VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

COMPUTER NETWORKS

Submitted by

KAMESH CHANDRA(1BM21CS271)

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

JUNE-2023 to SEPTEMBER-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 "LAB COURSE **COMPUTER NETWORKS**" carried out by **KAMESH CHANDRA(1BM21CS271),** 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 year2023. 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.

Sowmya T
Assistant Professor
Department of CSE
BMSCE, Bengaluru
`

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

Sl. No.	Experiment Title
1.	Create a topology and simulate sending a simple PDU from source to destination using hub and switch asconnecting devices and demonstrate ping message.
2	Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destinationunreachable, request timed out, reply
3	Configure default route, static route to the Router
4	Configure DHCP within a LAN and outside LAN. ,.
5	Configure Web Server, DNS within a LAN.
6	Configure RIP routing Protocol in Routers.
7	Configure OSPF routing protocol
8	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)
9	Demonstrate the TTL/ Life of a Packet
10	To understand the operation of TELNET by accessing therouter in server room from a PC in IT office.
11	To construct a VLAN and make the PC's communicate among a VLAN.
12	To construct a WLAN and make the nodes communicate wirelessly.
	Cycle - 2
13	Write a program for error detecting code using CRC-CCITT(16-bits).
14	Write a program for congestion control using Leakybucket algorithm.
15	Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to sendback the contents of the requested file if present.
16	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.

CN LAB 1

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

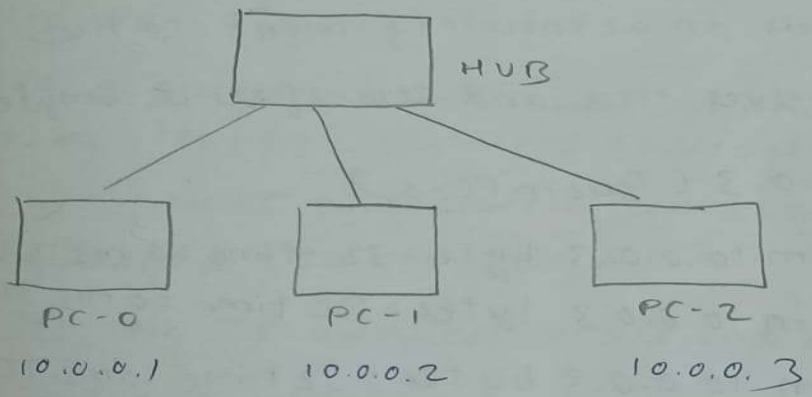
OBSERVATION

Lab-1

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

Aim: To understand working of hub and switch

Topology: (Hub)



Procedure:

1) Add a generic hub and PC's to it using copper straight through cable from fast ethernet interface of PC to hub.
2) Set the IP address of all PC's
3) Send a simple PDU message from PC-0 to PC-2 in the simulation mode.

a) In real-time mode click on PC0 and under desktop open a command prompt and ping another PC as

ex: Ping 10.0.0.3

Result:

1) The simple PDU is sent from PC0 to hub.

2) Hub sends PDU to all ports except the incoming port. The PDU is rejected by PC1

3) PC2 sends an acknowledgement to hub

4) PC0 receives this and transfer is complete.

Ping 10.0.0.3 (From PC-0)

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

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

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

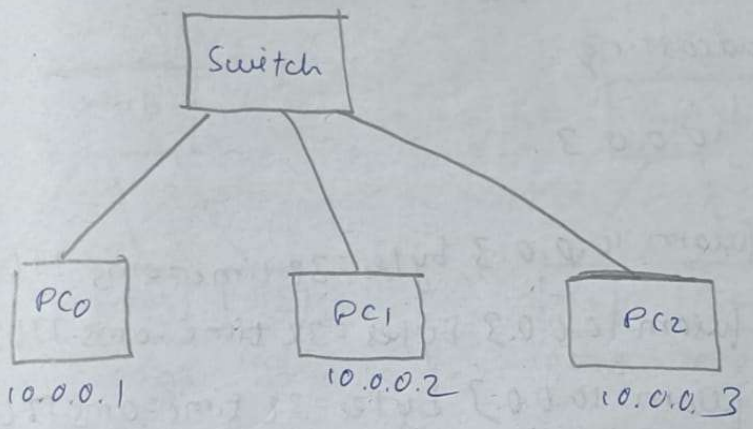
Reply from 10.0.0.3 bytes=32 time=0ms TTL=128

Ping statistics for 10.0.0.3

Packets: sent=4, received=4, lost=0

Approximate round trip time in milli-seconds minimum=0ms, maximum=0ms, Average=0ms

Topology (Switch)



Procedure:

1) Add a generic switch and connect 3 PC's to it using copper-straight through wire.
2) wait for the PC and switch connections to be established (30 seconds).
3) Set the IP address of all end devices.
4) Send a simple PDU request from PC0 to PC2 in simulation mode. In real time mode ping PC2 from PC0 i.e., Desktop > command Prompt.

Result:

1) The simple PDU is sent from PC0 to switch
2) The switch broadcasts the PDU to all output ports except input port.

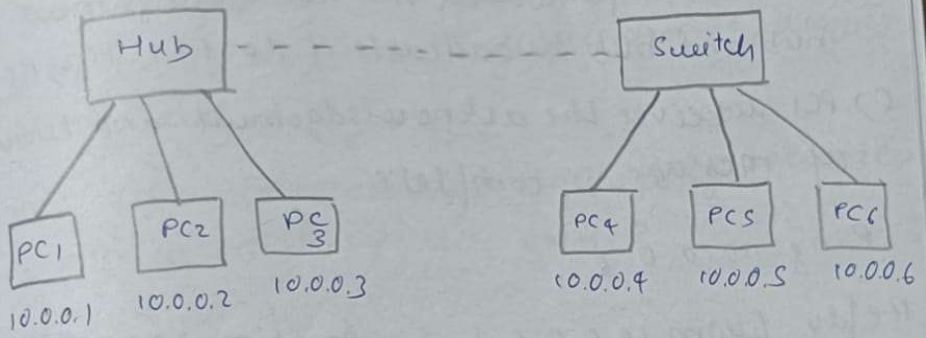
3) PC2 acknowledges and sends a packet to switch. the switch sends his to PC0 without broadcasting

Ping 10.0.0.3

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=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
Approximate round trip time (milli-seconds)
minimum=0ms, maximum=0ms, average=0ms

Topology (Hybrid)



Procedure :

1) Add a generic hub and a generic switch and connect 3 end-devices to each of them using copper-straight through wires
2) wait for switch-PC connection to be established
3) Connect the hub and switch using copper-cross-over wires
4) Set the IP address of the end devices.
5) Send simple PDU from PC1 TO PC6 and wait for simulation to finish
6) In real time mode Ping PC6 from PC1

Result :

1) Simple PDU is sent from PC1 to hub.
2) The hub sends copy of PDU to PC2, PC3 and switch.
3) The switch forwards the message to PC4 PC5 PC6

4) PC6 receives the message and sends an acknowledgement to switch.

5) The switch forwards the acknowledgement to hub and hub broadcasts it to PC1, PC2, PC3

6) PC1 receives the acknowledgement and transfer of message is complete.

Ping 10.0.0.6

Reply from 10.0.0.6 byte = 32 time = 0ms TTL = 128

Reply from 10.0.0.6 byte = 32 time = 0ms TTL = 128

Reply from 10.0.0.6 byte = 32 time = 0ms TTL = 128

Reply from 10.0.0.6 byte = 32 time = 0ms TTL = 128

Ping statistics for 10.0.0.6

Packets sent = 4, received = 4, loss = 0

Observations :-

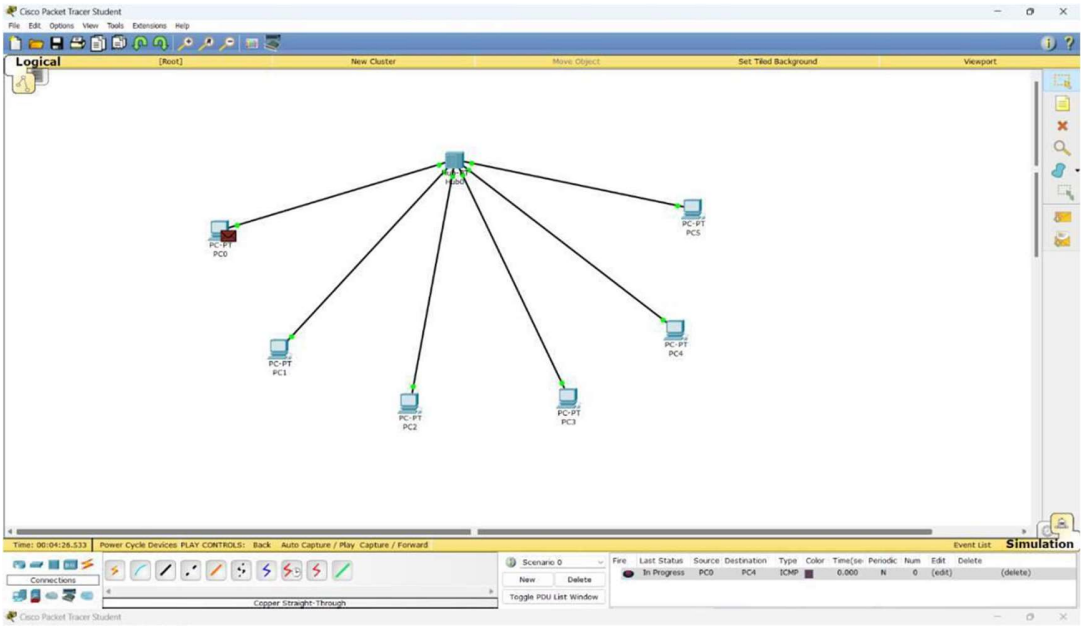
1) Hub doesn't store any data and all incoming packets are broadcasted to all ports except the input port. The receiver should acknowledge the packet when received.

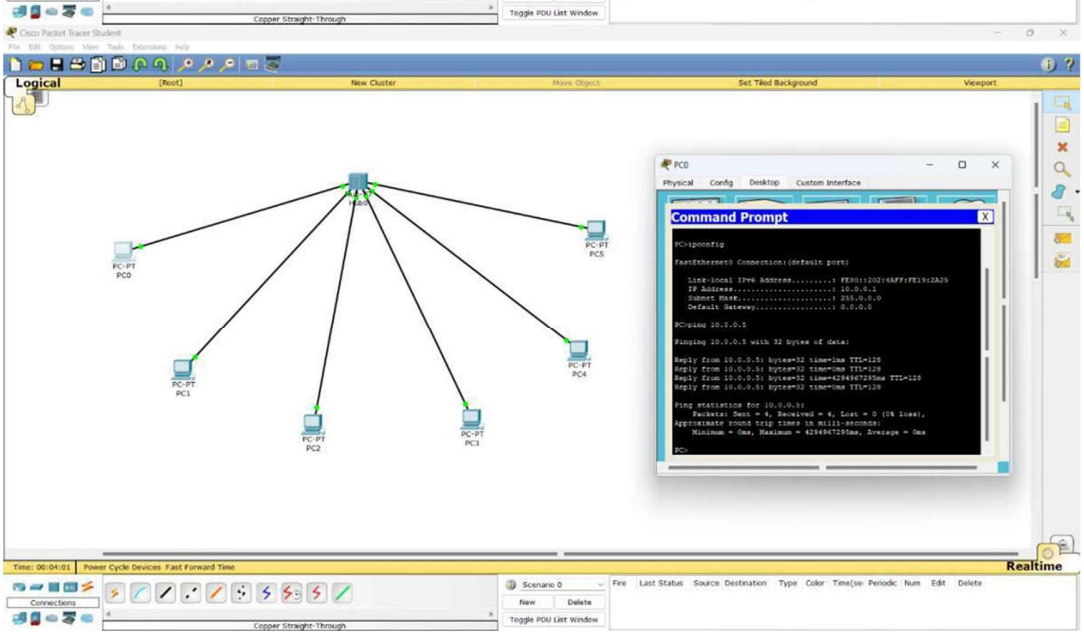
2) Switch takes time for new connection to be established. In the first transfer switch broadcasts the packet to all the end devices. In the following transfers switch transfers packets to destination end devices based on mac address.

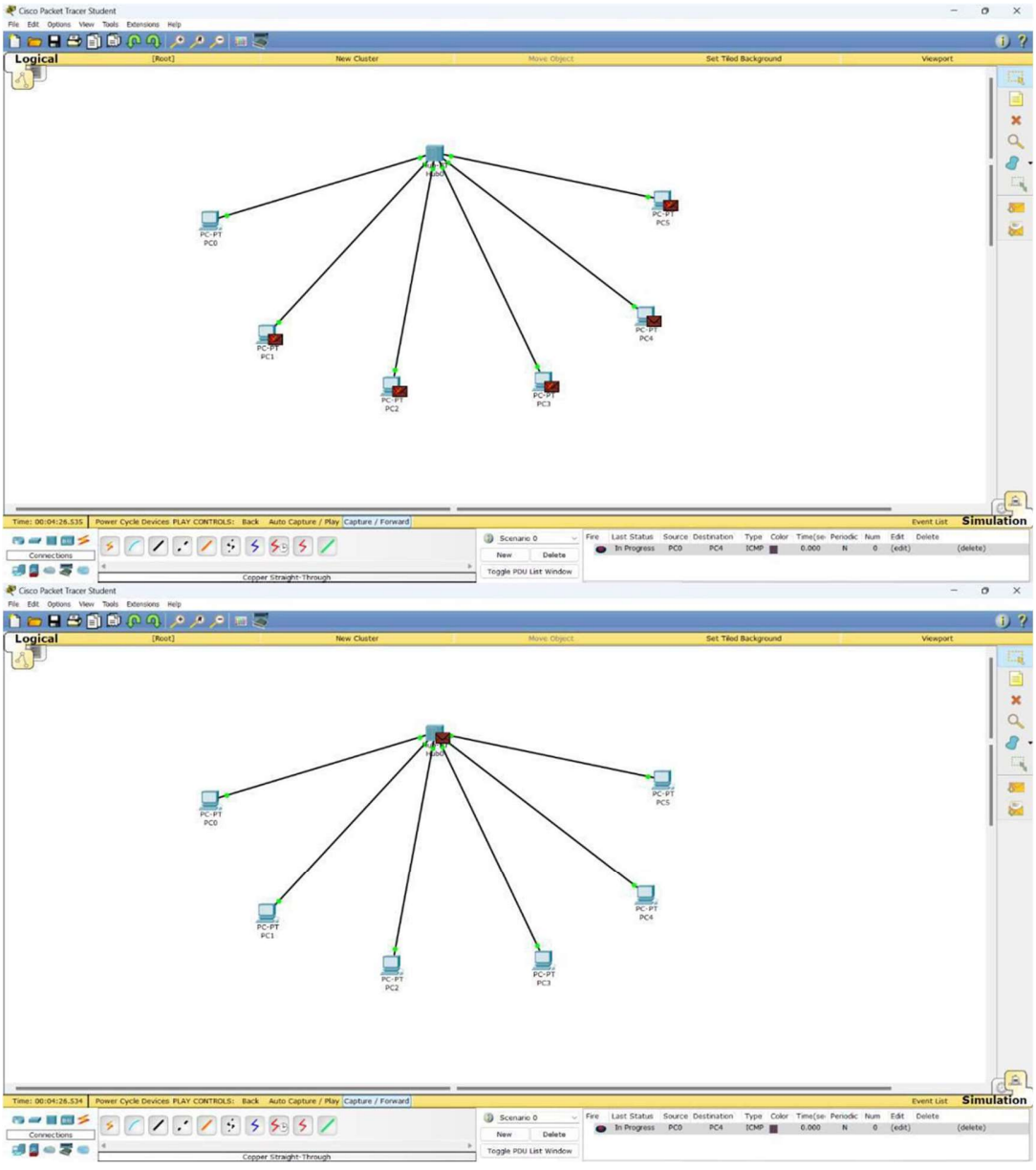
16/6/23

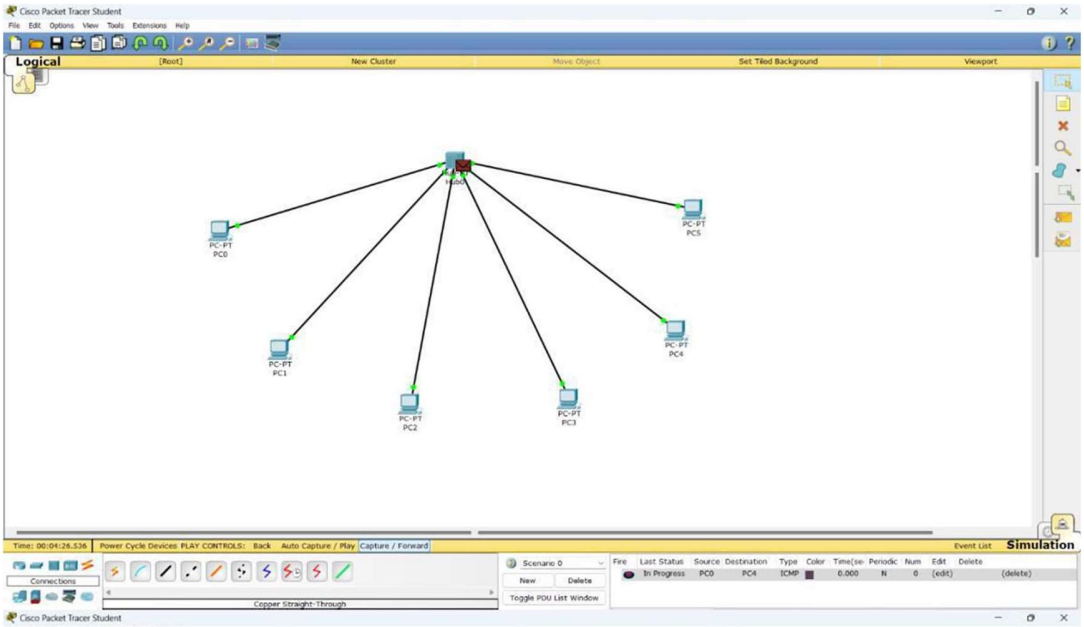
SCREENSHOTS

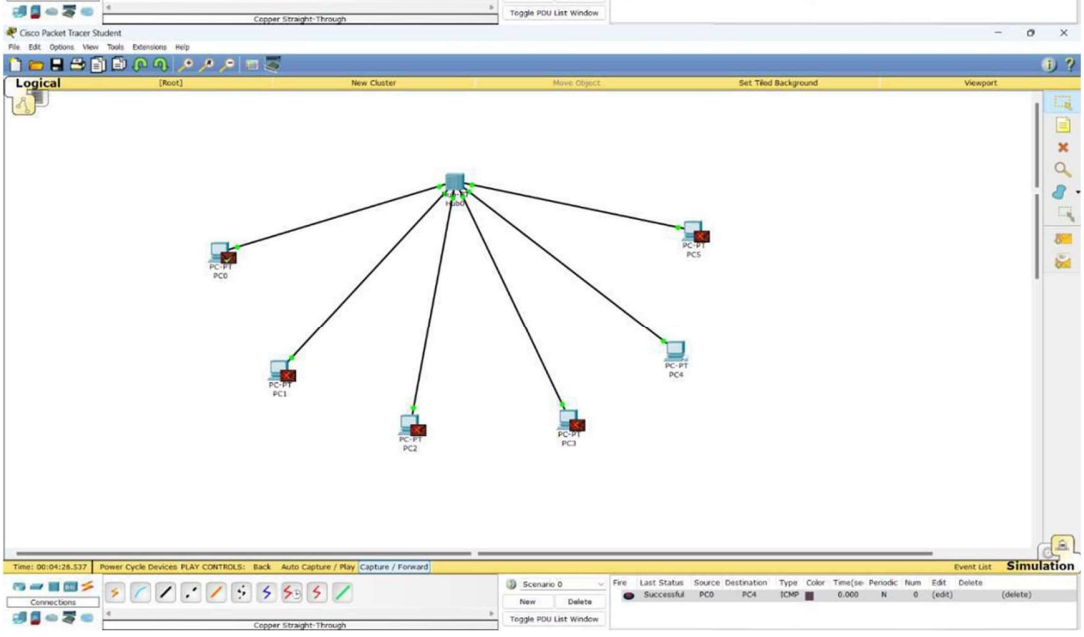
USING HUB:



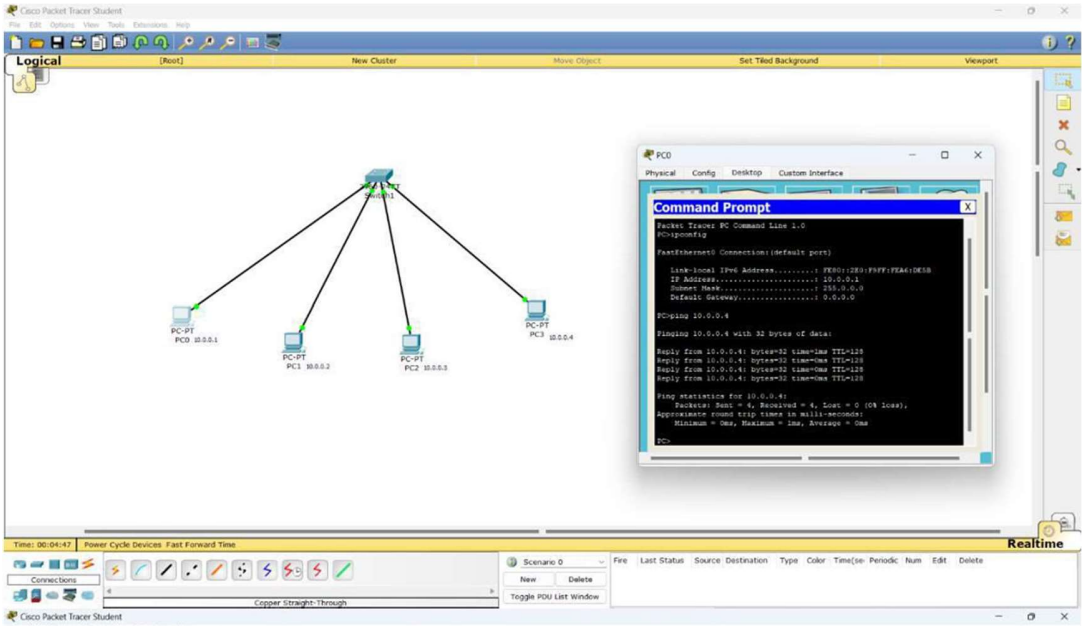


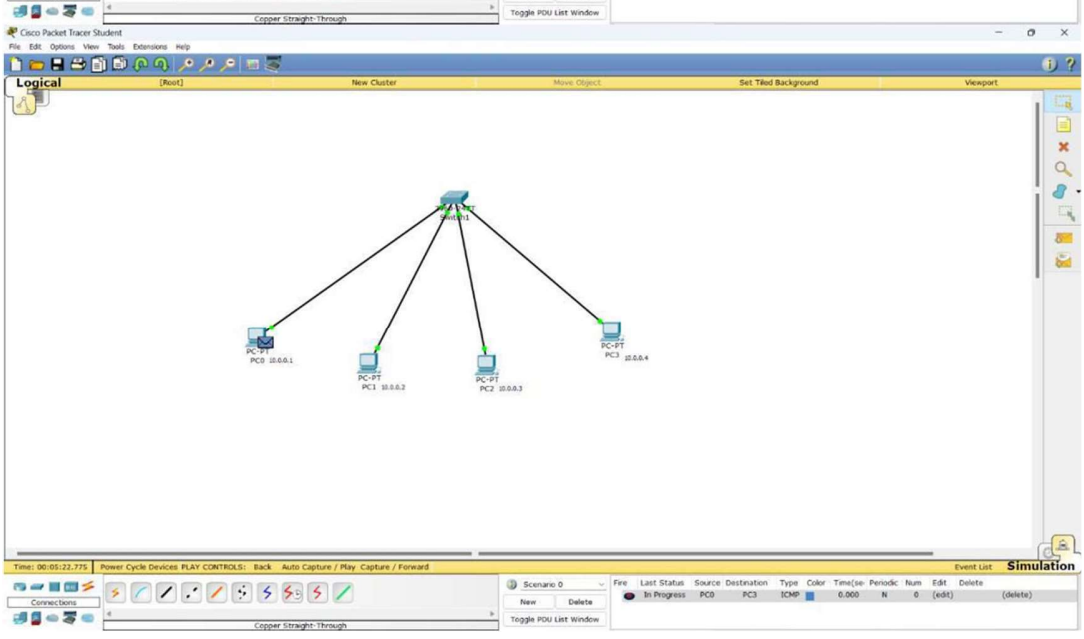

Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Hub0
PC-PT PC0
PC-PT PC1
PC-PT PC2
PC-PT PC3
PC-PT PC4
PC-PT PC5
Time: 00:04:26.535
Power Cycle Devices PLAY CONTROLS: Back Auto Capture / Play Capture / Forward
Event List
Simulation
Connections
Copper Straight-Through
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(se- Periodic Num Edit Delete
In Progress PC0 PC4 ICMP 0.000 N 0 (edit) (delete)
Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Hub0
PC-PT PC0
PC-PT PC1
PC-PT PC2
PC-PT PC3
PC-PT PC4
PC-PT PC5
Time: 00:04:26.534
Power Cycle Devices PLAY CONTROLS: Back Auto Capture / Play Capture / Forward
Event List
Simulation
Connections
Copper Straight-Through
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(se- Periodic Num Edit Delete
In Progress PC0 PC4 ICMP 0.000 N 0 (edit) (delete)

