VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

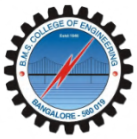
LAB REPORT
on

COMPUTER NETWORKS

Submitted by

VARUN RAJ S (1BM21CS264)

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



B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
JUN-2023 to SEP-2023

B. M. S. College of Engineering,
Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "COMPUTER NETWORKS" carried out by **VARUN RAJ S (1BM21CS264),** 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

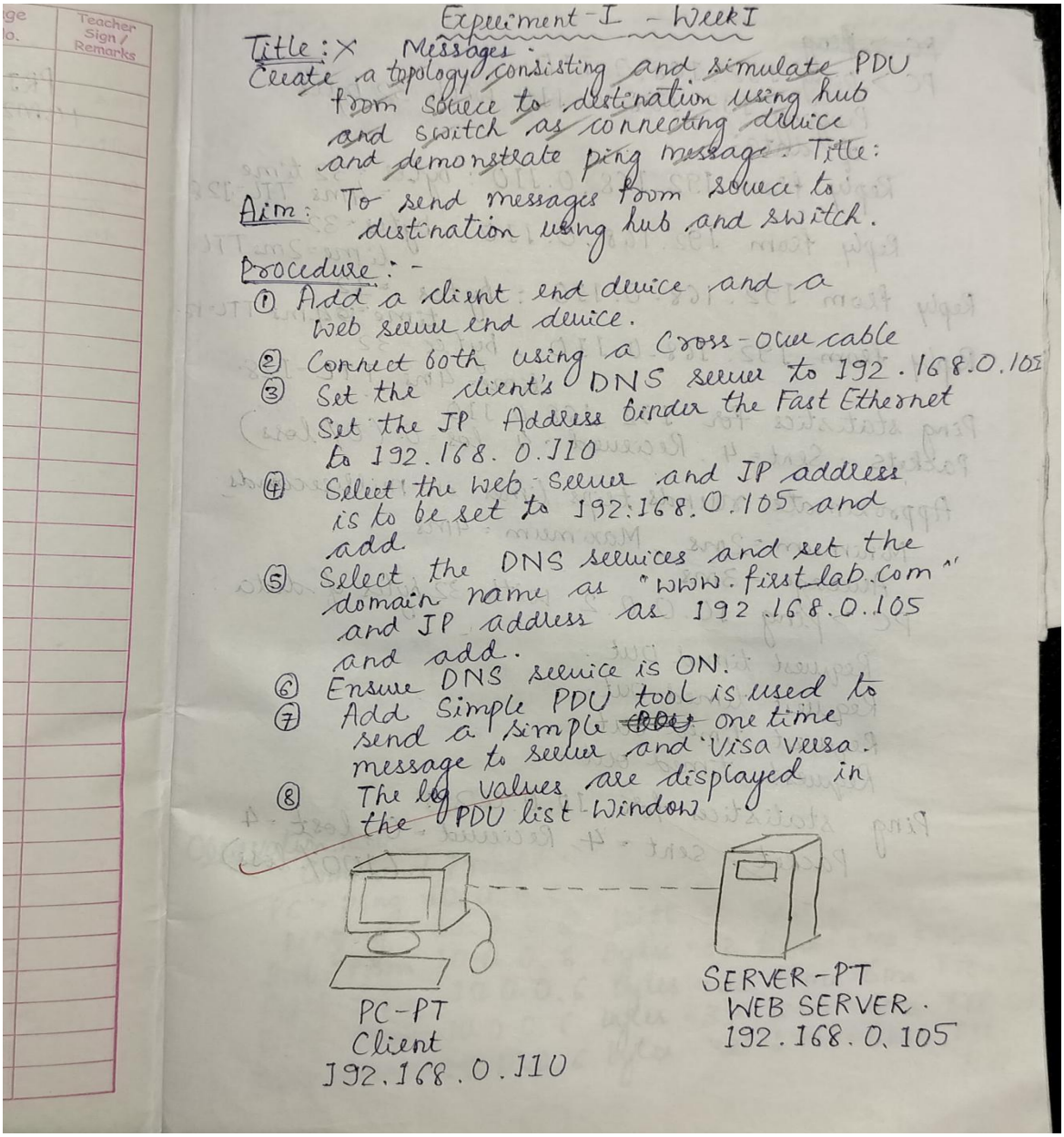
`

Index

WEEK 1

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

OBSERVATION



Experiment - I - Week I

Title: X Messages

Create a topology consisting and simulate PDU from source to destination using hub and switch as connecting device and demonstrate ping message. Title:

Aim: To send messages from source to destination using hub and switch.

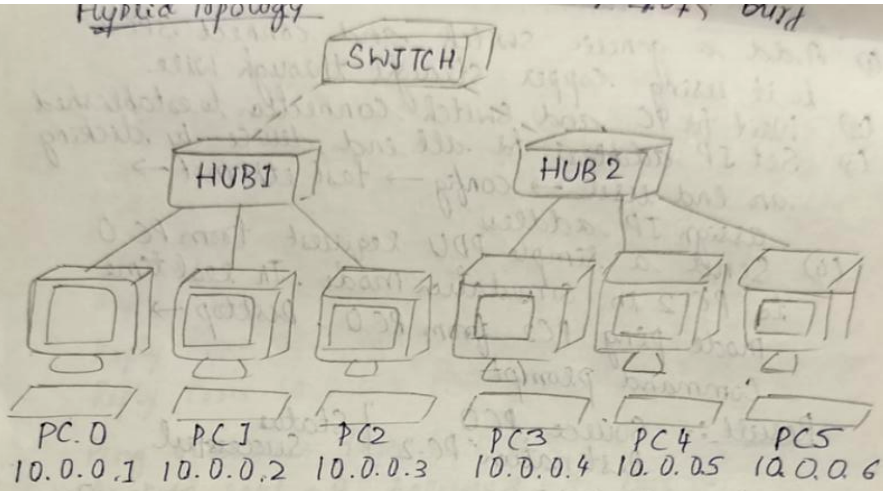
Procedure: -

1. Add a client end device and a web server end device.
2. Connect both using a Cross-over cable
3. Set the client's DNS server to 192.168.0.105 Set the IP Address Under the Fast Ethernet to 192.168.0.110
4. Select the web server and IP address is to be set to 192.168.0.105 and add
5. Select the DNS services and set the domain name as "www.first_lab.com" and IP address as 192.168.0.105 and add.
6. Ensure DNS service is ON.
7. Add Simple PDU tool is used to send a simple ~~PDU~~ one time message to server and Visa Versa.
8. The log values are displayed in the PDU list Window

PC-PT
Client
192.168.0.110

SERVER-PT
WEB SERVER.
192.168.0.105

Hybrid topology



Procedure

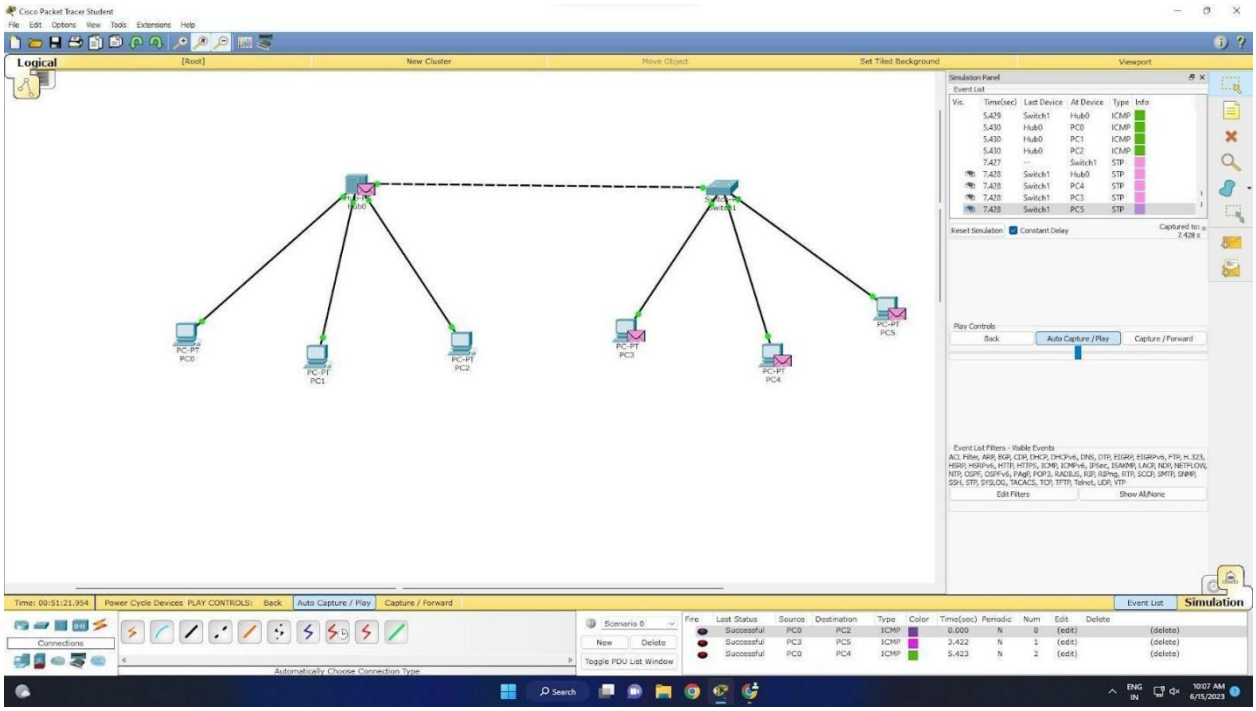
(i) Add generic hub, generic switch and connect 3 end devices. here PC
(ii) Connect end devices to the two different hubs the connect 2 hubs to a switch using copper cross over wire
(iii) Click on end device → config → fast ethernet → assign IP address.
(iv) Send PDU from source device to destination and wait for simulation to finish.
(v) In real time mode ping devices using command prompt.

Result

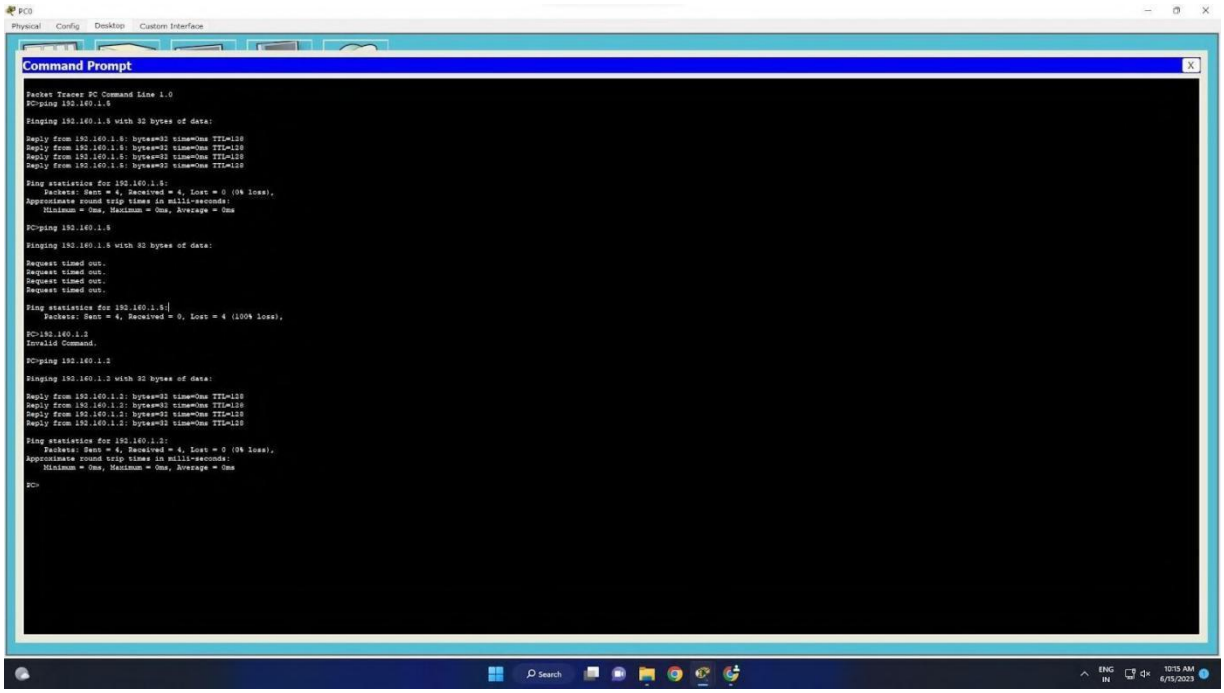
Source PC0
Destination PC5
] Status Successful.

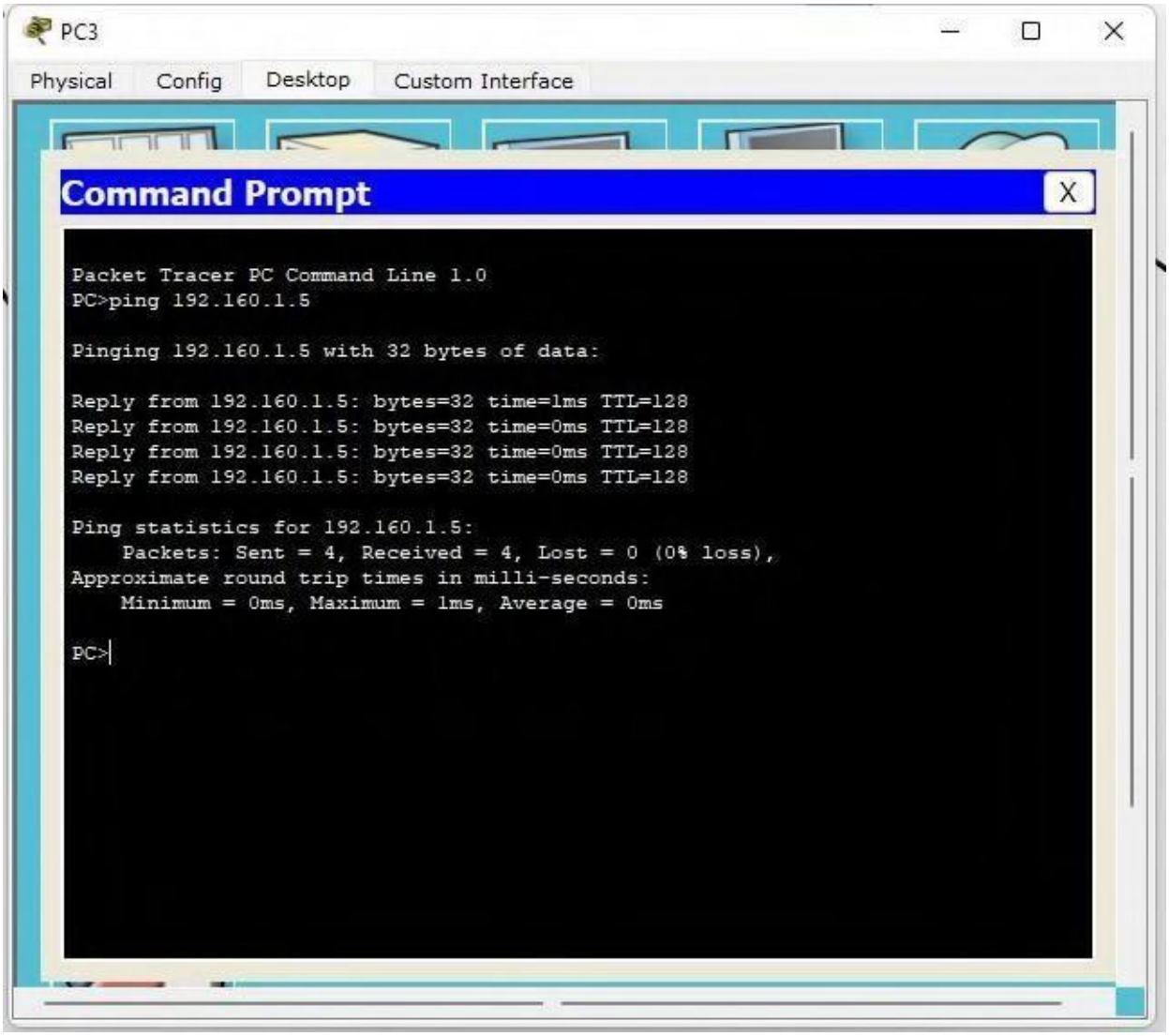
Ping 10.0.0.6
Reply from 10.0.0.6 byte=32 time=0ms TTL=128
Reply from 10.0.0.6 byte=32 time=0m TTL=128
Reply from 10.0.0.6 byte=32 time=0m TTL=128
Reply from 10.0.0.6 byte=32 time=0ms TTL=12

TOPOLOGY:



OUTPUT:

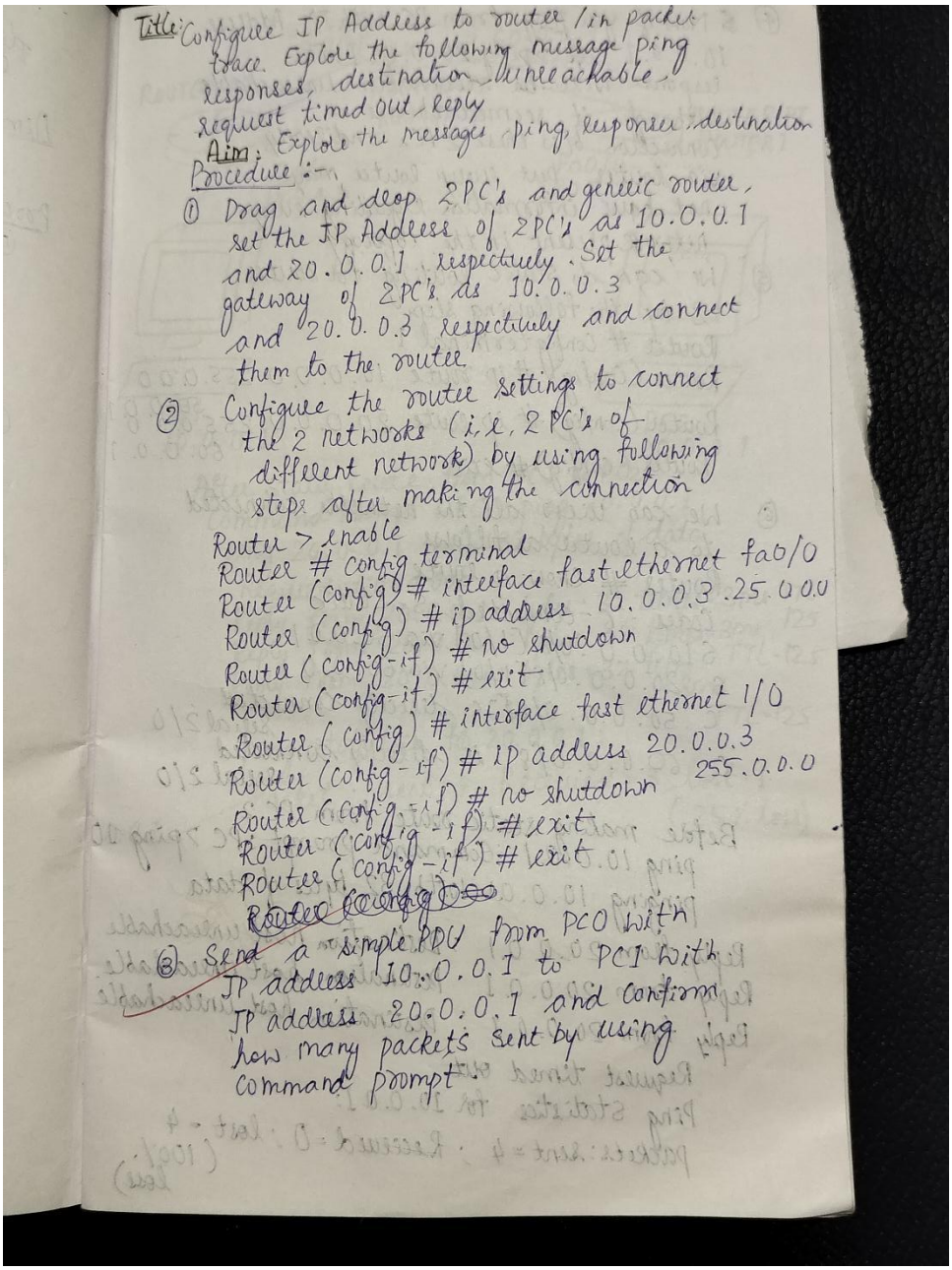



PC3
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 192.160.1.5
Pinging 192.160.1.5 with 32 bytes of data:
Reply from 192.160.1.5: bytes=32 time=1ms TTL=128
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
Reply from 192.160.1.5: bytes=32 time=0ms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

WEEK 2

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

OBSERVATION:



Title: Create a topology consisting of 2 devices connected with the help of a router.

Aim :- Connecting 2 devices with the help of a router.

Procedure :-

① Drag and drop 2 generic PC's and a router. Connect 2 PC's as per phesical to router after setting the IP addresses as 10.0.0.1 and 20.0.0.1 for PC_1 and PC_2 respectively and connect them.

② Configure router → enable → Configure terminal → interface fa0/0 → IP address (PC1 [subnet mask] → no shutdown

Repeat this for PC_2.

③ Ping from PC_1 to PC_2 and note the Observation give the gateway and ping again and noted down observation

Observation :-

PC > ping 10.0.0.1

ping 10.0.0.1 with 32 bytes of data

Request timed out.

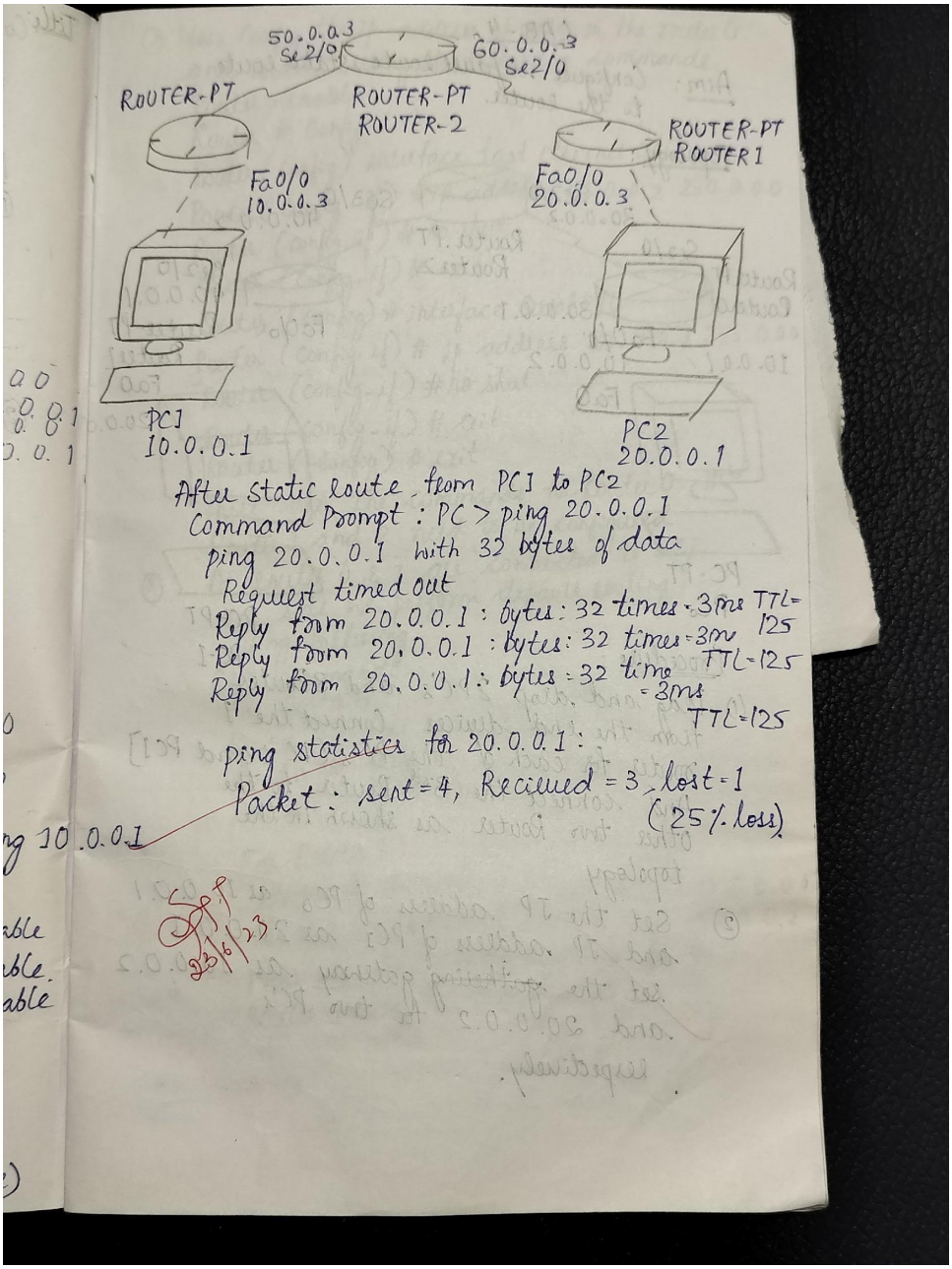
Reply from 10.0.0.1 : bytes = 32 time = 3ms TTL = 125

Reply from 10.0.0.1 : bytes : 32 time = 3ms TTL = 125

Reply from 10.0.0.1 : bytes : 32 time : 3ms TTL = 125

ping statistics for 10.0.0.1

packets sent = 4, Recieved = 2 lost = 2 (50% lose)



After static route from PC1 to PC2

Command Prompt : PC > ping 20.0.0.1

ping 20.0.0.1 with 32 bytes of data

Request timed out

Reply from 20.0.0.1 : bytes: 32 times=3ms TTL=125

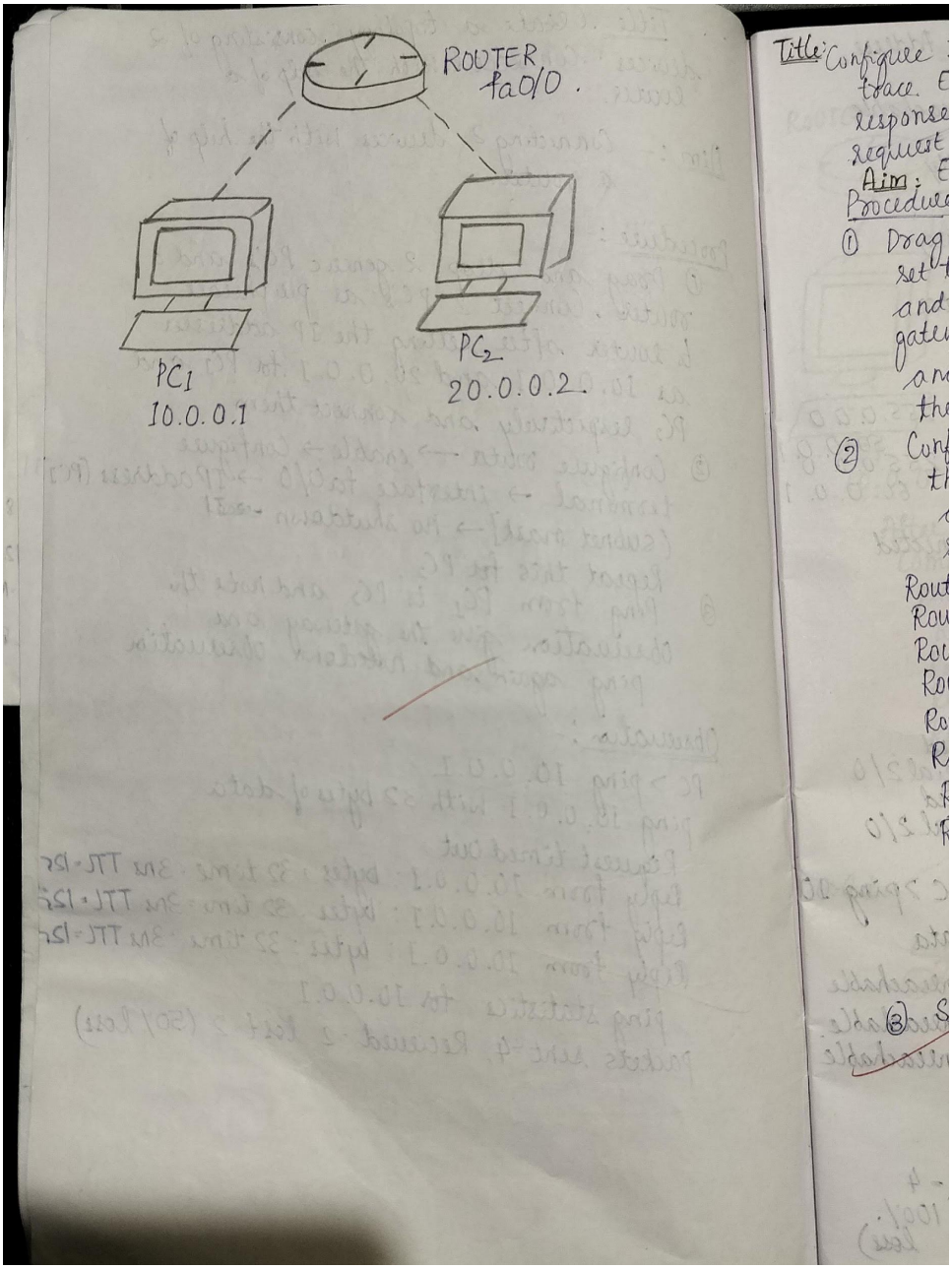
Reply from 20.0.0.1 : bytes: 32 times=3ms TTL=125

