VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

COURSE TITLE

Submitted by

AKSHANTH H M (1BM21CS014)

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 **AKSHANTH H M (1BM21CS014),** 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 lab - (22CS4PCCON)** work prescribed for the said degree.

Dr. Nandhini Vineeth
Assistant professor
Department of CSE
BMSCE, Bengaluru

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

`

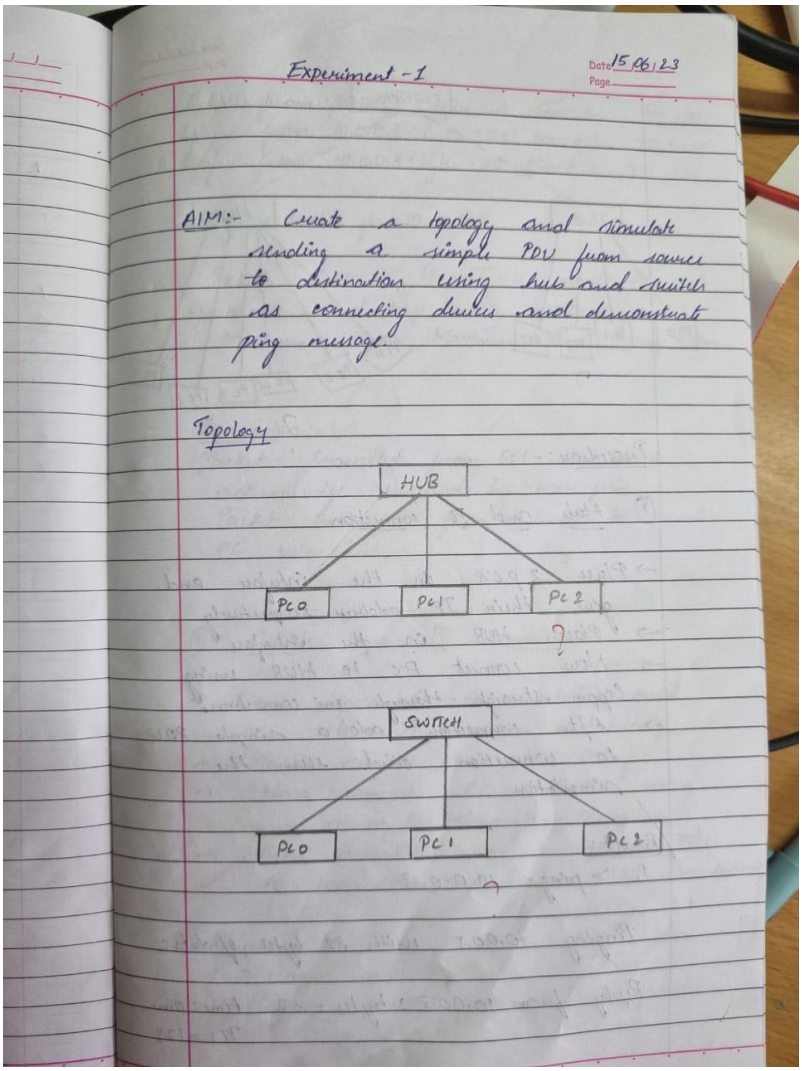
Index

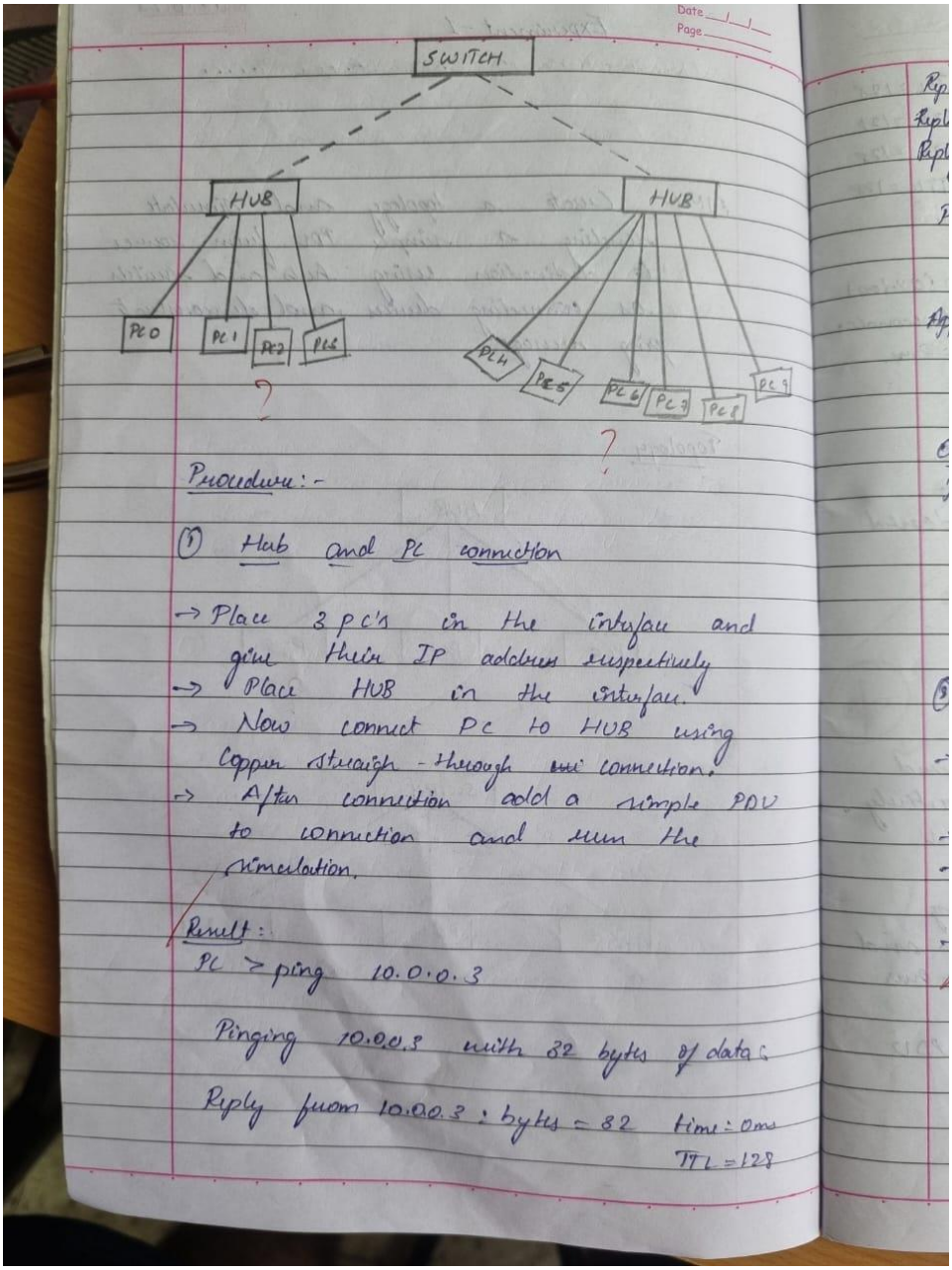
EXPERIMENT 1

AIM:

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:





Procedure:-

① Hub and PC connection

→ Place 3 pc's in the interface and give their IP address respectively
→ Place HUB in the interface.
→ Now connect PC to HUB using Copper Straight - through connection.
→ After connection add a simple PDU to connection and run the simulation.

Result:

PC > ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

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

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

Ping statistics for 10.0.0.3

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

Observation :-
Packet transmitted from PC1 to hub & acknowledge received by both pc's.
Packet transmitted from switch to under PC via hub.

② Switch and PC connection :-

→ Place 3 pc's in the interface and give their IP address respectively.
→ Place switch in the interface.
→ Now connect PC to SWITCH using copper straight through connection
→ After connection add a simple PDU to connection and run the simulation.

Result :-
PC > ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data :

Date / /
Page

Reply from 10.0.0.3 : bytes = 32 time = 1ms 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
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 time in milliseconds :
Minimum = 0ms, Maximum = 1ms, Average = 0ms.

Observation :-
Packet transmitted from source to destination without being broadcasted to all end devices.

③ Switch - Hub and PC connection :-

Procedure :-
→ Place the number of PC's required and give their IP address respectively.
→ Select two hubs and a switch and place it in the interface.
→ Now connect ~~switch~~ PC to hub using copper straight through connection and hub to switch using copper cross over connection.
→ After connection add a simple PDU and run the simulation

Result:-

PC > ping 10.0.0.11

Pinging 10.0.0.11 with 32 bits of data:

Reply from 10.0.0.11 : bytes = 32 time=3ms TTL=128
Reply from 10.0.0.11 : bytes = 32 time = 0ms TTL=128
Reply from 10.0.0.11 : bytes = 32 time = 1ms TTL=128
Reply from 10.0.0.11 : bytes = 32 time = 0ms TTL=128

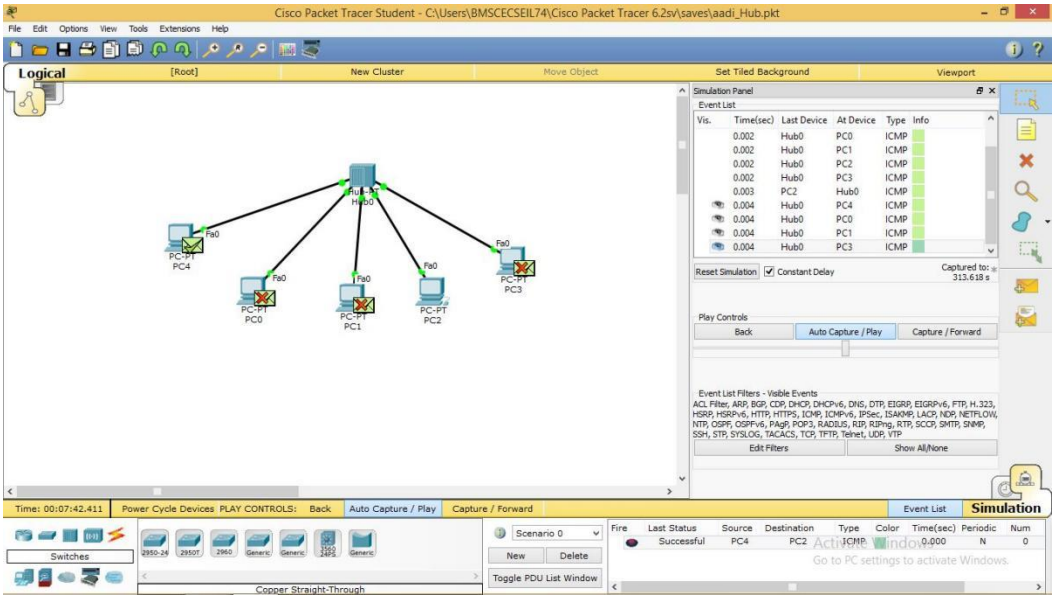
Ping statistics for 10.0.0.11:
Packets : Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milliseconds:
Minimum = 0ms, Maximum = 3ms, Average = 1ms

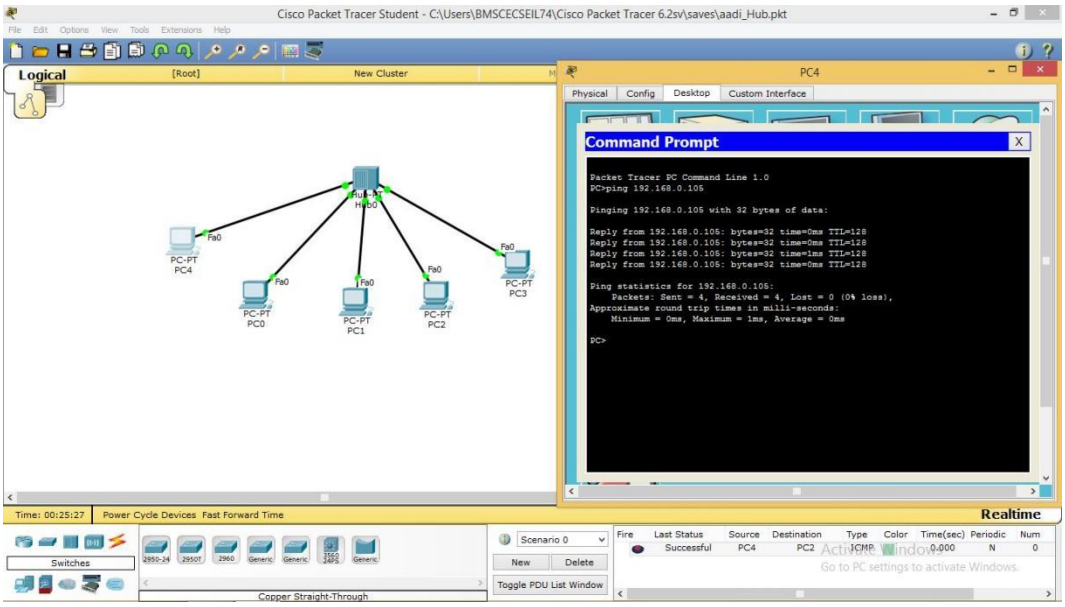
Observation:-

Packets transmitted from device to the destination by the hub, on both the hubs and in turn travels to switch and broadcast to all other devices.

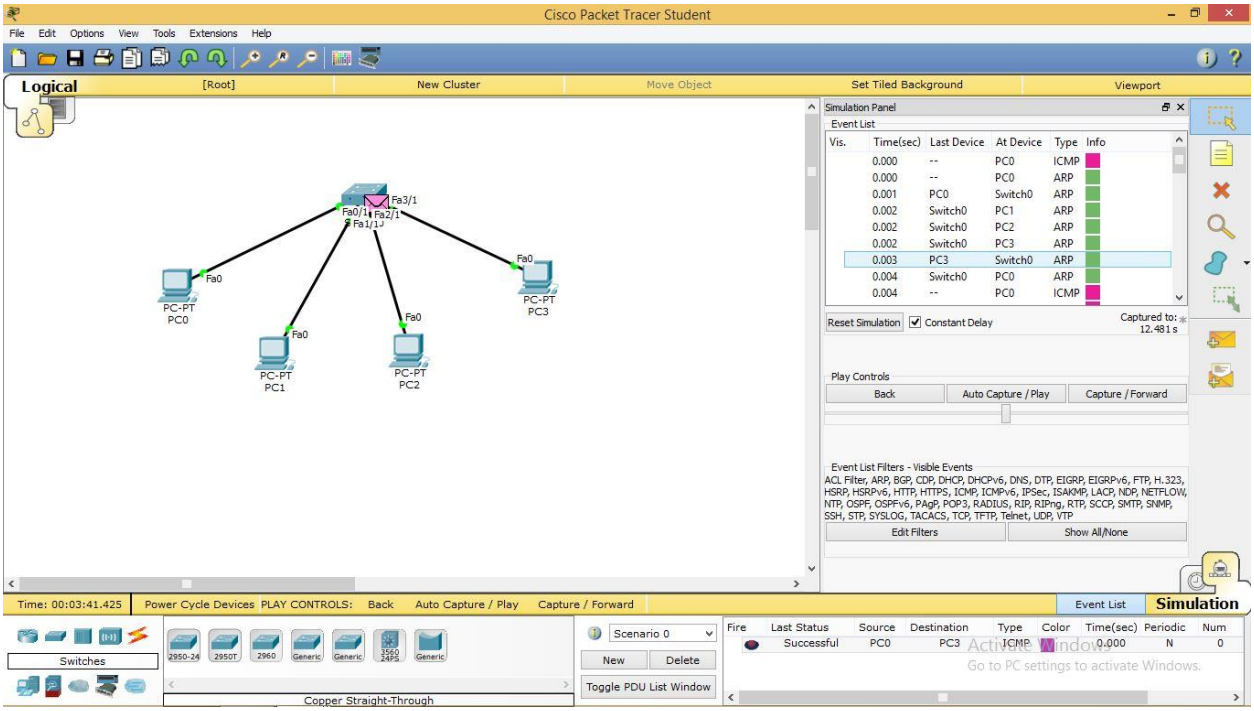
Result:

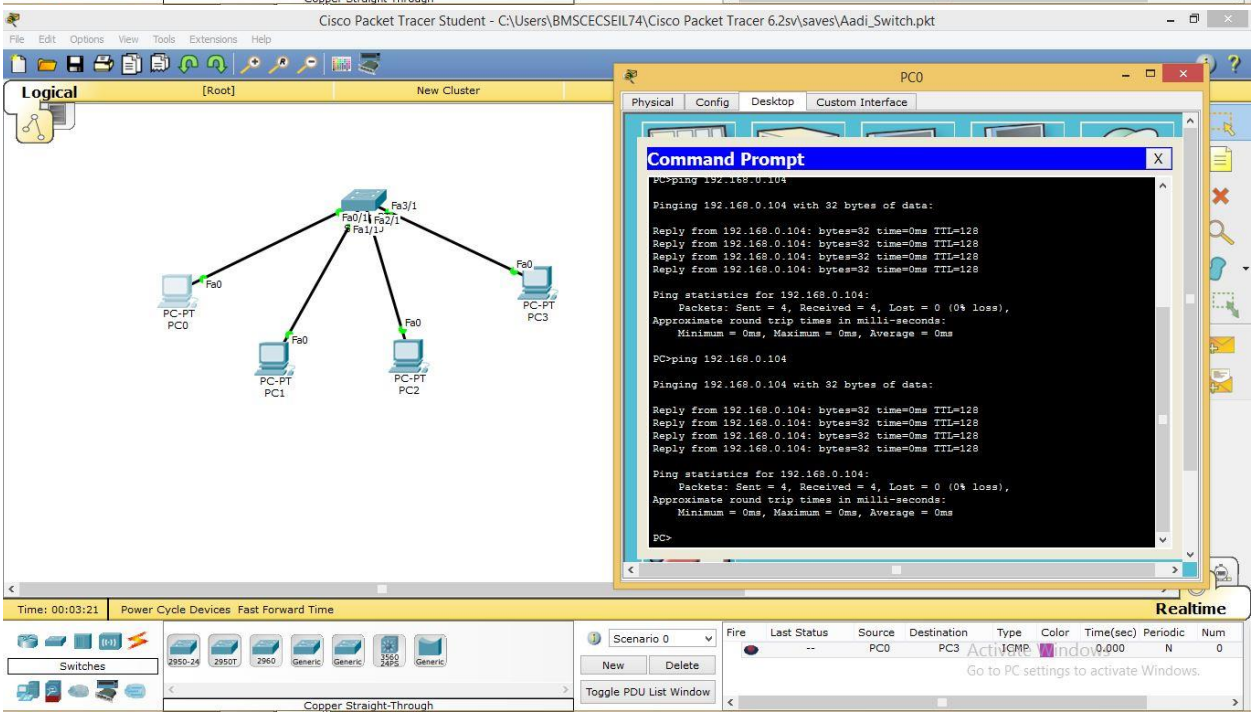
Hub:



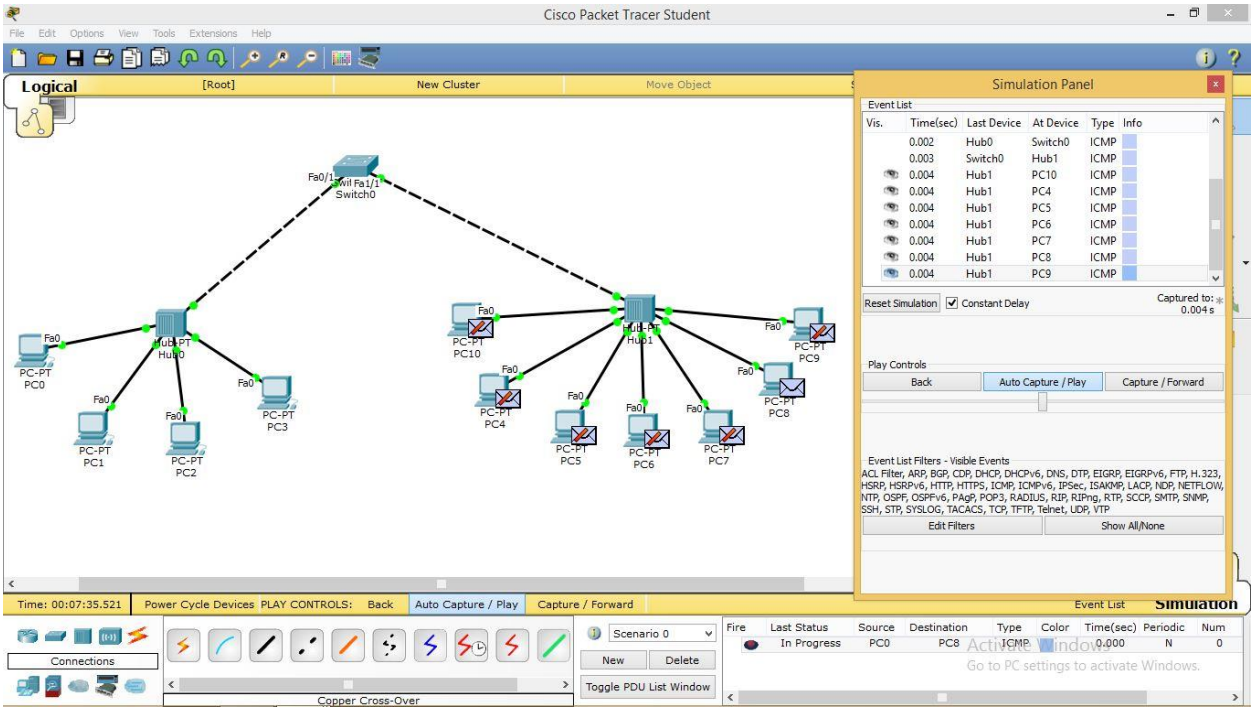


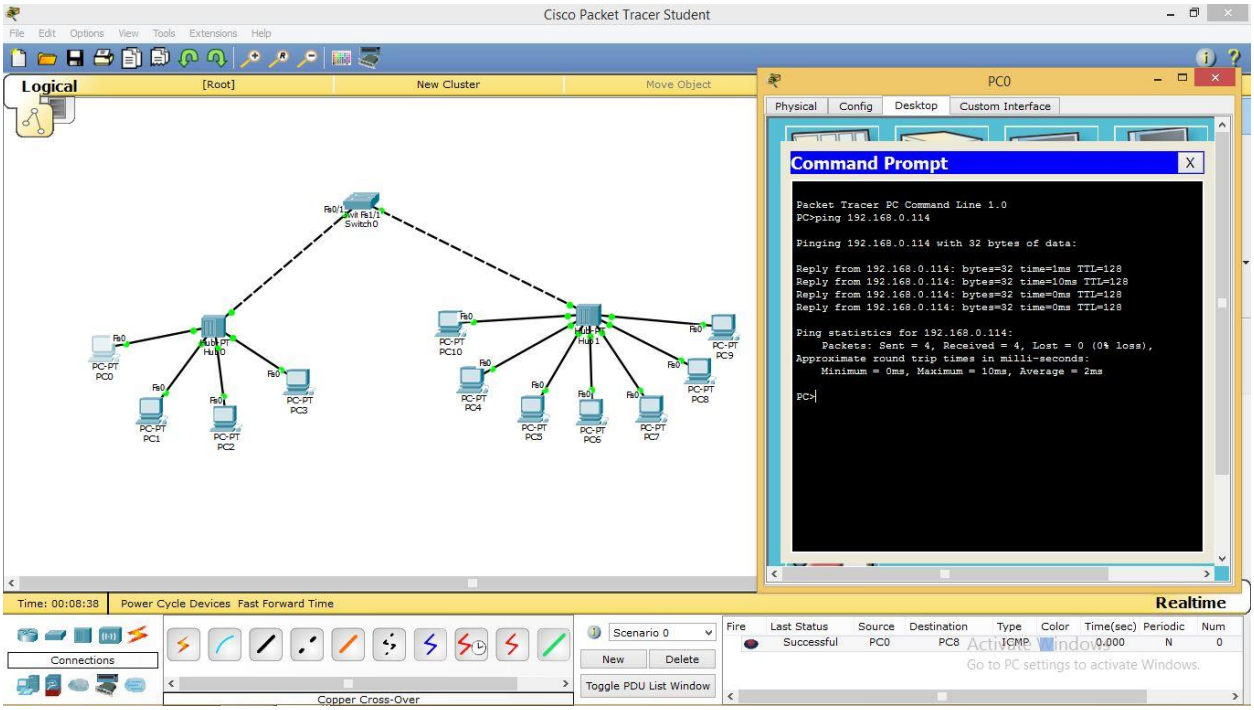
Switch:


Cisco Packet Tracer Student
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Simulation Panel
Event List
Vis. Time(sec) Last Device At Device Type Info
0.000 -- PC0 ICMP
0.000 -- PC0 ARP
0.001 PC0 Switch0 ARP
0.002 Switch0 PC1 ARP
0.002 Switch0 PC2 ARP
0.002 Switch0 PC3 ARP
0.003 PC3 Switch0 ARP
0.004 Switch0 PC0 ARP
0.004 -- PC0 ICMP
Reset Simulation
Constant Delay
Captured to: 12.481 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
Fa3/1
Fa0/1
Fa2/1
Fa1/1
Fa0
PC-PT PC0
PC-PT PC1
PC-PT PC2
PC-PT PC3
Time: 00:03:41.425
Power Cycle Devices PLAY CONTROLS: Back Auto Capture / Play Capture / Forward
Event List
Simulation
Switches
Copper Straight-Through
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(sec) Periodic Num
Successful PC0 PC3 ICMP 0.000 N 0
Activate Windows
Go to PC settings to activate Windows.


Cisco Packet Tracer Student - C:\Users\BMSCECSEIL74\Cisco Packet Tracer 6.2sv\saves\Aadi_Switch.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
PC>ping 192.168.0.104
Pinging 192.168.0.104 with 32 bytes of data:
Reply from 192.168.0.104: bytes=32 time=0ms TTL=128
Reply from 192.168.0.104: bytes=32 time=0ms TTL=128
Reply from 192.168.0.104: bytes=32 time=0ms TTL=128
Reply from 192.168.0.104: bytes=32 time=0ms TTL=128
Ping statistics for 192.168.0.104:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>ping 192.168.0.104
Pinging 192.168.0.104 with 32 bytes of data:
Reply from 192.168.0.104: bytes=32 time=0ms TTL=128
Reply from 192.168.0.104: bytes=32 time=0ms TTL=128
Reply from 192.168.0.104: bytes=32 time=0ms TTL=128
Reply from 192.168.0.104: bytes=32 time=0ms TTL=128
Ping statistics for 192.168.0.104:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>
Fa3/1
Fa0/1
Fa2/1
Fa1/1
Fa0
PC-PT PC0
PC-PT PC1
PC-PT PC2
PC-PT PC3
Time: 00:03:21
Power Cycle Devices Fast Forward Time
Realtime
Switches
Copper Straight-Through
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(sec) Periodic Num
-- PC0 PC3 ICMP 0.000 N 0
Activate Windows
Go to PC settings to activate Windows.

Hybrid:


Cisco Packet Tracer Student
Logical
[Root]
New Cluster
Move Object
Simulation Panel
Event List
Vis. Time(sec) Last Device At Device Type Info
0.002 Hub0 Switch0 ICMP
0.003 Switch0 Hub1 ICMP
0.004 Hub1 PC10 ICMP
0.004 Hub1 PC4 ICMP
0.004 Hub1 PC5 ICMP
0.004 Hub1 PC6 ICMP
0.004 Hub1 PC7 ICMP
0.004 Hub1 PC8 ICMP
0.004 Hub1 PC9 ICMP
Reset Simulation
Constant Delay
Captured to: 0.004 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
Switch0
Hub0
Hub1
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
PC8
PC9
PC10
Time: 00:07:35.521
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
In Progress PC0 PC8 ICMP 0.000 N 0
Activate Windows
Go to PC settings to activate Windows.

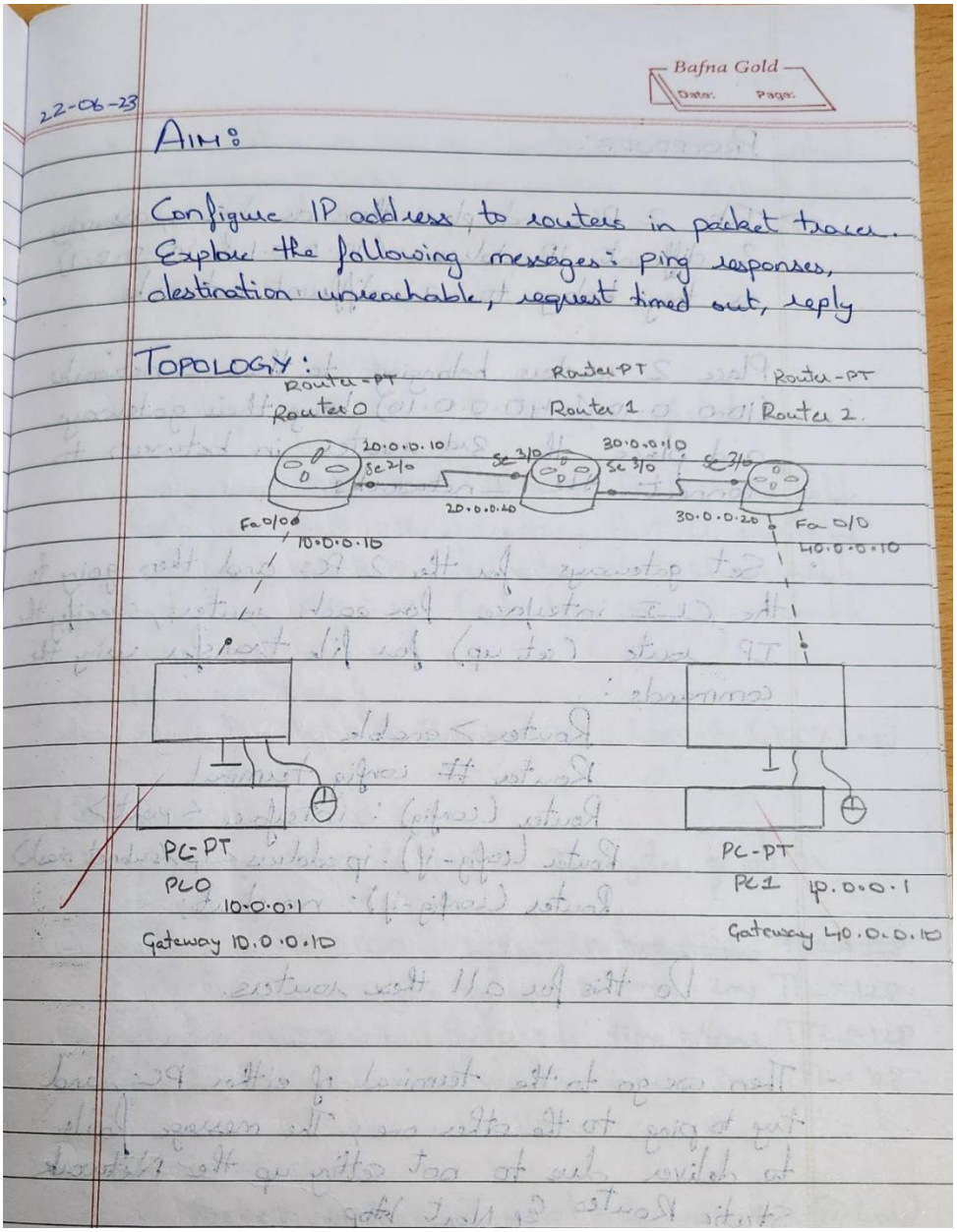

Cisco Packet Tracer Student
Logical
[Root]
New Cluster
Move Object
PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 192.168.0.114
Pinging 192.168.0.114 with 32 bytes of data:
Reply from 192.168.0.114: bytes=32 time=1ms TTL=128
Reply from 192.168.0.114: bytes=32 time=10ms TTL=128
Reply from 192.168.0.114: bytes=32 time=0ms TTL=128
Reply from 192.168.0.114: bytes=32 time=0ms TTL=128
Ping statistics for 192.168.0.114:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 10ms, Average = 2ms
PC>
Switch0
Hub0
Hub1
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
PC8
PC9
PC10
Time: 00:08:38
Power Cycle Devices Fast Forward Time
Realtime
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 PC8 ICMP 0.000 N 0
Activate Windows
Go to PC settings to activate Windows.

EXPERIMENT 2

AIM:

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

OBSERVATION:



PROCEDURE:

⇒ Take 2 PCs and place them as shown, assuming 2 different IP addresses (10.0.0.1 & 40.0.0.1) as they belong to 2 different networks.

- Place 2 routers belonging to these 2 networks (10.0.0.10 & 40.0.0.10) being their gateways and place the 3rd router in between to connect the 2 networks.

- Set gateways for the 2 PCs and then going to the CLI interface for each routers, specify the IP route (set up) for file transfer using the commands:

```
Router > enable
Router # config terminal
Router (config): interface <port>
Router (config-if): ip address <ip><subnet mask>
Router (config-if): no shut;
```

Do this for all these routers.

- Then we go to the terminal of either PCs, and try to ping to the other one; The message fails to deliver due to not setting up the Network Static Routes Eg Next Hop.

- We again go to CLI of each router and setup the "next hop" using the command:

```
> ip route <network-id> <mask> <next-hop>
```

eg: > ip route 40.0.0.0 255.0.0.0 20.0.0.20
(for router 1)

- This is done so that the router recognises which pathway to take when packet is received for a particular destination.

