VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT on

COMPUTER NETWORKS LAB

Submitted by

RAHUL RAJ (1BM21CS158)

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 **RAHUL RAJ (1BM21CS158),** 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 satisfies the academic requirements in respect of a **COMPUTER NETWORKS - (22CS4PCCON)** work prescribed for the said degree.

Nandini Vineeth
Assistant Professor
Department of CSE
BMSCE, Bengaluru

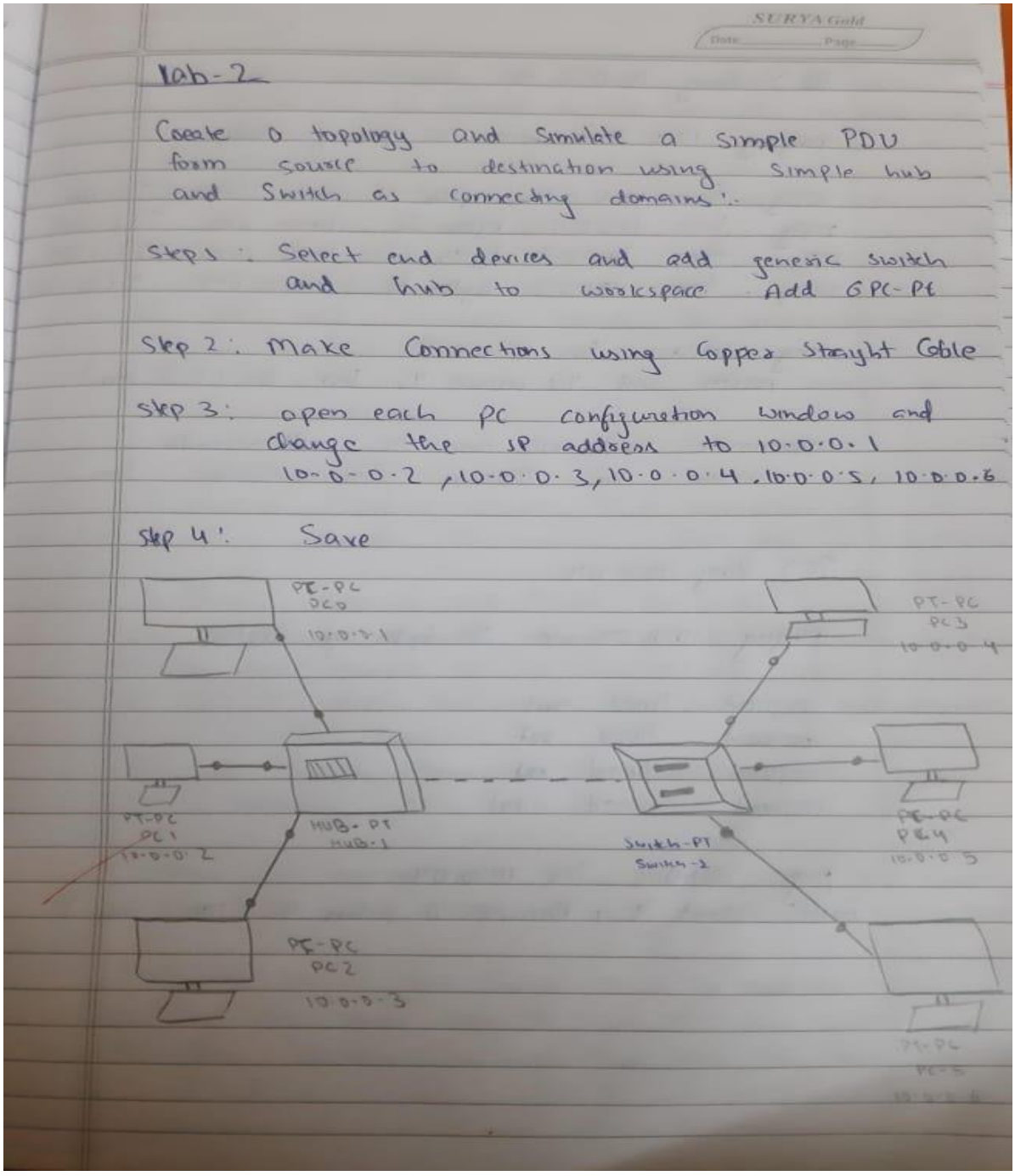
Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

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:



PC> Ping 10.0.0.4

pinging 10.0.0.4 with 32 bytes of data.

reply from 10.0.0.4 : bytes=32 time=12ms TTL=128
reply from 10.0.0.4 : bytes=32 time=6ms TTL=128
reply from 10.0.0.4 : bytes=32 time=6ms TTL=128
reply from 10.0.0.4 : bytes=32 time=6ms TTL=128

ping statistics for 10.0.0.4
packets: sent = 4, received = 4, lost = 0 (0% loss)

approximate round trip time in milli-seconds
Minimum = 6ms, Maximum = 12ms, Average = 7ms

PC) Ping 10.0.04

pinging 10.0.04 with 32 bytes of data:

request timed out.
request timed out.
request timed out.
request timed out.

ping statistics for 10.0.04:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss)

Scenario 1 : Sending btw PC's Connected to Hub
if message is sent from PC1 to PC2

1) Simple PDU is sent from PC1 to Hub.
2) Hub send the copies of the msg to PC0. PC2 and Switch.
3) The PC2 receives the message & sends back Acknowledgement to the hub.
4) The hub further forward the acknowledgement to PC0, PC1 & Switch
5) The PC1 receives it and the transfer is Complete.

Scenario 2: Sending between PC's Connected to Switch if message is sent from PC3 to PC4.

1) Simple PDU is sent from PC3 to Switch
2) The Switch forwards the message to PC5, PC4 and the Hub
3) The Hub forwards it to PC0, PC1 and PC2
4) the PC4 accepts the message and sends an acknowledgement to switch.
5) The Switch forwards the acknowledgement and Similarly PC3 receives it

Scenario 3: Sending btw PC's connected to Switch and Hub if msg is sent from PC0 to PC4

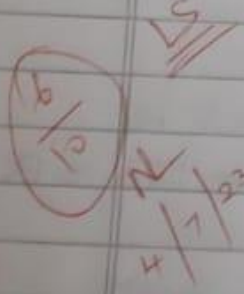
1) Simple PDU is sent from PC0 to ~~switch~~ Hub.
2) The Hub sends copy to PC1, PC2 and switch.
3) The Switch forwards the msg to PC3, PC4, PC5.

4) PC4 receives the msg and sends an acknowledgement to the Switch.

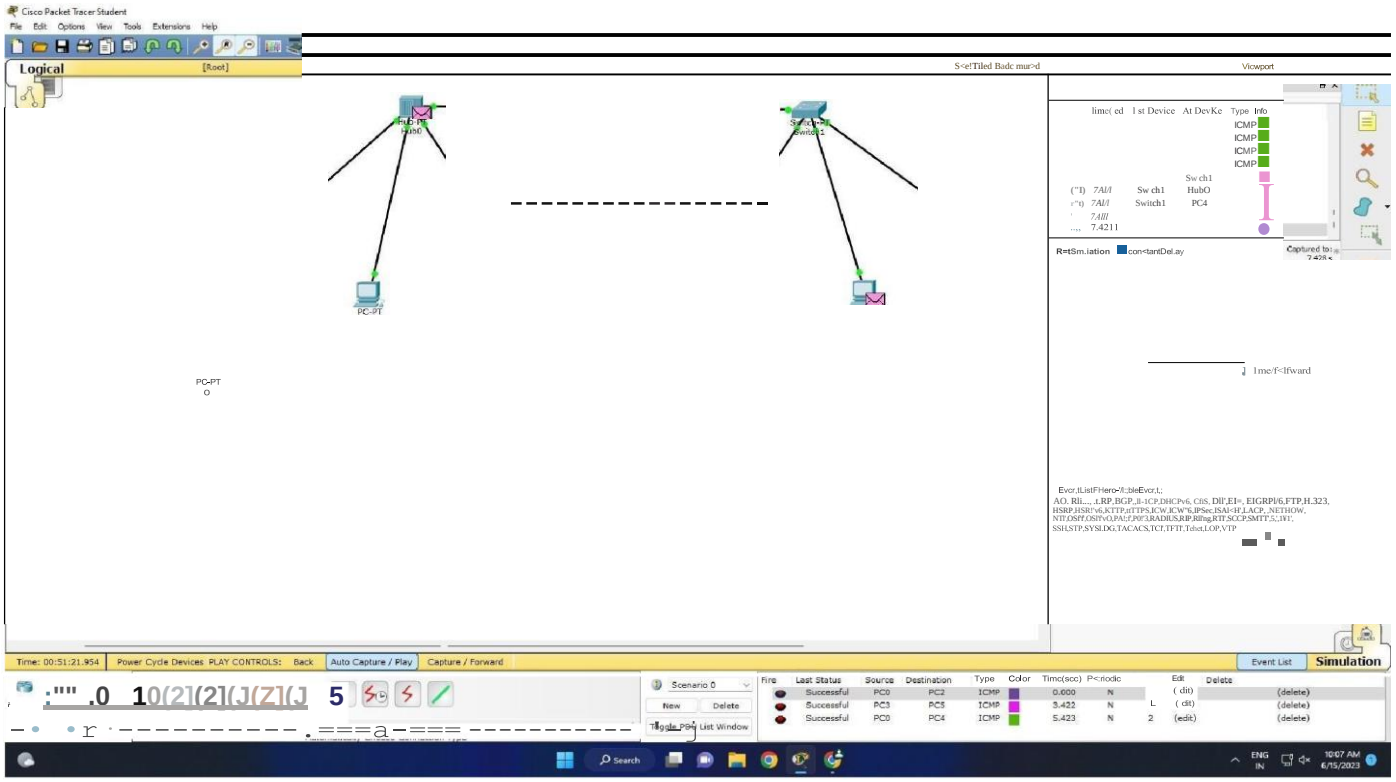
5) The Switch forwards the acknowledgment to the PC3, PC5 and the HUB.

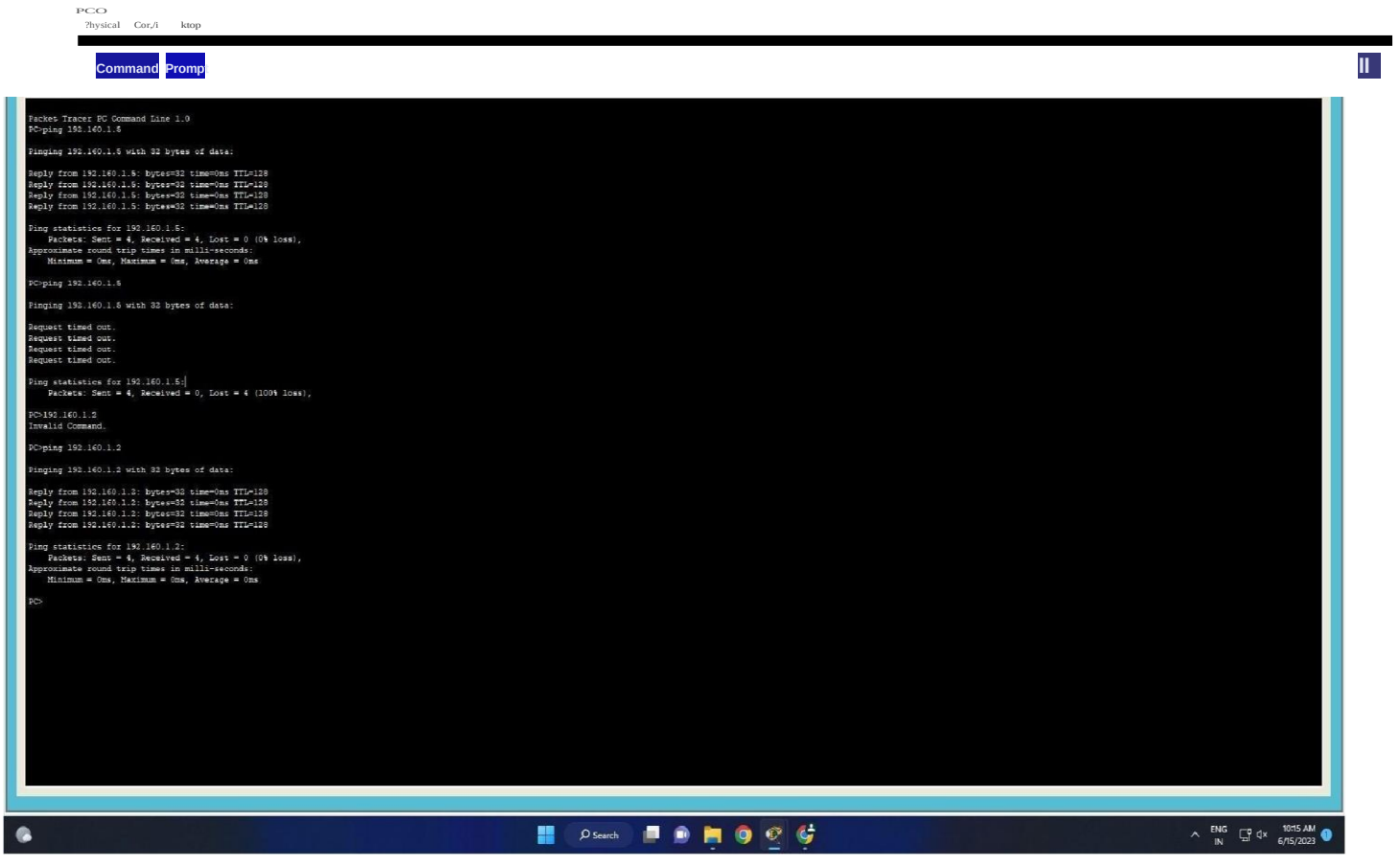
6) Hub sends copy of the acknowledgment to PC0, PC1 and PC4.

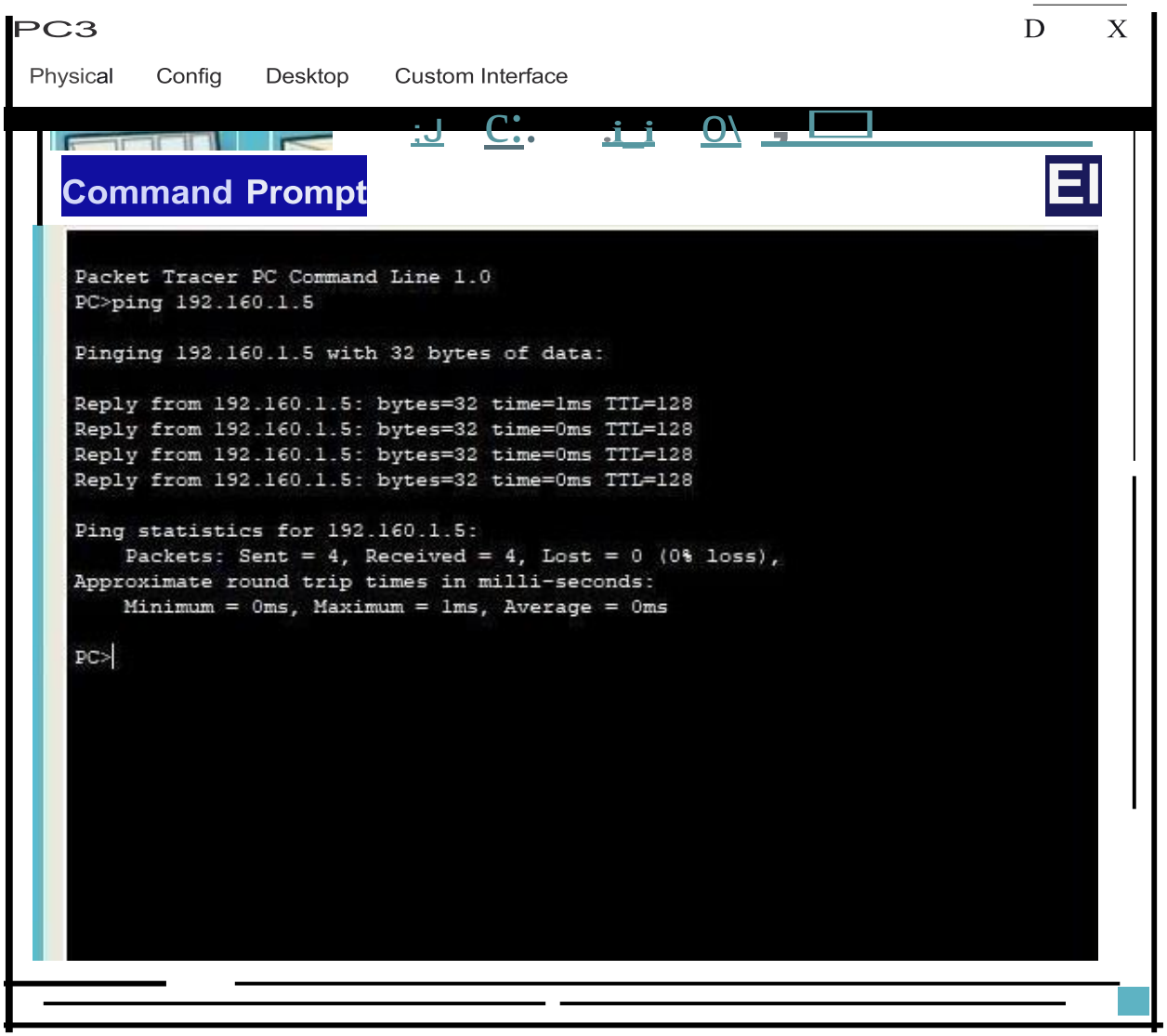
Topology?



OUTPUT:


Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
PC-PT
Time: 00:51:21.954
Power Cycle Devices PLAY CONTROLS: Back
Auto Capture / Play
Capture / Forward
Event List
Simulation
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(sec) Periodic Edit Delete
Successful PC0 PC2 ICMP 0.000 N (dit) (delete)
Successful PC3 PC5 ICMP 5.422 N (dit) (delete)
Successful PC0 PC4 ICMP 5.423 N 2 (edit) (delete)

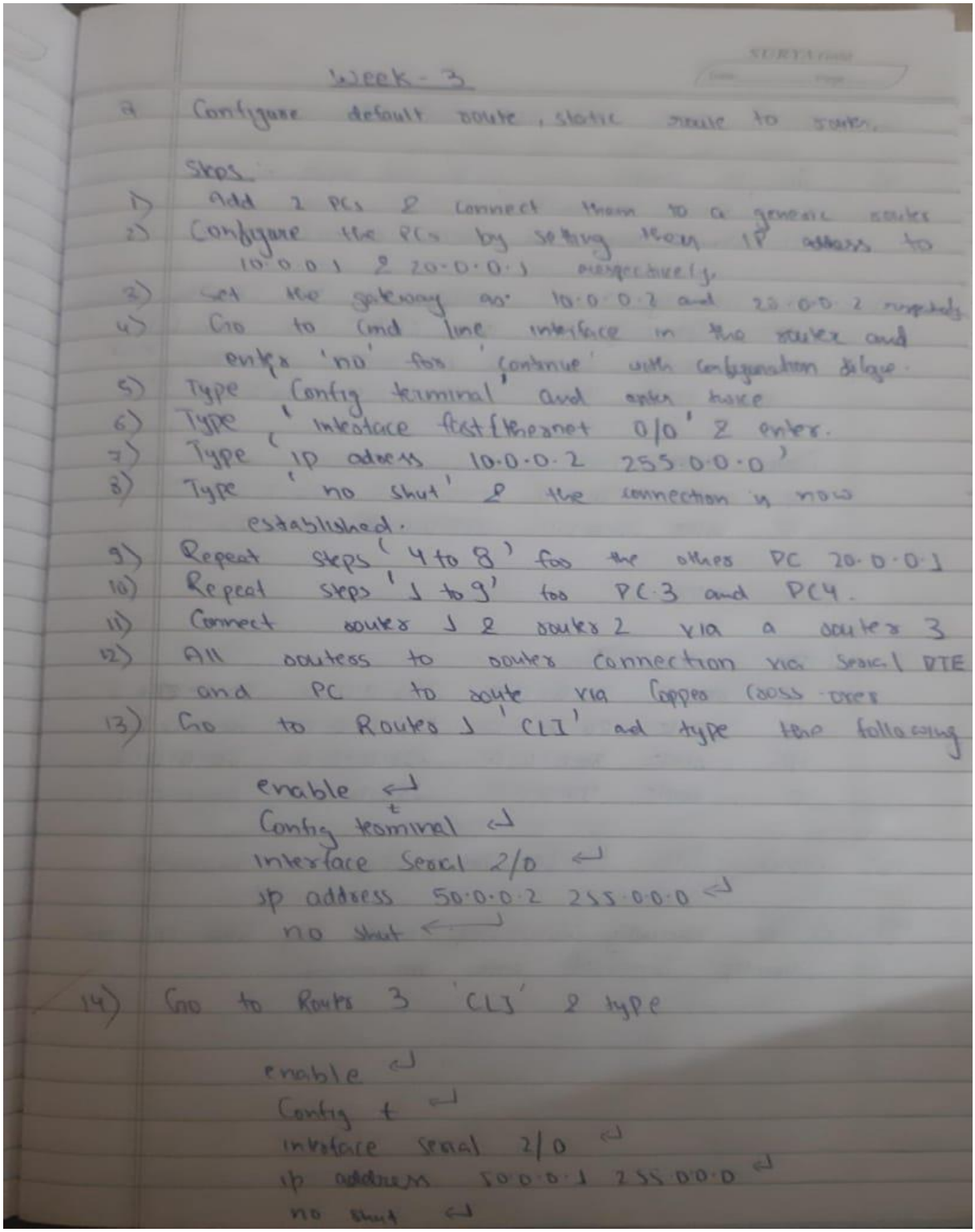

PCO
?hysical Cor,i ktop
Command Promp
Packet Tracer PC Command Line 1.0
PC>ping 192.160.1.5
Pinging 192.160.1.5 with 32 bytes of data:
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
Reply from 192.160.1.5: bytes=32 time=0ms TTL=128
Reply from 192.160.1.5: 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 192.160.1.6
Pinging 192.160.1.6 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 192.160.1.6:
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>


PC3
D X
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=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:



Week - 3

9 Configure default route, static route to router.

Steps

1) Add 2 PCs & Connect them to a generic router
2) Configure the PCs by setting their IP address to 10.0.0.1 & 20.0.0.1 respectively.
3) Set the gateway as 10.0.0.2 and 20.0.0.2 respectively
4) Go to Cmd line interface in the router and enter 'no' for 'continue' with configuration dialog.
5) Type 'Config terminal' and enter twice
6) Type 'interface fastEthernet 0/0' & enter.
7) Type 'ip address 10.0.0.2 255.0.0.0'
8) Type 'no shut' & the connection is now established.
9) Repeat steps '4 to 8' for the other PC 20.0.0.1
10) Repeat steps '1 to 9' too PC3 and PC4.
11) Connect router 1 & router 2 via a router 3
12) All routers to router connection via Serial DTE and PC to route via Copper Cross over
13) Go to Router 1 'CLI' and type the following

enable ↵
Config terminal ↵
interface Serial 2/0 ↵
ip address 50.0.0.2 255.0.0.0 ↵
no shut ↵

14) Go to Router 3 'CLI' & type

enable ↵
Config t ↵
interface serial 2/0 ↵
ip address 50.0.0.1 255.0.0.0 ↵
no shut ↵

15) Repeat steps (13) and (14) for Router 2 to 3.