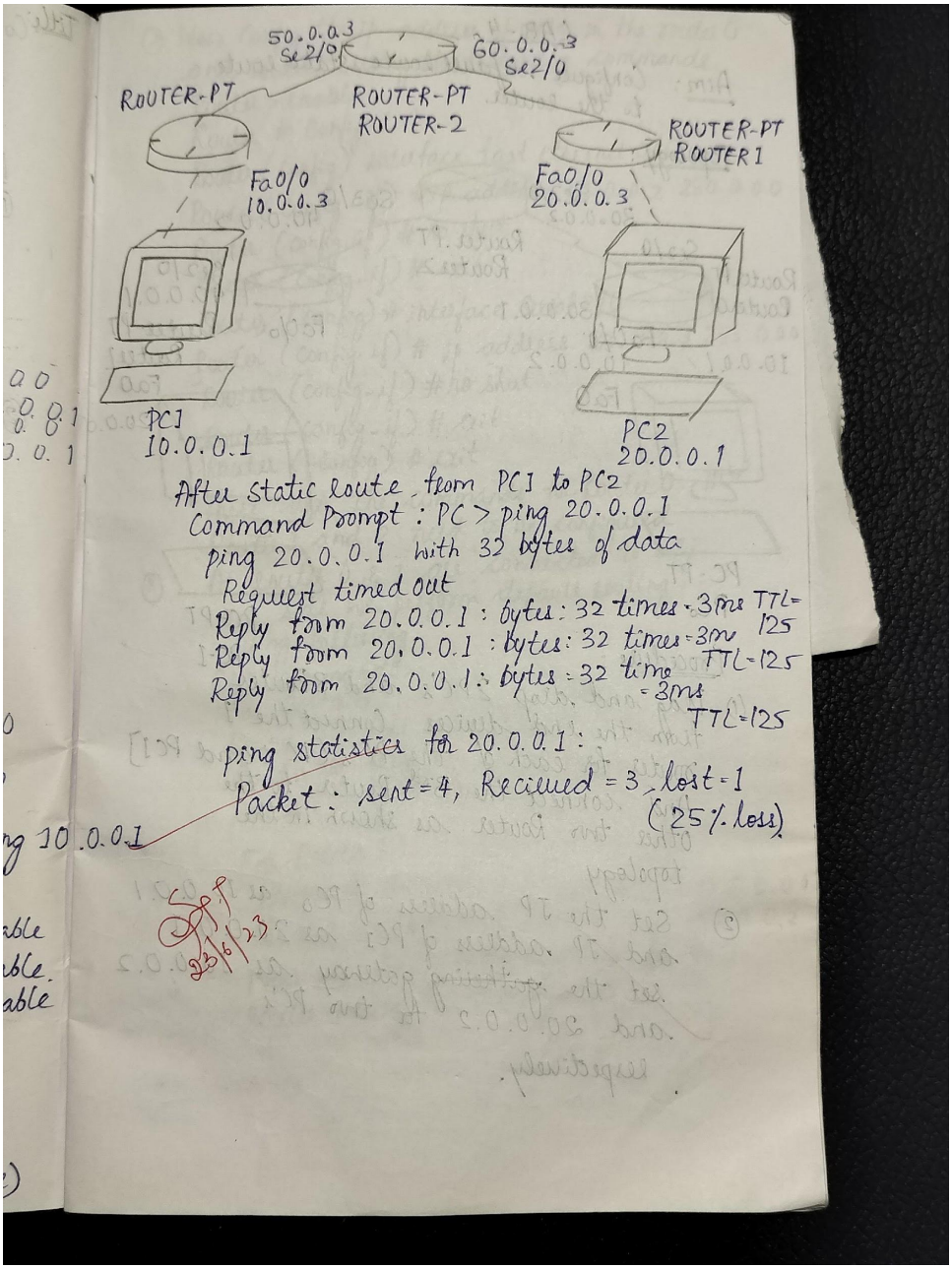
Reply from 20.0.0.1 : bytes=32 time=3ms TTL=125

ping statistics for 20.0.0.1 :

Packet : sent=4, Recieved=3, lost=1 (25% loss)

23/6/23


ROUTER
fa0/0.
PC1
10.0.0.1
PC2
20.0.0.2



After static route from PC1 to PC2

Command Prompt : PC > ping 20.0.0.1

ping 20.0.0.1 with 32 bytes of data

Request timed out

Reply from 20.0.0.1 : bytes: 32 times=3ms TTL=125

Reply from 20.0.0.1 : bytes: 32 times=3ms TTL=125

Reply from 20.0.0.1 : bytes=32 time=3ms TTL=125

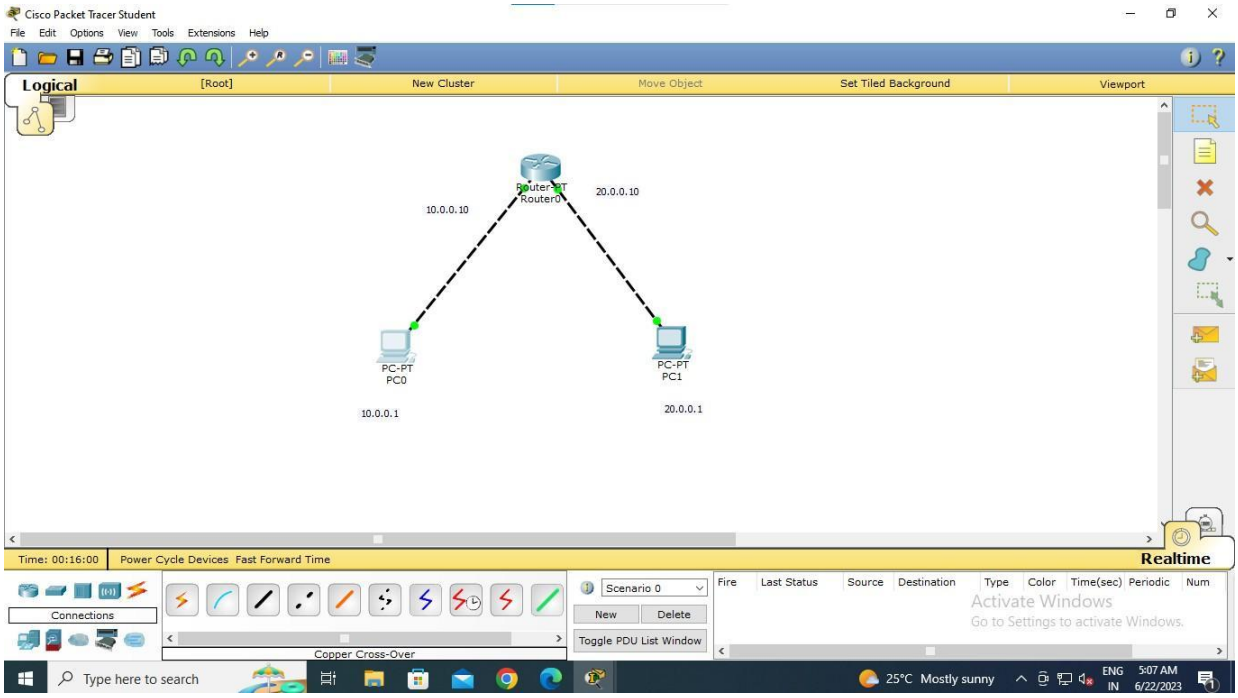
ping statistics for 20.0.0.1 :

Packet : sent=4, Recieved=3, lost=1 (25% loss)

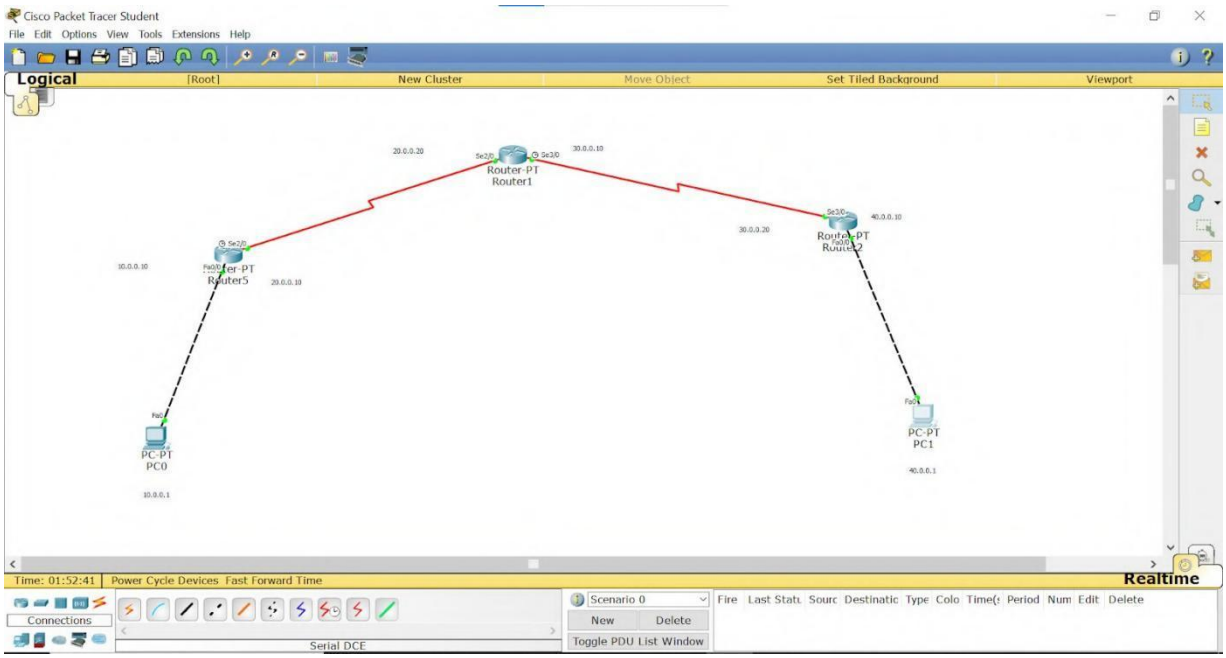
23/6/23

TOPOLOGY:

PROGRAM 2.1

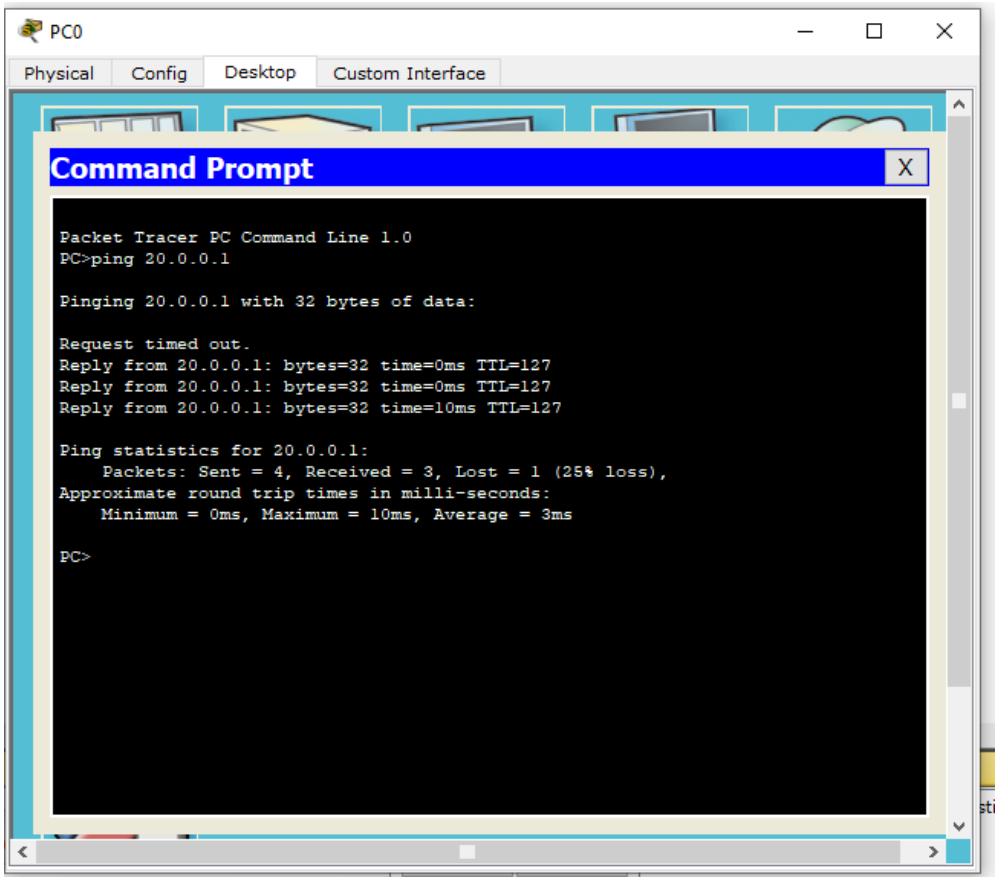


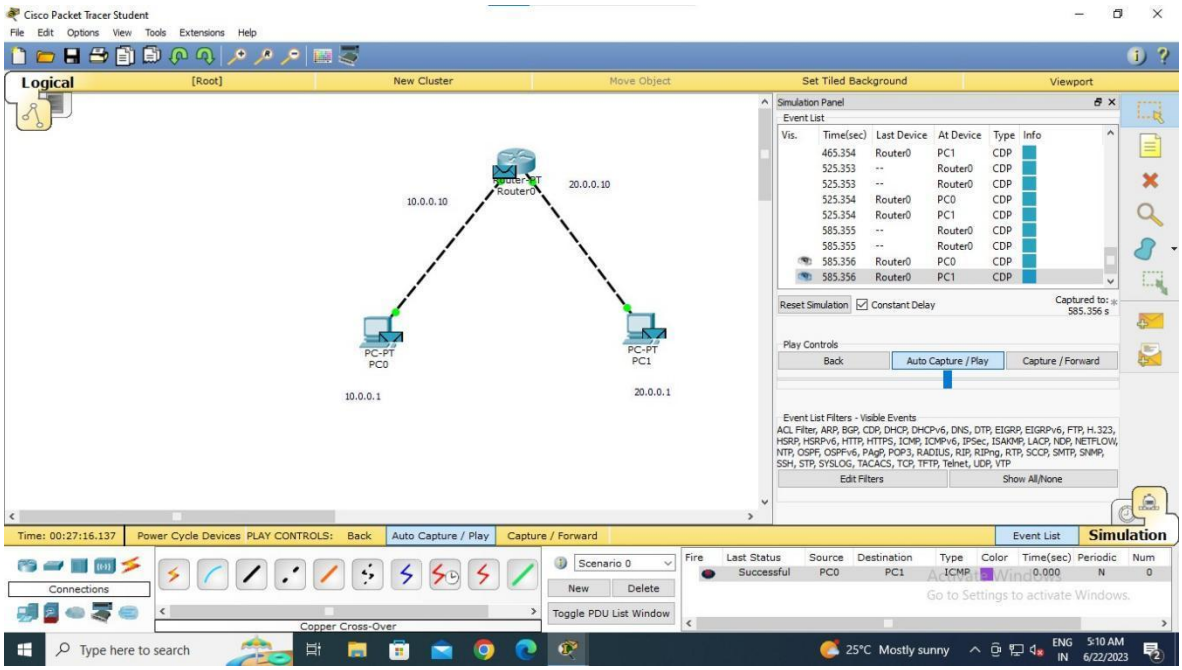
PROGRAM 2.2



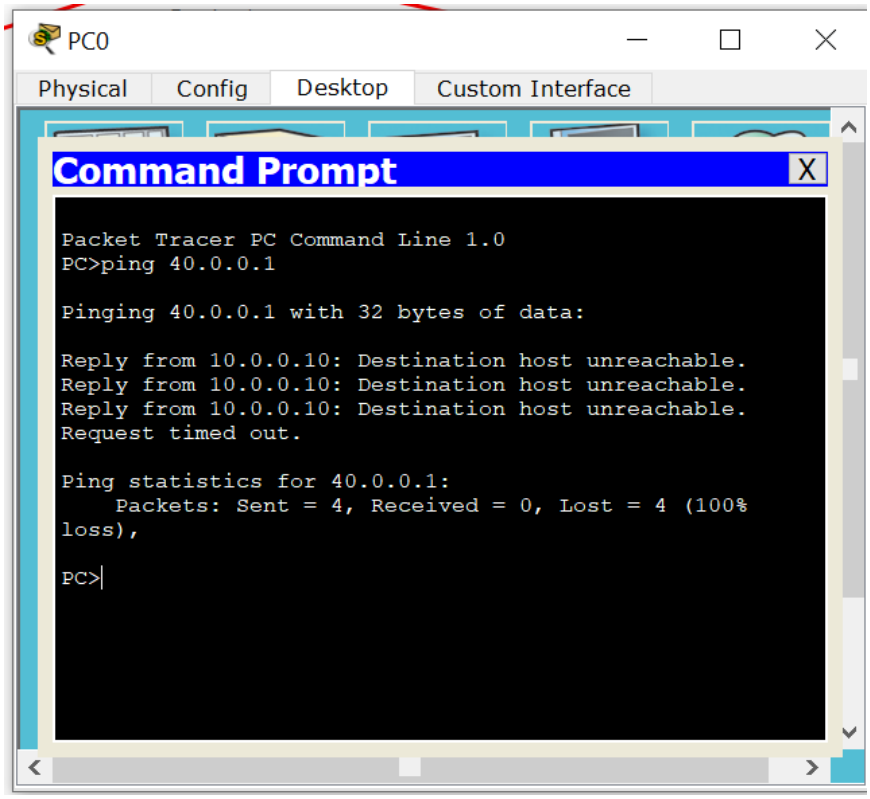
OUTPUT:

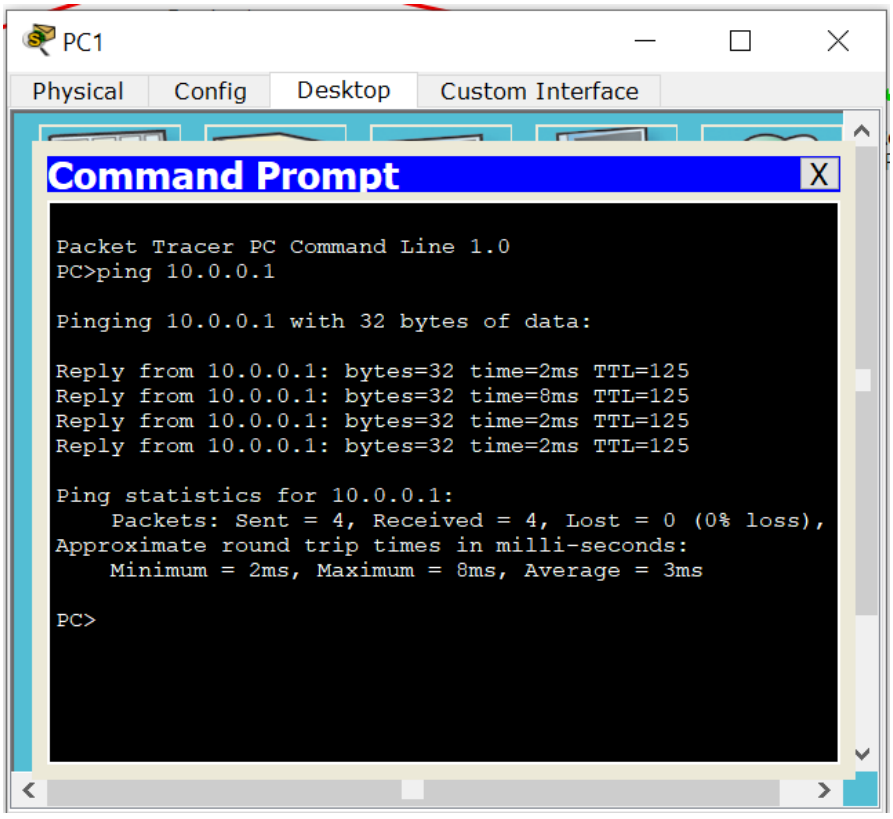
PROGRAM 2.1

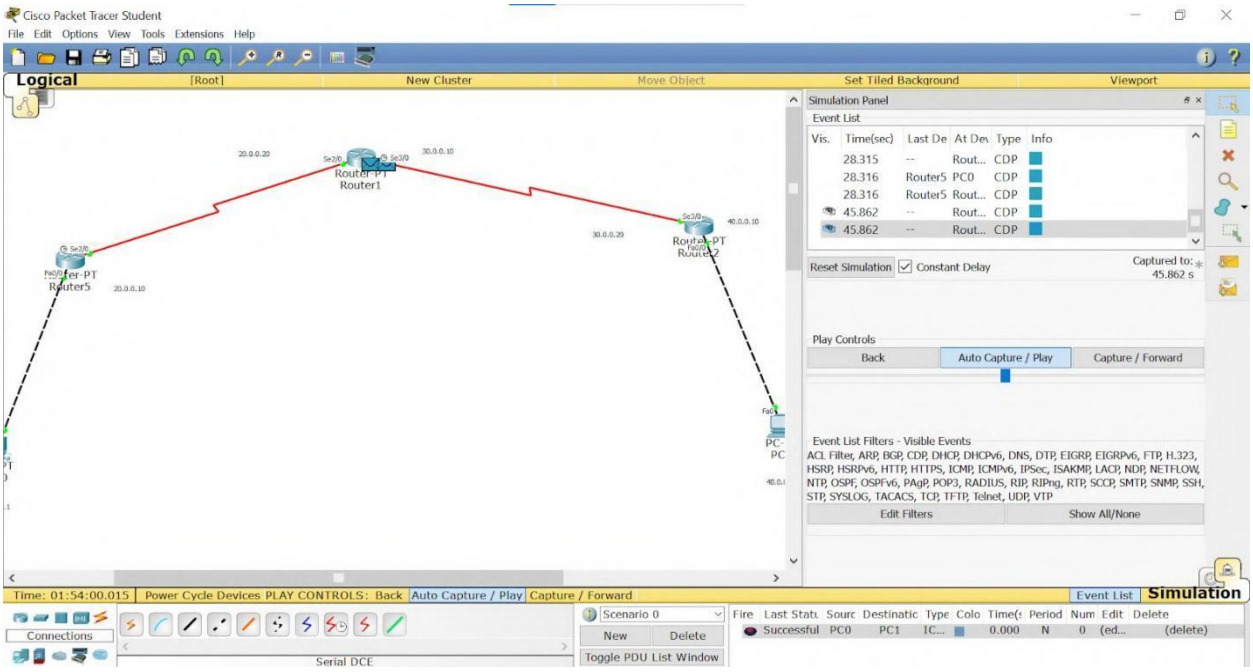




PROGRAM 2.2


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Reply from 10.0.0.10: Destination host unreachable.
Reply from 10.0.0.10: Destination host unreachable.
Reply from 10.0.0.10: Destination host unreachable.
Request timed out.
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 0, Lost = 4 (100%
loss),
PC>

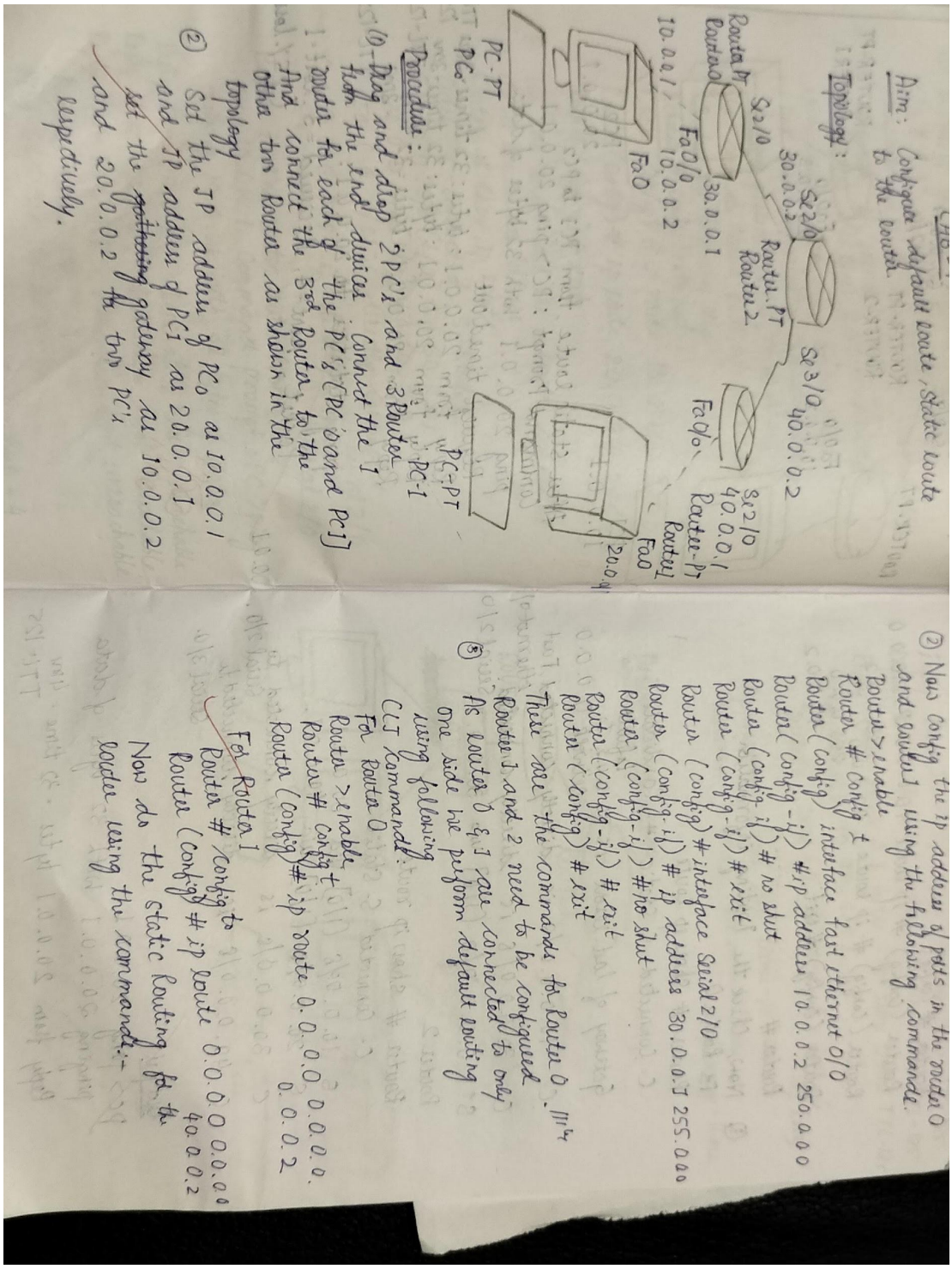

PC1
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=2ms TTL=125
Reply from 10.0.0.1: bytes=32 time=8ms TTL=125
Reply from 10.0.0.1: bytes=32 time=2ms TTL=125
Reply from 10.0.0.1: bytes=32 time=2ms TTL=125
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 8ms, Average = 3ms
PC>


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

WEEK 3

Configure default route, static route to the Router.

OBSERVATION:



Router # config t
Router (config) # ip route 10.0.0.0 255.0.0.0 30.0.0.2
Router (config) # ip route 20.0.0.0 255.0.0.0 40.0.0.2
Router (config) #exit
Router #

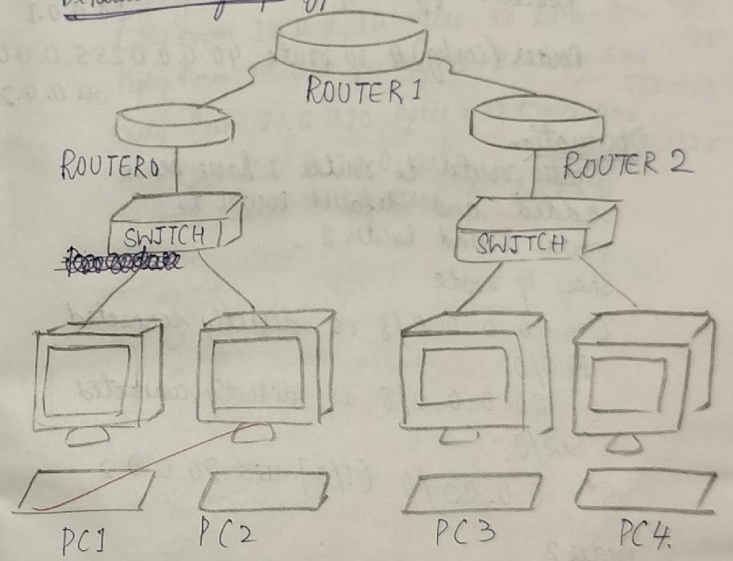
③ Now, Check the Routing information
For Router 0
Router # show ip route
C - Connected S - Static * - Candidate Default
Gateway of last resort is 30.0.0.2 to network 0.0.0.0
C 10.0.0.0/8 is directly connected, Fast ethernet 0/.
C. 30.0.0.0/8 is directly connected. Serial 2/0
S* 0.0.0.0/0 [1/0] via 30.0.0.1

Router 2
Router # show ip route
C - Connected S - Static
S 10.0.0.0/8 [1/0] via 30.0.0.1
S 20.0.0.0/8 [1/0] via 40.0.0.1.
C 30.0.0.0/8 is directly connected to Serial 2/0.
C 40.0.0.0/8 is directly connected to Serial 3/0.

Output

PC> ping 20.0.0.1
pinging 20.0.0.1 with 32 bytes of data
Reply from 20.0.0.1 bytes = 32 time = 4ms TTL = 125
Reply from 20.0.0.1 bytes = 32 time = 18ms TTL = 125
Reply from 20.0.0.1 bytes = 32 time = 17ms TTL = 125
Reply from 20.0.0.1 bytes = 32 time = 25ms TTL = 125
ping statistics for 20.0.0.1:
packets: Sent = 4. Received = 4, Lost = 0 (0% loss)
Approximate round trip times in ms.
Minimum = 4ms, Maximum = 25ms
Average = 16ms

Default Routing Topology



Procedure

Configuring default route to router 0 & 2

Router 0

Router (config)# ip Route 0.0.0.0 0.0.0.0 20.0.0.2

Router 2.

Router (config)# ip Route 0.0.0.0 0.0.0.0 30.0.0.1

Configure 2. static router of router 1

Router 1:

Router (config)# ip route 10.0.0.0 255.0.0.0 20.0.0.1

Router (config)# ip route 40.0.0.0 255.0.0.0 30.0.0.2

Observation

Static route to router 1 have been added and default router to router 0 and router 2.

Show ip route

C 10.0.0.0/8 is directly connected Fe0/0

C 20.0.0.0/8 is directly connected Se2/0.