RESULT:

(i) > 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.
Reply from 10.0.0.10: Destination Host Unreachable

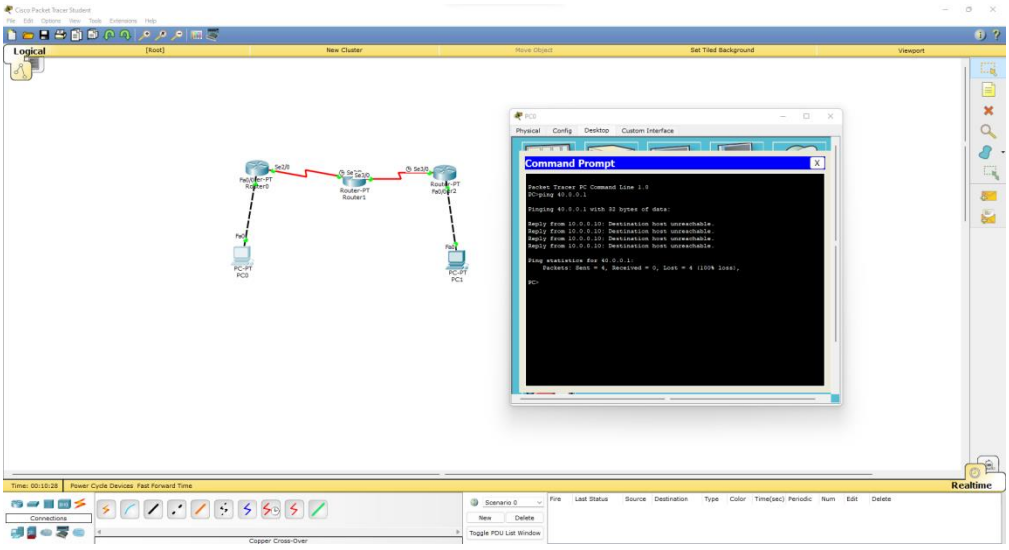
Ping statistics:
Packets: sent = 4; Received = 0; Loss = 4 (100% Loss)

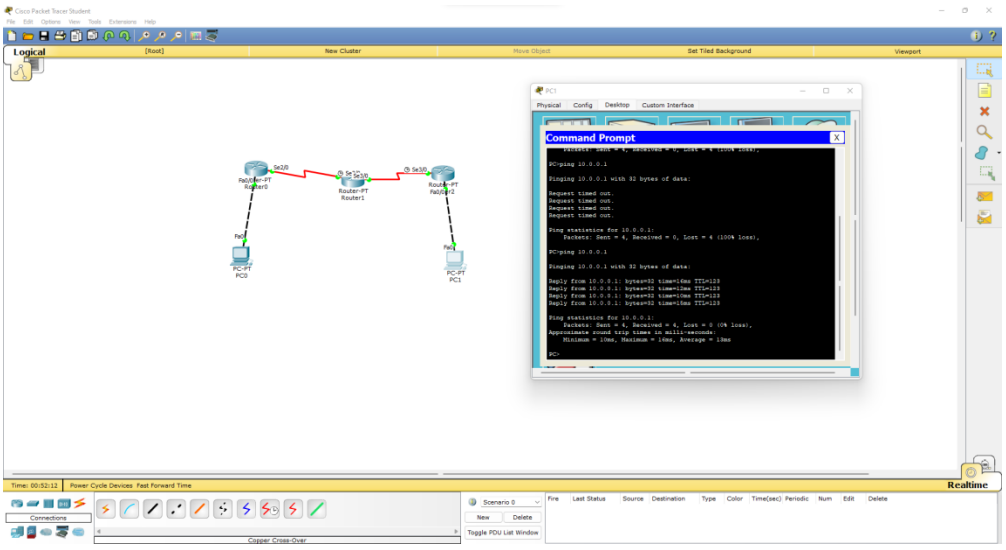
(ii) >ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data
Request Timed Out
~~Reply from 40.0.0.1 : Bytes=32 time=2ms TTL=128~~
Reply from 40.0.0.1 : Bytes = 32 time = 2ms TTL = 128
Reply from 40.0.0.1 : Bytes = 32 time = 4ms TTL = 128
Reply from 40.0.0.1 : Bytes = 32 time = 3ms TTL = 128

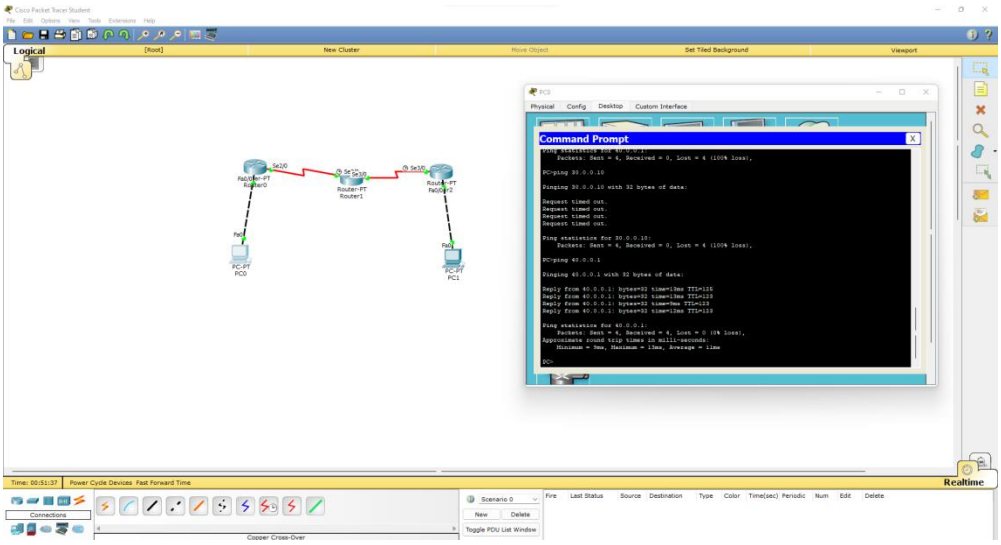
Ping statistics
Packets : sent = 4, Received = 1, Loss = 3 (75% Loss)
Approx time in ms:

minimum = 2ms Maximum = 4ms average = 3ms

Result:


Logical
Command Prompt
Realtime


Logical
Command Prompt
Realtime

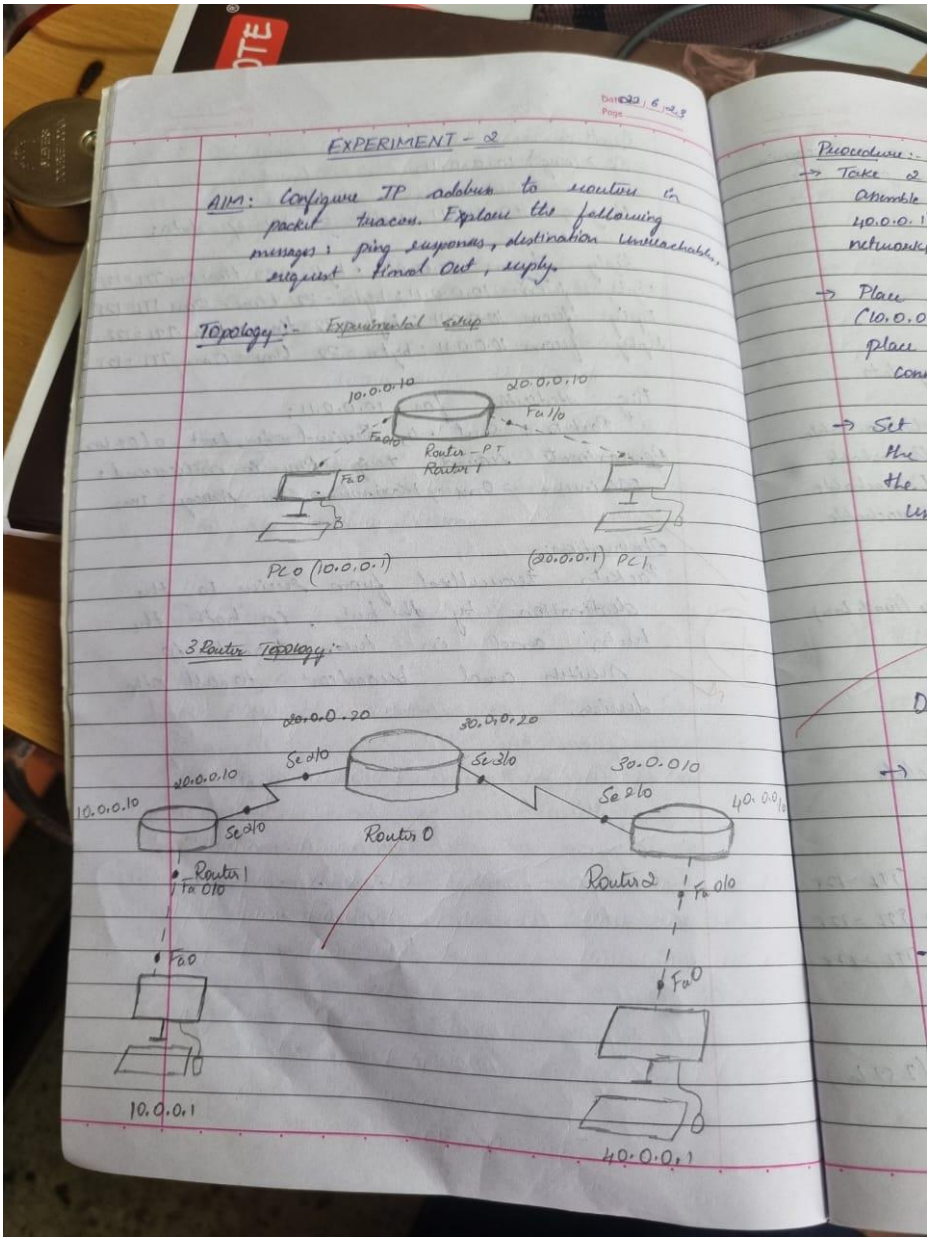

Logical
Command Prompt
Realtime

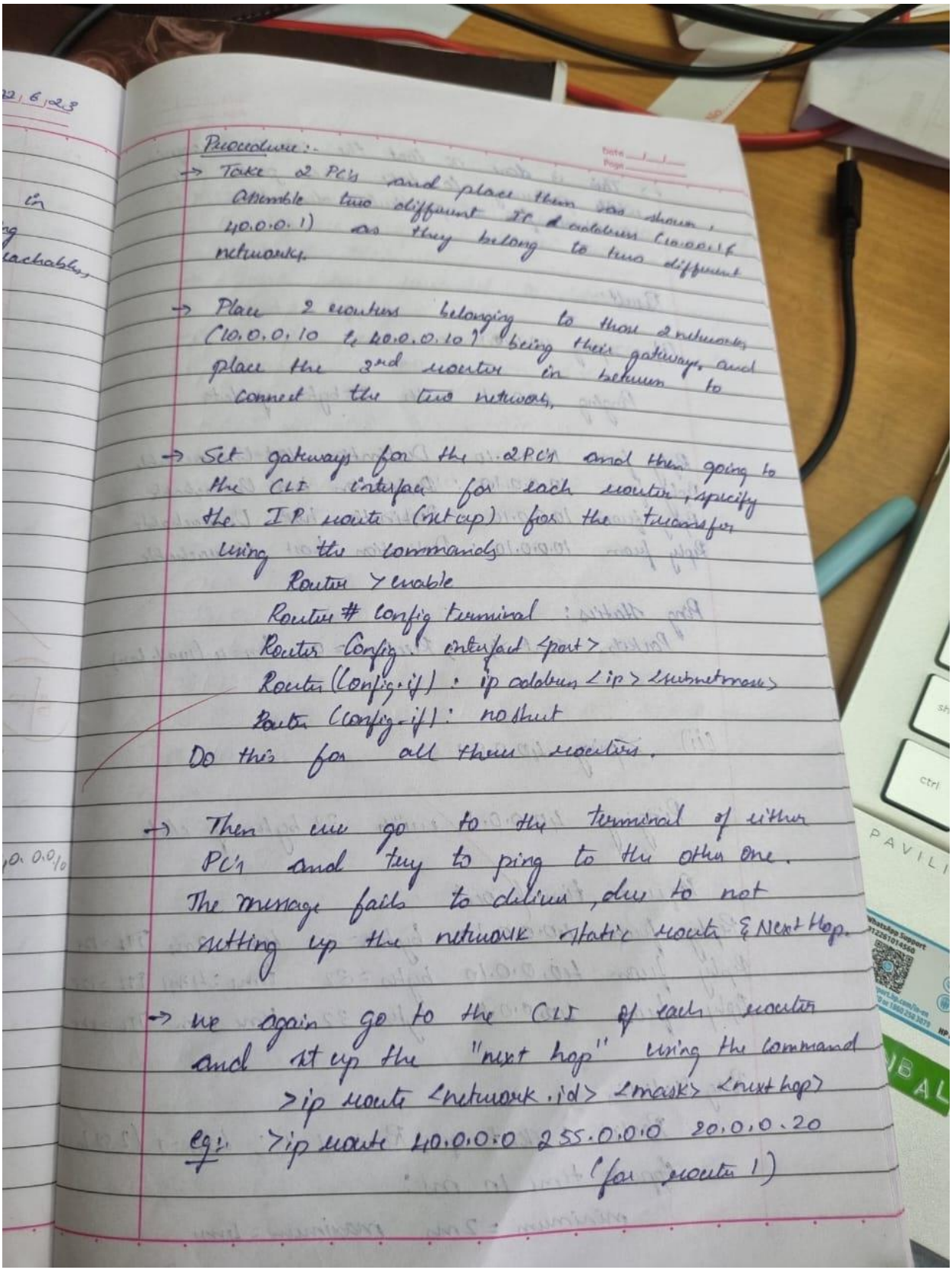
EXPERIMENT 3

AIM:

Configure default route, static route to the Router.

OBSERVATION:





Procedure:-

→ Take 2 PC's and place them as shown, Assemble two different IP address (10.0.0.1 & 40.0.0.1) as they belong to two different networks.

→ Place 2 routers belonging to those 2 networks (10.0.0.10 & 40.0.0.10) being their gateways, and place the 3rd router in between to connect the two networks.

→ Set gateways for the 2 PC's and then going to the CLI interface for each router, specify the IP route (next hop) for the transfer using the commands

Router > enable
Router # config terminal
Router Config: interface <port>
Router (config-if): ip address <ip> <subnetmask>
Router (config-if): no shut

Do this for all three routers.

→ Then we go to the terminal of either PC's and try to ping to the other one. The message fails to deliver, due to not setting up the network static route & Next Hop.

→ We again go to the CLI of each router and set up the "next hop" using the command

>ip route <network id> <mask> <next hop>

eg: >ip route 40.0.0.0 255.0.0.0 20.0.0.20 (for router 1)

→ This is done so that the router recognizes which pathway to take when packet is received for a particular destination.

Result:

(i) > 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
Reply from 10.0.0.10 : Destination host Unreachable

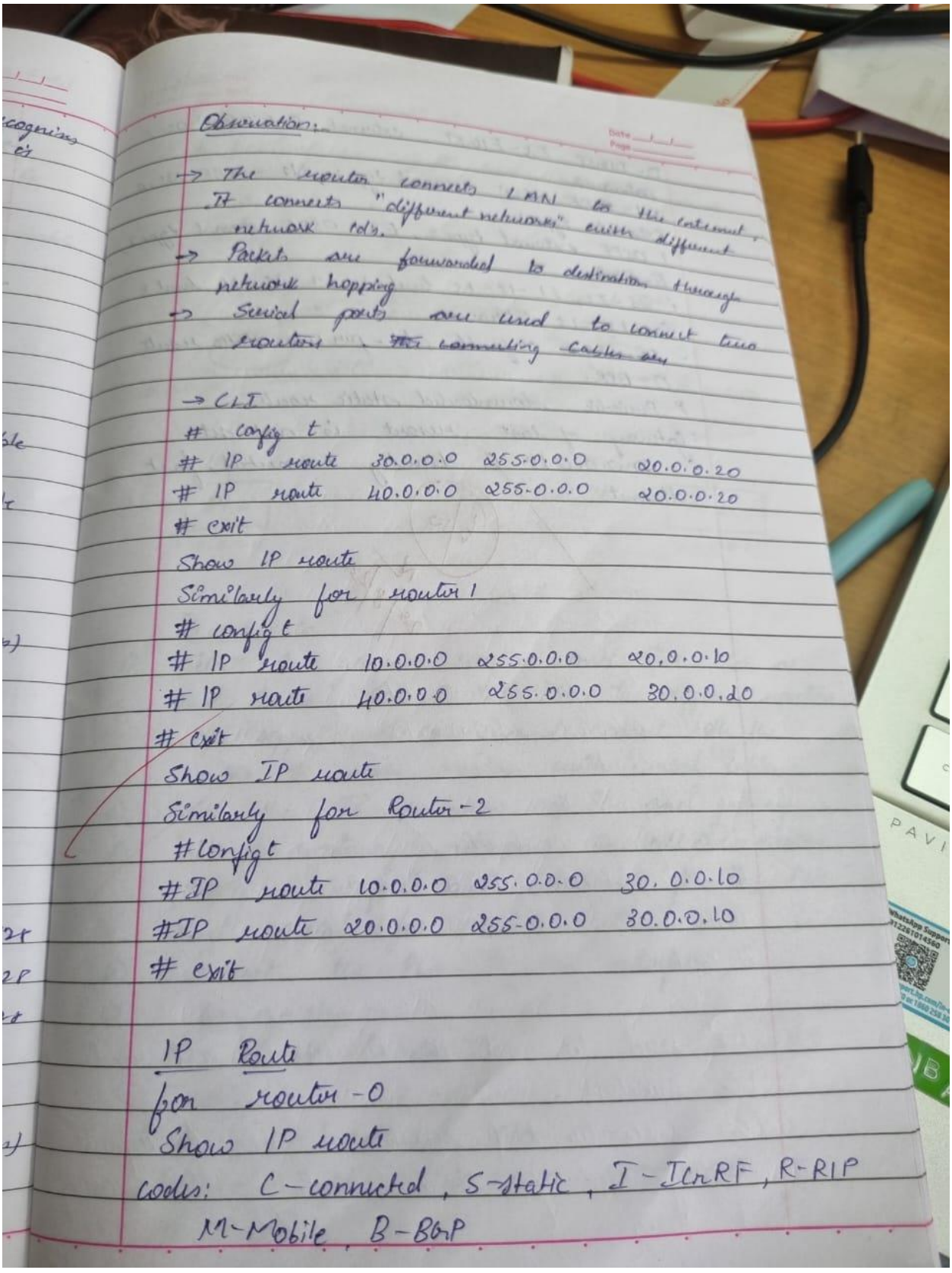
Ping Statics:
Packets: Sent = 4 ; Received = 0; Lost = 4 (100% Loss)

(ii) > ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data

Request timed out
Reply from 40.0.0.10 bytes = 32 time: 2ms TTL = 128
Reply from 40.0.0.10 bytes = 32 time: 4ms TTL = 128
Reply from 40.0.0.10 bytes = 32 time: 3ms TTL = 128

Ping Statistics
Packets : Sent = 4, Received = 1 ; Lost = 1 (25% loss)
Approx time in ms:
minimum = 2ms maximum = 4ms
average = 3ms



Observation:

→ The router connects LAN to the internet. It connects "different networks" with different network ids.
→ Packets are forwarded to destination through network hopping
→ Serial ports are used to connect two routers. The connecting cables are

→ CLI
config t
IP route 30.0.0.0 255.0.0.0 20.0.0.20
IP route 40.0.0.0 255.0.0.0 20.0.0.20
exit
Show IP route
Similarly for router 1
config t
IP route 10.0.0.0 255.0.0.0 20.0.0.10
IP route 40.0.0.0 255.0.0.0 30.0.0.20
exit
Show IP route
Similarly for Router-2
config t
IP route 10.0.0.0 255.0.0.0 30.0.0.10
IP route 20.0.0.0 255.0.0.0 30.0.0.10
exit

IP Route
for router-0
Show IP route
codes: C-connected, S-static, I-IGRF, R-RIP
M-Mobile, B-BGP

D - DIGRP, EX - EIGRT external, O - OSPF, A - OSPF interarea
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
1 - OSPF external type 2, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level 2
ia - IS-IS interarea
* - candidate default, U - per user static route
O - OER
P - Periodic downloaded static route
Gateway of last resort is not set
C - 10.0.0.0/8 is directly connected, fast ethernet

10/10
N
28/8/23

7/11/23

Aim:- Configu... connection

Topology

10.0.0.1

10.0.0.10

PC0

10.0.0

Proc...

1). Sele... end... with... no...

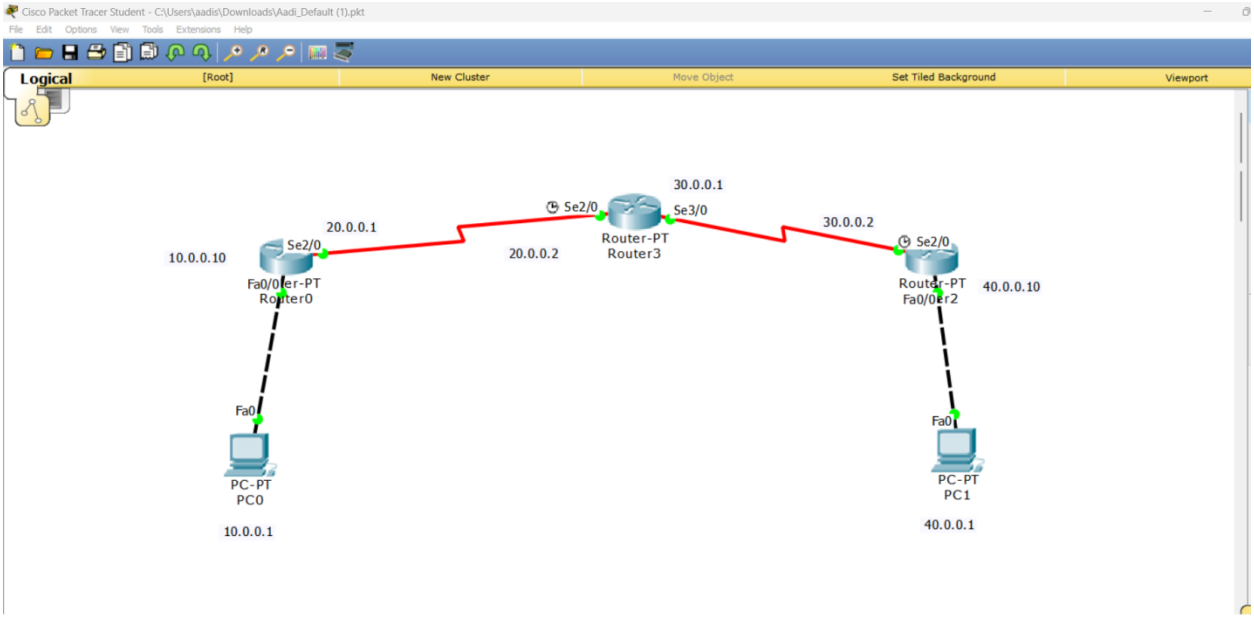
2).

3).

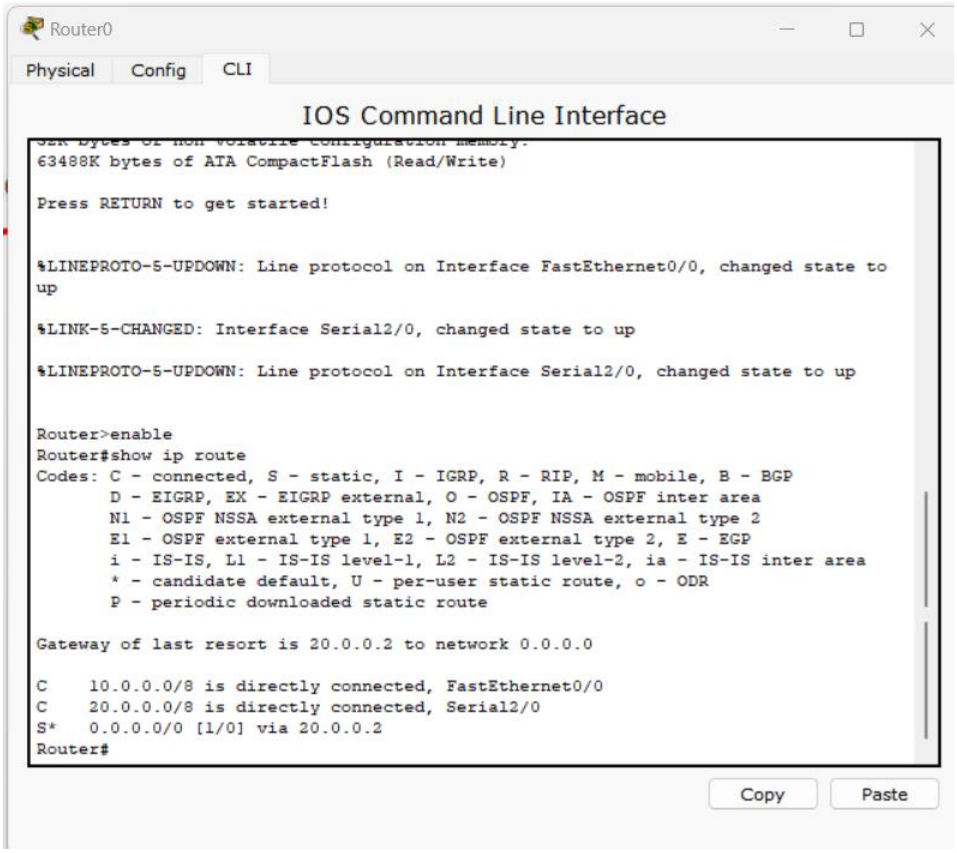
4).

5).

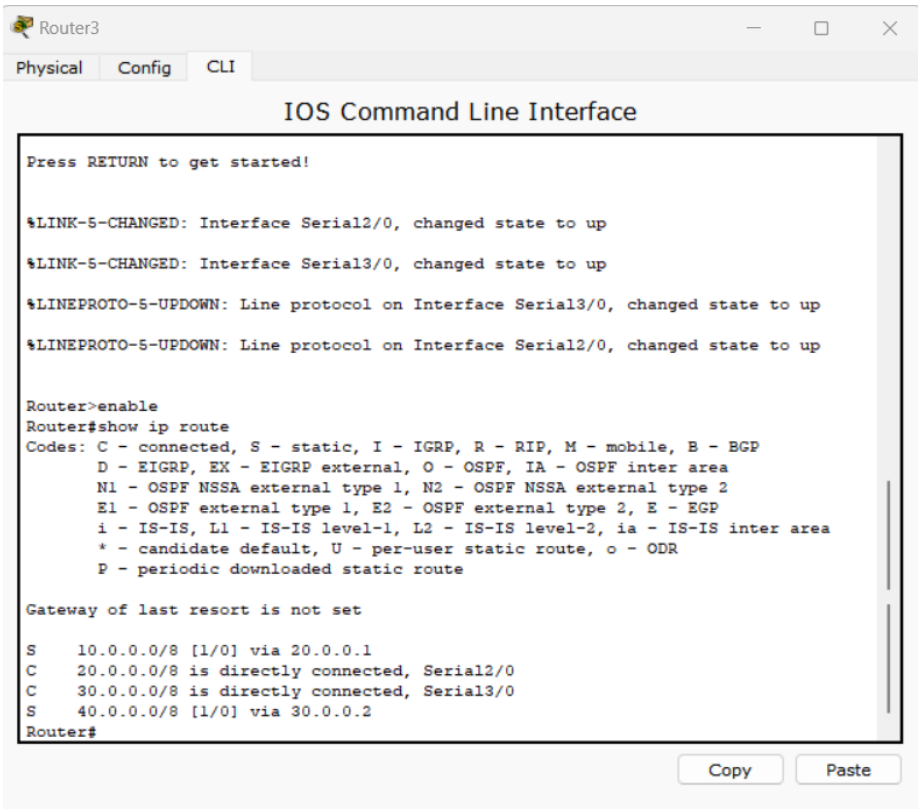
Result:



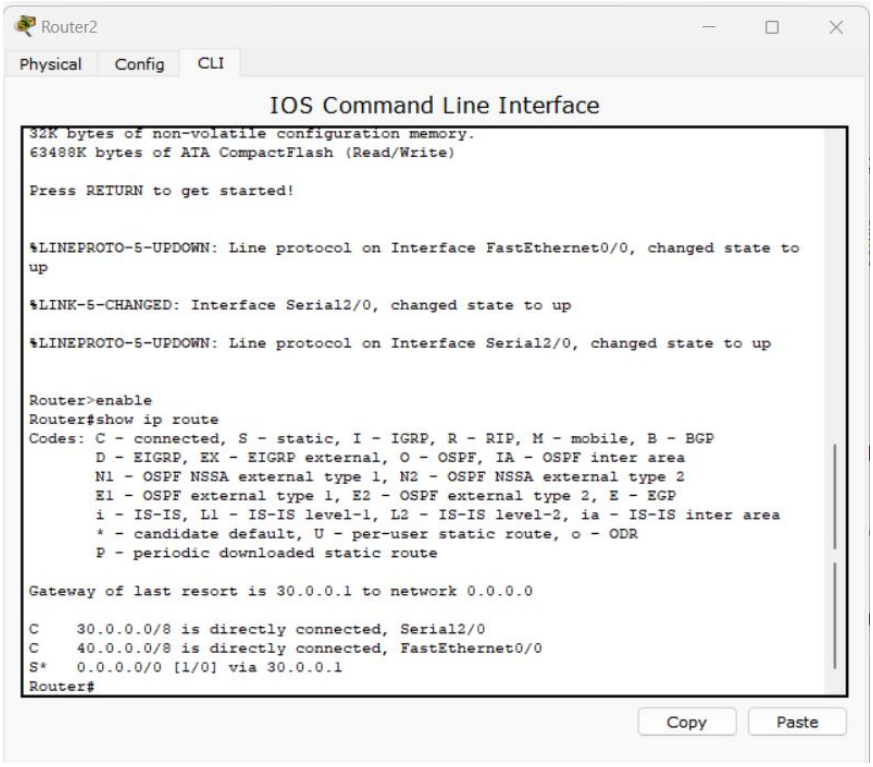
Router 0:

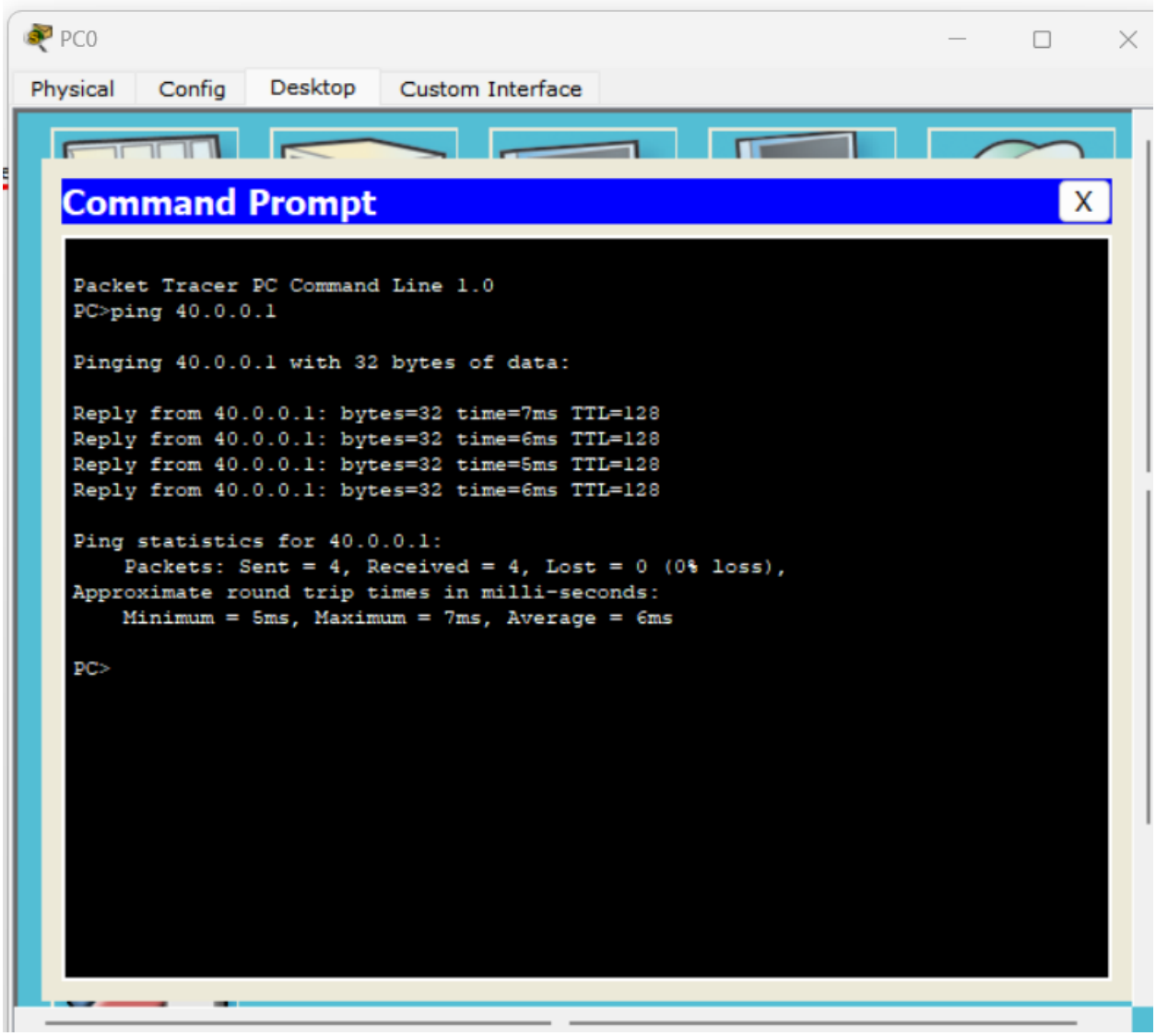

Router0
Physical
Config
CLI
IOS Command Line Interface
63488K bytes of ATA CompactFlash (Read/Write)
Press RETURN to get started!
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
Router>enable
Router#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 20.0.0.2 to network 0.0.0.0
C 10.0.0.0/8 is directly connected, FastEthernet0/0
C 20.0.0.0/8 is directly connected, Serial2/0
S* 0.0.0.0/0 [1/0] via 20.0.0.2
Router#
Copy
Paste

Router 1:


Router3
Physical
Config
CLI
IOS Command Line Interface
Press RETURN to get started!
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
%LINK-5-CHANGED: Interface Serial3/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
Router>enable
Router#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
S 10.0.0.0/8 [1/0] via 20.0.0.1
C 20.0.0.0/8 is directly connected, Serial2/0
C 30.0.0.0/8 is directly connected, Serial3/0
S 40.0.0.0/8 [1/0] via 30.0.0.2
Router#
Copy
Paste

Router 2:


Router2
Physical
Config
CLI
IOS Command Line Interface
32K bytes of non-volatile configuration memory.
63488K bytes of ATA CompactFlash (Read/Write)
Press RETURN to get started!
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
Router>enable
Router#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 30.0.0.1 to network 0.0.0.0
C 30.0.0.0/8 is directly connected, Serial2/0
C 40.0.0.0/8 is directly connected, FastEthernet0/0
S* 0.0.0.0/0 [1/0] via 30.0.0.1
Router#
Copy
Paste