16) Go to Router 1 'CLI' and type 'show ip route'
It shows only the direct connections.

17) We statically connect routers to the PCs by typing the following in the 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 routes 3):

~~(ip route 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

18) Now data transfer btw PCs is successful.

19) Before statically connecting routes ping btw PCs NOT directly connected was unsuccessful.

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

Reply from 20.0.0.2 : Destination host unreachable

ping statistics for 30.0.0.1

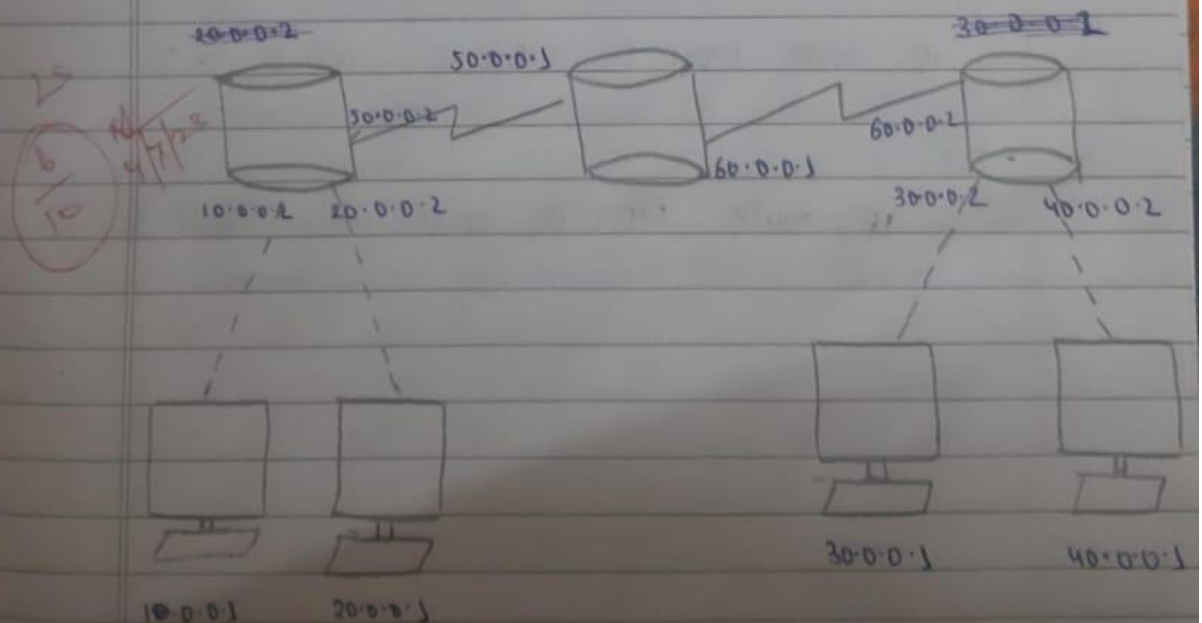
packets : Sent = 4 , Received = 0 , lost = 4 (100% lost)

After statically defining the route :

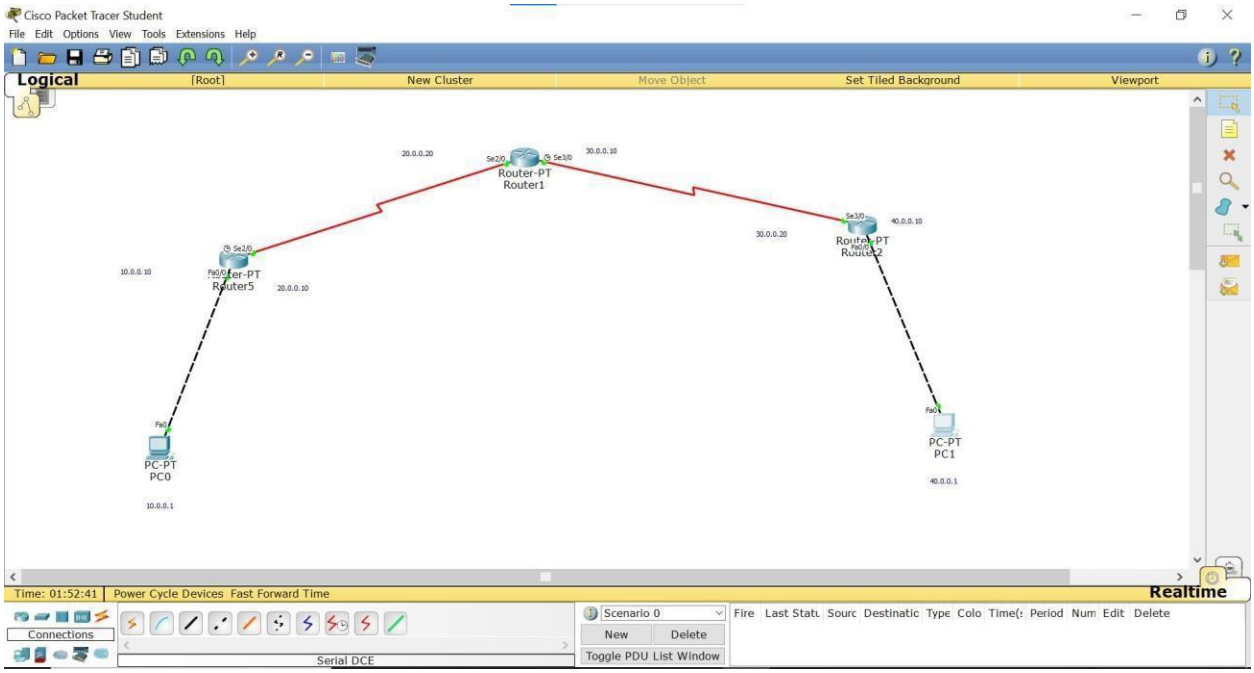
PC > ping 40.0.0.1

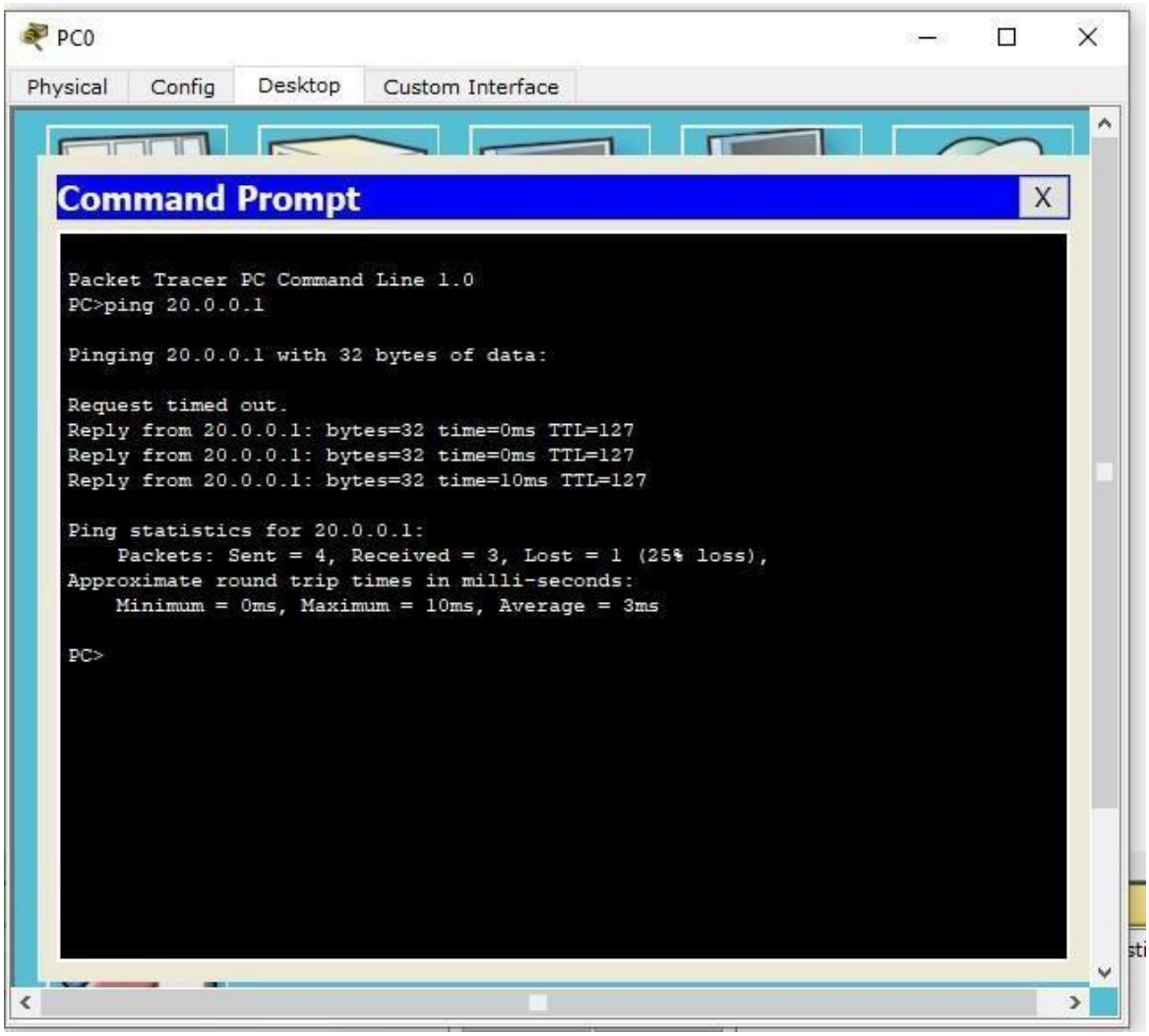
pinging 40.0.0.1 with 32 bytes of data :

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

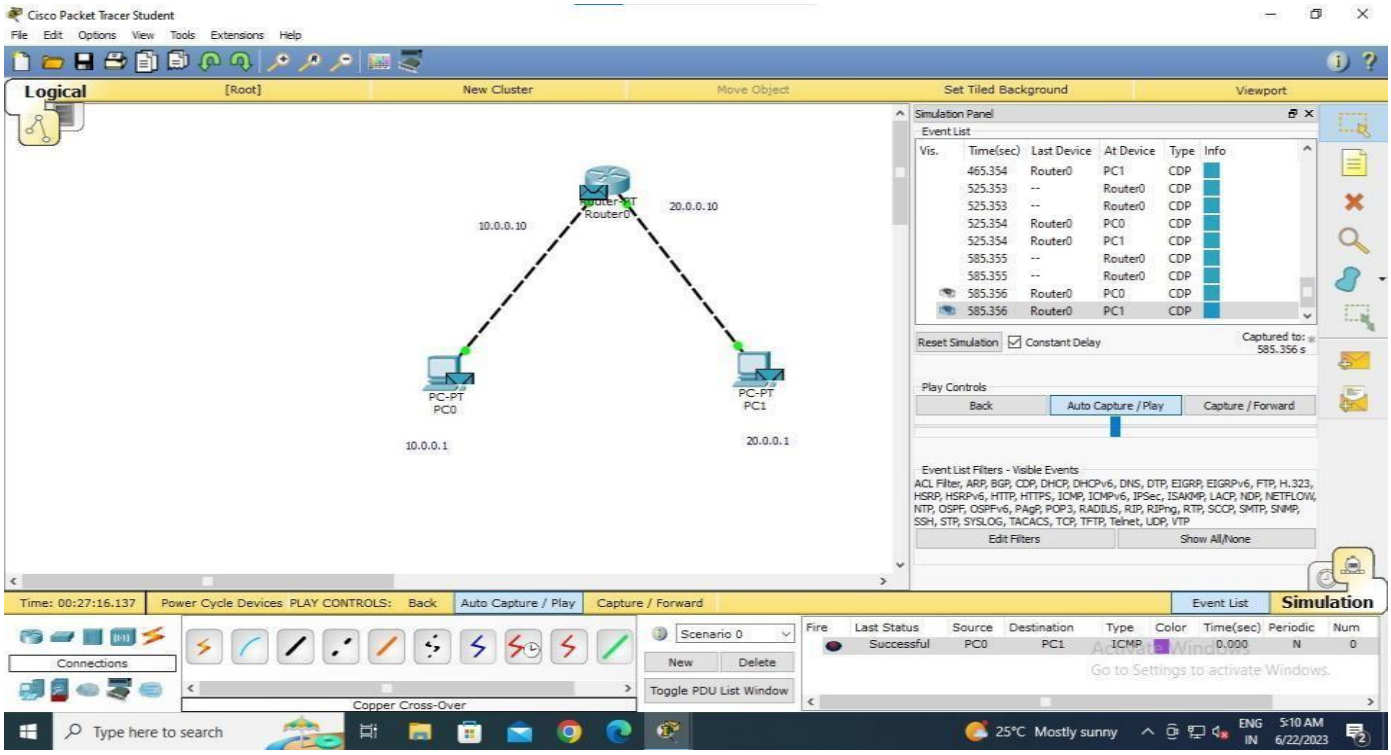


TOPOLOGY:


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
Time: 01:52:41
Power Cycle Devices Fast Forward Time
Realtime
Connections
Serial DCE
Scenario 0
New
Delete
Toggle PDU List Window


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
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>

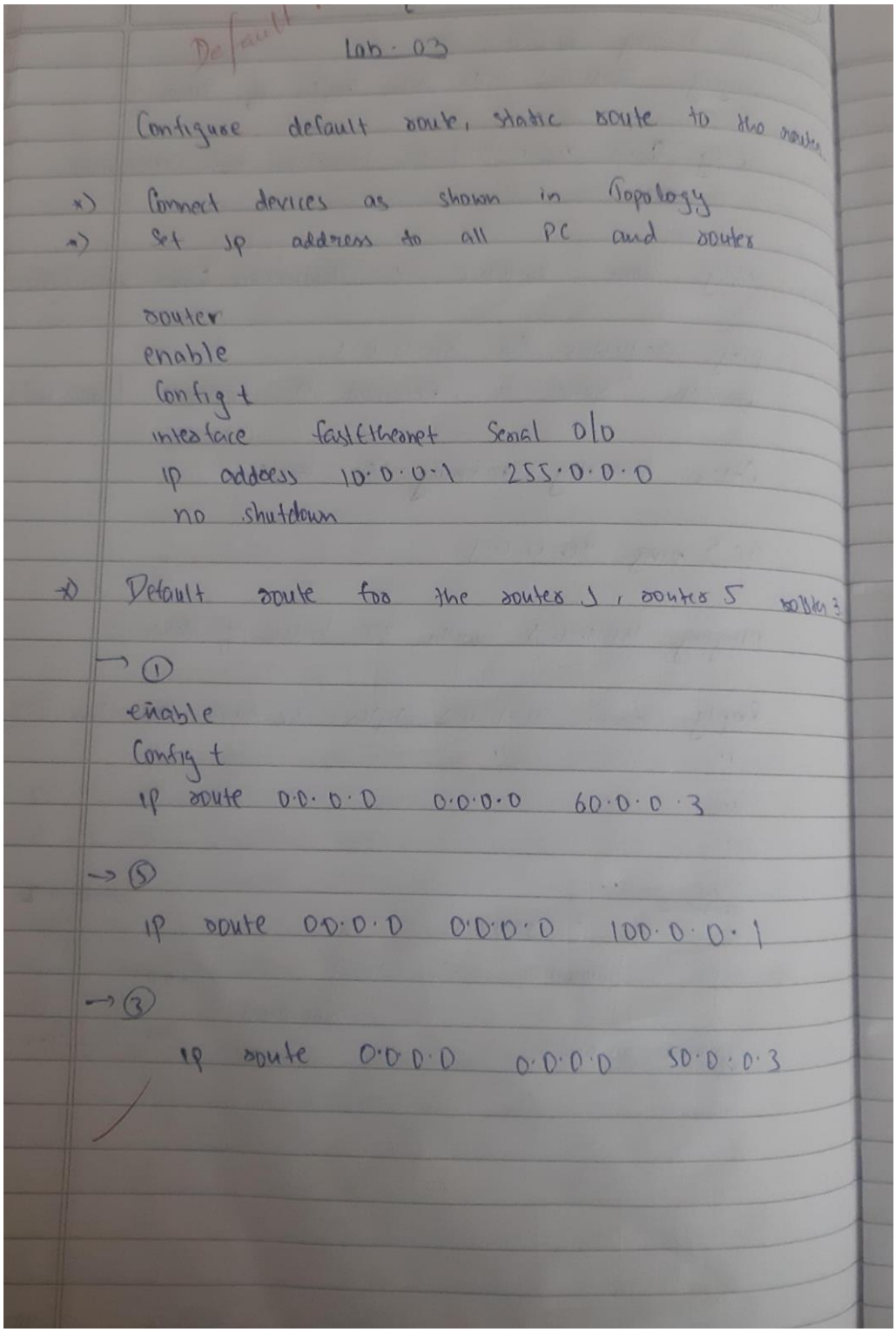
OUTPUT:


Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT
Router0
10.0.0.10
20.0.0.10
PC-PT
PC0
PC-PT
PC1
10.0.0.1
20.0.0.1
Simulation Panel
Event List
Vis. Time(sec) Last Device At Device Type Info
465.354 Router0 PC1 CDP
525.353 -- Router0 CDP
525.353 -- Router0 CDP
525.354 Router0 PC0 CDP
525.354 Router0 PC1 CDP
585.355 -- Router0 CDP
585.355 -- Router0 CDP
585.356 Router0 PC0 CDP
585.356 Router0 PC1 CDP
Reset Simulation
Constant Delay
Captured to: 585.356 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: 00:27:16.137
Power Cycle Devices PLAY CONTROLS: Back
Auto Capture / Play
Capture / Forward
Event List
Simulation
Connections
Copper Cross-Over
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(sec) Periodic Num
Successful PC0 PC1 ICMP 0.000 N 0
Activate Windows
Go to Settings to activate Windows.
Type here to search
25°C Mostly sunny
ENG IN
5:10 AM 6/22/2023

LAB 3

Configure default route, static route to the Router.

