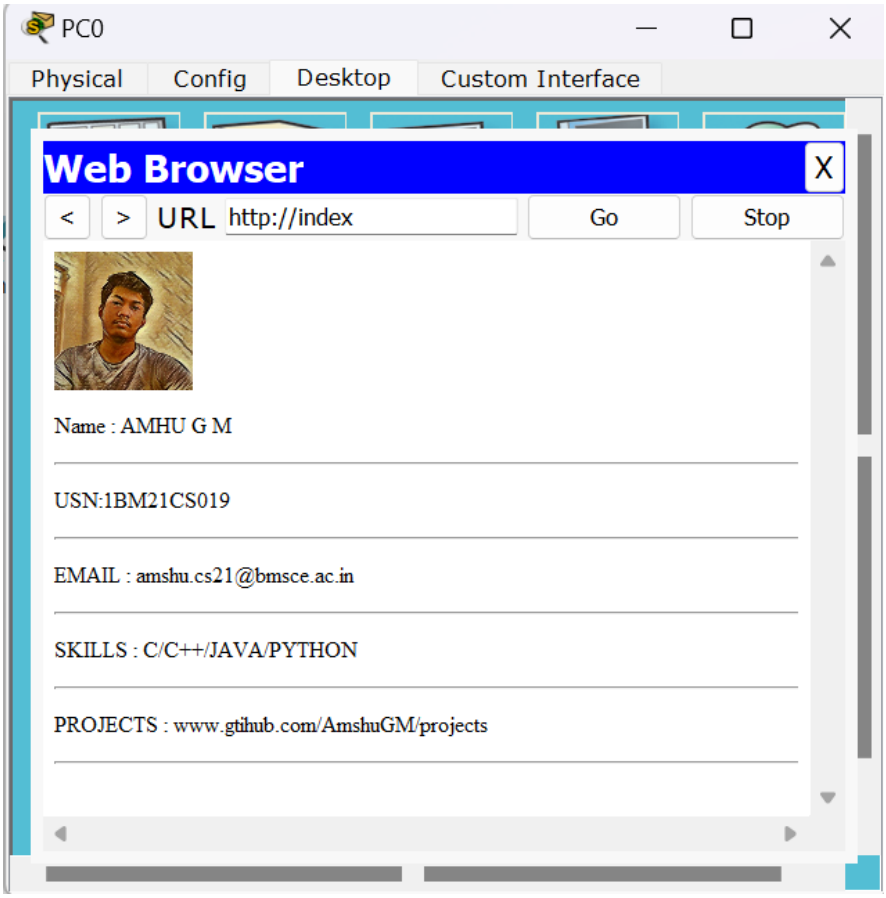
Observation:

* Routing Information Protocol (RIP) is a dynamic routing protocol that uses hop count as routing metric to find best path between source and destination network. Hop count is the number of routers occurring in between source and destination.

TOPOLOGY:



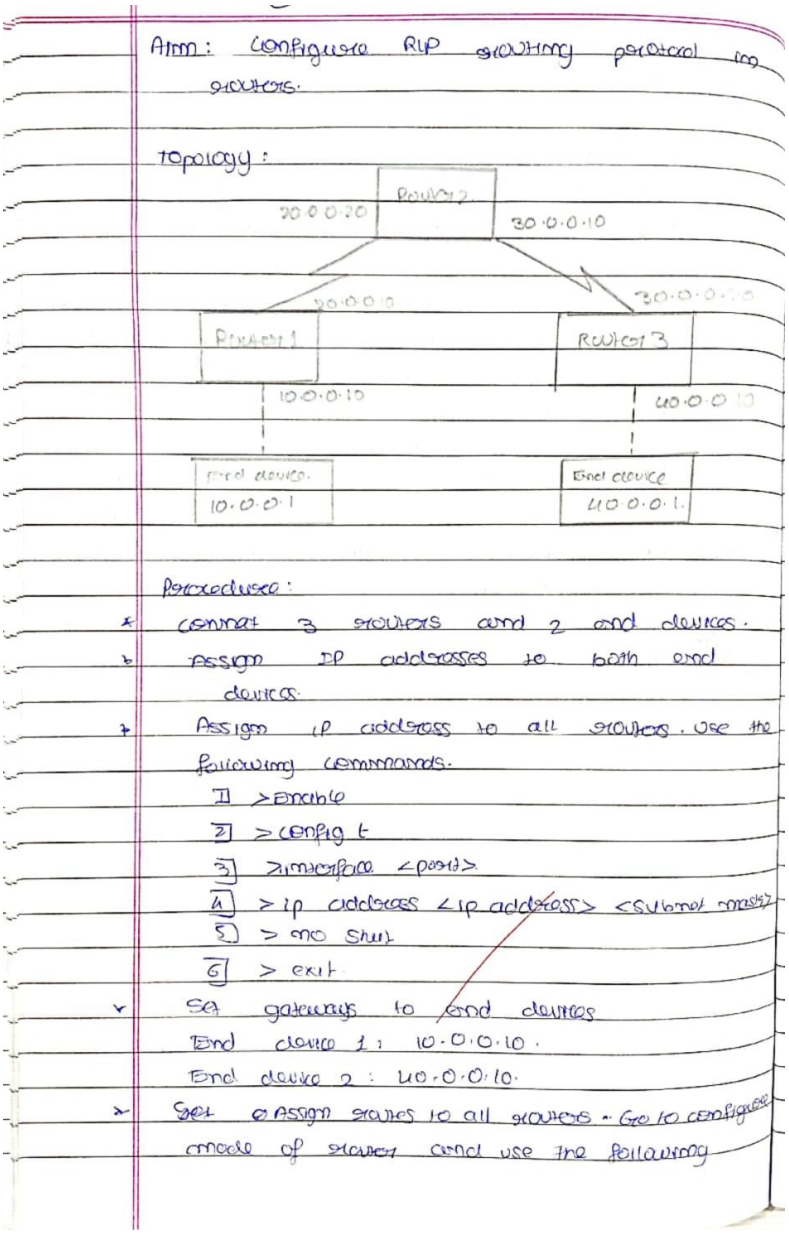
OUTPUT:



WEEK 6

Configure RIP routing Protocol in Routers.

OBSERVATION:



Date 20/7/23
Page

commands:

Router(config)# router rip
Router(config-router)# network <network address>
Router(config-router)# network <network address>
Router(config-router)# no shut.

For router 1:

Router(config)# router rip
Router(config-router)# network 10.0.0.0
Router(config-router)# network 20.0.0.0
* Ping end devices to test connection.

Result:

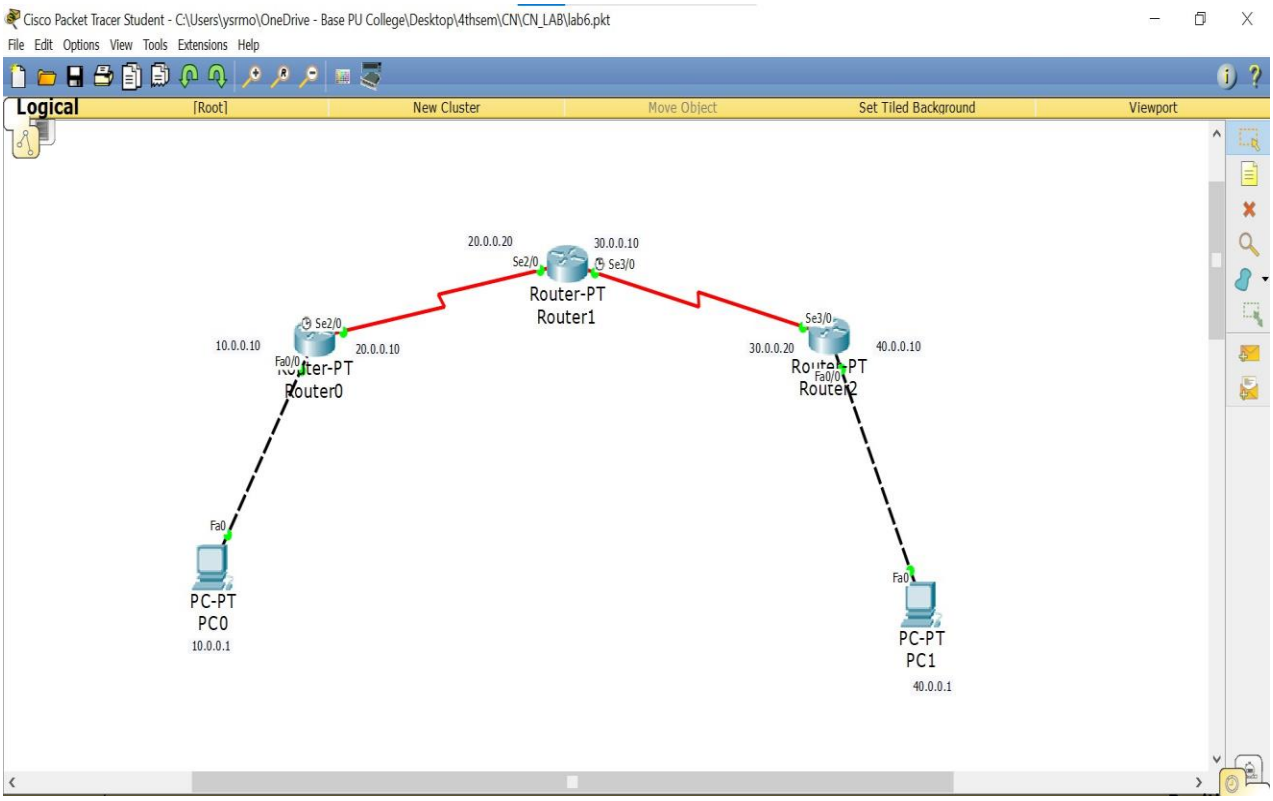
>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data
Request timed out.
Reply from 10.0.0.1: bytes=32 time=6ms TTL=125
Reply from 10.0.0.1: bytes=32 time=20ms TTL=125
Reply from 10.0.0.1: bytes=32 time=2ms TTL=125
Ping statistics for 10.0.0.1:
Packets: Sent=4 Received=3 Lost=1 (25% loss)

Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 20ms, Average = 10ms

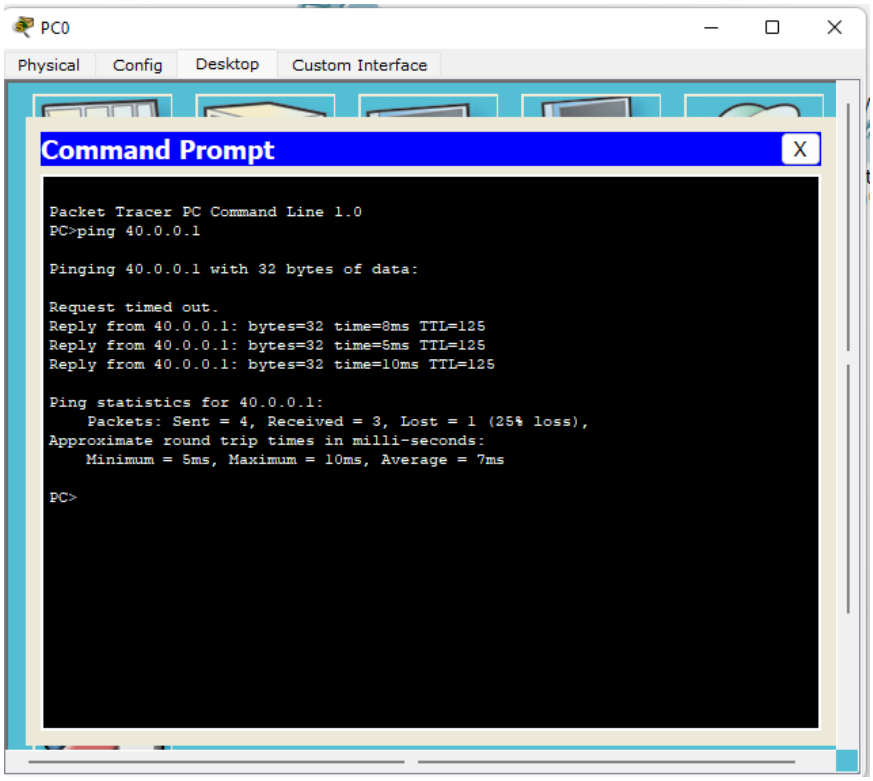
Observation:

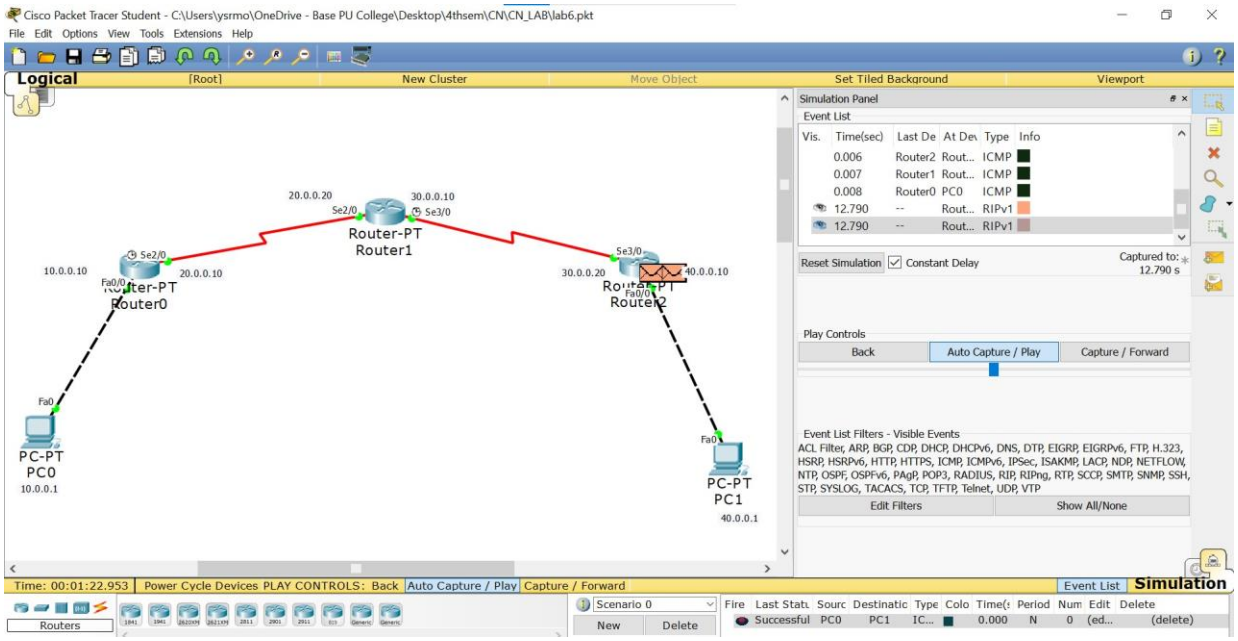
* Routing Information Protocol (RIP) is a dynamic routing protocol that uses hop count as routing metric to find best path between source and destination networks. Hop count is the number of routers occurring in between source and destination.

