VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT on

COMPUTER NETWORKS LAB

Submitted by

PRAVIJ GUPTA (1BM21CS141)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019 JUN-2023 to SEP-2023
B. M. S. College of Engineering,
Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "COMPUTER NETWORKS LAB" carried out by **PRAVIJ GUPTA (1BM21CS141),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2023. The Lab report has been approved as it satisfiesthe academic requirements in respect of a **COMPUTER NETWORKS - (22CS4PCCON)** work prescribed for the said degree.

Prof. Swathi Sridharan

Assistant Professor
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak

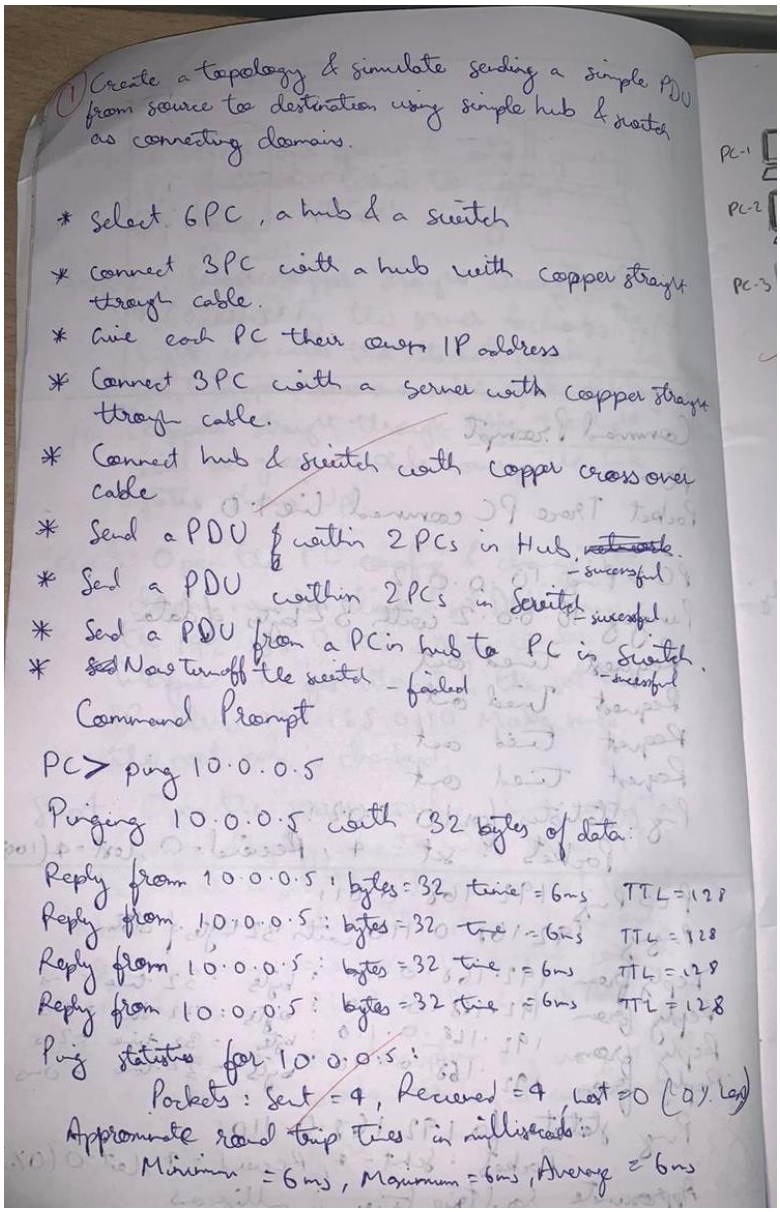
Professor and Head
Department of CSE
BMSCE, Bengaluru

Index

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:

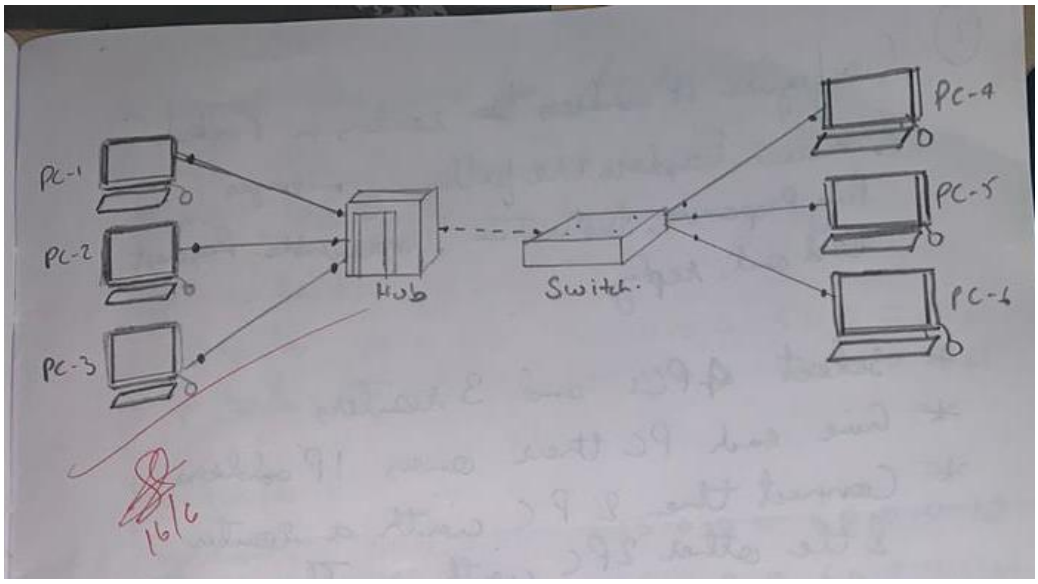


1) Create a topology & simulate sending a simple PDU from source to destination using simple hub & switch as connecting domains.

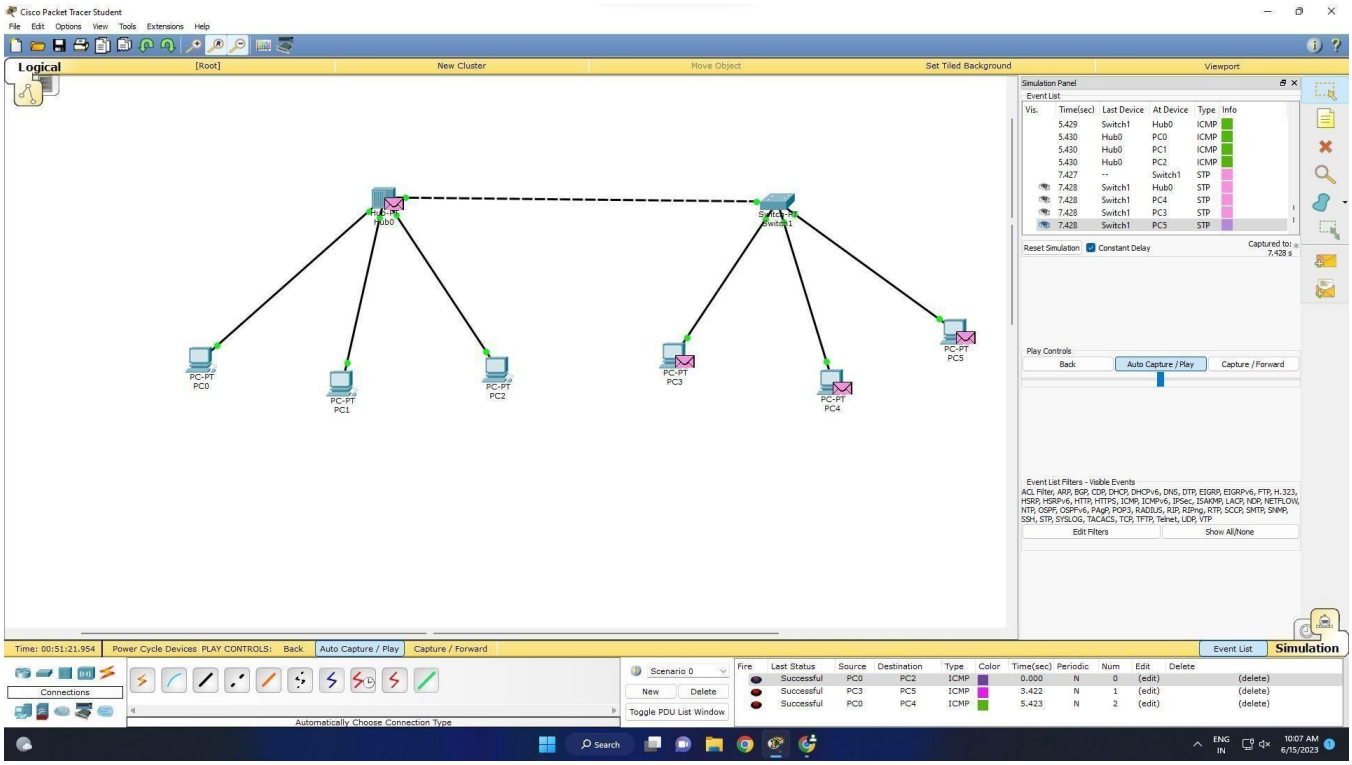
* Select 6PC, a hub & a switch
* Connect 3PC with a hub with copper straight through cable.
* Give each PC their own IP address
* Connect 3PC with a server with copper straight through cable.
* Connect hub & switch with copper cross over cable
* Send a PDU within 2PCs in Hub. - sucessful
* Send a PDU within 2PCs in Switch - sucessful
* Send a PDU from a PC in hub to PC in Switch. - sucessful
* Now turnoff the switch - failed

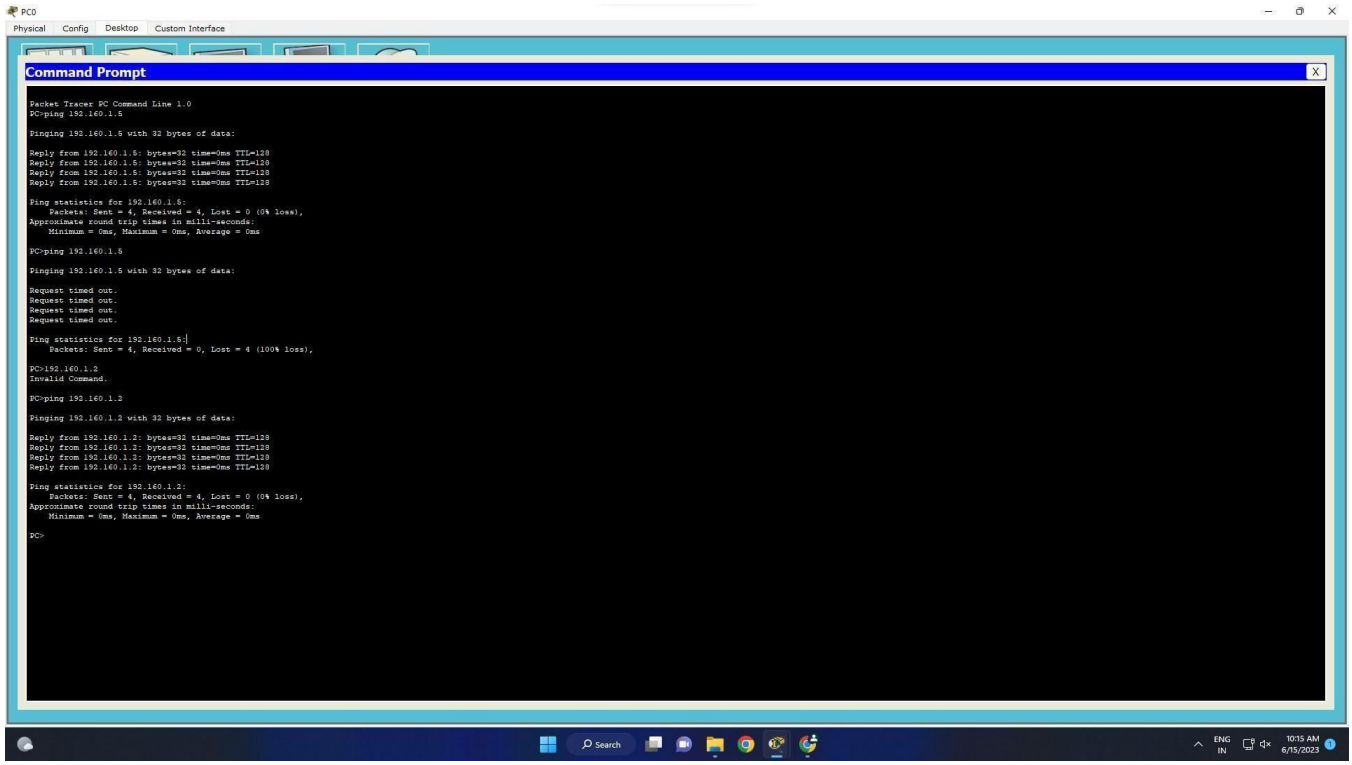
Command Prompt

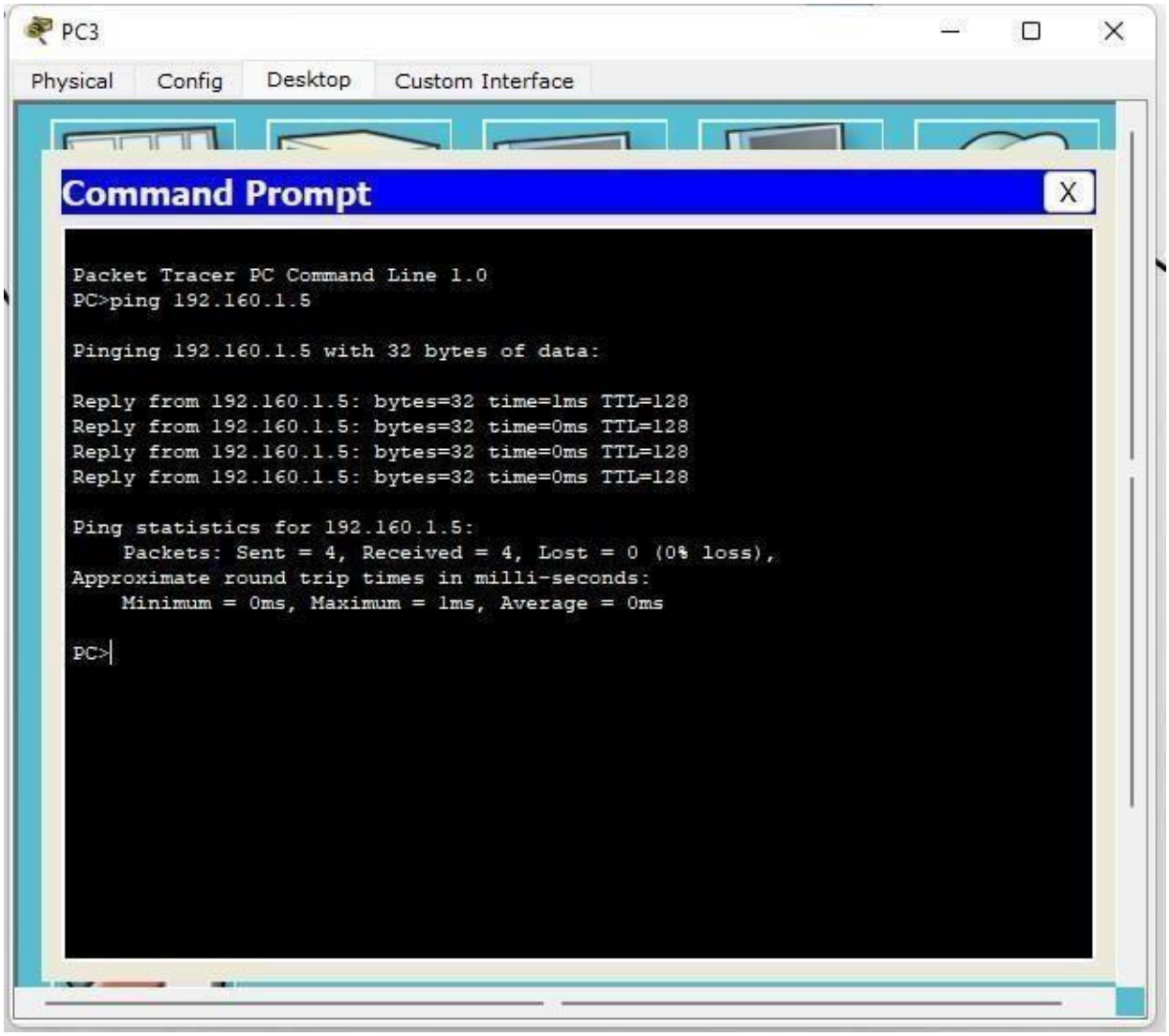
```
PC> ping 10.0.0.5
Pinging 10.0.0.5 with 32 bytes of data:
Reply from 10.0.0.5: bytes=32 time=6ms TTL=128
Reply from 10.0.0.5: bytes=32 time=6ms TTL=128
Reply from 10.0.0.5: bytes=32 time=6ms TTL=128
Reply from 10.0.0.5: bytes=32 time=6ms TTL=128
Ping statistics for 10.0.0.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)
Approximate round trip times in milliseconds:
    Minimum = 6ms, Maximum = 6ms, Average = 6ms
```


PC-1
PC-2
PC-3
Hub
Switch
PC-4
PC-5
PC-6
16/6

OUTPUT:


Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Hub-PT
Hub0
Switch-PT
Switch1
PC-PT
PC0
PC1
PC2
PC3
PC4
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)
ENG IN
10:07 AM 6/15/2023

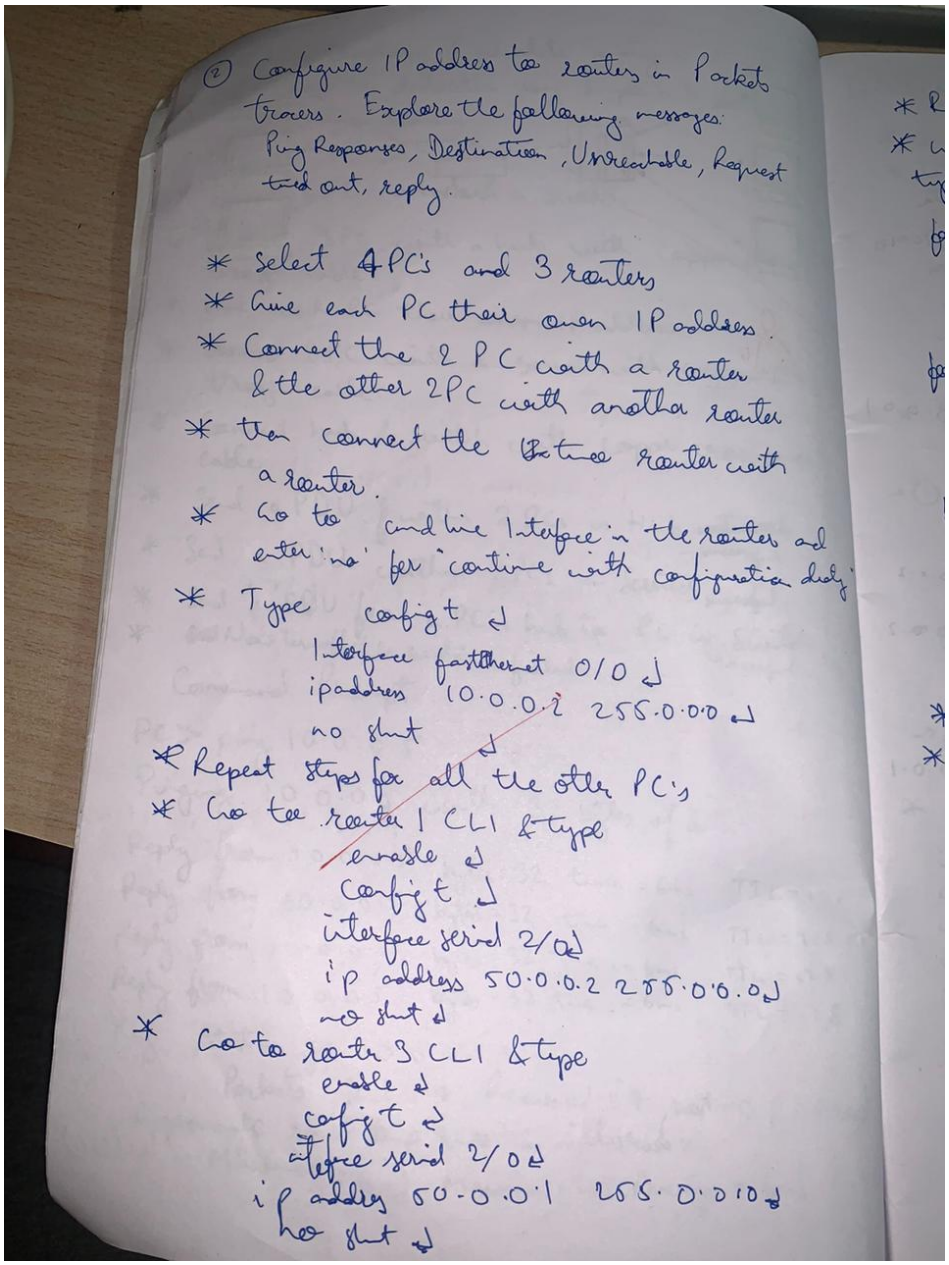

PC0
Physical Config Desktop Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 192.160.1.5
Pinging 192.160.1.5 with 32 bytes of data:
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
Ping statistics for 192.160.1.5:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>ping 192.160.1.5
Pinging 192.160.1.5 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 192.160.1.5:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>192.160.1.2
Invalid Command.
PC>ping 192.160.1.2
Pinging 192.160.1.2 with 32 bytes of data:
Reply from 192.160.1.2: bytes=32 time=0ms TTL=128
Reply from 192.160.1.2: bytes=32 time=0ms TTL=128
Reply from 192.160.1.2: bytes=32 time=0ms TTL=128
Reply from 192.160.1.2: bytes=32 time=0ms TTL=128
Ping statistics for 192.160.1.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>
ENG IN
10:15 AM 6/15/2023


PC3
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 192.160.1.5
Pinging 192.160.1.5 with 32 bytes of data:
Reply from 192.160.1.5: bytes=32 time=1ms TTL=128
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
Ping statistics for 192.160.1.5:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
PC>

LAB 2

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

OBSERVATION:



* Repeat the same for Router 2 & 3.
* We statically connect routers to the PCs by typing the following in CLI

for router 1:

```
ip route 30.0.0.0 255.0.0.0 50.0.0.1 ↵
ip route 40.0.0.0 255.0.0.0 40.0.0.1 ↵
```

for router 2:

```
ip route 10.0.0.0 255.0.0.0 60.0.0.1 ↵
ip route 20.0.0.0 255.0.0.0 60.0.0.1 ↵
```

for router 3:

```
ip route 10.0.0.0 255.0.0.0 50.0.0.1
ip route 20.0.0.0 255.0.0.0 50.0.0.1
ip route 30.0.0.0 255.0.0.0 60.0.0.1
ip route 40.0.0.0 255.0.0.0 60.0.0.1
```

* Now data transfer b/w PC's is sucessful
* Before statically connecting routers ping b/w PC's not directly connected was unsuccesful.

```
PC > Ping 30.0.0.1
  Pinging 30.0.0.1 with 32 bytes of data
  Reply from 20.0.0.2 : Destination host Unreachable
```

Ping statistics for 30.0.0.1

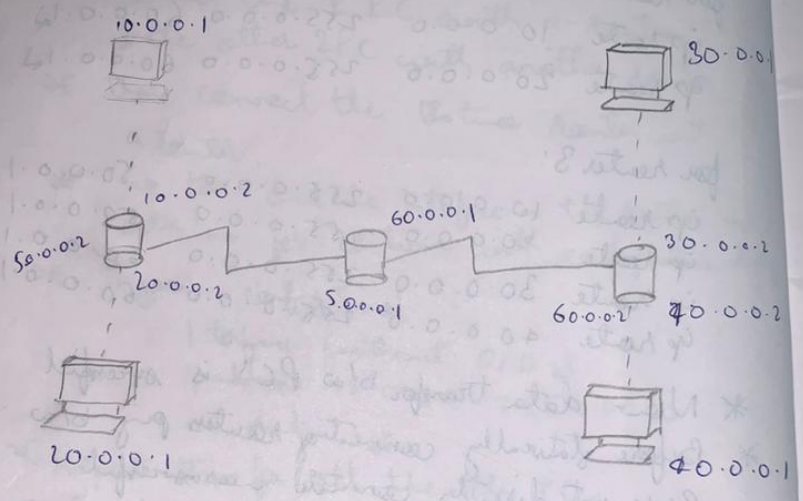
Packets : Sent = 4, Received = 0, Lost = 4 (100% loss)