OBSERVATION:



→ 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 routers.

output :-

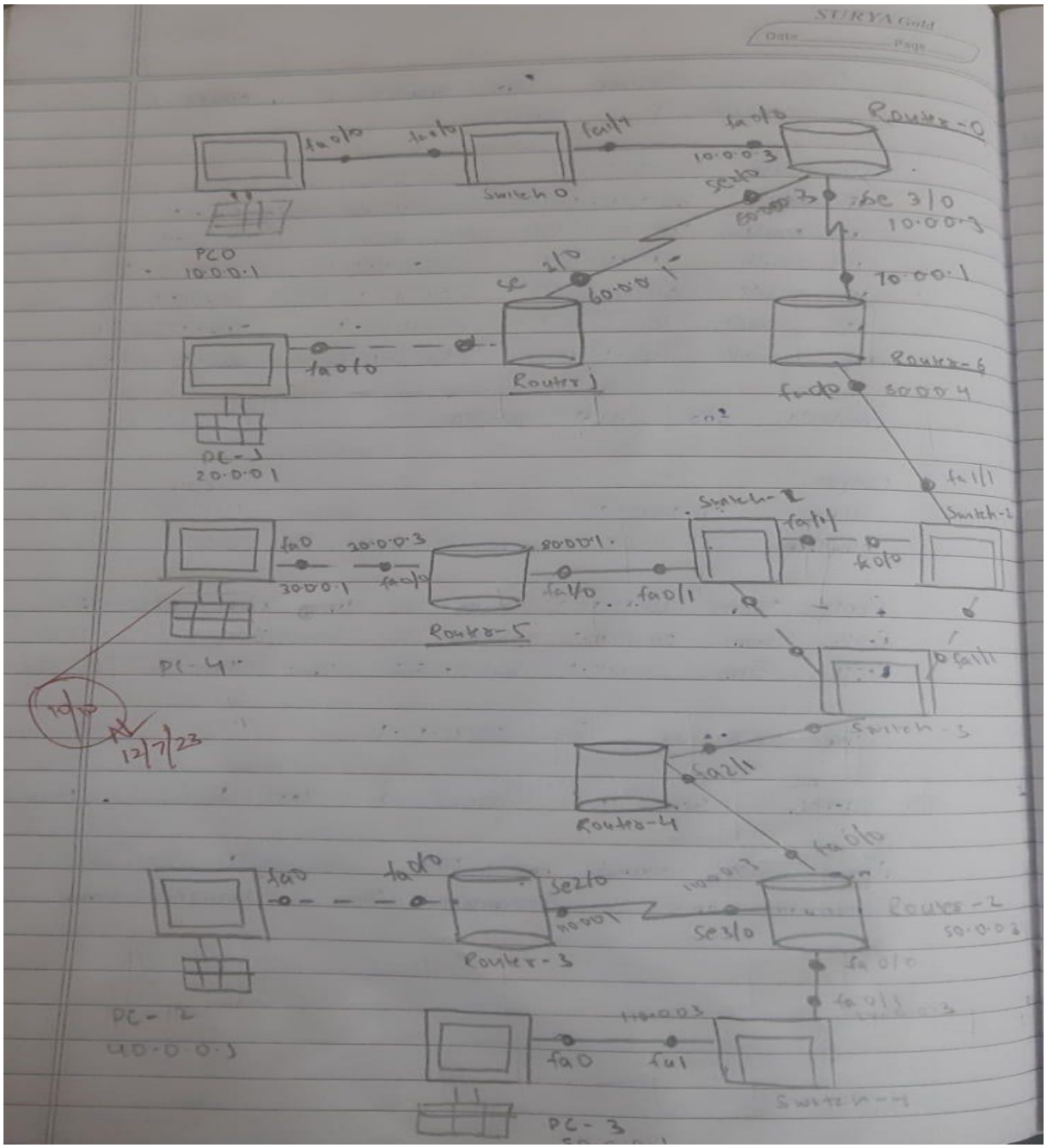
ping 20.0.0.1
pinging 20.0.0.1 with 32 bytes of data

Reply from 20.0.0.1 : bytes: 32 time: 7ms TTL: 126
Reply from 20.0.0.1 : bytes: 32 time: 6ms TTL: 126
Reply from 20.0.0.1 bytes = 32 time: 7ms TTL: 126

ping statistics for 20.0.0.1 :

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

Approximate round trip time in milli seconds:
minimum = 6ms , Maximum = 7ms , Average = 6ms

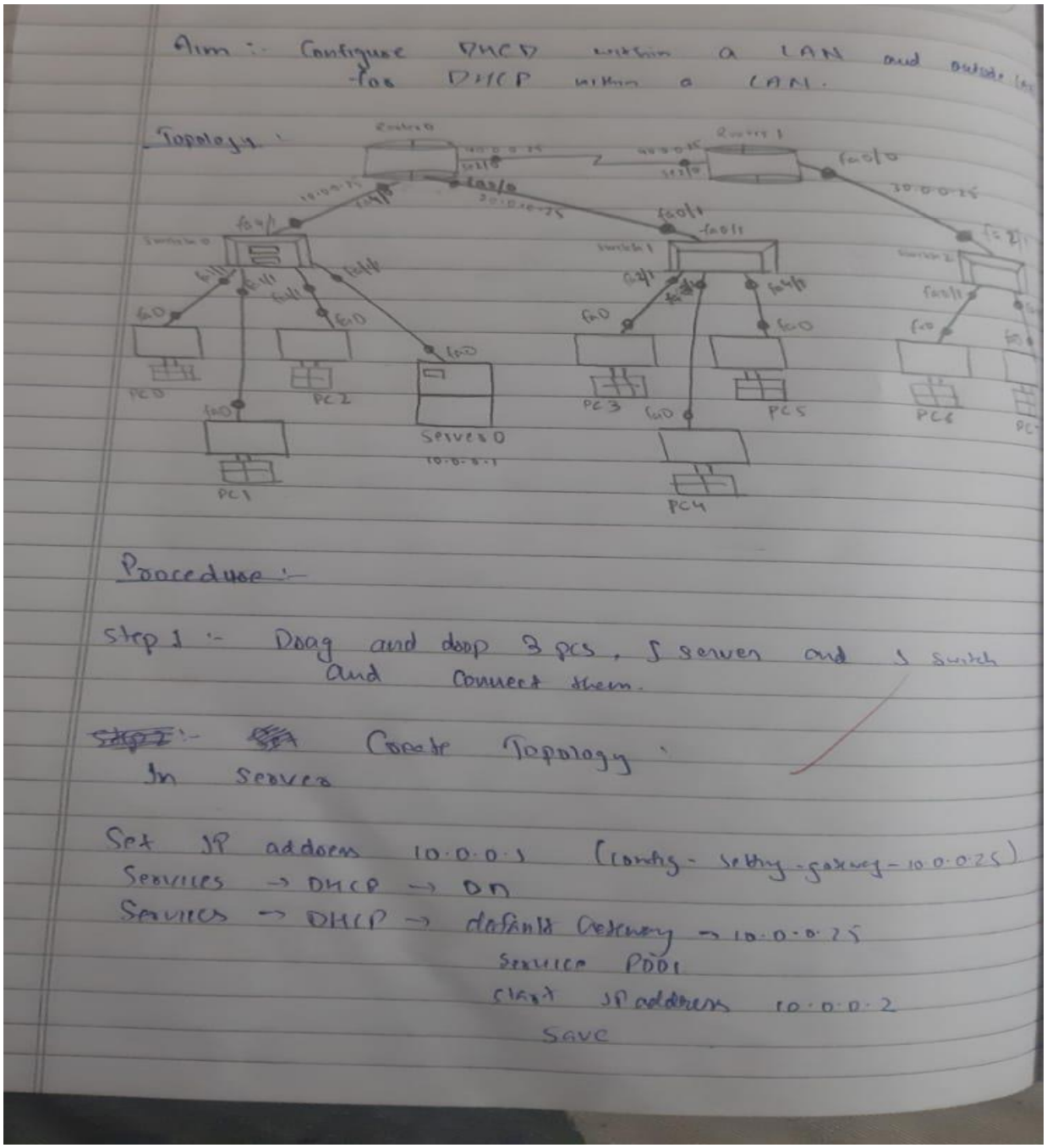

SURYA Gold
Date
Page
fa0/0
fa0/0
fa1/1
Switch 0
PC0
10.0.0.1
fa0/0
10.0.0.3
se2/0
Router-0
se 3/0
10.00.3
10.00.1
se 2/0
60.0.0.1
fa0/0
Router 1
PC-1
20.0.0.1
Router-6
50.0.0.4
fa1/1
Switch-2
Switch-1
fa0/1
fa0
30.0.0.3
30.0.0.1
fa0/0
80.0.0.1
fa1/0
fa0/1
Router-5
PC-4
fa0/0
12/7/23
Switch-3
fa2/1
Router-4
fa0/0
fa0
fa0/0
se2/0
Router-2
50.0.0.2
Router-3
se3/0
PC-2
40.0.0.5
fa0
fa1
PC-3
Switch-4

LAB 4

Configure DHCP within a LAN and outside LAN.

OBSERVATION:

TOPOLOGY:



Source POOL 1

gateway → 10.0.0.25

start ipaddress 20.0.0.2

add

Source POOL 2

gateway → 10.0.0.25

start ipaddress 40.0.0.2

add.

Gw routers 0 :-

Set two Network ip address

- Config t
- interface fastEthernet 0/0
- ip address 10.0.0.75 255.0.0.0
- no shut.

(Similarly 20.0.0.75)

- Config t
- interface fastEthernet 0/0
- ip address 10.0.0.1
- no shut.

→ static route (for 40 ip)

- Config t

→ ip route 40.0.0.0 255.0.0.0 30.0.0.26

On routers:

Set ip address
- Config t
- interface fa 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

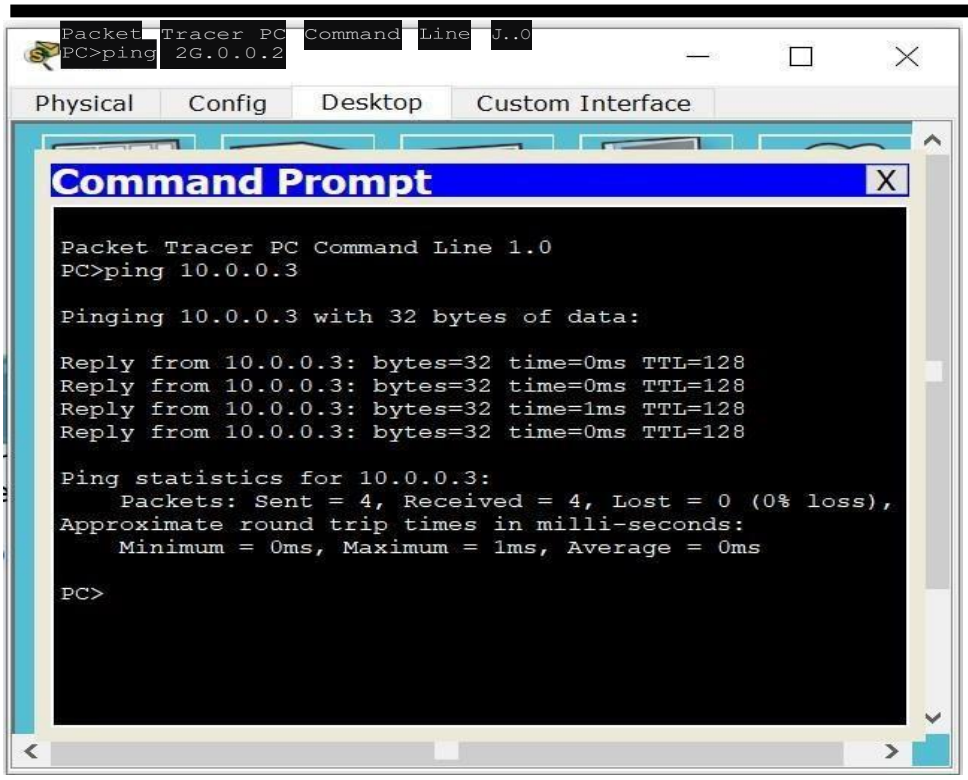
Static routing for 10 and 20 network

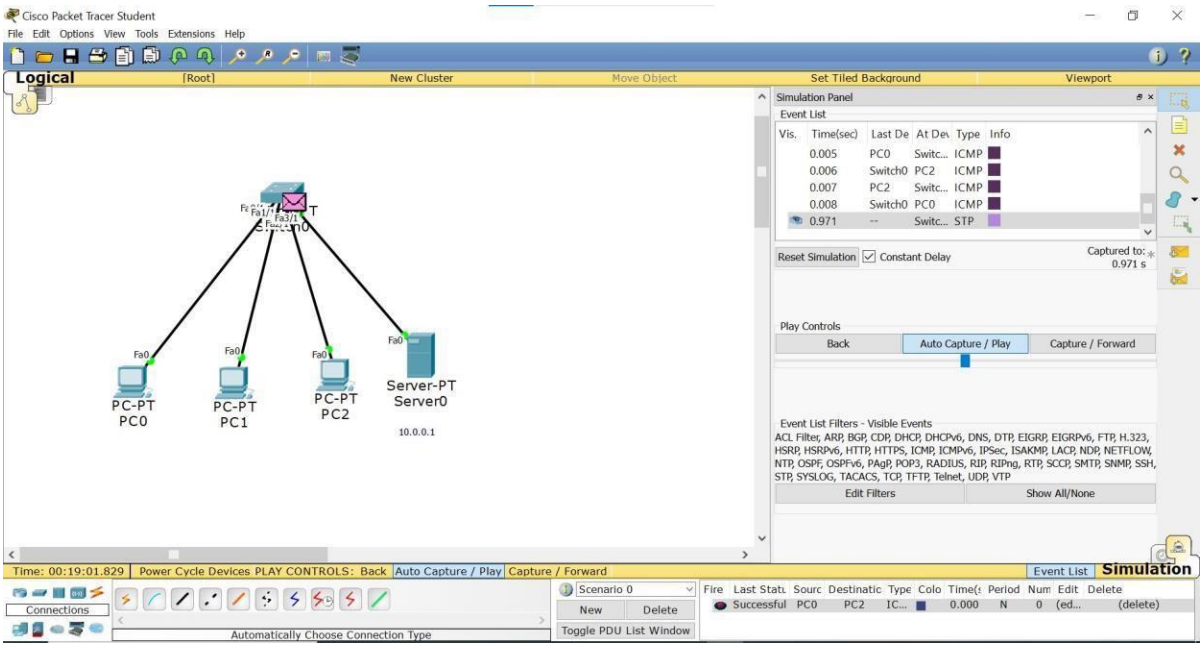
- Config t
- ip route 10.0.0.0 255.0.0.0 30.0.0.25
- ip route 20.0.0.0 255.0.0.0 30.0.0.25
- no shut.

→ Setting helper address
- Config t
- interface fastEthernet 0/0
- ip helper-address 10.0.0.1
- no shut.

Output :-
On any PC
Desktop → IP configuration → DHCP
(Dynamic IP address is assigned to all PC by

T:


Packet Tracer PC Command Line J..0
PC>ping 2G.0.0.2
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data:
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=1ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Ping statistics for 10.0.0.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
PC>

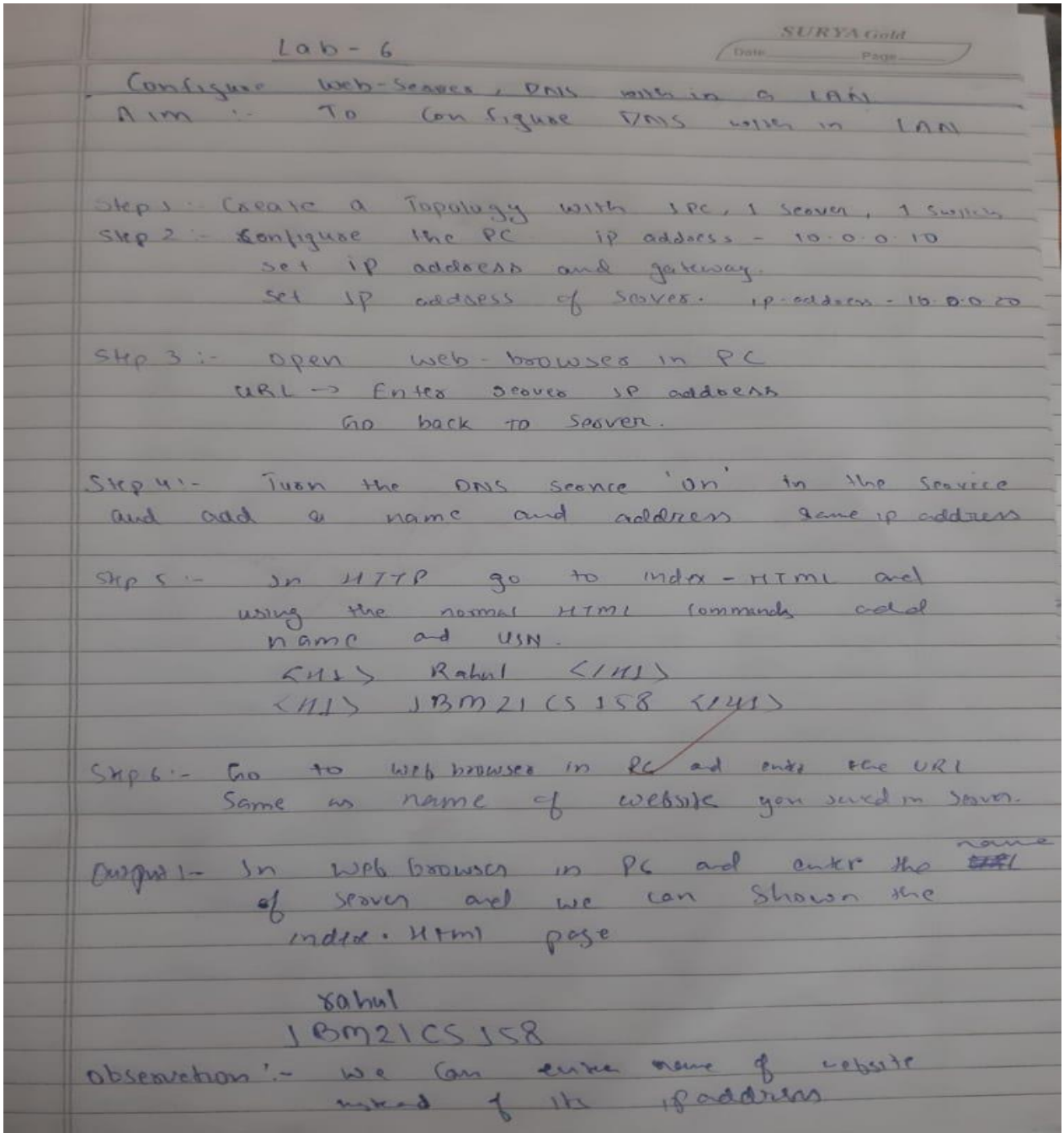

Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
PC-PT PC0
PC-PT PC1
PC-PT PC2
Server-PT Server0
10.0.0.1
Simulation Panel
Event List
Reset Simulation
Constant Delay
Play Controls
Back
Auto Capture / Play
Capture / Forward
Edit Filters
Show All/None
Time: 00:19:01.829
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

X

WEEK5

Configure Web Server, DNS within a LAN.

OBSERVATION:



Lab - 6

Configure Web-Server, DNS within a LAN

Aim :- To Configure DNS within LAN

Step 1 :- Create a Topology with 1 PC, 1 Server, 1 Switch

Step 2 :- Configure the PC ip address - 10.0.0.10
set ip address and gateway.
set ip address of server. ip address - 10.0.0.20

Step 3 :- Open web-browser in PC
URL → Enter Server ip address
Go back to Server.

Step 4 :- Turn the DNS service 'on' in the service and add a name and address same ip address

Step 5 :- In HTTP go to index.HTML and using the normal HTML commands add name and USN.
<H1> Rahul </H1>
<H1> 1BM21CS158 </H1>

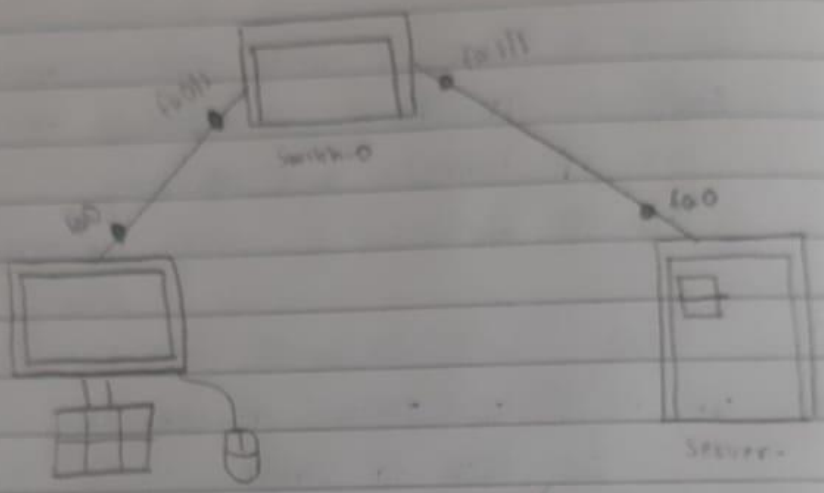
Step 6 :- Go to web browser in PC and enter the URL same as name of website you saved in Server.

Output :- In web browser in PC and enter the name of server and we can shown the index.Html page

Rahul
1BM21CS158

Observation :- We can enter name of website instead of its ipaddress

Topology for DNS server:-



PC-0

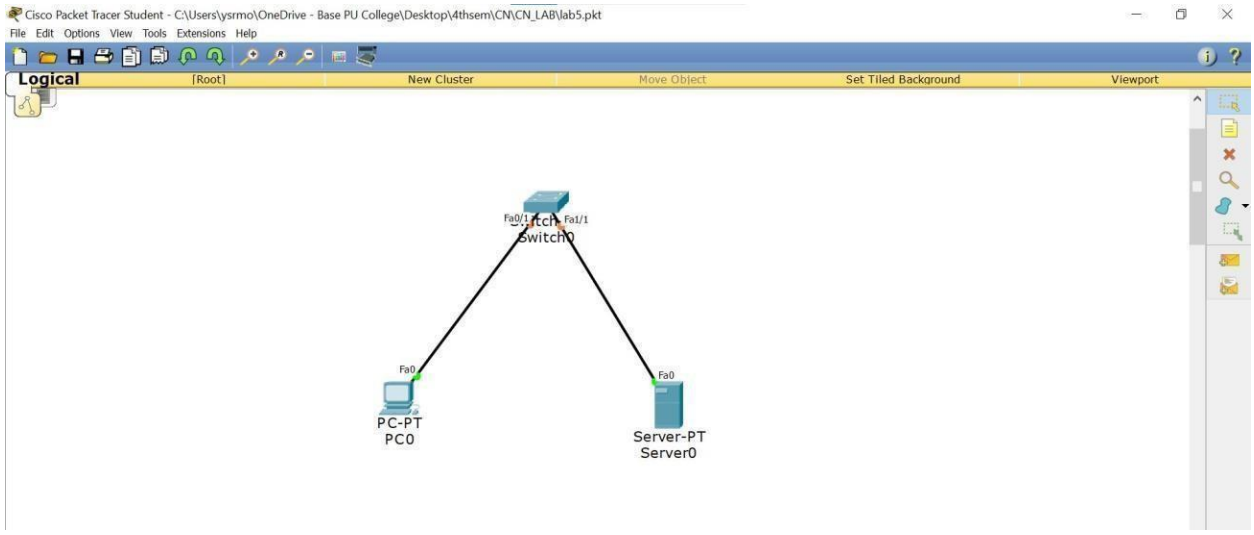
10.0.0.10

10.0.0.20

10/10

25/7/23

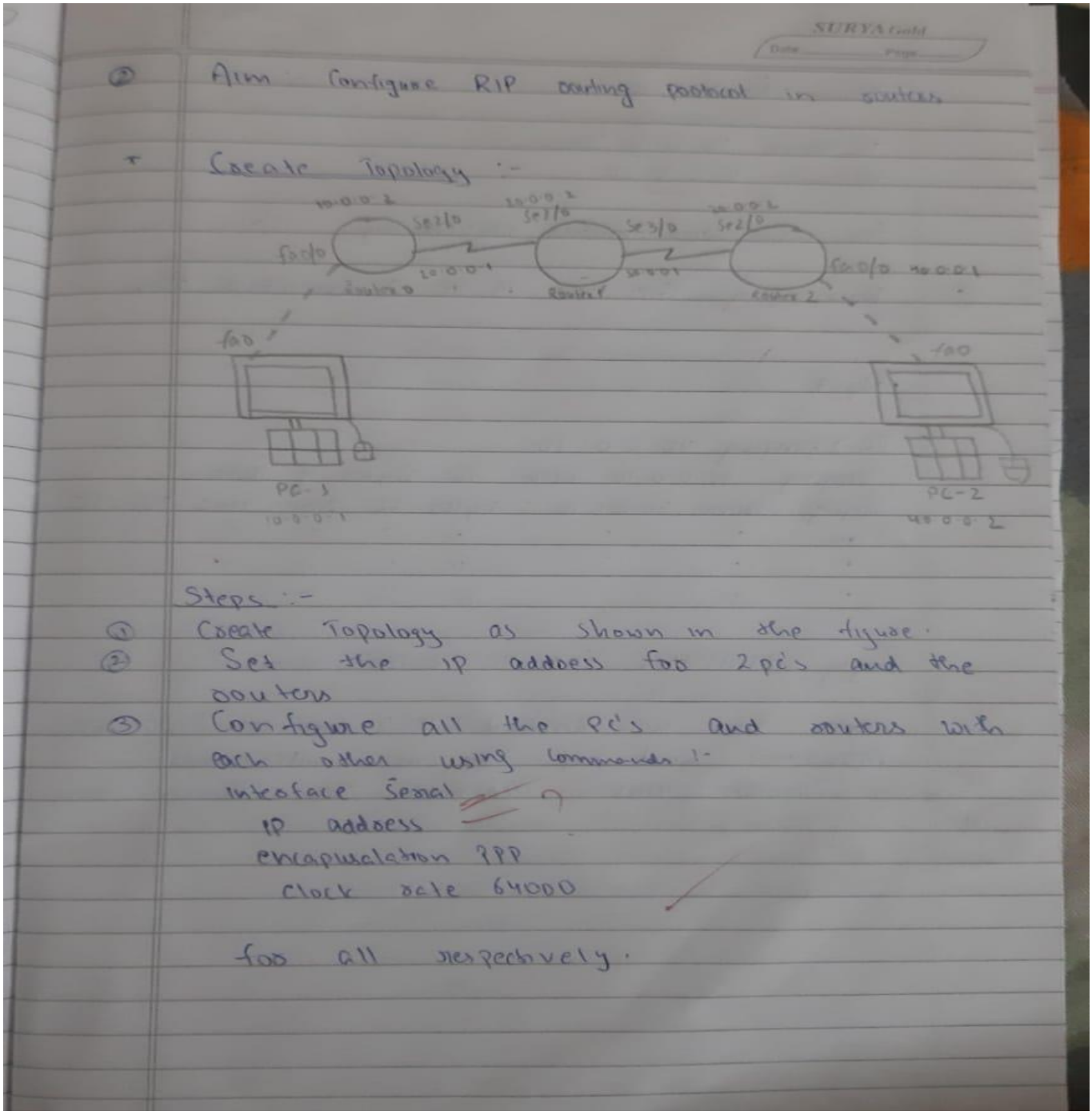
TOPOLOGY:


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:



Now Router ip

network 10.0.0.10
network 10.0.0.20

for all three routers respectively

(5) Now ping the PC's it is all connected to:

output :-

PC> pinging 40.0.0.10
pinging 40.0.0.10 with 32 bytes of data.
Reply from 40.0.0.10 : bytes : 32 time : 4ms TTL : 10
" " " " : " " "
" " " " : " " "
" " " " : " " "

Pinging statistics for 40.0.0.10 :

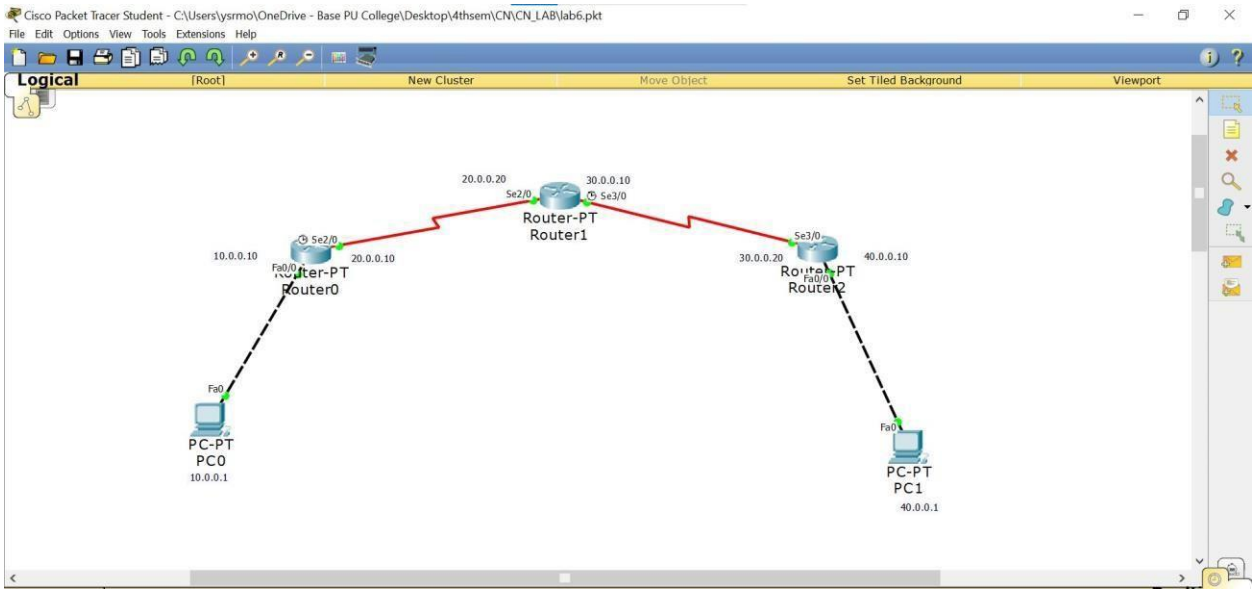
packets : sent : 4 , Received : 4 , lost : 0 (0% loss)
Approximate round trip time in millisecords :
Minimum : 4ms , Maximum : 2ms , Average : 4ms

Observation :

8/10

25/7/23

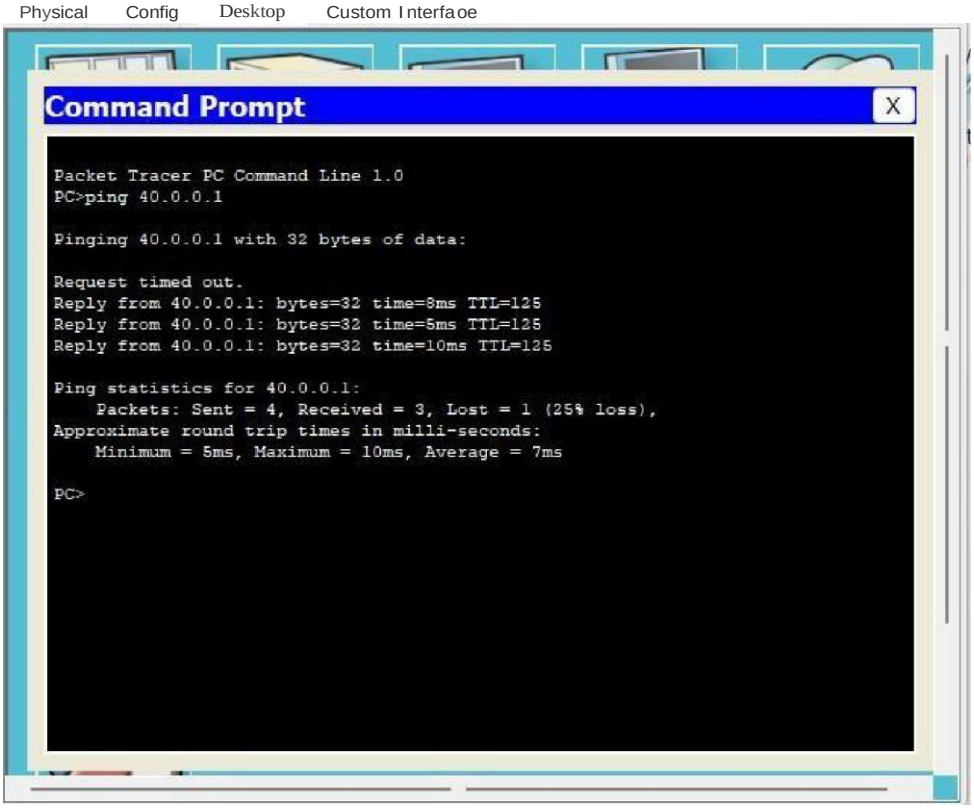
TOPOLOGY:



OUTPUT:

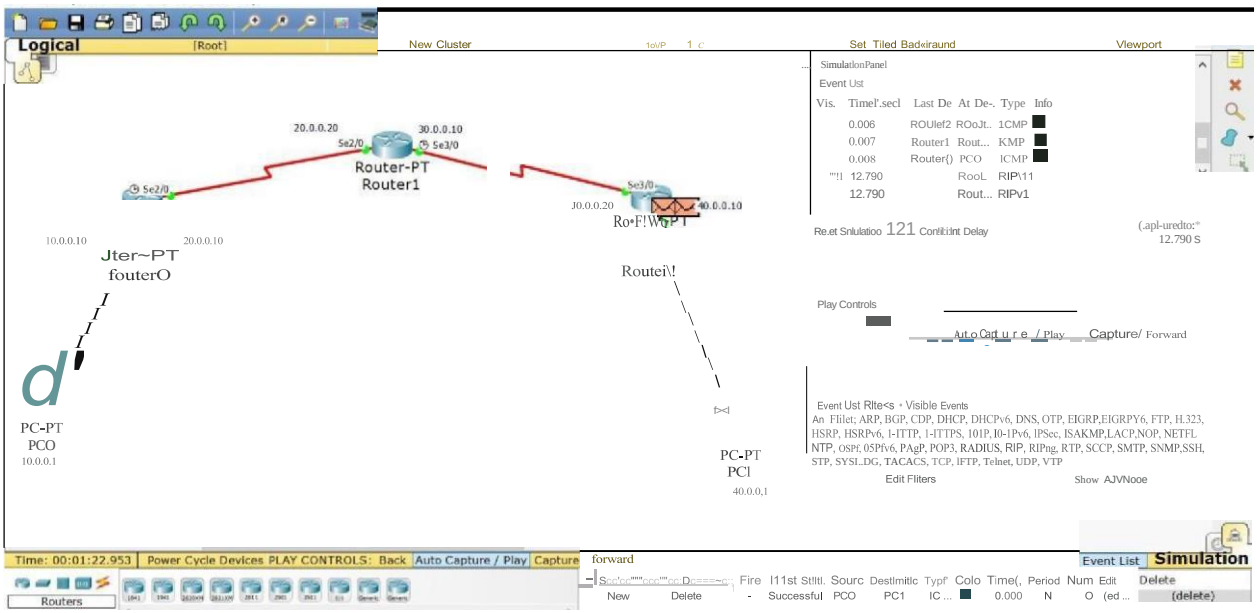
PCO ☐ X

Physical Config Desktop Custom Interfaoe



--t'Ci\iCOPad:et Tracer Studenl- C:\Ui;ers\y!iinno\OneDrive - Base PU College\ Idop\4thsem\CN\CN_LAB\l.ib6.pld 01 X

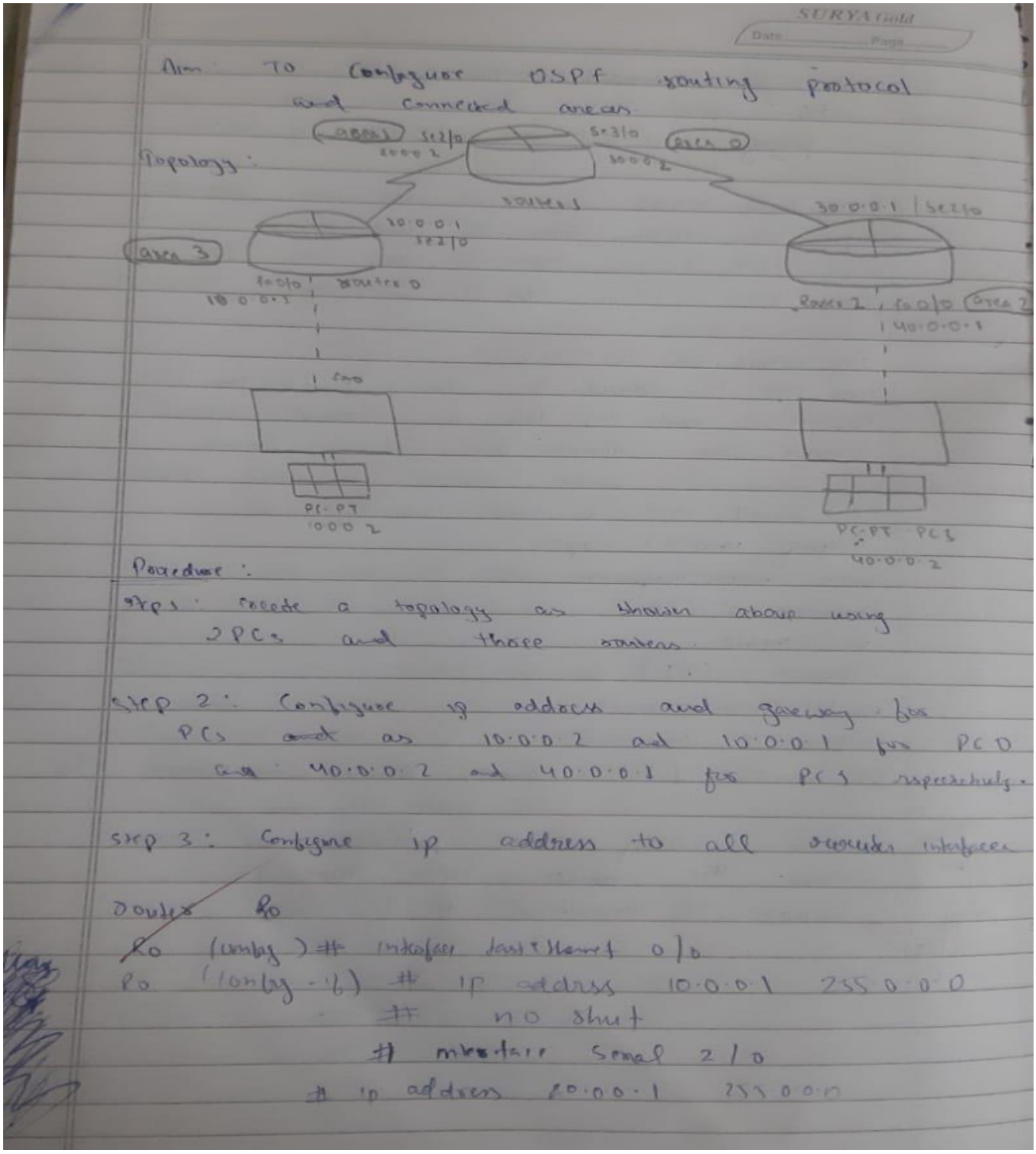
Ale Ecit Options Tools Exte11slons Help



LAB 7

Configure OSPF routing protocol.

OBSERVATION:



encapsulation PPP
clock rate 64000
No shut
exit

Similarly configure for R1 and R2.

Step 4: Now, enable ip routing by configuring OSPF routing Protocol in all routers.

Router R0:

(config)# router ospf 1
router-id 1.1.1.1
network 10.0.0.0 255.255.255.0 area 3
network 20.0.0.0 255.255.255.0 area 1
exit

Similarly, Configure for R1 and R2.

Step 5: Now, check routing table of R0

Router# show ip route

C - Connected
O - OSPF
C - 10.0.0.0/8 is directly connected, fa 0/0
C - 20.0.0.0/8 is directly connected, Serial 2/0
O IA 40.0.0.0/8 via 20.0.0.2, 00:04:23, Serial 2/0
O IA 30.0.0.0/8 via 20.0.0.2, 00:07:29, Serial 2/0

Here R1 knows area 0 Network 20.0.0.0 connected to R1 from R0. So R0 knows networks through this network.

Router (config) # router ospf 1, 1 → process id (1 - 65535)
There must be one interface up to keep ospf process up
So, its better to configure loopback address to router,
It is a virtual interface never goes down once we configure

R0 (config - if) # interface loopback 0
ip add 172.16.1.252 255.255.0.0
no shut

Similarly, configure for R1 and R2.

Step 6: Now check routing table for R3 ~~and R2~~
R3 # show ip route
Codes: O - Ospf C - Connected

O IA 20.0.0.0/8 via 30.0.0.2, 00:18:38, Serial 2/0
C 40.0.0.0/8 is directly connected, FastEthernet 0/0
C 30.0.0.0/8 is directly connected, Serial 2/0

Here, R3 doesnot know about the area 3 so we have to create Virtual link between R0 and R1.

Step 7: Create a virtual link between R0, R1, and by this we create a virtual link to connect area 3 to area 0.

SURYA Gold
Date Page

In R0.

R0 (config) # router ospf 1

area 1 virtual-link 2.2.2.2

In R1.

R1 (config) # router ospf 1.

area 1 virtual link 1.1.1.1

Step 8: R1 and R2 get updates about Area 1
Now, check routing table R2.

R2 # show ip route

Codes: O - OSPF C - Connected

O IA 20.0.0.0/8 via 30.0.0.2, 00:01:56, Serial 2/0
C 40.0.0.0/8 is directly connected, FastEthernet 0/0
O IA 10.0.0.0/8 via 30.0.0.2, 00:01:56, Serial 2/0
C 30.0.0.0/8 directly Connected, Serial 2/0

Step 9: Ping PCs from PC0.

In PC0,

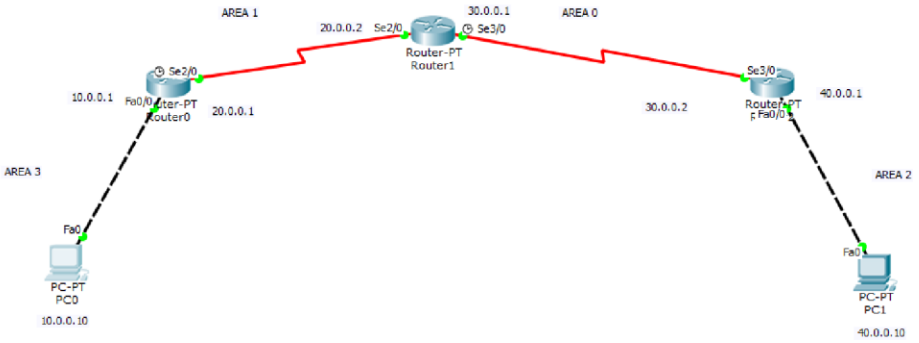
PC> Ping 40.0.0.2

8/10

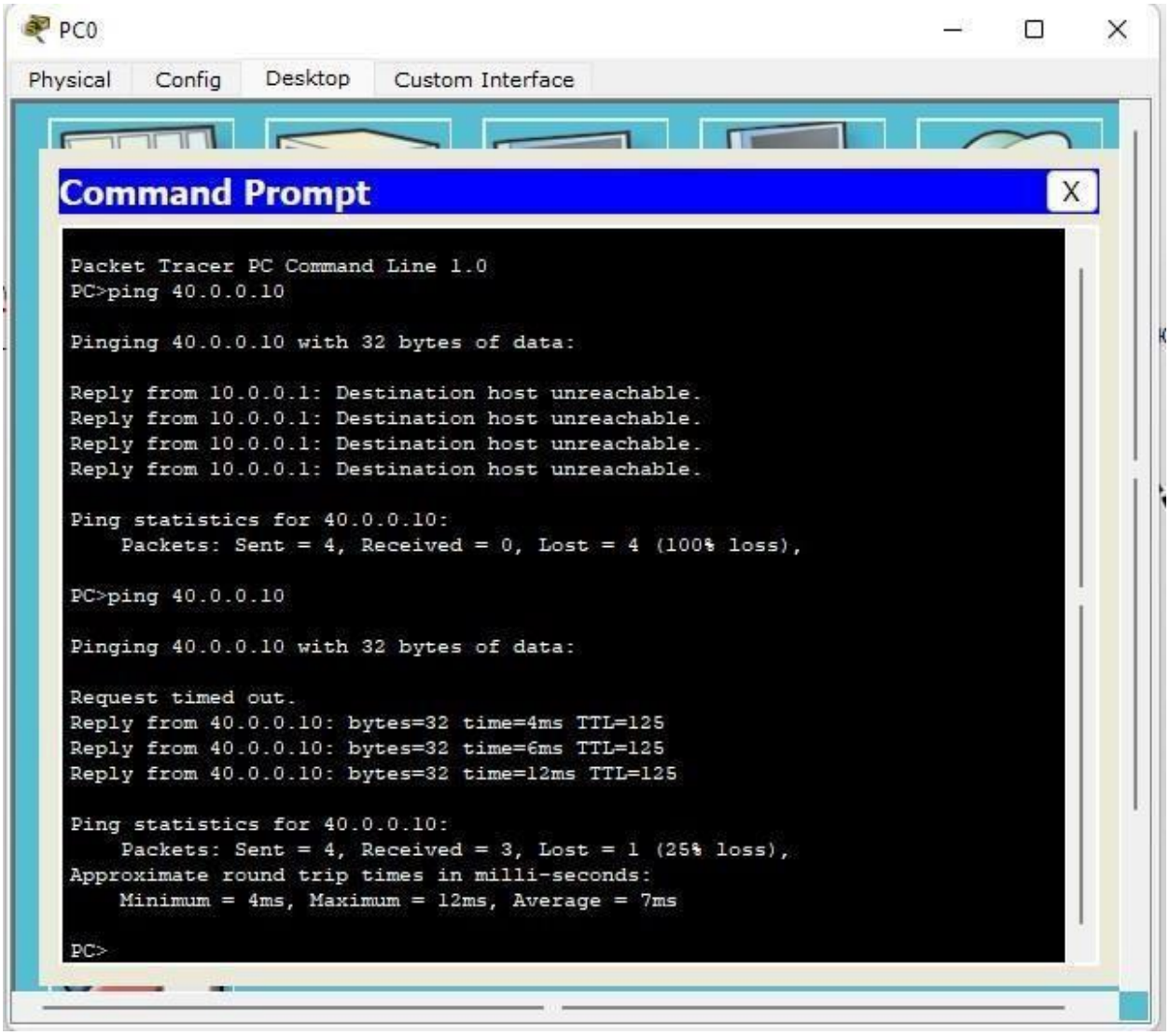
Pinging 40.0.0.2 with 32 bytes of data:
Reply from 40.0.0.2: bytes=32 time=2ms TTL=125
" " " " time=10ms TTL=125
" " " " time=14ms TTL=125
" " " " time=2ms TTL=125

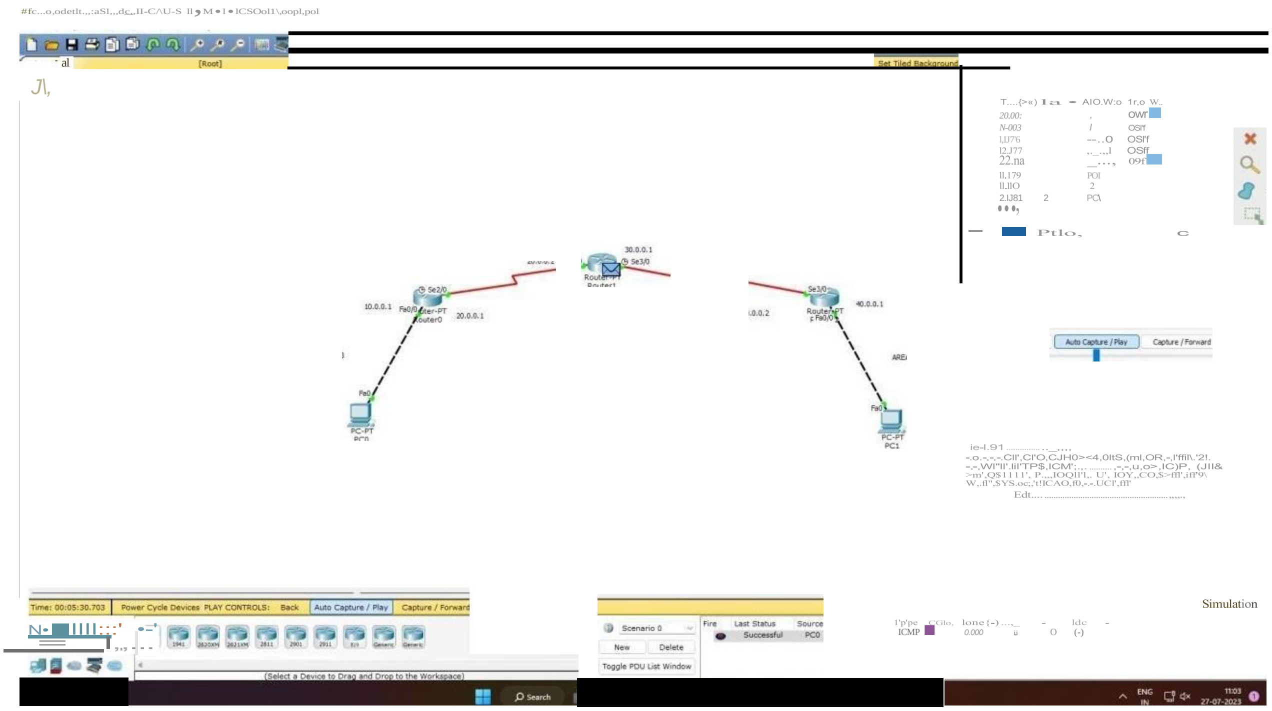
Packet: Sent: 4 Received: 4 Lost: 0 (0% loss)
Approximate round trip time in milliseconds
Minimum: 2ms, Maximum: 14ms, Average: 7ms.

