VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

COMPUTER NETWORKS

Submitted by

BHARATH M(1BM22CS405)

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

JUN2023to 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 **BHARATH M(1BM22CS405),** who is a bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2023. The Lab report has been approved as it satisfies the academic requirements in respect of a **Computer Networks (22CS4PCCON)** work prescribed for the said degree.

Dr.Latha N R
Assistant Professor

Department of CSE
BMSCE, Bengaluru
`

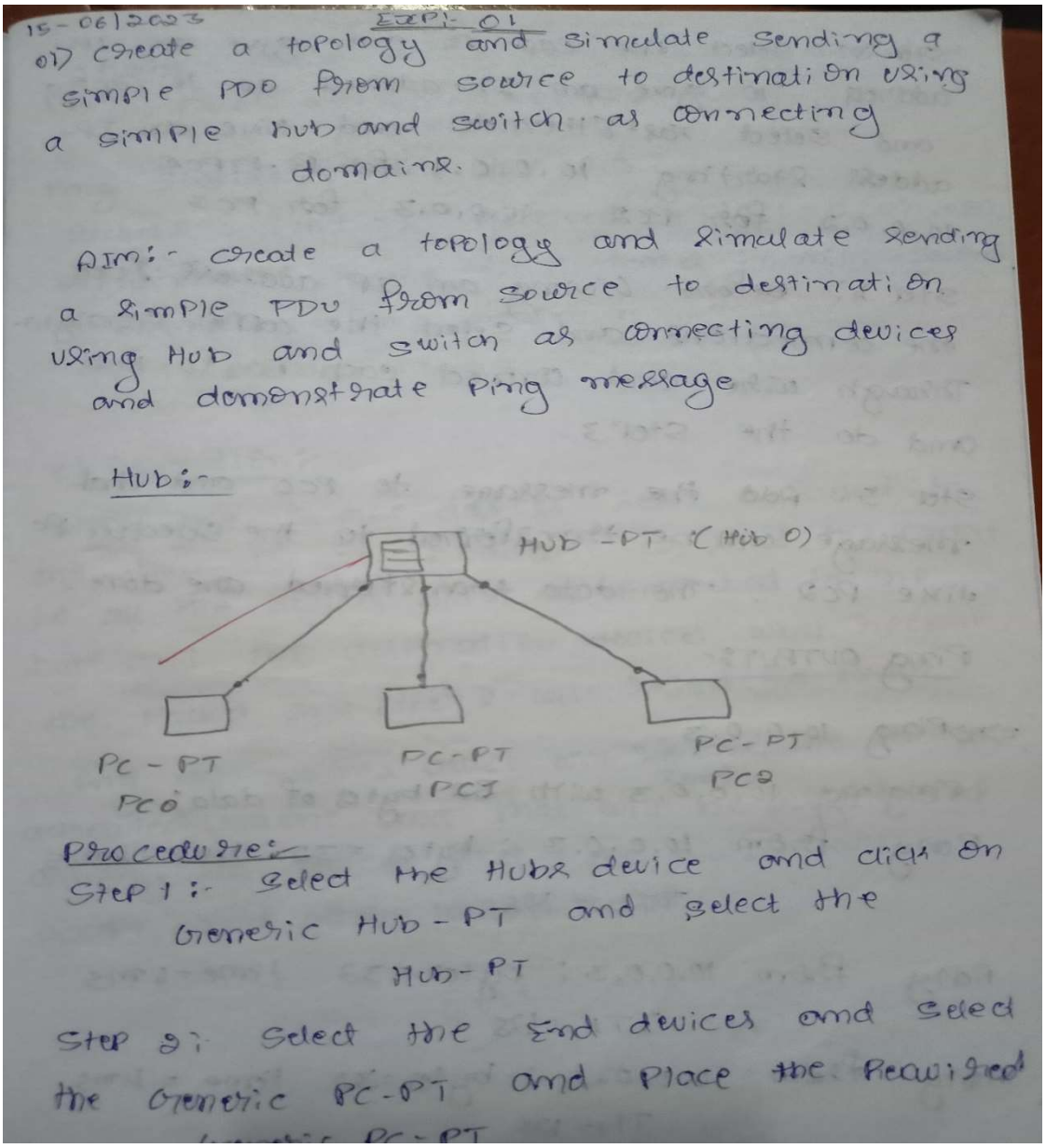
Dr. Jyothi S Nayak
Professor and
Head
Department of CSE
BMSCE, Bengaluru

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

LAB-1

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

OBSERVATION:



15-06/2023 Exp: 01

01) Create a topology and simulate sending a simple PDU from source to destination using a simple hub and switch as connecting domains.

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

Hub:-

Procedure:-

Step 1:- Select the Hubs device and click on Generic Hub-PT and select the Hub-PT

Step 2:- Select the End devices and select the Generic PC-PT and Place the Required

Step 3:- Select the each Pc and give an IP address, To give an IP-address go to config and select Fast Ethernet0 and Give an IP-address starting 10.0.0.1 for PC-PTPC0 10.0.0.2 for PC2 10.0.0.3 for PC3

Step 4:- Before Giving an IP address Select the connections and select the copper Straight-Through wire and connect each Pc to Hub. and do the Step 3.

Step 5:- Add the message to PC0 and that message will be transferred to the Selected Pc like PC2. The data transferred are done.

Ping OUTPUT:-

=> Ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:-

Reply from 10.0.0.3 ; bytes = 32 time = 4ms
TTL = 128

Reply from 10.0.0.3 ; bytes = 32 time = 4 ms
TTL = 128

Reply from 10.0.0.3 ; bytes = 32 time = 4ms
TTL = 128

Reply from 10.0.0.3: with 32 bytes of data
time = 4ms TTL = 128

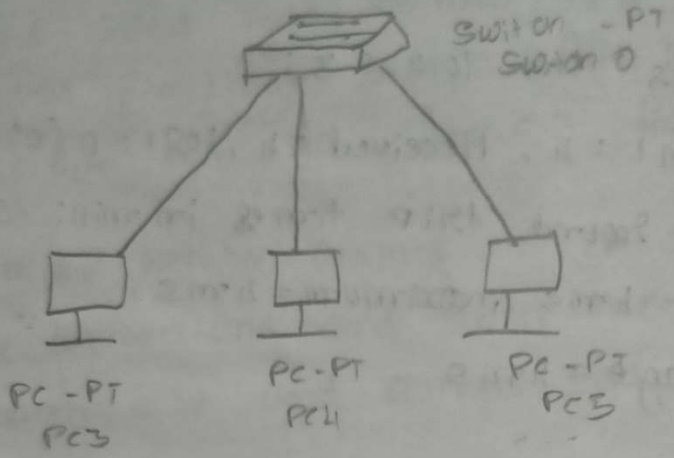
Ping statistics for 10.0.0.3:
Packets : Sent = 4, Received = 4, Lost = 0 (0% Loss),
Approximate round trip times in milli-seconds:
minimum = 4ms, maximum = 4ms,
average = 4ms

Observation:

When the source device sent a Packet to the hub it will bradcast or send the Packet to all the devices which all connected to the hub and the destination device will receive the Packet and other's will reject the Packet.

And destination's device will ~~receive~~ send the Packet acknowledgement and that will be distributed among all devices and the source will accept and other will discard.

Switch:



Procedure:-

Step 1:- Select the switch and 3 PC's

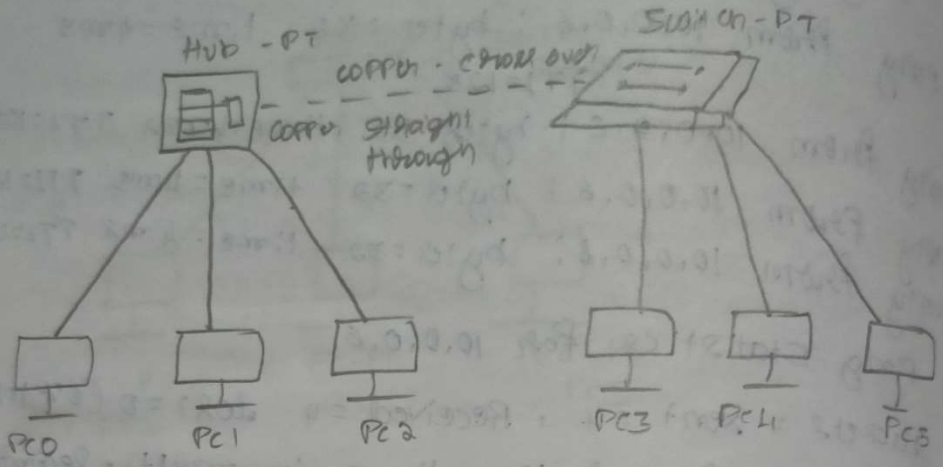
Step 2:- Connect the switch and 3 PC's with the copper straight - Through

Step 3:- Go to each PC and select the each PC and give an IP-address

Step 4:- Add Simple PDU(P) to one PC and that is the source PC and give an message to other PC that is destination PC.

Ping output:-

Hub - Switch connection:-



Step 1:- Previously drawn hub - topology and switch topology are connected through copper cross ones.

In Hub Port 3 is used in switch fast Ethernet 3.1 is used.

Step 2:- Add simple PDC from PDC to PC3

Ping output:

PC> Ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:

Reply from 10.0.0.4 ; bytes = 32 time = 1ms TTL = 128
Reply from 10.0.0.4 ; bytes = 32 time = 1 ms TTL = 128
Reply from 10.0.0.4 ; bytes = 32 time = 1ms TTL = 128
Reply from 10.0.0.4 ; bytes = 32 time = 1ms TTL = 128

Ping Satisfies for 10.0.0.4

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

Approximate round trip times in milli-seconds

minimum = 4ms maximum = 4ms

Average = 4ms

observation:-

In simulation mode PCD sends packet to hub

Hub sends it to PC1, PC2 and switch broadcasts it to PC3, PC4 and PC5

PC1 PC2, PC4 and PC5 discards item.

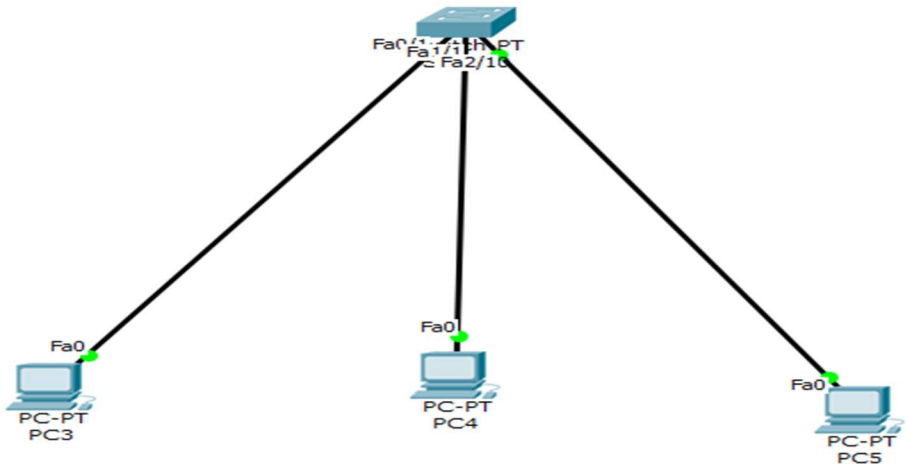
PC3 accepts and sends acknowledgement to hub through switch

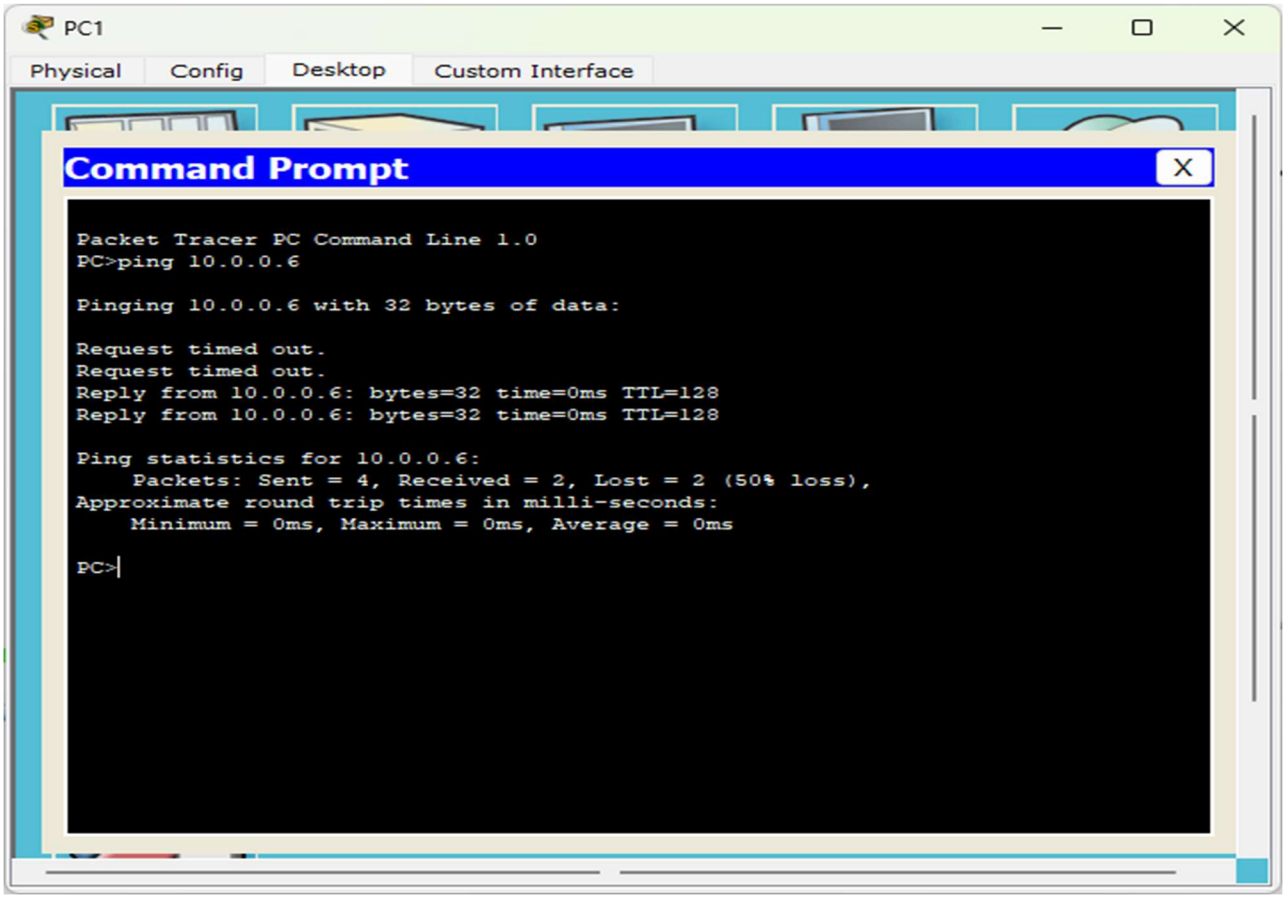
Hub is broadcasts it to all 3 PCs only PCD accepts it and others derived

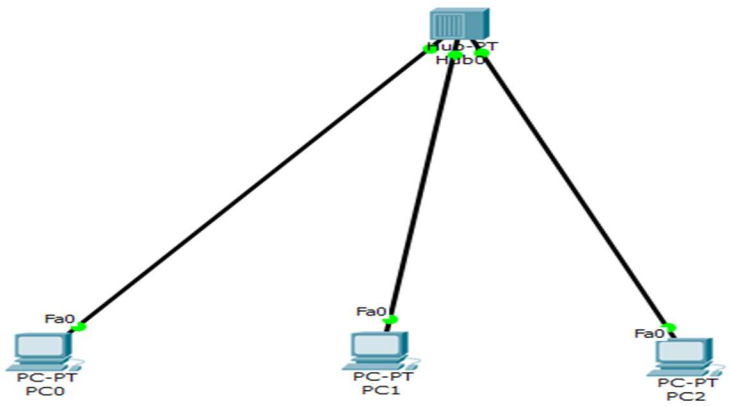
In second round PCD sends packet to HUB it, broadcasted to PC1, PC1 switch now switch broadcasts it only to PC3; Thus switch is smart device.

18/6/23

OUTPUT :


Fa2/16
Fa0
PC-PT
PC3
Fa0
PC-PT
PC4
Fa0
PC-PT
PC5


PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.6
Pinging 10.0.0.6 with 32 bytes of data:
Request timed out.
Request timed out.
Reply from 10.0.0.6: bytes=32 time=0ms TTL=128
Reply from 10.0.0.6: bytes=32 time=0ms TTL=128
Ping statistics for 10.0.0.6:
Packets: Sent = 4, Received = 2, Lost = 2 (50% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>

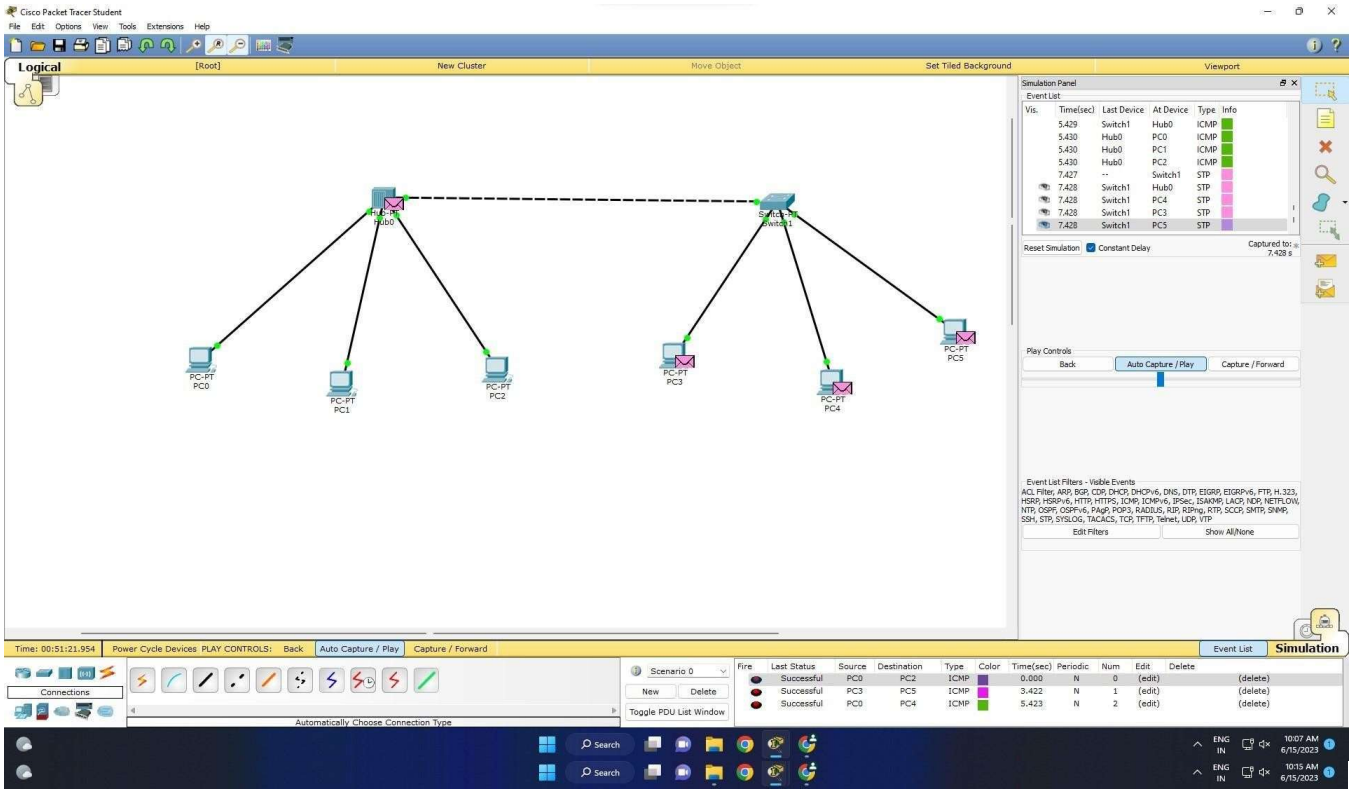


PC3

Physical | Config | Desktop | Custom Interface

Command Prompt

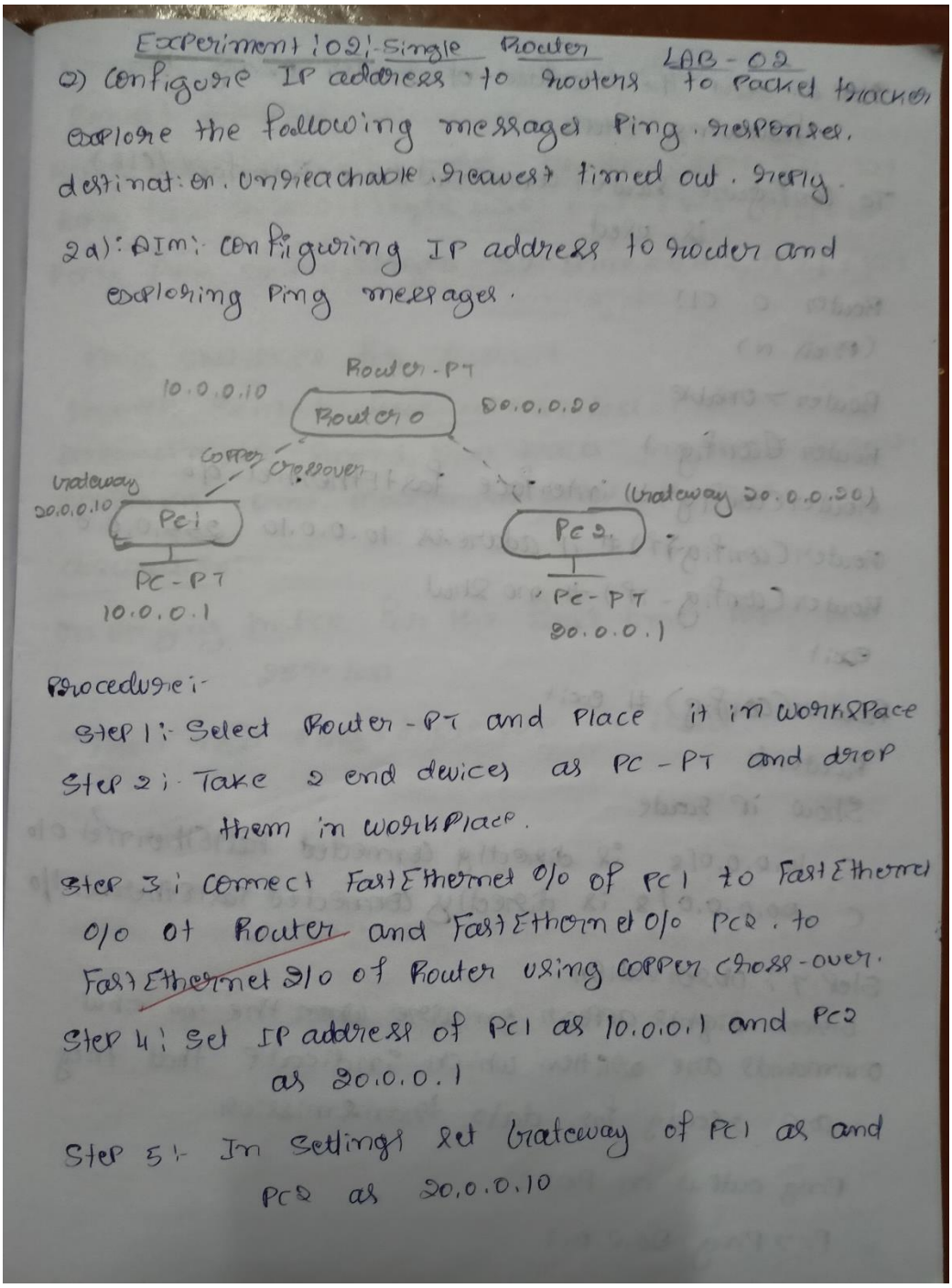
```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.6

Pinging 10.0.0.6 with 32 bytes of data:

Reply from 10.0.0.6: bytes=32 time=0ms TTL=128
Reply from 10.0.0.6: bytes=32 time=0ms TTL=128
Reply from 10.0.0.6: bytes=32 time=0ms TTL=128
Reply from 10.0.0.6: bytes=32 time=0ms TTL=128

Ping statistics for 10.0.0.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>
```


Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Hub0
Switch1
PC-PT PC0
PC-PT PC1
PC-PT PC2
PC-PT PC3
PC-PT PC4
PC-PT PC5
Simulation Panel
Event List
Vis. Time(sec) Last Device At Device Type Info
5.429 Switch1 Hub0 ICMP
5.430 Hub0 PC0 ICMP
5.430 Hub0 PC1 ICMP
5.430 Hub0 PC2 ICMP
7.427 -- Switch1 STP
7.428 Switch1 Hub0 STP
7.428 Switch1 PC4 STP
7.428 Switch1 PC3 STP
7.428 Switch1 PC5 STP
Reset Simulation
Constant Delay
Captured to: 7.428 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: 00:51:21.954
Power Cycle Devices PLAY CONTROLS: Back
Auto Capture / Play
Capture / Forward
Event List
Simulation
Connections
Automatically Choose Connection Type
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(sec) Periodic Num Edit Delete
Successful PC0 PC2 ICMP 0.000 N 0 (edit) (delete)
Successful PC3 PC5 ICMP 3.422 N 1 (edit) (delete)
Successful PC0 PC4 ICMP 5.423 N 2 (edit) (delete)
Search
ENG IN
10:07 AM 6/15/2023
10:15 AM 6/15/2023

LAB 2

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

OBSERVATION



Experiment 02: Single Router LAB - 02

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

2a): Aim: Configuring IP address to router and exploring Ping messages.

Procedure:-

Step 1:- Select Router-PT and Place it in WorkSpace

Step 2:- Take 2 end devices as PC-PT and drop them in WorkPlace.

Step 3: Connect FastEthernet 0/0 of Pc1 to FastEthernet 0/0 of Router and FastEthernet 0/0 Pc2 to FastEthernet 2/0 of Router using copper cross-over.

Step 4: Set IP address of Pc1 as 10.0.0.1 and Pc2 as 20.0.0.1

Step 5:- In Settings set Gateway of Pc1 as and Pc2 as 20.0.0.10

Step 6: Setup the interface of router using the following steps

To configure router command line interface (CLI) is used

Router 0 CLI

(Press N)

Router > enable

Router #config t

Router(config) # interface fastEthernet 0/0

Router(config-if) # ip address 10.0.0.10 255.0.0.0

Router(config-if) # no shut

exit

Router(config) # exit

Router #

Show ip route

C 10.0.0.0/8 is directly connected FastEthernet 0/0

C 20.0.0.0/8 is directly connected FastEthernet 1/0

Step 7: Observation

Green lights appear on wires when the no shut commands are written which indicate that they are ready for data transmission.

Ping output in PC0:

PC> Ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data

Request timedout.

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 = 0ms TTL = 127

Ping statistics for 20.0.0.1

Packets: Sent = 4, Received = 3, Lost = 1 (

Approximate round trip times in milli-seconds:

minimum - 0ms, maximum = 1ms, average.

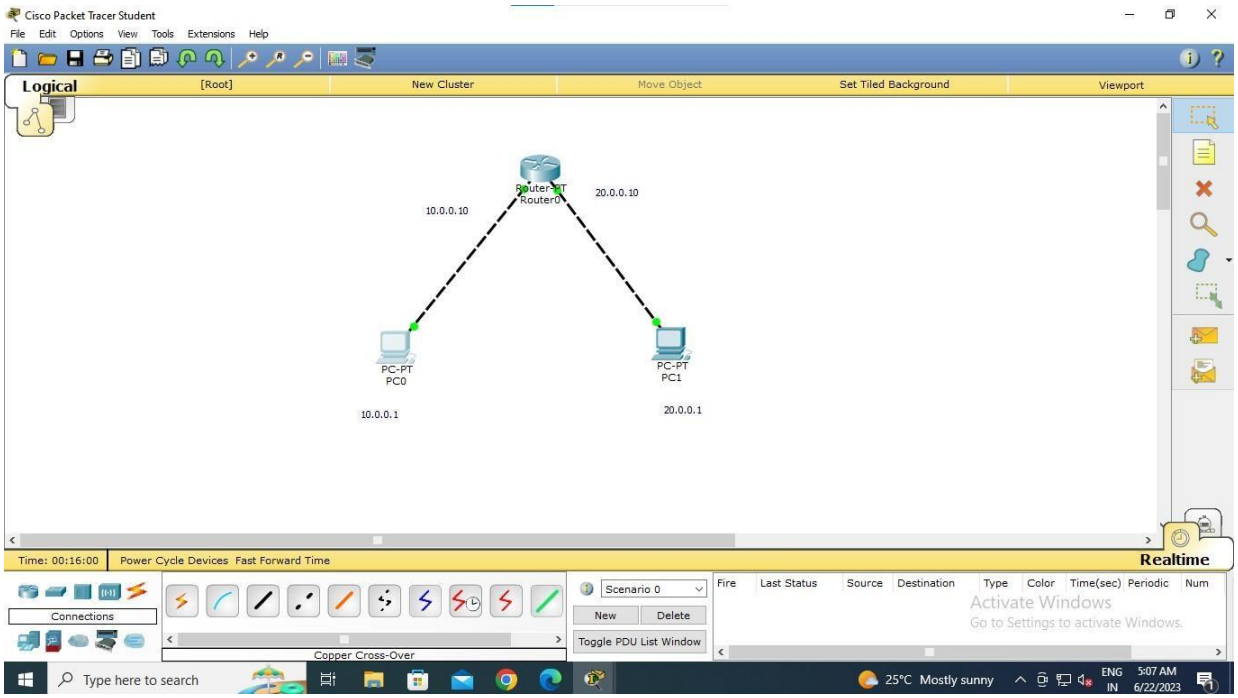
Observation:-

On Pinging in PC0 for the first time there is a 25% loss

From next Ping, there are no losses.