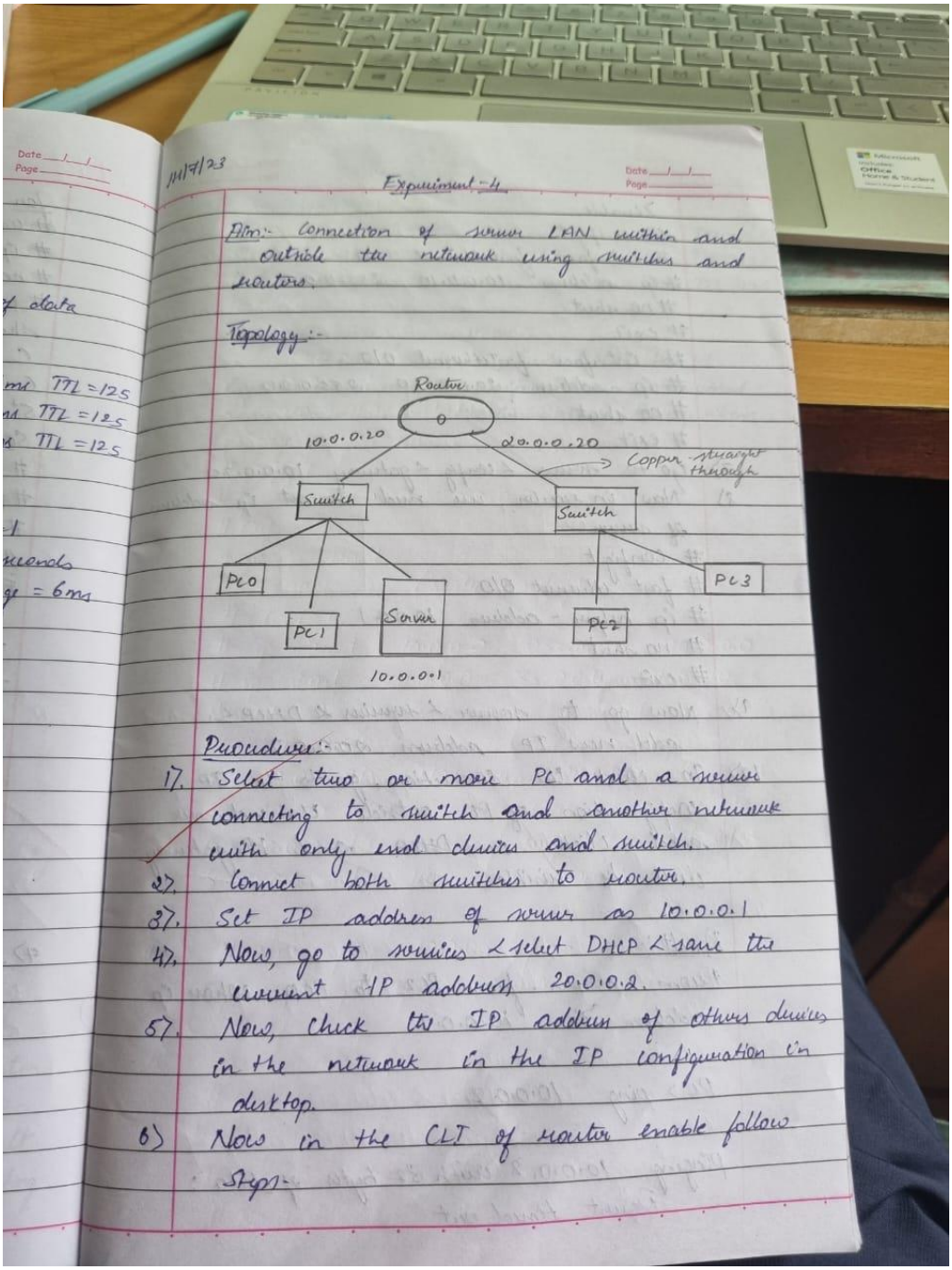

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Reply from 40.0.0.1: bytes=32 time=7ms TTL=128
Reply from 40.0.0.1: bytes=32 time=6ms TTL=128
Reply from 40.0.0.1: bytes=32 time=5ms TTL=128
Reply from 40.0.0.1: bytes=32 time=6ms TTL=128
Ping statistics for 40.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 5ms, Maximum = 7ms, Average = 6ms
PC>

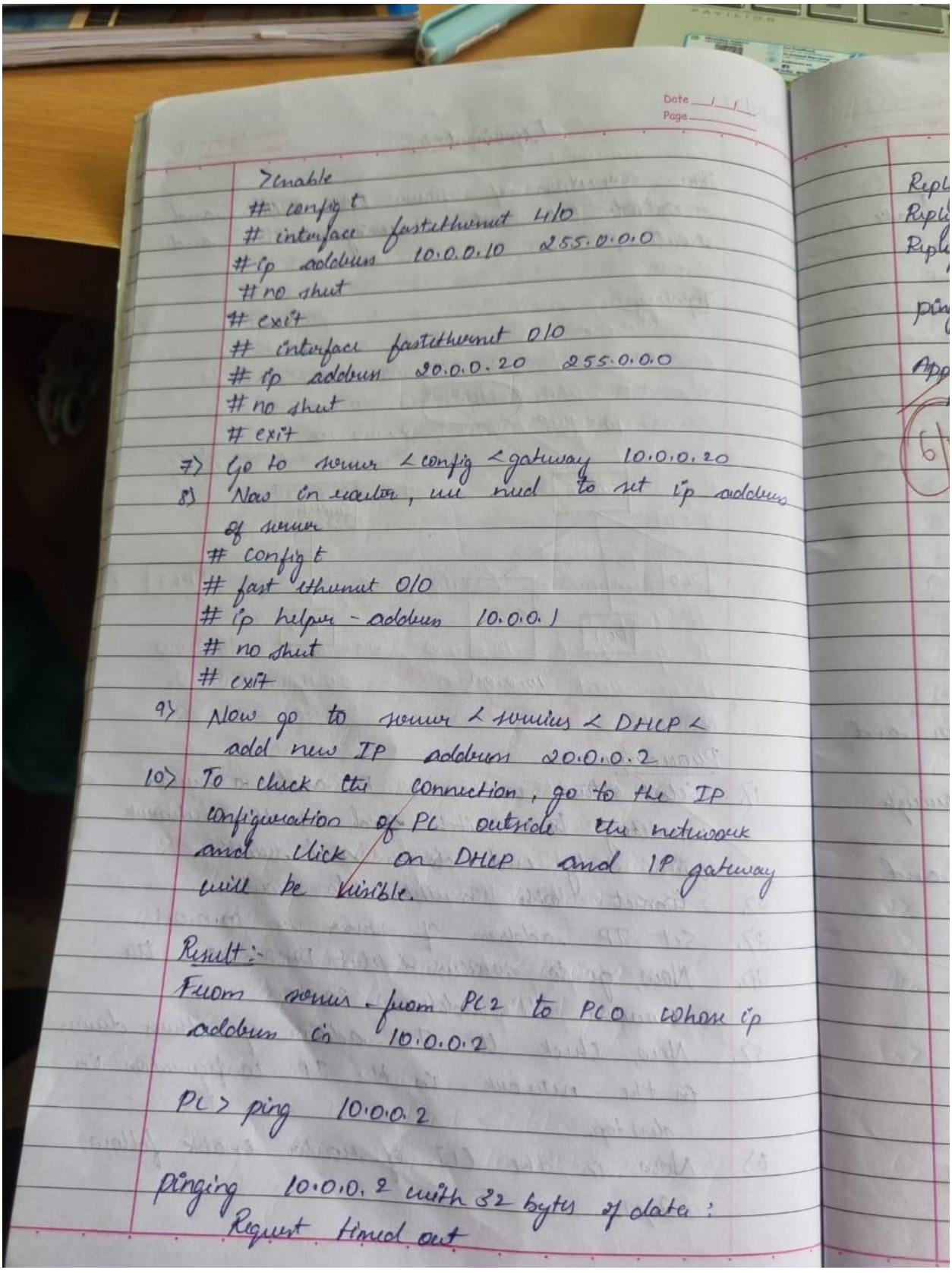
EXPERIMENT 4

AIM:

Configure DHCP within a LAN and outside LAN.

OBSERVATION:





> enable
config t
interface fastethernet 4/0
ip address 10.0.0.10 255.0.0.0
no shut
exit
interface fastethernet 0/0
ip address 20.0.0.20 255.0.0.0
no shut
exit

7) Go to server > config > gateway 10.0.0.20

8) Now in router, we need to set ip address of server
config t
fast ethernet 0/0
ip helper - address 10.0.0.1
no shut
exit

9) Now go to server > service > DHCP > add new IP address 20.0.0.2

10) To check the connection, go to the IP configuration of PC outside the network and click on DHCP and IP gateway will be visible.

Result:-
From server, from PC2 to PC0 whose ip address is 10.0.0.2

PC > ping 10.0.0.2

pinging 10.0.0.2 with 32 bytes of data:
Request timed out

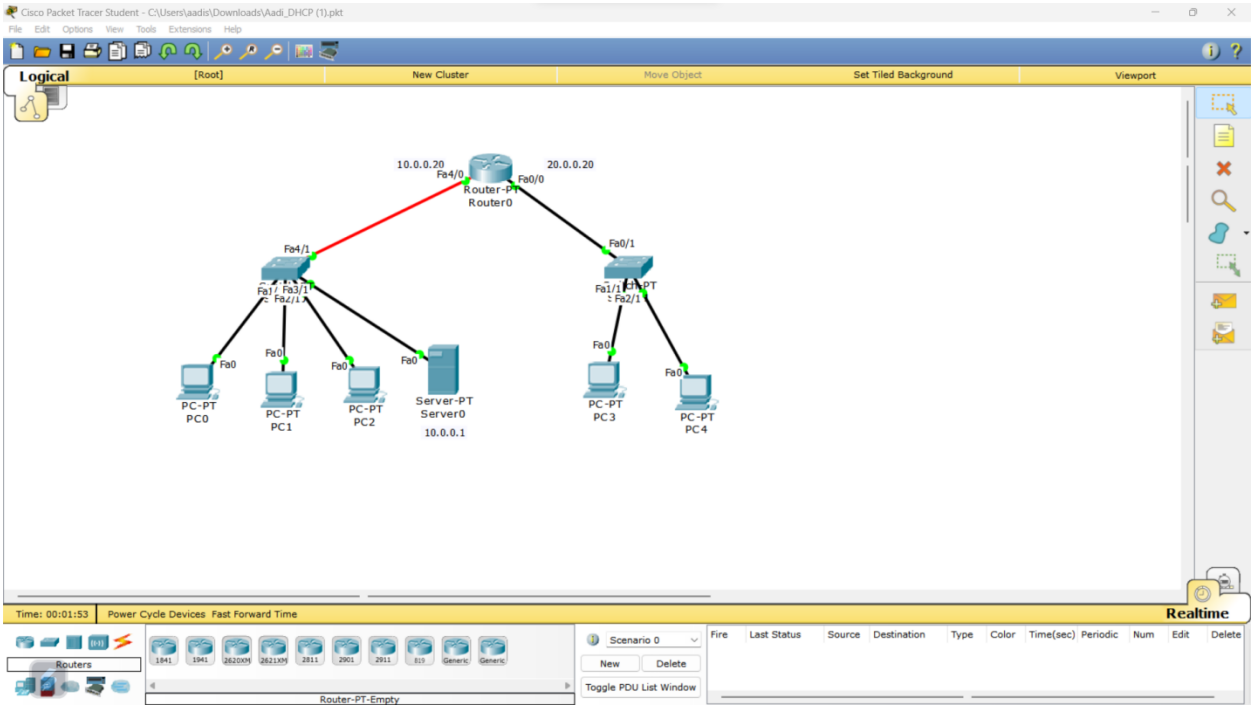
Reply from 10.0.0.2 : bytes = 32 time = 6ms TTL = 125
Reply from 10.0.0.2 : bytes = 32 time = 2ms TTL = 125
Reply from 10.0.0.2 : bytes = 32 time = 12ms TTL = 125

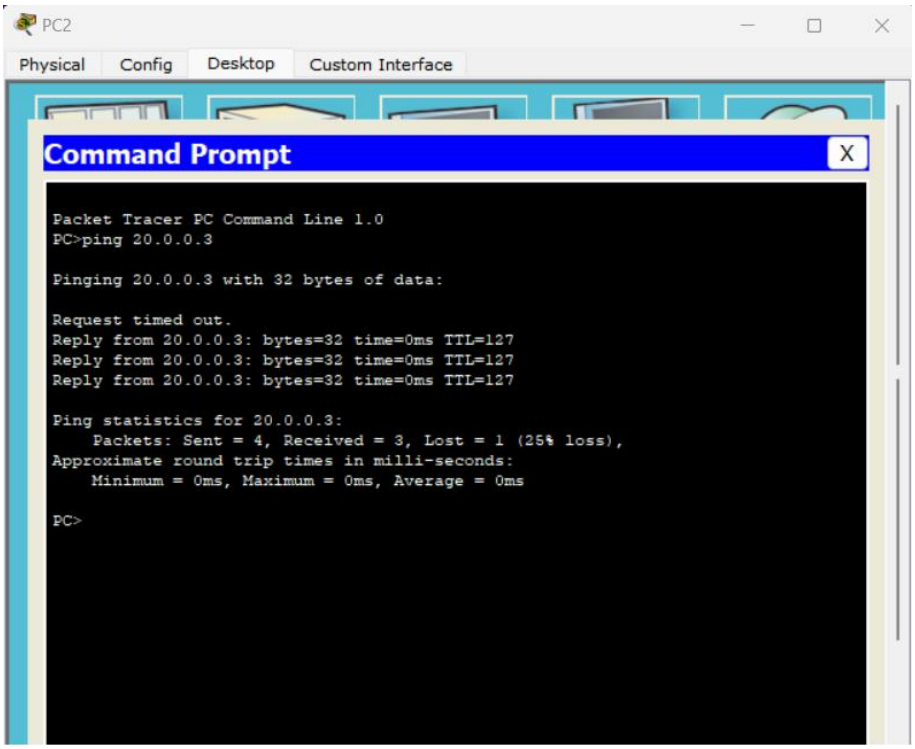
ping statistics for 10.0.0.2
Packets : Sent = 4, Rcvd = 3, Lost = 1
Approximate round trip time in millisecond
minimum = 2ms, Maximum = 12ms, Average = 6ms

6/10

NL
28/8/23

Result:


Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
10.0.0.20
Fa4/0
Router-PT
Router0
Fa0/0
20.0.0.20
Fa4/1
Fa0/1
Fa0
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
Server-PT
Server0
10.0.0.1
PC-PT
PC3
PC-PT
PC4
Time: 00:01:53
Power Cycle Devices
Fast Forward Time
Realtime
Routers
Router-PT-Empty
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Periodic
Num
Edit
Delete

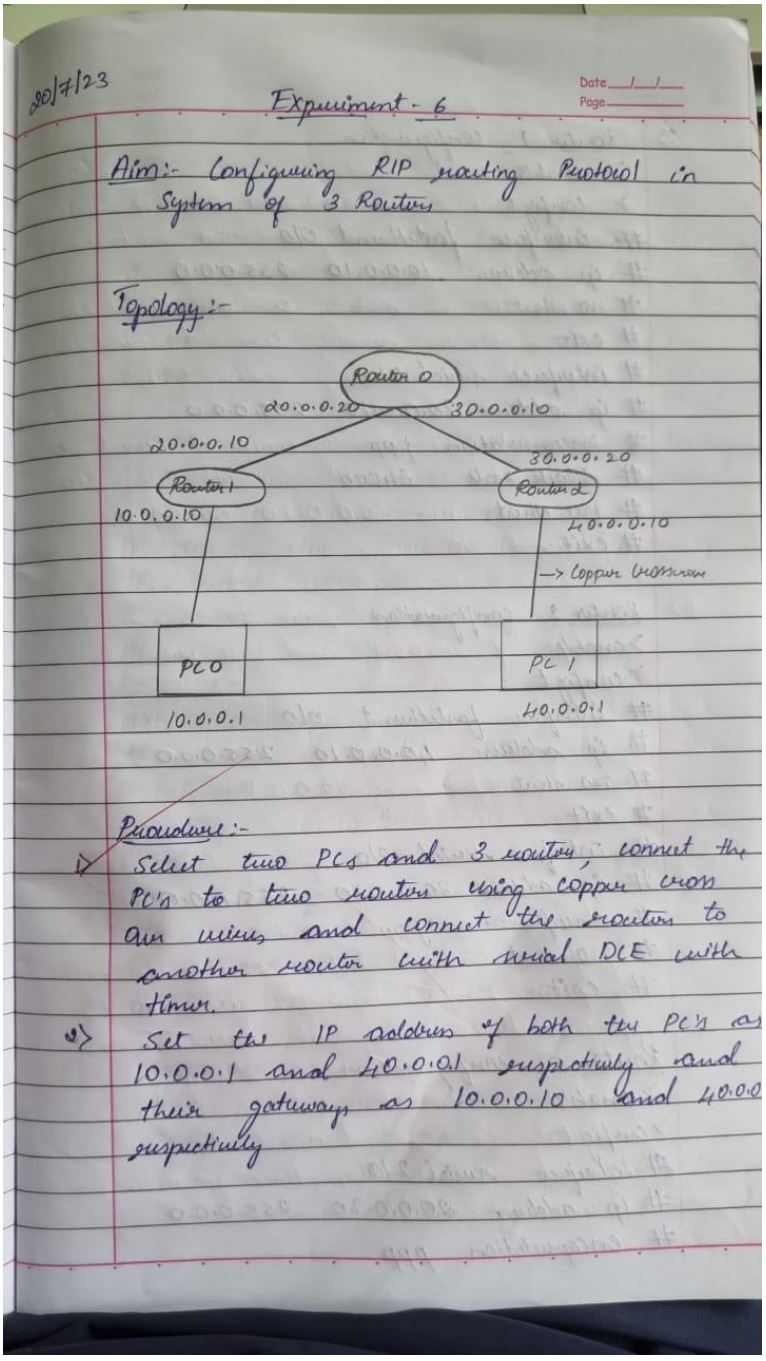

PC2
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.3
Pinging 20.0.0.3 with 32 bytes of data:
Request timed out.
Reply from 20.0.0.3: bytes=32 time=0ms TTL=127
Reply from 20.0.0.3: bytes=32 time=0ms TTL=127
Reply from 20.0.0.3: bytes=32 time=0ms 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>

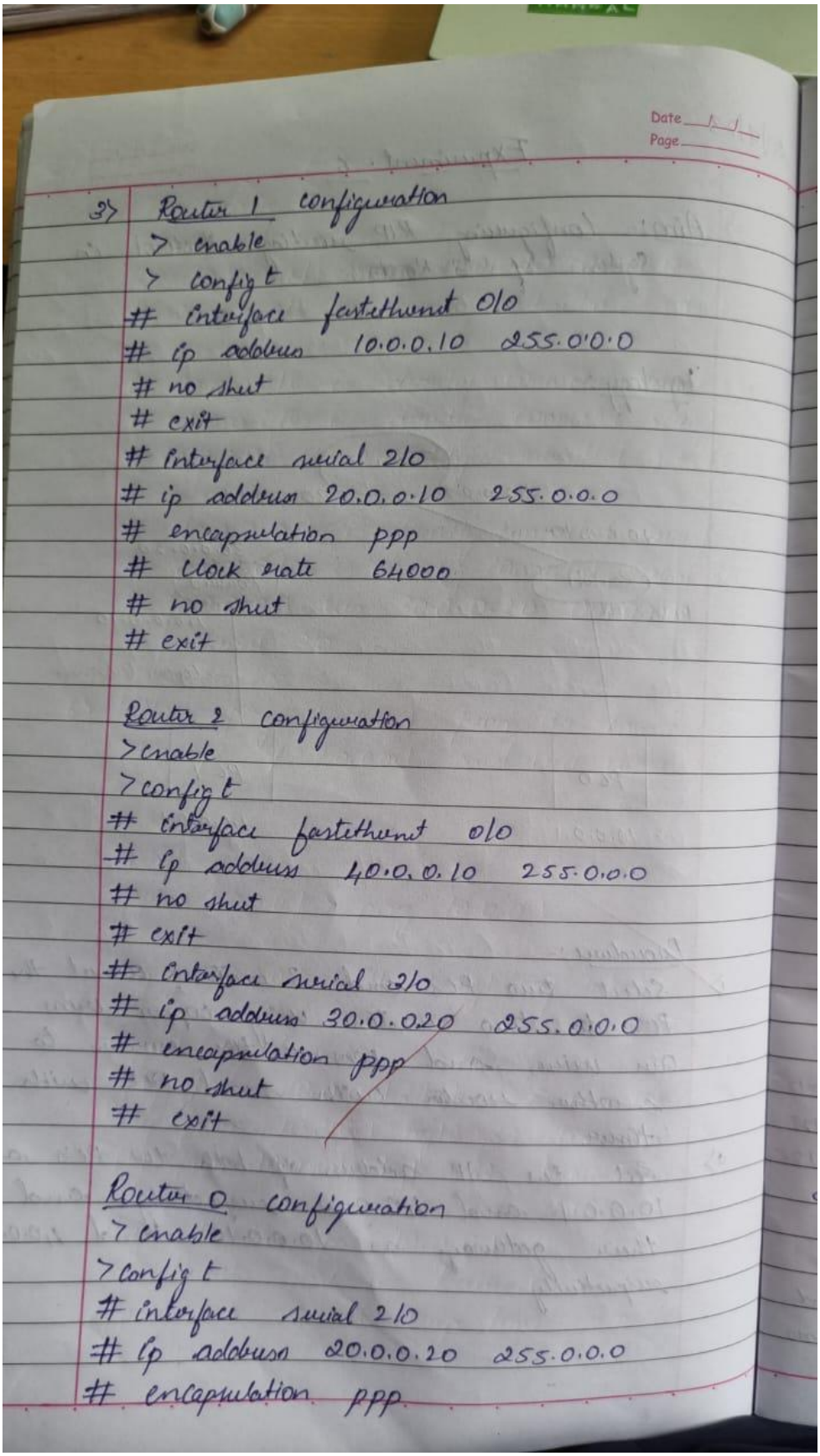
EXPERIMENT 5

AIM:

Configure RIP routing Protocol in Routers

OBSERVATION:

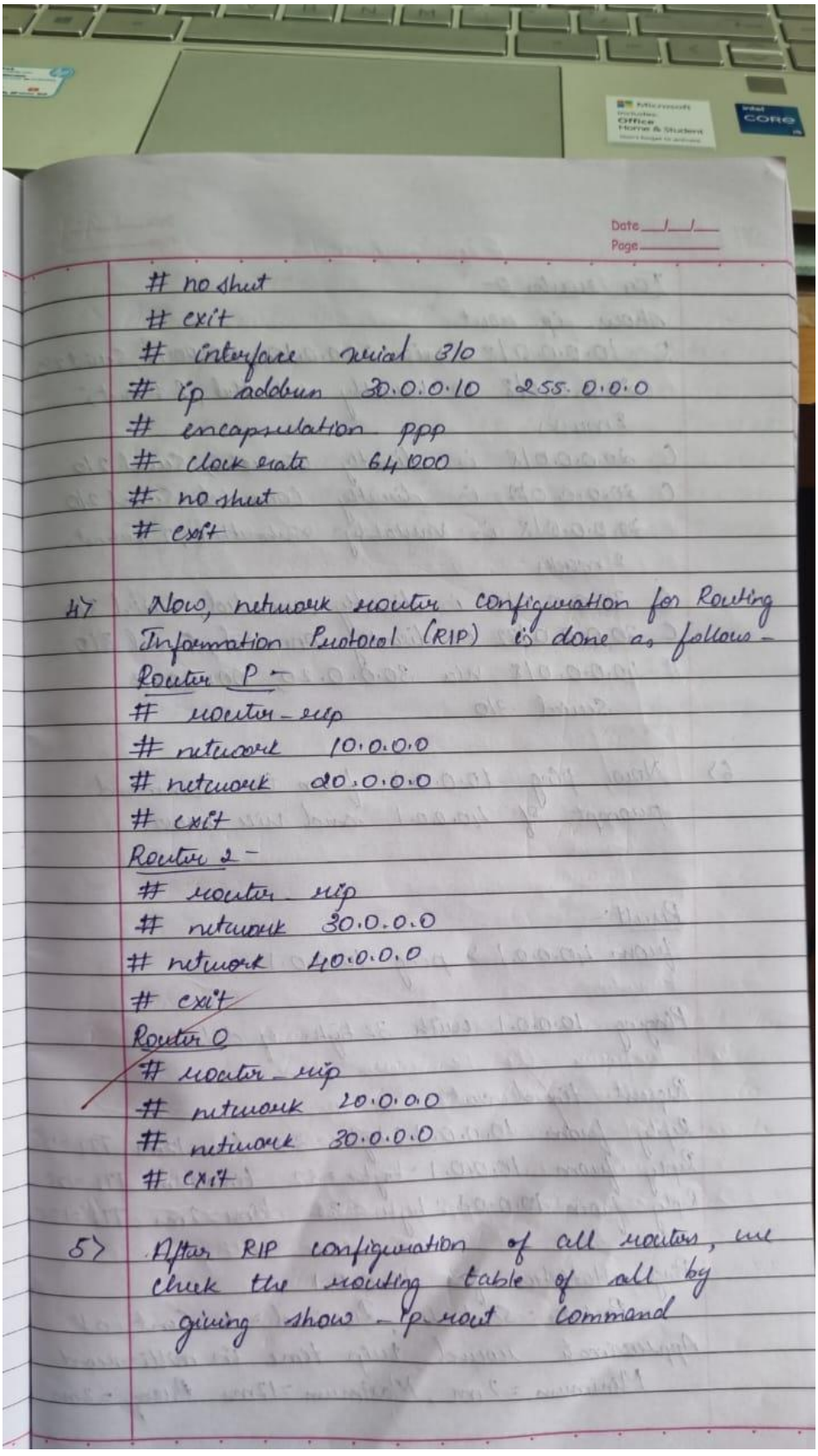




3) Router 1 configuration
> enable
> config t
interface fastethernet 0/0
ip address 10.0.0.10 255.0.0.0
no shut
exit
interface serial 2/0
ip address 20.0.0.10 255.0.0.0
encapsulation ppp
clock rate 64000
no shut
exit

Router 2 configuration
> enable
> config t
interface fastethernet 0/0
ip address 40.0.0.10 255.0.0.0
no shut
exit
interface serial 2/0
ip address 30.0.0.20 255.0.0.0
encapsulation ppp
no shut
exit

Router 0 configuration
> enable
> config t
interface serial 2/0
ip address 20.0.0.20 255.0.0.0
encapsulation ppp



Date __/__/__
Page _____

```
# no shut
# exit
# interface serial 3/0
# ip address 30.0.0.10 255.0.0.0
# encapsulation ppp
# clock rate 64000
# no shut
# exit
```

4) Now, network router configuration for Routing Information Protocol (RIP) is done as follows -

Router P -

```
# router rip
# network 10.0.0.0
# network 20.0.0.0
# exit
```

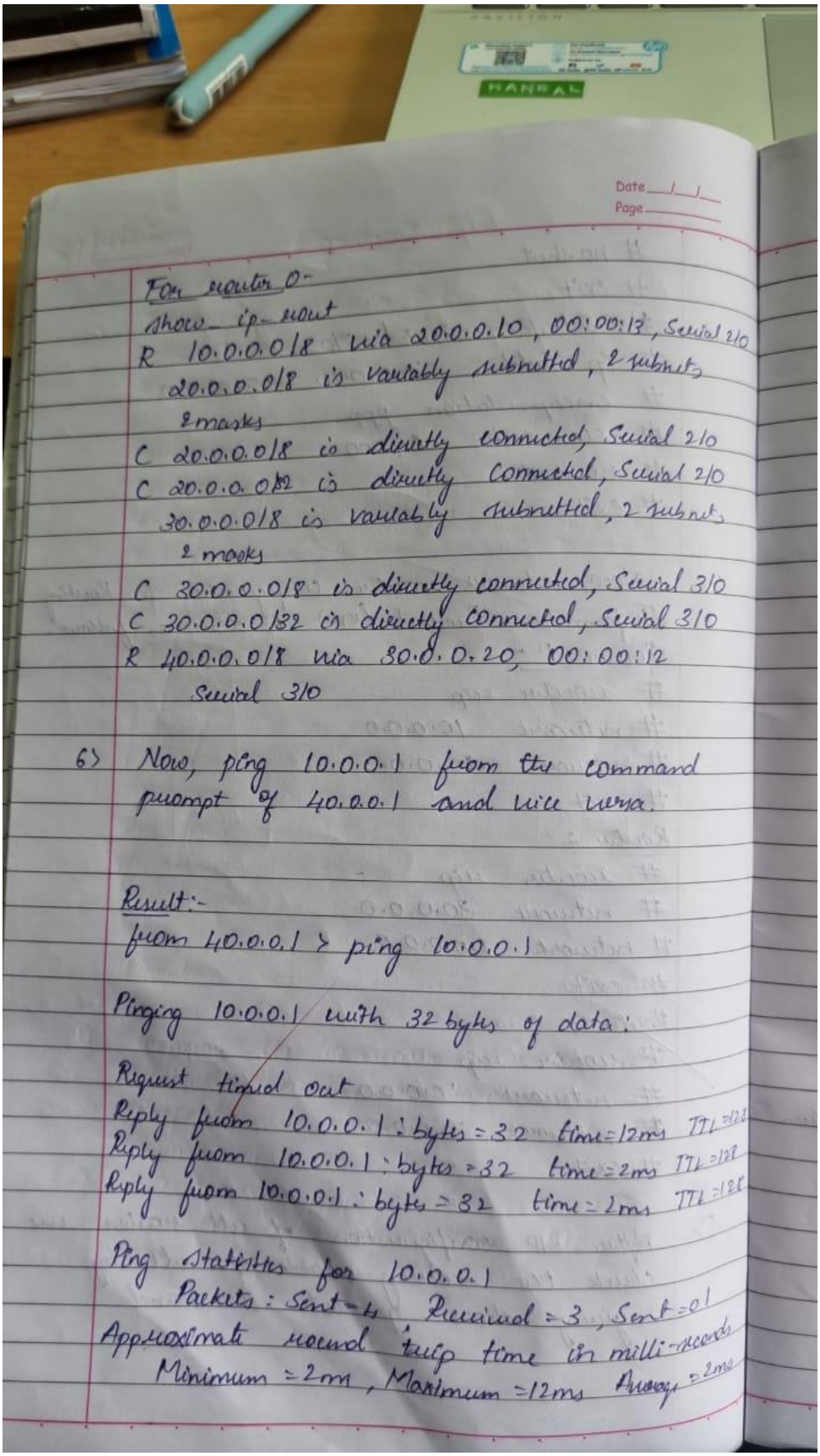
Router 2 -

```
# router rip
# network 30.0.0.0
# network 40.0.0.0
# exit
```

Router Q

```
# router rip
# network 20.0.0.0
# network 30.0.0.0
# exit
```

5) After RIP configuration of all routers, we check the routing table of all by giving show ip rout command



For router 0-
show ip route
R 10.0.0.0/8 via 20.0.0.10, 00:00:13, Serial 2/0
20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 20.0.0.0/8 is directly connected, Serial 2/0
C 20.0.0.0/32 is directly connected, Serial 2/0
30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 30.0.0.0/8 is directly connected, Serial 3/0
C 30.0.0.0/32 is directly connected, Serial 3/0
R 40.0.0.0/8 via 30.0.0.20, 00:00:12
Serial 3/0

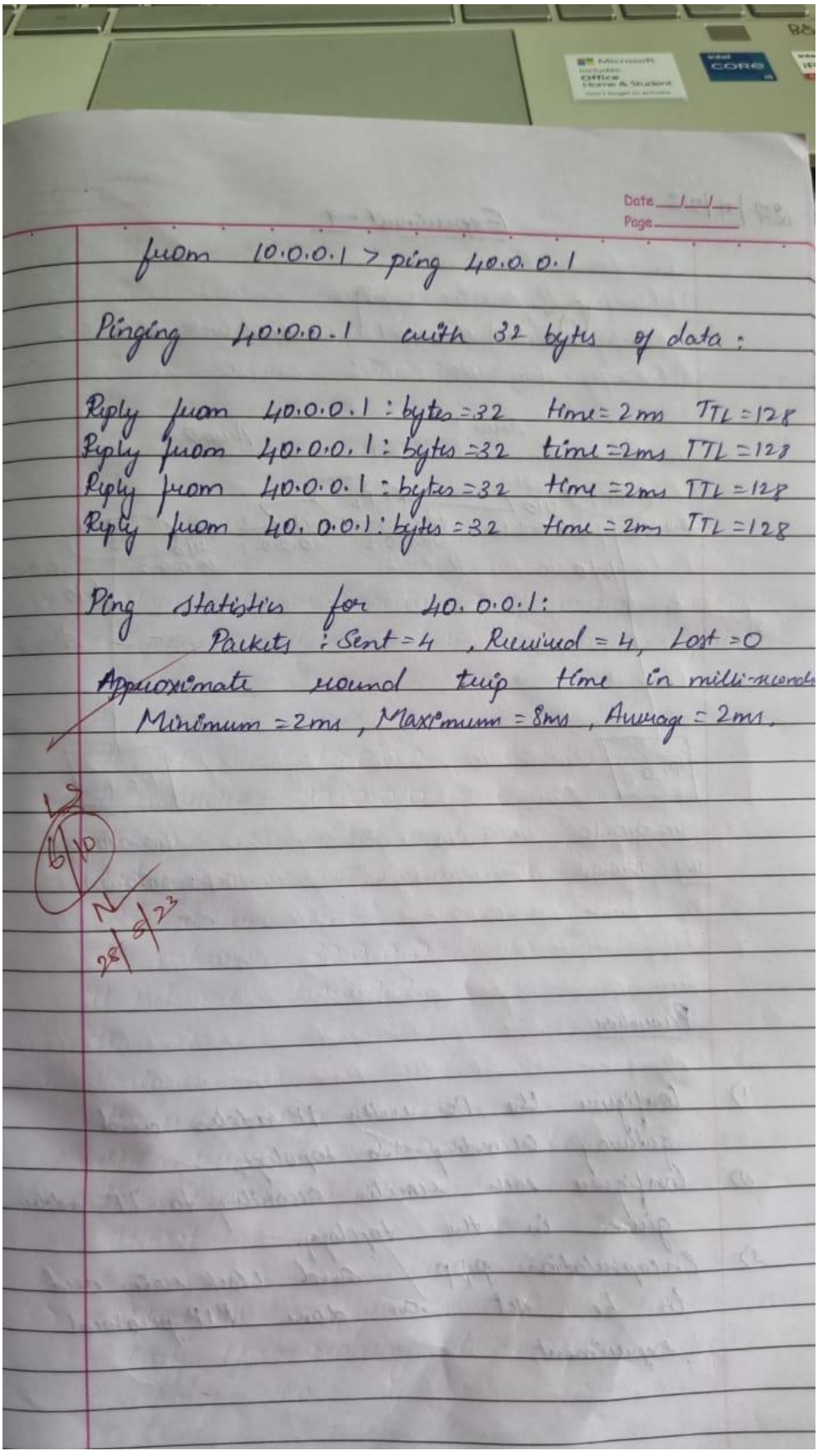
6) Now, ping 10.0.0.1 from the command prompt of 40.0.0.1 and vice versa.