TOPOLOGY:


AREA 1
20.0.0.2
Se2/0
30.0.0.1
Se3/0
AREA 0
Router-PT
Router1
Se2/0
10.0.0.1
Fa0/0
Router-PT
Router0
20.0.0.1
Se3/0
30.0.0.2
Router-PT
Fa0/0
40.0.0.1
AREA 3
AREA 2
Fa0
PC-PT
PC0
10.0.0.10
Fa0
PC-PT
PC1
40.0.0.10

OUTPUT:

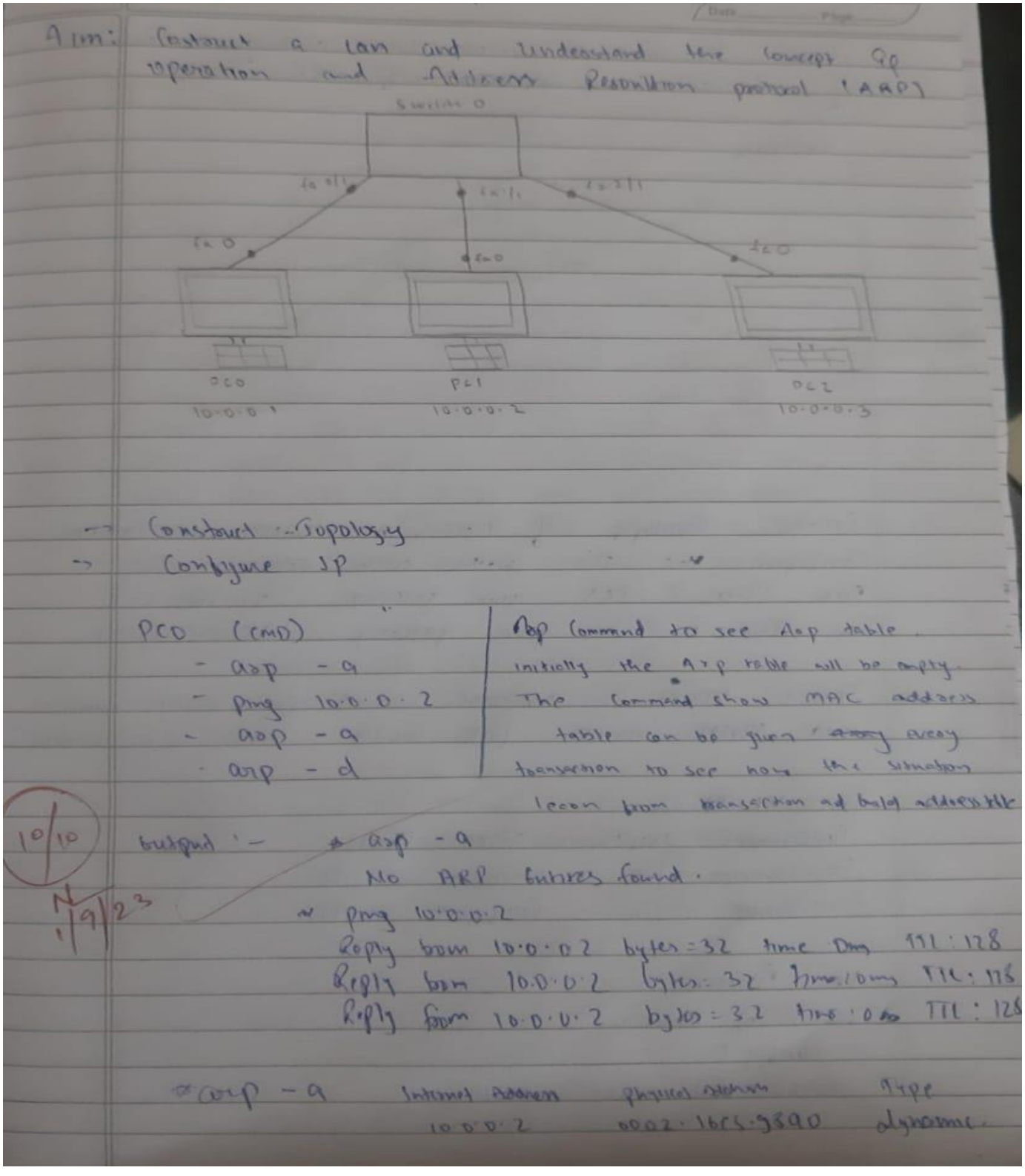

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.10
Pinging 40.0.0.10 with 32 bytes of data:
Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.
Ping statistics for 40.0.0.10:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 40.0.0.10
Pinging 40.0.0.10 with 32 bytes of data:
Request timed out.
Reply from 40.0.0.10: bytes=32 time=4ms TTL=125
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125
Reply from 40.0.0.10: bytes=32 time=12ms TTL=125
Ping statistics for 40.0.0.10:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 4ms, Maximum = 12ms, Average = 7ms
PC>

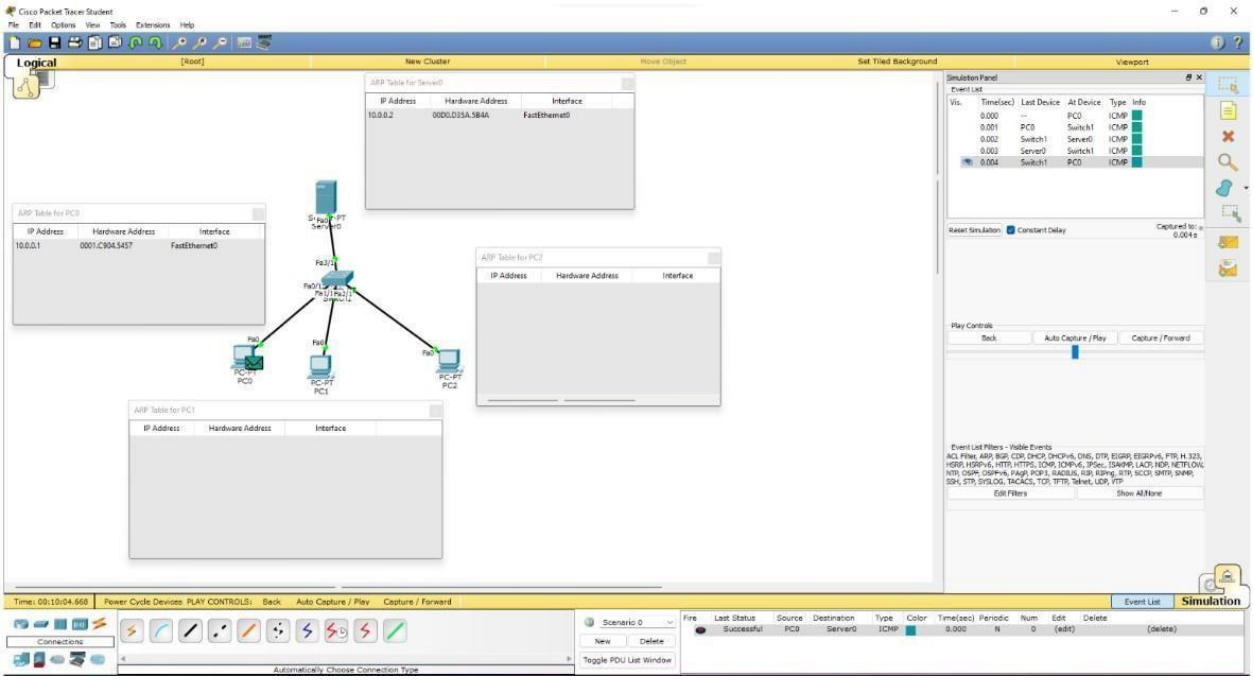

[Root]
Set Tiled Background
30.0.0.1
Se3/0
Se2/0
10.0.0.1
Fa0/0
Router0
20.0.0.1
Se3/0
40.0.0.1
PC-PT
PC1
Auto Capture / Play
Capture / Forward
Simulation
ICMP
Time: 00:05:30.703
Power Cycle Devices
PLAY CONTROLS:
Back
Scenario 0
Fire
Last Status
Source
Successful
PC0
New
Delete
Toggle PDU List Window
(Select a Device to Drag and Drop to the Workspace)
1941
2811
2901
2911
Generic
Search
ENG
IN
11:03
27-07-2023

LAB 8

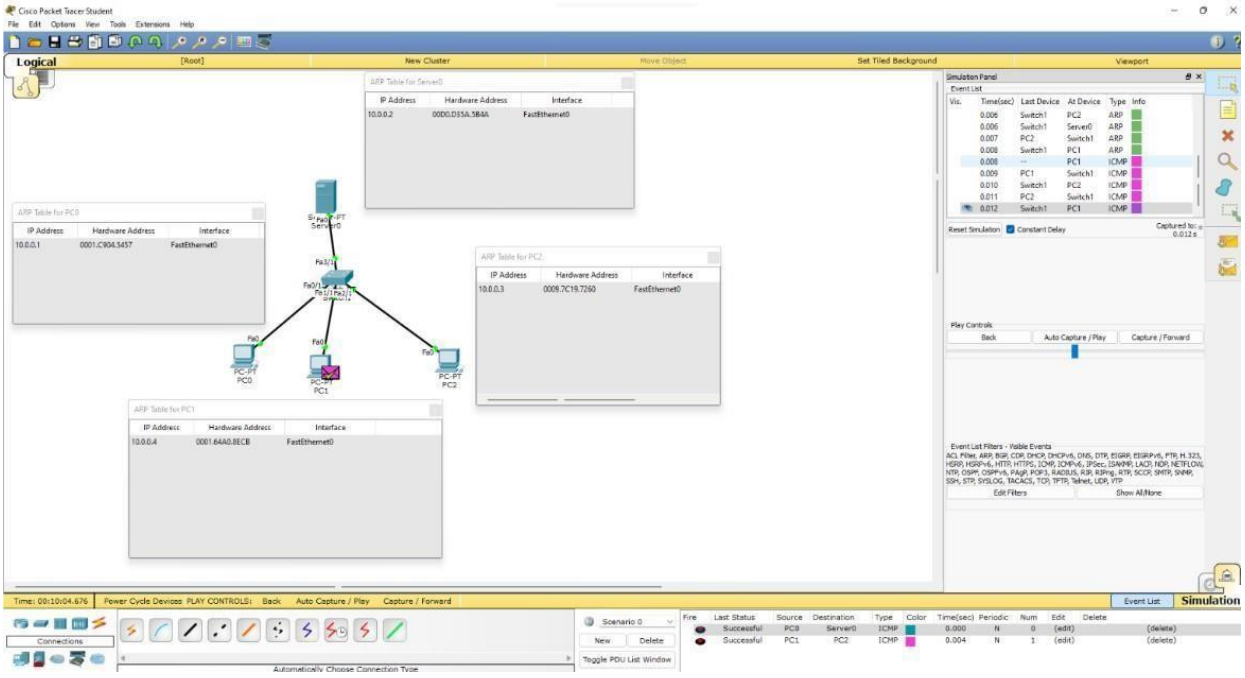
To construct a simple LAN and understand the concept and operation of Address Resolution Protocol (ARP).

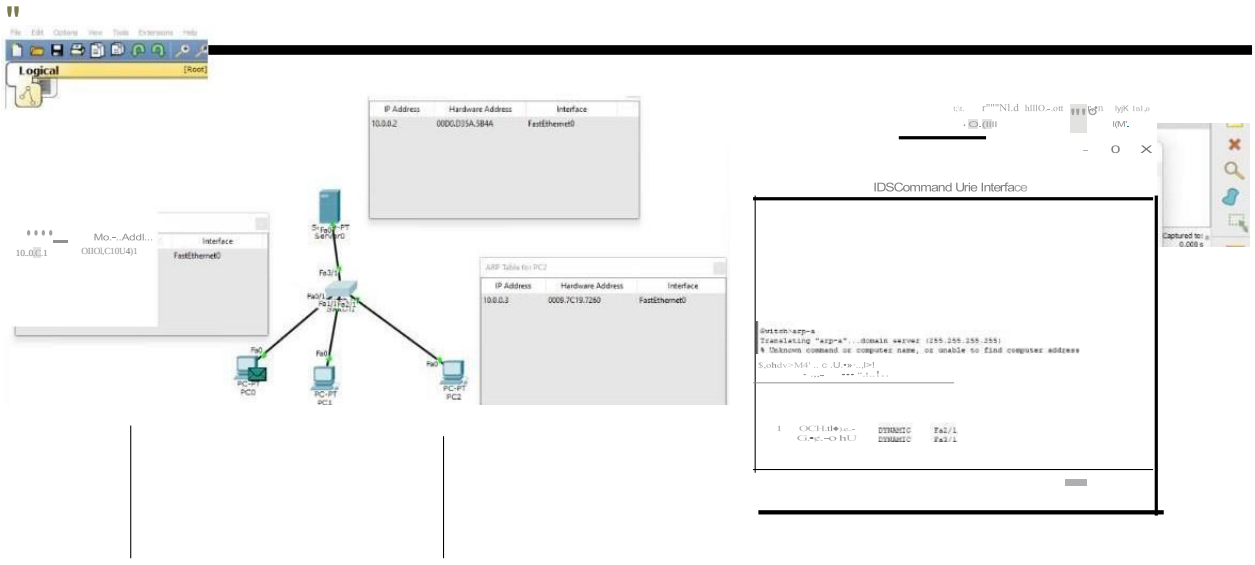
OBSERVATION:





OUTPUT:




Logical
[Root]
IP Address
Hardware Address
Interface
10.0.0.2
00D0.D35A.5B4A
FastEthernet0
Interface
FastEthernet0
ARP Table for PC2
IP Address
Hardware Address
Interface
10.0.0.3
0009.7C19.7260
FastEthernet0
IDSCommand Urie Interface
Switch>arp -a
Translating "arp-a"...domain server (255.255.255.255)
% Unknown command or computer name, or unable to find computer address
Captured to:
0.000 s

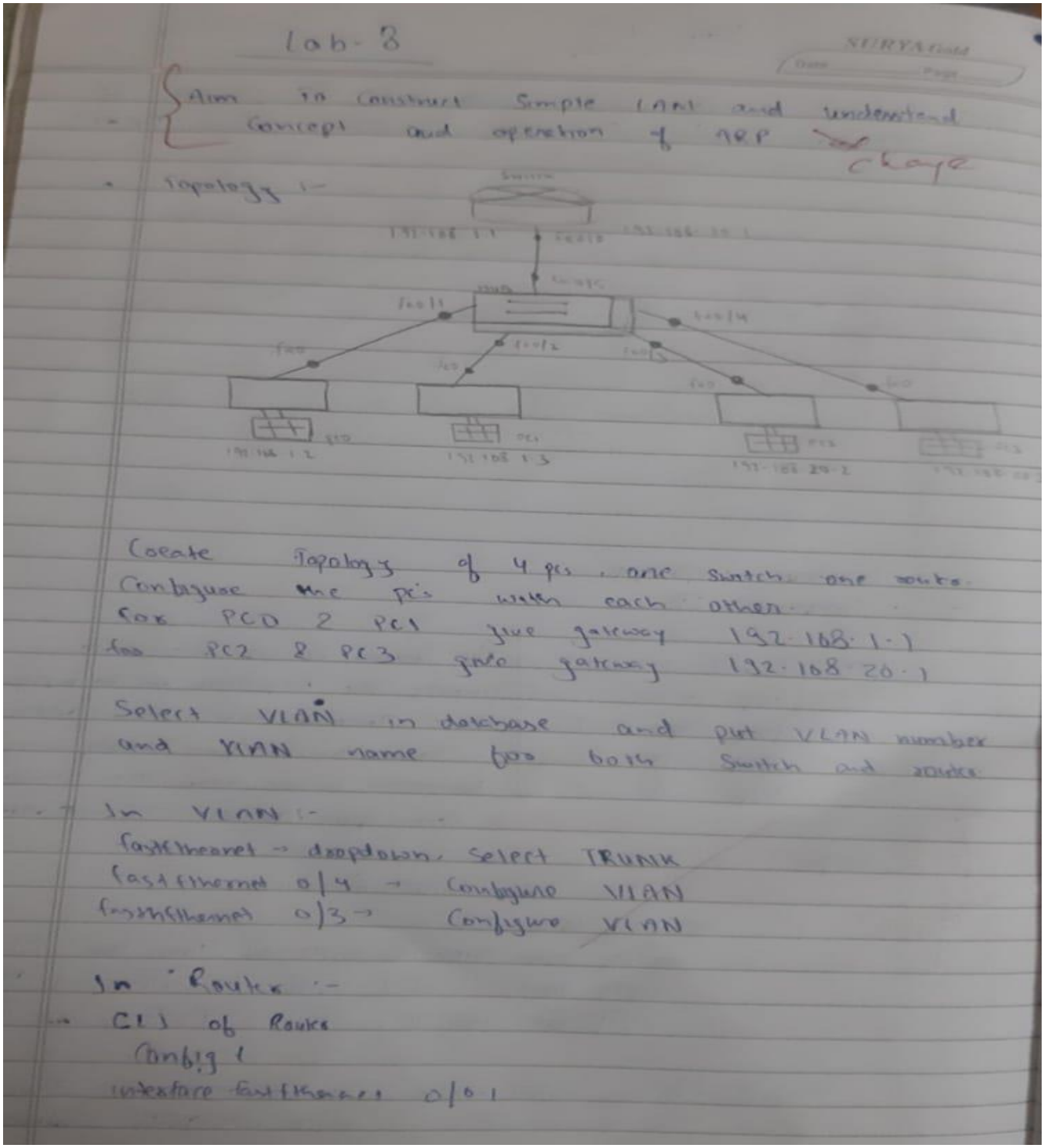
Simulation


Fire
Last Status
Source
Destination
In Progress
PC0
Server0
Successful
PC1
PC2
Num
Edit
Delete
0
(edit)
1
(edit)

LAB 9

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

OBSERVATION:



encapsulation dot 1Q 1
ip address 192.168.20.1 255.255.255.0
no shut
exit

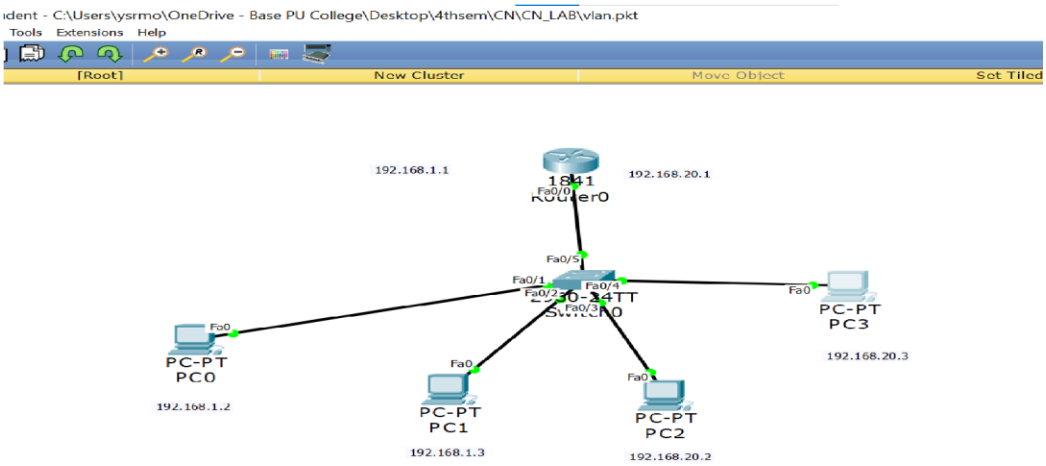
Output :-

ping 192.168.20.3
pinging 192.168.20.3 with 32 bytes of data
Reply from 192.168.20.3 : byte : 32 time : 0ms TTL 127
Reply from 192.168.20.3 : byte : 32 time 0ms TTL 127
Reply from 192.168.20.3 byte 32 time 0ms TTL 127
Reply from 192.168.20.3 byte 32 time 0ms TTL 127

Ping statics for 192.168.20.3
packets : Sent = 4, Received = 4 Lost : 0 (0% loss).