TOPOLOGY:


Cisco Packet Tracer Student - C:\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\CN\CN_LAB\lab6.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
20.0.0.20
30.0.0.10
Se2/0
Se3/0
Router-PT
Router1
Se2/0
10.0.0.10
20.0.0.10
Fa0/0
Router-PT
Router0
Se3/0
30.0.0.20
40.0.0.10
Fa0/0
Router-PT
Router2
Fa0
PC-PT
PC0
10.0.0.1
Fa0
PC-PT
PC1
40.0.0.1

OUTPUT:

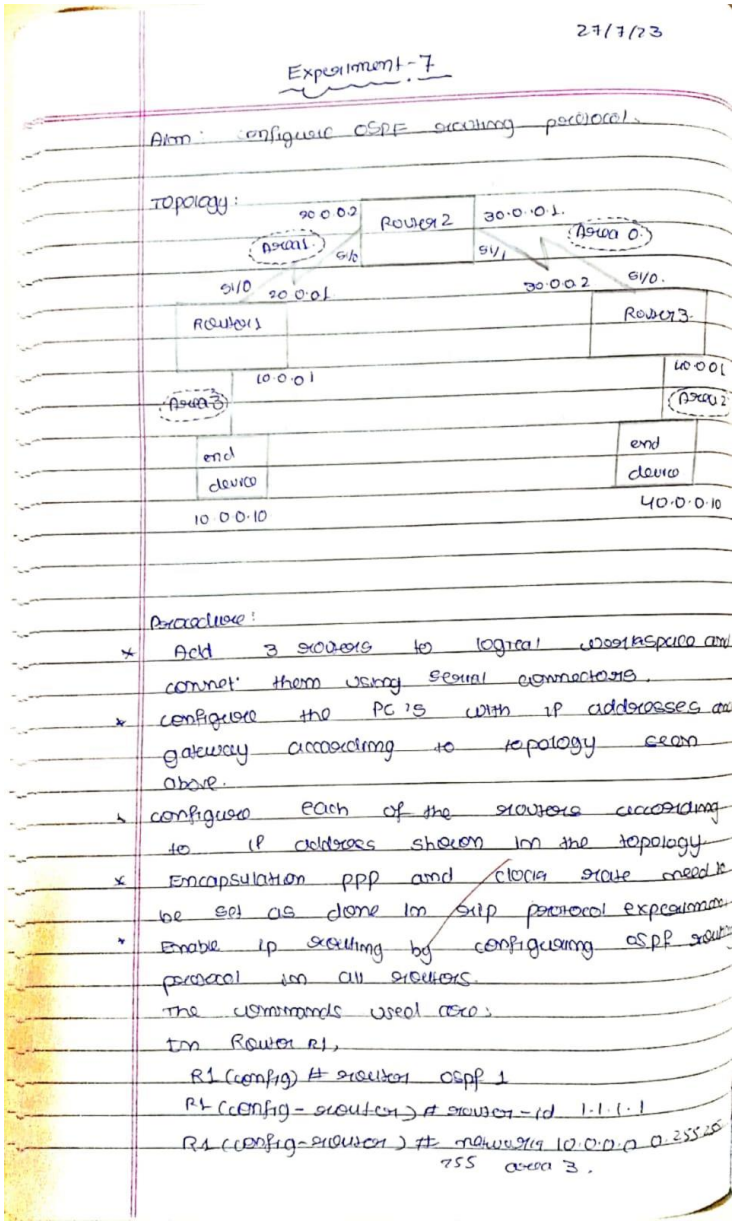

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Request timed out.
Reply from 40.0.0.1: bytes=32 time=8ms TTL=125
Reply from 40.0.0.1: bytes=32 time=5ms TTL=125
Reply from 40.0.0.1: bytes=32 time=10ms 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>


Cisco Packet Tracer Student - C:\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\CN\CN_LAB\lab6.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT
Router1
Router-PT
Router0
Router-PT
Router2
PC-PT
PC0
10.0.0.1
PC-PT
PC1
40.0.0.1
20.0.0.20
30.0.0.10
10.0.0.10
20.0.0.10
30.0.0.20
40.0.0.10
Simulation Panel
Event List
Vis. Time(sec) Last De At Dev Type Info
0.006 Router2 Rout... ICMP
0.007 Router1 Rout... ICMP
0.008 Router0 PC0 ICMP
12.790 -- Rout... RIPv1
12.790 -- Rout... RIPv1
Reset Simulation
Constant Delay
Captured to: 12.790 s
Play Controls
Back
Auto Capture / Play
Capture / Forward
Event List Filters - Visible Events
ACL Filter, ARP, BGP, CDP, DHCP, DHCPv6, DNS, DTP, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, LACP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, RADIUS, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, VTP
Edit Filters
Show All/None
Time: 00:01:22.953
Power Cycle Devices PLAY CONTROLS: Back
Auto Capture / Play
Capture / Forward
Event List
Simulation
Routers
Scenario 0
New
Delete
Fire Last Statu Sourc Destinatic Type Colo Time(s Period Num Edit Delete
Successful PC0 PC1 IC... 0.000 N 0 (ed... (delete)

WEEK 7

Configure OSPF routing protocol.

OBSERVATION:



27/7/23

Experiment - 7

Aim: configure OSPF routing protocol.

Topology:

Procedure:

* Add 3 routers to logical workspace and connect them using serial connections.
* configure the PC's with IP addresses and gateway according to topology given above.
* configure each of the routers according to IP address shown in the topology.
* Encapsulation ppp and clock rate need to be set as done in RIP protocol experiment.
* Enable ip routing by configuring OSPF routing protocol in all routers.

The commands used are:

In Router R1,

R1(config) # router ospf 1

R1(config-router) # router-id 1.1.1.1

R1(config-router) # network 10.0.0.0 0.255.255.255 area 3.

Date 27/7/23
Page

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

* In Router R2,
R2(config)# router ospf 1.
R2(config-router)# router-id 2.2.2.2
R2(config-router)# network 20.0.0.0 0.255.255.255 area 1.
R2(config-router)# network 30.0.0.0 0.255.255.255 area 0.
R2(config-router)# exit.

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

* Assign loopback interfaces.
In Router R1:
R1(config-if)# interface loopback 0.
R1(config-if)# ip address 172.16.1.252 255.255.0.0
R1(config-if)# no shutdown

* In Router R2:
R2(config-if)# interface loopback 0.
R2(config-if)# ip add 172.16.1.253 255.255.0.0
R2(config-if)# no shutdown

* R3(config-if)# interface loopback 0
R3(config-if)# ip address 172.16.1.254 255.255.0.0.
R3(config-if)# no shutdown.

* Now create virtual links between R1, R2 by this using the following commands:

In Router R1,

R1(config)# router ospf 1

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

R1(config-router)# exit

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

R2(config-router)# exit.

* To test the connection ping from 10.0.0.10 to 40.0.0.10.

Result:

>> 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=11ms TTL = 125

Reply from 40.0.0.10 : bytes=32 time=11ms TTL = 125

Reply from 40.0.0.10 : bytes=32 time=8ms TTL = 125

ping statistics for 40.0.0.10

packets sent = 4, Received = 3, lost = 1 (25% loss)

Approximate round trip in milli-seconds

minimum = 8ms, Maximum = 11ms, Average = 10ms

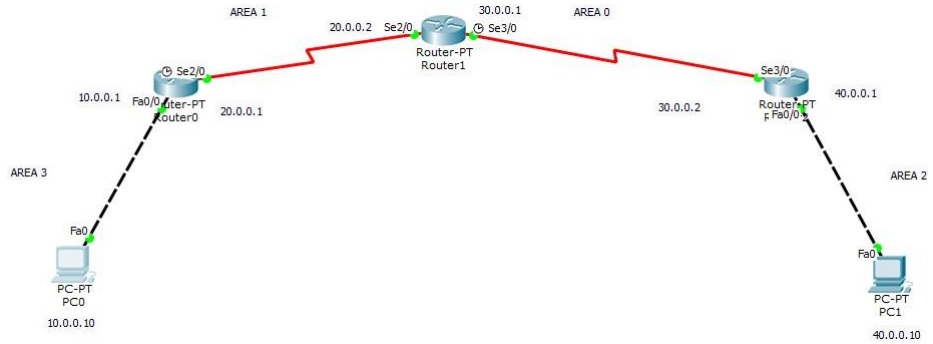
Observation:

* OSPF is a link-state routing protocol that is used to find the best path between source and destination routers using its own SPF algorithm.

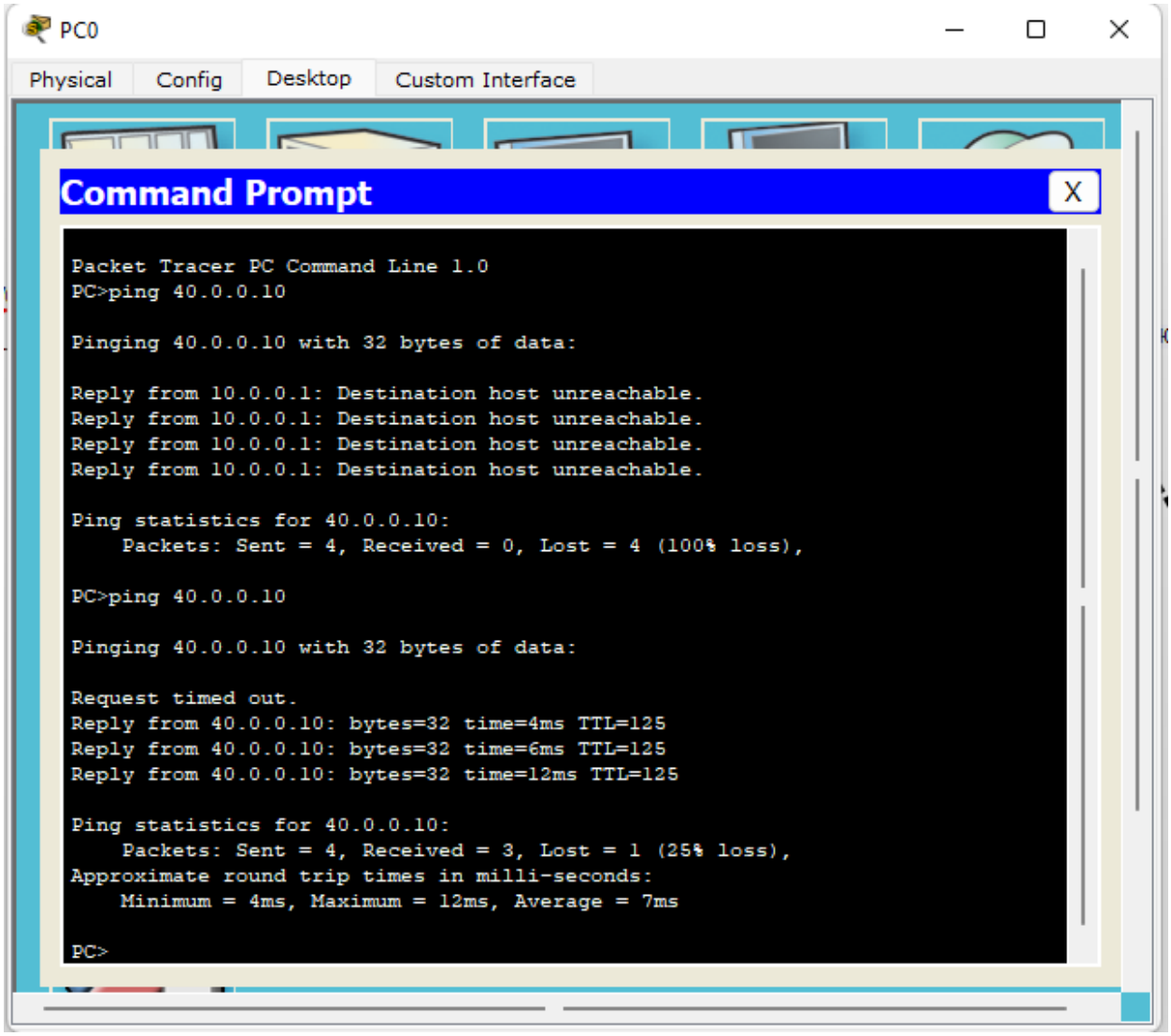
Date __/__/__
Page _____

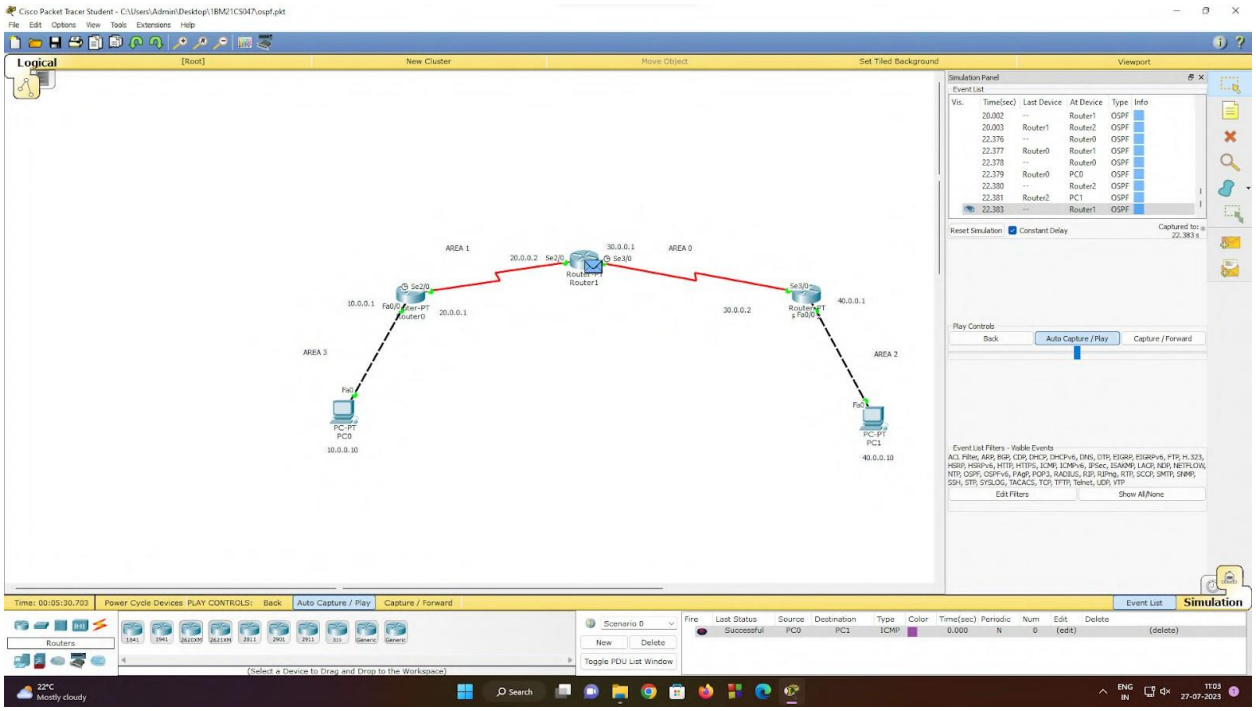
* This network is divided into 4 areas where area 0 is the backbone.
* After we make the virtual - link between the area which is not connected to the backbone area 0, we can ping messages successfully.

TOPOLOGY:



OUTPUT:

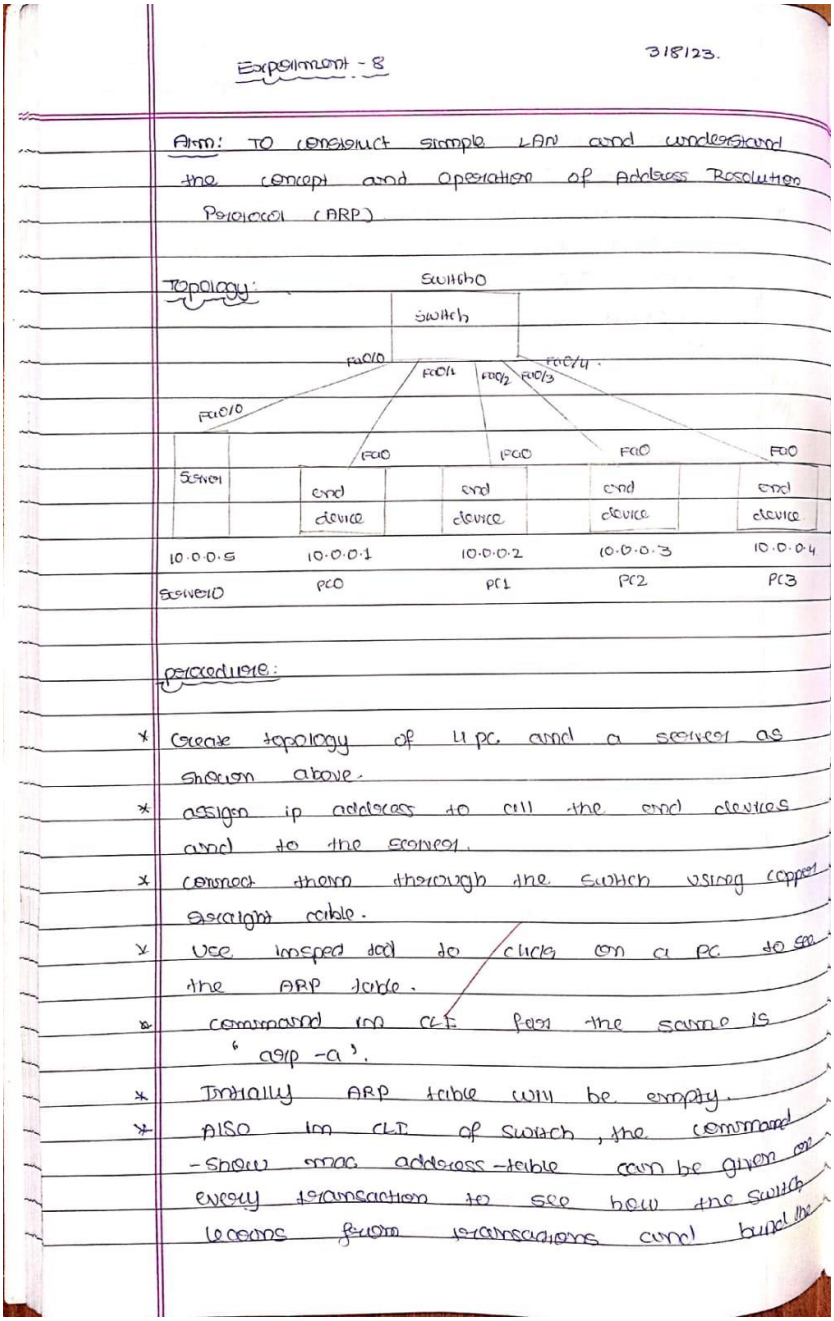



Cisco Packet Tracer Student - C:\Users\Admin\Desktop\1BM21CS047\ospf.pkt
Logical
AREA 1
AREA 0
AREA 3
AREA 2
20.0.0.2
30.0.0.1
10.0.0.1
20.0.0.1
30.0.0.2
40.0.0.1
10.0.0.10
40.0.0.10
Router1
Router0
PC0
PC1
Simulation Panel
Event List
Play Controls
Back
Auto Capture / Play
Capture / Forward
Simulation

WEEK 8

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

OBSERVATION:



Experiment - 8

318123.