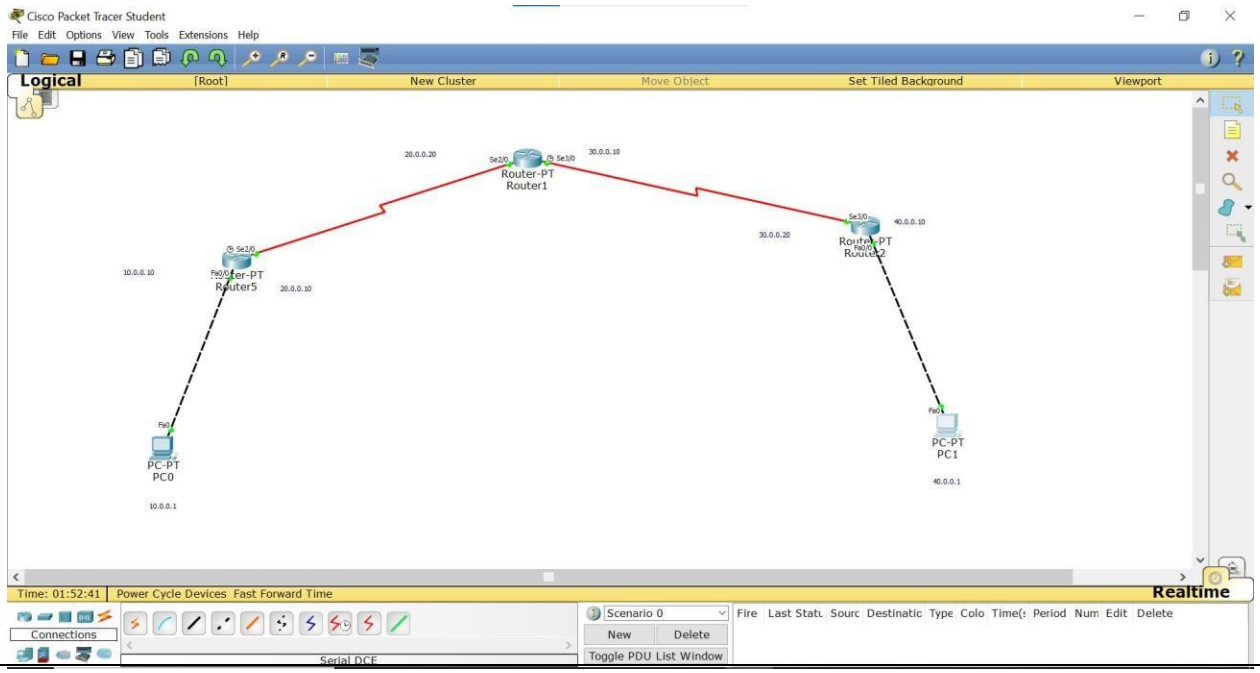
Output :

TOPOLOGY:

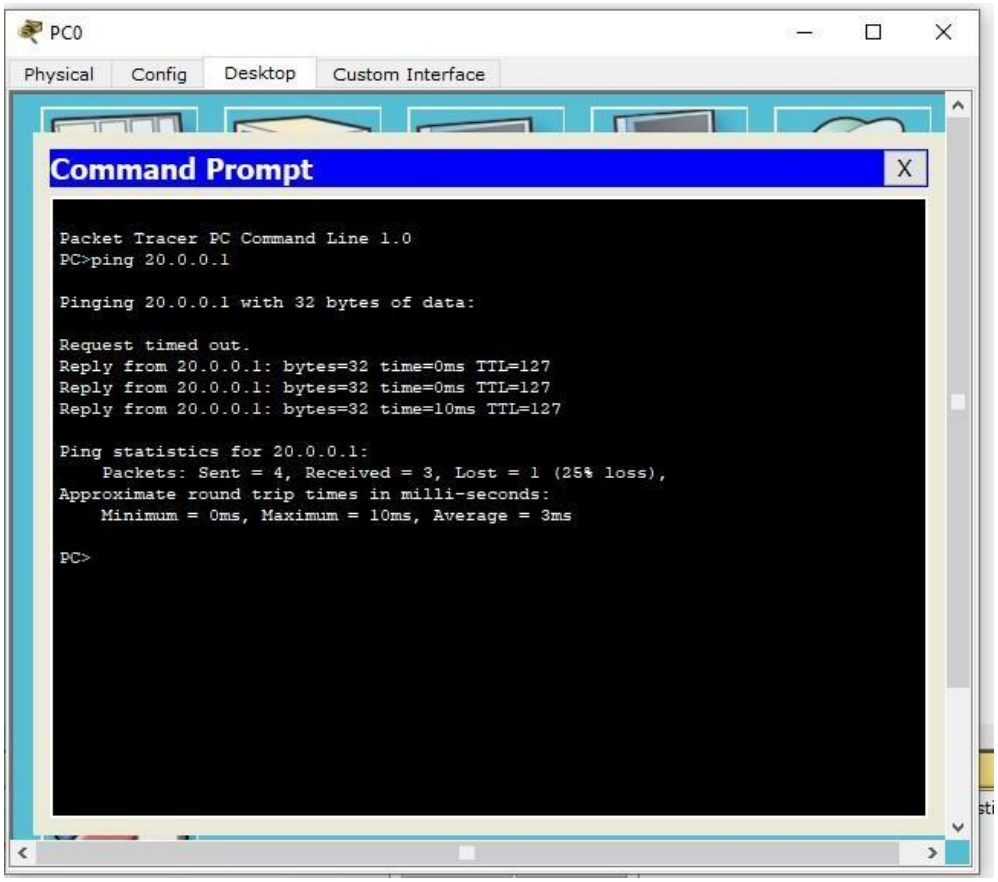


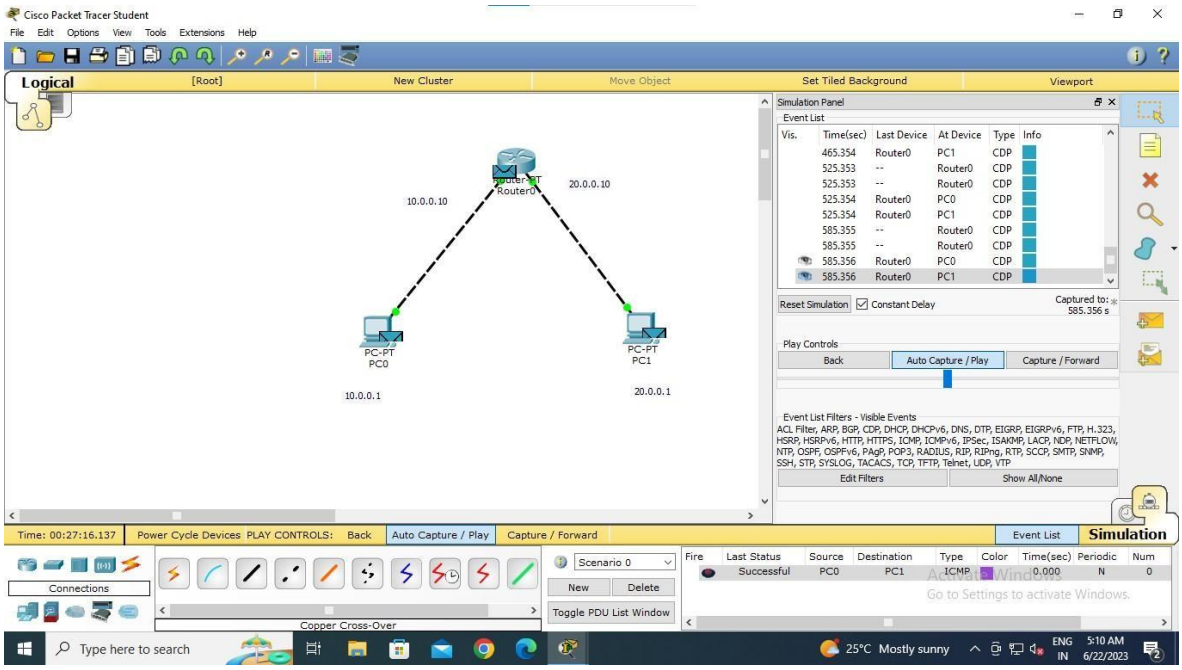
PROGRAM 2.1

PROGRAM 2.2

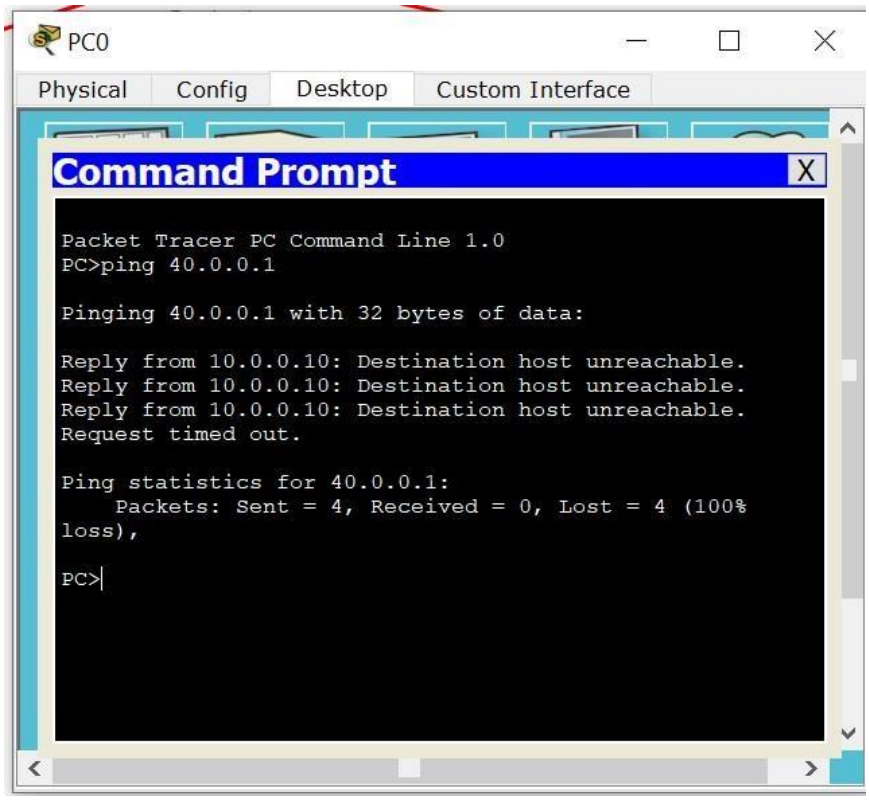


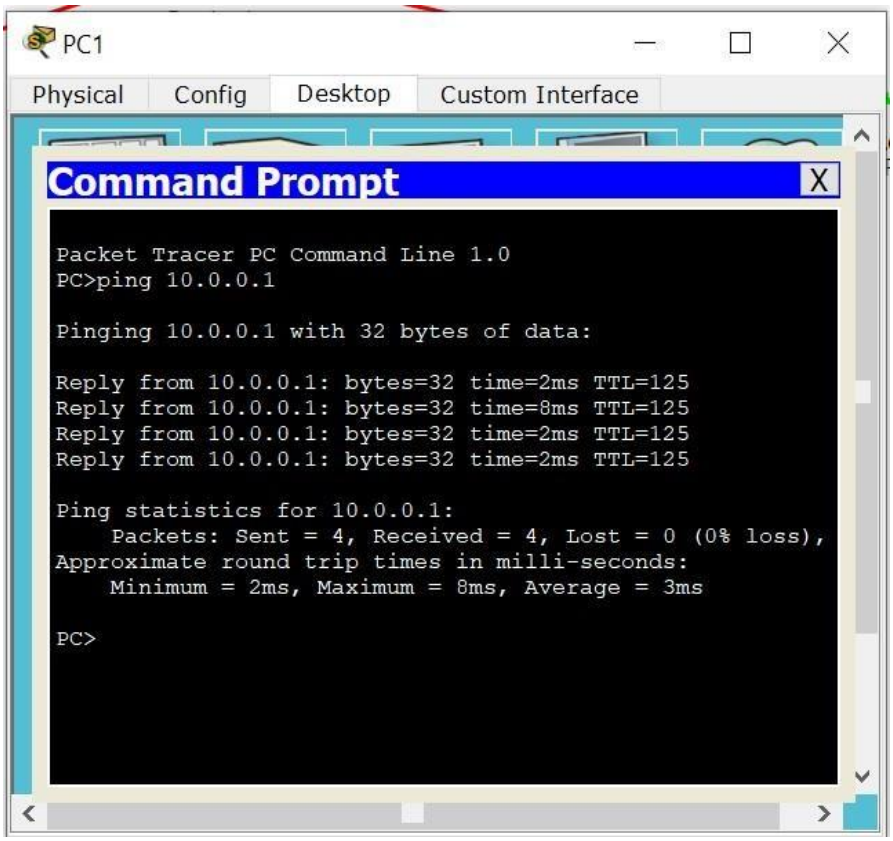
PROGRAM 2.1

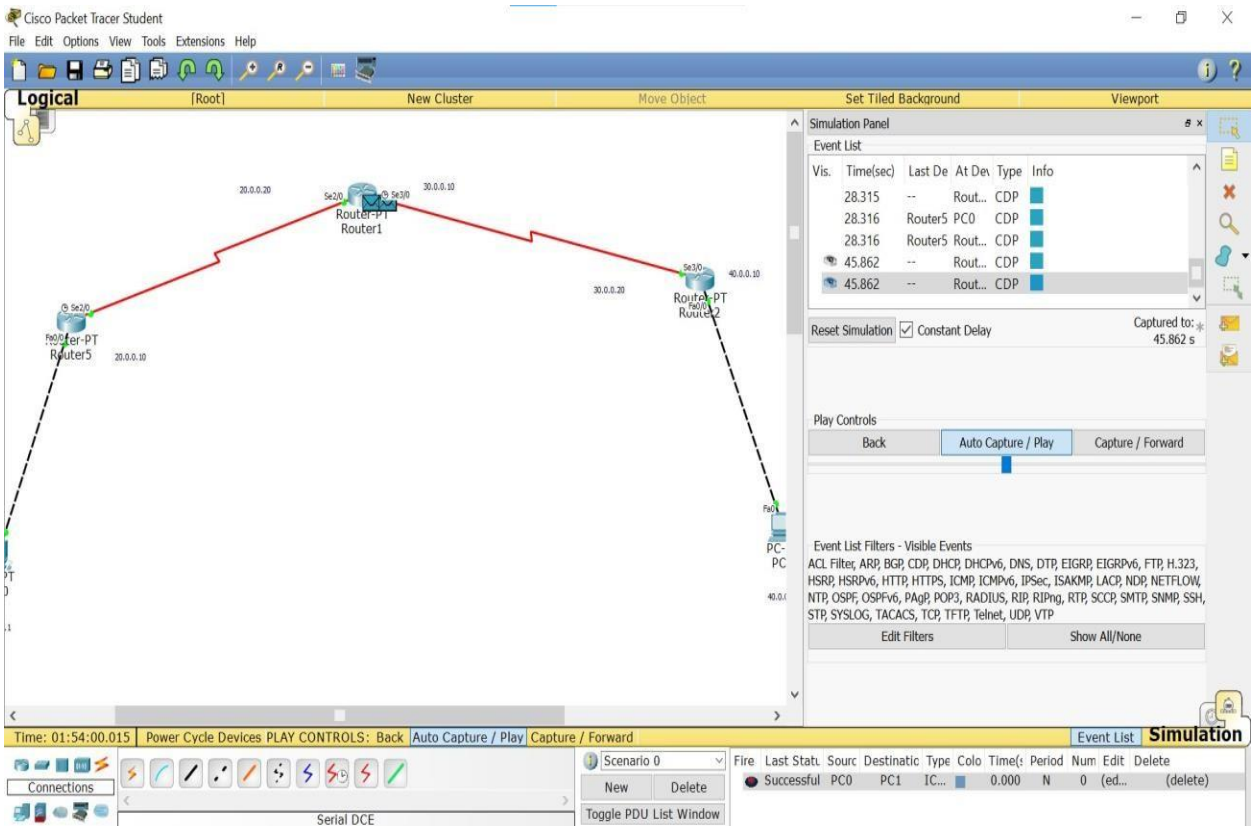

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Request timed out.
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=10ms 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 = 10ms, Average = 3ms
PC>


Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
10.0.0.10
20.0.0.10
Router-PT
Router0
PC-PT
PC0
PC-PT
PC1
10.0.0.1
20.0.0.1
Simulation Panel
Event List
Reset Simulation
Constant Delay
Play Controls
Back
Auto Capture / Play
Capture / Forward
Edit Filters
Show All/None
Time: 00:27:16.137
Power Cycle Devices PLAY CONTROLS: Back
Auto Capture / Play
Capture / Forward
Event List
Simulation
Connections
Copper Cross-Over
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Periodic
Num
Successful
PC0
PC1
ICMP
0.000
N
0
Type here to search
25°C Mostly sunny
ENG
IN
5:10 AM
6/22/2023

PROGRAM 2.2


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>

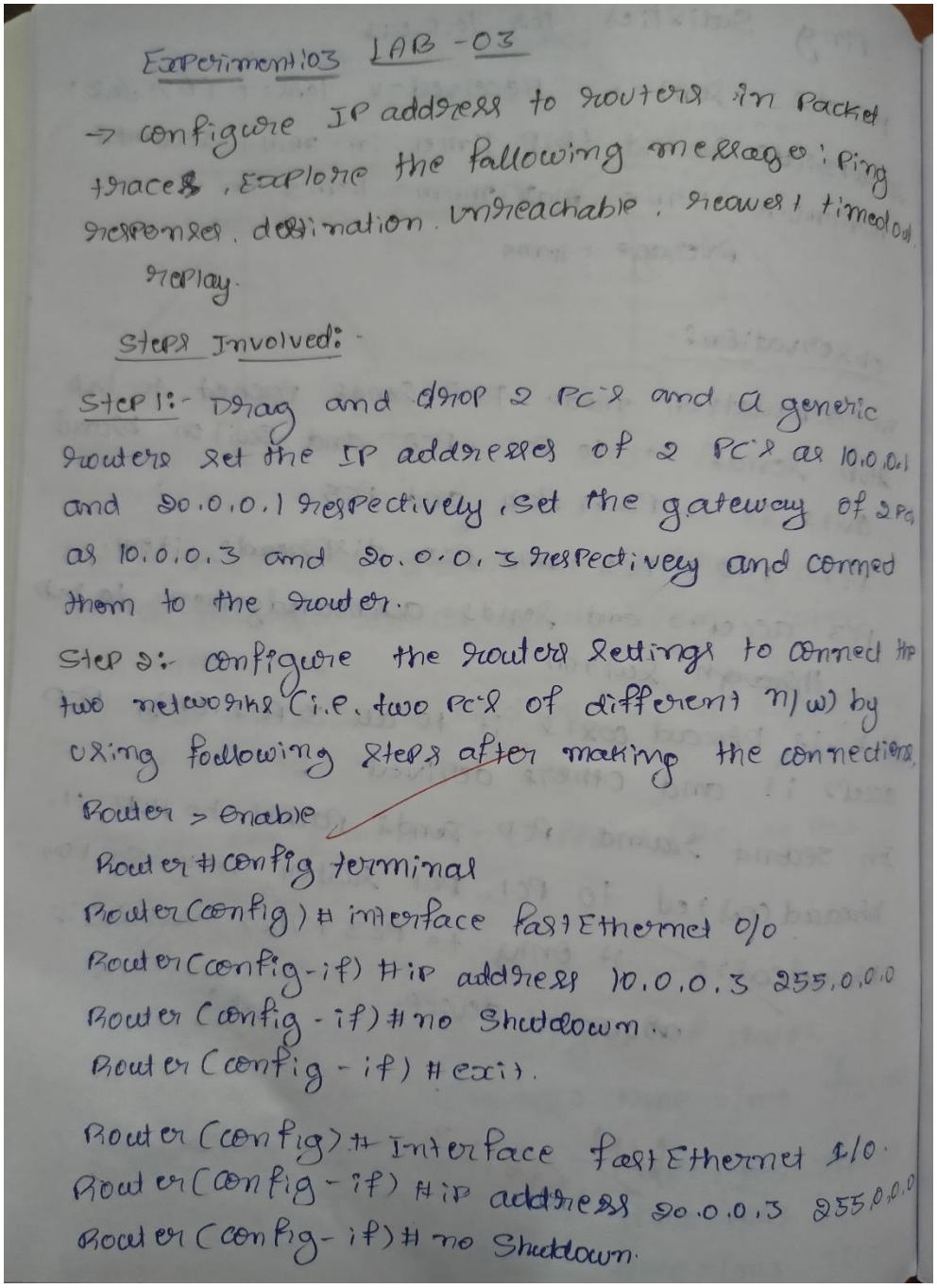

PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=2ms TTL=125
Reply from 10.0.0.1: bytes=32 time=8ms TTL=125
Reply from 10.0.0.1: bytes=32 time=2ms TTL=125
Reply from 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
Router-PT
Router1
Router-PT
Router2
Router-PT
Router5
PC-PT
PC
20.0.0.20
30.0.0.10
30.0.0.20
40.0.0.10
20.0.0.10
Simulation Panel
Event List
Vis. Time(sec) Last De At De 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 Stat. Sourc Destinatic Type Colo Time(s Period Num Edit Delete
Successful PC0 PC1 IC... 0.000 N 0 (ed... (delete)

LAB 3

Configure default route to the Router.

OBSERVATION:



Experiment:03 LAB - 03

→ configure IP address to routers in Packet tracer. Explore the following message: Ping responses, destination unreachable, request timed out, reply.

Steps Involved:

Step 1:- Drag and drop 2 PC's and a generic routers set the IP addresses of 2 PC's as 10.0.0.1 and 20.0.0.1 respectively, set the gateway of 2 PCs as 10.0.0.3 and 20.0.0.3 respectively and connect them to the router.

Step 2:- configure the routers settings to connect the two networks (i.e. two PC's of different n/w) by using following steps after making the connections.

Router > Enable
Router # config terminal
Router(config) # interface fast Ethernet 0/0
Router(config-if) # ip address 10.0.0.3 255.0.0.0
Router (config - if) # no Shutdown.
Router (config - if) # exit.

Router (config) # Interface fast Ethernet 1/0.
Router(config - if) # ip address 20.0.0.3 255.0.0.0
Router (config - if) # no Shutdown.

Router (config-if) # Exit
Router (config-if) # Exit
Router #

Step 3: Send a simple PDU from with IP address 10.0.0.1 to PC1 with IP address 30.0.0.1 and confirms how many Packets sent by using Ping command.

Step 4:
Similarly, connect two more Pc's with a router and configure by following above mentioned steps. Introduce one more router and connect it to the existing two routers of different network and configure it.

Step 5:
Now if you Ping from the Pc with IP address 10.0.0.1 as → Ping 40.0.0.1 the response will be destination unreachable, Although it seemed there's a connection between these two Pc's indirectly via routers. But every routers may not have information regarding every network present in the topology so these Pc's cannot communicate, To eliminate this, we should use static routing to teach every router manually.

Step 6:

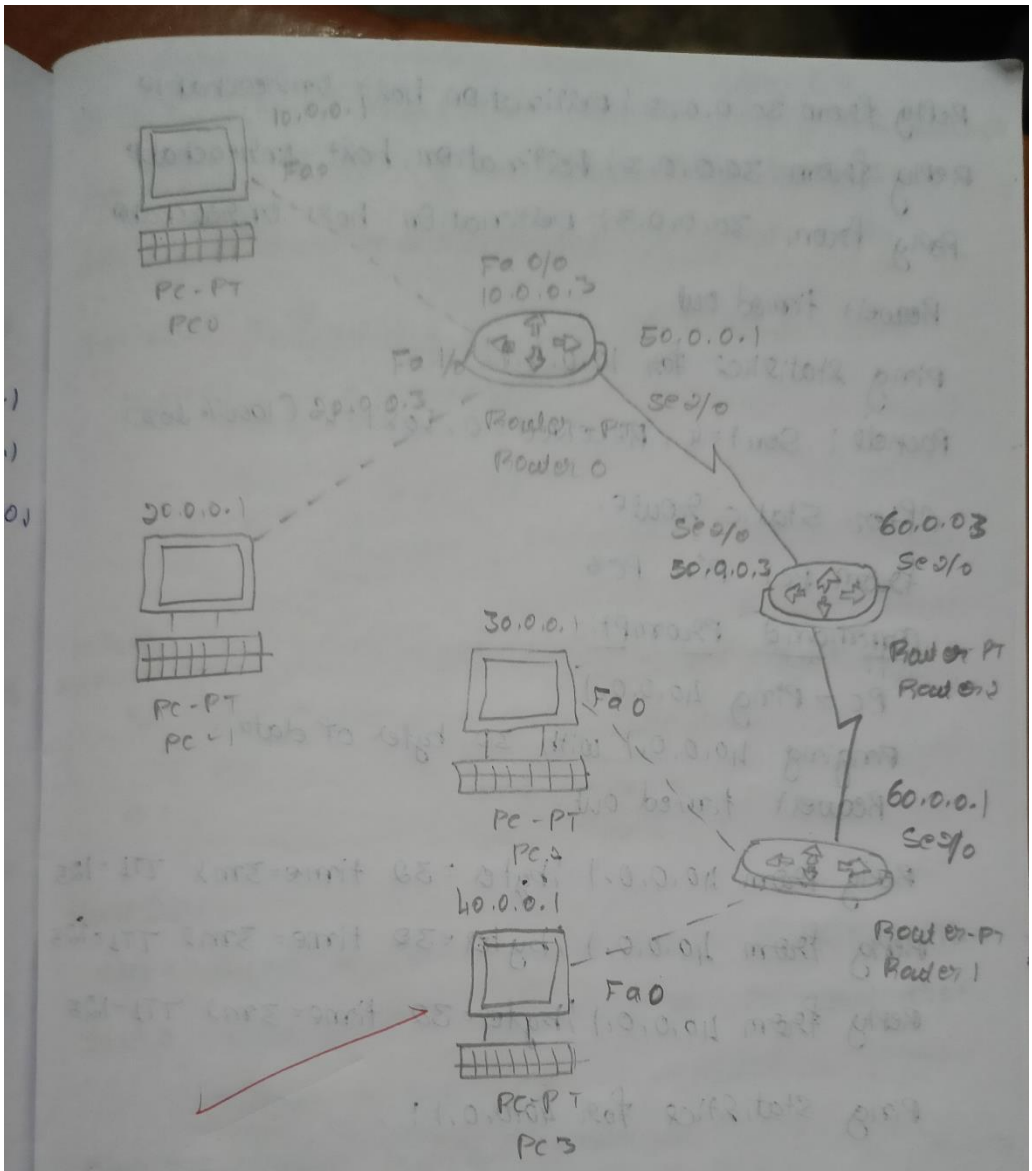
We can do static routing for router2 by the following steps.

```
Router # config t
Router (config)# ip route 10.0.0.0 255.0.0.0 50.0.0.1
Router (config) # ip route 20.0.0.0 255.0.0.0 50.0.0.1
Router (config) # ip route 30.0.0.0 255.0.0.0 60.0.0.1
Router (config) # ip route 40.0.0.0 255.0.0.0 60.0.0.1
Router (config) # exit
Router #
```

Step 7:

We can view all the networks connected to a router as follows:

```
Router # show IP route
codec  C - connected, S - Static
S  10.0.0.0/8 [2/0] via 50.0.0.1
S  20.0.0.0/8 [2/0]
S  30.0.0.0/8 [2/0]
S  40.0.0.0/8 [2/0]
C  50.0.0.0/8 is directly connected. Serial 2/0
C  60.0.0.0/8 is directly connected Serial 2/0
```



Before making Static route from PC2

Ping 10.0.0.1

Command Prompt

Pc -> Ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Reply from 30.0.0.3 : Destination host unreachable

Reply from 30.0.0.3 : Destination host unreachable

Reply from 30.0.0.3 : Destination host unreachable

Request timed out.

Ping statistics for 10.0.0.1 :

Packets : Sent = 4, Received = 0, Lost = 4 (100% loss)

After Static route.

from PC1 Pin PC3

Command Prompt :-

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 = 3ms TTL = 125

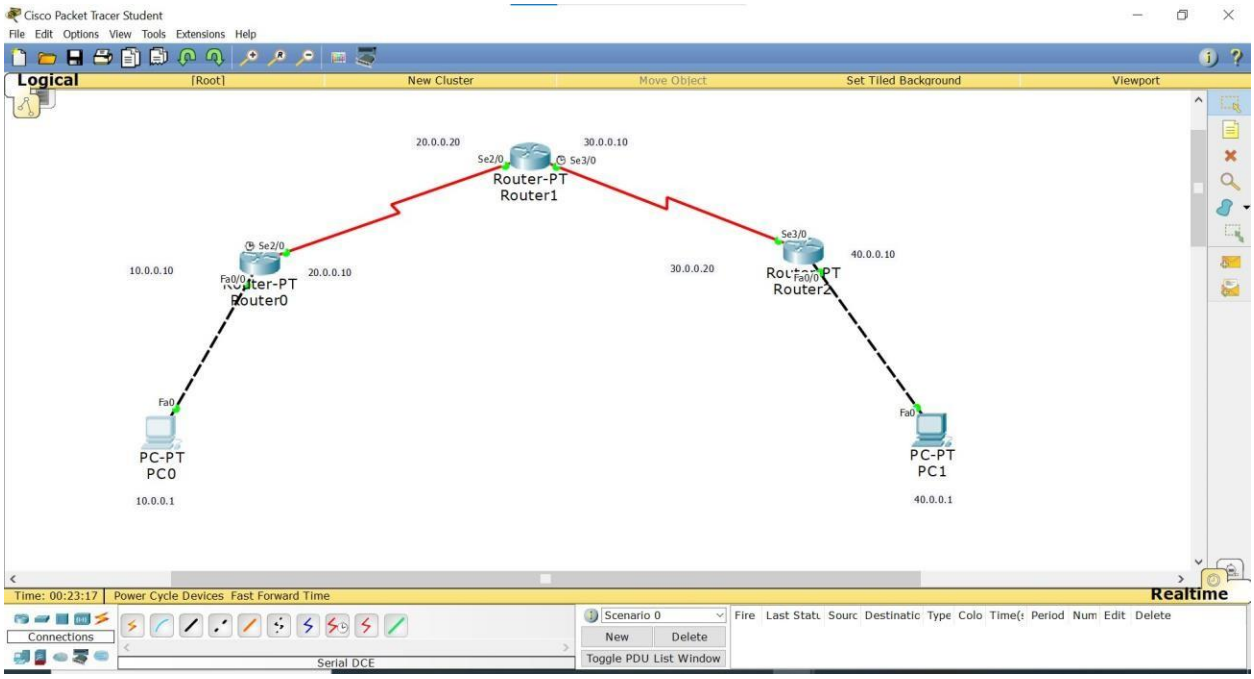
Reply from 40.0.0.1 : bytes = 32 time = 3ms TTL = 125

Reply from 40.0.0.1 : bytes = 32 time = 3ms TTL = 125

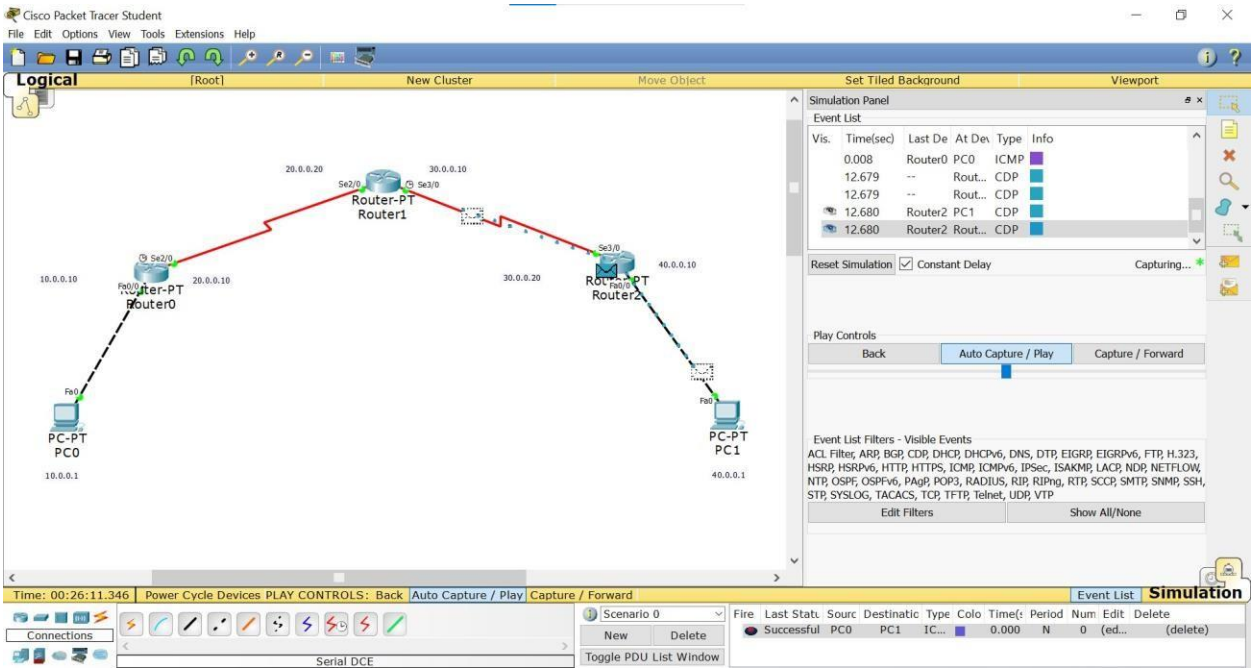
Ping statistics for 40.0.0.1 :

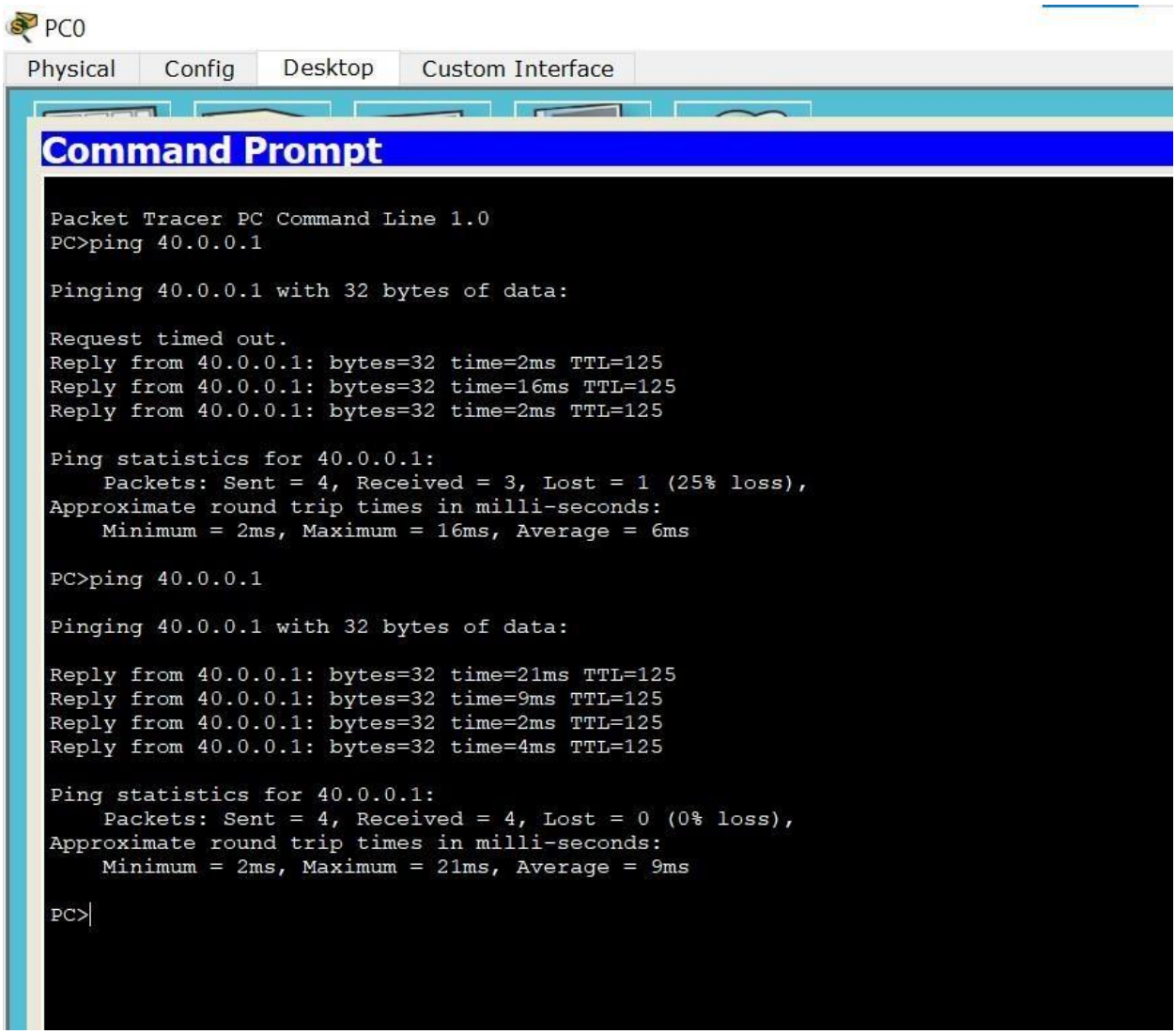
Packets : Sent = 4, Received = 3, Lost = 1 (25% loss)