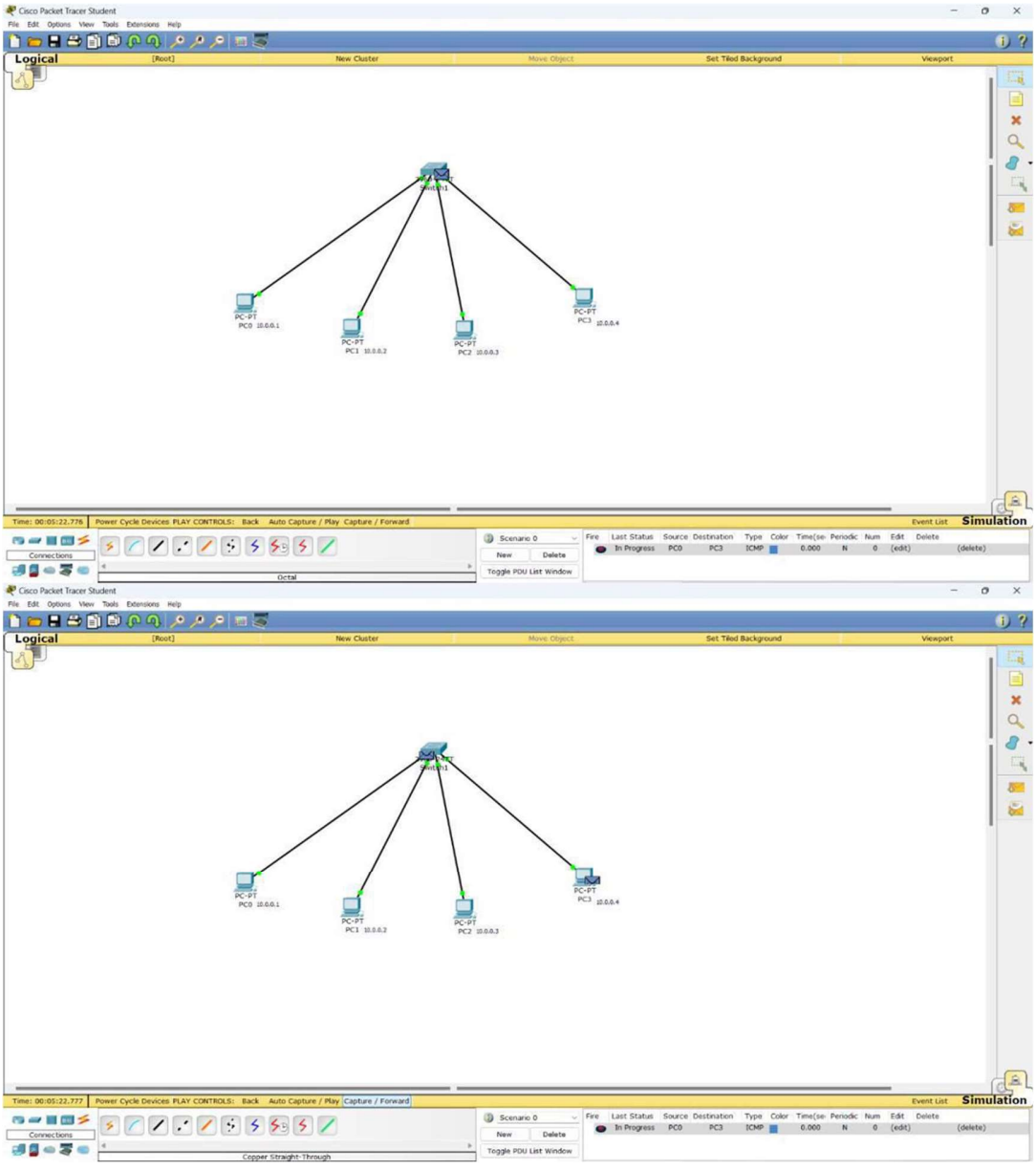

Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Hub0
PC-PT PC0
PC-PT PC1
PC-PT PC2
PC-PT PC3
PC-PT PC4
PC-PT PC5
Time: 00:04:26.536
Power Cycle Devices PLAY CONTROLS: Back Auto Capture / Play Capture / Forward
Event List
Simulation
Connections
Copper Straight-Through
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(se- Periodic Num Edit Delete
In Progress PC0 PC4 ICMP 0.000 N 0 (edit) (delete)

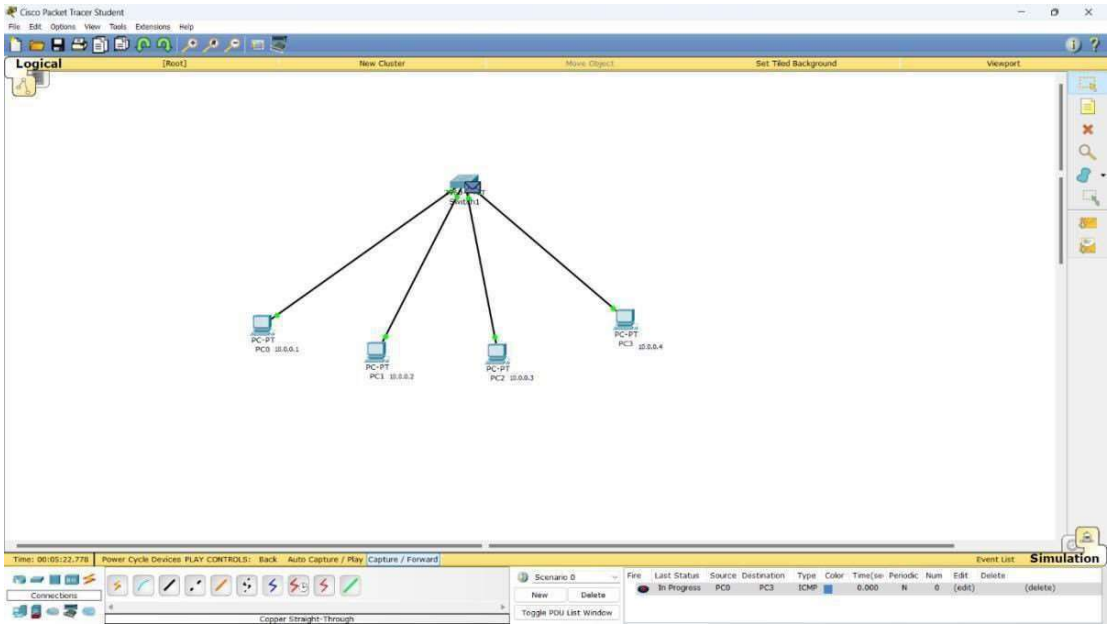

Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Hub0
PC-PT PC0
PC-PT PC1
PC-PT PC2
PC-PT PC3
PC-PT PC4
PC-PT PC5
Time: 00:04:26.537
Power Cycle Devices PLAY CONTROLS: Back Auto Capture / Play Capture / Forward
Event List
Simulation
Connections
Copper Straight-Through
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(se- Periodic Num Edit Delete
Successful PC0 PC4 ICMP 0.000 N 0 (edit) (delete)

USING SWITCH:

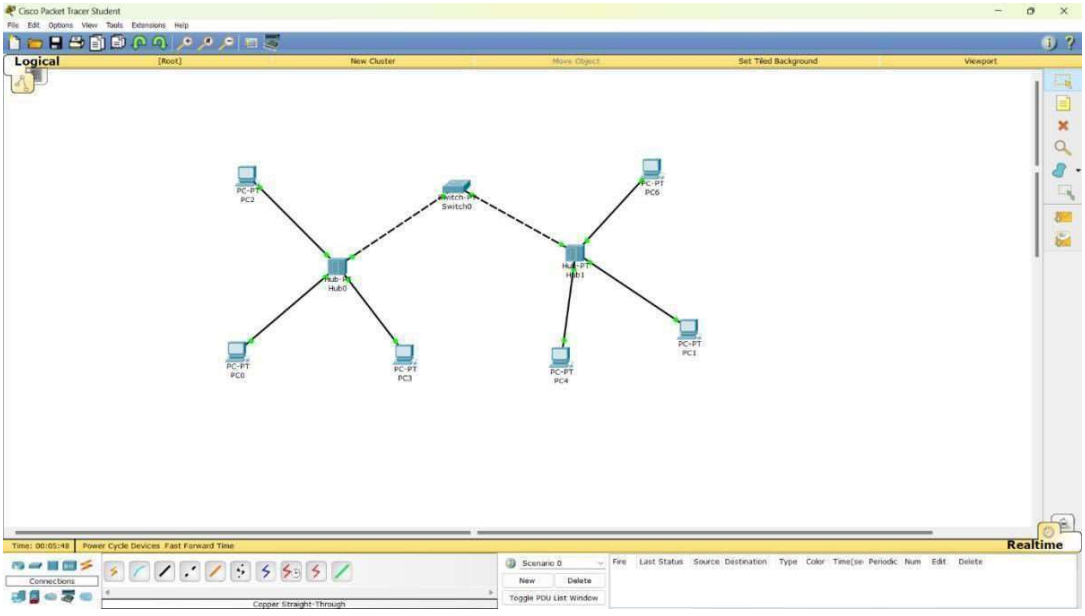


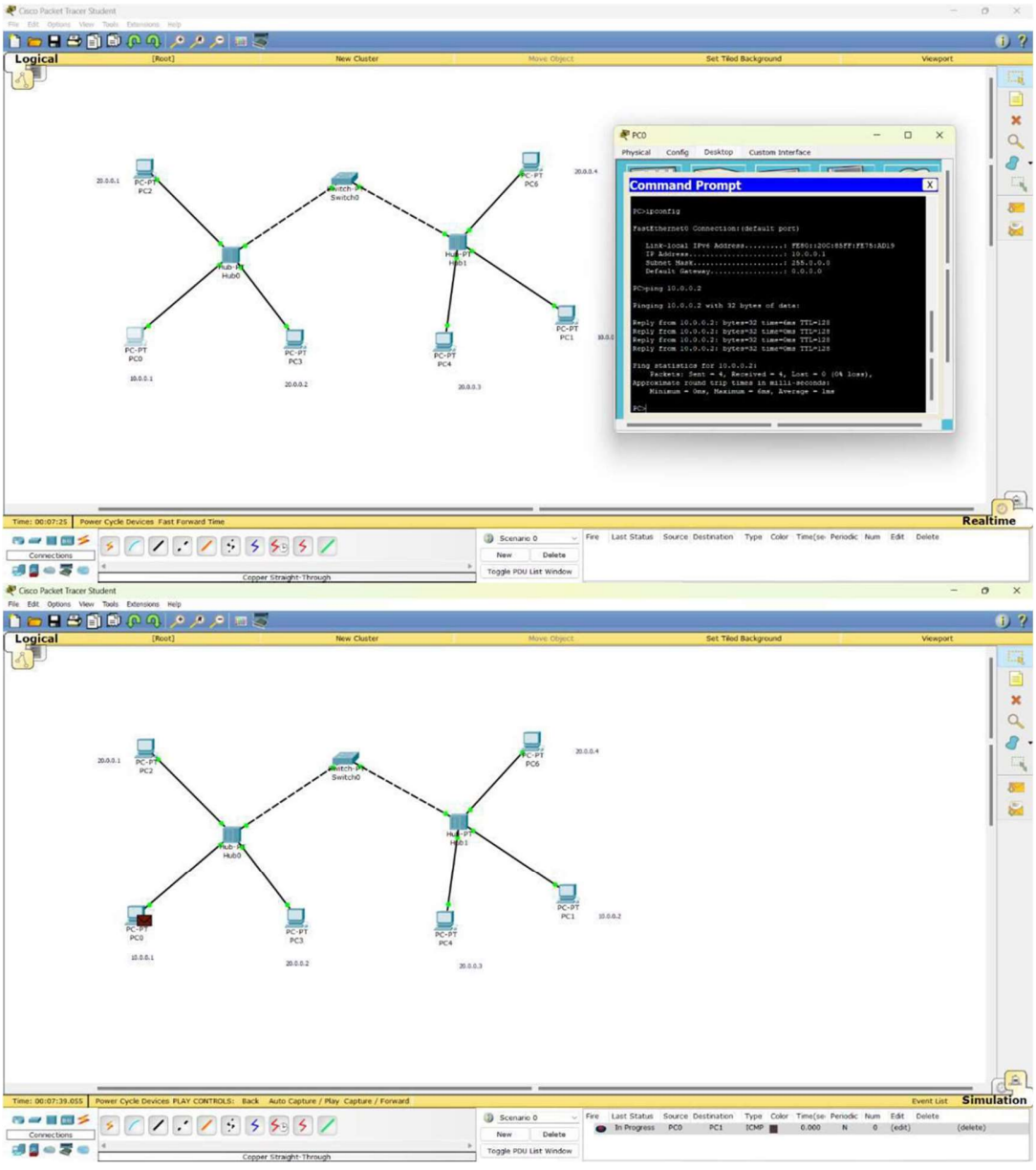


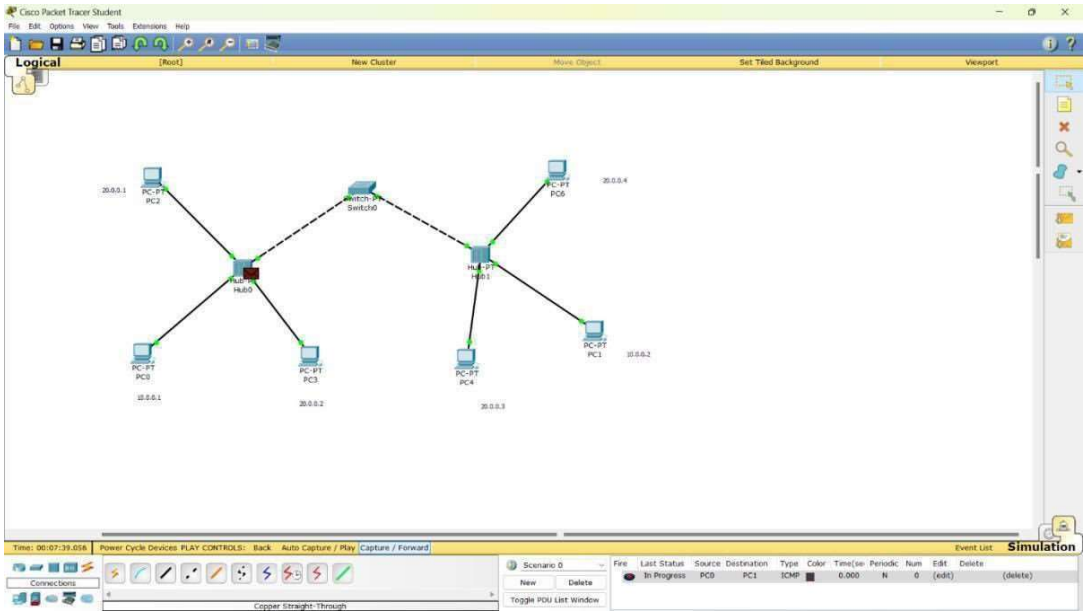

Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Switch1
PC-PT PC0 10.0.0.1
PC-PT PC1 10.0.0.2
PC-PT PC2 10.0.0.3
PC-PT PC3 10.0.0.4
Time: 00:05:22.776
Power Cycle Devices PLAY CONTROLS: Back Auto Capture / Play Capture / Forward
Event List
Simulation
Connections
Octal
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(se- Periodic Num Edit Delete
In Progress PC0 PC3 ICMP 0.000 N 0 (edit) (delete)
Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Switch1
PC-PT PC0 10.0.0.1
PC-PT PC1 10.0.0.2
PC-PT PC2 10.0.0.3
PC-PT PC3 10.0.0.4
Time: 00:05:22.777
Power Cycle Devices PLAY CONTROLS: Back Auto Capture / Play Capture / Forward
Event List
Simulation
Connections
Copper Straight-Through
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(se- Periodic Num Edit Delete
In Progress PC0 PC3 ICMP 0.000 N 0 (edit) (delete)



USING HUB AND SWITCH:



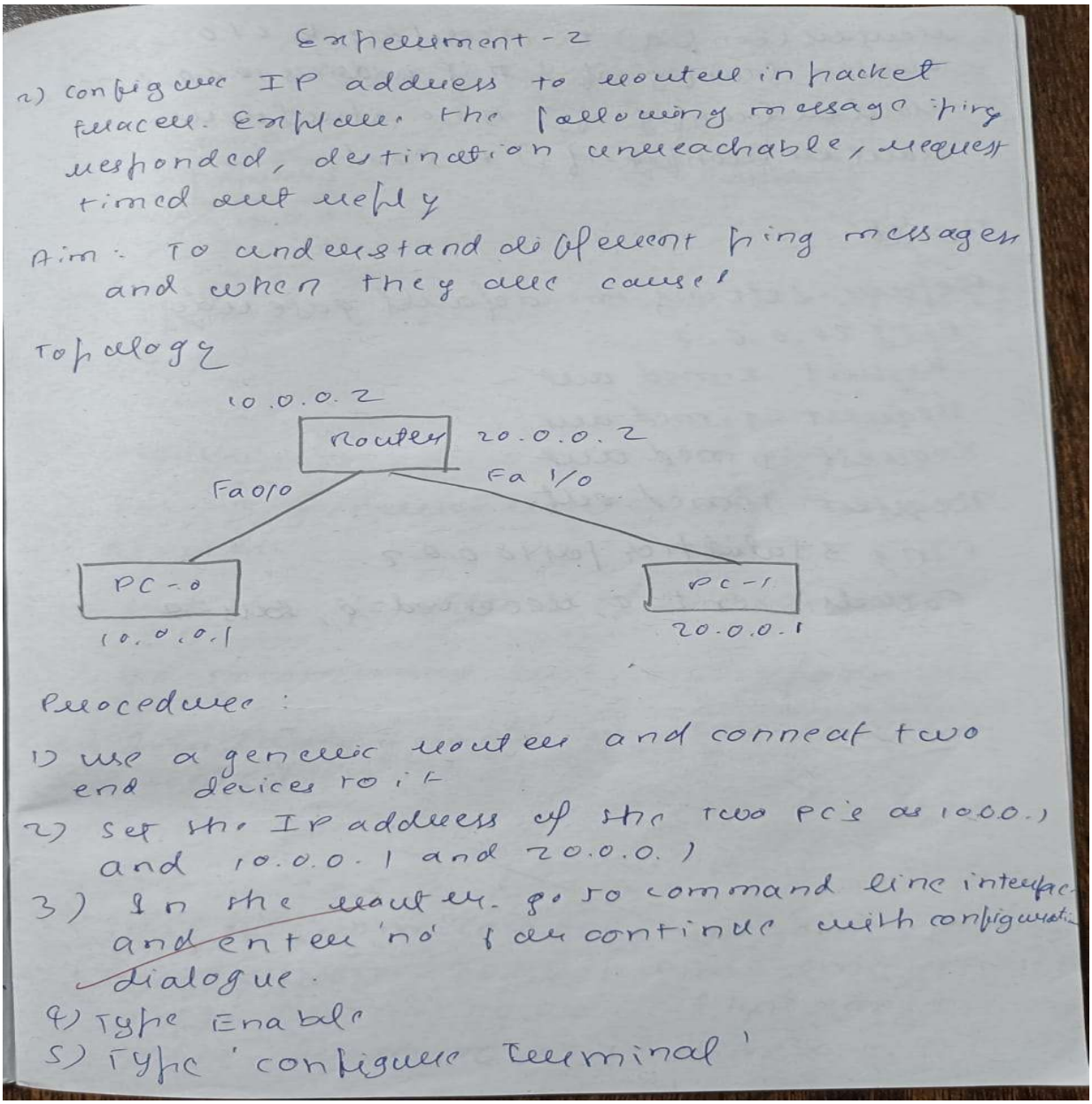

Cisco Packet Tracer Student
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
PC>ipconfig
FastEthernet0 Connection:(default port)
Link-local IPv6 Address.....: FE80::20C:85FF:FE75:AD19
IP Address.....: 10.0.0.1
Subnet Mask.....: 255.0.0.0
Default Gateway.....: 0.0.0.0
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=6ms 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=0ms 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 = 6ms, Average = 1ms
PC>
Switch0
Hub0
Hub1
PC0
PC1
PC2
PC3
PC4
PC6
Time: 00:07:25
Power Cycle Devices Fast Forward Time
Realtime
Connections
Copper Straight-Through
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time(se
Periodic
Num
Edit
Delete
Cisco Packet Tracer Student
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Switch0
Hub0
Hub1
PC0
PC1
PC2
PC3
PC4
PC6
Time: 00:07:39.055
Power Cycle Devices PLAY CONTROLS: Back Auto Capture / Play Capture / Forward
Event List
Simulation
Connections
Copper Straight-Through
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time(se
Periodic
Num
Edit
Delete
In Progress
PC0
PC1
ICMP
0.000
N
0
(edit)
(delete)


Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
PC-PT PC2
Switch-PT Switch0
PC-PT PC6
Hub-PT Hub0
Hub-PT Hub1
PC-PT PC3
PC-PT PC4
PC-PT PC1
Time: 00:07:39.056
Power Cycle Devices PLAY CONTROLS: Back Auto Capture / Play Capture / Forward
Event List
Simulation
Connections
Copper Straight-Through
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(se Periodic Num Edit Delete
In Progress PC0 PC1 ICMP 0.000 N 0 (edit) (delete)

CN LAB 2

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

OBSERVATION



router (config) # interface Fa 0/0
router (config-if) # ip address 10.0.0.2 255.0.0.0
router (config-if) # no shutdown
router (config-if) # exit

Result:

Before setting the default gateway

Ping 20.0.0.1

Request timed out

Request timed out

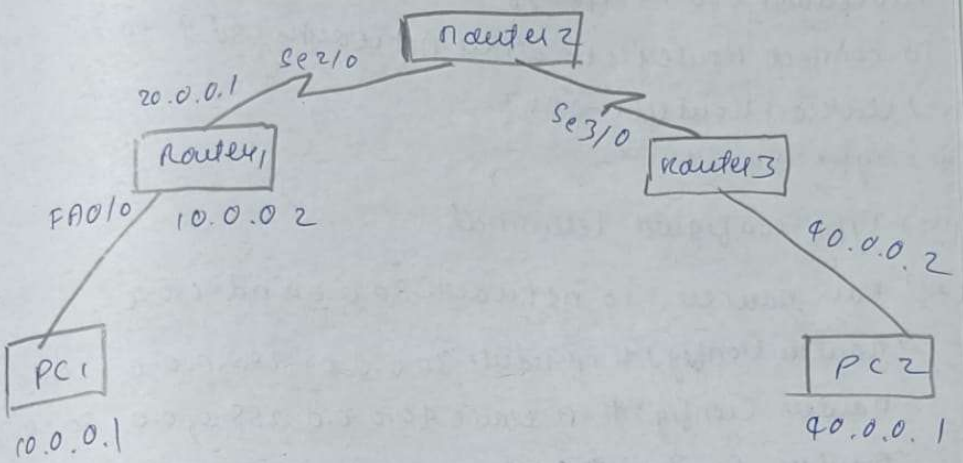
Request timed out

Request timed out

Ping statistics for 10.0.0.2.

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

Topology



Procedure:

1) Take 3 generic routers and two end devices.

2) Set the ip address of end devices

3) Go to command line interface

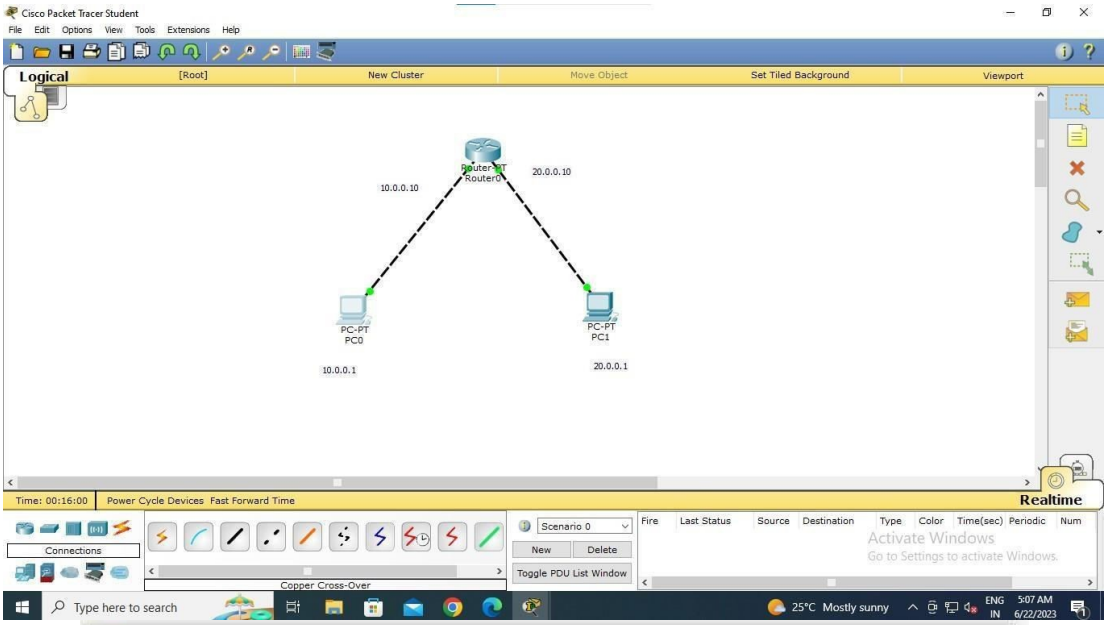
4) Similarly for the router-router connection.