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

Router 2

Show ip route

C 30.0.0.0/8 is directly connected Se2/0

C 40.0.0.0/8 is directly connected Fe0/0

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

Router 1

Show ip route

S 10.0.0.0/8 [1/0] via 20.0.0.1

C 20.0.0.0/8 is directly connected Se2/0

C 30.0.0.0/8 is directly connected Se3/0

S 40.0.0.0/8 [1/0] via 30.0.0.2.

Output

Ping request from PC4

→ Ping 10.0.0.10

Pinging 10.0.0.10 with 32 bytes of data.

Reply from 10.0.0.10 bytes=32 time=9ms TTL=125

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

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

Ping statistics for 10.0.0.10

Packets: Sent=4, received=4 lost=0.

Sept 21/7/23

Procedure

Configuring default route to router 0 & 2

Router 0

Router (config) # ip Route 0.0.0.0 0.0.0.0 20.0.0.2

Router 2.

Router (config) # ip Route 0.0.0.0 0.0.0.0 30.0.0.1

Configure 2. static routes of router 1

Router 1:

Router (config)# ip route 10.0.0.0 255.0.0.0 20.0.0.1

Router (config)# ip route 40.0.0.0 255.0.0.0 30.0.0.2

Observation

Static route to router 1 have been added and default route to router 0 and router 2.

Show ip route

C 10.0.0.0/8 is directly connected Fe 0/0

C 20.0.0.0/8 is directly connected Se2/0.

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

Router 2

Show ip route

C 30.0.0.0/8 is directly connected Se2/0

C 40.0.0.0/8 is directly connected Fe0/0

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

Router 1

Show ip route

S 10.0.0.0/8 [1/0] via 20.0.0.1

C 20.0.0.0/8 is directly connected Se2/0

C 30.0.0.0/8 is directly connected Se3/0

S 40.0.0.0/8 [1/0] via 30.0.0.2.

Output

Ping request from PC4

→ Ping 10.0.0.10

Pinging 10.0.0.10 with 32 bytes of data

Reply from 10.0.0.10 bytes=32 time=9ms TTL=125

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

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

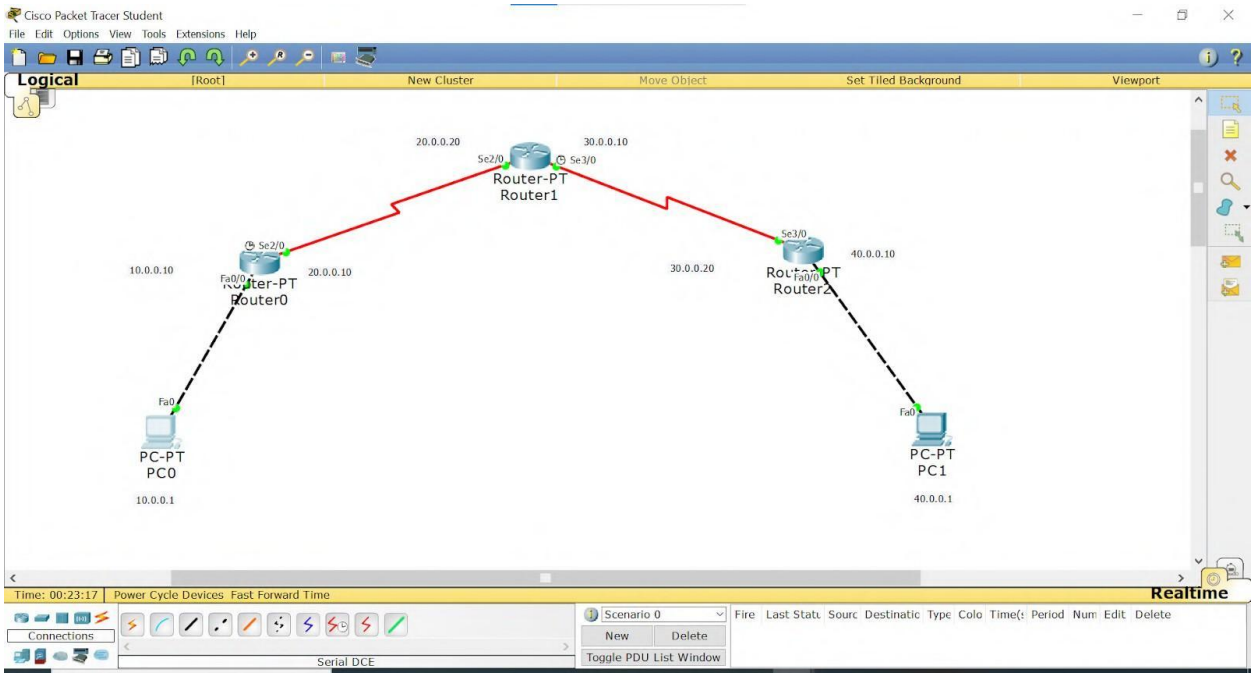
Ping statistics for 10.0.0.10

Packets: Sent=4, received=4 lost=0.

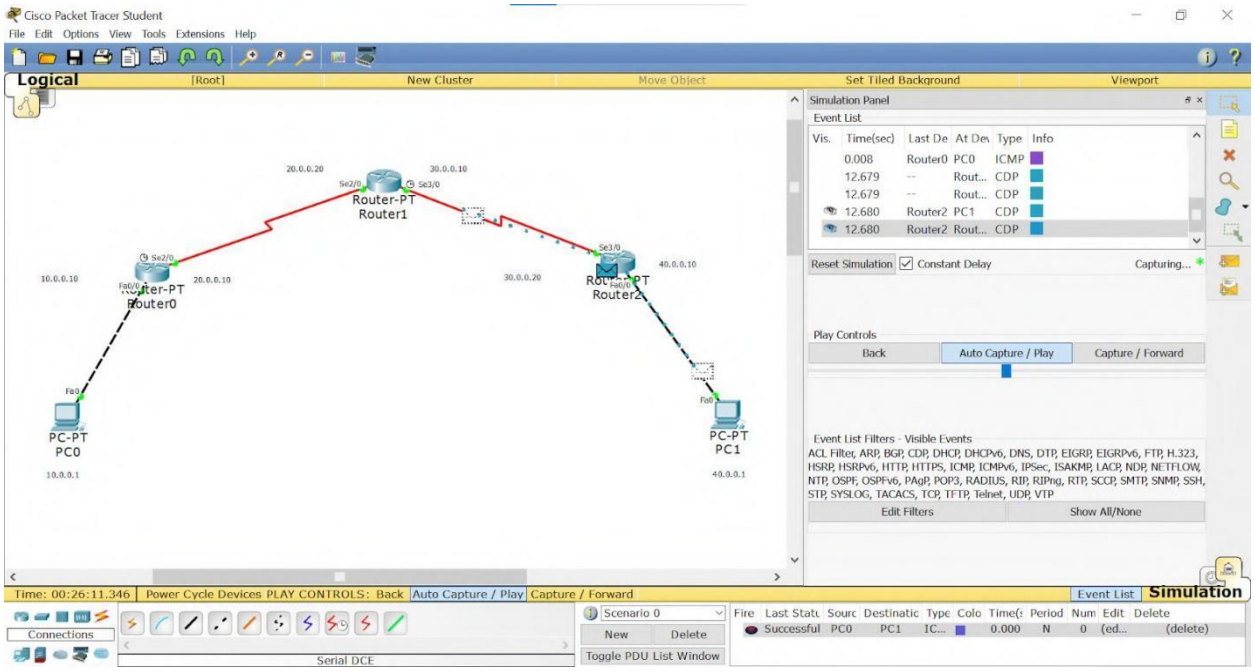
Sd. 1

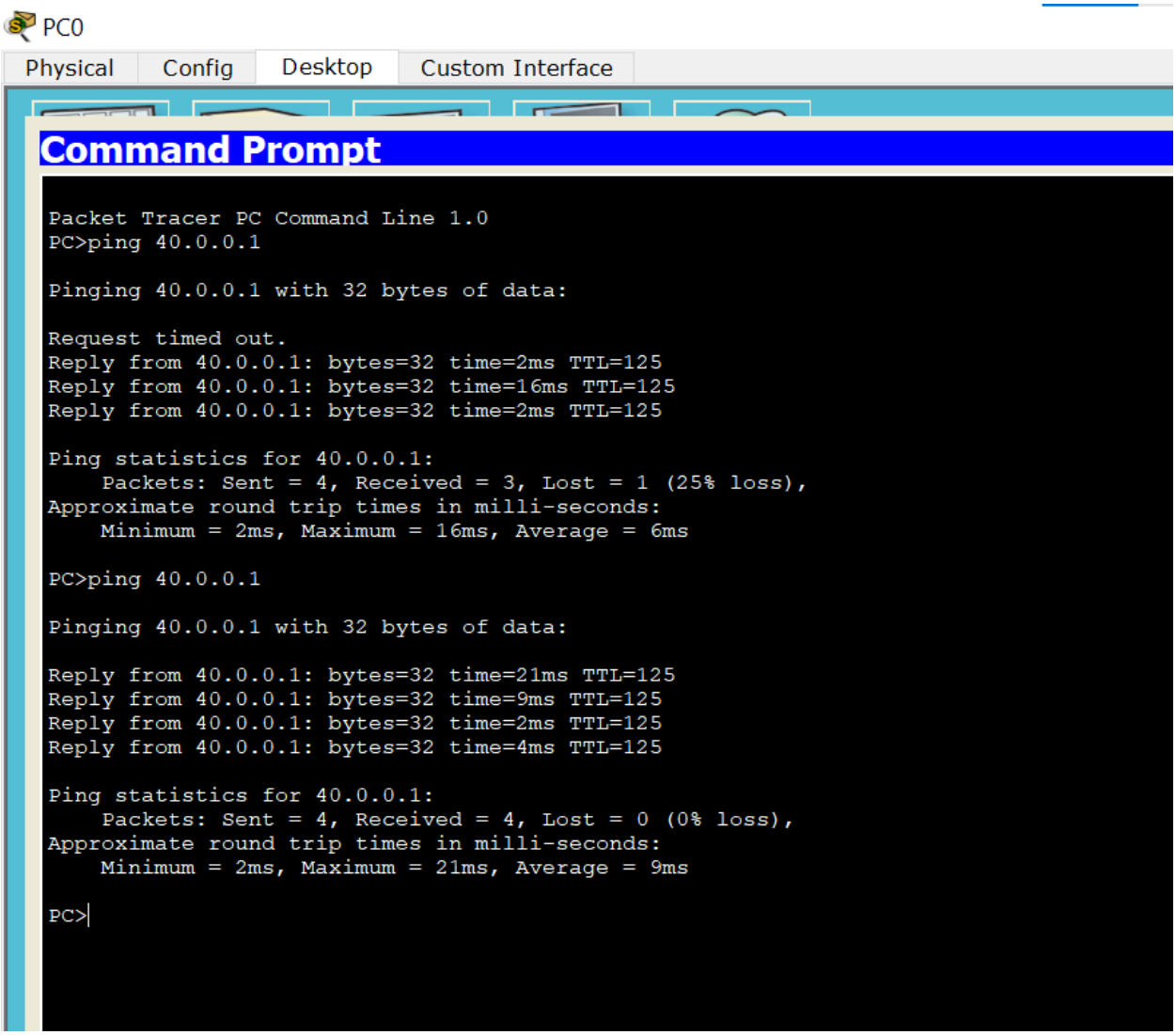
21/7/23

TOPOLOGY:



OUTPUT:

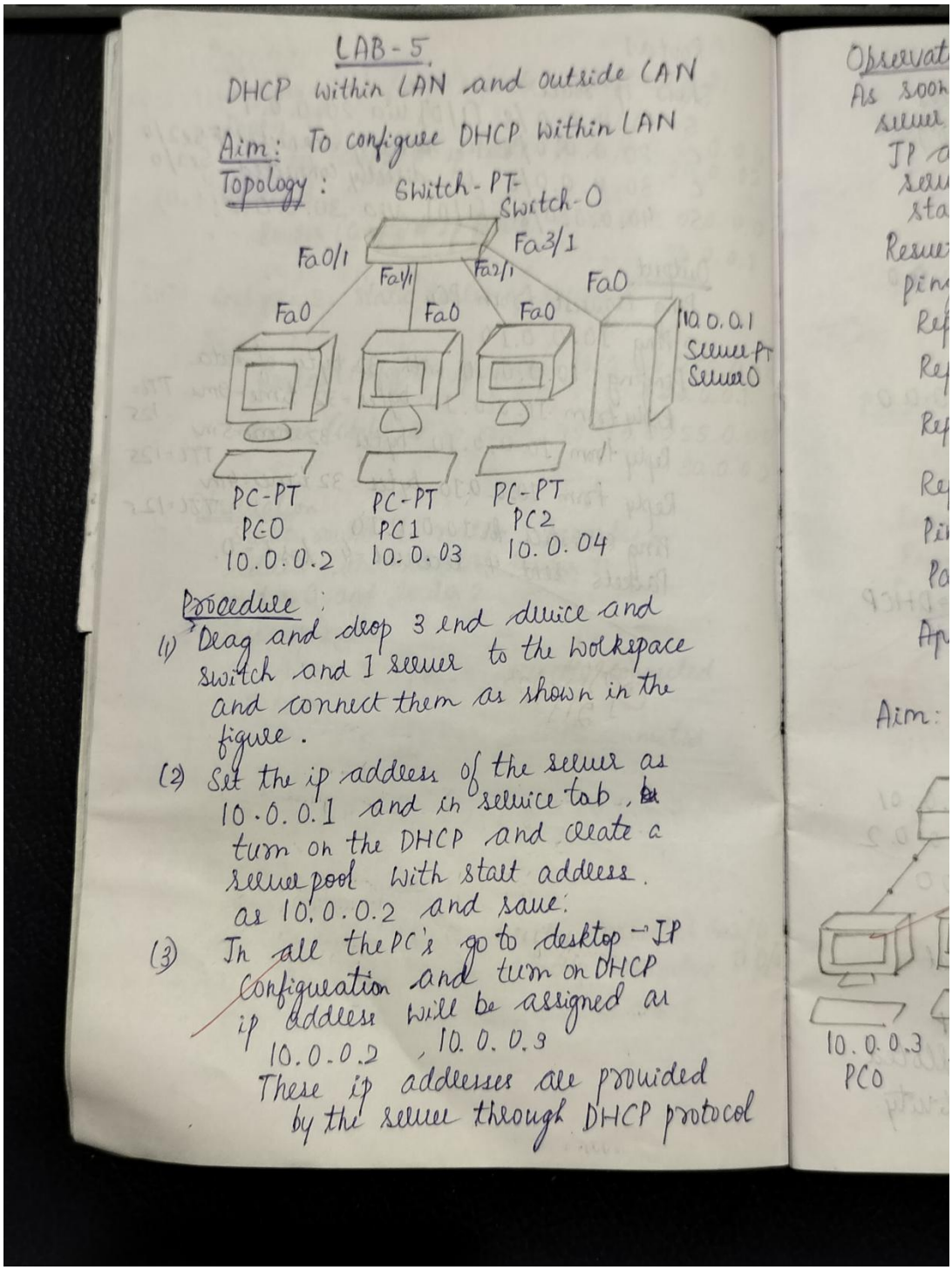



PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Request timed out.
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125
Reply from 40.0.0.1: bytes=32 time=16ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 16ms, Average = 6ms
PC>ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Reply from 40.0.0.1: bytes=32 time=21ms TTL=125
Reply from 40.0.0.1: bytes=32 time=9ms TTL=125
Reply from 40.0.0.1: bytes=32 time=2ms TTL=125
Reply from 40.0.0.1: bytes=32 time=4ms TTL=125
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 21ms, Average = 9ms
PC>

WEEK 4

Configure DHCP within a LAN and outside LAN.

OBSERVATION:



LAB-5

DHCP within LAN and outside LAN

Aim: To configure DHCP within LAN

Topology:

Procedure:

(1) Drag and drop 3 end device and switch and 1 server to the workspace and connect them as shown in the figure.

(2) Set the ip address of the server as 10.0.0.1 and in service tab, turn on the DHCP and create a server pool with start address as 10.0.0.2 and save.

(3) In all the PC's go to desktop → IP configuration and turn on DHCP ip address will be assigned as 10.0.0.2, 10.0.0.3

These ip addresses are provided by the server through DHCP protocol

Observat

As soon

server

IP

ser

sta

Reser

pin

Re

Re

Re

Re

Pi

Po

Ap

Aim:

10.0.0.3

PC0

Procedure:

(1) Drag and drop the end devices connect them as shown in the figure
Click on the router - CLI
→ enable
→ Configure terminal
→ interface Fa0/0
→ ip address 10.0.0.4 255.0.0.0
→ no shutdown
→ exit
→ interface Fa1/0
→ ip address 20.0.0.1 255.0.0.0
→ no shutdown
→ exit
→ interface Fa1/0
→ ip helper address 10.0.0.1
→ exit

(2) Click on the server → services → DHCP
server pool → Gateway 10.0.0.4
Starting address 10.0.0.2
Maximum no of device = 10
Click on save
Server pool new → gateway 20.0.0.1
Starting address 20.0.0.2
Maximum no of devices = 100
Click on add.

(3) For each PC set fastethernet 0/0 to DHCP

(4) All PC's will have their IP address automatically alloted

(5) Ping PC's to test connectivity

Observa

For a s
allott
A gate
for s
the se
and

Once t
serve
be

Result:

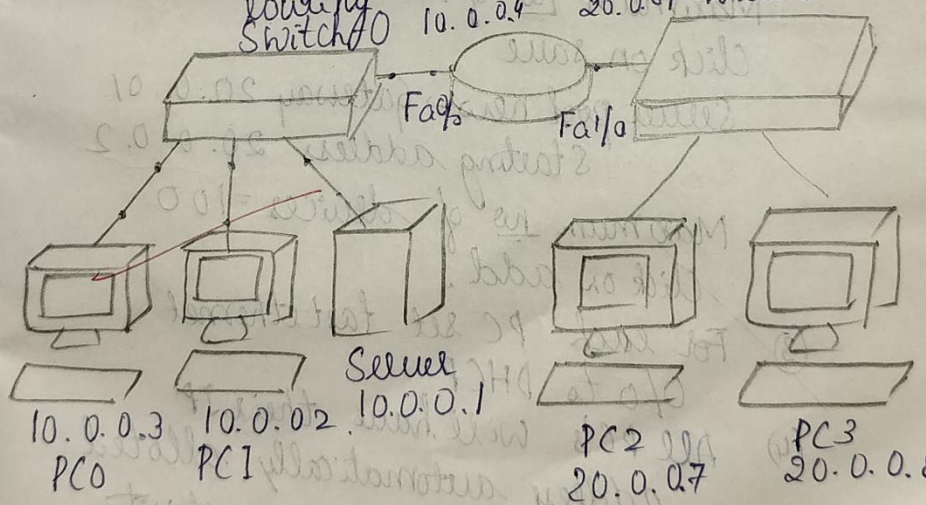
Ping
Reply
Reply
Reply
Reply

SPT
21/7/22

Observation:
As soon as the server is assigned a server pool with the appropriate starting IP address all devices within that server pool will have an IP address starting from that starting address.

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=125
Reply from 10.0.0.3 bytes=32 time=0ms TTL=125
Reply from 10.0.0.3 bytes=32 time=0ms TTL=125
Reply from 10.0.0.3 bytes=32 time=0ms TTL=125
Ping Statistics for 10.0.0.3
Packets sent=4 Received=4 lost=0 (0% loss)
Approximate=0ms, Maximum=0ms, Average=0ms

Aim: Diagram for internal LAN DHCP



Observation

For a server, IP address must be allotted statically
A gateway is assigned to the server for server pool now, the PC's outside the server LAN will have 2 random address 169.254.0.1
Once the gateway is assigned to the server the PCs will automatically be assigned the gateway.

Result:

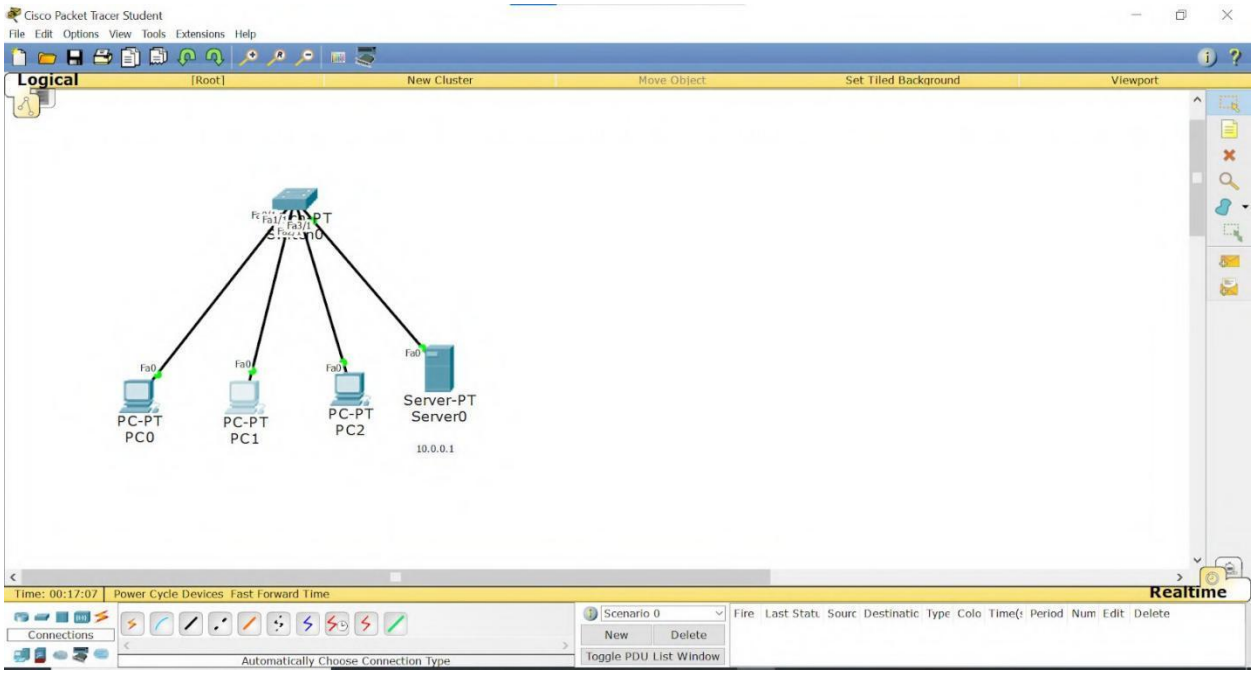
Ping 20.0.0.7 (from 10.0.0.2)
Reply from 20.0.0.7 byte=32 time=1ms TTL=127
Reply from 20.0.0.7 byte=32 time=0ms TTL=127
Reply from 20.0.0.7 byte=32 time=1ms TTL=127
Reply from 20.0.0.7 byte=32 time=0ms TTL=127

SJT
21/7/23

Output

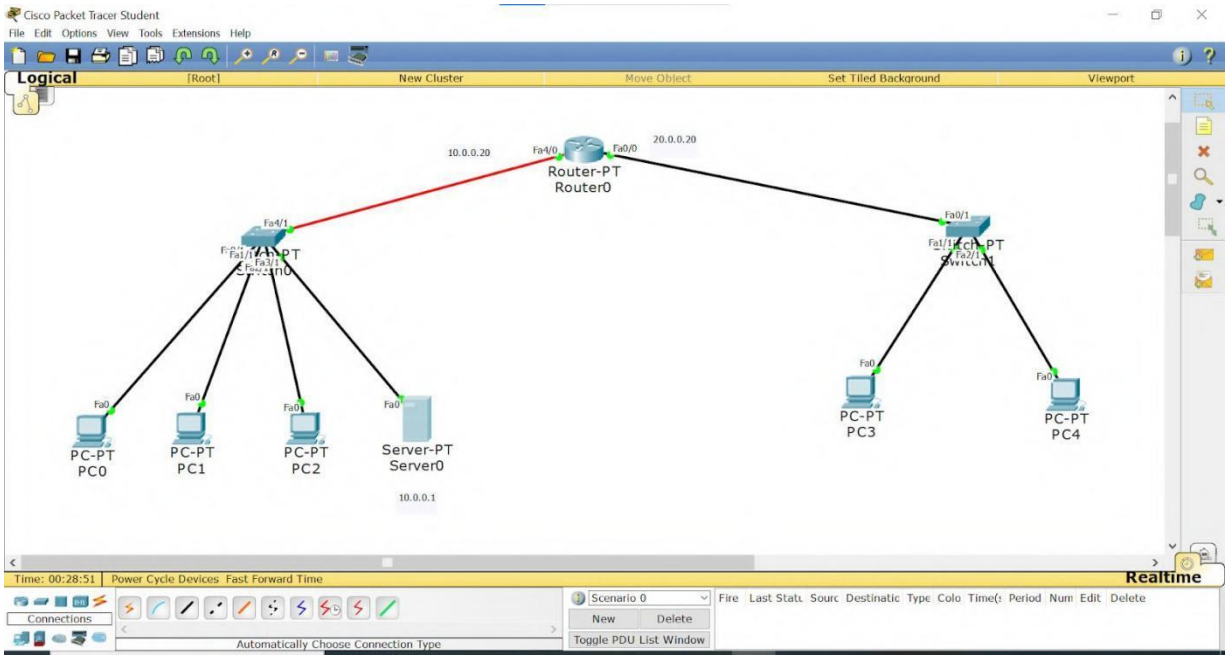
TOPOLOGY:

PROGRAM



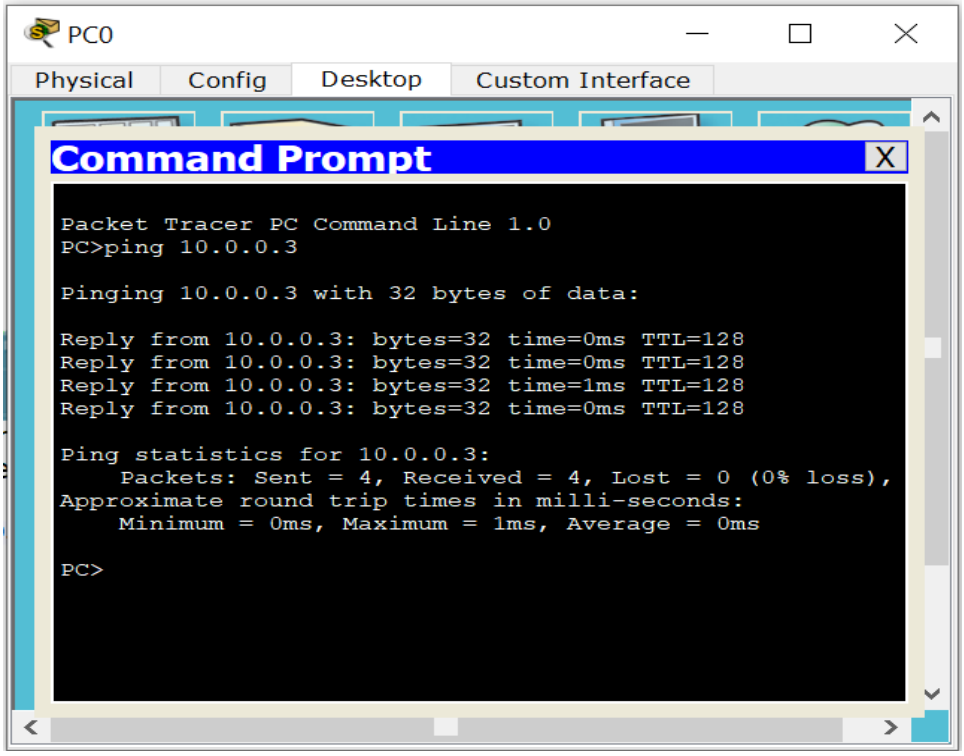
4.1:

PROGRAM 4.2:

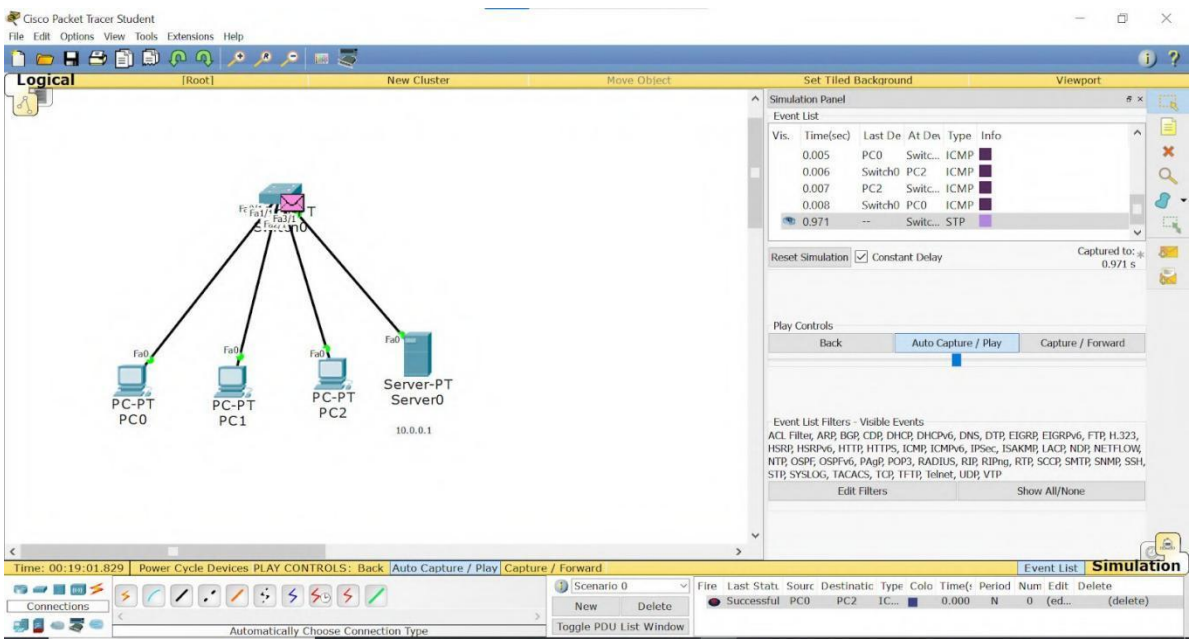


OUTPUT:

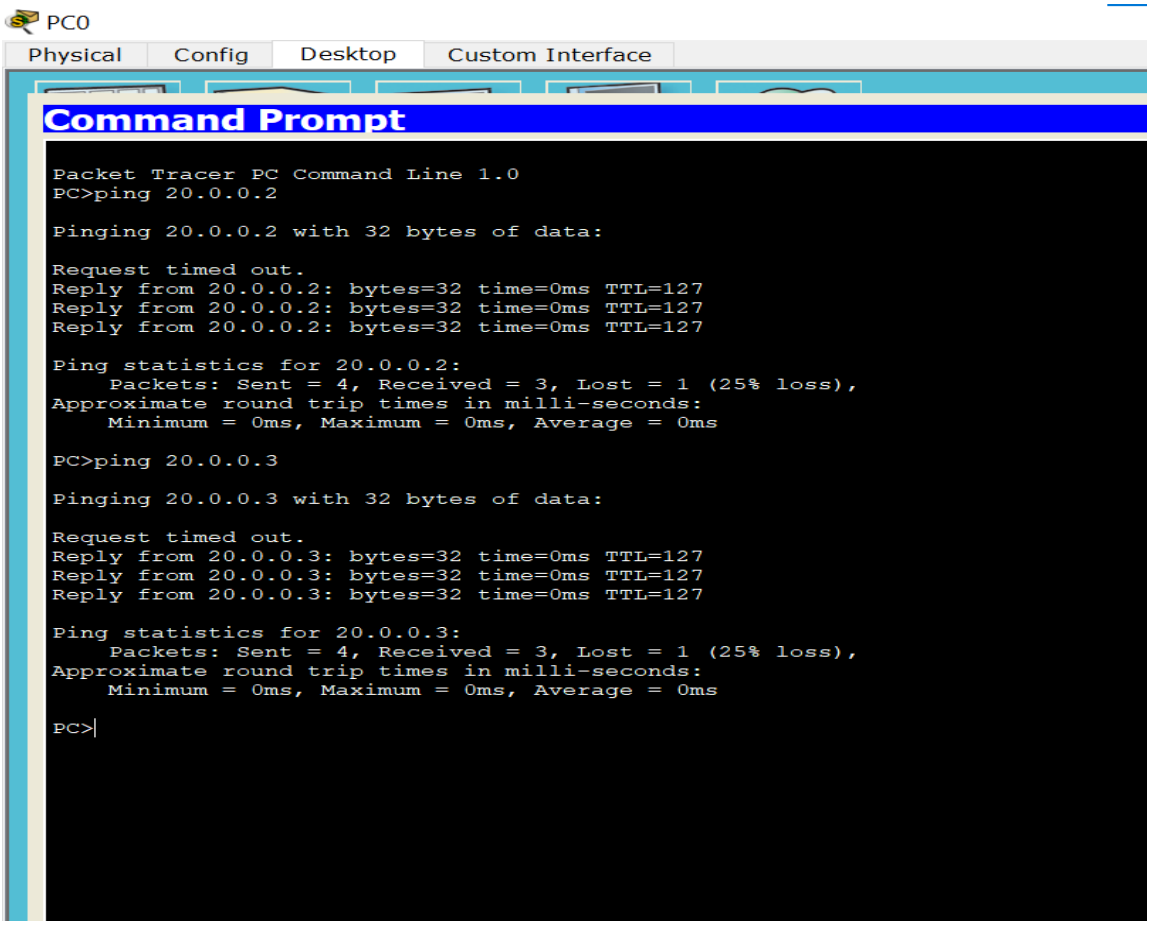
PROGRAM

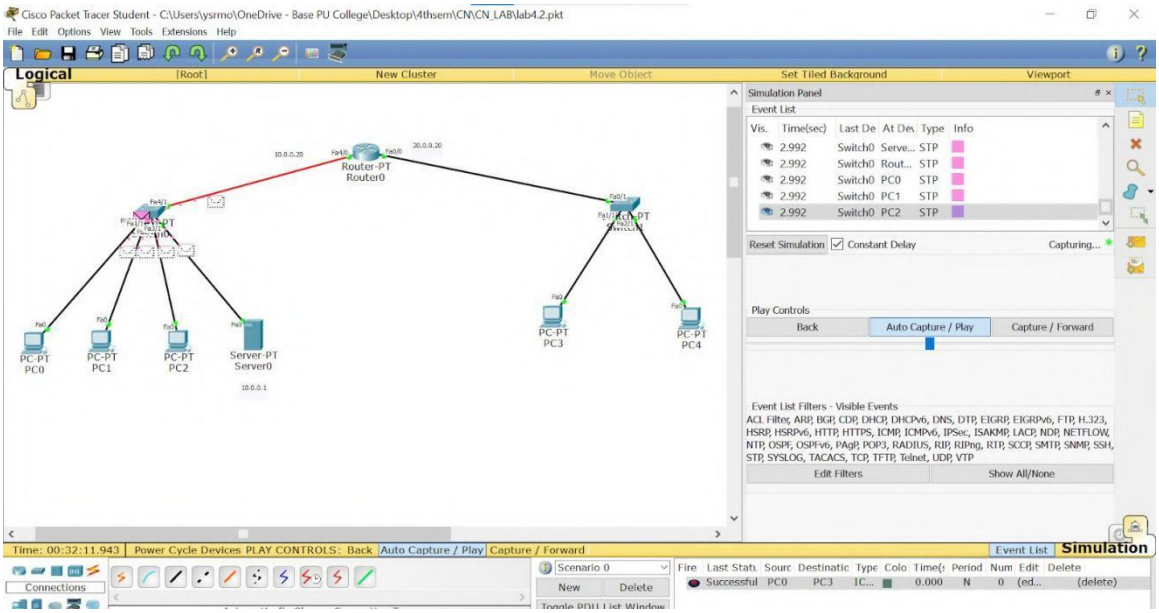


4.1:



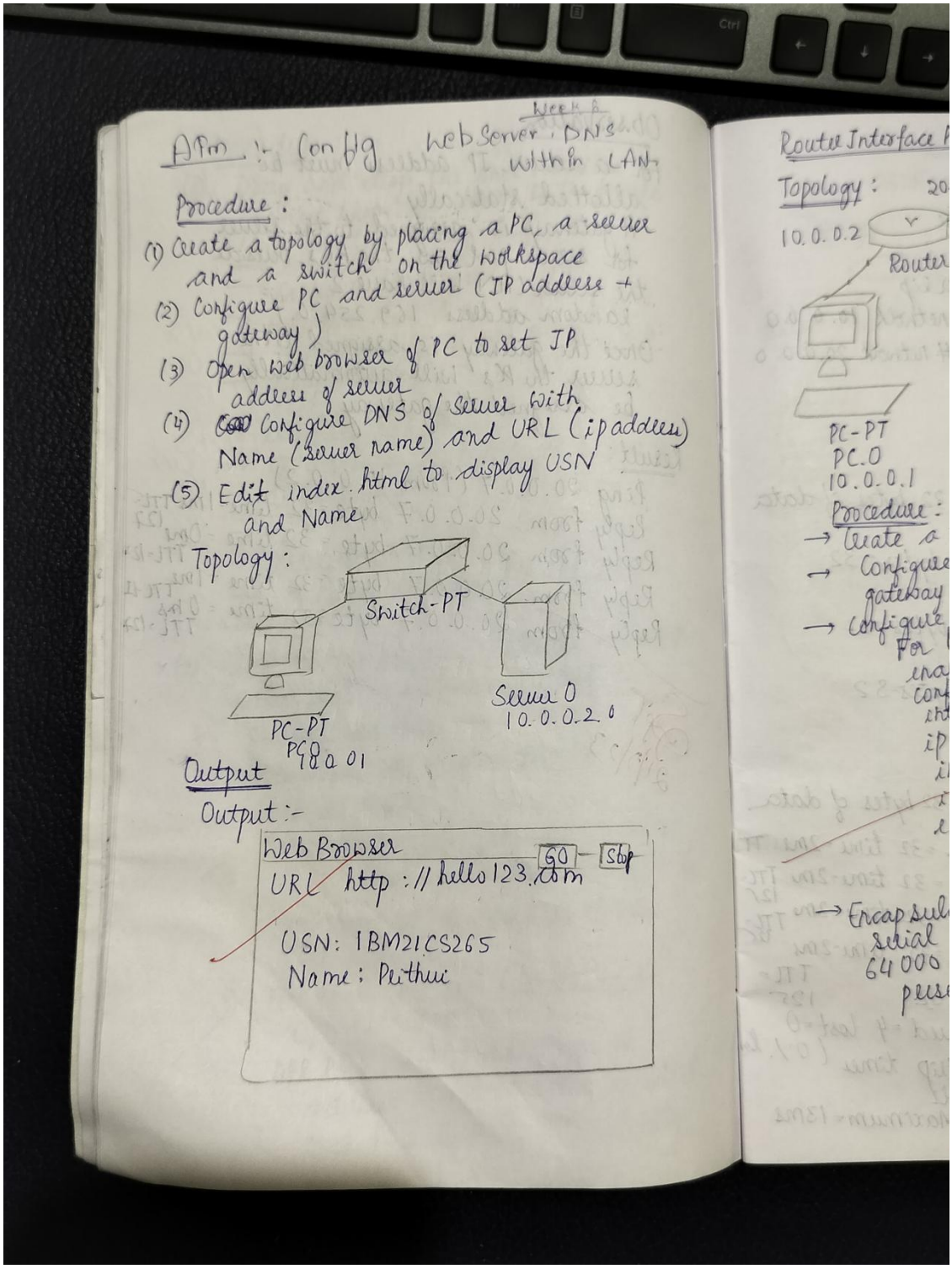
PROGRAM 4.2:


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>

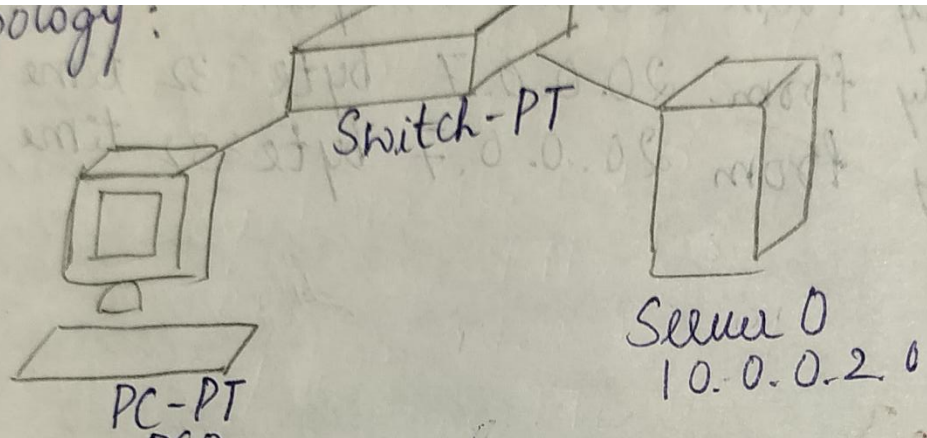

Cisco Packet Tracer Student - C:\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\CN\CN_LAB\lab4.2.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT
Router0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
Server-PT
Server0
PC-PT
PC3
PC-PT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

WEEK 5

Configure Web Server, DNS within a LAN.
OBSERVATION:



Topology :



Output

Output :-

Web Browser

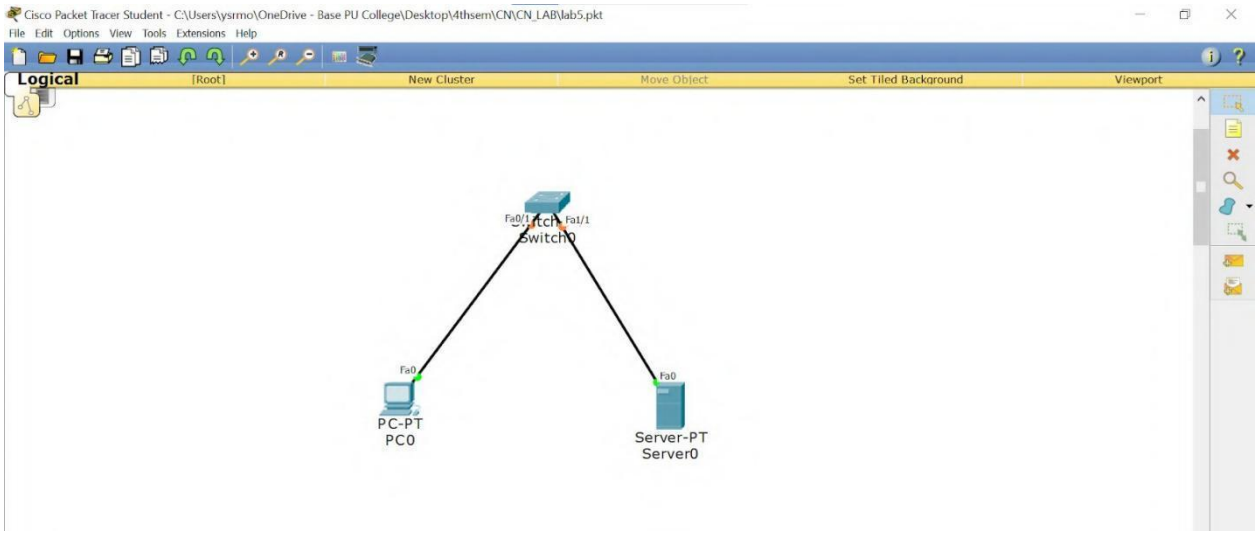
GO | Stop

URL http://hello123.com

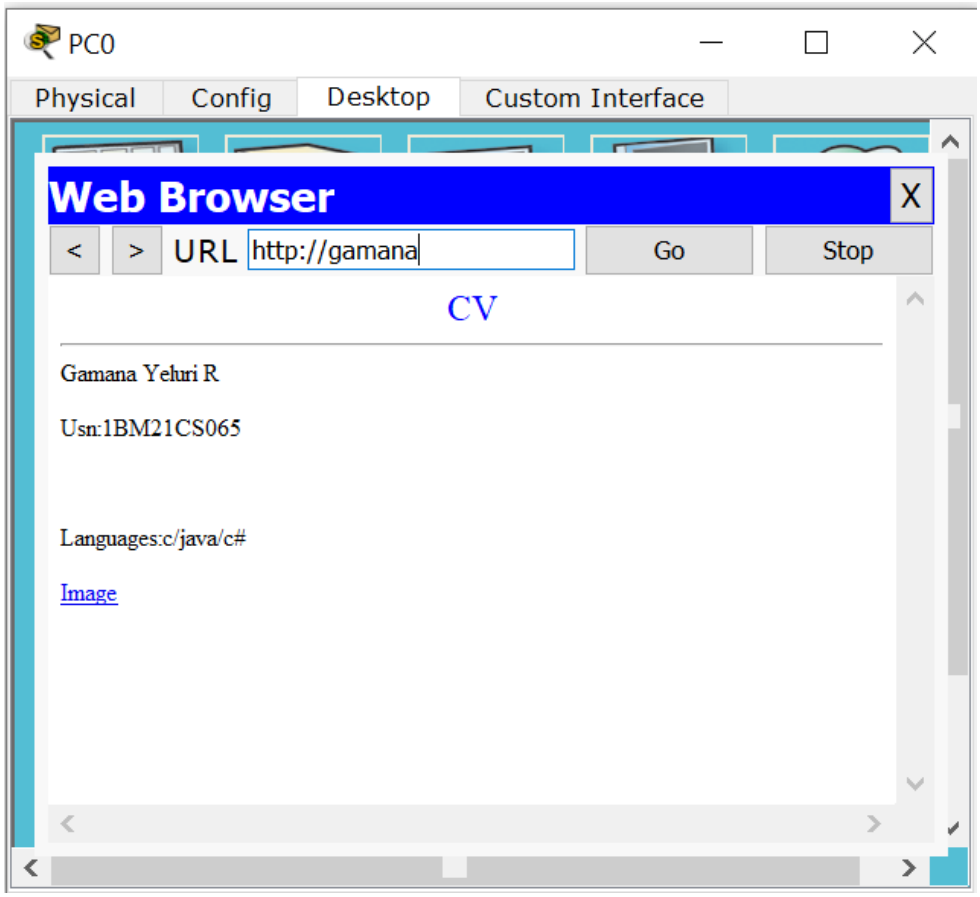
USN: 1BM21CS265

Name: Pathui

TOPOLOGY:



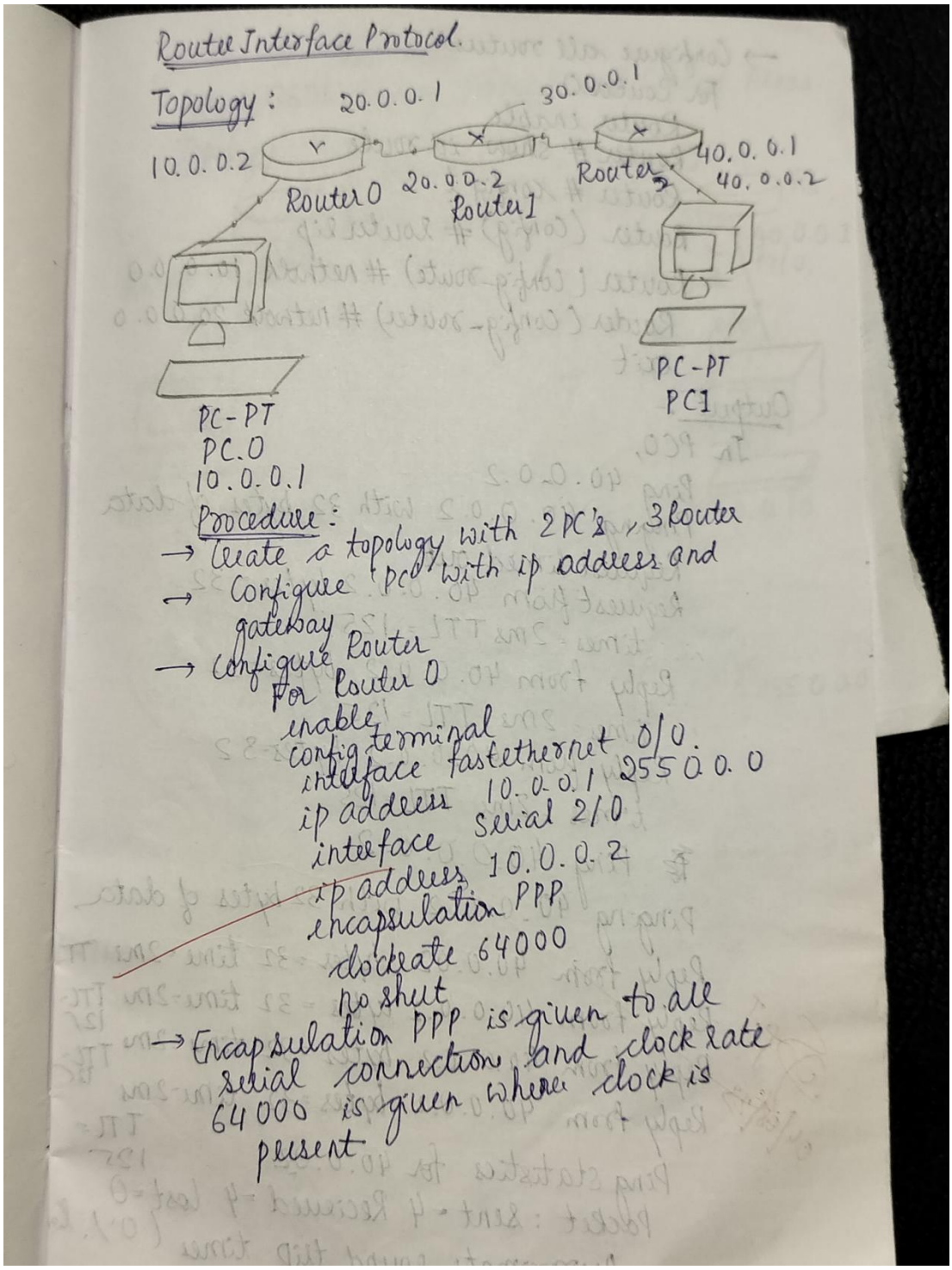
OUTPUT:



WEEK 6

Configure RIP routing Protocol in Routers.

OBSERVATION:



→ Configure all routers

For Router 0

Router enable
Router # show id route
Router # config t
Router (Config) # router rip
Router (Config-route) # network 10.0.0.0
Router (Config-router) # network 20.0.0.0
exit

Output :-

In PC0,

Ping 40.0.0.2

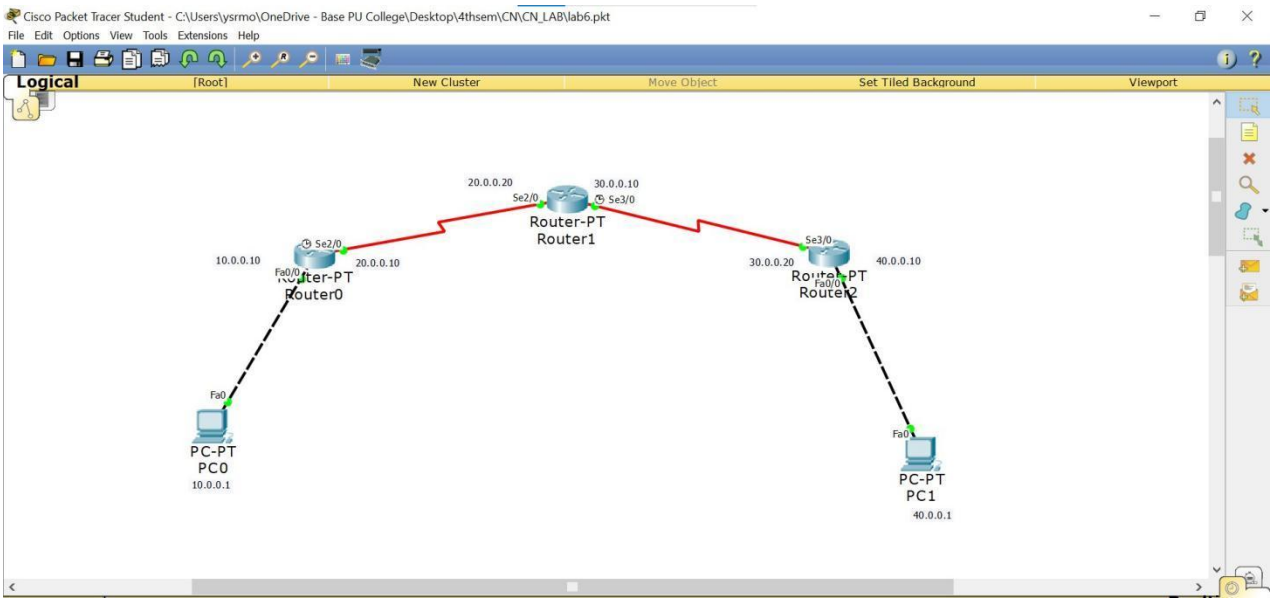
Pinging 40.0.0.2 with 32 bytes of data
Request timed out
Request from 40.0.0.2 bytes = 32
times = 2ms TTL = 125
Reply from 40.0.0.2 bytes = 32
times = 2ms TTL = 125
Reply from 40.0.0.2 bytes = 32
times = 2ms TTL = 125

Ping 40.0.0.0.2

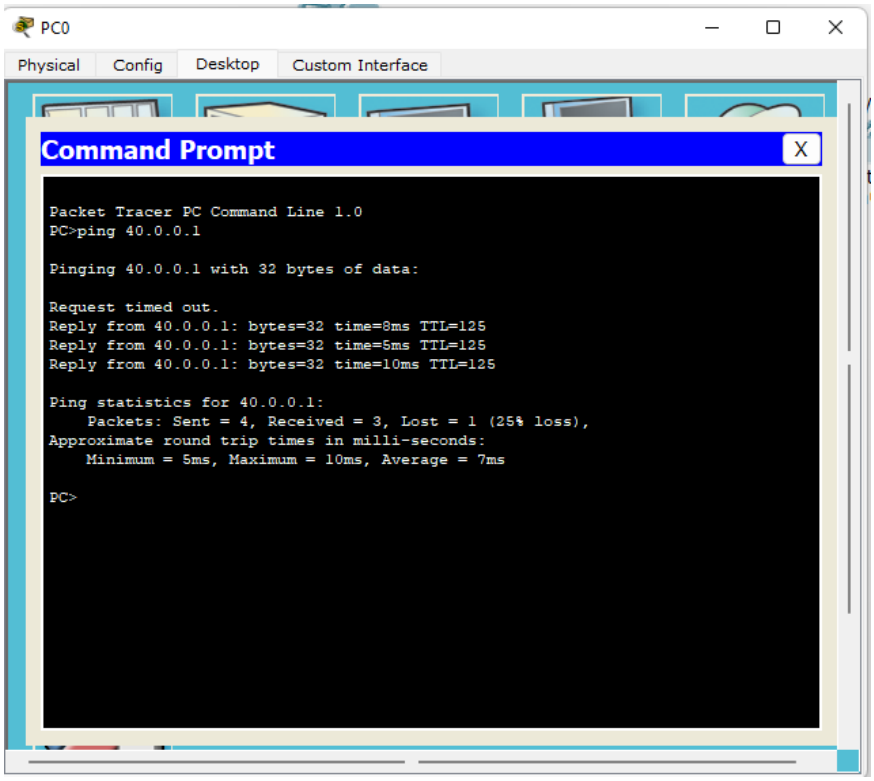
Pinging 40.0.0.2 with 32 bytes of data
Reply from 40.0.02 bytes = 32 time = 2ms TTL = 125
Reply from 40.0.0.2 bytes = 32 time = 2ms TTL = 125
Reply from 40.0.0.2 bytes = 32 time = 2ms TTL = 125
Reply from 40.0.0.2 bytes = 32 time = 2ms TTL = 125

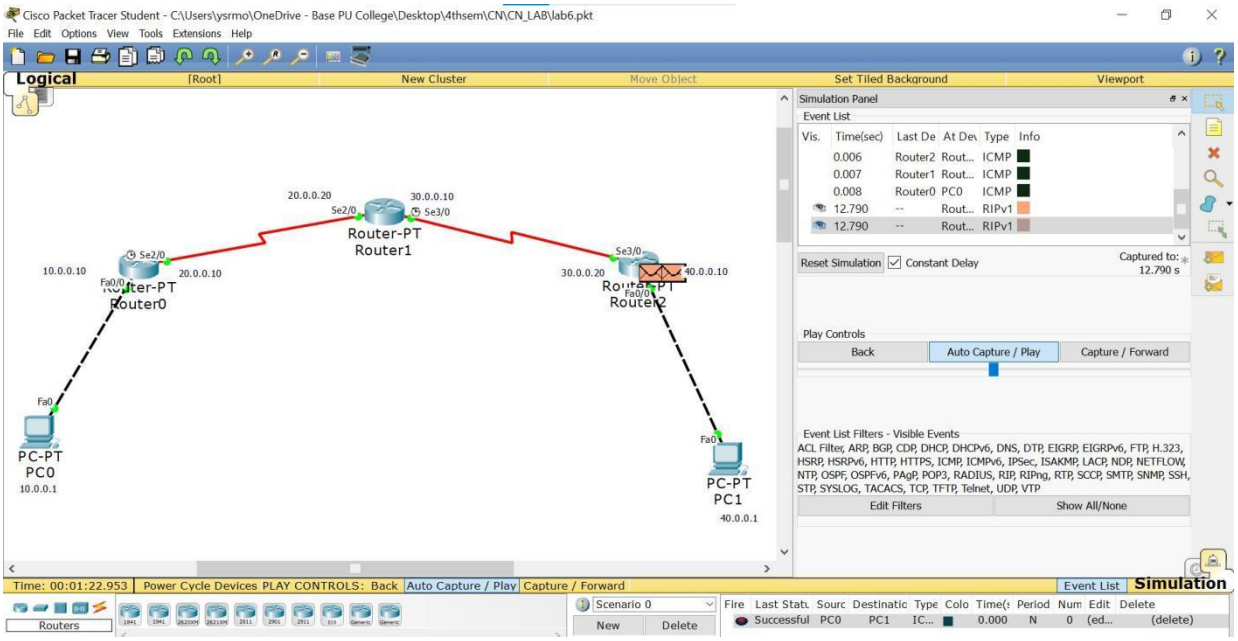
Ping statistics for 40.0.0.2
Packet : Sent = 4 Recieved = 4 lost = 0 (0% loss)
Approximate round trip times
in milli-seconds
Minimum = 2ms, Maximum = 13ms

TOPOLOGY:



OUTPUT:

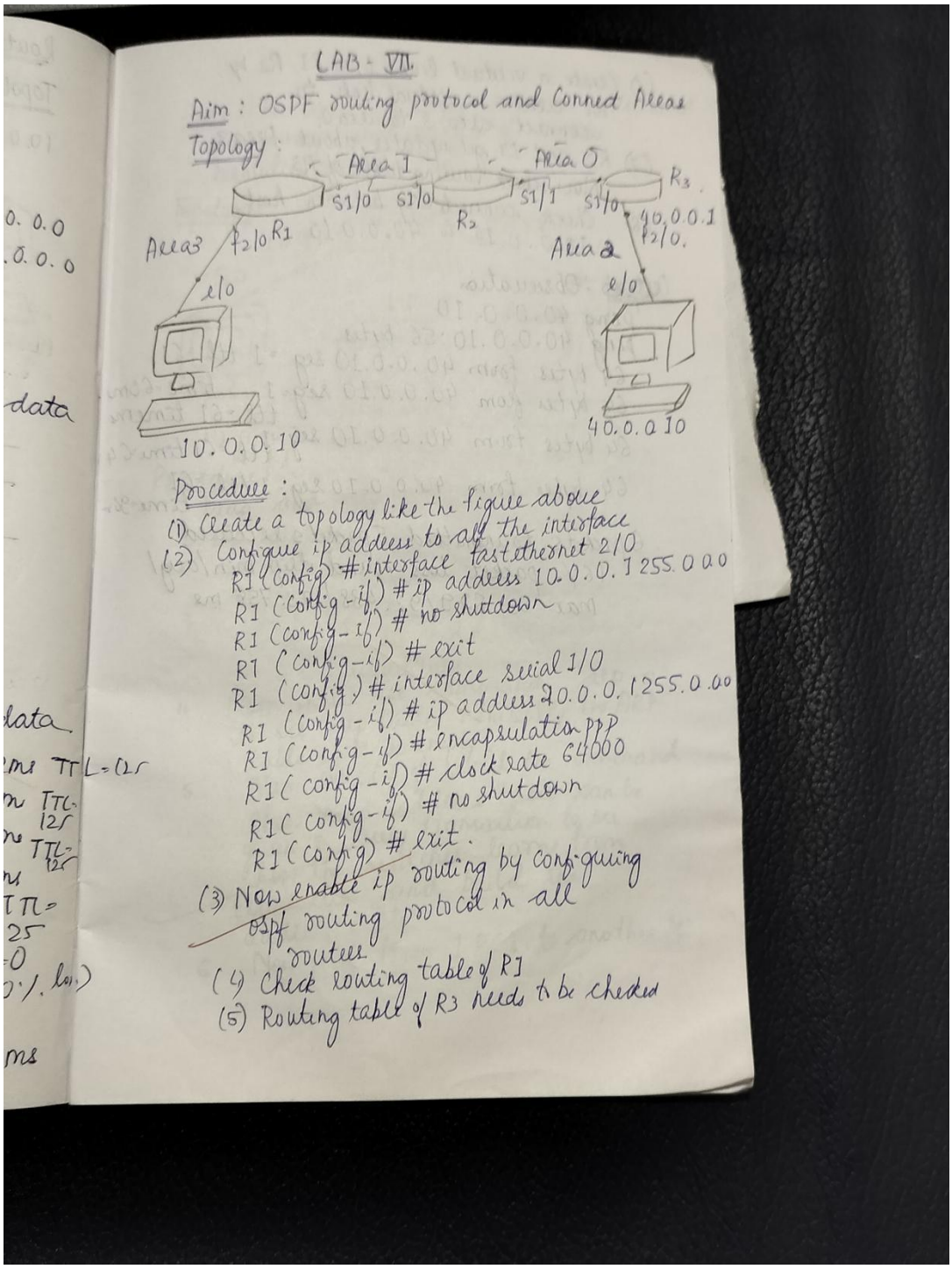



Cisco Packet Tracer Student - C:\Users\ysrmo\OneDrive - Base PU College\Desktop\4thsem\CN\CN_LAB\lab6.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT
Router1
Router-PT
Router0
Router-PT
Router2
PC-PT
PC0
10.0.0.1
PC-PT
PC1
40.0.0.1
Simulation Panel
Event List
Reset Simulation
Constant Delay
Play Controls
Back
Auto Capture / Play
Capture / Forward
Event List Filters - Visible Events
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

WEEK 7

Configure OSPF routing protocol.