TOPOLOGY:



OUTPUT:

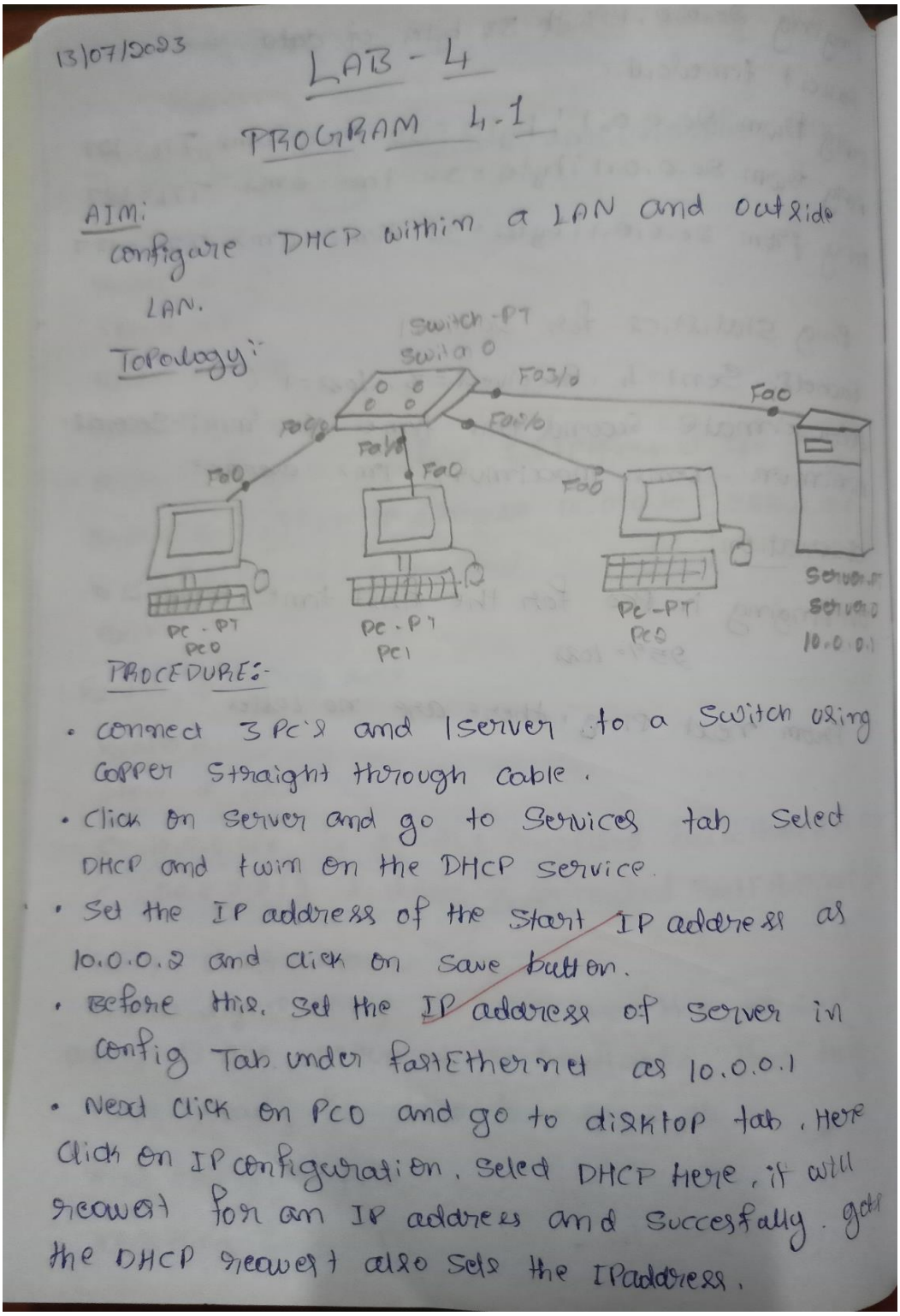



PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Request timed out.
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125
Reply from 40.0.0.1: bytes=32 time=16ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms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

LAB 4

Configure DHCP within a LAN and outside LAN.

OBSERVATION:



13/07/2023

LAB - 4

PROGRAM 4.1

AIM:

Configure DHCP within a LAN and outside LAN.

Topology:

PROCEDURE:-

- connect 3 PC's and 1 server to a switch using copper straight through cable.
- Click on server and go to Services tab select DHCP and turn on the DHCP service.
- Set the IP address of the start IP address as 10.0.0.2 and click on save button.
- Before this, set the IP address of server in config Tab under FastEthernet as 10.0.0.1
- Next click on PC0 and go to desktop tab. Here click on IP configuration, select DHCP here, it will request for an IP address and succesfully. gets the DHCP request also sets the IPaddress.

- Repeat this steps for other 2 PC's.
- To send a Packet across PC's go to PC's Command Prompt and type Ping destination IP address.

PING OUTPUT:

Packet Tracer PC Command Line 1.0:

PC>Ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data.

Reply from 10.0.0.3 bytes=32 time=0ms TTL=128
Reply from 10.0.0.3 bytes=32 time=0ms TTL=128
Reply from 10.0.0.3 bytes=32 time=0ms 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=0ms, Maximum=1ms, Average=0ms.

OBSERVATION:-

- DHCP is used to dynamically assign an IP address to any device or node.
- It is a client-server protocol in which servers manage a pool of unique IP address and also about client configuration parameters.
- DHCP enabled clients sends a request to DHCP server when they want to connect to a network.

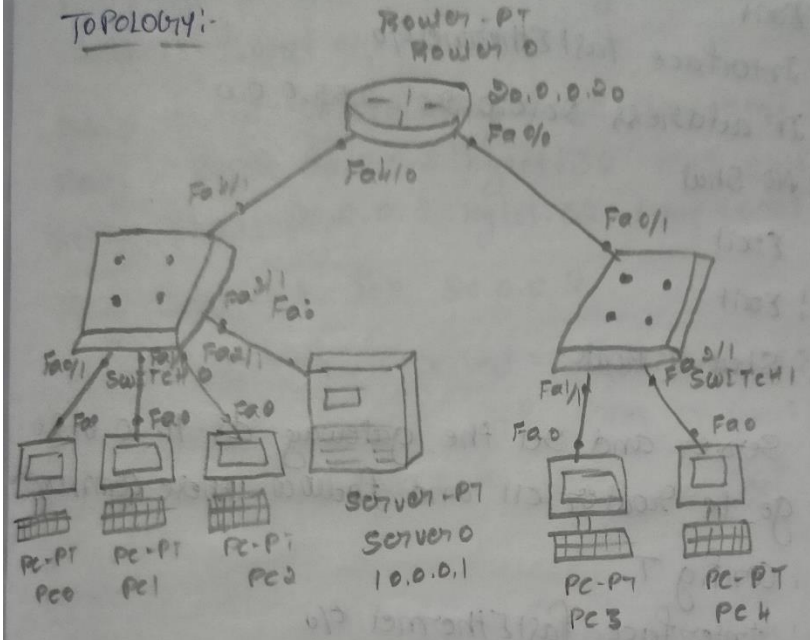
- The DHCP server responds to the client request by providing IP configuration information from address pools. Previously specified by network administrator.

PROGRAM 4.2

AIM:

Configure DHCP within a LAN and outside LAN.

TOPOLOGY:-



PROCEDURE:-

* Add a Router, a Switch and 2 PC's to 4.1 Program network and connect the Router to both Switches.
* Set the Server IP address of Server and with the help of Server Set the first 3 PC's IP address through DHCP.
* Now Set the Router IP address with the following commands Statically.

Step 1 : No

Step 2 : Enable

Step 3 : Config t

Step 4 : Interface fastEthernet 4/0
Step 5 : IP address 10.0.0.20 255.0.0.0
Step 6 : No Shut
Step 7 : Exit
Step 8 : Interface fastEthernet 0/0.
Step 9 : IP address 20.0.0.20 255.0.0.0
Step 10 : No Shut
Step 11 : Exit
Step 12 : Exit
Step 13 : Show IP Route.

- Go to Server and set the gateway as 10.0.0.20
- Again go to router cli and fodlow these commands
  Step 14 : Config T
  Step 15 : Interface fastEthernet 0/0
  Step 16 : IP helper address 10.0.0.1
  Step 17 : No Shut
  Step 18 : Exit
- Now, go to Server Services and add one more Pool as Server Pool, Start IP address as 20.0.0.2 and default gateway as 20.0.0.20, Then click add & Save.
- Now Set the other two PC's IP address by going to there Desktop → IP configuration and selecting DHCP which will automatically generate its IP address
- Now the network is complete and can send Packets from any PC to other by typing ping destination IP address in their respective command prompts.

PING OUTPUT:

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

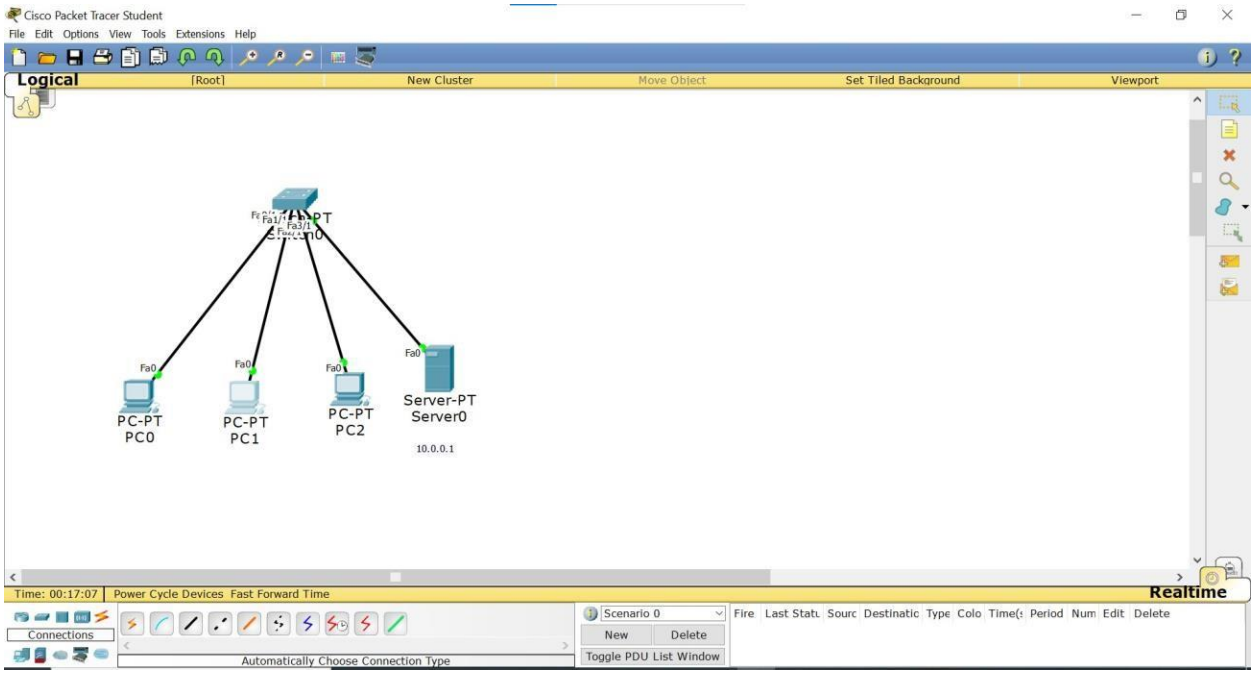
minimum = 0ms, Maximum = 0ms, Average = 0ms.

OBSERVATION:-

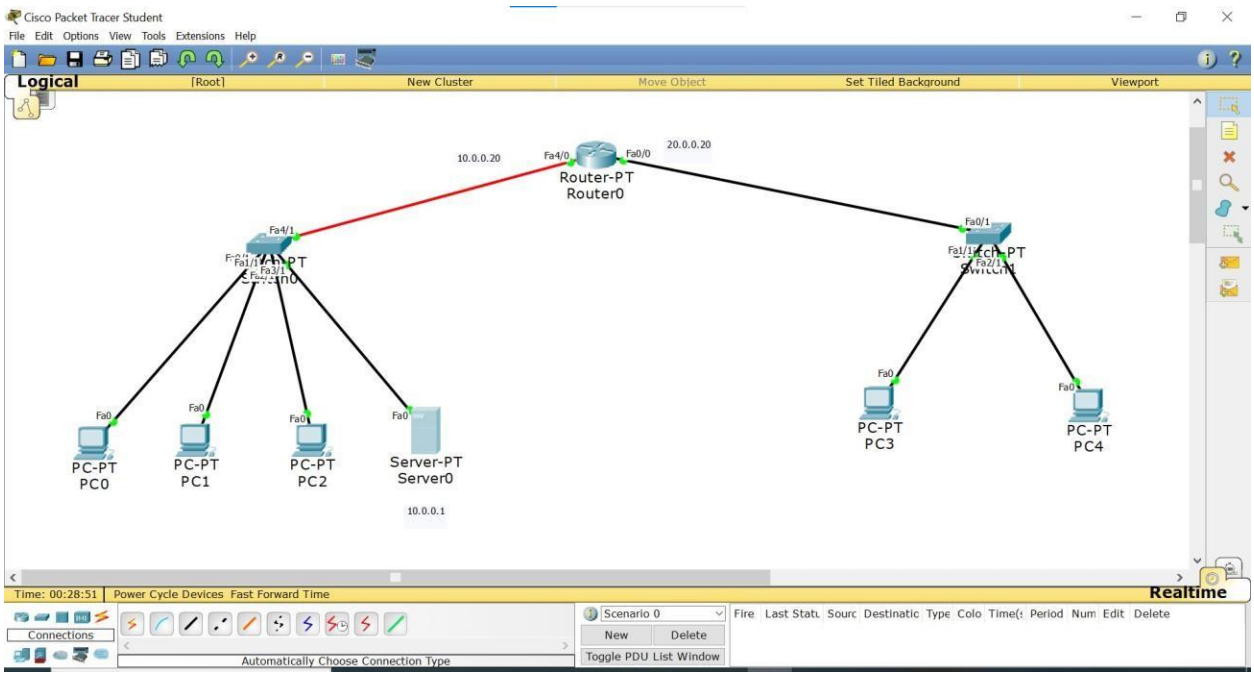
- DHCP is used to assign IP addresses dynamically to different devices.
- To assign continous IP address we create a Server Pool where we assign the starting IP address and a default gateway number. For PC's under different switches we create a different Server Pool again & Start This takes cares of delivering the Packets to correct destination IP address and also sends back the Ack to the initial device.

TOPOLOGY:

PROGRAM 4.1:

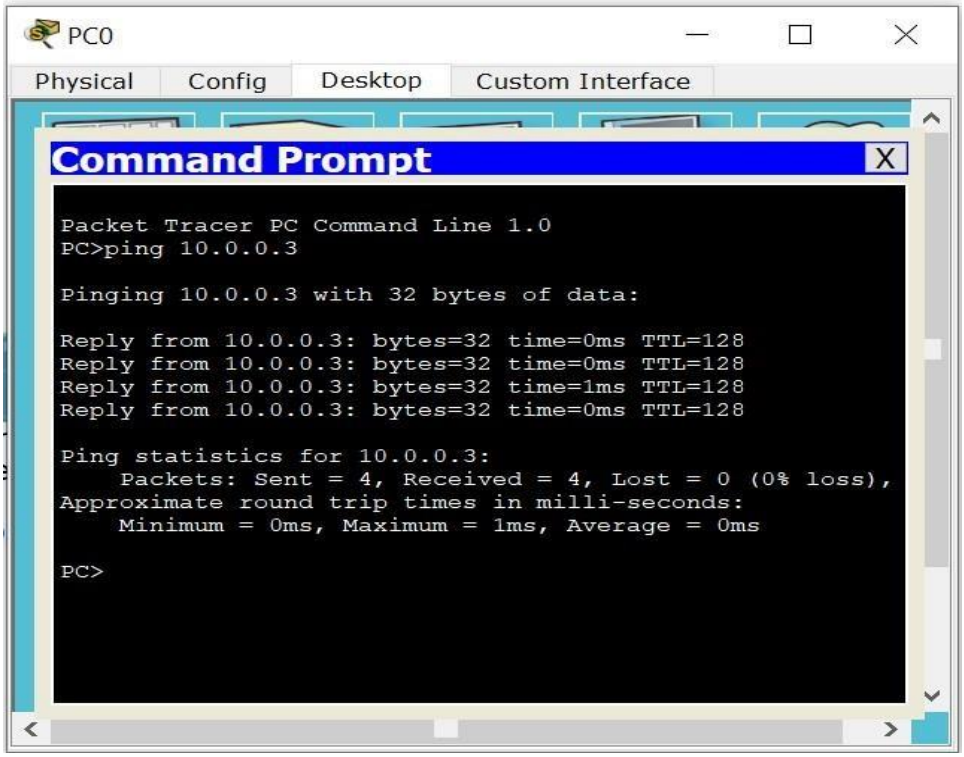


PROGRAM 4.2:

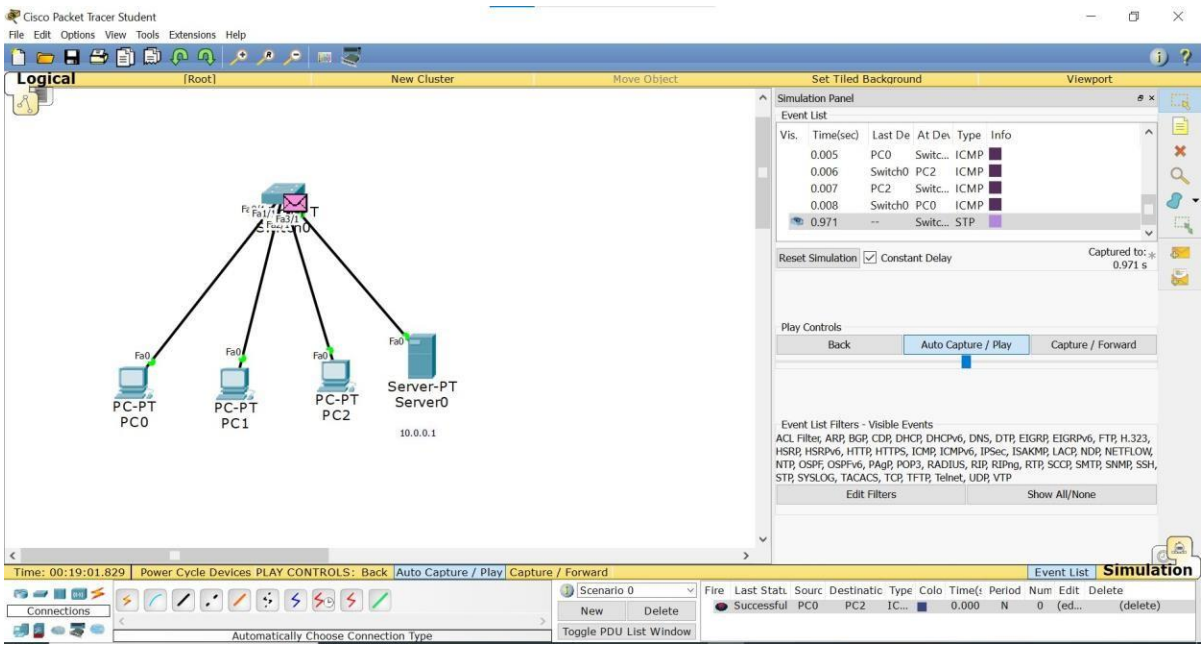


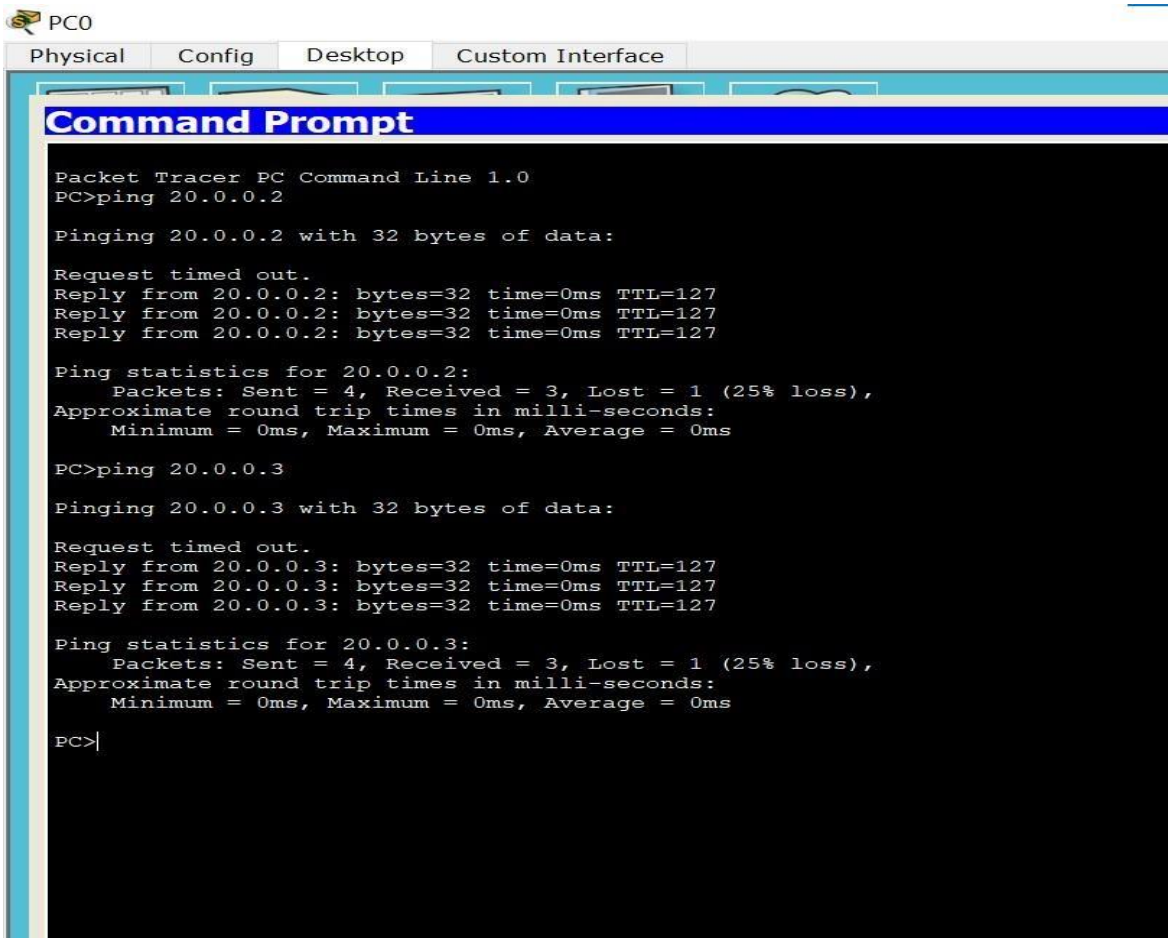
OUTPUT:

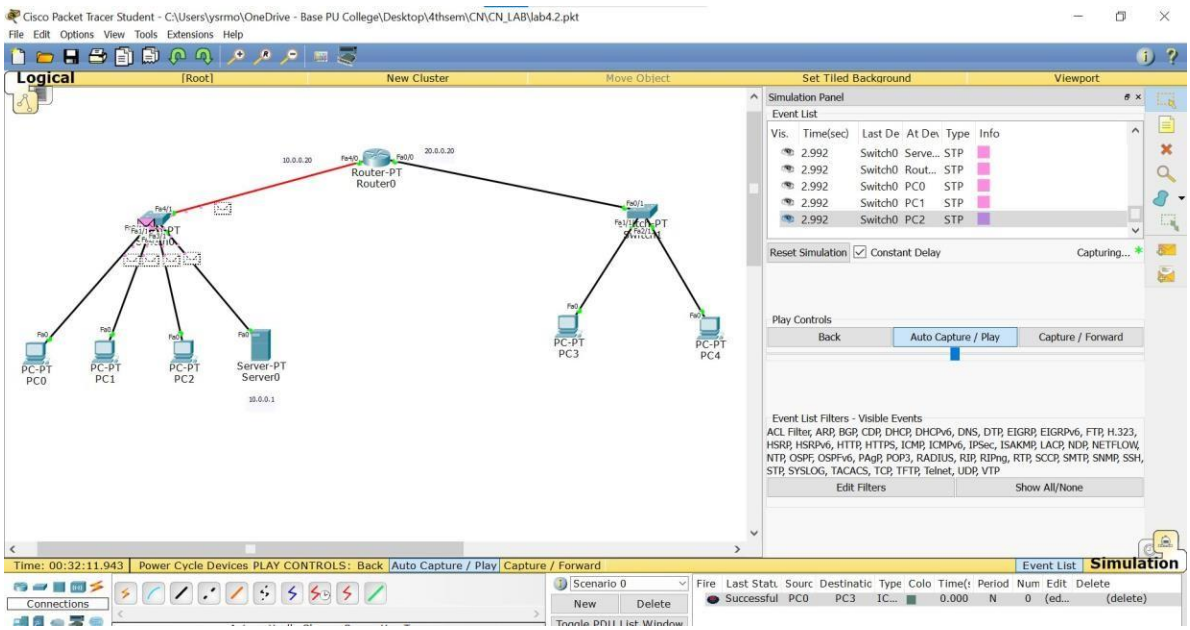
PROGRAM 4.1:



PROGRAM 4.2:



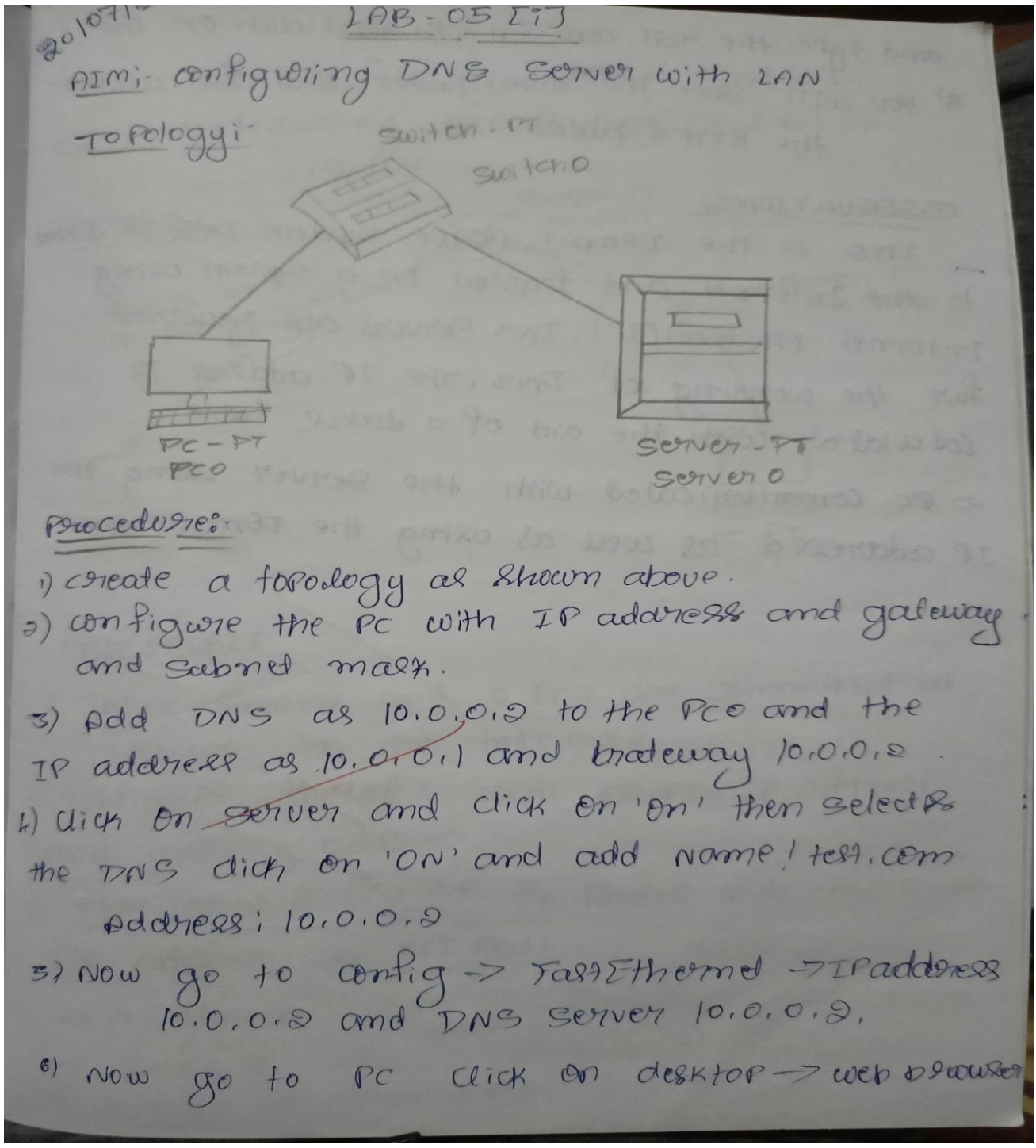

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data:
Request timed out.
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Reply from 20.0.0.2: bytes=32 time=0ms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


Logical
Router-PT
Router0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
Server-PT
Server0
PC-PT
PC3
PC-PT
PC4
Simulation Panel
Event List
Reset Simulation
Constant Delay
Play Controls
Back
Auto Capture / Play
Capture / Forward
Edit Filters
Show All/None
Simulation

LAB 5

Configure Web Server, DNS within a LAN.

OBSERVATION:



LAB - 05 [i]

Aim:- configuring DNS Server with LAN

Topology:-

Procedure:-

1) create a topology as shown above.
2) configure the PC with IP address and gateway and subnet mask.
3) Add DNS as 10.0.0.2 to the PC0 and the IP address as 10.0.0.1 and gateway 10.0.0.2.
4) Click on server and click on 'On' then selects the DNS click on 'ON' and add name: test.com address: 10.0.0.2
5) Now go to config -> FastEthernet -> IP address 10.0.0.2 and DNS server 10.0.0.2.
6) Now go to PC click on desktop -> web browser

and type the 'test.com' in URL and click on Go.

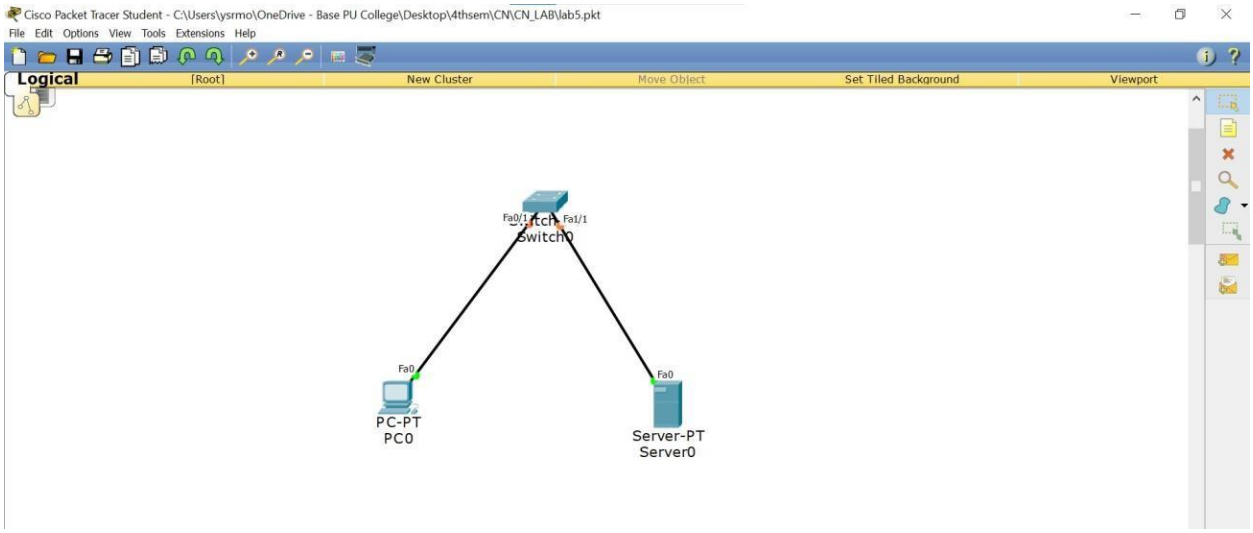
8) You will see the Html, Index that has written the HTTPS button.