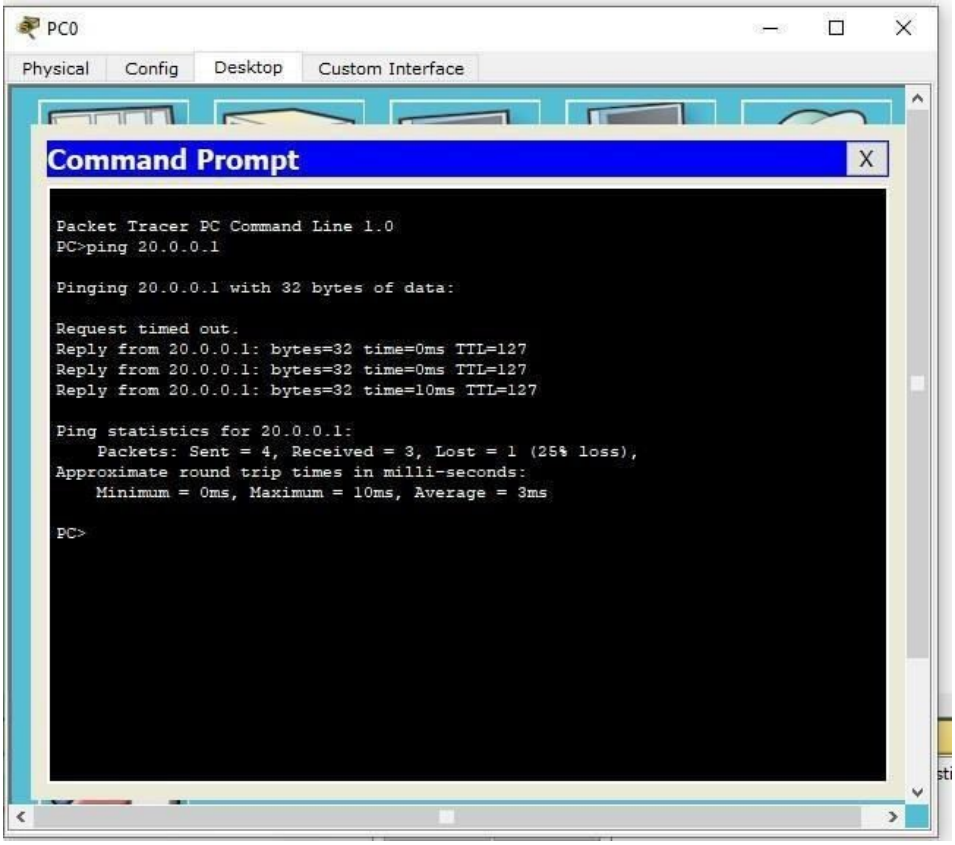
Result:

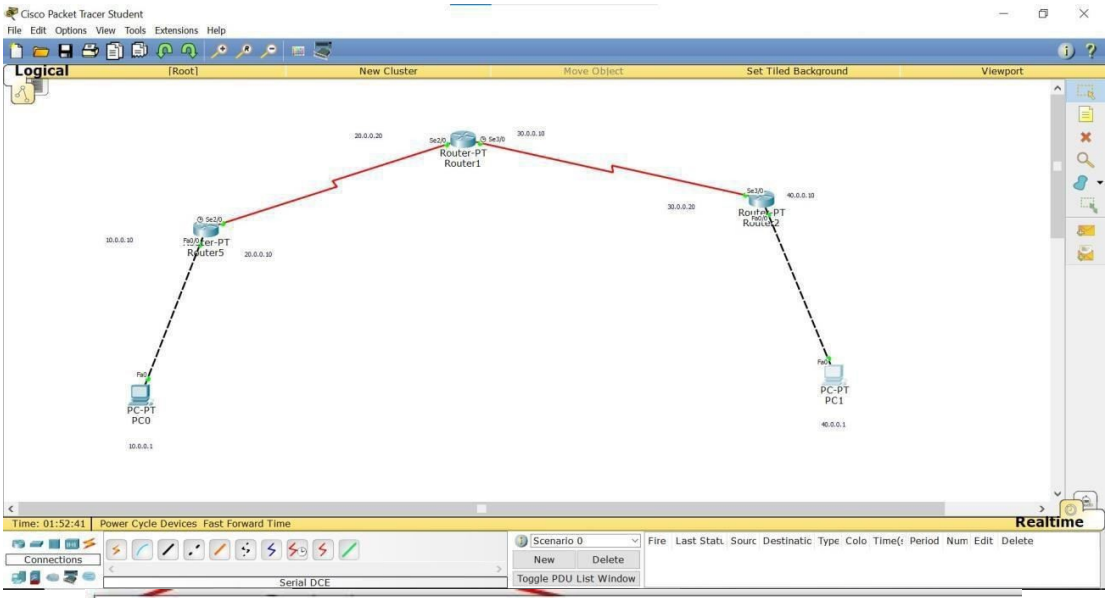
23/8/23 Ping 20.0.0.1

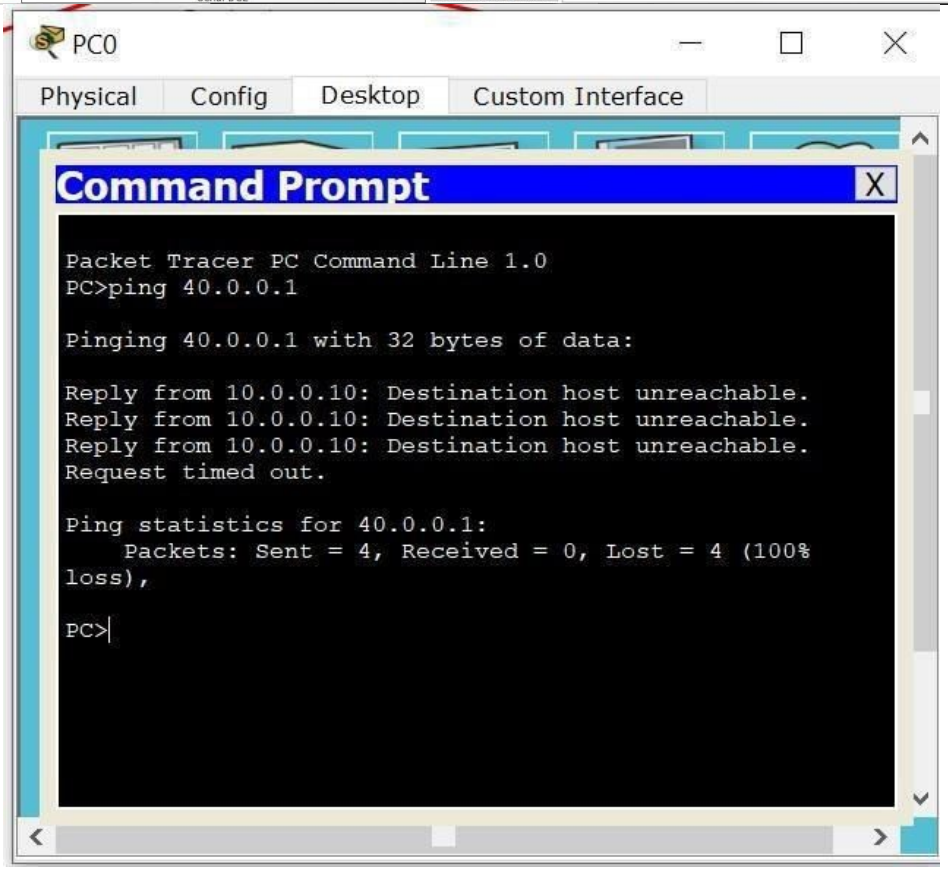
Reply from 20.0.0.1 time=0ms TTL=128
Reply from 20.0.0.1 time=0ms TTL=128
Reply from 20.0.0.1 time=0ms TTL=128
Reply from 20.0.0.1 time=0ms TTL=128

SCREENSHOTS





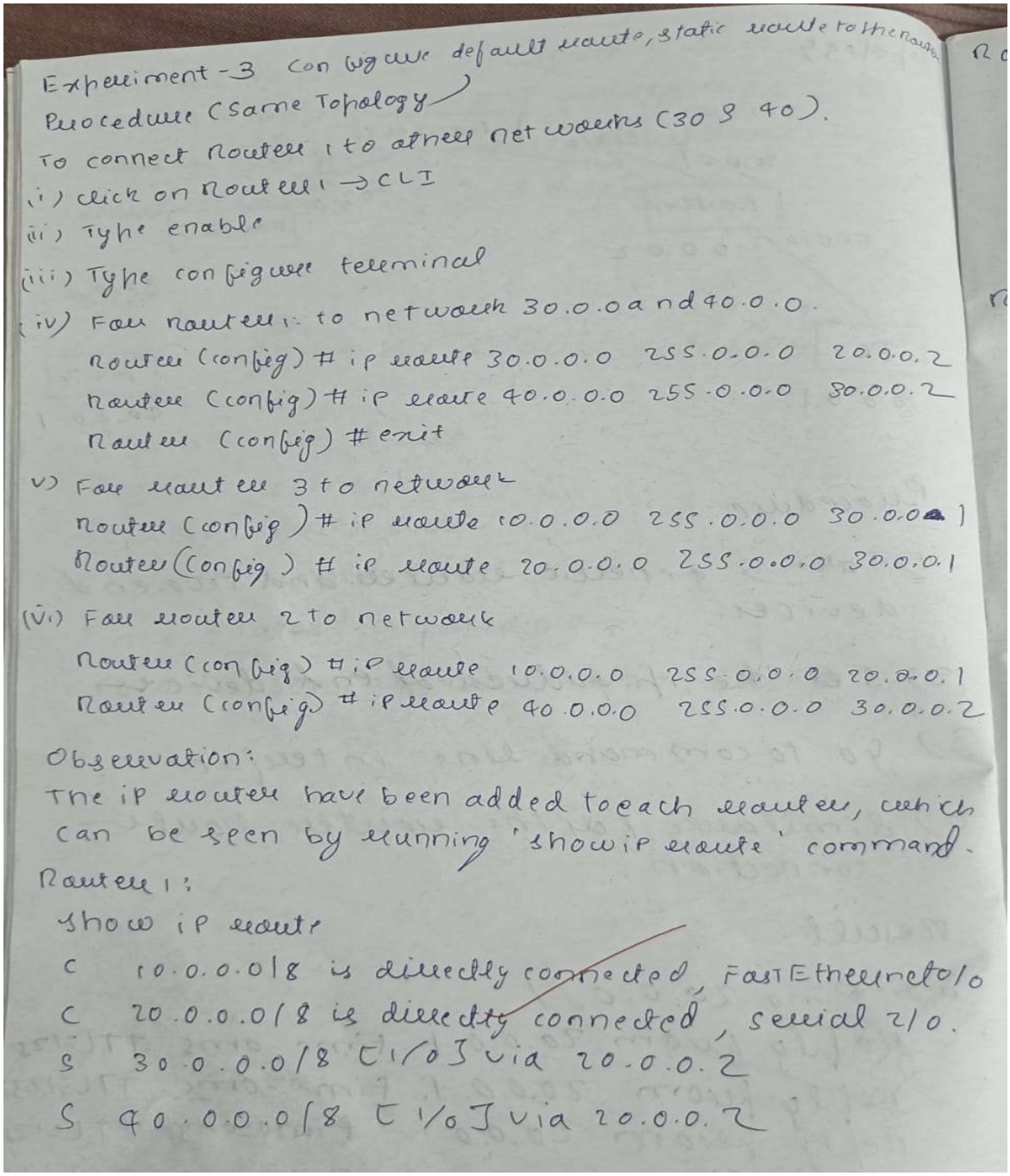

Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT
Router1
Router-PT
Router5
Router-PT
Router2
PC-PT
PC0
PC-PT
PC1
Time: 01:52:41
Power Cycle Devices Fast Forward Time
Realtime
Connections
Serial DCE
Scenario 0
New
Delete
Toggle PDU List Window


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:
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>

CN LAB 3

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

OBSERVATION:



Experiment - 3 Configure default route, static route to the router.

Procedure (Same Topology)

To connect Router 1 to other networks (30 & 40).

(i) click on Router 1 → CLI

(ii) Type enable

(iii) Type configure terminal

(iv) For router 1 to network 30.0.0 and 40.0.0.

Router (config) # ip route 30.0.0.0 255.0.0.0 20.0.0.2

Router (config) # ip route 40.0.0.0 255.0.0.0 30.0.0.2

Router (config) # exit

v) For router 3 to network

Router (config) # ip route 10.0.0.0 255.0.0.0 30.0.0.1

Router (config) # ip route 20.0.0.0 255.0.0.0 30.0.0.1

(vi) For router 2 to network

Router (config) # ip route 10.0.0.0 255.0.0.0 20.0.0.1

Router (config) # ip route 40.0.0.0 255.0.0.0 30.0.0.2

Observation:

The ip routes have been added to each router, which can be seen by running 'show ip route' command.

Router 1:

show ip route

C 10.0.0.0/8 is directly connected, FastEthernet0/0

C 20.0.0.0/8 is directly connected, serial 2/0.

S 30.0.0.0/8 [1/0] via 20.0.0.2

S 40.0.0.0/8 [1/0] via 20.0.0.2

Router 2

show ip route

S 10.0.0.0/8 [1/0] via 20.0.0.1

S 40.0.0.0/8 [1/0] via 30.0.0.2

C 20.0.0.0/8 is directly connected, Serial 2/0

C 30.0.0.0/8 is directly connected, Serial 3/0

Router 3

show ip route

S 10.0.0.0/8 [1/0] via 30.0.0.1

S 20.0.0.0/8 [1/0] via 30.0.0.1

C 30.0.0.0/8 is directly connected Se 3/0

C 40.0.0.0/8 is directly connected Fa 0/0

Output

The ping requests to all networks are successful.

From 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=10ms TTL=125

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

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

Reply from 40.0.0.1 : bytes=32 time=8ms TTL=125

Ping statistics for 40.0.0.1

Packets sent = 4, Received = 4, Lost = 0 (0% loss).

→ Ping 30.0.0.2

Pinging 30.0.0.2 with 32 bytes of data

Reply from 30.0.0.2: byes=32, time=7ms TTL=253

Reply from 30.0.0.2: bytes=32, time=7ms, TTL=253

Reply from 30.0.0.2: bytes=32, time=7ms, TTL=253

Reply from 30.0.0.2: bytes=32, time=7ms, TTL=253

Ping statistics for 30.0.0.2

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

For PC 2

→ Ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data

Reply from 20.0.0.1 : bytes = 32 time = 2ms TTL = 253

Reply from 20.0.0.1 : bytes = 32 time = 9ms TTL = 253

Reply from 20.0.0.1 : bytes = 32 time = 6ms TTL = 253

Reply from 20.0.0.1 : bytes = 32 time = 7ms TTL = 253

Ping statistics for 20.0.0.1

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

→ Ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data

Reply from 10.0.0.1 : bytes = 32 time = 7ms TTL = 125

Reply from 10.0.0.1 : bytes = 32 time = 7ms TTL = 125

Reply from 10.0.0.1 : bytes = 32 time = 7ms TTL = 125

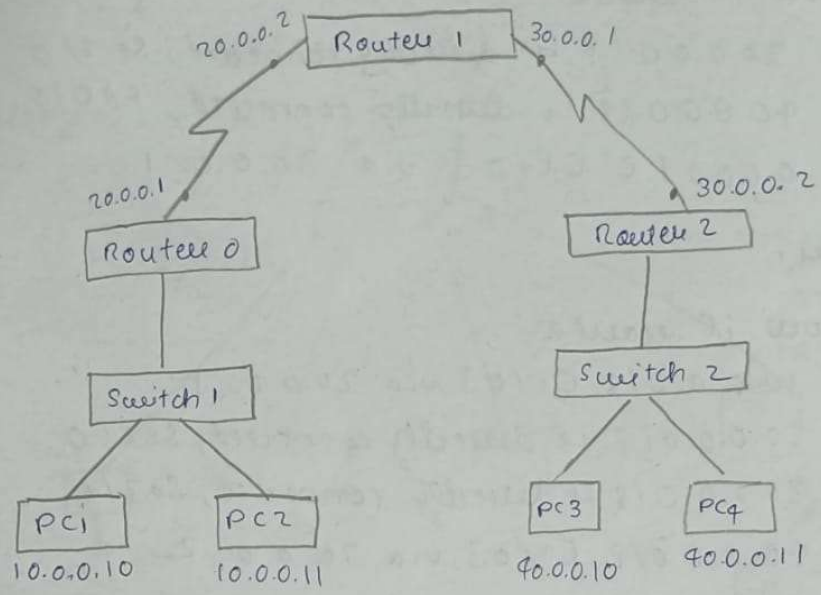
Reply from 10.0.0.1 : bytes = 32 time = 7ms TTL = 125

Ping statistics for 10.0.0.1

Packets, Sent = 4, Received = 4, LosT = 0

DEFAULT Routing

TOPOLOGY



Procedure:

configuring default route to Router 0 and Router 2

Router 0:

Router (config) # ip route 0.0.0.0 0.0.0.0 20.0.0.2

Router 2:

Router (config) # ip route 0.0.0.0 0.0.0.0 30.0.0.1

Configure 2 static routes of Router 1

Router 1:

~~Router (config) # ip route 10.0.0.0 255.0.0.0 20.0.0.1~~

Router (config) # ip route 40.0.0.0 255.0.0.0 30.0.0.2

Observation:

Static routes to router 1 have been added and default routes to router 0 and router 2

Router 0:

show ip route

C 10.0.0.0/8 is directly connected, Fe0/0

C 20.0.0.0/8 is directly connected, Se2/0.

S* 0.0.0.0/0 [1/0] via 20.0.0.2

Router 2

show ip route

C 30.0.0.0/8 is directly connected, Se 2/0

C 40.0.0.0/8 is directly connected, Fe0/0

S* 0.0.0.0/0 [1/0] via 30.0.0.1

Router 1

Show ip route

S 10.0.0.0/8 [1/0] via 20.0.0.1

C 20.0.0.0/8 is directly connected, Se2/0

C 30.0.0.0/8 is directly connected, Se3/0

S 40.0.0.0/8 [1/0] via 30.0.0.2

Output

Ping requests from PC4

→ Ping 10.0.0.10

Pinging 10.0.0.10 with 32 bytes of data

Reply from 10.0.0.10: bytes=32 time=4ms TTL=125

Reply from 10.0.0.10: bytes=32 time=5ms TTL=125

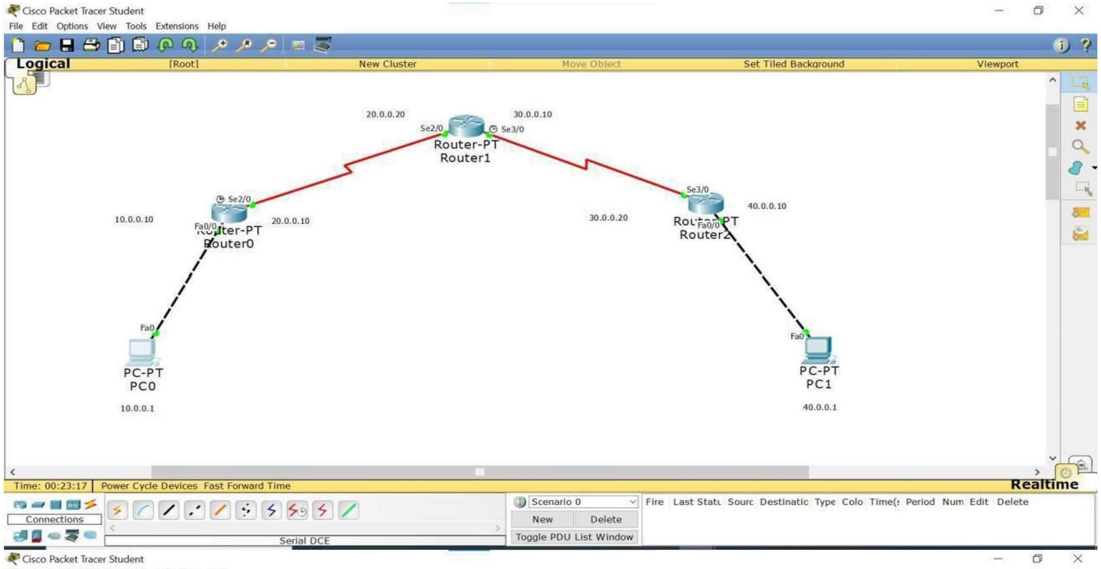
Reply from 10.0.0.10: bytes=32 time=4ms TTL=125

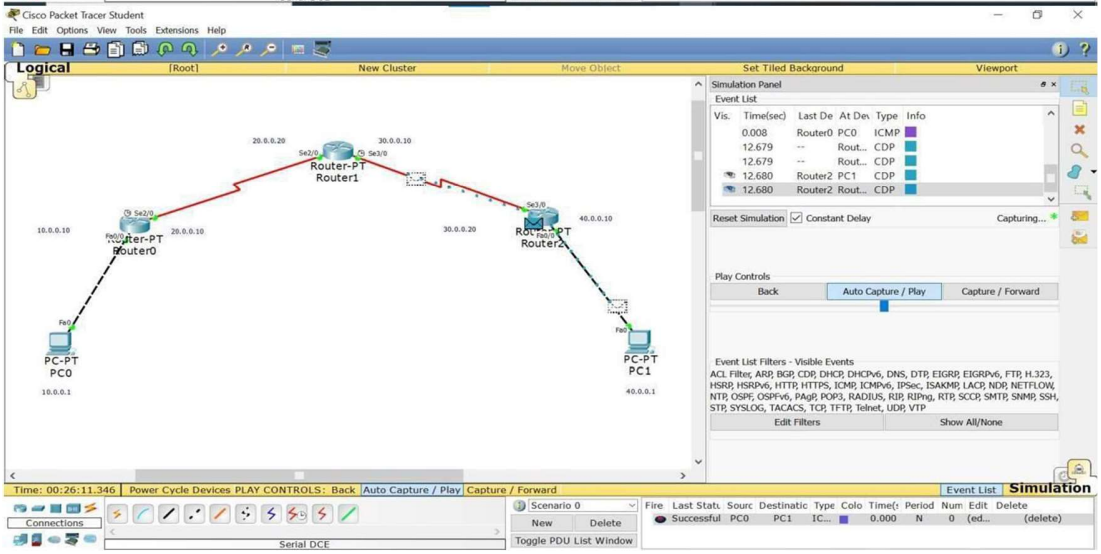
Reply from 10.0.0.10: bytes=32 time=4ms TTL=125

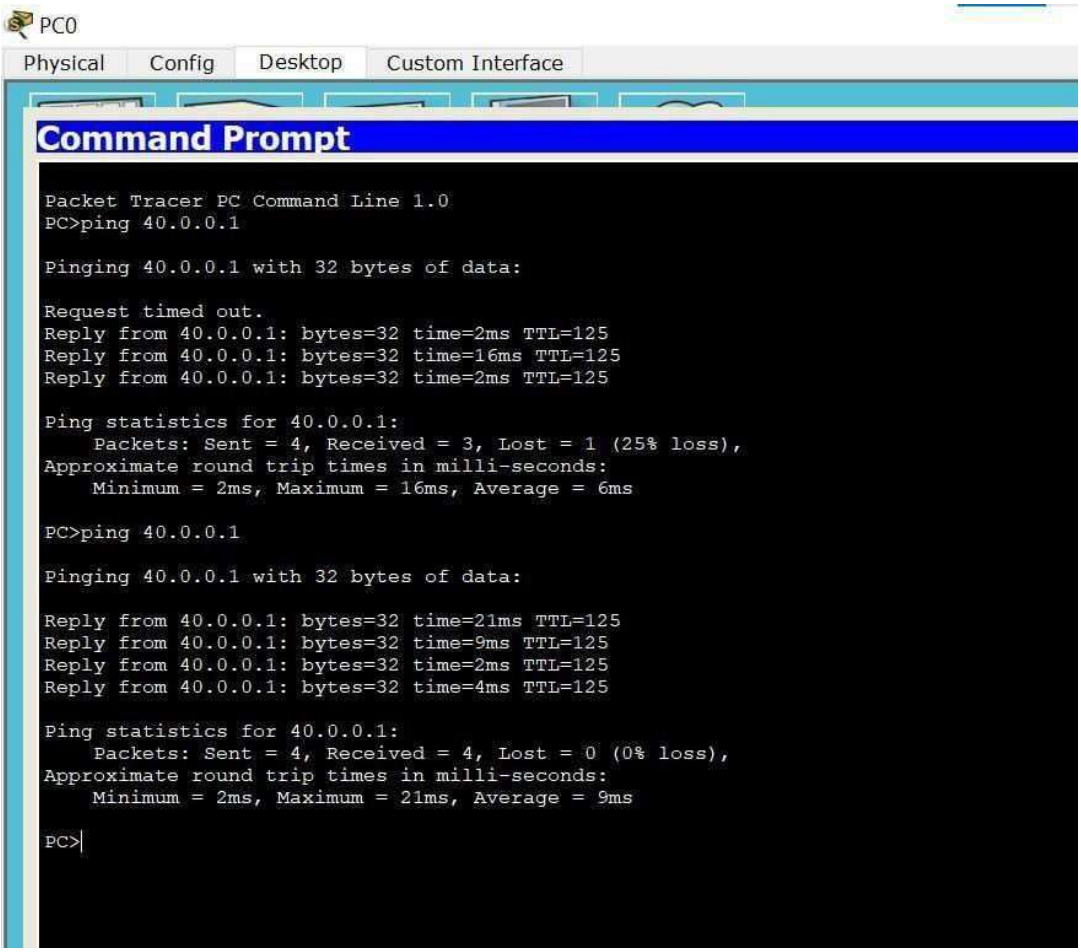
Ping statistics for 10.0.0.10

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

SCREENSHOTS






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>

CN LAB 4

AIM: Configure DHCP within a LAN and outside LAN.

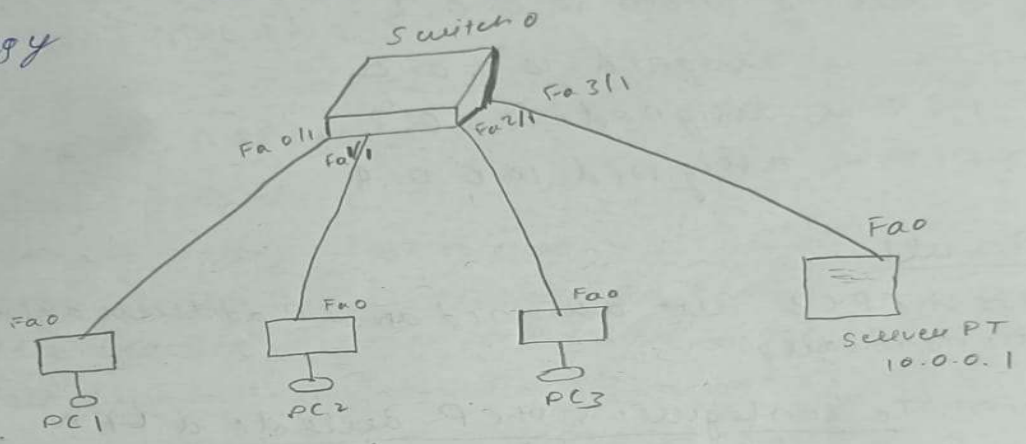
OBSERVATION:

Experiment - 4 : configure DHCP within a LAN and outside LAN.

Aim: configure DHCP within a LAN and outside LAN

within a LAN

Topology



Procedure

→ Drag and drop 3 PC's, 1 switch and 1 server

→ connect all the PC's and server to the switch

→ Set the IP address of the server manually. Say 10.0.0.1

→ click on server services then select DHCP

→ Set the start IP address say 10.0.0.2

→ Add the configuration

→ ~~Now~~ click on each PC and go to config and select IP configuration as DHCP.

Observation

→ All the PC's connected to the switch and server gets automatically assigned IP address starting from 10.0.0.2.