OBSERVATION:



(6) Create a virtual link b/w R1, R2 by
We create a virtual link to
connect area 3 to area 0.
(7) R2 and R3 get updates about Area 3,
Now check routing table of R3
(8) Check connectivity between host
10.0.0.10 to 40.0.0.10

Output : Observation

ping 40.0.0.10

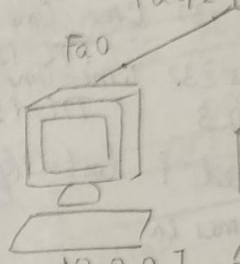
ping 40.0.0.10 :56 bytes
64 bytes from 40.0.0.10 seq =1 ttl =61
64 bytes from 40.0.0.10 seq=1 time =60m
ttl =61 time=
64 bytes from 40.0.0.10 seq=1
ttl =61 time=6
64 bytes from 40.0.0.10 seq=1 ttl =61
time=9

6 packets transmitted, 5 packets received
16% packet loss round trip min/avg/
max 60.829/92.438/173.758 ms

LAB

Aim : To cons
the concept
resolution pro

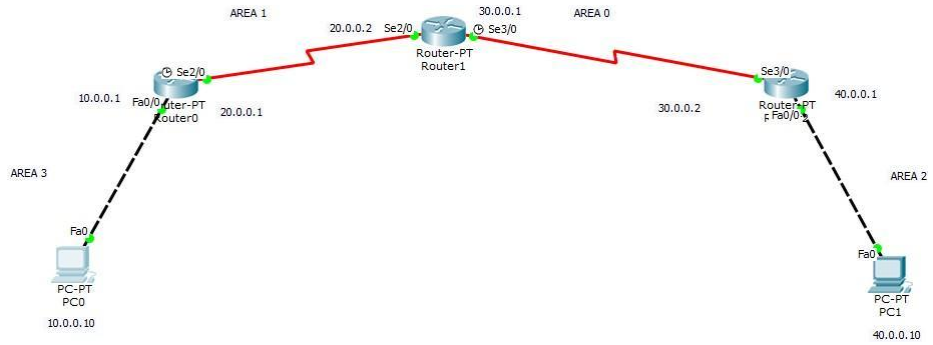
Topology



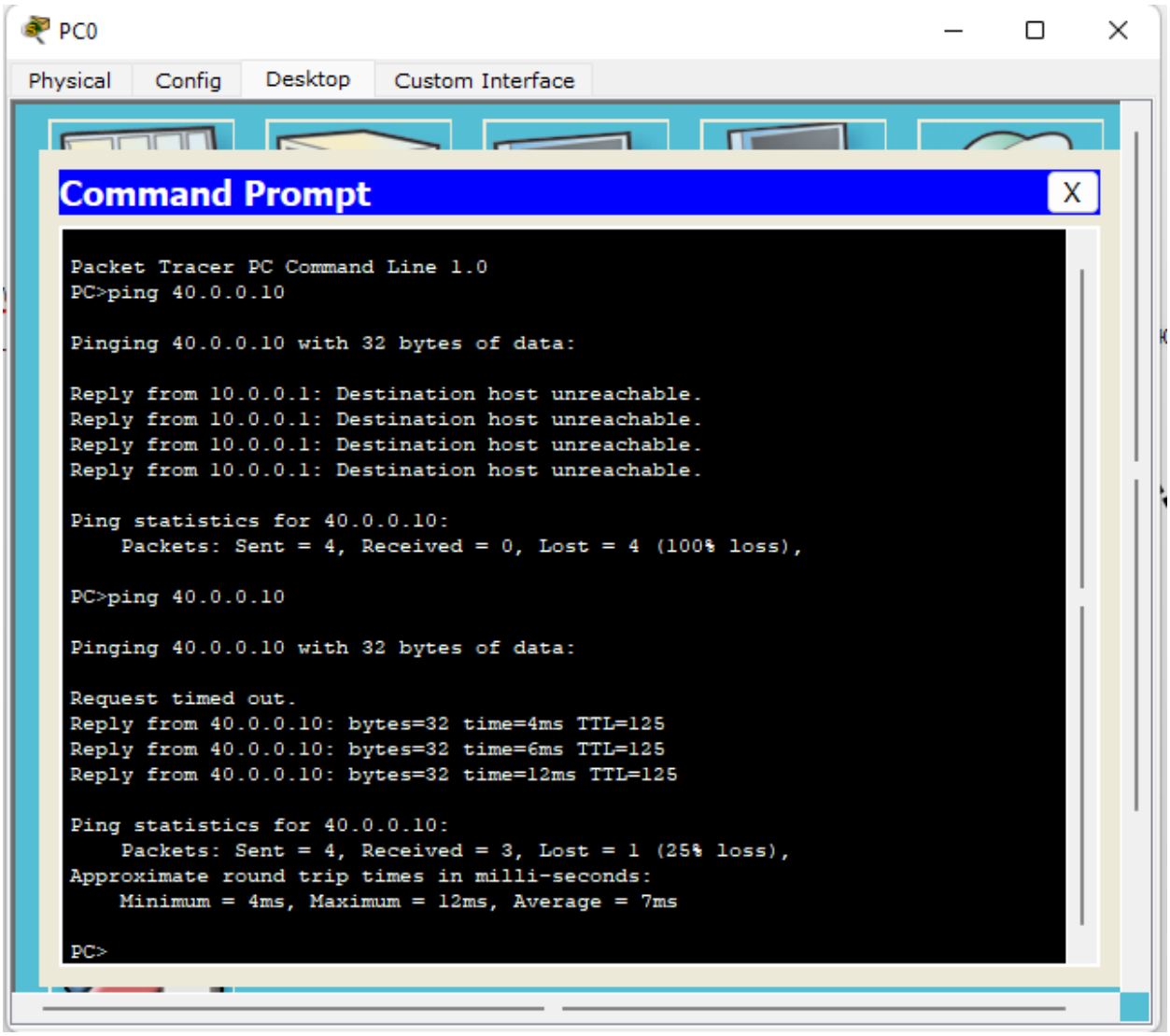
Procedure PC.

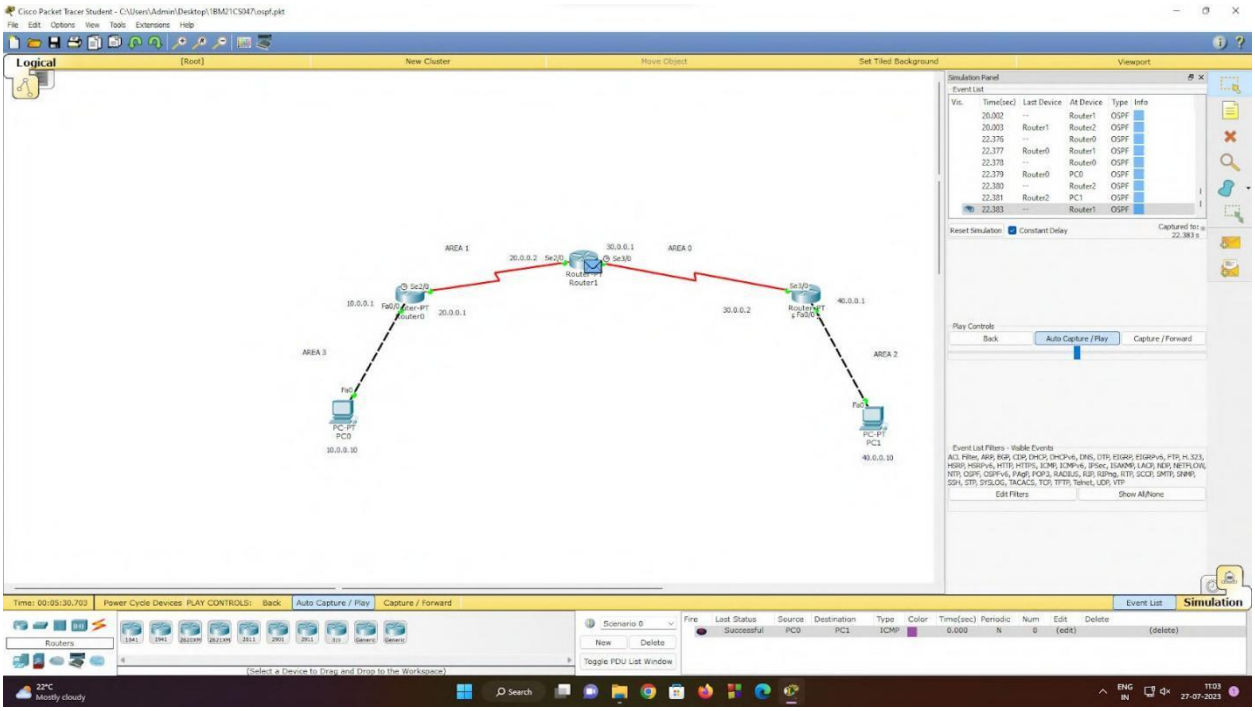
1. Drag and
the devic
2. Connect
as show
3. Config
PC1, R2
10.0.0.
4. Now
to see
table
5. Also
Show
given
host
transa
adde
6 Now

TOPOLOGY:



OUTPUT:




Logical
AREA 1
AREA 0
AREA 3
AREA 2
PC-PT PC0
PC-PT PC1
Simulation

WEEK 8

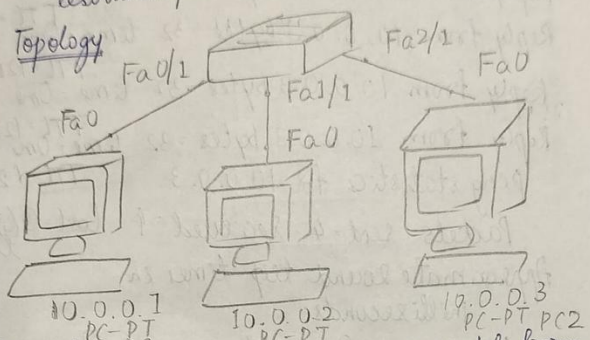
To construct a simple LAN and understand the concept and operation of Address Resolution Protocol (ARP).

OBSERVATION:

LAB - VIII

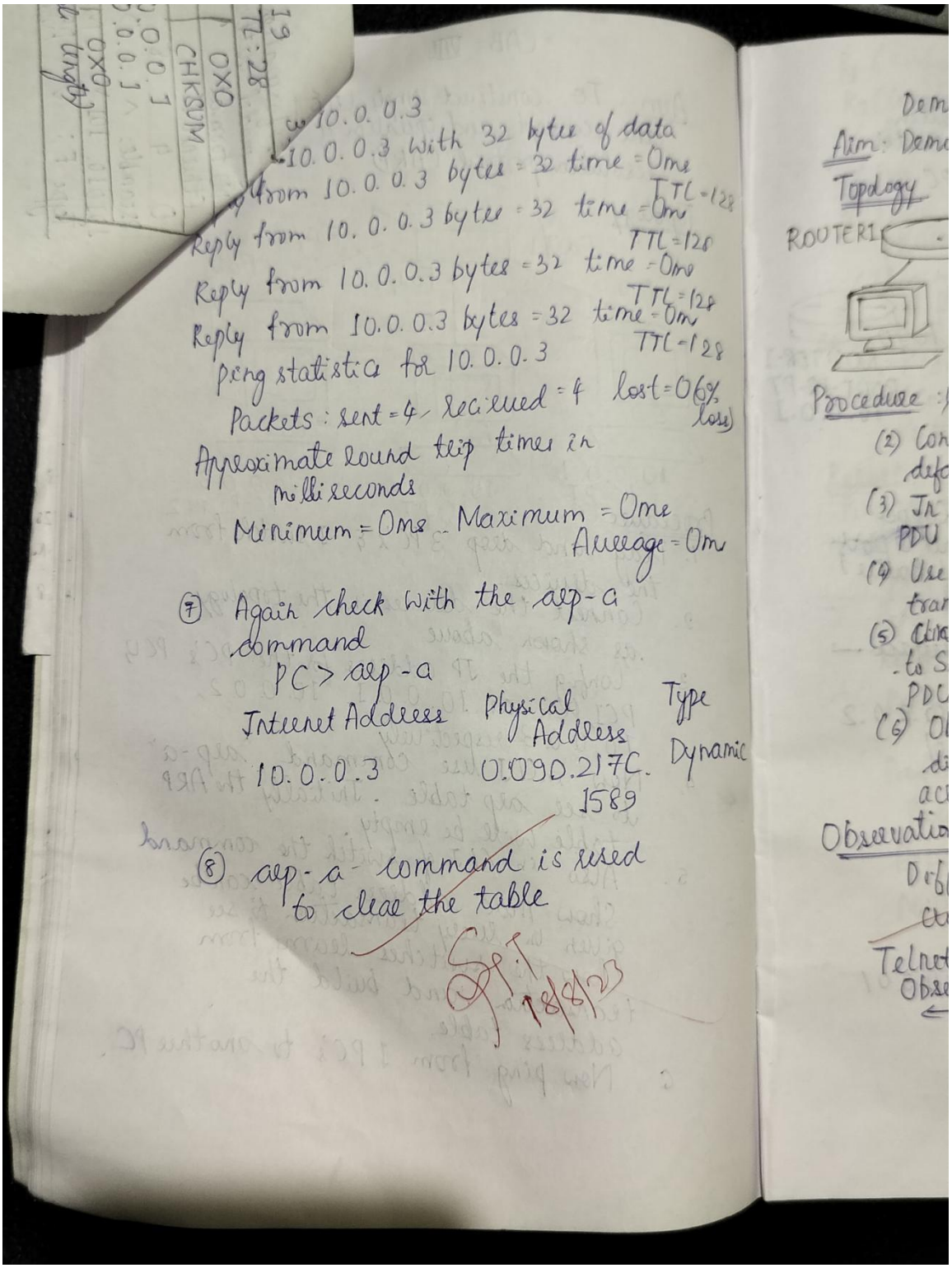
Aim : To construct simple LAN and understand the concept and operation of address resolution protocol (ARP)

Topology



Procedure

1. Drag and drop 3 PC's & 1 switch from the devices.
2. Connect the devices in the topology as shown above
3. Config the IP address for the PC's PC0, PC1, PC2 as 10.0.0.1, 10.0.0.2, 10.0.0.3 respectively
4. Now in CLI use command "arp -a" to see arp table. Initially the ARP table will be empty.
5. Also in CLI of switch the command Show mac address table can be given on every teansaction to see how the switches learns from teansaction and build the address table.
6. Now ping from 1 PC's to another PC.



10.0.0.3
10.0.0.3 with 32 bytes of data
from 10.0.0.3 bytes = 32 time = 0ms TTL = 128
Reply from 10.0.0.3 bytes = 32 time = 0ms TTL = 128
Reply from 10.0.0.3 bytes = 32 time = 0ms TTL = 128
Reply from 10.0.0.3 bytes = 32 time = 0ms TTL = 128
ping statistics for 10.0.0.3
Packets: sent = 4, received = 4 lost = 0 (0% loss)
Approximate round trip times in milliseconds
Minimum = 0ms Maximum = 0ms Average = 0ms

(7) Again check with the arp -a command

PC > arp -a

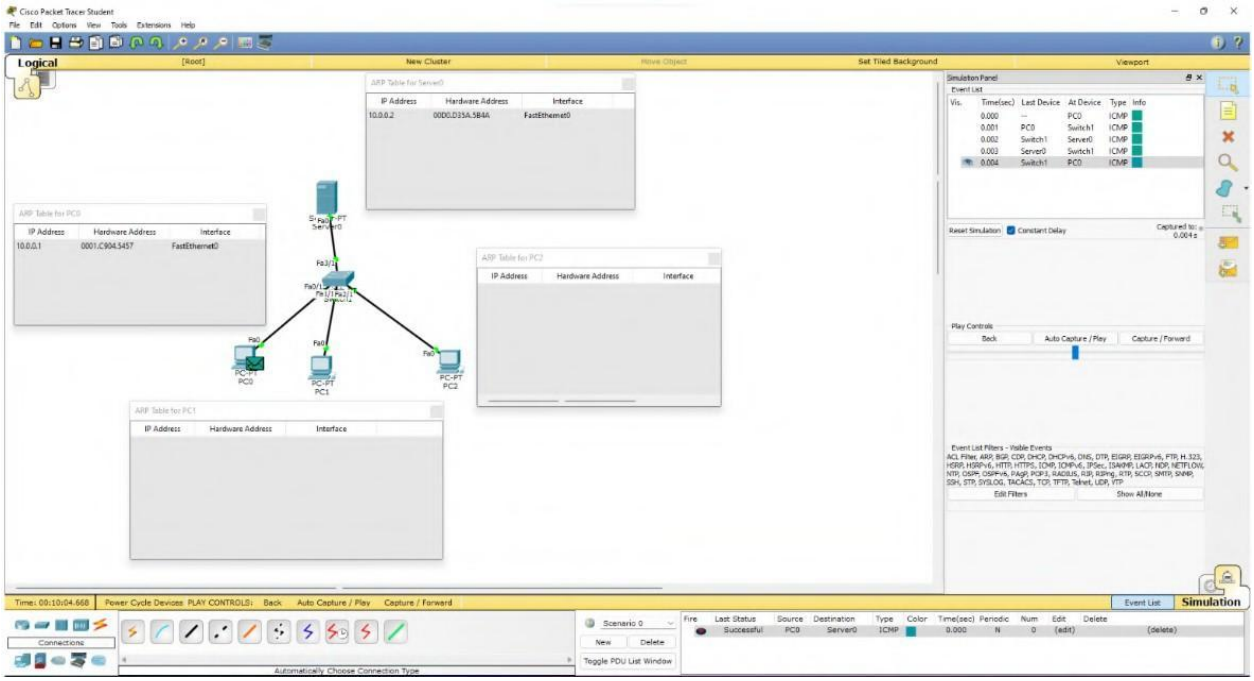
Internet Address	Physical Address	Type
10.0.0.3	0.090.217C.1589	Dynamic

(8) arp -a command is used to clear the table

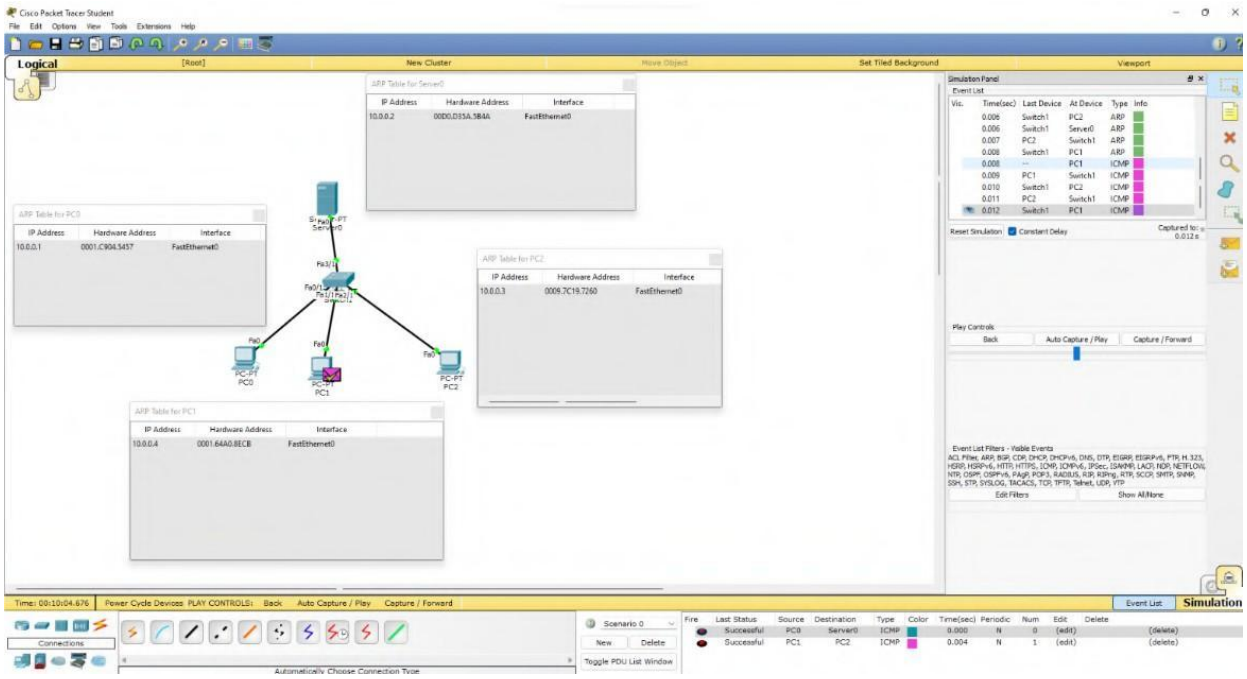
S.J. 18/8/23

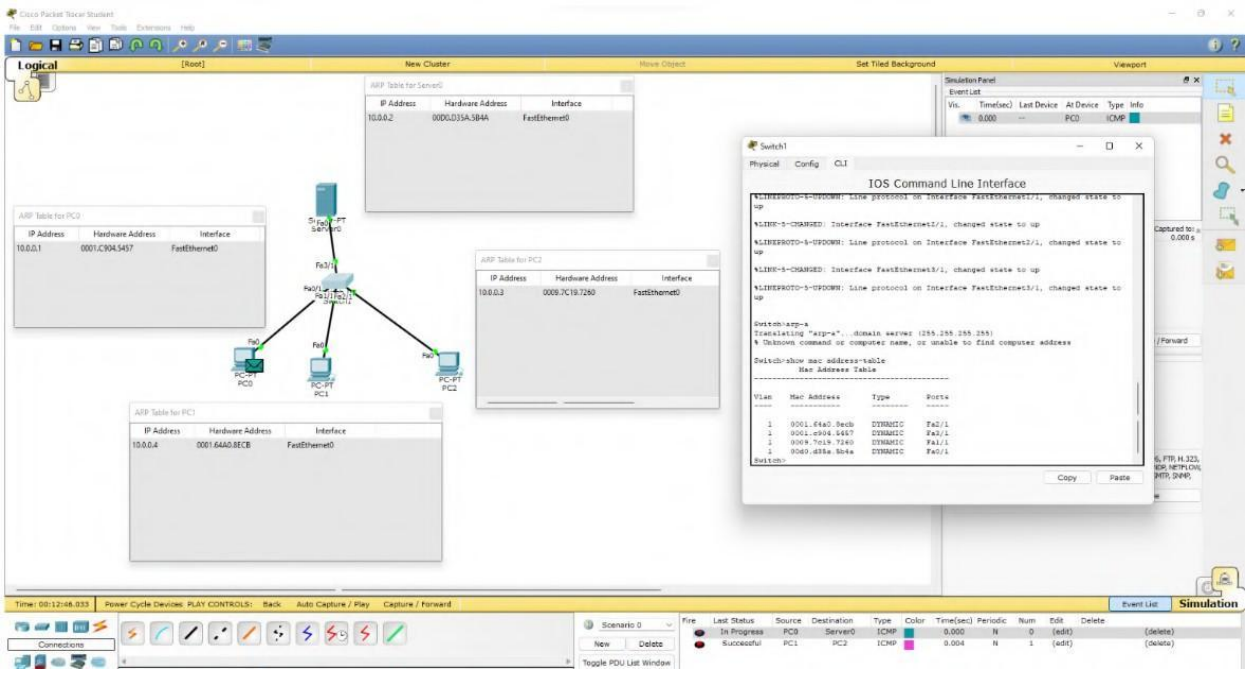
Aim: Demo
Topology
ROUTER1
Procedure :
(2) Con
defa
(3) In
PDU
(4) Use
tran
(5) Che
to S
PDU
(6) O
di
ac
Observatio
Defi
co
Telnet
Obse

TOPOLOGY:



OUTPUT:

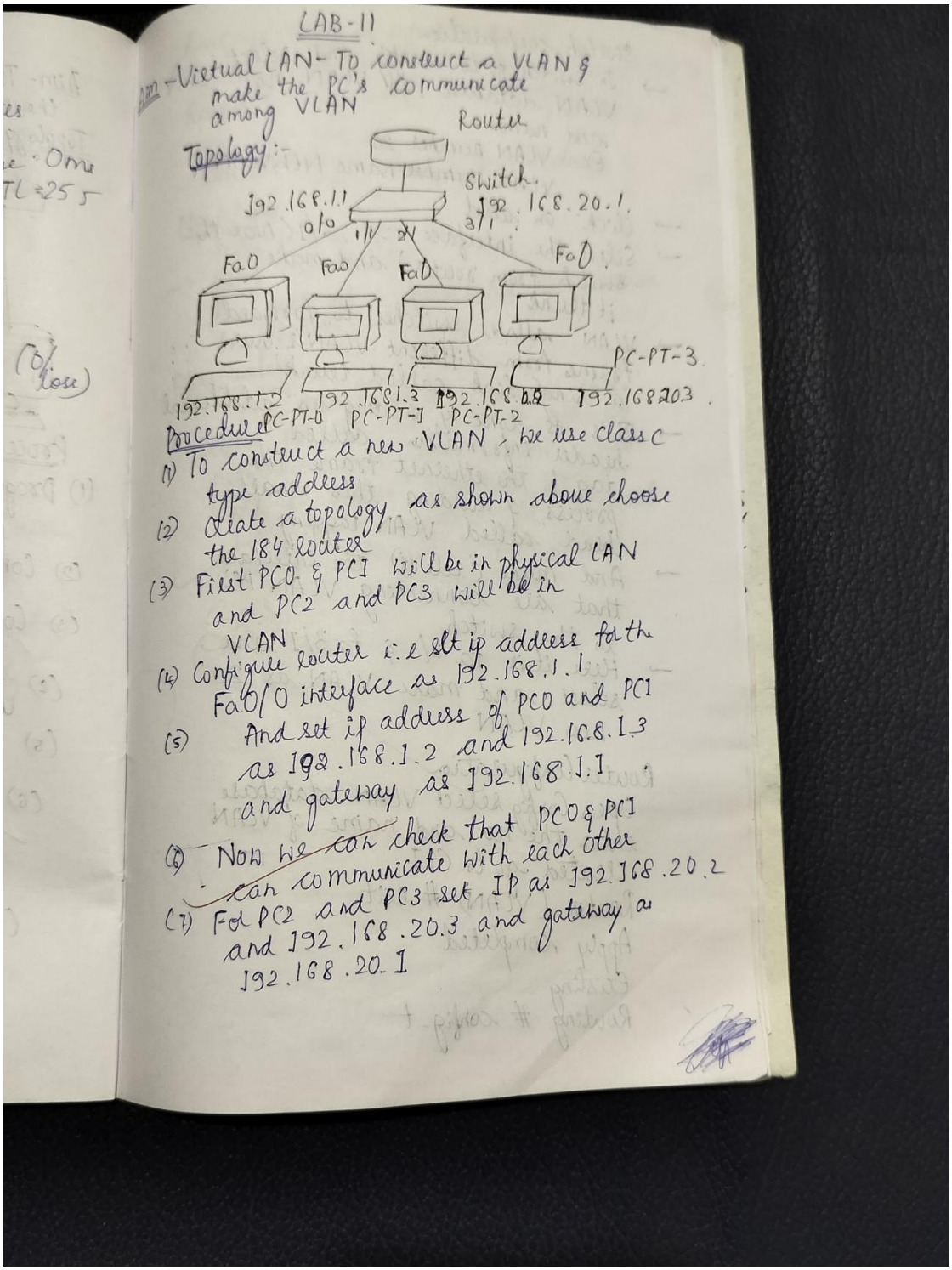



Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
IOS Command Line Interface
Switch1
Physical
Config
CLI
Copy
Paste
Simulation
Event List