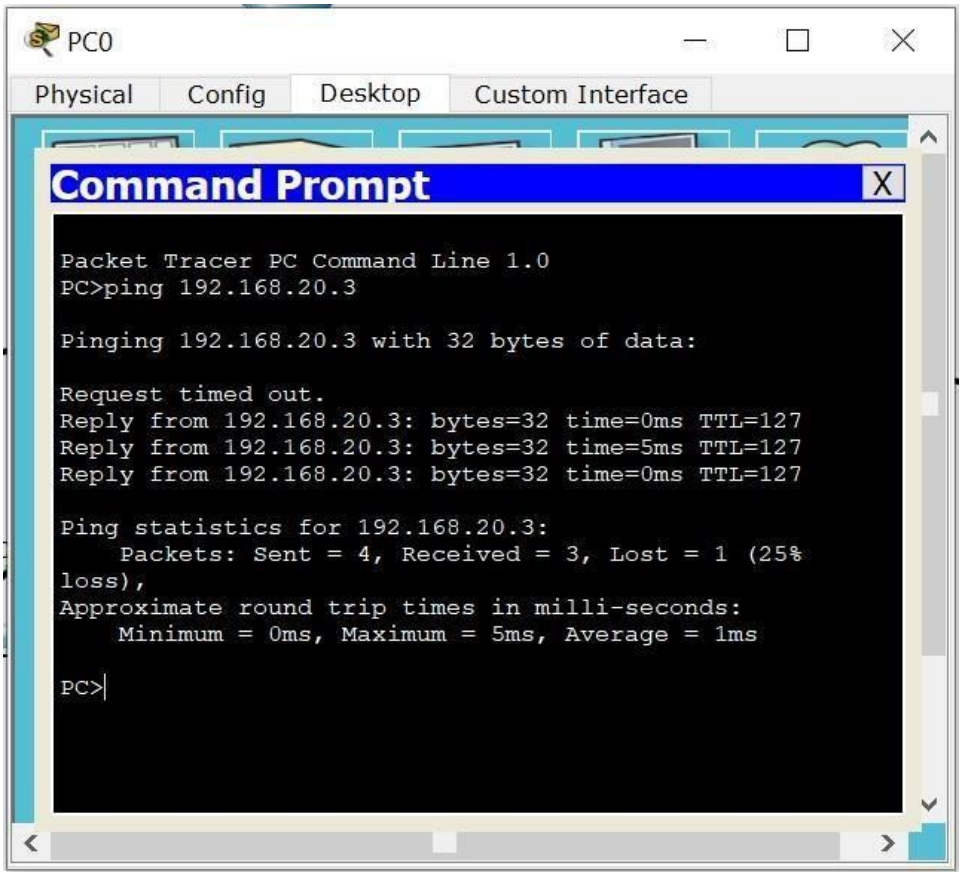
Aim : To construct VLAN and make the pc communication among VLAN

19/23


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
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

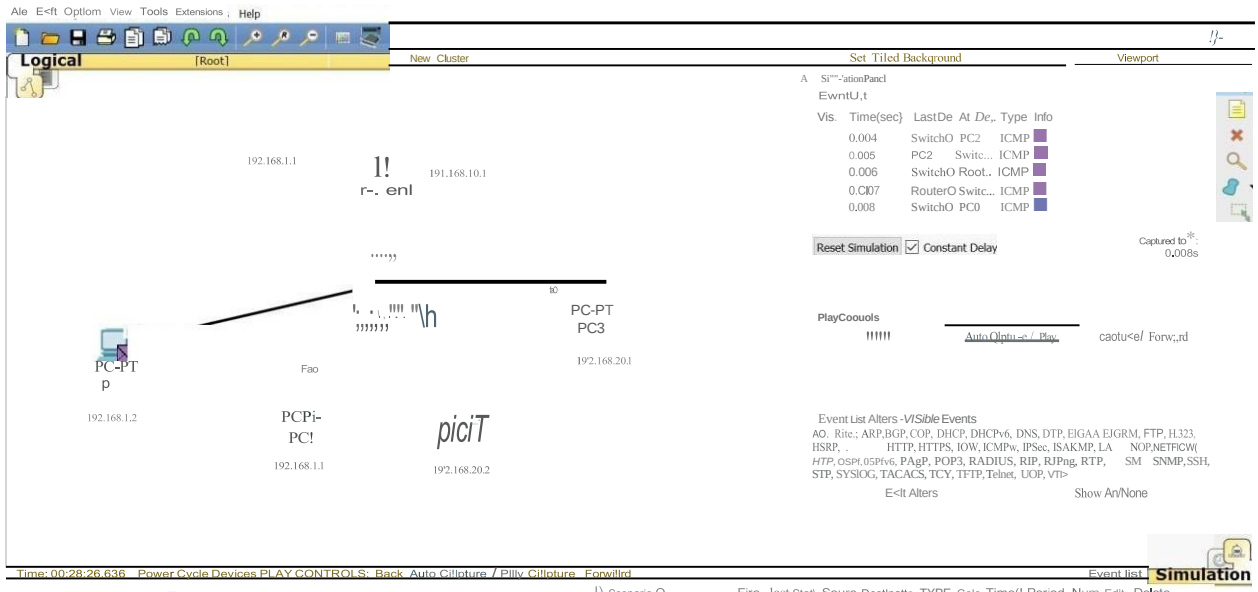
T

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>

Cisco Pad:et Tracer Student - C:\IJsefs\ysrmo\OneOrive- Base PU College\Desktop\4thsem\CN\CN_LAB\wlan.plcr

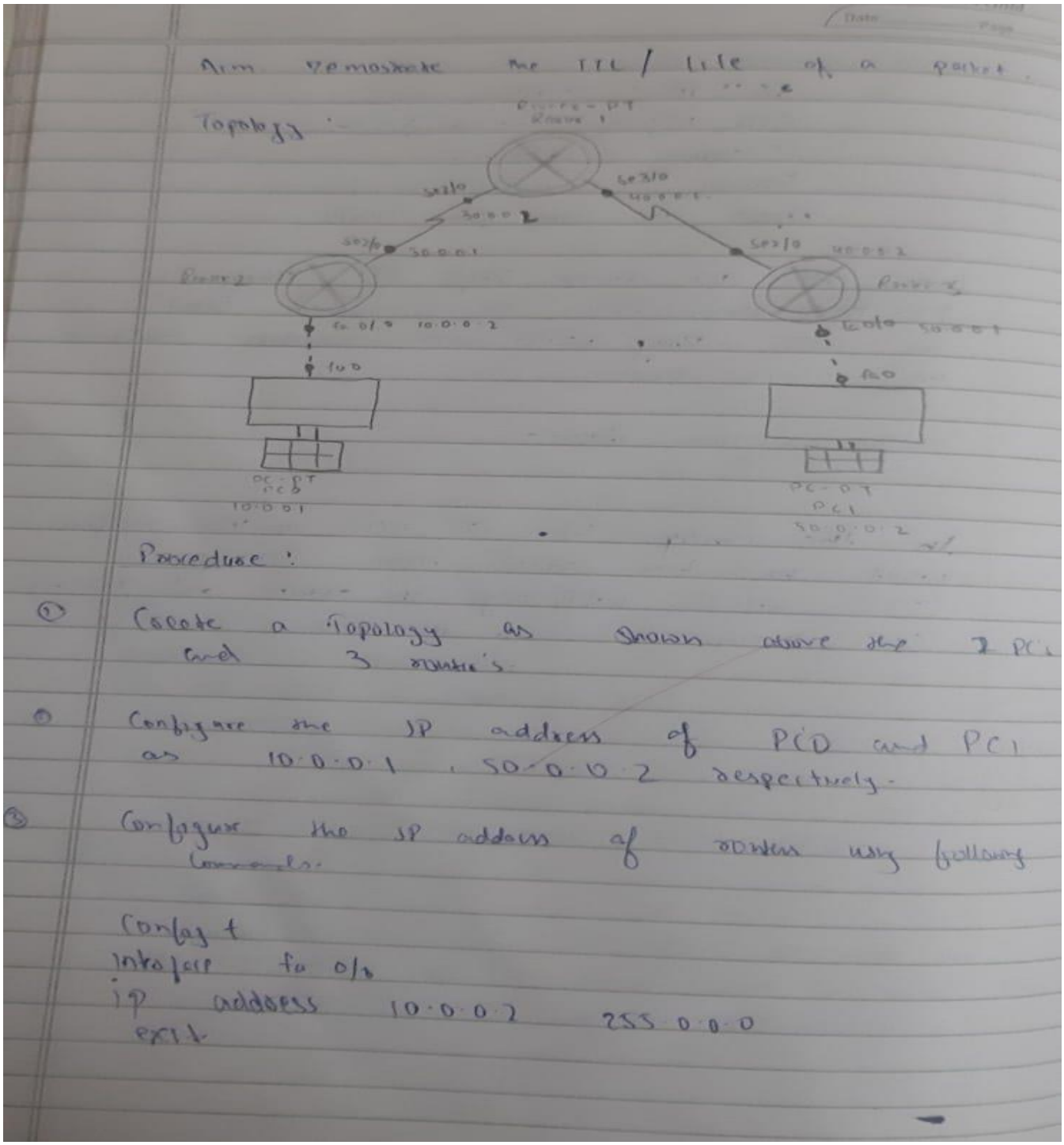
CJ X



LAB 10

Demonstrate the TTL/ Life of a Packet.

OBSERVATION:

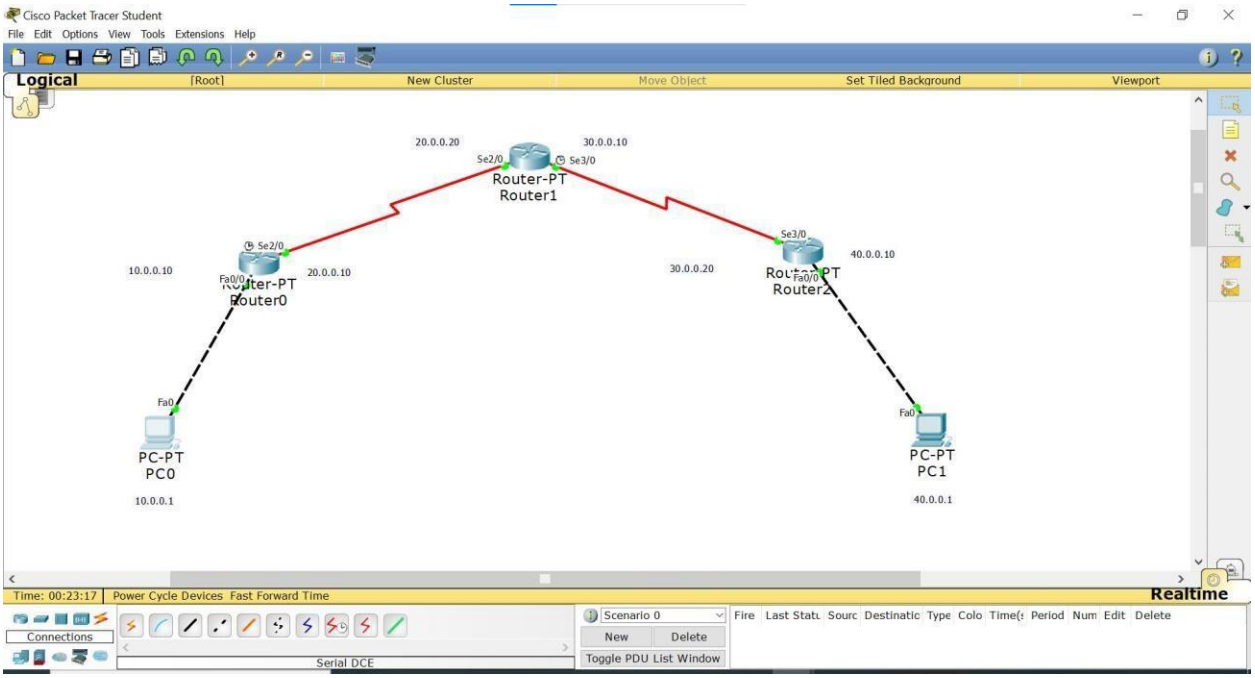


SURYA Gold
Date Page

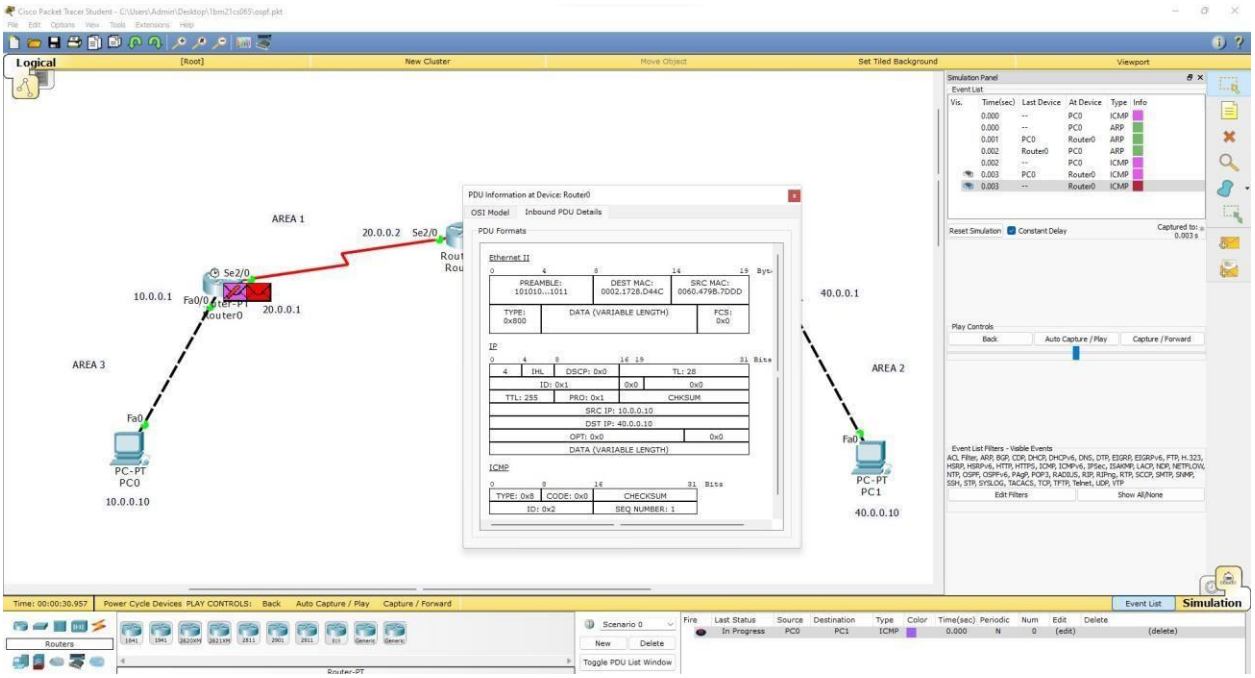
1. Configure the source using default / static routing.
2. In Simulation mode, send a simple PDU from one PC to another.
3. Use Capture button to capture every transfer.
4. Click on PDU during ICMP transfer to see the Inbound & outbound PDU details.

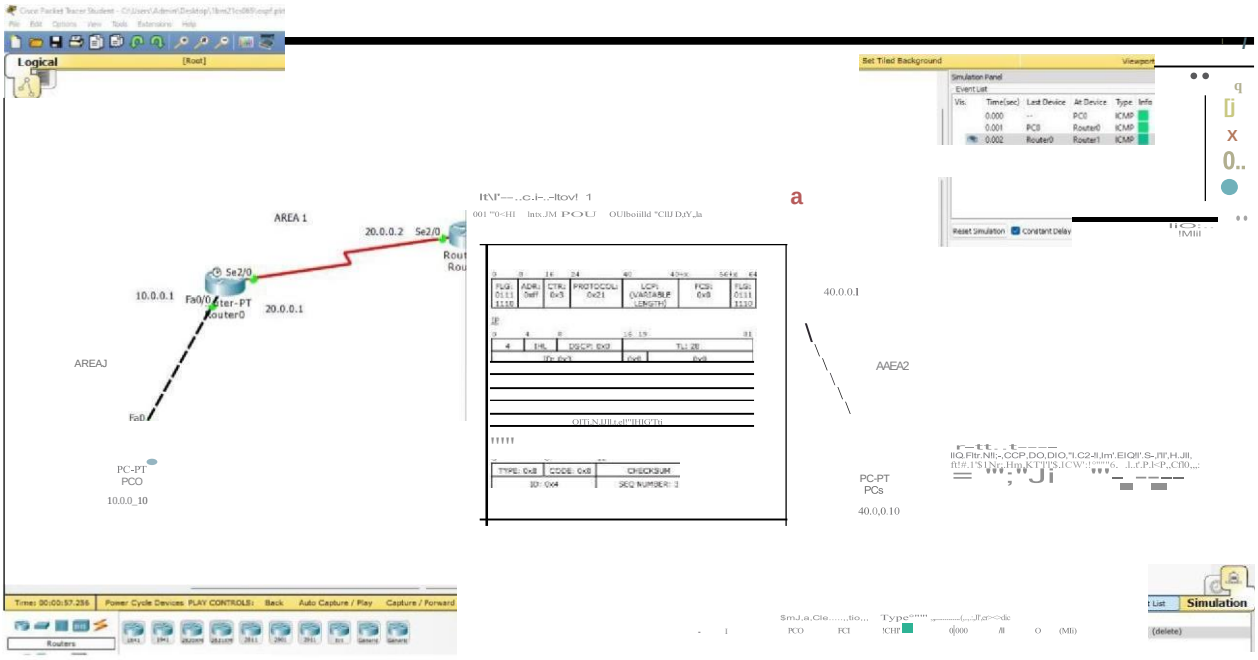
10/10

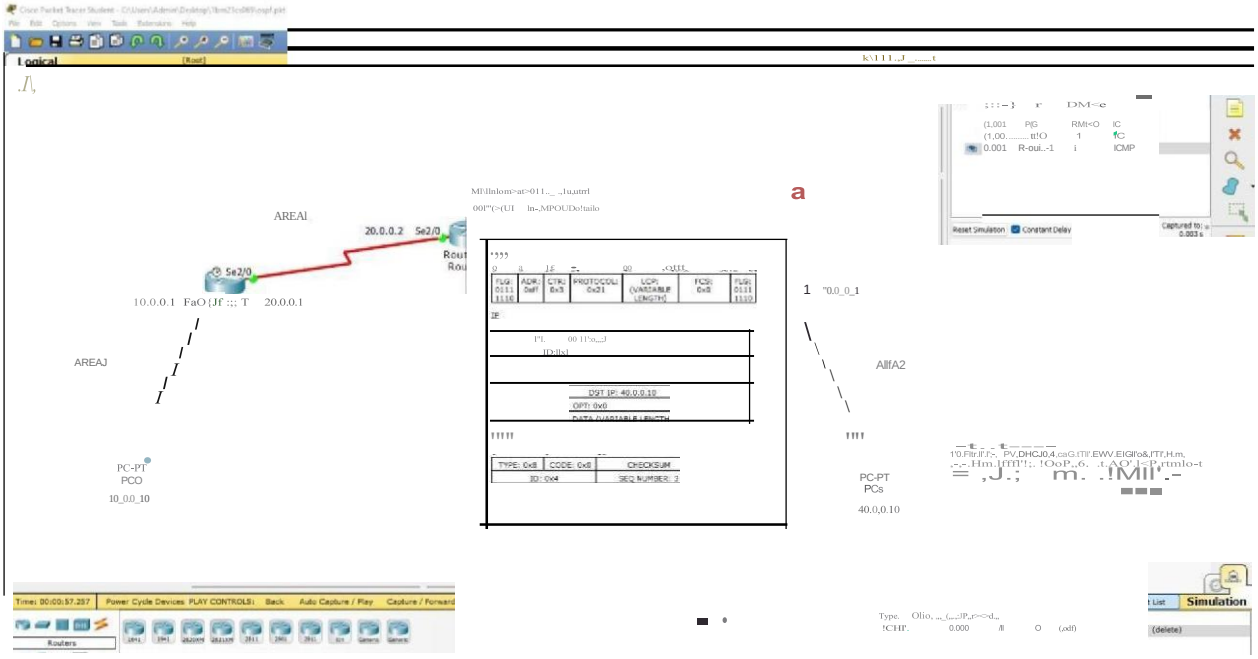
TOPOLOGY:

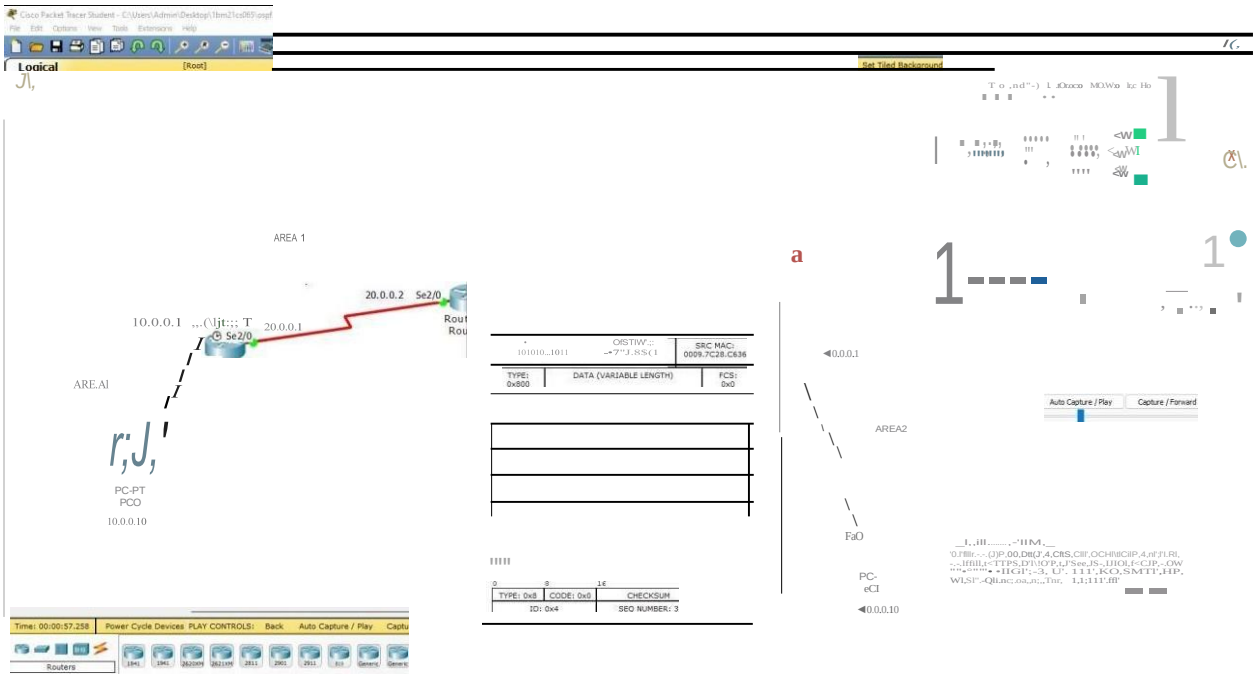


OUTPUT:




Logical
[Root]
AREA 1
20.0.0.2
Se2/0
Se2/0
10.0.0.1
Fa0/0
20.0.0.1
AREAJ
PC-PT
PCO
10.0.0.10
40.0.0.1
AAEA2
PC-PT
PCs
40.0.0.10
Simulation

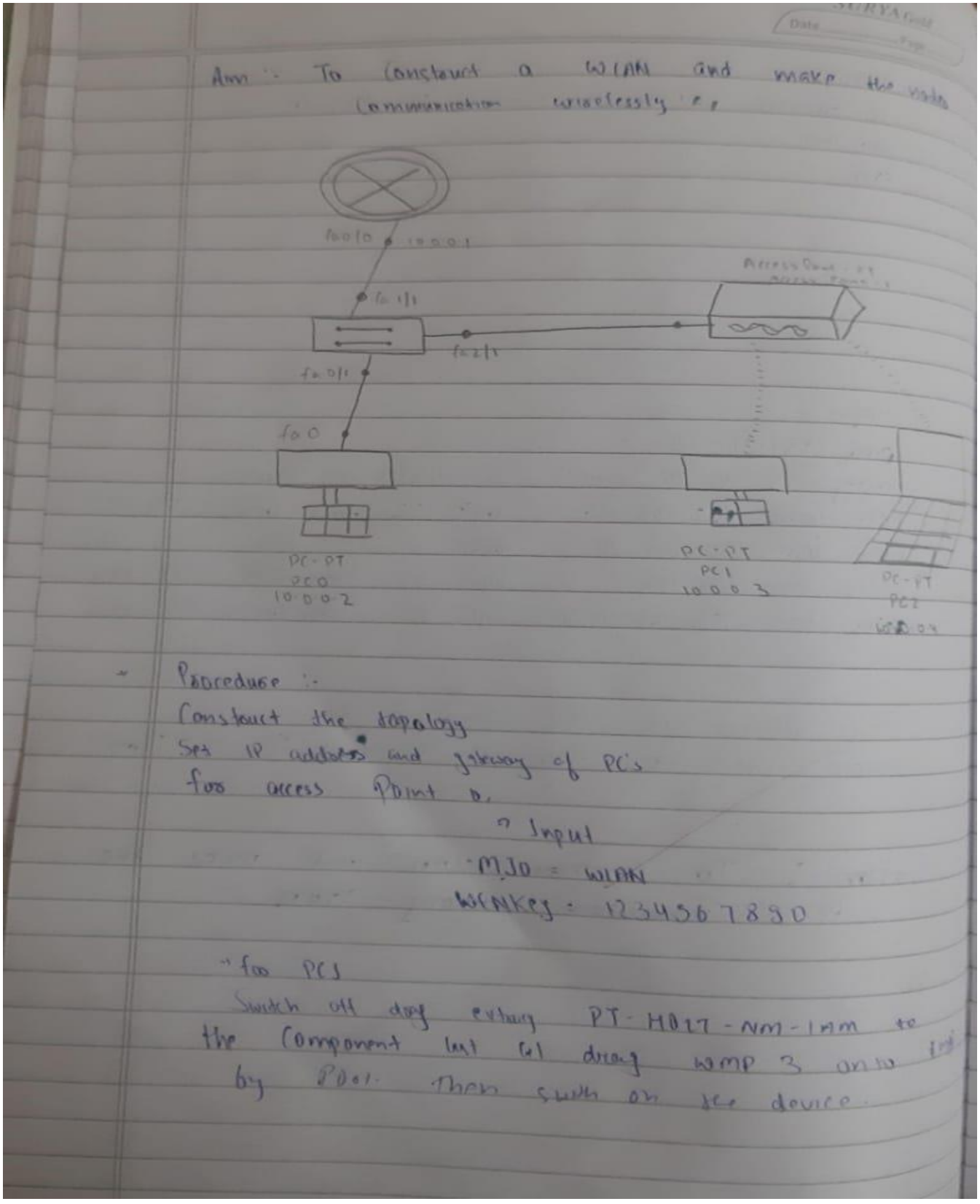

Logical
[Root]
AREAl
20.0.0.2
Se2/0
Se2/0
10.0.0.1
20.0.0.1
AREAJ
PC-PT
PCO
10_0.0_10
1 "0.0_0_1
AllfA2
PC-PT
PCs
40.0.0.10
Simulation


Logical
[Root]
Set Tiled Background
AREA 1
10.0.0.1
Se2/0
20.0.0.1
20.0.0.2
Se2/0
ARE.Al
PC-PT
PCO
10.0.0.10
SRC MAC: 0009.7C28.C636
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x4
SEQ NUMBER: 3
a
◄0.0.0.1
AREA2
FaO
PC-
eCI
◄0.0.0.10
Auto Capture / Play
Capture / Forward
Time: 00:00:57.258
Power Cycle Devices
PLAY CONTROLS:
Back
Auto Capture / Play
Routers

Simulation

LAB 11

construct a WLAN and make the nodes communicate wirelessly
OBSERVATION



In wireless 0:

SSID - WLAN

WEN key - 1234567890

IP address - 10.0.0.3

Gateway - 10.0.0.1

Repeat the same for laptop

Switch off the device. Drag the existing PT-HOST-NM-1AM to component listed in LHS Drag WMP300N Wireless interface to empty port.

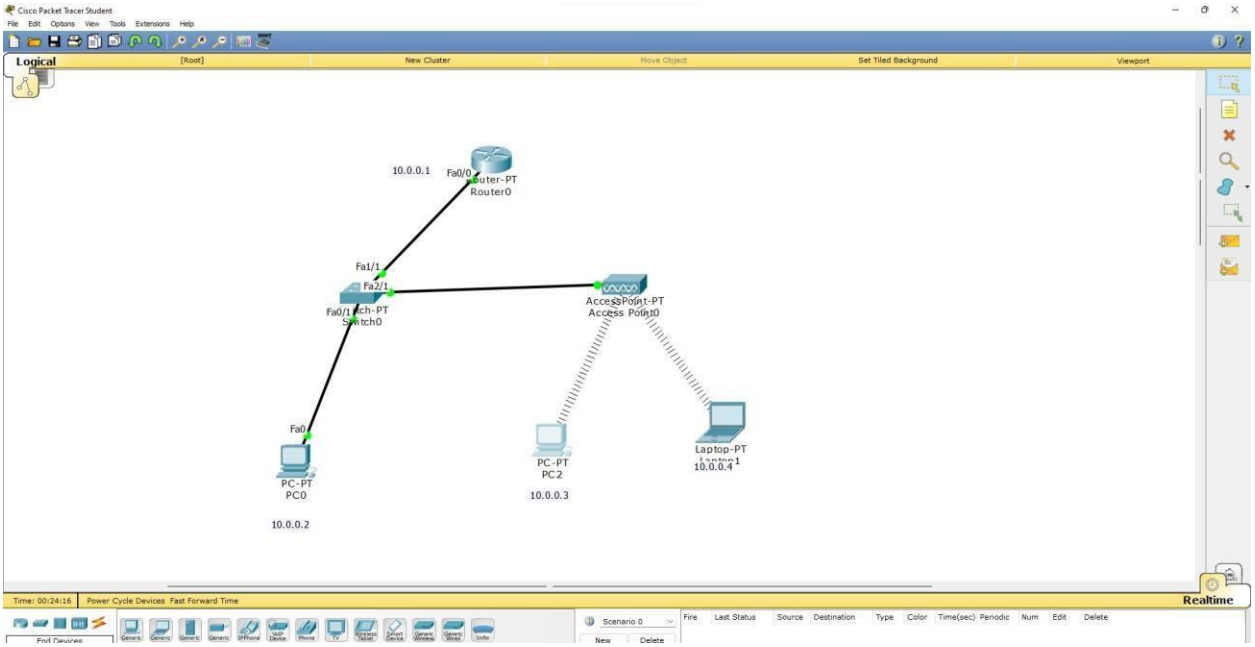
Switch on the device.

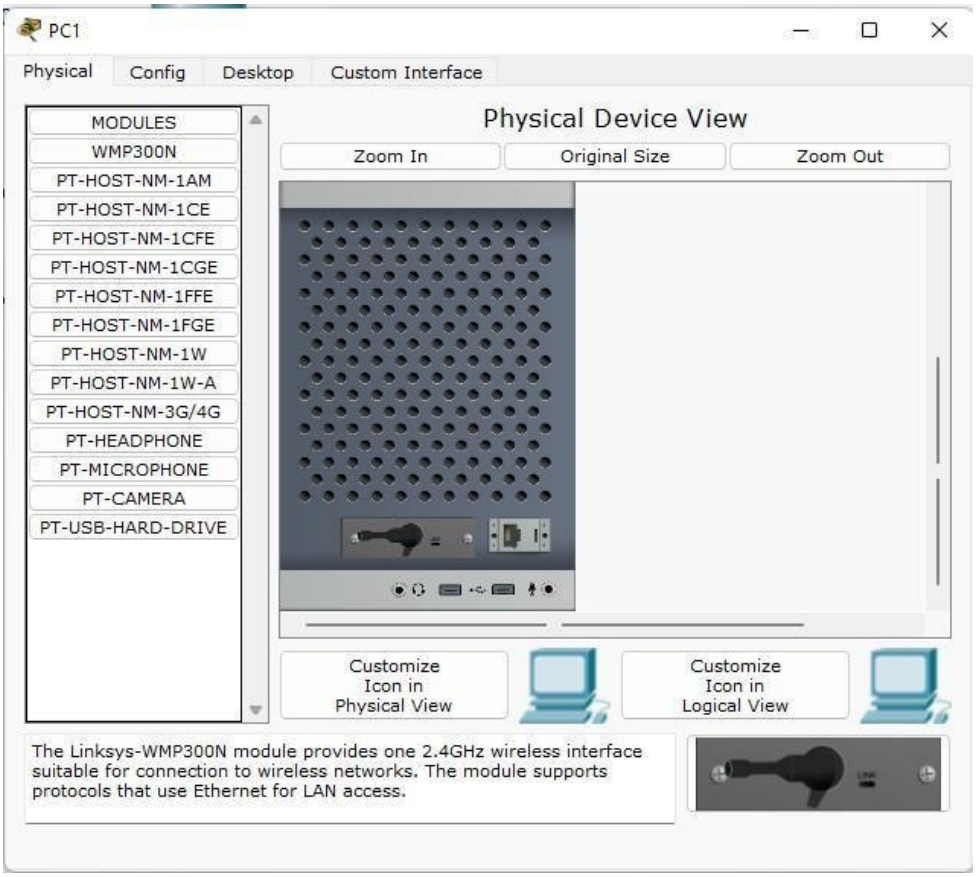
In the Config tab. a new wlan1 interface would have been added. Now Configure SSID, WEP, WPP key. Ip address and gateway (as manually done) to the device

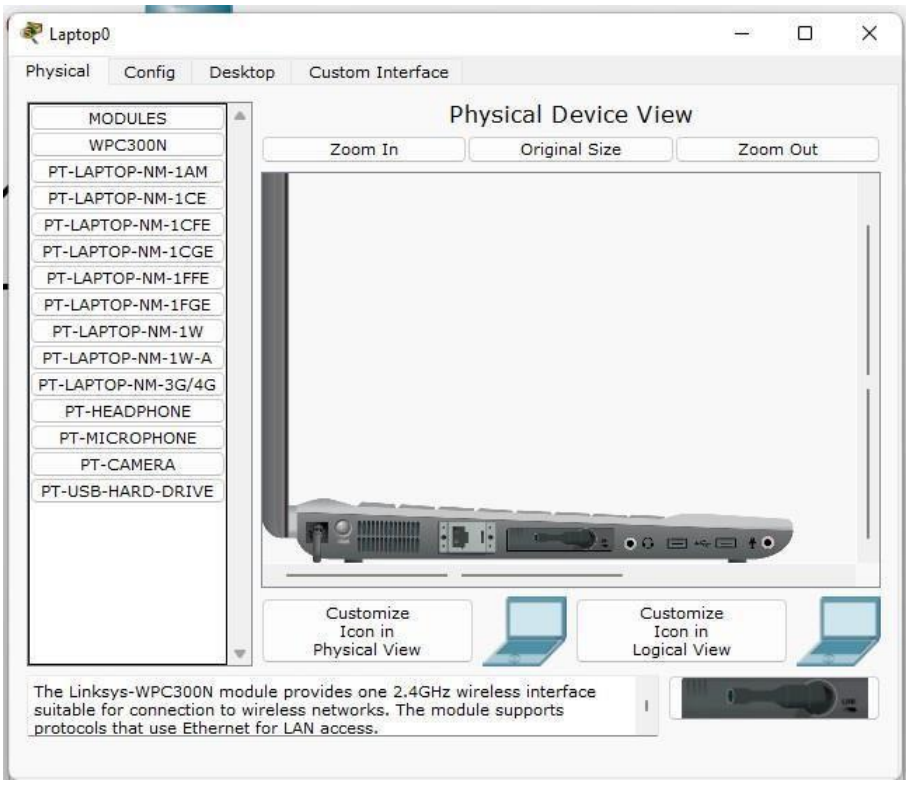
10/10

N

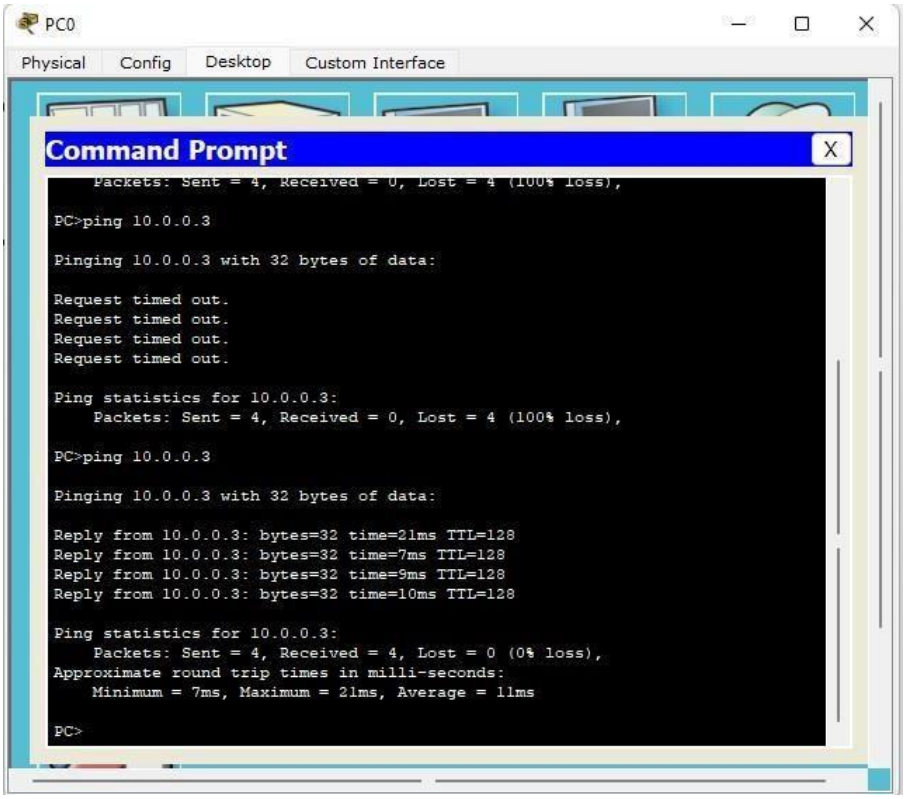
TOPOLOGY:


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

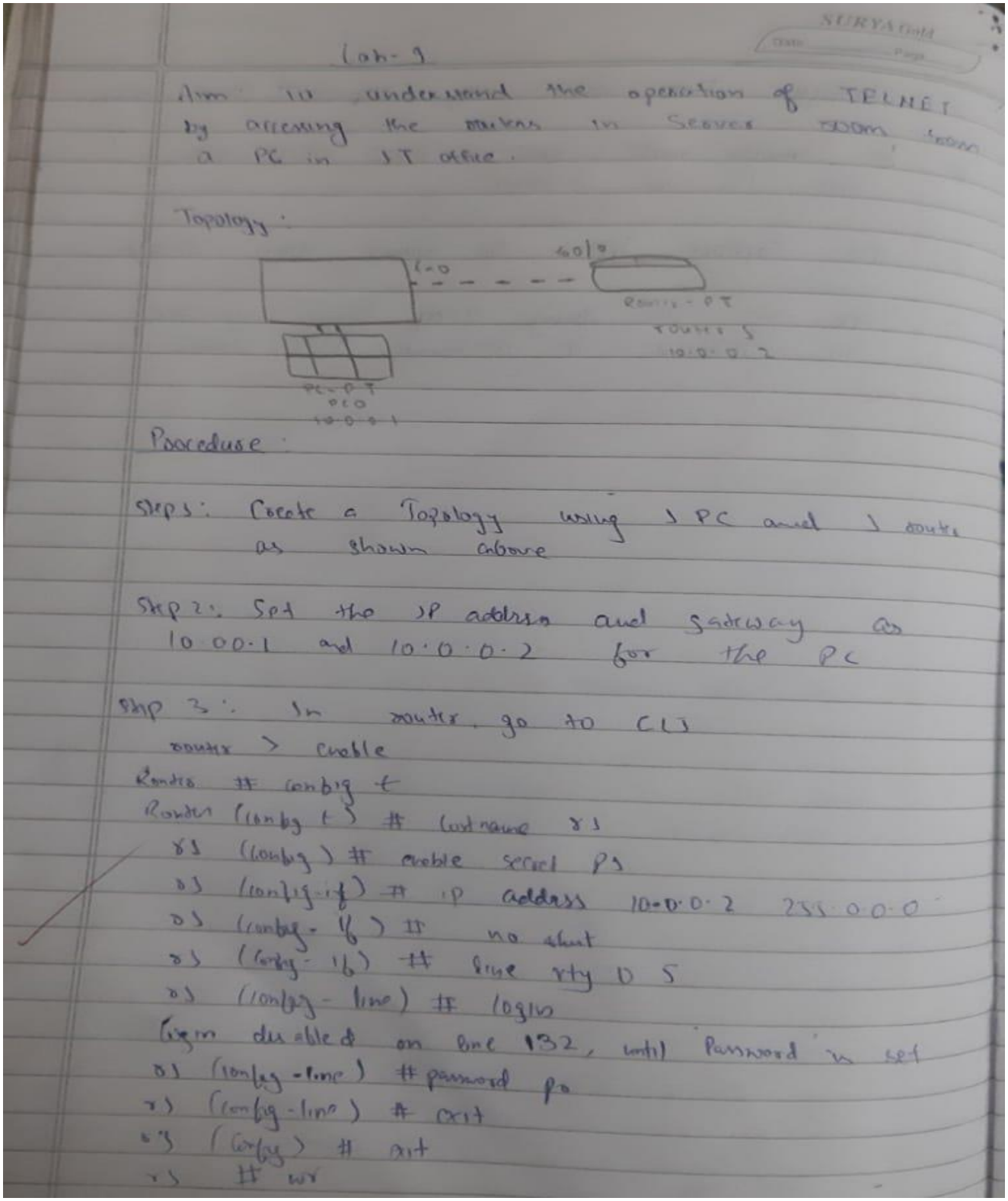
OUTPUT:


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:



SURYA Gold
Date Page

Step 4: In Command prompt of PC,

PC > ping 10.0.0.2

pinging 10.0.0.2 with 32 bytes of data:

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

Ping statistics for 10.0.0.2:
Packet Sent: 4, Received: 4, Lost: 0 (0% loss)
Approximate round trip time in milliseconds
Minimum: 0ms, Maximum: 0ms, Average: 0ms

PC > telnet 10.0.0.2
Trying 10.0.0.2 open
User Access verification
Password: (PO)
R1 > enable
Password: (PS)
R1 # show ip route
Codes: C - connected
C 10.0.0.0/8 is directly connected, FastEthernet 0/0

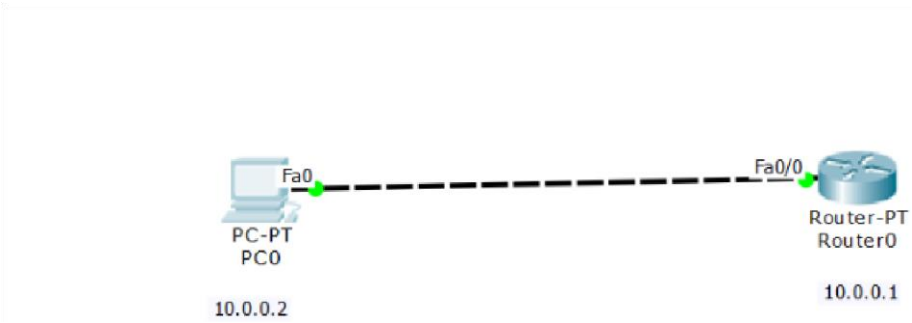
LS

observation: using telnet protocol we can access the router from the PC (which is connected to it).