→ PC1 is assigned 10.0.0.2

→ PC2 is assigned 10.0.0.3

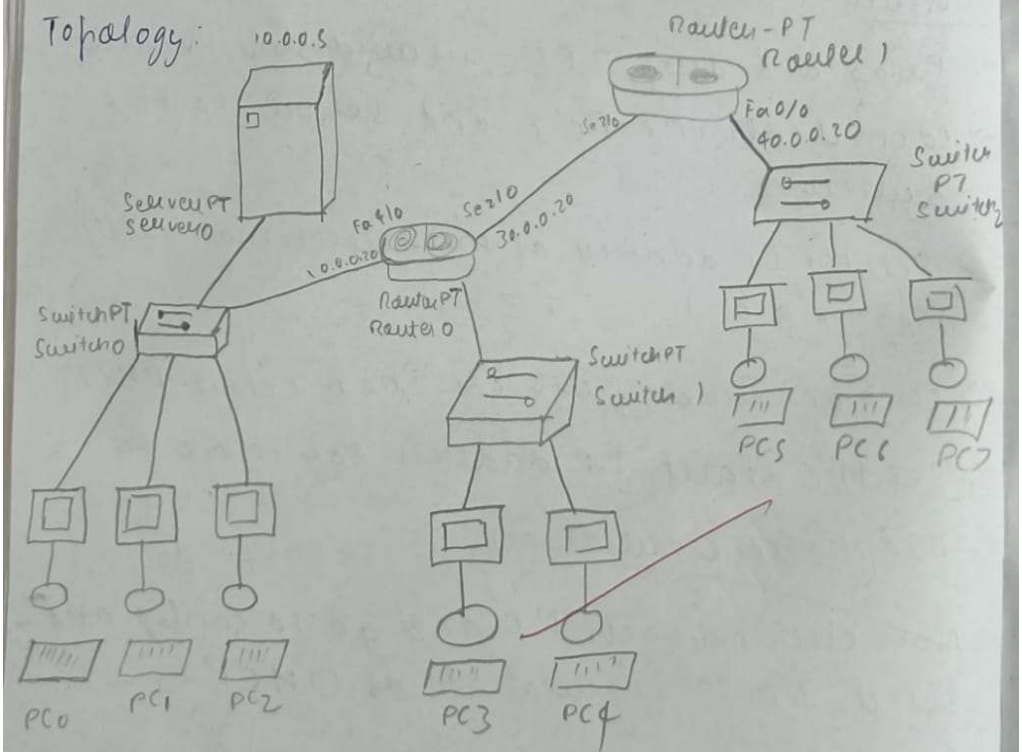
→ PC2 is assigned 10.0.0.4

Result

All the PC's are assigned an IP address automatically/ Dynamically.

Aim: To configure DHCP outside a LAN.

Topology:



Procedure:

1) Repeat the procedure we did for the LAN (DHCP within a LAN).

2) Now add 2 routers, another set of PC's and switch 1 (LAN2)

3) Config the IP address of ports Fa 4/0 and Fa 0/0 of Router 0.

Router > enable
router # config terminal
Router (config) # interface Fa 4/0
Router (config-if) # ip address 10.0.0.2 255.0.0.0
Router (config-if) # ip helper-address 10.0.0.5
Router (config-if) # no shutdown
Router (config-if) # exit

Similarly for Fa 0/0.

Here ip helper address is the ip address of the server 0

4) Go to PC3 and PC4 and switch to DHCP in system → IP configuration

5) Connect Router1 to Router 0. Router 1 is connected to switch 2 and 3 PC's (PC5, PC6, PC7)

6) Config IP address of Router 1 for interface serial 2/0 and Fa 0/0 with 30.0.0.30 and 40.0.0.20 respectively

7) Config IP address of Router 0 for serial 2/0 as 30.0.0.20

8) Perform static routing for router 0 →

Router (config) # ip route 40.0.0.0 255.0.0.0 30.0.0.30

for Router 1 =)

Router (config) # ip route 10.0.0.0 255.0.0.0 30.0.0.20

9) go to Server and create 2 more server pools with different names.

	Default Gateway	DNS Server	Start IP	Subnet mask
Server Pool 1	10.0.0.20	10.0.0.5	10.0.0.10	255.0.0.0
Server Pool 2	10.0.0.20	10.0.0.5	20.0.0.20	255.0.0.0
Server Pool 3	10.0.0.20	10.0.0.5	40.0.0.10	255.0.0.0

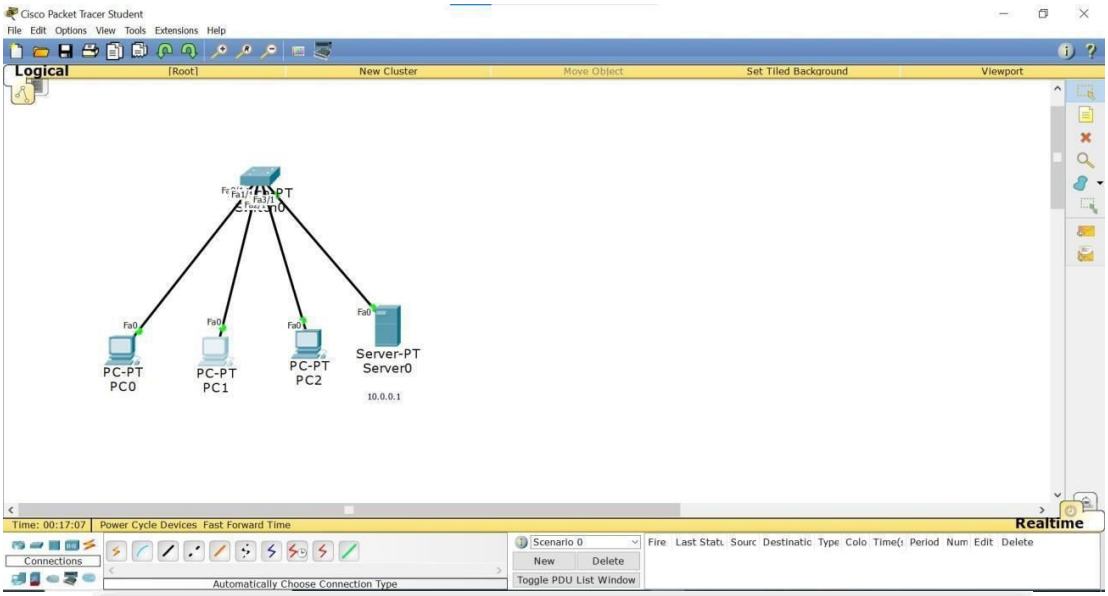
10) go to PC5, PC6, PC7 switch to DHCP in IP configuration. An IP address will be given to each PCs.

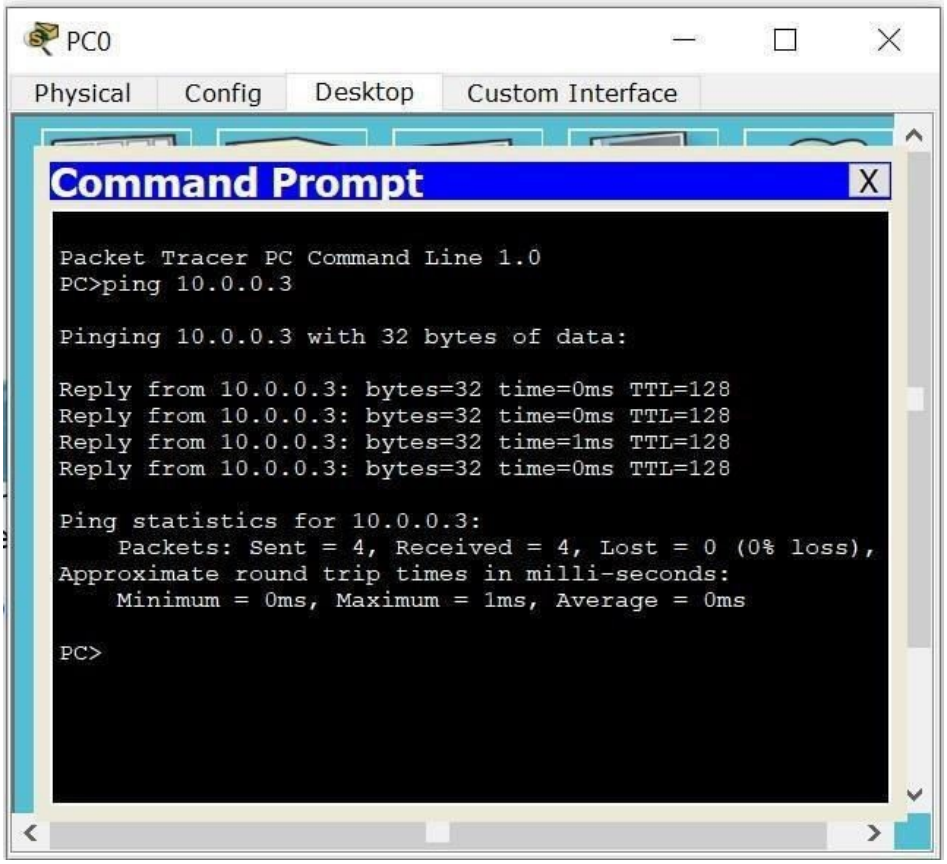
Observation / Result

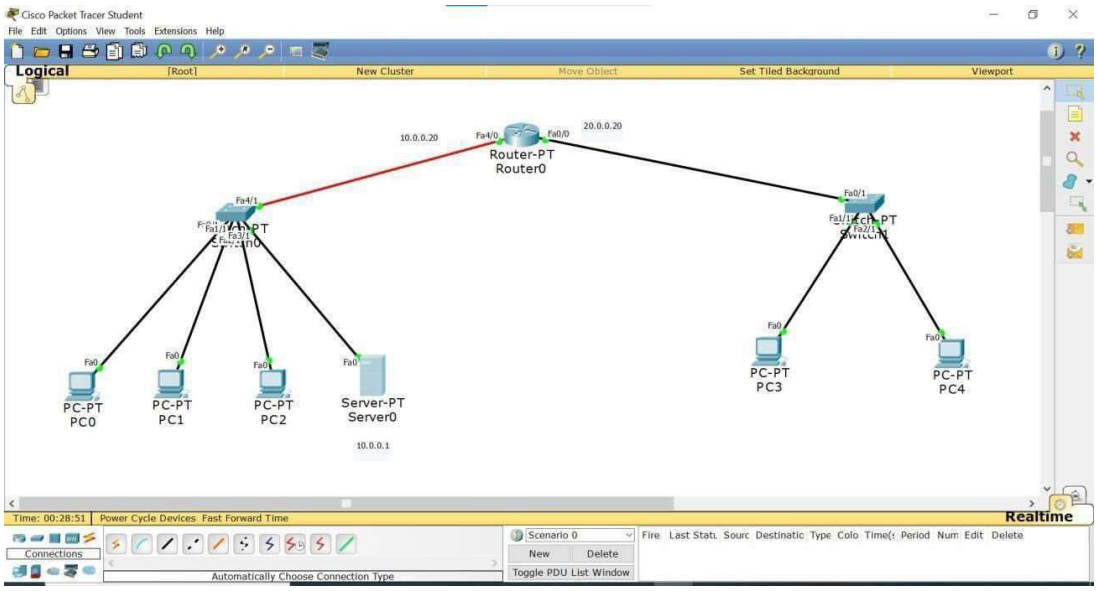
IP addresses are set automatically using DHCP protocol by the server.

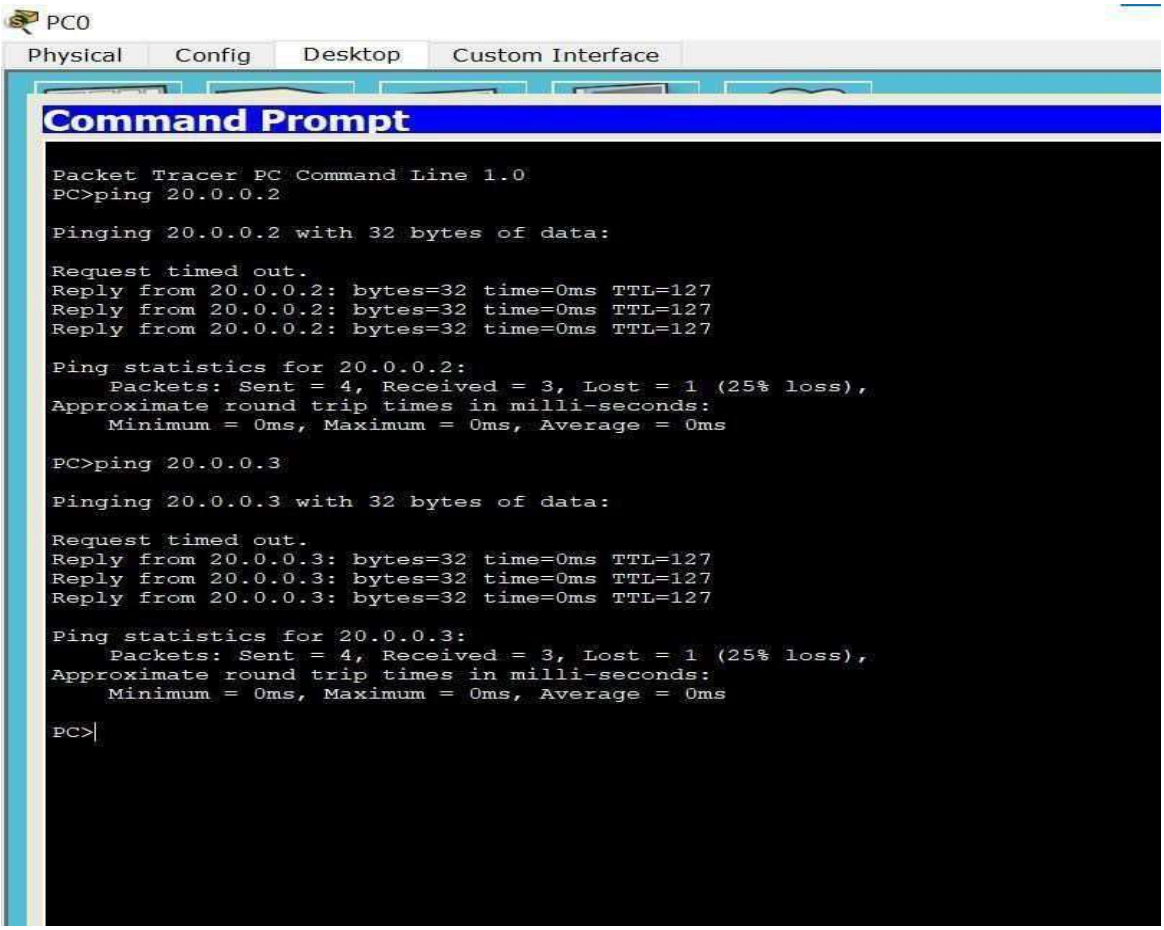
Sri

SCREENSHOTS





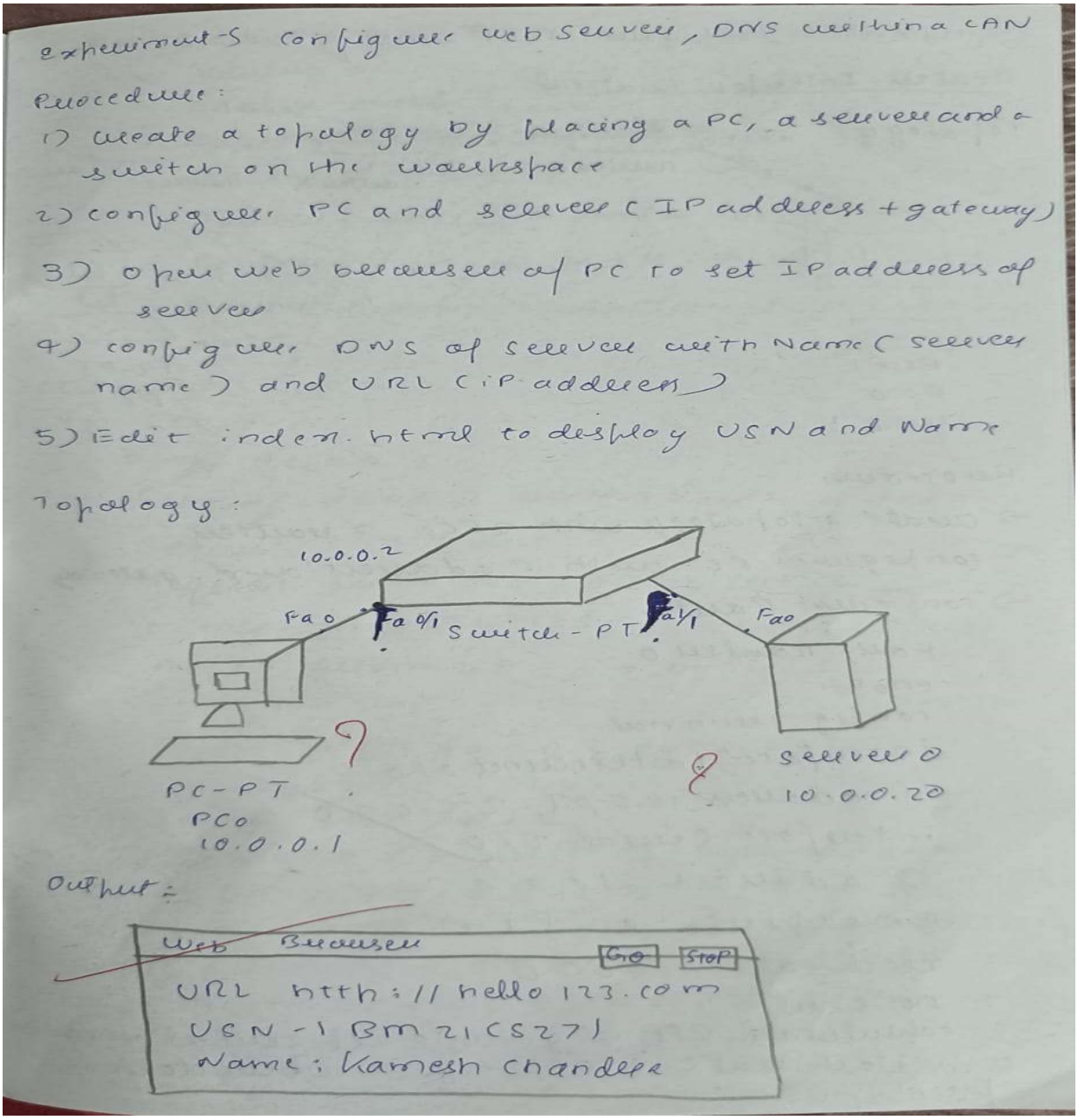

Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
10.0.0.20
Fa4/0
Fa0/0
20.0.0.20
Router-PT
Router0
Fa4/1
Fa0/1
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
Server-PT
Server0
10.0.0.1
PC-PT
PC3
PC-PT
PC4
Time: 00:28:51
Power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


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>

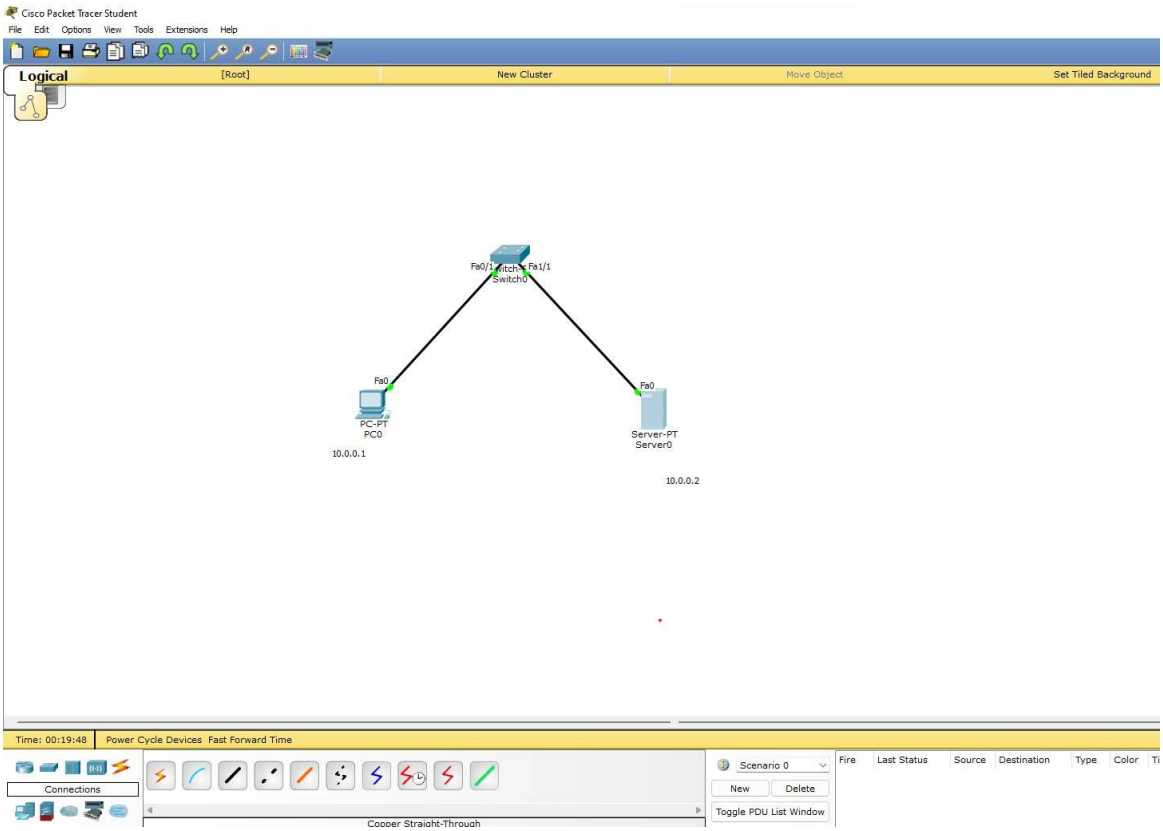
CN LAB 5

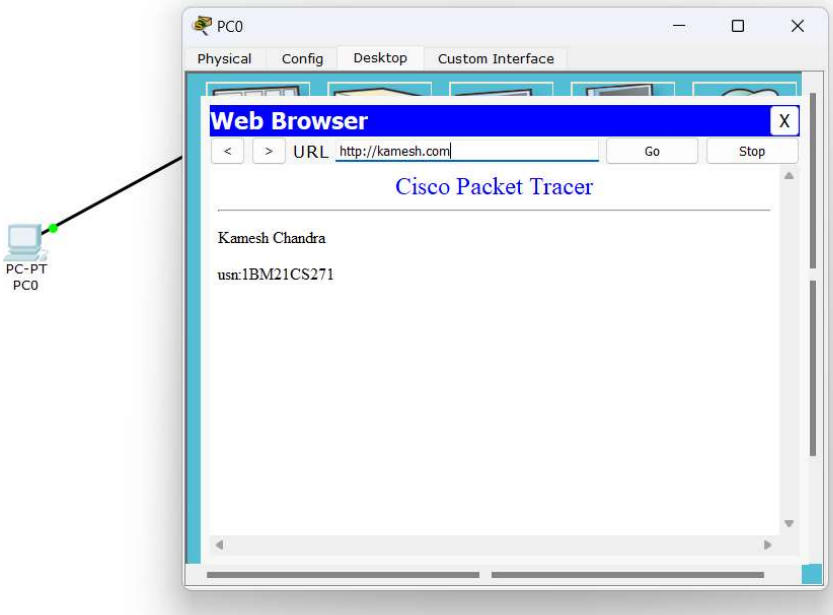
AIM: Configure Web Server, DNS within a LAN.

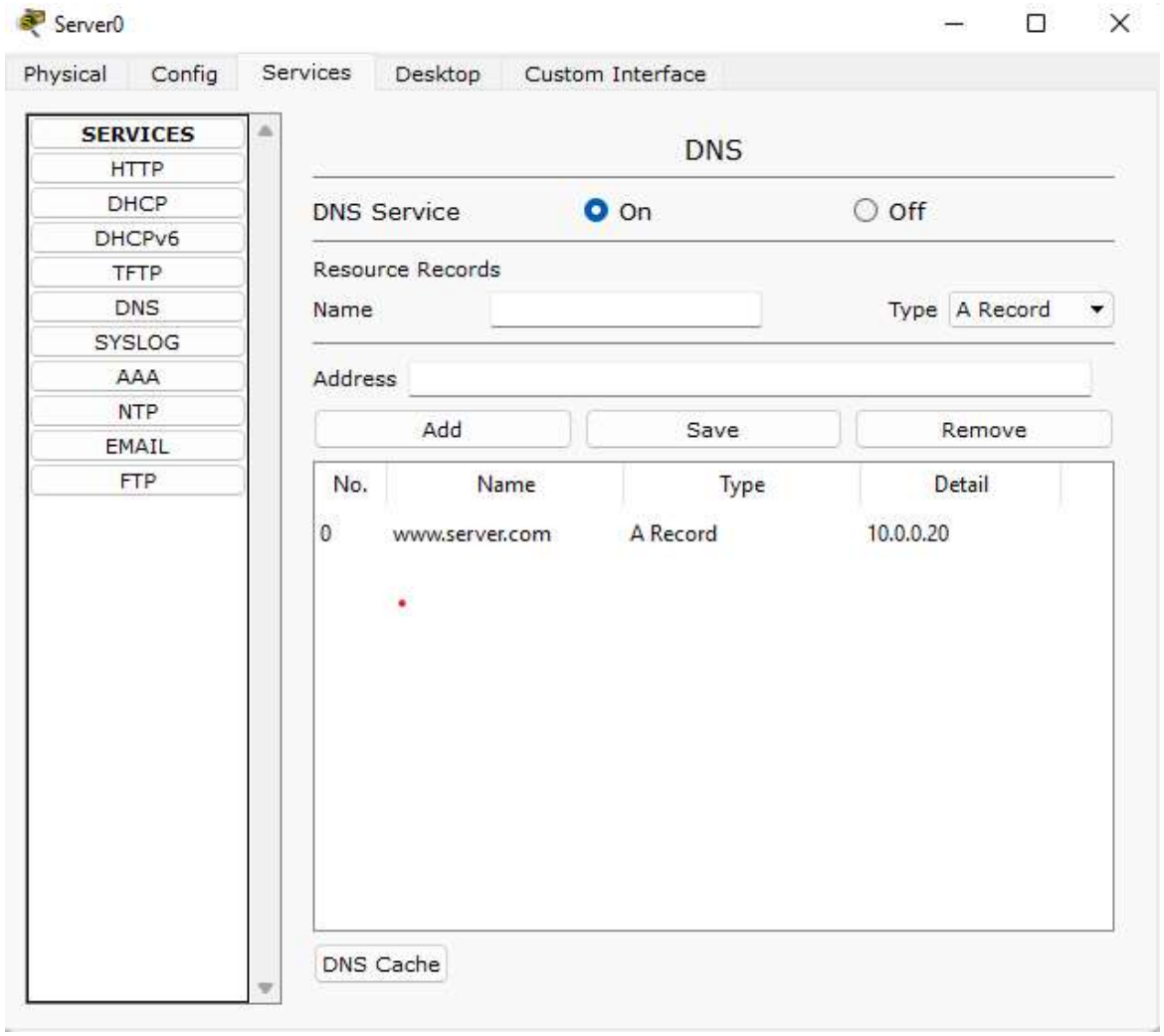
OBSERVATION:



SCREENSHOTS






Server0
Physical
Config
Services
Desktop
Custom Interface
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
DNS
DNS Service
On
Off
Resource Records
Name
Type
A Record
Address
Add
Save
Remove
No.
Name
Type
Detail
0
www.server.com
A Record
10.0.0.20
DNS Cache

CN LAB 6

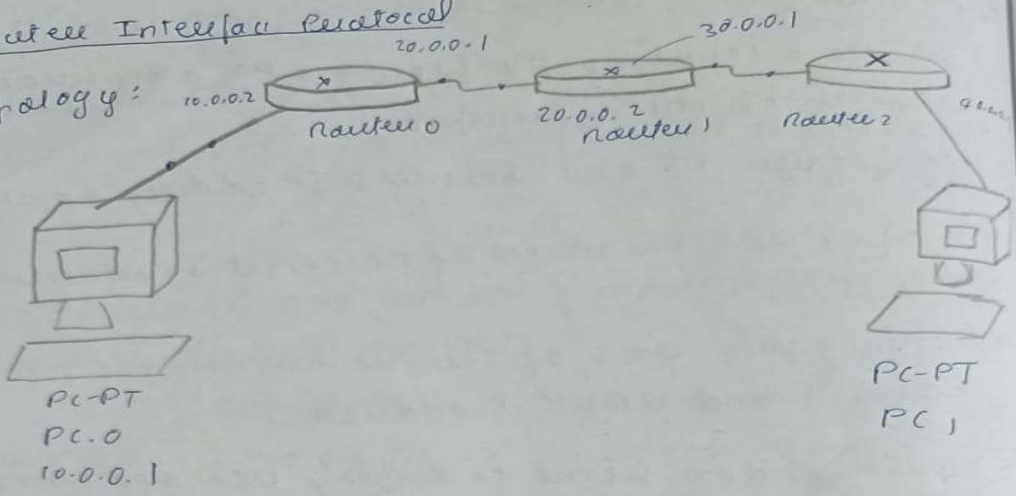
AIM: Configure RIP routing Protocol in Routers.

