VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

COMPUTER NETWORKS

Submitted by

ANITHA U(1BM22CS402)

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 **ANITHA U(1BM22CS402),** 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
`

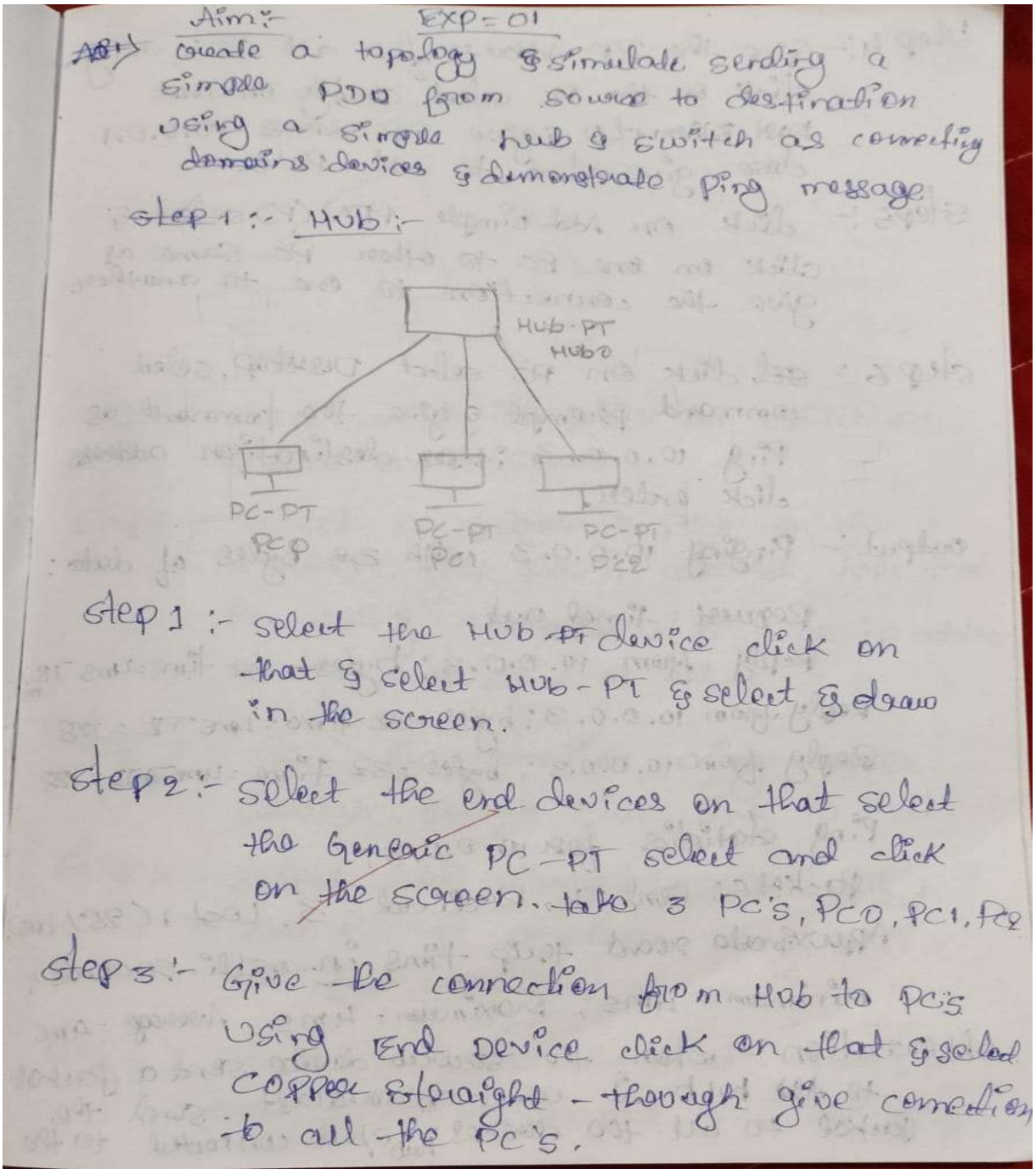
Dr. Jyothi S Nayak
Professor and
Head
Department of CSE
BMSCE, Bengaluru

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:



Aim:- EXP=01

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

Step 1 :- Hub :-

Step 1 :- Select the Hub-PT device click on that & select Hub-PT & select & draw in the screen.

Step 2 :- Select the end devices on that select the Generic PC-PT select and click on the screen. take 3 PC's, PC0, PC1, PC2

Step 3 :- Give the connection from Hub to PC's using End Device click on that & select copper straight-through give connection to all the PC's.

step 4:- Give the IP Address for all PC's, using click on the PC select config on that FastEthernet0 & give IP address 10.0.0.1 close. give same to all PC's.

step5:- click on Add simple PDU (P) select & click on one PC to other PC same as give the connection to one to another.

step 6:- set click on PC select Desktop, select command prompt & give the command as Ping 10.0.0.3 it as destination address click enter.

output:- Pinging 10.0.0.3 with 32 bytes of data:

Request timed out.

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

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

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

Ping statistics for 10.0.0.3;

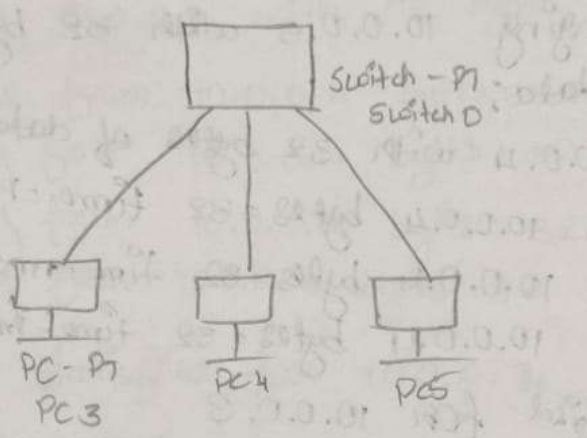
Packets: Sent = 4, Received = 3. Lost = 1 (25% Loss),

Approximate round trip tims in milli-seconds;

Minimum = 4ms, Maximum = 4ms, Average = 4ms

observation:- when the source device send a packet to the hub it will broadcast. send the packet to all the devices hub all connected to the

Switch:



Step 1 :- Select switch icon & click on switch-PT & click on the screen.

Step 2 :- Select PC & draw 3 PC's on the screen. give the ip address for that PC using Fast Ethernet 0. Give ip address as 10.0.0.4. Same as all.

Step 3 :- Give the connection using copper. Straigh through give link to all

Step 4 :- Select Add simple PDU(P) select one PC to another PC give link to that message.

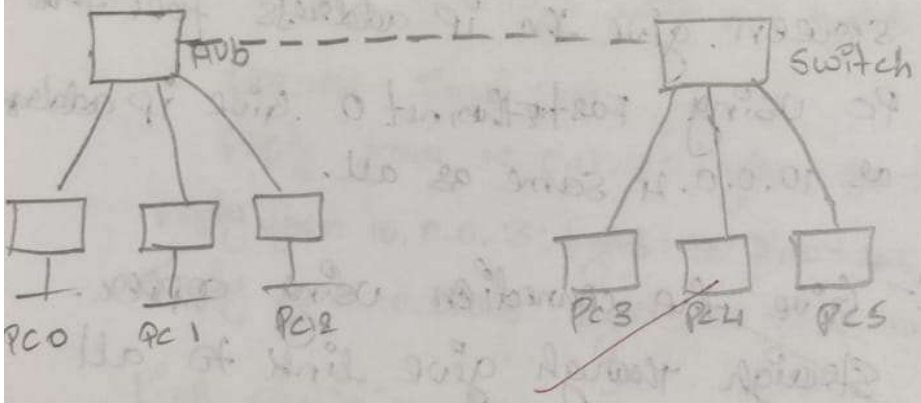
Step 5 :- select Pc click on that click on Desk-top -> Command give the ping address.

Step 6 :- PC> Ping 10.0.0.5

Output :- Pinging 10.0.0.5 with 32 bytes of data;

Pinging 10.0.0.4 with 32 bytes of data:
Reply from 10.0.0.4 bytes = 32 time = 1 ms TTL = 128
Reply from 10.0.0.4 bytes = 32 time = 1 ms TTL = 128
Reply from 10.0.0.4 bytes = 32 time = 1 ms TTL = 128
Ping satisfied for 10.0.0.3
Packets : Send = 4, Recived = 4 lost = 0 (0% loss)
Approximat round toup times in milli-seconds.

Observation :- when the source device sent a packet to all hub it will broadcast or send the packet to all the devices which all connected to the hub.



Step 1 :- Previously drawn hub topology & switch topology are connected through copper cross over In hub port 3 is used, in switch fast ethernet is used.

Step 2 :- Add simple PDC from PC0 to PC3.

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 = 1ms 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

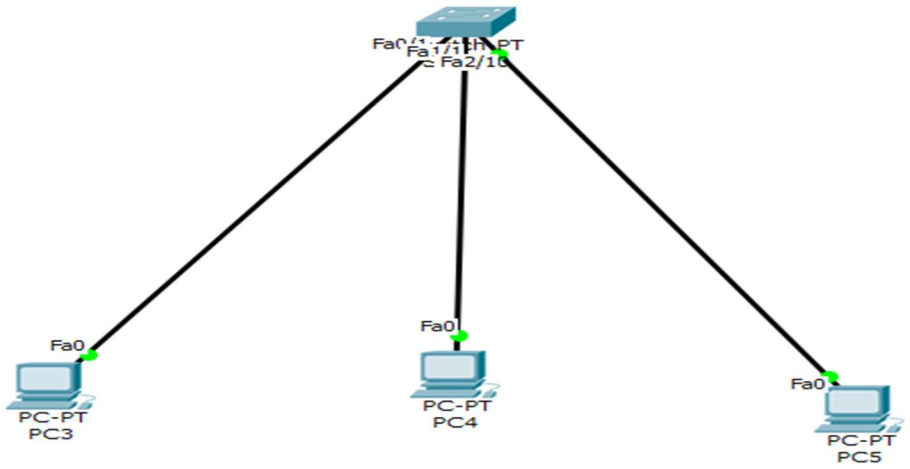
Ping statistics for 10.0.0.4

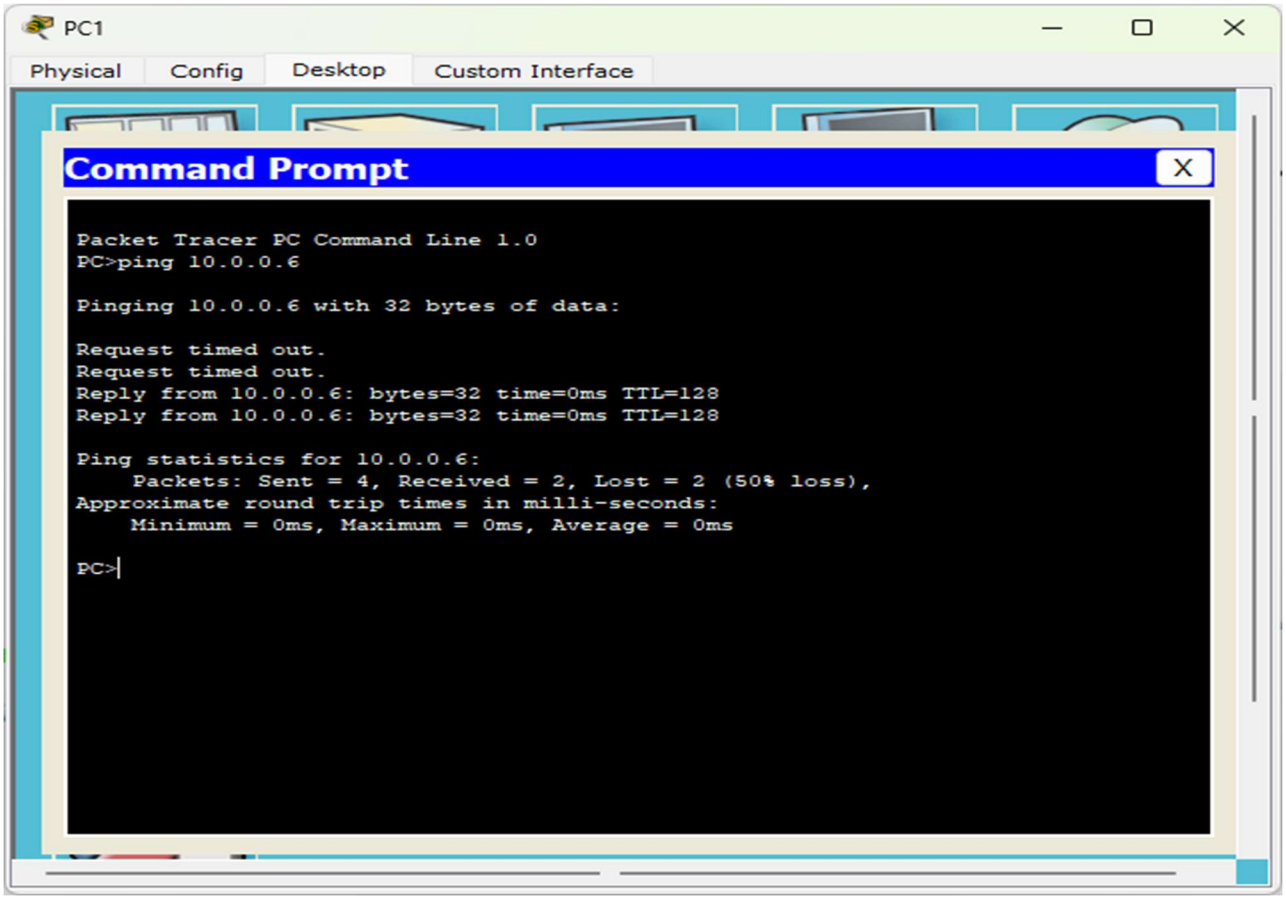
Packets = sent = 4, Received = 4 lost = 0 (0% loss)
Approximate round trip times in milli-second.
minimum = 4ms maximum = 4ms Average = 4ms

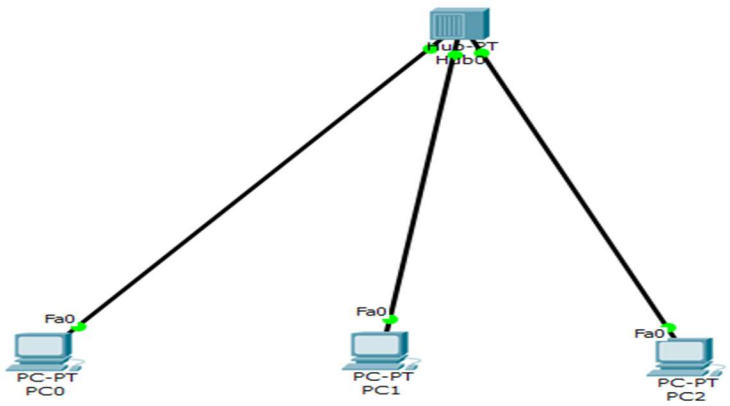
Observation:-

In simulation mode PC0 sends packet to hub. hub send it to PC1 PC2 & select broad casts it to PC3, PC4 & PC5
PC1, PC2, PC4 & PC5 discards them.
PC3 accepts & sends acknowledgment to hub through switch.
Hub is broad casts it to all 3PCs
only PC0 accepts it & others discard.
In second round PC0 sends packet to hub it broad casted to PC1, PC2 switch.
Now switch broad casts it only to PC3.
thus. switch is smart device.

OUTPUT :