Result:-
from 40.0.0.1 > ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Request timed out
Reply from 10.0.0.1: bytes=32 time=12ms TTL=128
Reply from 10.0.0.1: bytes=32 time=2ms TTL=128
Reply from 10.0.0.1: bytes=32 time=2ms TTL=128

Ping statistics for 10.0.0.1
Packets: Sent=4, Received=3, Lost=0
Approximate round trip time in milli-seconds
Minimum=2ms, Maximum=12ms Average=2ms



Date / /
Page

from 10.0.0.1 > ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

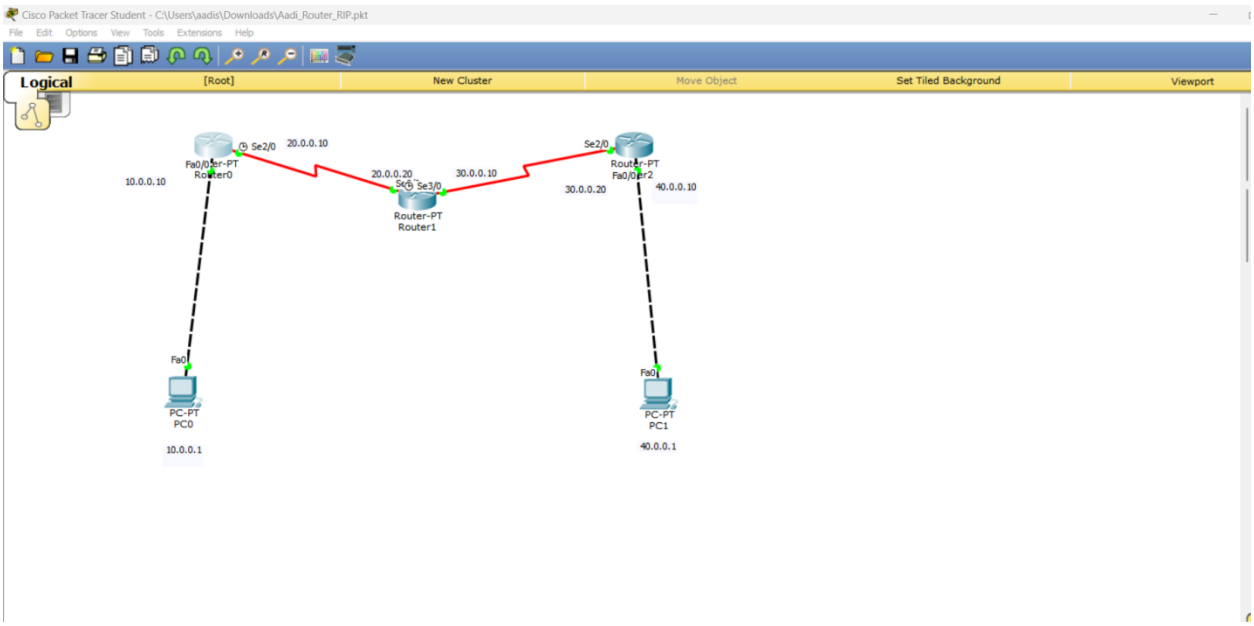
Reply from 40.0.0.1 : bytes = 32 time = 2ms TTL = 128
Reply from 40.0.0.1 : bytes = 32 time = 2ms TTL = 128
Reply from 40.0.0.1 : bytes = 32 time = 2ms TTL = 128
Reply from 40.0.0.1 : bytes = 32 time = 2ms TTL = 128

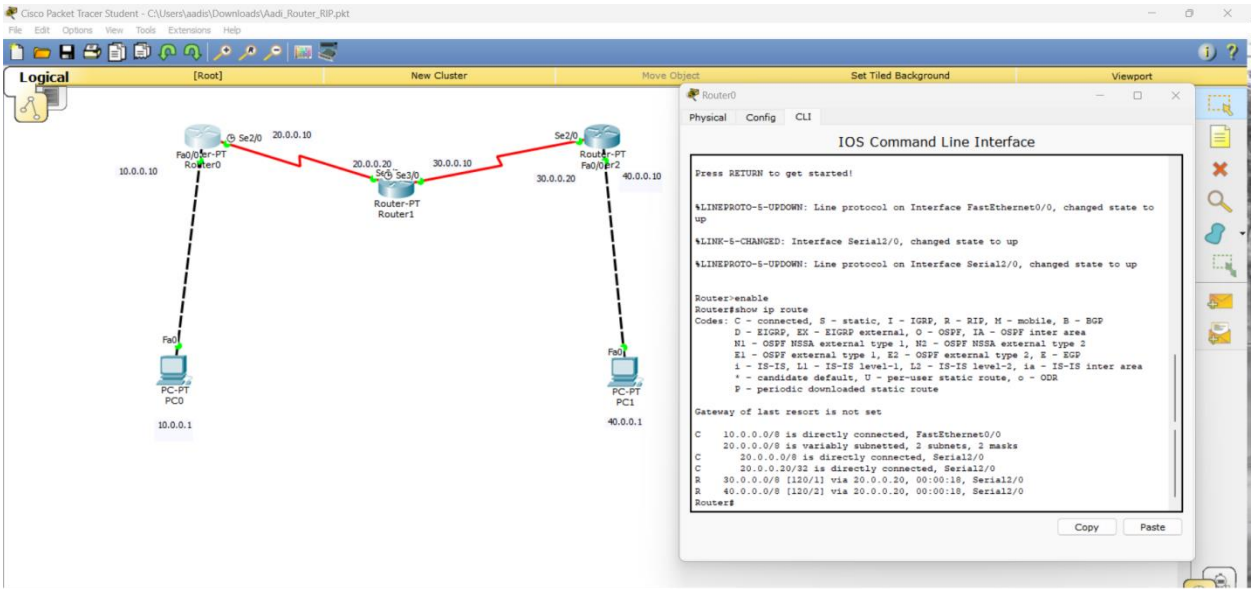
Ping statistics for 40.0.0.1:
Packets : Sent = 4, Received = 4, Lost = 0
Approximate round trip time in milli-seconds
Minimum = 2ms, Maximum = 8ms, Average = 2ms.

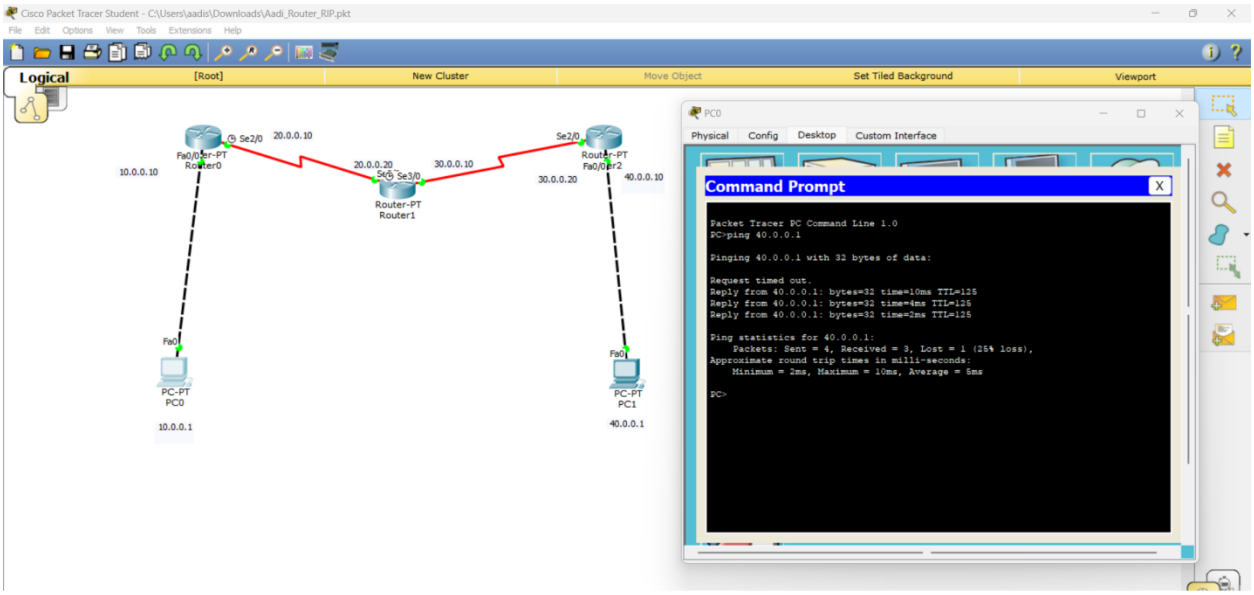
6/10

28/8/23

Result:


Cisco Packet Tracer Student - C:\Users\aadis\Downloads\Aadi_Router_RIP.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Se2/0
20.0.0.10
Fa0/0
Router-PT
Router0
10.0.0.10
20.0.0.20
Se3/0
30.0.0.10
Router-PT
Router1
Se2/0
Router-PT
Fa0/0
Router2
30.0.0.20
40.0.0.10
Fa0
PC-PT
PC0
10.0.0.1
Fa0
PC-PT
PC1
40.0.0.1


Cisco Packet Tracer Student - C:\Users\aadis\Downloads\Aadi_Router_RIP.pkt
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router0
Physical
Config
CLI
IOS Command Line Interface
Press RETURN to get started!
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
Router>enable
Router#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
20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 20.0.0.0/8 is directly connected, Serial2/0
C 20.0.0.20/32 is directly connected, Serial2/0
R 30.0.0.0/8 [120/1] via 20.0.0.20, 00:00:18, Serial2/0
R 40.0.0.0/8 [120/2] via 20.0.0.20, 00:00:18, Serial2/0
Router#
Copy
Paste
Router-PT Router0
Router-PT Router1
Router-PT Router2
PC-PT PC0
PC-PT PC1
10.0.0.10
20.0.0.10
20.0.0.20
30.0.0.10
30.0.0.20
40.0.0.10
10.0.0.1
40.0.0.1

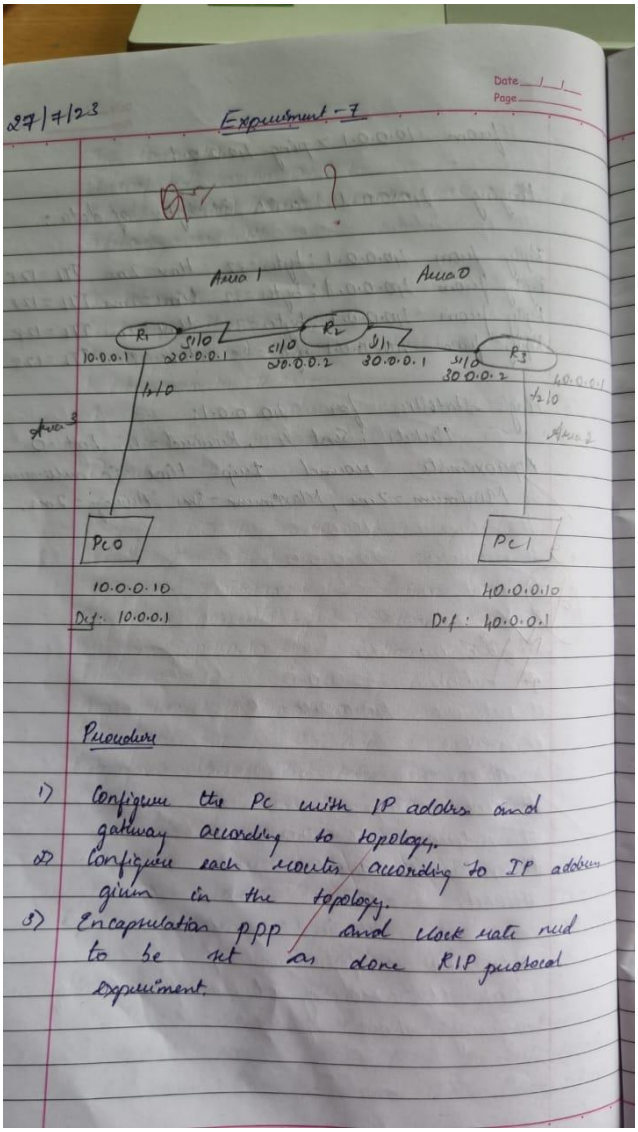

Cisco Packet Tracer Student - C:\Users\aadis\Downloads\Aadi_Router_RIP.pkt
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Request timed out.
Reply from 40.0.0.1: bytes=32 time=10ms TTL=125
Reply from 40.0.0.1: bytes=32 time=4ms 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 = 10ms, Average = 5ms
PC>
Router-PT Router0
Router-PT Router1
Router-PT Router2
PC-PT PC0
PC-PT PC1
10.0.0.10
20.0.0.10
20.0.0.20
30.0.0.10
30.0.0.20
40.0.0.10
10.0.0.1
40.0.0.1

EXPERIMENT 6

AIM:

Configure OSPF routing protocol.

OBSERVATION:



Date: / /
Page:

4). In Router R_1

```
R1 (Config)# router ospf 1
R1 (Config-router)# router-id 1.1.1.1
R1 (Config-router)# network 10.0.0.0 0.255.255.253 area 2
R1 (Config-router)# network 20.0.0.0 0.255.255.255 area 1
R1 (Config-router)# exit
```

In Router R_2

```
R2 (Config)# router ospf 1
R2 (Config-router)# router-id 2.2.2.2
R2 (Config-router)#network 20.0.0.0 0.255.255.255 area 1
R2 (Config-router)#network 30.0.0.0 0.255.255.255 area 0
R2 (Config-router)# exit
```

In Router R_3

```
R3 (Config)# router ospf 1
R3 (Config-router)# router-id 3.3.3.3
R3 (Config-router
```

```
R1 (Config-if)# interface loopback 0
R1 (Config-if)# ip add 172.16.1.252 255.255.0.0
R1 (Config-if)# no shutdown
```

```
R2 (Config-if)# interface loopback 0
R2 (Config-if)# ip add 172.16.1.253 255.255.0.0
R2 (Config-if)# no shutdown
```

```
R3 (Config-if)# interface loopback 0
R3 (Config-if)# ip add 172.16.1.254 255.255.0.0
R3 (Config-if)# no shutdown
```

Date / /
Page

In Router R_1

R_1(config)# router ospf 1

R_1(config-router)# area 1 virtual-link 2.2.2.2

R_1(config-router)#

In Router R_2

R_2(config)# router ospf 1

R_2(config-router)# area 1 virtual-link 1.1.1.1

R_2(config-router)# exit

Output:-

Pinging 40.0.0.10 with 32 bytes of data

Request timed out

Reply from 40.0.0.10: bytes = 32 time = 2ms TTL = 125

Reply from 40.0.0.10: bytes = 32 time = 2ms TTL = 125

Reply from 40.0.0.10: bytes = 32 time = 2ms TTL = 125

Ping statistics for 40.0.0.10

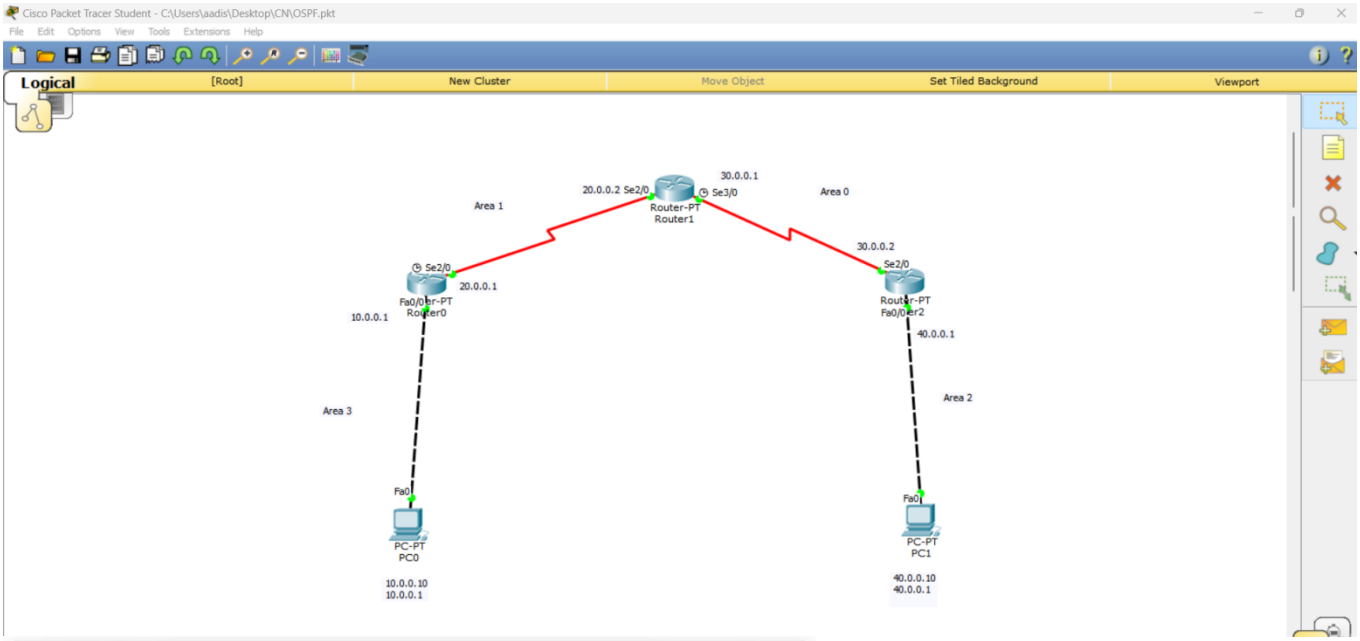
Packets: sent = 4, Received = 3, Lost = 1 (25% loss)

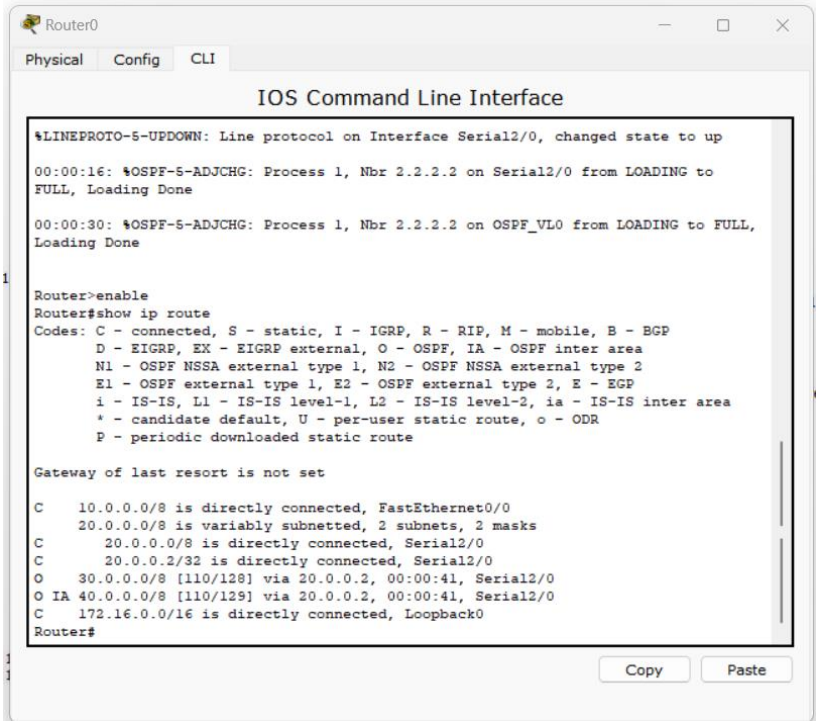
Approx round trip times in milliseconds

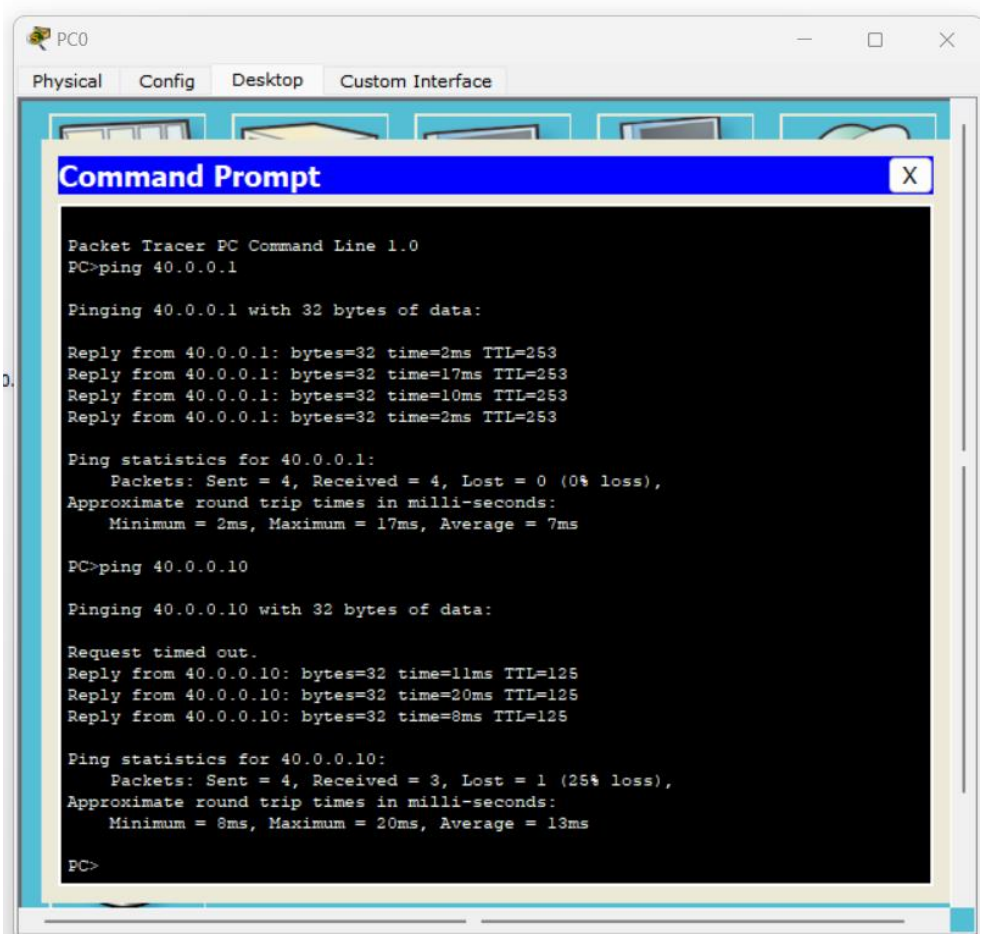
min = 2ms, max = 10ms, Average = 7ms

6/10 LS N 28/8/23

Result:


Cisco Packet Tracer Student - C:\Users\aadis\Desktop\CN\OSPF.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Area 1
Area 0
Area 3
Area 2
20.0.0.2 Se2/0
30.0.0.1
Se3/0
Router-PT
Router1
Se2/0
20.0.0.1
10.0.0.1
30.0.0.2
Se2/0
40.0.0.1
Fa0
PC-PT
PC0
10.0.0.10
10.0.0.1
Fa0
PC-PT
PC1
40.0.0.10
40.0.0.1


Router0
Physical
Config
CLI
IOS Command Line Interface
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
00:00:16: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Serial2/0 from LOADING to FULL, Loading Done
00:00:30: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on OSPF_VL0 from LOADING to FULL, Loading Done
Router>enable
Router#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
20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 20.0.0.0/8 is directly connected, Serial2/0
C 20.0.0.2/32 is directly connected, Serial2/0
O 30.0.0.0/8 [110/128] via 20.0.0.2, 00:00:41, Serial2/0
O IA 40.0.0.0/8 [110/129] via 20.0.0.2, 00:00:41, Serial2/0
C 172.16.0.0/16 is directly connected, Loopback0
Router#
Copy
Paste

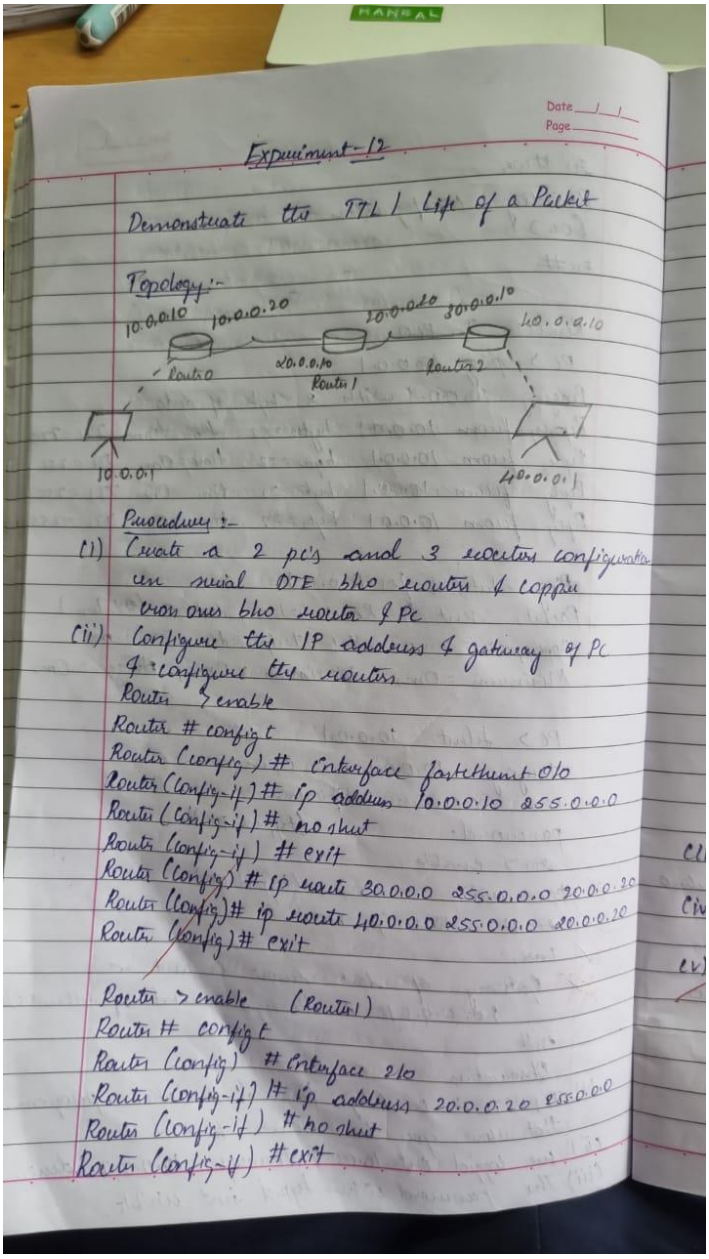

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Reply from 40.0.0.1: bytes=32 time=2ms TTL=253
Reply from 40.0.0.1: bytes=32 time=17ms TTL=253
Reply from 40.0.0.1: bytes=32 time=10ms TTL=253
Reply from 40.0.0.1: bytes=32 time=2ms TTL=253
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 = 17ms, Average = 7ms
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=11ms TTL=125
Reply from 40.0.0.10: bytes=32 time=20ms TTL=125
Reply from 40.0.0.10: bytes=32 time=8ms 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 = 8ms, Maximum = 20ms, Average = 13ms
PC>

EXPERIMENT 7

AIM:

Demonstrate the TTL/ Life of a Packet.

OBSERVATION:



Date / /
Page

Router (config) # interface serial 2/0
Router (config-if) # ip address 30.0.0.10 255.0.0.0
Router (config-if) # no shut
Router (config-if) # exit

Router (config)# ip route 10.0.0.0 255.0.0.0 20.0.0.10
Router (config)# ip route 40.0.0.0 255.0.0.0 30.0.0.20
Router (config) # exit
Router > enable (Router 2)

Router # config t
Router (config) # interface serial 2/0
Router (config-if) # ip address 30.0.0.20 255.0.0.0
Router (config-if) # no shut
Router (config-if) # exit
Router (config)# interface fastethernet 0/0
Router (config-if) # ip address 40.0.0.10 255.0.0.0
Router (config-if) # no shut
Router (config-if) # exit
Router (config) # ip route 10.0.0.0 255.0.0.0 30.0.0.10
Router (config) # ip route 20.0.0.0 255.0.0.0 30.0.0.10

(iii) Select simulation mode, select simple PDU and select source & destination PC's.
(iv) Use Capture button to sent PDU & acknowledgment from PC to router, router to PC.
(v) Click on PDU during every transfer to see the Inbound and Outbound PDU details. Observe the differences in the PDUs

Result:-

PDU information at device : PC0

Outbound PDU details

TTL : 255

PDU information at Device : PC0

Inbound PDU details

TTL : 255

PDU information at Device Router 0

Outbound PDU details

TTL : 255

PDU information at Device Router 1

Inbound PDU details

TTL : 254

Outbound PDU details

TTL : 253

PDU information at Device Router 2

Inbound PDU details

TTL : 253

Outbound PDU details

TTL : 252

PDU information at device PC1

Inbound PDU details

TTL : 252

An example :- for Inbound details of Router 0

O/P

4	IHL	DSCP : 0x0	TL : 28
ID : 0x0		0x0	0x0
TTL : 255	PRO : 0x1	CHECKSUM	
SRC IP 10.0.0.1			
DST IP 40.0.0.1			
OPT : 0x0		0x0	
DATA (Variable length)			

Date / /
Page

Observation

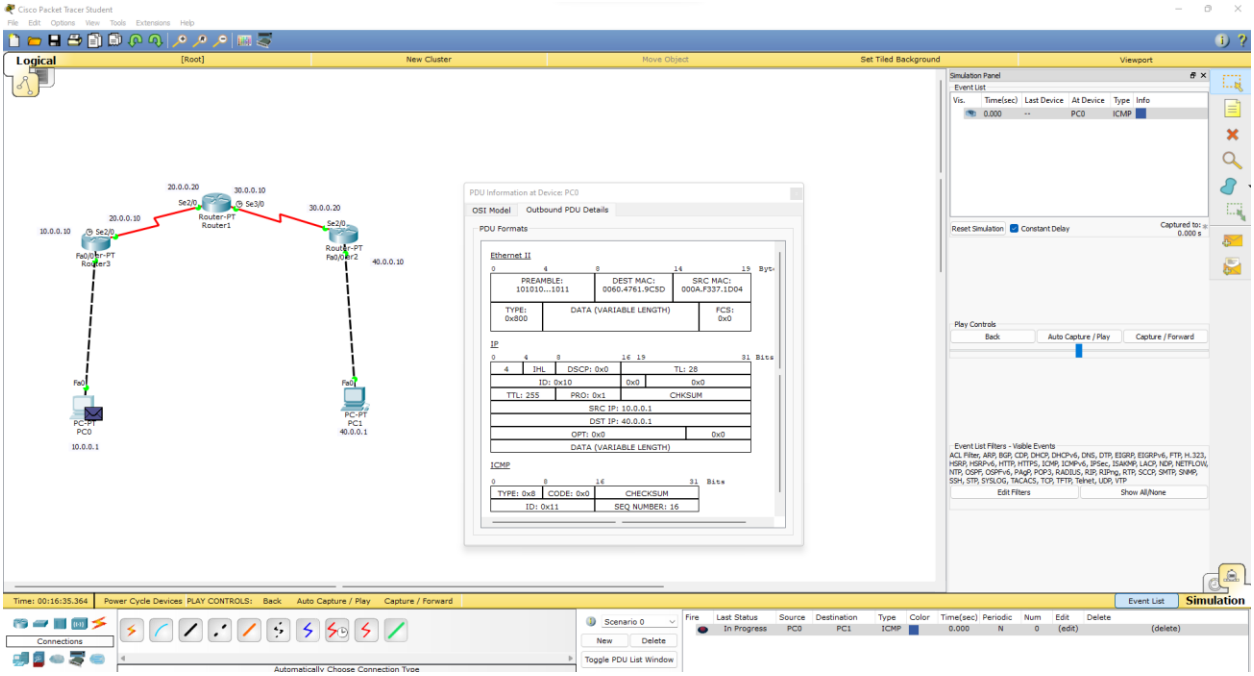
(i). The TTL is reduced by 1 in every router. Time to live (TTL) is a mechanism which limits the lifeform or lifetime of data in a computer or network. It is a counter 255 is set as max TTL.