OBSERVATION:

Experiment-6 Configure RIP routing Protocol in Routers

Routers Interface Protocol

Topology:



Procedure:

→ create a topology with 2 PC's, 3 routers

→ configure 'PC' with IP address and gateway

→ configure Router

For Router 0

enable

config terminal

interface fastethernet 0/0

ip address 10.0.0.1 255.0.0.0

interface serial 2/0

ip address 10.0.0.2

encapsulation PPP

clockrate 64000

no shut

→ encapsulation PPP is given to all serial connection and clock rate 64000 is given where clock is present.

configure all routers

For router 0

router enable

router # show ip route

Router # config - t

Router (config) # router rip

Router (config - router) # network 10.0.0.0

Router (config - router) # network 20.0.0.0

exit

output :-

In PC0,

Ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data

Request timed out

Request from 40.0.0.2 bytes = 32

time = 2ms TTL = 125

Reply from 40.0.0.2 bytes = 32

time = 2ms TTL = 125

Reply from 40.0.0.2 bytes = 32

time = 2ms TTL = 125

ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data

Reply from 40.0.0.2 bytes = 32 time = 2ms TTL = 125

Reply from 40.0.0.2 bytes = 32 time = 2ms TTL = 125

Reply from 40.0.0.2 bytes = 32 time = 2ms TTL = 125

Reply from 40.0.0.2 bytes = 32 time = 2ms TTL = 125

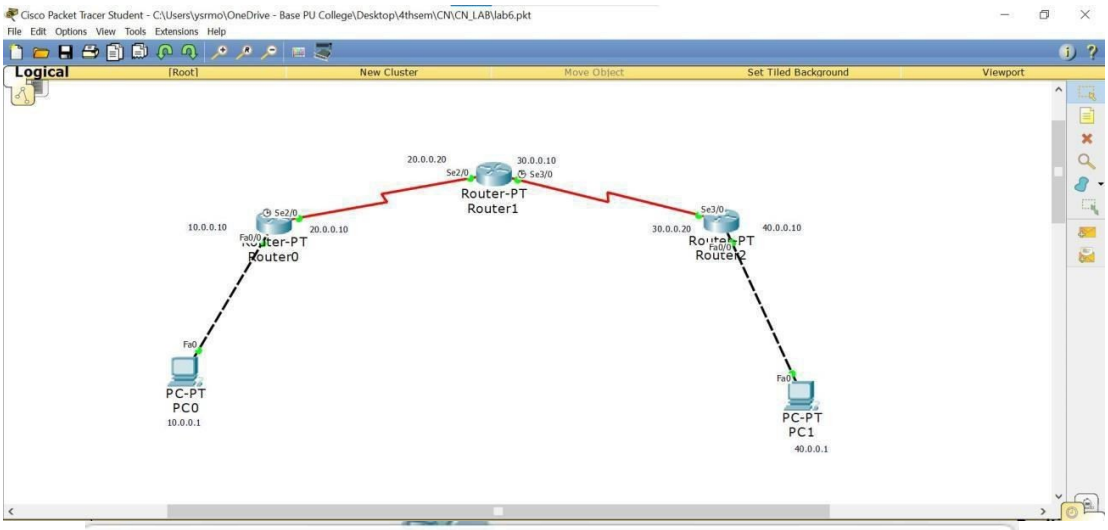
Ping statistics for 40.0.0.2

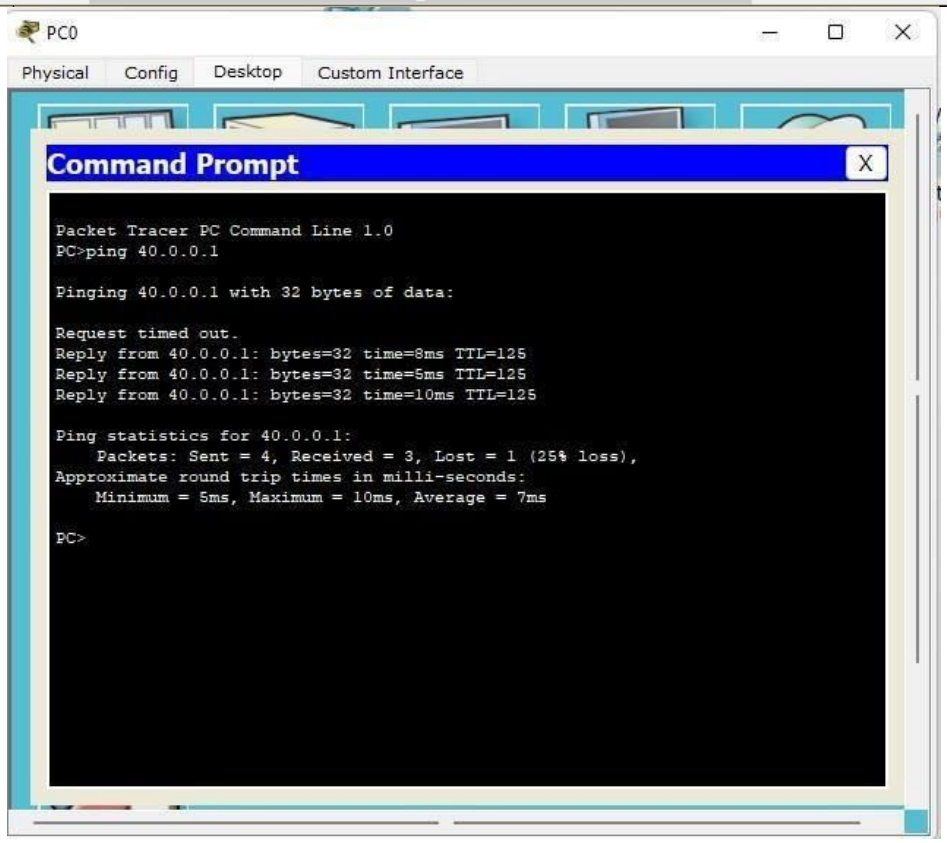
Packets: Sent = 4 Received = 4 Lost = 0 (0% loss)

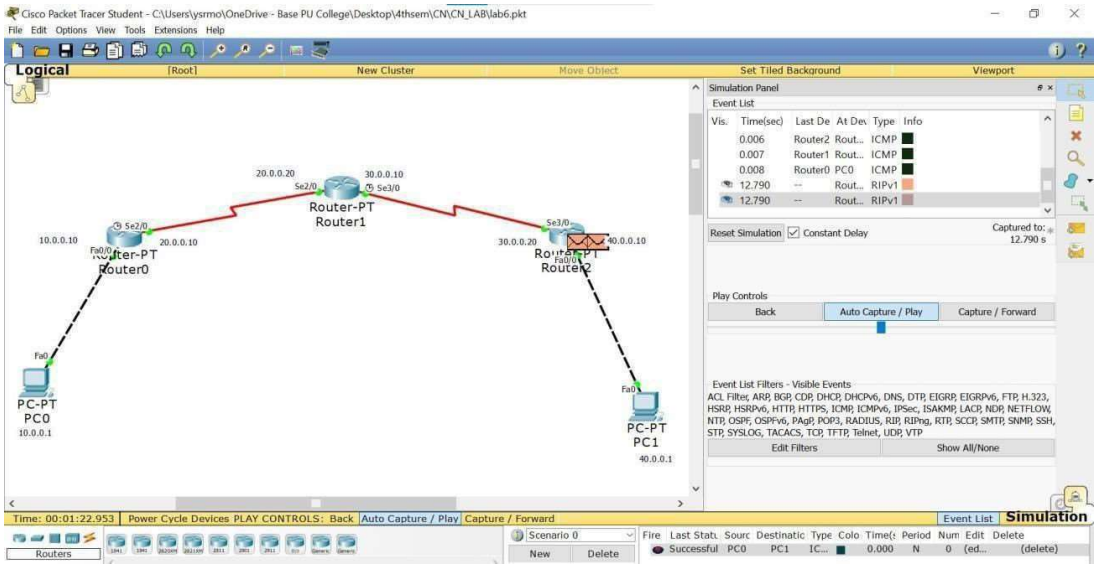
Approximate round trip times in milli-seconds

minimum = 2ms, maximum = 13 ms

SCREENSHOTS



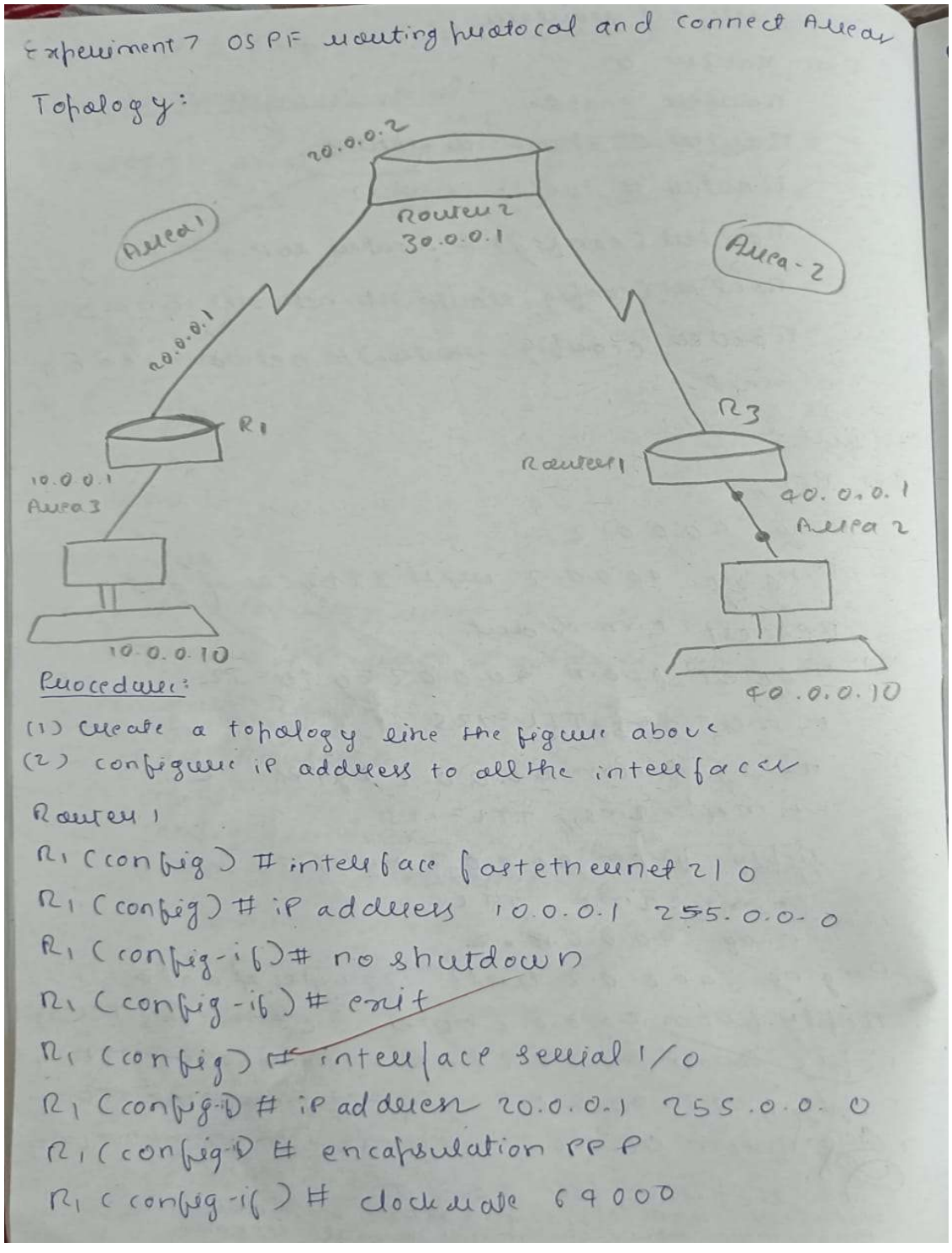



Cisco Packet Tracer Student - C:\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\CN\CN_LAB\lab6.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
20.0.0.20
30.0.0.10
Se2/0
Se3/0
Router-PT
Router1
10.0.0.10
Fa0/0
20.0.0.10
Router0
30.0.0.20
40.0.0.10
Router2
Fa0
PC-PT
PC0
10.0.0.1
PC-PT
PC1
40.0.0.1
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 Stat Sourc Destinatic Type Colo Time(s Period Num Edit Delete
Successful PC0 PC1 IC... 0.000 N 0 (ed... (delete)

CN LAB 7

AIM: Configure OSPF routing protocol.

OBSERVATION:



R1(config-if)# no shutdown
R1(config-if)# exit

Router2,

R2(config)# interface serial 1/0
R2(config-if)# ip address 20.0.0.2 255.0.0.0
R2(config-if)# encapsulation PPP
R2(config-if)# no shutdown
R2(config-if)# exit

R2(config)# interface serial 1/1
R2(config-if)# ip address 30.0.0.1 255.0.0.0
R2(config-if)# encapsulation PPP
R2(config-if)# clock rate 64000
R2(config-if)# no shutdown
R2(config-if)# exit

Router R3
R3(config)# interface serial 1/0
R3(config-if)# ip address 30.0.0.2 255.0.0.0
R3(config-if)# encapsulation PPP
R3(config-if)# no shutdown
R3(config-if)# exit
R3(config)# interface fastethernet 2/0
R3(config-if)# ip address 40.0.0.1 255.0.0.0
R3(config-if)# no shutdown
R3(config-if)# exit

3) Now, Enable IP routing by configuring OSPF routing protocol in all routers.

In router R_1,

```
R1(config)# router ospf 1
R1(config-router)# router-id 1.1.1.1
R1(config-router)# network 10.0.0.0 255.255.255.255 area 0
R1(config-router)# network 20.0.0.0 255.255.255.255 area 1
R1(config-router)# exit
```

Repeat the procedure for router 2 & Router 3

(4)
```
R1(config-if)# interface loopback 0
R1(config-if)# ip add 172.16.1.252 255.255.0.0
R1(config-if)# no shutdown
```

Repeat the procedure for R2 and R3

(5) Create virtual link b/w R_1, R_2 by this we create a virtual link to connect area 3 to area 0.

Router R_1

```
R1(config)# router ospf 1
R1(config-router)# area 1 virtual-link 2.2.2.2
R1(config-router)#
*Feb 10 10:29:23:267: %OSPF-5-ADJCHG:
Process 1, Nbr 2.2.2 on OSPF_VL0 from loading
to full. Loading Done
```

Repeat the procedure for R2 and R3

6) Check connectivity b/w host 10.0.0.10 to 40.0.0.10

Output / Observation :

Ping 40.0.0.10

Ping 40.0.0.10 : 56 bytes

64 bytes from 40.0.0.10 Seq = 1 ttl = 61 time = 73 ms

64 bytes from 40.0.0.10 seq = 1 ttl = 61 time = 64 ms

64 bytes from 40.0.0.10 Seq = 1 ttl = 61 time = 66 ms

64 bytes from 40.0.0.10 seq = 1 ttl = 61 time = 60 ms

64 bytes from 40.0.0.10 seq = 1 ttl = 61 time = 96 ms

5 packets transmitted, 5 packets received

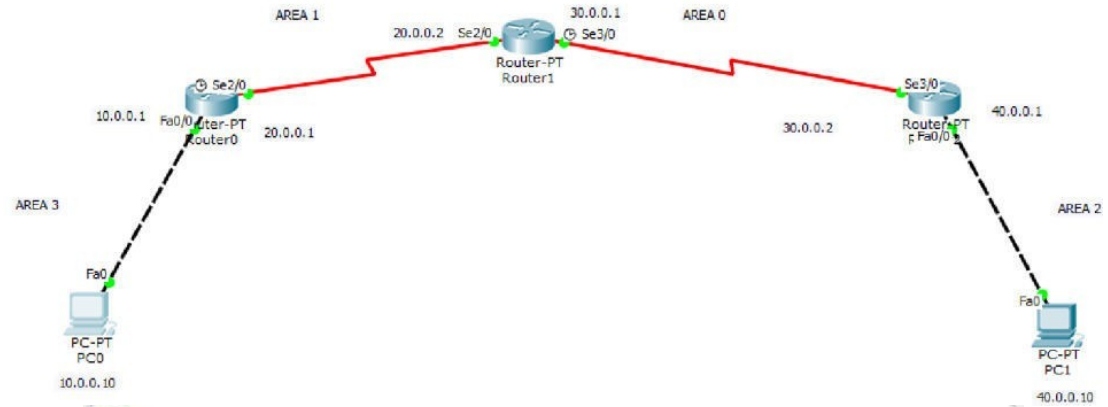
16 % packet loss round-trip min / avg / max

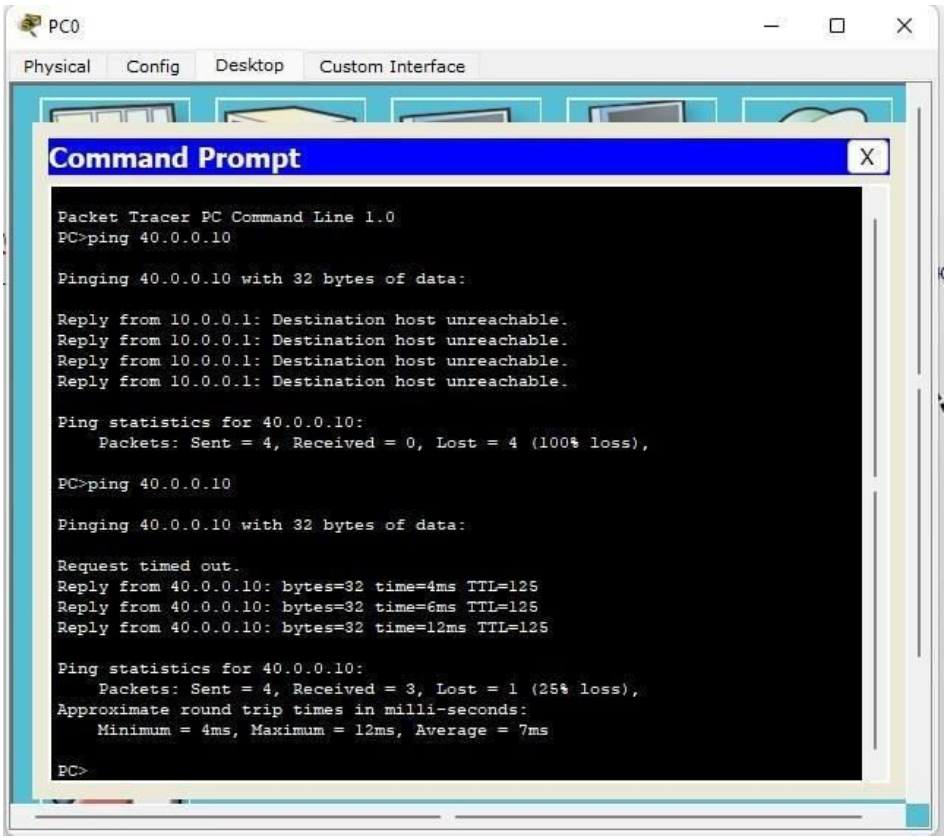
60.829/92.438/173.753 ms

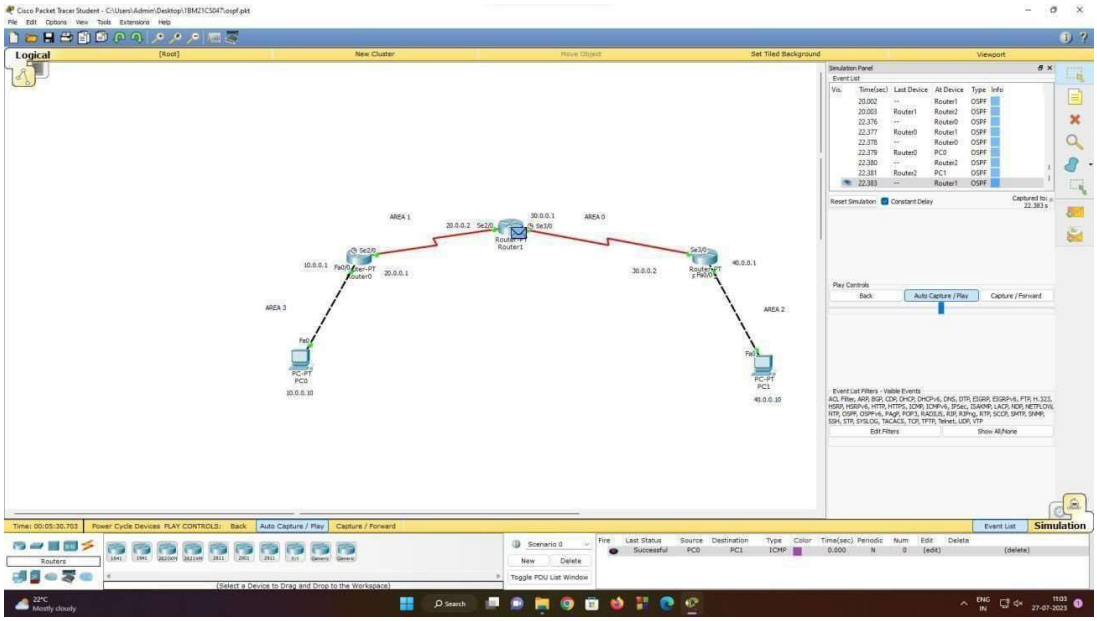
Sp.1

18/8/23

SCREENSHOTS:



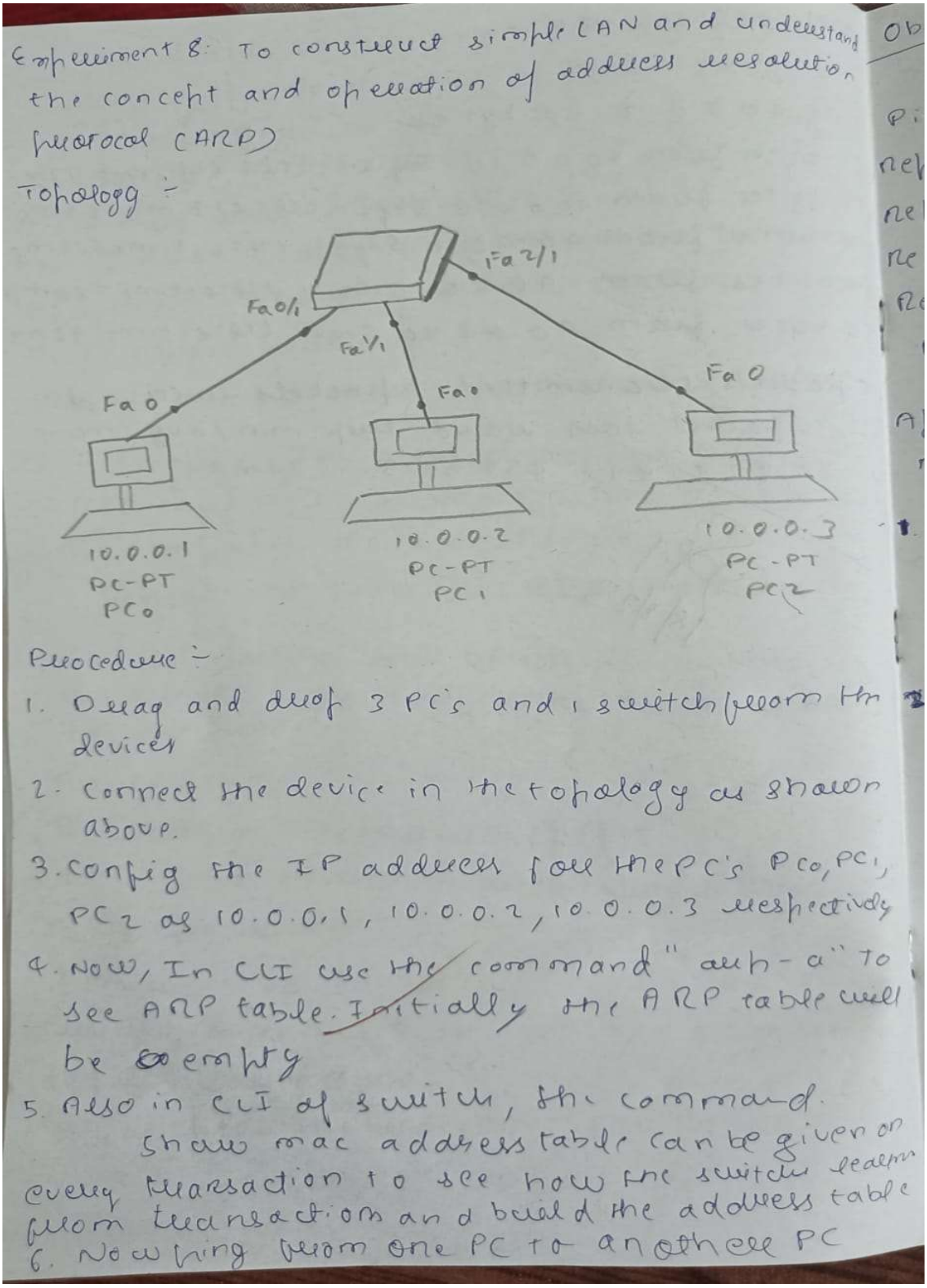



Cisco Packet Tracer Student - C:\Users\Admin\Desktop\1BM21CS047\ospf.pkt
Logical
AREA 1
AREA 0
AREA 3
AREA 2
20.0.0.2
30.0.0.1
10.0.0.1
20.0.0.1
30.0.0.2
40.0.0.1
10.0.0.10
40.0.0.10
Router1
Router0
PC0
PC1
Simulation Panel
Event List
Play Controls
Auto Capture / Play
Capture / Forward
Simulation