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>

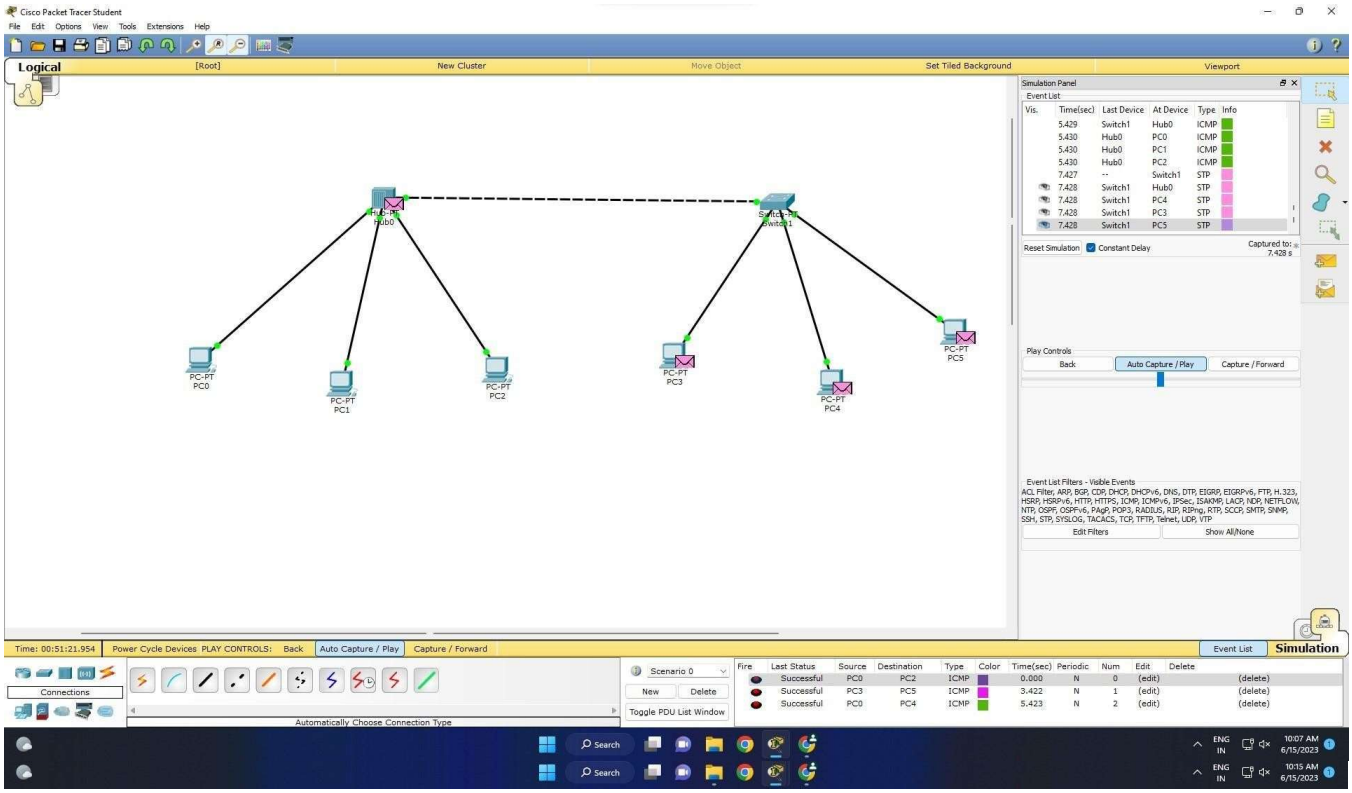


PC3

Physical | Config | Desktop | Custom Interface

Command Prompt

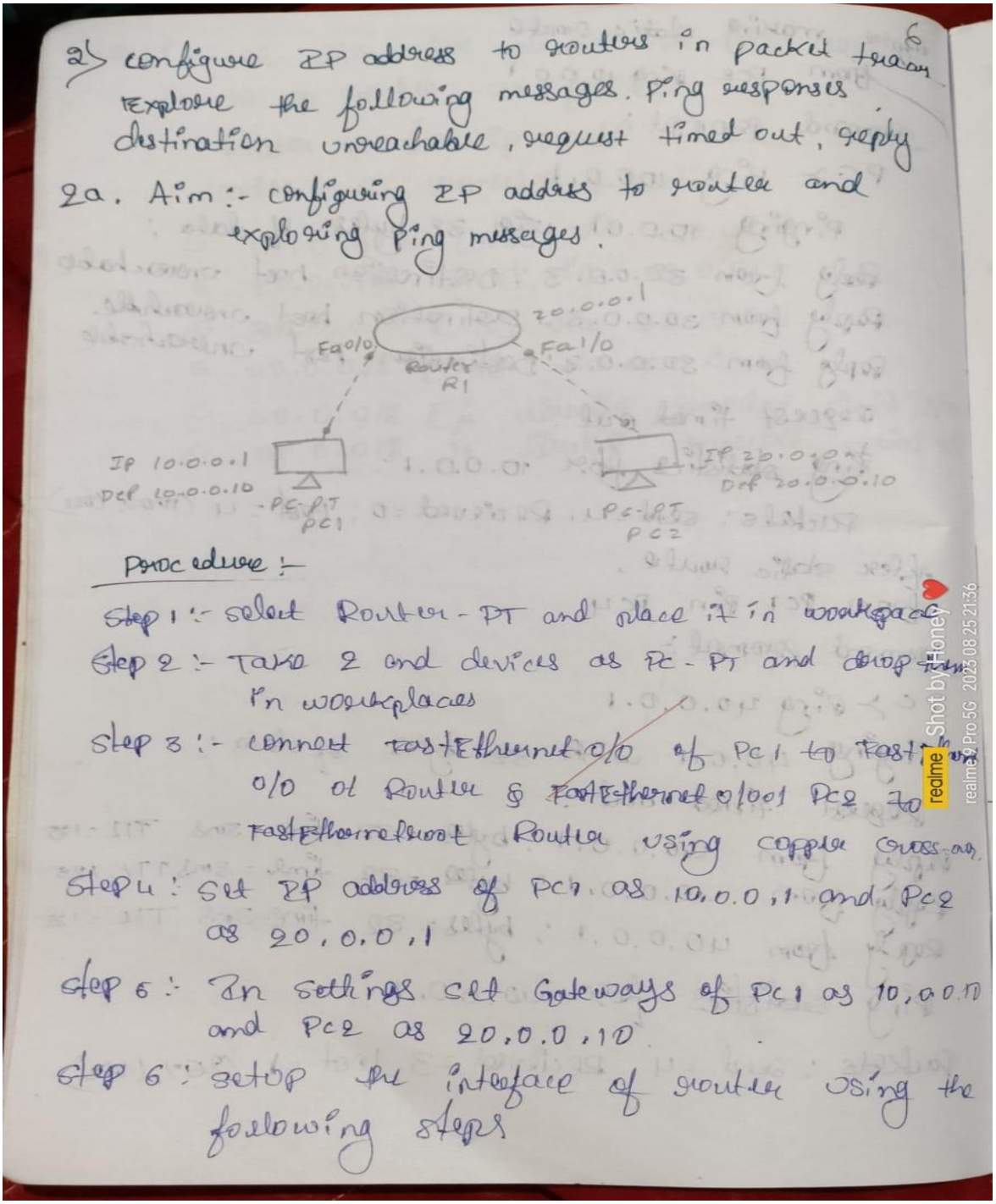
```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.6

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:



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)# interface fastEthernet 1/0
Router (config-if)# ip address 20.0.0.10 255.00.0
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, FastEthet 1/0

step 7: observation

Green lights appear on wires when 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 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=0ms 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 = 1ms, Average

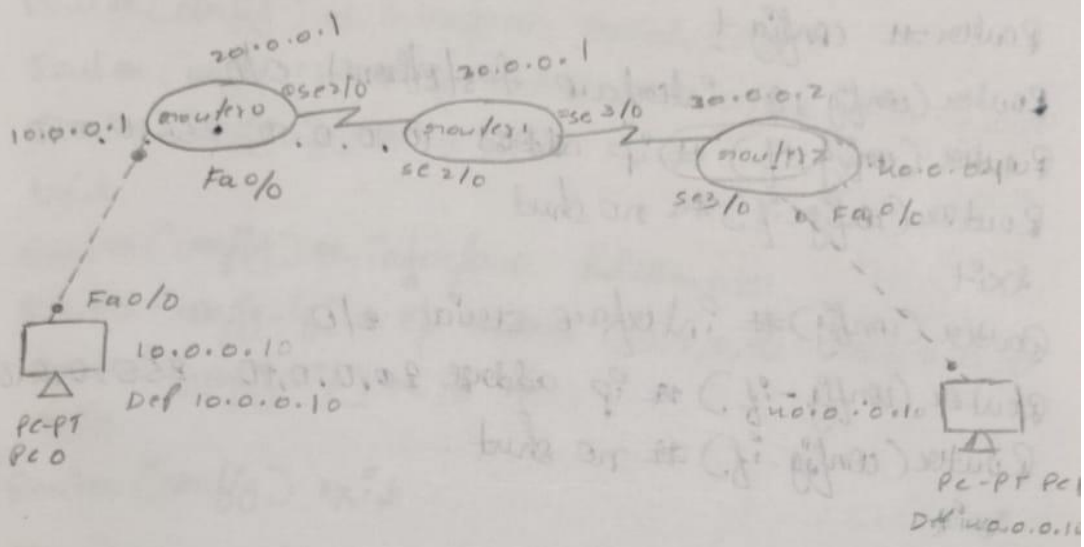
Observation

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

from next ping, there are no losses.

26. Aim :- configure using 3 routers and 2 PCs ?

Topology :-



Procedure :-

Step 1 :- The network is started by selecting end devices PC0 & PC1 i.e. generic PCs and placing them in work space.

Step 2 :- Select 3 Router-PT & place them as Router0, Router1 and Router2 in workspace

Step 3 :- PC0 & PC1 are connected to router 0 & Router 2 respectively using copper cross over.

Step 4 :- connect Router 0 to Router 1, Router 1 to Router 2 using.

Step 5 :- Set up IP address of PC0 to 10.0.0.1 PC1 to 40.0.0.1 setup gateway of PC0 as 10.0.0.20 and PC1 as 40.0.0.10

configure the router by opening CLI

In Router 0

Router > enable

Router # config t

Router (config) # interface fastEthernet 0/0

Router (config-if) # ip address 10.0.0.10 255.0.0.0

Router (config-if) # no shut

exit

Router (config) # interface serial 2/0

Router (config-if) # ip address 20.0.0.10 255.0.0.0

Router (config-if) # no shut

exit

exit

In Router 1

Router > enable

Router # config t

Router (config) # interface serial 2/0

Router (config-if) # ip address 20.0.0.20 255.0.0.0

Router (config-if) # no shut

exit

Router (config) # interface serial 3/0

Router (config-if) # ip address 30.0.0.20 255.0.0.0

Router (config-if) # no shut

exit

Router (config) # exit

In Router 2

Router > enable
Router # config t
Router (config) # interface serial 2/0
Router (config-if) # ip address 30.0.0.20 255.0.0.0
Router (config-if) # no shut
exit
Router (config) # interface fastEthernet 0/0
Router (config-if) # ip address 40.0.0.10 255.0.0.0
Router (config-if) # no shut
exit
Router (config) exit

IP Route table :-

Router 0 :

Router # show ip route
C 20.0.0.0/8 is directly connected, FastEthernet 0/0
C 20.0.0.0/8 is directly connected, serial 2/2

Router 1 :

Router # show ip route
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 2

Router # show ip route
C 30.0.0.0/8 is directly connected, serial 2/0
C 40.0.0.0/8 is directly connected, FastEthernet 0/0

Ping output in PC0

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.

Ping statistics for 40.0.0.1:

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

Observations:-

Green lights appears on the wires when no shut is written.

Now configure the router which does not have data of other network add the network in CLI. In all 3 routers CLI write config t. then ip route

Route 0:

IP route 30.0.0.0 255.0.0.0 20.0.0.30
IP route 40.0.0.0 255.0.0.0 20.0.0.20

Router 1:

ip route 10.0.0.0 255.0.0.0 20.0.0.10
ip route 40.0.0.0 255.0.0.0 30.0.0.20

Router 2:

ip route 10.0.0.0 255.0.0.0 30.0.0.10

ip route 20.0.0.0 255.0.0.0 30.0.0.10

new ip route table

exit

Router 0

C 10.0.0.0/8 is directly connected, FastEthernet 0/0

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

S 30.0.0.0/8 [1/0] via 20.0.0.20

S 40.0.0.0/8 [1/0] via 20.0.0.20

Router 1

S 10.0.0.0/8 [1/0] via 20.0.0.10

C 20.0.0.0/8 is directly connected serial 2/0

C 30.0.0.0/8 is directly connected serial 3/0

S 40.0.0.0/8 [1/0] via 30.0.0.20

Router 2

S 10.0.0.0/8 [1/0] via 30.0.0.10

S 20.0.0.0/8 [1/0] via 30.0.0.10

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

C 40.0.0.0/8 is directly connected, FastEthernet 0/0

Ping messages

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 = 2ms TTL = 125

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

Ping statistics for 40.0.0.1

Packets sent = 4, Recived = 3, lost = 1 (25% loss)

Approximate round trip times in milli-seconds

minimum = 2ms, maximum = 2ms, Average = 2ms

Observation:

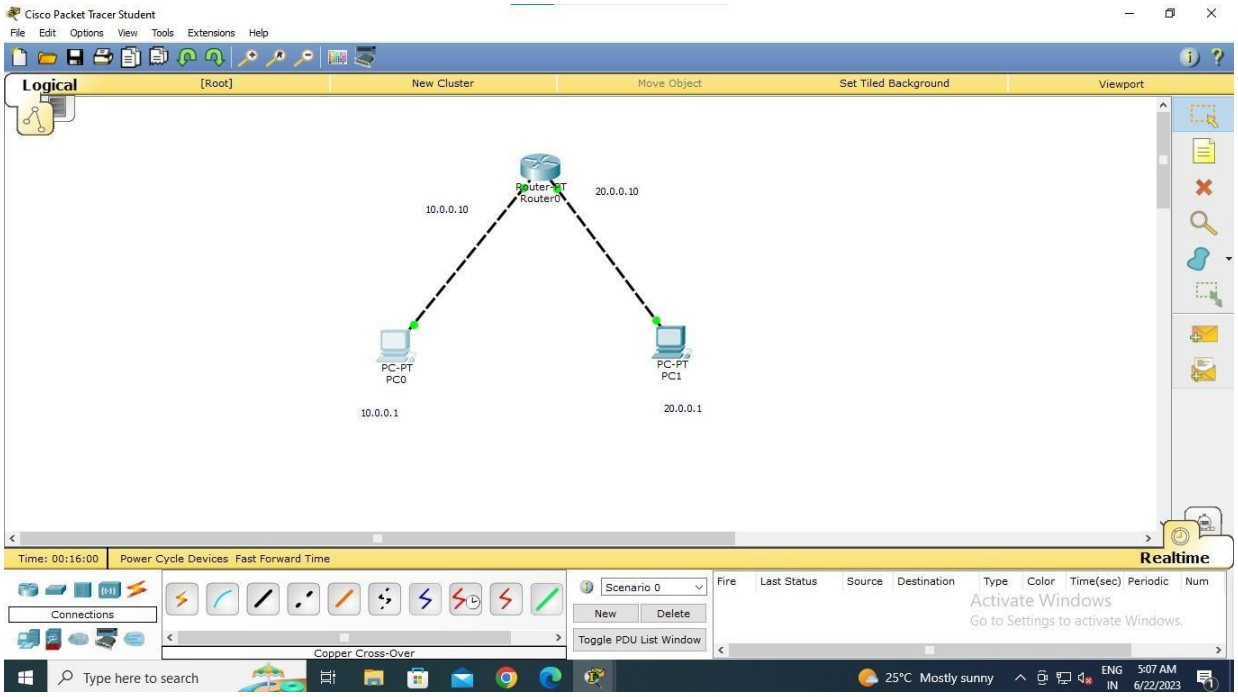
In first Ping destination host was unreachable as Router 0 has no knowledge about the network 30.0.0.0 and 40.0.0.0 & the packets got stuck or lost.

After this ip route is explicitly

Now on pinging there is 25% loss in first time, the following one's has no loss