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

Topology:

procedure:

* Create topology of 4 pc and a server as shown above.
* assign ip address to all the end devices and to the server.
* connect them through the switch using copper straight cable.
* Use inspect tool to click on a PC to see the ARP table.
* command in CLI for the same is 'arp -a'.
* Initially ARP table will be empty.
* Also in CLI of switch, the command - show mac address-table can be given on every transaction to see how the switch learns from transactions and build the

Date 3/8/23.
Page

address table.

* Use capture button in the simulation panel to go step by step so that the changes in ARP can be clearly noted.
* Observe the switch as well the nodes update the ARP table as and when a new communication starts.

Result:

SERVER > arp -a.

Internet address	Physical Address	Type
10.0.0.1	0040.0b3a.7d29	dynamic
10.0.0.2	000b.be3e.708c	dynamic
10.0.0.3	0002.16d6.2205	dynamic
10.0.0.4	00e0.8f57.838a	dynamic

Switch> show mac address-table

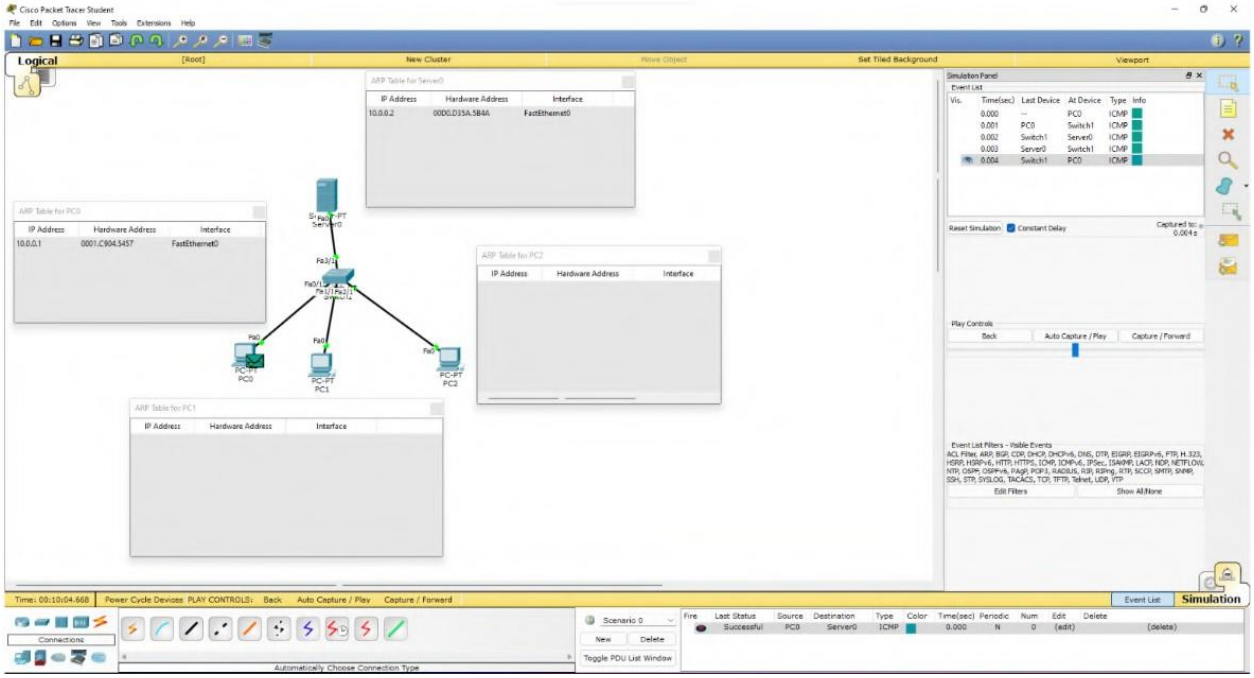
Mac address table

Vlan	Mac Address	Type	Ports
1	0002.16d6.2205	DYNAMIC	Fa2/1
1	0004.9a36.38e0	DYNAMIC	Eth 6/1
1	000b.be3e.708c	DYNAMIC	Fa1/1
1	0040.0b3a.7d29	DYNAMIC	Fa0/1
1	00e0.8f57.838a	DYNAMIC	Fa3/1

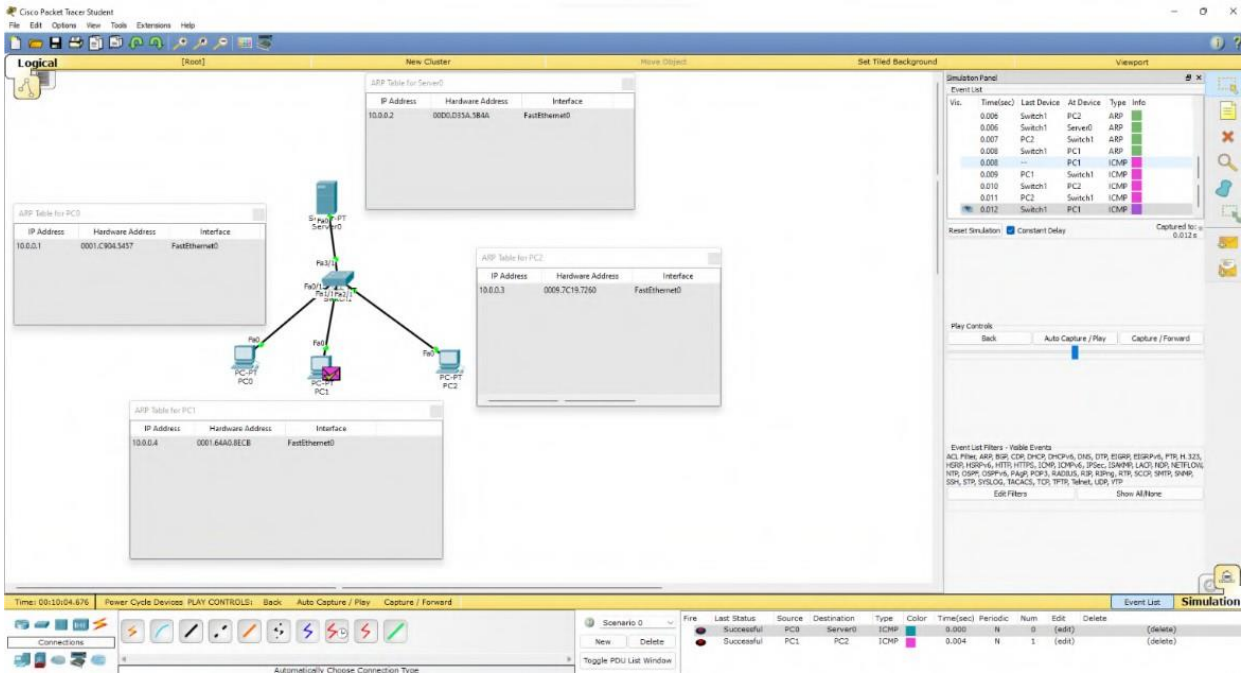
Observation:

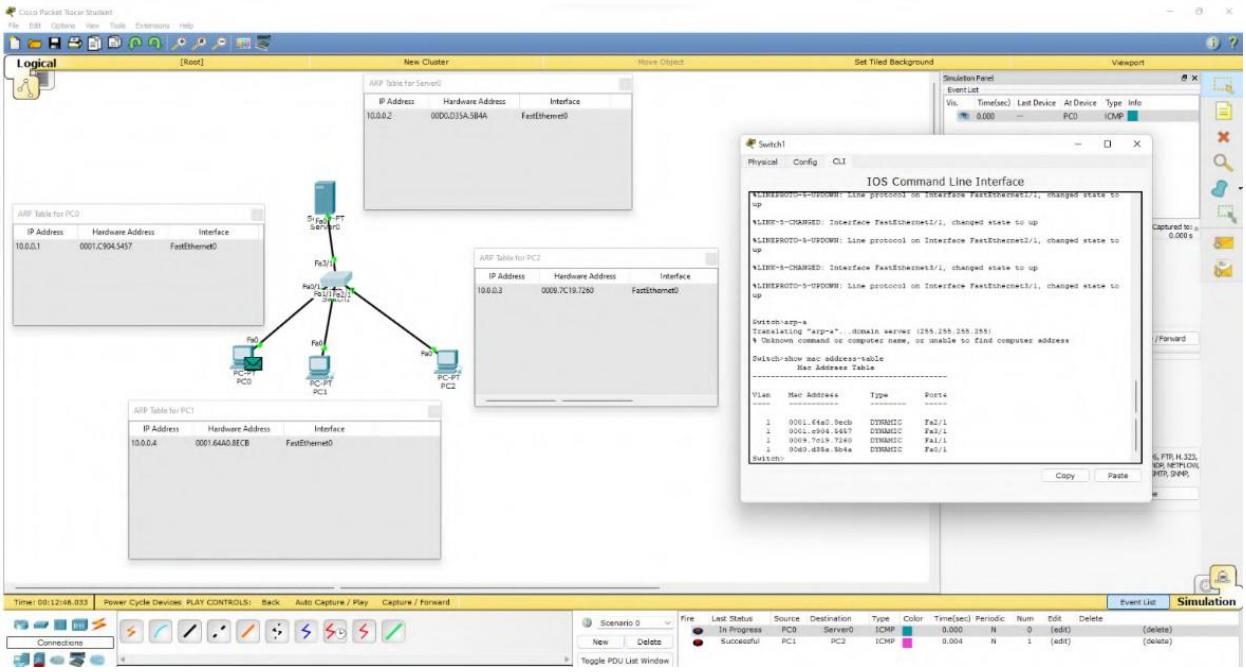
* We have observed that the switch as well as the nodes has updated the ARP table as and when a new communication starts.

TOPOLOGY:



OUTPUT:

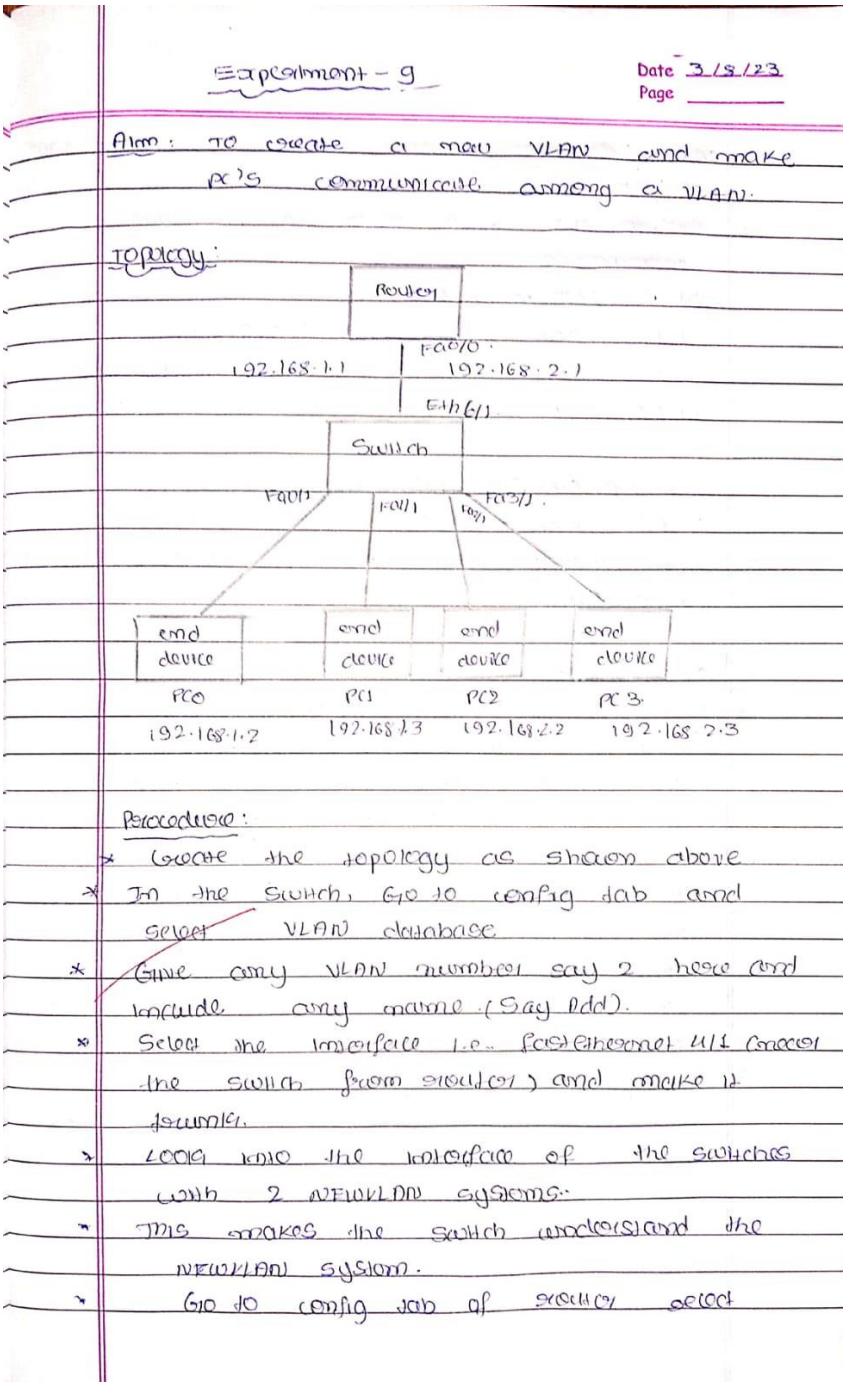



Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Switch1
Physical
Config
CLI
IOS Command Line Interface
Copy
Paste
Simulation

WEEK 9

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

OBSERVATION:



Experiment - 9

Date 3/8/23
Page

Aim: To create a new VLAN and make PC's communicate among a VLAN.

Topology:

Procedure:

* Create the topology as shown above
* In the switch, go to config tab and select VLAN database
* Give any VLAN number say 2 here and include any name (Say Add).
* Select the interface i.e. fastethernet 1/1 (where the switch from router) and make it dynamic.
* Look into the interface of the switches with 2 NEWVLAN systems.
* This makes the switch understand the NEWVLAN system.
* Go to config tab of router select

VLAN database enter the number and name of the VLAN created.

* Go to CLI and enter the following commands:

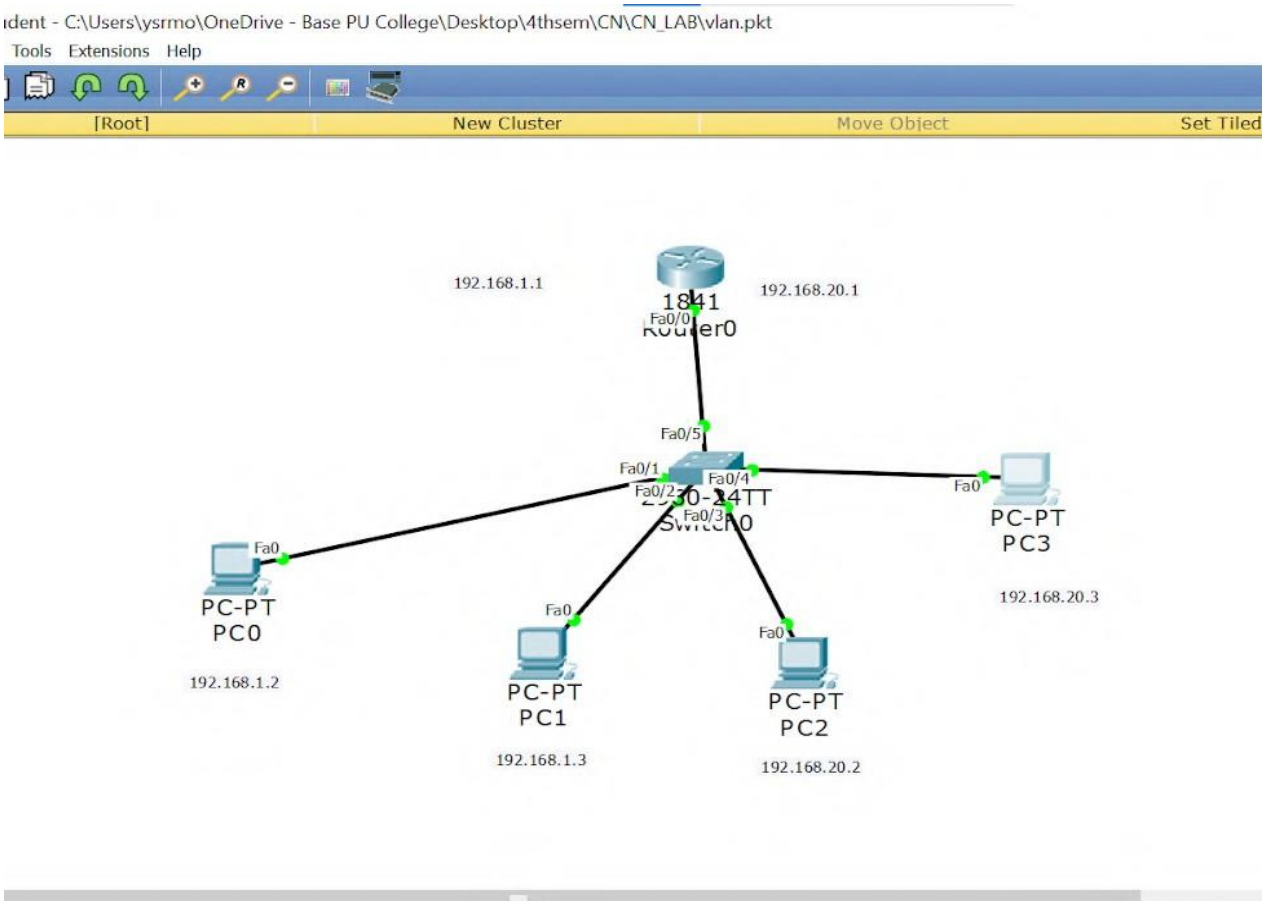
1] Router# config t
2] Router(config)# interface fa0/0.1
3] Router(config-subif)# encapsulation dot 1q 2
4] Router(config-subif)# ip address 192.168.2.1 255.255.255.0
5] Router(config-subif)# no shut
6] Router(config-subif)# exit

* This makes the router understand about new VLAN.