CN LAB 8

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

OBSERVATION:



Observation :

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=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 times in milli-seconds

minimum = 0ms, maximum = 0ms, Average = 0ms

1. Again check with the arp -a command

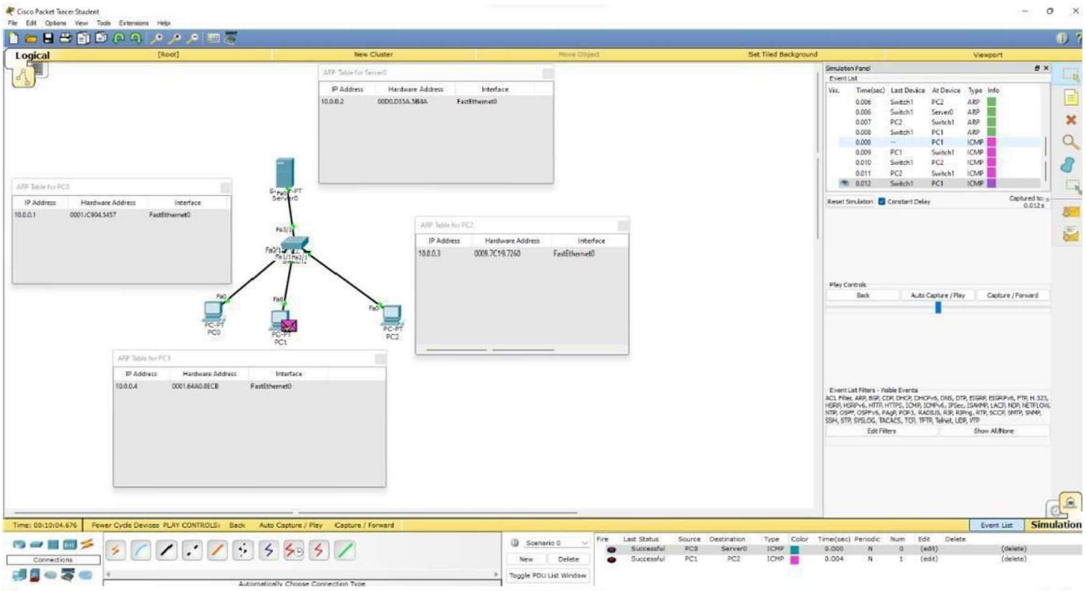
PC > arp -a

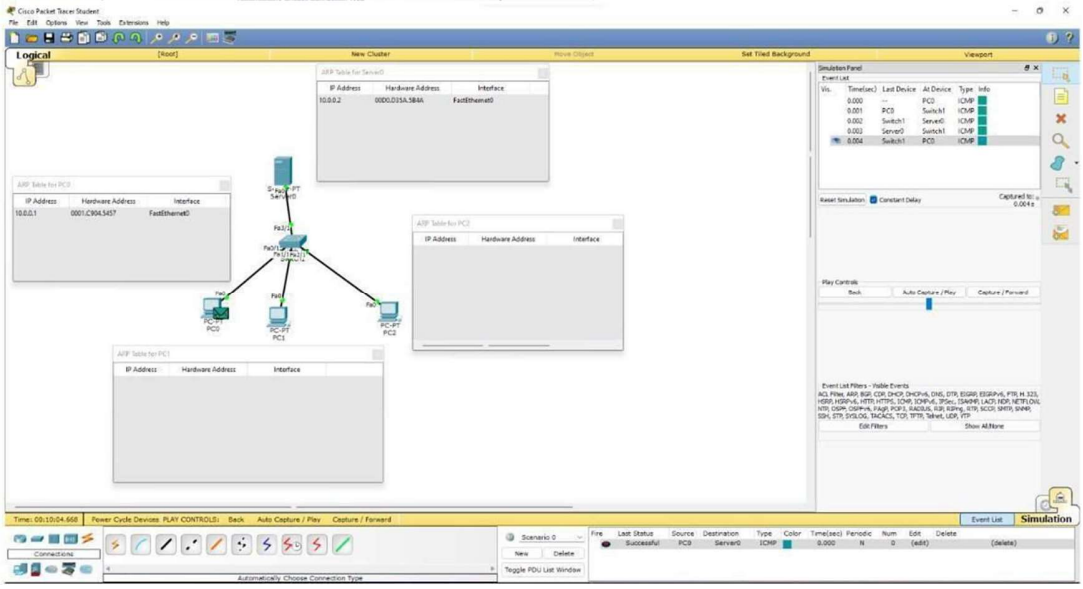
Internet address	Physical address	Type
10.0.0.3	0090.217C.1589	dynamic

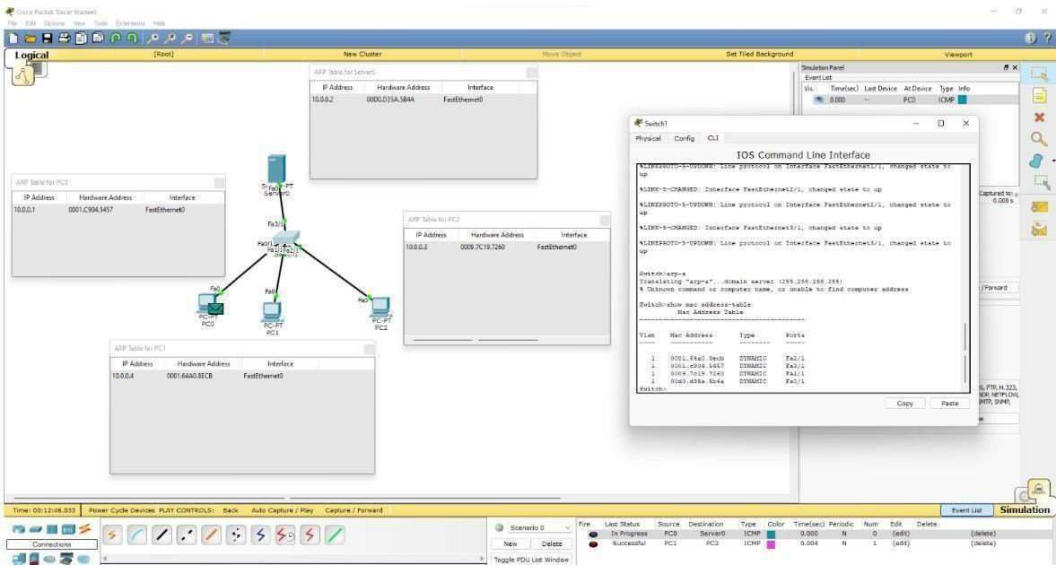
2. arp -a command is used to clear the table.

S.J
18/8/23

SCREENSHOTS:






Logical
IOS Command Line Interface
Switch1
Physical
Config
CLI
Copy
Paste
Simulation

CN LAB 9

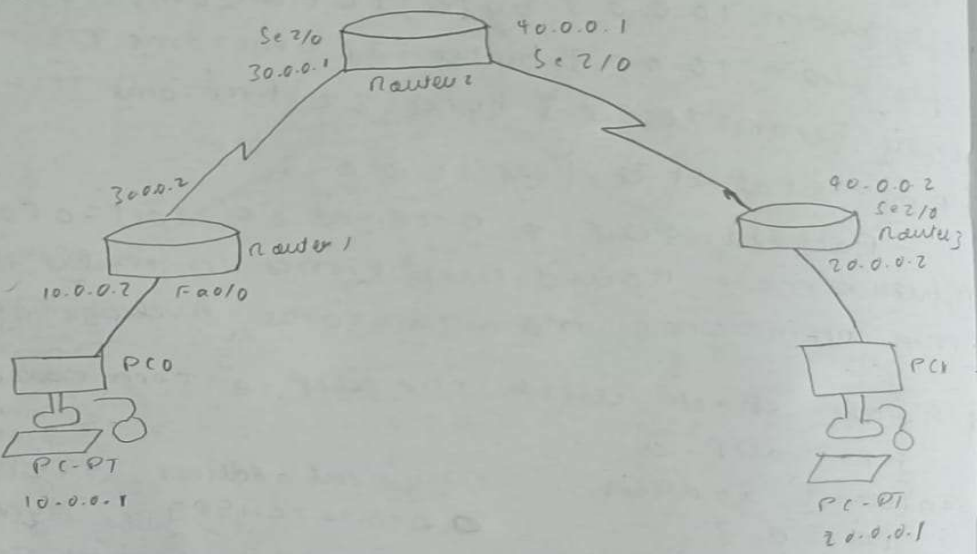
AIM: Demonstrate the TTL/ Life of a Packet.

OBSERVATION:

Experiment 9. Demonstrate the TTL/ life of a packet

Aim : Demonstrating the TTL

Topology :



Procedure :

→ Create a topology as seen above

→ Configure the device as per static / default / dynamic routing (same as CAB3))

→ In the simulation mode send a simple PDU from one PC to another.

→ use capture button to capture every transfer

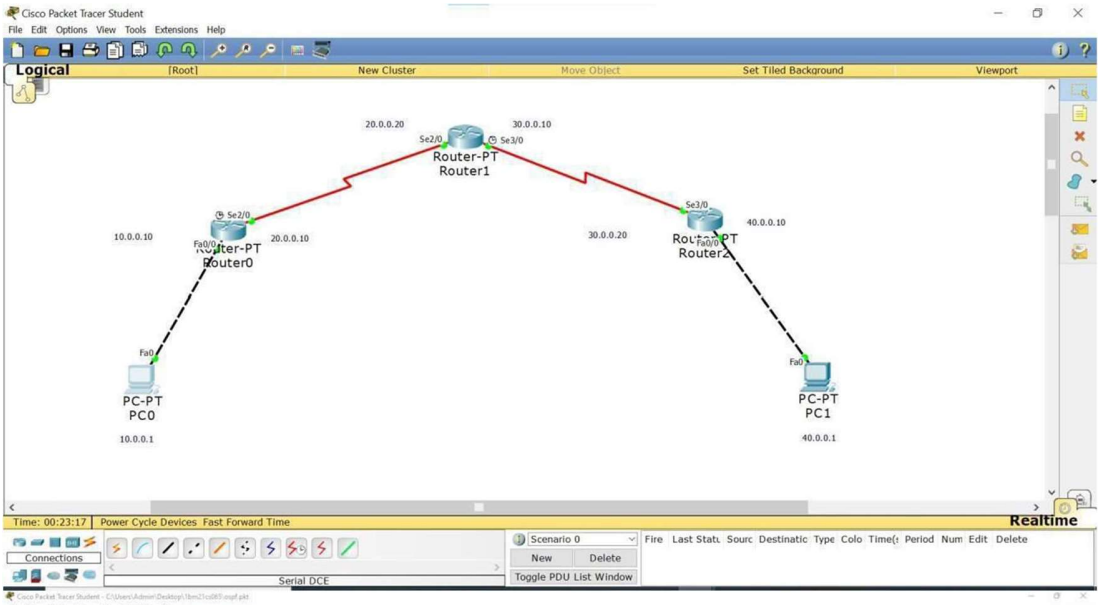
→ Click on the PDU during every transfer to see the Inbound & Outbound PDU details.

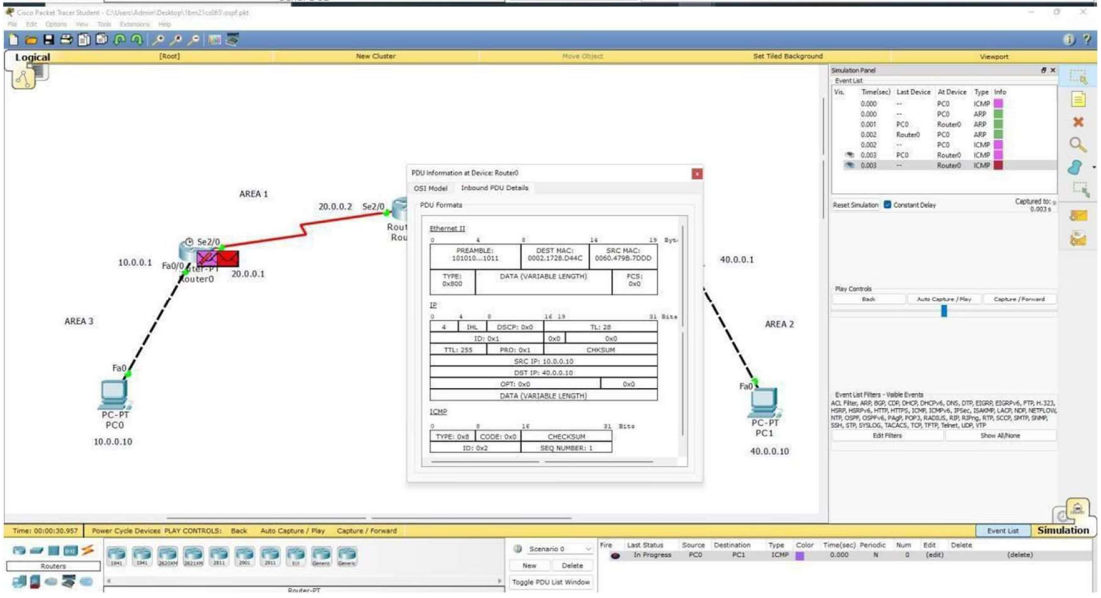
→ Observe that there is a difference of 1 in TTL when it cross every router.

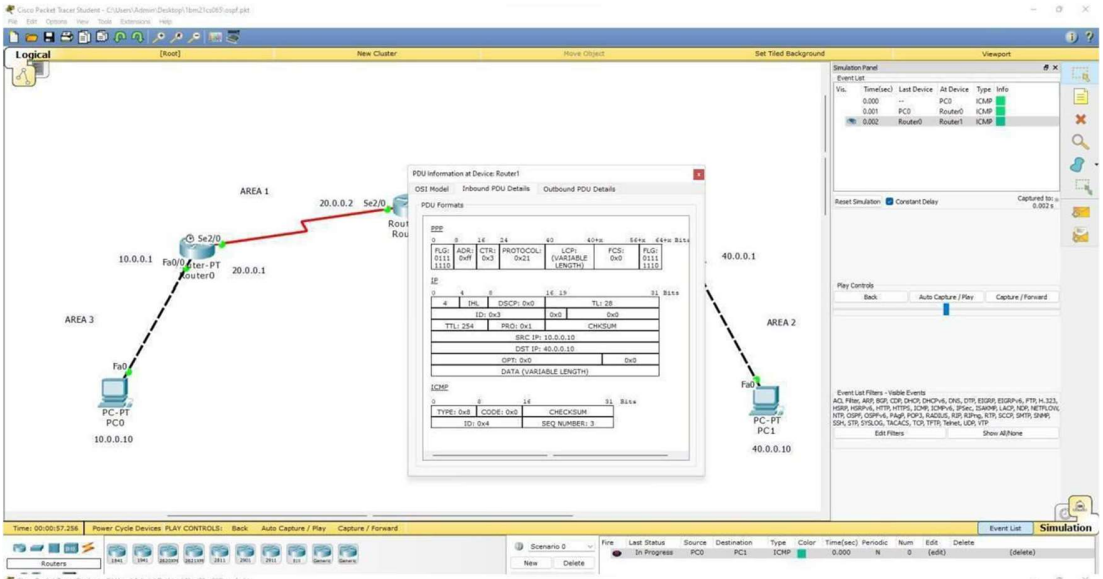
Observation & Result :-

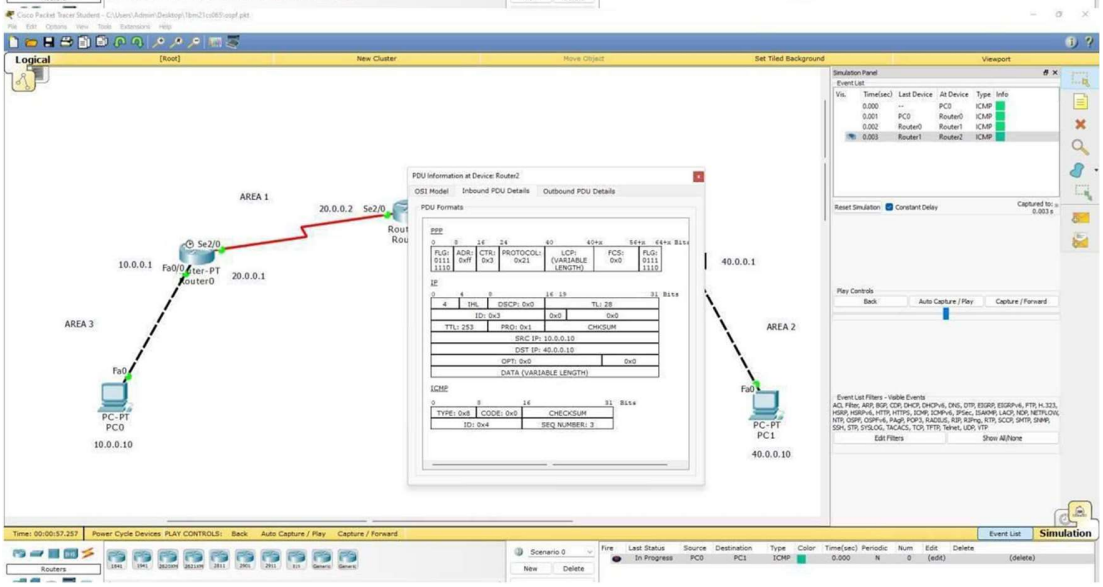
Difference of 1 in TTL when the POV crosses every router.

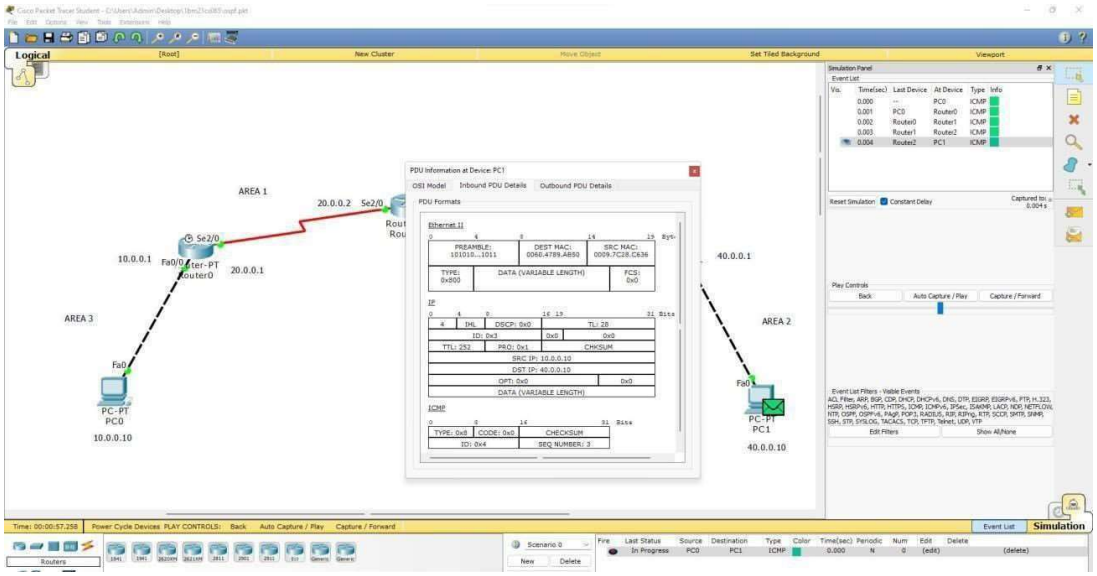
SCREENSHOTS:






Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
AREA 1
AREA 2
AREA 3
20.0.0.2
Se2/0
10.0.0.1
Fa0/0
Router-PT
Router0
20.0.0.1
Fa0
PC-PT
PC0
10.0.0.10
40.0.0.1
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
Simulation Panel
Event List
Reset Simulation
Play Controls
Edit Filters
Show All/None
Time: 00:00:57.256
Power Cycle Devices
PLAY CONTROLS:
Back
Auto Capture / Play
Capture / Forward
Event List
Simulation
Scenario 0
New
Delete


Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
AREA 1
AREA 2
AREA 3
20.0.0.2
Se2/0
10.0.0.1
Fa0/0
Router-PT
Router0
20.0.0.1
Fa0
PC-PT
PC0
10.0.0.10
40.0.0.1
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
Simulation Panel
Event List
Reset Simulation
Play Controls
Edit Filters
Show All/None
Time: 00:00:57.257
Power Cycle Devices
PLAY CONTROLS:
Back
Auto Capture / Play
Capture / Forward
Event List
Simulation
Scenario 0
New
Delete


Logical
AREA 1
20.0.0.2
Se2/0
10.0.0.1
Fa0/0
Router-PT
Router0
20.0.0.1
AREA 3
Fa0
PC-PT
PC0
10.0.0.10
40.0.0.1
AREA 2
Fa0
PC-PT
PC1
40.0.0.10
PDU Information at Device: PC1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Simulation Panel
Event List
Simulation

CN LAB 10

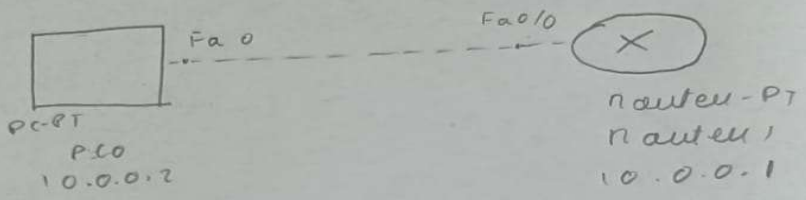
AIM: To understand the operation of TELNET by accessing the router in server room from a PC in IT office.

OBSERVATION:

Experiment - 10

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

Topolog:



Procedure:

1. Add a PC and a Router
   connect PC to Router-1 using copper cross-over wire from FastEthernet 0 port of PC to FastEthernet 0/0 port of router-1.
2. PC-1 configuration:
   Click on PC-1 → config → Interface → FastEthernet 0
   Assign a static IP address (10.0.0.2 and subnet mask 255.0.0.0)
3. Router-1 configuration
   Router> enable
   Router # config t
   Router(config) # hostname R1
   R1(config) # enable secret P1
   R1(config) # interface Fa 0/0
   R1(config-if) # ip address 10.0.0.1 255.0.0.0
   R1(config-if) # no shut

R1(config-if)# line vty 0 5

> login
> password Po
>exit
>exit
> wr - (To save changes in router)

Observation
PC> Ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data
Reply from 10.0.0.1 bytes=32 time=0ms TTL=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 ms
Minimum = 0ms, Maximum=0ms, Average=0ms
PC> telnet 10.0.0.1
Trying 10.0.0.1.. Open
User Access verification
Password:
% is enable:
Password:
R1# show IP route
codes C - connected, S-Static -1, IGRP
R-RIP m-Mobile
D - EIGRD, EX - EIGRD external,
O - OSPF - IA - OSPF

N1 - OSPF NSSA external type 1, N2 - External type 2
E1 - OSPF External type 1 - E2 - OSPF External type 2
i - IS-IS, L1 - IS-IS level-1 L2 - IS-IS level - 2
* - candidate default, U - Per-user static route
P - Periodic download static route.

Gateway of last resort is not set

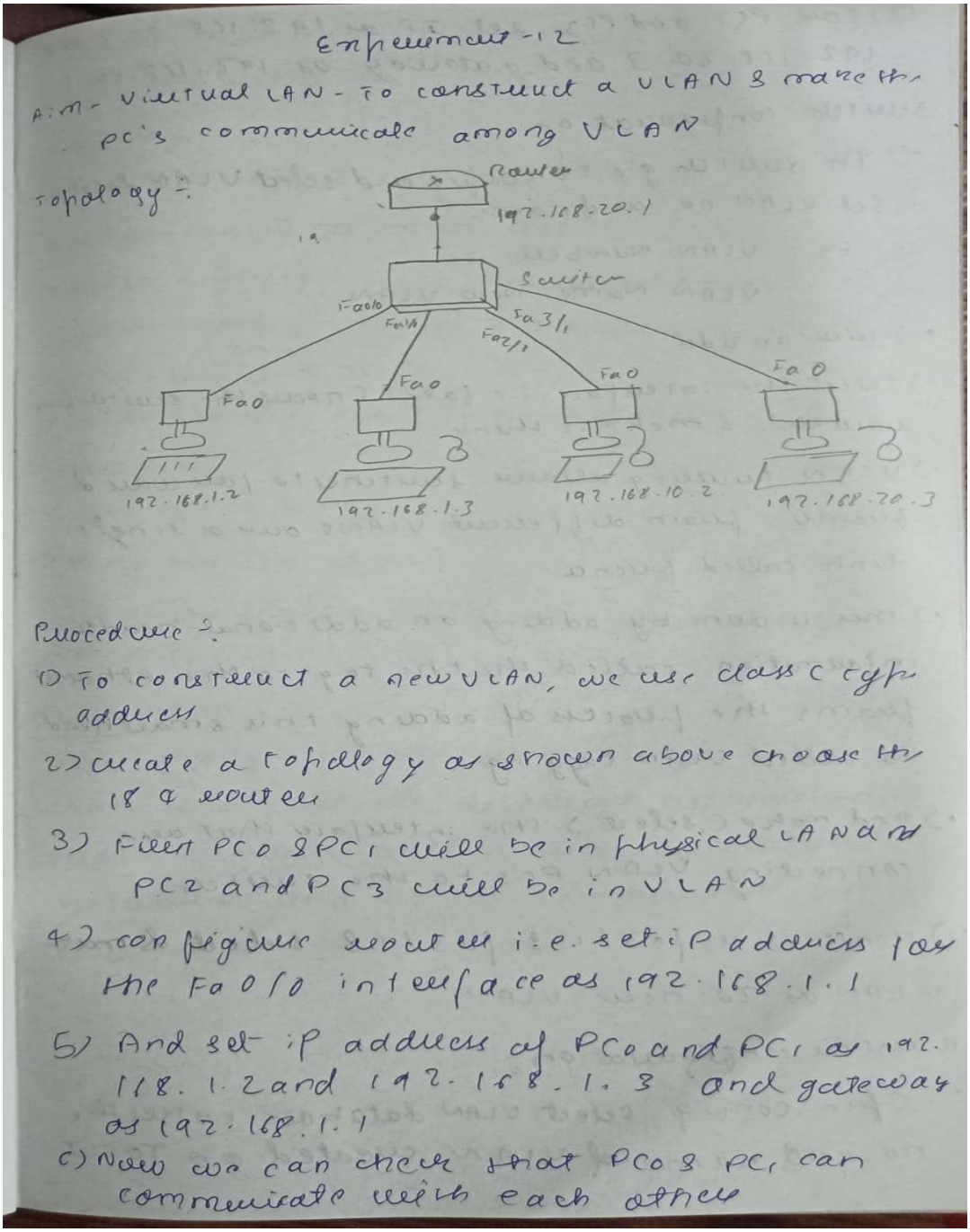
C 10.0.0.0/0 is directly connected,
FastEthernet 0/0

SCREENSHOTS:

CN LAB 11

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

OBSERVATION:



7) For PC2 and PC3 set IP as 192.168.20.2 and 192.168.20.3 and gateway as 192.168.20.1

switch configuration

→ In switch go to config and select VLAN database. Set VLAN no and name.