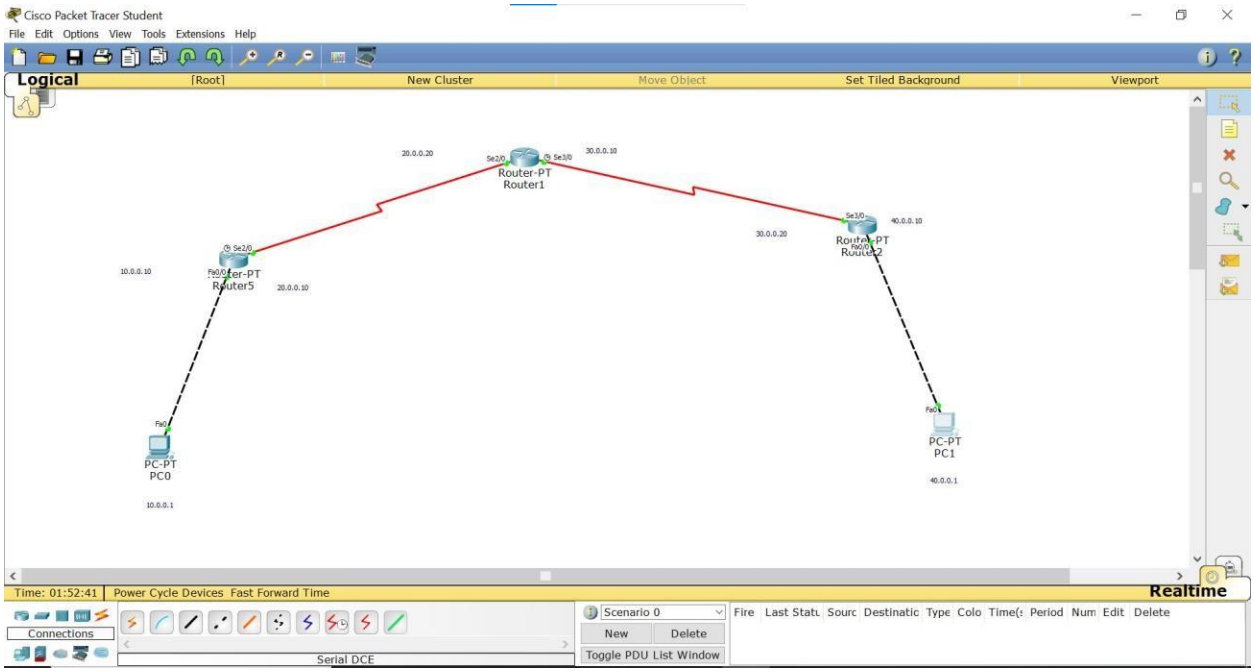
Output :

TOPOLOGY:

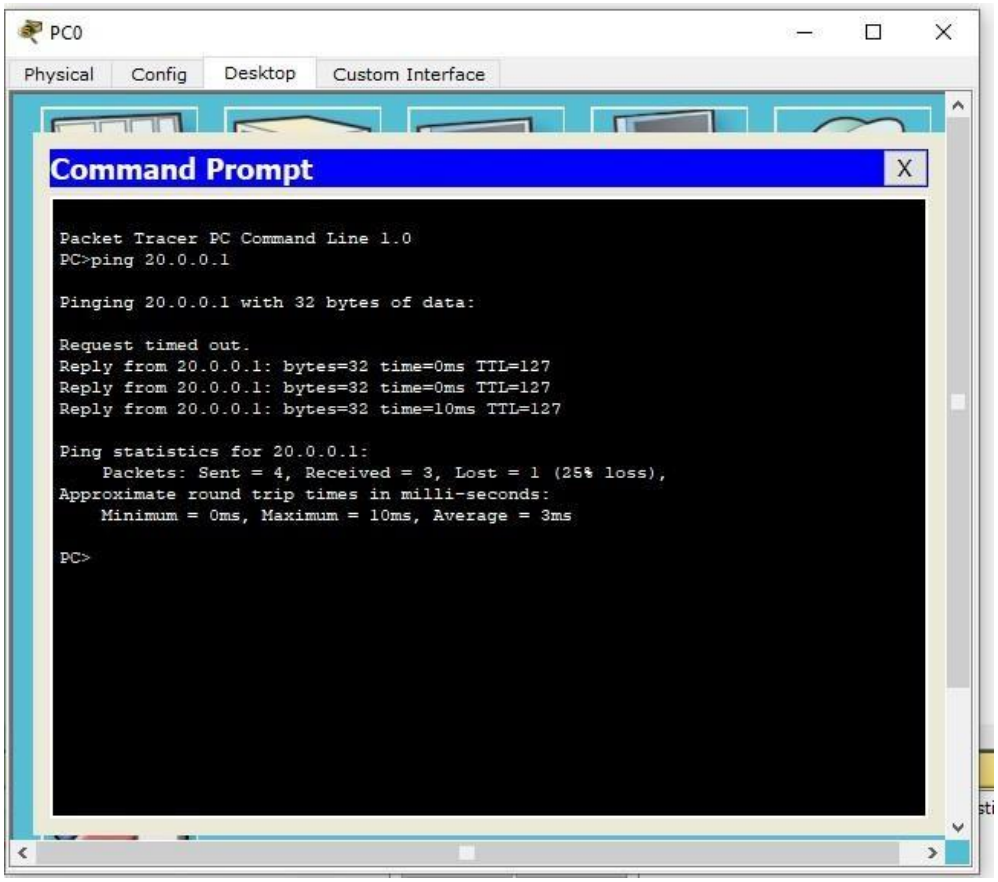


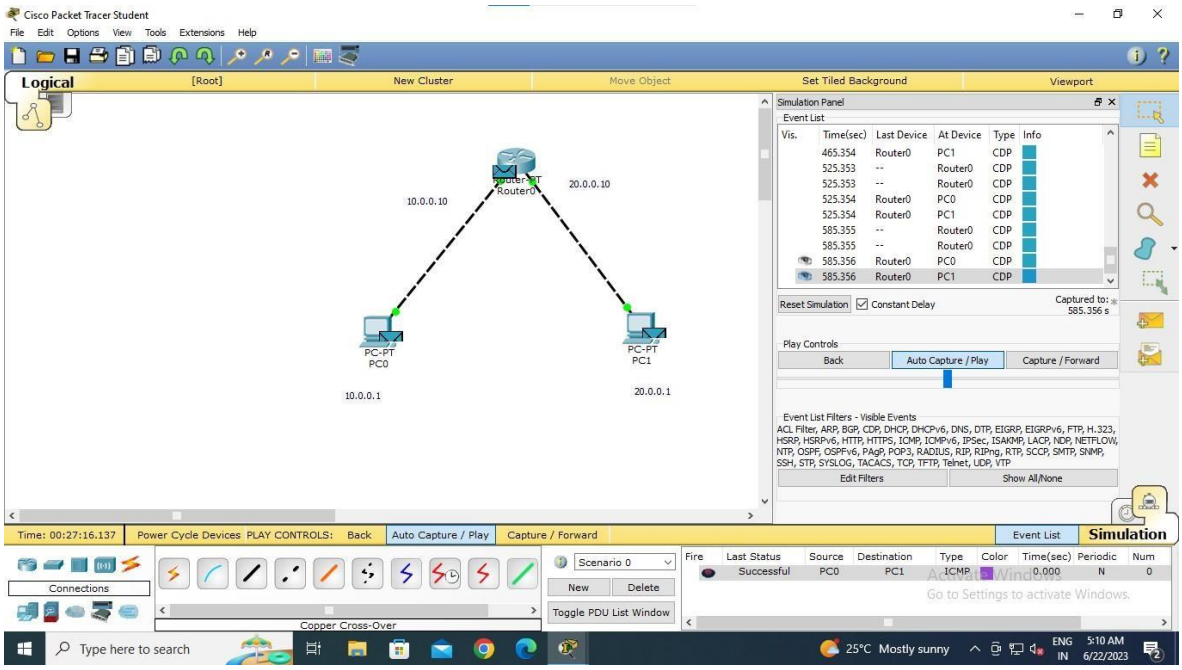
PROGRAM 2.1

PROGRAM 2.2

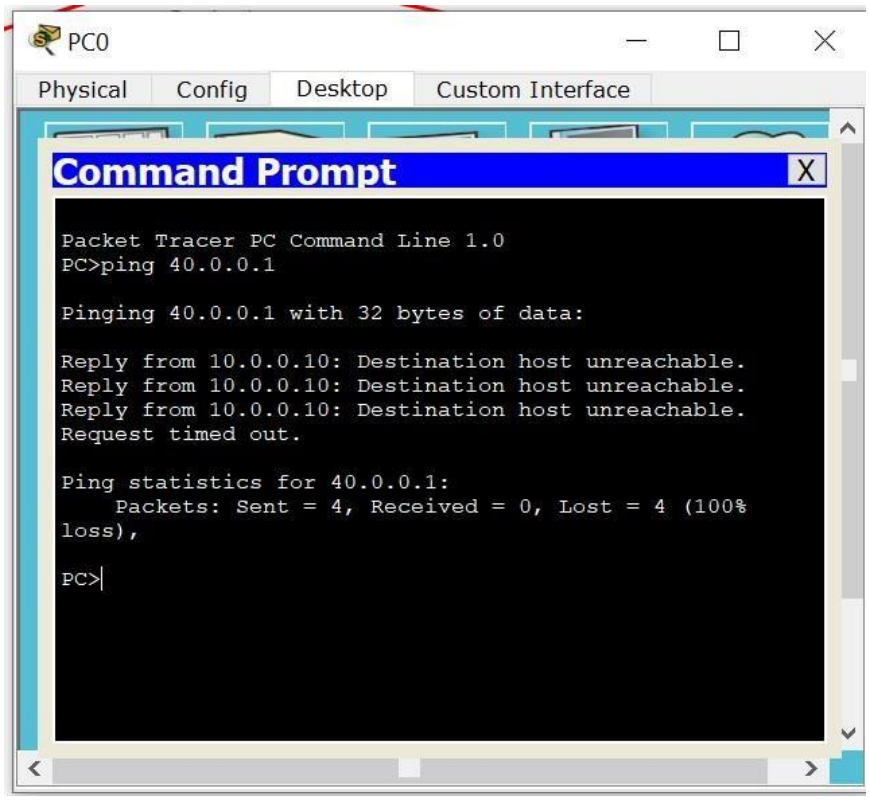


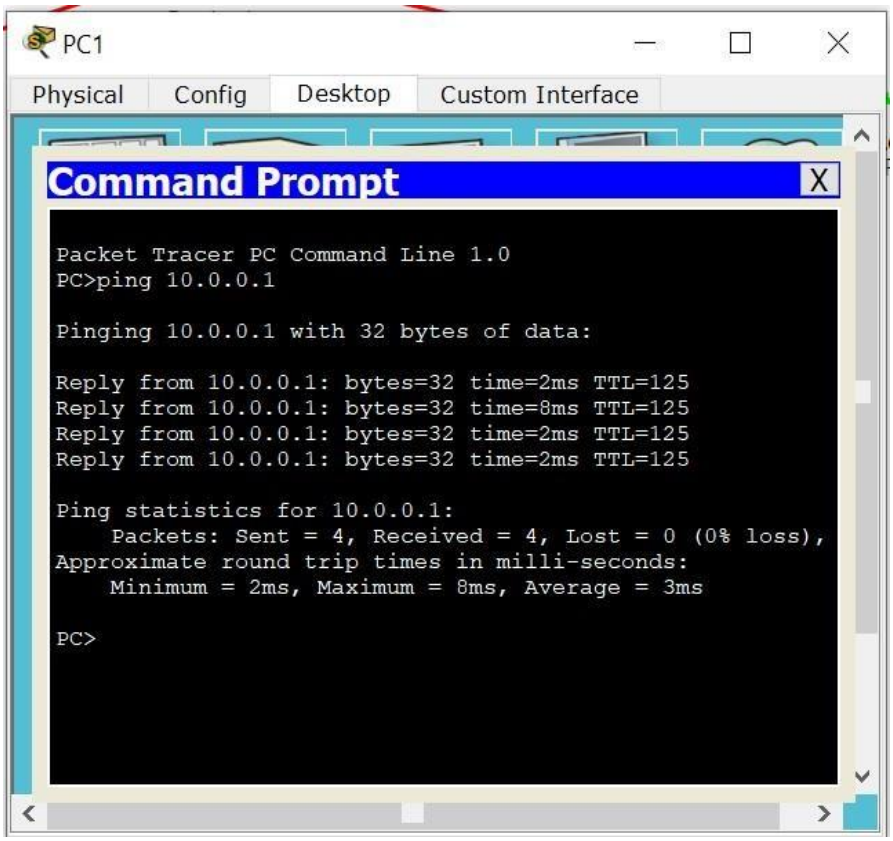
PROGRAM 2.1

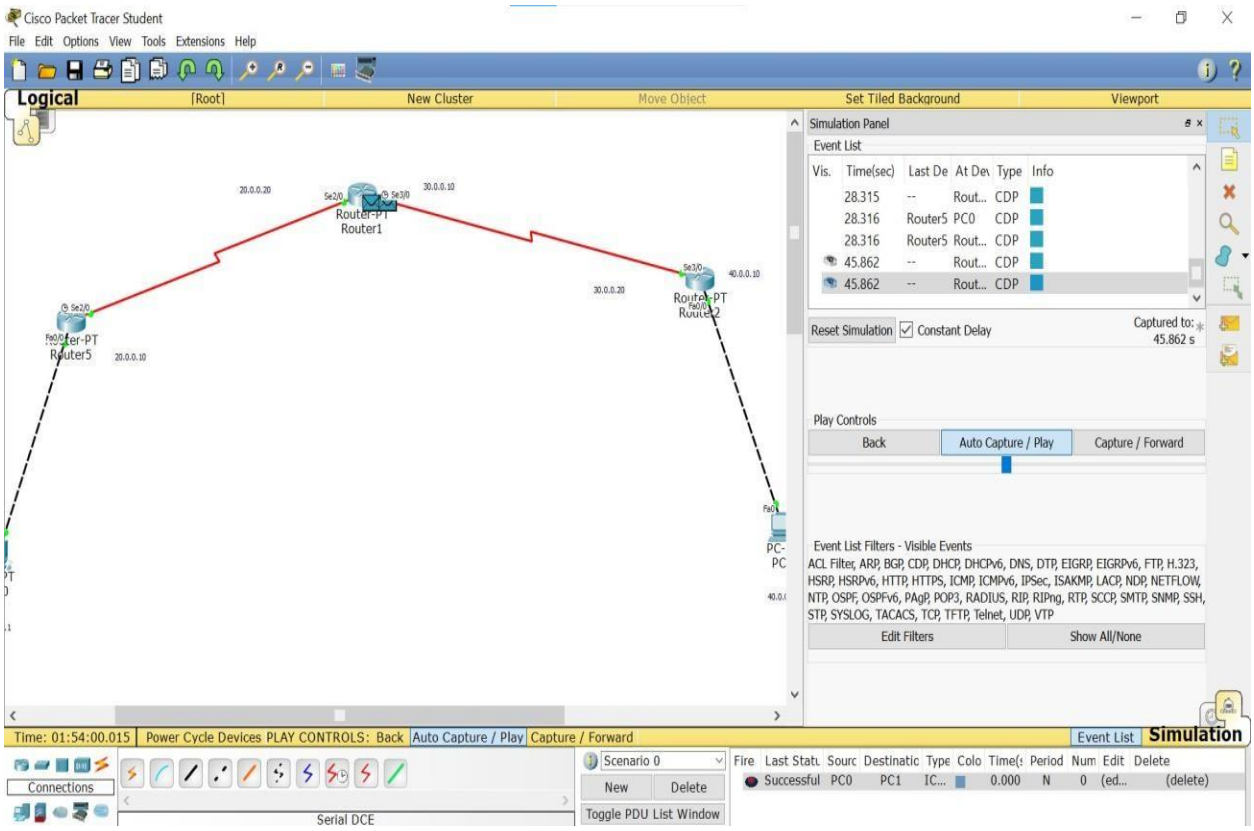

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Request timed out.
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=10ms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

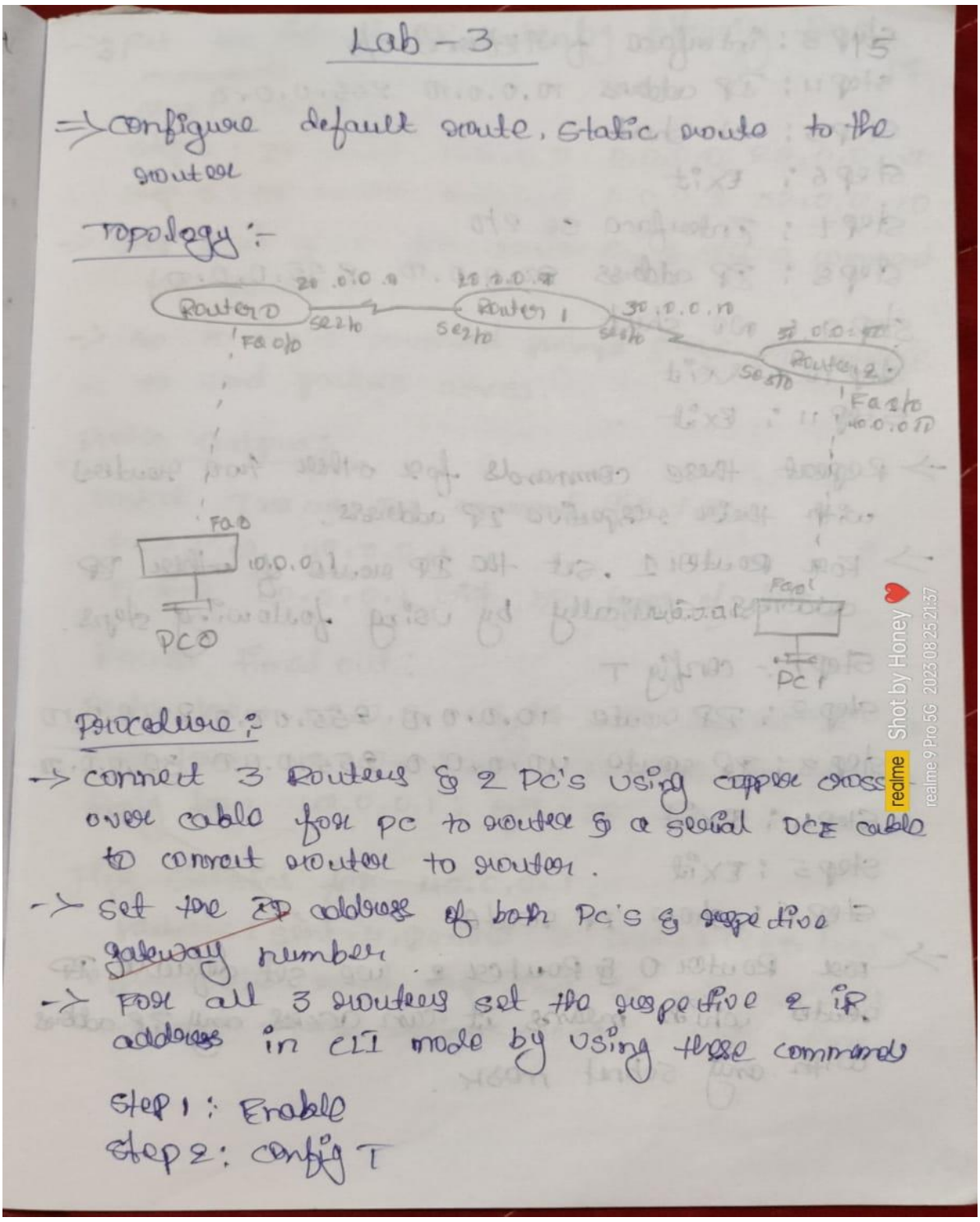

PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=2ms TTL=125
Reply from 10.0.0.1: bytes=32 time=8ms TTL=125
Reply from 10.0.0.1: bytes=32 time=2ms TTL=125
Reply from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



step 3 : interface fastEthernet 0/0
step 4 : IP address 10.0.0.10 255.0.0.0
step 5 : No shut
step 6 : Exit
step 7 : Interface se 2/0
step 8 : IP address 20.0.0.10 255.0.0.0
step 9 : no shut
step 10 : Exit
step 11 : Exit

-> Repeat these commands for other two routers with their respective IP address.

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

Step 1 :- config T
step 2 : IP route 10.0.0.0 255.0.0.0 20.0.0.0
step 3 : IP route 40.0.0.0 255.0.0.0 30.0.0.0
Step 4 : Exit.
step 5 : Exit
step 6 : show IP route

-> For Router 0 & Router 2 we set default IP route which means it can access any IP addres with any subnet mask.

-> Set the default IP route by following these commands

Step 1 : config T

Step 2 : IP route 0.0.0.0 0.0.0.0 20.0.0.20

Step 3 : IP route 0.0.0.0 0.0.0.0 30.0.0.10

-> step 2 is given for Router 0 & step 3 command for Router 1

-> Go to PC's command prompt & type ping message to send packets across.

PING Output :-

Packet Tracer PC command Line 1.0

PC> Ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data :

Request timed out :

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

Reply from 40.0.0.1 : bytes = 32 time = 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, Recived = 3 Lost = 1 (25% Loss)

Approximate round trip times in milli-seconds

minimum = 2ms, maximum = 16ms, Average = 6ms

Observation:

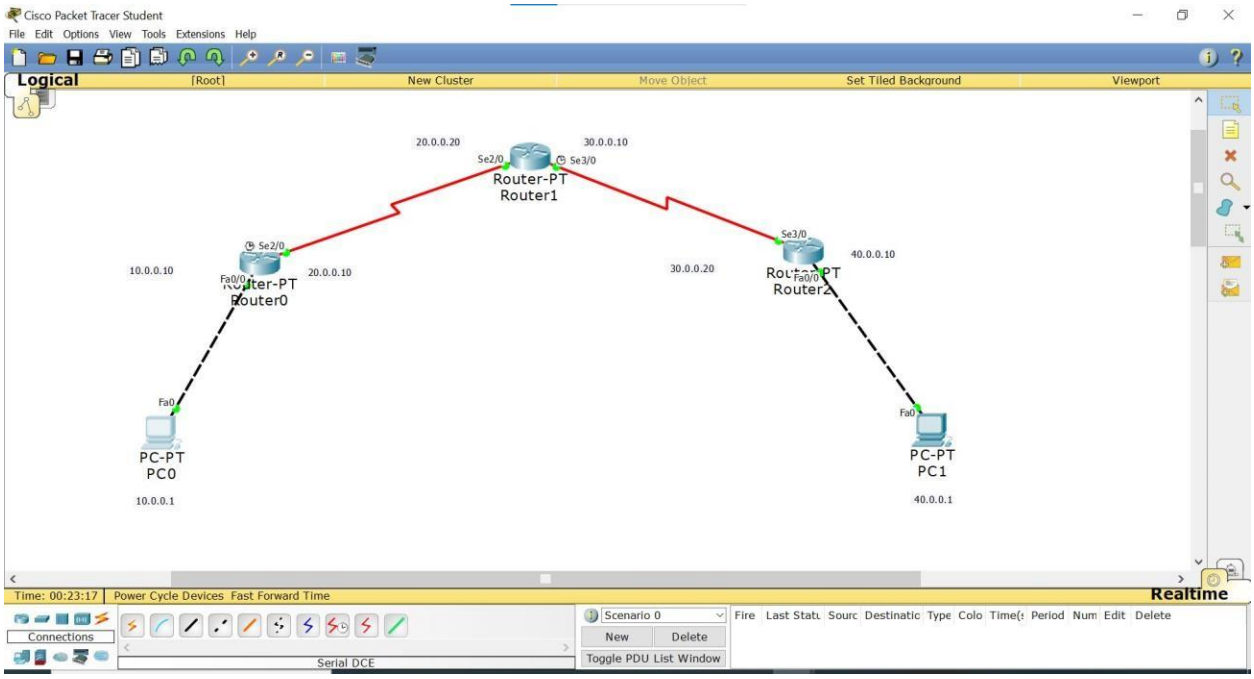
-> A default route is the route which takes effect when no other route is available for an IP address destination.

-> If a packet is received, the devices first checks the IP destination address, if the destination address is not local the device checks its routing table

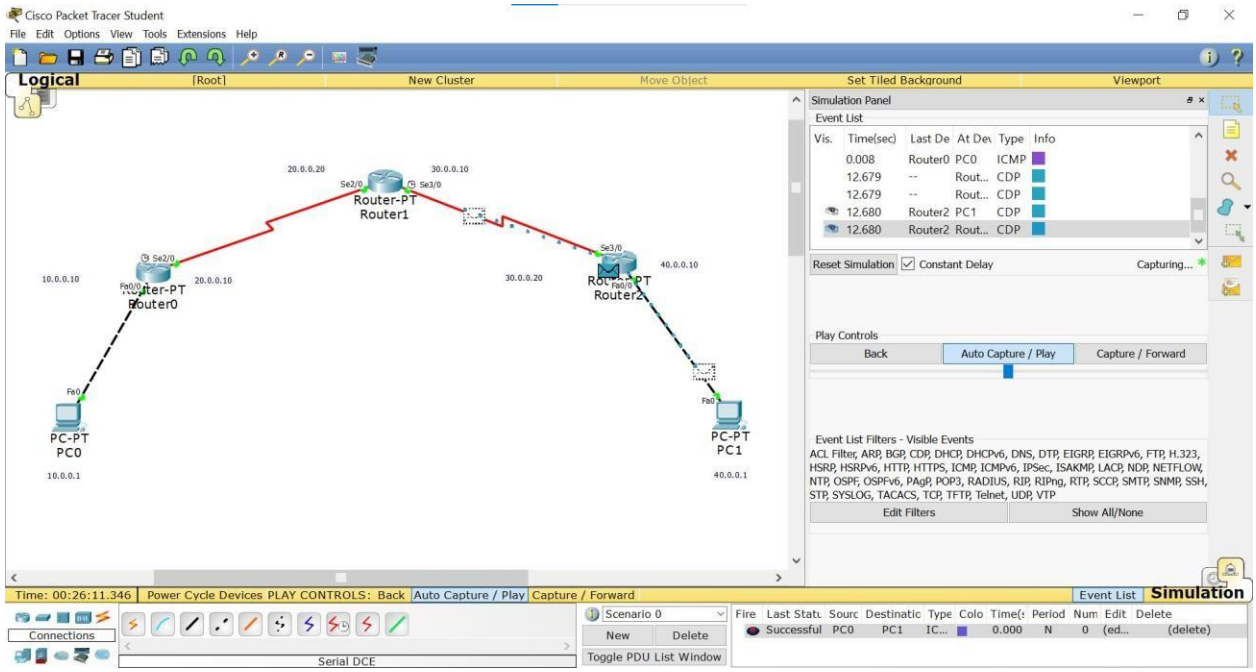
-> If the remote destination subnet is not listed then the packet is forwarded to the next hop toward the destination using the default route

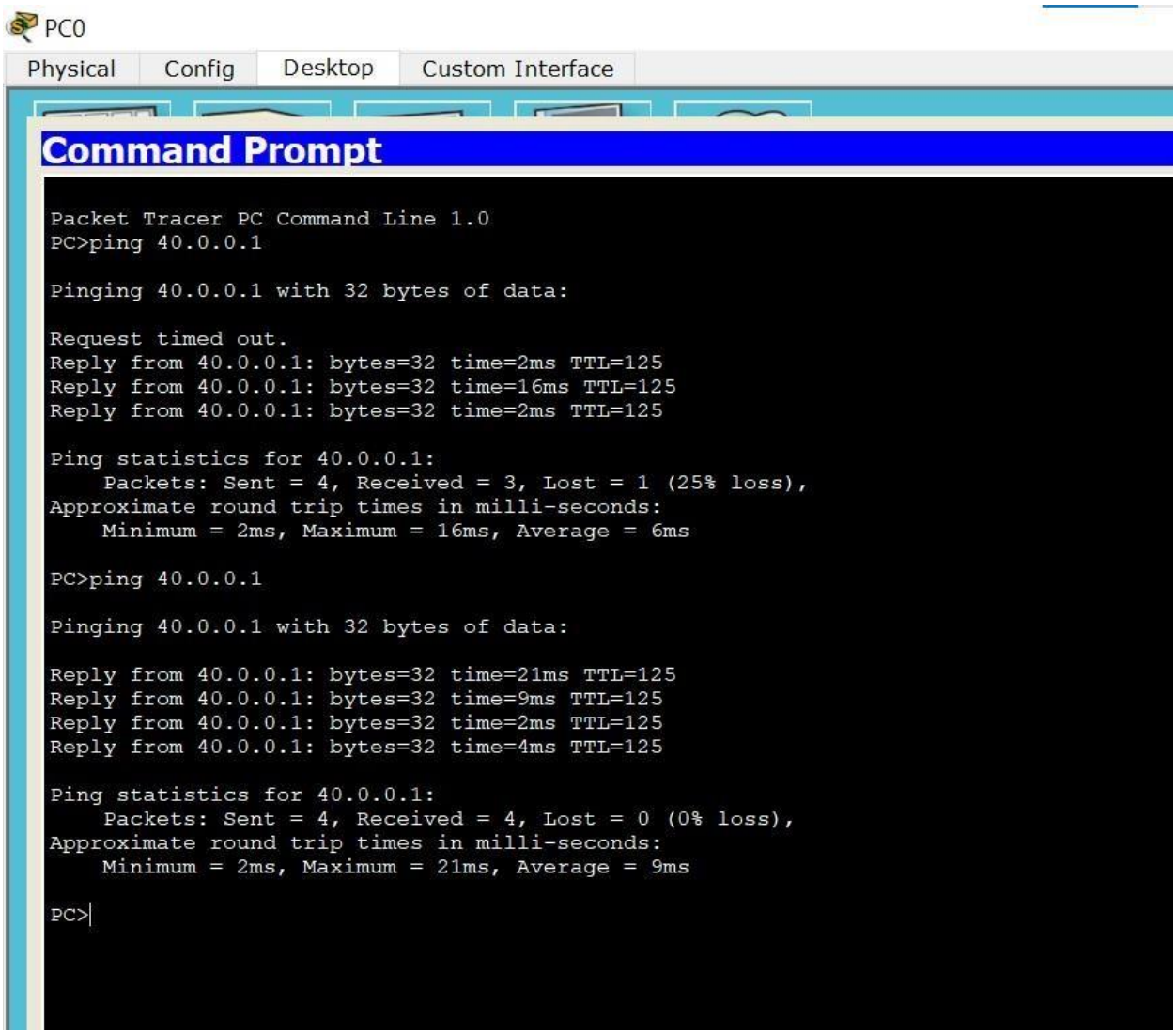
-> the process repeats until the packet is delivered.

TOPOLOGY:



OUTPUT:

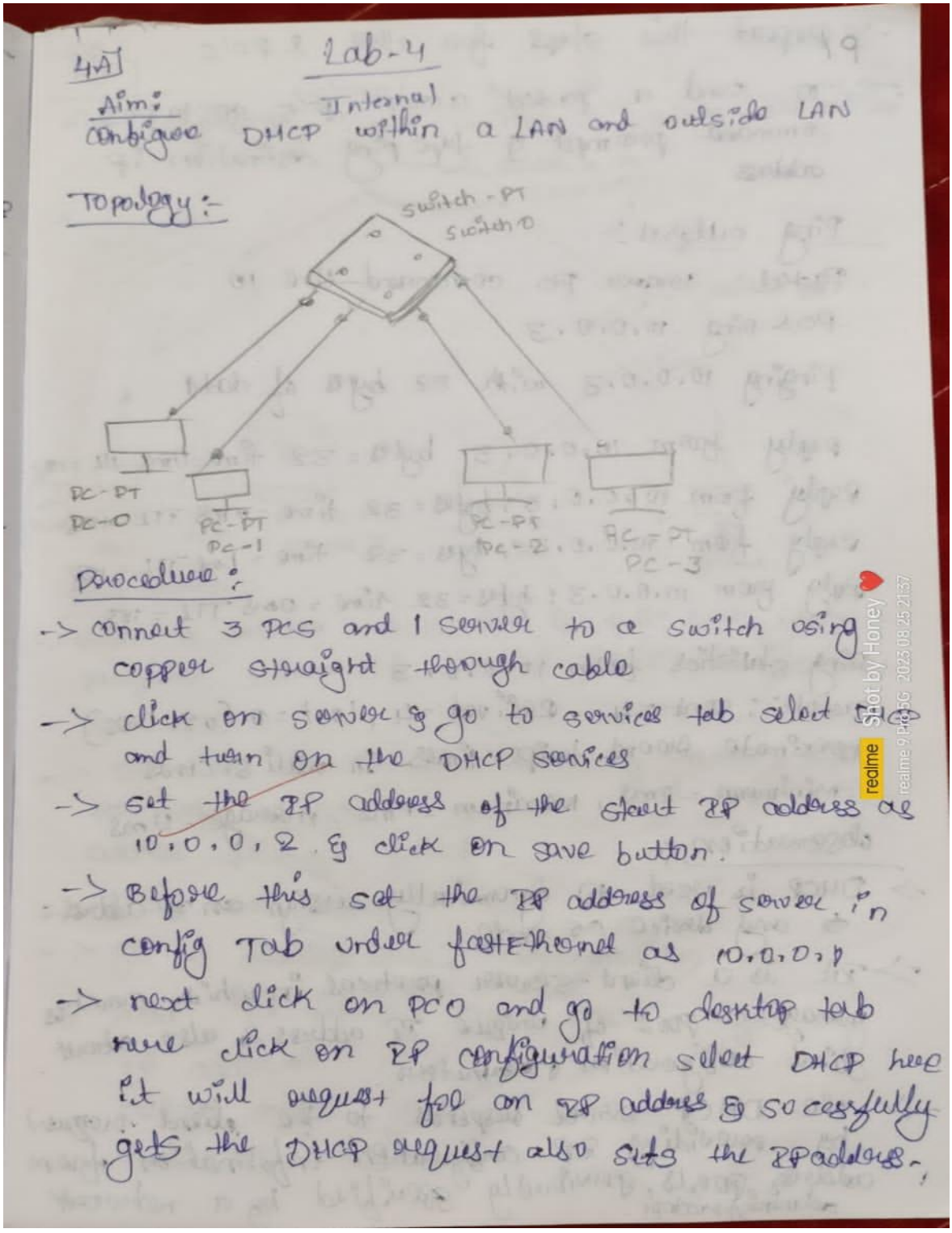



PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Request timed out.
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125
Reply from 40.0.0.1: bytes=32 time=16ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms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



→ Repeat this steps for other 2 PC's

→ To send a packet across PC's go to Pc's command prompt & type ping destination ip address

Ping output :-

Packet tracer PC command line 10

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

observation :-

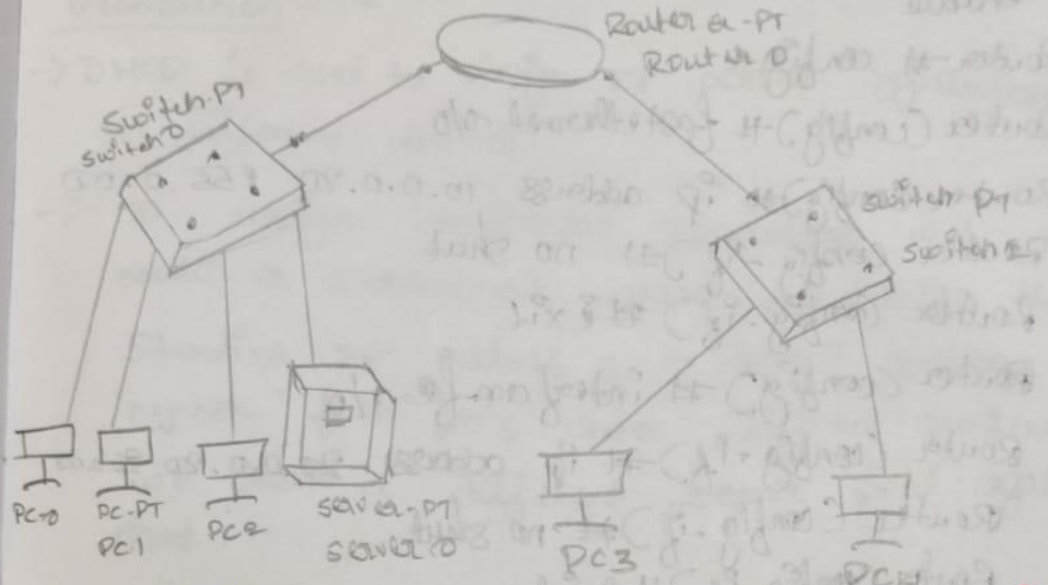
→ DHCP is used to dynamically assign an IP address to any device or node

→ It is a client-server protocol in which servers manage a pool of unique IP address & also about client configuration parameters.

→ the DHCP server responds to the client request by providing IP configuration information from address pools, previously specified by a network administrator.

4[B]

Aim:- configure external DHCP within a LAN & outside LAN



Procedure:-

-> Add a router, a switch & 2 PC's to LAN) program network & connect the router to both switches.

-> set the server IP address of server & with the help of server set the first 3 PC's IP address through DHCP.

-> click on server

-> go to desktop -> IP configuration

-> Add IP address, subnet mask & gateway

IP address 10.0.0.1

subnet mask 255.0.0.0

Gateway 10.0.0.20

step 3:- configure the router

-> click on router go to CLI

enable

Router# config t

Router(config)# fastethernet 0/0

Router(config)# ip address 10.0.0.20 255.0.0.0

Router(config-if)# no shut

Router(config-if)# exit

Router(config)# interface 1/0

Router(config-if)# ip address 20.0.0.20 25.0.0.0

Router(config-if)# no shut

Router(config-if)# exit

exit

Routing enable

Router -> show ip route

C: 10.0.0.0/1 is directly connected fe 0

step 4:- Go to server [DHCP server config]