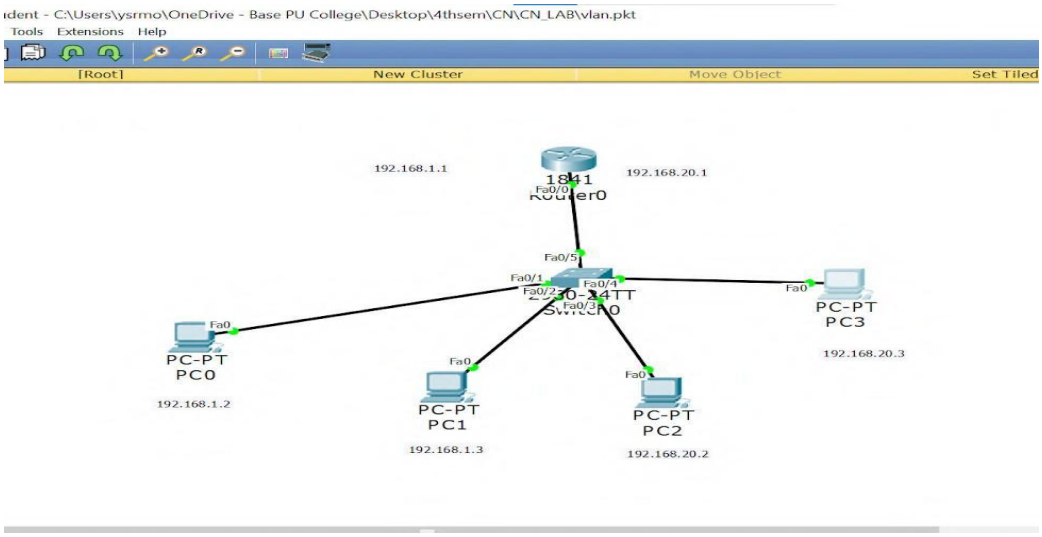
WEEK 9

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

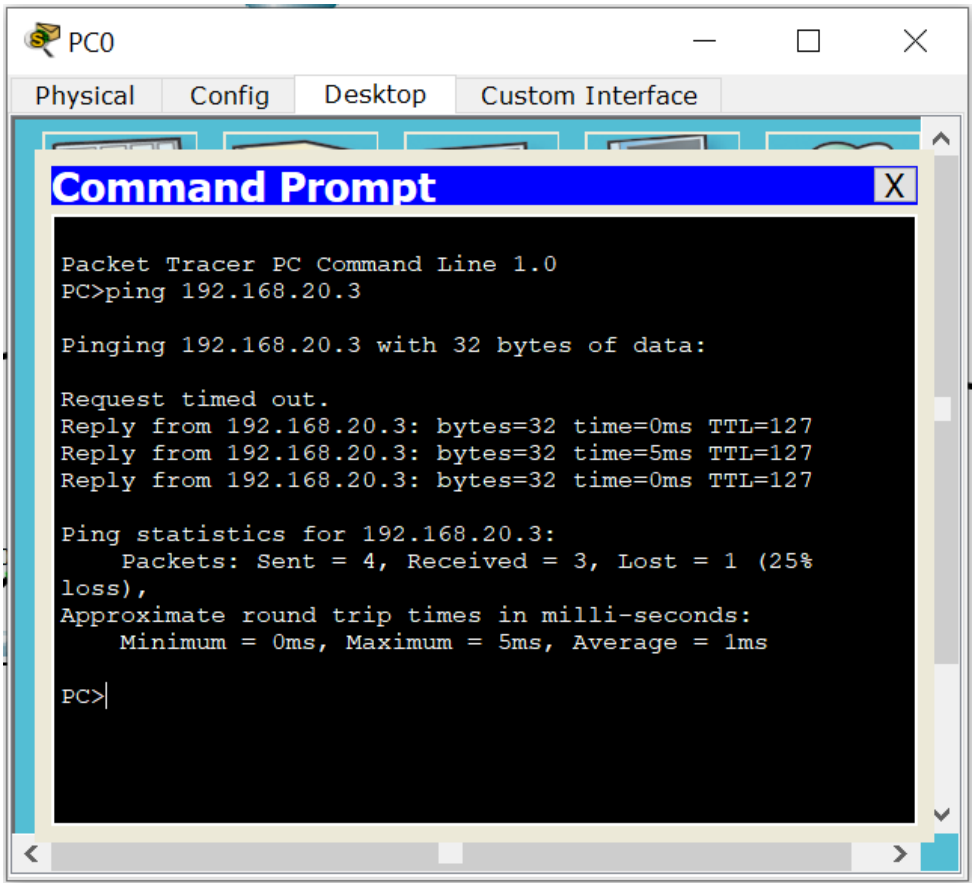
OBSERVATION:

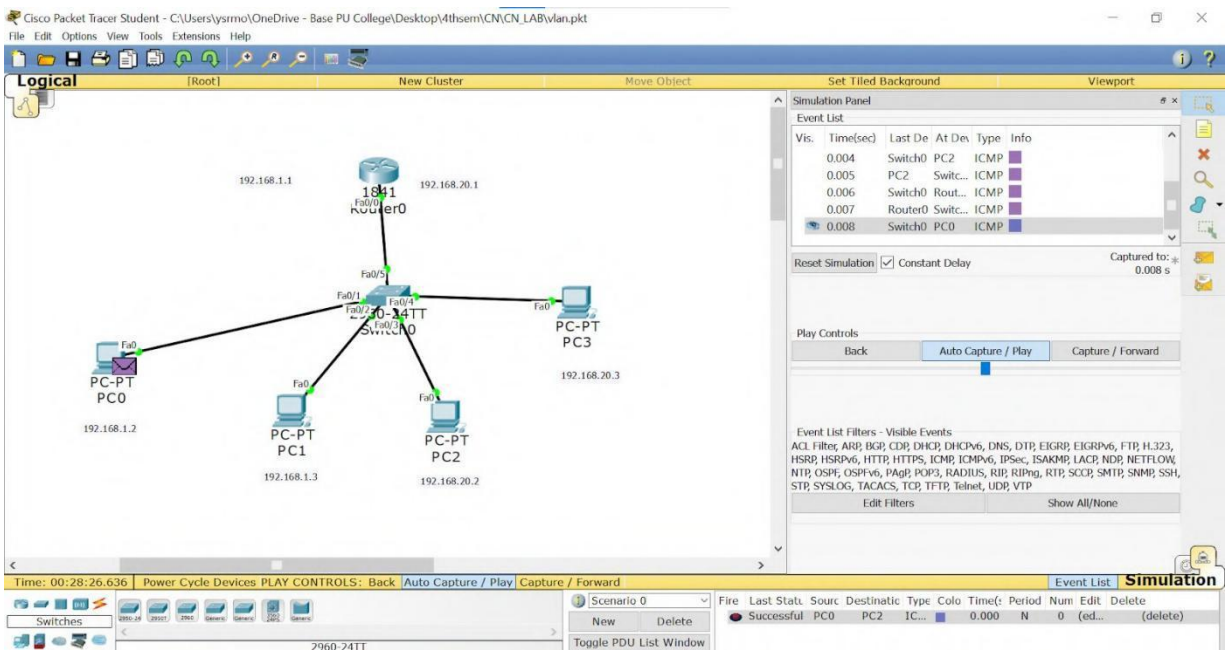


TOPOLOGY:


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
192.168.20.1
Fa0/5
Fa0/1
Fa0/4
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:

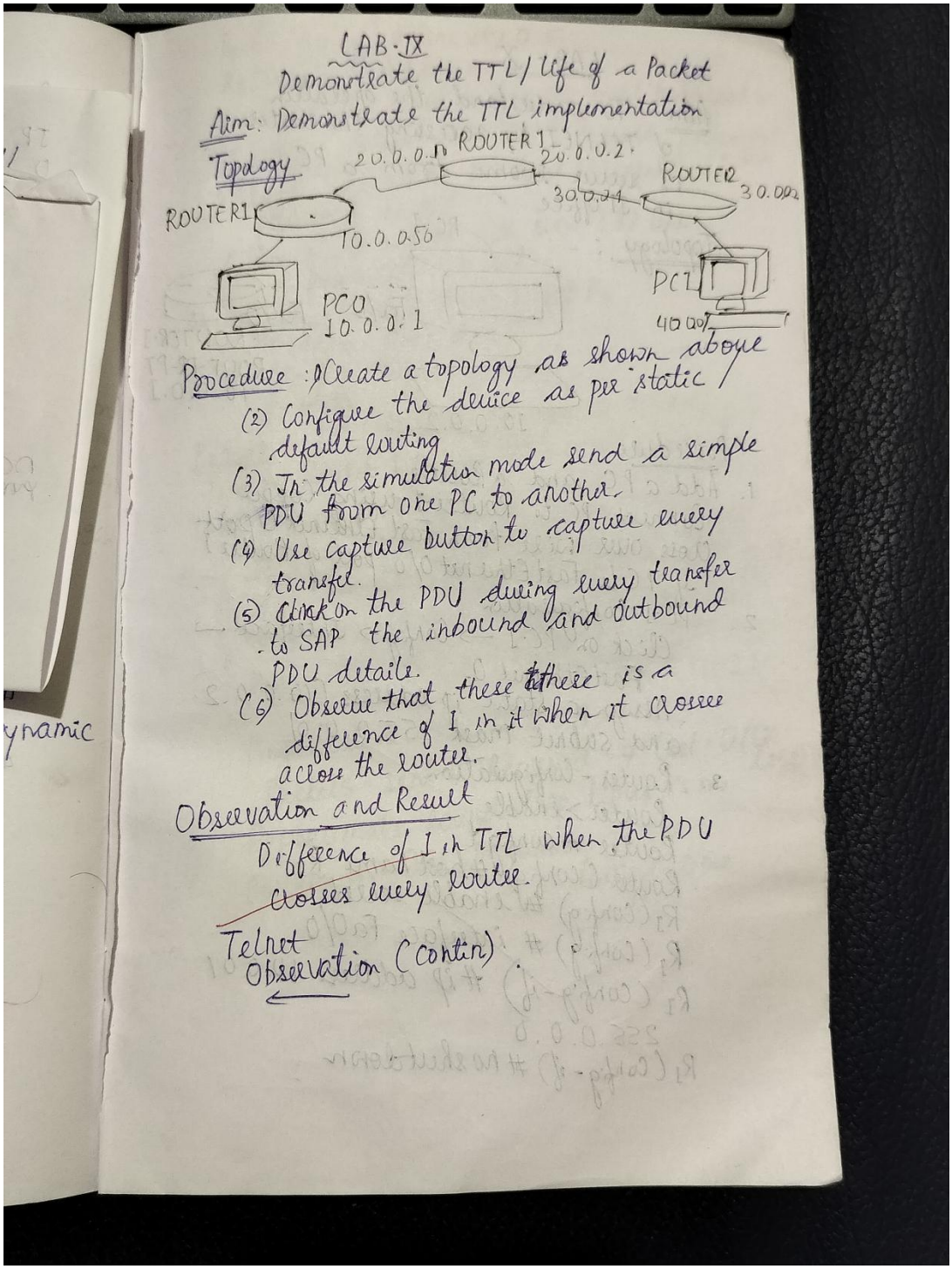

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 192.168.20.3
Pinging 192.168.20.3 with 32 bytes of data:
Request timed out.
Reply from 192.168.20.3: bytes=32 time=0ms TTL=127
Reply from 192.168.20.3: bytes=32 time=5ms TTL=127
Reply from 192.168.20.3: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


Logical
PC-PT PC0
PC-PT PC1
PC-PT PC2
PC-PT PC3
192.168.1.1
192.168.20.1
192.168.1.2
192.168.1.3
192.168.20.2
192.168.20.3
Simulation Panel
Event List
Reset Simulation
Constant Delay
Play Controls
Back
Auto Capture / Play
Capture / Forward
Edit Filters
Show All/None
Simulation

WEEK 10

Demonstrate the TTL/ Life of a Packet.

OBSERVATION:



Observation

PDU information at Device: PCU
OSI model outbound PDU details
PDU format
Ethernet II

0 4	8 14	19 Bytes.
Preamble: 101010...101 Type F:	DEST MAC	SRC MAC
	DATA (Variable)	FCS

0 4 8 16	31 Bytes
4 IHL DSCP	TL: 28
ID: 0X7 0X	0X0
TTL: 255 PRO: 0X1	CHKSUM
SCR IP: 10.0.0.1	
DEST IP: 40.0.0.1	
OUPT 0X0	0X0
DATA (Variable length)	

ICMP

0 8 16	31 Bits
TYPE CODE	CHECKSUM
ID: 0X8	SEQNUMBER: 7

PDU Information at Device Router I
OSI Model Inbound PDU details Outbound PDU

PDU: Formats

Ethernet II

0 4 8 14	19 Bytes
Preamble: 101010: TYPE:	DEST MAC	SRC MAC: FCFS:
	DATA (Variable length)	

10.0.0.3 with 32 bytes of data

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

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

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

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

ping statistics for 10.0.0.3

Packets: sent = 4, received = 4 lost = 0 (0% loss)

Approximate round trip times in milliseconds

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

(7) Again check with the arp -a command

PC > arp -a

Internet Address	Physical Address	Type
10.0.0.3	0.090.217C.1589	Dynamic

(8) arp -a command is used to clear the table

SJT 18/8/23

Aim:

Topolog

ROUTER1

Procedu

(2)

(3)

(4)

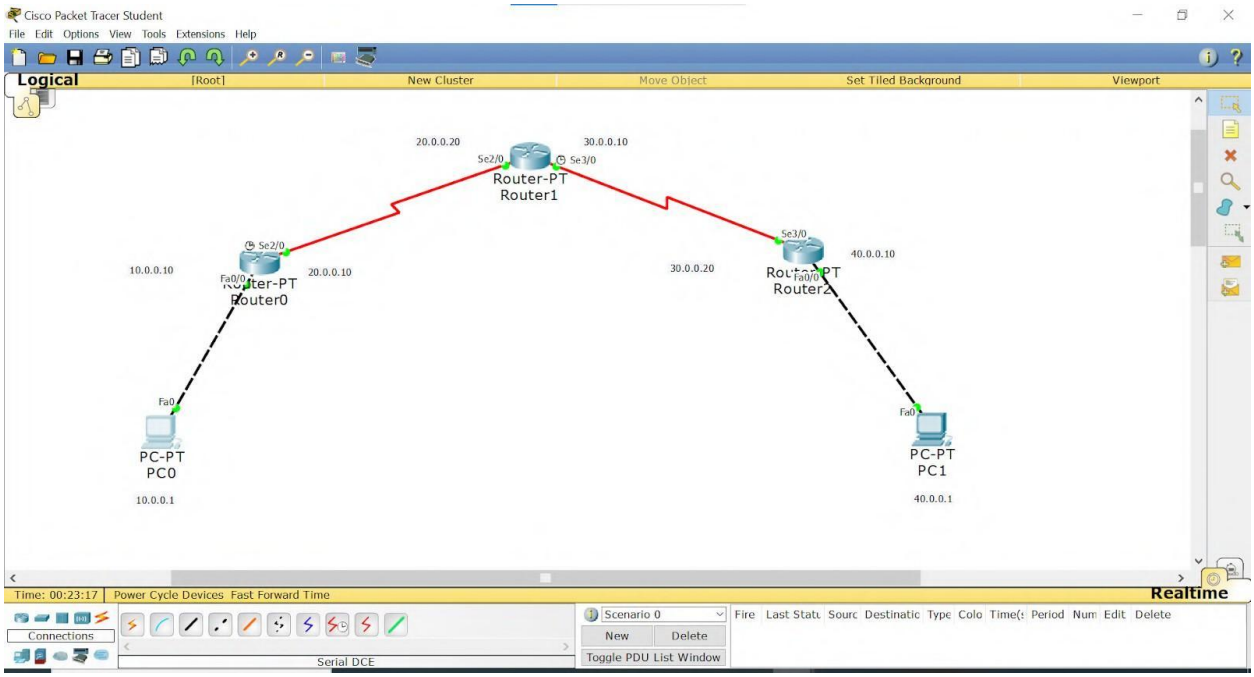
(5)

(6)

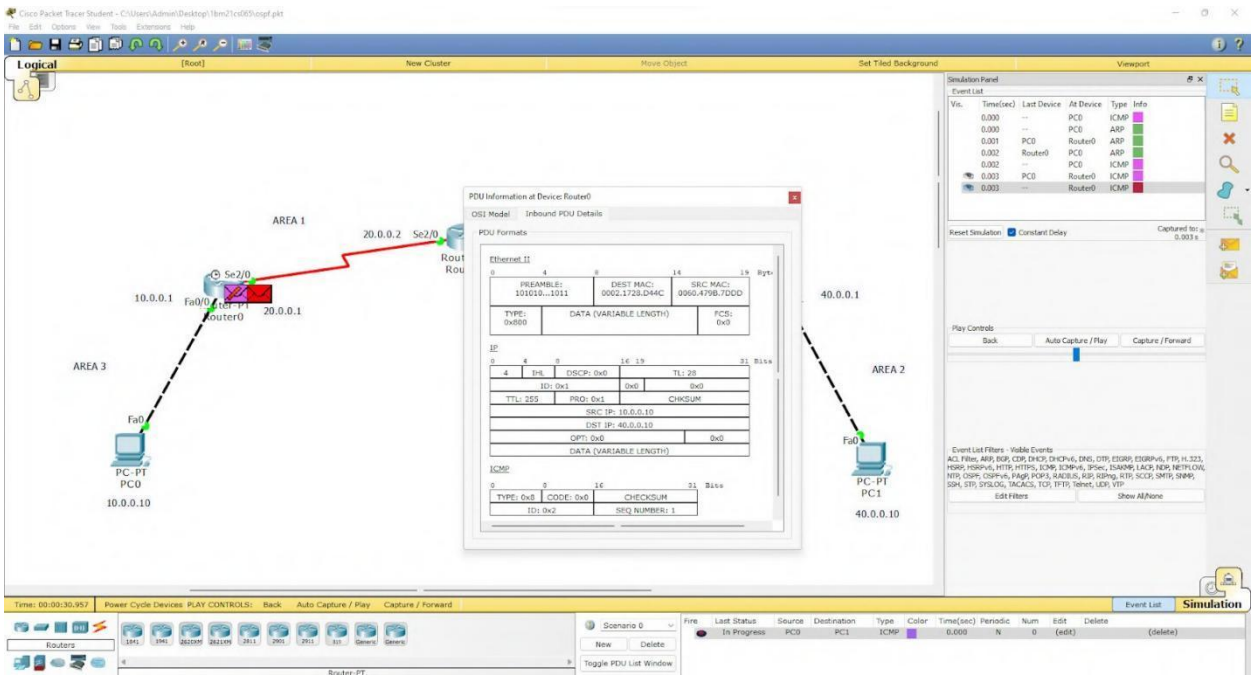
Observa

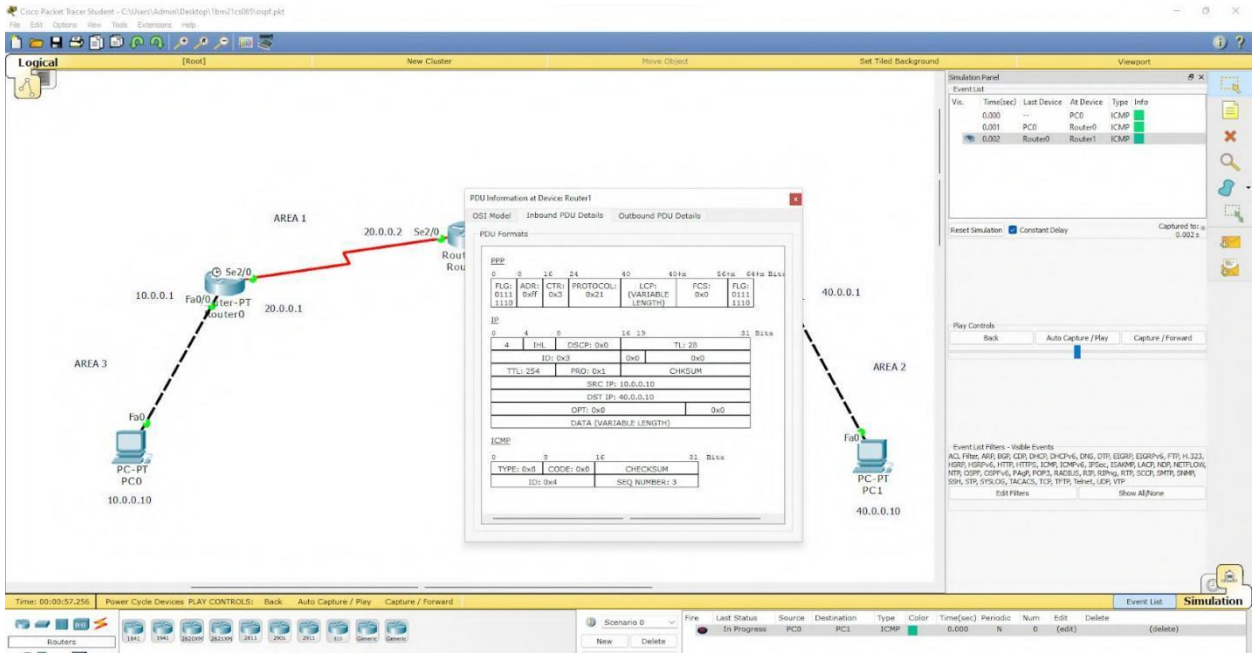
Te

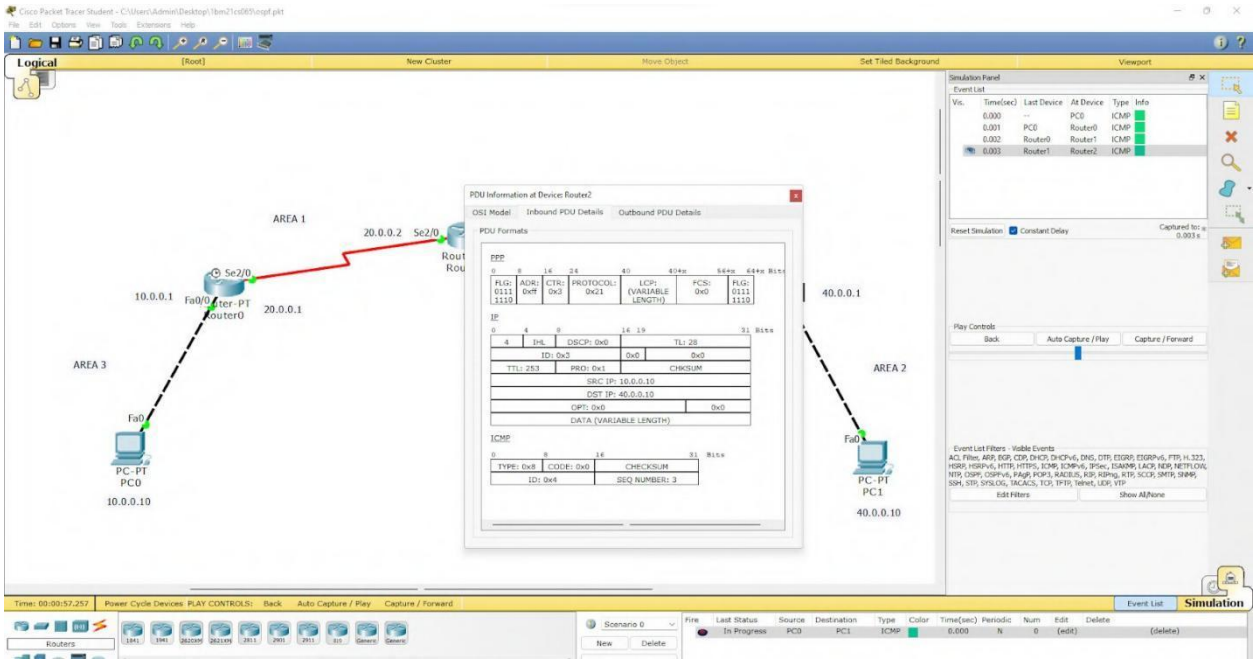
TOPOLOGY:

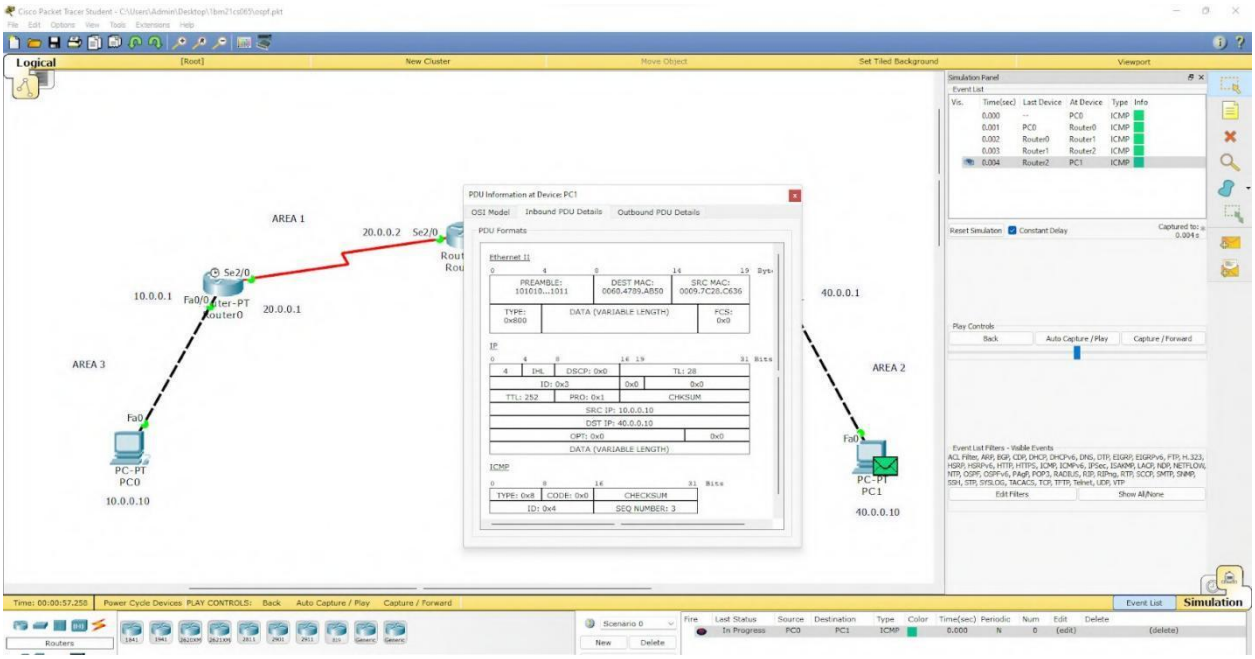


OUTPUT:




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
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x4
SEQ NUMBER: 3
Simulation Panel
Event List
Simulation

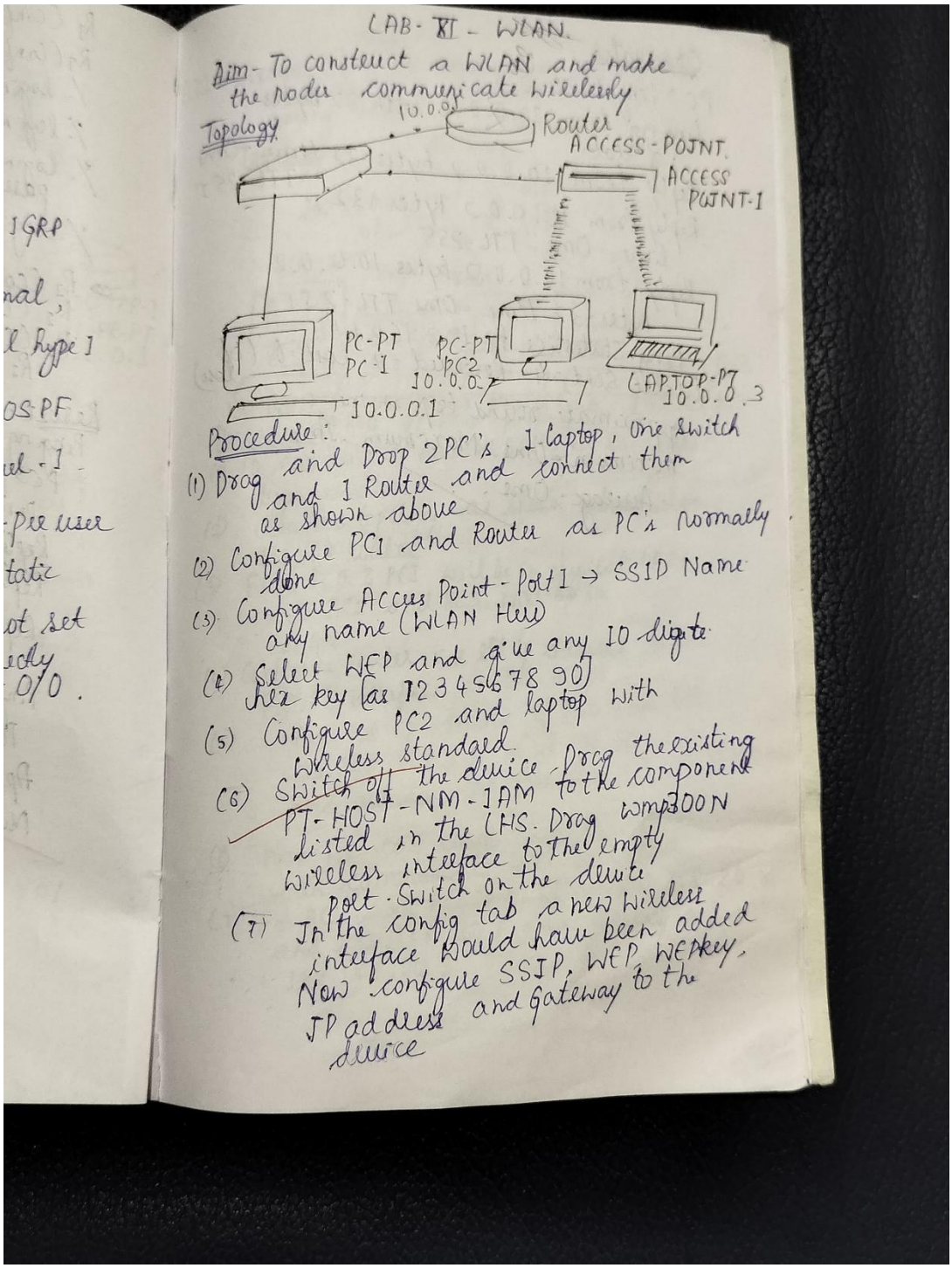

Logical
AREA 1
AREA 2
AREA 3
20.0.0.2
Se2/0
10.0.0.1
Fa0/0
20.0.0.1
40.0.0.1
Fa0
PC-PT
PC0
10.0.0.10
PC-PT
PC1
40.0.0.10
PDU Information at Device: Router2
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
PPP
IP
ICMP
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x4
SEQ NUMBER: 3
Simulation Panel
Event List
Simulation


Logical
AREA 1
20.0.0.2
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
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
SRC IP: 10.0.0.10
DST IP: 40.0.0.10
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
Simulation

WEEK 11

To construct a WLAN and make the nodes communicate wirelessly

OBSERVATION:



Observation

PC > Ping 10.0.0.2

Pinging from 10.0.0.2 with 32 bytes of data

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

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

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

Ping Statistics for 192.168.1.1

Packet Sent=4 Recieved=4 lost=0 (0% loss)

Approximate round trip in ms

minimum=0ms Maximum=1ms

Average=0ms.