After statically defining the route

PC> Ping 40.0.0.1

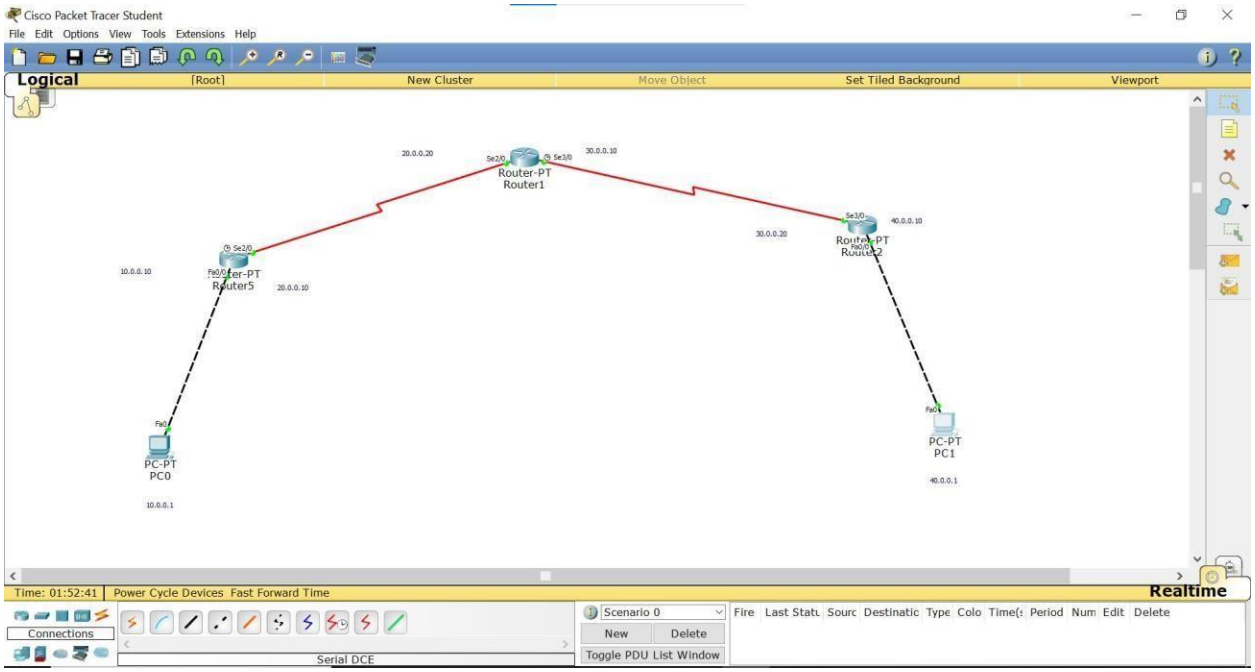
Pinging 40.0.0.1 with 32 bytes of data

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

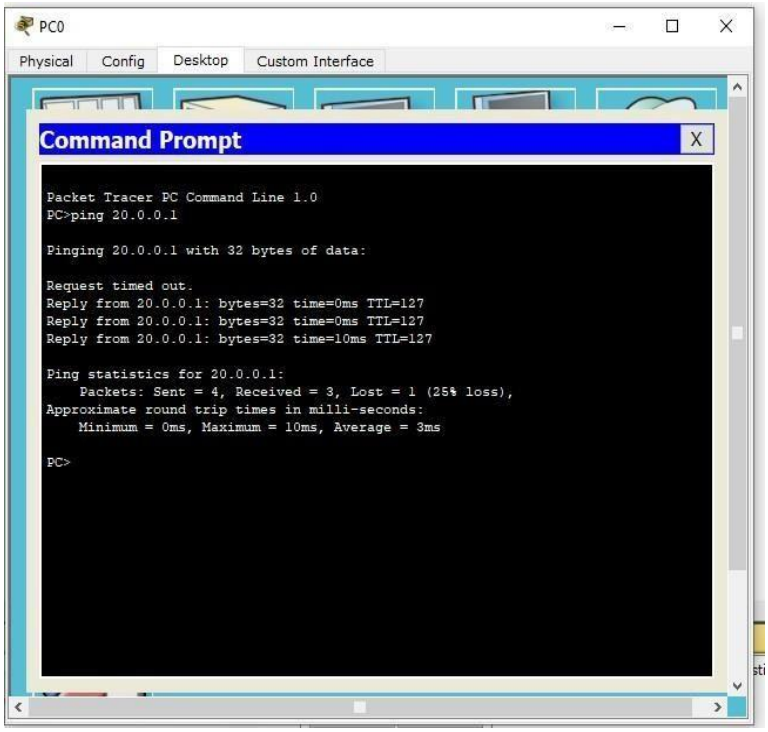


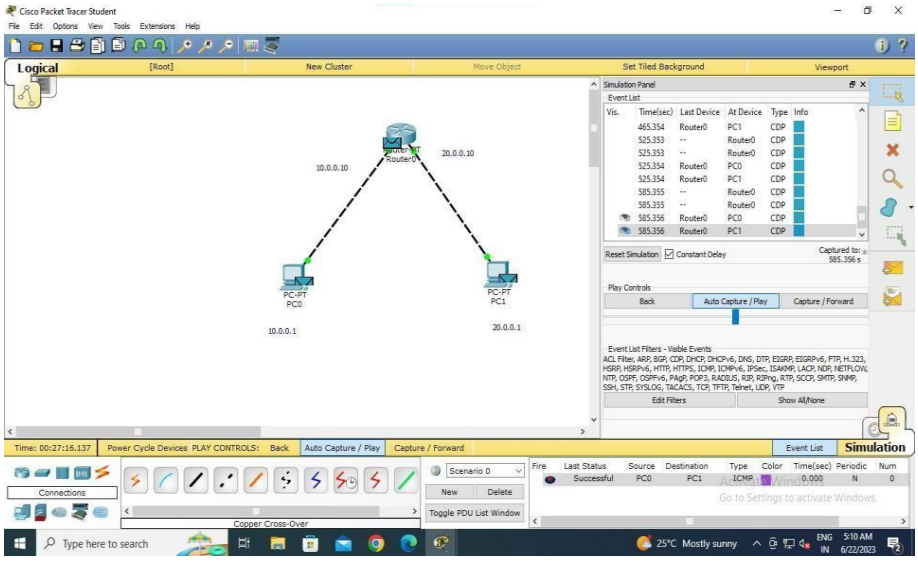
26/6/23

TOPOLOGY:


Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT
Router1
Router-PT
Router5
Router-PT
Router2
PC-PT
PC0
PC-PT
PC1
20.0.0.20
30.0.0.10
30.0.0.20
40.0.0.10
10.0.0.10
20.0.0.10
10.0.0.1
40.0.0.1
Time: 01:52:41
Power Cycle Devices Fast Forward Time
Realtime
Connections
Serial DCE
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Statu
Sourc
Destinatic
Type
Colo
Time(s
Period
Num
Edit
Delete

OUTPUT:

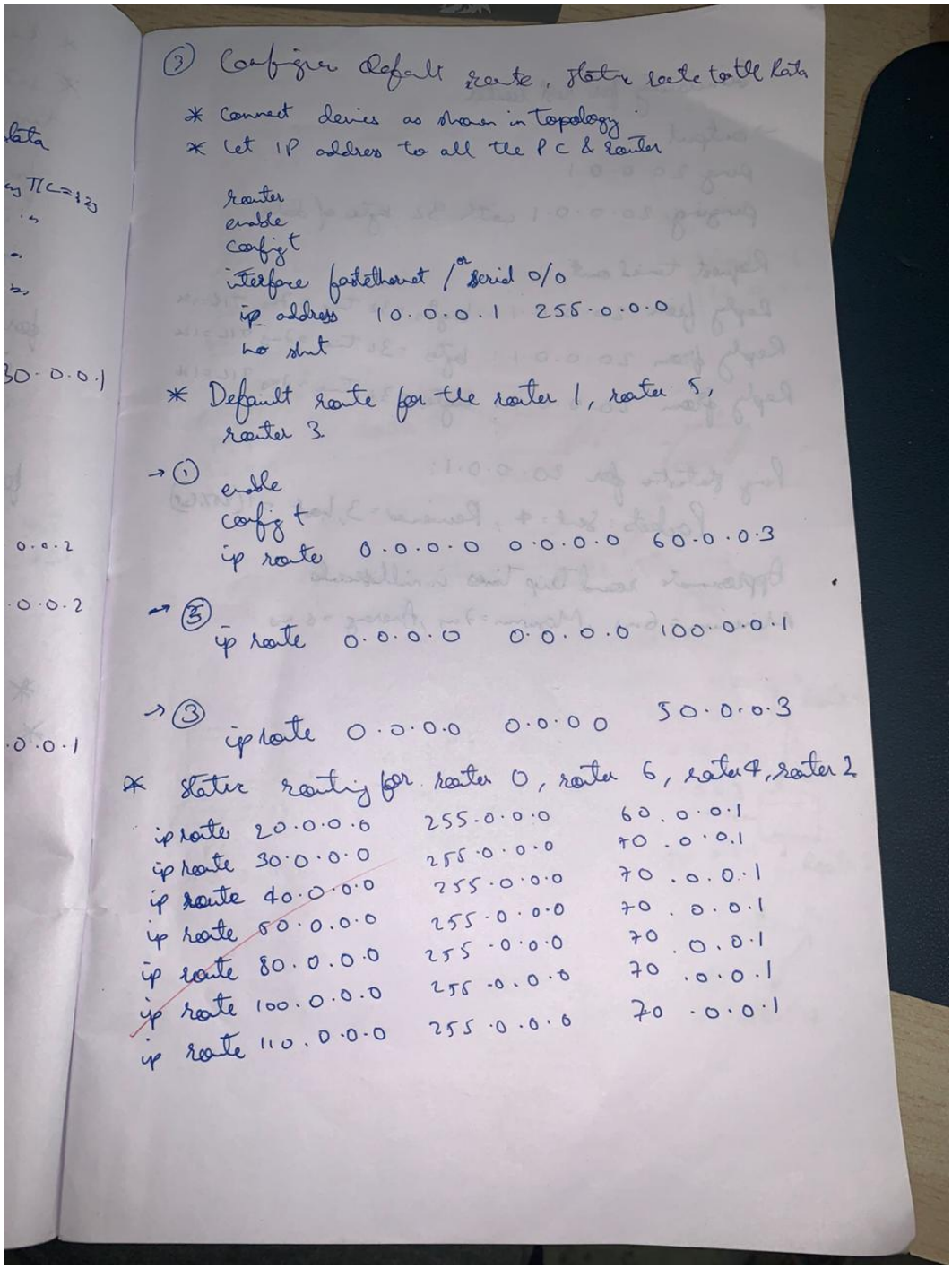

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Request timed out.
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=10ms 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
Logical
10.0.0.10
20.0.0.10
PC-PT PC0
PC-PT PC1
10.0.0.1
20.0.0.1
Simulation
Type here to search
25°C Mostly sunny

LAB 3

Configure default route, static route to the Router.

OBSERVATION:



③ Configure Default route, Static route to the Rata

* Connect devices as shown in topology.
* Let IP address to all the PC & router

router
enable
config t
interface fastethernet / or serial 0/0
ip address 10.0.0.1 255.0.0.0
no shut

* Default route for the router 1, router 5, router 3.

→ ① enable
config t
ip route 0.0.0.0 0.0.0.0 60.0.0.3

→ ⑤ ip route 0.0.0.0 0.0.0.0 100.0.0.1

→ ③ ip route 0.0.0.0 0.0.0.0 50.0.0.3

* Static routing for router 0, router 6, router 4, router 2

ip route 20.0.0.0 255.0.0.0 60.0.0.1
ip route 30.0.0.0 255.0.0.0 70.0.0.1
ip route 40.0.0.0 255.0.0.0 70.0.0.1
ip route 50.0.0.0 255.0.0.0 70.0.0.1
ip route 80.0.0.0 255.0.0.0 70.0.0.1
ip route 100.0.0.0 255.0.0.0 70.0.0.1
ip route 110.0.0.0 255.0.0.0 70.0.0.1

Similarly for rest router

→ output :-

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 tie = 7ms TTL = 12

Reply from 20.0.0.1 : bytes = 32 tie = 7ms TTL = 12

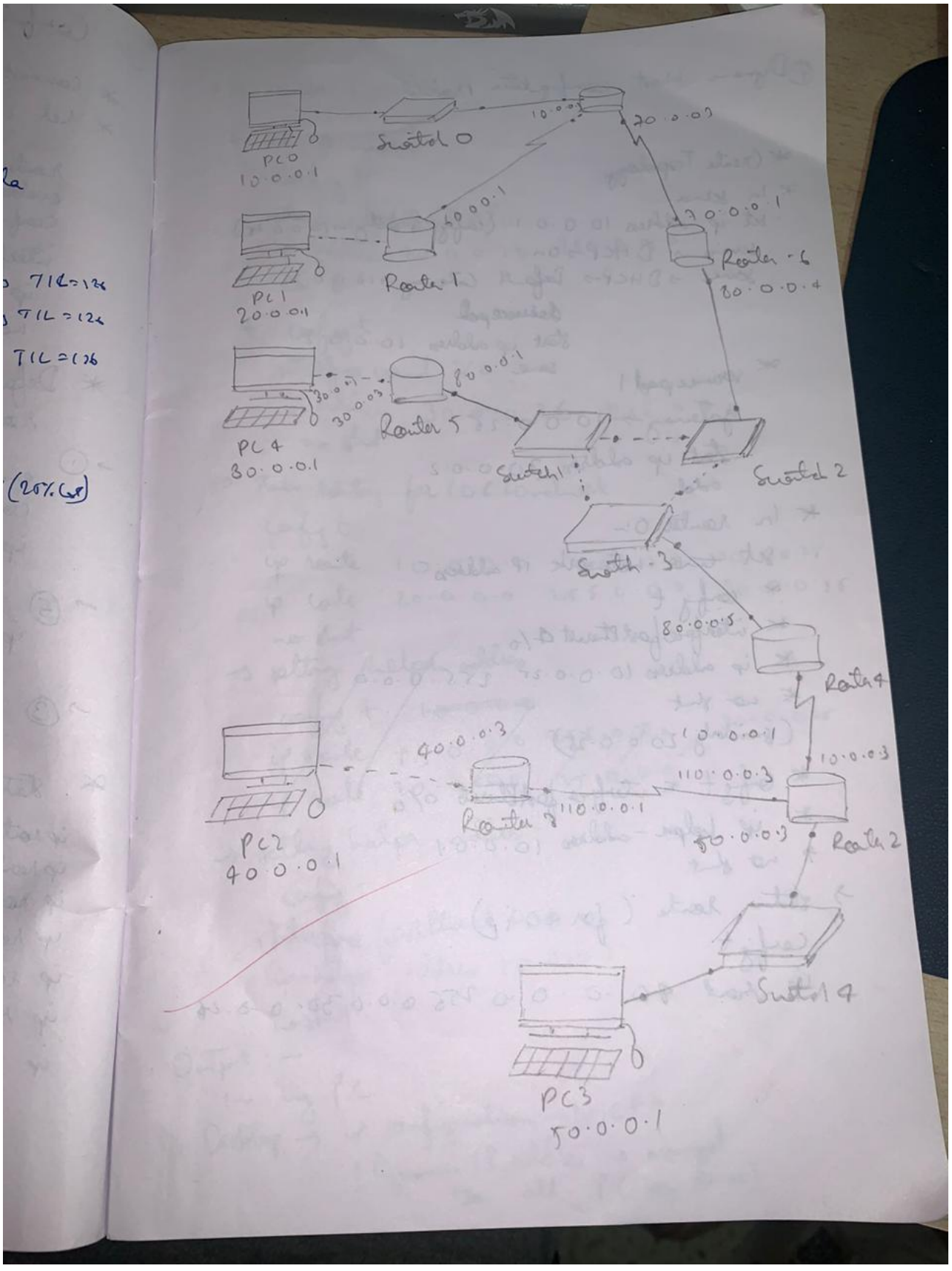
Reply from 20.0.0.1 : bytes = 32 Time = 7ms TTL = 12

Ping statistics for 20.0.0.1 :

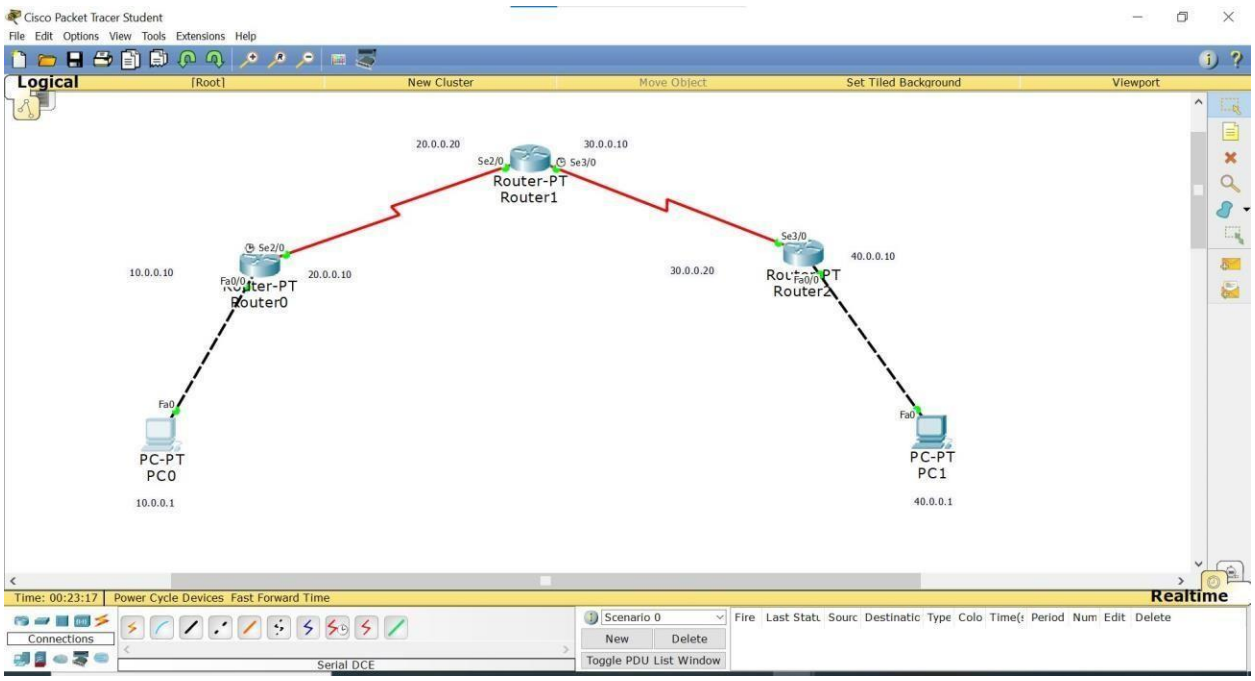
Packets : Sent = 4, Received = 3, Lost = 1 (25% loss)

Approximate round trip times in milliseconds

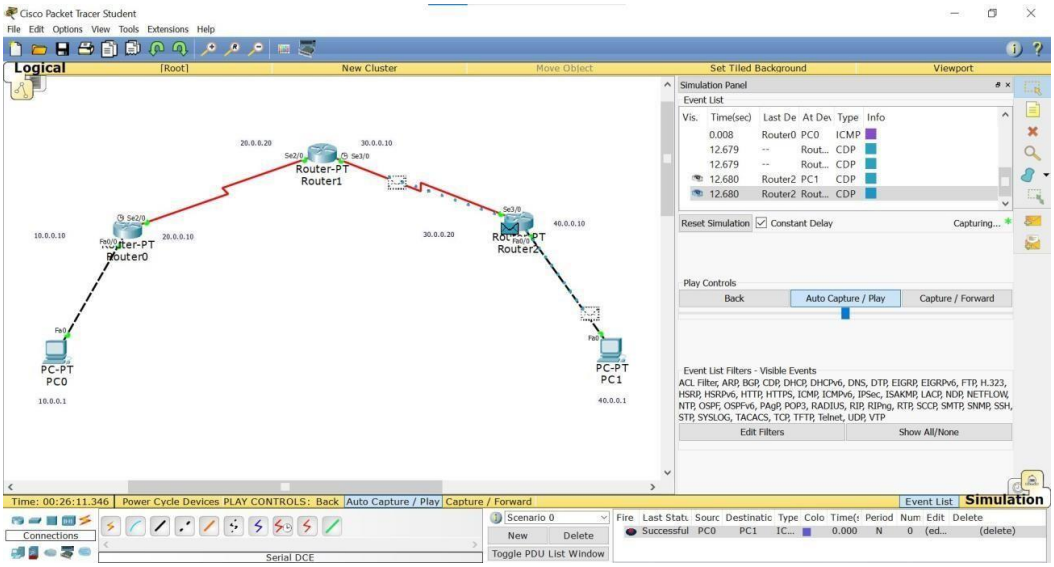
Minimum = 6ms, Maximum = 7ms, Average = 6 ms

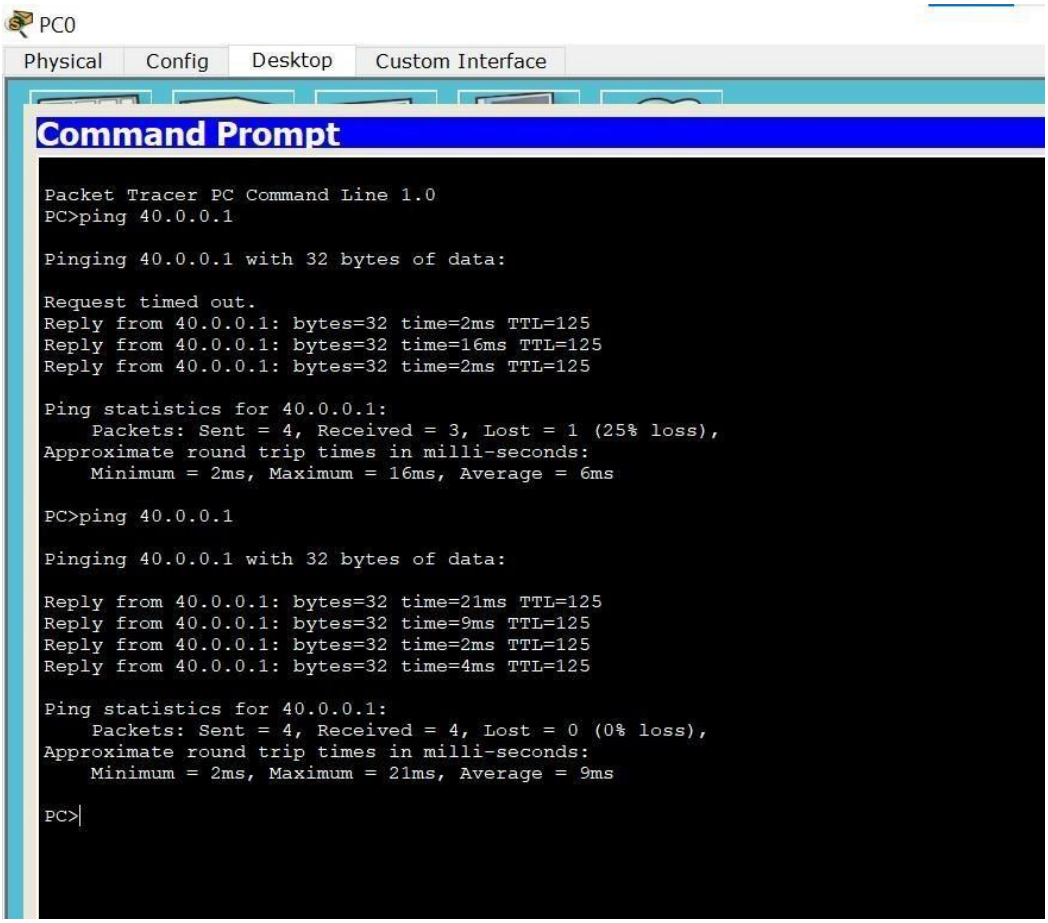

PC0
10.0.0.1
Switch 0
10.0.0.2
20.0.0.2
PC1
20.0.0.1
40.0.0.1
Router 1
20.0.0.1
Router 6
30.0.0.4
PC4
30.0.0.1
30.0.0.2
30.0.0.3
Router 5
80.0.0.1
Switch 1
Switch 2
Switch 3
80.0.0.5
Router 4
10.0.0.1
10.0.0.3
110.0.0.3
PC2
40.0.0.1
40.0.0.3
Router 3
110.0.0.1
50.0.0.3
Router 2
Switch 4
PC3
50.0.0.1

TOPOLOGY:


Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
20.0.0.20
30.0.0.10
Se2/0
Se3/0
Router-PT
Router1
Se2/0
Fa0/0
Router-PT
Router0
10.0.0.10
20.0.0.10
Se3/0
30.0.0.20
40.0.0.10
Fa0/0
Router2
Fa0
PC-PT
PC0
10.0.0.1
Fa0
PC-PT
PC1
40.0.0.1
Time: 00:23:17
Power Cycle Devices Fast Forward Time
Realtime
Connections
Serial DCE
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Statu
Sourc
Destinatic
Type
Colo
Time(s
Period
Num
Edit
Delete

OUTPUT:

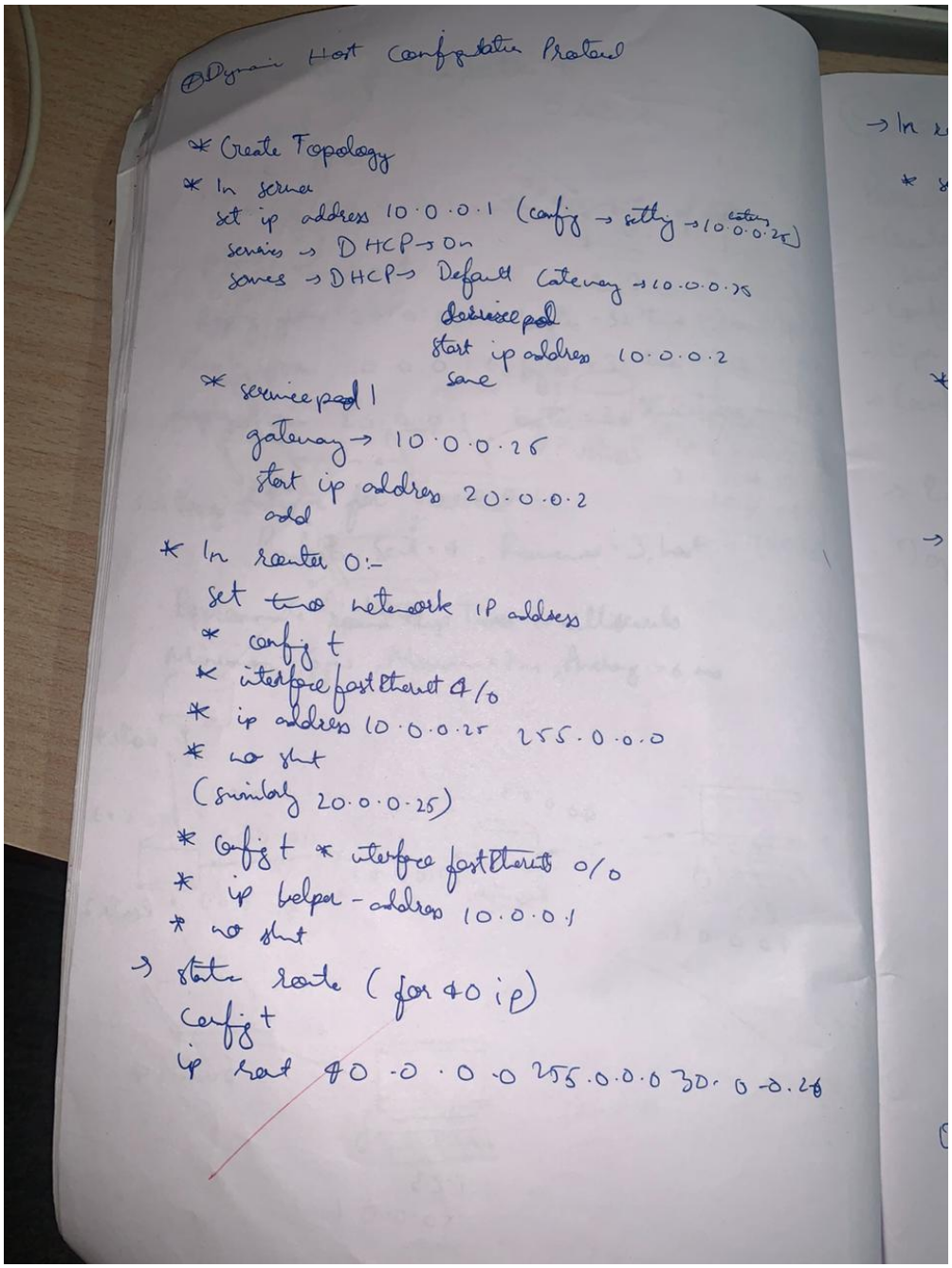

Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT
Router1
Router-PT
Router0
Router-PT
Router2
PC-PT
PC0
PC-PT
PC1
Simulation Panel
Event List
Reset Simulation
Constant Delay
Capturing...
Play Controls
Back
Auto Capture / Play
Capture / Forward
Edit Filters
Show All/None
Time: 00:26:11.346
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


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:



→ In router 1 :-

* set ip address
  config t
  interface fastethernet 0/0
  ip address 40.0.0.26 255.0.0.0
  no shut

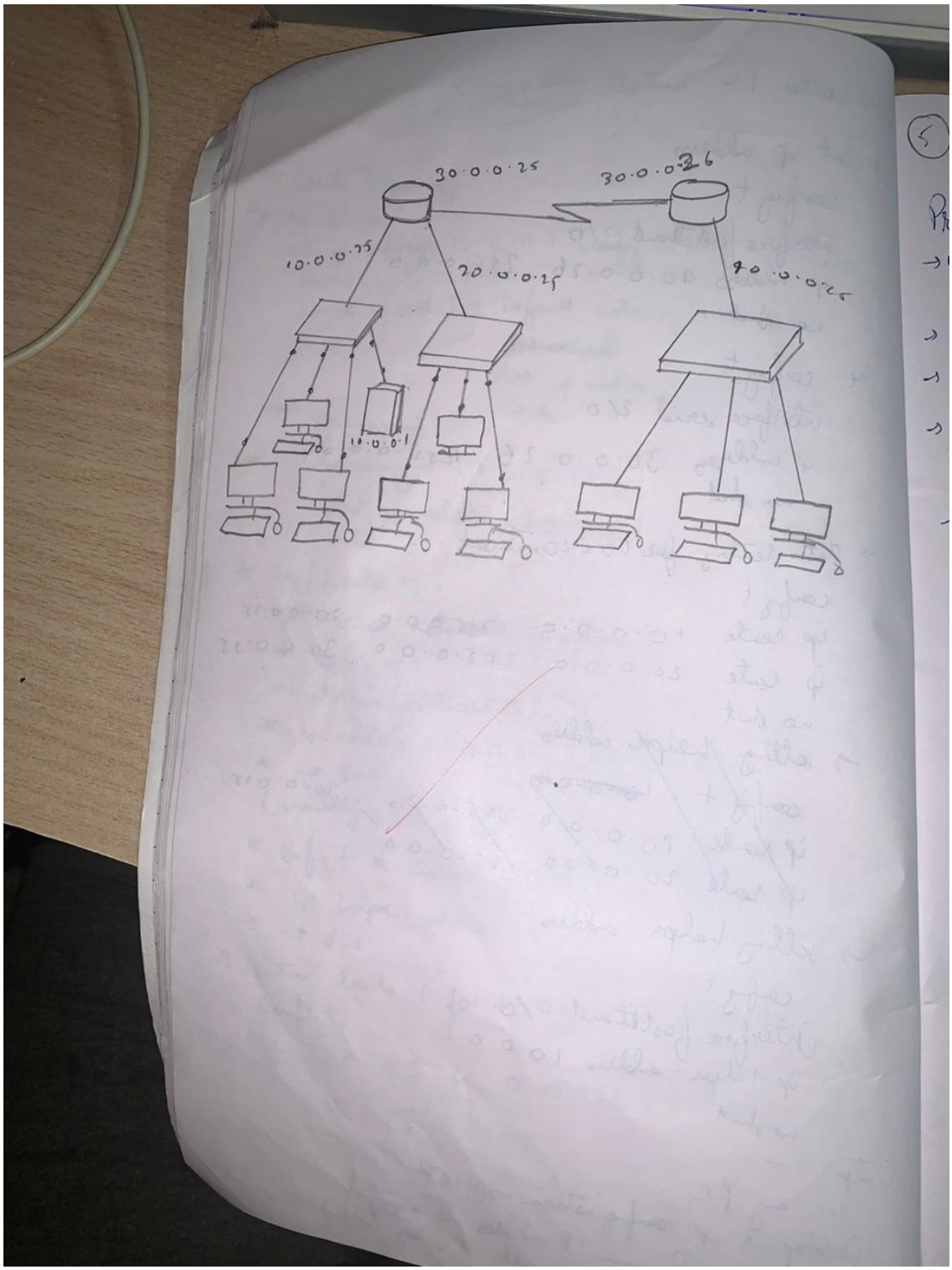
* config t
  interface serial 2/0
  ip address 30.0.0.26 255.0.0.0
  no shut

→ state routing for 10 & 20 network
  config t
  ip route 10.0.0.0 255.0.0.0 30.0.0.25
  ip rote 20.0.0.0 255.0.0.0 30.0.0.25
  no shut

→ setting helpa addres
  config t
  ip route 10.0.0.0 255.0.0.0 30.0.0.25
  ip rote 20.0.0.0 25.0.0.0

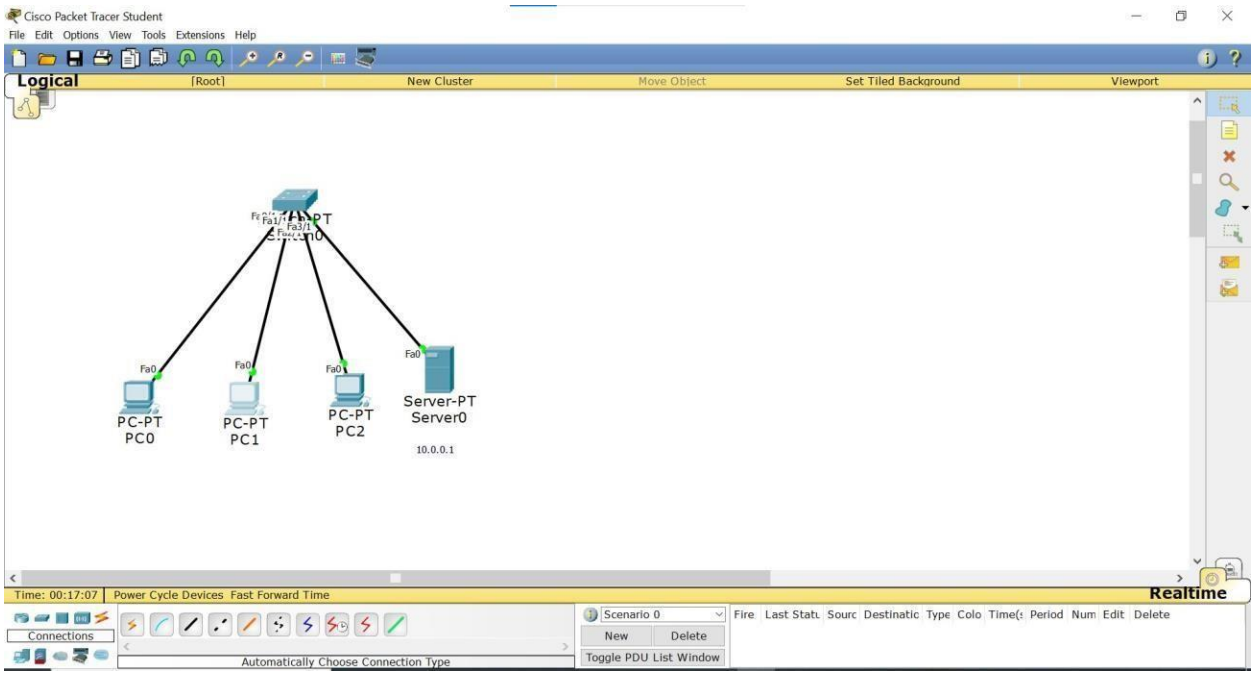
→ setting helpa addres
  config t
  interface fastEthernet 0/0
  ip-helpa-addres 10.0.0.1
  no shut

Output :-
  In any PC
Desktop → ip configuration → DHCP
  (Dynamic IP addres is assigned to all PC by dhcp)

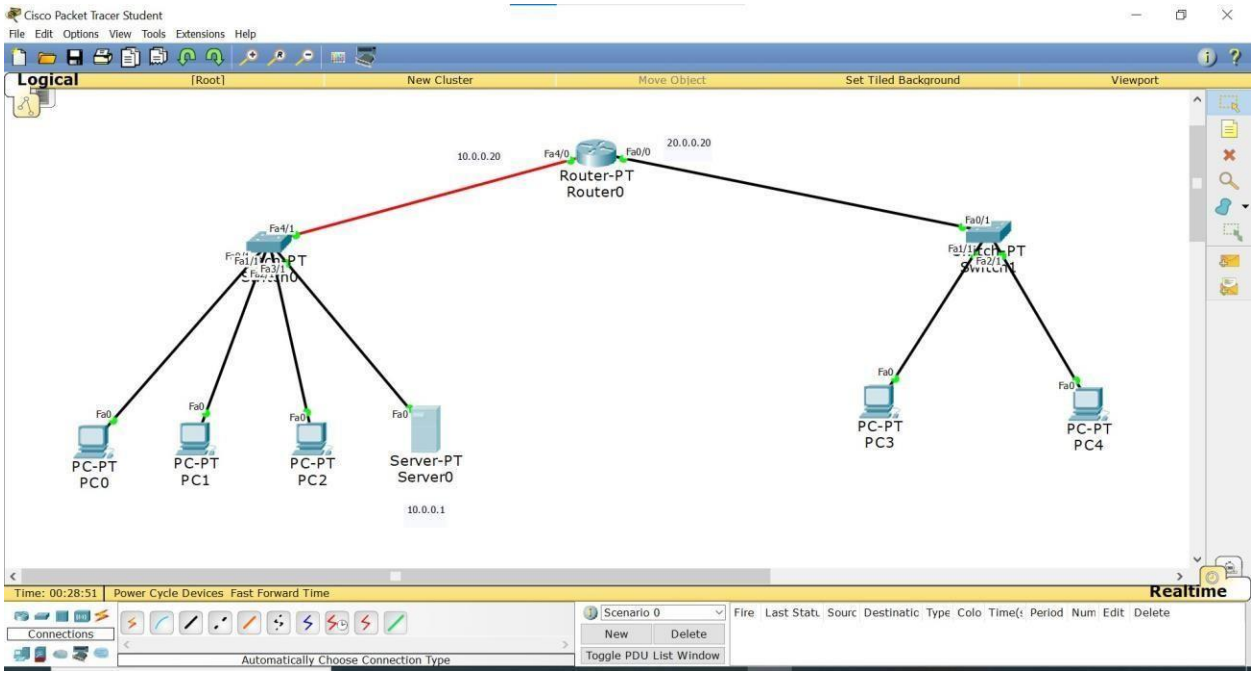

30.0.0.25
30.0.0.26
10.0.0.25
20.0.0.25
40.0.0.25
10.0.0.1

TOPOLOGY:

PROGRAM 4.1:

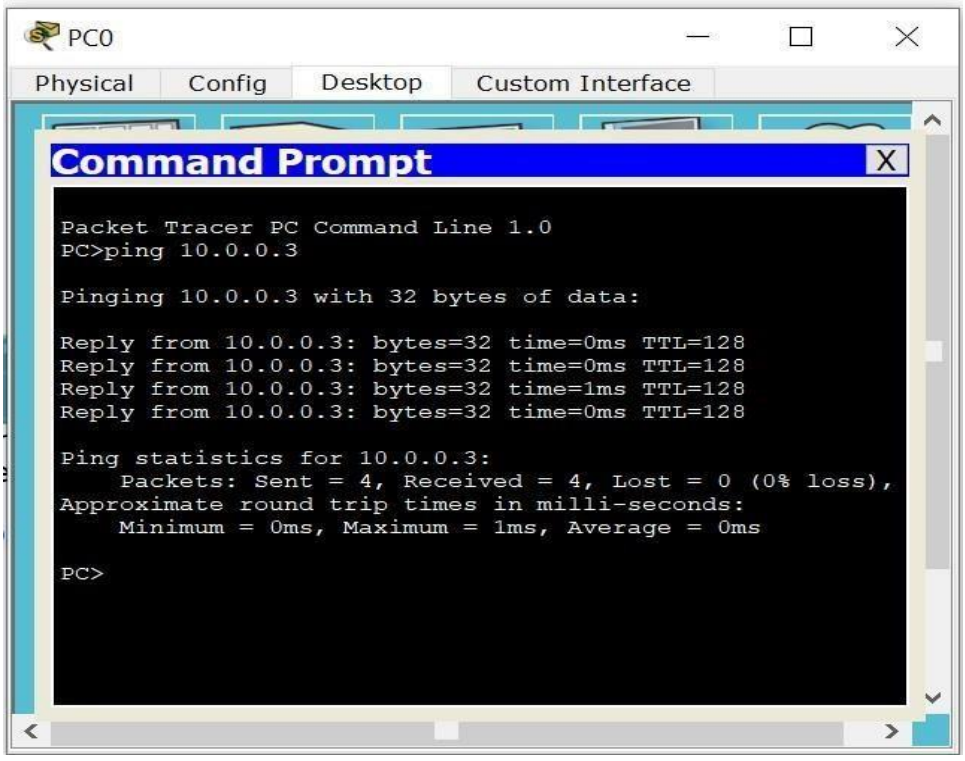


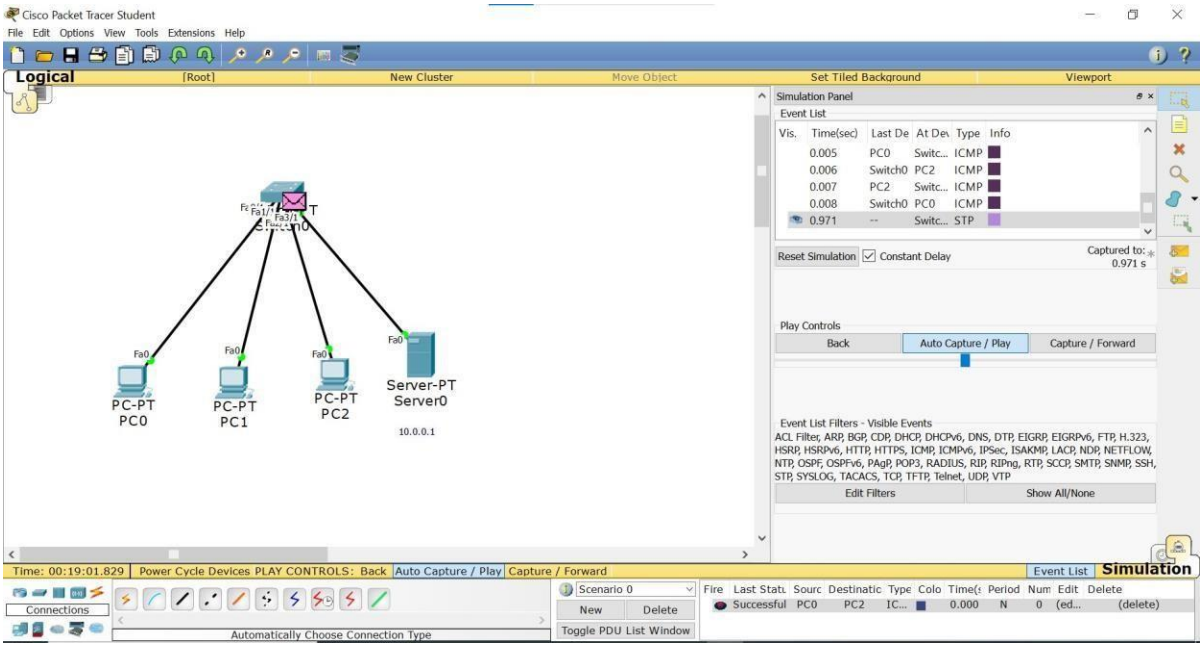
PROGRAM 4.2:



OUTPUT:

PROGRAM 4.1:





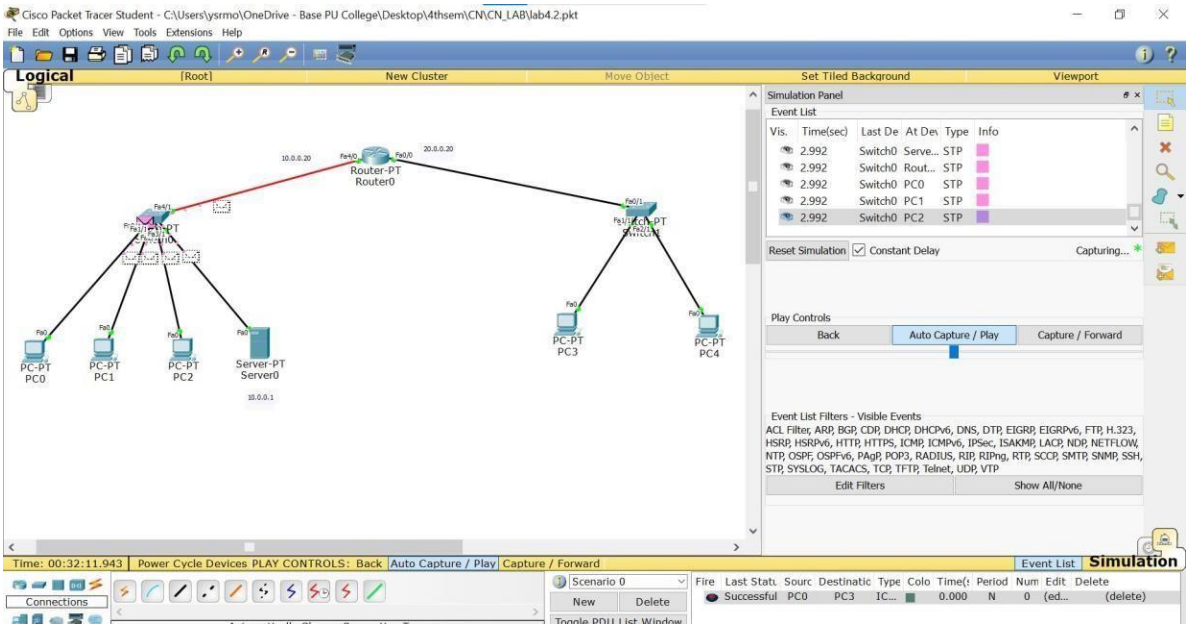
PROGRAM 4.2:

PC0

Physical | Config | Desktop | Custom Interface

Command Prompt

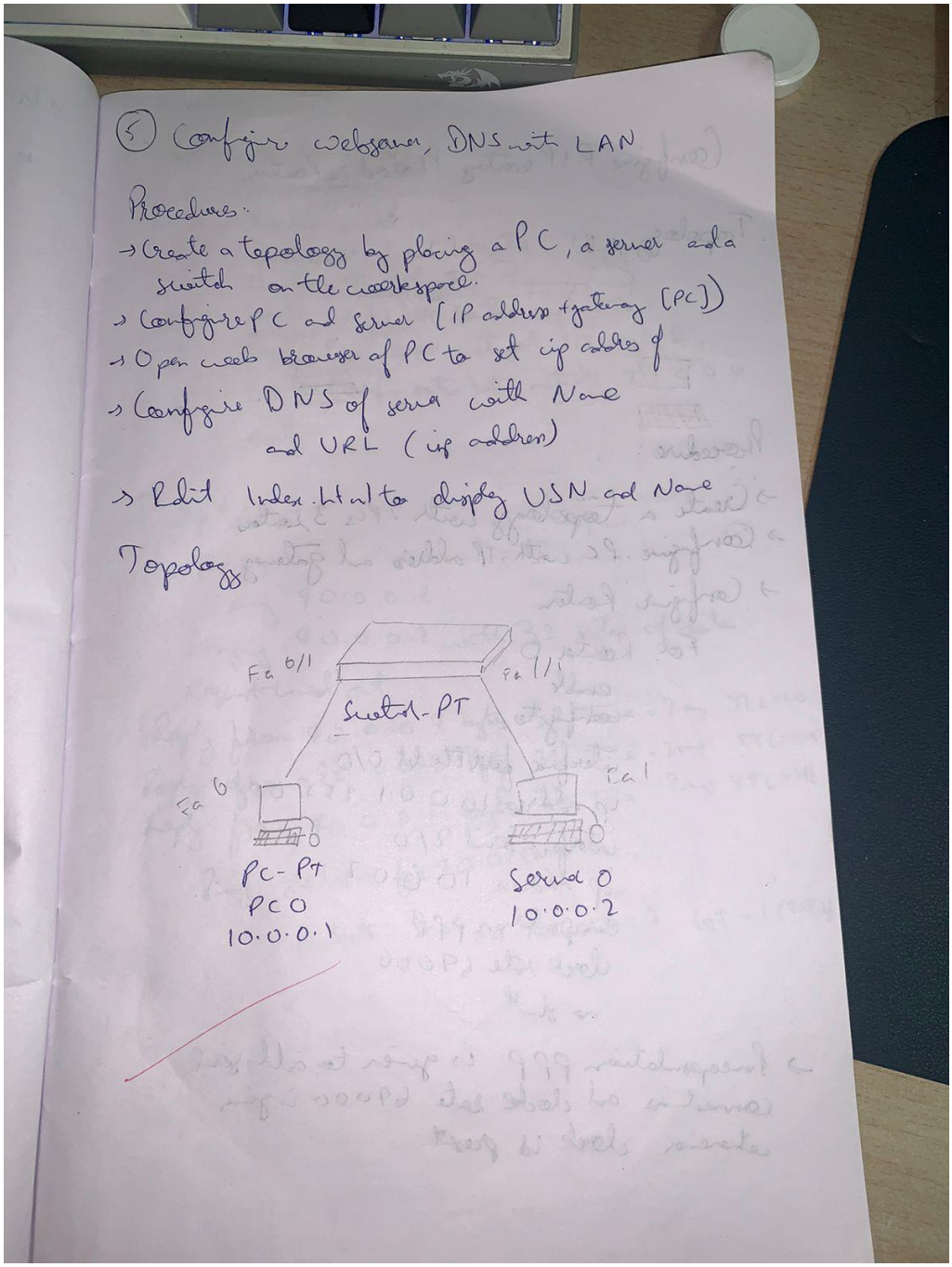
```
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.2

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>
```



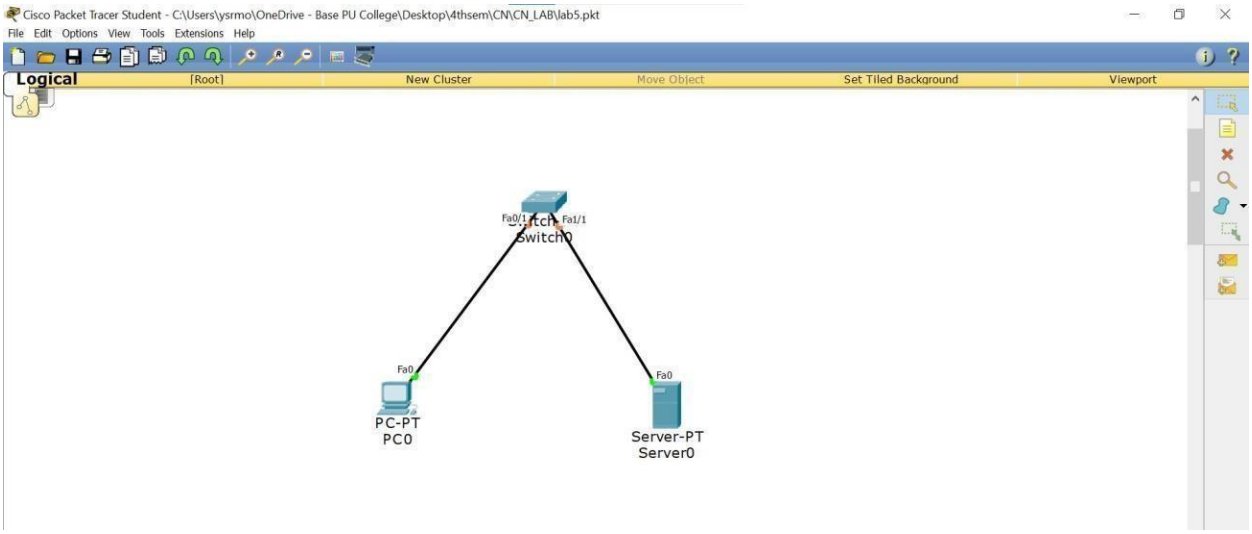
WEEK5

Configure Web Server, DNS within a LAN.