Eg: VLAN Number 20
VLAN Name New VLAN

-) click on add
-) select the interface i.e fa0/1 (near the switch from router) & make it think.
-) VLAN trunking allows switches to pass data frames from different VLANs over a single link called trunk.
-) This is done by adding an additional header information called the tag to the ethernet frame the process of adding this small head is called VLAN tagging.
-) And make (select) the interface that are connecting VLAN PC's to the switch.
-) Hence it is fa2/1 & fa3/1 & select & make VLAN as 20: new VLAN

Router configuration:

- Open config select VLAN database enter the no. and name of VLAN created go To CLI.

router(VLAN)# exit
Apply completed
exiting
Routing # config t
Router(config)# interface fa0/0
Router(config-subif)# encapsulation dot1Q 2
Router(config-subif)# ip address 192.168.20.1
255.255.255.0
Router(config-subif)# no shutdown

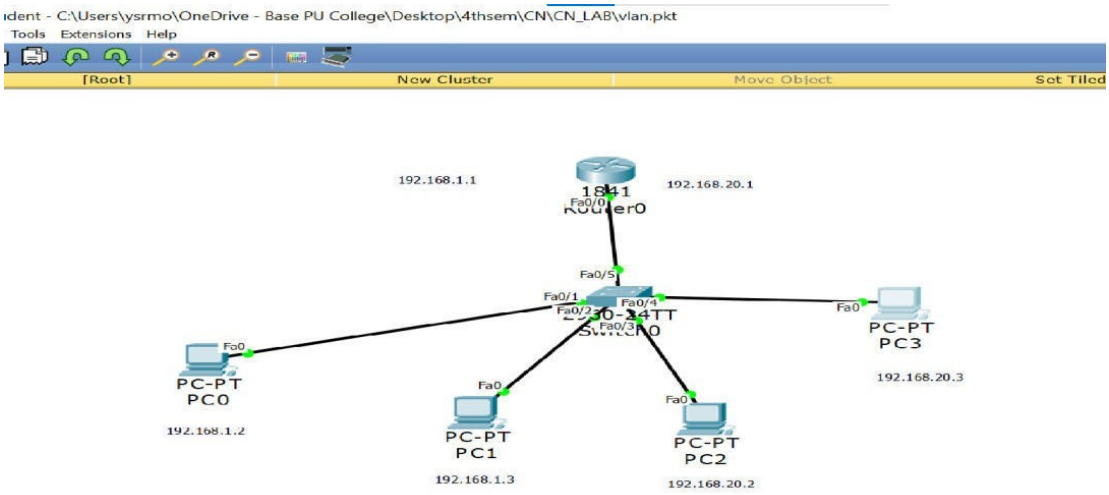
→ Now Ping
from PC0 to PC3

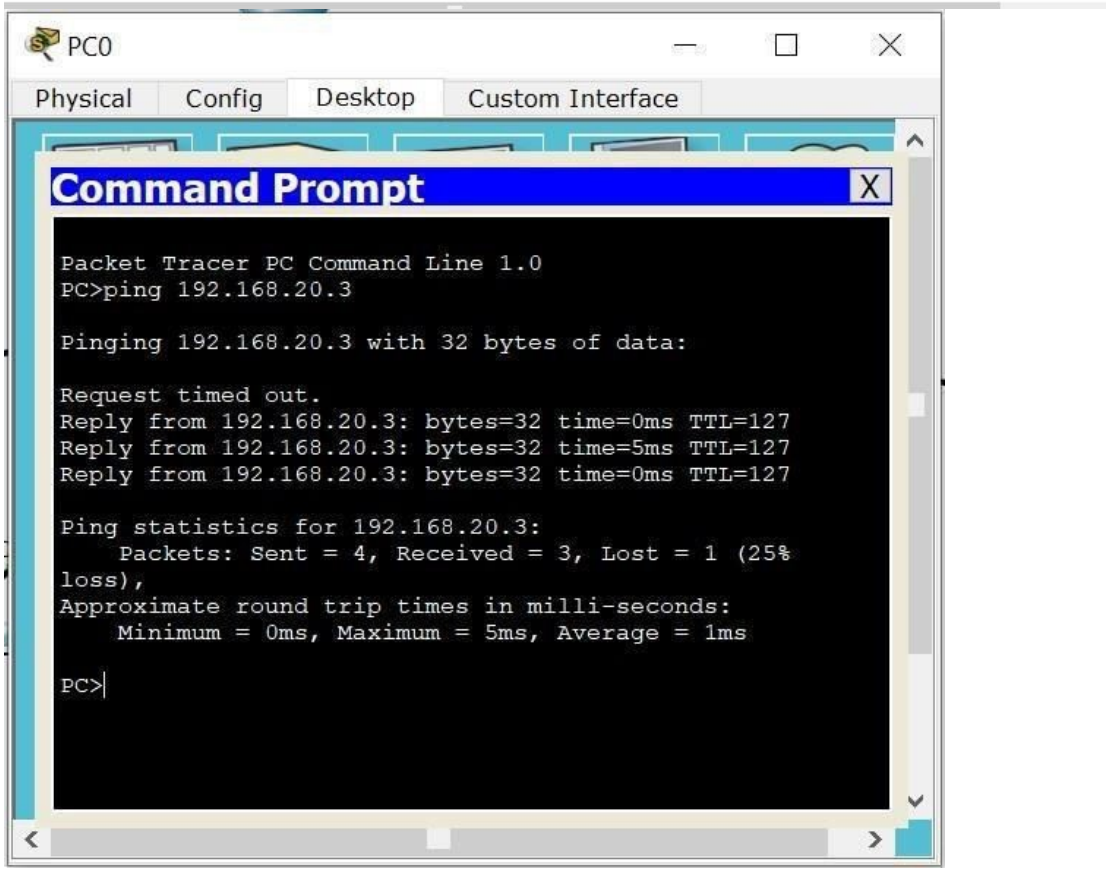
→ Ping 192.168.20.3
you will get a successful transmission from
PC0 to PC3

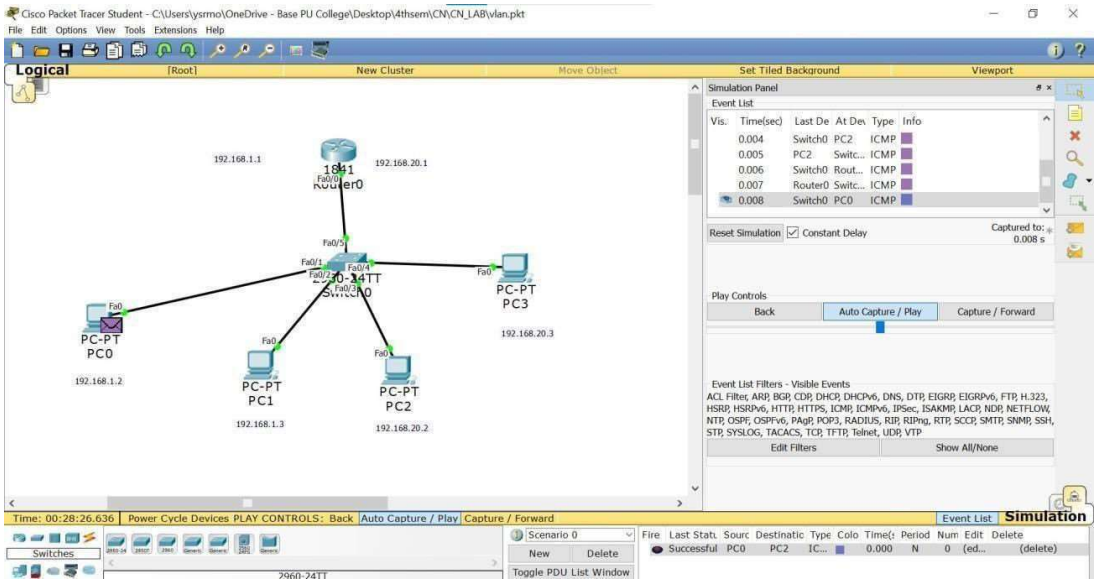
Observation

Even though we are using a single router we can use multiple different networks and those networks will works as virtual networks. And we can communicate from Physical LAN to VLAN and vice versa.

SCREENSHOTS:



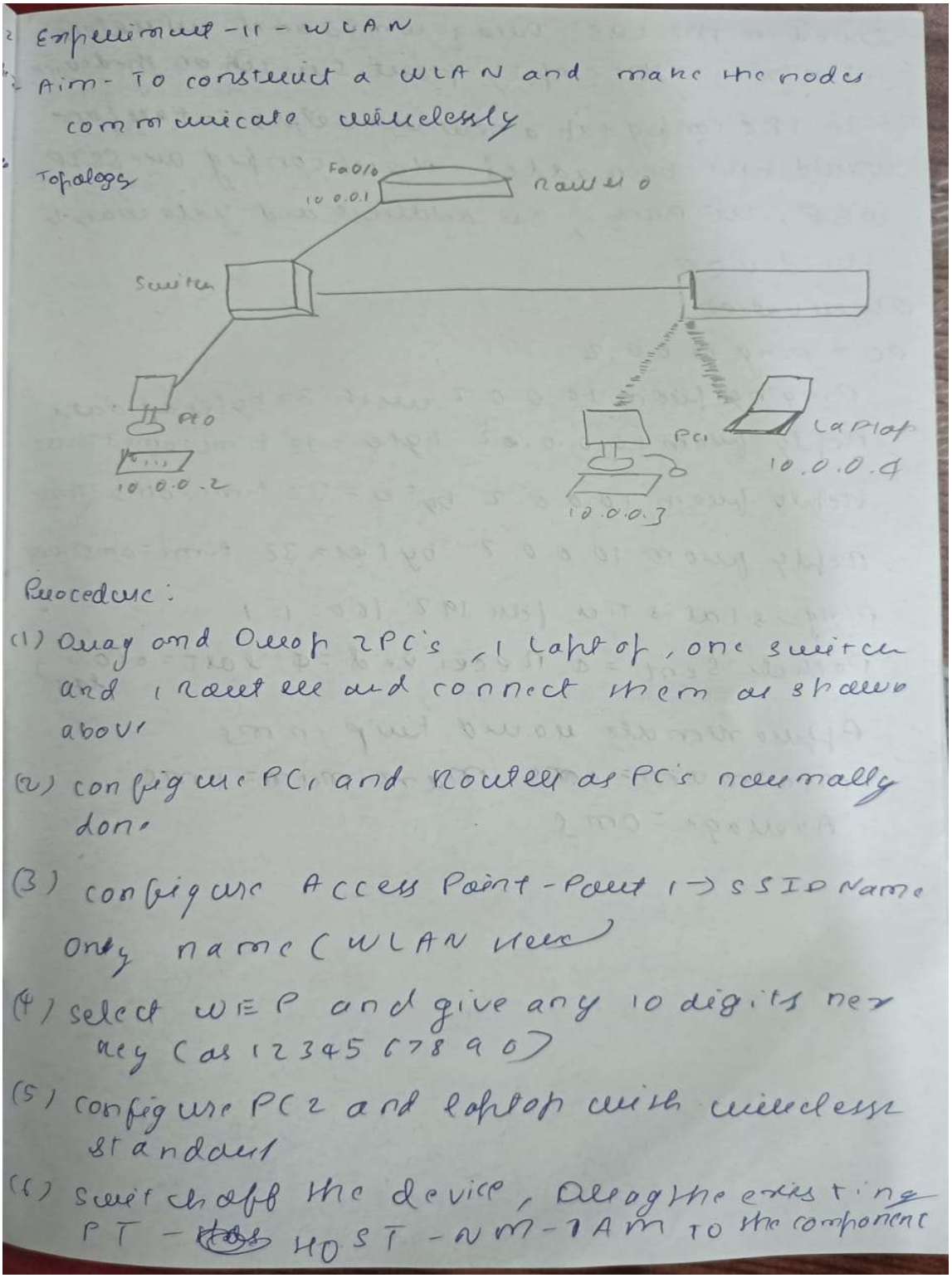



Cisco Packet Tracer Student - C:\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\CN\CN_LAB\vlan.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
192.168.1.1
192.168.20.1
Router0
Fa0/5
Fa0/1
Fa0/24
Fa0/3
Switch0
Fa0
PC-PT PC0
192.168.1.2
PC-PT PC1
192.168.1.3
PC-PT PC2
192.168.20.2
PC-PT PC3
192.168.20.3
Simulation Panel
Event List
Vis. Time(sec) Last De At De Type Info
0.004 Switch0 PC2 ICMP
0.005 PC2 Switc... ICMP
0.006 Switch0 Rout... ICMP
0.007 Router0 Switc... ICMP
0.008 Switch0 PC0 ICMP
Reset Simulation
Constant Delay
Captured to: 0.008 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: 00:28:26.636
Power Cycle Devices PLAY CONTROLS: Back
Auto Capture / Play
Capture / Forward
Event List
Simulation
Switches
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Stat. Sourc Destinatic Type Colo Time(: Period Num Edit Delete
Successful PC0 PC2 IC... 0.000 N 0 (ed... (delete)

CN LAB 12

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

OBSERVATION:



listed in the CHS. Drag WMP300N wireless interface to the empty front switch on the device.

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

Observation

PC > ping 10.0.0.2

Pinging from 10.0.0.2 with 32 bytes of data

Reply from 10.0.0.2 byte = 32 time=0ms TTL=255

Reply from 10.0.0.2 byte = 32 time=0ms TTL=255

Reply from 10.0.0.2 bytes = 32 time=0ms TTL=255

Ping statistics for 192.168.1.1

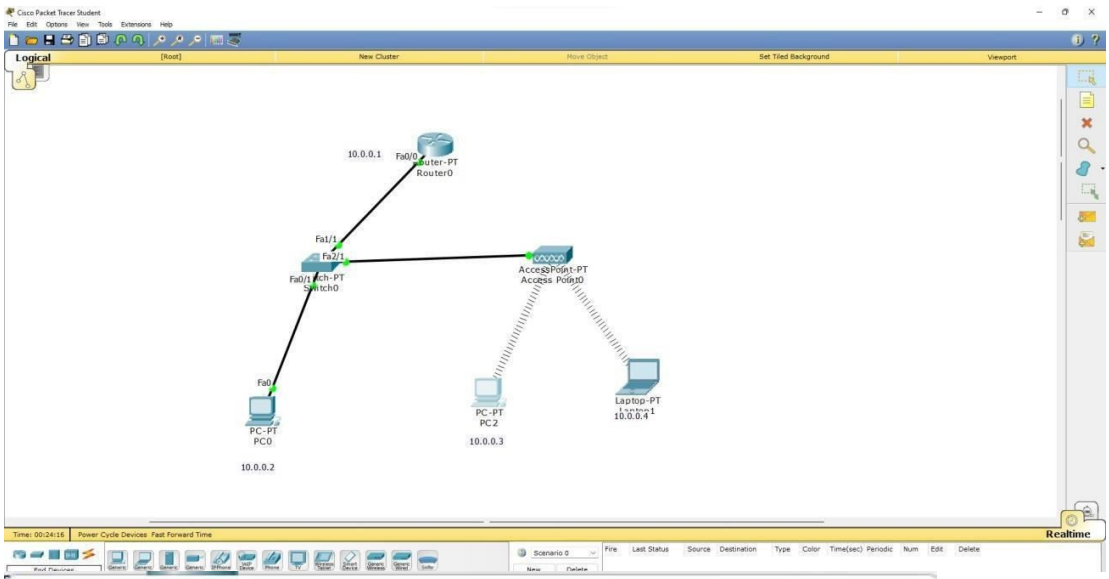
Packets Sent = 4 Received = 4 lost = 0 (0% loss)

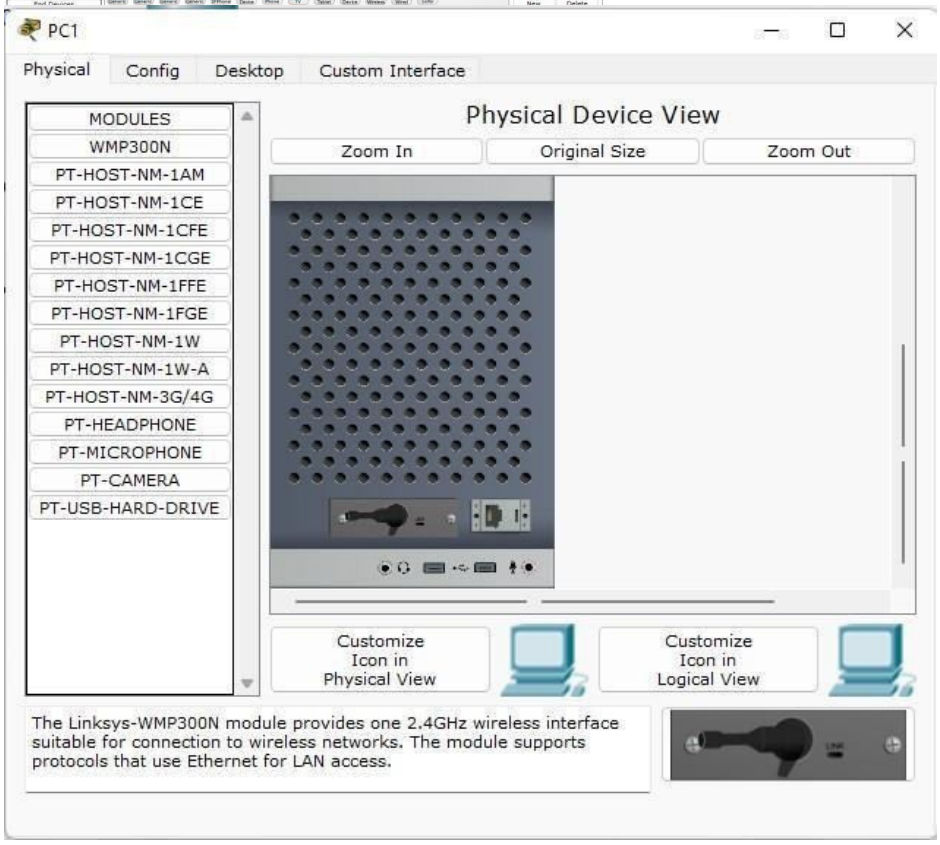
Approximate round trip in ms

minimum = 0ms Maximum = 1ms

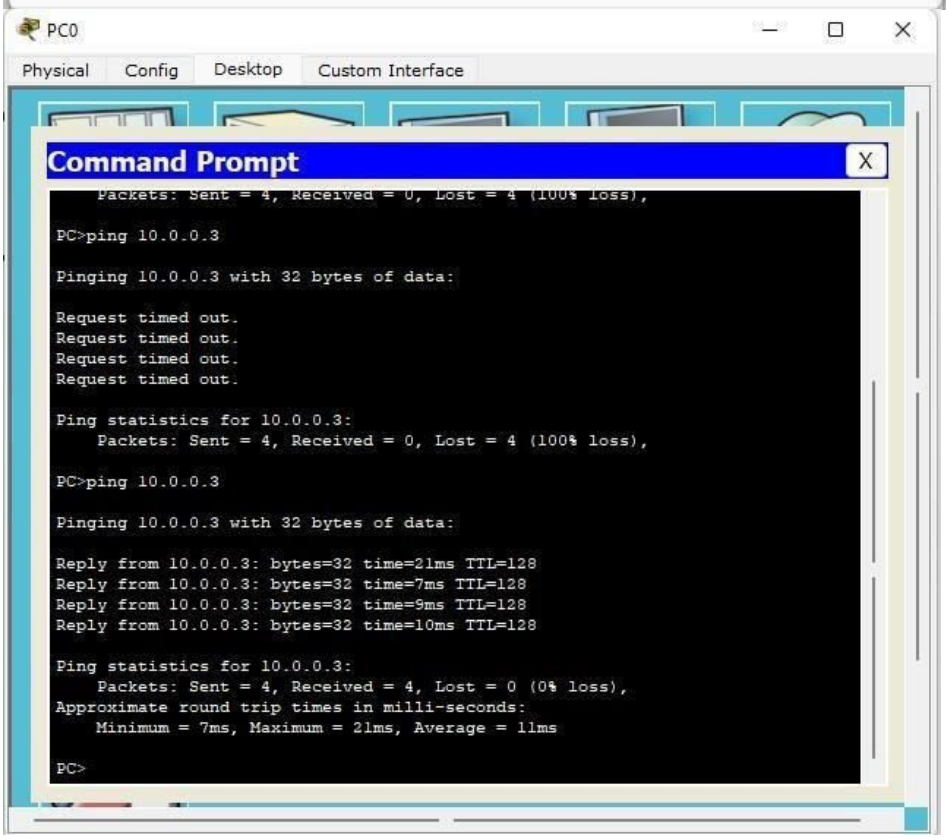
Average = 0ms

SCREENSHOTS:





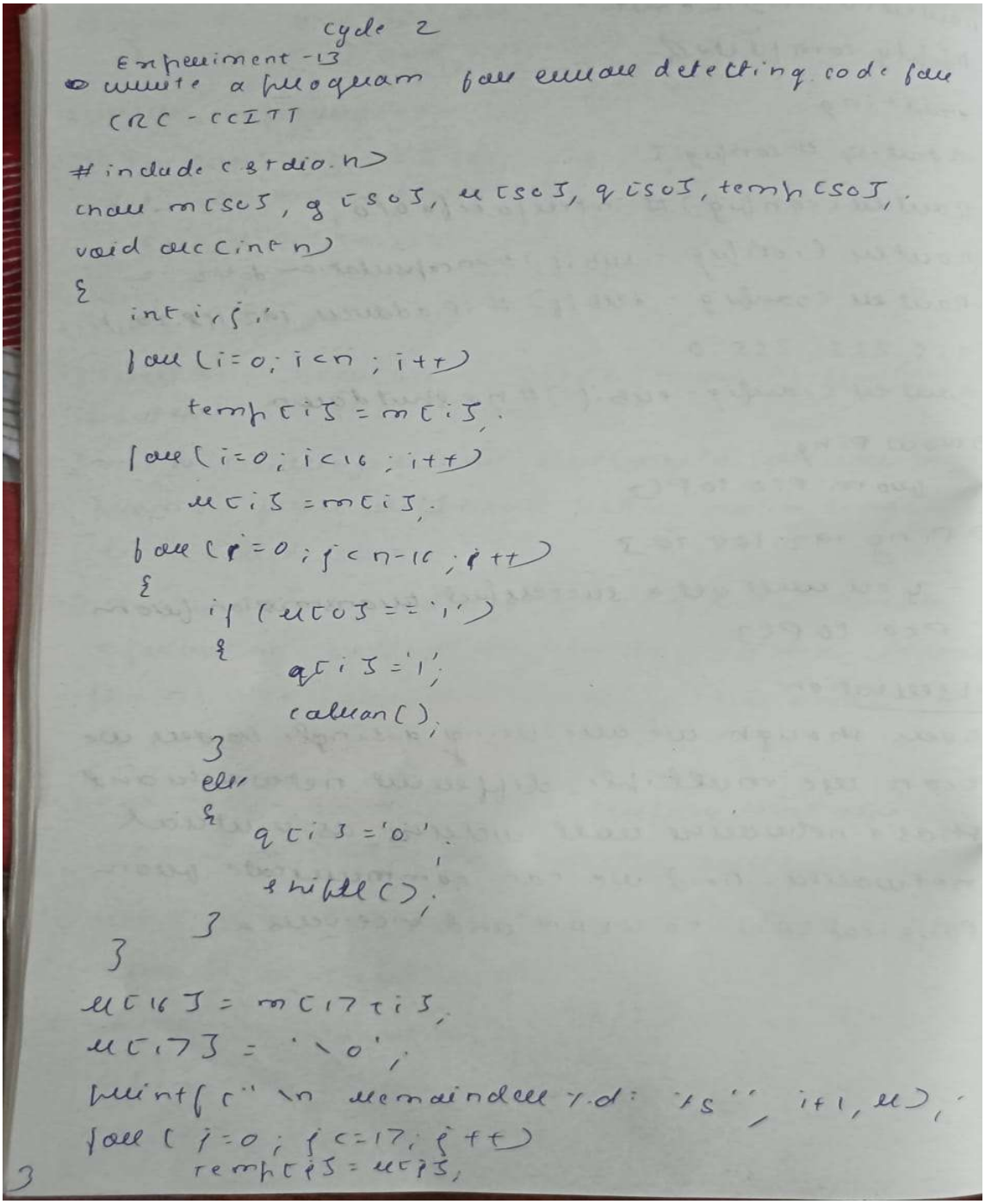

Laptop0
Physical
Config
Desktop
Custom Interface
MODULES
WPC300N
PT-LAPTOP-NM-1AM
PT-LAPTOP-NM-1CE
PT-LAPTOP-NM-1CFE
PT-LAPTOP-NM-1CGE
PT-LAPTOP-NM-1FFE
PT-LAPTOP-NM-1FGE
PT-LAPTOP-NM-1W
PT-LAPTOP-NM-1W-A
PT-LAPTOP-NM-3G/4G
PT-HEADPHONE
PT-MICROPHONE
PT-CAMERA
PT-USB-HARD-DRIVE
Physical Device View
Zoom 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.


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>

CN LAB 13

AIM: Write a program for error detecting code using CRCCCITT (16-bits).