OBSERVATIONS:

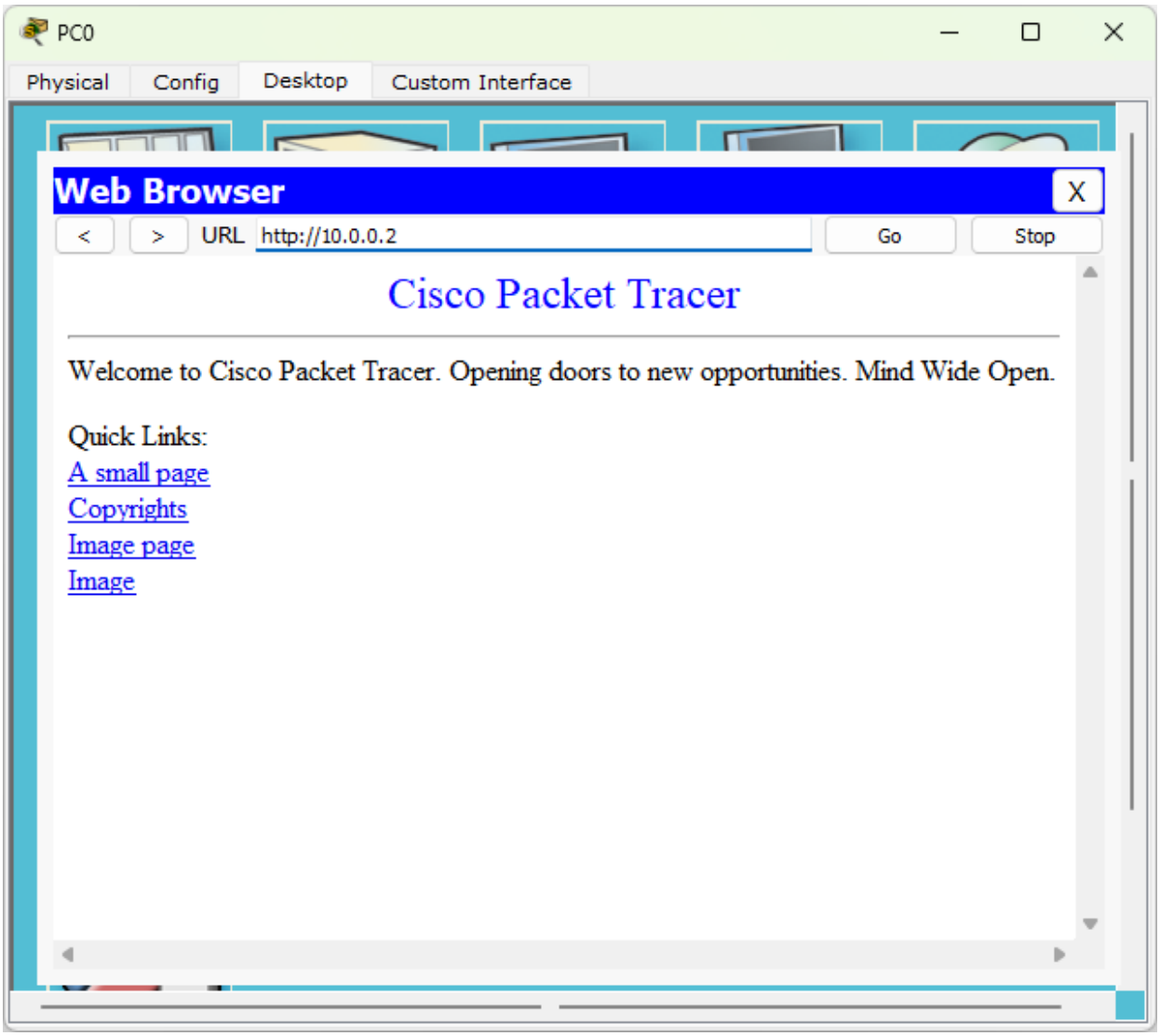
DNS is the Domain Name System. DNS is similar to the Internet and focused on a system using Internet Protocol (IP). DNS servers are required for the working of DNS, The IP address is calculated with the aid of a lookup table.

-> PC communicated with the server using the IP addresses as well as using the Domain Name.

TOPOLOGY:


Cisco Packet Tracer Student - C:\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\CN\CN_LAB\lab5.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Fa0/1
Fa1/1
Switch0
Fa0
PC-PT
PC0
Fa0
Server-PT
Server0

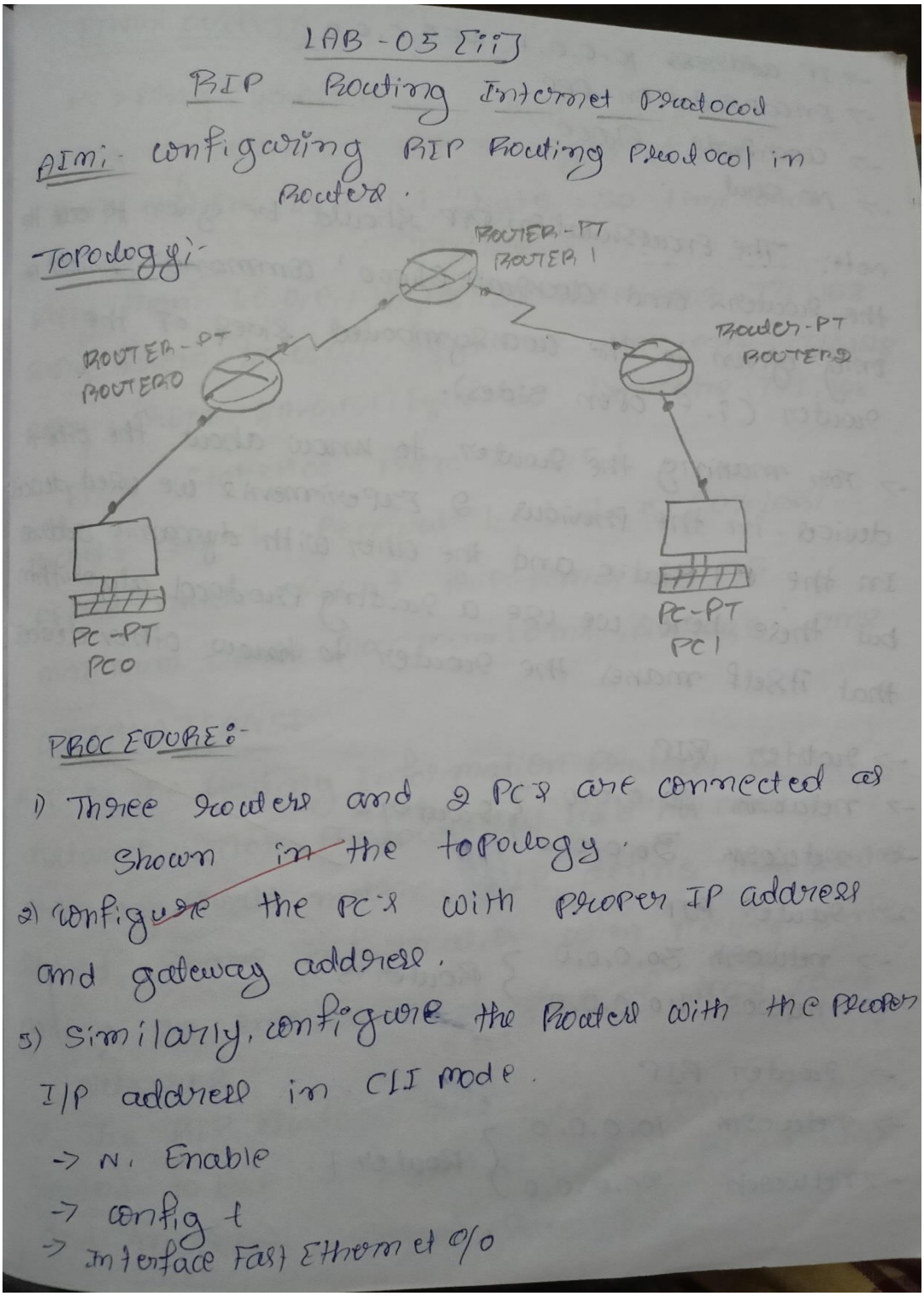
OUTPUT:


PC0
Physical
Config
Desktop
Custom Interface
Web Browser
X
<
>
URL
http://10.0.0.2
Go
Stop
Cisco Packet Tracer
Welcome to Cisco Packet Tracer. Opening doors to new opportunities. Mind Wide Open.
Quick Links:
A small page
Copyrights
Image page
Image

LAB 6

Configure RIP routing Protocol in Routers.

OBSERVATION:



-> IP address 10.0.0.1 255.0.0.0
-> Encapsulation PPP
-> clock rate 64000
-> No Shut

Note: The Encapsulation PPP should be given to all the routers and 'clockrate 64000' command should be only given to the clock symboled sides of the router (i.e. open sides).

-> For making the router to know about the other devices, in the previous 2 Experiments we used devices In the #1 Static and the other with dynamic address but these here we use a routing protocol algorithm that itself makes the router to know other devices

-> router RIP
-> network 20.0.0.0
-> network 30.0.0.0
} Router 2

-> router RIP
-> network 30.0.0.0
-> network 40.0.0.0
} Router 3

-> router RIP
-> network 10.0.0.0
-> network 20.0.0.0
} Router 1.

PING OUTPUT:

PC>Ping 40.0.0.0

Pinging 40.0.0.1 with 32 byte of data.

Reply from 40.0.0.1 ; byte = 32 time ; 0ms
TTL ; 128

Reply from 40.0.0.1 ; bytes = 32 time ; 0ms ; TTL ; 128

Reply from 40.0.0.1 ; bytes = 32 time ; 0ms ; TTL ; 128

Reply from 40.0.0.1 ; bytes = 32 time ; 0ms ; TTL ; 128

Ping Statistics from 40.0.0.1

Packets Sent = 4 Received = 4 Lost = 0 (0% Loss)

Approximate Round trip times in ms

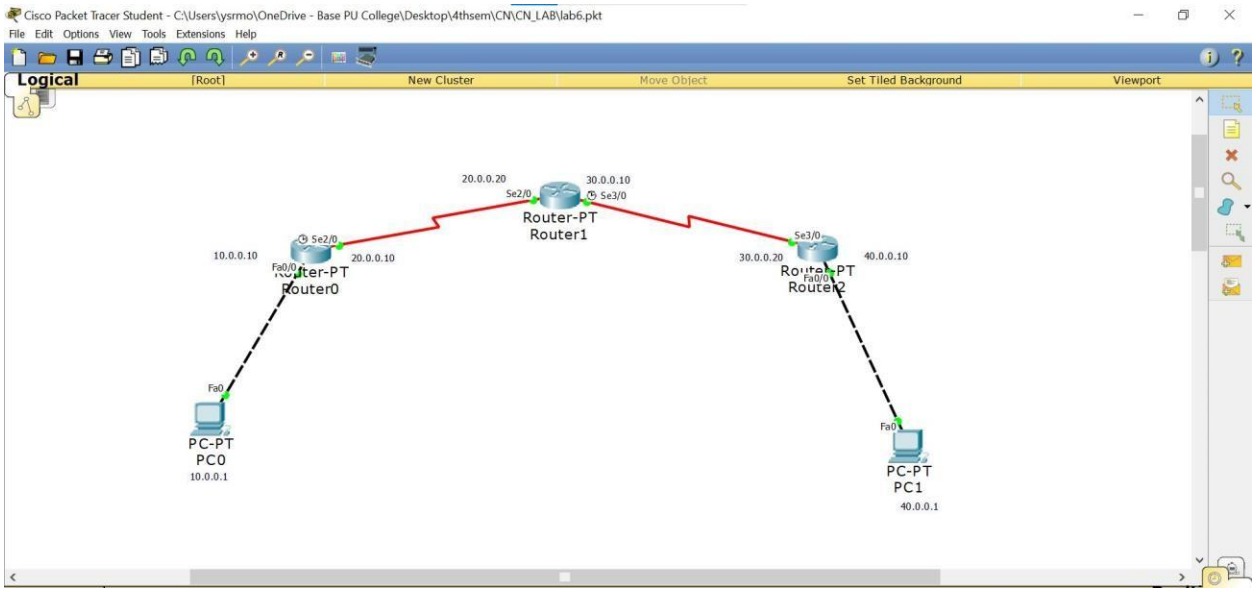
minimum = 0ms . Maximum ; 0ms , Average = 0ms.

OBSERVATIONS:-

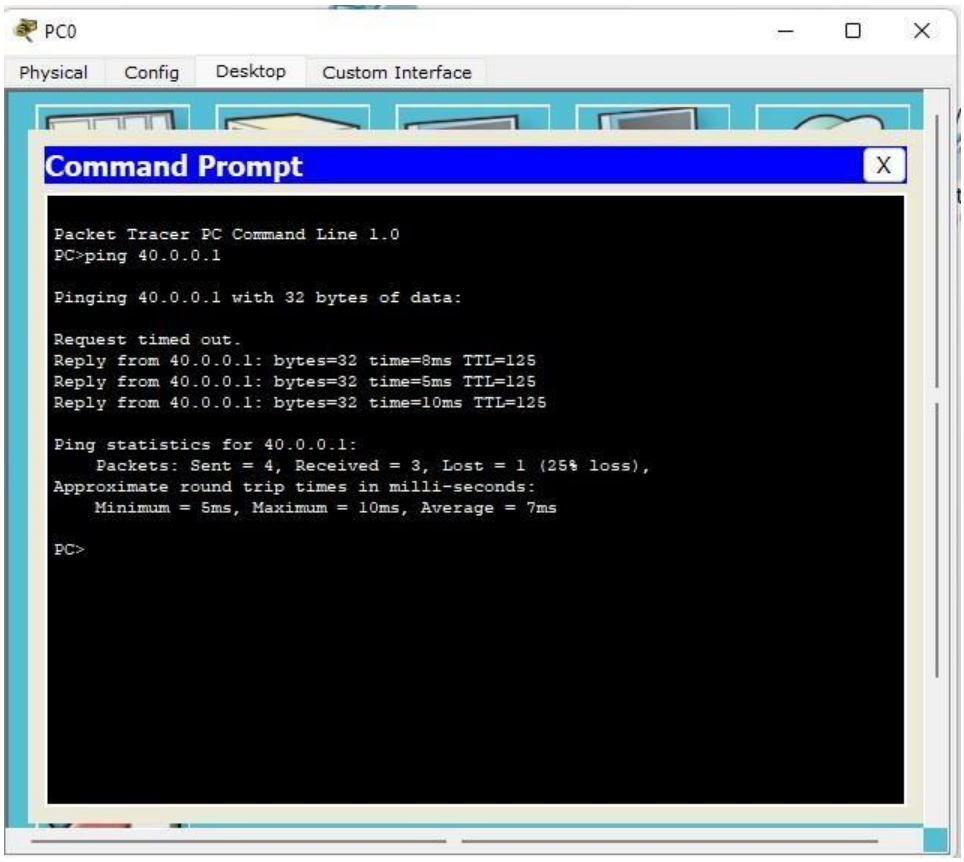
RIP is the Routing Information Protocol is a distance vector Protocol that bases hop count as its primary should metric, RIP defines how routers should share information when moving traffic among an interconnected group of local area Networks.

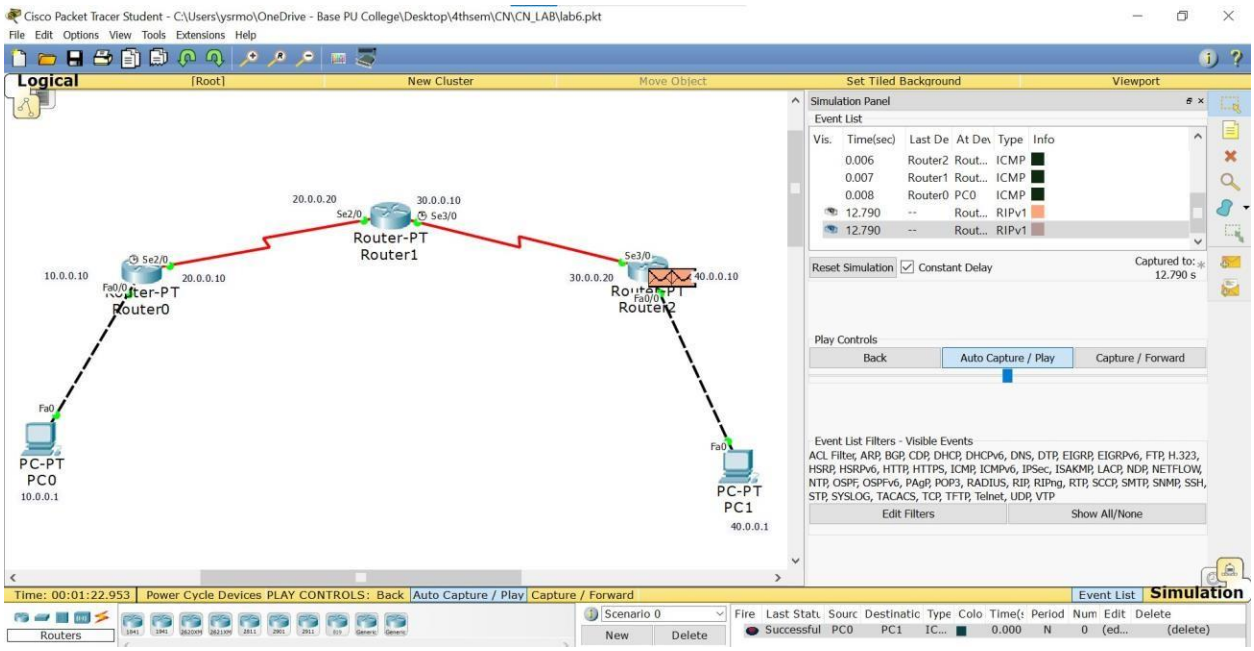
-> The RIP Protocol Here used to connect the routers to one other and PC's using RIP Protocol and message is Pinged Successfully.

TOPOLOGY:



OUTPUT:

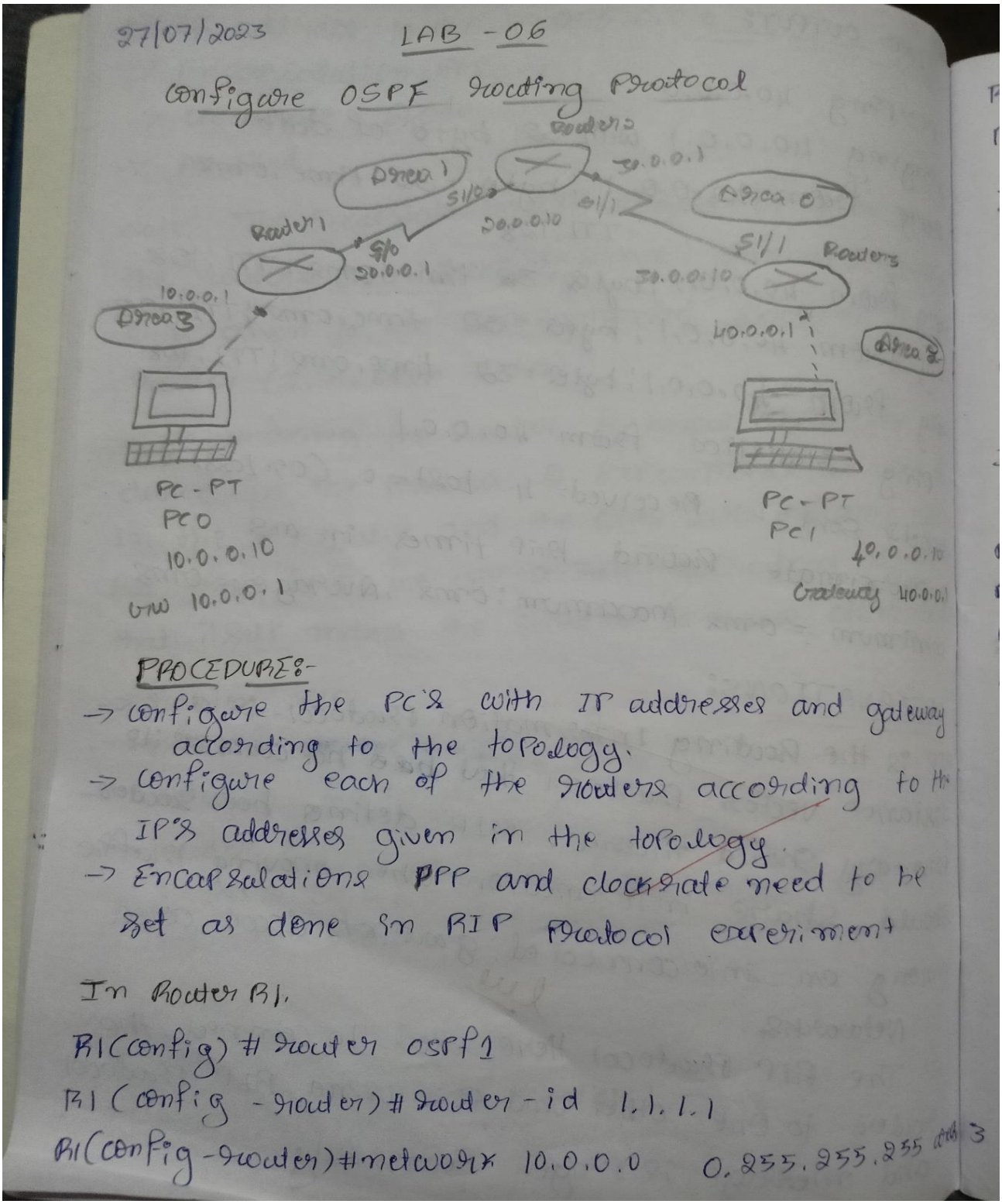



Cisco Packet Tracer Student - C:\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\CN\CN_LAB\lab6.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT
Router1
Router-PT
Router0
Router-PT
Router2
PC-PT
PC0
10.0.0.1
PC-PT
PC1
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)

LAB 7

Configure OSPF routing protocol.

OBSERVATION:



Virtual link:-

In Router R1.

R1(config)# router ospf 1

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

In Router R2.

R2(config)# router ospf 1

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

R2(config-router)# exit

→ Show IP route

OUTPUT:-

Reply from 40.0.0.10: bytes:32 time:0ms TTL:128
Reply from 40.0.0.10: bytes:32 time:0ms TTL:128
Reply from 40.0.0.10: bytes:32 time:0ms TTL:128
Reply from 40.0.0.10: bytes:32 time:0ms TTL:128

Pinging statistics from 40.0.0.10

Packet sent = 4 Received = 4 lost = 0 (0% loss)

Approximate round trip time in ms

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

31/7/23

R1(config-router)# network 20.0.0.0 0.255.255.255 area1
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 area1
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

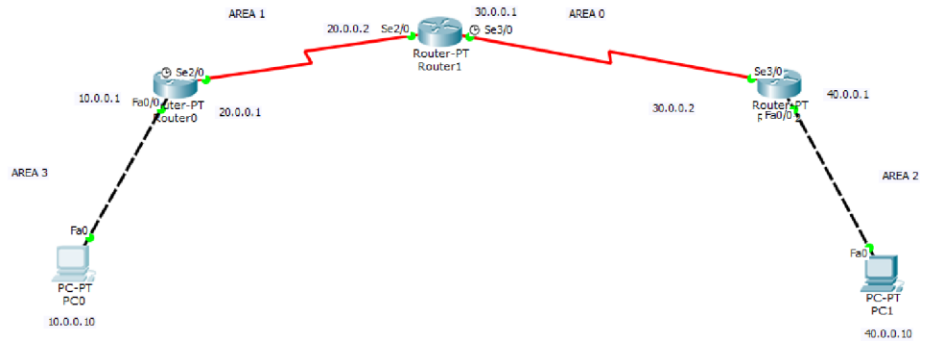
Interfaces Loopback:-

R1(config-if)# interface loopback 0
R1(config-if)# ip address 172.16.1.252 255.255.0.0
R1(config-if)# no shutdown

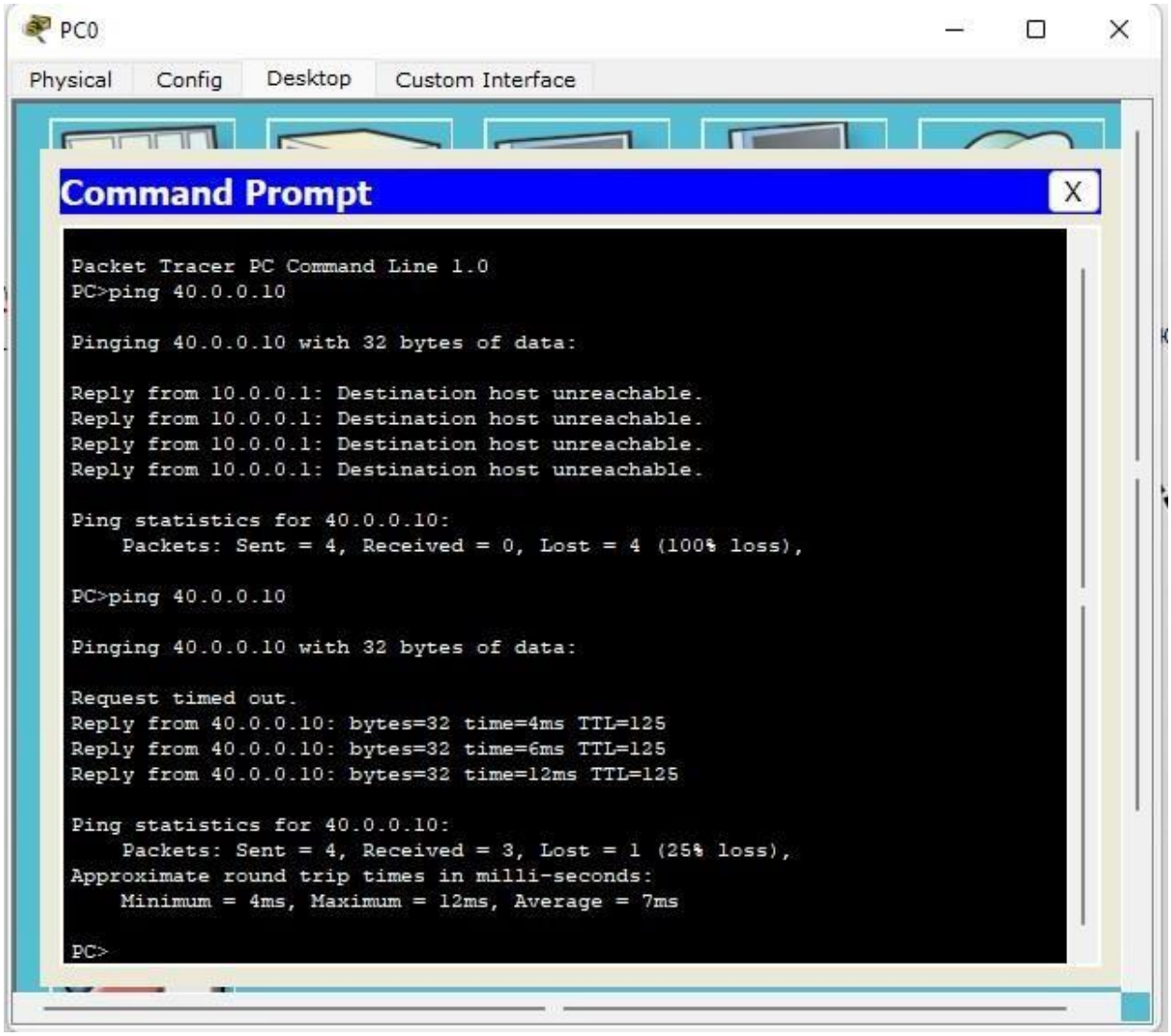
R2(config-if)# interface loopback 0
R2(config-if)# ip address 172.16.1.253 255.255.0.0
R2(config-if)# no shutdown

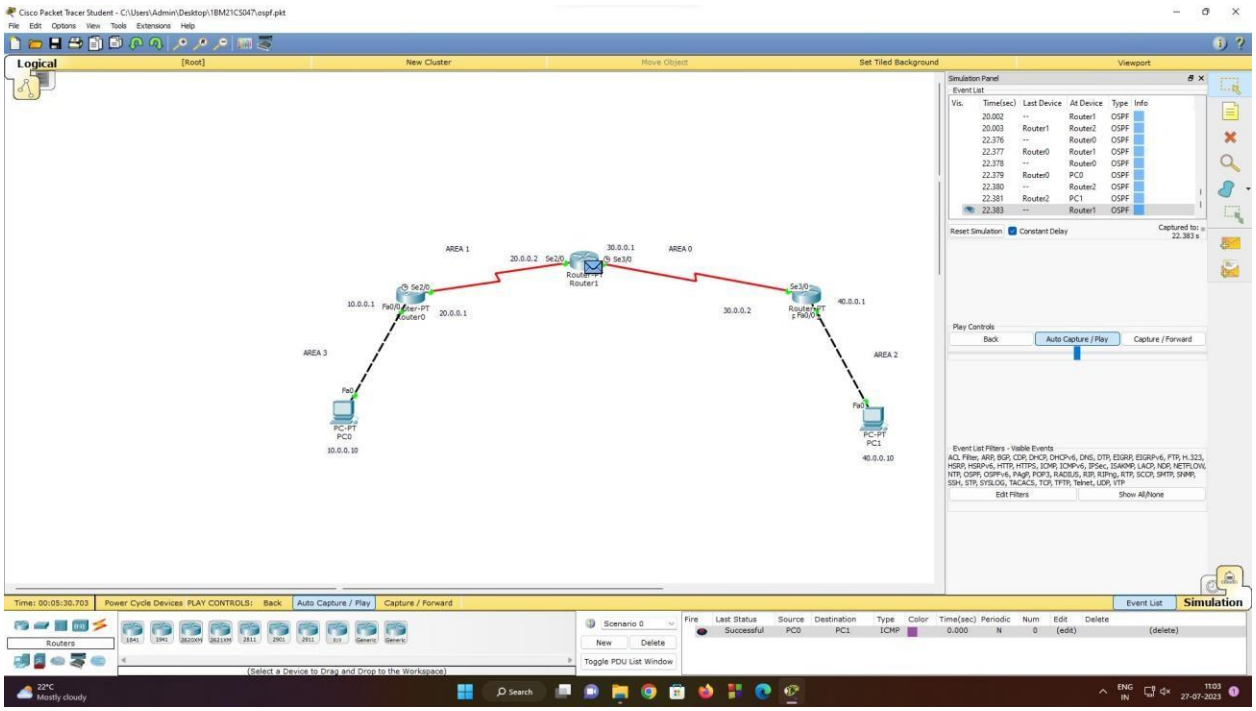
R3(config-if)# interface loopback 0
R3(config-if)# ip address 172.16.1.254 255.255.0.0
R3(config-if)# no shutdown.