OBSERVATION:



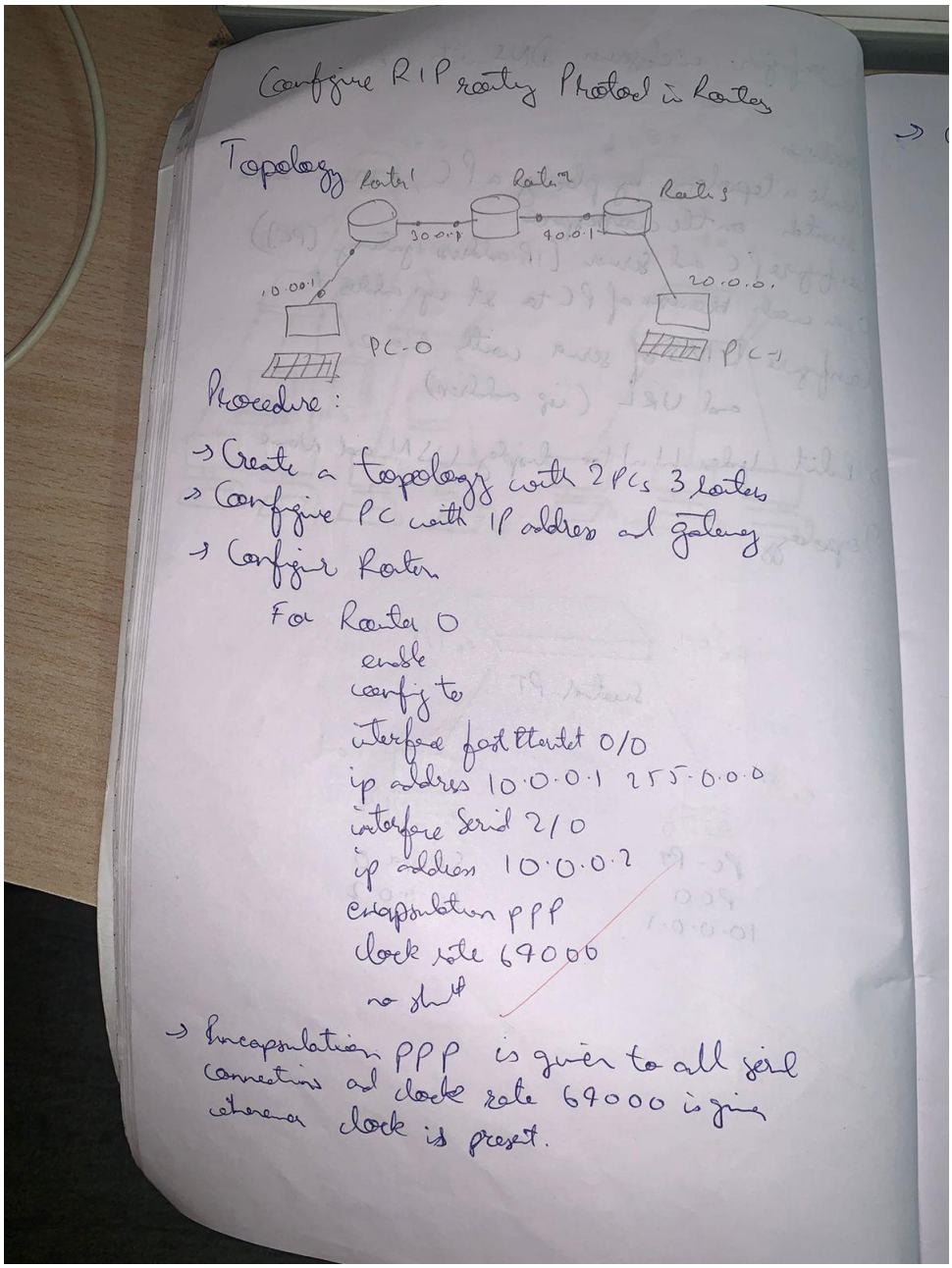
TOPOLOGY:


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

LAB 6

Configure RIP routing Protocol in Routers.

OBSERVATION:



→ Configure all routers

For router0

Router enable

Router # show ip route

Router # config t

Router (config) # router rip

Router (config) # network 10.0.0.0

Router (config-router) # network 20.0.0.0

exit

Output:-

In PC0

Ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data

Request timed out

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

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

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

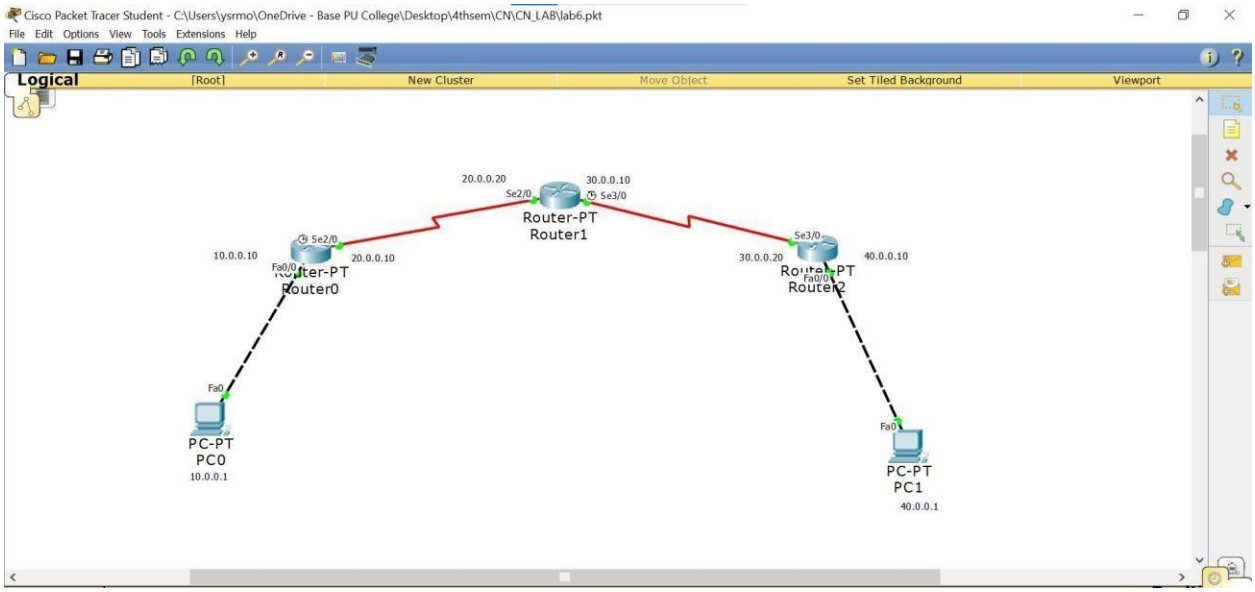
Ping statistics for 40.0.0.2

Packet Sent = 4 Recieved = 3 Lost = 1 (25%)

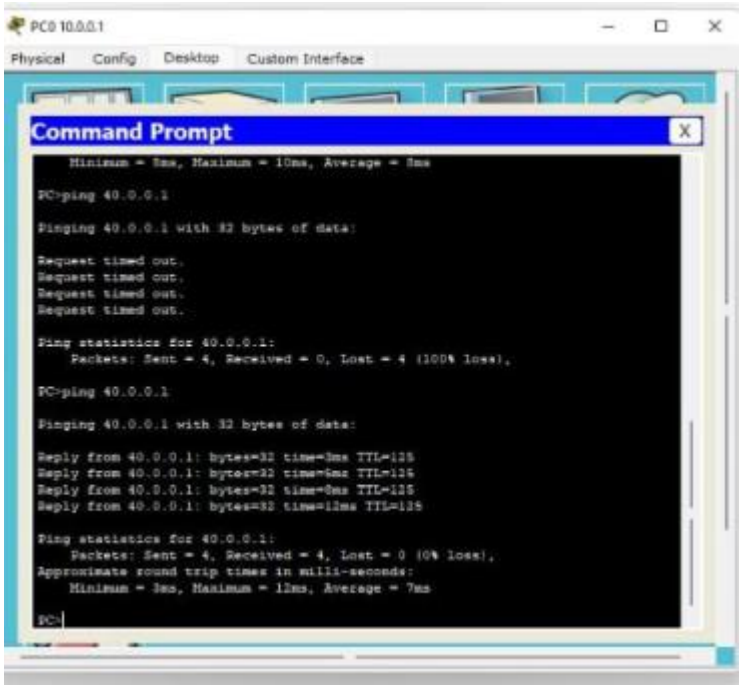
Late Submission

5/11/23

TOPOLOGY:



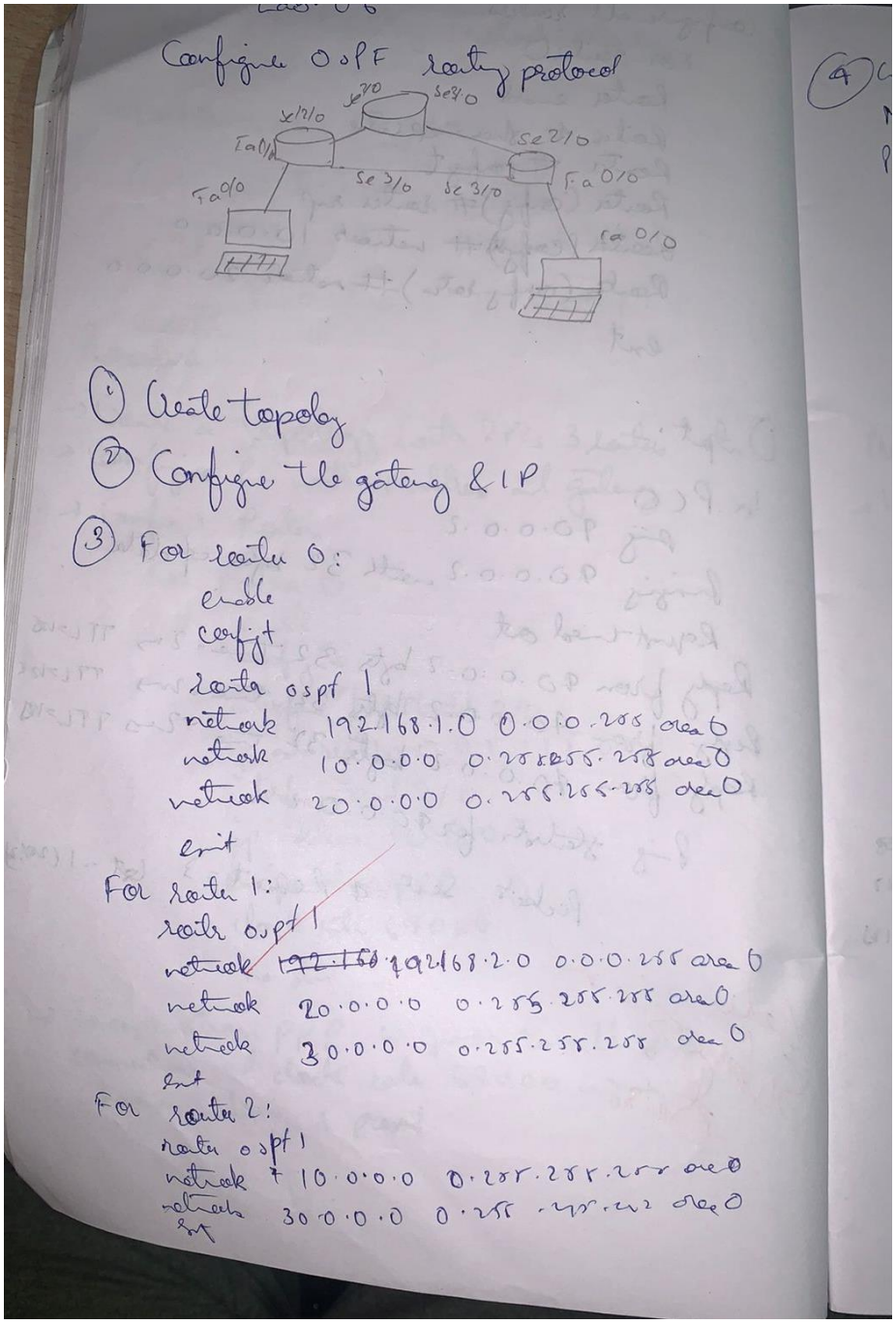
OUTPUT:



LAB 7

Configure OSPF routing protocol.

OBSERVATION:



(4) Go to Simulation & uncheck all the filters

Now Ping 192.168.2.1 in PC0

Pinging 192.168.2.1 with 32 bytes of data

Reply from 192.168.2.1 bytes=32 time=4ms TTL=128

Reply from 192.168.2.1 bytes=32 time=4ms TTL=128

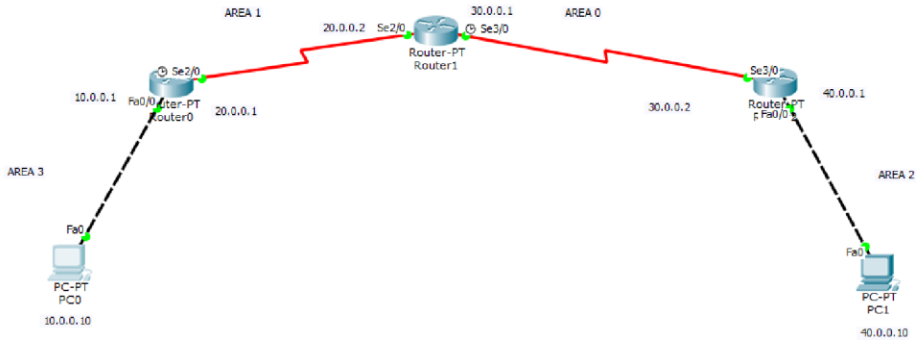
Reply from 192.168.2.1 bytes=32 time=4ms TTL=128

Reply from 192.168.2.1 bytes=32 time=4ms TTL=128

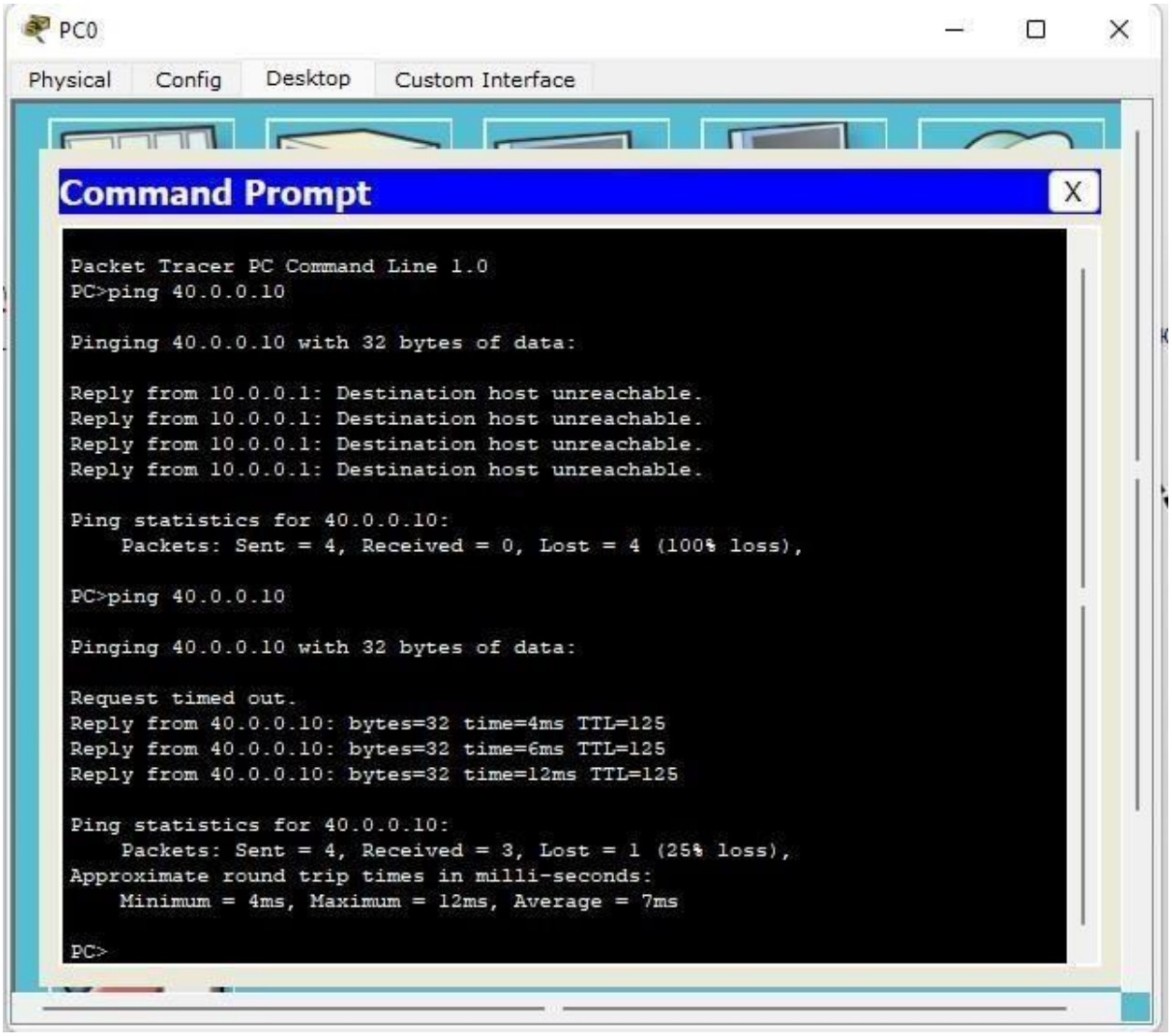
Ping statistics for 192.168.2.1:

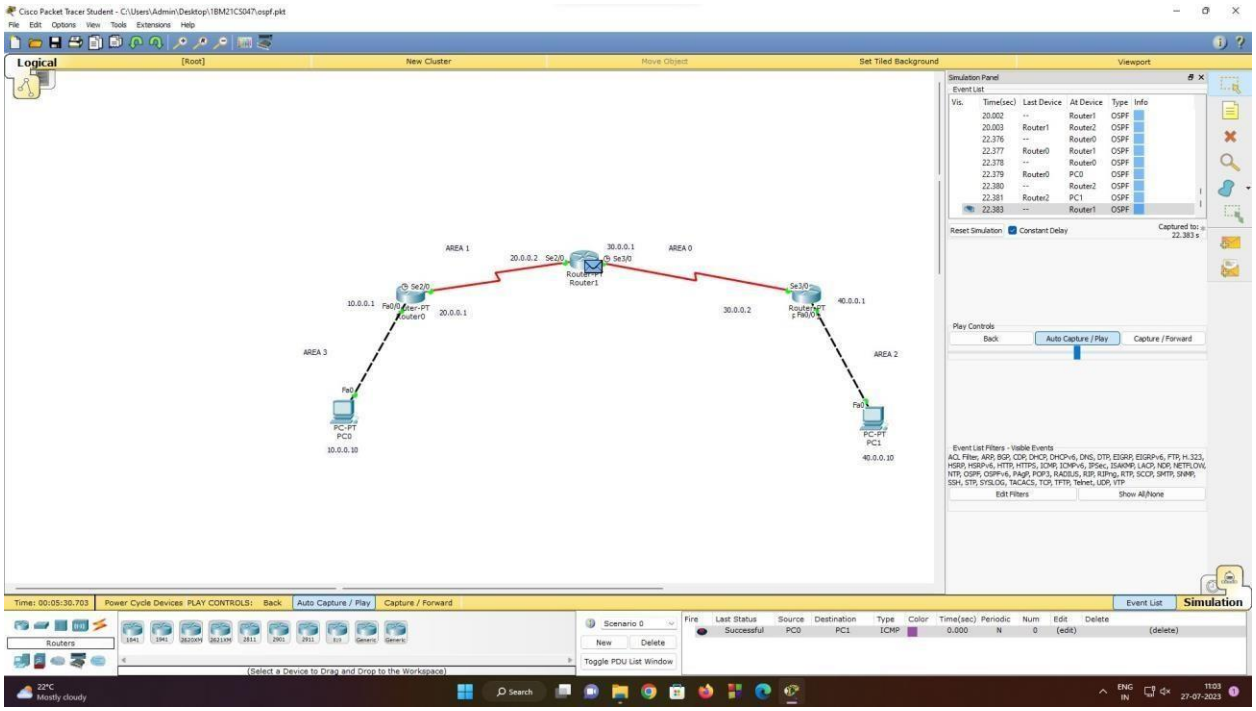
Packets: Sent = 4, Received = 4 Lost 0 (0% loss)

TOPOLOGY:



OUTPUT:

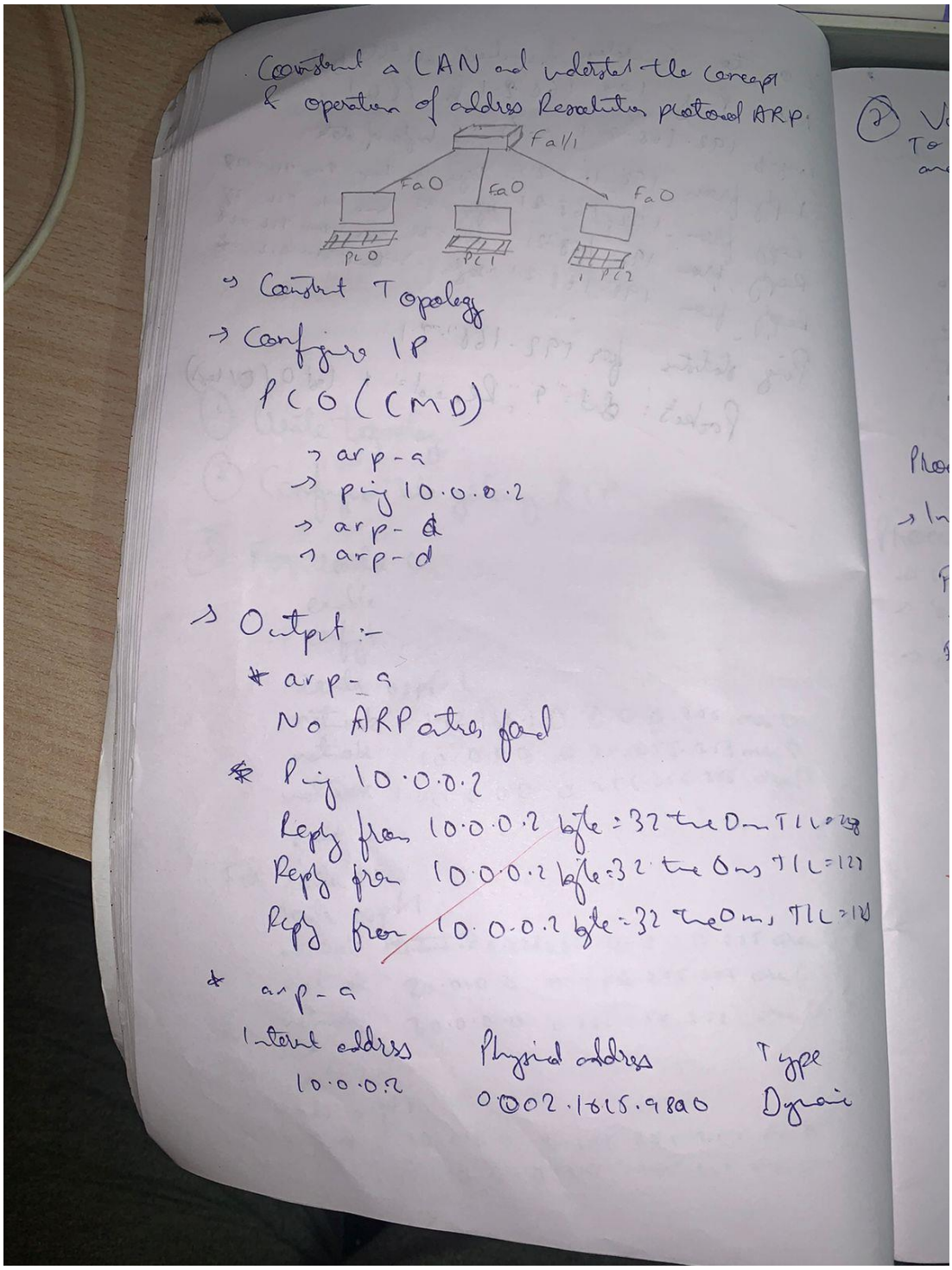



Cisco Packet Tracer Student - C:\Users\Admin\Desktop\1BM21CS047\ospf.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
AREA 1
AREA 0
AREA 3
AREA 2
20.0.0.2
Se2/0
30.0.0.1
Se3/0
Router1
10.0.0.1
Fa0/0
20.0.0.1
30.0.0.2
40.0.0.1
PC-PT PC0
10.0.0.10
PC-PT 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