-> select services then go to DHCP

-> set service on

-> set start IP address from (ex-20.0.0.0) th save

-> step 5:- then configure the PC's

Select a pc then desktop - go to IP config select DHCP

-> Repeat the same procedure for all other PC's

Observation:-

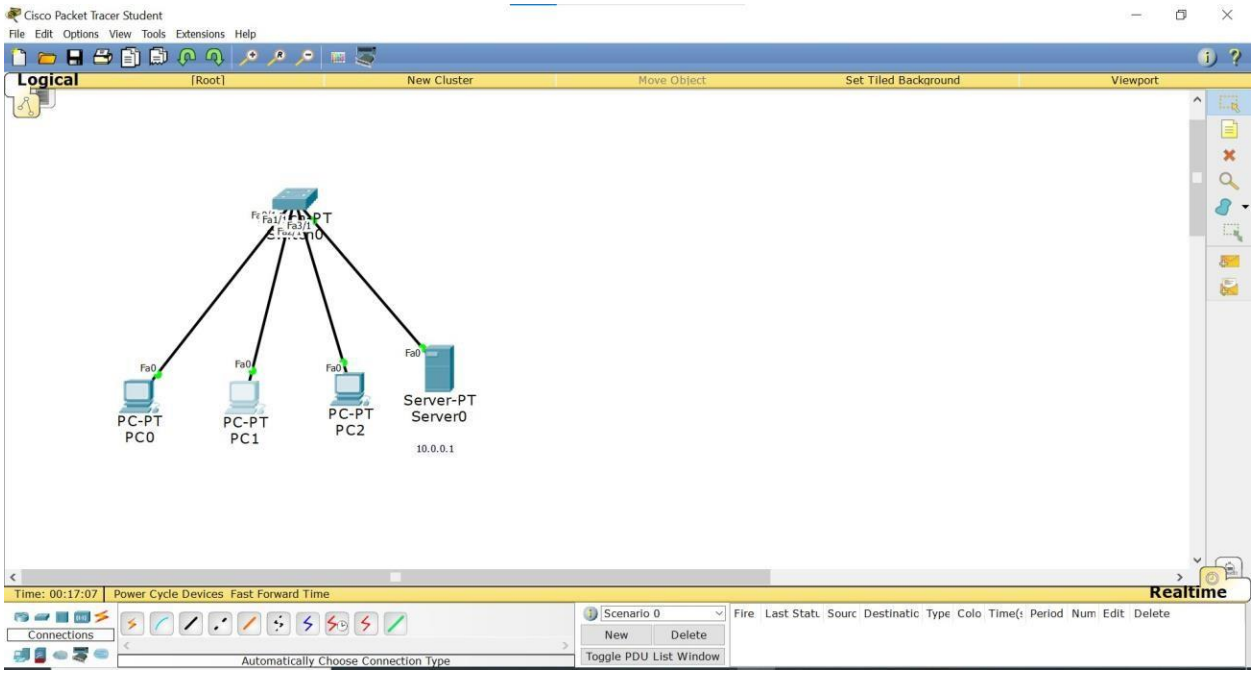
-> DHCP is used to design IP address dynamically to different devices

-> To assign continuous IP address we create a serverpool where we assign the starting IP address & a default gateway number for PC's under different switches we create a different server pool again & start.

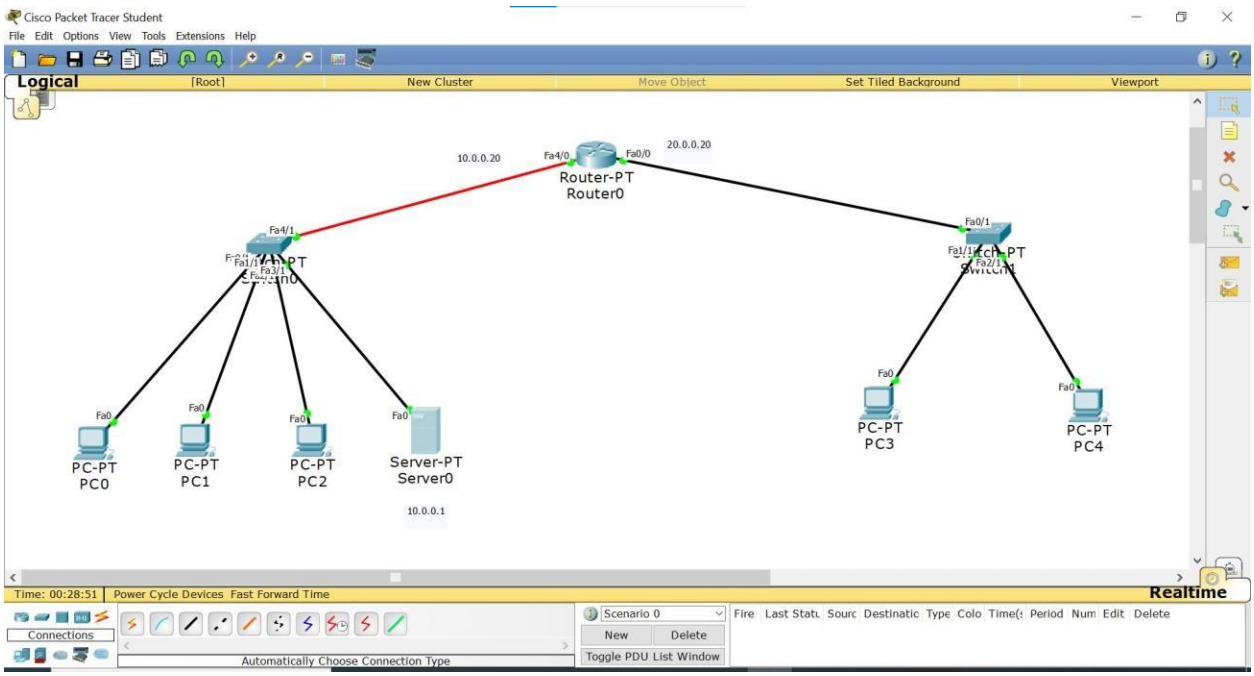
Rw
19/7/23

TOPOLOGY:

PROGRAM 4.1:

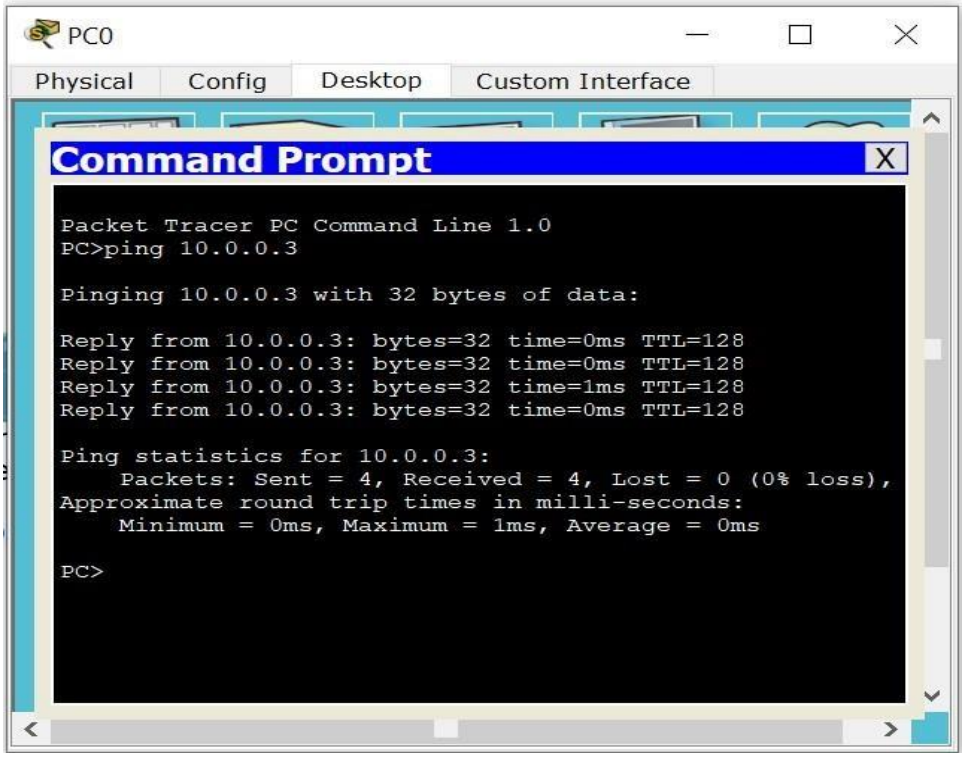


PROGRAM 4.2:

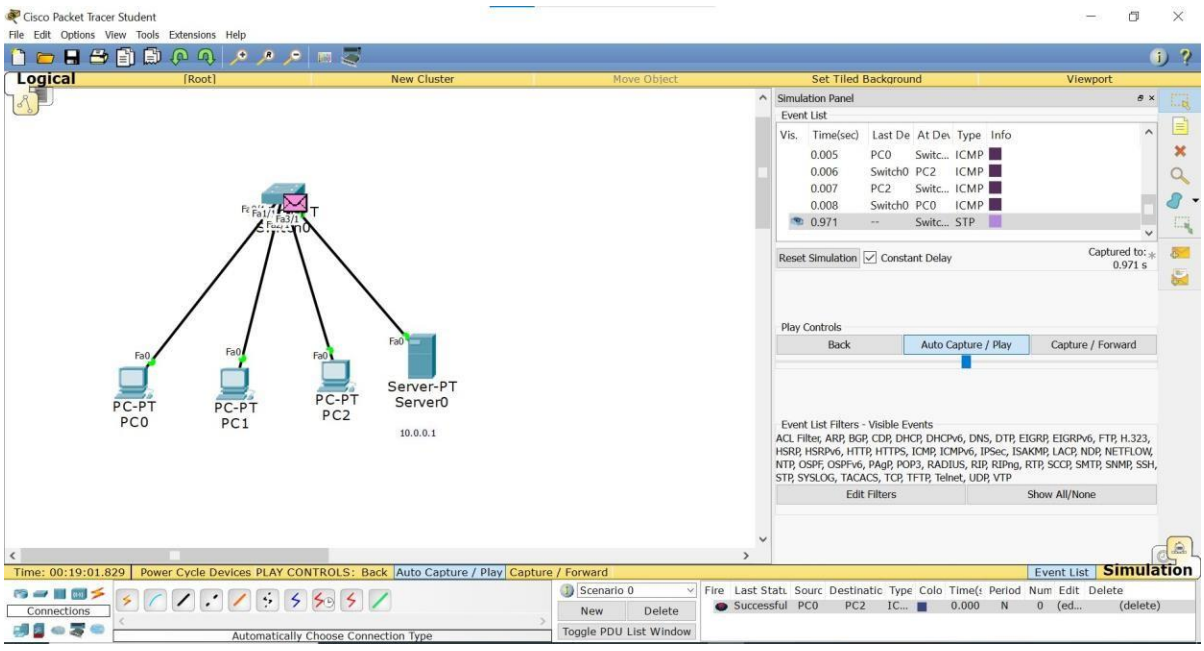


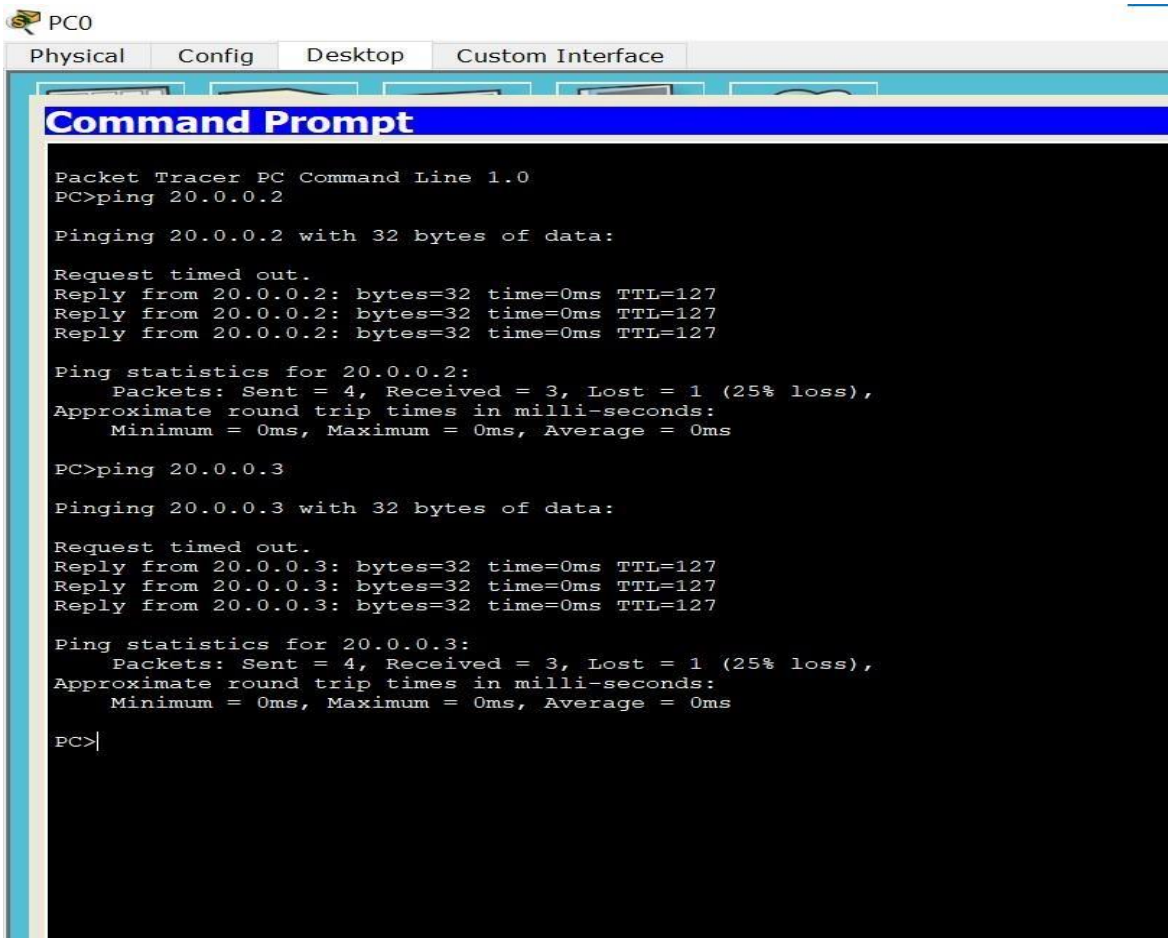
OUTPUT:

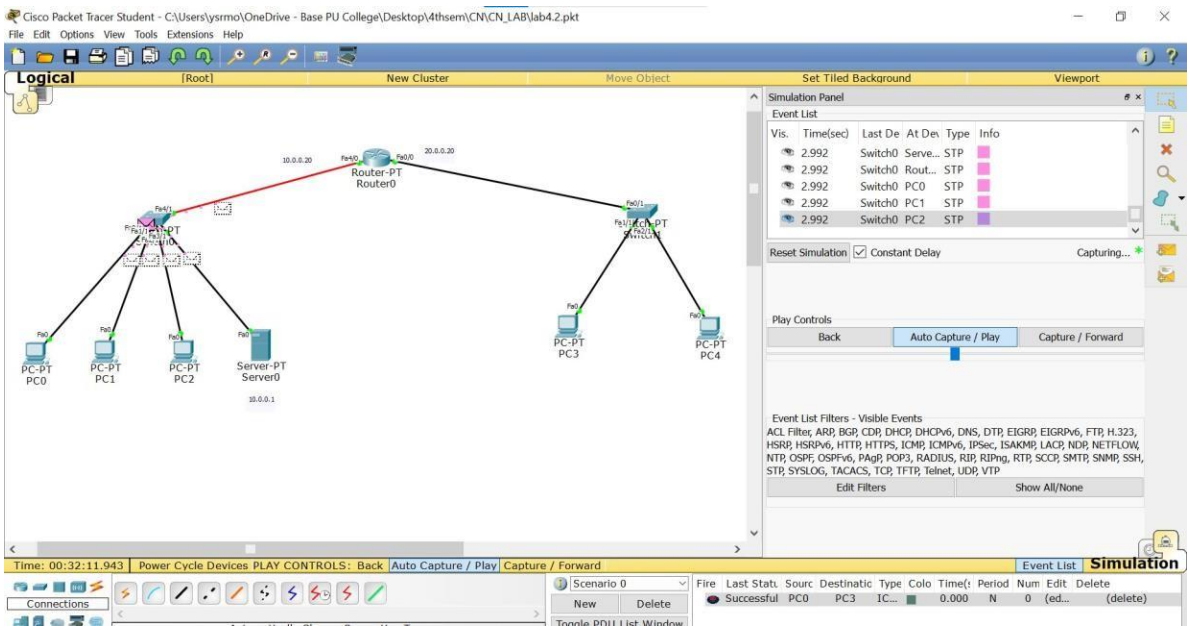
PROGRAM 4.1:



PROGRAM 4.2:



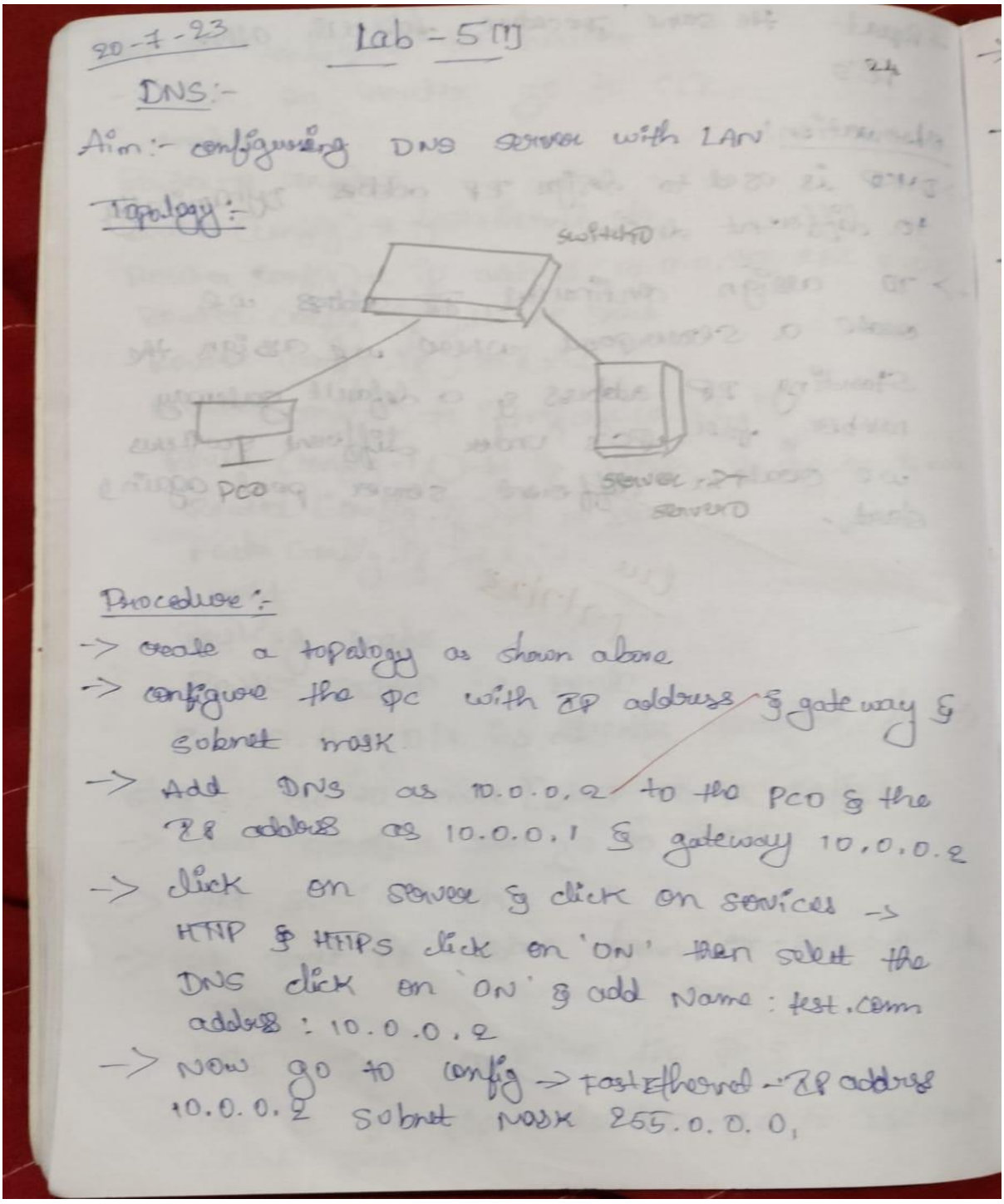

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data:
Request timed out.
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Reply from 20.0.0.2: bytes=32 time=0ms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



20-7-23 Lab-5 [1]

24

DNS:-

Aim:- configuring DNS server with LAN

Topology:-

Procedure:-

-> create a topology as shown above
-> configure the pc with IP address & gateway & subnet mask
-> Add DNS as 10.0.0.2 to the PC0 & the IP address as 10.0.0.1 & gateway 10.0.0.2
-> click on server & click on services -> HTTP & HTTPS click on 'ON' then select the DNS click on 'ON' & add Name: test.com address: 10.0.0.2
-> Now go to config -> FastEthernet -> IP address 10.0.0.2 Subnet mask 255.0.0.0,

-> Now go to settings in the config gateway 10.0.0.2 & DNS server 10.0.0.2

-> Now go to pc click on desktop -> web browser and type the 'test.com' in URL & click on go.

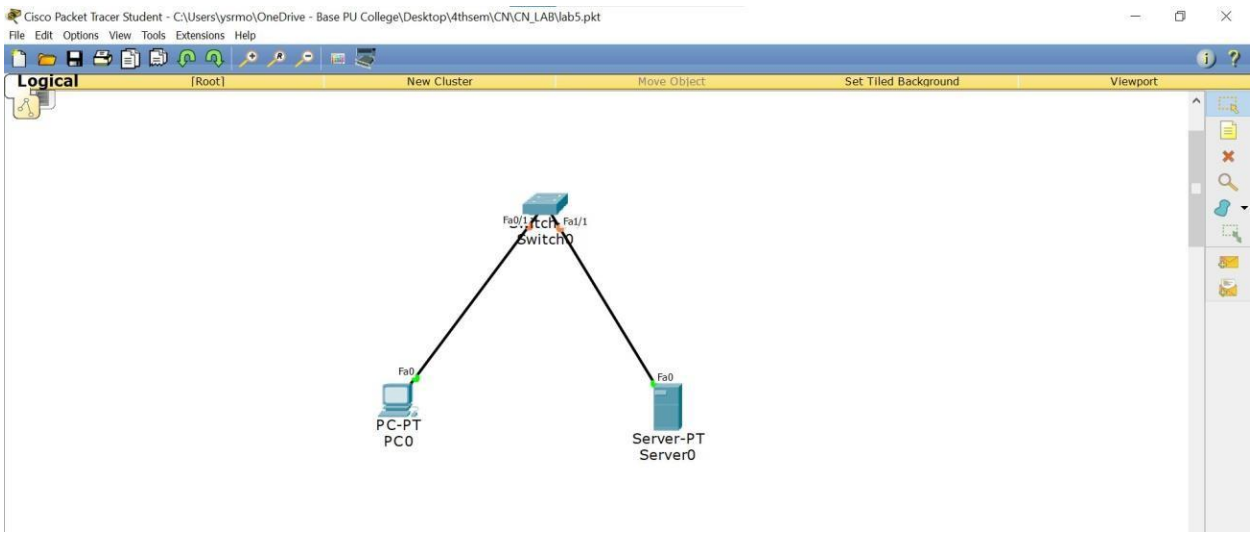
-> you will see the html. index that has written the HTTP button.

Observation:

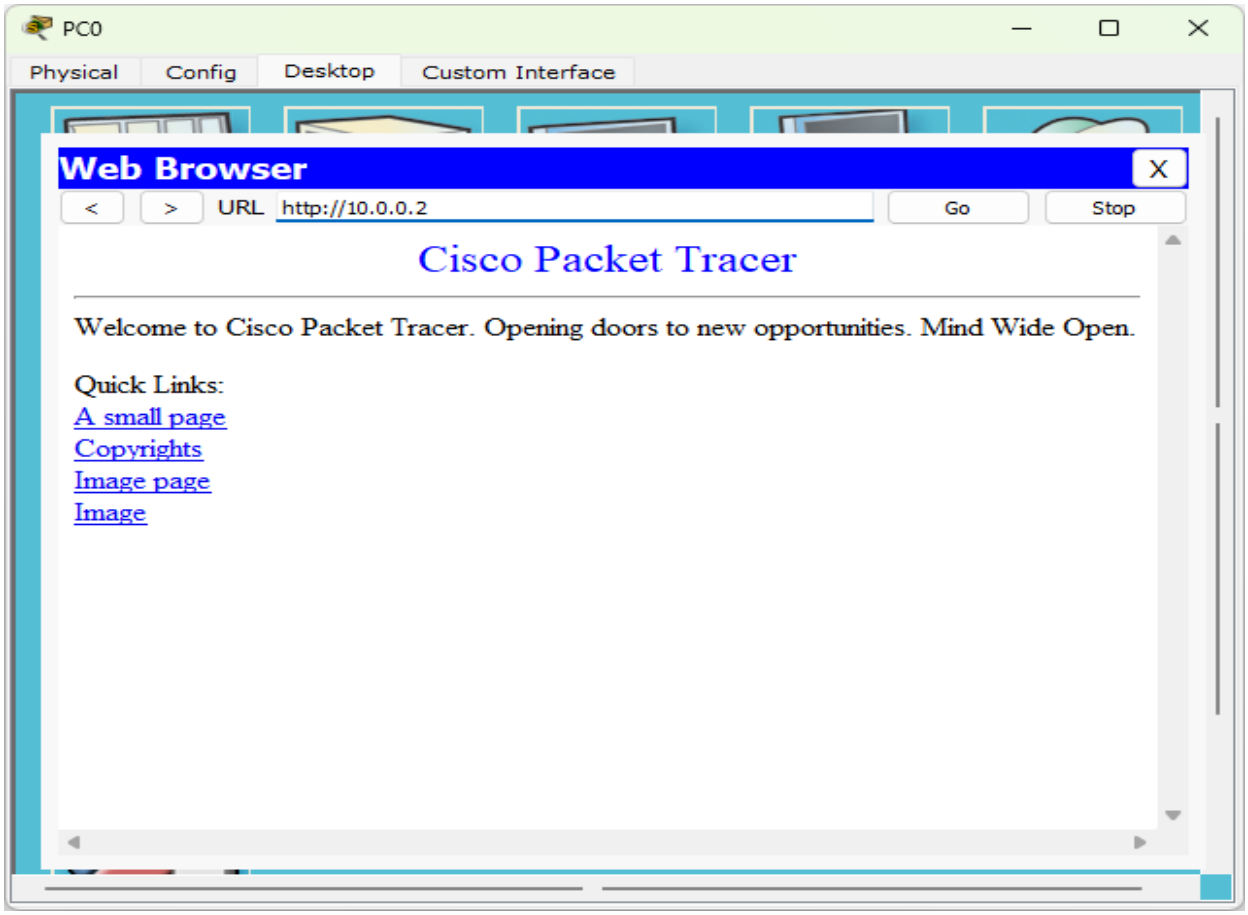
DNS is the Domain Name system. DNS is linked to the Internet & focused on a system using Internet protocol (IP). DNS servers are required for the working of DNS the IP address is calculated with the aid of a lookup table

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

TOPOLOGY:



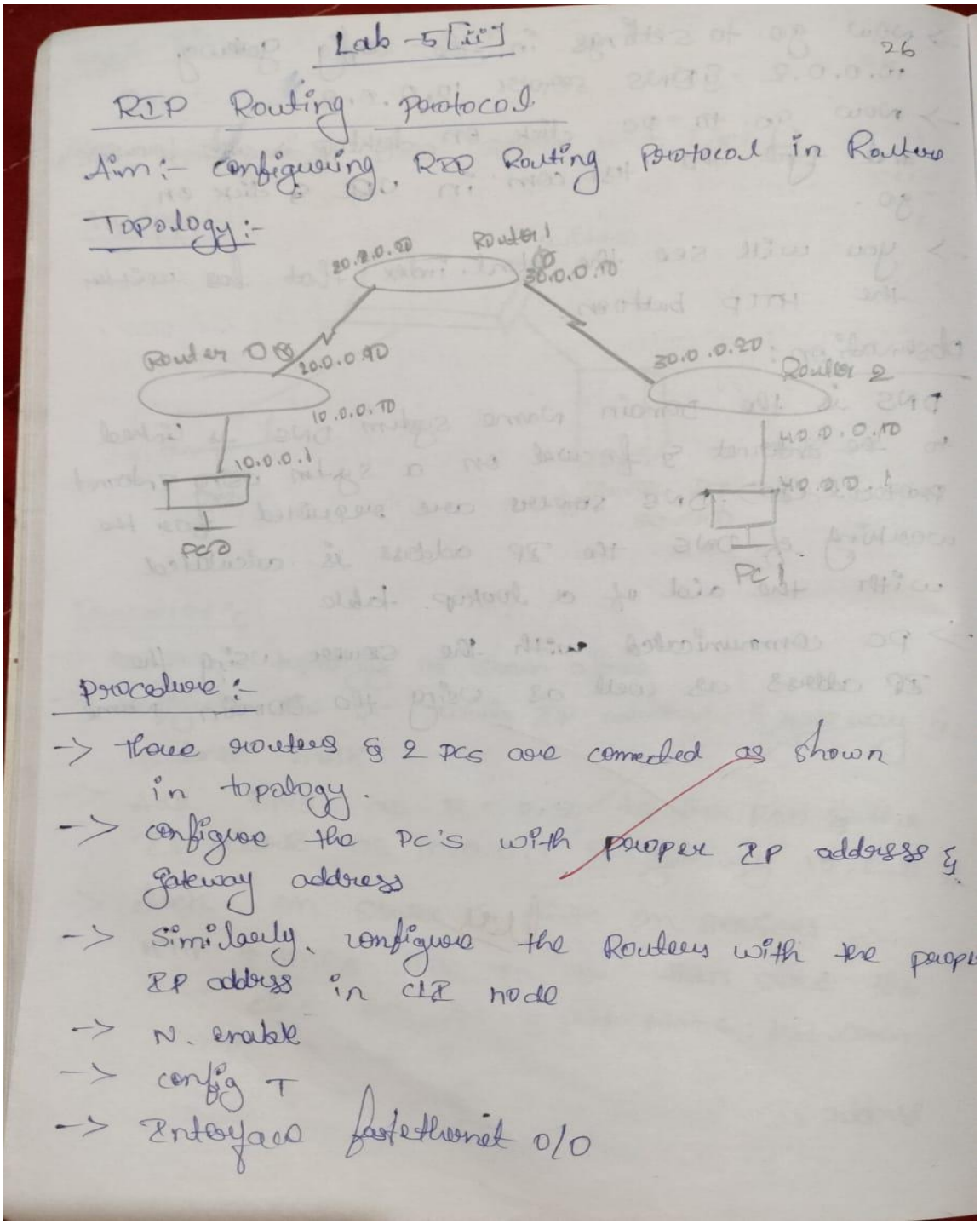
OUTPUT:



LAB 6

Configure RIP routing Protocol in Routers.

OBSERVATION:



-> IP address 10.0.0.1 255.0.0.0
-> encasulation PPP
-> clockrate 64000
no shut

Note:- the encapsulation PPP should be given to all the routers & 'clockrate 64000' command should be only given to the clock symboled sides of the routers (i.e. open sides).

-> For making the routers to know about the other devices in the previous 2 experiments we used 1 static & the other with dynamic address but 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 } router 3
-> network 40.0.0.0 }

-> router rip
-> network 10.0.0.0
-> network 20.0.0.0 } router 1

Ping output:

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:0ms TTL:12

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.

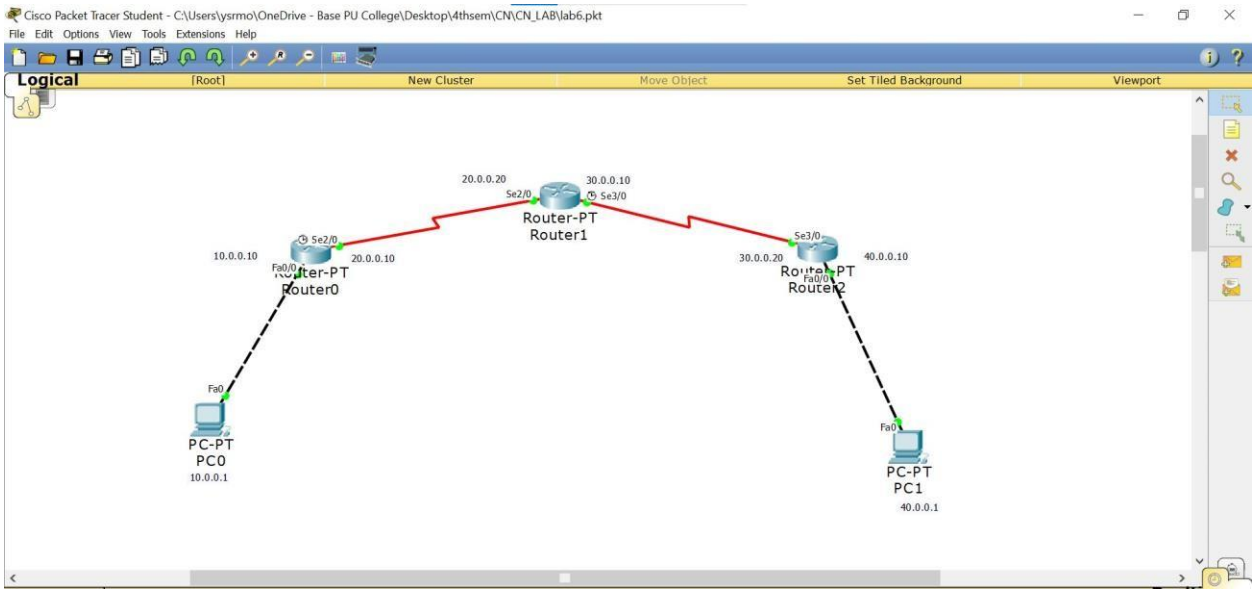
minimum = 0ms, maximum, 0ms Average = 0ms

observation :-

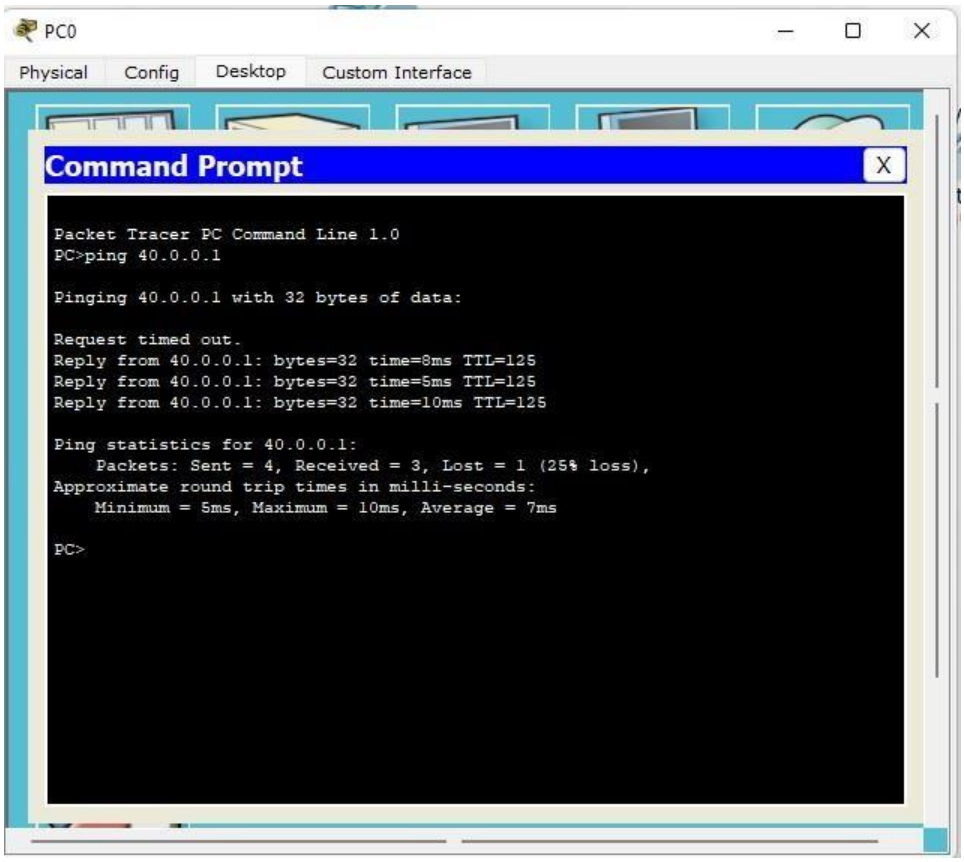
RIP is the routing Information protocol is a distance vector protocol that uses hop count as its primary metric. RIP defines how routers should share information when moving traffic among an interconnected group of local area networks.

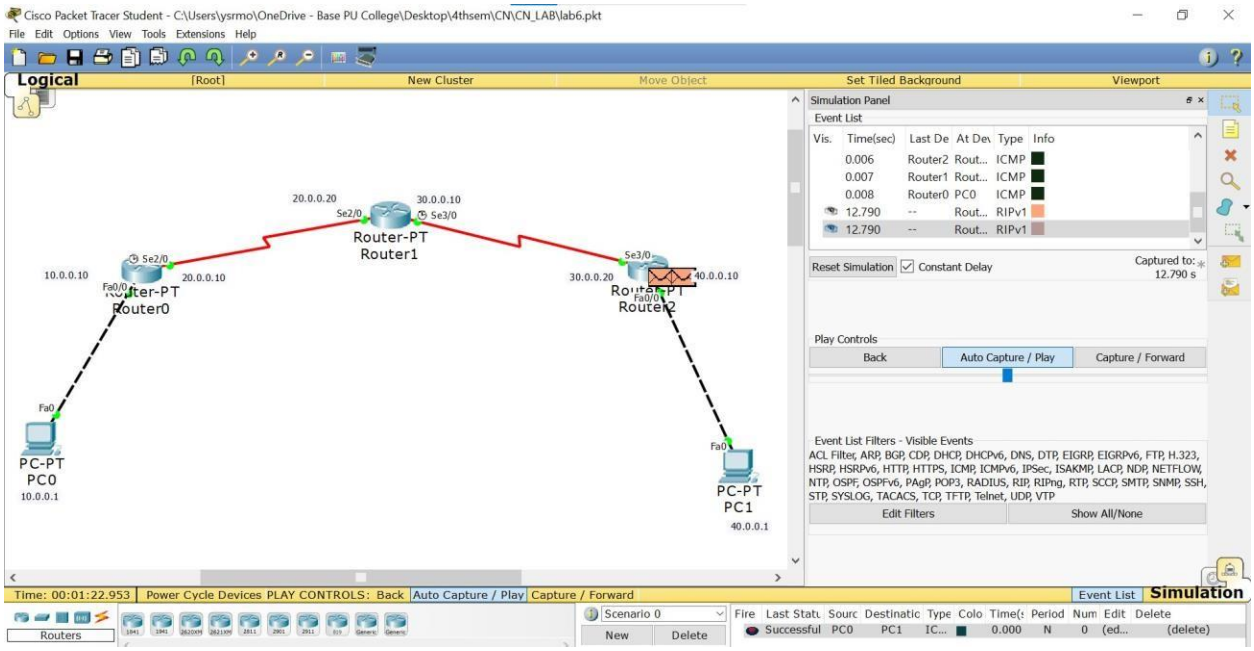
-> the RIP protocol here used to connect the routers to one other and PC's using RIP protocol and message is Pinged successfully.

TOPOLOGY:



OUTPUT:

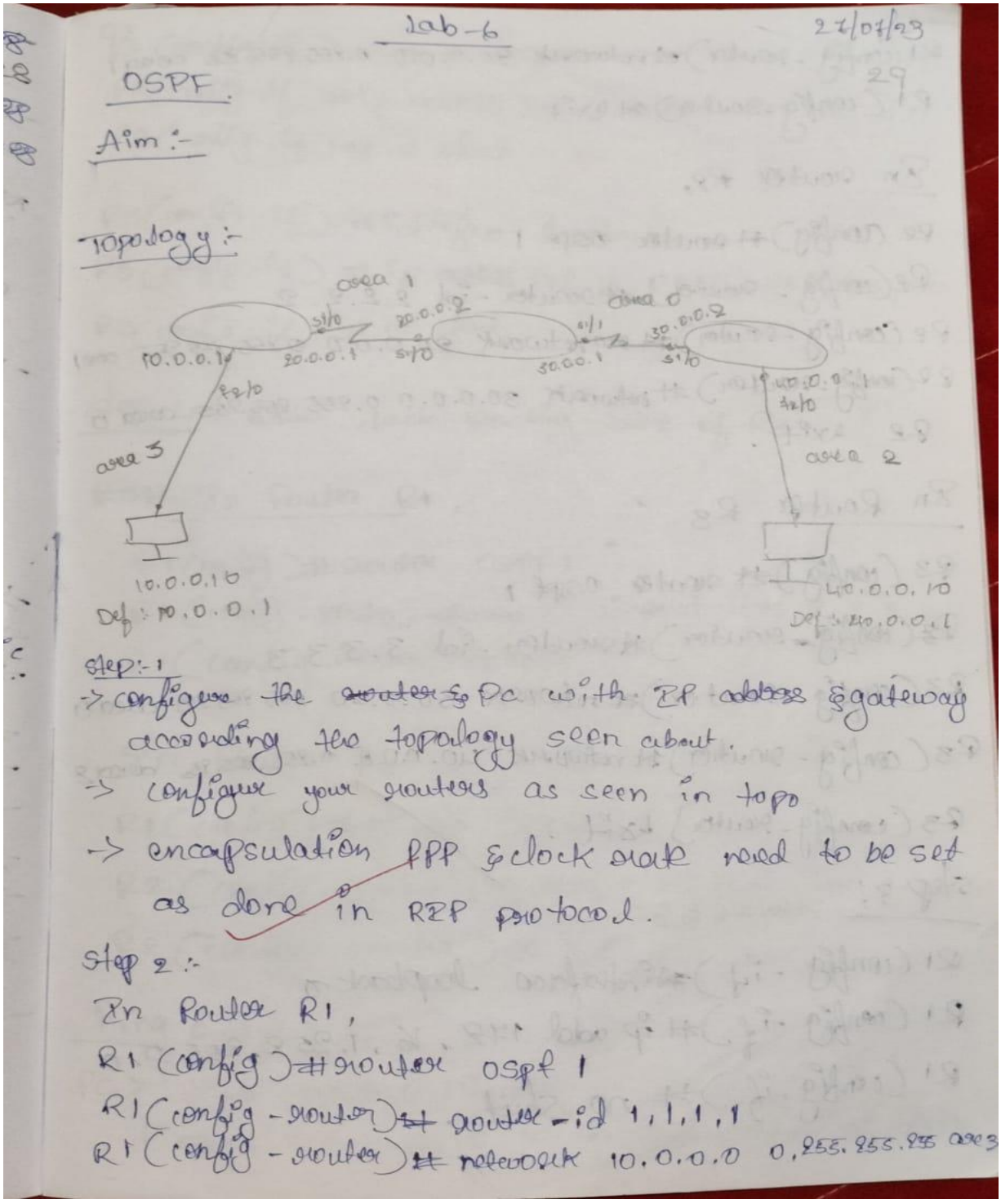



Cisco Packet Tracer Student - C:\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\CN\CN_LAB\lab6.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT
Router1
20.0.0.20
30.0.0.10
Se2/0
Se3/0
Router-PT
Router0
10.0.0.10
20.0.0.10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



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

R1 (config-router)# exit.

In router R2,

R2 (config)# router ospf 1

R2 (config-router)# router-id 2.2.2.2

R2 (config-router)# network 20.0.0.0 0.255.255.255 area 1

R2 (config-router)# network 30.0.0.0 0.255.255.255 area 0

R2 exit.

In Router R3

R3 (config)# router ospf 1

R3 (config-router)# router-id 3.3.3.3

R3 (config-router)# network 30.0.0.0 255.255.255 area 0

R3 (config-router)# network 40.0.0.0 0.255.255.255 area 2

R3 (config-router) exit.

Step 3: Interface f0/0

R1 (config-if)# interface loopback 0

R1 (config-if)# ip add 172.16.1.252.255.0.0

R1 (config-if)# no shut.

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