6/10

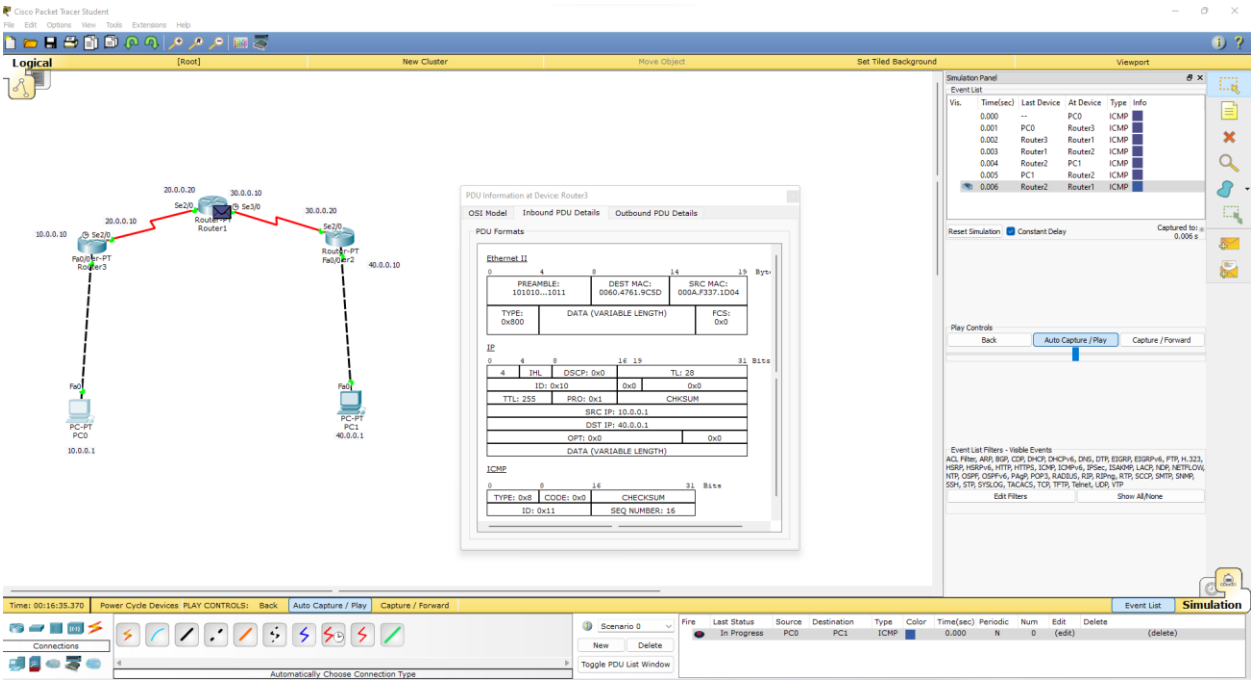
25 | 8 | 23

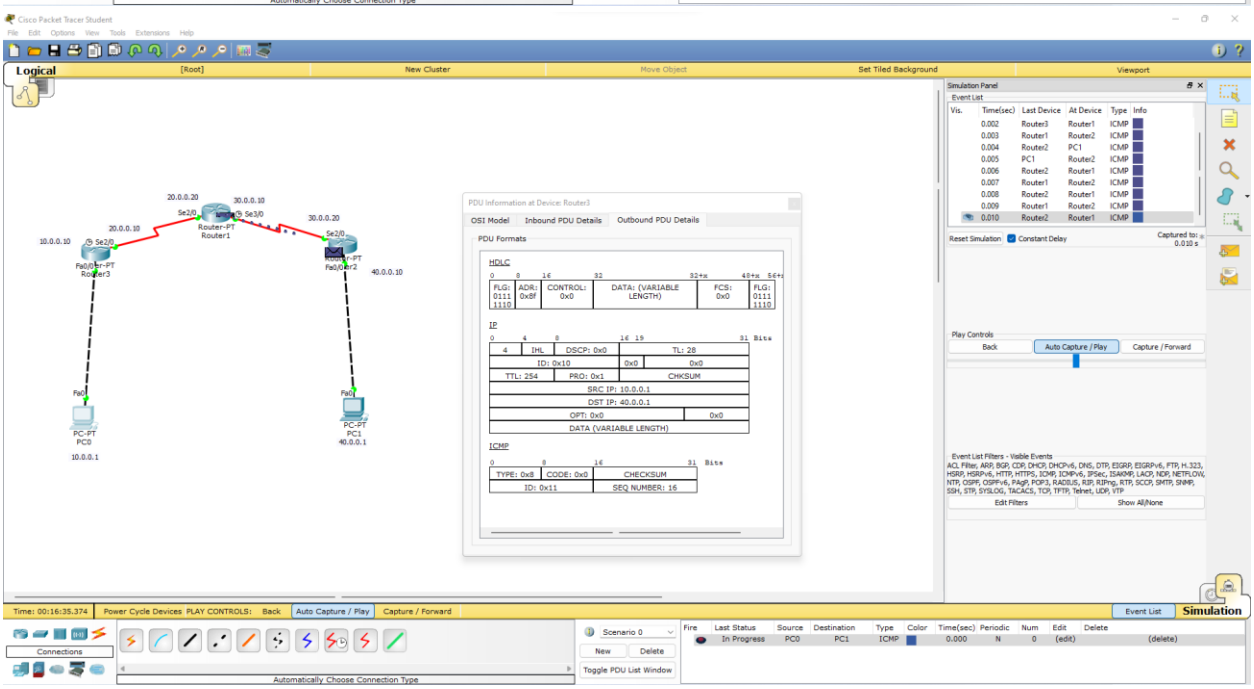
Result:

PC0:

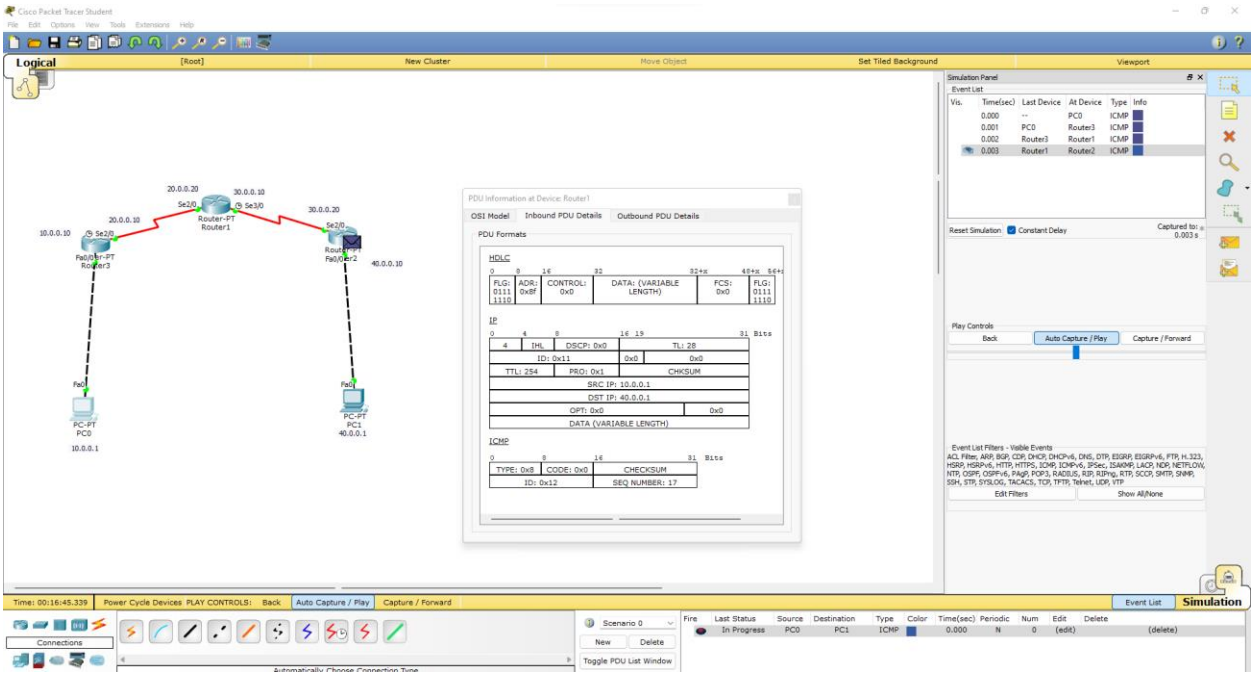


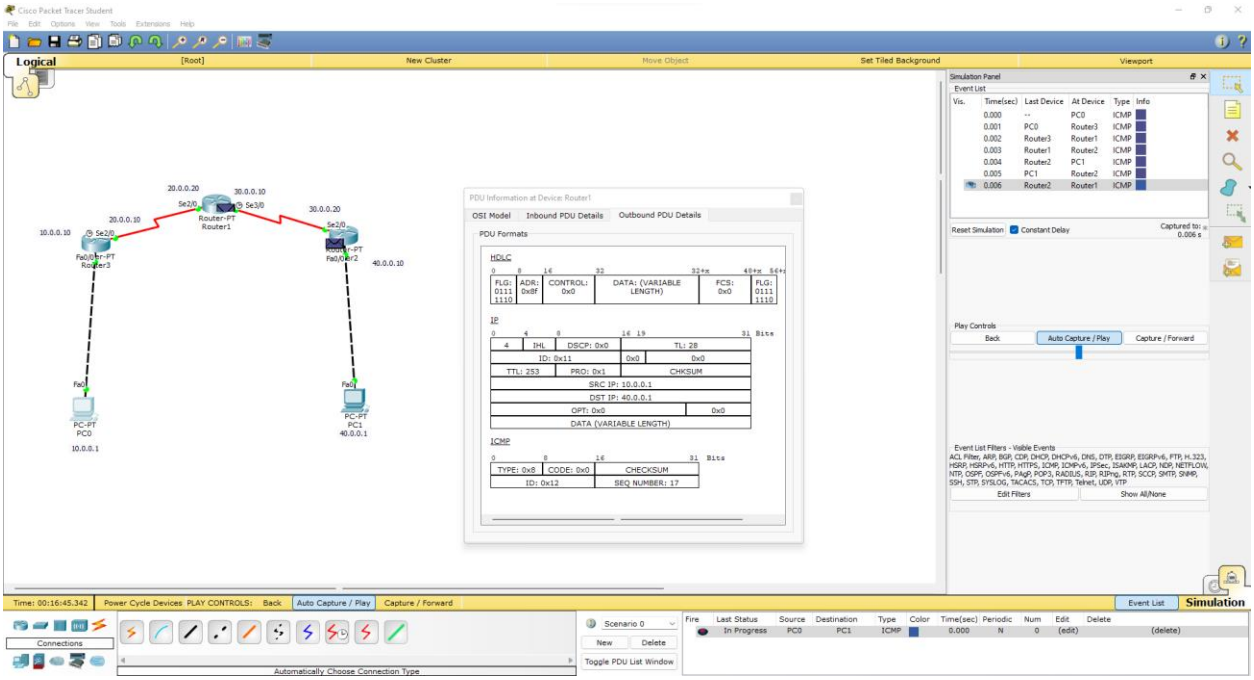
Router 0:


Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router1
Router3
PC0
PC1
PDU Information at Device: Router3
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
IP
ICMP
Simulation Panel
Event List
Simulation

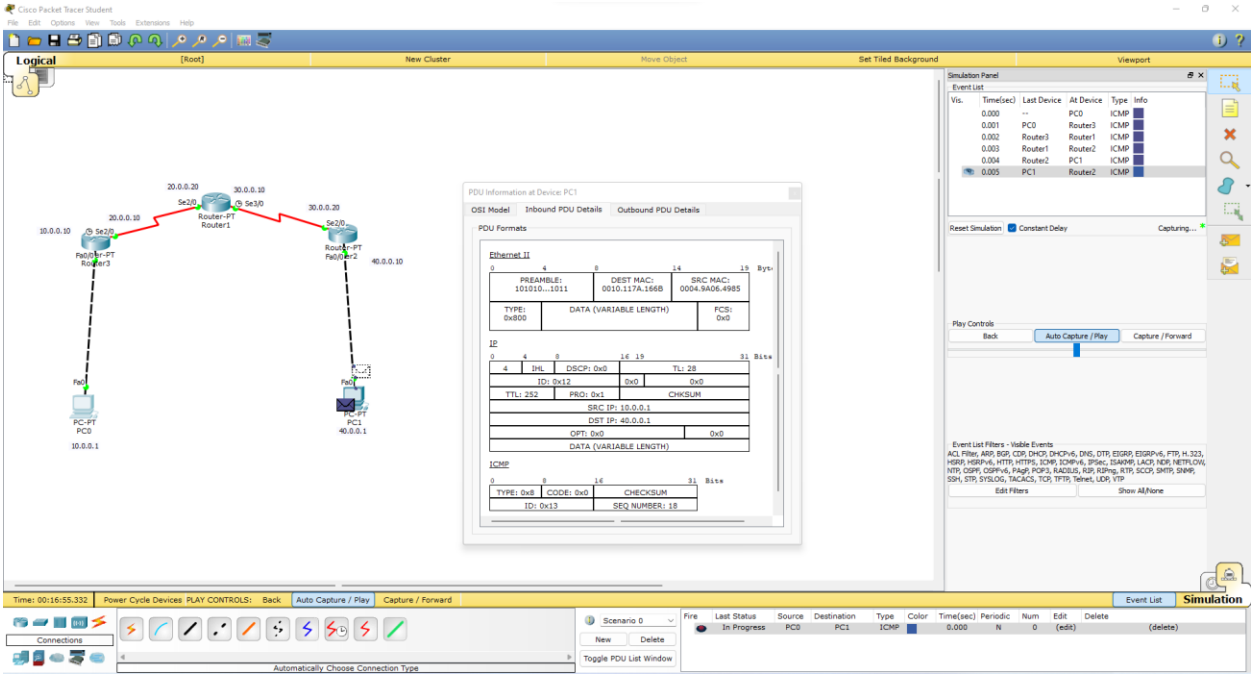

Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router1
Router3
PC0
PC1
PDU Information at Device: Router3
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
IP
ICMP
Simulation Panel
Event List
Simulation

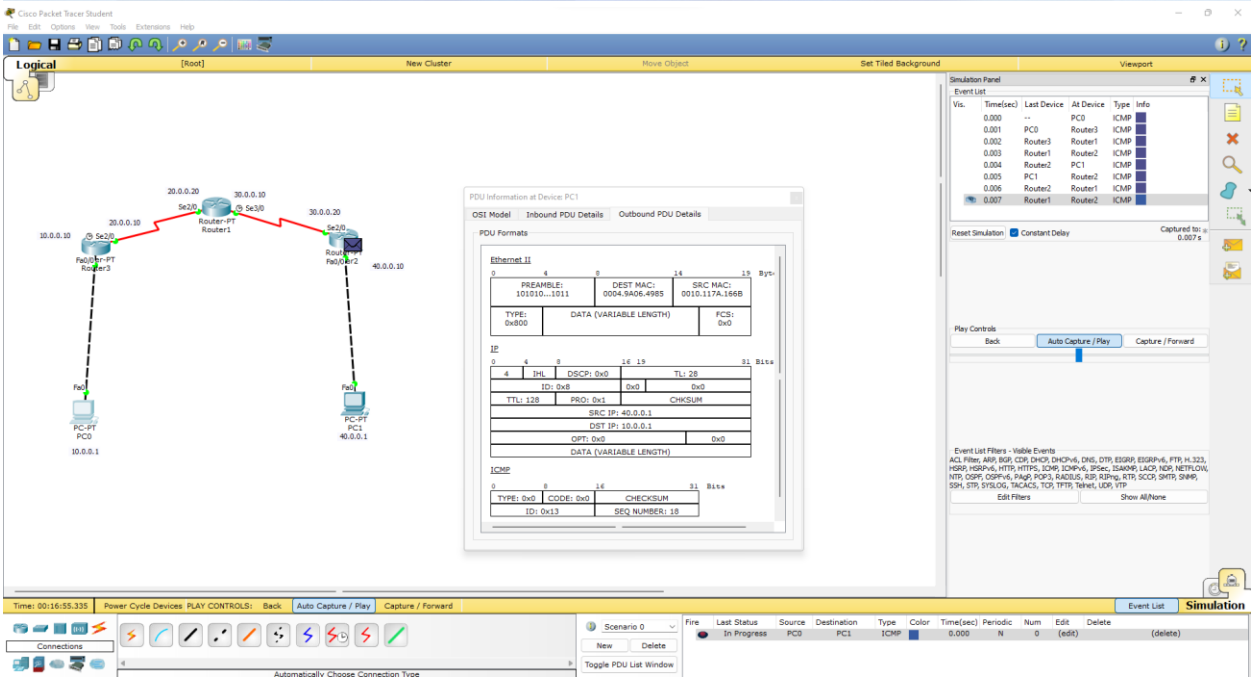
Router 1:


Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT Router1
Router-PT Router3
Router-PT Router2
PC-PT PC0
PC-PT PC1
20.0.0.20
30.0.0.10
20.0.0.10
30.0.0.20
10.0.0.10
40.0.0.10
10.0.0.1
40.0.0.1
PDU Information at Device: Router1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
IP
ICMP
Simulation Panel
Event List
Play Controls
Back
Auto Capture / Play
Capture / Forward
Time: 00:16:45.339
Power Cycle Devices
PLAY CONTROLS:
Event List
Simulation
Connections
Scenario 0
New
Delete
Toggle PDU List Window

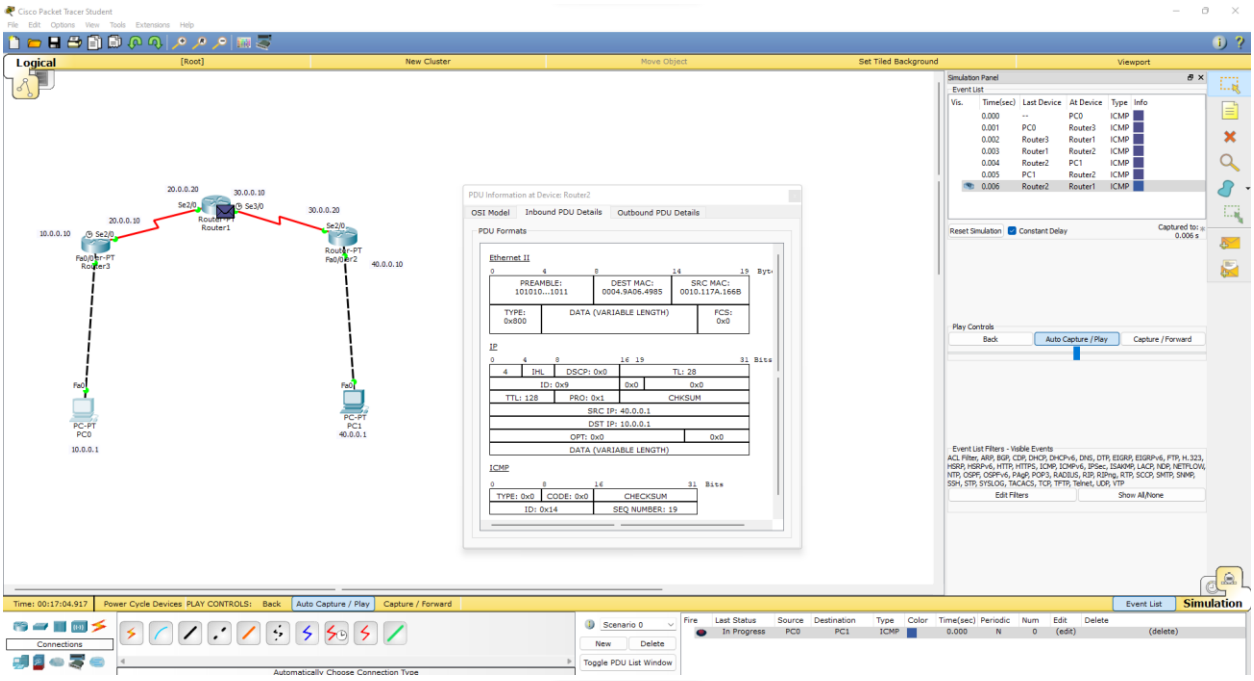

Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT Router1
Router-PT Router3
Router-PT Router2
PC-PT PC0
PC-PT PC1
20.0.0.20
30.0.0.10
20.0.0.10
30.0.0.20
10.0.0.10
40.0.0.10
10.0.0.1
40.0.0.1
PDU Information at Device: Router1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
IP
ICMP
Simulation Panel
Event List
Play Controls
Back
Auto Capture / Play
Capture / Forward
Time: 00:16:45.342
Power Cycle Devices
PLAY CONTROLS:
Event List
Simulation
Connections
Scenario 0
New
Delete
Toggle PDU List Window
Automatically Choose Connection Type

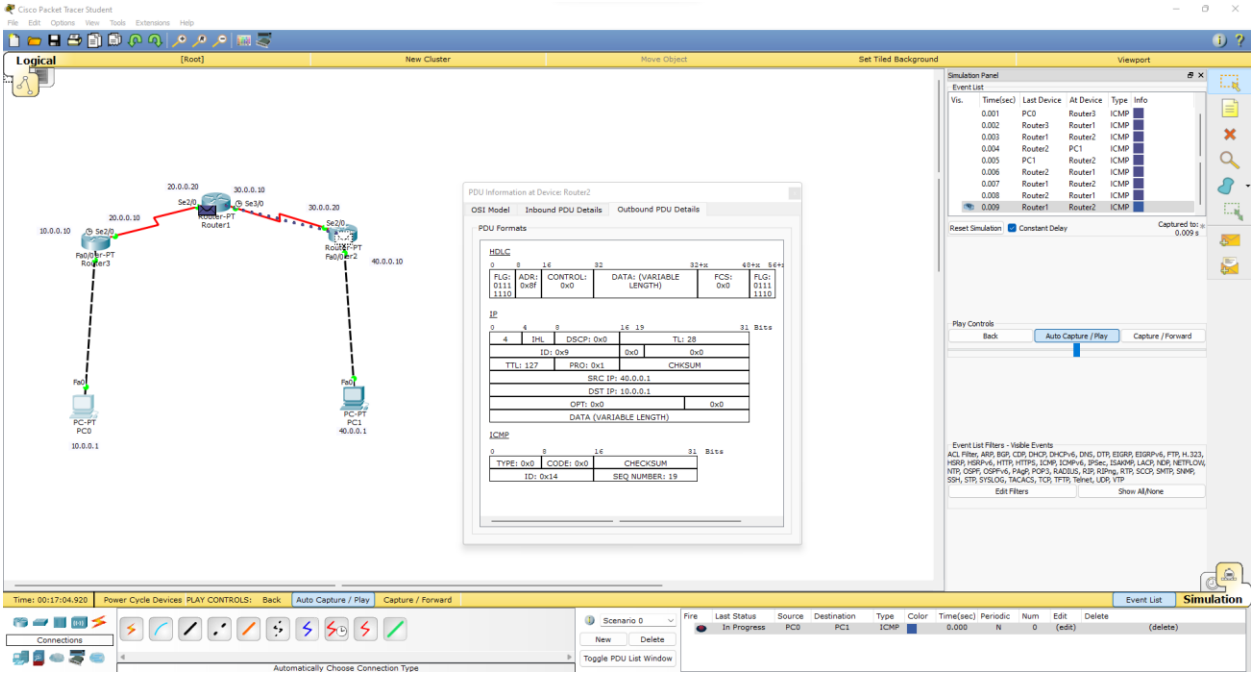
Router 2:


Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT
Router1
Router-PT
Router3
Router-PT
Router2
PC-PT
PC0
PC-PT
PC1
20.0.0.20
30.0.0.10
20.0.0.10
30.0.0.20
10.0.0.10
40.0.0.10
10.0.0.1
40.0.0.1
PDU Information at Device: PC1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
PREAMBLE: 101010...1011
DEST MAC: 0010.117A.166B
SRC MAC: 0004.9A06.4985
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
DSCP: 0x0
TL: 28
ID: 0x12
TTL: 252
PRO: 0x1
CHKSUM
SRC IP: 10.0.0.1
DST IP: 40.0.0.1
OPT: 0x0
DATA (VARIABLE LENGTH)
ICMP
TYPE: 0x8
CODE: 0x0
CHECKSUM
ID: 0x13
SEQ NUMBER: 18
Simulation Panel
Event List
Reset Simulation
Constant Delay
Play Controls
Back
Auto Capture / Play
Capture / Forward
Edit Filters
Show All/None
Time: 00:16:55.332
Power Cycle Devices
PLAY CONTROLS:
Event List
Simulation
Connections
Automatically Choose Connection Type
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Periodic
Num
Edit
Delete
In Progress
PC0
PC1
ICMP
(edit)
(delete)


Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT
Router1
Router-PT
Router3
Router-PT
Router2
PC-PT
PC0
PC-PT
PC1
20.0.0.20
30.0.0.10
20.0.0.10
30.0.0.20
10.0.0.10
40.0.0.10
10.0.0.1
40.0.0.1
PDU Information at Device: PC1
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
PREAMBLE: 101010...1011
DEST MAC: 0004.9A06.4985
SRC MAC: 0010.117A.166B
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
DSCP: 0x0
TL: 28
ID: 0x8
TTL: 128
PRO: 0x1
CHKSUM
SRC IP: 40.0.0.1
DST IP: 10.0.0.1
OPT: 0x0
DATA (VARIABLE LENGTH)
ICMP
TYPE: 0x0
CODE: 0x0
CHECKSUM
ID: 0x13
SEQ NUMBER: 18
Simulation Panel
Event List
Reset Simulation
Constant Delay
Play Controls
Back
Auto Capture / Play
Capture / Forward
Edit Filters
Show All/None
Time: 00:16:55.335
Power Cycle Devices
PLAY CONTROLS:
Event List
Simulation
Connections
Automatically Choose Connection Type
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Periodic
Num
Edit
Delete
In Progress
PC0
PC1
ICMP
(edit)
(delete)

PC1:


Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router1
Router3
Router2
PC0
PC1
20.0.0.20
30.0.0.10
20.0.0.10
30.0.0.20
10.0.0.10
40.0.0.10
10.0.0.1
40.0.0.1
PDU Information at Device: Router2
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
PREAMBLE: 101010...1011
DEST MAC: 0004.9A06.4985
SRC MAC: 0010.117A.166B
TYPE: 0x800
DATA (VARIABLE LENGTH)
FCS: 0x0
IP
DSCP: 0x0
TL: 28
ID: 0x9
TTL: 128
PRO: 0x1
CHKSUM
SRC IP: 40.0.0.1
DST IP: 10.0.0.1
OPT: 0x0
DATA (VARIABLE LENGTH)
ICMP
TYPE: 0x0
CODE: 0x0
CHECKSUM
ID: 0x14
SEQ NUMBER: 19
Simulation Panel
Event List
Reset Simulation
Constant Delay
Play Controls
Back
Auto Capture / Play
Capture / Forward
Edit Filters
Show All/None
Time: 00:17:04.917
Power Cycle Devices
PLAY CONTROLS:
Event List
Simulation
Connections
Automatically Choose Connection Type
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Periodic
Num
Edit
Delete
In Progress
PC0
PC1
ICMP
0.000
N
0
(edit)
(delete)

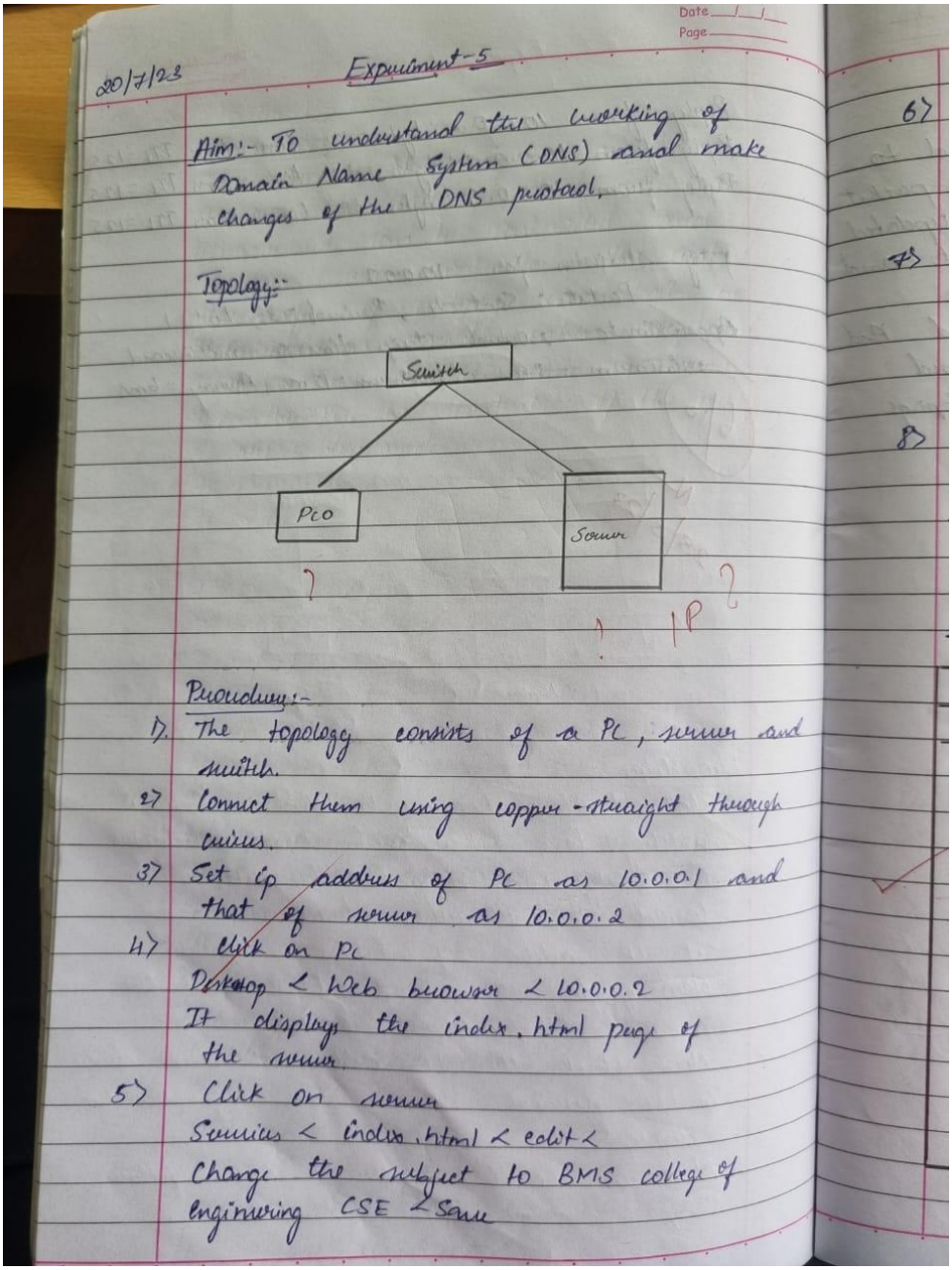

Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router1
Router3
Router2
PC0
PC1
20.0.0.20
30.0.0.10
20.0.0.10
30.0.0.20
10.0.0.10
40.0.0.10
10.0.0.1
40.0.0.1
PDU Information at Device: Router2
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
HDLC
FLG: 0111 1110
ADR: 0x8F
CONTROL: 0x0
DATA: (VARIABLE LENGTH)
FCS: 0x0
IP
DSCP: 0x0
TL: 28
ID: 0x9
TTL: 127
PRO: 0x1
CHKSUM
SRC IP: 40.0.0.1
DST IP: 10.0.0.1
OPT: 0x0
DATA (VARIABLE LENGTH)
ICMP
TYPE: 0x0
CODE: 0x0
CHECKSUM
ID: 0x14
SEQ NUMBER: 19
Simulation Panel
Event List
Reset Simulation
Constant Delay
Play Controls
Back
Auto Capture / Play
Capture / Forward
Edit Filters
Show All/None
Time: 00:17:04.920
Power Cycle Devices
PLAY CONTROLS:
Event List
Simulation
Connections
Automatically Choose Connection Type
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Periodic
Num
Edit
Delete
In Progress
PC0
PC1
ICMP
0.000
N
0
(edit)
(delete)

EXPERIMENT 8

AIM:

Configure Web Server, DNS within a LAN.

OBSERVATION:



20/7/23 Experiment-5

Aim:- To understand the working of Domain Name System (DNS) and make changes of the DNS protocol.

Topology:-

Procedure:-

1) The topology consists of a PC, server and switch.
2) Connect them using copper-straight through wires.
3) Set ip address of PC as 10.0.0.1 and that of server as 10.0.0.2
4) Click on PC
Desktop < Web browser < 10.0.0.2
It displays the index.html page of the server.
5) Click on server
Services < index.html < edit <
Change the subject to BMS college of Engineering CSE < Save

Date / /
Page

6) Click on Pc
Desktop < Web browser < 10.0.0.2
It changes the content of index.html page as set by us i.e, BMS College of Engineering CSE

7) Click on server
Services < DNS < ON <
give domain name as bmscecse <
set address as 10.0.0.2
< add

8) Click on Pc
Desktop < Web browser < bmscecse
It display the same index.html of page of the server (with ip address 10.0.0.2)

Result:-

Web Browser			X
< >	URL	http://bmscecse	Go	Stop

Cisco Packet Tracer

BMS COLLEGE OF ENGINEERING Opening doors to new opportunities. Mind with open

Quick Links:
A small page
Copyrights
Image page
Image.

Date __/__/__
Page

20/7/2

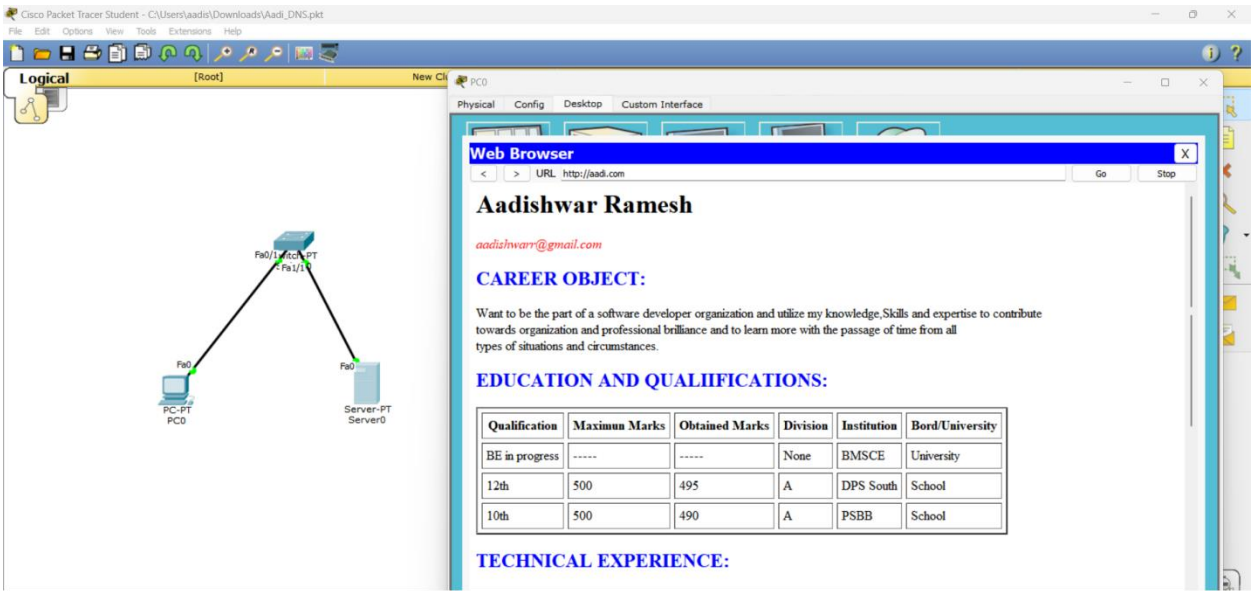
Observation -

The Domain Name System can be used to address a webpage and in cisco packet tracer, it can be controlled and updated in the server controller. The contents and the name of webpage can also be changed as and when required. And when this domain name is browsed on the PC. the contents of the webpage are displayed.