TOPOLOGY:



OUTPUT:

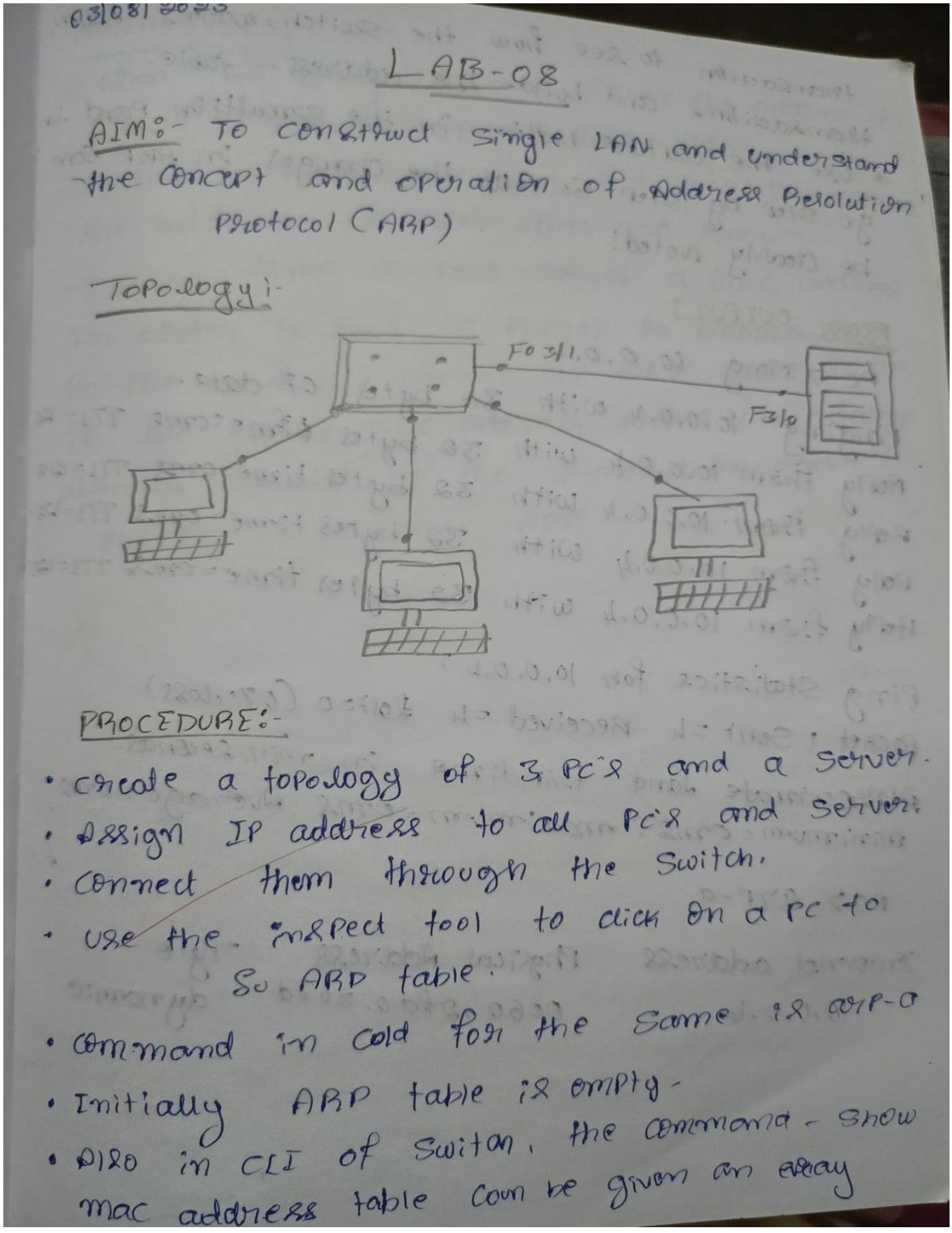



Cisco Packet Tracer Student - C:\Users\Admin\Desktop\1BM21CS047\ospf.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
AREA 1
AREA 0
AREA 3
AREA 2
20.0.0.2
Se2/0
30.0.0.1
Se3/0
Router1
10.0.0.1
Fa0/0
Router0
20.0.0.1
30.0.0.2
40.0.0.1
PC-PT
PC0
10.0.0.10
PC1
40.0.0.10
Simulation Panel
Event List
Vis. Time(sec) Last Device At Device Type Info
20.002 -- Router1 OSPF
20.003 Router1 Router2 OSPF
22.376 -- Router0 OSPF
22.377 Router0 Router1 OSPF
22.378 -- Router0 OSPF
22.379 Router0 PC0 OSPF
22.380 -- Router2 OSPF
22.381 Router2 PC1 OSPF
22.383 -- Router1 OSPF
Reset Simulation
Constant Delay
Captured to: 22.383 s
Play Controls
Back
Auto Capture / Play
Capture / Forward
Event List Filters - Visible Events
Edit Filters
Show All/None
Time: 00:05:30.703
Power Cycle Devices PLAY CONTROLS: Back Auto Capture / Play Capture / Forward
Event List
Simulation
Routers
(Select a Device to Drag and Drop to the Workspace)
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(sec) Periodic Num Edit Delete
Successful PC0 PC1 ICMP 0.000 N 0 (edit) (delete)
22°C
Mostly cloudy
Search
ENG
IN
11:03
27-07-2023

LAB 8

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

OBSERVATION:



03/08/2023

LAB-08

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

Topology:-

PROCEDURE:-

- Create a topology of 3 PC's and a Server.
- Assign IP address to all PC's and Server.
- Connect them through the Switch.
- Use the inspect tool to click on a PC to see ARP table.
- Command in cold for the same is arp-a
- Initially ARP table is empty.
- Also in CLI of Switch, the command - show mac address table can be given an easy

transaction to see how the switch learns from transactions and build the address-table.

- Use the capture button in the simulation panel and go step by step so that the changes in ARP can be clearly noted!

PING OUTPUT:-

PC -> Ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data.
Reply from 10.0.0.4 with 32 bytes time=0ms TTL=128
Reply from 10.0.0.4 with 32 bytes time=0ms TTL=128
Reply from 10.0.0.4 with 32 bytes time=0ms TTL=128
Reply from 10.0.0.4 with 32 bytes time=0ms TTL=128

Ping Statistics for 10.0.0.4:
Packet: Sent=4 Received=4 Lost=0 (0% loss)
Approximate round trip time in milliseconds.
Minimum=0ms, Maximum=0ms Average=0ms

PC> arp-a

Internet address	Physical Address	Type
10.0.0.4	0060.2fa0.304d	dynamic

OBSERVATION:-

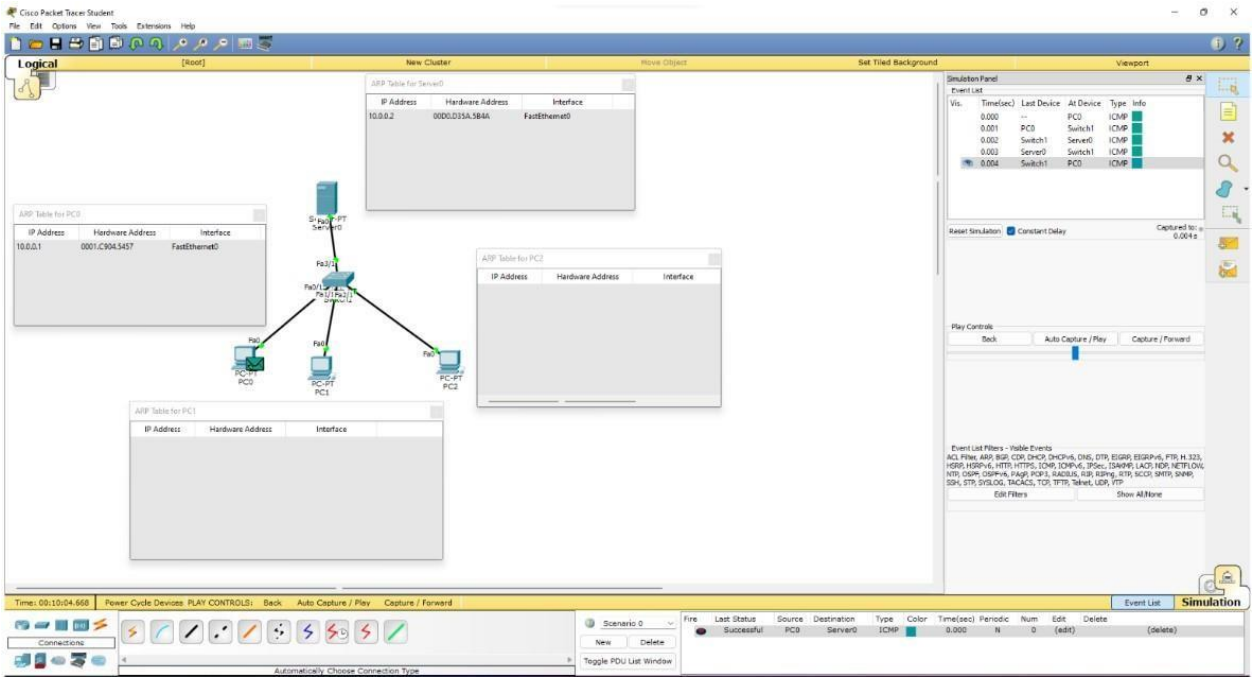
→ When we Ping PC and Server the address of server is known to PC & vice-versa.

→ when we Ping b/w other two PC's simultaneously the address of each other are known.

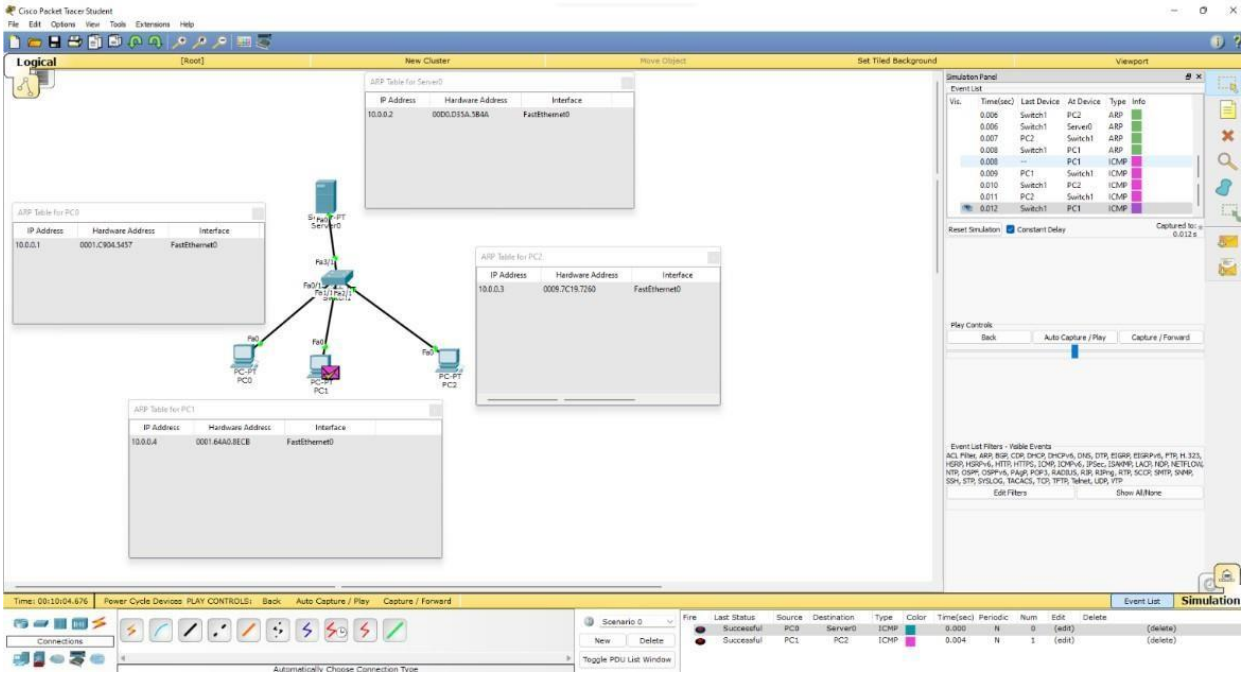
→ Every time a host requires a MAC address in order to send a packet to another host in the LAN, it checks its ARP cache to see if the IP to MAC address translation address already exists of the translation does not exists it performs ARP.

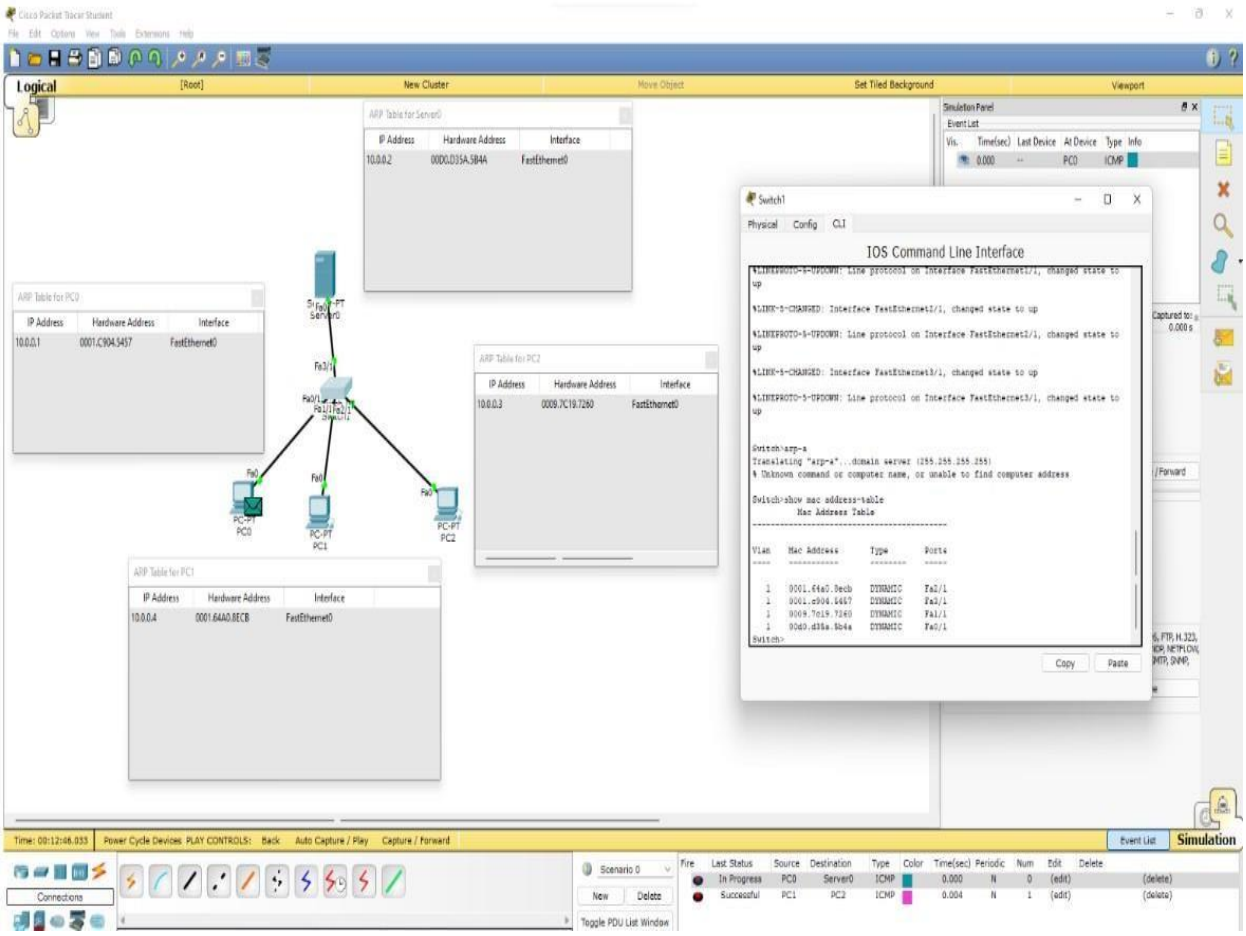
18/8

TOPOLOGY:



OUTPUT:

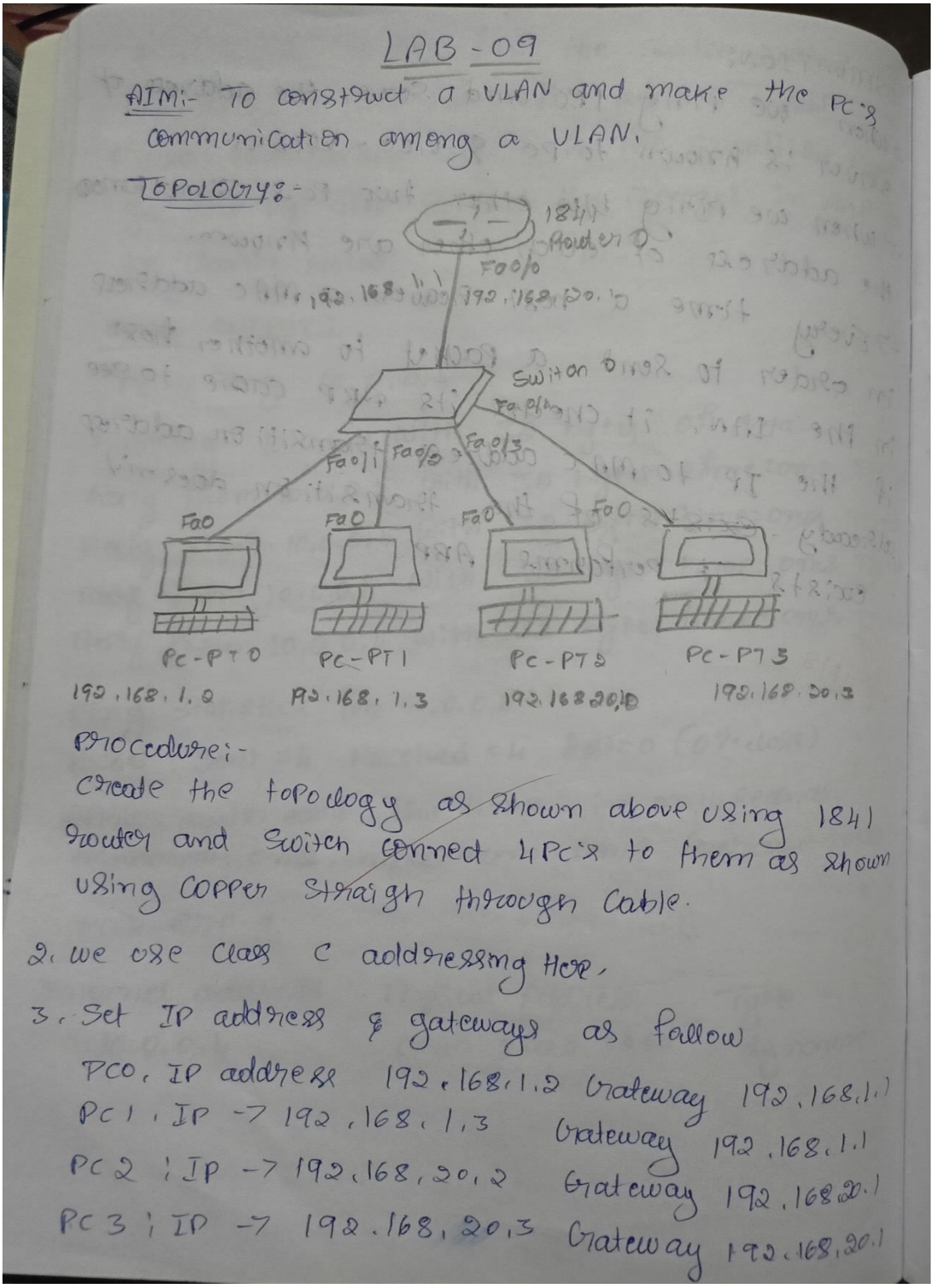



Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
ARP Table for Server0
IP Address
Hardware Address
Interface
10.0.0.2
0060.D35A.5B4A
FastEthernet0
ARP Table for PC0
10.0.0.1
0001.C904.5457
ARP Table for PC2
10.0.0.3
0009.7C19.7260
ARP Table for PC1
10.0.0.4
0001.64A0.8ECB
Switch1
Physical
Config
CLI
IOS Command Line Interface
%LINK-5-CHANGED: Interface FastEthernet2/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet2/1, changed state to up
%LINK-5-CHANGED: Interface FastEthernet3/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet3/1, changed state to up
Switch>arp -a
Translating "arp-a"...domain server (255.255.255.255)
% Unknown command or computer name, or unable to find computer address
Switch>show mac address-table
Mac Address Table
Vlan Mac Address Type Ports
1 0001.64a0.8ecb DYNAMIC Fa2/1
1 0001.c904.5457 DYNAMIC Fa2/1
1 0009.7c19.7260 DYNAMIC Fa2/1
1 00d0.d35a.5b4a DYNAMIC Fa3/1
Switch>
Copy
Paste
PC-PT PC0
PC-PT PC1
PC-PT PC2
Server-PT Server0
Time: 00:12:46.033
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 Status
Source
Destination
Type
Color
Time(sec)
Periodic
Num
Edit
Delete
In Progress
PC0
Server0
ICMP
0.000
N
0
(edit)
(delete)
Successful
PC1
PC2
ICMP
0.004
N
1
(edit)
(delete)

LAB 9

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

OBSERVATION:



LAB - 09

AIM:- To construct a VLAN and make the PC's communication among a VLAN.

TOPOLOGY:-

Procedure:-

Create the topology as shown above using 1841 router and switch connect 4 PC's to them as shown using copper straight through cable.

2. We use class C addressing here.

3. Set IP address & gateways as follow

PC0, IP address 192.168.1.2 Gateway 192.168.1.1

PC1, IP -> 192.168.1.3 Gateway 192.168.1.1

PC2 ; IP -> 192.168.20.2 Gateway 192.168.20.1

PC3 ; IP -> 192.168.20.3 Gateway 192.168.20.1

4. Go to config tab of switch
   Open VLAN database
   Set VLAN Number = 20
   VLAN name = NEWVLAN
   Click on add.

5. In switch Go to FastEthernet 5/0 under that interfaces as it is connected to router
   Select Trunk and choose 20. NEWVLAN.

6. For Fa0/3 and Fa0/4 Select 20 : NEWLAN and keep access as it is.

7. Open config tab in router.
   Go to VLAN Database.
   Add VLAN Number 20
   VLAN Name : NEWVLAN

8. In Router 0 go to CLI mode.

Router (vlan) # exit
Router # config t
Router (config) # int fa 0/0
Router (config-if)# ip address 192.168.1.1 255.255.255.0
Router (config-if) # No Shut
Router (config-if) exit

Router (config) # int fa 0/0.1
Router (config-subif) # encapsulation dot1q 20
Router (config-subif) # ip address 192.168.20.1 255.255.255.0

Router (config-subif) # No Shut

Router (config-subif) # exit
Router (config) # exit.

Ping OUTPUT:-

PC > Ping 192.168.20.2
Pinging 192.168.20.2 with 32 bytes of data.
Request timed out.
Reply from 192.168.20.2 : bytes : 32 time=0 ms TTL=127
Reply from 192.168.20.2 : bytes : 32 time=0 ms TTL=127
Reply from 192.168.20.2 : bytes : 32 time=0 ms TTL=127

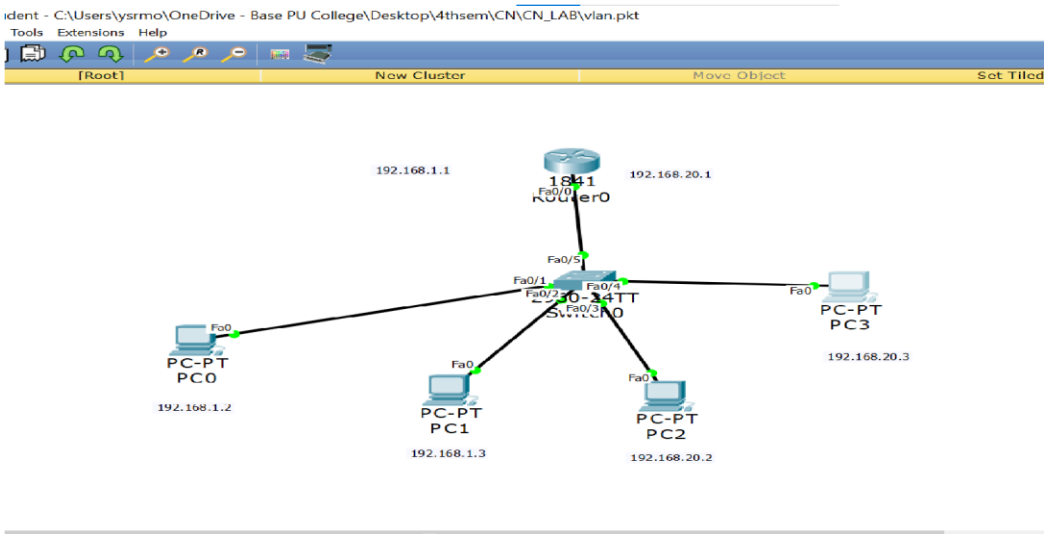
Ping statistics for 192.168.20.2
Packets: Sent = 4, Received = 3 Lost = 1 (25% loss),
Approximate round trip times in milli seconds:
Minimum = 0 ms Maximum = 1 ms, Average = 0 ms.

OBSERVATION:-

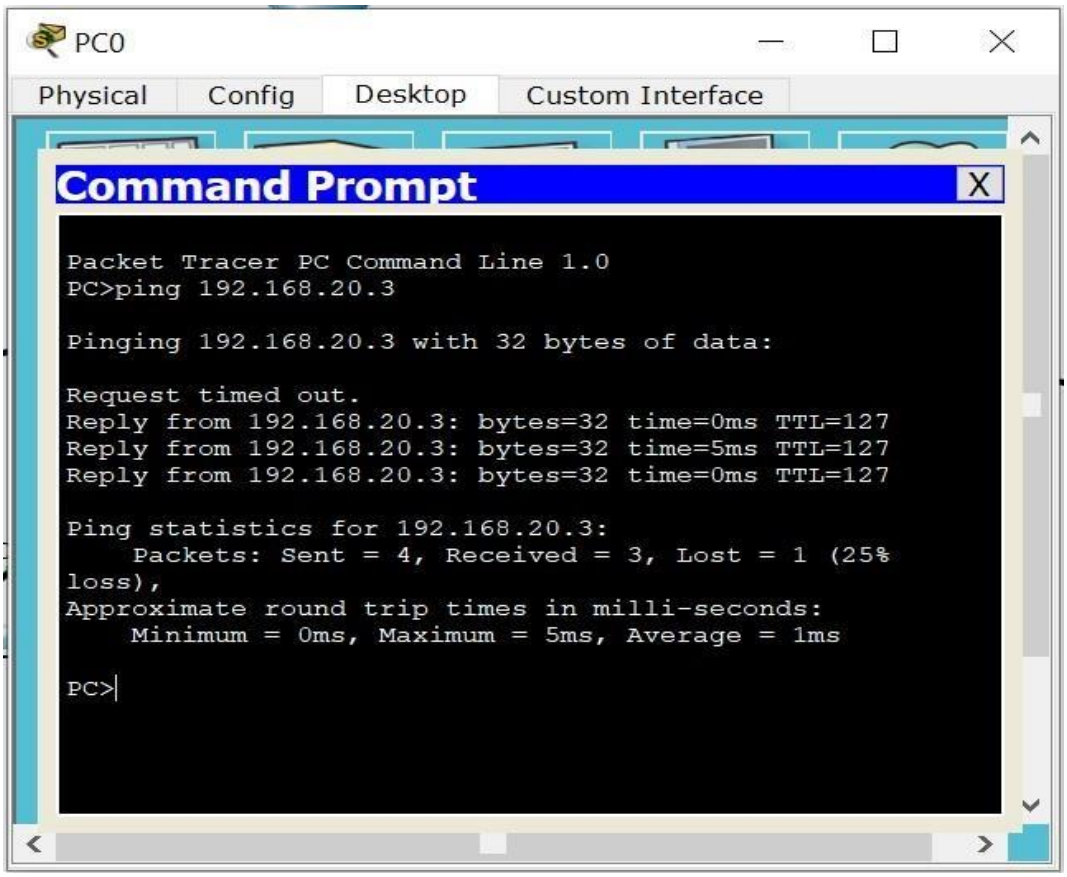
* We can observe that after VLAN is configured we can successfully ping PC2 (192.168.20.2) from PC0 (192.168.1.2)

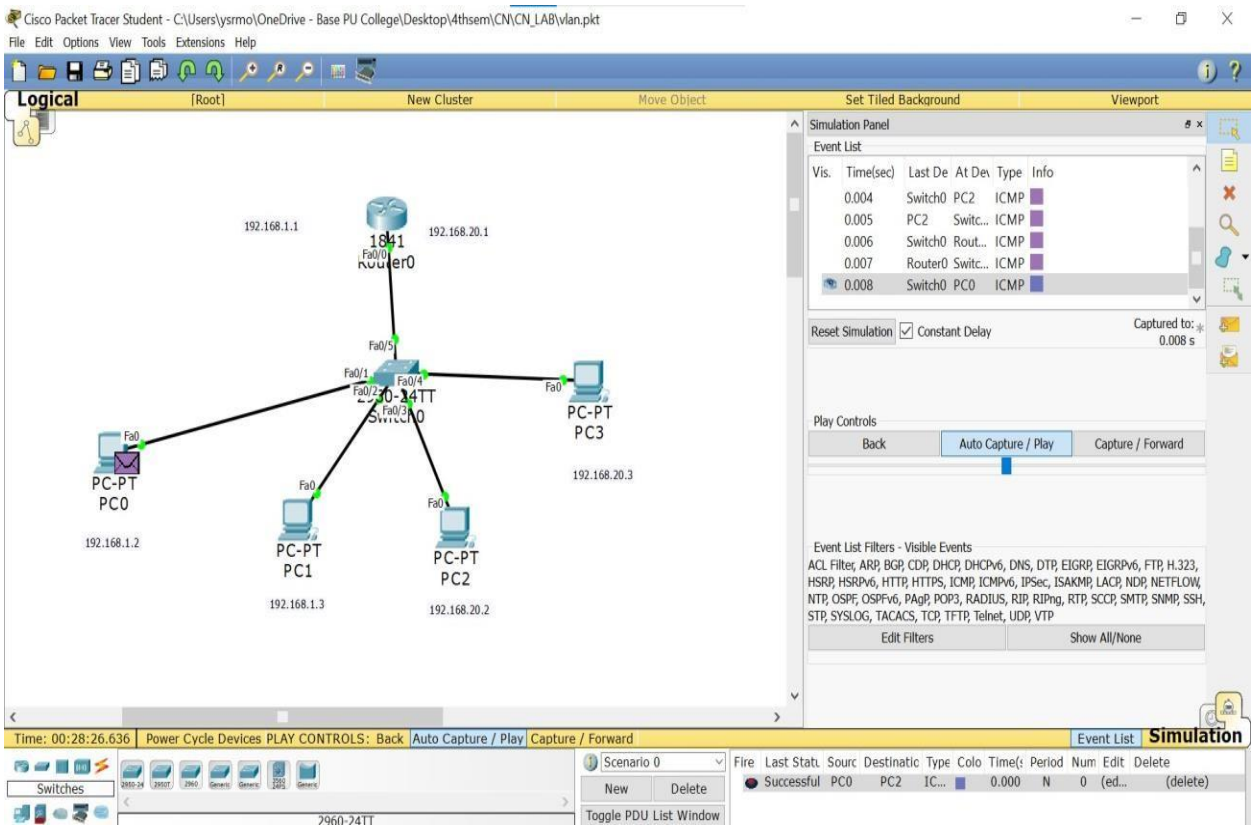
PC2 and PC3 are grouped together and communication among them is done via VLAN.