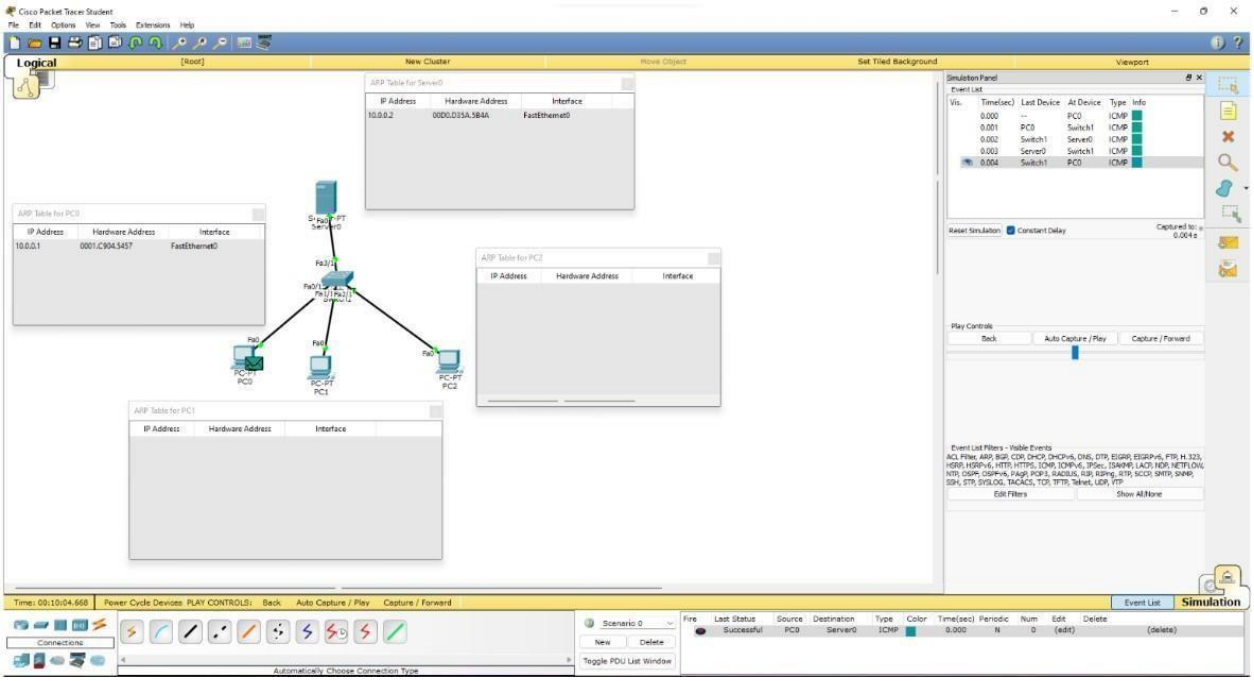
LAB 8

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

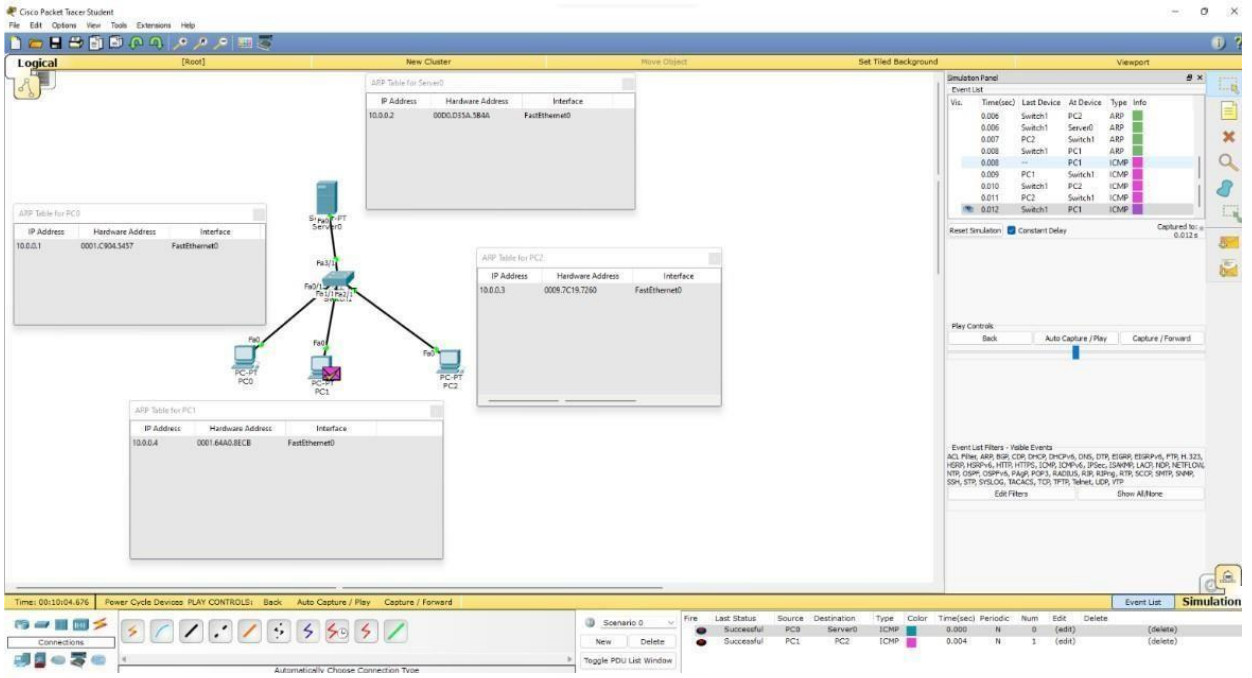
OBSERVATION:

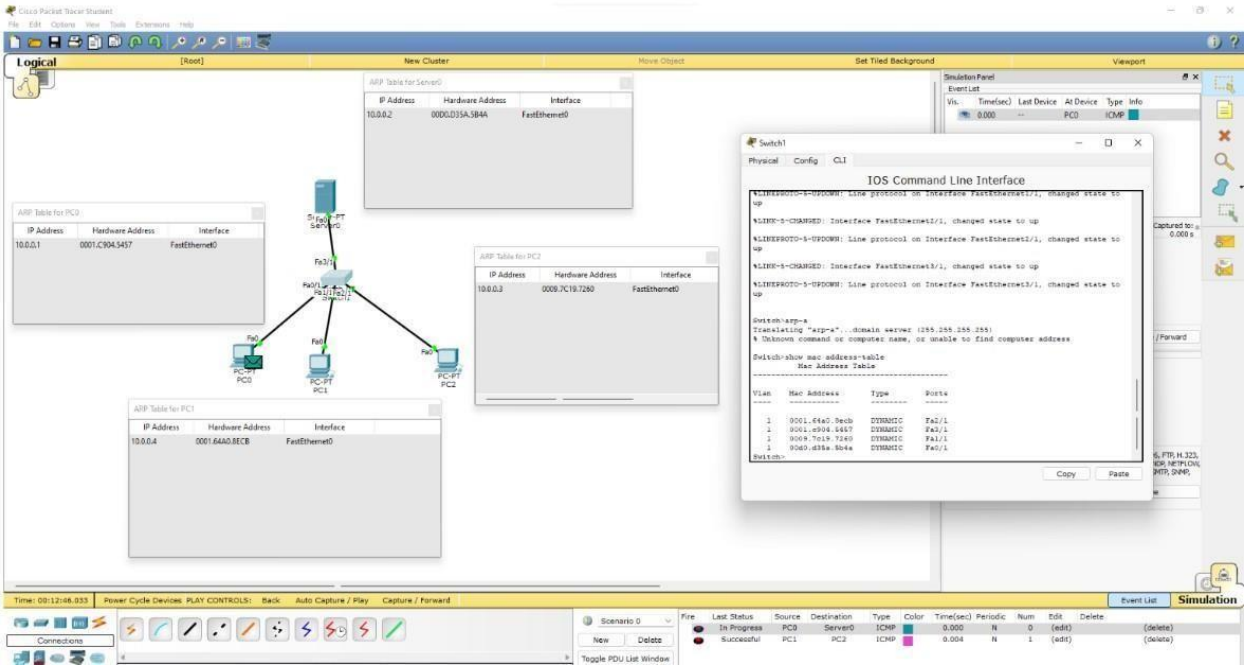


TOPOLOGY:


Cisco Packet Tracer Student
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
ARP Table for Server0
IP Address
Hardware Address
Interface
10.0.0.2
00D0.D35A.5B4A
FastEthernet0
ARP Table for PC0
10.0.0.1
0001.C904.5457
FastEthernet0
ARP Table for PC2
ARP Table for PC1
Server-PT Server0
PC-PT PC0
PC-PT PC1
PC-PT PC2
Simulation Panel
Event List
Vis.
Time(sec)
Last Device
At Device
Type
Info
Reset Simulation
Constant Delay
Play Controls
Back
Auto Capture / Play
Capture / Forward
Edit Filters
Show All/None
Time: 00:10:04.668
Power Cycle Devices
PLAY CONTROLS:
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
Server0
ICMP
0.000
N
0
(edit)
(delete)

OUTPUT:

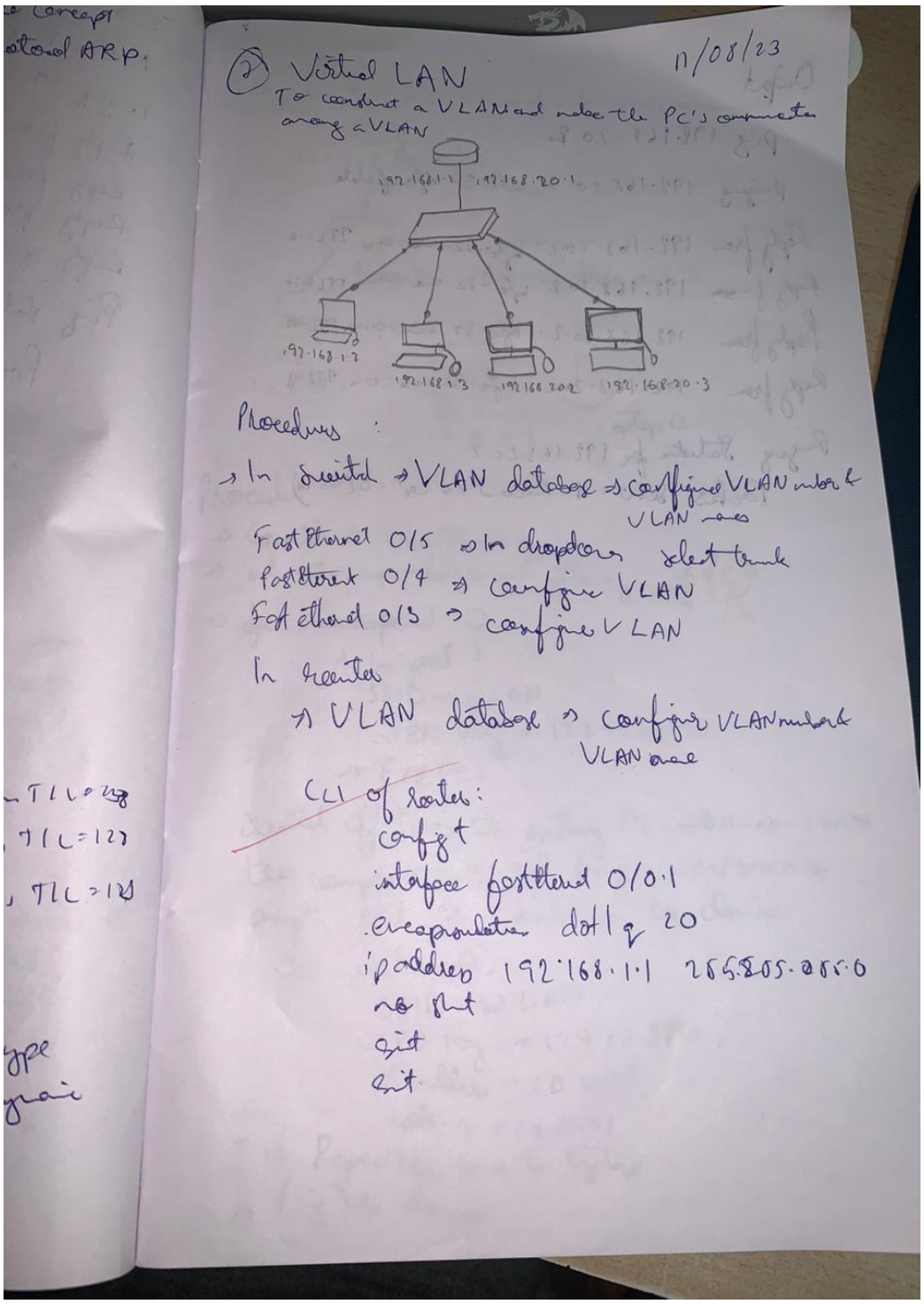




LAB 9

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

OBSERVATION:



Output

Ping 192.168.20.2

Pinging 192.168.20.2 with 32 bytes of data:

Reply from 192.168.20.2: bytes=32 time=0ms TTL=128

Reply from 192.168.20.2: bytes=32 time=0ms TTL=128

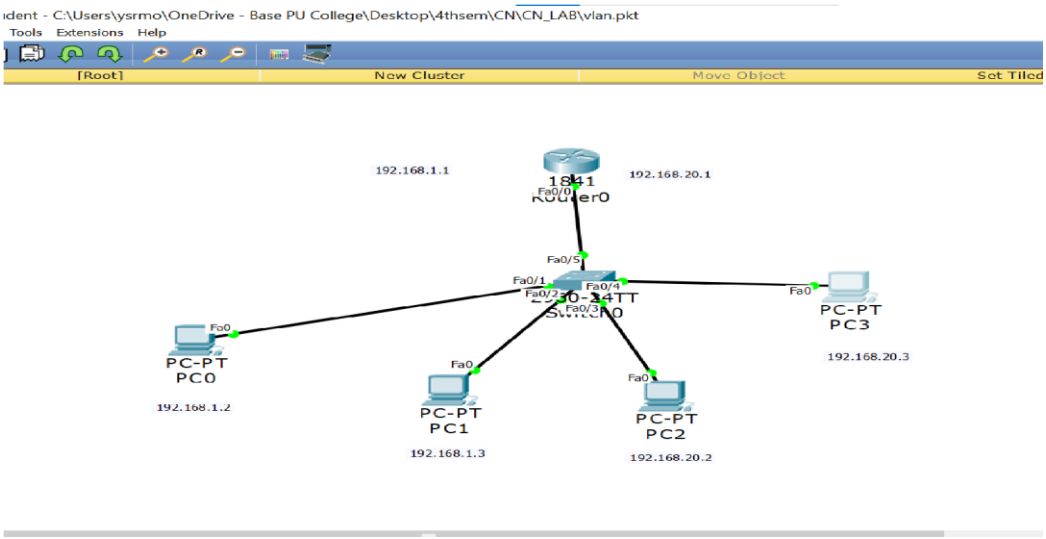
Reply from 192.168.20.2: bytes=32 time=0ms TTL=128

Reply from 192.168.20.2: bytes=32 time=0ms TTL=128

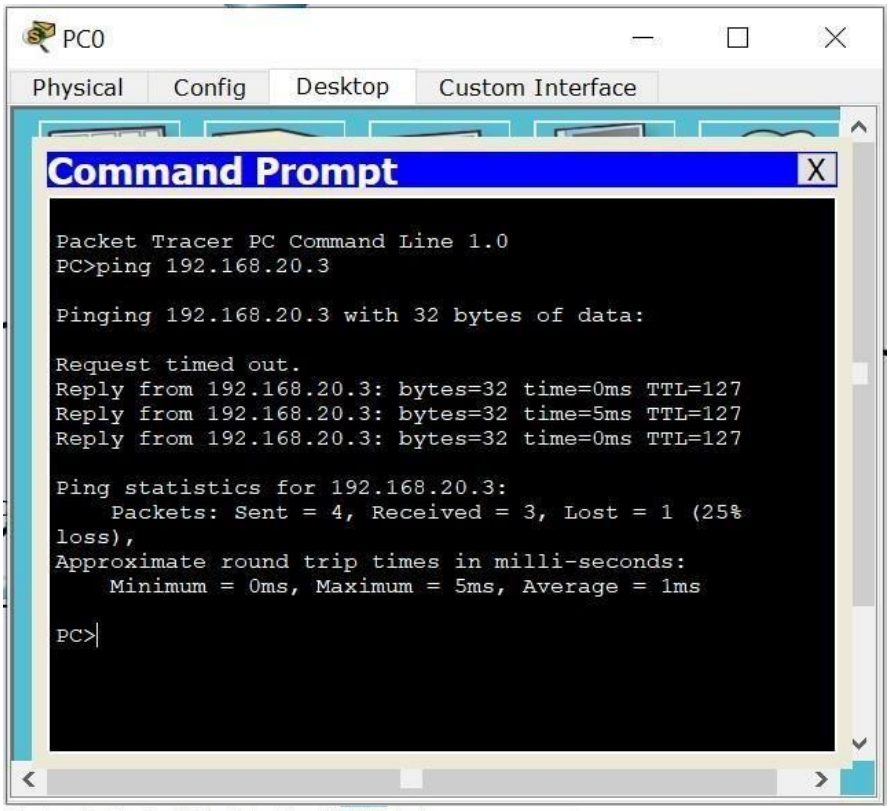
Ping statistics for 192.168.20.2:

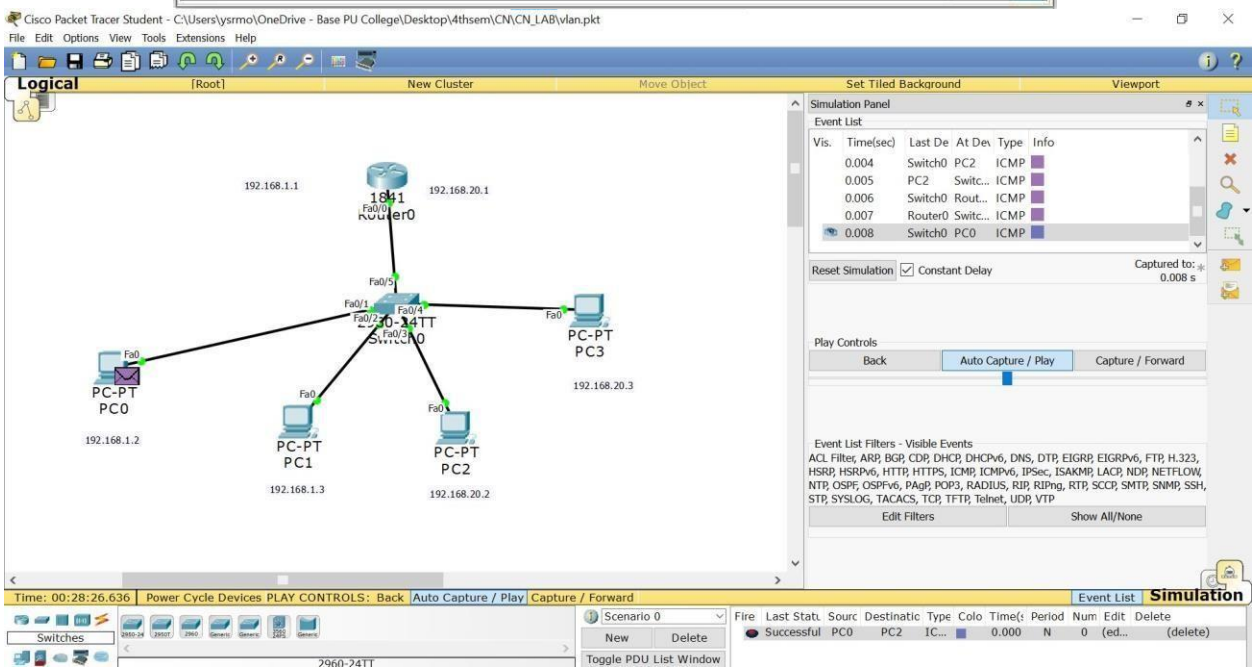
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)

TOPOLOGY:


ident - C:\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\CN\CN_LAB\vlan.pkt
Tools Extensions Help
[Root]
New Cluster
Move Object
Set Tiled
192.168.1.1
1841
Router0
Fa0/0
192.168.20.1
Fa0/5
Fa0/1
Fa0/4
Fa0/3
Fa0
PC-PT
PC3
192.168.20.3
Fa0
PC-PT
PC0
192.168.1.2
Fa0
PC-PT
PC1
192.168.1.3
Fa0
PC-PT
PC2
192.168.20.2

OUTPUT:

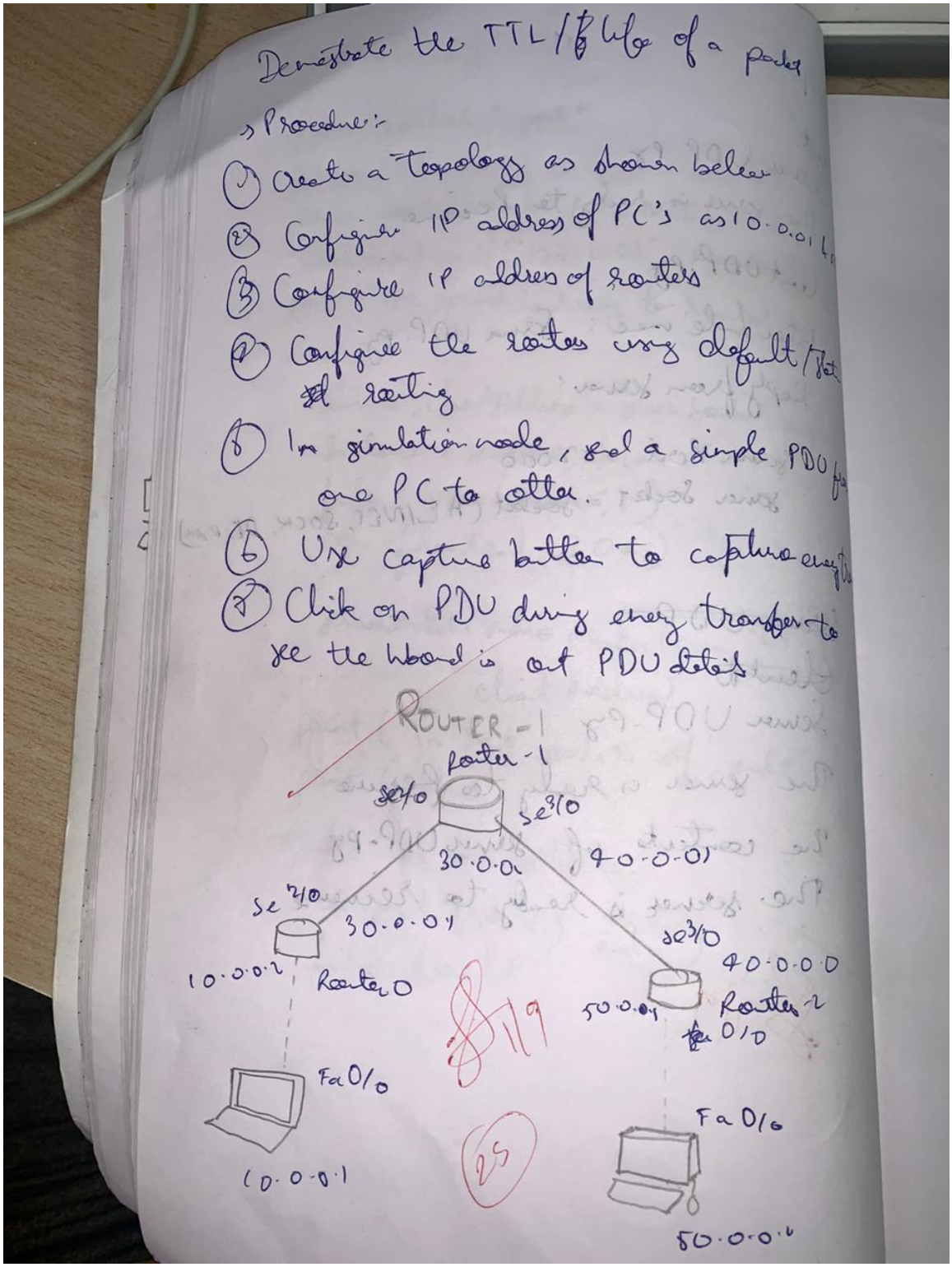

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 192.168.20.3
Pinging 192.168.20.3 with 32 bytes of data:
Request timed out.
Reply from 192.168.20.3: bytes=32 time=0ms TTL=127
Reply from 192.168.20.3: bytes=32 time=5ms TTL=127
Reply from 192.168.20.3: bytes=32 time=0ms TTL=127
Ping statistics for 192.168.20.3:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 5ms, Average = 1ms
PC>