6/10

28/8/23

Result:

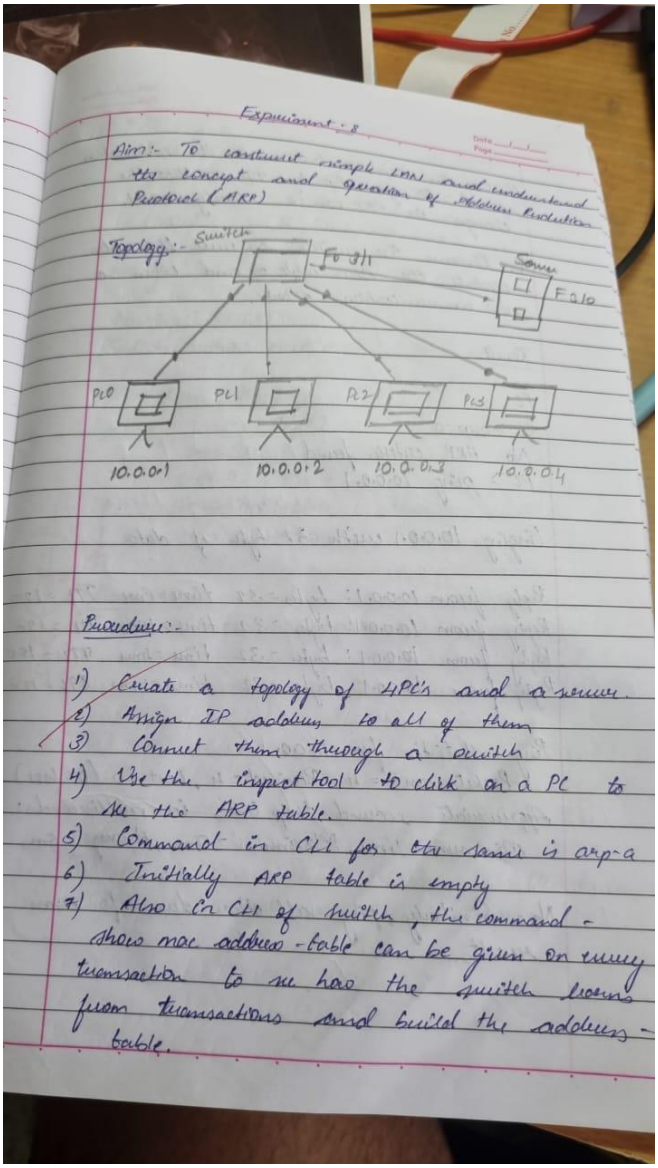

Cisco Packet Tracer Student - C:\Users\aadis\Downloads\Aadi_DNS.pkt
Logical
[Root]
PC0
Physical
Config
Desktop
Custom Interface
Web Browser
URL http://aadi.com
Go
Stop
Aadishwar Ramesh
aadishwarr@gmail.com
CAREER OBJECT:
Want to be the part of a software developer organization and utilize my knowledge,Skills and expertise to contribute towards organization and professional brilliance and to learn more with the passage of time from all types of situations and circumstances.
EDUCATION AND QUALIIFICATIONS:
Qualification | Maximun Marks | Obtained Marks | Division | Institution | Bord/University
BE in progress | ----- | ----- | None | BMSCE | University
12th | 500 | 495 | A | DPS South | School
10th | 500 | 490 | A | PSBB | School
TECHNICAL EXPERIENCE:
Fa0
PC-PT
PC0
Fa0
Server-PT
Server0

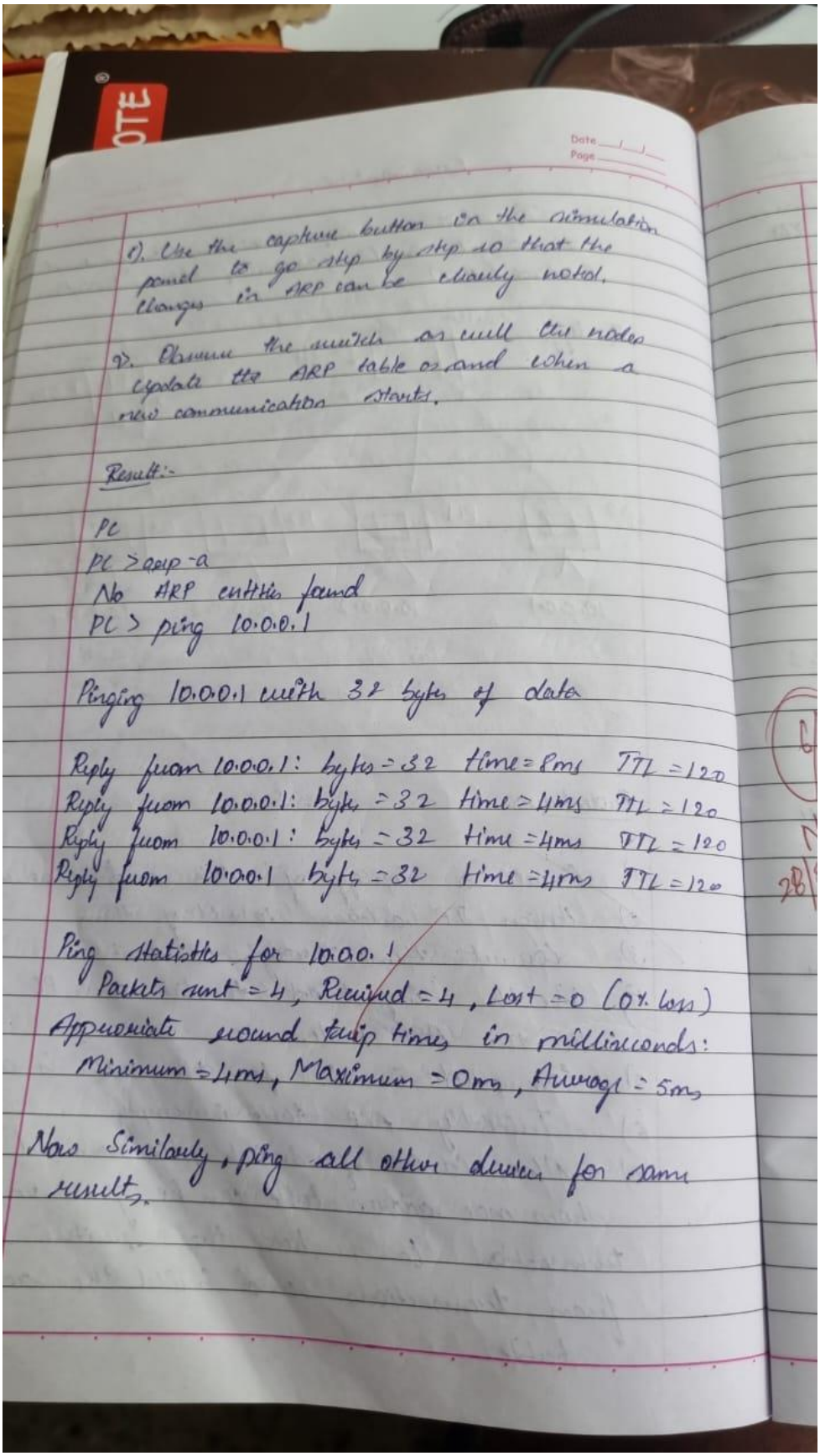
EXPERIMENT 9

AIM:

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

OBSERVATION:





6). Use the capture button on the simulation panel to go step by step so that the changes in ARP can be clearly noted.

7). Observe the switch as well the nodes update the ARP table or and when a new communication starts.

Result:-

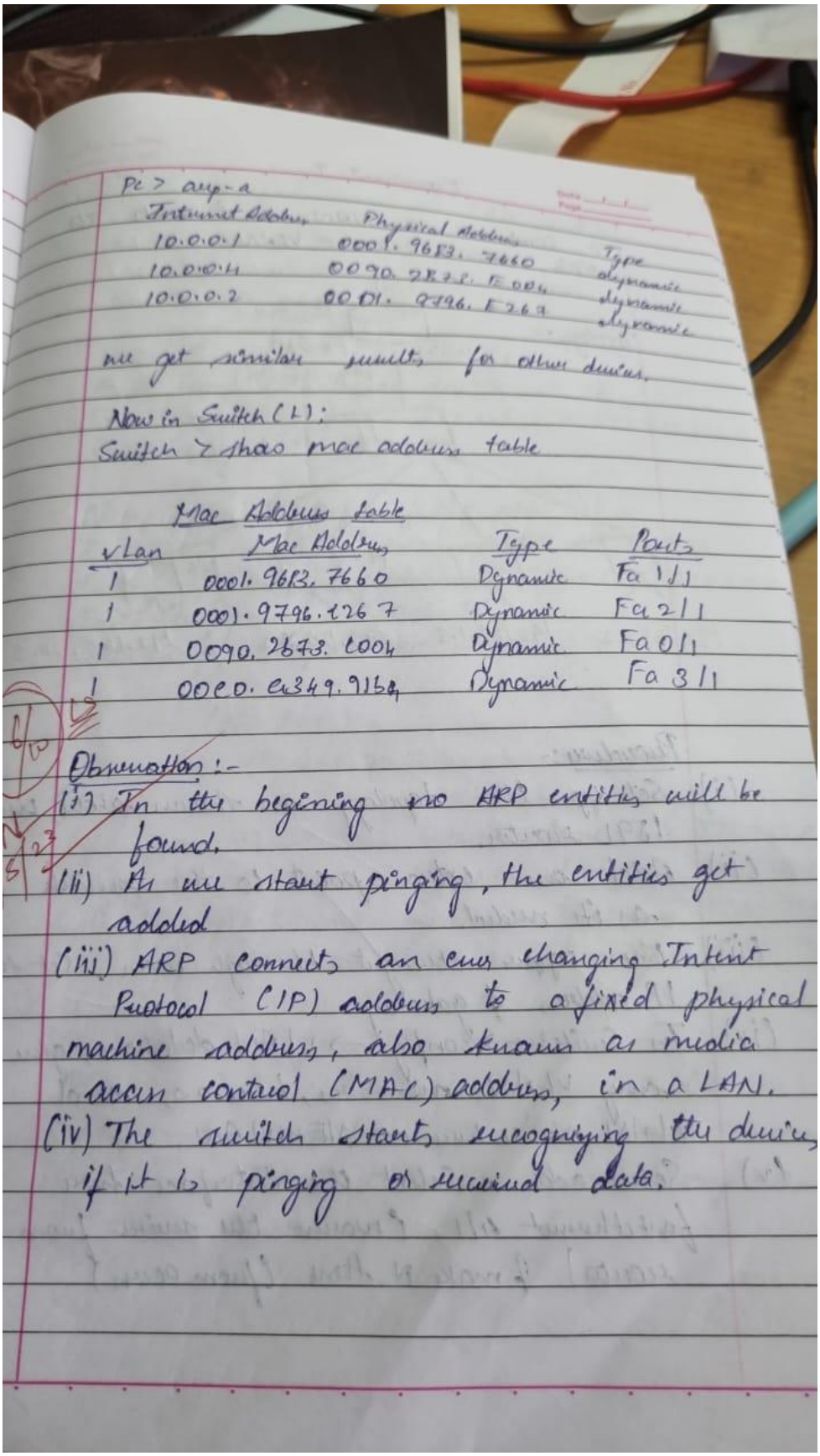
PC
PC > arp -a
No ARP entries found
PC > ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data

Reply from 10.0.0.1: bytes = 32 time = 8ms TTL = 120
Reply from 10.0.0.1: bytes = 32 time = 4ms TTL = 120
Reply from 10.0.0.1: bytes = 32 time = 4ms TTL = 120
Reply from 10.0.0.1 bytes = 32 time = 4ms TTL = 120

Ping Statistics for 10.0.0.1
Packets sent = 4, Received = 4, Lost = 0 (0% loss)
Approximate round trip times in milliseconds:
Minimum = 4ms, Maximum = 0ms, Average = 5ms

Now Similarly, ping all other devices for same result.



PC > arp -a

Internet Address	Physical Address	Type
10.0.0.1	0001.9683.7660	dynamic
10.0.0.4	0090.2873.E004	dynamic
10.0.0.2	0001.9796.E267	dynamic

We get similar results for other devices.

Now in Switch (1):

Switch > show mac address table

Mac Address table

Vlan	Mac Address	Type	Ports
1	0001.9683.7660	Dynamic	Fa 1/1
1	0001.9796.E267	Dynamic	Fa 2/1
1	0090.2873.E004	Dynamic	Fa 0/1
1	00E0.e349.9154	Dynamic	Fa 3/1

Observation :-

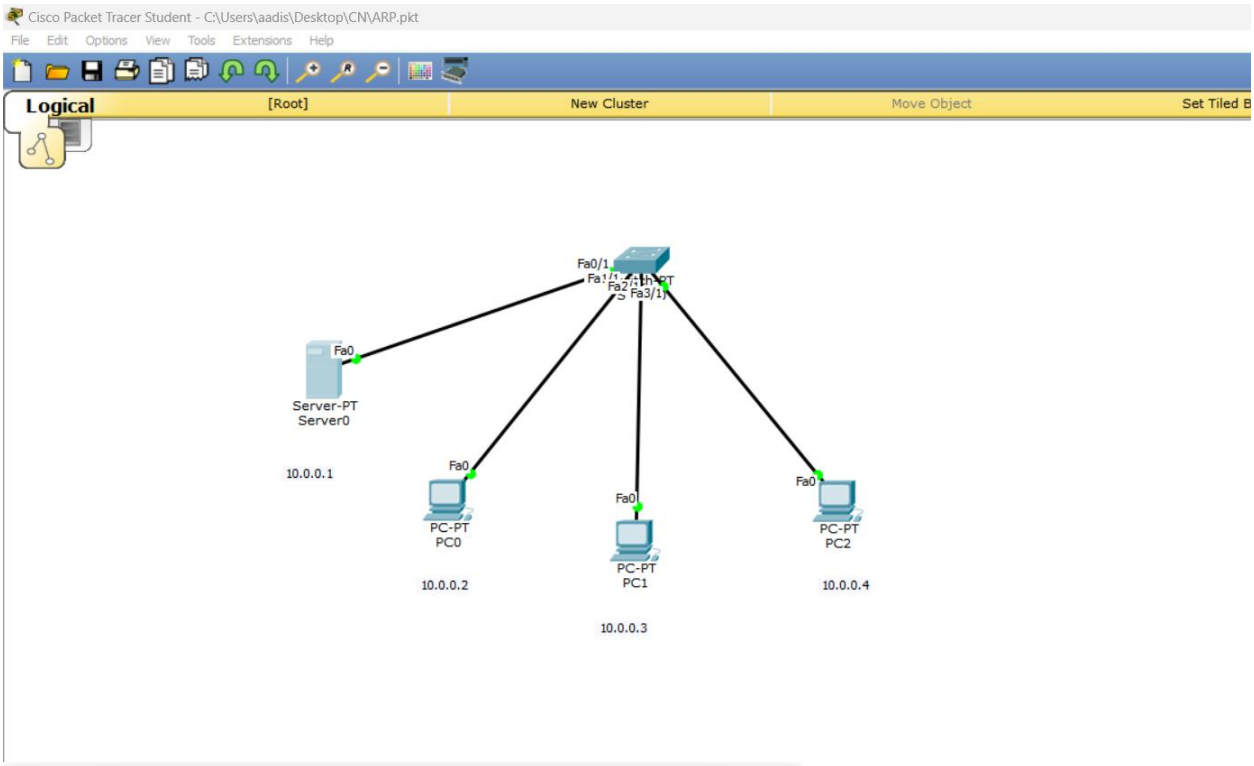
(i) In the begining no ARP entities will be found.

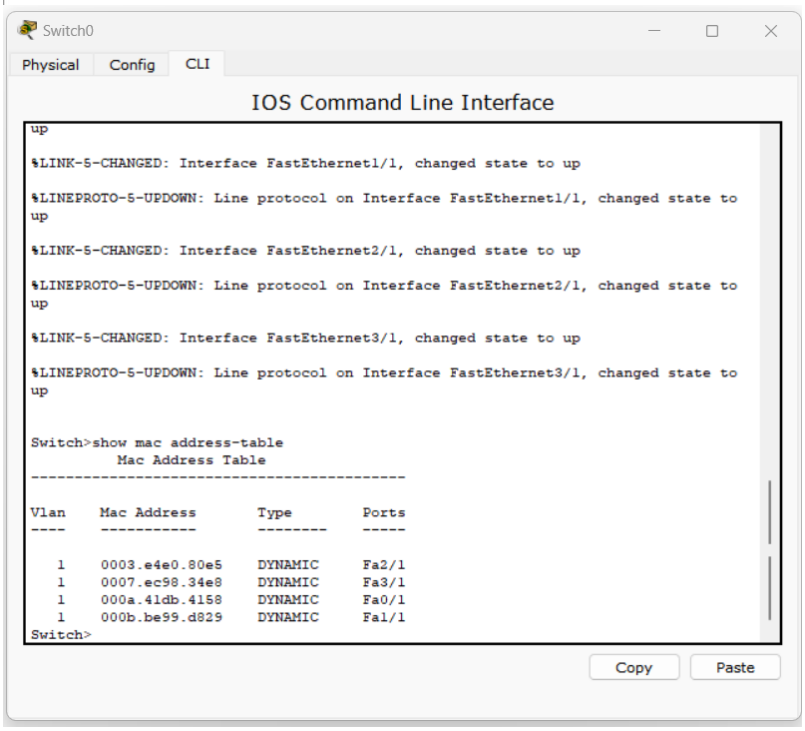
(ii) As we start pinging, the entities get added

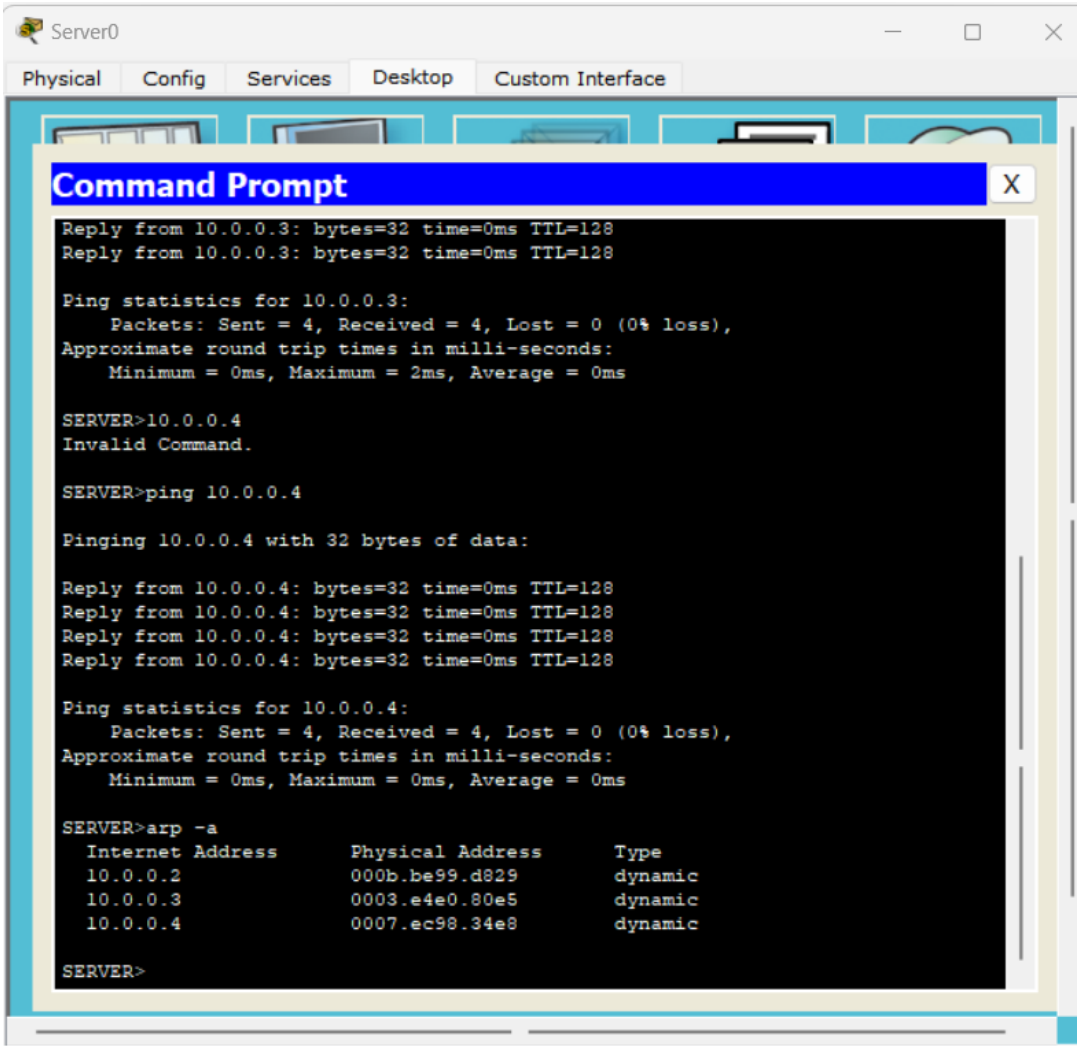
(iii) ARP connects an ever changing Internet Protocol (IP) address to a fixed physical machine address, also known as media access control (MAC) address, in a LAN.

(iv) The switch starts recognizing the device, if it is pinging or received data.

Result:


Cisco Packet Tracer Student - C:\Users\aadis\Desktop\CN\ARP.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled B
Fa0/1
Fa3/1
Fa0
Server-PT
Server0
10.0.0.1
Fa0
PC-PT
PC0
10.0.0.2
Fa0
PC-PT
PC1
10.0.0.3
Fa0
PC-PT
PC2
10.0.0.4


Switch0
Physical
Config
CLI
IOS Command Line Interface
up
%LINK-5-CHANGED: Interface FastEthernet1/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/1, changed state to up
%LINK-5-CHANGED: Interface FastEthernet2/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet2/1, changed state to up
%LINK-5-CHANGED: Interface FastEthernet3/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet3/1, changed state to up
Switch>show mac address-table
Mac Address Table
Vlan Mac Address Type Ports
1 0003.e4e0.80e5 DYNAMIC Fa2/1
1 0007.ec98.34e8 DYNAMIC Fa3/1
1 000a.41db.4158 DYNAMIC Fa0/1
1 000b.be99.d829 DYNAMIC Fa1/1
Switch>
Copy
Paste

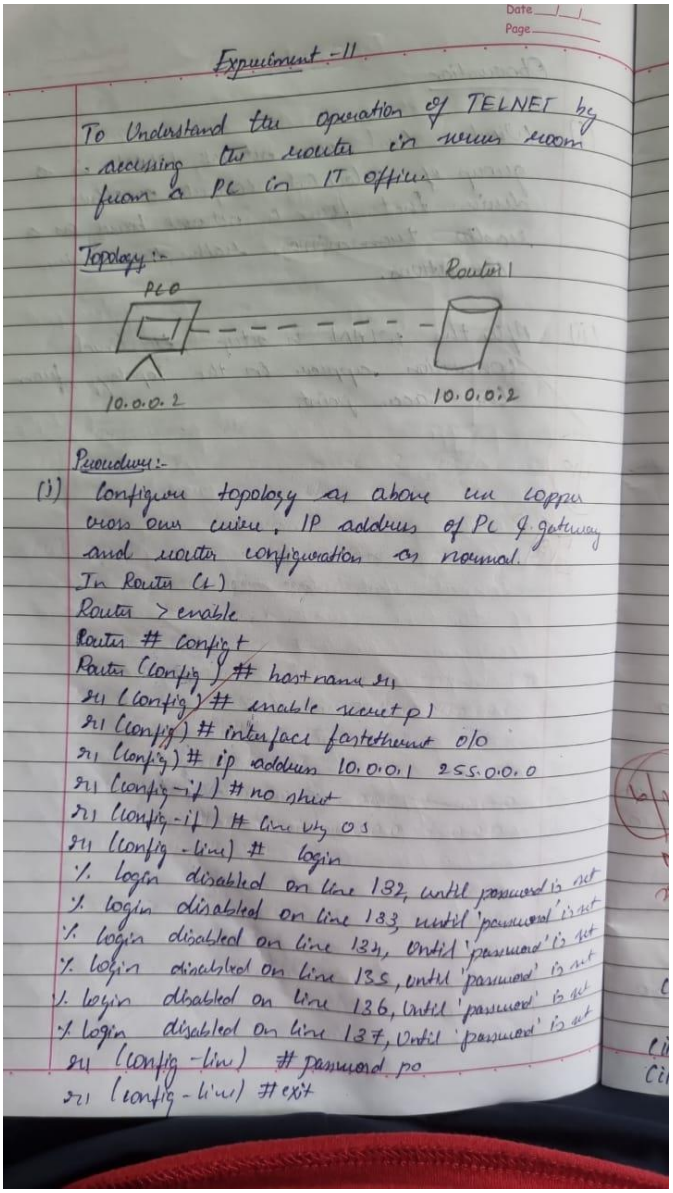

Server0
Physical
Config
Services
Desktop
Custom Interface
Command Prompt
X
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Reply from 10.0.0.3: bytes=32 time=0ms TTL=128
Ping statistics for 10.0.0.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 2ms, Average = 0ms
SERVER>10.0.0.4
Invalid Command.
SERVER>ping 10.0.0.4
Pinging 10.0.0.4 with 32 bytes of data:
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Ping statistics for 10.0.0.4:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
SERVER>arp -a
Internet Address Physical Address Type
10.0.0.2 000b.be99.d829 dynamic
10.0.0.3 0003.e4e0.80e5 dynamic
10.0.0.4 0007.ec98.34e8 dynamic
SERVER>

EXPERIMENT 10

AIM:

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

OBSERVATION:



Date / /
Page

R1 #wr
Building configuration
(OK)
R1#

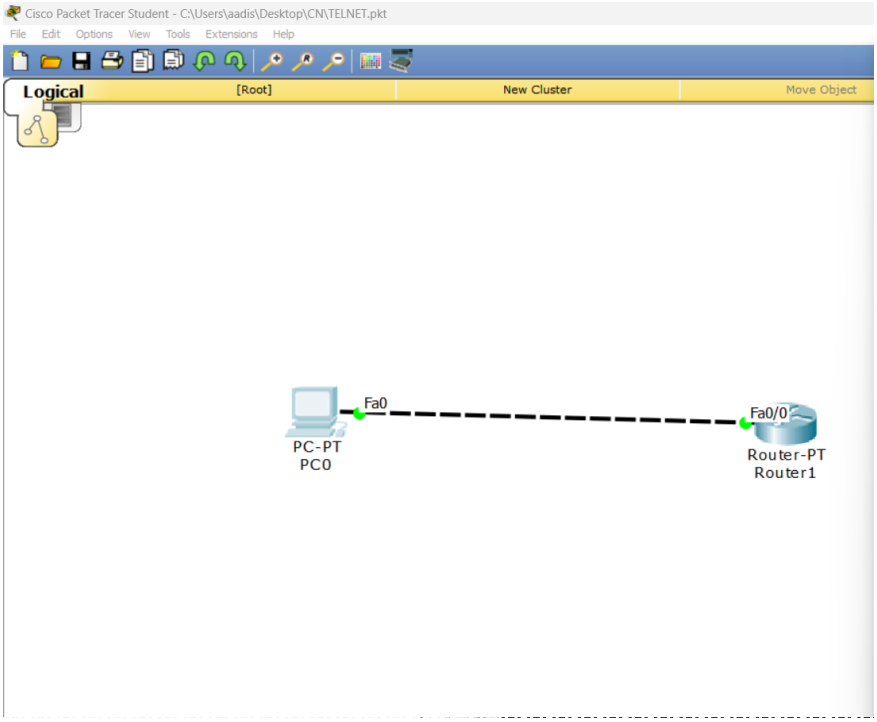
Result:- In PC0
PC > ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data
Reply from 10.0.0.1 bytes=32 time=0ms TTL=255
Reply from 10.0.0.1 bytes=32 time=0ms TTL=255
Reply from 10.0.0.1 bytes=32 time=0ms TTL=255
Reply from 10.0.0.1 bytes=32 time=0ms TTL=255

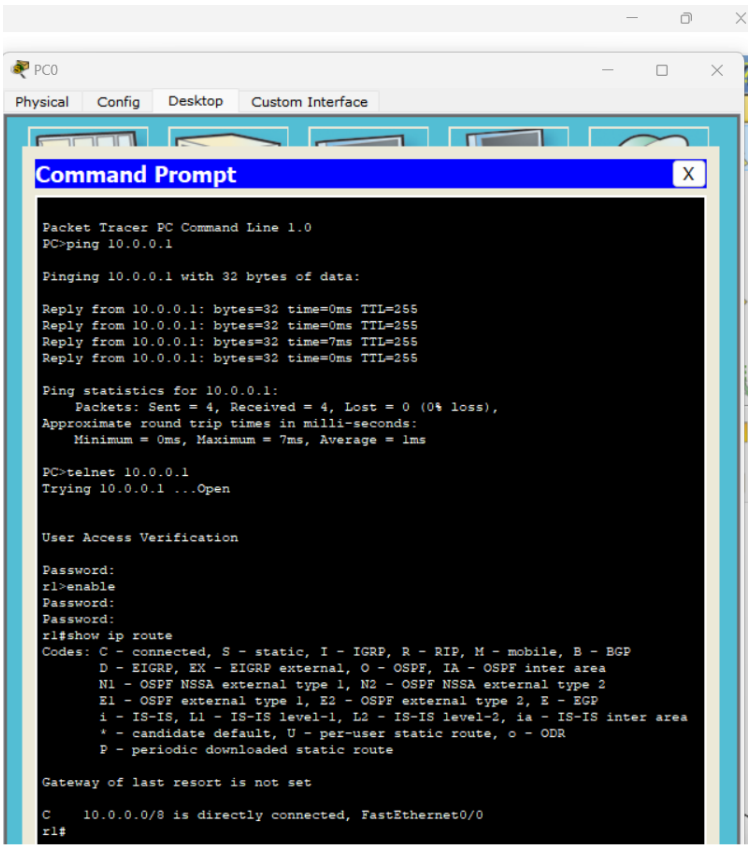
Ping statistics for 10.0.0.1
Packets sent=4, Received=4, lost=0 (0% loss)
Approximate round trip time in ms:
Minimum=0ms, Maximum=0ms, Average=0ms

PC > telnet 10.0.0.1
Trying 10.0.0.1 ... open
User access verification
password: (typed p0)
R1> enable
password: (typed p1)
R1# show ip route
Codes:
Gateway of last resort is not set
10.0.0.0/8 is directly connected, FastEthernet 0/0
R1#

Observation
(i) TELNET is used by terminal emulation programs that allow you to remote host.
(ii) we logged int 10.0.0.1 through 10.0.0.2 device
(iii) The password when typed isn't visible.

Result:


Cisco Packet Tracer Student - C:\Users\aadis\Desktop\CN\TELNET.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Fa0
PC-PT
PC0
Fa0/0
Router-PT
Router1

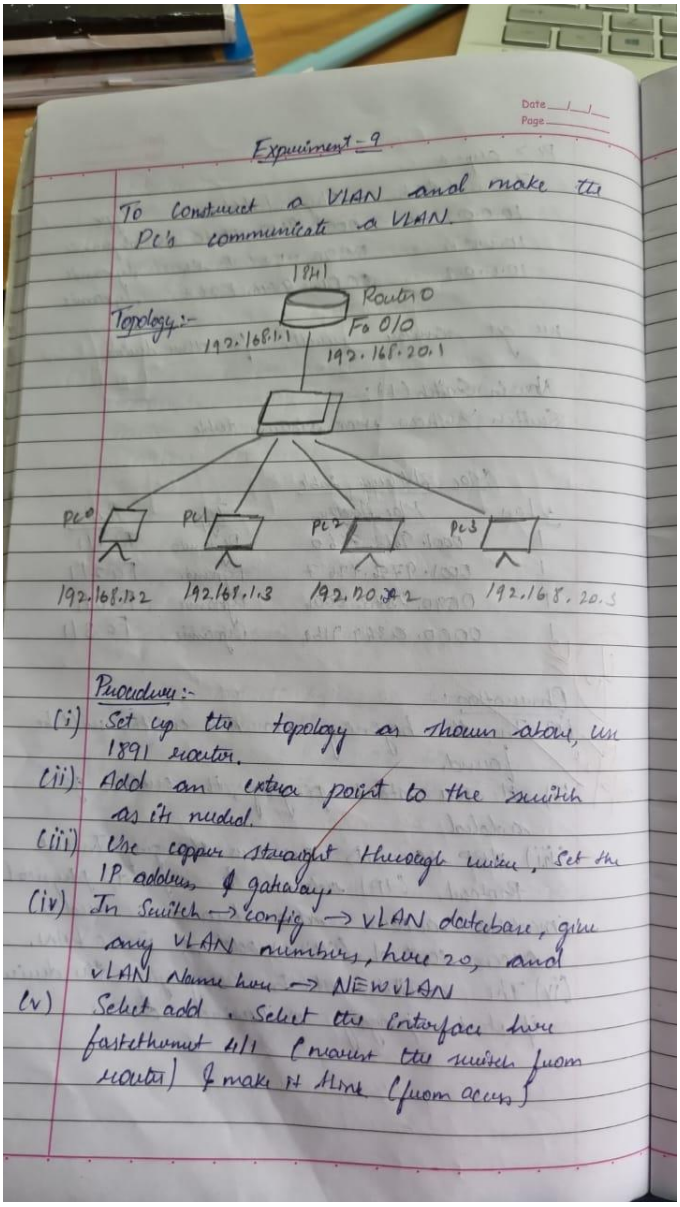

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=0ms TTL=255
Reply from 10.0.0.1: bytes=32 time=7ms TTL=255
Reply from 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 = 7ms, Average = 1ms
PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open
User Access Verification
Password:
r1>enable
Password:
Password:
r1#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
r1#

EXPERIMENT 11

AIM:

To construct a VLAN and make the PC's communicate among a VLAN.

OBSERVATION:



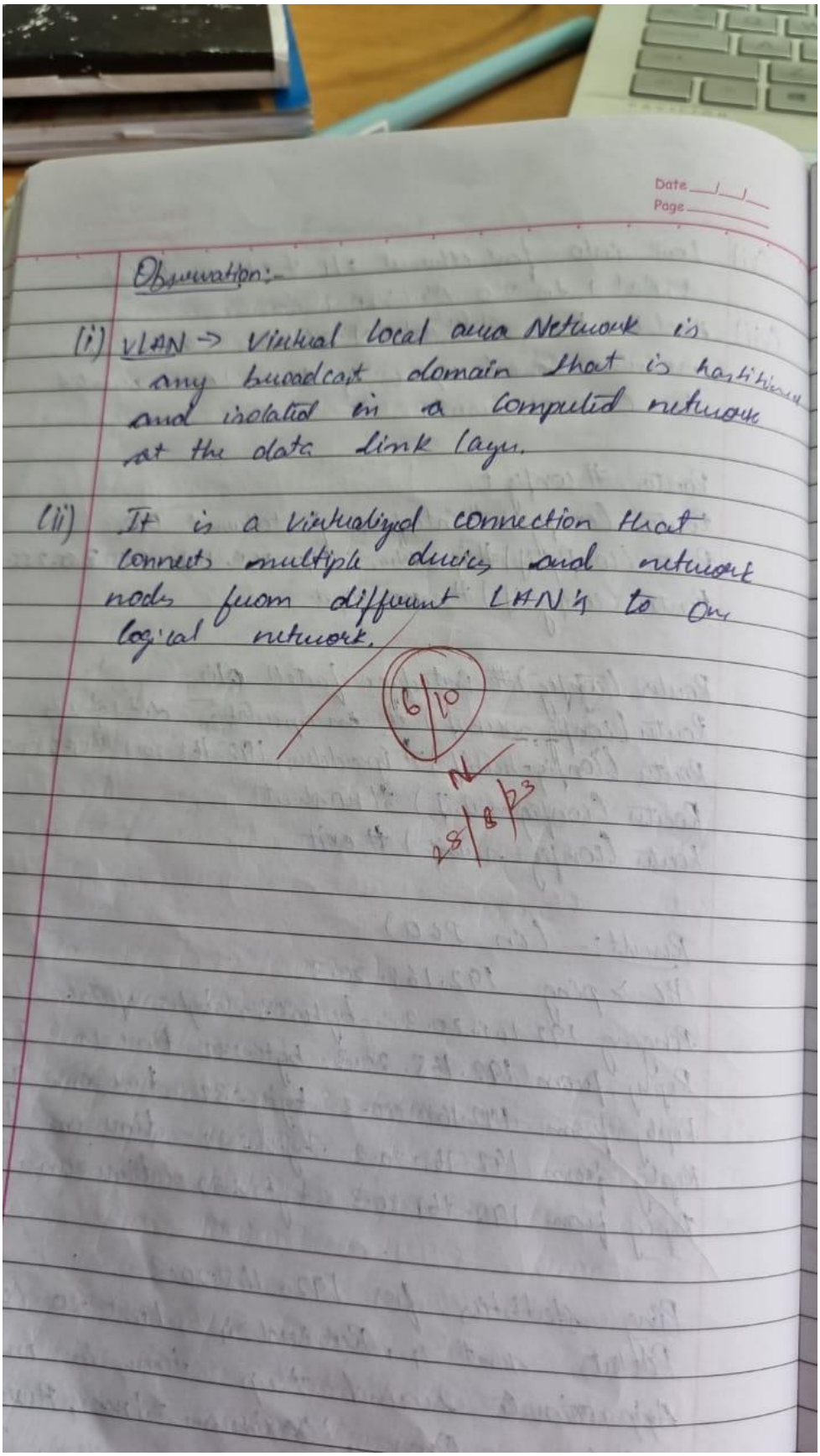
(vi) Look into fast ethernet 0/1 & 3/1 and set VLAN 1 to 20 New VLAN

(vii) And in Router sight VLAN database, enter the number and name of the VLAN created In CLI of router

```
Router # config t
Router (config) # interface fastethernet 0/0
Router (config-if) # ip address 192.168.1.1 255.255.255.0
Router (config-if) # no shut

Router (config) # interface fastet 0/0
Router (config-subif) # encapsulation dot1q 20
Router (config-subif) # ip address 192.168.20.1 255.255.255.
Router (config-subif) # no shut
Router (config-subif) # exit
```

Result:- (on PC0)

```
PC > ping 192.168.20.3
Ponging 192.168.20.3 : bytes:32 bytes of data
Reply from 192.168.20.3 : bytes:32 time=1ms TTL=12
Reply from 192.168.20.3 bytes:32 time=0ms TTL=1
Reply from 192.168.20.3 bytes:32 time=0ms TTL=1
Reply from 192.161.20.3 bytes:32 time=0ms TTL=

Ping statistics for 192.168.20.3
Packets sent=4, Received=4, Lost=0 (0% loss)
Approximate round trip times in millisecon
minimum=0ms, Maximum=1ms, Average=0m
```



Date __/__/__
Page

Observation:-

(i) VLAN → Virtual local area Network is any broadcast domain that is partitioned and isolated in a computer network at the data link layer.