TOPOLOGY:



OUTPUT:

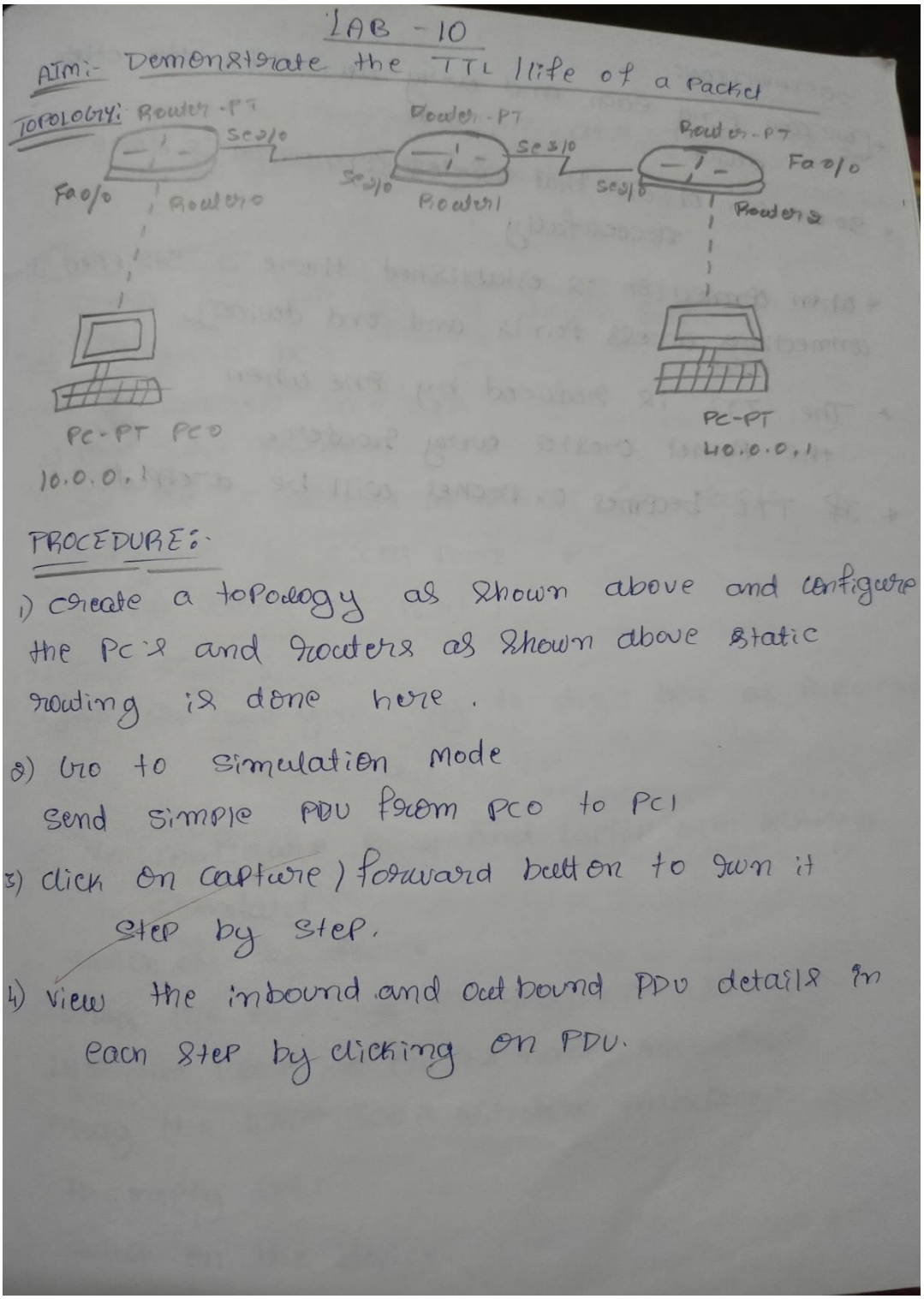



Cisco Packet Tracer Student - C:\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\CN\CN_LAB\vlan.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
192.168.1.1
1841
Router0
192.168.20.1
Fa0/0
Fa0/5
Fa0/1
Fa0/2
Fa0/4
Fa0/3
2960-24TT
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
2960-24TT
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Statu Sourc Destinatic Type Colo Time(s Period Num Edit Delete
Successful PC0 PC2 IC... 0.000 N 0 (ed... (delete)

LAB 10

Demonstrate the TTL/ Life of a Packet.

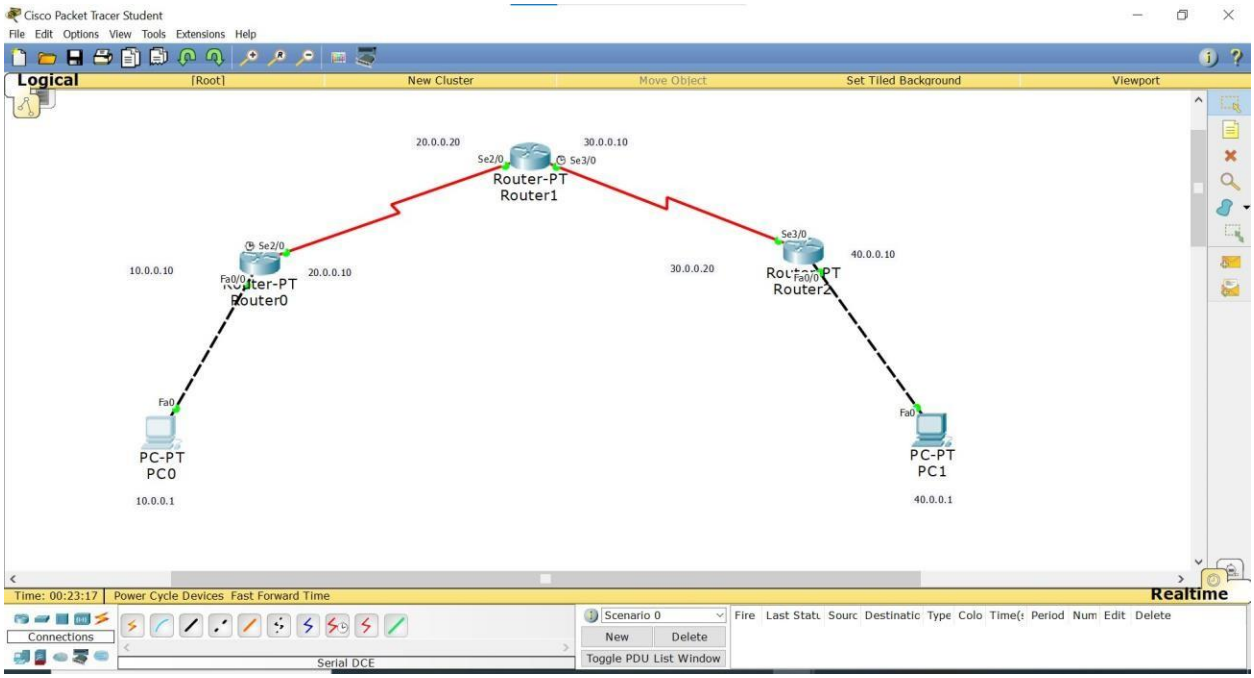
OBSERVATION:



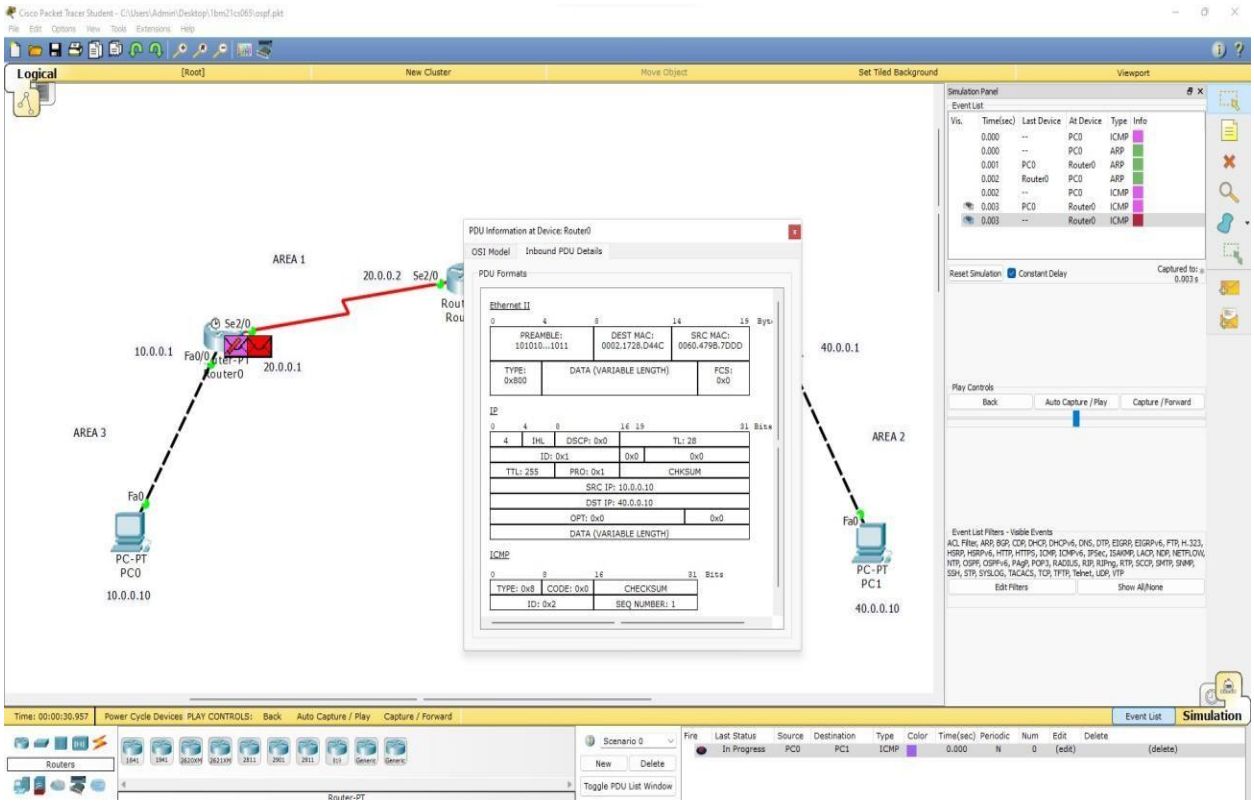
OBSERVATION:-

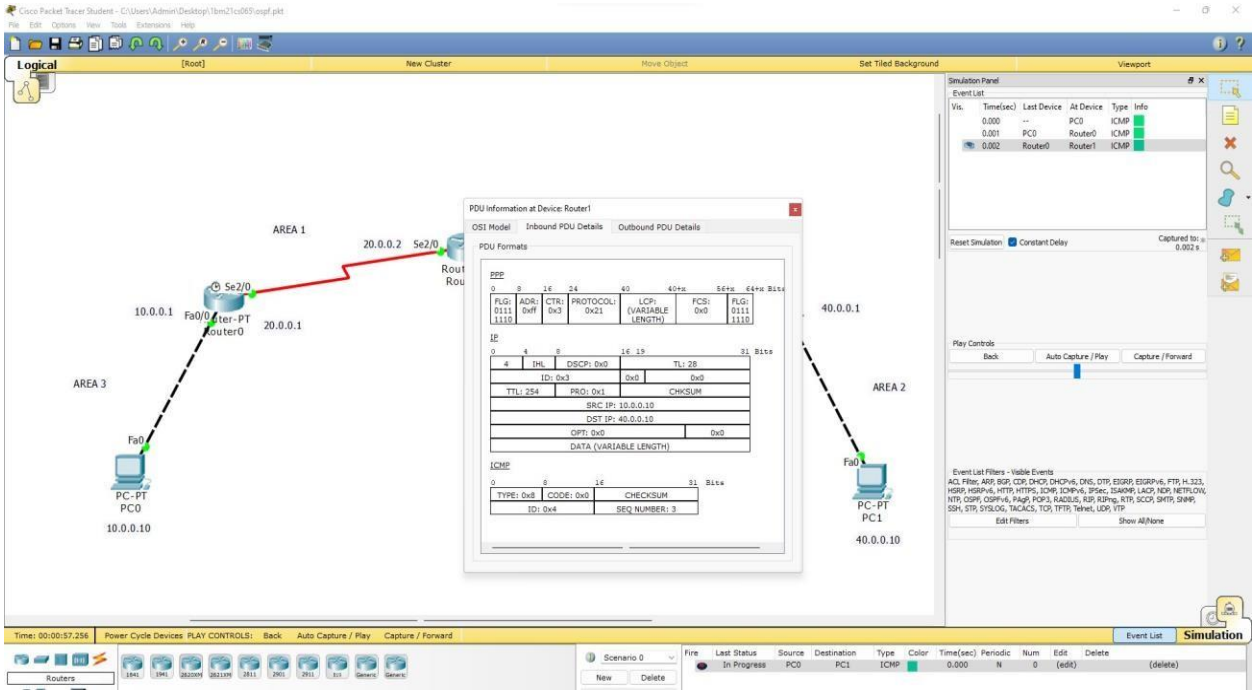
* We can Ping each and every device to the other device.
* So we can observe that wireless connection is done successfully.
* When connection is established there is stripped line connecting access points and end devices.
* The TTL is reduced by one when the packet crossed every router.
* If TTL becomes 0, packet will be dropped.

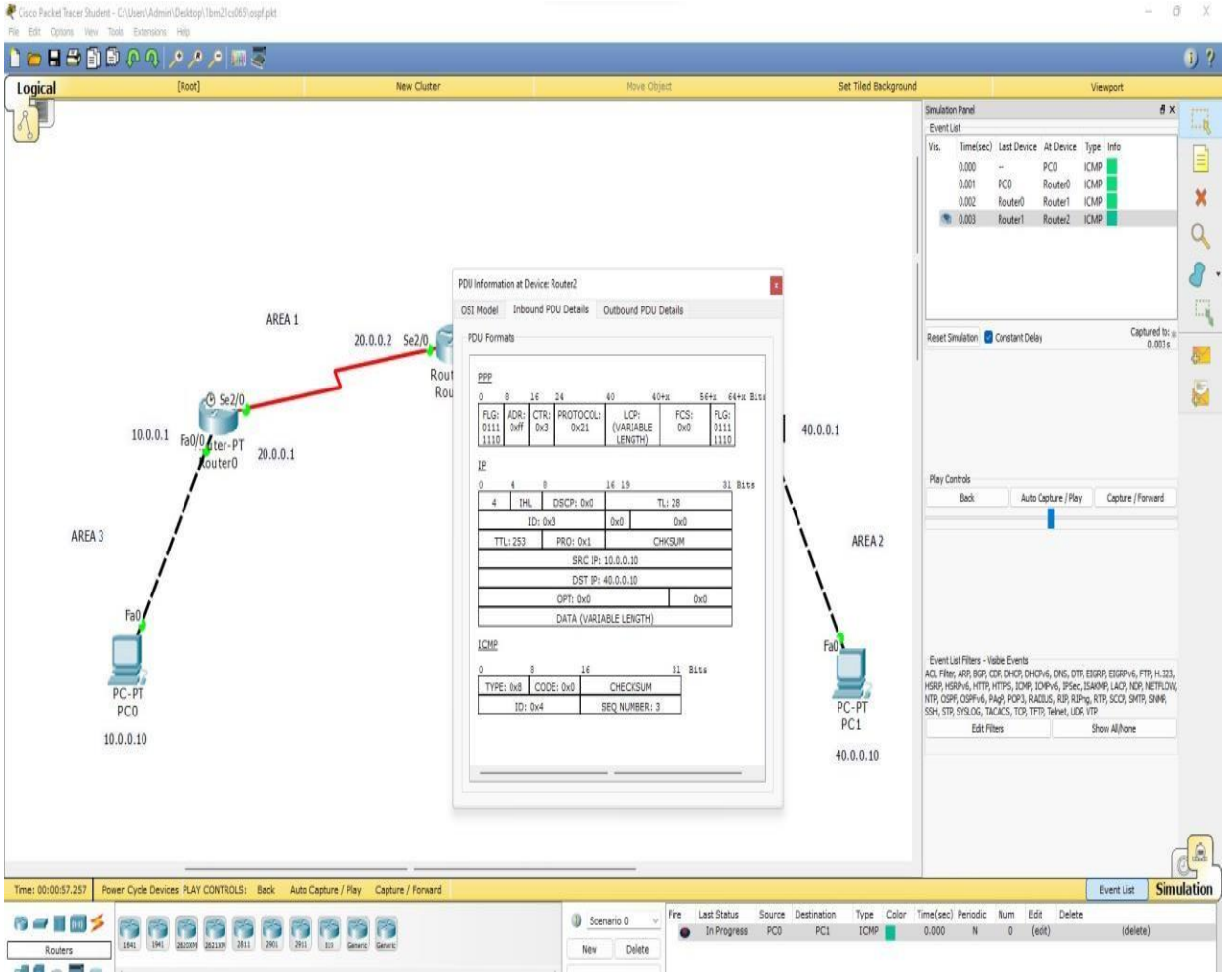
TOPOLOGY:

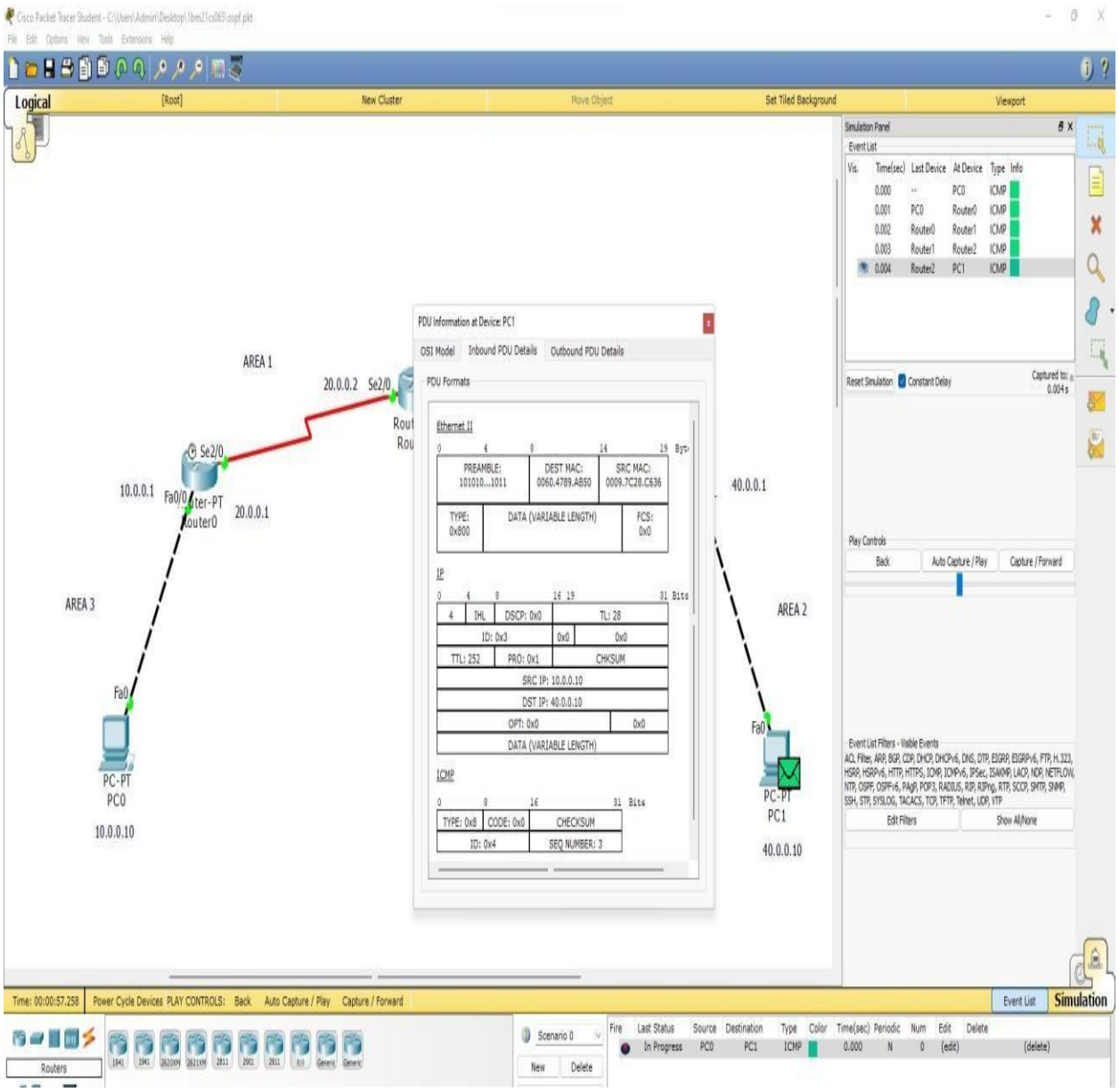


OUTPUT:




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
20.0.0.1
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
Simulation Panel
Event List
Time: 00:00:57.256
Power Cycle Devices
PLAY CONTROLS:
Back
Auto Capture / Play
Capture / Forward
Event List
Simulation

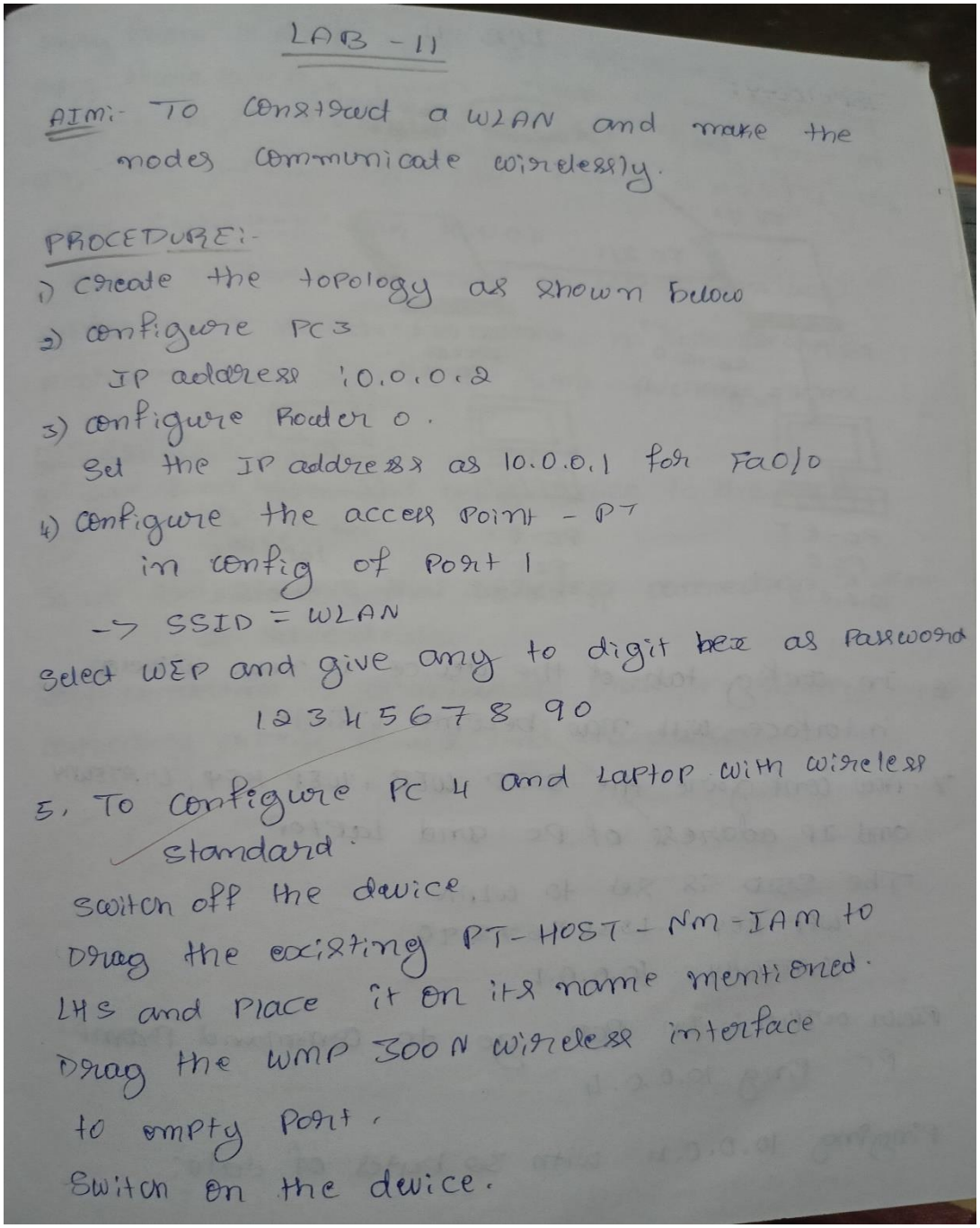

Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
AREA 1
AREA 2
AREA 3
20.0.0.2
Se2/0
10.0.0.1
Fa0/0
20.0.0.1
40.0.0.1
Fa0
PC-PT
PC0
10.0.0.10
PC-PT
PC1
40.0.0.10
PDU Information at Device: Router2
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
PPP
IP
ICMP
Simulation Panel
Event List
Time: 00:00:57.257
Power Cycle Devices
PLAY CONTROLS:
Back
Auto Capture / Play
Capture / Forward
Event List
Simulation


Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
AREA 1
20.0.0.2
Se2/0
Se2/0
10.0.0.1
Fa0/0
20.0.0.1
AREA 3
AREA 2
40.0.0.1
Fa0
PC-PT
PC0
10.0.0.10
Fa0
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
DSCP: 0x0
TL: 28
ID: 0x3
CHKSUM
TTL: 252
PRO: 0x1
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
Reset Simulation
Constant Delay
Play Controls
Back
Auto Capture / Play
Capture / Forward
Edit Filters
Show All/None
Time: 00:00:57.258
Power Cycle Devices
PLAY CONTROLS:
Back
Auto Capture / Play
Capture / Forward
Event List
Simulation
Routers
Scenario 0
New
Delete
In Progress
PC0
PC1
ICMP
(edit)
(delete)

LAB 11

To construct a WLAN and make the nodes communicate wirelessly

OBSERVATION:



LAB - 11

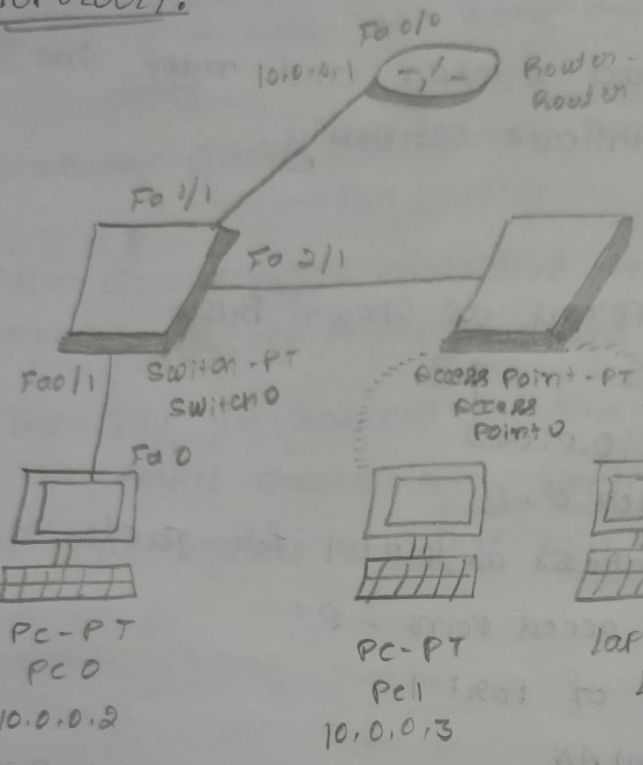
AIM: To construct a WLAN and make the nodes communicate wirelessly.

PROCEDURE:-

1) Create the topology as shown below
2) configure PC3
   IP address : 10.0.0.2
3) configure Router 0.
   Set the IP address as 10.0.0.1 for Fa0/0
4) Configure the access Point - PT
   in config of Port 1
   -> SSID = WLAN
Select WEP and give any 10 digit hex as Password
   1234567890
5. To configure PC 4 and Laptop with wireless standard.
   Switch off the device.
   Drag the existing PT-HOST-NM-1AM to LHS and Place it on its name mentioned.
   Drag the WMP 300N wireless interface to empty Port.
   Switch on the device.

LAB - 11

TOPOLOGY:-



6. In config tab of the device a new wireless interface will now become visible.

7. Now configure the SSID, WEP, WEP KEY, GATEWAY and IP address of Pc and Laptop

The SSID is set to WLAN

WEP KEY = 1234567890

GATEWAY = 10.0.0.1

PING OUTPUT :- In PC0 go to Command Prompt

PC Ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:

Reply from 10.0.0.4: bytes = 32 time = 24ms TTL = 128
Reply from 10.0.0.4: bytes = 32 time = 5ms TTL = 128
Reply from 10.0.0.4: bytes = 32 time = 5ms TTL = 128
Reply from 10.0.0.4: bytes = 32 time = 12ms TTL = 128

Ping statistics for 10.0.0.4

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

Approximate round trip time. in milli seconds:

minimum = 5ms. maximum = 24ms. Average = 14ms.

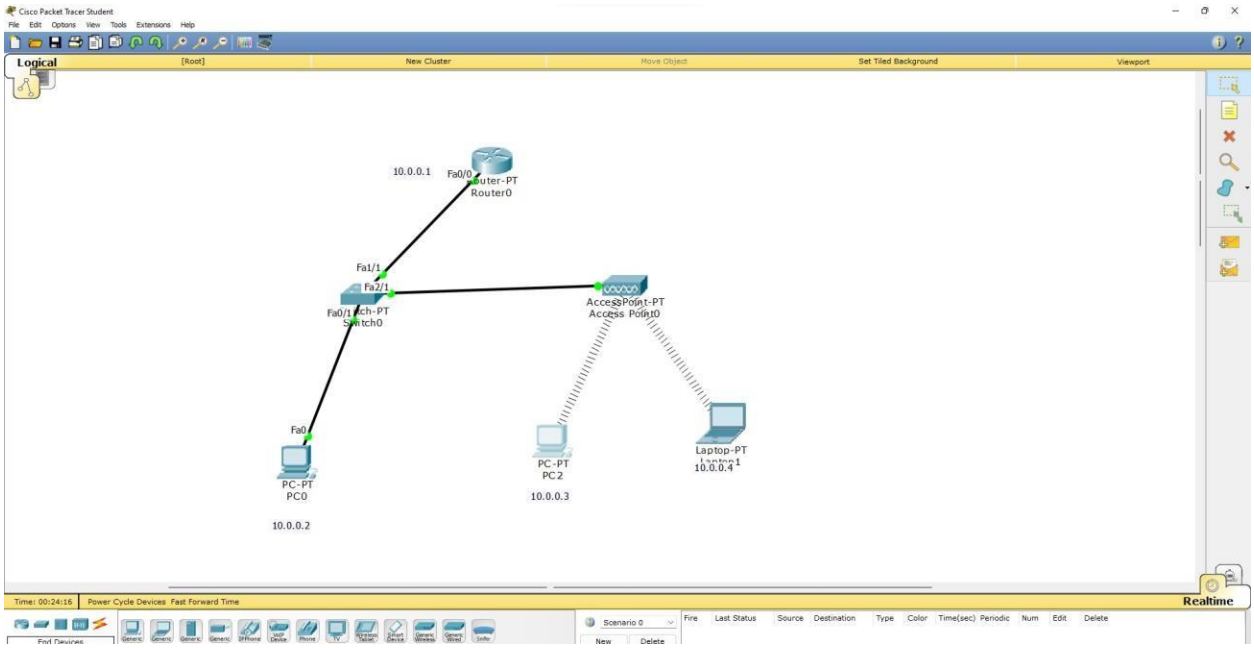
OBSERVATION:

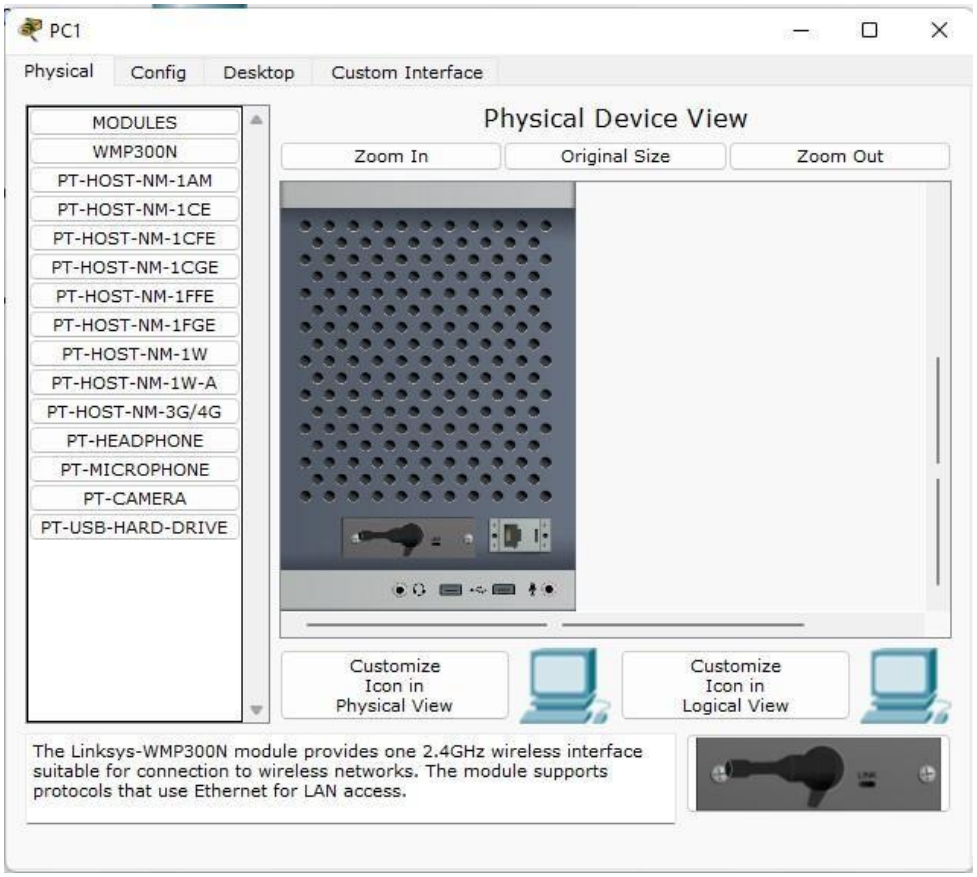
We can Ping each and every device to the other device.