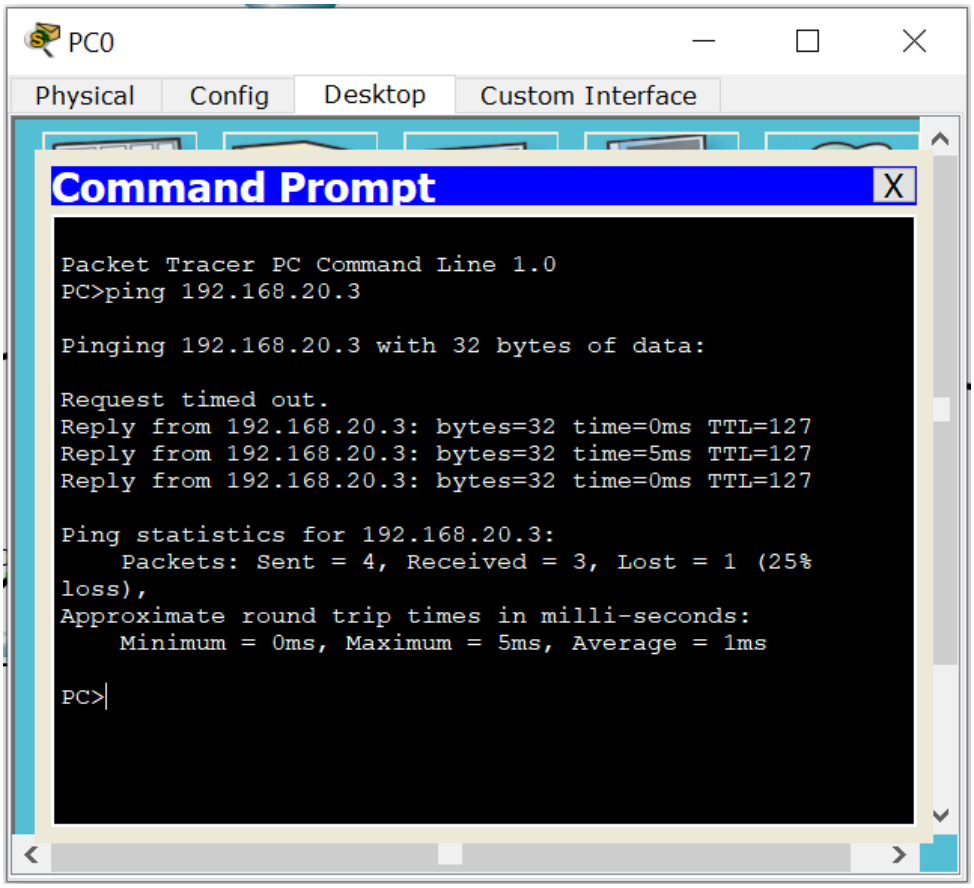
Observation:

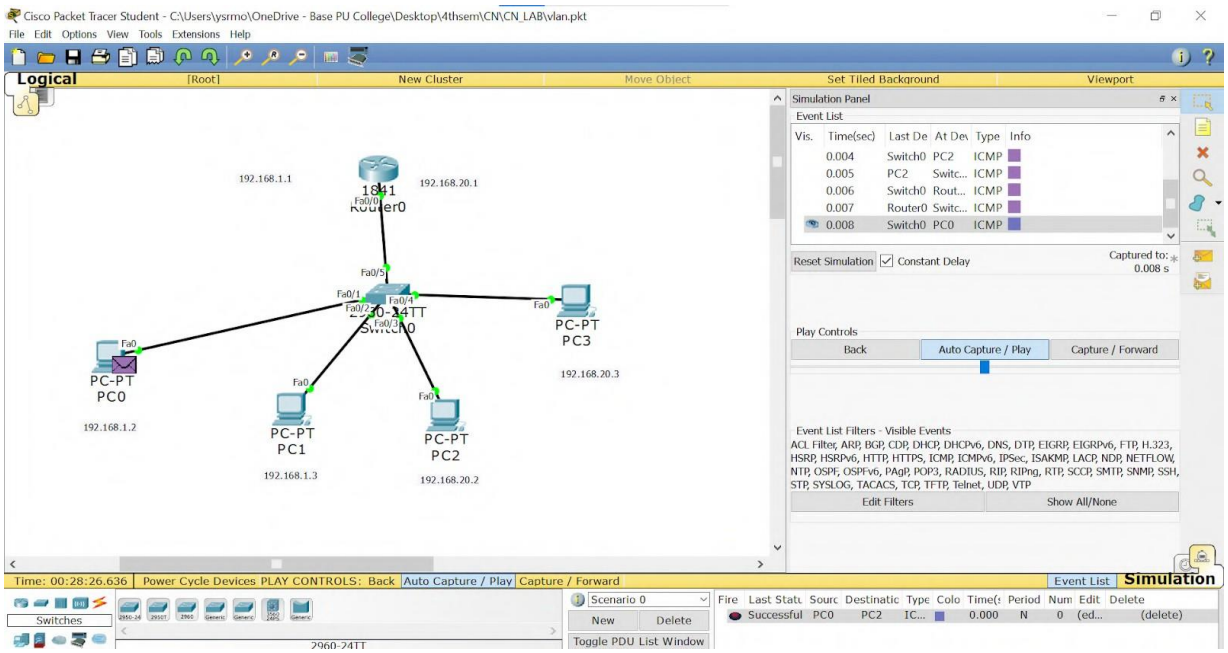
* VLAN stands for virtual local area network.
* VLAN is a custom network which is created from one or more local area networks. It enables a group of device available in multiple networks to be combined into one logical network. The result becomes a virtual LAN that is administered exactly like a physical VLAN. It is a virtual extension of LAN.
* encapsulation dot1q is the networking standard that supports virtual LANs on an IEEE 802.3 Ethernet network.

TOPOLOGY:


ident - C:\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\CN\CN_LAB\vlan.pkt
Tools Extensions Help
[Root]
New Cluster
Move Object
Set Tiled
192.168.1.1
1841
Fa0/0
192.168.20.1
Fa0/5
Fa0/1
Fa0/4
Fa0/3
Fa0
PC-PT
PC0
192.168.1.2
Fa0
PC-PT
PC1
192.168.1.3
Fa0
PC-PT
PC2
192.168.20.2
Fa0
PC-PT
PC3
192.168.20.3

OUTPUT:

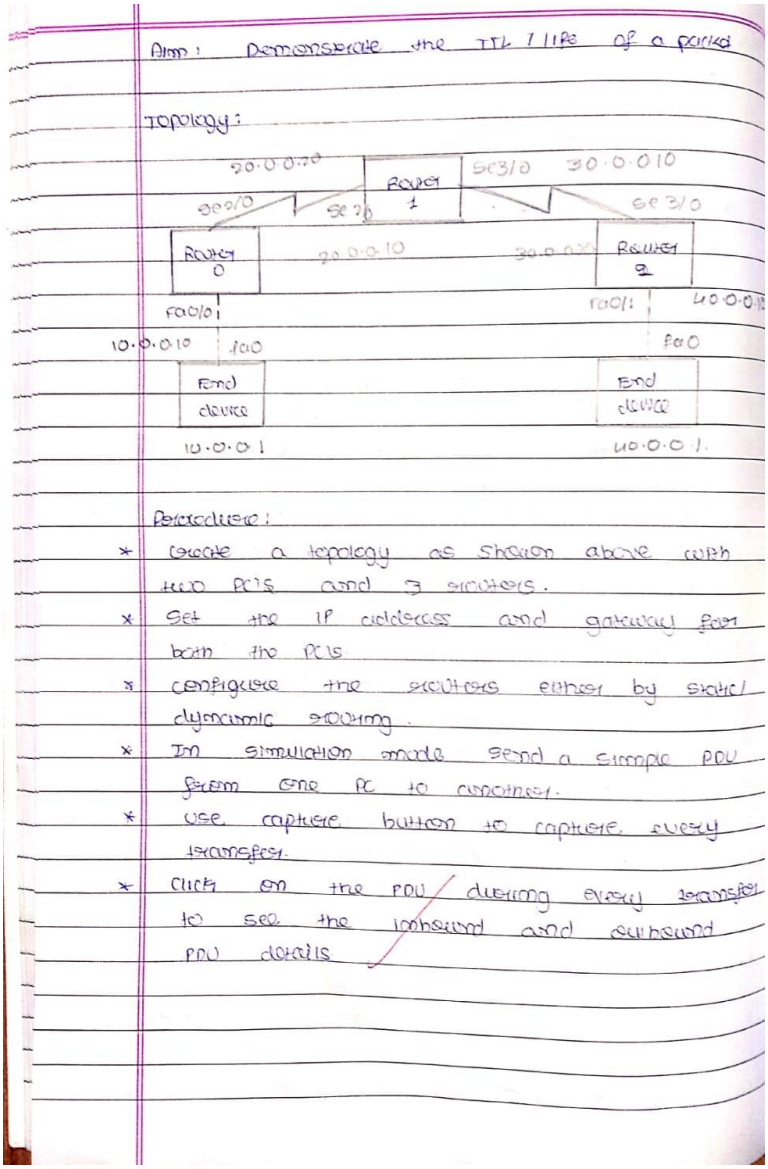

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 192.168.20.3
Pinging 192.168.20.3 with 32 bytes of data:
Request timed out.
Reply from 192.168.20.3: bytes=32 time=0ms TTL=127
Reply from 192.168.20.3: bytes=32 time=5ms TTL=127
Reply from 192.168.20.3: 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>


Logical
PC-PT PC0
PC-PT PC1
PC-PT PC2
PC-PT PC3
192.168.1.1
192.168.20.1
192.168.1.2
192.168.1.3
192.168.20.2
192.168.20.3
Simulation Panel
Event List
Play Controls
Back
Auto Capture / Play
Capture / Forward
Simulation

WEEK 10

Demonstrate the TTL/ Life of a Packet.

OBSERVATION:



Date __/__/__
Page _____

device.

* In the config-tab a new wireless interface would have been added. Now configure SSID, WEP, WEP key, IP address and gateway to the device.

x ping from every device to every other device.

Result:

Ping output:

packet tracer PC command line

PC> ping 10.0.0.3

pinging 10.0.0.3 with 32 bytes of data:

Request timed out.

Reply from 10.0.0.3 : bytes=32 time=0ms TTL=127

Reply from 10.0.0.3 : bytes=32 time=0ms TTL=127

Reply from 10.0.0.3 : bytes=32 time=2ms TTL=127

Ping statistics for 10.0.0.3

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

Approximate round trip time in milliseconds:

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

Observation:

* A WLAN is a group of connected devices that forum a network based on radio transmissions.

x data sent in packets contains layers which labels and instructions. MAC address to endpoints for routing.

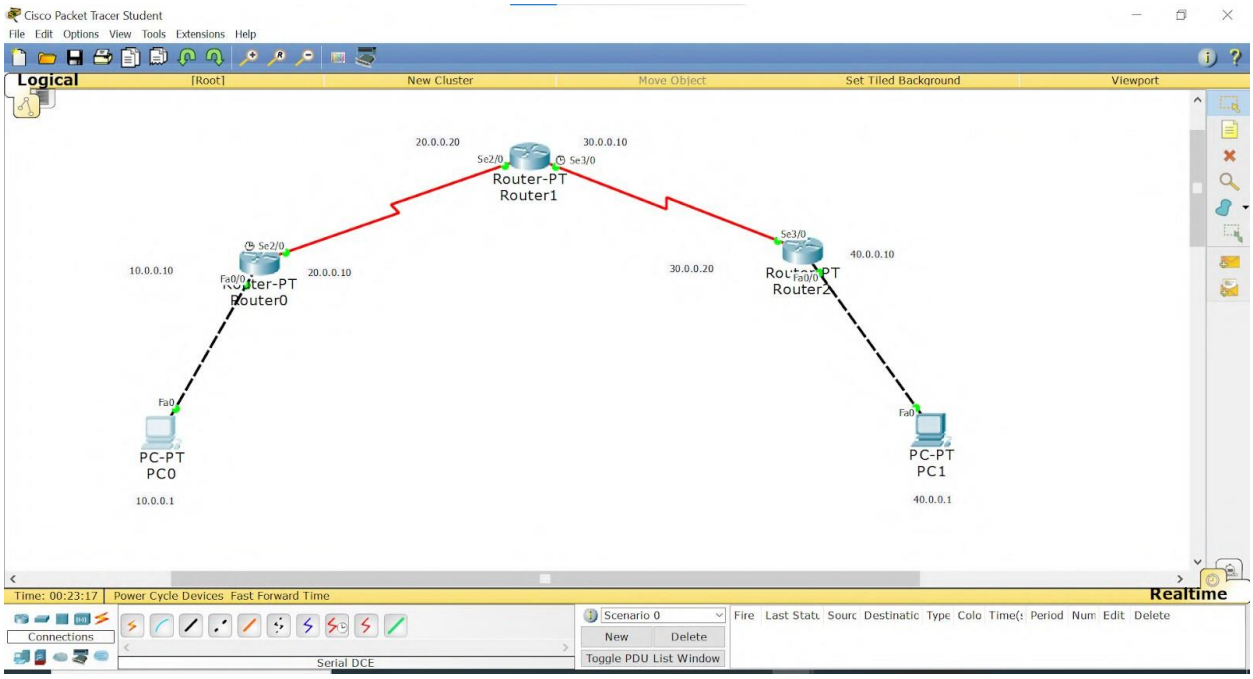
* The access point is the base station that serves as a hub to which other stations can connect.

* With one access point we can connect multiple devices.

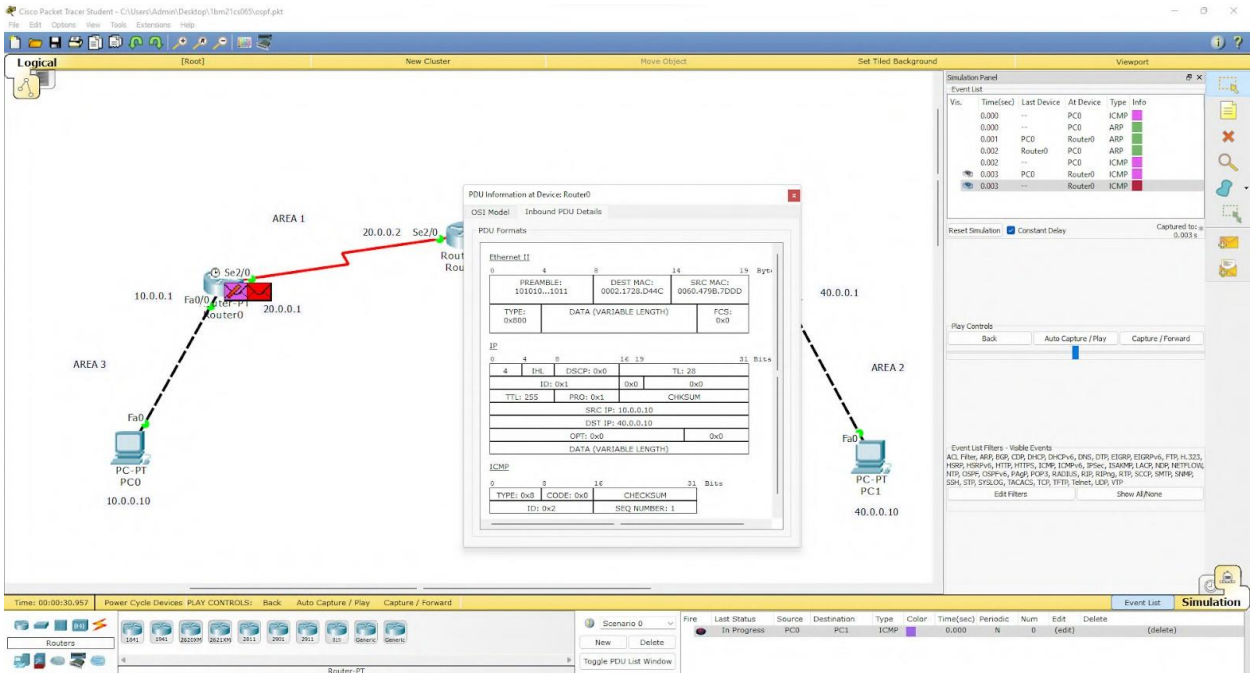
Observation:

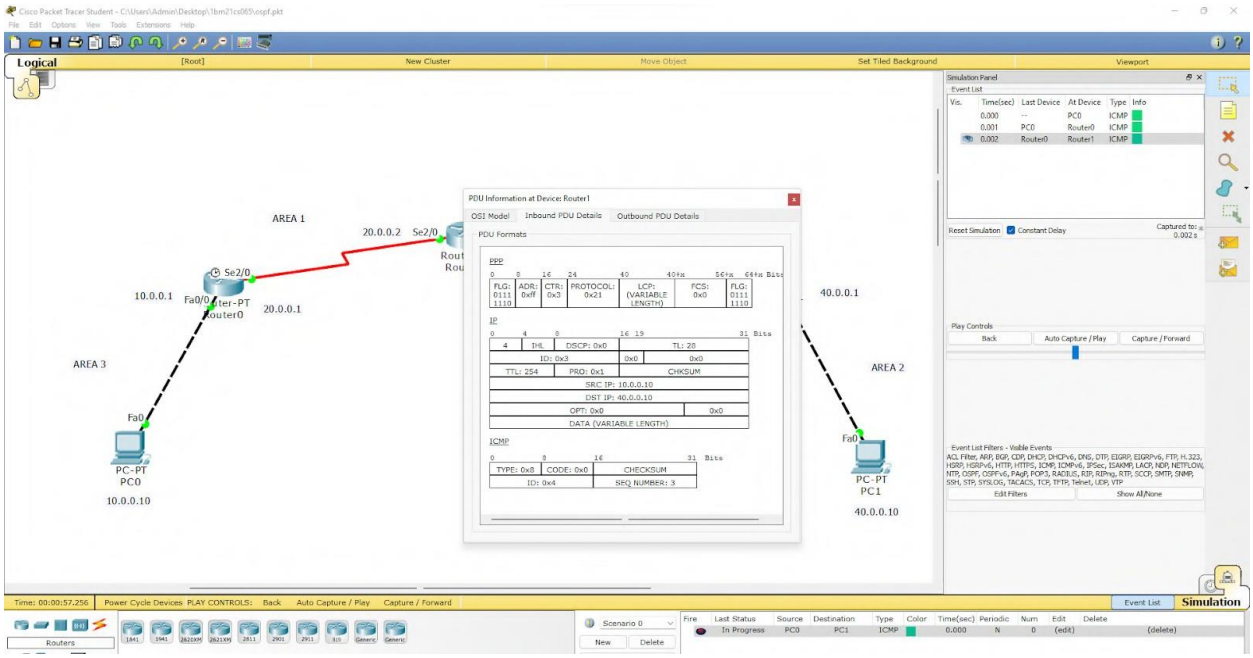
* TELNET stands for Teletype network. It is a type of protocol that enables one computer to connect to the local computer.
* It is used as a standard TCP/IP protocol for virtual terminal provided by ISO.
* During TELNET operation, whatever is being performed on the remote computer will be displayed by the local computer. TELNET operates on the client / server principle.

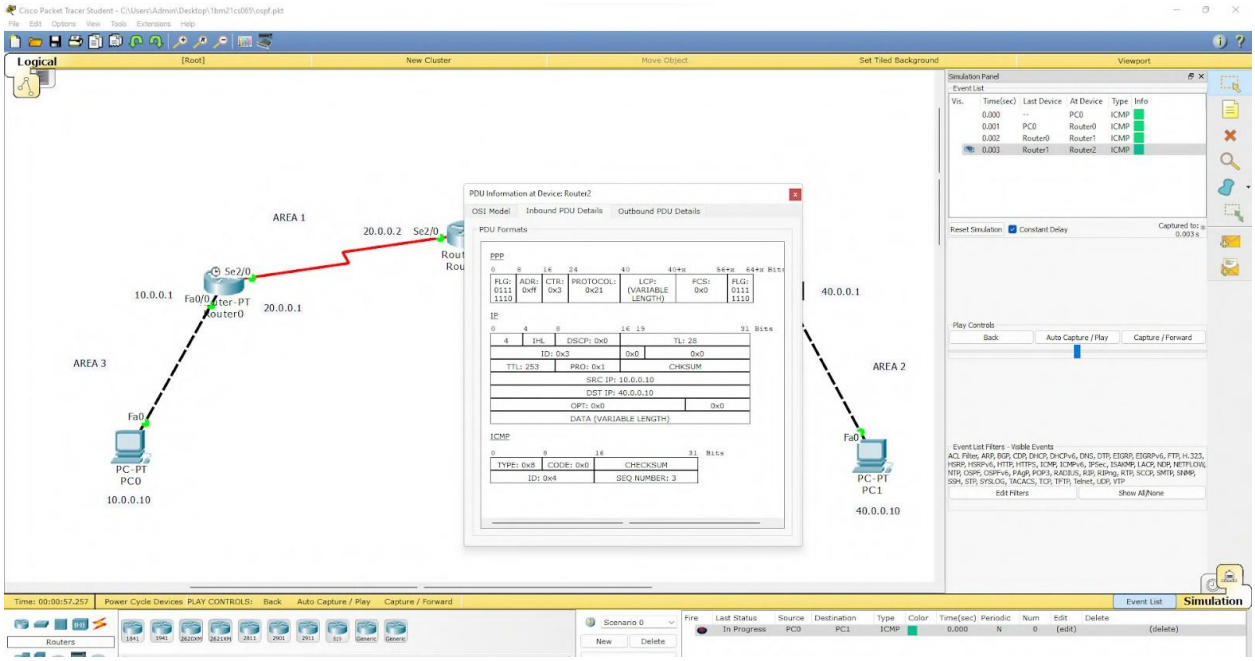
TOPOLOGY:

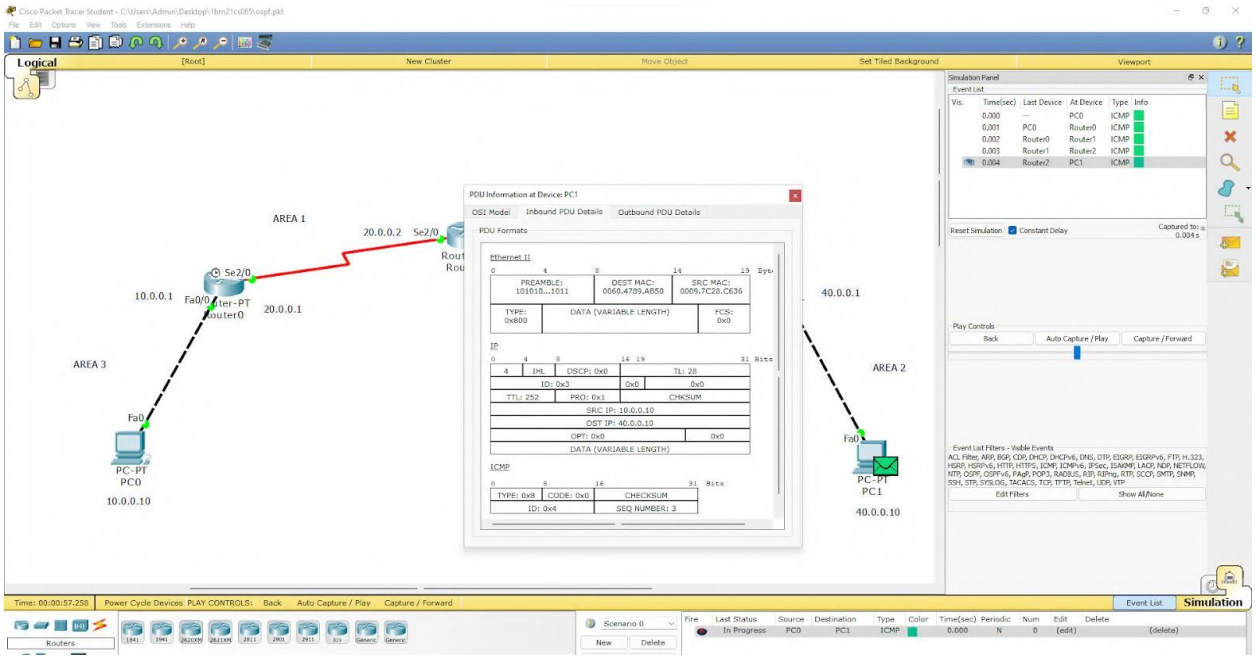


OUTPUT:




Logical
AREA 1
20.0.0.2
Se2/0
10.0.0.1
Fa0/0
Router-PT
Router0
20.0.0.1
AREA 3
AREA 2
40.0.0.1
Fa0
PC-PT
PC0
10.0.0.10
PC-PT
PC1
40.0.0.10
PDU Information at Device: Router1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
PPP
IP
ICMP
TTL: 254
TYPE: 0x8
CODE: 0x0
CHECKSUM
SEQ NUMBER: 3
Simulation Panel
Event List
Time: 00:00:57.256
Power Cycle Devices
PLAY CONTROLS:
Back
Auto Capture / Play
Capture / Forward
Event List
Simulation

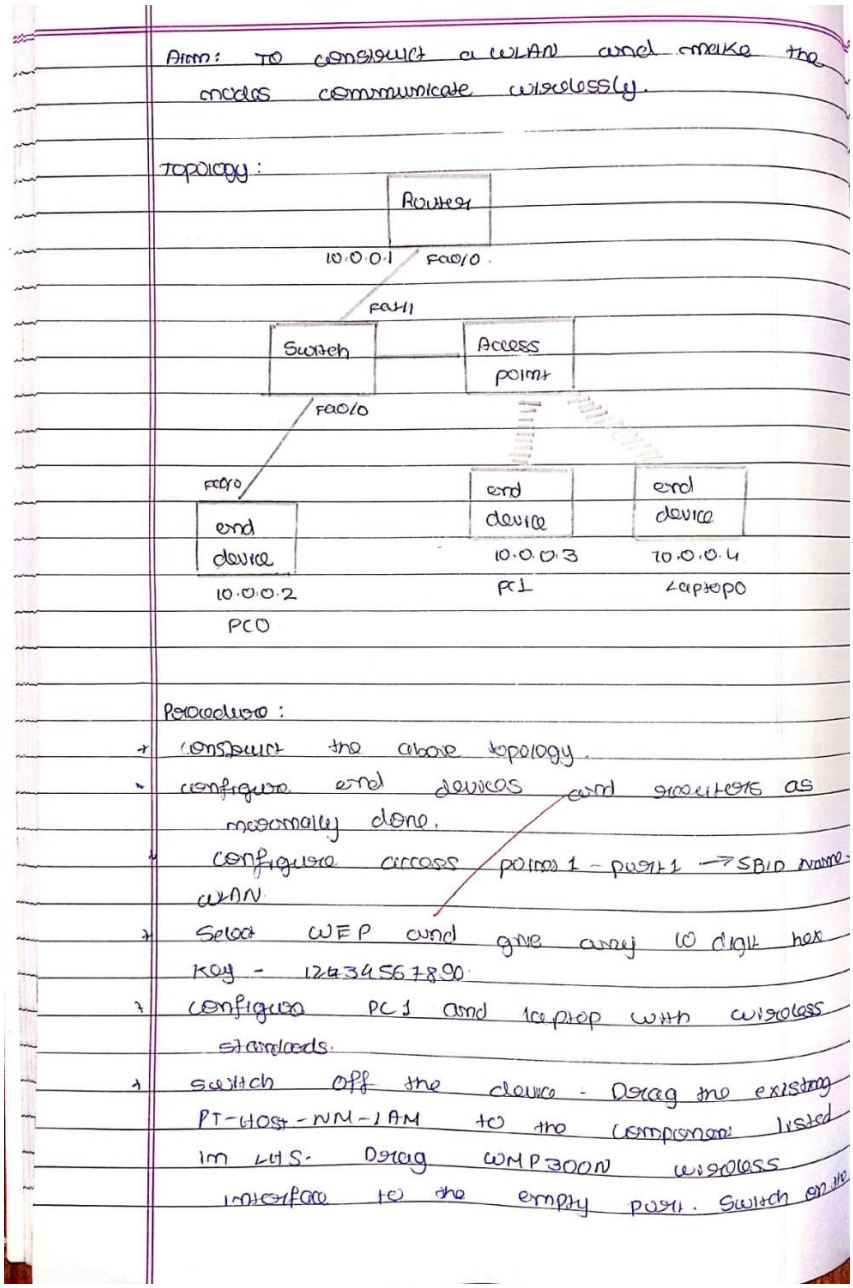

Logical
AREA 1
20.0.0.2
Se2/0
10.0.0.1
Fa0/0
Router-PT
Router0
20.0.0.1
AREA 3
AREA 2
40.0.0.1
Fa0
PC-PT
PC0
10.0.0.10
PC-PT
PC1
40.0.0.10
PDU Information at Device: Router2
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
PPP
IP
ICMP
TTL: 253
TYPE: 0x8
CODE: 0x0
CHECKSUM
SEQ NUMBER: 3
Simulation Panel
Event List
Time: 00:00:57.257
Power Cycle Devices
PLAY CONTROLS:
Back
Auto Capture / Play
Capture / Forward
Event List
Simulation


Logical
AREA 1
20.0.0.2
Se2/0
10.0.0.1
Fa0/0
Router-PT
Router0
20.0.0.1
AREA 3
AREA 2
40.0.0.1
Fa0
PC-PT
PC0
10.0.0.10
PC-PT
PC1
40.0.0.10
PDU Information at Device: PC1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
PREAMBLE: 101010...1011
DEST MAC: 0060.4789.AB50
SRC MAC: 0009.7C28.C636
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
IHL
DSCP: 0x0
TL: 28
ID: 0x3
TTL: 252
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.10
DST IP: 40.0.0.10
OPT: 0x0
DATA (VARIABLE LENGTH)
ICMP
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x4
SEQ NUMBER: 3
Simulation Panel
Event List
Play Controls
Back
Auto Capture / Play
Capture / Forward
Event List Filters - Visible Events
Edit Filters
Show All/None
Time: 00:00:57.258
Power Cycle Devices
PLAY CONTROLS:
Routers
Scenario 0
New
Delete
Simulation

WEEK 11

To construct a WLAN and make the nodes communicate wirelessly

OBSERVATION:



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

Topology:

Procedure:

- Construct the above topology.
- Configure end devices and switches as normally done.
- Configure access point 1 - port 1 → SSID name: WLAN
- Select WEP and give any 10 digit hex key - 1234567890
- Configure PC 1 and laptop with wireless standards.
- Switch off the device - Drag the existing PT-HOST-NM-1AM to the components listed in LHS. Drag WMP300N wireless interface to the empty port. Switch on the

10/8/23

Date __/__/__
Page _____

of a packet

0.0.0.10

Router 2

Fa0/1 | 40.0.0.10

Fa0

End device

40.0.0.1.

above with

gateway for

by static/

simple PDU

ne every

very transfer

outbound

Output:

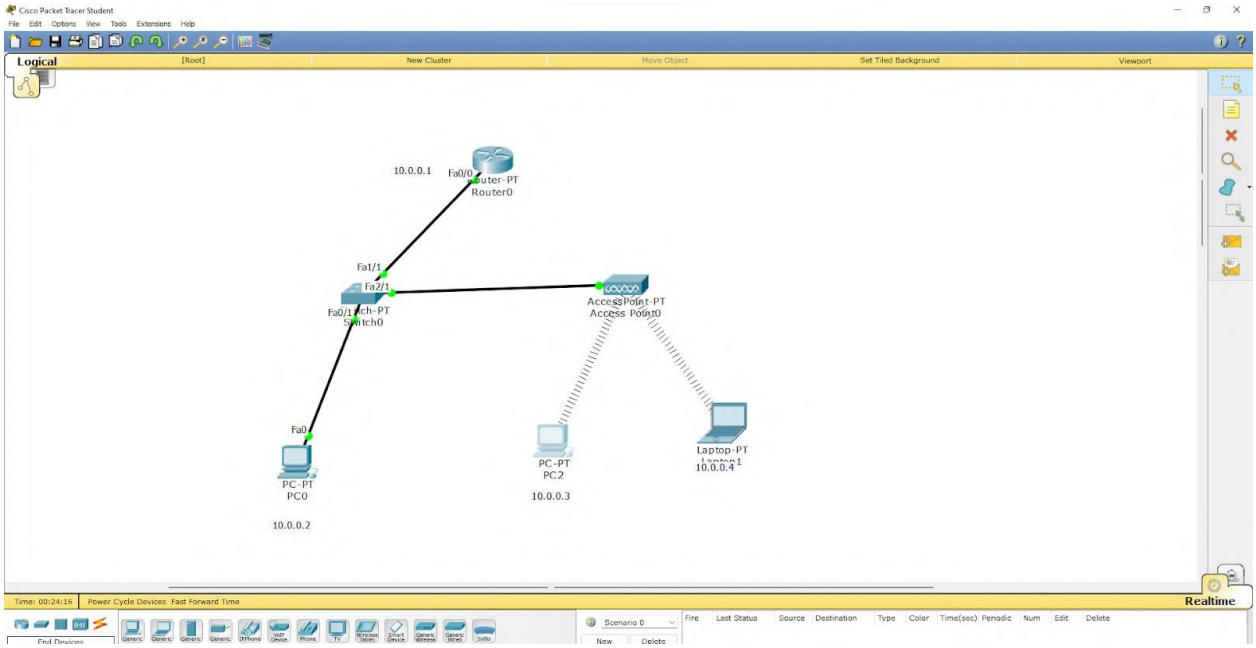
IP

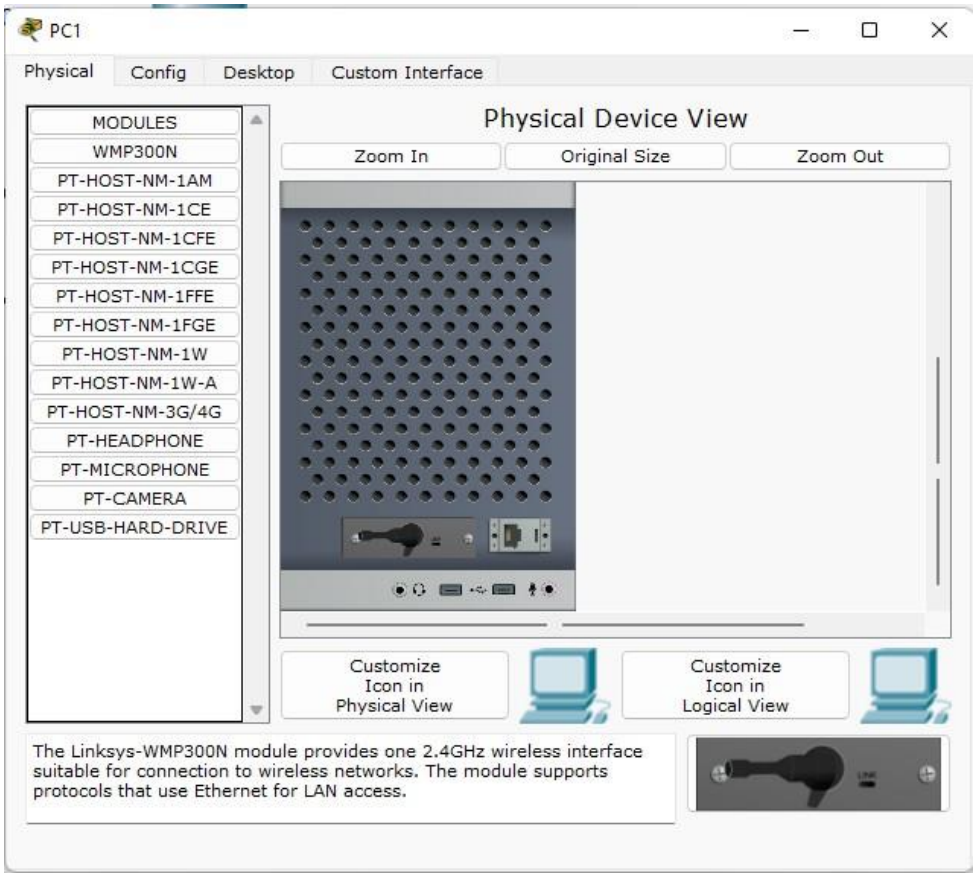
0	4	8	16	19	31
4	IHL	DSCP		TL: 28	
ID: 0x6			0x	0x0	
TTL: 255		PRO: 0x1		CHKSUM	
SRC IP: 10.0.0.1					
DST IP: 20.0.0.1					
OPT 0x0					0x0
DATA (VARIABLE LENGTH)					

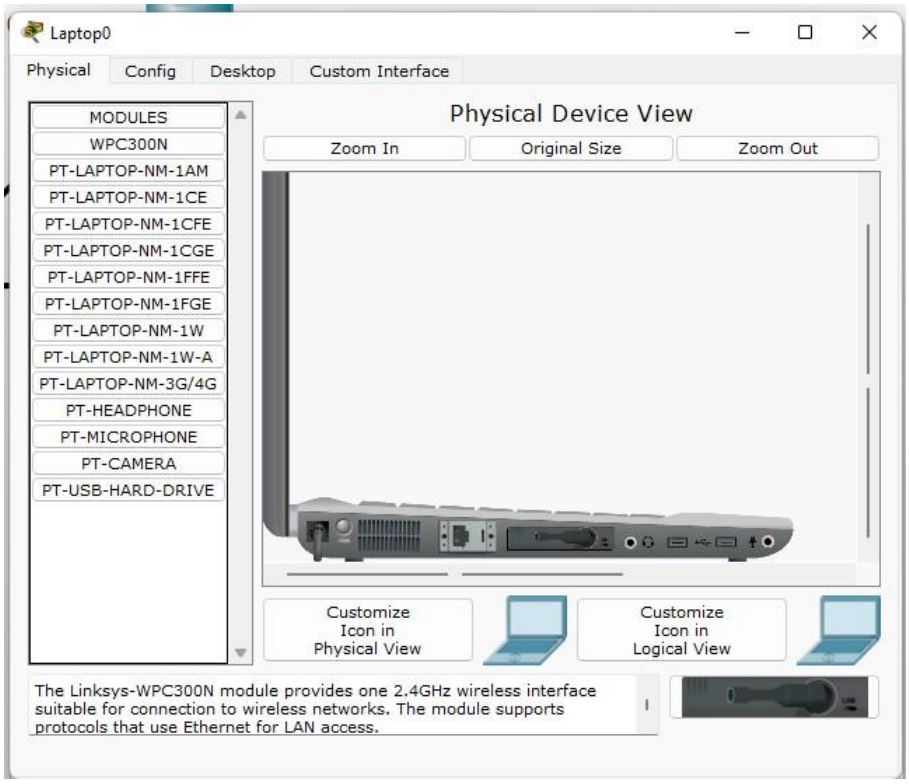
Observation:

* The no of hops the packet travel before being discarded is recorded as TTL.
* Datagrams TTL field is set by sender and reduced by each router along the path to its destination.
* The router reduces TTL value by one while forwarding the packets.
* When TTL value is 0, the router discards it and sends ICMP message.

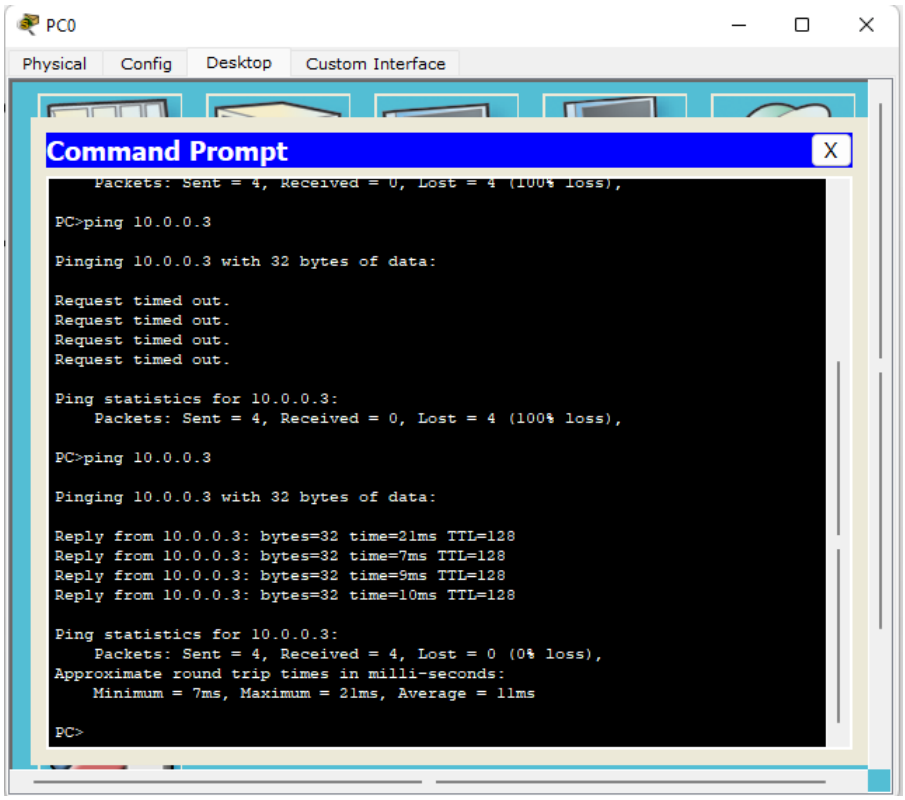
TOPOLOGY:


Logical
10.0.0.1
Fa0/0
Router-PT
Router0
Fa1/1
Fa2/1
Fa0/1
Switch0
Access Point-PT
Access Point0
Fa0
PC-PT
PC0
10.0.0.2
PC-PT
PC2
10.0.0.3
Laptop-PT
10.0.0.4
Realtime


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.

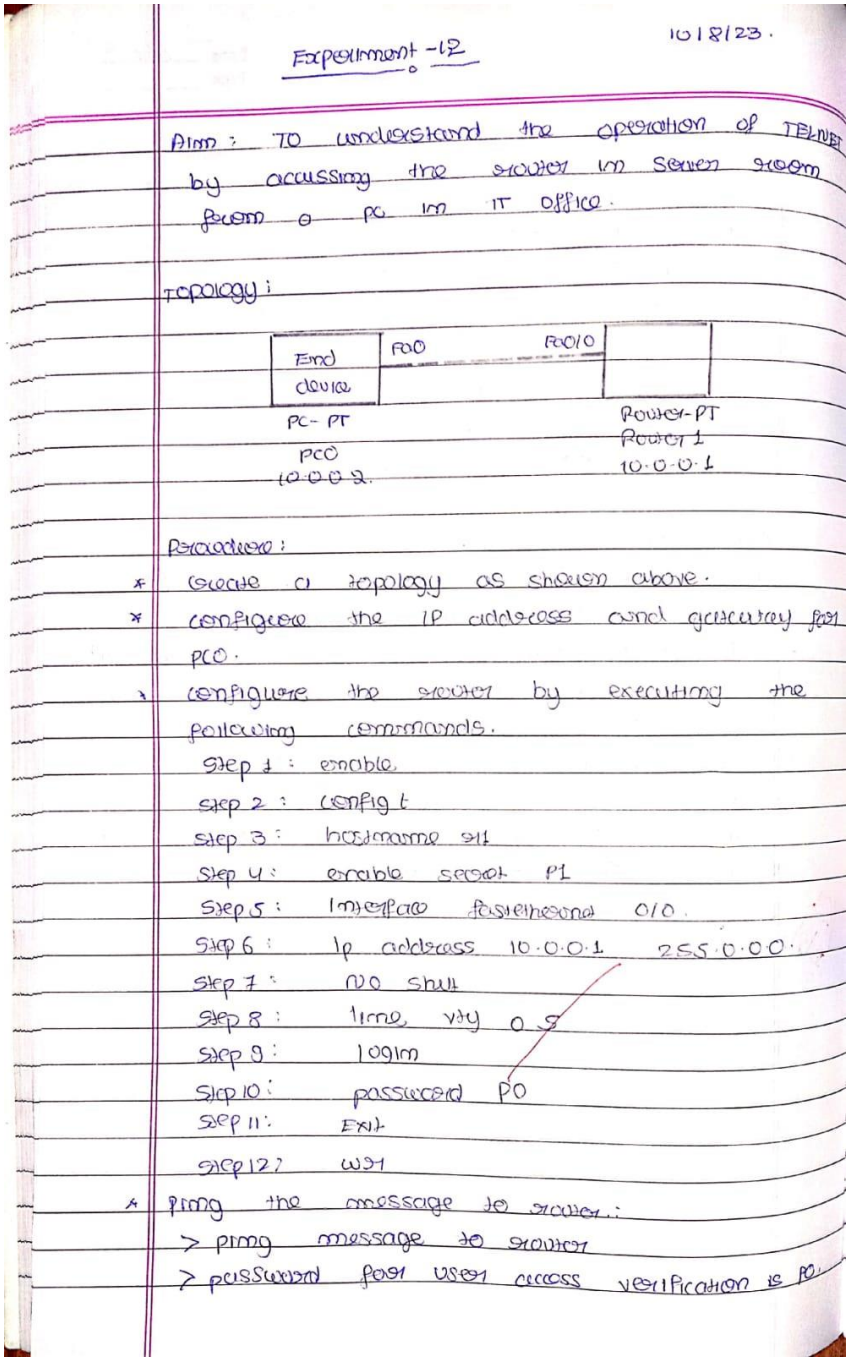
OUTPUT:


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
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:



Experiment -12

10/8/23.

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

Topology:

End device	Fa0	Fa0/0	
PC-PT PC0 10.0.0.2			Router-PT Router1 10.0.0.1

Procedure:

* Create a topology as shown above.
* Configure the IP address and gateway for PC0.
* Configure the router by executing the following commands.
  - Step 1: enable
  - Step 2: config t
  - Step 3: hostname R1
  - Step 4: enable secret P1
  - Step 5: interface fastethernet 0/0
  - Step 6: ip address 10.0.0.1 255.0.0.0
  - Step 7: no shut
  - Step 8: line vty 0 5
  - Step 9: login
  - Step 10: password P0
  - Step 11: Exit
  - Step 12: wr
* Ping the message to router:
  - > ping message to router
  - > password for user access verification is P0.

3/23.

of TELNET
room

Date __/__/__
Page ______

> password for enable is P1.
> Accessing router CLI from PC
> show IP route

ccuracy for

the

Result:

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 milliseconds:
minimum: 0ms, Maximum: 0ms, Average: 0ms.

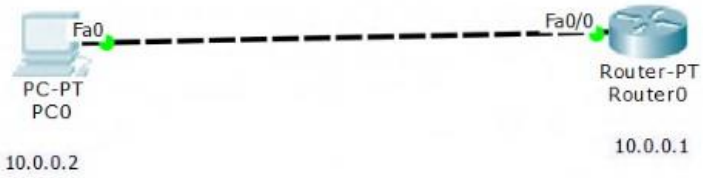
PC> telnet 10.0.0.1.
Typing 10.0.0.1 ... open
user access verification
Password: P0
R1 > enable
Password: P1
R1# show ip route.

C 10.0.0.0/8 is directly connected, Fa0/0.

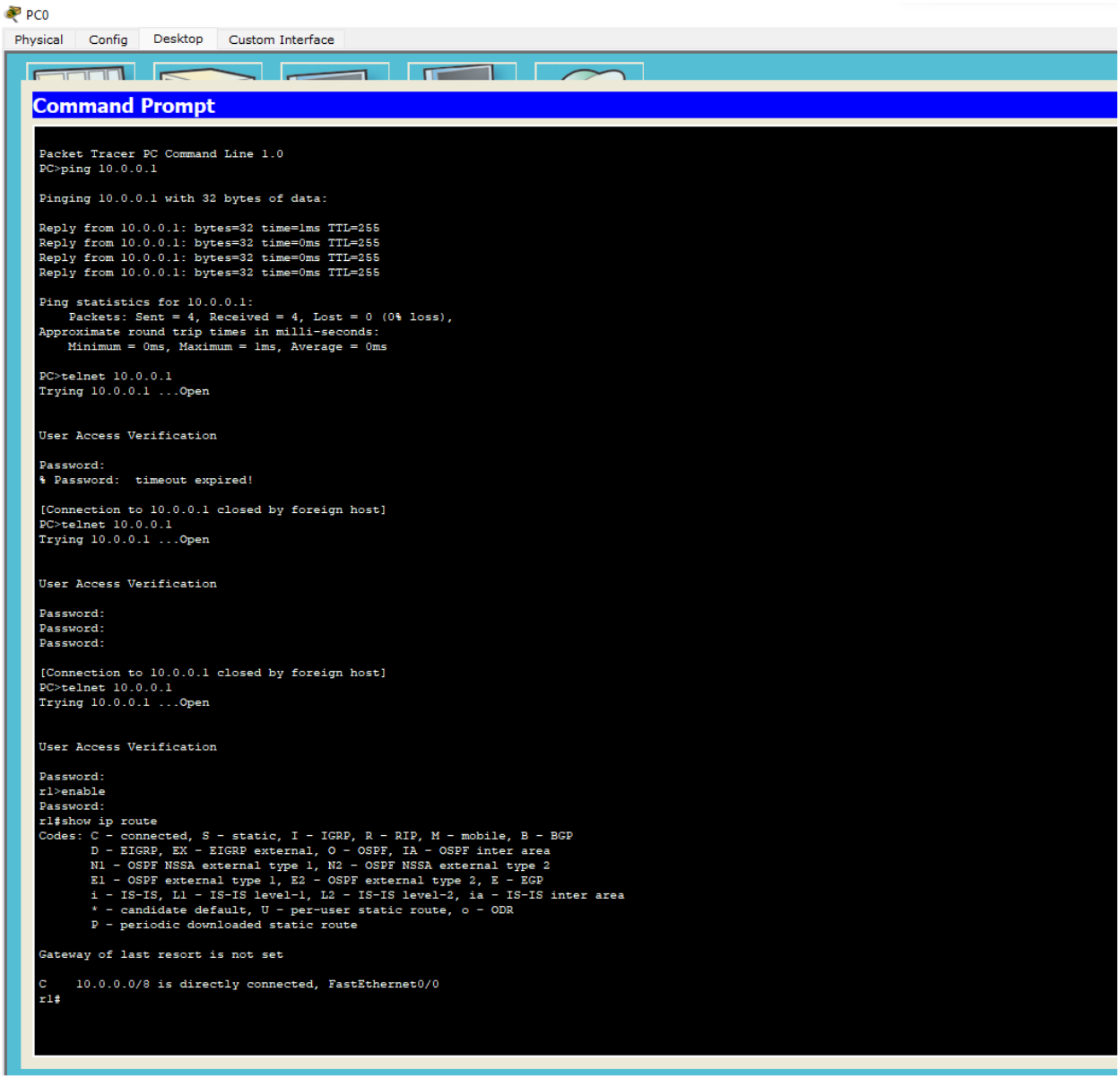
0.0.0.

tion is P0.