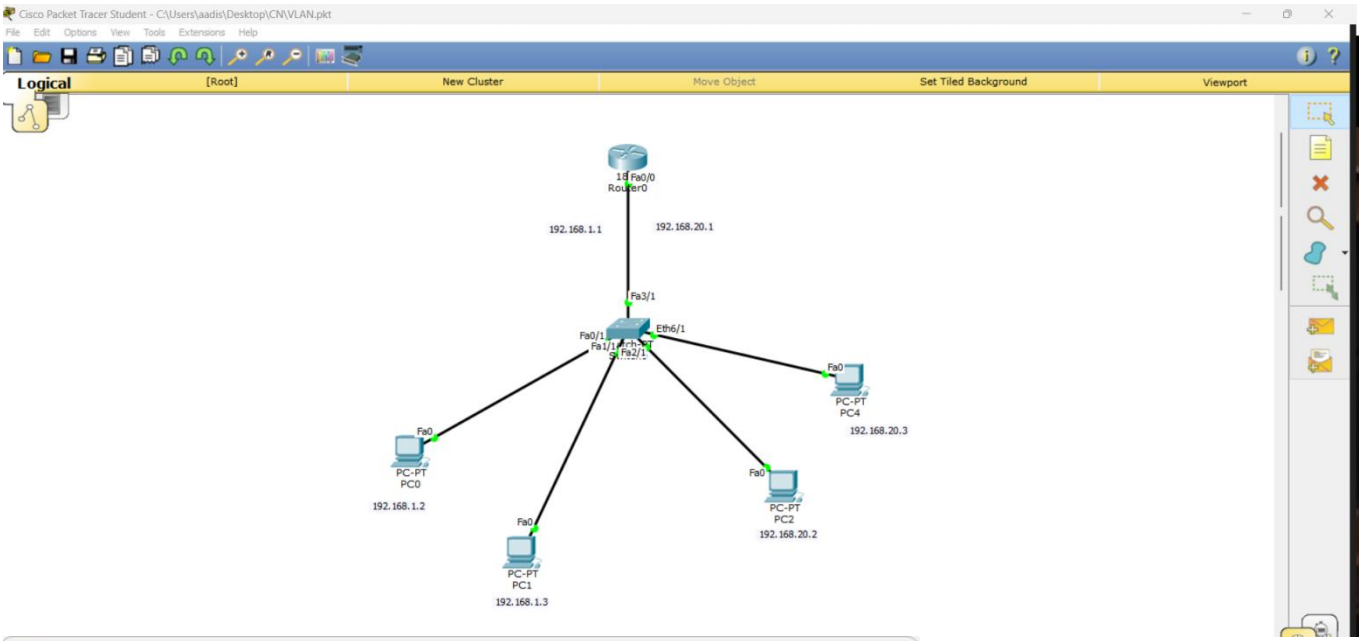
(ii) It is a virtualized connection that connects multiple devices and network nodes from different LAN's to one logical network.

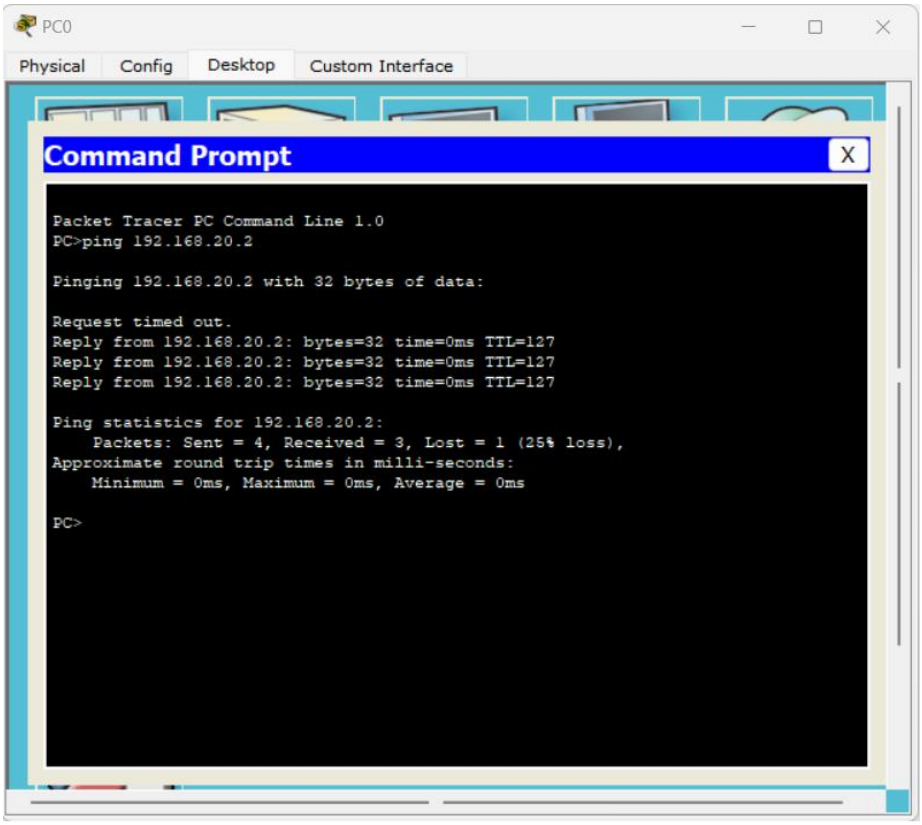
6/10

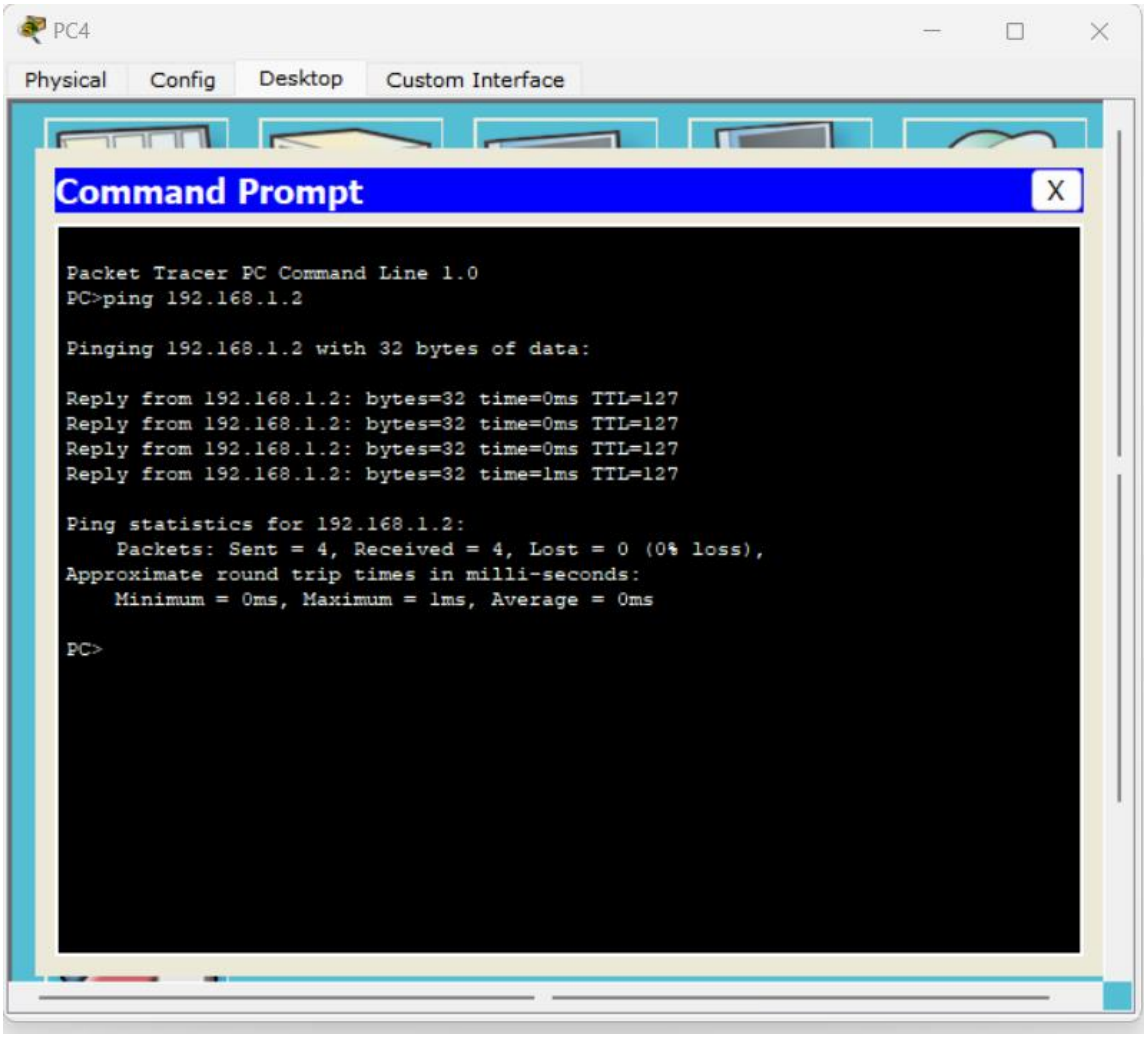
N

28/8/23

Result:


Cisco Packet Tracer Student - C:\Users\aadis\Desktop\CN\VLAN.pkt
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router0
Fa0/0
192.168.1.1
192.168.20.1
Fa3/1
Eth6/1
Fa0/1
Fa1/1
Fa2/1
PC-PT PC0
192.168.1.2
PC-PT PC1
192.168.1.3
PC-PT PC2
192.168.20.2
PC-PT PC4
192.168.20.3


PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 192.168.20.2
Pinging 192.168.20.2 with 32 bytes of data:
Request timed out.
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127
Ping statistics for 192.168.20.2:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>


PC4
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 192.168.1.2
Pinging 192.168.1.2 with 32 bytes of data:
Reply from 192.168.1.2: bytes=32 time=0ms TTL=127
Reply from 192.168.1.2: bytes=32 time=0ms TTL=127
Reply from 192.168.1.2: bytes=32 time=0ms TTL=127
Reply from 192.168.1.2: bytes=32 time=1ms TTL=127
Ping statistics for 192.168.1.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
PC>

EXPERIMENT 12

AIM:

To construct a WLAN and make the nodes communicate wirelessly.

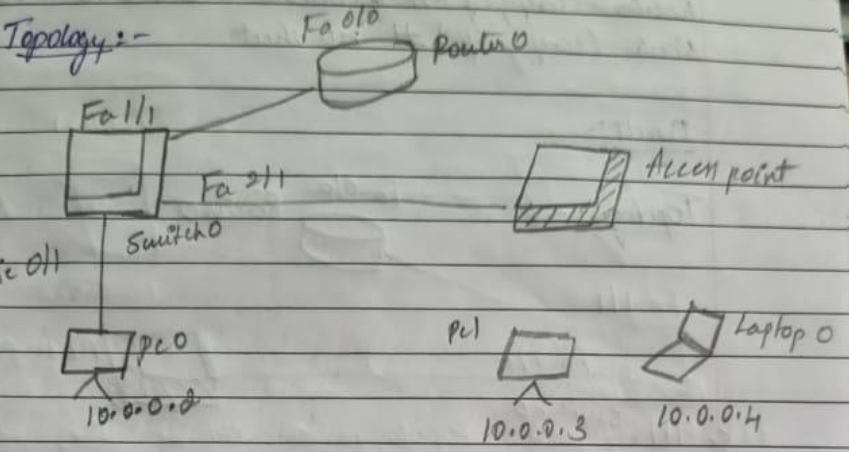
OBSERVATION:

Experiment - 10

Date __/__/__
Page

To construct a WLAN and make the nodes communicate wirelessly.

Topology:-



Procedure:-

(i) Construct above topology. Use Access point PT connect that to router SET the IP address of the PC connected with wire and configure router 1.

(ii) Configure access point 1 → port 1 → SSID Name - any - name (WLAN has) Select WEP and give 10 digit hex 1234567890 hex.

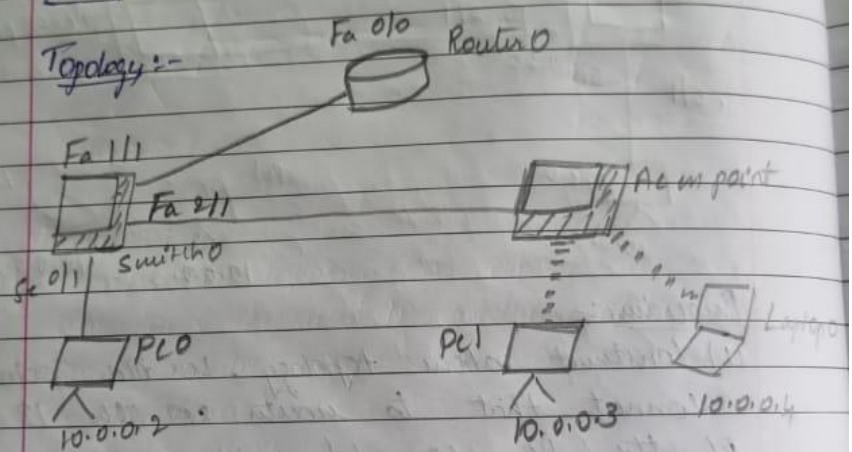
(iii) To configure PC4 & laptop wirelessly, switch off the device. Drag the existing PT-Host-NM-1AM to the component listed in the LHS. Drag WMP300N wireless interface to the empty port & switch on the device.

(iv) Now, in the config tab, a new wireless interface would have been added, configure SSID, WEP, WEP4, IP address & gateway (as normally done) to the device.

Router > enable
Router # config t
Router (config) # interface fastethernet 0/0
Router (config-if) # ip address 10.0.0.10 255.0.0.0
Router (config-if) # no shut

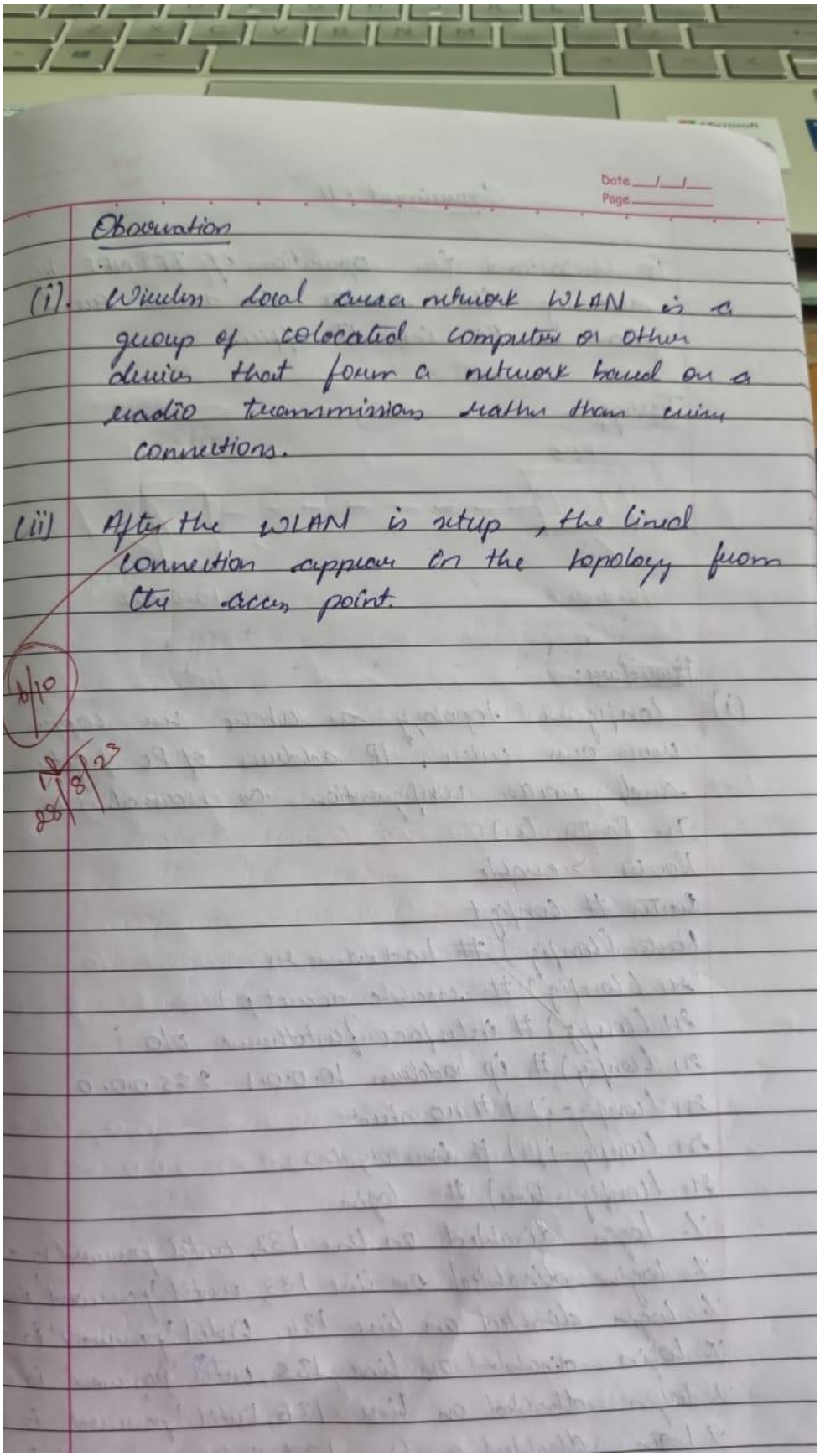
Result:-

Topology:-



In PC 0 (10.0.0.2)
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 = 1ms TTL =
Reply from 10.0.0.3 bytes = 32 time = 13ms TTL:
Reply from 10.0.0.3 bytes = 32 time = 6ms TTL:
Reply from 10.0.0.3 bytes = 32 time = 0ms TTL:

Ping statistics for 10.0.0.2
Packets sent = 4, Received = 4, Lost = 0 (0% loss)
Approximate round trip times in milli-seconds
Minimum = 6ms, Maximum = 21 ms, Avg =



Date / /
Page

Observation

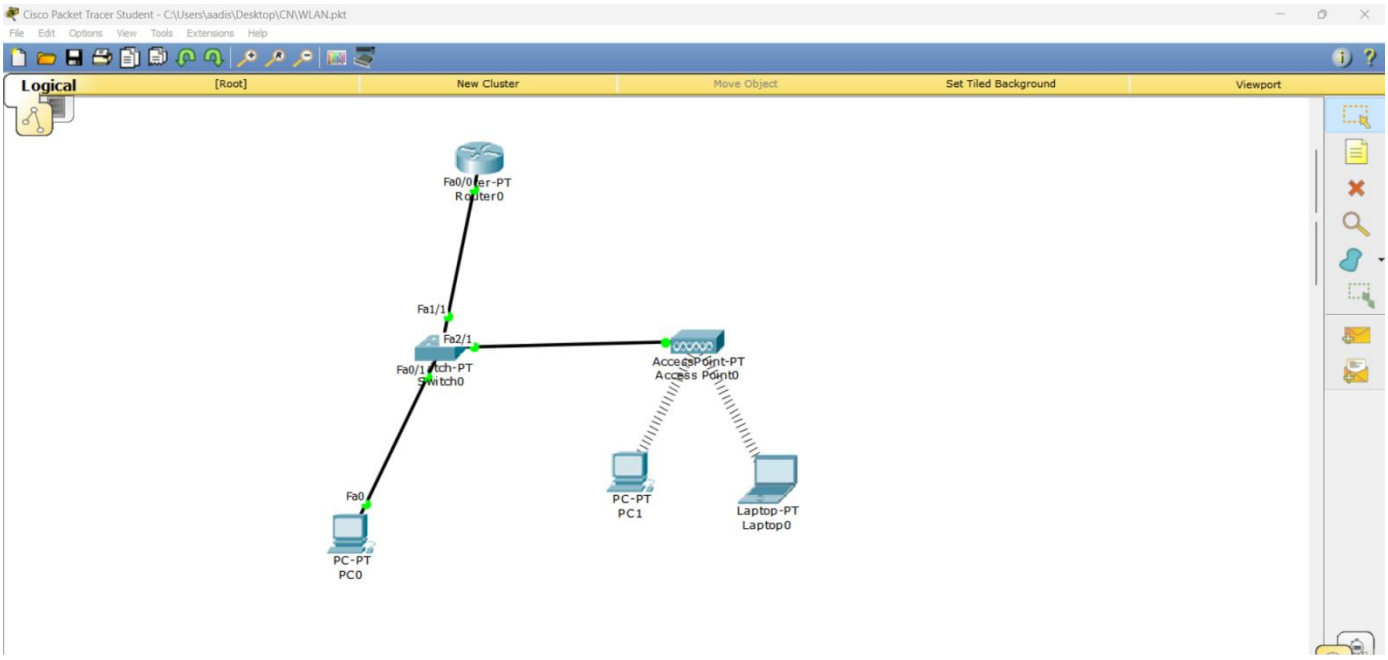
(i). Wireless local area network WLAN is a group of colocated computers or other devices that form a network based on a radio transmission rather than wire connections.

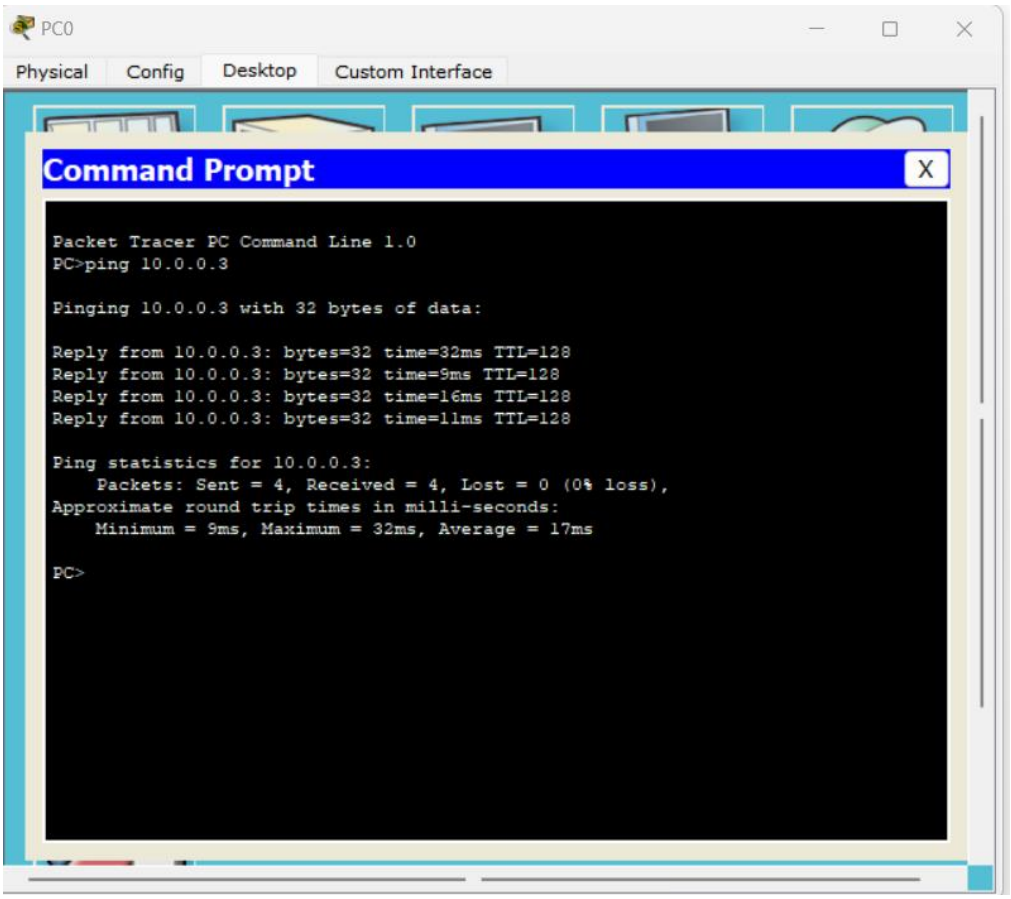
(ii) After the WLAN is setup, the final connection appear on the topology from the access point.

10/10

1 2 3
8
88

Result:


Cisco Packet Tracer Student - C:\Users\aadis\Desktop\CN\WLAN.pkt
File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Fa0/0er-PT
Router0
Fa1/1
Fa2/1
Fa0/1tch-PT
Switch0
AccessPoint-PT
Access Point0
Fa0
PC-PT
PC0
PC-PT
PC1
Laptop-PT
Laptop0

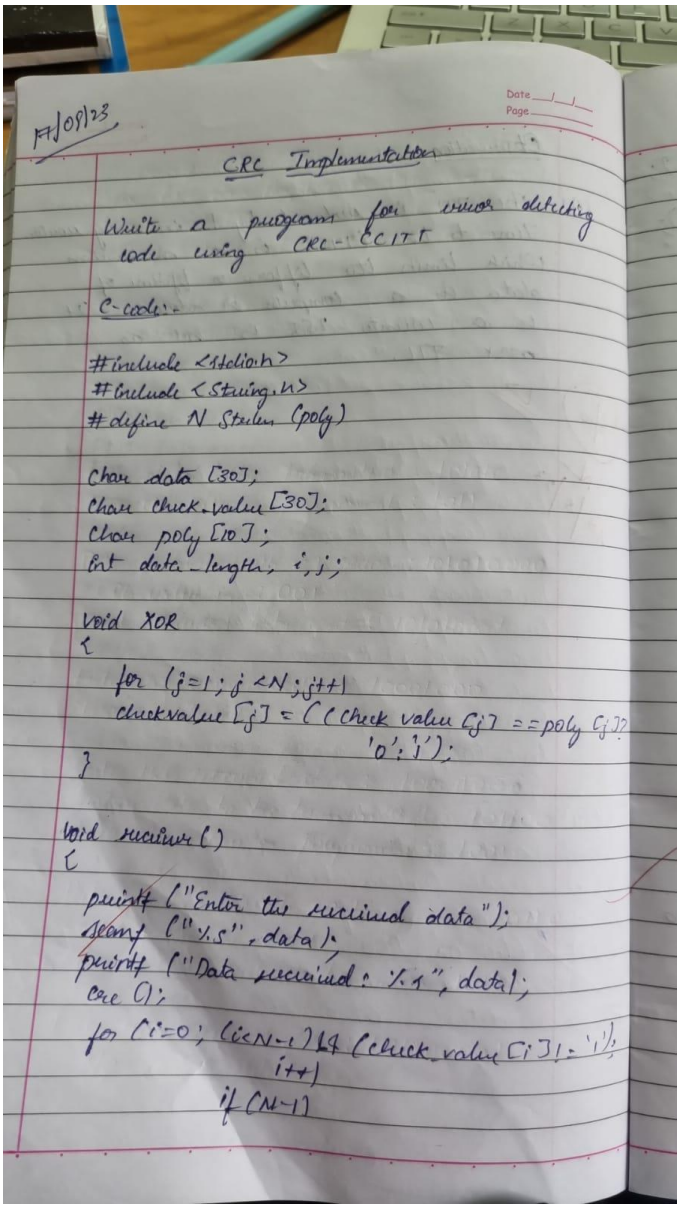

PC0
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data:
Reply from 10.0.0.3: bytes=32 time=32ms TTL=128
Reply from 10.0.0.3: bytes=32 time=9ms TTL=128
Reply from 10.0.0.3: bytes=32 time=16ms TTL=128
Reply from 10.0.0.3: bytes=32 time=11ms 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 = 9ms, Maximum = 32ms, Average = 17ms
PC>

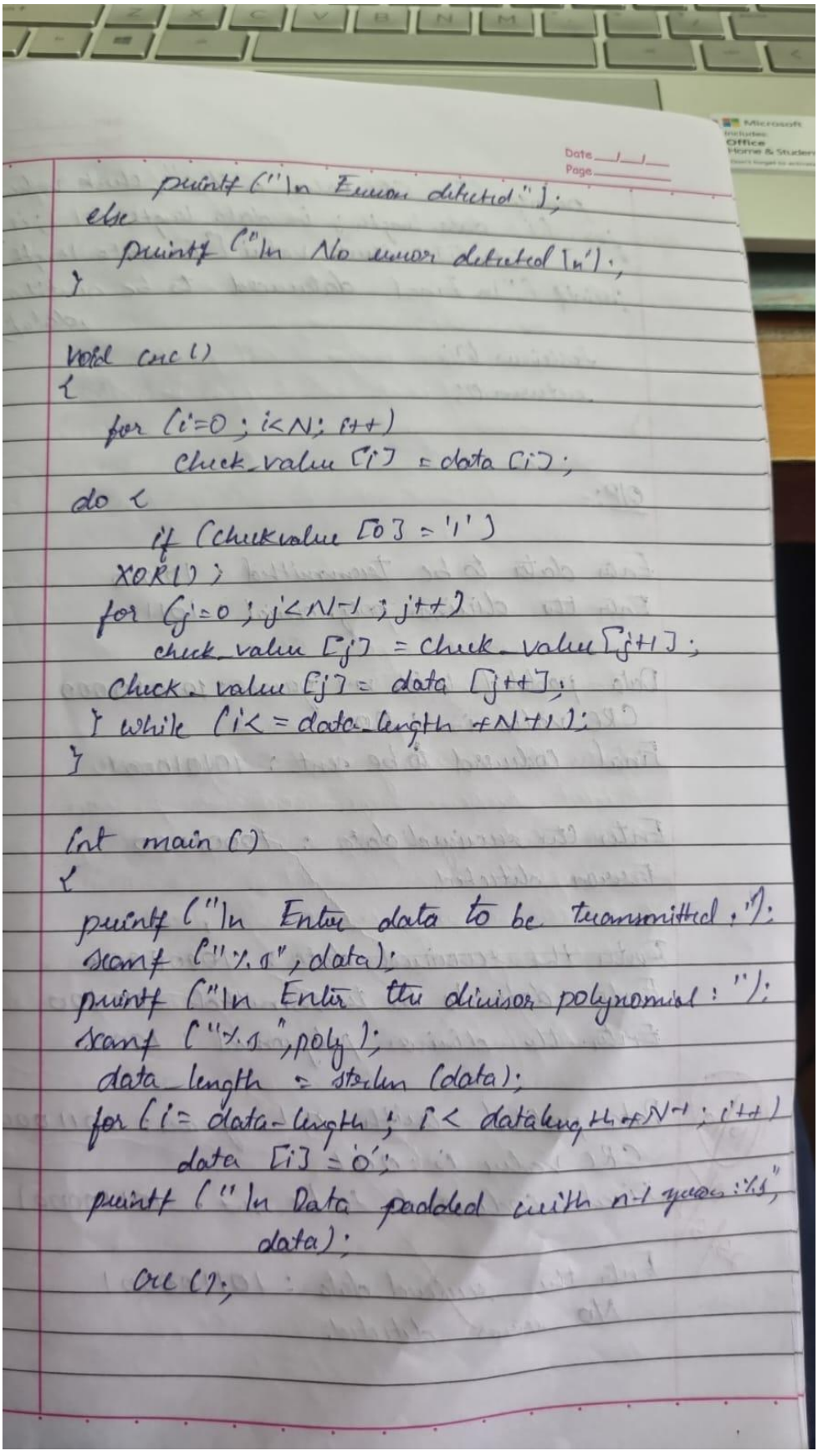
CYCLE 2

AIM:

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

PROGRAM:





```
        printf("\n Error detected.");
    else
        printf("\n No error detected\n");
}

void crc()
{
    for(i=0; i<N; i++)
        check_value[i] = data[i];
    do {
        if(checkvalue[0] == '1')
        XOR();
        for(j=0; j<N-1; j++)
            check_value[j] = check_value[j+1];
        check_value[j] = data[j++];
    } while(i<=data_length+N+1);
}

int main()
{
    printf("\n Enter data to be transmitted: ");
    scanf("%s", data);
    printf("\n 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("\n Data padded with n-1 zeros: %s",
            data);
    crc();
```

```
printf("\n CRC value is %s", check_value);
for (i = data_length; i < data_length + N - 1; i++)
    data[i] = check_value[i - data_length];
printf("\n Final datawword to be sent: %s",
                                    , data);

    receiver();
    return 0;
}
```

O/P:-

Enter data to be transmitted : 101010
Enter the divisor polynomial : 1011

Data padded with n-1 zeros : 101010000
CRC value is : 001
Final codeword to be sent : 101010001

Enter the received data : 10001000
Error detected.

~~Enter the received data : 10001000~~
Enter data to be transmitted: 101100
Enter the divisor polynomial : 1001

Data padded with n-1 zeros : 101100000
CRC value is : 001
Final codeword to be sent : 101100001

Enter the received data : 101100001
No errors detected.