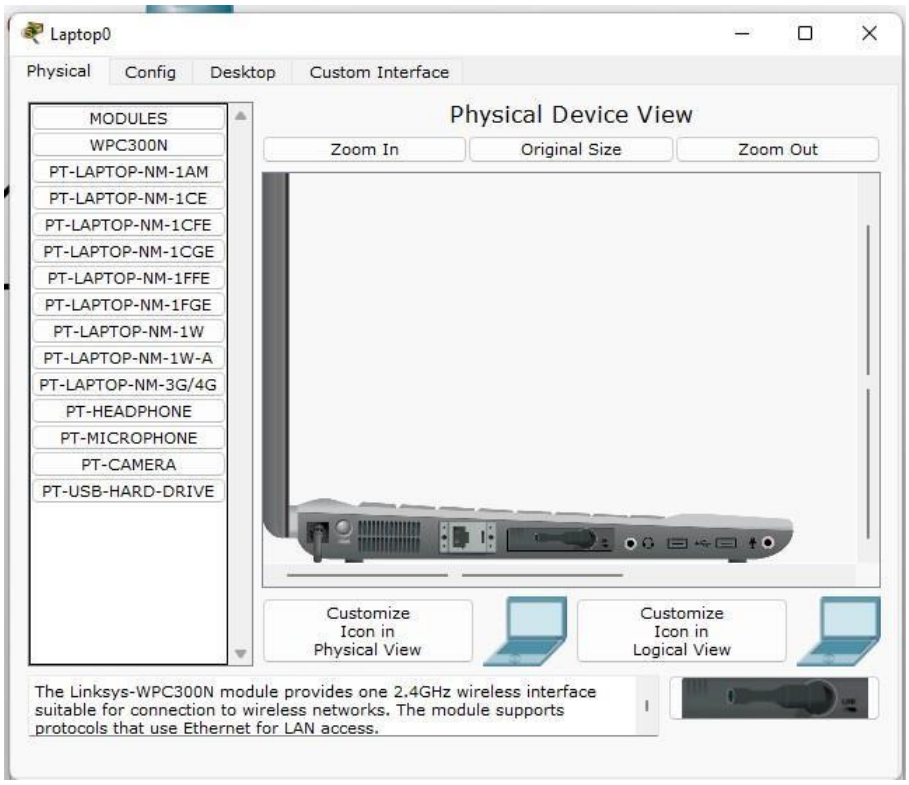
So we can observe that wireless connection is done successfully.

When connection is established there is stripped lines connecting access points and end devices.

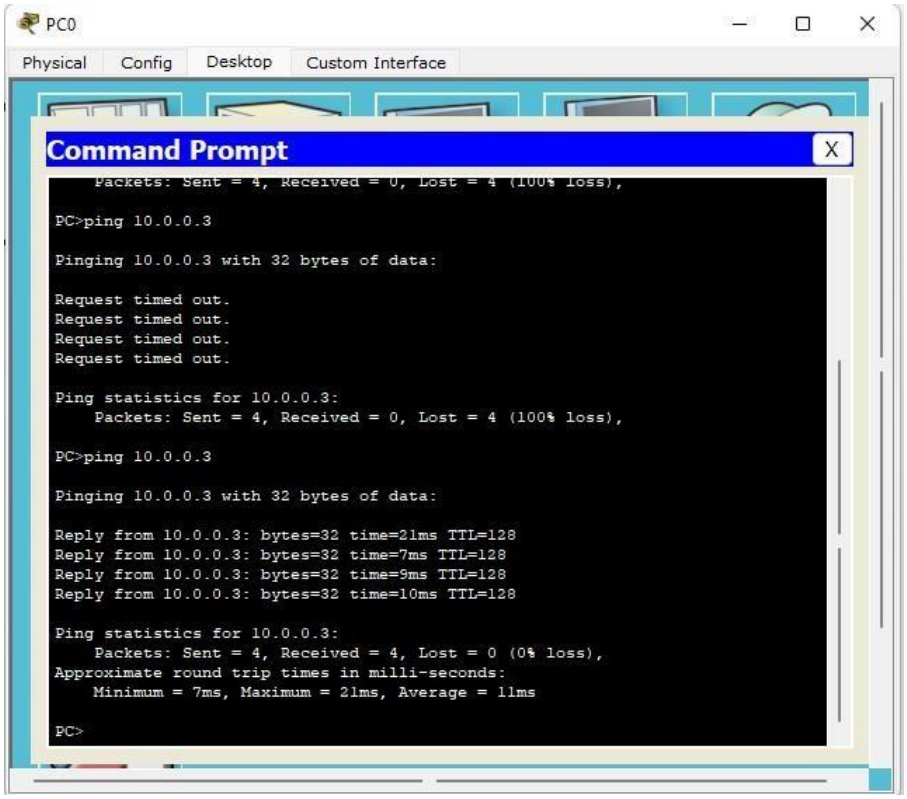
TOPOLOGY:


Logical
10.0.0.1
Fa0/0
Router-PT
Router0
Fa1/1
Fa2/1
Fa0/1
Switch0
AccessPoint-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.

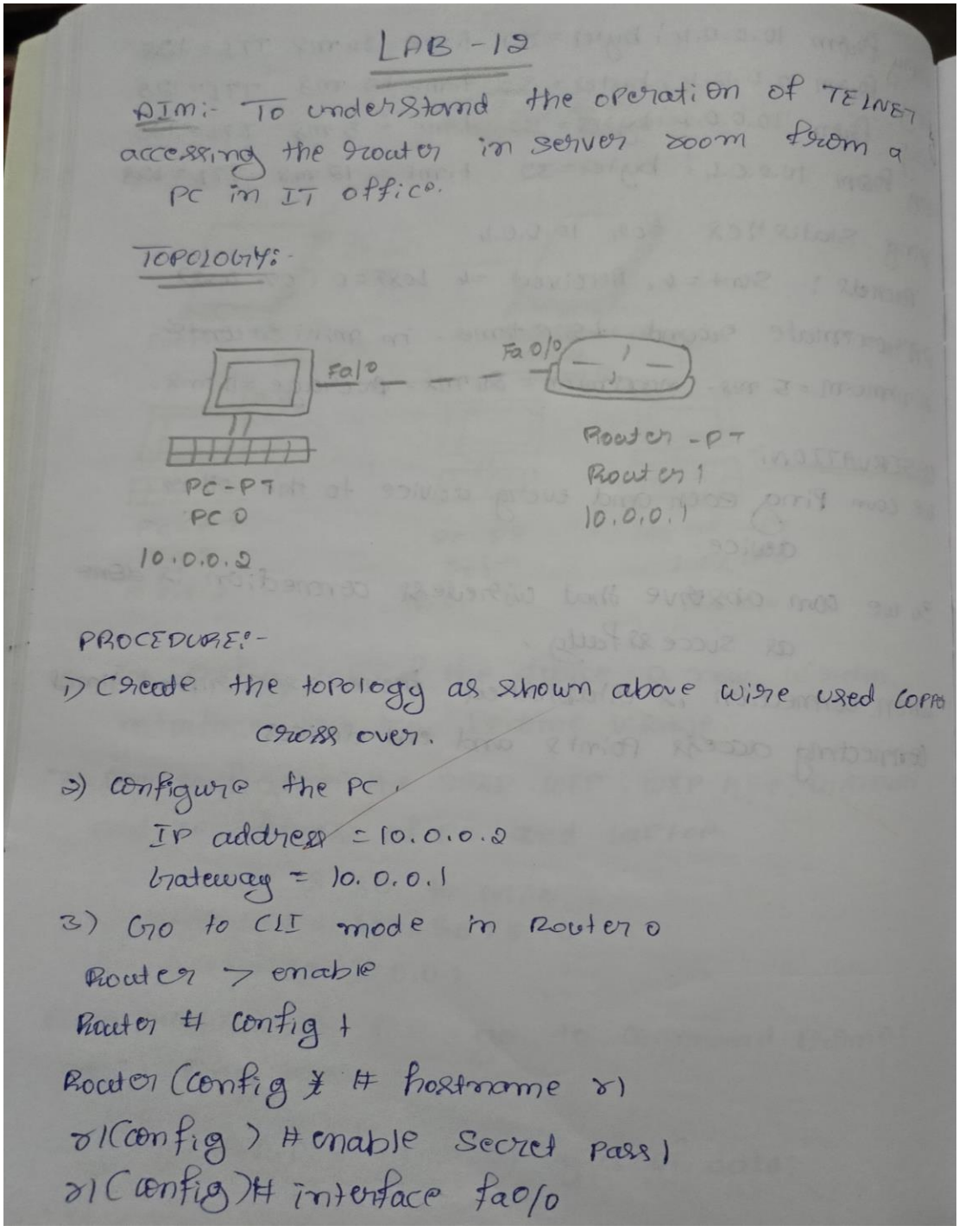
OUTPUT:


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>

LAB 12

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

OBSERVATION:



LAB - 12

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

TOPOLOGY:-

PROCEDURE:-

1) Create the topology as shown above wire used copper cross over.

2) Configure the PC.
   IP address = 10.0.0.2
   Gateway = 10.0.0.1

3) Go to CLI mode in Router 0
   Router > enable
   Router # config t
   Router (config) # hostname r1
   r1(config) # enable secret pass1
   r1(config)# interface fa0/0

R1(config)# ipaddress 10.0.0.1 255.0.0.0
R1(config-if)# no shut
R1(config-if)# line vty 0 5
R1(config-line)# login
R1(config-line)# Password Pass 0
R1(config-line)# exit
R1# wr

PING OUTPUT in PC0

we can successfully ping 10.0.0.1 from

PC> telnet 10.0.0.1

Trying 10.0.0.1 ... open

User Access Verification.

Password : Pass 0

R1> enable

Password : Pass 1

R1# Show IP route

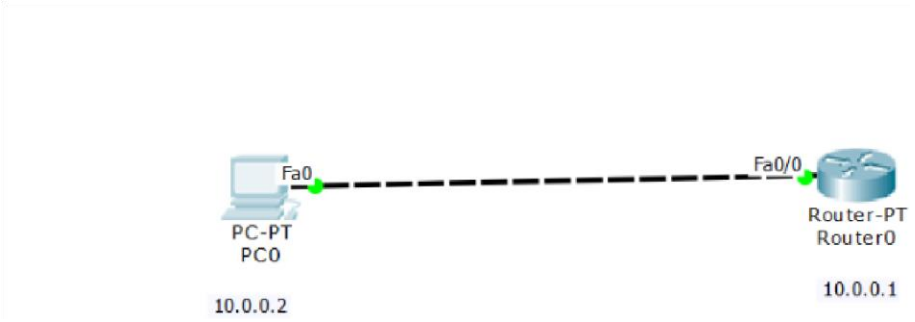
C 10.0.0.0/8 is directly connected, Fast Ethernet 0/0

OBSERVATION:-

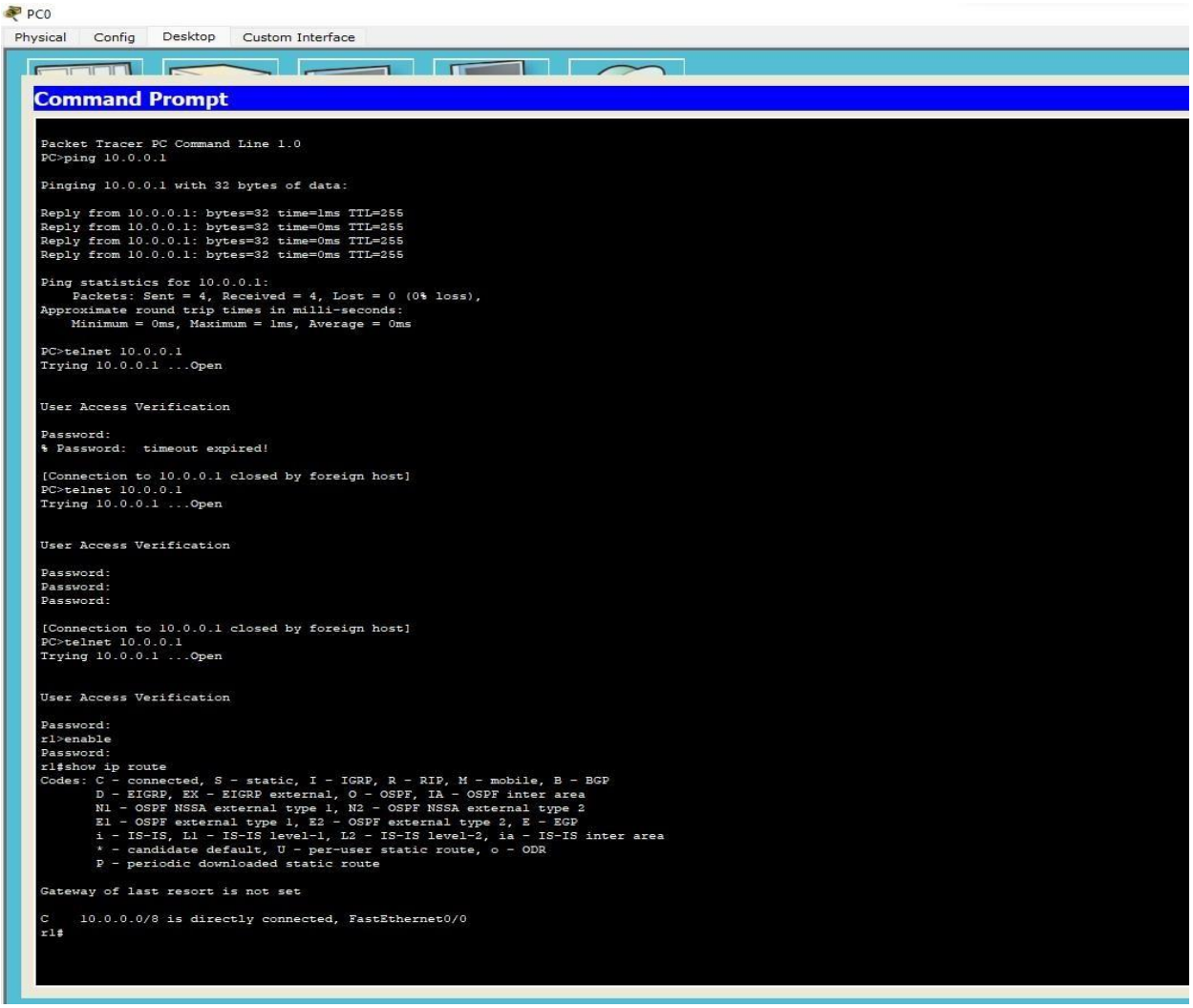
we can observe that the admin in PC is able to run commands as run in router in CLI and see the result from the PC.

So with the help of TELNET, we can access the router in server room from a PC.

TOPOLOGY:



OUTPUT:



LAB 13

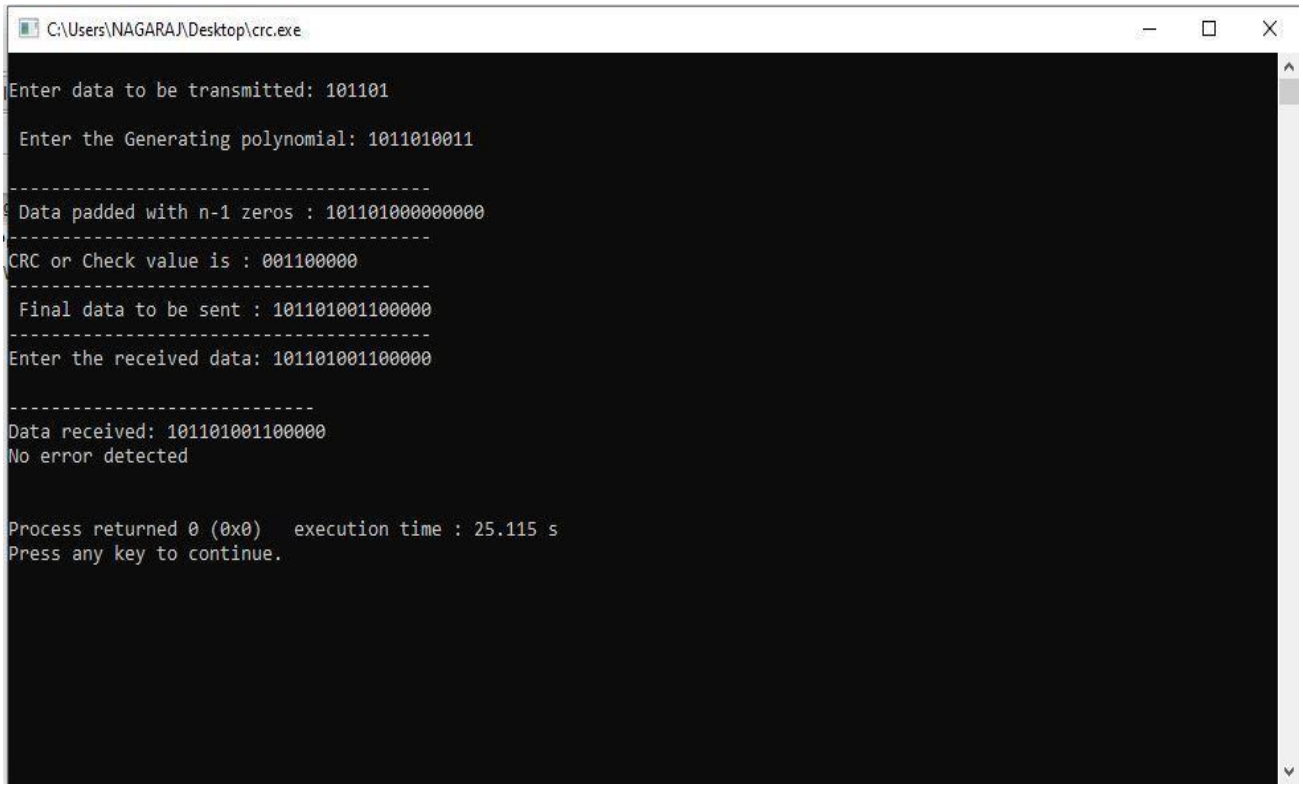
Write a program for error detecting code using CRCCCITT (16-bits).

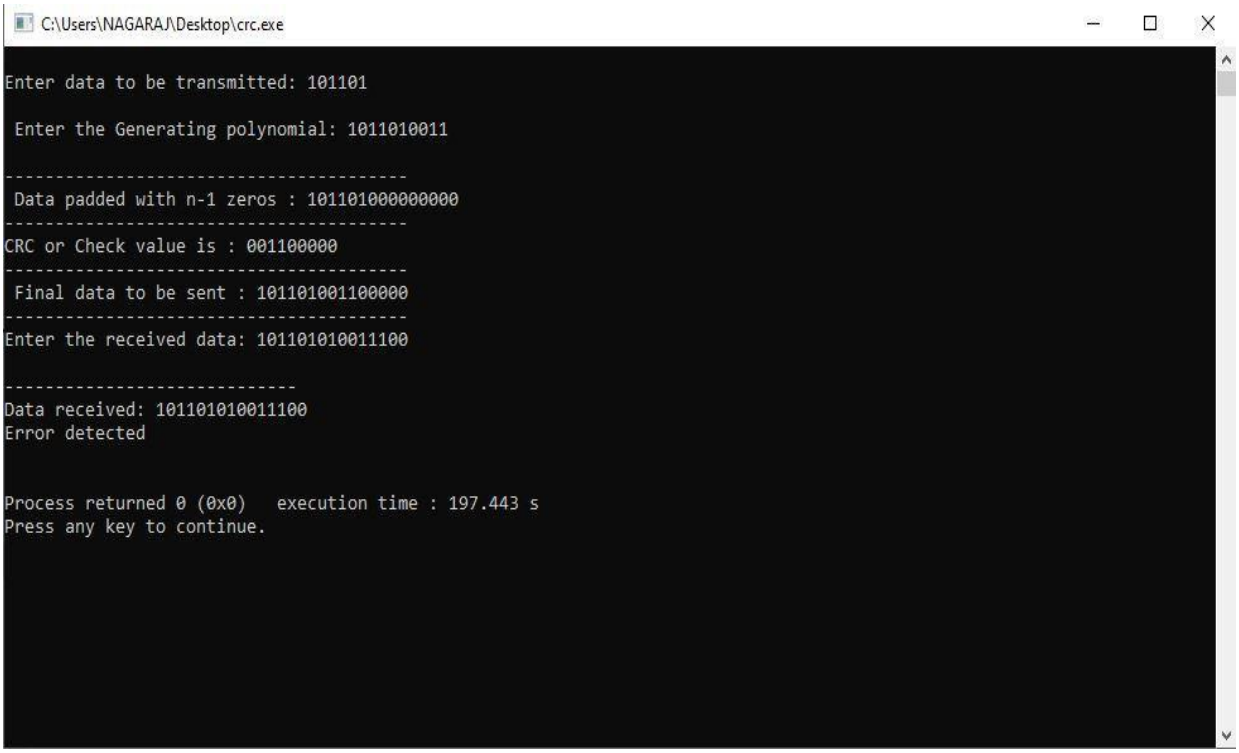
Code:

```
#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 :


C:\Users\NAGARAJ\Desktop\crc.exe
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.

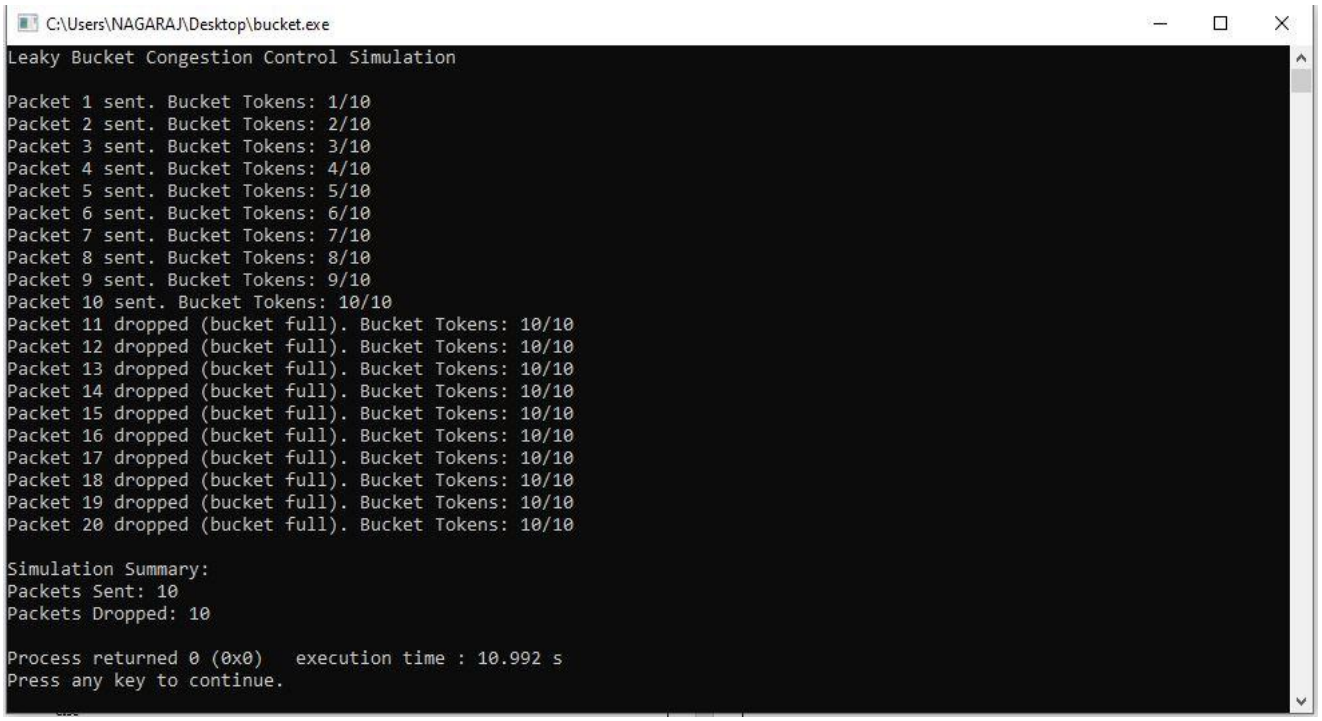

C:\Users\NAGARAJ\Desktop\crc.exe
Enter data to be transmitted: 101101
Enter the Generating polynomial: 1011010011
Data padded with n-1 zeros : 101101000000000
CRC or Check value is : 001100000
Final data to be sent : 101101001100000
Enter the received data: 101101010011100
Data received: 101101010011100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

b. Write a program for congestion control using Leaky bucket algorithm.

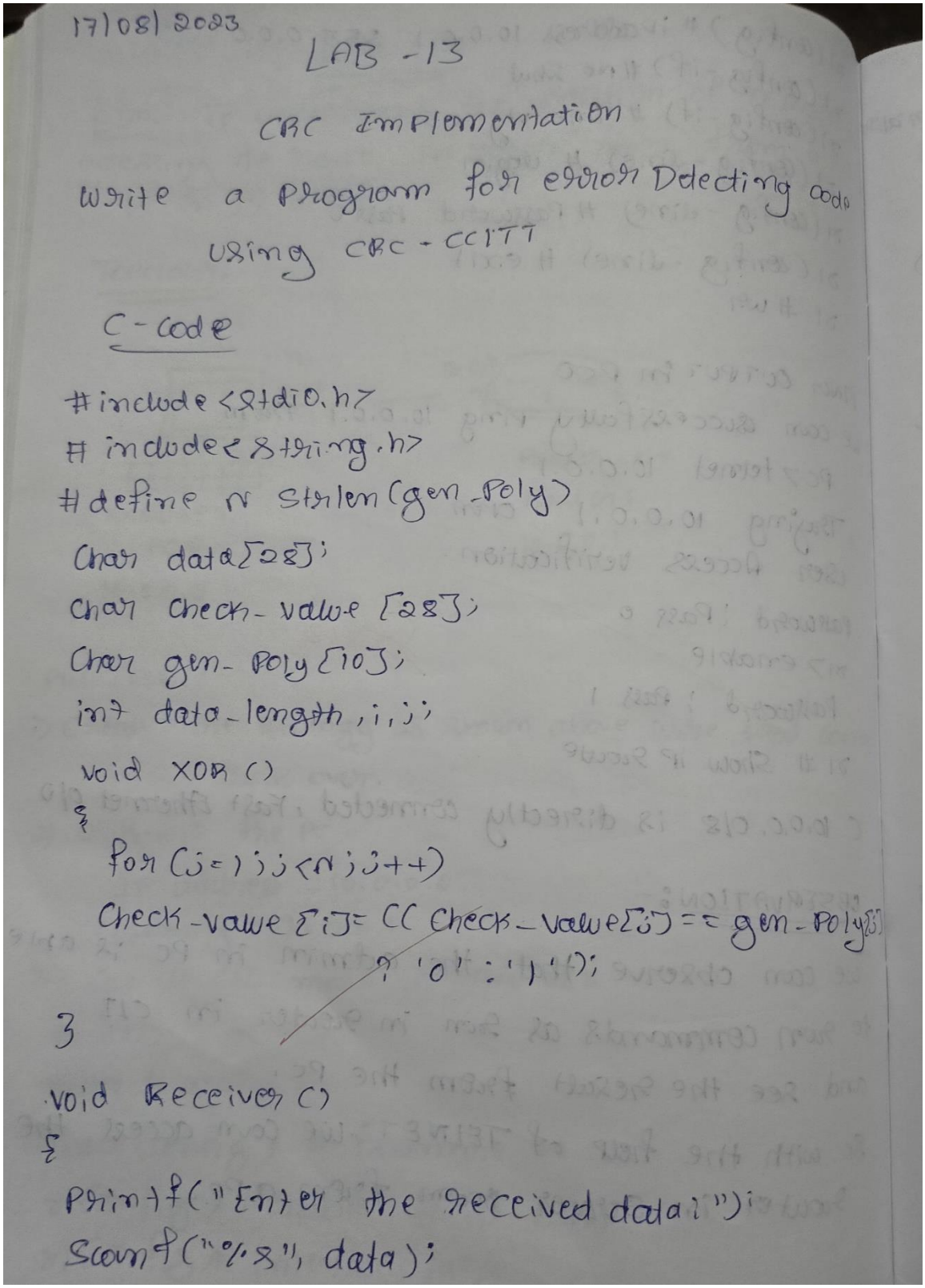
Code :

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#define BUCKET_SIZE 10
#define RATE 1
#define PACKETS 20

int main() {
   int bucket = 0;
   int sent = 0;
   int dropped = 0;
   int i;

   printf("Leaky Bucket Congestion Control Simulation\n\n");

   for (i = 0; i < PACKETS; i++) {
      usleep(500000);

      if (bucket < BUCKET_SIZE) {
         printf("Packet %d sent. Bucket Tokens: %d/%d\n", i + 1, bucket + 1,
BUCKET_SIZE);
         bucket++;
         sent++;
      } else {
         printf("Packet %d dropped (bucket full). Bucket Tokens: %d/%d\n", i + 1,
bucket, BUCKET_SIZE);
         dropped++;
      }
   }

   printf("\nSimulation Summary:\n");
   printf("Packets Sent: %d\n", sent);
   printf("Packets Dropped: %d\n", dropped);

   return 0;
}
```

Output :


C:\Users\NAGARAJ\Desktop\bucket.exe
Leaky Bucket Congestion Control Simulation
Packet 1 sent. Bucket Tokens: 1/10
Packet 2 sent. Bucket Tokens: 2/10
Packet 3 sent. Bucket Tokens: 3/10
Packet 4 sent. Bucket Tokens: 4/10
Packet 5 sent. Bucket Tokens: 5/10
Packet 6 sent. Bucket Tokens: 6/10
Packet 7 sent. Bucket Tokens: 7/10
Packet 8 sent. Bucket Tokens: 8/10
Packet 9 sent. Bucket Tokens: 9/10
Packet 10 sent. Bucket Tokens: 10/10
Packet 11 dropped (bucket full). Bucket Tokens: 10/10
Packet 12 dropped (bucket full). Bucket Tokens: 10/10
Packet 13 dropped (bucket full). Bucket Tokens: 10/10
Packet 14 dropped (bucket full). Bucket Tokens: 10/10
Packet 15 dropped (bucket full). Bucket Tokens: 10/10
Packet 16 dropped (bucket full). Bucket Tokens: 10/10
Packet 17 dropped (bucket full). Bucket Tokens: 10/10
Packet 18 dropped (bucket full). Bucket Tokens: 10/10
Packet 19 dropped (bucket full). Bucket Tokens: 10/10
Packet 20 dropped (bucket full). Bucket Tokens: 10/10
Simulation Summary:
Packets Sent: 10
Packets Dropped: 10
Process returned 0 (0x0) execution time : 10.992 s
Press any key to continue.

OBSERVATION :



17/08/2023

LAB - 13

CRC Implementation

Write a Program for error Detecting code using CRC - CCITT

C - 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("\n Error detected \n\n");
else
printf("\n No 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]=data[i++];
}while(i<=data_length+N-1);
}
int main()
{
printf("\n Enter the 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("\n CRC 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 : 101010
Enter the divisor Polynomial : 1011
Data Padded with n-1 zeroes : 101010000
CRC value is : 001
Final codeword to be sent = 101010001

Enter the received data : 10001000
Error detected

Enter data to be transmitted : 101100
Enter the divisor Polynomial : 1001
Data Padded with n-1 zeroes : 101100000
CRC value is : 001
Final codeword to be sent : 101100001

Enter the received data : 101100001

No Error Detected.

18/8

Write a program for congestion control using Leaky Bucket algorithm

C- code

```
#include<stdio.h>
int main()
{
   int incoming, outgoing, buck_size, n, store
Printf("Enter Bucket size:");
Scanf("%d", &buck_size);
Printf ("Enter outgoing size:");
Scanf("%d", &outgoing);
Printf("Enter the number of inputs:")
Scanf("%d", &n);
while (n!=0)
{  Printf("Enter the incoming bucket size:")
   Scanf("%d", &incoming);
   if (incoming <= (buck_size - store))
   {
     store += incoming;
    Printf("Bucket buffer size %d out of
          %d' \n", store, buck_size)

  }
  else
  {
```

```
        incoming -(buck_size-store));
printf("Bucket buffer size %d out of %d \n",
        store, buck_size);
    store = store - outgoing
printf("After outgoing %d Packets left out of
    %d in buffer \n", store, buck_size);
    n--;
  }
}
```

18/8

OUTPUT:-

Enter bucket size: 5000
Enter outgoing rate: 2000
Enter number of inputs: 2
Enter the incoming Packet size: 3000
Bucket Buffer size 3000 out of 5000
After outgoing 1000 Packets left out of 5000
in buffer.
Enter the incoming Packet size: 1000
Bucket buffer size 2000 out of 5000
After outgoing 0 Packets left out of 5000
in Buffer.