Aim – Virtual LAN
make th
among

Topology :-

192.1

Fa0

192.168.1.2

Procedure PC-

(1) To conste
type a

(2) Create
the 184

(3) First PC
and P
VLAN

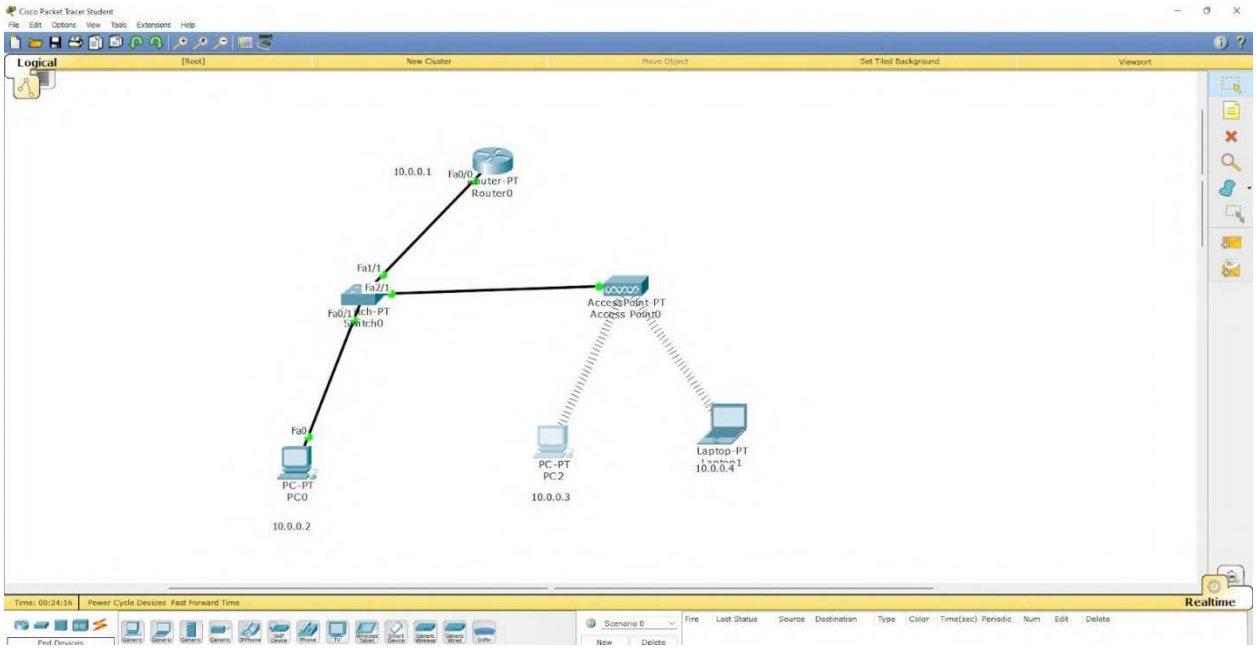
(4) Configure
Fa0/(

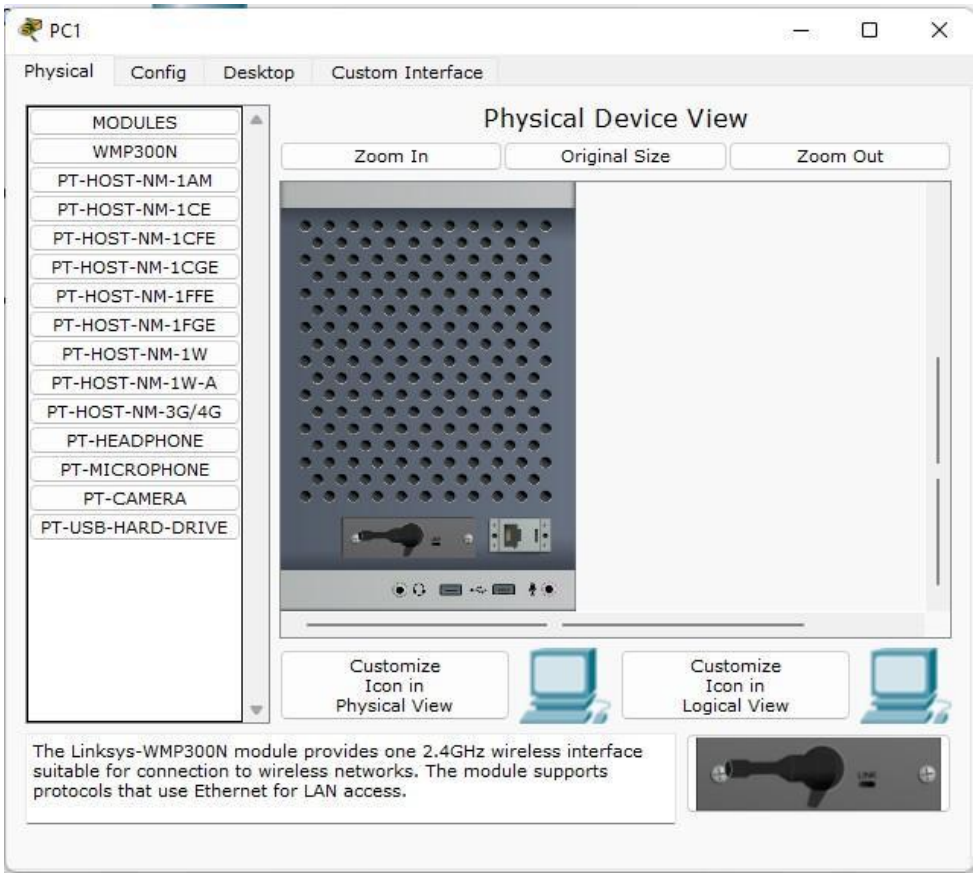
(5) Ano
as
and

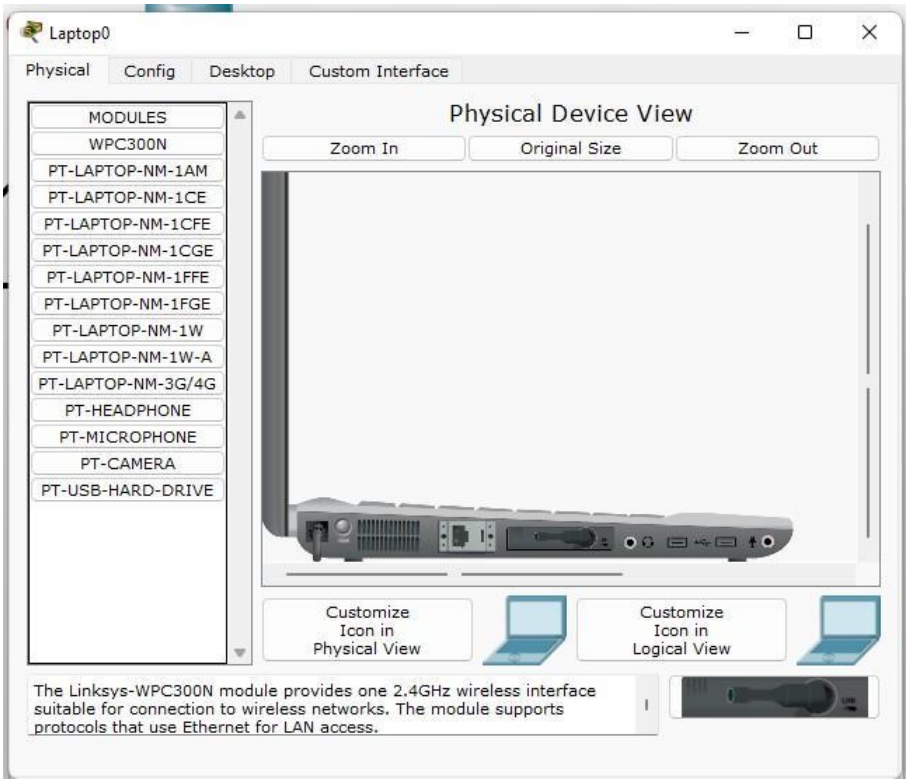
(6) Now
can

(7) For P
and
192

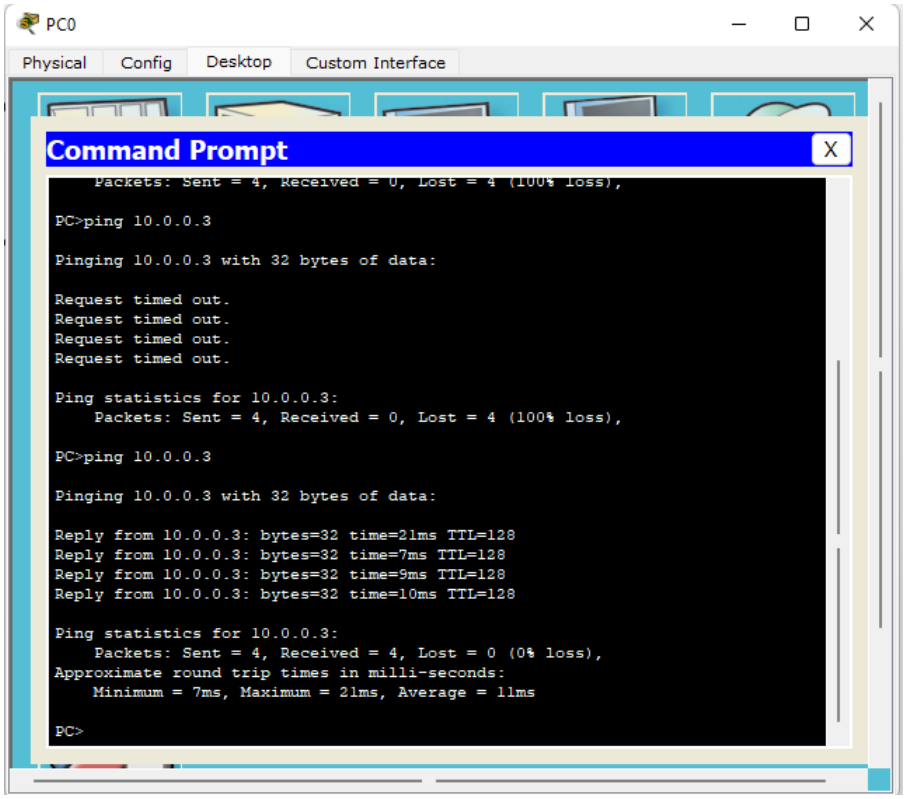
TOPOLOGY:


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


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


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

OUTPUT:


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

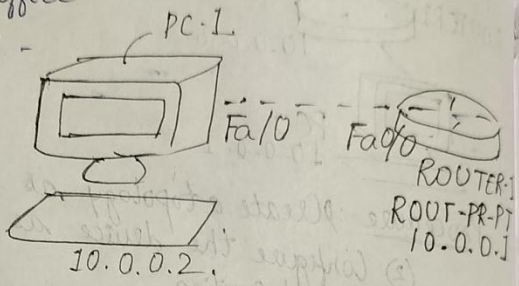
WEEK 12

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

OBSERVATION:

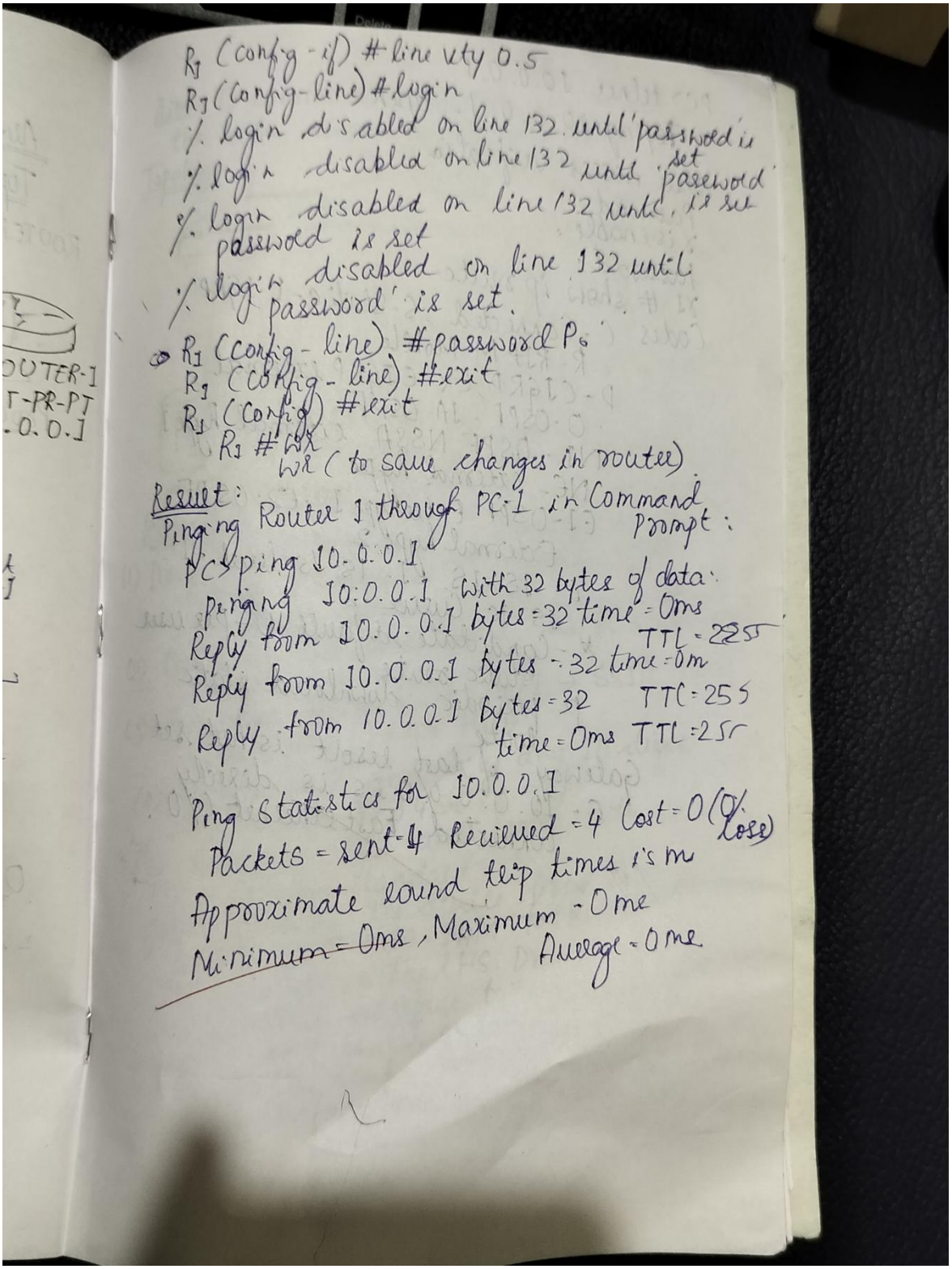
Aim : To understand the operation of TELNET by accessing the router in server rooms from a PC in IT office

Topology :-



Procedure

1. Add a PC and a Router
   Connect PC to Router-1 using Coppie cross over wire from Fast Ethernet 0 port of PC to Fast Ethernet 0/0 port of Router-1
2. PC-1 Configuration
   Click on PC-1 → Config → Interface → FastEthernet 0.
   Assign a static ip address [10.0.0.2 and subnet mask 255.0.0.0]
3. Router - Configuration
   Router > enable
   Router # config t
   Router (config) # host name R_1
   R_1 (config) # enable secret
   R_1 (config) # interface Fa0/0
   R_1 (config-if) # ip address 10.0.0.1 255.0.0.0
   R_1 (config-if) # no shutdown



R1 (config-if) # line vty 0 5
R1 (config-line) # login
% login disabled on line 132 until 'password is
% login disabled on line 132 until 'password set'
% login disabled on line 132 until 'is set password is set
% login disabled on line 132 until password' is set.
R1 (config-line) # password P0
R1 (config-line) # exit
R1 (config) # exit
R1 # wr
wr (to save changes in router)

Result:
Pinging Router 1 through PC-1 in Command Prompt:

PC> ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1 bytes=32 time=0ms TTL=255
Reply from 10.0.0.1 bytes=32 time=0m
Reply from 10.0.0.1 bytes=32 TTL=255
time=0ms TTL=25

Ping statistics for 10.0.0.1
Packets = sent=4 Recieved=4 Lost=0 (0% loss)
Approximate round trip times in m
Minimum=0ms, Maximum=0ms
Average=0ms

PC> telnet 10.0.0.1
Trying 10.0.0.1.. open
User Access Verification
Password:
R1 > enable:
Password:
R1 # show ip route
Codes C - Connected, S - Static, I IGRP
R - RIP M - Mobile
D - CJGRP, EX - EIGRP external,
O - OSPF - IA - OSPF
N1 - OSPF NSSA external type 1
N2 - external type
E1 - OSPF Exte type 1 . E2 - OS PF
External type 2
i - IS-IS L1 - IS-Is level-1 -
IS-Is Cueue-2
* - Candidate default U - per user
static route
P - periodic download static
route
Gateway of last resort is not set
C - 10.0.0.0/0 is is directly
connected, Fast ethernet 0/0.

Aim - T
the
Topology

Proce

(1) Drag

(2) Con

(3) Co

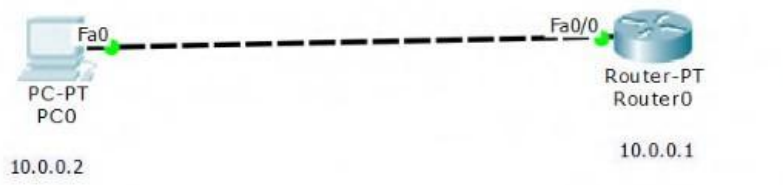
(4)

(5)

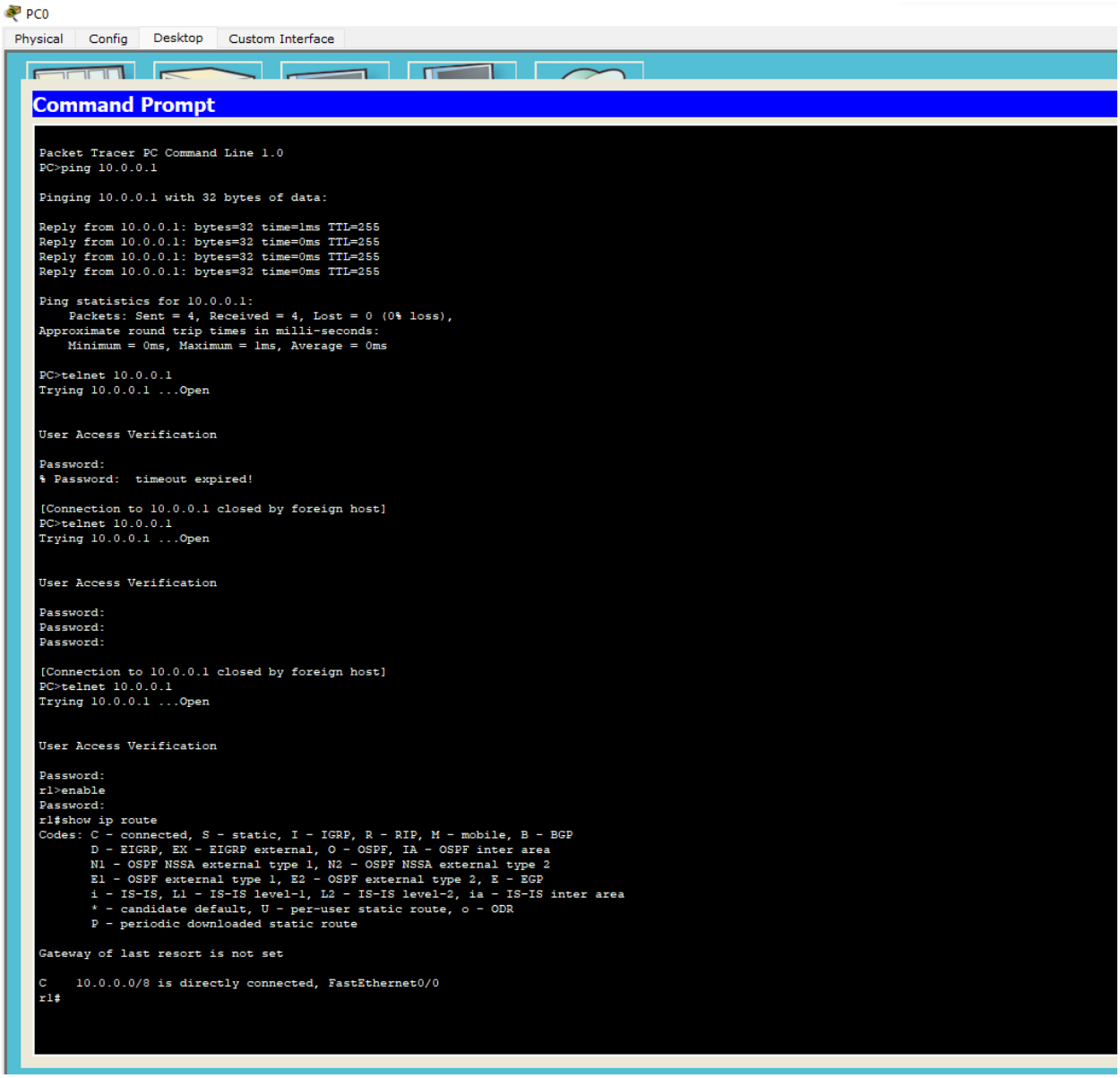
(6)

(7

TOPOLOGY:



OUTPUT:


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=1ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open
User Access Verification
Password:
% Password: timeout expired!
[Connection to 10.0.0.1 closed by foreign host]
PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open
User Access Verification
Password:
Password:
Password:
[Connection to 10.0.0.1 closed by foreign host]
PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open
User Access Verification
Password:
rl>enable
Password:
rl#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route
Gateway of last resort is not set
C 10.0.0.0/8 is directly connected, FastEthernet0/0
rl#

WEEK 13

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

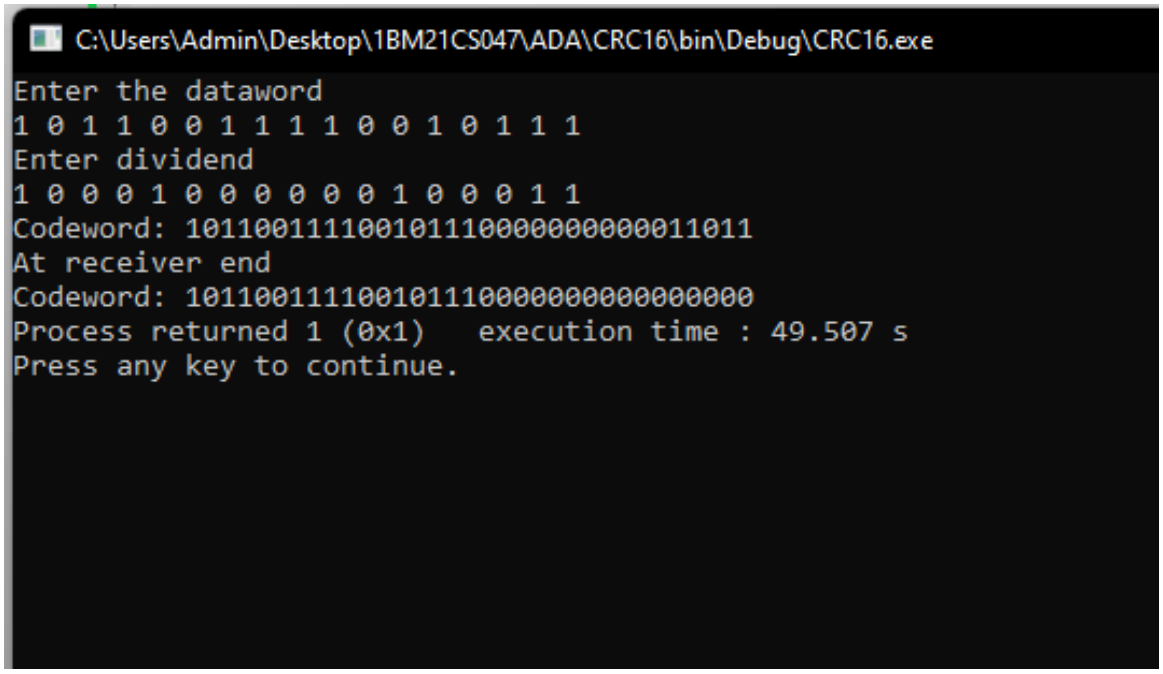
CODE:

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

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

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

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

   for(i=i;i<33;i++)
      div[i]=0;

   for(i=0;i<17;i++)
      ze[i]=0;
   printf("Enter dividend \n");
```

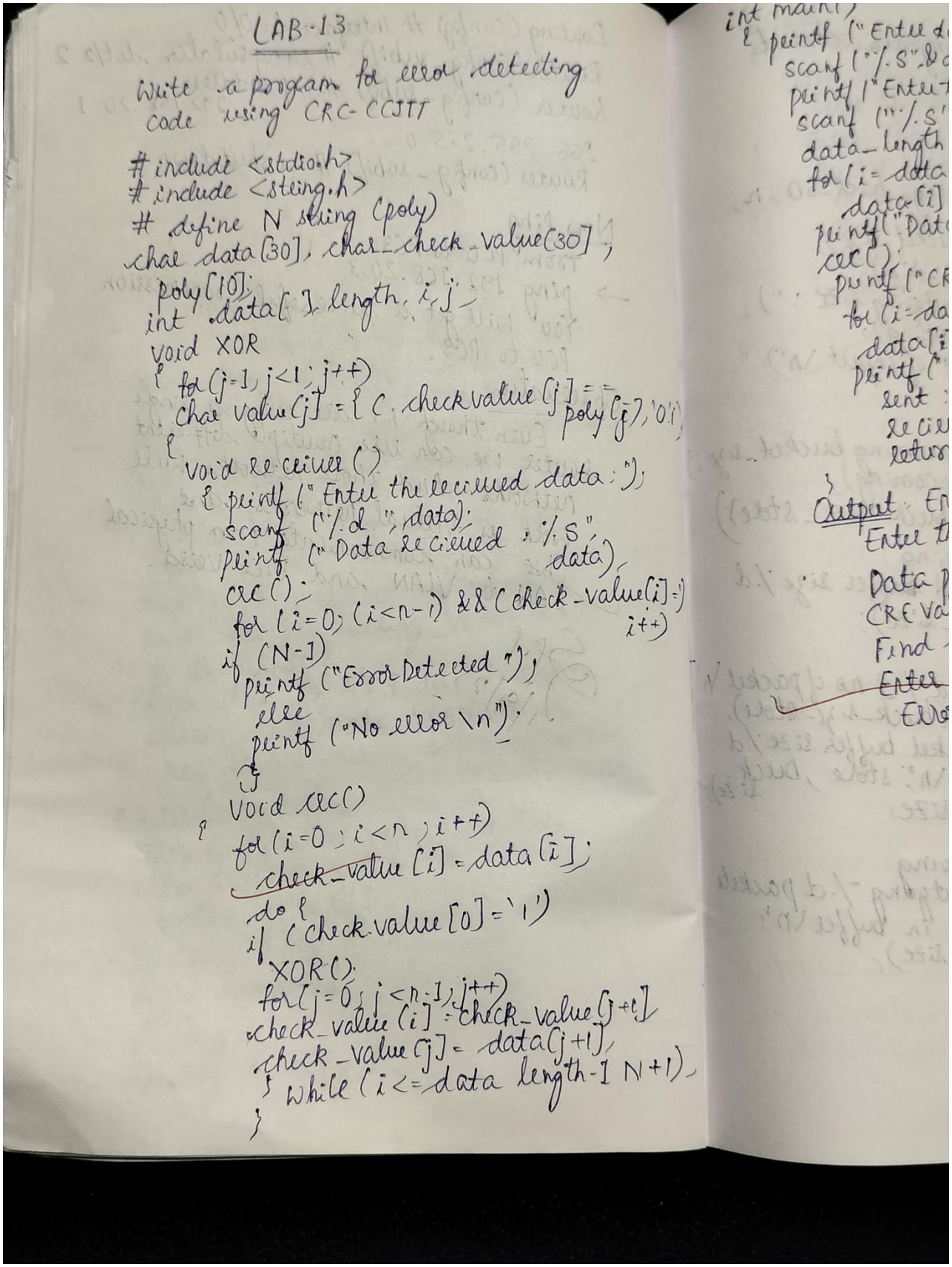
```
for(i=0;i<17;i++)
   scanf("%d",&dd[i]);

i=0;
k=0;
   for(i=i;i<17;i++)
      arr[k++]=div[i];
while(i<33)
{
   if(arr[0]==0)
      xor(arr,ze);
   else
      xor(arr,dd);

   arr[16]=div[i++];

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

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


 k=0;


   for(i=i;i<17;i++)
      arr[k++]=div[i];
while(i<33)
{
```

```
        if(arr[0]==0)
            xor(arr,ze);
        else
            xor(arr,dd);

        arr[16]=div[i++];

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

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

OUTPUT:



OBSERVATION:



LAB-13

Write a program for error detecting code using CRC-CCITT

```
#include <stdio.h>
#include <string.h>
#define N strlen(poly)
char data[30], check_value[30];
poly[10];
int data[], length, i, j;
void XOR
{ for(j=1; j<1; j++)
  char value[j] = { C. check value[j] == poly[j], '0':'1';
{
  void receiver()
  { printf("Enter the received data:");
    scanf("%d", data);
    printf("Data received : %s", data);
    crc();
    for(i=0; (i<n-1) && (check_value[i]=='0'); i++)
    if (N-1)
    printf("Error Detected");
     else
     printf("No error\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[i] = check_value[j+1]
    check_value[j] = data[j+1];
    } while(i<= data length-1 N+1);
  }
```

```
int main()
{ printf("Enter d
  scanf("%s", &d
  printf("Enter
  scanf("%s"
  data_length
  for(i=data
    data[i]
  printf("Dat
  crc();
  printf("CR
  for(i=da
    data[i
  printf(
    sent
    rec
    retur
}
Output: En
  Enter t
  Data
  CRC va
  Find
  Enter
  Err
```

```
int main()
{ printf ("Enter data to be transmitted ");
  scanf ("%s",&data);
  printf ("Enter the divisor polynomial:")
  scanf ("%s", +poly);
  data_length = strlen (data);
  for (i= data_length ; i < data_length +N-1; i++)
     data[i] = 0);
  printf ("Data padded with n-1 zeros:%s",
                                    data);
  crc();
  printf ("CRC value is:%s", check_value);
  for (i=data_length ; i< data_length+N-1;
  data[i]= checkvalue [i - data_length];
  printf ("Final datawoed to be
     sent : %s", data)
     receiver();
     return0);
}
```

Output: Enter the data to be transmitted: 101010
Enter the divisor polynomial 101
Data padded with 0's: 101010 00
CRC value is : 001
Find codeword to be sent : 10101001
Enter the receiver data: 10001000
Error detected

WEEK 14

Write a program for congestion control using Leaky bucket algorithm.