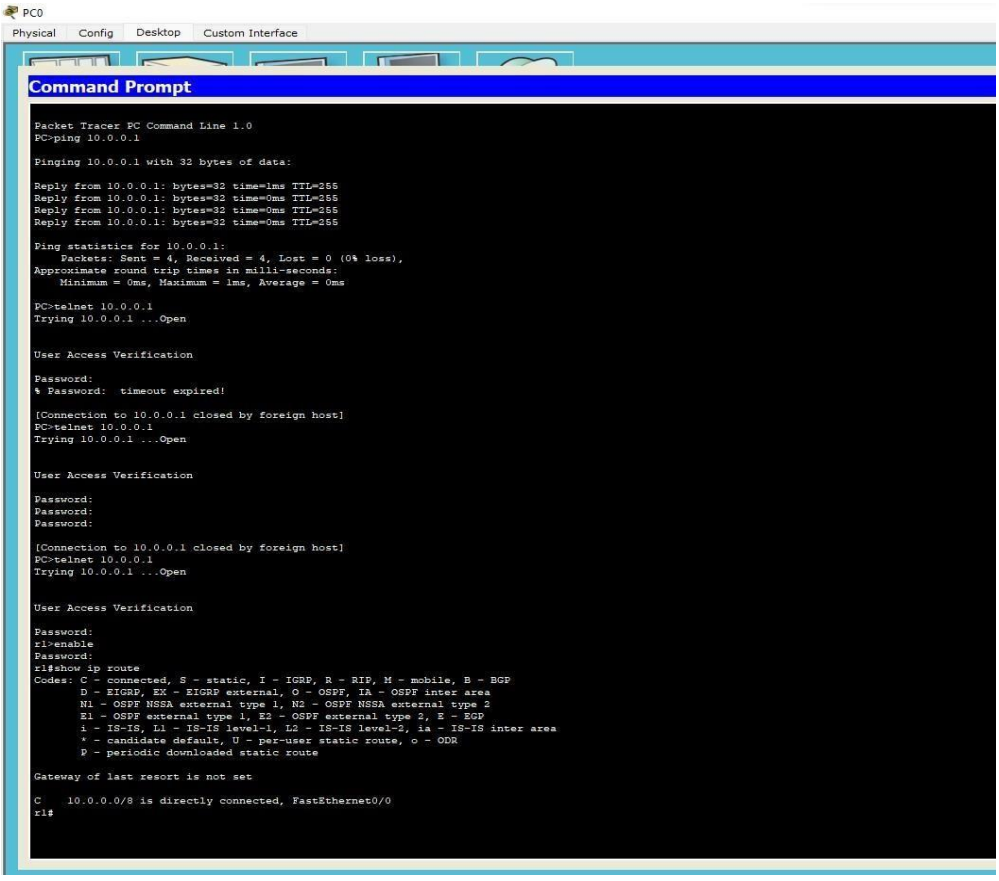
8/10

1/9/23

TOPOLOGY:



OUTPUT:



LAB 13

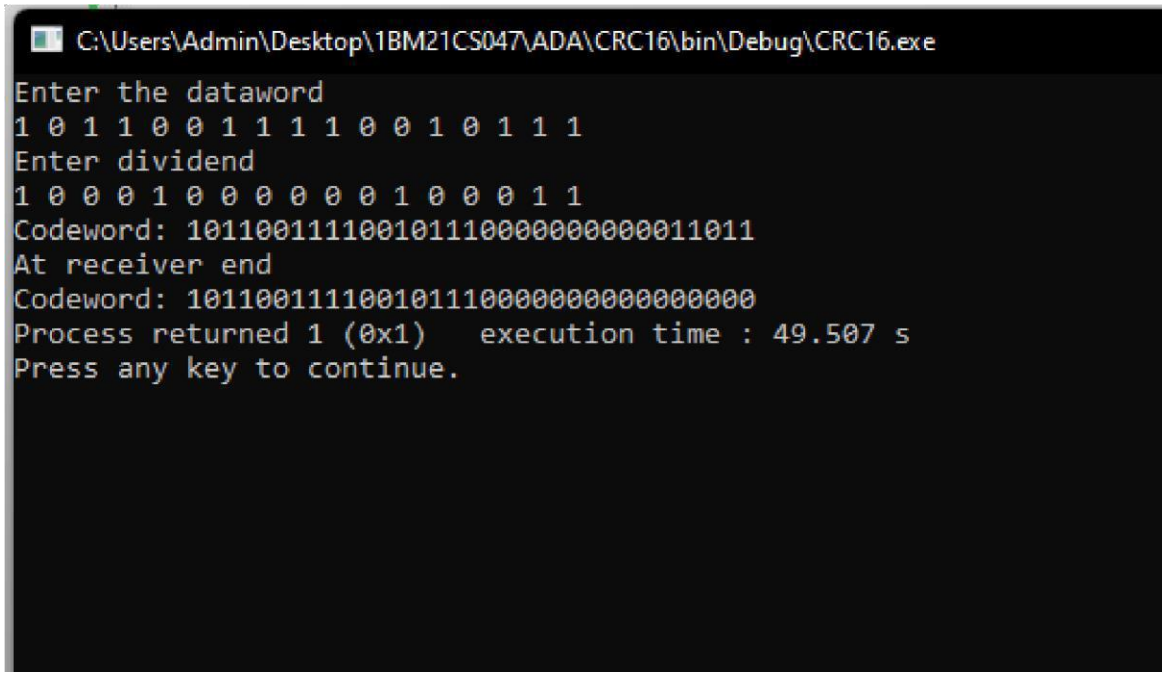
Write a program for error detecting code using CRC- CCITT (16bits).

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:



OBSERVATION:

Cycle - 2

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

Program

```
#include <stdio.h>
char m[50], g[50], a[50], q[50], temp[50];
void caltans(int);
void xor(int);
void aloram();
void shift();
void main()
{
    int n, i=0;
    char ch, flag = 0;
    printf("Enter the frame bits:");
    while((ch=getc(stdin)) != '\n');
        m[i++] = ch;
        n = i;
    for(i=0; i<16; i++)
        m[n++] = '0';
        n[n] = '\0';

printf("message after appending 16 zeros : %s", m);
for(i=0; i<16; i++)
    g[i] = '0';
g[0] = g[4] = g[11] = g[16] = '1';
g[17] = '\0';
printf("\n generator : %s \n", g);
xor(n);
```

```
printf ("\n quotient %s", q);
calltrans (n);
printf ("\n transmitted frame %s", m);
printf ("\n Enter transmitted frame ");
scanf ("%s", m);
printf ("CRC checking \n");
crc(n);
printf ("\n \n last remainder %s", r)
  for (i=0; i<16; i++)
    if (r[i] != '0')
      flag = 1;
    else
      continue;
  if (flag == 1)
    printf ("Error during transmission");
  else
    printf ("\n Received frame is correct");
}

Void crc (int n)
{ int i,j;
  for (i=0; i<n; i++)
     temp[i] = m[i];
  for (i=0; i<n-16; i++)
    { if (r[0] == '1')
      { q[i] = '1';
        calcram ();
      }
    else
       q[i] = '0';     shift ();
    }
```

```
r[10] = m[17+i];
r[17] = '\0';
for (j=0; j<17; j++)
    temp[j] = r[j];
}
g[n-16] = '\0';
}

Void calcrem ()
{   int i, j;
    for(i=1; i<=16; i++)
      r[i-1] = ((int)temp[i]-48) ^ ((int)g[i]-48)+48;
}

Void shift()
{ int i;
  for(i=1; i<16; i++)
    r[i-1]=r[i];
}

Void getrem (int n)
{ int i, k=0;
  for(i=n-16; i<n; i++)
  m[i]= ((int)m[i]-48) ^ ((int)r[k++]-48]+48;
    m[i] = '\0';
}
```

Output: Enter bits: 1011

Message after appending 16 zeros: 1011 0000 0000 0000 0000

generator: 10010000010001

Quotient: 1011

transmitted frame: 1011 1011 0001 0110 1011

last remainder: 0000 0000 0000 0000

Received frame is correct

Lab 14

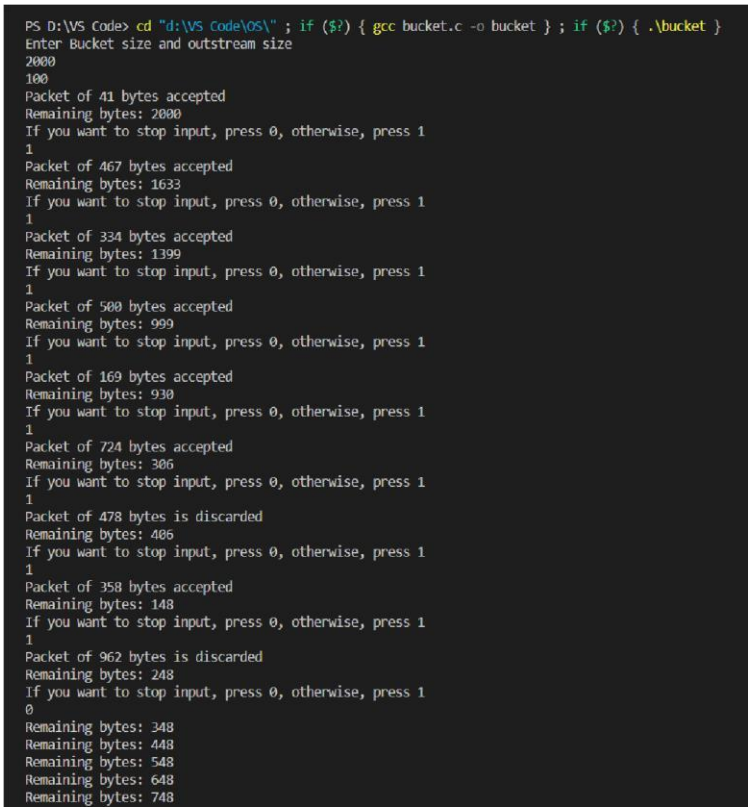
Write a program for congestion control using Leaky bucket algorithm.

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:



```
Remaining bytes: 3-48
Remaining bytes: 448
Rema.in.ing bytes: 548
H.emaining bytes: 648
Remaining bytes: 748
Remaining bytes: 848
Remaining bytes: 948
Remaining bytes: 1048
Remainine bytes:   48
Rema.in.ing bytes: 1248
Remaining bytes: 1348
Remaining bytes: 1448
Remainine bytes:  548
Remaining bytes: 1648
Remaining bytes: 1748
Remaining bytes: 1848
Rema n ng bytes: 1948
Rema..1.n..1.ng bytes: 2 00
PS D: ,vs  code,os> D
```

OBSERVATION:

2. Write a program for congestion control using leaky bucket algorithm

```
#include <stdio.h>
int main ()
{ int incoming, outgoing, bucket_size, n, store = 0;
printf (" Enter bucket size, outgoing rate and no of inputs : ");
scanf ("%d %d %d", &bucket_size, &outgoing, &n);
while (n != 0 )
{ printf (" Enter the incoming packet size : ");
  scanf ("%d", &incoming);
printf (" Incoming packet size %d out of %d in", store, bucket_size);
} else {
    printf (" Dropped %d no of packets in", incoming ((bucket - size. store));
    store = bucket - size;
}
printf (" After Outgoing %d packet left out %d in buffer
    in", store, bucket_size );
    n - -;
} return 0;
}
```

Output: Enter bucket size, Outgoing rate and no of inputs : 20 10 2
Enter incoming packet size : 30
Incoming packet size : 30
Dropped 10 no of packets
Bucket buffer size 0 out of 20
After Outgoing 10 packets left out 20 in buffer

9/10

1/9/23

Enter the incoming packet size : 10
Incoming packet size 10
Bucket buffer size 10 out of 20
After Outgoing 10 packets left out of 20 in buffer

Scan

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:

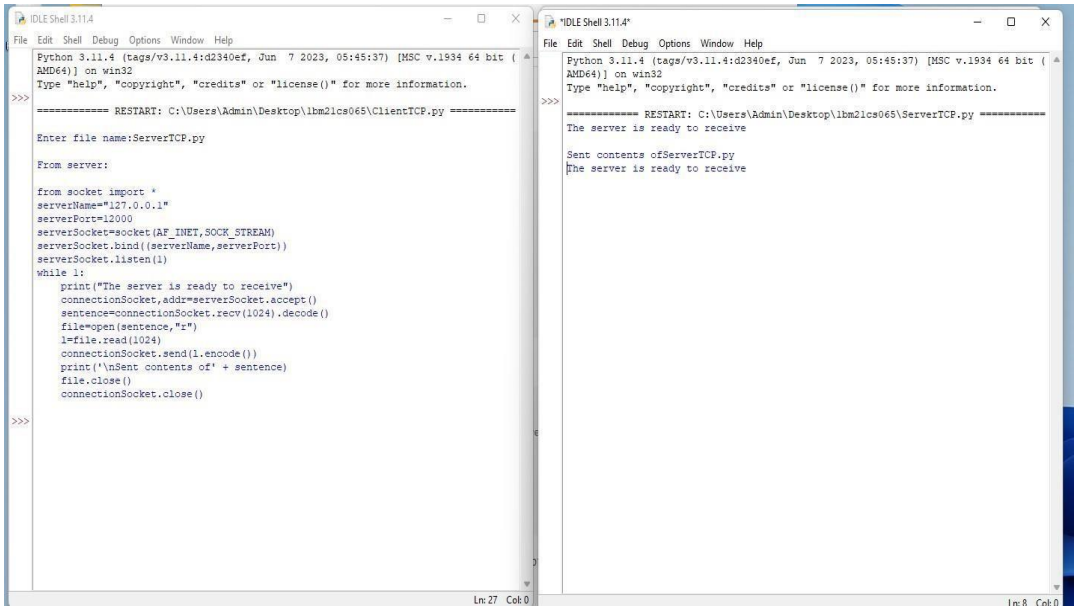
ClientTCP.py from socket import * serverName =

“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()

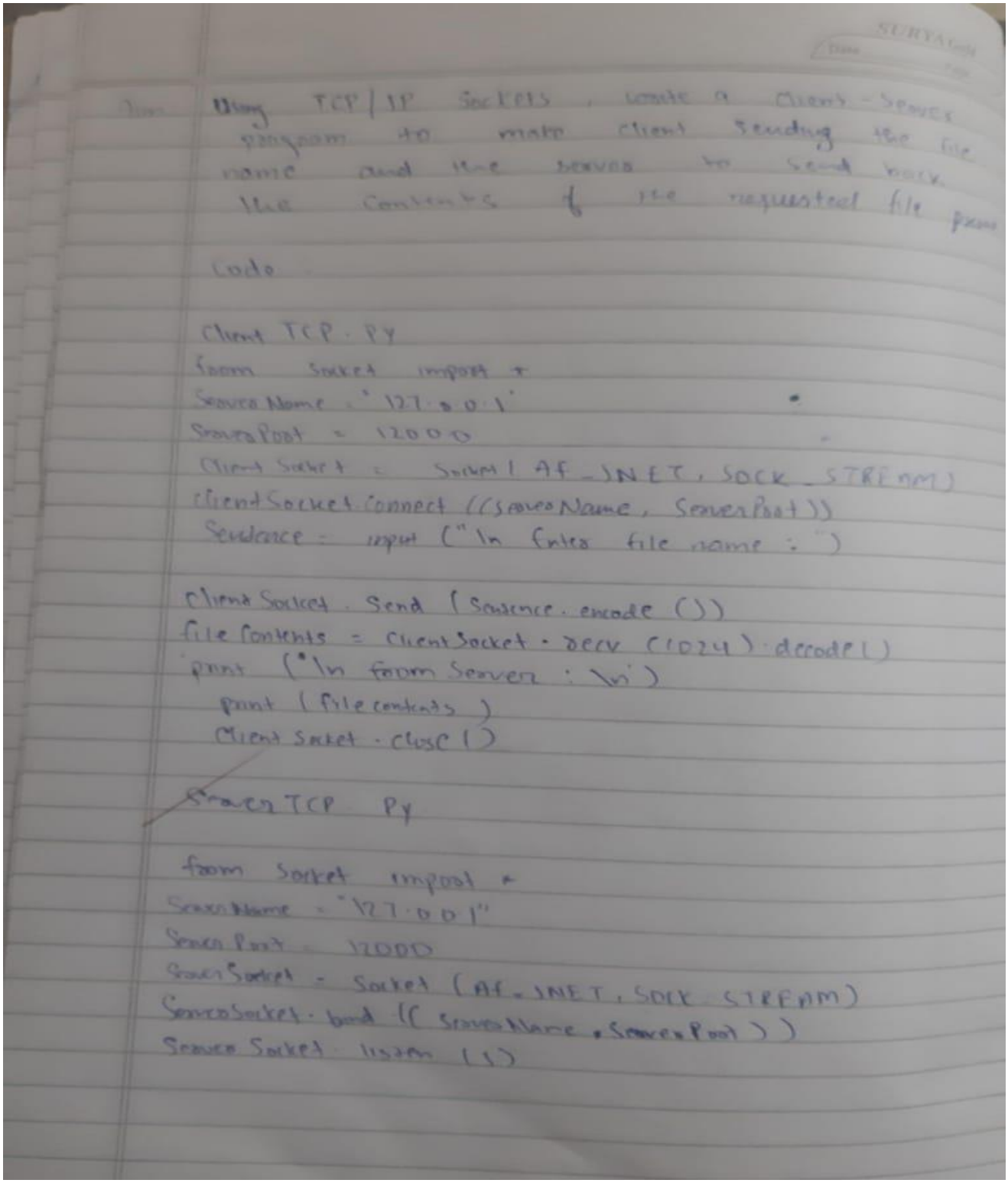
ServerTCP.py from socket import * serverName=”127.0.0.1” serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort)) serverSocket.listen(1) while 1: print
(“The server is ready to receive”)

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:



Using TCP/IP Sockets, create a client-server program to make client sending the file name and the server to send back the contents of the requested file prom

Code

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("In Enter file name : ")

ClientSocket.Send (Sentence.encode())
file contents = ClientSocket.recv(1024).decode()
print ("In from Server : \n")
  print (file contents)
  ClientSocket.close()
```

Server TCP Py

```
from Socket import *
ServerName = '127.0.0.1'
ServerPort = 12000
ServerSocket = Socket (AF_INET, SOCK STREAM)
ServerSocket.bind ((ServerName, ServerPort))
ServerSocket.listen (1)
```

while 1:

print ("The server is ready to receive")
ConnectionSocket, addr = serverSocket.accept()
Sentence = connectionSocket.recv(1024).decode()

file = open(Sentence, "r")
l = file.read(1024)

ConnectionSocket.send(l.encode())
print ('\n Sent contents of' + Sentence)
file.close()
connectionSocket.close()

bind() - method bind to specific IP and port so that it can listen to incoming req on the IP

listn()

10/10

Client output

```
>>> The server is ready to receive
```

9/22

Server output

```
Enter the filename : Server TCP.py
from Server
/*
    Content of the file will be displayed here
*/
```

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:

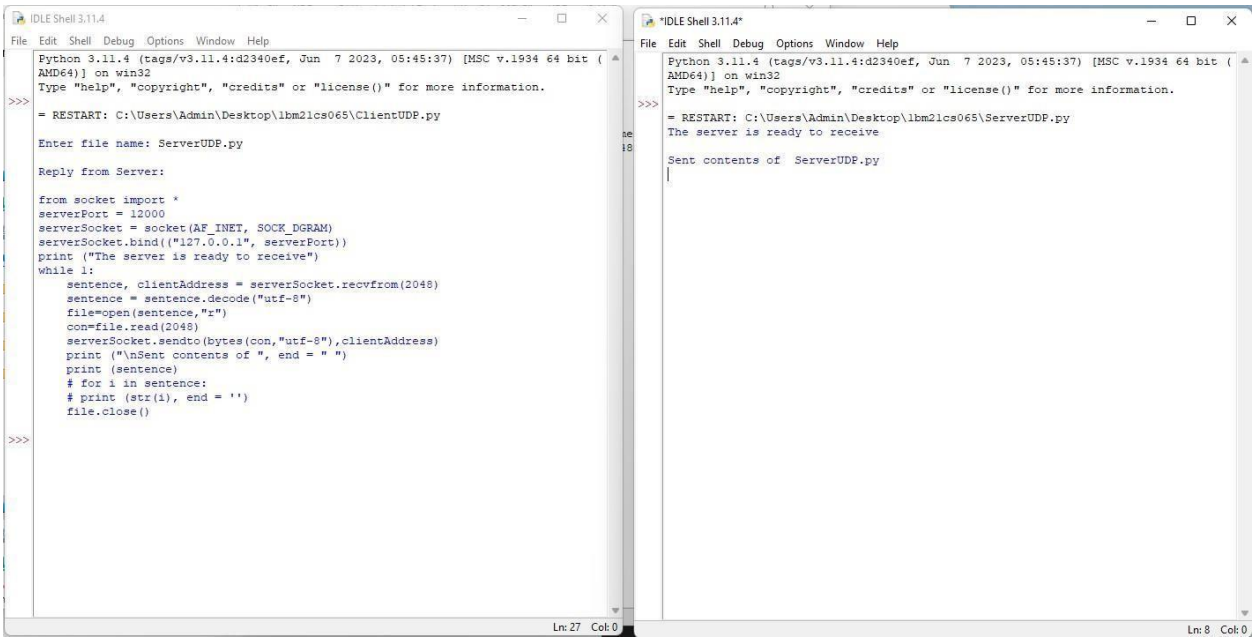
sentence, clientAddress = serverSocket.recvfrom(2048) sentence

= 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:



OBSERVATION:

Q. Using UDP sockets. Write a Client - Server program to make client sending the file name and the Server to send back the content 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))

fileContent, 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(("127.0.0.1", ServerPort))
print ("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 Content of', end = ' ')
    print(sentence)
        ## for i in sentence:
        # print(str(i), end = "")
            file.close()
```

output:

```
The Server is ready to receive
Sent Content of SeaverTCP.py
The Server is ready to receive
```

Server UDP output

```
Enter the file name: serverUDP.py
Reply from Server

/*
    content of the file
    displayed well
*/
```