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

Step 4 :- Now, check Routing table of R3.

R3# In Router R1,

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

In Router R2.

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

Ping output :-

PC> Ping 40.0.0.10
Pinging 40.0.0.10 with 32 bytes of data.

32

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

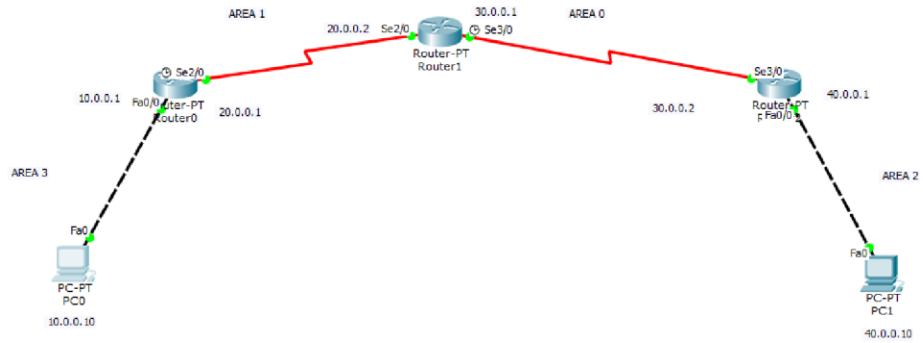
Ping statistics from 40.0.0.10

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

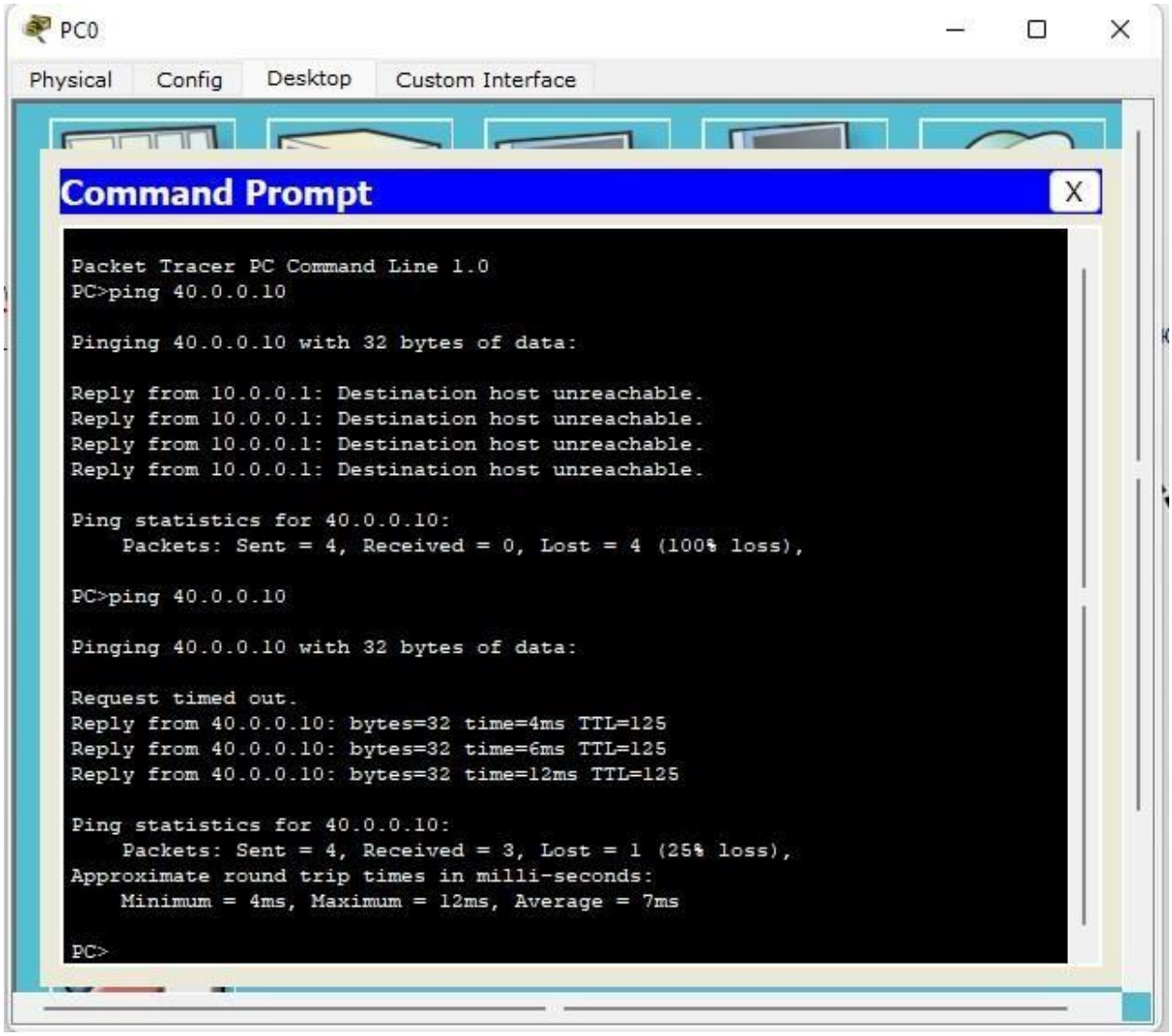
Approximate round trip times in ms

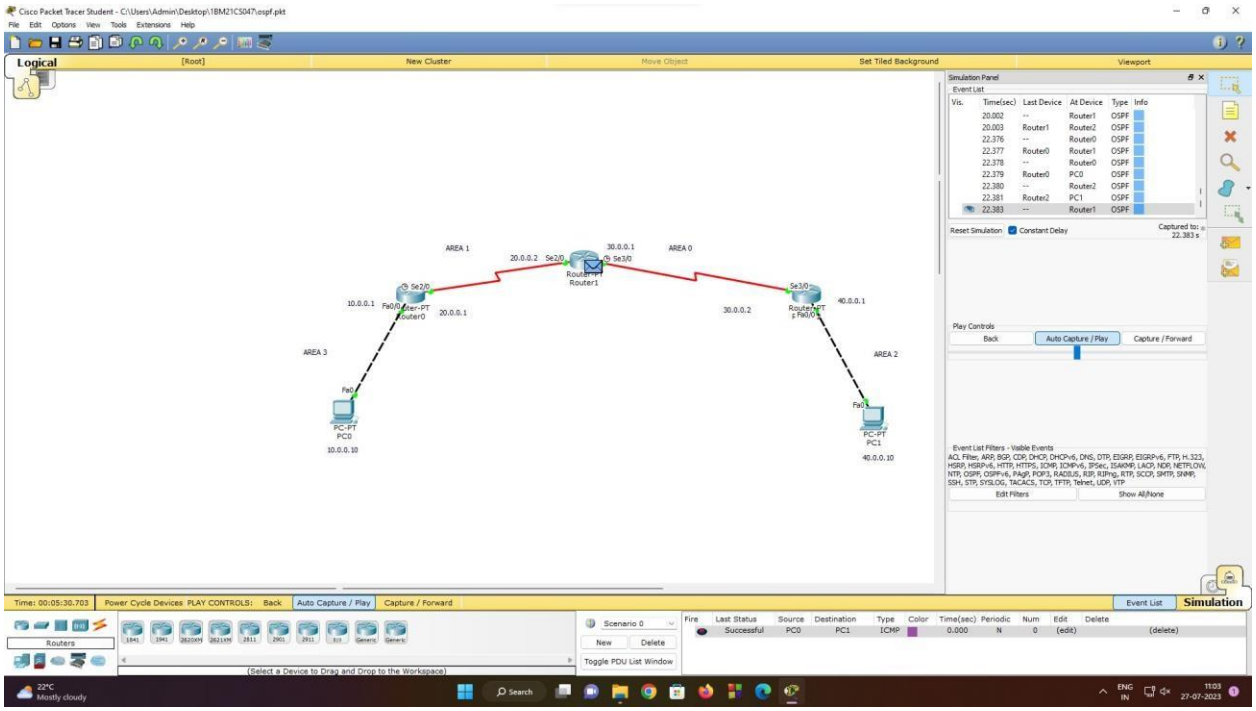
minimum = 0 ms, maximum 0ms Average = 0ms

TOPOLOGY:



OUTPUT:

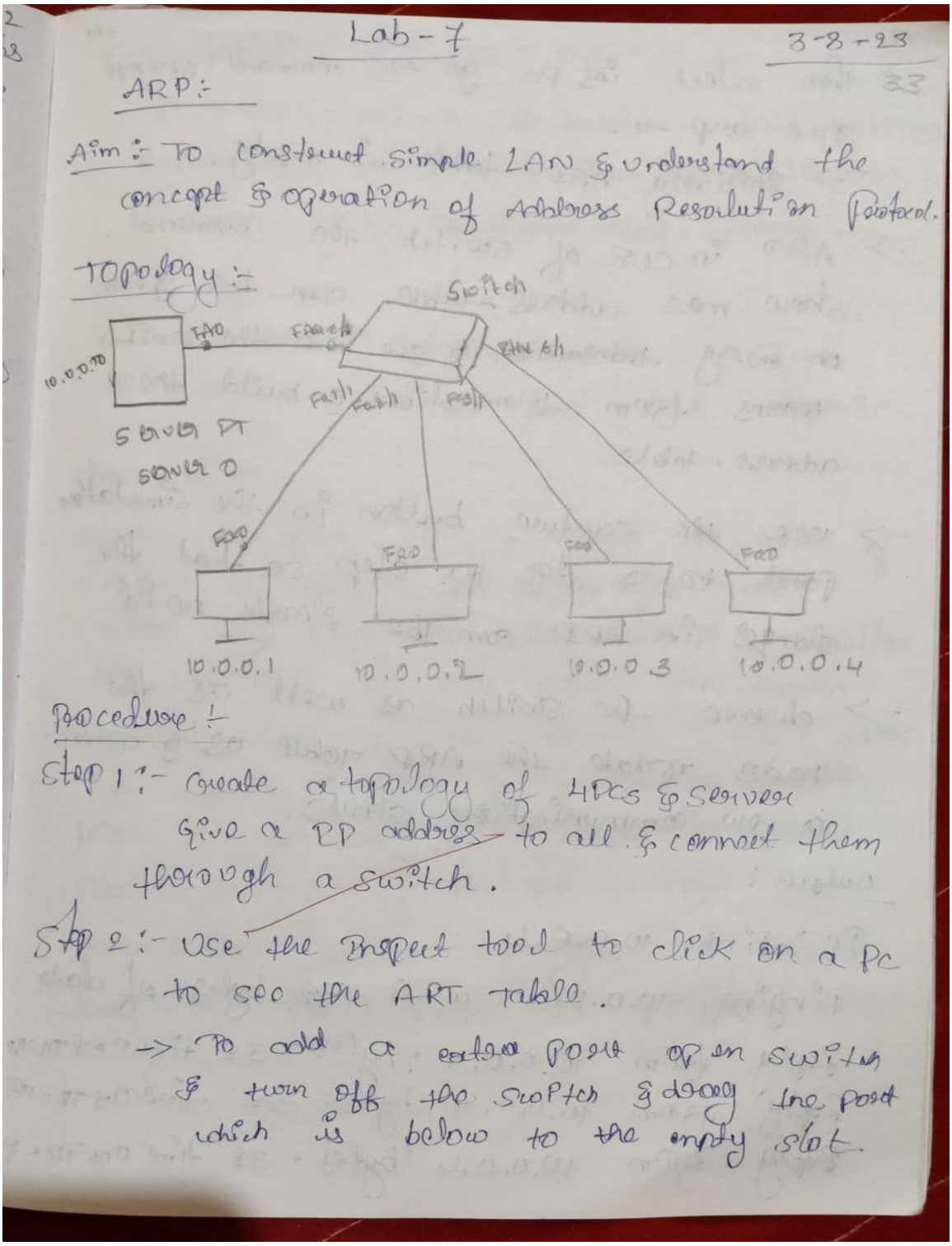



Cisco Packet Tracer Student - C:\Users\Admin\Desktop\1BM21CS047\ospf.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
AREA 1
AREA 0
AREA 3
AREA 2
20.0.0.2
Se2/0
30.0.0.1
Se3/0
Router1
10.0.0.1
Fa0/0
20.0.0.1
30.0.0.2
40.0.0.1
PC-PT
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
Scenario 0
New
Delete
Toggle PDU List Window
(Select a Device to Drag and Drop to the Workspace)
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:



→ then select any PC go to command prompt give arp -a.

→ Initially ARP table is empty.

→ Also in CLI of switch the command show mac address-table can be given on every transaction to see how the switch learns from transactions & build the address-table

→ use the capture button in the simulation panel to go step by step so that the changes in ARP can be clearly noted

→ observe the switch as well as the nodes update the ARP table as & when a new communication starts.

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 0ms TTL=128

Reply from 10.0.0.4 bytes = 32 time 0ms TTL=128

Reply from 10.0.0.4 bytes = 32 time 0ms TTL<128

Ping statistics for 10.0.0.4:

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

Approximate round trip time in milliseconds

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

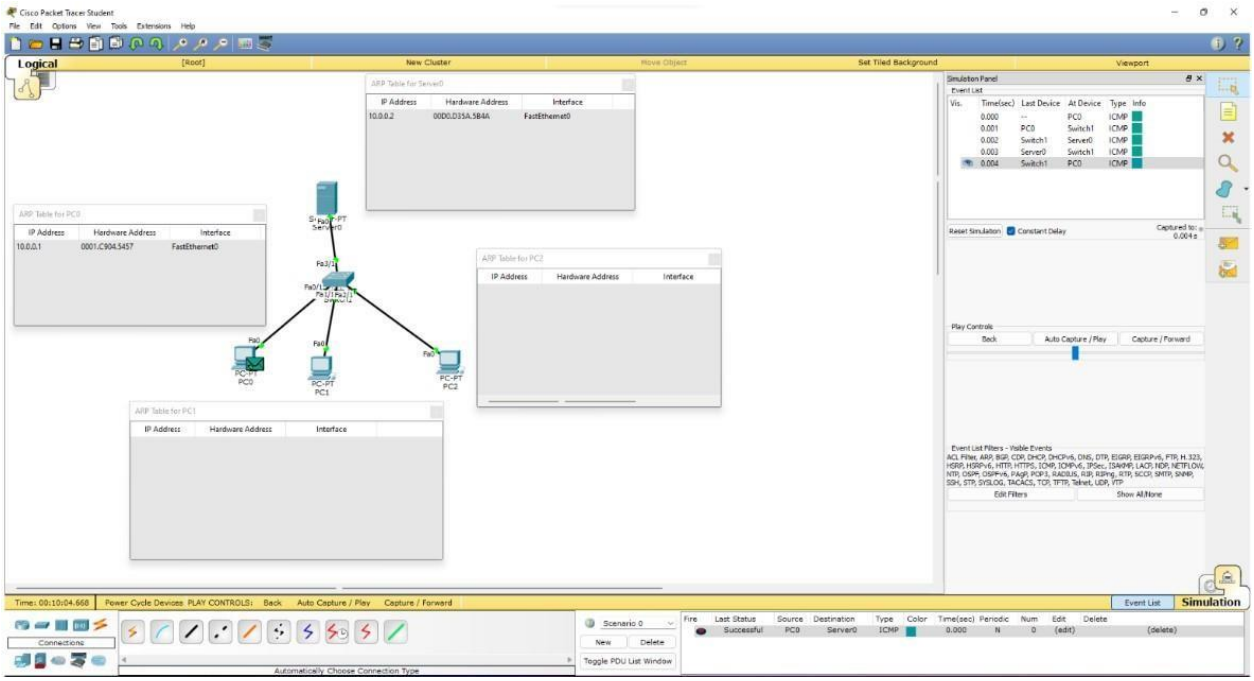
PC > arp -a

Internet address	physical address	Type
10.0.0.4	0060.2f00.3244	dynamic.

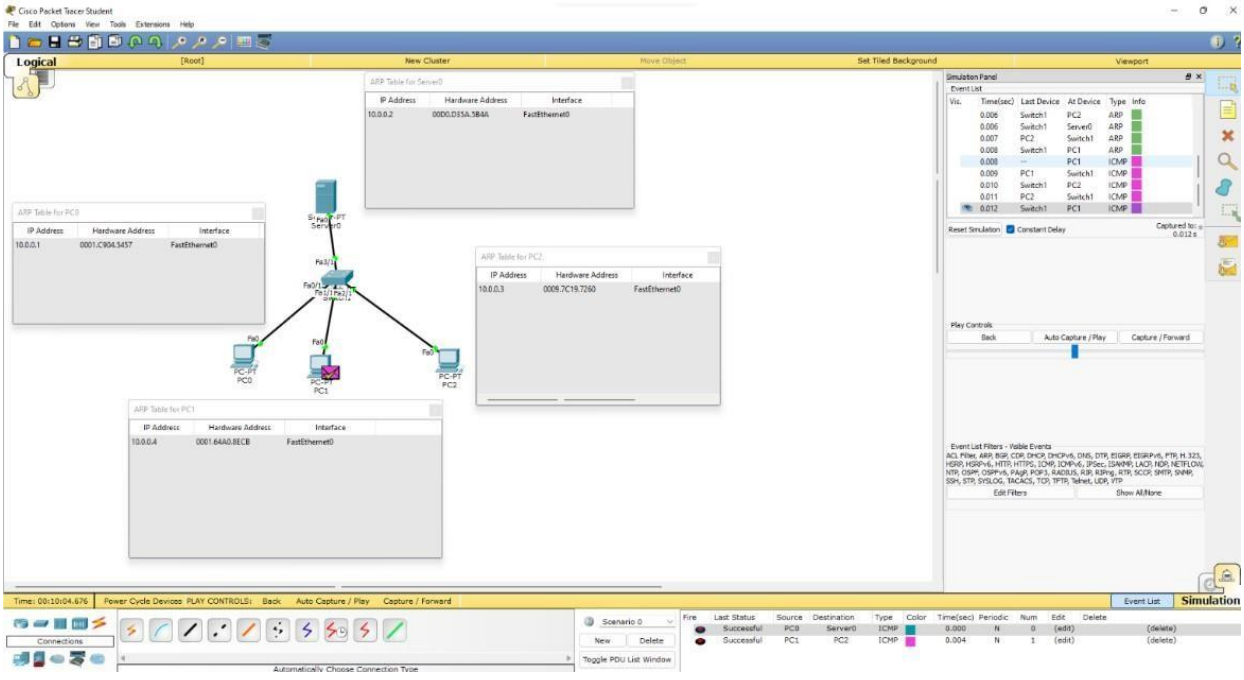
observation:

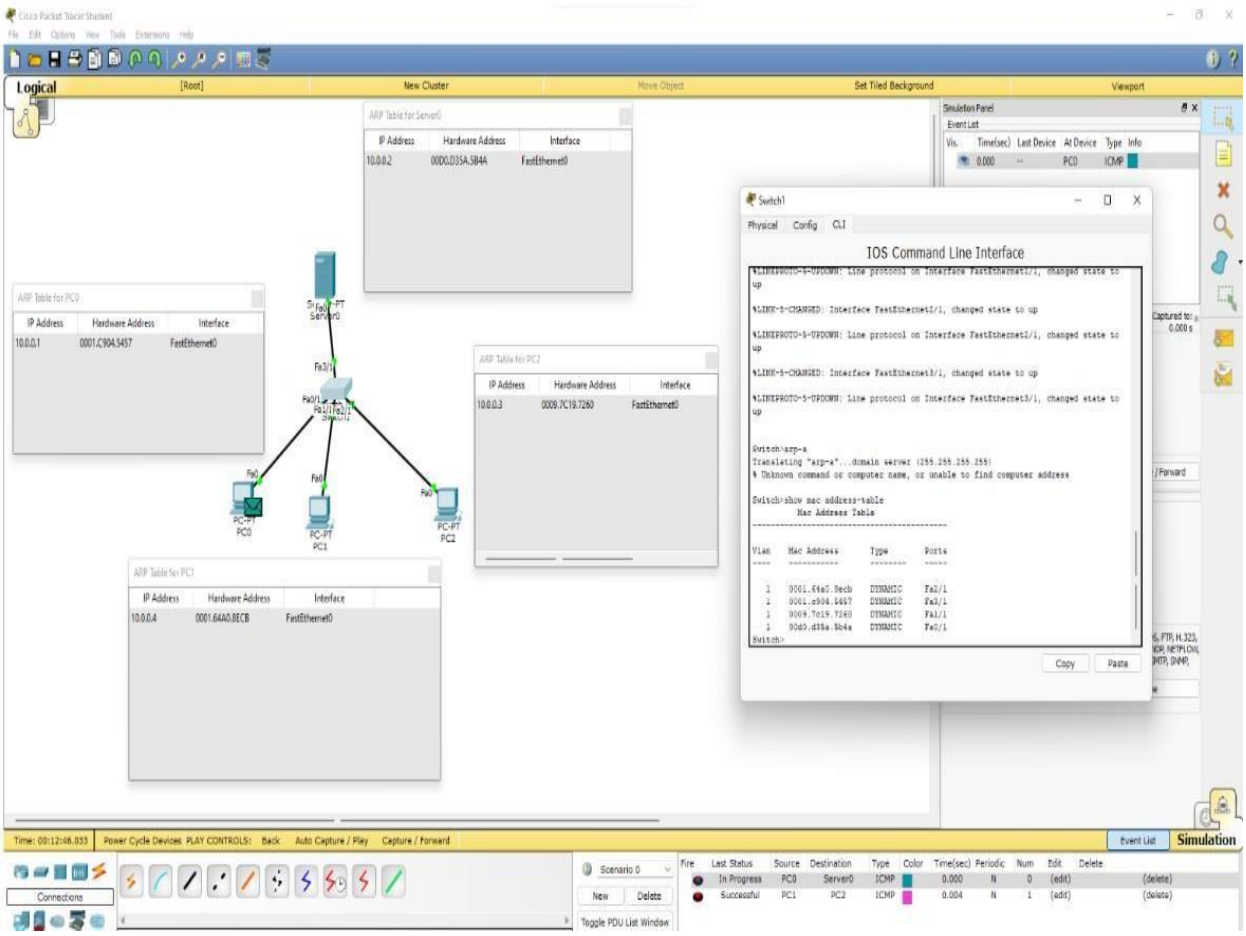
- when we ping PC & server the address of server is known to PC & vice-versa.
- when we ping b/w other two PC's simultaneously the addresses of each other are known.
- Every time a host requests a MAC address in order to send a packet to another host in the LAN, it checks its ARP cache to see if the IP to MAC address translation address already exists of the translation doesn't exists it performs ARP.

TOPOLOGY:



OUTPUT:

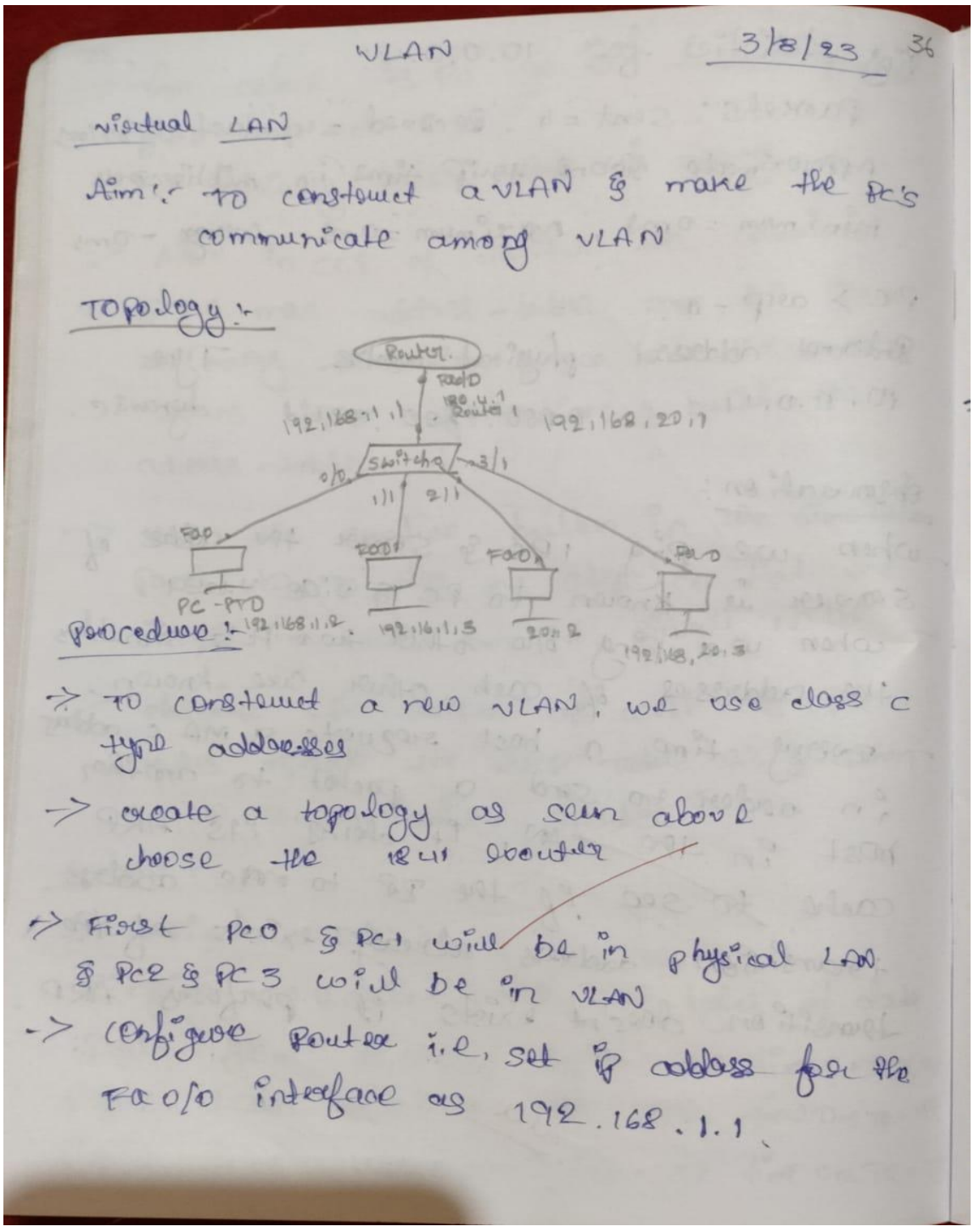



Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
ARP Table for Server0
IP Address
Hardware Address
Interface
10.0.0.2
0060.D35A.5B4A
FastEthernet0
ARP Table for PC0
10.0.0.1
0001.C904.5457
FastEthernet0
ARP Table for PC2
10.0.0.3
0009.7C19.7260
FastEthernet0
ARP Table for PC1
10.0.0.4
0001.64A0.8ECB
FastEthernet0
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 Fa2/1
Switch>
Copy
Paste
Simulation Panel
Event List
Vis.
Time(sec)
Last Device
At Device
Type
Info
0.000
PC0
ICMP
Captured to: 0.000 s
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:



VLAN 3/8/23 36

Virtual LAN

Aim:- To construct a VLAN & make the PC's communicate among VLAN

Topology:-

Procedure:-

-> To construct a new VLAN, we use class C type addresses

-> create a topology as seen above choose the 18 4 1 2 routers

-> First PC0 & PC1 will be in physical LAN & PC2 & PC3 will be in VLAN

-> Configure Router i.e, set IP address for the Fa 0/0 interface as 192.168.1.1

- And set ip address of PC0 & PC1 as 192.168.1.2 & 192.168.1.3 & gateway as 192.168.1.1

-> now we can check that PC0 & PC1 can communicate with each other.

-> Now PC2 & PC3 set IP as 192.168.20.2 & 192.168.20.3 & gateway as 192.168.20.1

=> Switch configuration

-> In switch go to config & select VLAN data base. Set VLAN no & name:-

ex: VLAN Number 20
VLAN Name Newvlan.

-> click on add.

-> select the interface. i.e fa 6/1 (near the switch from router) & make it trunk.

-> VLAN trunking allows switches to forward frames from different VLANs over a single link called trunk.

-> this is done by adding an additional header information called tag to the ethernet frame. the process of adding this small header is called VLAN tagging.

→ And make (select) the interface that are connecting VLAN PC's to the switch..

→ Here it is fa 2/1 & fa 3/1 & select & make VLAN as 20 : new vlan.

VLAN [20]

☐ 1 : default

☑ 20 : newvlan

Router configuration:

→ open config selet VLAN database enter the no & name of vlan created.