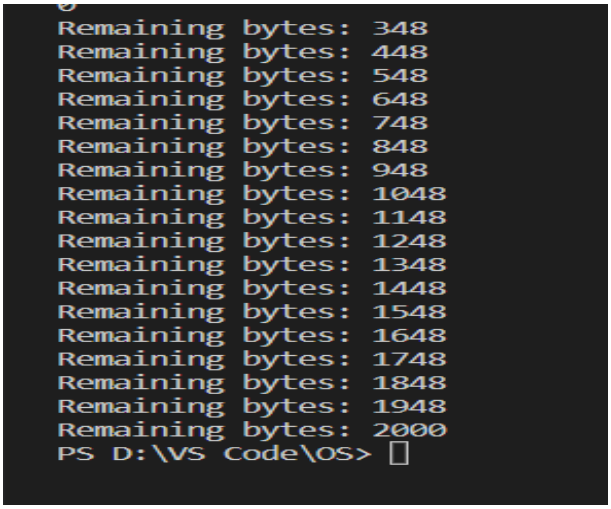
CODE:

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

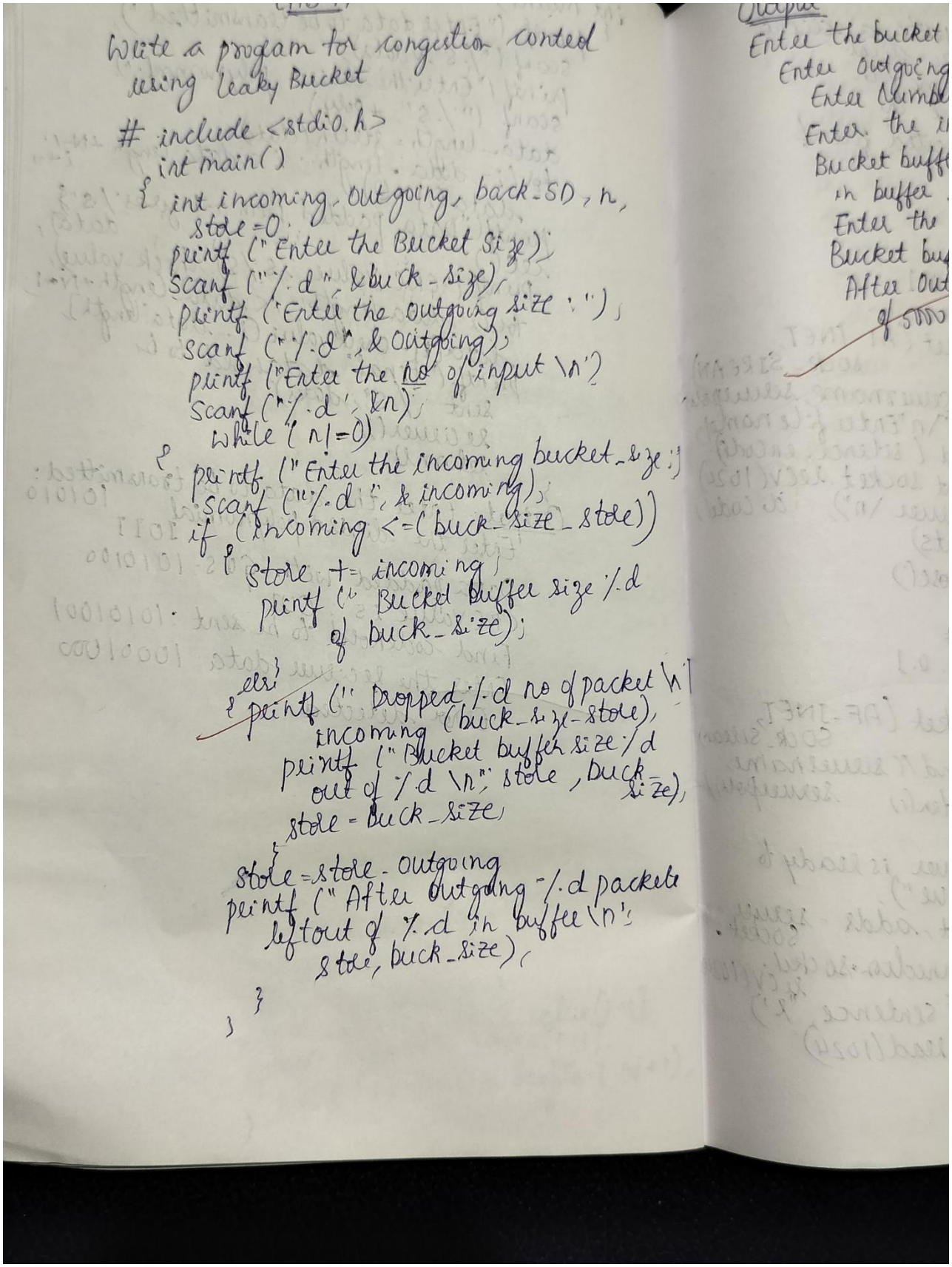
```
    }
    while (remaining < buckets) // Fixed the condition
    {
        if (buckets - remaining > outlets)
        {
            remaining += outlets; // Fixed the calculation
        }
        else
            remaining = buckets;
        printf("Remaining bytes: %d \n", remaining);
    }
  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

OUTPUT:

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


Remaining bytes: 348
Remaining bytes: 448
Remaining bytes: 548
Remaining bytes: 648
Remaining bytes: 748
Remaining bytes: 848
Remaining bytes: 948
Remaining bytes: 1048
Remaining bytes: 1148
Remaining bytes: 1248
Remaining bytes: 1348
Remaining bytes: 1448
Remaining bytes: 1548
Remaining bytes: 1648
Remaining bytes: 1748
Remaining bytes: 1848
Remaining bytes: 1948
Remaining bytes: 2000
PS D:\VS Code\OS>

OBSERVATION:



Write a program for congestion control using Leaky Bucket

```
# include <stdio.h>
int main()
{ int incoming, outgoing, back_SD, n,
    store=0;
  printf ("Enter the Bucket Size");
  scanf ("%d", &buck_size);
  printf ("Enter the outgoing size :");
  scanf ("%d", &outgoing);
  printf ("Enter the no of input \n")
  scanf ("%d", &n);
   while (n!=0)
{ printf ("Enter the incoming bucket_size");
  scanf ("%d", &incoming);
if (incoming <=(buck_size_store))
{ store += incoming;
   printf (" Bucked buffer size %d
        of buck_size);
}
else
{ printf (" Dropped %d no of packet \n"
        incoming (buck_size-store),
   printf ("Bucket buffer size %d
     out of %d \n", store, buck_size),
   store = buck_size
}
store=store-outgoing
printf ("After outgoing %d packets
   leftout of %d in buffer \n",
      store, buck_size);
}
}
```

Enter the bucket
Enter outgoing
Enter Numb
Enter the i
Bucket buff
in buffer
Enter the
Bucket bu
After Out
of 5000

Output

Enter the bucket size = 5000
Enter outgoing rate = 2000
Enter number of input = 2
Enter the incoming packet size = 3000
Bucket buffer size 3000 out of 5000
in buffer.
Enter the incoming packet size = 1000
Bucket buffer size 2000 out of 5000
After outgoing a packet out
of 5000 in buffer

WEEK 15

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

CODE:

ClientTCP.py

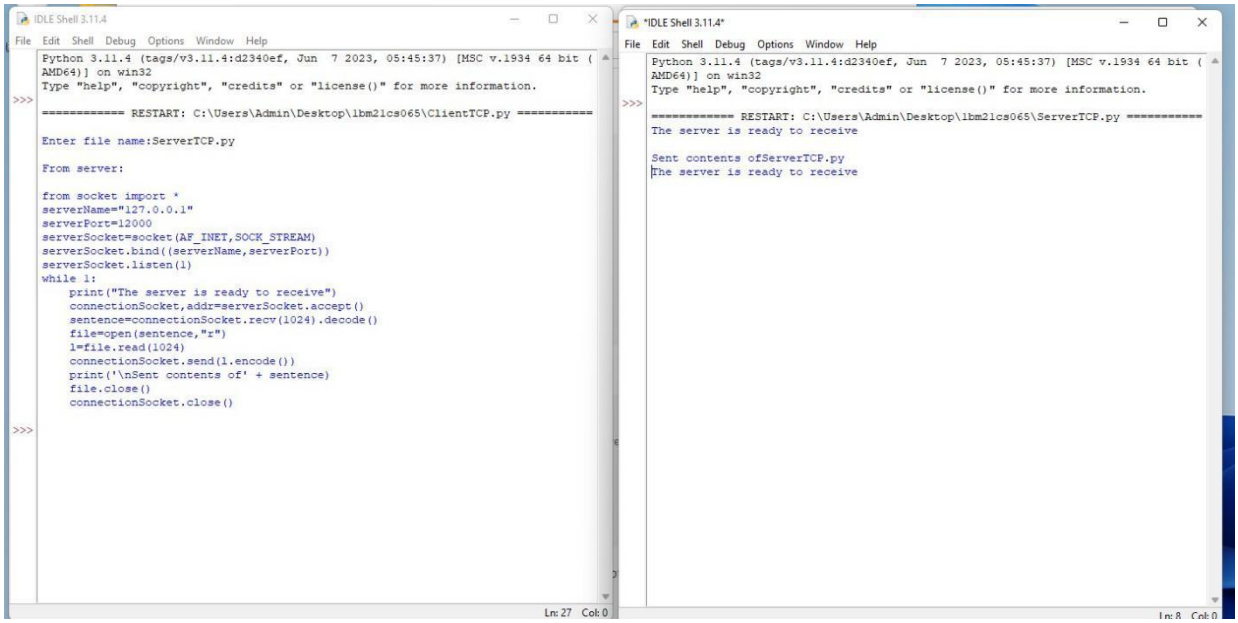
```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input("\nEnter file name: ")
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print ("\nFrom Server:\n")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

ServerTCP.py

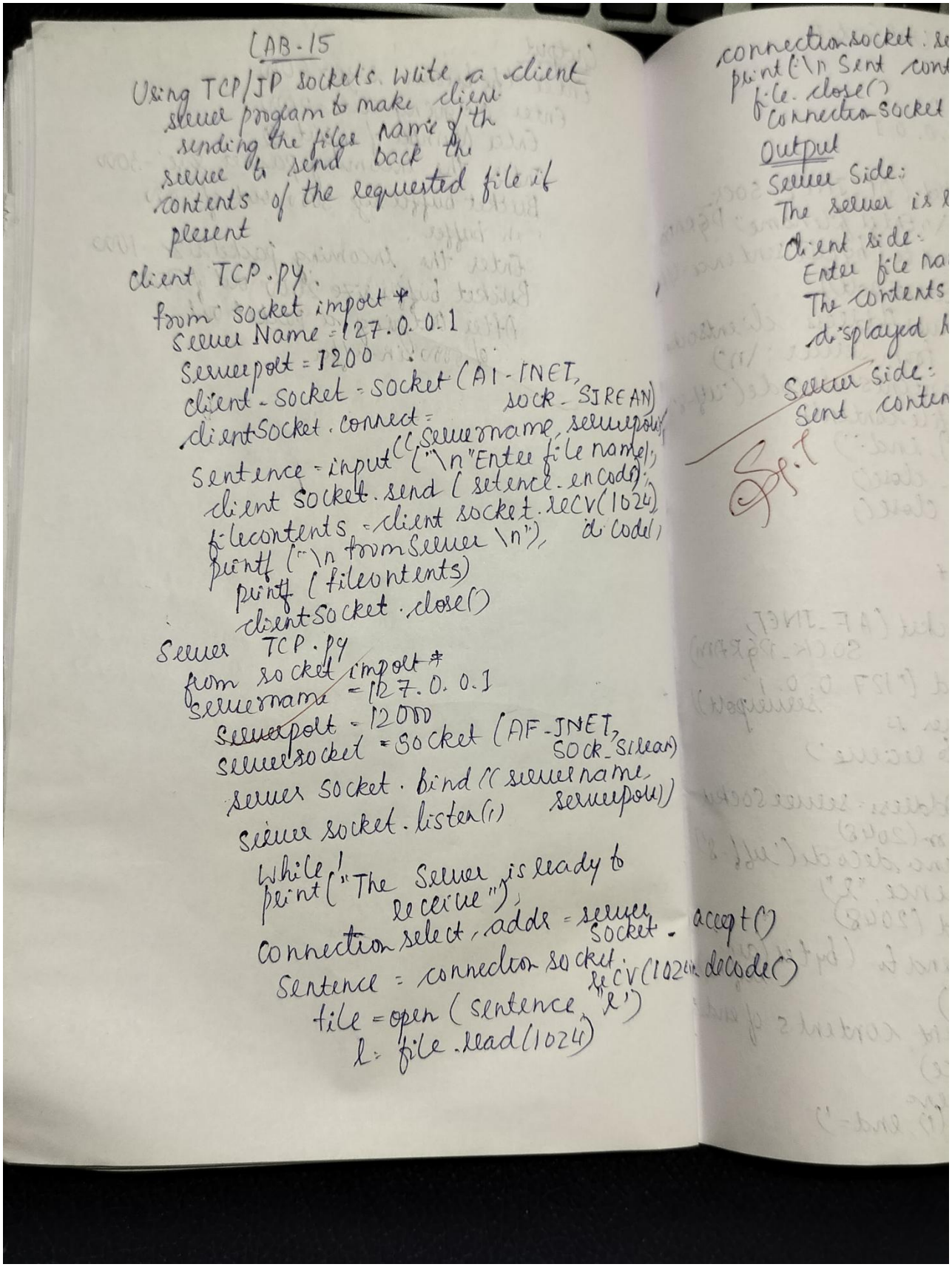
```
from socket import *
serverName="127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
print ("The server is ready to receive")
connectionSocket, addr = serverSocket.accept()
sentence = connectionSocket.recv(1024).decode()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

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

OUTPUT:



OBSERVATION:



LAB-15

Using TCP/IP sockets. Write a client server program to make client sending 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 = 1200
clientSocket = socket(AI-INET, SOCK_STREAM)
clientSocket.connect((ServerName, serverport))
Sentence = input("\n"Enter file name);
clientSocket.send(setence.encode());
filecontents = clientsocket.recv(1024).decode()
print("\n fromServer \n"),
print(filecontents)
clientSocket.close()
```

Server TCP.py

```
from socket import *
Servername = 127.0.0.1
Serverport = 12000
serversocket = Socket(AF-INET, SOCK_STREAM)
server socket.bind((servername, serverport))
server socket.listen(1)
While 1:
print("The Server is ready to receive"),
connection socket, addr = server socket.accept()
Sentence = connectionsocket.recv(1024).decode()
file = open(sentence, 'r')
l = file.read(1024)
```

```
connectionsocket.se
print("\n Sent cont
file.close()
Connection-socket
```

Output

Server Side:
The server is

Client side:
Enter file na
The contents
displayed

Server side:
Sent conten

Sg.T

WEEK 16

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

CODE:

ClientUDP.py

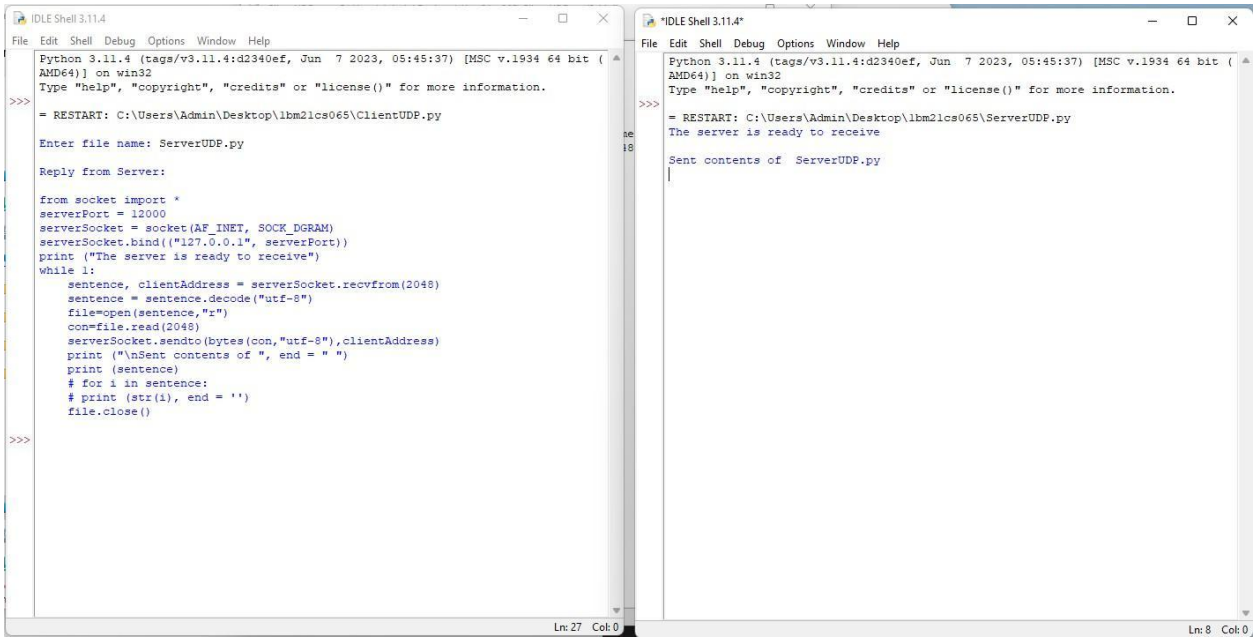
```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("\nEnter file name: ")
clientSocket.sendto(bytes(sentence,"utf-8"),(serverName, serverPort))
filecontents,serverAddress = clientSocket.recvfrom(2048)
print ("\nReply from Server:\n")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

ServerUDP.py

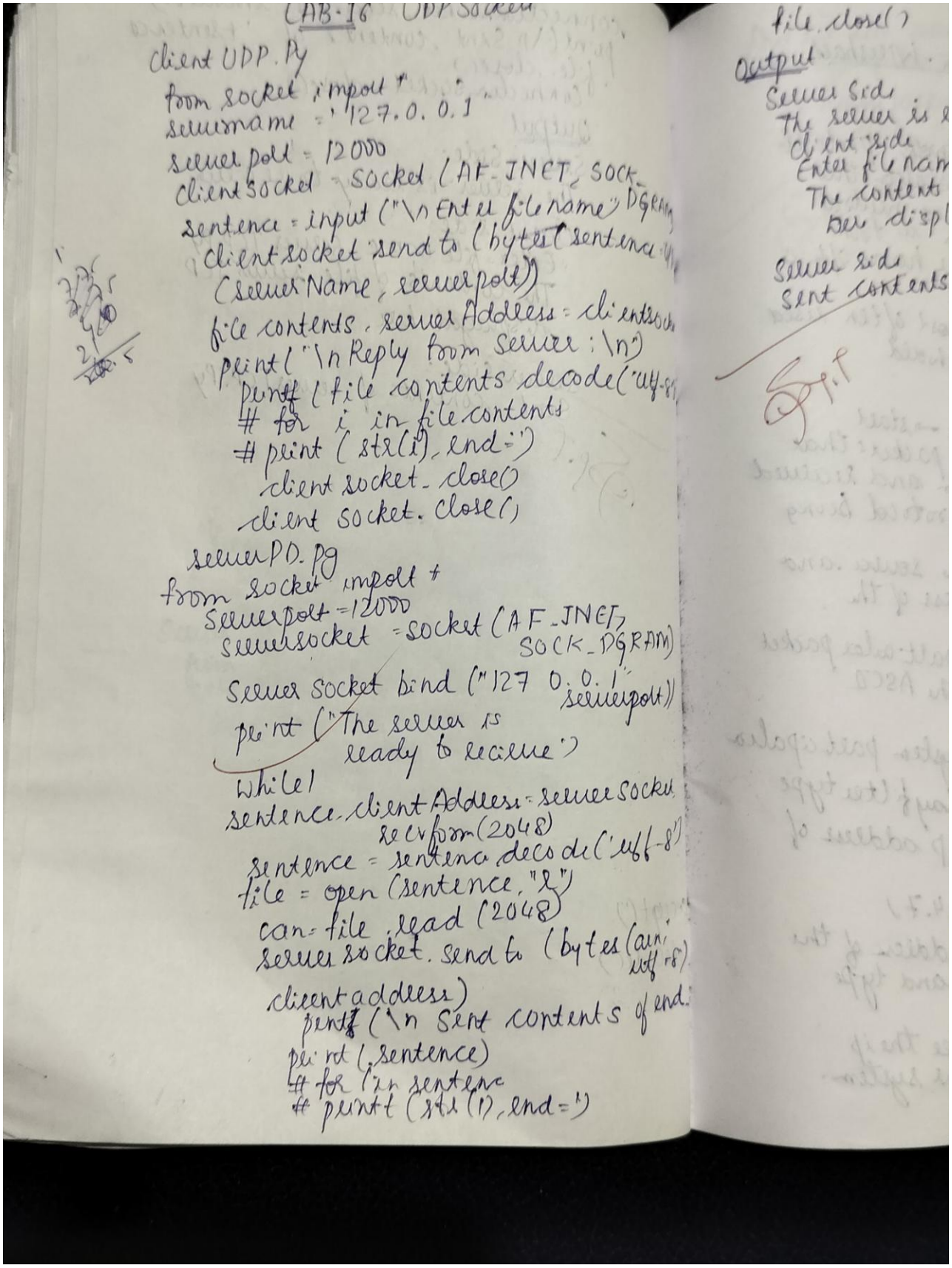
```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 1:
sentence, clientAddress = serverSocket.recvfrom(2048)
sentence = sentence.decode("utf-8")
file=open(sentence,"r")
```

```
con=file.read(2048)
serverSocket.sendto(bytes(con,”utf-8”),clientAddress
) print (“\nSent contents of “, end = “ “)
print (sentence)
# for i in sentence:
# print (str(i), end = “ “)
file.close()
```

OUTPUT:



OBSERVATION:



LAB-16 UDP Socket

client UDP.Py

```
from socket import *
servername = '127.0.0.1'
serverport = 12000
clientSocket = socket(AF_INET, SOCK_
sentence = input("\n Enter file name: DGram
clientsocket.sendto(bytes(sentence
(serverName, serverport))
file contents, serverAddress = clientsock
print("\n Reply from Server: \n")
print(file contents.decode("utf-8
# for i in file contents
# print(str(i), end=')
clientsocket.close()
clientSocket.close()
```

serverUDP.Py

```
from socket import *
serverport = 12000
serversocket = socket(AF_INET,
                      SOCK_DGRAM)
Server Socket.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()

Output

Server side
The server is
Client side
Enter file nam
The contents
Ser displ

Server side
Sent contents

F. INET, SOCK_
er filename") DGRAM)
(bytes(sentence-'ulf
port))
ddress = clientsocket
server : \n")
ts decode('utf-8')
ontents
d=')
se()
e()

(AF_INET,
SOCK_DGRAM)
127 0.0.1", serverport))
s
ceive")
ess = serversocket
2048)
decode('utf-8')
e, "r")
2048)
to (bytes(ans, utf-8')
contents of end=
end=')

file.close()

Output

Server Side
The server is ready to receive
client side
Enter file name : serverUDP.py
The contents of the file serverUDP
are displayed here

Server side
Sent contents of serverUDP.py

WEEK 17

Tool Exploration -Wireshark

OBSERVATION:

LAB-17

Tool Exploration - Wireshark

Wireshark

Wireshark is a network protocol analyser or an application that captures packet from a network communication such as from your computer to your home office or the internet

Wireshark is the most often used packets shiffer in the world

Open Wireshark

And click on capture → start

Now you can see the packets that are sent by the system and received by the system and the protocol being used

And we can see the source and destination address of the packet

If we click on a particular packet now you can see the ASCII code in the bottom

To see only your system participants network in display filter type ip.address == ip address of the pc.

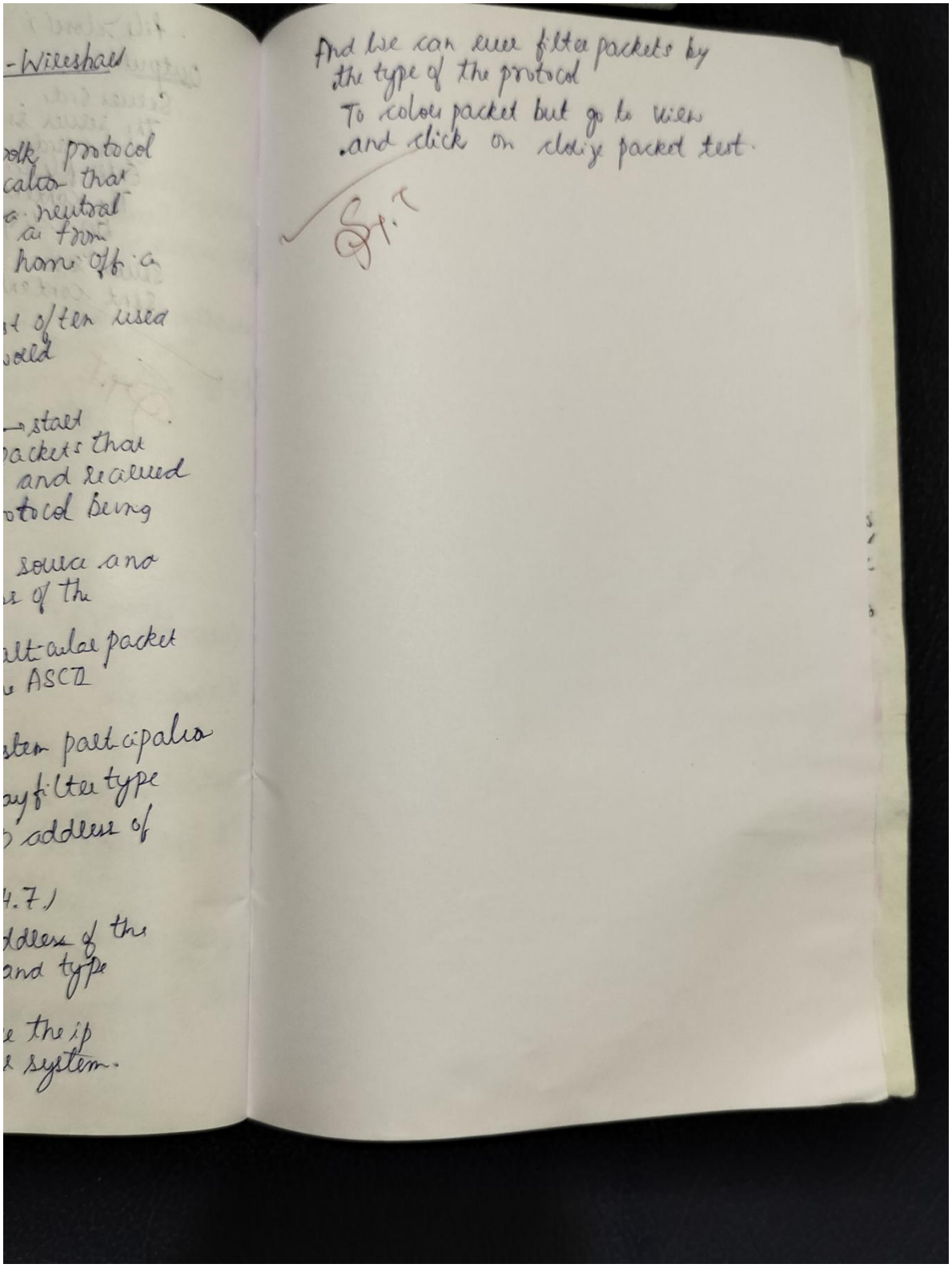
ipaddress - 10.124.7.1

Note: To know ip address of the system open cmp and type ipconfig

Now you can now see the ip address of your system.

And we can see the type of the pr

To color packet and click on



-Wireshark

olk protocol
calco that
a neutral
ai from
home offi a

d often used
ould

stall
ackets that
and received
otocol being

source and
s of the

lt-value packet
ASCII

ster participa
ay filter type
address of

4.7)
ddress of the
and type

e the ip
l system.

And we can even filter packets by
the type of the protocol
To colour packet but go to view
and click on colorize packed text.

Sir