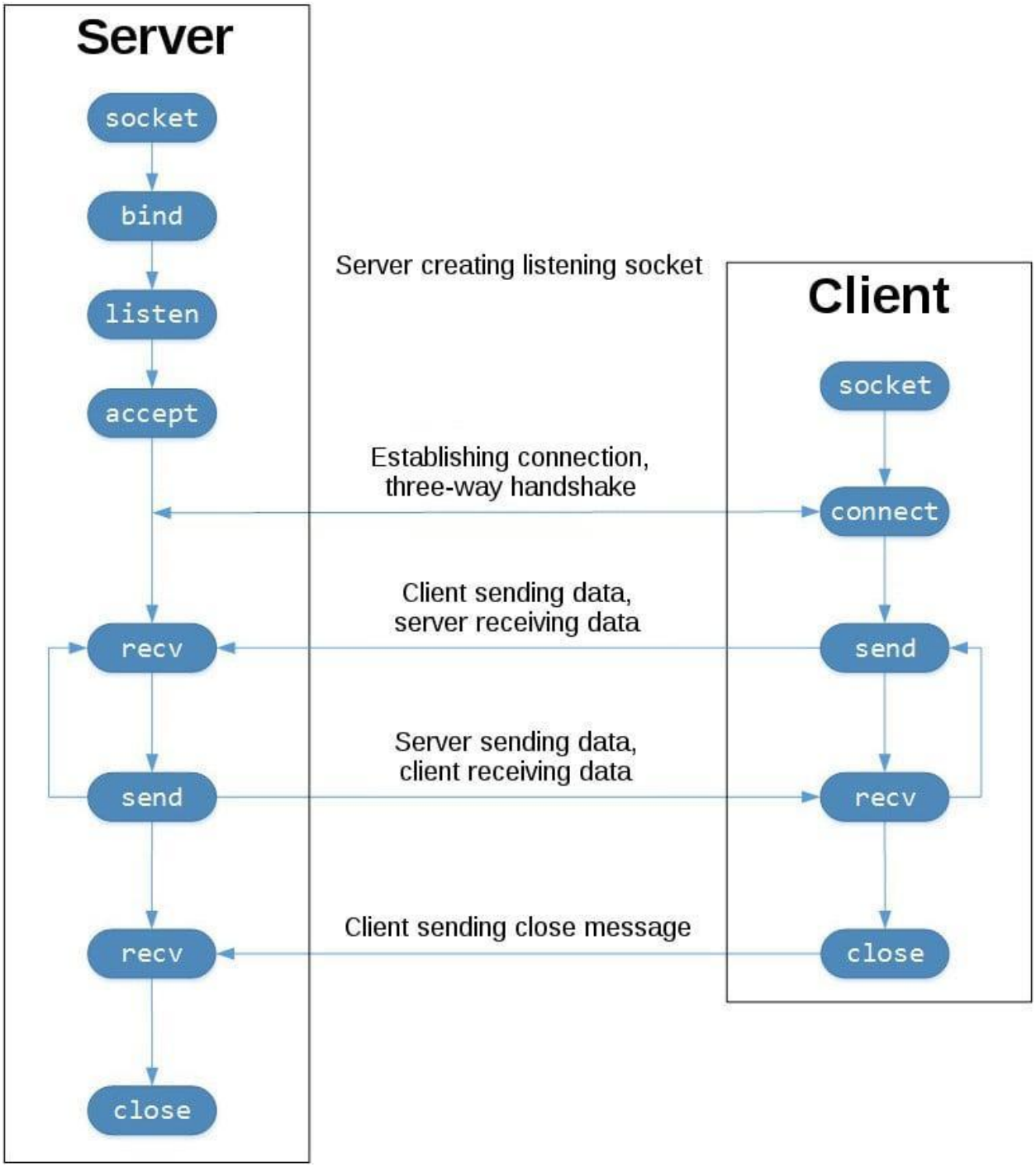
CYCLE 3

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.**

2. **Server :**
3. A server has a bind() method which binds it to a specific IP and port so that it can listen to incoming requests on that IP and port. A server has a listen() method which puts the server into listening mode. This allows the server to listen to incoming connections. And last a server has an accept() and close() method. The accept method initiates a connection with the client and the close method closes the connection with the client.
4. First of all, we import socket which is necessary.
5. Then we made a socket object and reserved a port on our pc.
6. After that, we bound our server to the specified port. Passing an empty string means that the server can listen to incoming connections from other computers as well. If we would have passed 127.0.0.1 then it would have listened to only those calls made within the local computer.
7. After that we put the server into listening mode.5 here means that 5 connections are kept waiting if the server is busy and if a 6th socket tries to connect then the connection is refused.
8. At last, we make a while loop and start to accept all incoming connections and close those connections after a thank you message to all connected sockets.

This output shows that our server is working.
Now for the client-side:

- First of all, we make a socket object.
- Then we connect to localhost on port 12345 (the port on which our server runs) and lastly, we receive data from the server and close the connection.
- Now save this file as client.py and run it from the terminal after starting the server script.



The arguments passed to socket() are constants used to specify the address family and socket type. AF_INET is the Internet address family for IPv4. SOCK_STREAM is the socket type for TCP, the protocol that will be used to transport messages in the network. The IP address 127.0.0.1 is the standard IPv4 address for the loopback interface, so only processes on the host will be able to connect to the server. port represents the TCP port number to accept connections on from clients. It should be an integer from 1 to 65535, as 0 is reserved.

The .bind() method is used to associate the socket with a specific network interface and port number

The listen() method has a backlog parameter. It specifies the number of unaccepted connections that the system will allow before refusing new connections. If your server receives a lot of connection requests simultaneously, increasing the backlog value may help by setting the maximum length of the queue for pending connections.

The accept() method blocks execution and waits for an incoming connection. When a client connects, it returns a new socket object representing the connection and a tuple holding the address of the client. The tuple will contain (host, port) for IPv4 connections

SOLUTION:

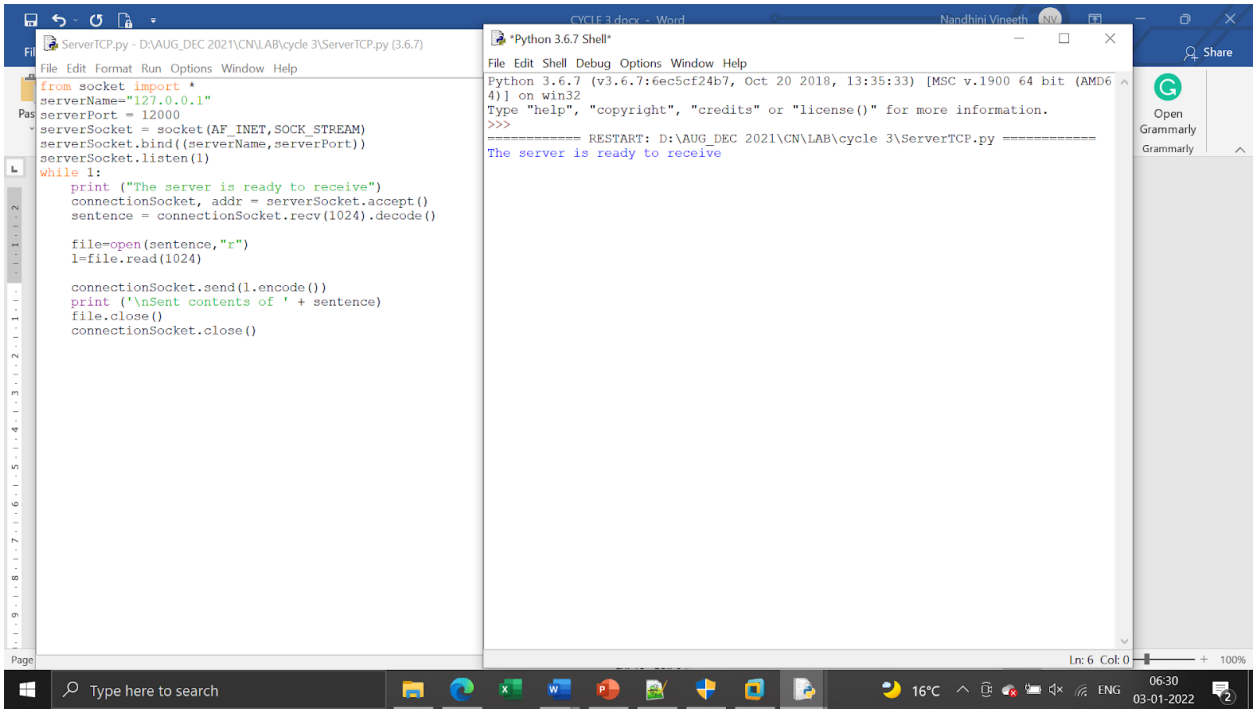
ClientTCP.py

```
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()
```

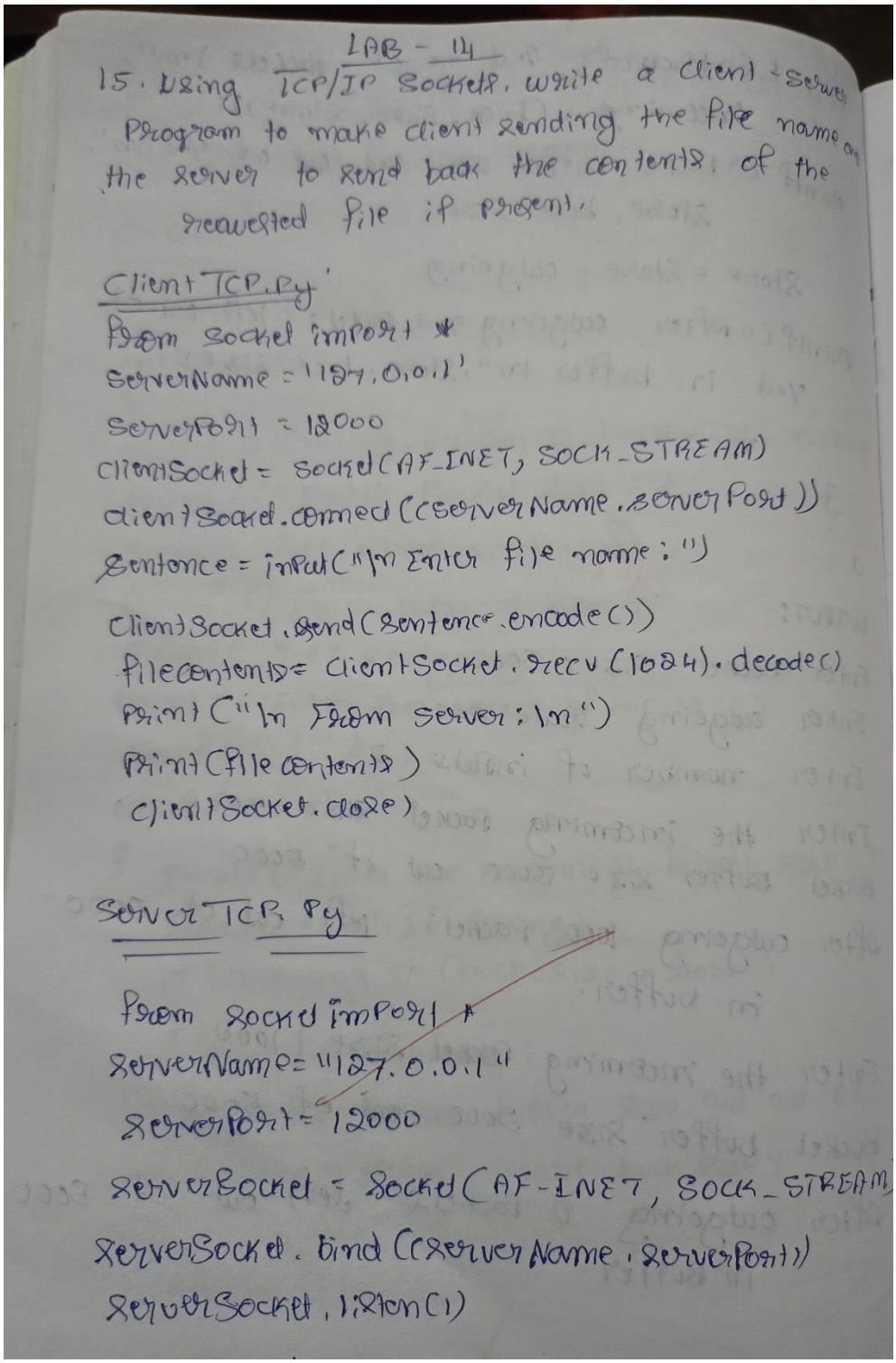
ServerTCP.py

```
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:


ServerTCP.py - D:\AUG_DEC 2021\CN\LAB\cycle 3\ServerTCP.py (3.6.7)
File Edit Format Run Options Window Help
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()
Python 3.6.7 Shell
File Edit Shell Debug Options Window Help
Python 3.6.7 (v3.6.7:6ec5cf24b7, Oct 20 2018, 13:35:33) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:\AUG_DEC 2021\CN\LAB\cycle 3\ServerTCP.py ==========
The server is ready to receive
Ln: 6 Col: 0
Type here to search
16°C
ENG
06:30
03-01-2022

OBSERVATION :



LAB - 14

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.

Client TCP.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()
```

Server TCP. 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")
d=file.read(1024)
connectionSocket.send(1.encode())
print("\n Sent contents of " + sentence)
file.close()
connectionSocket.close()
```

OUTPUT:-

* When you Run ServerTCP.Py
  The server is ready to receive.

* When you run clientTCP.Py
  Entile file Name: ServerTCP.Py
  From Server:
  (The File from ServerTCP.Py will be copied and displayed here)

* In ServerTCP.Py
  The server is ready to receive
  Sent contents of ServerTCP.Py The server is ready to receive.

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.

ClientUDP.Py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input(~~AF_INET, SOCK_DGRAM~~)
                ("\n Enter file name:")
clientSocket.sendto(bytes(sentence, "utf-8"),
    (serverName, serverPort))
fileContents, serverAddress = clientSocket.recvfr
                (2048)
print("\n Reply 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("\n Sent contents of ", end = ' ')
print(sentence)
# for i in sentence:
# print(str(i), end = "")
file.close()
```

* When you run ServerUDP.Py

The Server is ready to receive

* When you ready to receive

* When you run ClientUDP.Py

Enter file name: ServerUDP.Py

Reply from server:

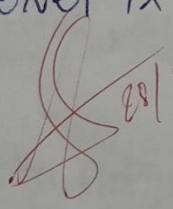
(The file from ServerUDP.Py will be copied and displayed here)

* In ServerUDP.Py

The server is ready to receive

Sent contents of ServerUDP.Py

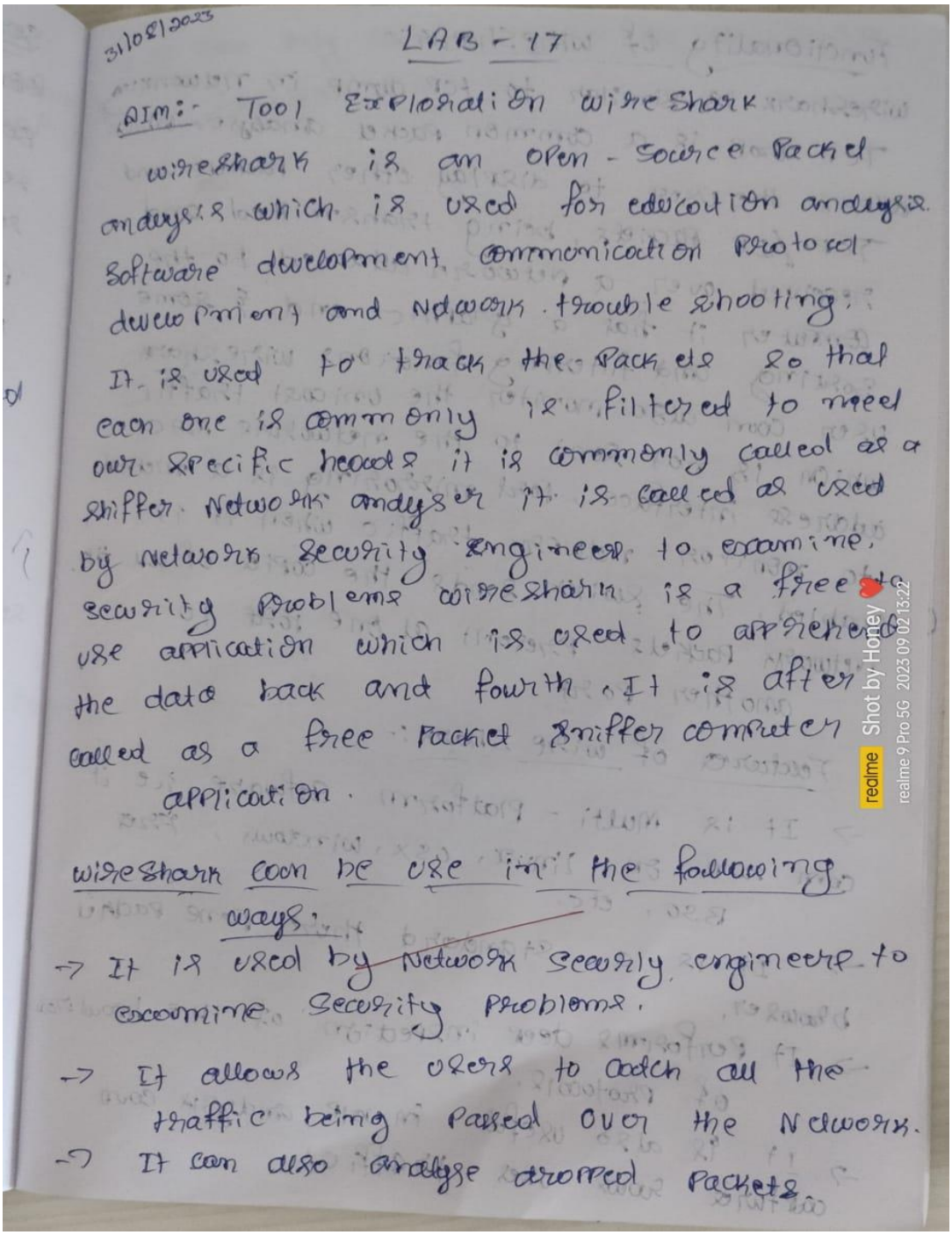
The Server is ready to receive



LAB 15

Tool Exploration -Wireshark.

OBSERVATION :



31/08/2023

LAB - 17

AIM: Tool Exploration wireshark

wireshark is an open-source packet analysis which is used for education analysis software development communication protocol development and network trouble shooting.

It is used to track the packets so that each one is commonly is filtered to meet our specific needs it is commonly called as a sniffer network analyser it is called as used by network security engineer to examine security problems wireshark is a free to use application which is used to apprehend the data back and fourth. It is after called as a free packet sniffer computer application.

wireshark can be use in the following ways:

-> It is used by network security engineers to examine security problems.

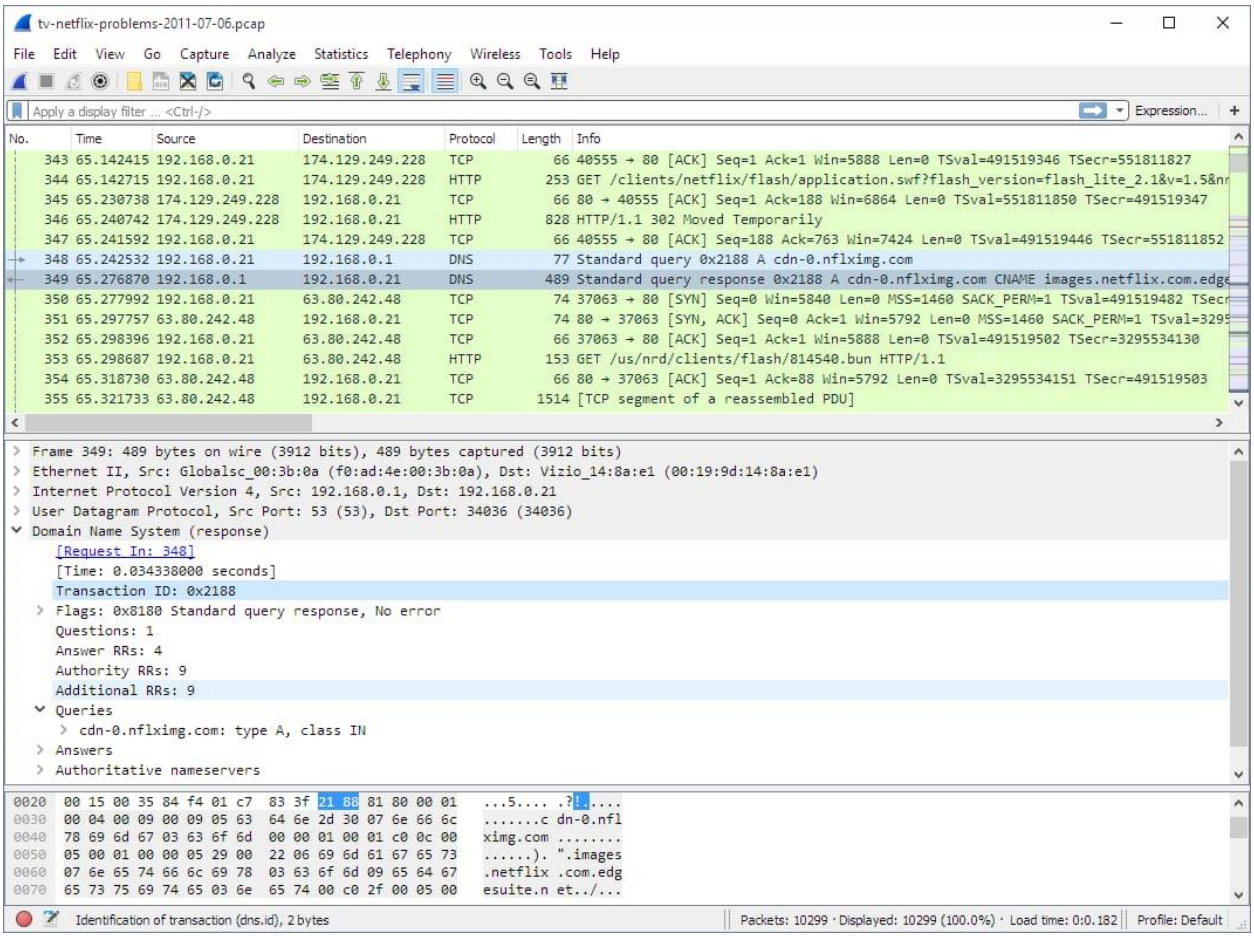
-> It allows the users to watch all the traffic being passed over the Network.

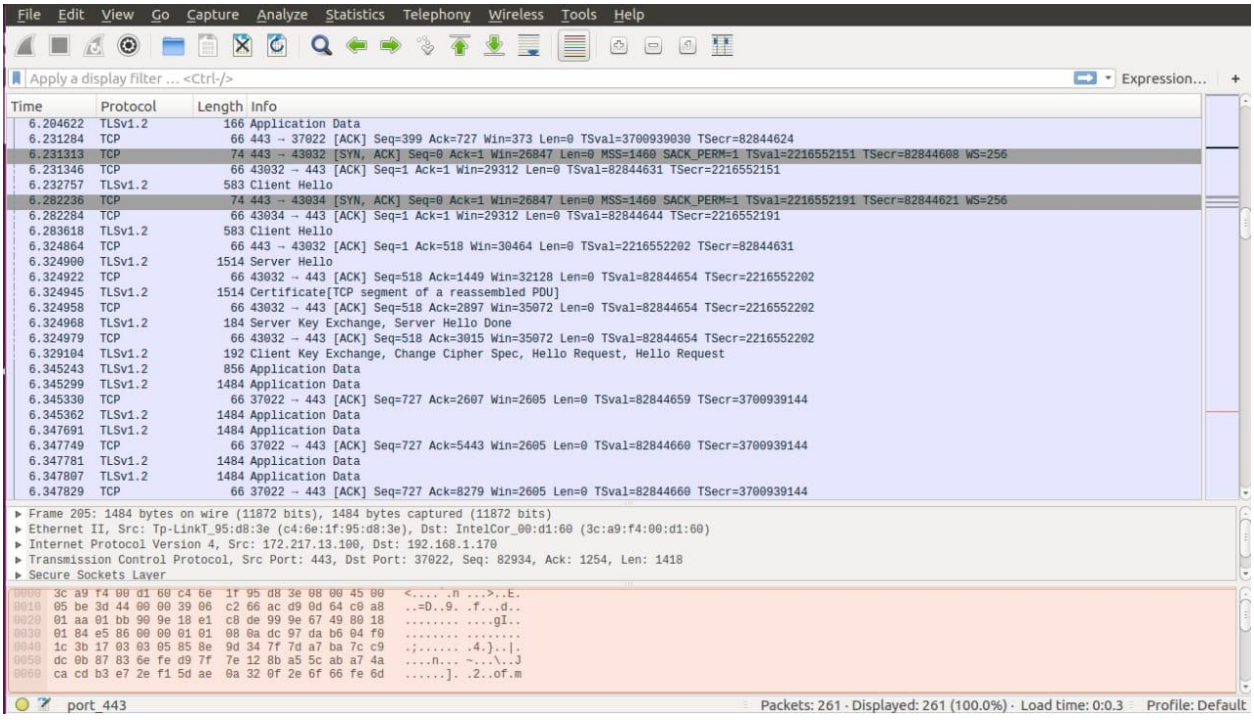
-> It can also analyse dropped packets.

- It can only capture packets on the PCAP supported networks.

NP
11/9/2023

OUTPUT :


tv-netflix-problems-2011-07-06.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
No. Time Source Destination Protocol Length Info
343 65.142415 192.168.0.21 174.129.249.228 TCP 66 40555 → 80 [ACK] Seq=1 Ack=1 Win=5888 Len=0 TSval=491519346 TSecr=551811827
344 65.142715 192.168.0.21 174.129.249.228 HTTP 253 GET /clients/netflix/flash/application.swf?flash_version=flash_lite_2.1&v=1.5&nr
345 65.230738 174.129.249.228 192.168.0.21 TCP 66 80 → 40555 [ACK] Seq=1 Ack=188 Win=6864 Len=0 TSval=551811850 TSecr=491519347
346 65.240742 174.129.249.228 192.168.0.21 HTTP 828 HTTP/1.1 302 Moved Temporarily
347 65.241592 192.168.0.21 174.129.249.228 TCP 66 40555 → 80 [ACK] Seq=188 Ack=763 Win=7424 Len=0 TSval=491519446 TSecr=551811852
348 65.242532 192.168.0.21 192.168.0.1 DNS 77 Standard query 0x2188 A cdn-0.nflximg.com
349 65.276870 192.168.0.1 192.168.0.21 DNS 489 Standard query response 0x2188 A cdn-0.nflximg.com CNAME images.netflix.com.edge
350 65.277992 192.168.0.21 63.80.242.48 TCP 74 37063 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=491519482 TSecr
351 65.297757 63.80.242.48 192.168.0.21 TCP 74 80 → 37063 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TSval=3295
352 65.298396 192.168.0.21 63.80.242.48 TCP 66 37063 → 80 [ACK] Seq=1 Ack=1 Win=5888 Len=0 TSval=491519502 TSecr=3295534130
353 65.298687 192.168.0.21 63.80.242.48 HTTP 153 GET /us/nrd/clients/flash/814540.bun HTTP/1.1
354 65.318730 63.80.242.48 192.168.0.21 TCP 66 80 → 37063 [ACK] Seq=1 Ack=88 Win=5792 Len=0 TSval=3295534151 TSecr=491519503
355 65.321733 63.80.242.48 192.168.0.21 TCP 1514 [TCP segment of a reassembled PDU]
Frame 349: 489 bytes on wire (3912 bits), 489 bytes captured (3912 bits)
Ethernet II, Src: Globalsc_00:3b:0a (f0:ad:4e:00:3b:0a), Dst: Vizio_14:8a:e1 (00:19:9d:14:8a:e1)
Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.21
User Datagram Protocol, Src Port: 53 (53), Dst Port: 34036 (34036)
Domain Name System (response)
[Request In: 348]
[Time: 0.034338000 seconds]
Transaction ID: 0x2188
Flags: 0x8180 Standard query response, No error
Questions: 1
Answer RRs: 4
Authority RRs: 9
Additional RRs: 9
Queries
cdn-0.nflximg.com: type A, class IN
Answers
Authoritative nameservers
0020 00 15 00 35 84 f4 01 c7 83 3f 21 88 81 80 00 01 ...5.... .?!.....
0030 00 04 00 09 00 09 05 63 64 6e 2d 30 07 6e 66 6c ....c dn-0.nfl
0040 78 69 6d 67 03 63 6f 6d 00 00 01 00 01 c0 0c 00 ximg.com ....
0050 05 00 01 00 00 05 29 00 22 06 69 6d 61 67 65 73 ......). ".images
0060 07 6e 65 74 66 6c 69 78 03 63 6f 6d 09 65 64 67 .netflix .com.edg
0070 65 73 75 69 74 65 03 6e 65 74 00 c0 2f 00 05 00 esuite.n et../...
Identification of transaction (dns.id), 2 bytes
Packets: 10299 · Displayed: 10299 (100.0%) · Load time: 0:0.182
Profile: Default


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
Time Protocol Length Info
6.204622 TLSv1.2 166 Application Data
6.231284 TCP 66 443 → 37022 [ACK] Seq=399 Ack=727 Win=373 Len=0 TSval=3700939030 TSecr=82844624
6.231313 TCP 74 443 → 43032 [SYN, ACK] Seq=0 Ack=1 Win=26847 Len=0 MSS=1460 SACK_PERM=1 TSval=2216552151 TSecr=82844608 WS=256
6.231346 TCP 66 43032 → 443 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=82844631 TSecr=2216552151
6.232757 TLSv1.2 583 Client Hello
6.282236 TCP 74 443 → 43034 [SYN, ACK] Seq=0 Ack=1 Win=26847 Len=0 MSS=1460 SACK_PERM=1 TSval=2216552191 TSecr=82844621 WS=256
6.282284 TCP 66 43034 → 443 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=82844644 TSecr=2216552191
6.283618 TLSv1.2 583 Client Hello
6.324864 TCP 66 443 → 43032 [ACK] Seq=1 Ack=518 Win=30464 Len=0 TSval=2216552202 TSecr=82844631
6.324900 TLSv1.2 1514 Server Hello
6.324922 TCP 66 43032 → 443 [ACK] Seq=518 Ack=1449 Win=32128 Len=0 TSval=82844654 TSecr=2216552202
6.324945 TLSv1.2 1514 Certificate[TCP segment of a reassembled PDU]
6.324958 TCP 66 43032 → 443 [ACK] Seq=518 Ack=2897 Win=35072 Len=0 TSval=82844654 TSecr=2216552202
6.324968 TLSv1.2 184 Server Key Exchange, Server Hello Done
6.324979 TCP 66 43032 → 443 [ACK] Seq=518 Ack=3015 Win=35072 Len=0 TSval=82844654 TSecr=2216552202
6.329104 TLSv1.2 192 Client Key Exchange, Change Cipher Spec, Hello Request, Hello Request
6.345243 TLSv1.2 856 Application Data
6.345299 TLSv1.2 1484 Application Data
6.345330 TCP 66 37022 → 443 [ACK] Seq=727 Ack=2607 Win=2605 Len=0 TSval=82844659 TSecr=3700939144
6.345362 TLSv1.2 1484 Application Data
6.347691 TLSv1.2 1484 Application Data
6.347749 TCP 66 37022 → 443 [ACK] Seq=727 Ack=5443 Win=2605 Len=0 TSval=82844660 TSecr=3700939144
6.347781 TLSv1.2 1484 Application Data
6.347807 TLSv1.2 1484 Application Data
6.347829 TCP 66 37022 → 443 [ACK] Seq=727 Ack=8279 Win=2605 Len=0 TSval=82844660 TSecr=3700939144
Frame 205: 1484 bytes on wire (11872 bits), 1484 bytes captured (11872 bits)
Ethernet II, Src: Tp-LinkT_95:d8:3e (c4:6e:1f:95:d8:3e), Dst: IntelCor_00:d1:60 (3c:a9:f4:00:d1:60)
Internet Protocol Version 4, Src: 172.217.13.100, Dst: 192.168.1.170
Transmission Control Protocol, Src Port: 443, Dst Port: 37022, Seq: 82934, Ack: 1254, Len: 1418
Secure Sockets Layer
0000 3c a9 f4 00 d1 60 c4 6e 1f 95 d8 3e 08 00 45 00 <.... .n ...>..E.
0010 05 be 3d 44 00 00 39 06 c2 66 ac d9 0d 64 c0 a8 ..=D..9. .f...d..
0020 01 aa 01 bb 90 9e 18 e1 c8 de 99 9e 67 49 80 18 ....gI..
0030 01 84 e5 86 00 00 01 01 08 0a dc 97 da b6 04 f0 ....
0040 1c 3b 17 03 03 05 85 8e 9d 34 7f 7d a7 ba 7c c9 .;..... .4.}..|.
0050 dc 0b 87 83 6e fe d9 7f 7e 12 8b a5 5c ab a7 4a ....n... ~...\..J
0060 ca cd b3 e7 2e f1 5d ae 0a 32 0f 2e 6f 66 fe 6d ......]. .2..of.m
port_443
Packets: 261 · Displayed: 261 (100.0%) · Load time: 0:0.3
Profile: Default