go to CLI

Router (vlan) #exit.

Apply completed.

exiting...

Router # config t.

Router (config)# interface fa 0/0.1

Router (config-subif)# encapsulation dot1q 2.

Router (config-subif)# ip address 192.168.20.1 255.255.255.0

Router (config-subif)# no shut.

Now Ping

from PC0 to PC3

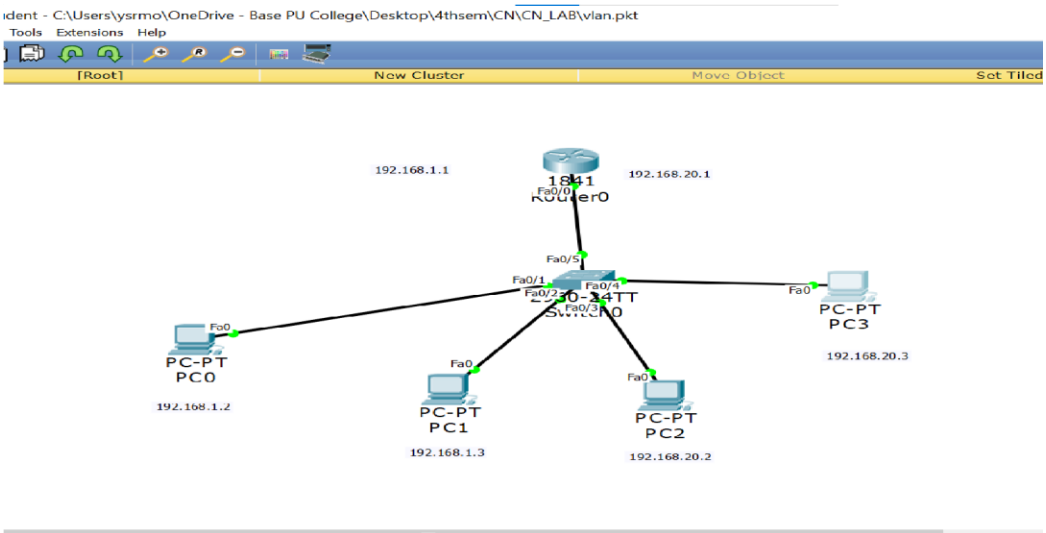
→ Ping 192.168.20.3

you will get a successful transmission from PC0 to PC3.

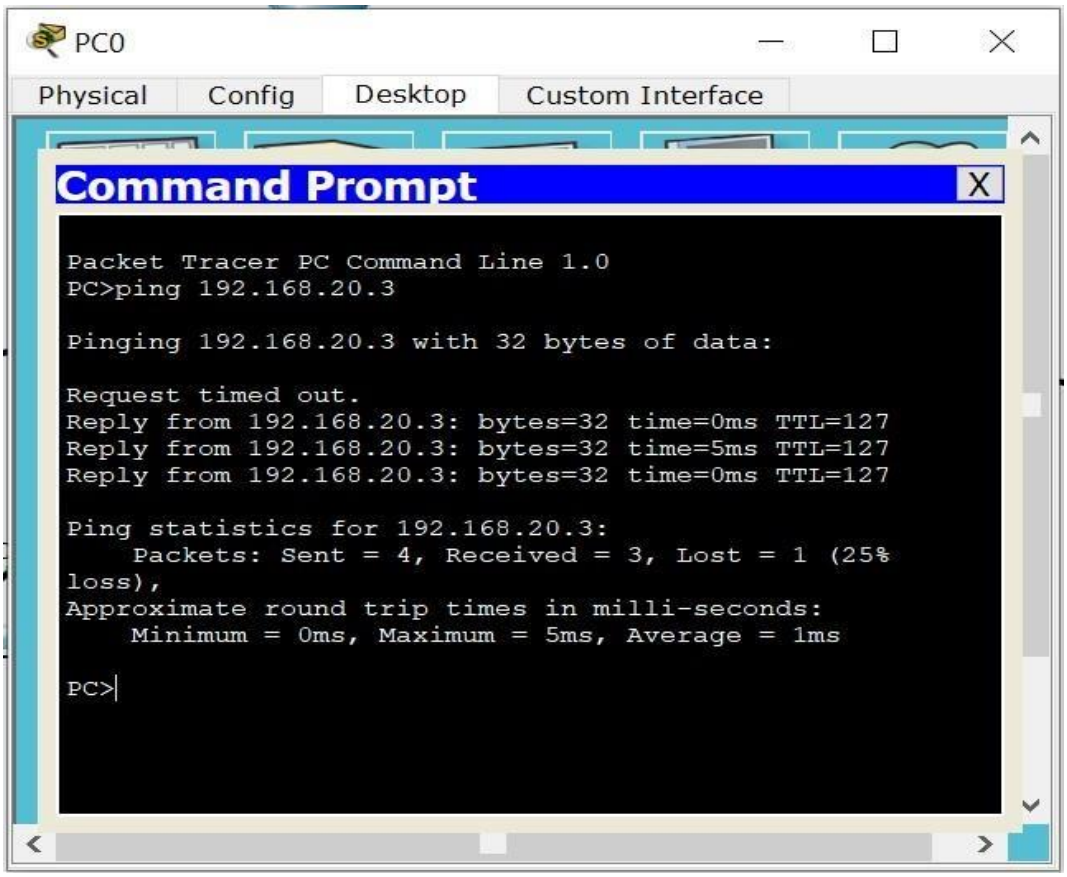
Observation :-

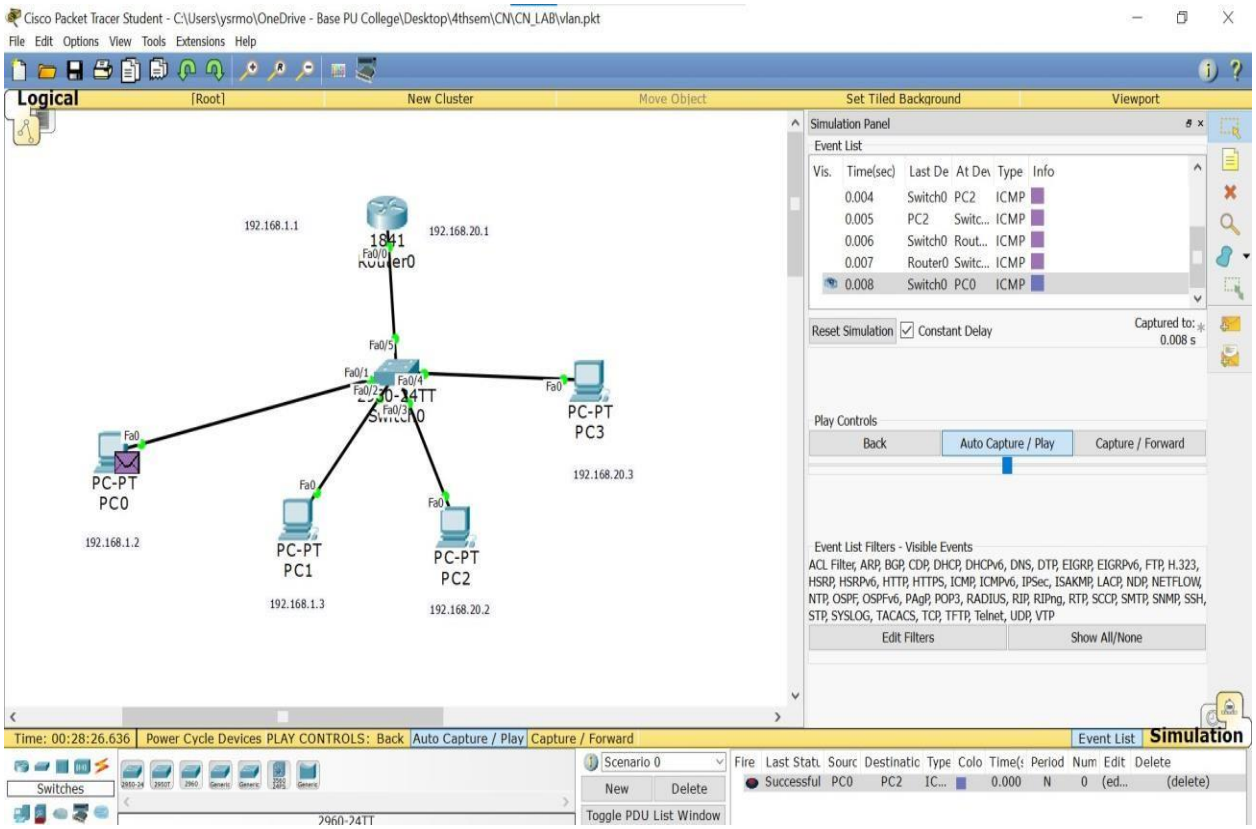
Even though we are using a single router we can use multiple different networks. And those networks will works as virtual networks. And we can communicate from physical LAN to VLAN & vice versa.

TOPOLOGY:



OUTPUT:

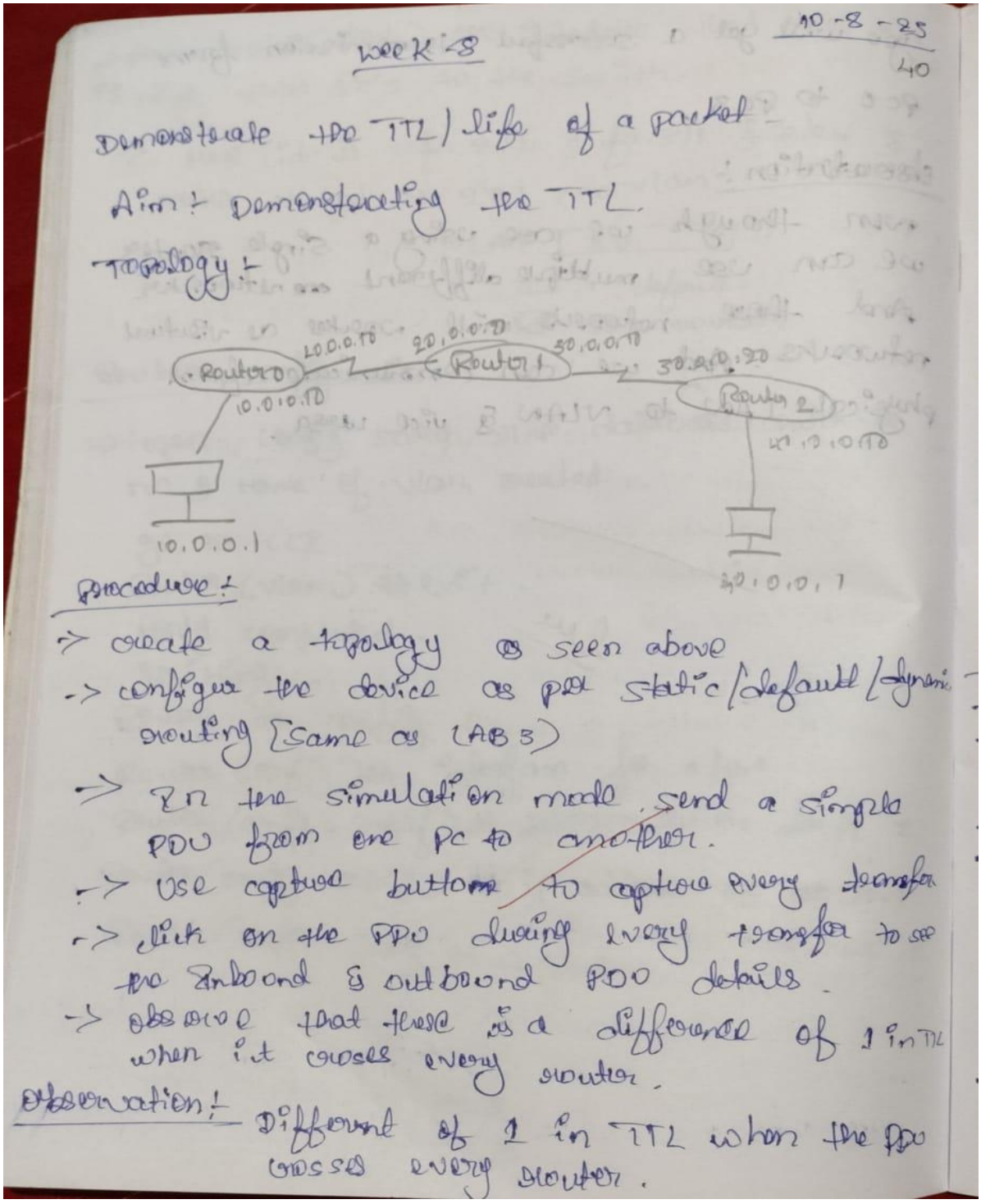



Cisco Packet Tracer Student - C:\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\CN\CN_LAB\vlan.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
192.168.1.1
1841
Router0
192.168.20.1
Fa0/0
Fa0/5
Fa0/1
Fa0/2
Fa0/4
Fa0/3
2960-24TT
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:



Week-8 10-8-25

40

Demonstrate the TTL / life of a packet

Aim :- Demonstrating the TTL

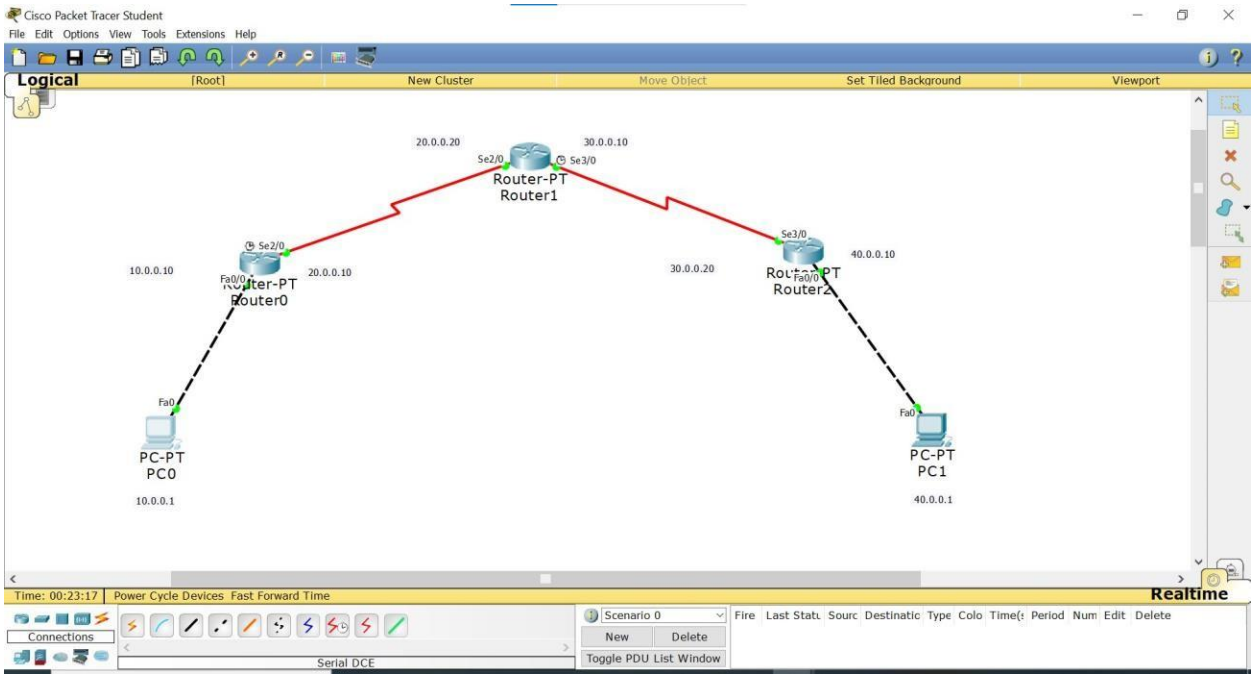
Topology :-

Procedure :-

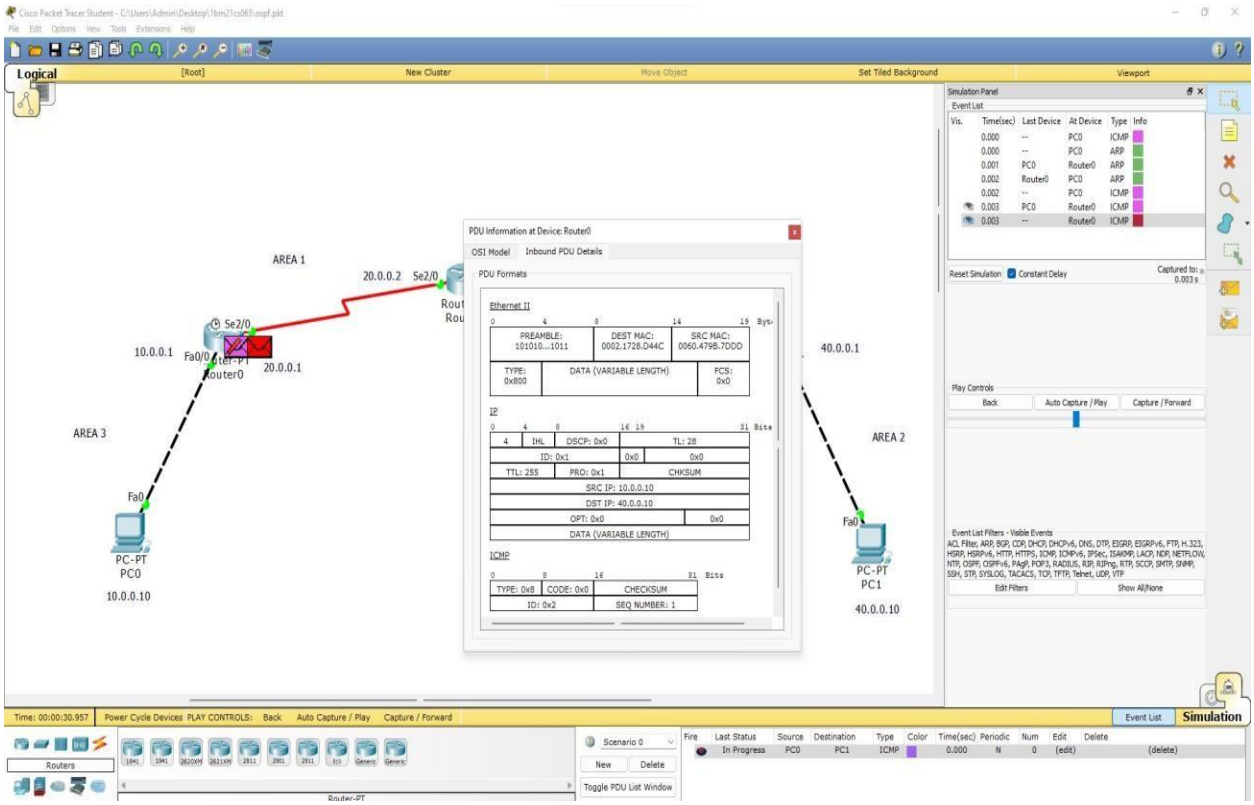
-> create a topology as seen above
-> configure the device as per static / default / dynamic routing [Same as LAB 3]
-> In the simulation mode, send a simple PDU from one PC to another.
-> Use capture buttons to capture every transfer
-> click on the PDU during every transfer to see the inbound & outbound PDU details.
-> observe that there is a difference of 1 in TTL when it crosses every router.

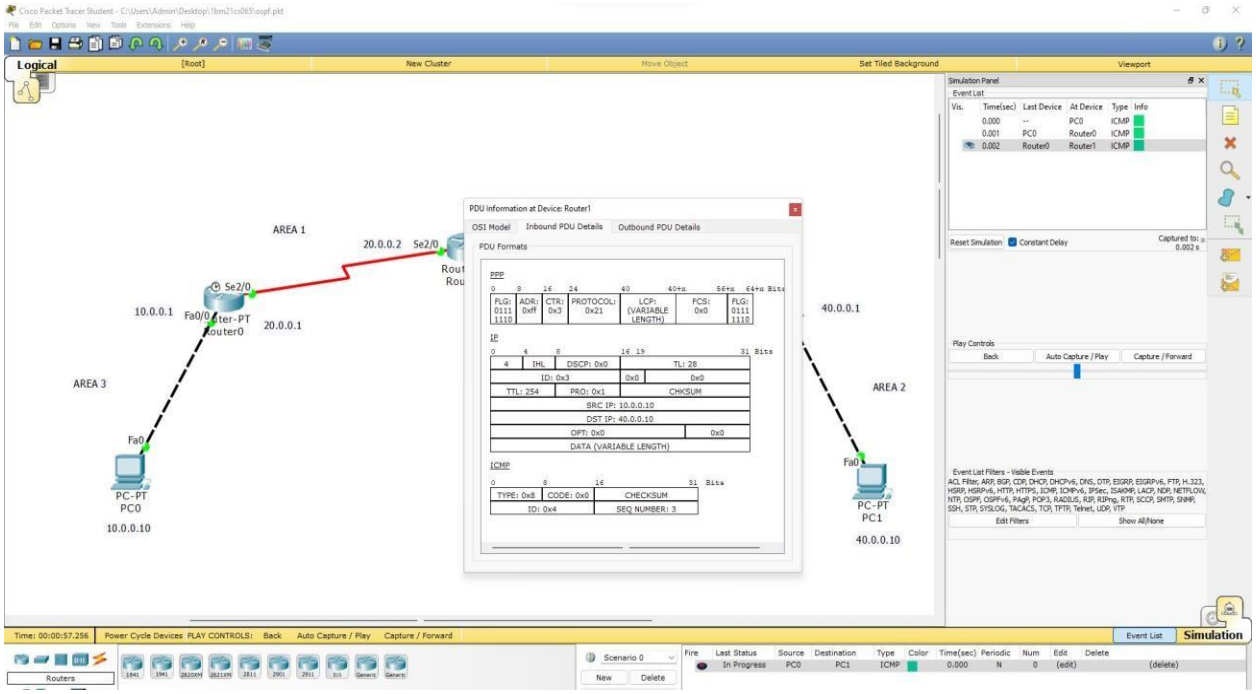
Observation :- Different of 1 in TTL when the PDU crosses every router.

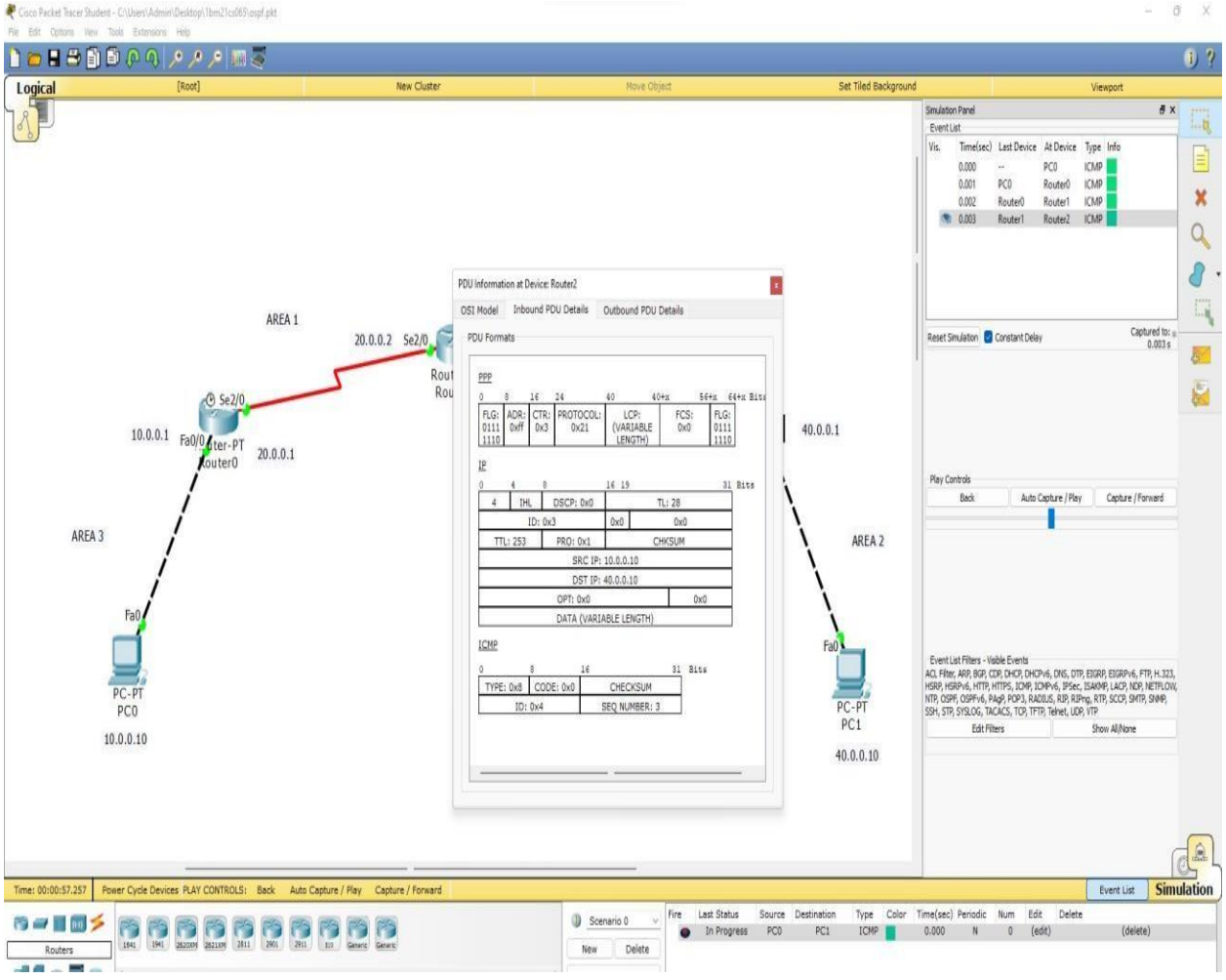
TOPOLOGY:

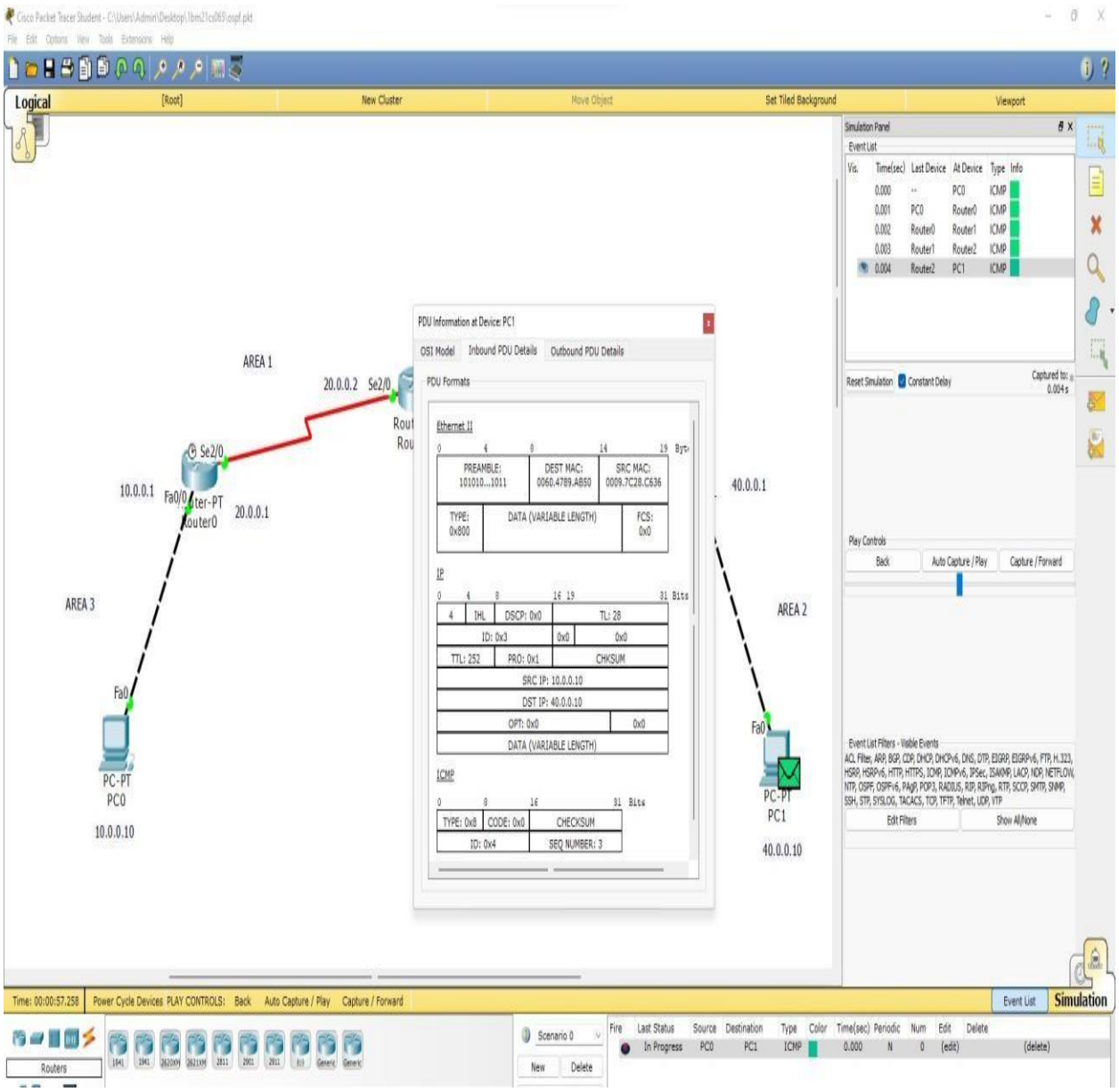


OUTPUT:




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

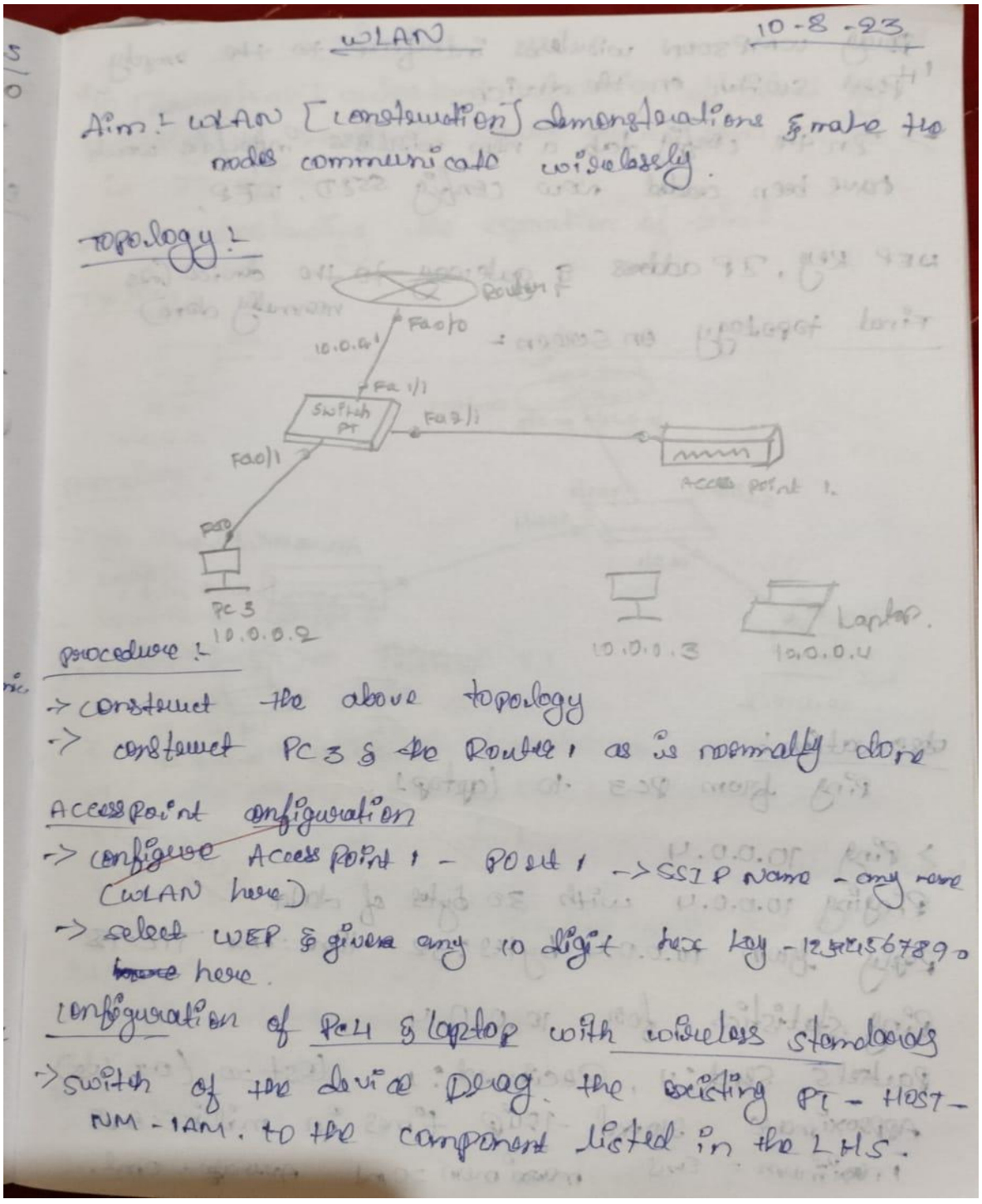

Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
AREA 1
AREA 2
AREA 3
20.0.0.2
Se2/0
10.0.0.1
Fa0/0
20.0.0.1
40.0.0.1
Fa0
PC-PT
PC0
10.0.0.10
PC-PT
PC1
40.0.0.10
PDU Information at Device: Router2
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
PPP
IP
ICMP
Simulation 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:

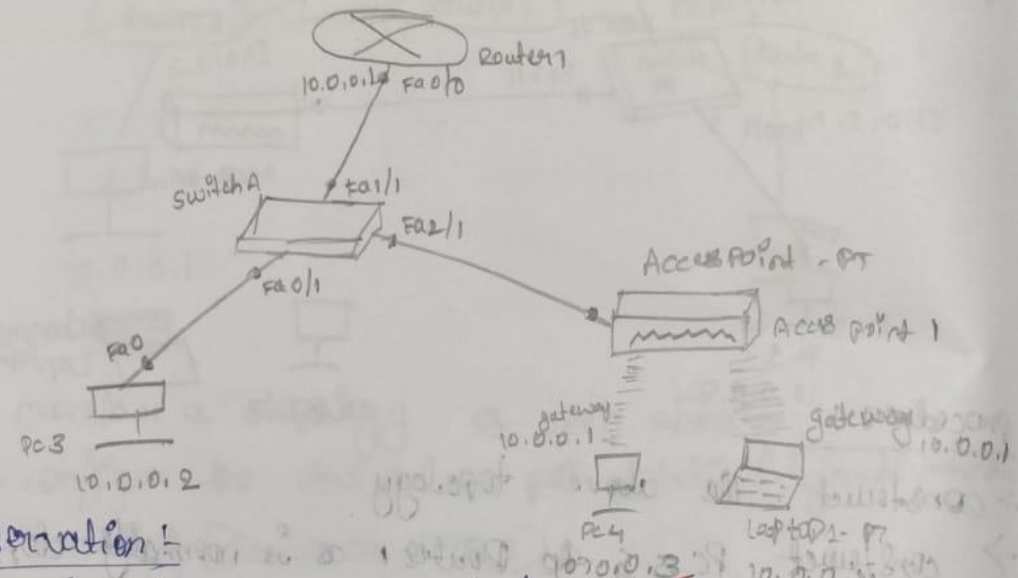


Drag WMP300N wireless interface to the empty PORT slot on the device.

→ In the config tab a new wireless interface would have been added now config SSID, WEP,

WEP key, IP address & gateway to the device (as normally done)

Final topology on screen:-



observation:-

Ping from PC3 to laptop1.

> Ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data

Reply from 10.0.0.4 bytes: 32 time=0ms TTL=128

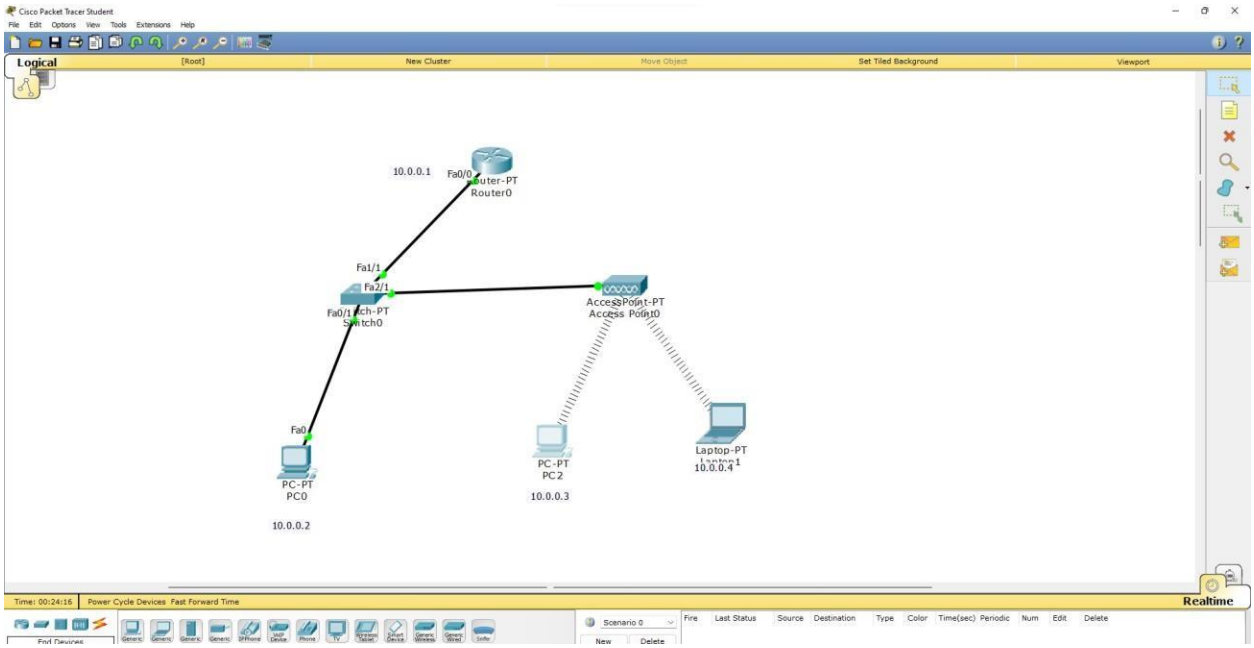
Ping statistics for 10.0.0.4

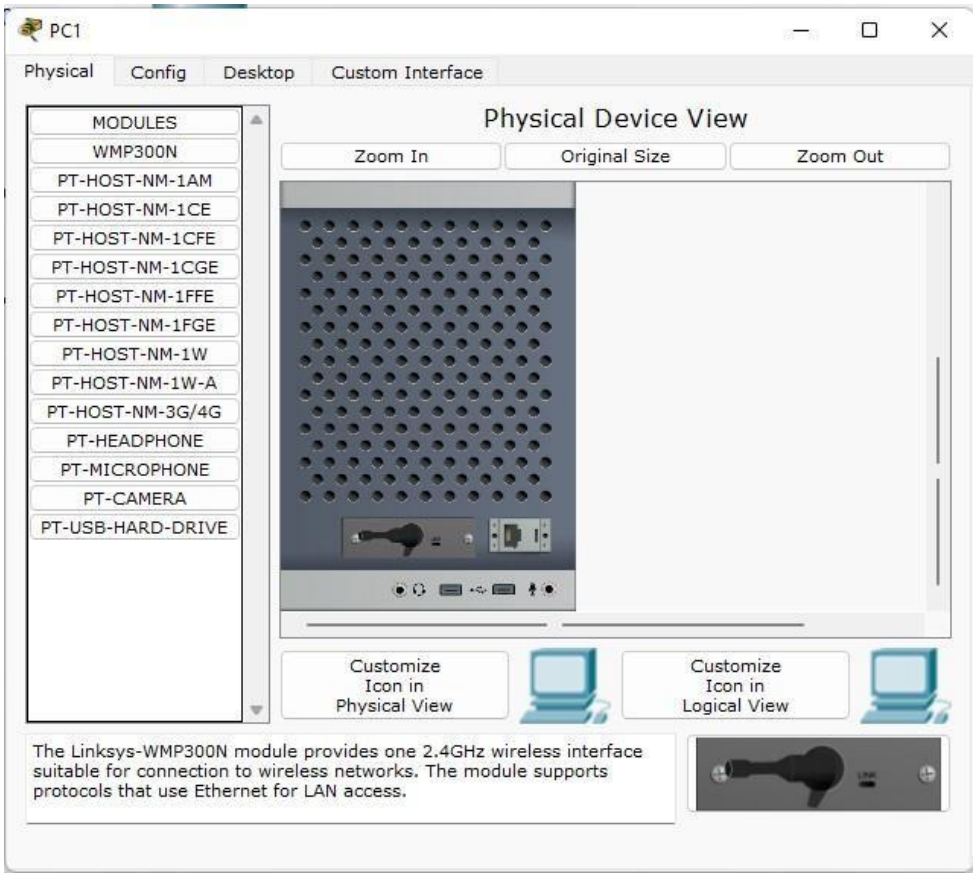
Packets sent: 4, Received: 4 Lost = 0 (0% loss)

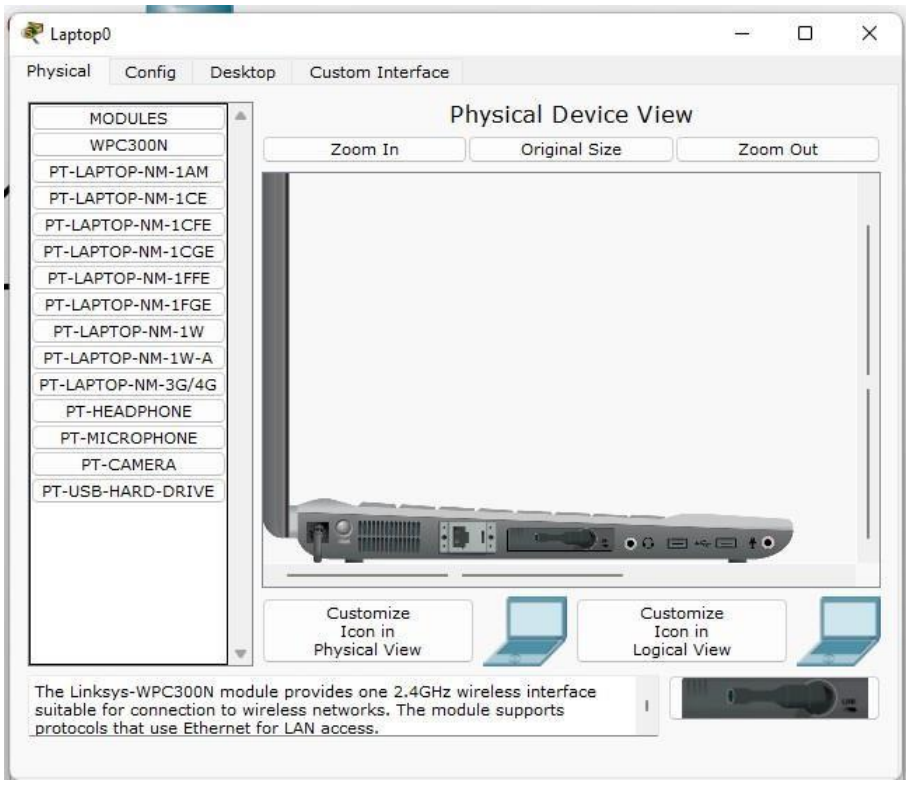
Approximate round trip times in milliseconds

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

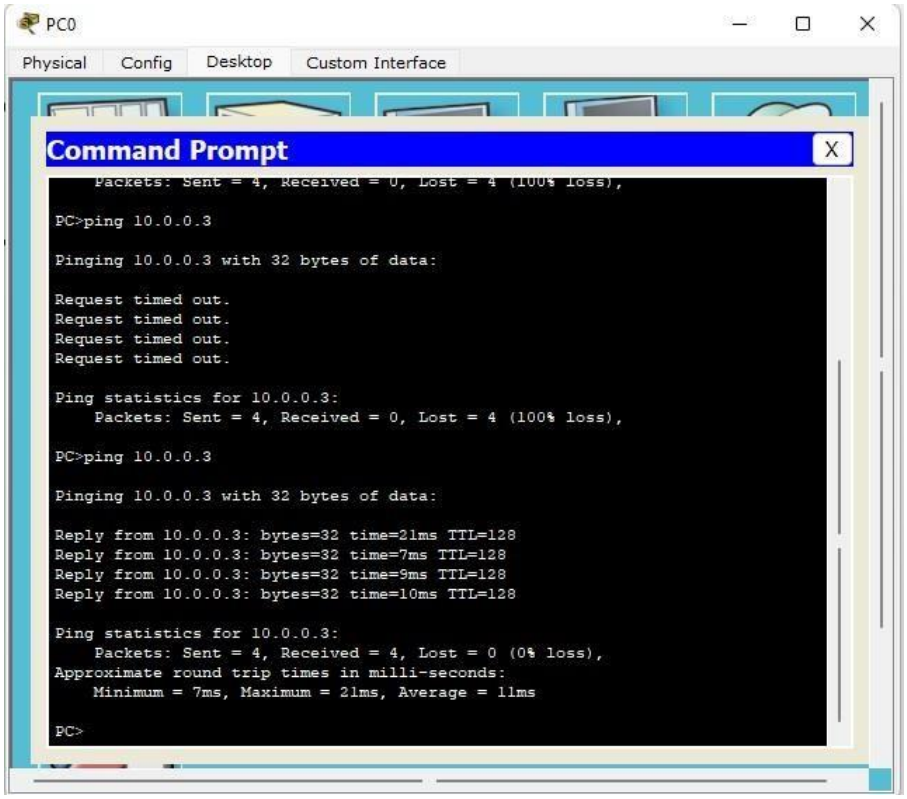
TOPOLOGY:


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.

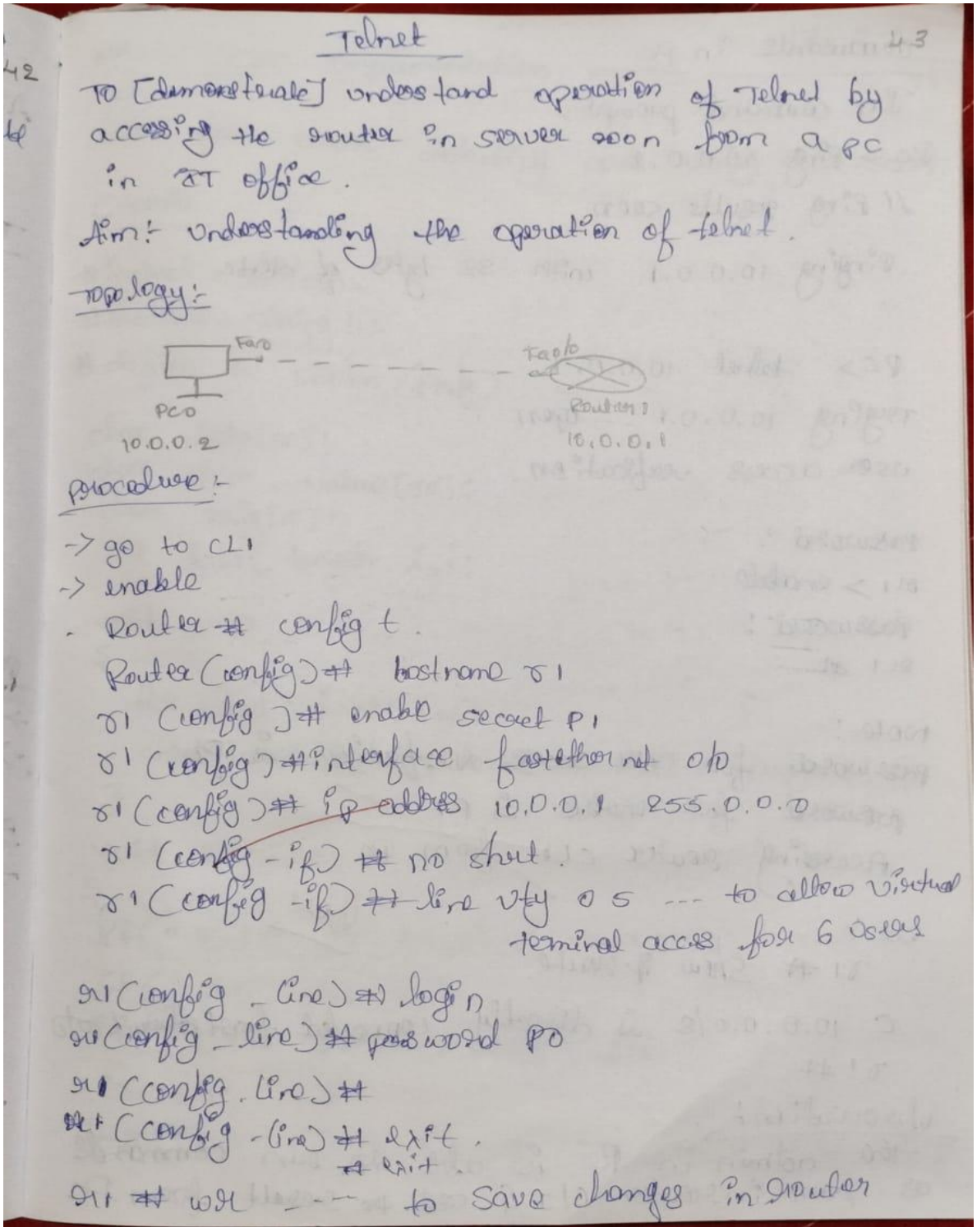
OUTPUT:


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:



Telnet

To [demonstrate] understand operation of Telnet by accessing the router in server room from a PC in IT office.

Aim:- Understanding the operation of telnet.

Topology:-

Procedure:-

-> go to CLI
-> enable
- Router# config t.
Router(config)# hostname r1
r1(config)# enable secret p1
r1(config)# interface fastethernet 0/0
r1(config)# ip address 10.0.0.1 255.0.0.0
r1(config-if)# no shut.
r1(config-if)# line vty 0 5 --- to allow virtual terminal access for 6 users
r1(config-line)# login
r1(config-line)# password p0
r1(config-line)#
r1(config-line)# exit.
r1# exit
r1# wr --- to save changes in router

commands in PC

In command prompt;

PC > Ping 10.0.0.1

// Ping results seen

Pinging 10.0.0.1 with 32 bytes of data

PC> telnet 10.0.0.1

Trying 10.0.0.1 ... open

User access verification

Password:

R1 > enable

Password:

R1 #

Note:

Password for user access verification is P0

Password for enable is P1

Accessing router CLI from PC

ex:

R1 # show ip route

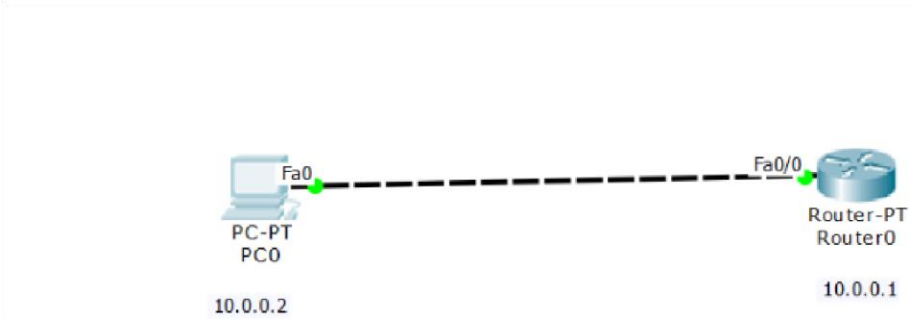
C 10.0.0.0/8 is directly connected fastethernet 0/0

R1 #

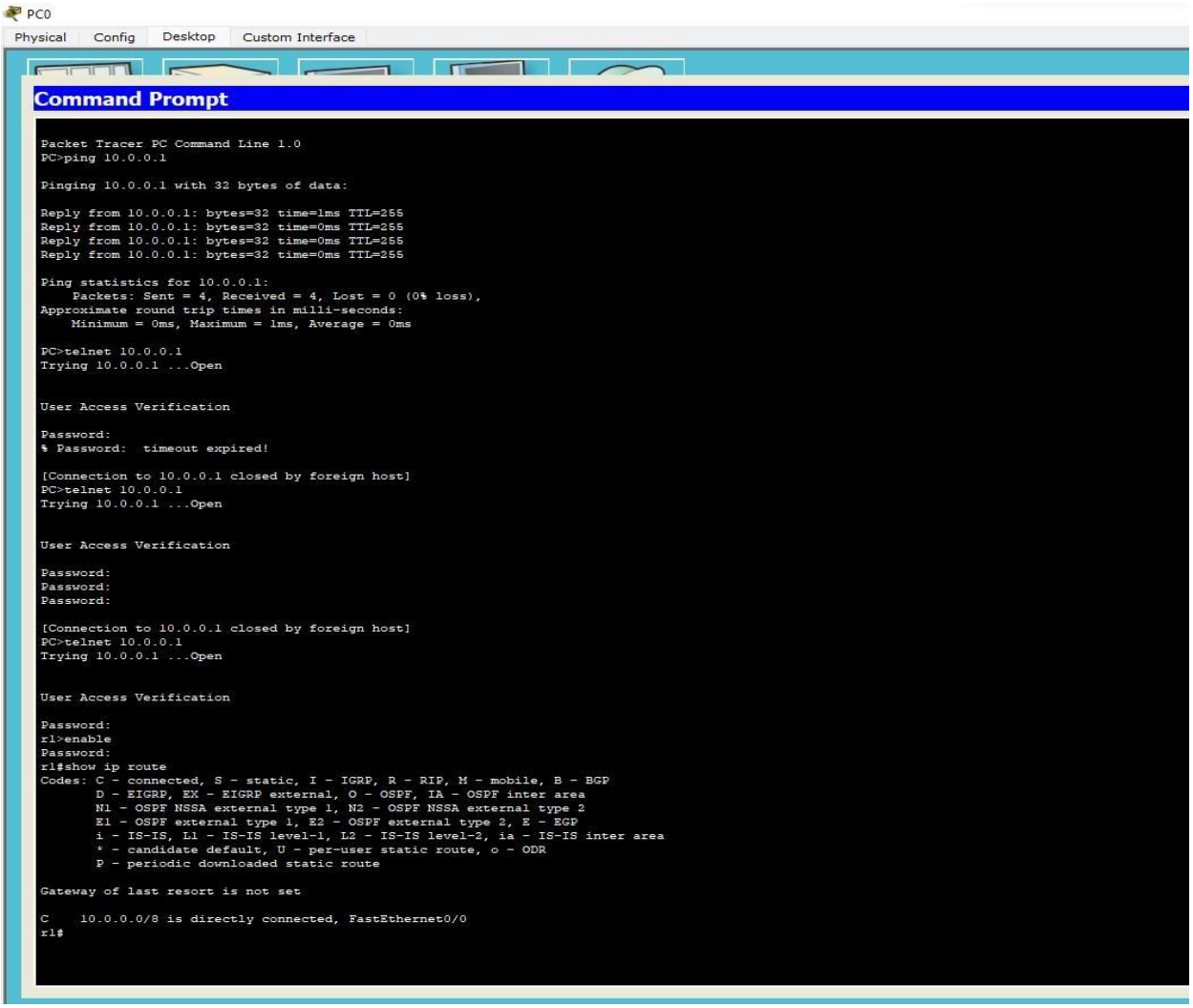
observation:

the admin in PC is able to run commands as run in router CLI & see the result from PC

TOPOLOGY:



OUTPUT:



LAB 13

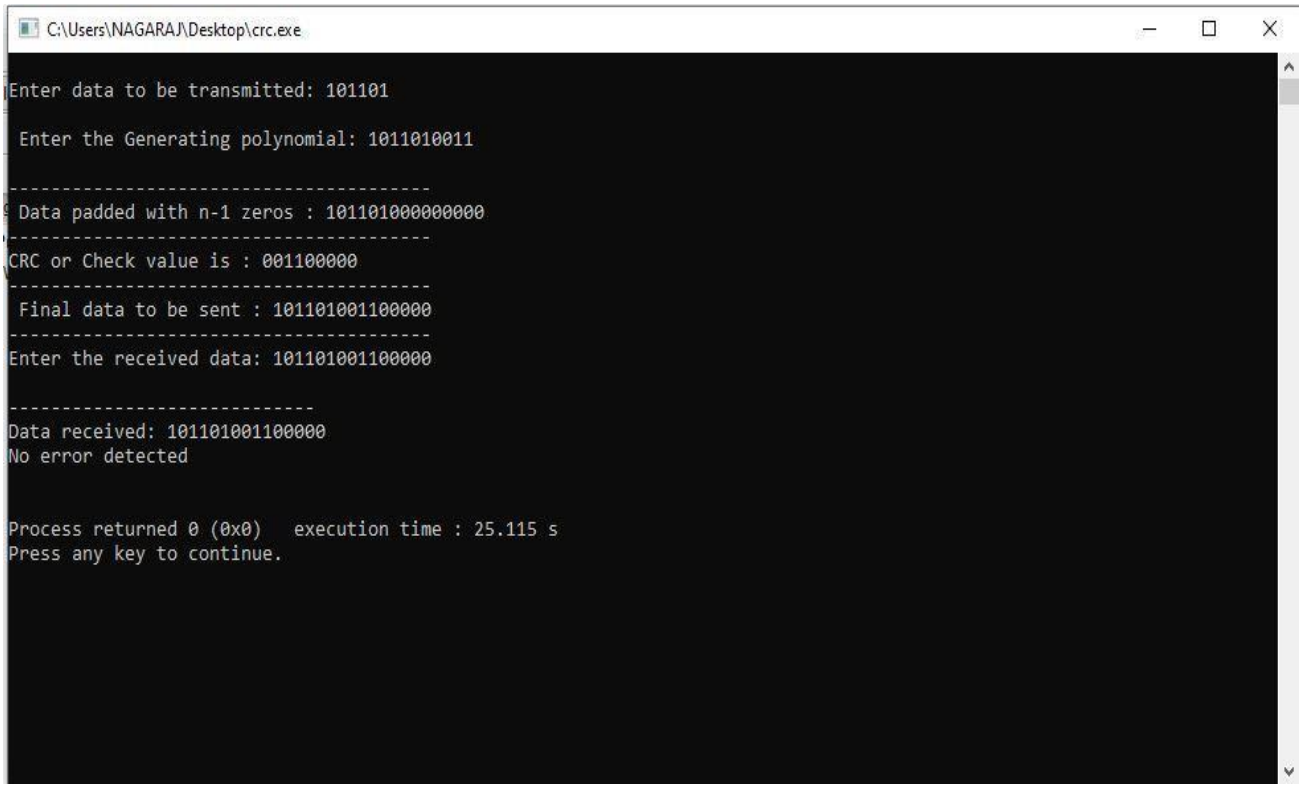
Write a program for error detecting code using CRCCCITT (16-bits).

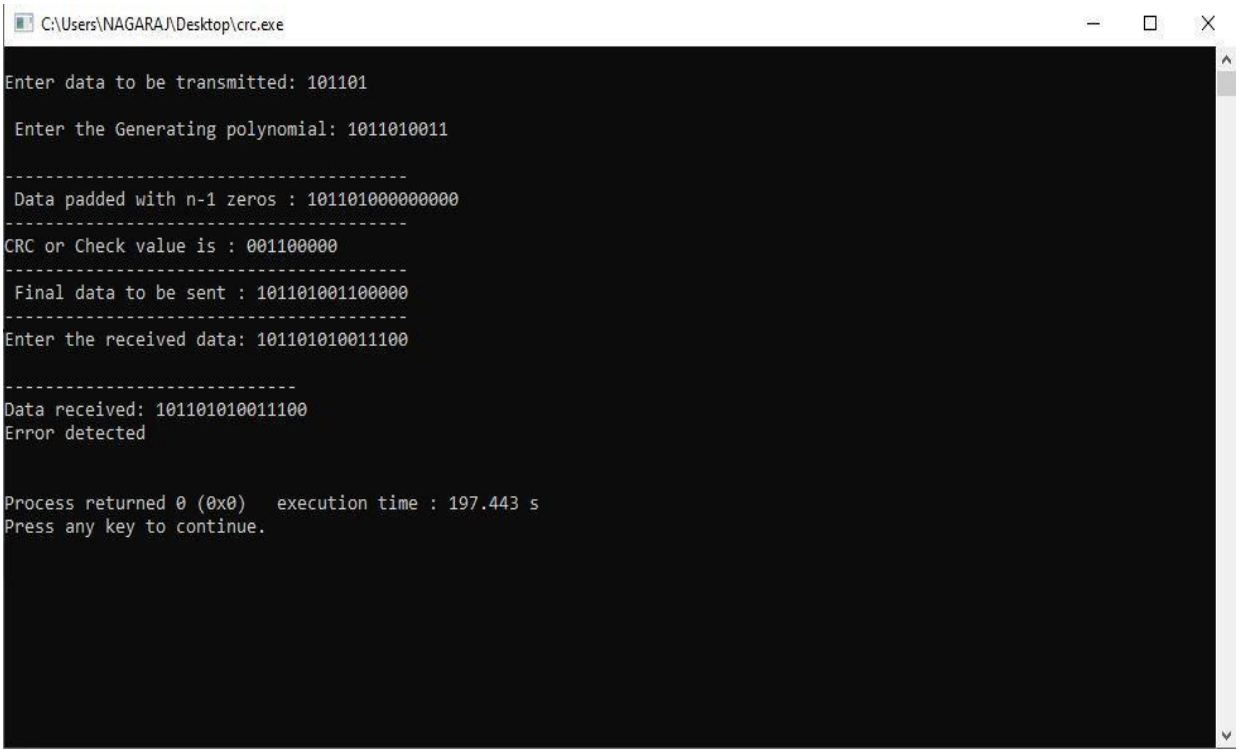
Code:

```
#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.

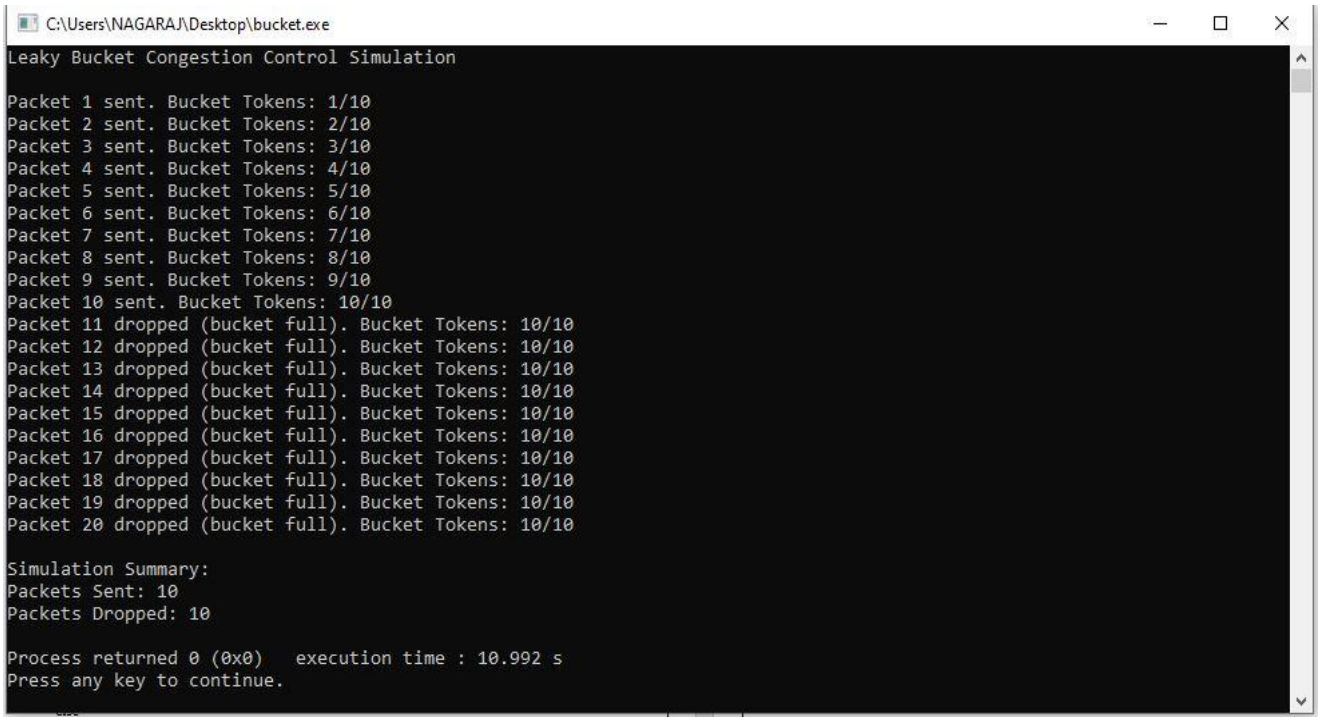

C:\Users\NAGARAJ\Desktop\crc.exe
Enter data to be transmitted: 101101
Enter the Generating polynomial: 1011010011
Data padded with n-1 zeros : 101101000000000
CRC or Check value is : 001100000
Final data to be sent : 101101001100000
Enter the received data: 101101010011100
Data received: 101101010011100
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 :