TOPOLOGY:



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:
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).

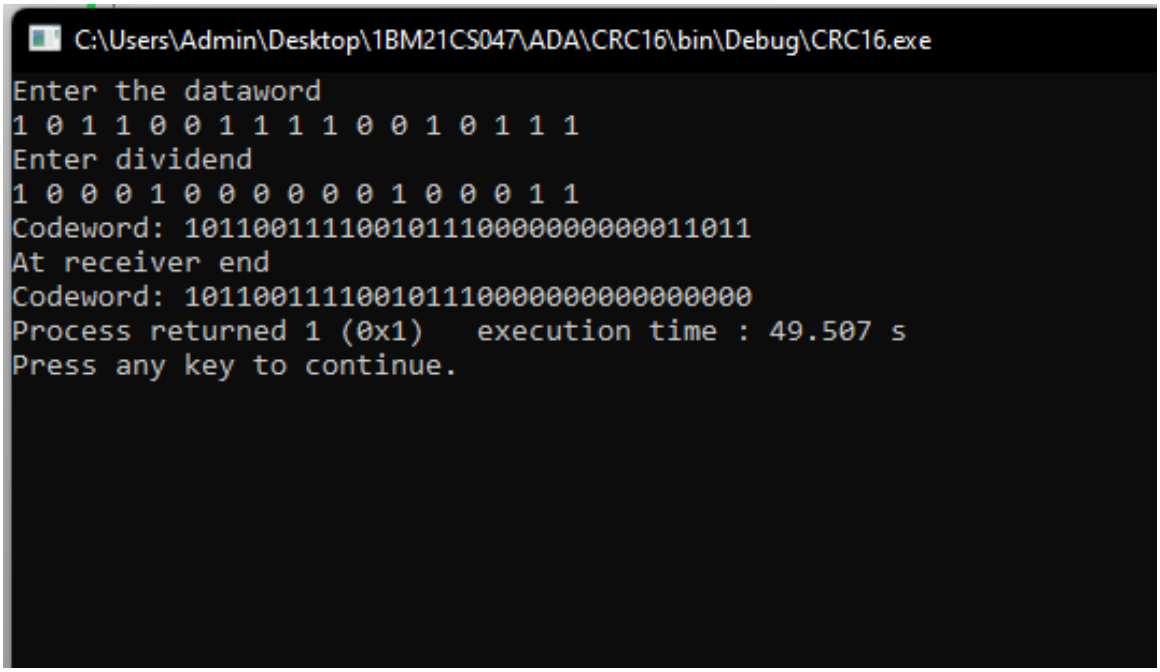
CODE:

```
#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");
```

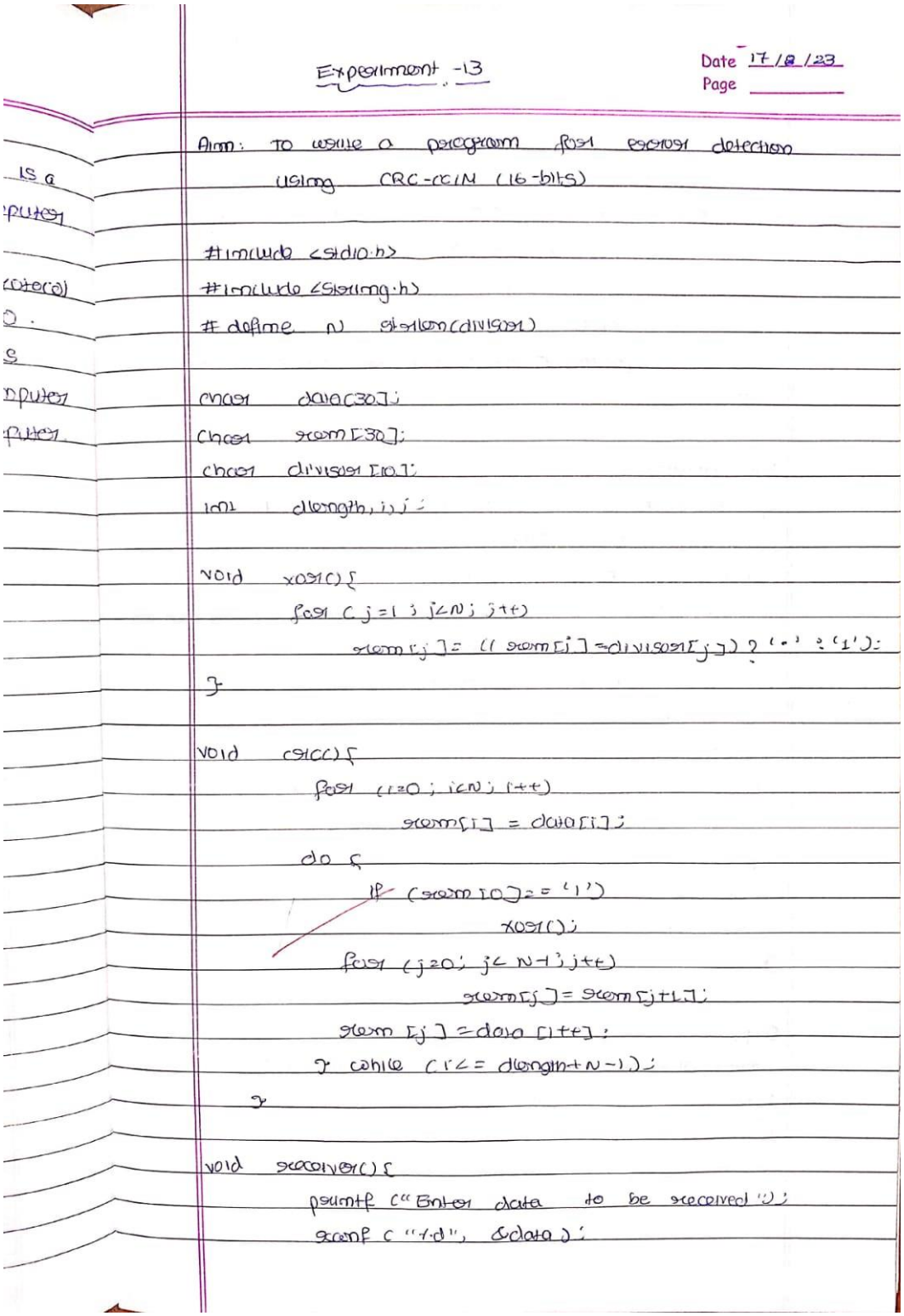
```
for(i=0;i<17;i++)
   scanf("%d",&dd[i]);

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:



OBSERVATION:



Experiment -13

Date 17/8/23
Page

Aim: To write a program for error detection using CRC-CCITT (16-bits)

```
#include <stdio.h>
#include <string.h>
#define N strlen(divisor)

char data[30];
char rem[30];
char divisor[10];
int length, i, j;

void xor() {
    for (j=1; j<N; j++)
        rem[j] = ((rem[j]==divisor[j]) ? '0' : '1');
}

void crc() {
    for (i=0; i<N; i++)
        rem[i] = data[i];
    do {
        if (rem[0]=='1')
            xor();
        for (j=0; j<N-1; j++)
            rem[j] = rem[j+1];
        rem[j] = data[i++];
    } while (i<=length+N-1);
}

void receiver() {
    printf("Enter data to be received: ");
    scanf("%d", &data);
```

```
printf("Data received : %s", data);
crc();
for(i=0; i<N-1 && rem[i] != '1'; i++);
    if (i<N-1)
        printf("Error in transmission");
    else
        printf("No error in transmission");
}

int main(){
printf("Enter data to be transmitted");
scanf("%s", data);
printf("Enter divisor");
scanf("%s", divisor);
dlength = strlen(data);
for(i=dlength; i<dlength+N-1; i++)
    data[i] = '0';
crc();
for(i=dlength; i<dlength+N-1; i++)
    data[i] = rem[i-dlength];
printf("Data being sent : %s", data);
receiver();
return 0;
}
```

Output :

Enter data to be transmitted : 1100101011001001

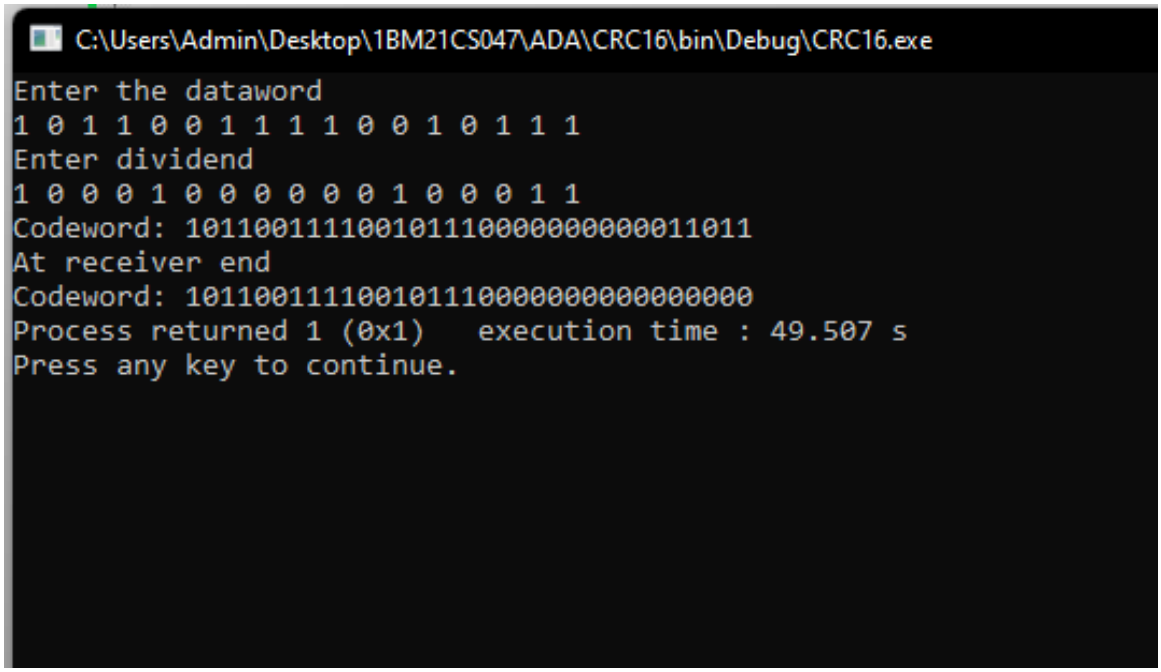
Enter divisor : 1000100000010001

Data being sent : 1110010010111001001

Data received : 1110010010111001001

No error in transmission.

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: 10110011110010111000000000011011
At receiver end
Codeword: 10110011110010111000000000000000
Process returned 1 (0x1)   execution time : 49.507 s
Press any key to continue.

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);
```

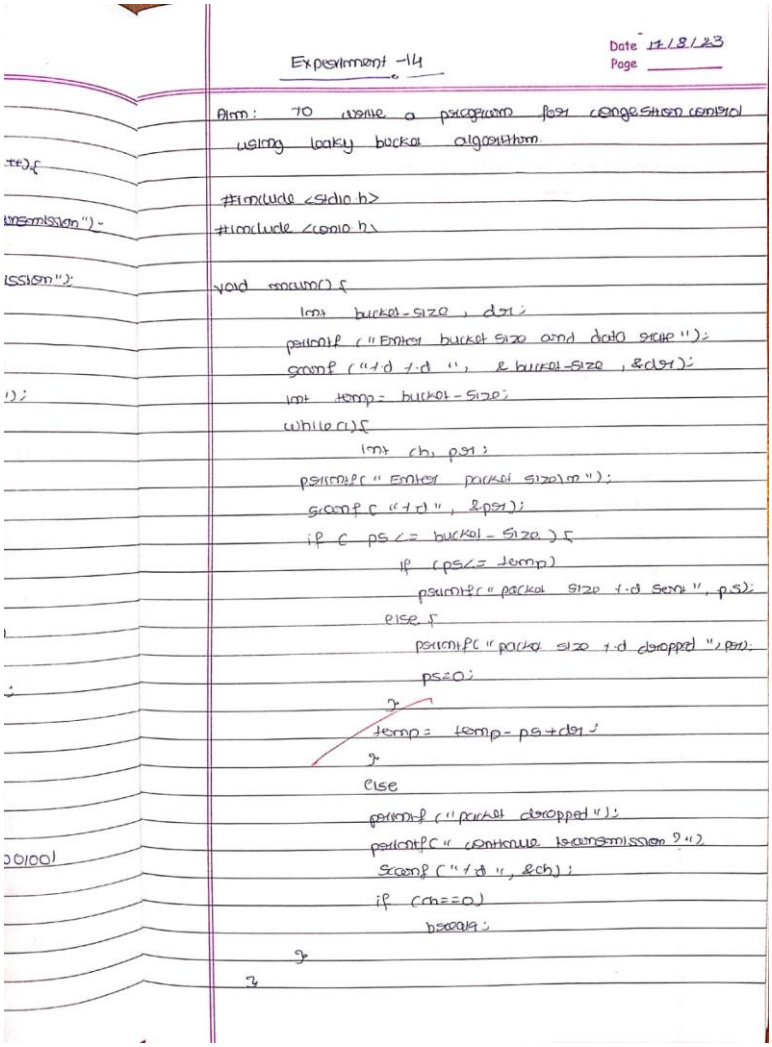
```
    }
    while (remaining < buckets) // Fixed the condition
    {
        if (buckets - remaining > outlets)
        {
            remaining += outlets; // Fixed the calculation
        }
        else
            remaining = buckets;
        printf("Remaining bytes: %d \n", remaining);
    }
    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:



Experiment -14

Date 12/8/23
Page

Aim: To write a program for congestion control using leaky bucket algorithm

```
#include <stdio.h>
#include <conio.h>

void main()
{
    int bucket_size, drp;
    printf("Enter bucket size and drop rate");
    scanf("%d %d", &bucket_size, &drp);
    int temp = bucket_size;
    while(1){
        int ch, ps;
        printf("Enter packet size\n");
        scanf("%d", &ps);
        if(ps <= bucket_size){
            if(ps <= temp)
                printf("packet size %d sent", ps);
            else{
                printf("packet size %d dropped", ps);
                ps=0;
            }
            temp = temp - ps + drp;
        }
        else
        printf("packet dropped");
        printf("continue transmission?\n");
        scanf("%d", &ch);
        if(ch==0)
            break;
    }
}
```

Output :

Enter bucksize and data erate
4000 250
Enter packet size
5000
packet dropped
continue transmission?
1
Enter packet size
1000
packet size 1000 sent
continue transmission?
1
Enter packet size
3000
packet size 3000 sent
continue transmission?
1
Enter packet size
750
packet size 750 dropped
continue transmission?
0

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

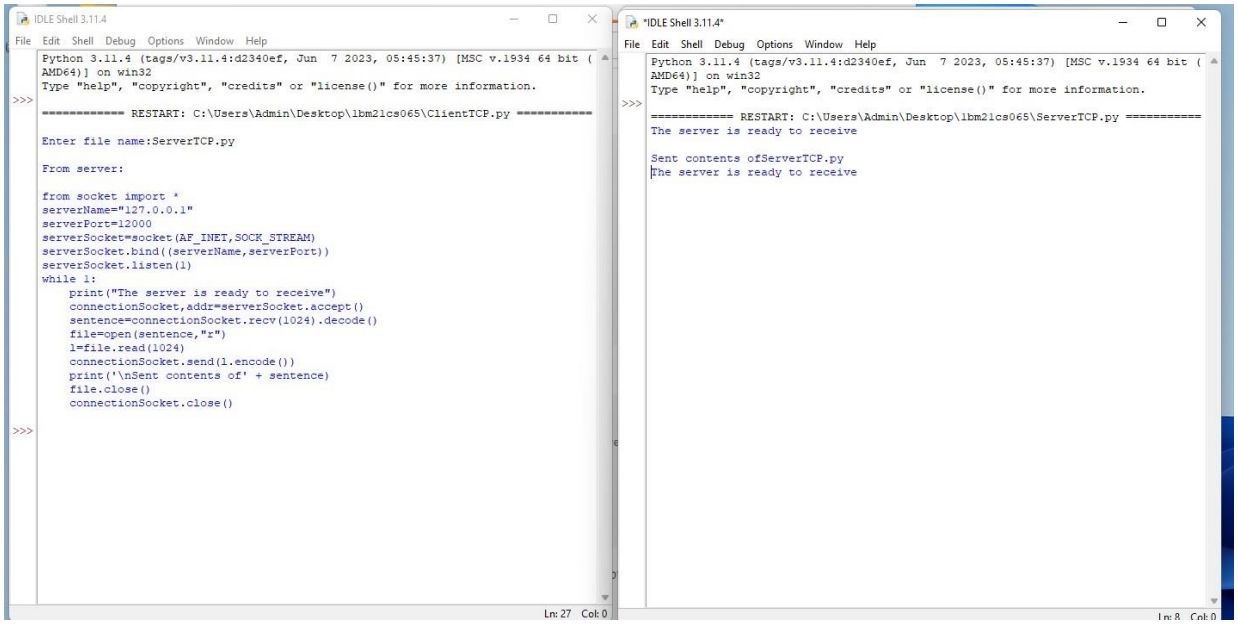
```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input("\nEnter file name: ")
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print ("\nFrom Server:\n")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