Cisco Packet Tracer Student - C:\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\CN\CN_LAB\vlan.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
192.168.1.1
192.168.20.1
Router0
Fa0/5
Switch0
PC-PT PC0
192.168.1.2
PC-PT PC1
192.168.1.3
PC-PT PC2
192.168.20.2
PC-PT PC3
192.168.20.3
Simulation Panel
Event List
Vis. Time(sec) Last De At De Type Info
0.004 Switch0 PC2 ICMP
0.005 PC2 Switc... ICMP
0.006 Switch0 Rout... ICMP
0.007 Router0 Switc... ICMP
0.008 Switch0 PC0 ICMP
Reset Simulation
Constant Delay
Captured to: 0.008 s
Play Controls
Back
Auto Capture / Play
Capture / Forward
Event List Filters - Visible Events
ACL Filter, ARP, BGP, CDP, DHCP, DHCPv6, DNS, DTP, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, LACP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, RADIUS, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, VTP
Edit Filters
Show All/None
Time: 00:28:26.636
Power Cycle Devices PLAY CONTROLS: Back
Auto Capture / Play
Capture / Forward
Event List
Simulation
Switches
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Statu Sourc Destinatic Type Colo Time(s Period Num Edit Delete
Successful PC0 PC2 IC... 0.000 N 0 (ed... (delete)

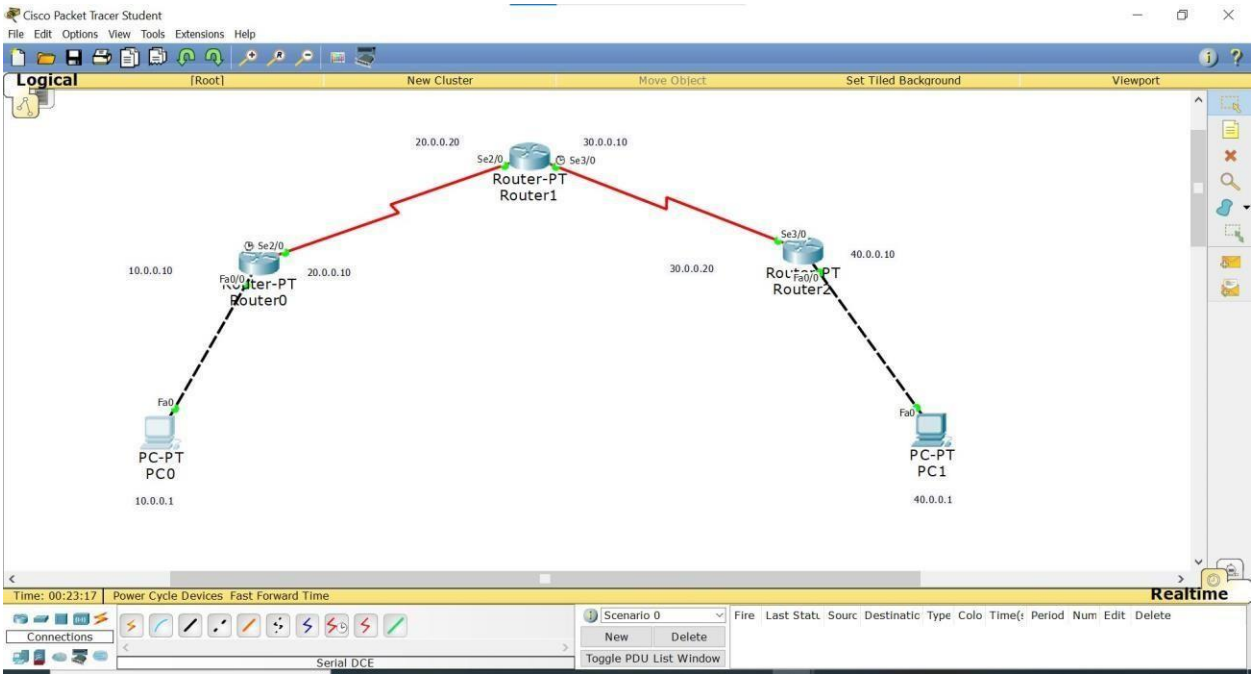
LAB 10

Demonstrate the TTL/ Life of a Packet.

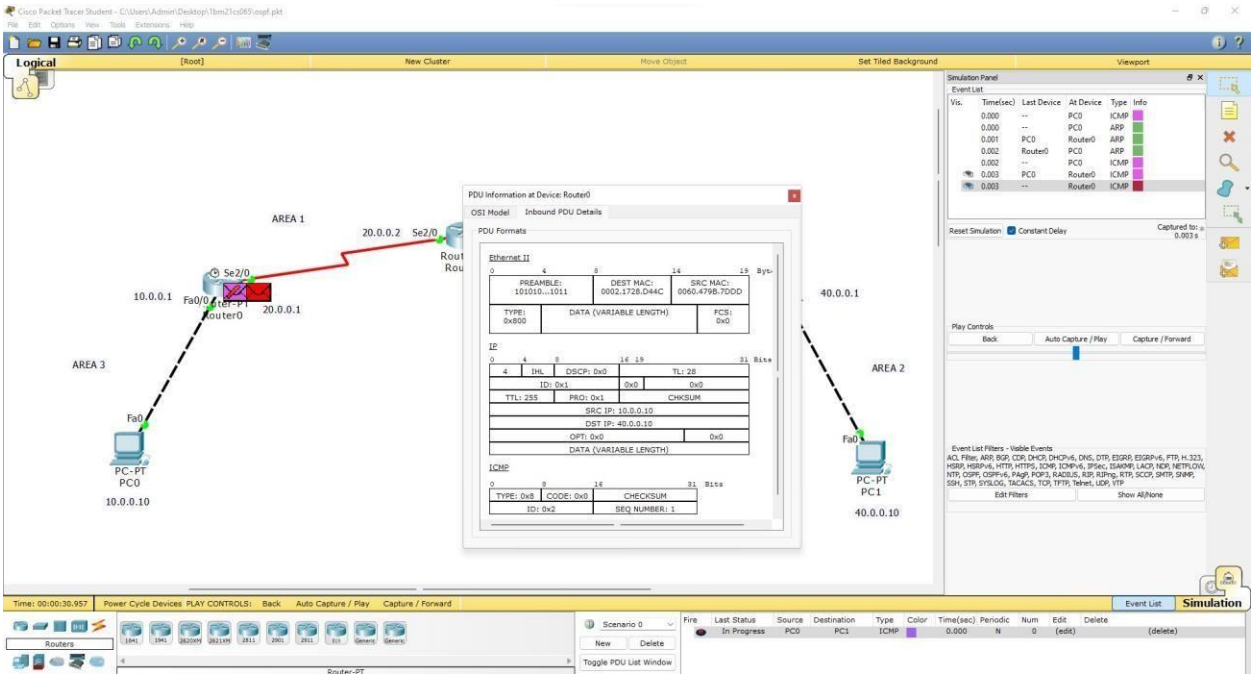
OBSERVATION:

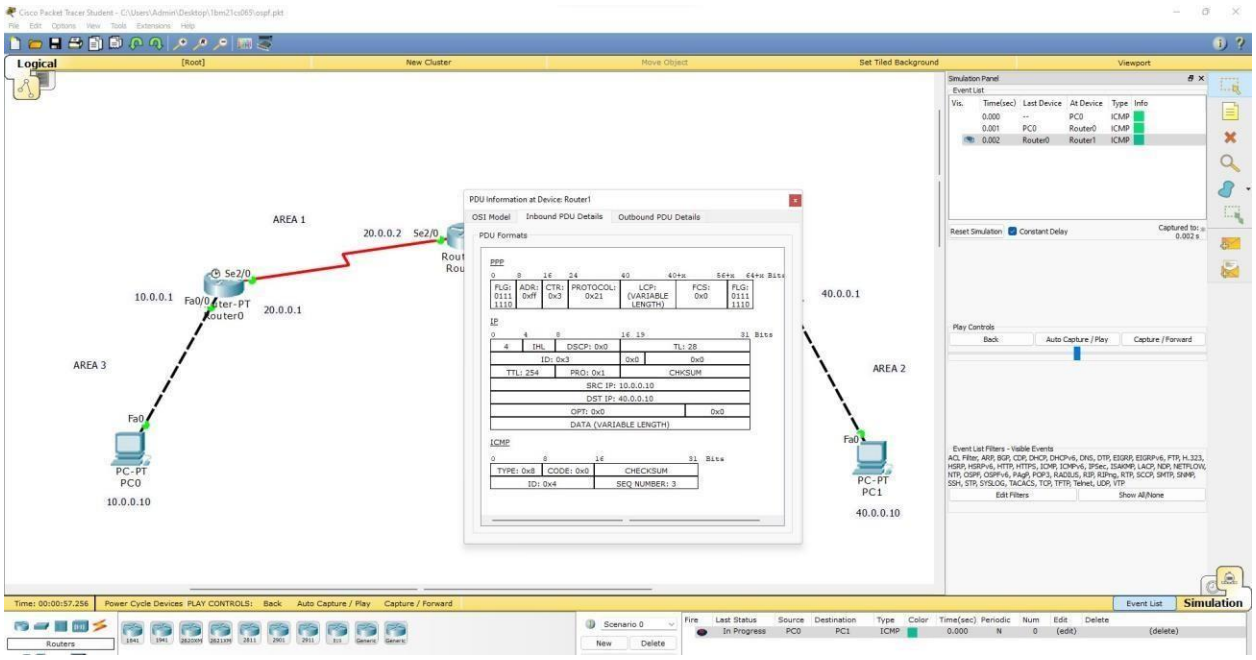


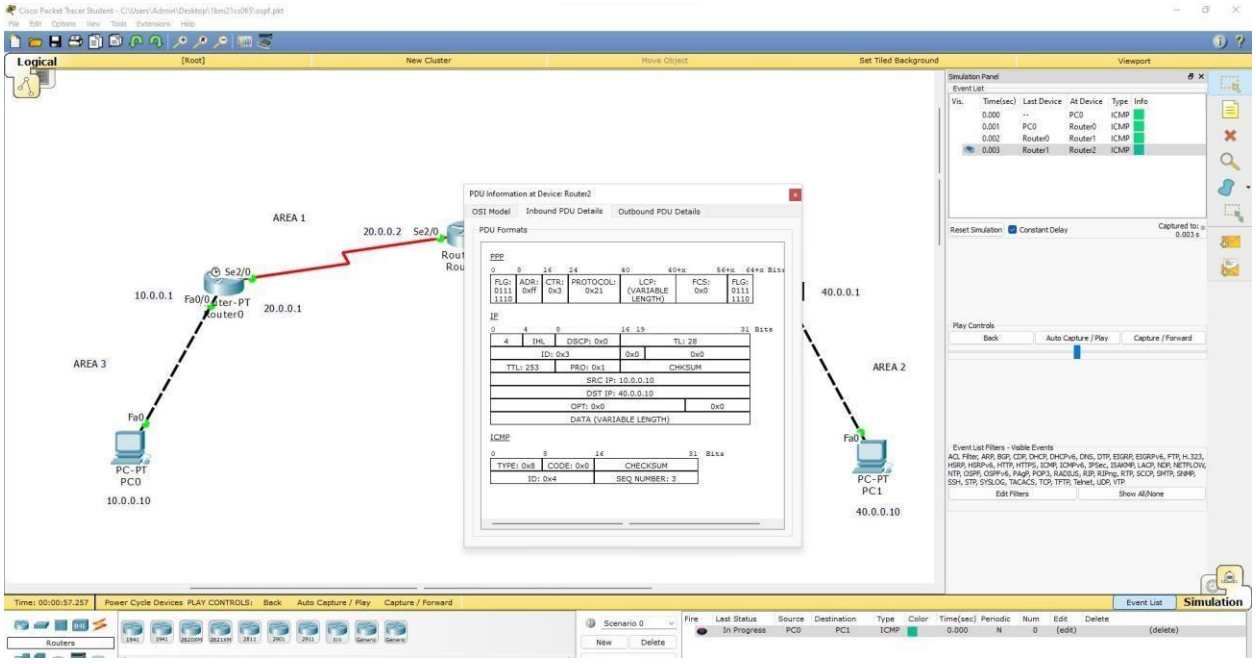
TOPOLOGY:

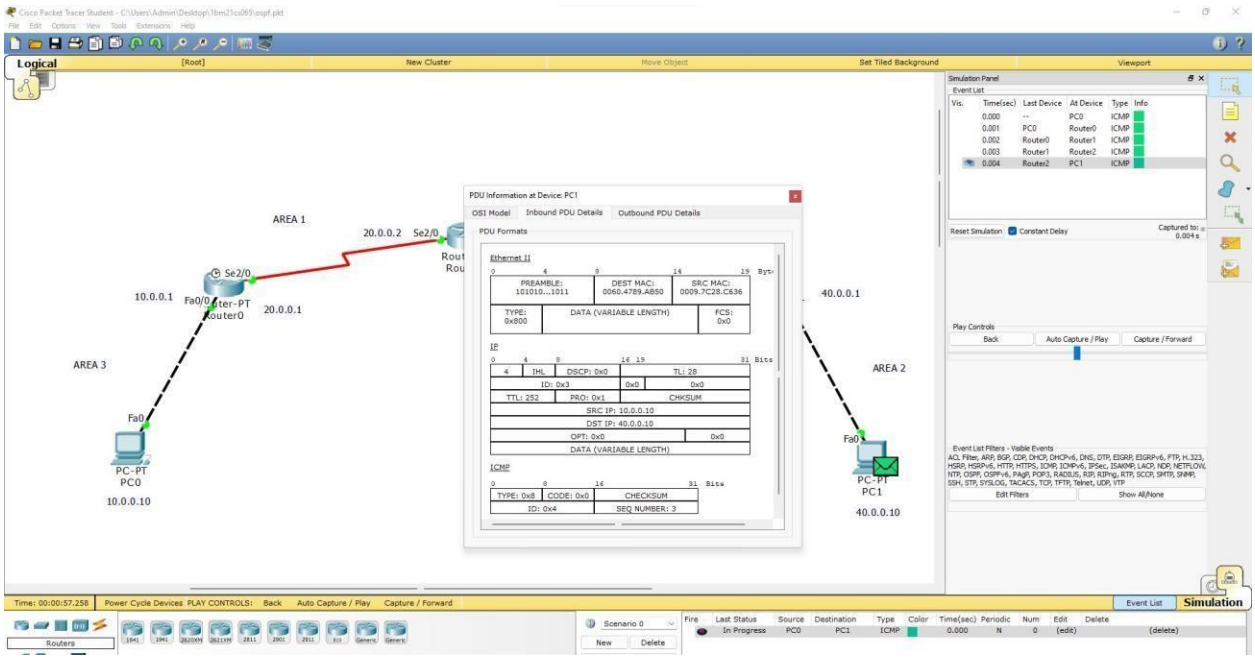


OUTPUT:




Logical
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
SRC IP: 10.0.0.10
DST IP: 40.0.0.10
DATA (VARIABLE LENGTH)
CHECKSUM
SEQ NUMBER: 3
Simulation Panel
Event List
Play Controls
Back
Auto Capture / Play
Capture / Forward
Edit Filters
Show All/None
Time: 00:00:57.256
Power Cycle Devices
PLAY CONTROLS:
Event List
Simulation

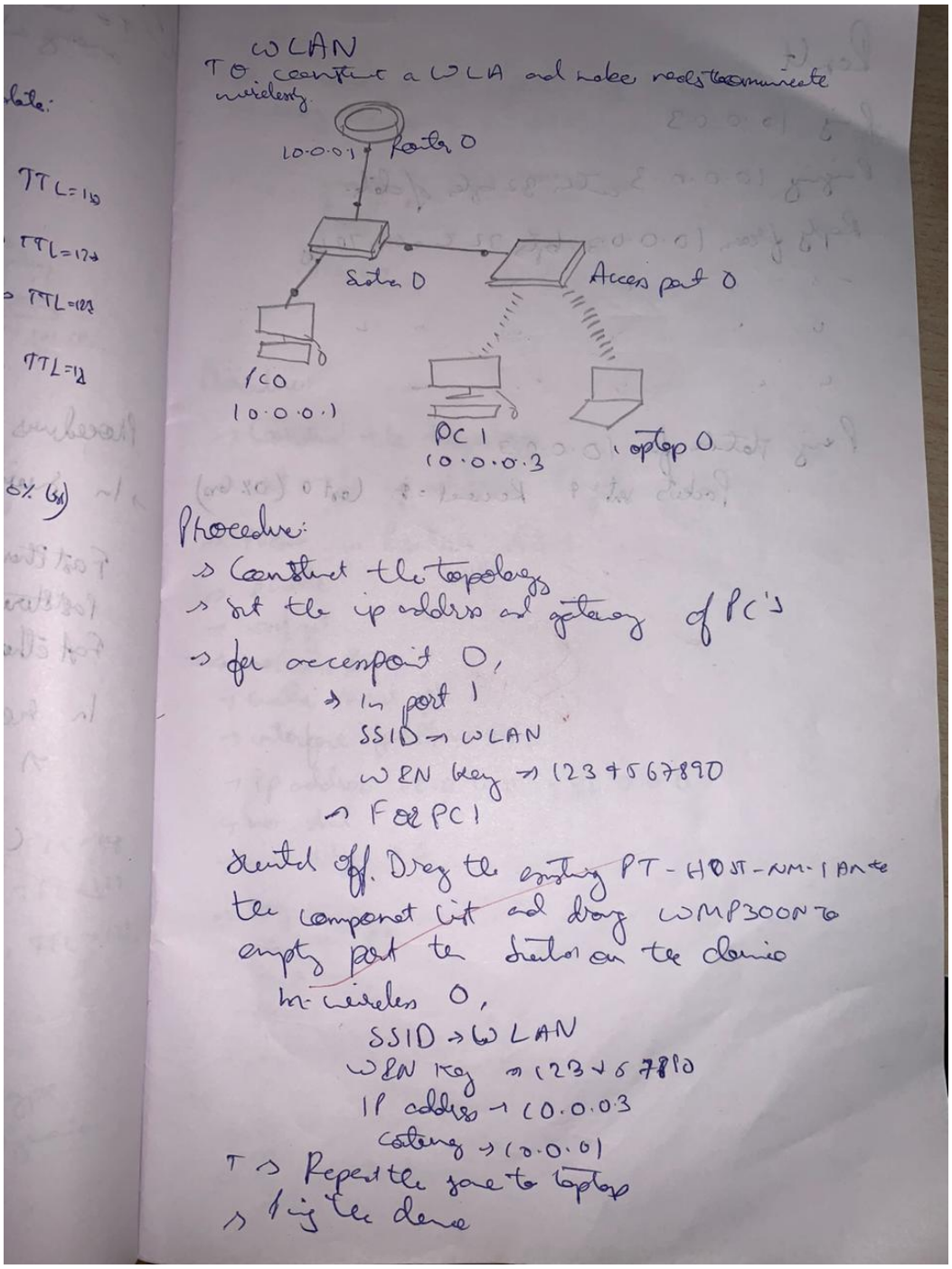

Logical
AREA 1
AREA 2
AREA 3
20.0.0.2
Se2/0
10.0.0.1
Fa0/0
20.0.0.1
40.0.0.1
Fa0
PC-PT
PC0
10.0.0.10
PC-PT
PC1
40.0.0.10
PDU Information at Device: Router2
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
PPP
IP
ICMP
SRC IP: 10.0.0.10
DST IP: 40.0.0.10
DATA (VARIABLE LENGTH)
CHECKSUM
SEQ NUMBER: 3
Simulation Panel
Event List
Play Controls
Back
Auto Capture / Play
Capture / Forward
Edit Filters
Show All/None
Time: 00:00:57.257
Power Cycle Devices
PLAY CONTROLS:
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: PC1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
PREAMBLE: 10101010...1011
DEST MAC: 0060.4789.AB50
SRC MAC: 0009.7C28.C636
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
THL
DSCP: 0x0
TL: 28
ID: 0x3
TTL: 252
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.10
DST IP: 40.0.0.10
OPT: 0x0
DATA (VARIABLE LENGTH)
ICMP
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x4
SEQ NUMBER: 3
Simulation Panel
Event List
Play Controls
Back
Auto Capture / Play
Capture / Forward
Edit Filters
Show All/None
Time: 00:00:57.258
Power Cycle Devices
PLAY CONTROLS:
Event List
Simulation
Routers
Scenario 0
New
Delete
In Progress
ICMP
(edit)
(delete)

LAB 11

To construct a WLAN and make the nodes communicate wirelessly

OBSERVATION:



Result

Ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data.

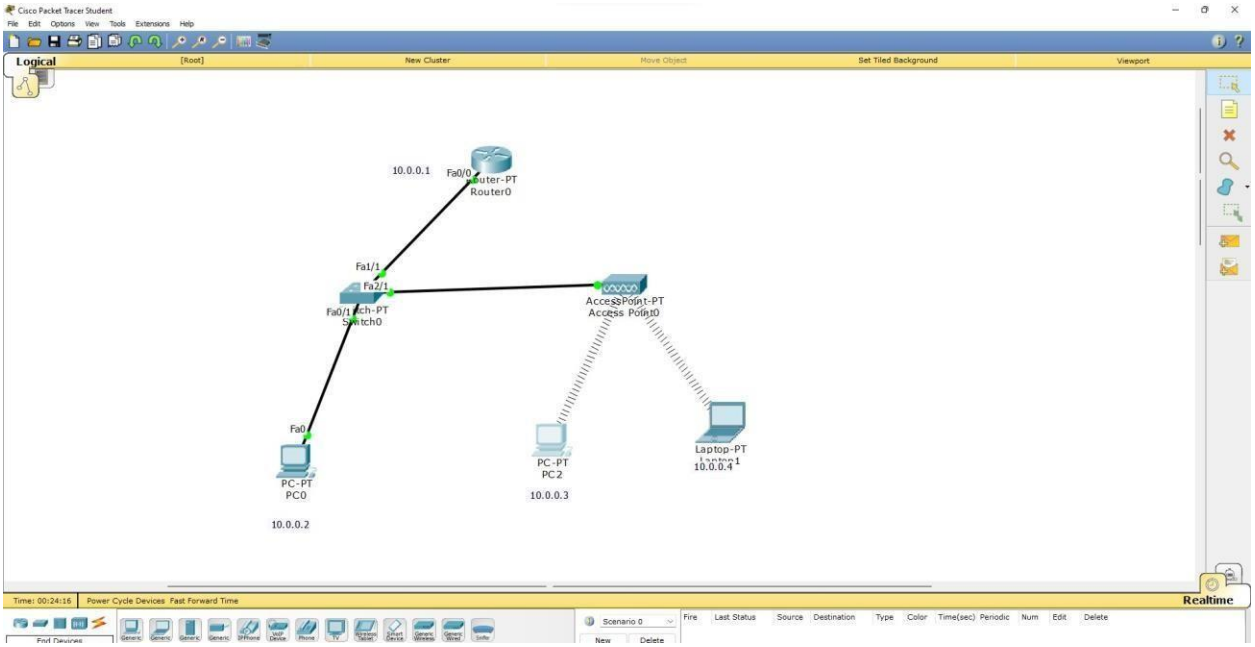
Reply from 10.0.0.3: bytes 32 time 0ms TTL 64