OBSERVATION:



```
                    cycle 2
Experiment -13
• write a program for error detecting code for
  CRC - CCITT

#include <stdio.h>
char m[50], g[50], a[50], q[50], temp[50];
void calc(int n)
{
   int i, j;
   for(i=0; i<n; i++)
      temp[i] = m[i];
   for(i=0; i<16; i++)
      a[i] = m[i];
   for(j=0; j<n-16; j++)
   {
      if(a[0] == '1')
      {
         q[j] = '1';
         caltxor();
      }
      else
      {
         q[j] = '0';
         shiftl();
      }
   }
a[16] = m[17+j];
a[17] = '\0';
printf("\n remainder %d: %s", j+1, a);
for(j=0; j<=17; j++)
      temp[j] = a[j];
}
```

```
q[n-1] = '\0';
}
void callam()
{
    int i,j;
    for(i=1;i<16;i++)
        u[i-1] = ((int)temp[i]-48)^((int)q[i]-
                                    48) + 48;
}
void shift()
{
    int i;
    for(i=1;i<=16;i++)
        u[i-1] = u[i];
}
void calfulams(int n)
{
    int i,k=0;
    for(i=n-16;i<n;i++)
        m[i] = ((int)m[i]-48)^((int)u[k++]-48)+
                48;
        m[i] = '\0';
}
int main()
{
    int n,i=0;
    char ch,flag=0;
    printf("Enter plaintext:");
    while((ch=getchar())!='\n')
        m[i++] = ch;
```

```
n=i;
for(i=0;i<b;i++)
  m[n++]='0';
m[n]='\0';
printf("message after appending 16 zeros:%s",m);
for(i=0;i<=16;i++)
   g[i]='0';
g[0]=g[4]=g[11]=g[16]=' ';
g[17]='\0';
printf("\n generator:%s\n",g);
rec(n);
printf("\n\n quotient:%s\n",q);
calcrem(n);
printf("\n transmitted frame:%s",m);
printf("\n Enter transmitted frame:");
scanf("\n%s",rm);
printf("CRC checking\n");
rec(n);
printf("\n\n last remainder:%s",e);
for(i=0;i<16;i++)
    if(e[i]!='0')
        flag=1;
    else
        continue;
```

```
if (flag == 1)
    printf("Error during transmission");
else printf("Received frame is correct");
}
```

Output

Enter frame bits : 1011

message after appending 16 zeroes :

1011 0000 0000 0000 0000

generator : 1000 1000 0001 0000 1

quotient : 10 11

transmitted : 1011 1011 0010 1101 0101 1

Enter transmitted frame

1011 1011 0001 0110 1011

Last remainder : 0000 0000 0000 0000

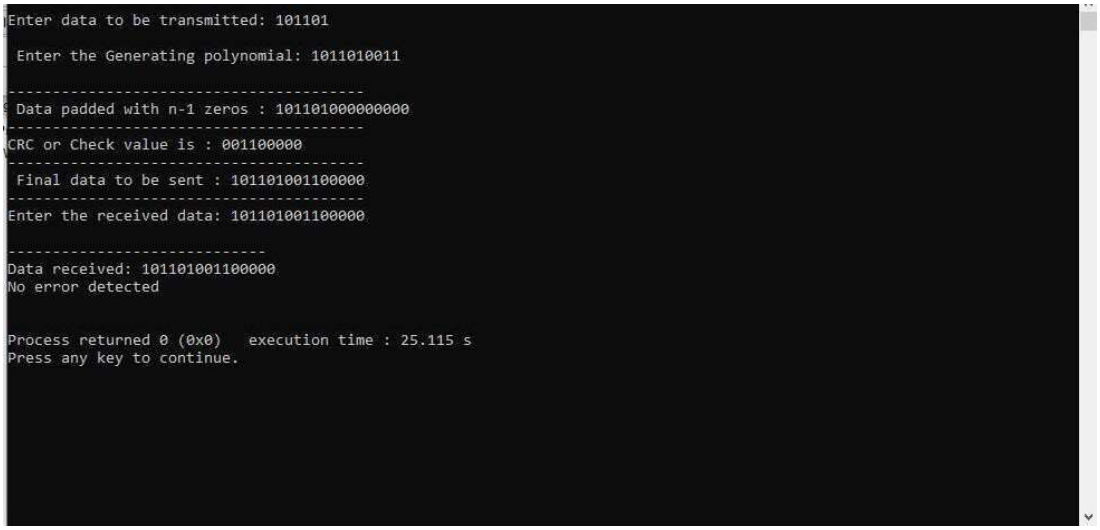
Received frame is correct.

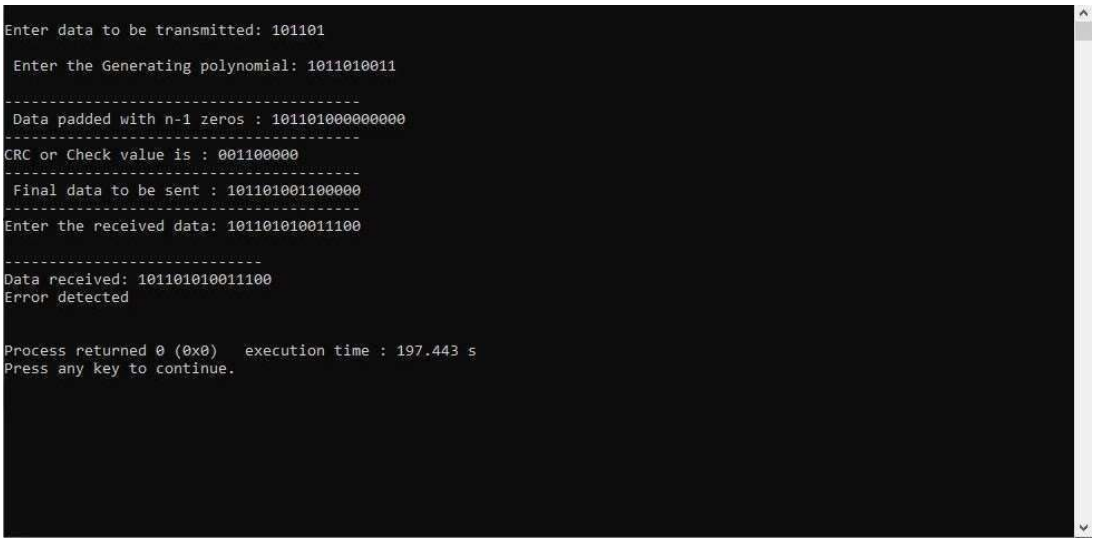
CODE:

```
 #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 receiver(){
printf("Enter the received data: "); scanf("%s", data);
printf("\n \n"); printf("Data received: %s", data); crc();
for(i=0;(i<N-1)  &&  (check_value[i]!='1');i++);  if(i<N-1)
printf("\nError detected\n\n"); else
printf("\nNo error 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("\nEnter data to be transmitted: ");
 scanf("%s",data);
printf("\n Enter the Generating polynomial: ");
scanf("%s",gen_poly);
data_length=strlen(data);
for(i=data_length;i<data_length+N-1;i++)
data[i]='0';
printf("\n ");
printf("\n Data padded with n-1 zeros : %s",data);
printf("\n ");
crc();
printf("\nCRC or Check value is : %s",check_value);
for(i=data_length;i<data_length+N-1;i++)
data[i]=check_value[i-data_length];
printf("\n ");
 printf("\n Final data to be sent : %s",data);
printf("\n \n");
receiver();
return 0;
}
```

OUTPUT

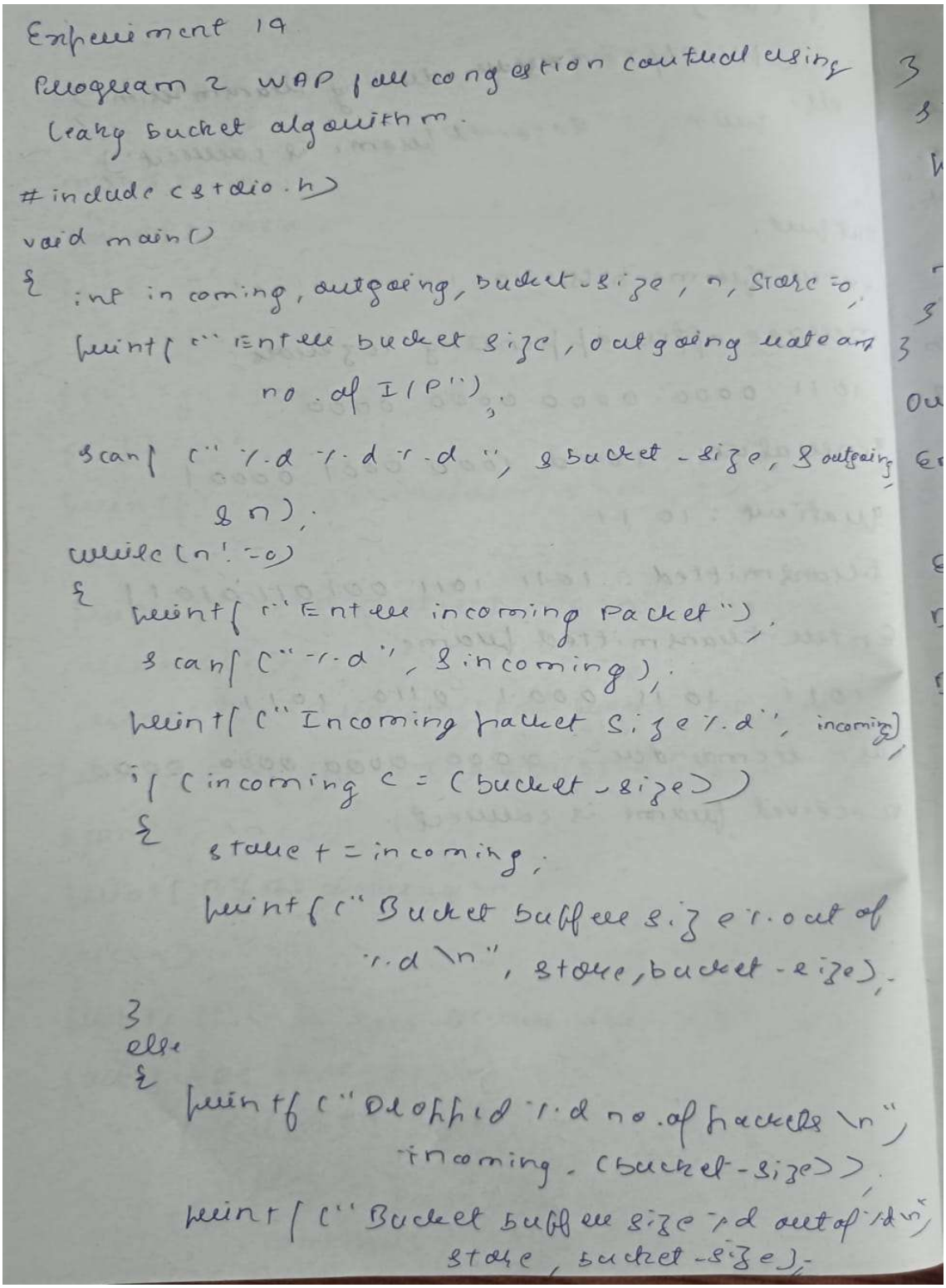

Enter data to be transmitted: 101101
Enter the Generating polynomial: 1011010011
Data padded with n-1 zeros : 101101000000000
CRC or Check value is : 001100000
Final data to be sent : 101101001100000
Enter the received data: 101101001100000
Data received: 101101001100000
No error detected
Process returned 0 (0x0) execution time : 25.115 s
Press any key to continue.


Enter data to be transmitted: 101101
Enter the Generating polynomial: 1011010011
Data padded with n-1 zeros : 101101000000000
CRC or Check value is : 001100000
Final data to be sent : 101101001100000
Enter the received data: 101101010011100
Data received: 101101010011100
Error detected
Process returned 0 (0x0) execution time : 197.443 s
Press any key to continue.

CNLAB 14

AIM: Write a program for congestion control using Leaky bucket algorithm**.**

OBSERVATION:



```
Experiment 19
Program 2 WAP for congestion control using
Leaky bucket algorithm.

#include<stdio.h>
void main()
{
  int incoming, outgoing, bucket_size, n, store=0;
  printf("Enter bucket size, outgoing rate and
          no. of I/P");
  scanf("%d %d %d", &bucket_size, &outgoing,
        &n);
  while(n!=0)
  {
    printf("Enter incoming packet");
    scanf("%d", &incoming);
    printf("Incoming packet size %d", incoming);
    if(incoming <= (bucket_size))
    {
      store += incoming;
      printf("Bucket buffer size %d out of
              %d\n", store, bucket_size);
    }
    else
    {
      printf("Dropped %d no. of packets\n",
              incoming - (bucket_size));
      printf("Bucket buffer size %d out of %d\n",
              store, bucket_size);
```

```
store = bucket_size;
}
store = store - outgoing;
printf("After outgoing %d packets left out
        %d in buffer\n", store, bucket_size);
n--;
}
}
```

Output

Enter bucket size, outgoing rate and no. of IP:
20 10 2

Enter incoming packet size = 30

Dropped 10 no of packet,

Bucket buffer size 0 out of 20

After outgoing 10 packet left out 20 in buffer

Enter incoming packet size = 10

Bucket buffer size 10 out of 20

After outgoing 10 packets left and 20 in buffer

CODE:

```
#include<stdio.h> void main()
{ int b_size,d_rate,in_d_rate,rem_b_size;
printf("Enter the bucket size:\n");
scanf("%d",&b_size);
rem_b_size=b_size;
printf("Enter the outgoing data rate:\n");
scanf("%d",&d_rate); while(1) {
printf("Enter the size of incoming packet\n");
scanf("%d",&in_d_rate);
if(in_d_rate<=b_size)
{
if(in_d_rate<=rem_b_size)
{
 rem_b_size=rem_b_size-in_d_rate;
 rem_b_size=rem_b_size+d_rate;
 printf("Data packet is accepted\n");
printf("Remaining space in bucket is.....%d\n",rem_b_size);
printf("\n");
}
 else
{
printf("Data packet is dropped because the bucket size is less than the packet
size\n");
printf("\n");
}
}
}
}
```

OUTPUT

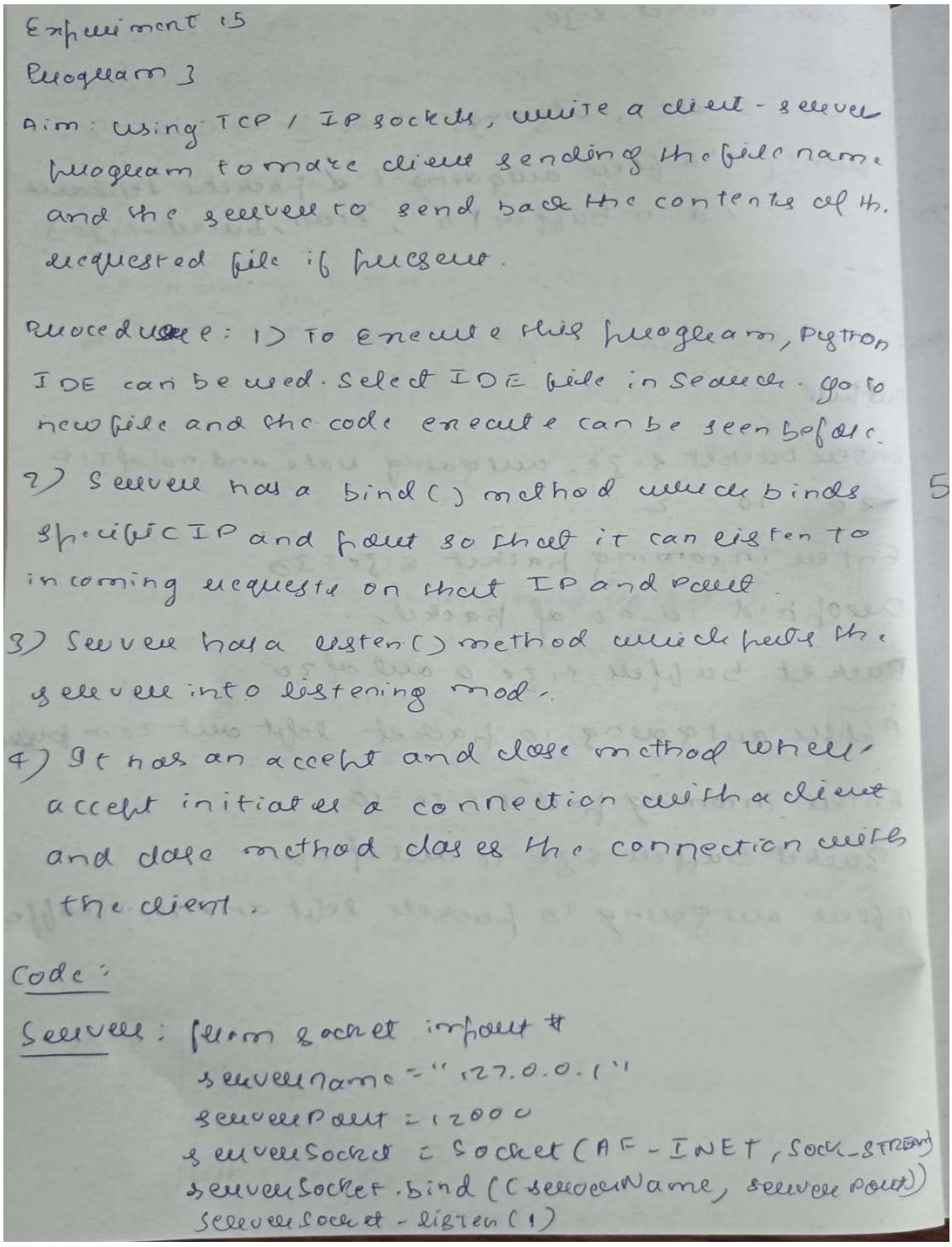
```
Enter the bucket size:
5000
Enter the outgoing data rate:
200
Enter the size of incoming packet
3000
Data packet is accepted
Remaining space in bucket is.... 2200

Enter the size of incoming packet
2500
Data packet is dropped because the bucket size is less than the packet size

Enter the size of incoming packet
```

CN LAB 15

Program 1

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.

Observation:



Experiment 15

Program 3

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 the requested file if present.

Procedure: 1) To execute this program, Python IDE can be used. Select IDE file in search. go to new file and the code execute can be seen before.

2) Server has a bind() method which binds specific IP and port so that it can listen to incoming requests on that IP and Port.

3) Server has a listen() method which puts the server into listening mod.

4) It has an accept and close method where accept initiates a connection with a client and close method closes the connection with the client.

Code:

Server:

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
```

while 1:

print ("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 ("\n sent contents of " + sentence)

file.close()

connectionSocket.close()

5) Next we apply client code

code:

client: from socket import *

serverPort = 12000

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("\n From Server \n")

print (filecontents)

client socket.close()

Observation

When we run the server code, the output displayed was "Server is ready to receive"
Then we run the client code
→ Enter file name: new.txt
→ From Server: The whole content was displayed

Output

Server: The server is ready to receive sent content of new.txt

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:

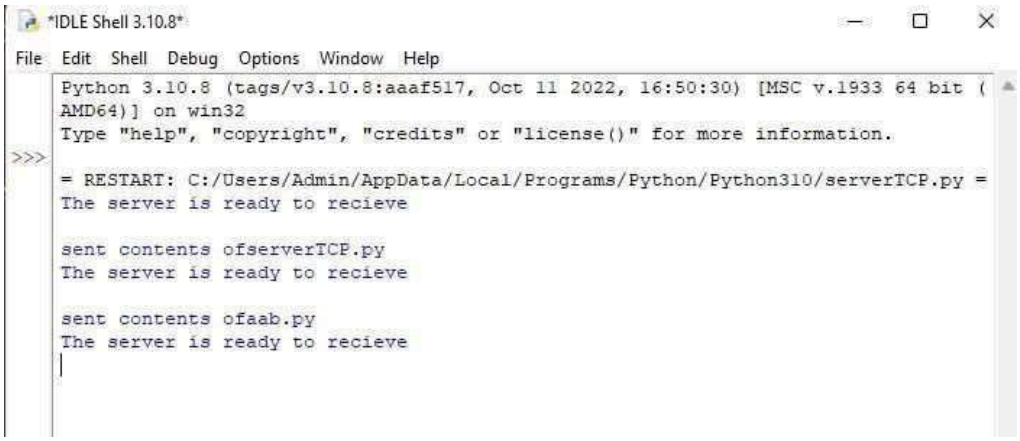
Client:

IDLE Shell 3.10.8

File Edit Shell Debug Options Window Help

```
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/clientTCP.py =

Enter file name:serverTCP.py

From Server:

from socket import *
serverName = "127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while(1):
    print("The server is ready to recieve")
    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()


>>>
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/clientTCP.py =

Enter file name:aab.py

From Server:

Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
class Node:
    def _init_(self,data):
        self.data=data
        self.left=None
        self.right=None
        self.height=1

class AVL Tree:
    def getHeight(self,root):
        if not root:
            return 0
        return root.height

    def getBalance(self,root):
        if not root:
            return 0
        return self.getHeight(root.left)-self.getHeight(root.right)

    def rightRotate(self,z):
        y=z.left
        T3=y.right

        y.right=z
        z.left=T3

        z.height=1+max(self.getHeight(z.left),self.getHeight(z.right))
        y.height=1+max(self.getHeight(y.left),self.getHeight(y.right))

        return y

    def insert(self,root,data):
        if not root:
            return Node(data)
        if data < root.data
            root.left=self.insert(root.left,data)
        else:
            root.right=se
>>>
```

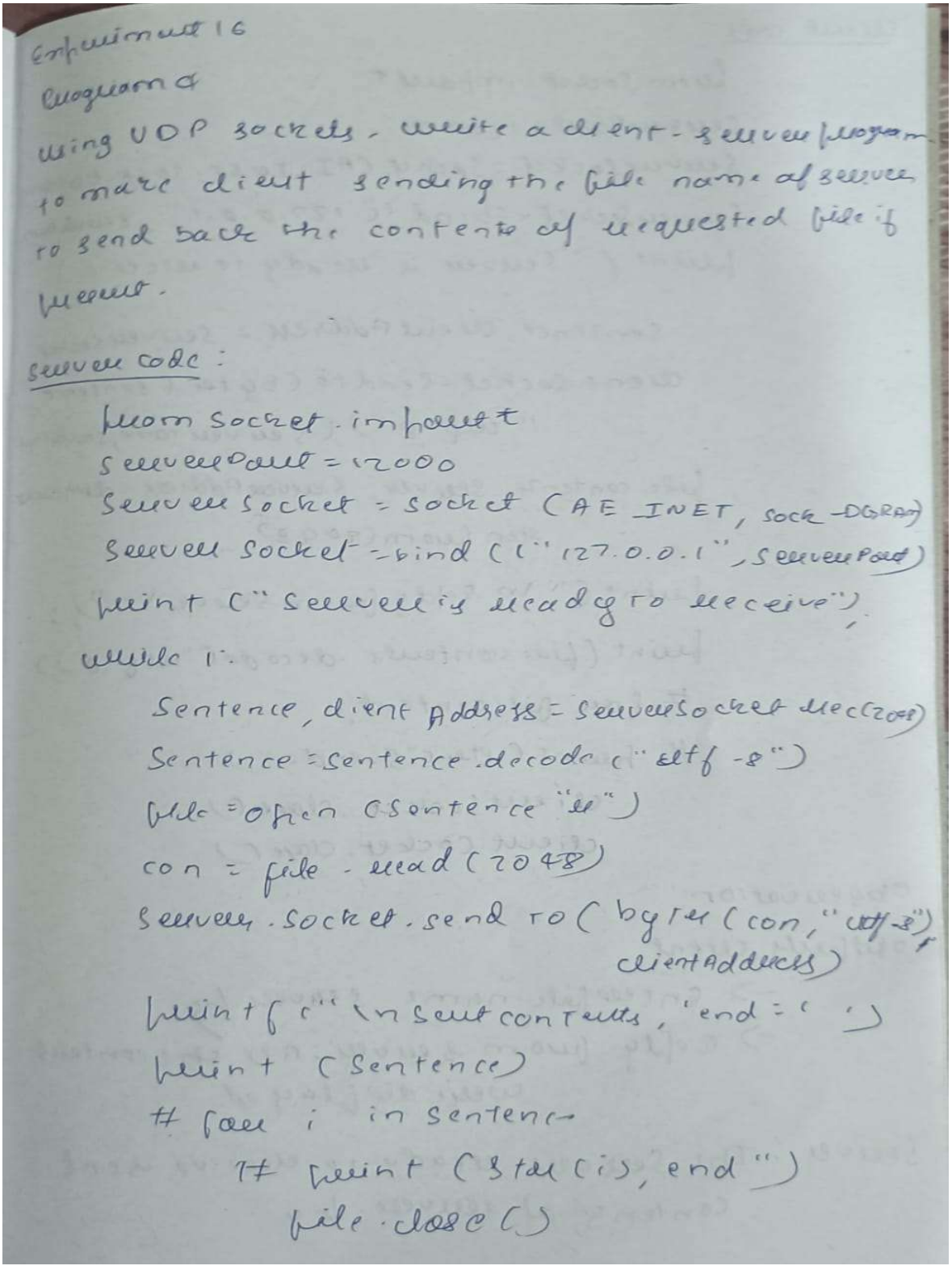
Server:


IDLE Shell 3.10.8
File Edit Shell Debug Options Window Help
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/serverTCP.py =
The server is ready to recieve
sent contents ofserverTCP.py
The server is ready to recieve
sent contents ofaab.py
The server is ready to recieve

Program 16

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.

Observation:



Experiment 16

Program a

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

Server code:

```
from socket import
serverPort = 12000
Server socket = socket (AF_INET, sock_DGRAM)
Server Socket = bind (("127.0.0.1", serverPort)
print ("Server is ready to receive");
while 1:
    Sentence, client Address = ServerSocket rec(2048)
    Sentence = Sentence.decode ("utf-8")
    file = open (Sentence "r")
    con = file . read (2048)
    Server . socket . send to ( byte (con, "utf-8"),
                                clientAddress)
    print f ("\n Sent contents, 'end = ' ')
    print (Sentence)
    # for i in Sentence
        # print (str(i), end ")
        file . close ()
```

client code

```
from socket import *
serverPort = 12000
serverSocket = socket(AI_INET, SOCK_DGRAM)
serverSocket = bind(("127.0.0.1", serverPort))
print("server is ready to receive")

    sentence, clientAddress = serverSocket
    clientSocket.sendto(bytes(sentence,
        "utf-8"),(servername, serverPort))

    file contents, server, serverAddress = clientSocket
        recvfrom(2048)
    print("\n reply from server \n");
    print(file contents.decode("utf-8"))
    # for file content;
    # print(str(i5.end=""))
        clientSocket.close()
        clientSocket.close()
```

Observation

Output: client

→ Enter file name: server.by

→ reply from server: All the contents were displayed

Server: The server ready to receive sent contents of server.by

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:

Client:

```
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/clientUDP.py =

Enter file name:  serverUDP.py

Reply from Server:

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

     serverSocket.sendto(bytes(con,"utf-8"),clientAddress)

     print ('\nSent contents of ', end = ' ')
     print (sentence)
    #  for i in sentence:
        #  print (str(i), end = '')
     file.close()

>>>
```

Server:

```
>>>
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/serverUDP.py =
The server is ready to receive

Sent contents of  serverUDP.py
```