ServerTCP.py

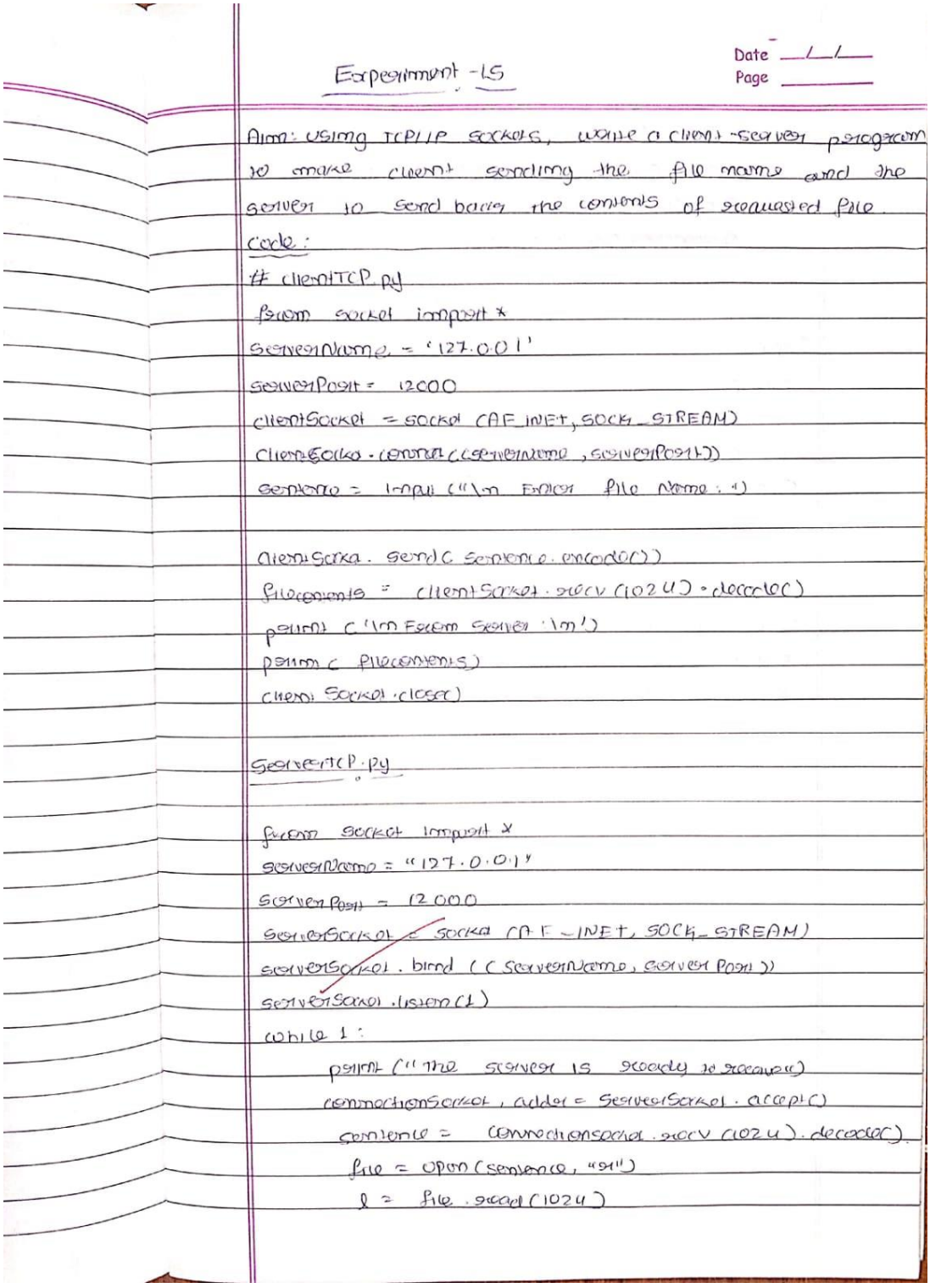
```
from socket import *
serverName="127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
print ("The server is ready to receive")
connectionSocket, addr = serverSocket.accept()
sentence = connectionSocket.recv(1024).decode()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

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

OUTPUT:



OBSERVATION:



Experiment - 15

Date __/__/__
Page _____

Aim: 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 requested file.

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("\n Enter file Name: ")

clientSocket.send(sentence.encode())
fileContents = clientSocket.recv(1024).decode()
print("\n From 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('Sent contents of ' + sentence)
file.close()
connectionSocket.close()
```

Output:

server.py :

```
The server is ready to receive
Sent contents of server.py
```

client.py

```
Enter file name: server.py
From Server
# All contents of server.py are shown 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

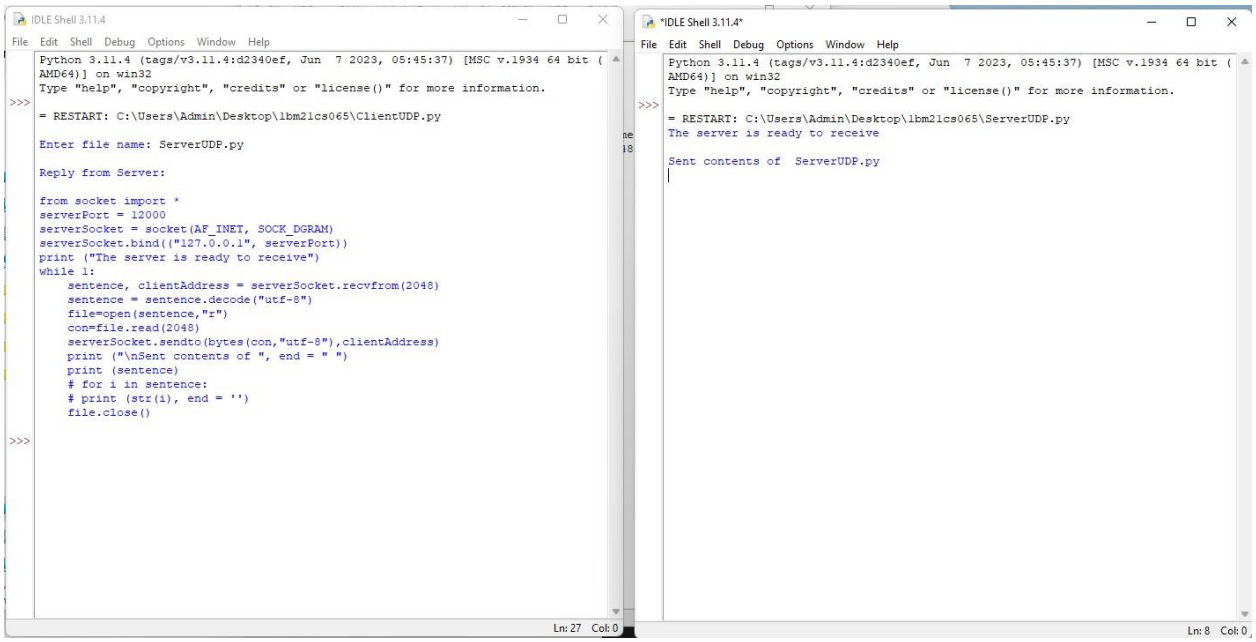
```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("\nEnter file name: ")
clientSocket.sendto(bytes(sentence,"utf-8"),(serverName, serverPort))
filecontents,serverAddress = clientSocket.recvfrom(2048)
print ("\nReply from Server:\n")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

ServerUDP.py

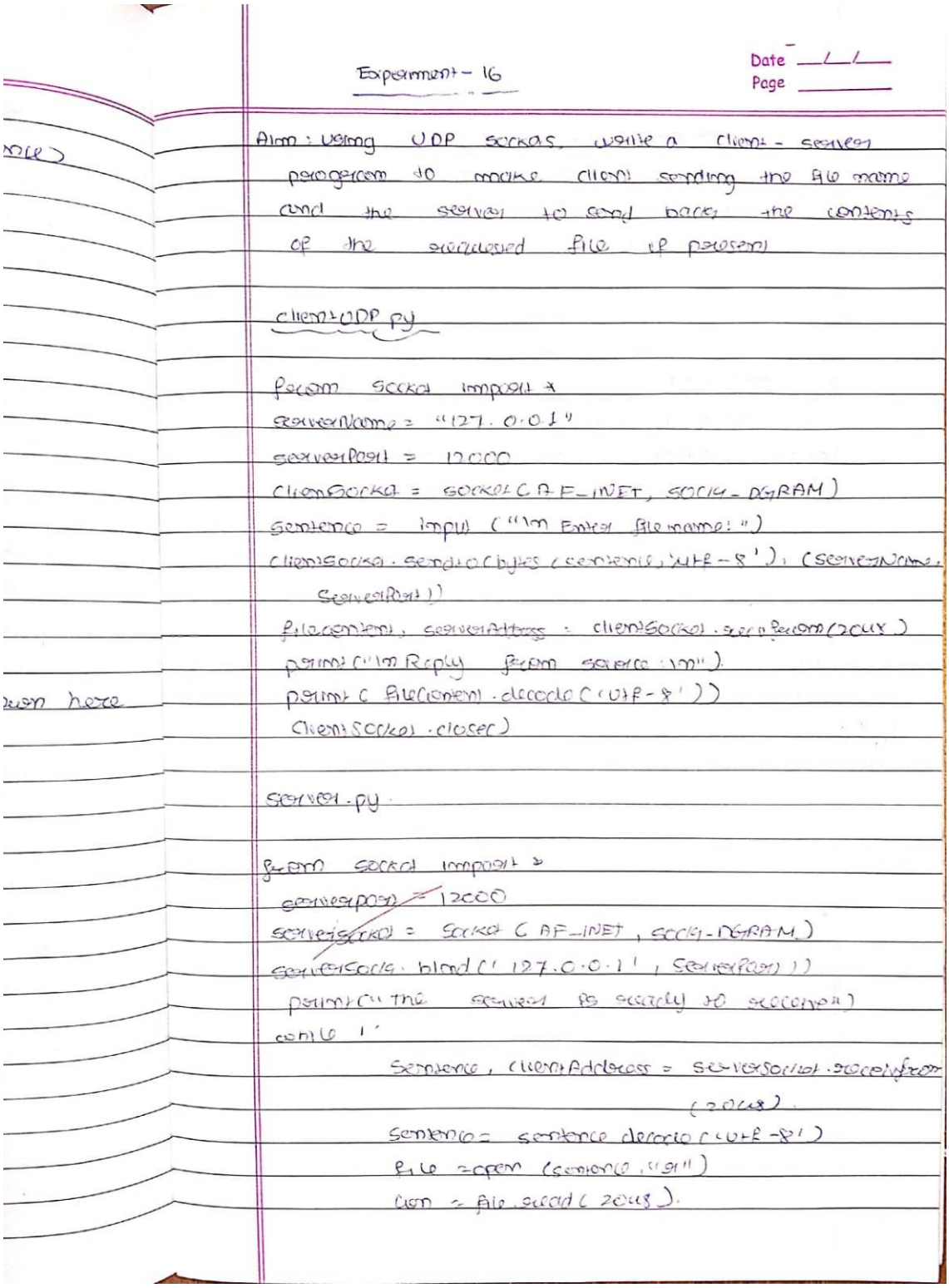
```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 1:
sentence, clientAddress = serverSocket.recvfrom(2048)
sentence = sentence.decode("utf-8")
file=open(sentence,"r")
```

```
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:



Experiment - 16

Date __/__/__
Page _____

Aim : 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

client-UDP.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("\n Enter file name: ")
clientSocket.sendto(bytes(sentence,'utf-8'), (serverName,
    serverPort))
fileContent, serverAddress = clientSocket.recvfrom(2048)
print("\n Reply from server: \n")
print(fileContent.decode('utf-8'))
clientSocket.close()
```

server.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("\n Sent content of ", end=" ")
print(sentence)
file.close()
```

output:

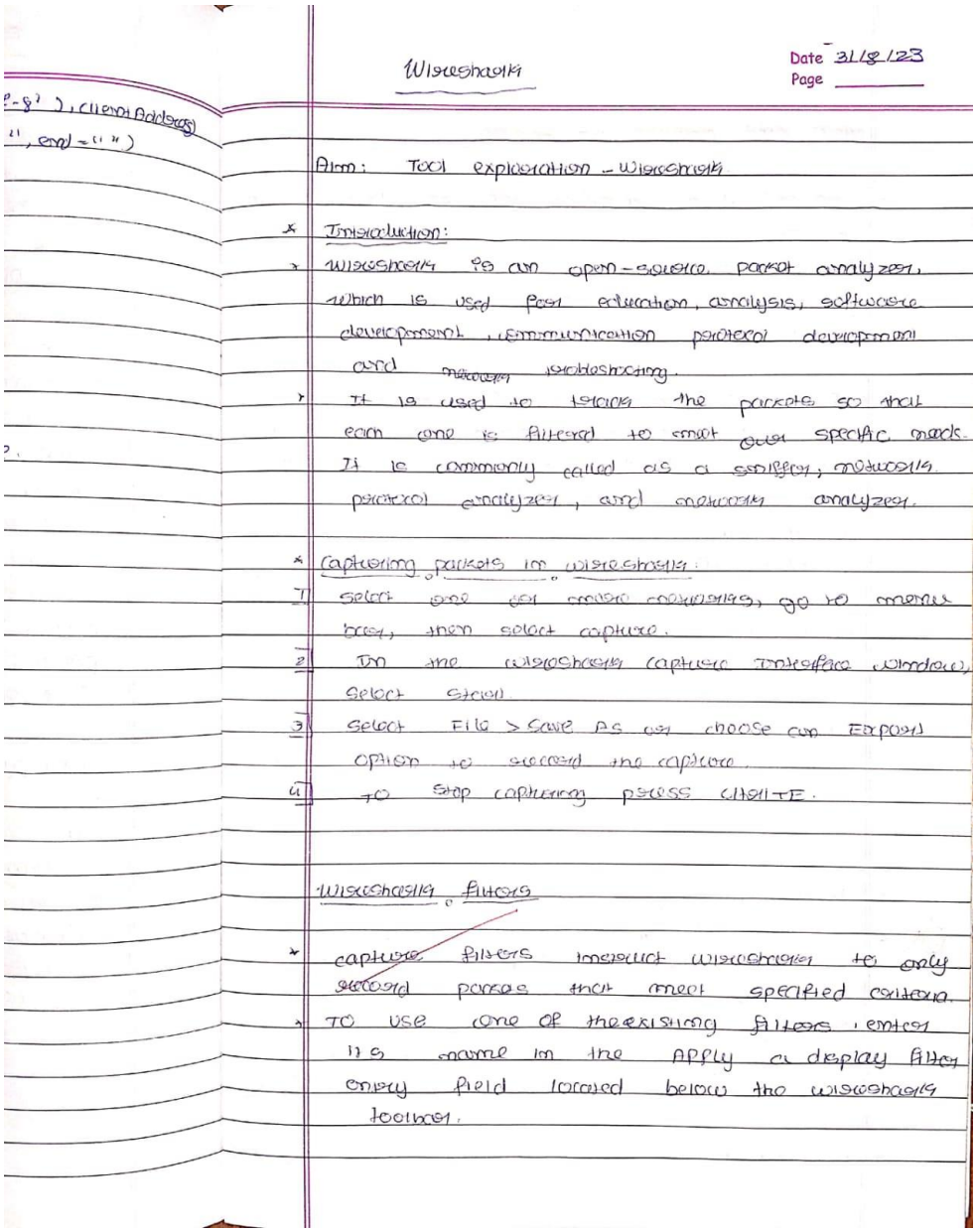
```
>>> Server.py
The Server is ready to receive
Sent content of ServerUDP.py
The Server is ready to receive.

>>> client.py

Enter file name: ServerUDP.py
Reply from server:
" Content of ServerUDP.py
```

31/1/23

WEEK 17

Tool Exploration -Wireshark

OBSERVATION:



Wireshark

Date 31/8/23
Page

Aim: Tool exploration - Wireshark

Introduction:

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

Capturing packets in wireshark:

1. Select and for more interfaces, go to menu bar, then select capture.
2. In the wireshark capture interface window, select start.
3. Select File > Save As or choose an Export option to record the capture.
4. To stop capturing press CTRL+E.

Wireshark filters

- Capture filters instruct wireshark to only record packets that meet specified criteria.
- To use one of the existing filters, enter its name in the Apply a display filter entry field located below the wireshark toolbar.

View and analyze the packets

- Captured data interface contains three main sections:
  1] The packet list pane (the top section)
  2] the packet details pane (middle section)
  3] The packet bytes pane (bottom section)

* Packet list pane shows all packets found in active capture file. Each packet has its own row and a corresponding number assigned to it. Each packet contains
  - Timestamp
  - Source IP
  - Destination IP
  - Protocol
  - Length

- The details pane, presents protocols and protocol fields of the selected packet in a collapsible format, which can be expanded on click.

* Packet bytes pane is present at the bottom of the bytes pane, which displays the raw data of the selected packet in hexadecimal bytes.
- Selecting a specific position of this data automatically highlights its corresponding section in the packet details and vice versa.
- Any bytes that cannot be parsed are represented