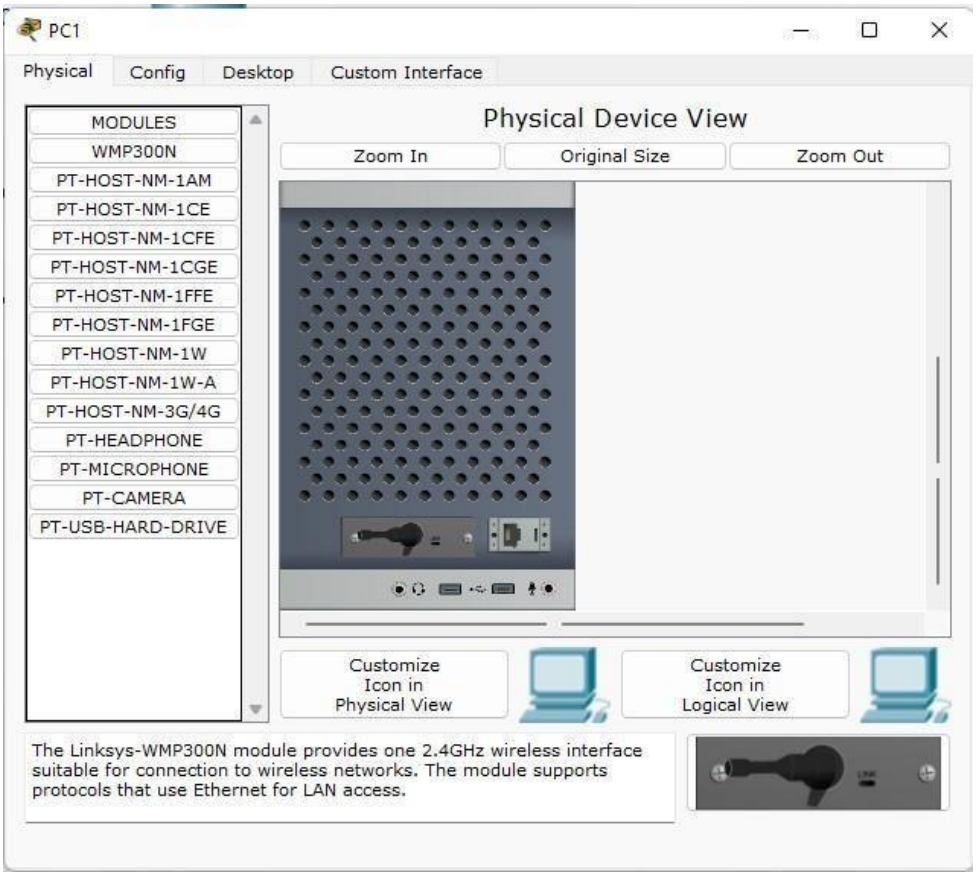
Ping statistics for 10.0.0.3

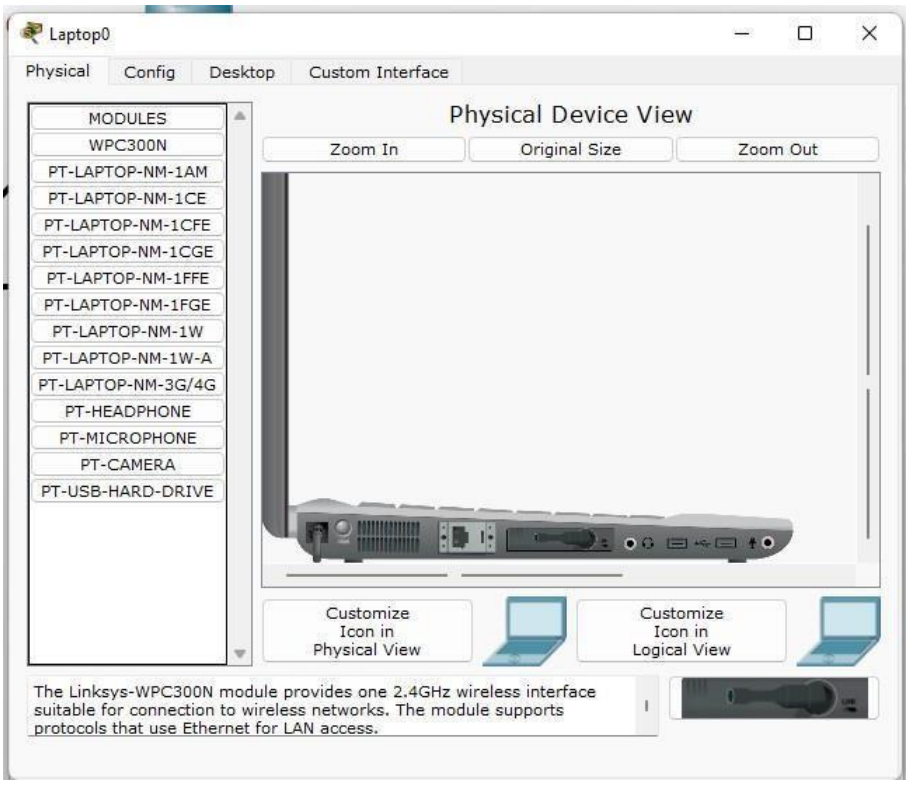
Packets sent = 4 Received = 4 Lost 0 (0%

8/9

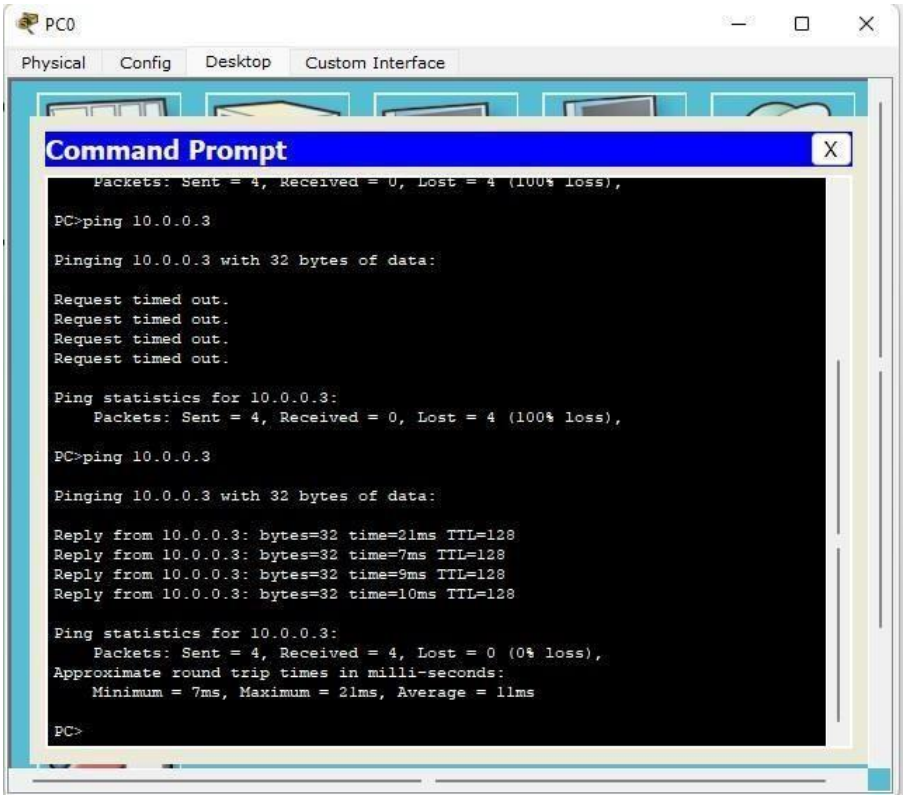
TOPOLOGY:


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


PC1
Physical
Config
Desktop
Custom Interface
MODULES
WMP300N
PT-HOST-NM-1AM
PT-HOST-NM-1CE
PT-HOST-NM-1CFE
PT-HOST-NM-1CGE
PT-HOST-NM-1FFE
PT-HOST-NM-1FGE
PT-HOST-NM-1W
PT-HOST-NM-1W-A
PT-HOST-NM-3G/4G
PT-HEADPHONE
PT-MICROPHONE
PT-CAMERA
PT-USB-HARD-DRIVE
Physical Device View
Zoom In
Original Size
Zoom Out
Customize Icon in Physical View
Customize Icon in Logical View
The Linksys-WMP300N module provides one 2.4GHz wireless interface suitable for connection to wireless networks. The module supports protocols that use Ethernet for LAN access.


Laptop0
Physical
Config
Desktop
Custom Interface
MODULES
WPC300N
PT-LAPTOP-NM-1AM
PT-LAPTOP-NM-1CE
PT-LAPTOP-NM-1CFE
PT-LAPTOP-NM-1CGE
PT-LAPTOP-NM-1FFE
PT-LAPTOP-NM-1FGE
PT-LAPTOP-NM-1W
PT-LAPTOP-NM-1W-A
PT-LAPTOP-NM-3G/4G
PT-HEADPHONE
PT-MICROPHONE
PT-CAMERA
PT-USB-HARD-DRIVE
Physical Device View
Zoom In
Original Size
Zoom Out
Customize Icon in Physical View
Customize Icon in Logical View
The Linksys-WPC300N module provides one 2.4GHz wireless interface suitable for connection to wireless networks. The module supports protocols that use Ethernet for LAN access.

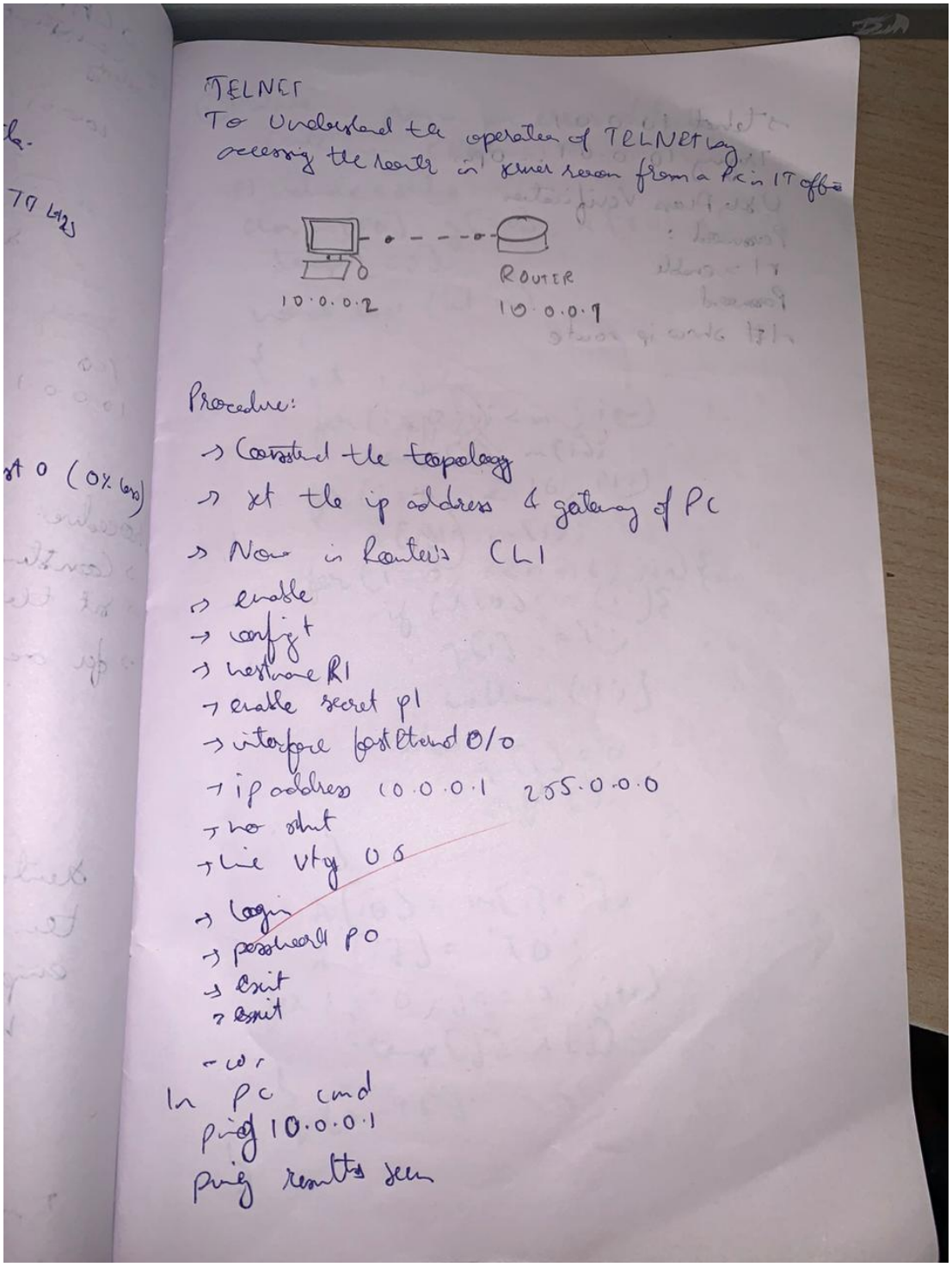
OUTPUT:

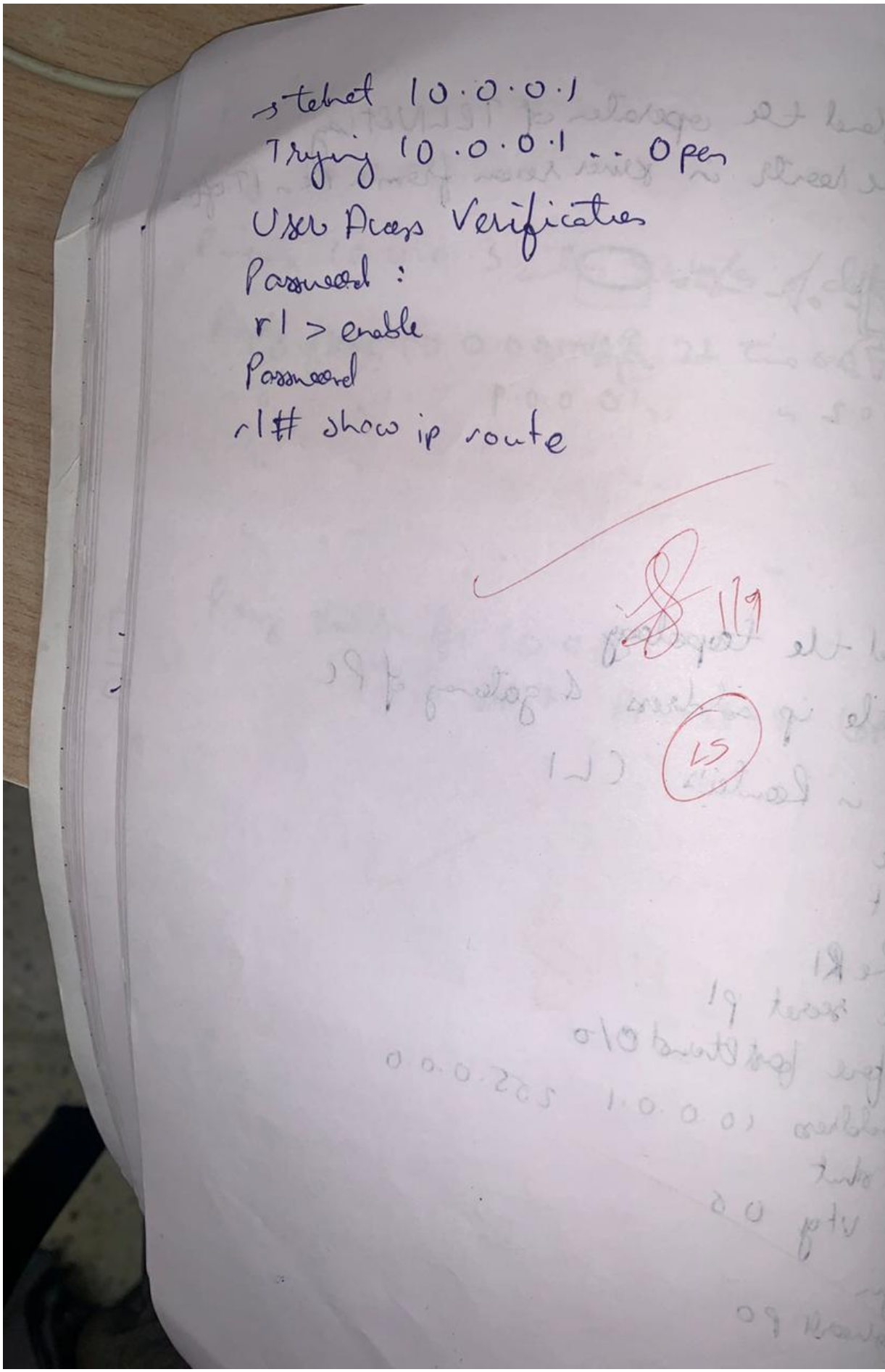

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 10.0.0.3:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data:
Reply from 10.0.0.3: bytes=32 time=21ms TTL=128
Reply from 10.0.0.3: bytes=32 time=7ms TTL=128
Reply from 10.0.0.3: bytes=32 time=9ms TTL=128
Reply from 10.0.0.3: bytes=32 time=10ms TTL=128
Ping statistics for 10.0.0.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 7ms, Maximum = 21ms, Average = 11ms
PC>

LAB 12

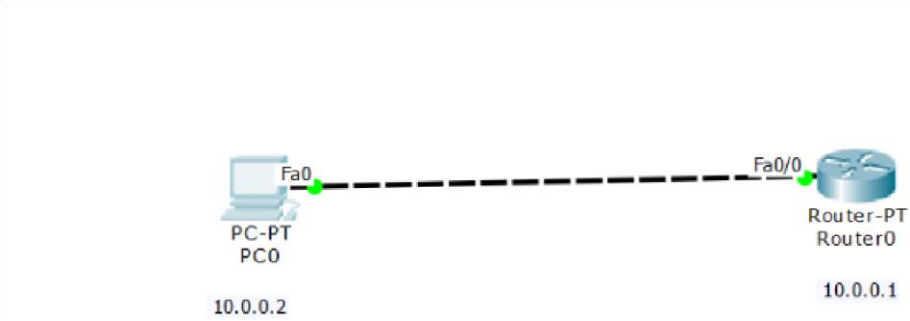
To understand the operation of TELNET by accessing the router in server room from a PC in IT office.

OBSERVATION:

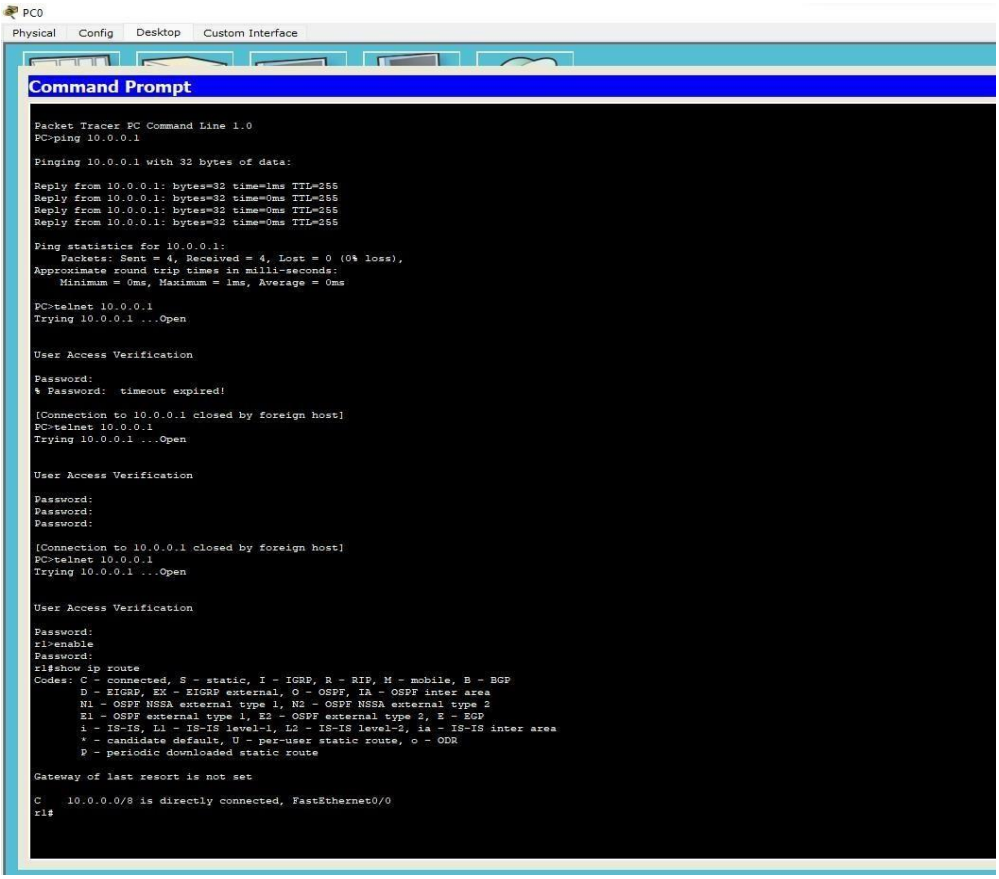



telnet 10.0.0.1
Trying 10.0.0.1 ... Open
User Acces Verificates
Password :
r1 > enable
Password
r1# show ip route

TOPOLOGY:



OUTPUT:



LAB 13

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

CODE:

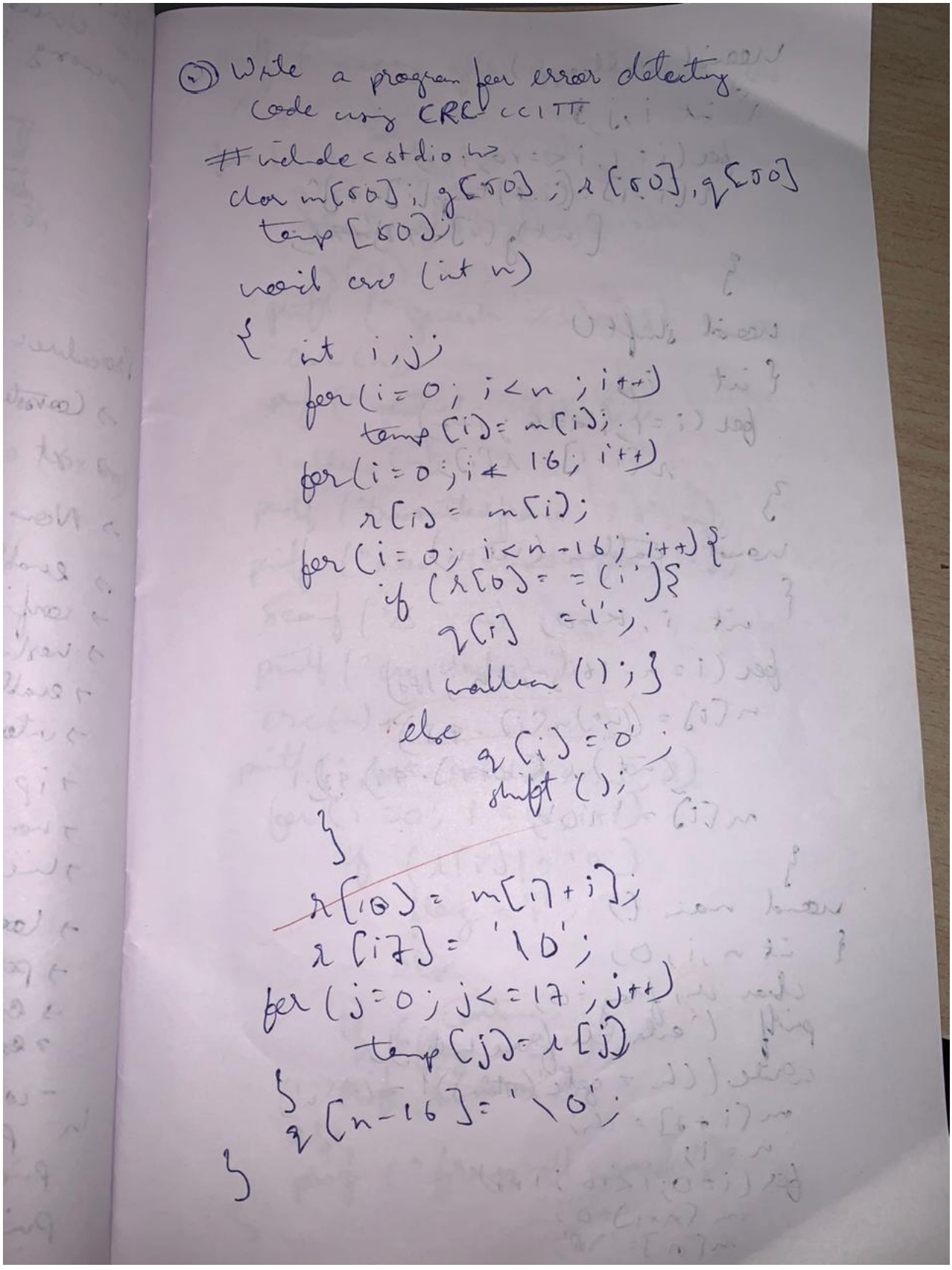
```
#include<stdio.h> int arr[17];

void xor(int x[], int y[])
{
    int k=0;
    for(int i=1;i<16;i++)
    { if(x[i]==y[i]) arr[k++]=0;
        else arr[i]=1;
    }
}

void main()
{ int dd[17],div[33],ze[17],i,k;

    printf("Enter the dataword \n");
    for(i=0;i<17;i++) scanf("%d",&div[i]);
    for(i=i;i<33;i++) div[i]=0;

    for(i=0;i<17;i++) ze[i]=0;
    printf("Enter dividend \n"); for(i=0;i<17;i++)
        scanf("%d",&dd[i]);
    i=0; k=0;
        for(i=i;i<17;i++)
            arr[k++]=div[i];
    while(i<33)
    { if(arr[0]==0) xor(arr,ze);
        else
```

```
            xor(arr,dd); arr[16]=div[i++];

    }
    k=0; for(i=17;i<33;i++) div[i]=arr[k++];
    printf("Codeword: "); for(i=0;i<33;i++)
    printf("%d",div[i]);


    for(i=0;i<17;i++) arr[i]=0; printf("\nAt receiver end

    \n");


     k=0;

        for(i=i;i<17;i++)

            arr[k++]=div[i];

    while(i<33)
    { if(arr[0]==0) xor(arr,ze); else xor(arr,dd);
        arr[16]=div[i++];

    }
    k=0;
    for(i=17;i<33;i++) div[i]=arr[k++]; printf("Codeword: ");

    for(i=0;i<33;i++) printf("%d",div[i]);

}
```

OUTPUT:

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

OBSERVATION:



(2) Write a program for error detecting code using CRC CCITT

```
#include<stdio.h>
char m[50], g[50], r[50], q[50]
  temp[50];
void crc (int n)
{  int i,j;
   for(i=0; i<n ; i++)
      temp[i]= m[i];
   for(i=0;i< 16; i++)
      r[i]= m[i];
   for(i=0; i<n-16; i++){
      if(r[0]==('1'){
         q[i] ='1';
         xor();}
      else q[i]='0';
         shift();
   }
   r[16]= m[17+i];
   r[17]= '10';
  for(j=0; j<=17 ; j++)
      temp[j]=r[j]
   }
   q[n-16]='\0';
}
```

```
void callen()
{ int i,j;
  for(i=1;i<=10;i++)
    a[i-1]=((int)temp[i]-48)^
           [(int)g[i]-48]+48;
}
void shiftU
{ int i;
  for(i=1;i<=16;i++)
      a[i-1]=a[i];
}
void callen(int n)
{ int i,k=0;
  for(i=n-16;i<n;i++)
    n[i]=((int)n[i]-48)^
         ((int)a[k++]-48)+48;
    n[i]='10';
}
void main()
{ int n,i,0;
  char ch,flag=0;
  printf("enter the freehosts");
  while((ch=getc(stdin))!='\n')
  n[i++]=ch;
    n=i;
  for(i=0;i<16;i++)
    n[n++]='0';
```

```
printf("message after appending 7 zeros
%s",m);
for(i=0;i<=10;i++)
    g[i]='0';
    g[0]=g[7]=g[10]=g[14]='1';
    g[17]='\0';
printf("generator: %s\n",g);
crc(m);
printf("quotient is %s",q);
xaltem(m);
printf("transmitted frame %s",m);
printf("after transmitted frame:");
scanf("%s",m);
printf("crc decoder\n");
crc(m);
printf("last remainder: %s",R);
for(i=0;i<10;i++)
    if(r[i]!='0')
        flag='1';
    else
        continue;
if(flag=='1')
printf("error");
else
printf("frame all correct");
}
```

Output

Enter frame bits : 1011

message after appendy 16 zero

1011 0000 0000 0000 0000

generate : 1000 1000 0001 00001

quotient : 1011

transmitted : 1011 1011 0001 0110[illegible]

after - transmitted frame

1011 1011 0001 0110 1011

Last remainder 0000 0000 0000 0000

Received frame is correct

Lab 14

Write a program for congestion control using Leaky bucket algorithm.