17-8-23

CRC Implementation

WAP for error detecting code using CRC-CCITT

C-code

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

char data[30];
char check_value[30];
char poly[10];
int data_length, i, j;

void XOR
{
  for (j=1; j<N; j++)
  check_value[j] = ((check_value[j] == poly[j] ?
                                    '0' : '1');
}
void receiver()
{
Pf("Enter the received data: ");
Sf("%s", data);
Pf("Data received: %s", data);
CRC();
for(i=0; (i<N-1) && (check_value[i] != '1'); i++)
```

```
if (N-1)
    pf(" Error detected \n");
  else
    pf(" No error detected \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[j+i];
      }
   while(i<=data_length + N+i);
}

int main()
{
 Printf(" Enter data to be transmitted ");
 Scanf("%s", data);
 Printf(" Enter divisor polynomial: ");
 Scanf(" %s", Poly);
 data_length = Strlen(data);
```

```
for (i = data_length ; i < data_length + N-1 ; i++)
    data[i] = '0';
Pf ("Data padded with n-1 zeroes : %s", data);
crc();
Pf ("CRC value is %s", check_value);
for(i = data_length ; i < data_length + N-1 ; i++)
    data[i] = check_value[i - data_length];
Pf ("Final datawords to be sent : %s", data);
receiver();
return 0;
}
```

Output :-

Enter data to be transmitted : 101010
Enter the divisor polynomial : 101
Data padded with n-1 zeroes : 101010.0000
CRC value is : 001
Final codeword to be sent : 101 01000

Enter the received data : 10001000
Error detected

Enter data to be transmitted : 101100
Enter divisor poly : 1001
Data padded with n-1 zeroes : 101100000
CRC value is : 001
Final codeword to be sent : 101100001
Enter received data : 101100000
NO error detected

WAP for congestion control using Leaky Bucket algm:

C-code:-

```
#include<stdio.h>
int main()
{
int incoming, outgoing, buck_size, n, store=0;
Pf("Enter bucket size");
Sf("%d", &buck_size);
Pf("Enter outgoing size");
Sf("%d", &outgoing);
Pf("Enter no of inputs:");
Sf("%d", &n);
while(n!=0)
{
Pf("Enter the incoming bucket size:");
Sf("%d", &incoming);
if(incoming<=(buck_size-store))
{
   store+=incoming;
   Pf("Bucket buffer size %d out of %d",
                  store, buck_size);
}
else
```

```
{
Pf("Dropped %d no of packet\n", incoming -
                        (bock_size - store));
Pf("Bucket buffer size %d out of %d in", store,
store = bock_size;                    bock_size);
}
store = store - outgoing;
Pf("After outgoing %d packets left out of
    %d in buffer\n", store, bock_size);
    n--;
  }
}
```

Output:-

Enter bucket size: 5000
Enter outgoing rate: 2000
Enter no 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.

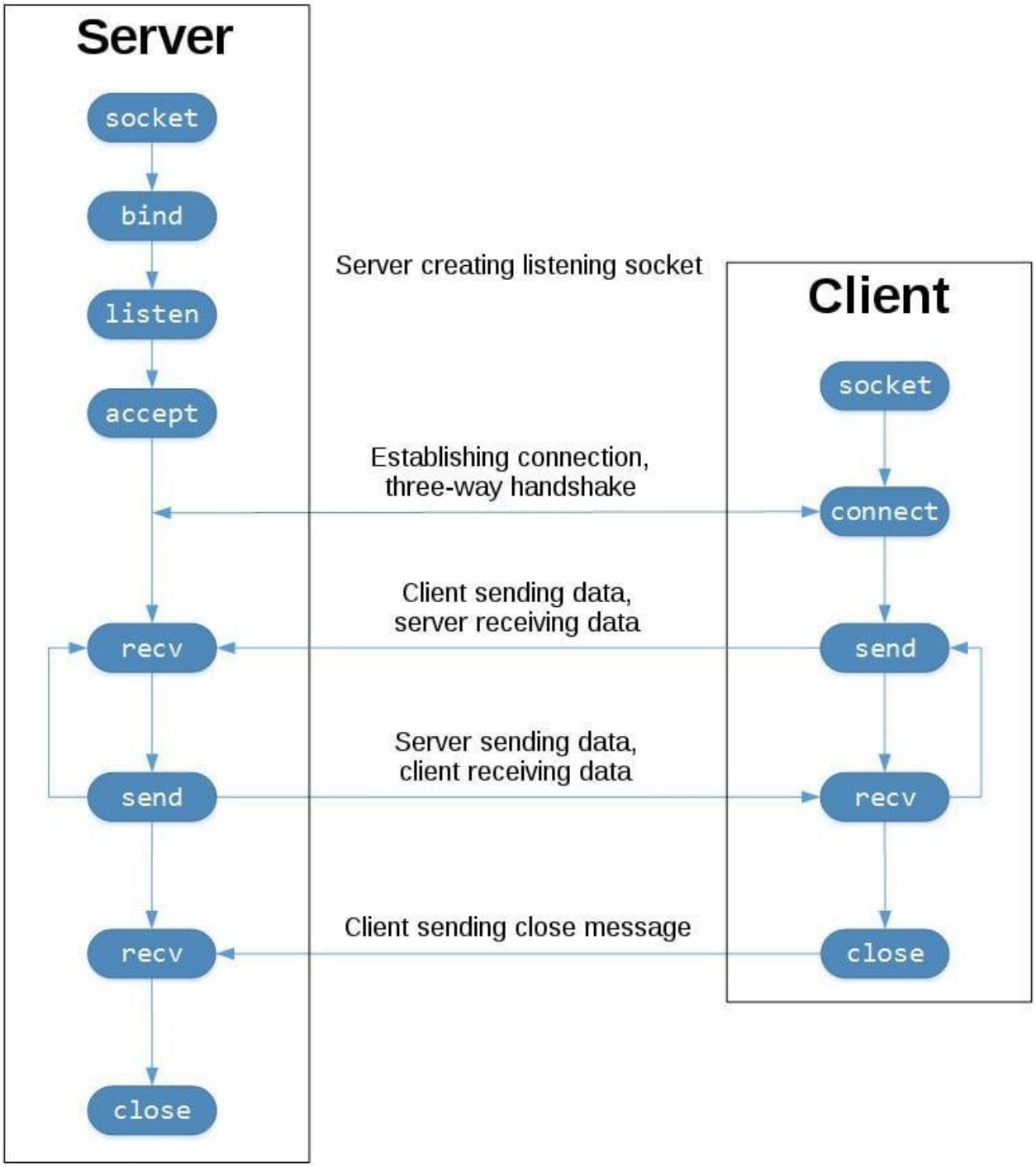
CYCLE 3

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:

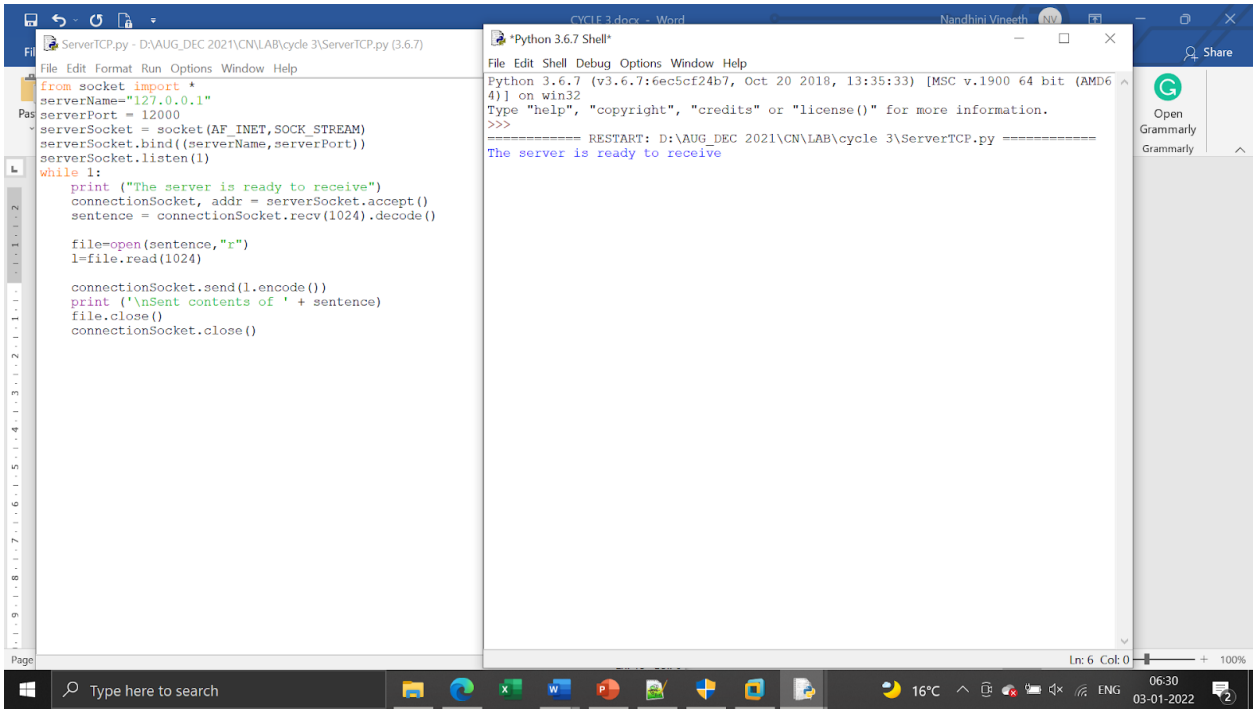
ClientTCP.py

```
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()
```

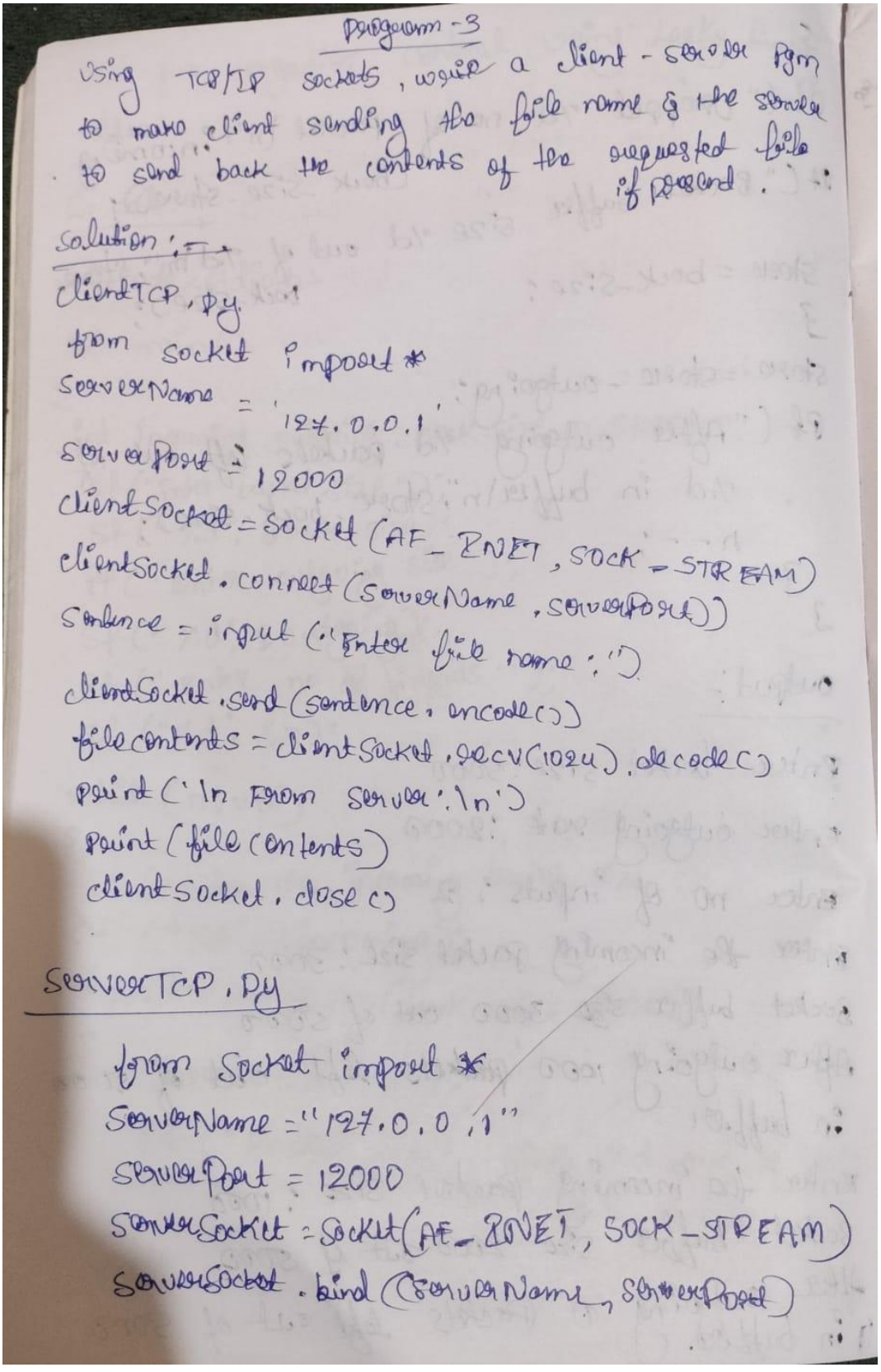
ServerTCP.py

```
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 :



Program - 3

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

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("Enter file name :")
clientSocket.send(sentence.encode())
fileContents = clientSocket.recv(1024).decode()
print('In From Server:\n')
print(fileContents)
clientSocket.close()
```

serverTCP.py

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

Output:-

First run serverUDP.py then run clientUDP.py

Output on serverUDP.py

the server is ready to receive

Send contents of serverUDP.py

Output on clientUDP.py

Enter file name: serverUDP.py

Reply from server:

contents of serverUDP.py is displayed here same as serverUDP.py file.

28/9

KLD
1/9/2023

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()
```

9) Using UDP sockets, write a client-server program to make client sending the file name to 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("\n Enter file name: ")
clientSocket.sendto(bytes(sentence, "utf-8"), (serverName, serverPort))
fileContents, serverAddress = clientSocket.recvfrom(2048)
print("\n Reply from server: \n")
print(fileContents.decode("utf-8"))
# for i in fileContents:
# print(str(i), end="")
clientSocket.close()
clientSocket.close()
```

```
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('\n Sent contents of ' + sentence)
file.close()
connectionSocket.close()
```

output :-

First Run serverTCP.py then run clientTC.py

output on serverTCP

the server is ready to receive

sent contents of serverTCP.py

the server is ready to receive

output on clientTCP terminal.

Enter file name : serverTCP.py

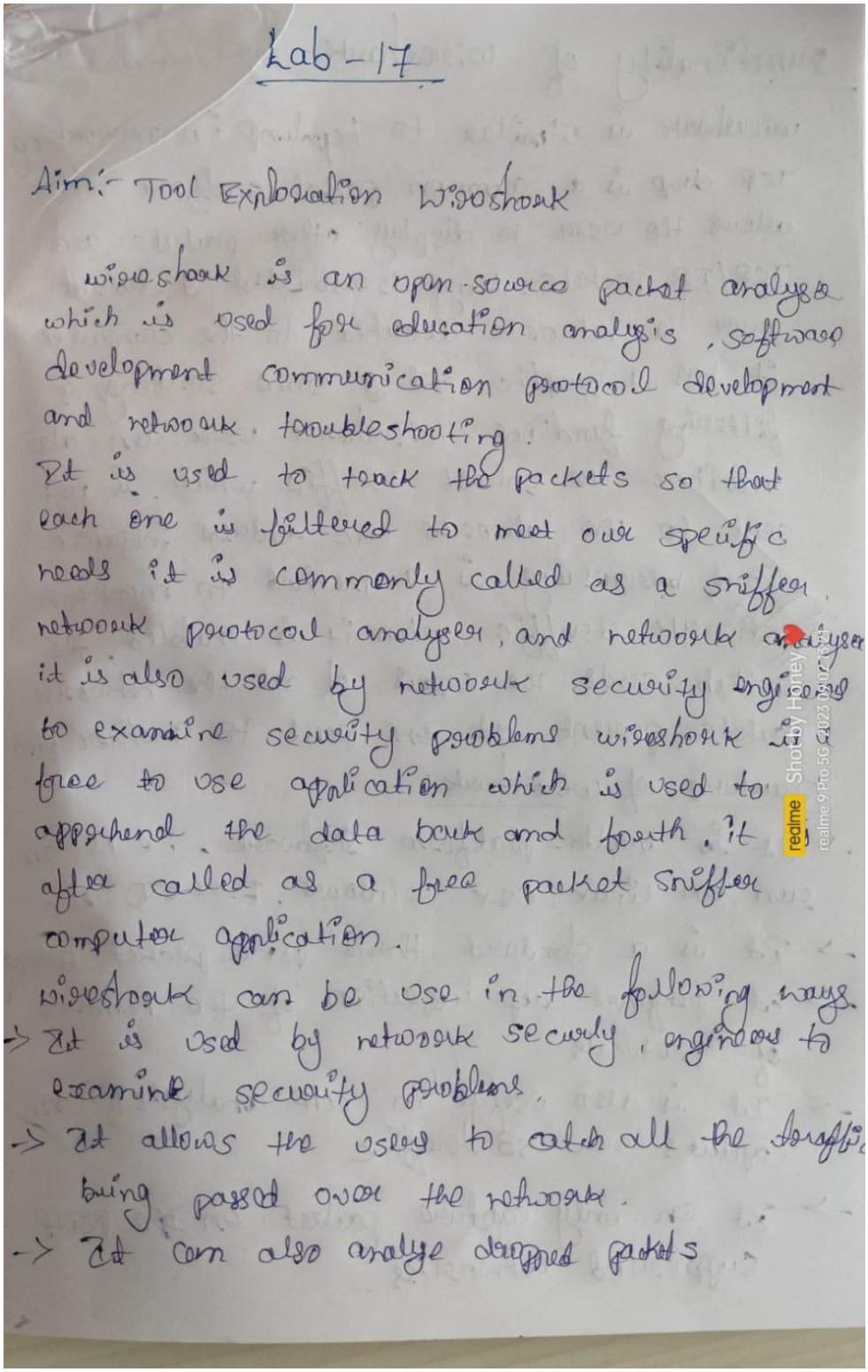
From Server :

contents in serverTCP.py is displayed here

LAB 15

Tool Exploration -Wireshark.

OBSERVATION :



Lab - 17

Aim:- Tool Exploration Wireshark

Wireshark is an open-source packet analyser which is used for education analysis, software development communication protocol development and network troubleshooting.
It is used to track the packets so that each one is filtered to meet our specific needs it is commonly called as a sniffer network protocol analyser, and network analyser it is also used by network security engineers to examine security problems wireshark is free to use application which is used to apprehend the data back and forth. It is often 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 catch all the traffic being passed over the network.
-> It can also analyse dropped packets.

Functionality of wireshark:-

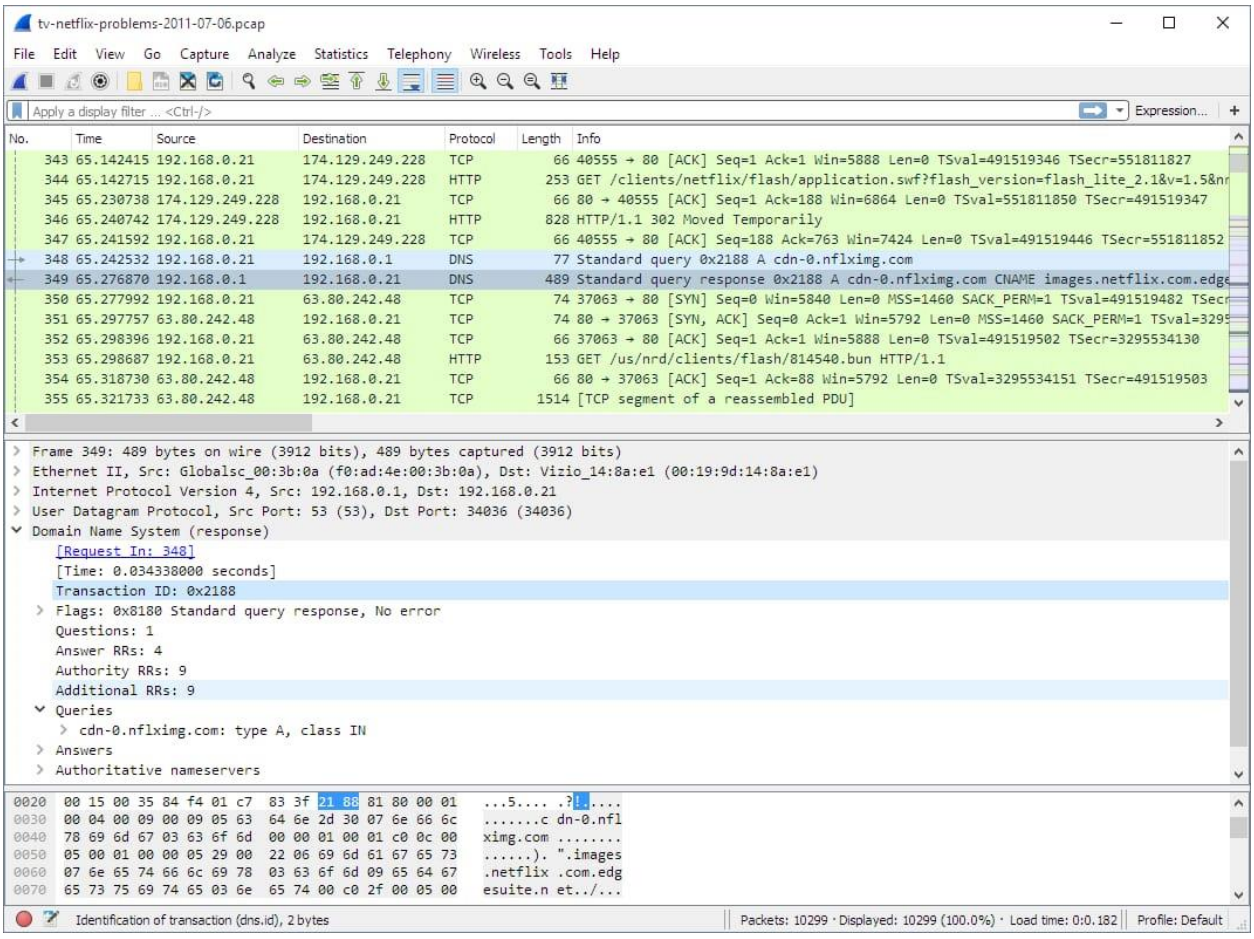
wireshark is similar to tcpdump in networking TCP dump is a common packet analyzer which allows the user to display other packets and TCP/IP packets, being transmitted & received over a network attached to the computer it has a graphic end & some sorting & filtering functions wireshark user can also monitor the unicast traffic which is not sent to the networks MAC address interface.

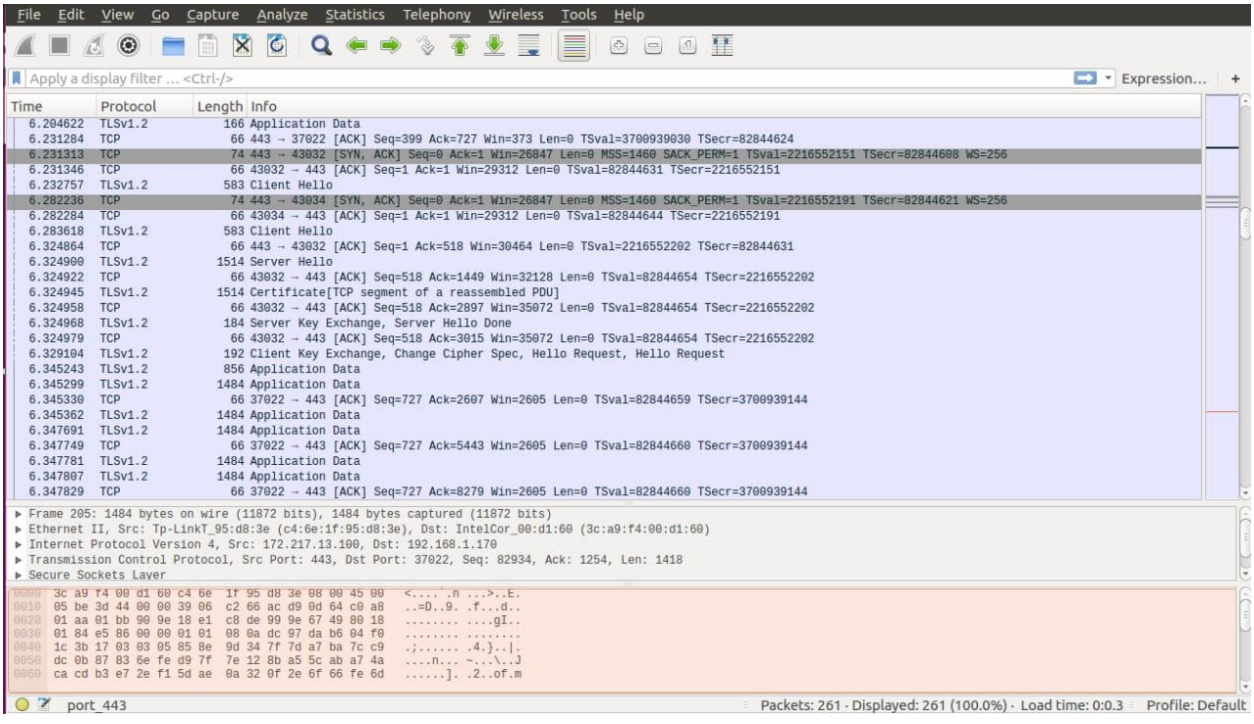
Port mirroring is a method to monitor network traffic when it is enabled. the switch sends the copies of all the network packets present at one port to another port

Features of wireshark:

-> It is multi-platform software, i.e it can run on linux, OSX windows, FreeBSD etc
-> It is a standard three pane packet browser
-> It performs deep inspection of the hundreds of protocals.
-> It is also useful in VOIP analysis & can captures raw USB traffic
-> It can only capture packet on the PCAP supported networks.

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