CODE:

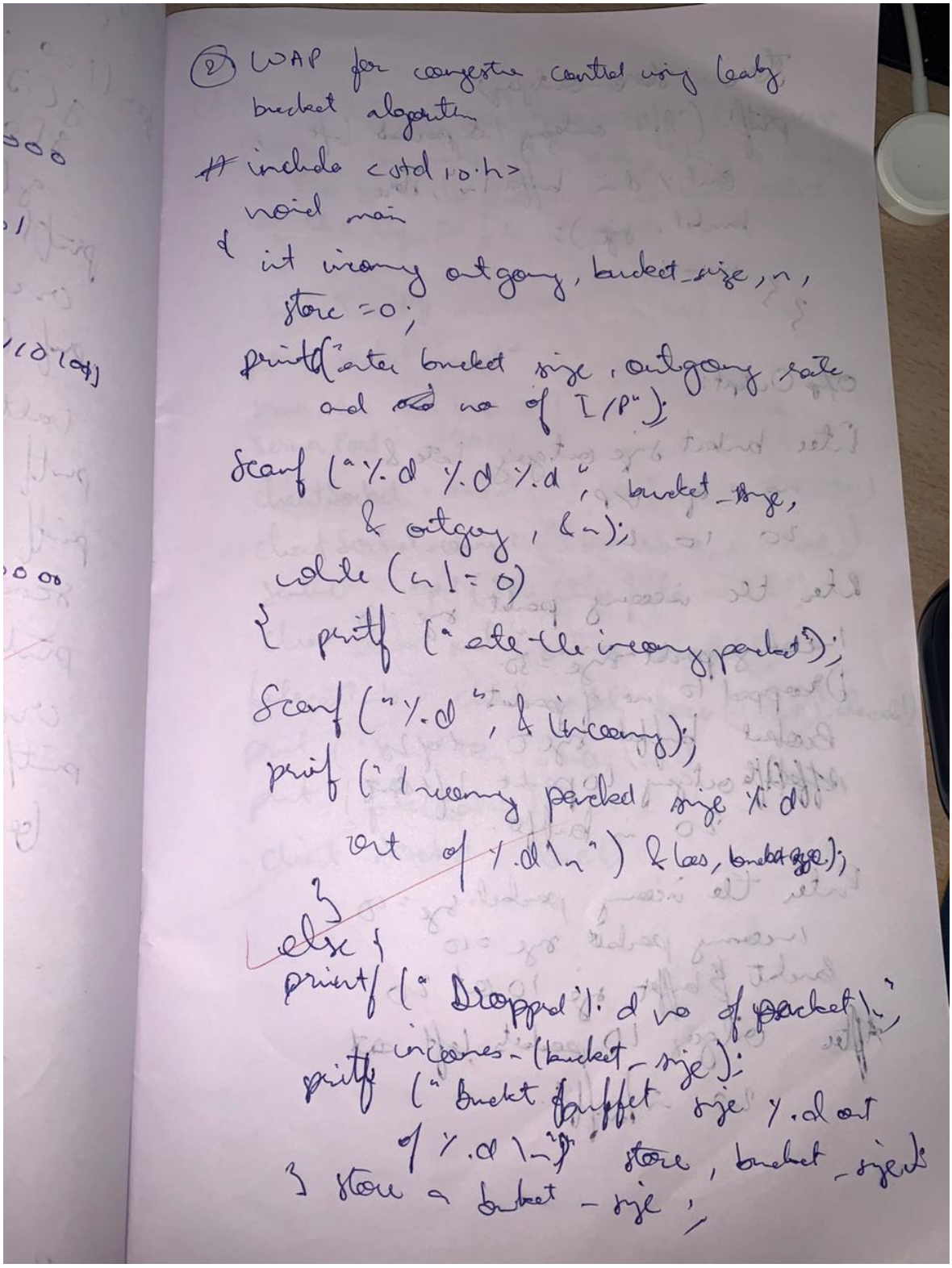
```
#include <stdio.h>
#include <stdlib.h> // Include this for the rand() function int main()
{
    int buckets, outlets, k = 1, num, remaining;
    printf("Enter Bucket size and outstream size\n"); scanf("%d %d", &buckets, &outlets);
    remaining = buckets; while (k)
    {
        num = rand() % 1000; // Generate a random number between 0 and 999 if (num < remaining)
        {
                                    remaining = remaining - num; printf("Packet of %d bytes
            accepted\n", num); // Added missing
variable
        }
        else
        {
            printf("Packet of %d bytes is discarded\n", num);
        }
        if (buckets - remaining > outlets)
        {
            remaining += outlets; // Fixed the calculation
        }
        else remaining = buckets; printf("Remaining
        bytes: %d \n", remaining); printf("If you want to stop input, press 0, otherwise, press
        1\n"); scanf("%d", &k);
    }
    while (remaining < buckets) // Fixed the condition
    {
        if (buckets - remaining > outlets)
```

```
        {
            remaining += outlets; // Fixed the calculation
        }
        else remaining = buckets;
        printf("Remaining bytes: %d \n", remaining);
    }
                                    return 0; // Added a return statement to indicate successful completion
}
```

OUTPUT:

```
PS D:\VS Code> cd "d:\VS Code\OS\" ; if ($?) { gcc bucket.c -o bucket } ; if ($?) { .\bucket }
Enter Bucket size and outstream size
2000
100
Packet of 41 bytes accepted
Remaining bytes: 2000
If you want to stop input, press 0, otherwise, press 1
1
Packet of 467 bytes accepted
Remaining bytes: 1633
If you want to stop input, press 0, otherwise, press 1
1
Packet of 334 bytes accepted
Remaining bytes: 1399
If you want to stop input, press 0, otherwise, press 1
1
Packet of 500 bytes accepted
Remaining bytes: 999
If you want to stop input, press 0, otherwise, press 1
1
Packet of 169 bytes accepted
Remaining bytes: 930
If you want to stop input, press 0, otherwise, press 1
1
Packet of 724 bytes accepted
Remaining bytes: 306
If you want to stop input, press 0, otherwise, press 1
1
Packet of 478 bytes is discarded
Remaining bytes: 406
If you want to stop input, press 0, otherwise, press 1
1
Packet of 358 bytes accepted
Remaining bytes: 148
If you want to stop input, press 0, otherwise, press 1
1
Packet of 962 bytes is discarded
Remaining bytes: 248
If you want to stop input, press 0, otherwise, press 1
0
Remaining bytes: 348
Remaining bytes: 448
Remaining bytes: 548
Remaining bytes: 648
Remaining bytes: 748
Remaining bytes: 348
Remaining bytes: 448
Remaining bytes: 548
Remaining bytes: 648
Remaining bytes: 748
Remaining bytes: 848
Remaining bytes: 948
Remaining bytes: 1048
Remaining bytes: 1148
Remaining bytes: 1248
Remaining bytes: 1348
Remaining bytes: 1448
Remaining bytes: 1548
Remaining bytes: 1648
Remaining bytes: 1748
Remaining bytes: 1848
Remaining bytes: 1948
Remaining bytes: 2000
PS D:\VS Code\OS>
```

OBSERVATION:



(2) WAP for congestion control using leaky bucket algorithm

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

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

Output:

Enter bucket size outgoing rate &
no of I/P
20 10 2

Enter the incoming packet size = 8
Incoming packet size = 30
Dropped 10 no of packet
Bucket buffer size 0 out of 20
After outgoing 10 packet left out
20 in buffer.

Enter the incoming packet size = 40
Incoming packet size = 40
Bucket buffer size 10 out 20
After outgoing 10 packets left out
20 in buffer.

WEEK15

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

CODE:

ClientTCP.py from socket import * serverName =

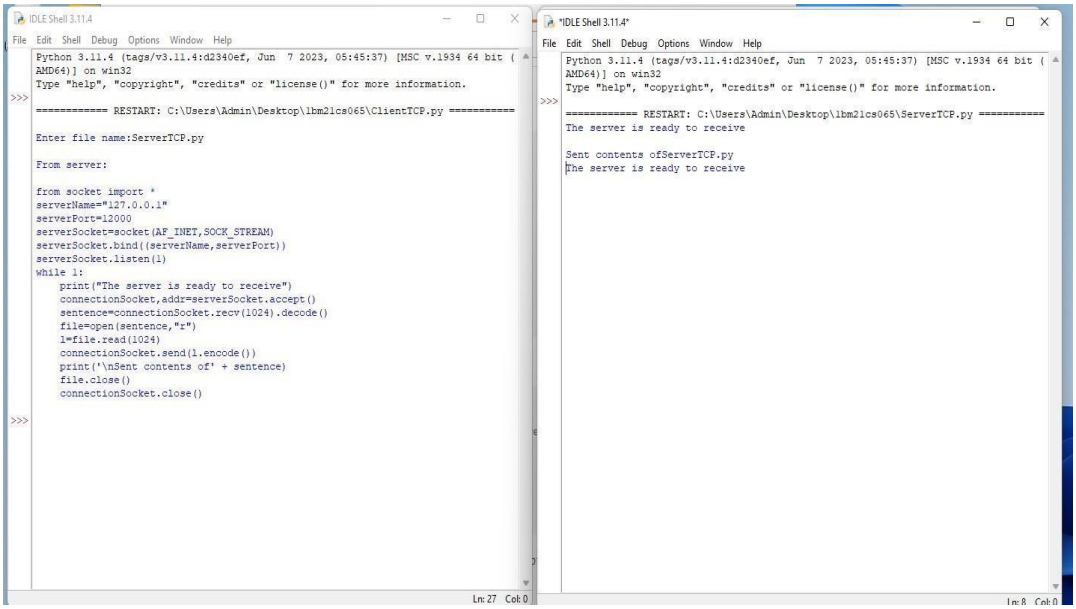
“127.0.0.1” serverPort = 12000 clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort)) sentence = input(“\nEnter file
name: “) clientSocket.send(sentence.encode()) filecontents =
clientSocket.recv(1024).decode() print (“\nFrom Server:\n”) print(filecontents)
clientSocket.close()

ServerTCP.py from socket import * serverName=”127.0.0.1” serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort)) serverSocket.listen(1) while 1: print
(“The server is ready to receive”)

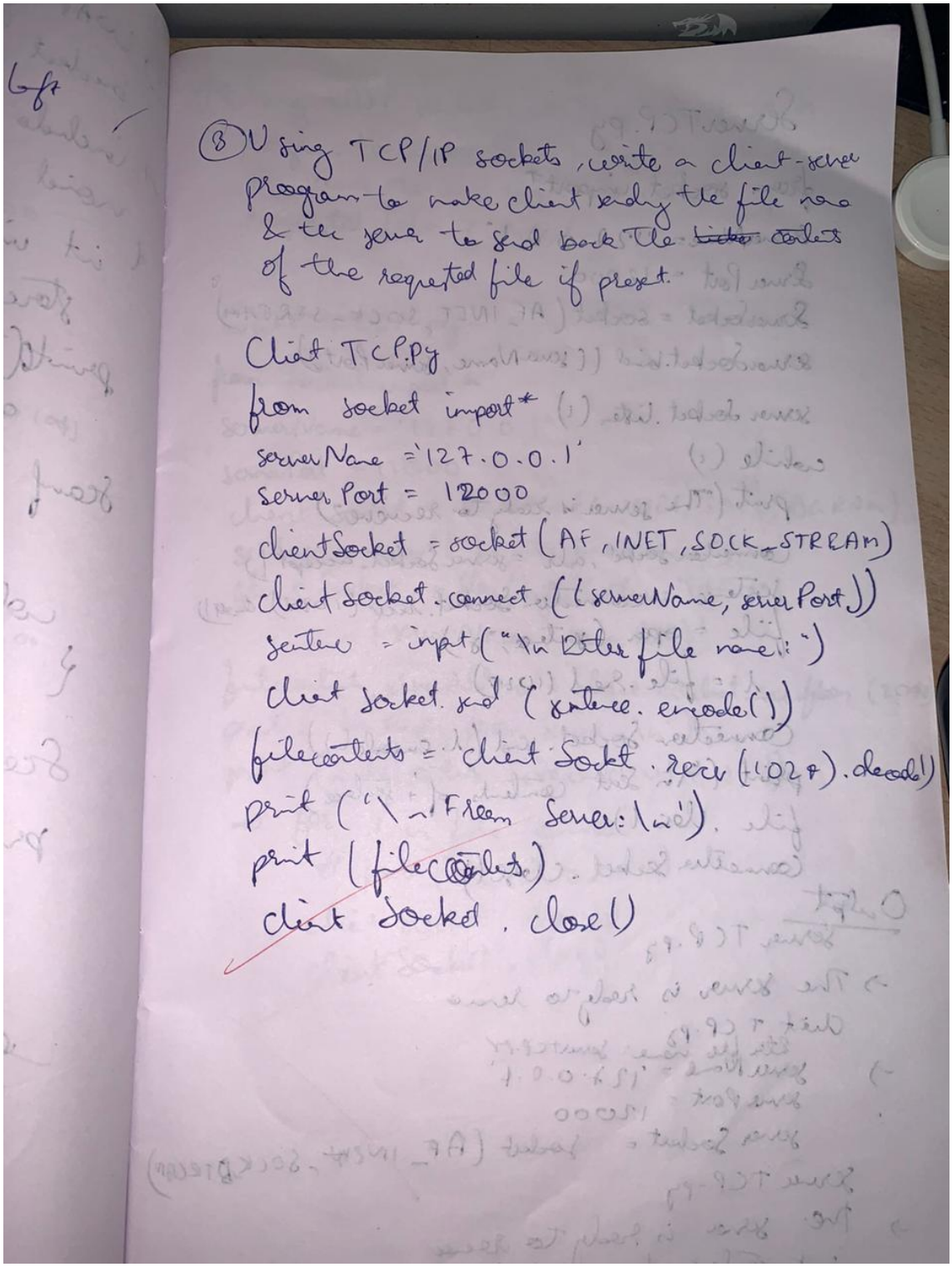
connectionSocket, addr = serverSocket.accept() sentence =
connectionSocket.recv(1024).decode() file=open(sentence,”r”) l=file.read(1024)
connectionSocket.send(l.encode()) print (“\nSent contents of “ + sentence)

file.close() connectionSocket.close()

OUTPUT:


IDLE Shell 3.11.4
File Edit Shell Debug Options Window Help
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\Admin\Desktop\1bm21cs065\ClientTCP.py ==========
Enter file name:ServerTCP.py
From server:
from socket import *
serverName="127.0.0.1"
serverPort=12000
serverSocket=socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
print("The server is ready to 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()
>>>
Ln: 27 Col: 0
IDLE Shell 3.11.4
File Edit Shell Debug Options Window Help
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\Admin\Desktop\1bm21cs065\ServerTCP.py ==========
The server is ready to receive
Sent contents ofServerTCP.py
The server is ready to receive
Ln: 8 Col: 0

OBSERVATION:



(8) Using TCP/IP sockets, write a client-server program to make client send the file name & 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()
```

ServerTCP.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREA
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
while (1)
    print("The server is ready to recieve")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).
    file = open(sentence, "r")
    l = file.read(1024)
    connectionSocket.send(l.encode(1))
    print("\n Sent contents of" + sentence)
    file.close()
    connectionSocket.close()
```

Output

ServerTCP.py

→ The server is ready to receive

clientTCP.py

Enter file name: serverTCP.py

→ serverName = '127.0.0.1'
serverPort = 12000
serverSocket = socket(AF_INET, SOCK

ServerTCP.py

→ The server is ready to receive

LAB 16

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

CODE:

ClientUDP.py

from socket import * serverName = “127.0.0.1” serverPort = 12000 clientSocket = socket(AF_INET, SOCK_DGRAM) sentence = input(“\nEnter file name: “) clientSocket.sendto(bytes(sentence,”utf8”),(serverName, serverPort)) filecontents,serverAddress = clientSocket.recvfrom(2048) print (“\nReply from Server:\n”) print (filecontents.decode(“utf-8”)) # for i in filecontents: # print(str(i), end = “

”) clientSocket.close() clientSocket.close() ServerUDP.py from socket import * serverPort =

12000

serverSocket = socket(AF_INET, SOCK_DGRAM)

serverSocket.bind((“127.0.0.1”, serverPort)) print (“The server is ready to

receive”) while 1:

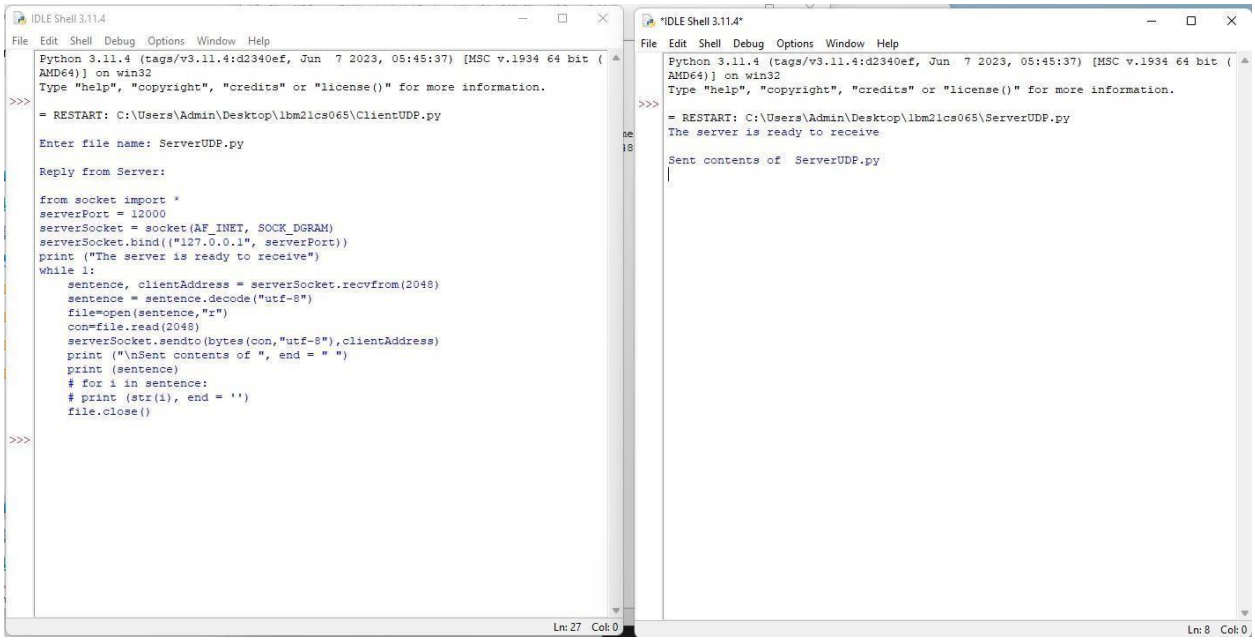
sentence, clientAddress = serverSocket.recvfrom(2048) sentence

= sentence.decode(“utf-8”) file=open(sentence,”r”) con=file.read(2048) serverSocket.sendto(bytes(con,”utf-

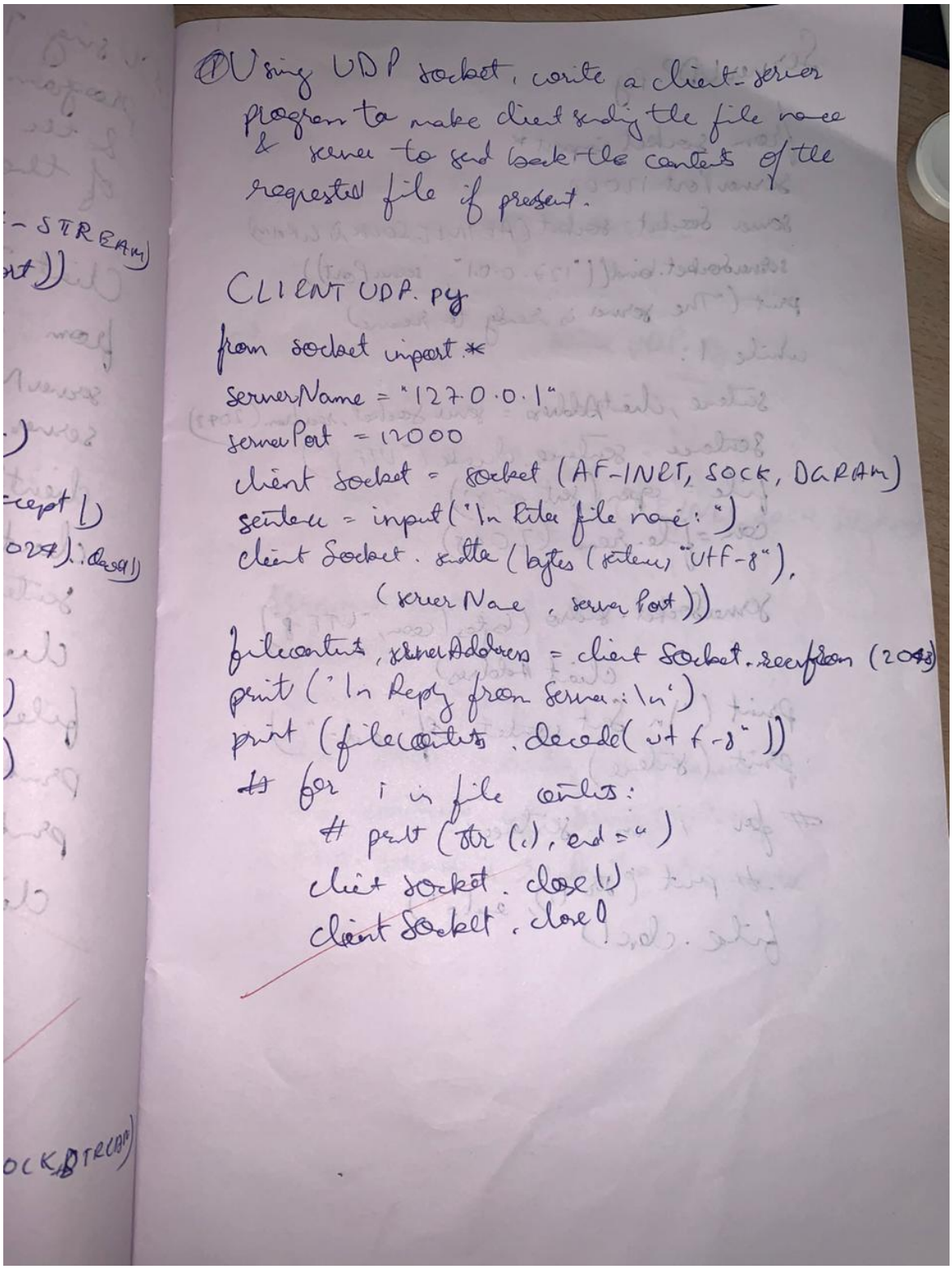
8”),clientAddress) print (“\nSent contents of “, end = “ “) print

(sentence) # for i in sentence: # print (str(i), end = “ “) file.close()

OUTPUT:


IDLE Shell 3.11.4
File Edit Shell Debug Options Window Help
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Admin\Desktop\1bm21cs065\ClientUDP.py
Enter file name: ServerUDP.py
Reply from Server:
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 1:
sentence, clientAddress = serverSocket.recvfrom(2048)
sentence = sentence.decode("utf-8")
file=open(sentence,"r")
con=file.read(2048)
serverSocket.sendto(bytes(con,"utf-8"),clientAddress)
print ("\nSent contents of ", end = " ")
print (sentence)
# for i in sentence:
# print (str(i), end = '')
file.close()
>>>
Ln: 27 Col: 0
IDLE Shell 3.11.4
File Edit Shell Debug Options Window Help
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Admin\Desktop\1bm21cs065\ServerUDP.py
The server is ready to receive
Sent contents of ServerUDP.py
Ln: 8 Col: 0

OBSERVATION:



(1) Using UDP socket, write a client-server program to make client sending the file name & server to send back the contents of the requested file if present.

CLIENT UDP.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("\n Enter file name: ")
clientSocket.sendto(bytes(sentence, "utf-8"),
                    (serverName, serverPort))
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()
```

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(2
    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()
```

Output

~~Client~~

Server UDP.py

→ The server is ready to Receive

Client UDP.py

→ Enter file name: Server UDP.py

Reply from server:

Server Port = 12000

server Socket = socket (AF_INET, SOCK_DGRAM)

~~Server UDP.py~~

~~Client~~

Server UDP.py

→ The server is ready to Receive

The contents of server UDP.py

The server is ready to receive

8 1/9