PROGRAM:

```
#include<stdio.h>
#include<string.h>
#define N strlen(divisor)
char data[28];
char rem[28];
char divisor[10];
int dlength,i,j;
void XOR(){
   for(j = 1;j < N; j++)
   rem[j] = (( rem[j] == divisor[j])?'0':'1');

}

void receiver(){

   printf("Enter the received data: ");
   scanf("%s", data);
   printf("\n\n");
   printf("Data received: %s", data);

   crc();

   for(i=0;(i<N-1) && (rem[i]!='1');i++);
      if(i<N-1)
         printf("\nError detected\n\n");
      else
         printf("\nNo error detected\n\n");
}

void crc(){

   for(i=0;i<N;i++)
      rem[i]=data[i];
   do{

      if(rem[0]=='1')
         XOR();
```

```
        for(j=0;j<N-1;j++)
            rem[j]=rem[j+1];

        rem[j]=data[i++];
    }
    while(i<=dlength+16);

}

int main()
{  int c=0;

    printf("\nEnter data to be transmitted: ");
    scanf("%s",data);
    printf("\n Enter the Divisor: ");
    scanf("%s",divisor);
    dlength=strlen(data);
    for(i=dlength;i<dlength+16;i++)
        data[i]='0';
    printf("\n");
    printf("\n Data padded with n-1 zeros : %s",data);
    printf("\n");
    crc();
    printf("\nCRC or Check value is : %s",rem);
    printf("\n rem strlen is : %d ", strlen(rem));
    for(i=dlength+13;i<dlength+16;i++)
      {    printf("\n %s",data);
           data[i]= rem[c++];
           }
    printf("\n");

    printf("\n Final data to be sent : %s",data);
    printf("\n\n");

    receiver();
        return 0;
}
```

Output:

```
Enter data to be transmitted: 1001101

 Enter the Divisor: 1011


 Data padded with n-1 zeros : 10011010000000000000000

CRC or Check value is : 111
 rem strlen is : 3
 10011010000000000000000
 10011010000000000000100
 10011010000000000000110

 Final data to be sent : 10011010000000000000111

Enter the received data: 10011010000000000000111


Data received: 10011010000000000000111
No error detected
```

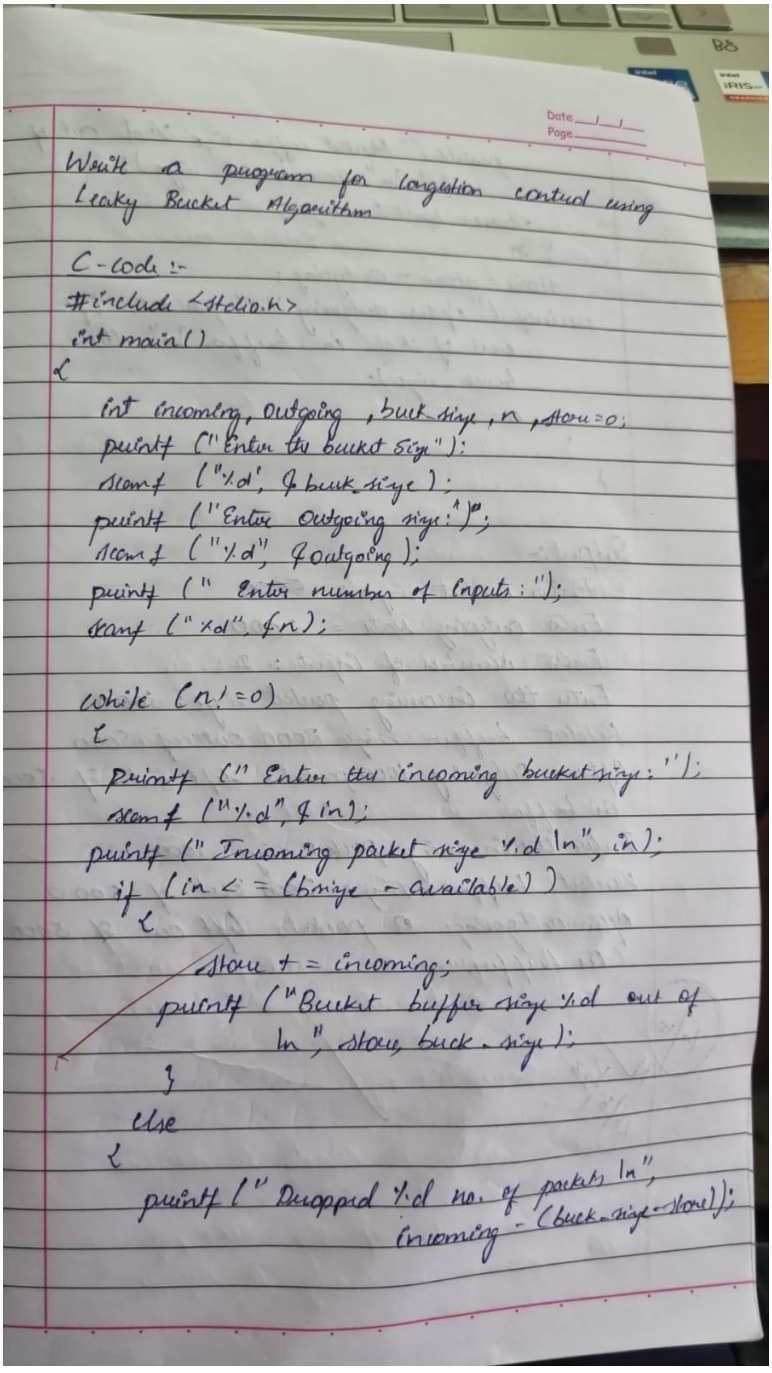
```
Enter data to be transmitted: 1001101

 Enter the Divisor: 1011


 Data padded with n-1 zeros : 10011010000000000000000

CRC or Check value is : 111
 rem strlen is : 3
 10011010000000000000000
 10011010000000000000100
 10011010000000000000110

 Final data to be sent : 10011010000000000000111

Enter the received data: 10011010000000000001111


Data received: 10011010000000000001111
Error detected



...Program finished with exit code 0
Press ENTER to exit console.
```

CYCLE 2

AIM:

Write a program for congestion control using Leaky bucket algorithm.

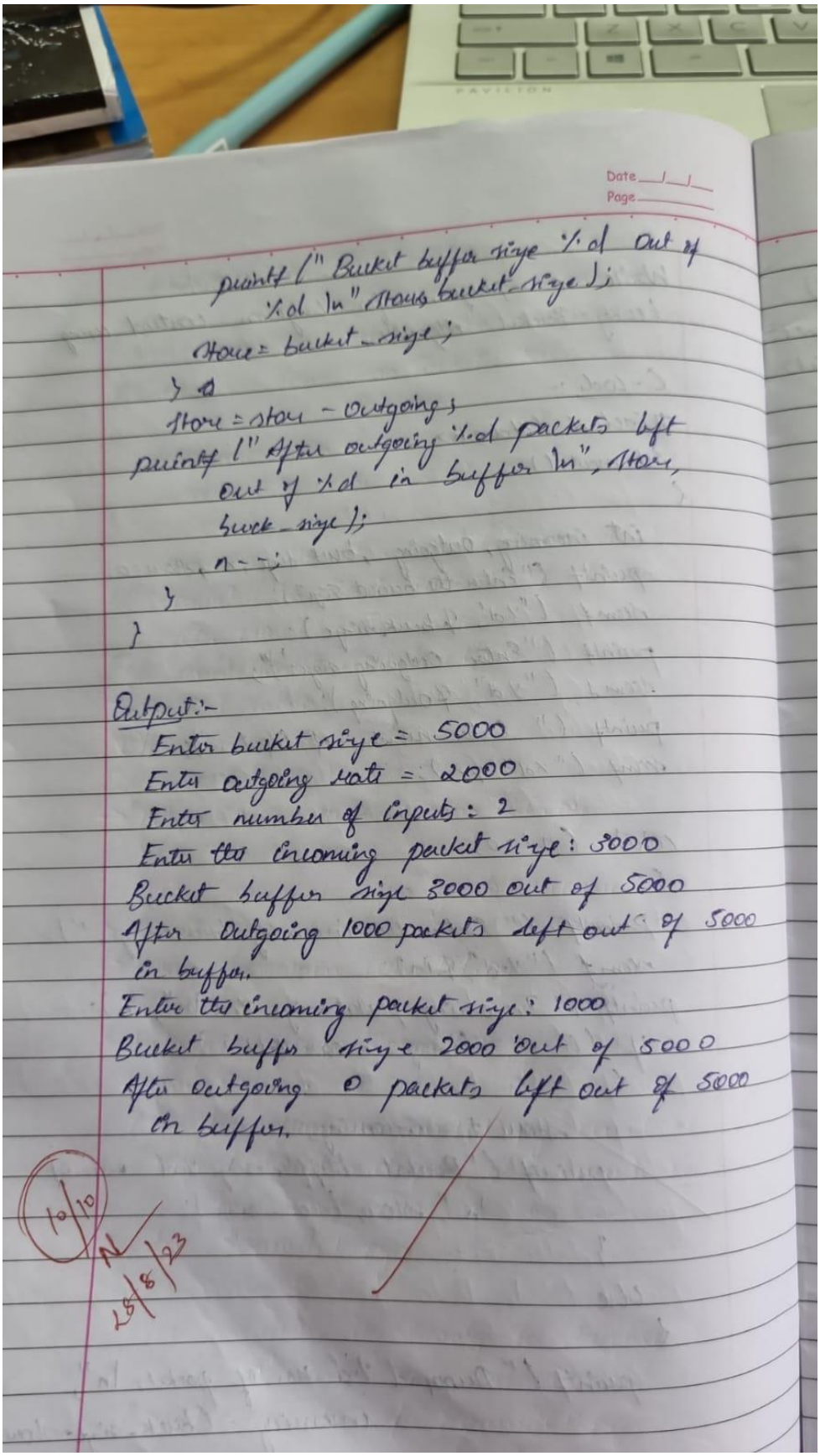
PROGRAM:



Write a program for congestion control using Leaky Bucket Algorithm

C-code :-

```
#include <stdio.h>
int main()
{
  int incoming, outgoing, buck_size, n, store=0;
  printf("Enter the bucket size");
  scanf("%d", &buck_size);
  printf("Enter outgoing size:");
  scanf("%d", &outgoing);
  printf("Enter number of inputs:");
  scanf("%d", &n);

  while(n!=0)
  {
    printf("Enter the incoming bucket size:");
    scanf("%d", &in);
    printf("Incoming packet size %d \n", in);
    if(in <= (bsize - available))
    {
      store += incoming;
      printf("Bucket buffer size %d out of
              %n", store, buck_size);
    }
    else
    {
      printf("Dropped %d no. of packets \n",
              incoming - (buck_size - store));
```



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

Output:-

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

10/10
AL
25/8/23

PROGRAM:

```
#include<stdio.h>

int main(){
    int in, out, bsize, n, available = 0;
    printf("Enter the bucket size: ");
    scanf("%d", &bsize);
    printf("Enter the outgoing rate: ");
    scanf("%d", &out);
    printf("Enter the no of inputs: ");
    scanf("%d", &n);

    while (n != 0) {
        printf("Enter the incoming packet size : ");
        scanf("%d", &in);
        printf("Incoming packet size %d\n", in);
        if (in <= (bsize - available)){
            available += in;
            printf("Bucket buffer size %d out of %d\n", available, bsize);
        } else {
            printf("Dropped %d no of packets\n", in - (bsize - available));
            printf("Bucket buffer size %d out of %d\n", available, bsize);
            available = bsize;
        }
        available = available - out;
        printf("After outgoing %d packets left out of %d in buffer\n", available, bsize);
        n--;
    }
}
```

Output:

```
Enter the bucket size: 1000
Enter the outgoing rate: 200
Enter the no of inputs: 6
Enter the incoming packet size : 200
Incoming packet size 200
Bucket buffer size 200 out of 1000
After outgoing 0 packets left out of 1000 in buffer
Enter the incoming packet size : 200
Incoming packet size 200
Bucket buffer size 200 out of 1000
After outgoing 0 packets left out of 1000 in buffer
Enter the incoming packet size : 400
Incoming packet size 400
Bucket buffer size 400 out of 1000
After outgoing 200 packets left out of 1000 in buffer
Enter the incoming packet size : 450
Incoming packet size 450
Bucket buffer size 650 out of 1000
After outgoing 450 packets left out of 1000 in buffer
Enter the incoming packet size : 500
Incoming packet size 500
Bucket buffer size 950 out of 1000
After outgoing 750 packets left out of 1000 in buffer
Enter the incoming packet size : 100
Incoming packet size 100
Bucket buffer size 850 out of 1000
```

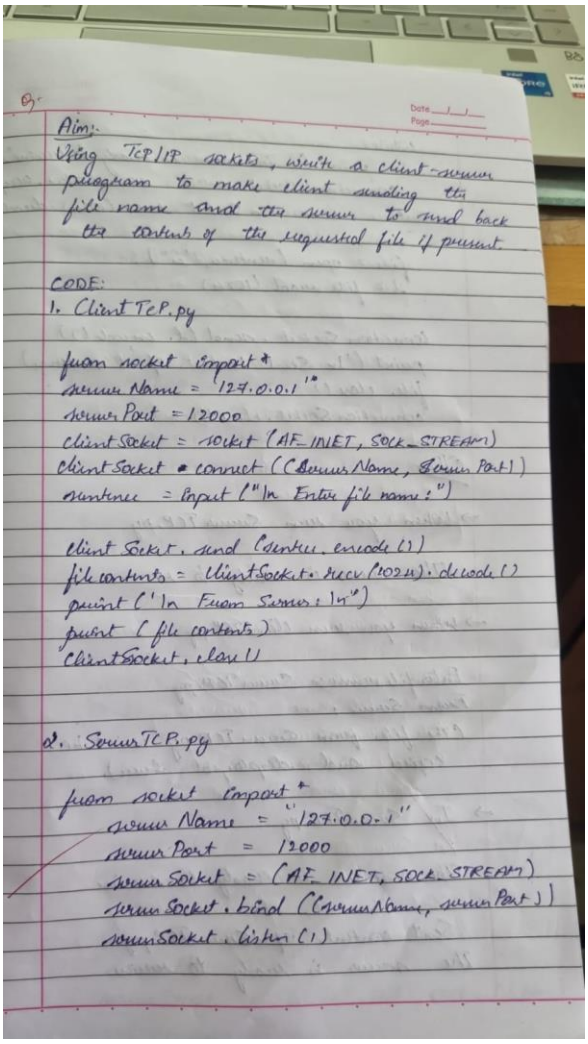
CYCLE 2

AIM:

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.

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

PROGRAM:



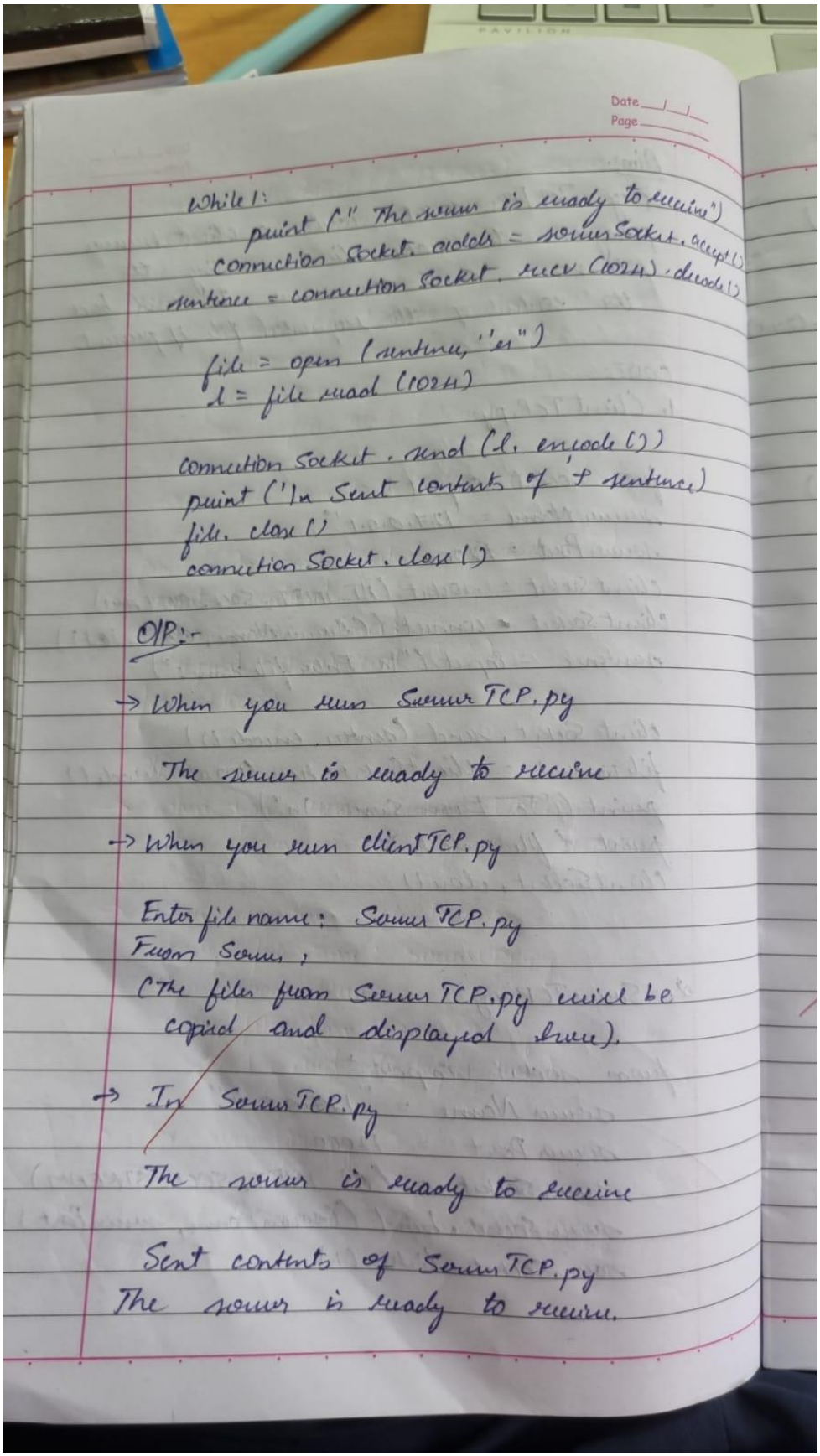
Aim:
Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.

CODE:
1. ClientTCP.py

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = input("\n Enter file name:")

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

2. ServerTCP.py

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
serverSocket = (AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
```



```
while 1:
    print ("The server is ready to receive")
    connection Socket, addr = serverSocket.accept()
    sentence = connection Socket.recv(1024).decode()

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

    connection Socket.send(l.encode())
    print ('\n Sent contents of ' + sentence)
    file.close()
    connection Socket.close()
```

O/P:-

→ When you run ServerTCP.py

The server is ready to receive

→ When you run clientTCP.py

Enter file name: ServerTCP.py
From Server:
(The files from ServerTCP.py will be copied and displayed here).

→ In ServerTCP.py

The server is ready to receive

Sent contents of ServerTCP.py
The server is ready to receive.

Date / /
Page

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

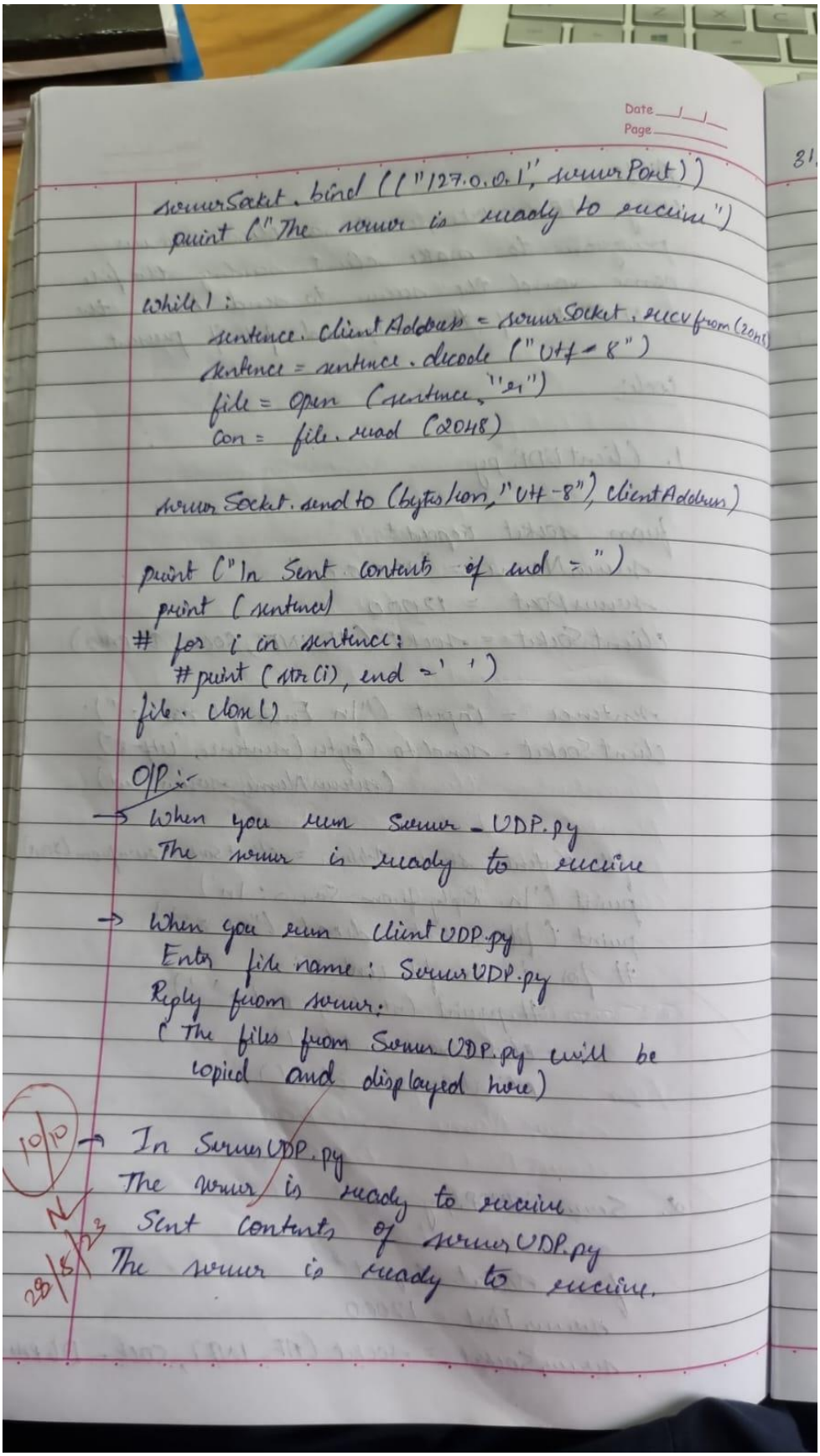
Code:-

1. 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))

file contents, serverAddress = clientSocket.recvfrom(2048)
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()
```

2. Server UDP.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("In Sent contents of end = ")
    print(sentence)
    # for i in sentence:
    #     print(str(i), end = '')
    file.close()
```

O/P :-

→ When you run Server_UDP.py
The server is ready to receive

→ When you run ClientUDP.py
Enter file name : ServerUDP.py
Reply from server:
(The files from ServerUDP.py will be copied and displayed here)

10/10

→ In ServerUDP.py
The server is ready to receive
Sent contents of serverUDP.py
The server is ready to receive.

28/8

PROGRAM:

1.

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()
```

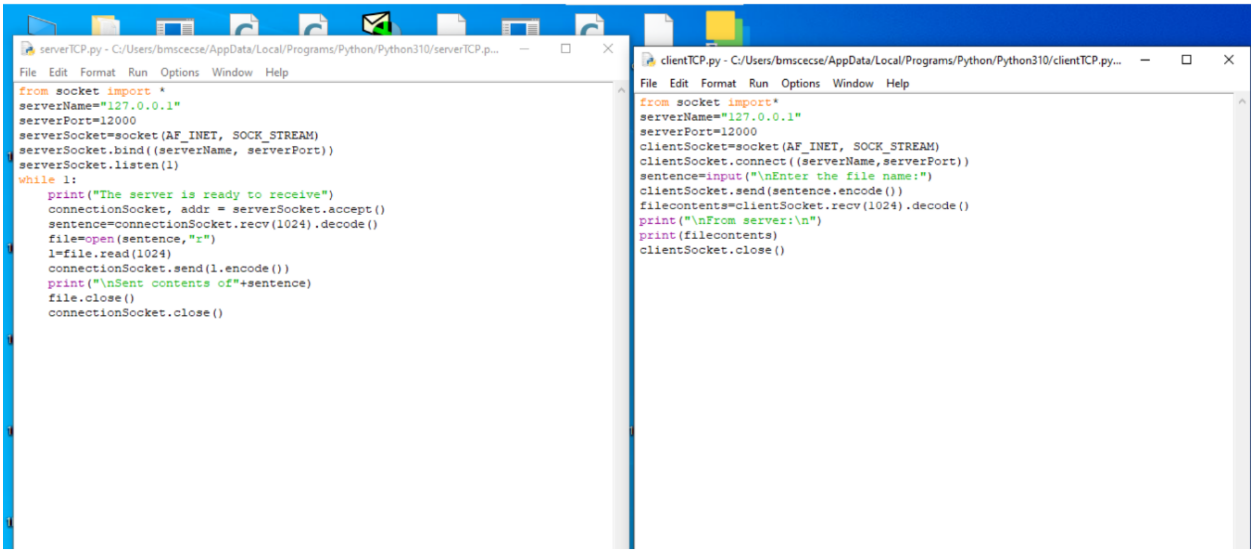
2.
ClientUDP.py

```
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:



serverTCP.py - C:/Users/bmscecse/AppData/Local/Programs/Python/Python310/serverTCP.p...

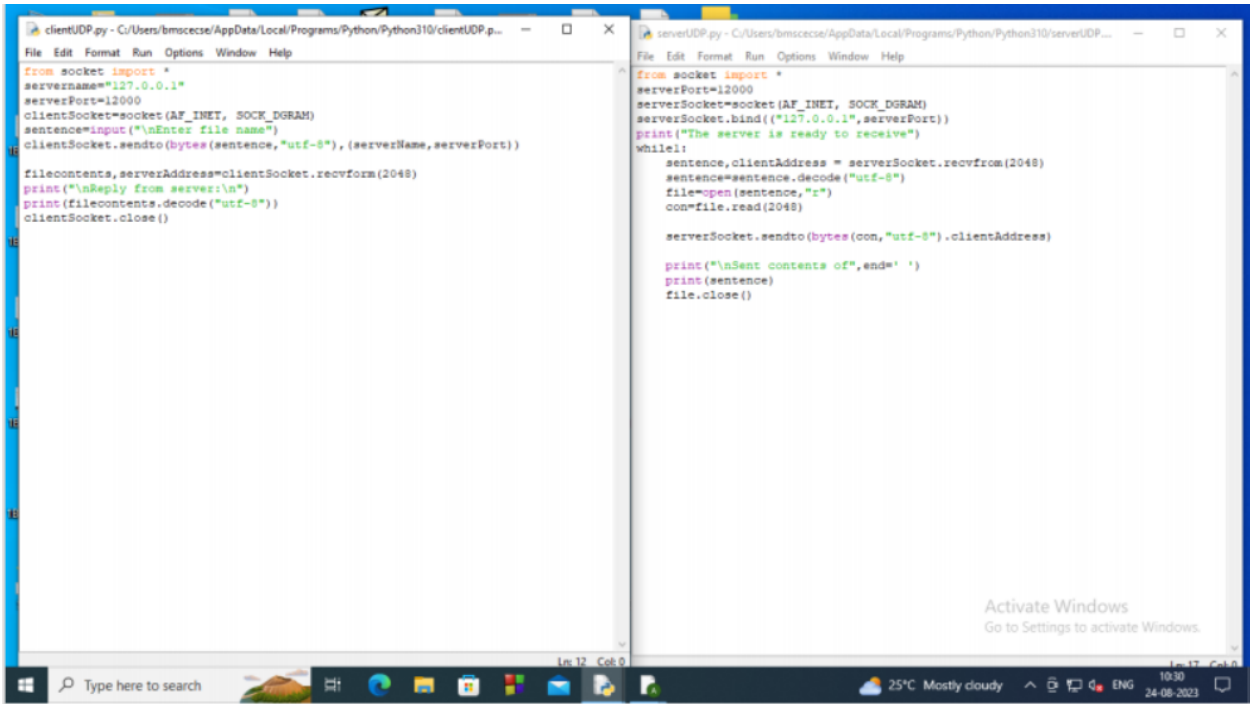
File Edit Format Run Options Window Help

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

clientTCP.py - C:/Users/bmscecse/AppData/Local/Programs/Python/Python310/clientTCP.py...

File Edit Format Run Options Window Help

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



clientUDP.py - C:/Users/bmscecse/AppData/Local/Programs/Python/Python310/clientUDP.p...

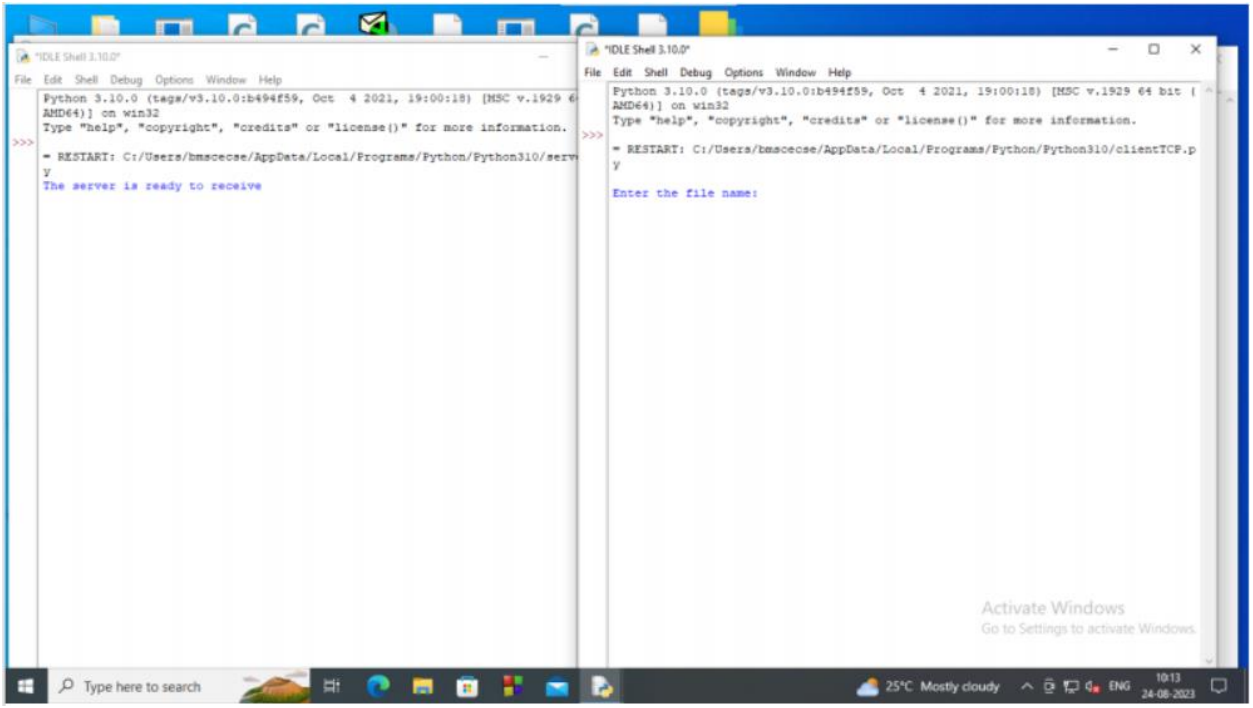
File Edit Format Run Options Window Help

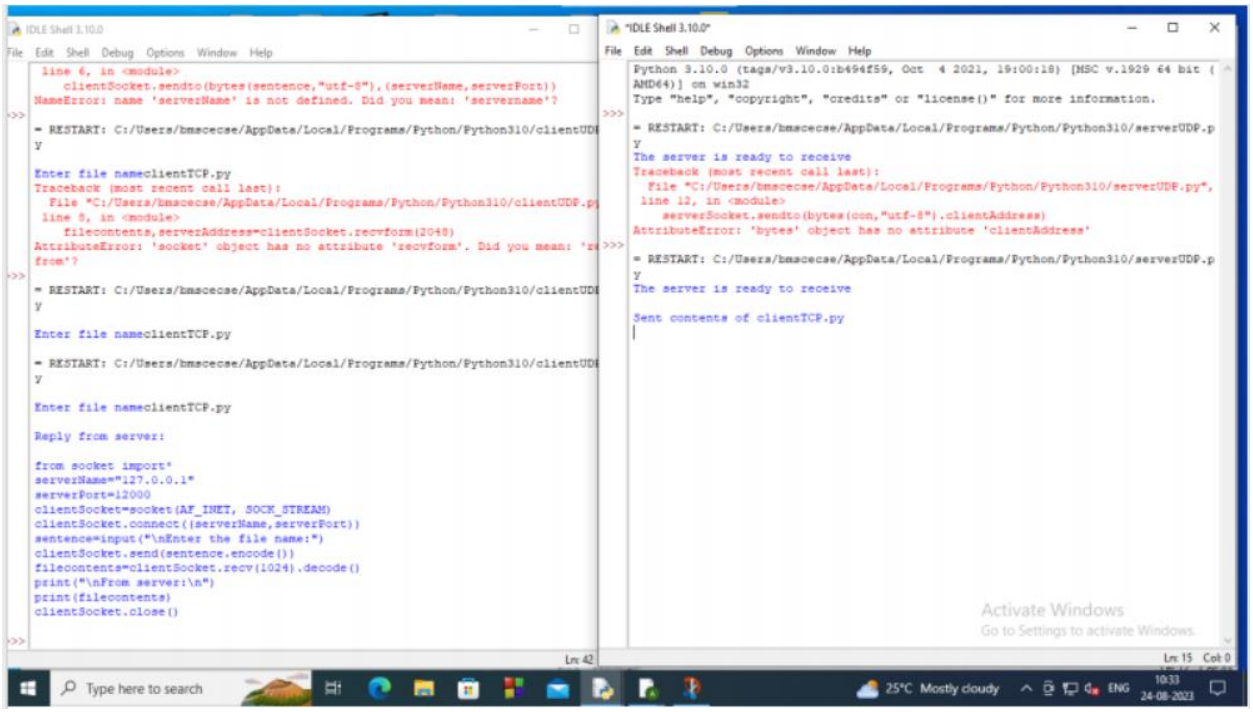
```
from socket import *
servername="127.0.0.1"
serverPort=12000
clientSocket=socket(AF_INET, SOCK_DGRAM)
sentence=input("\nEnter file name")
clientSocket.sendto(bytes(sentence,"utf-8"),(serverName,serverPort))

filecontents,serverAddress=clientSocket.recvfrom(2048)
print("\nReply from server:\n")
print(filecontents.decode("utf-8"))
clientSocket.close()
```

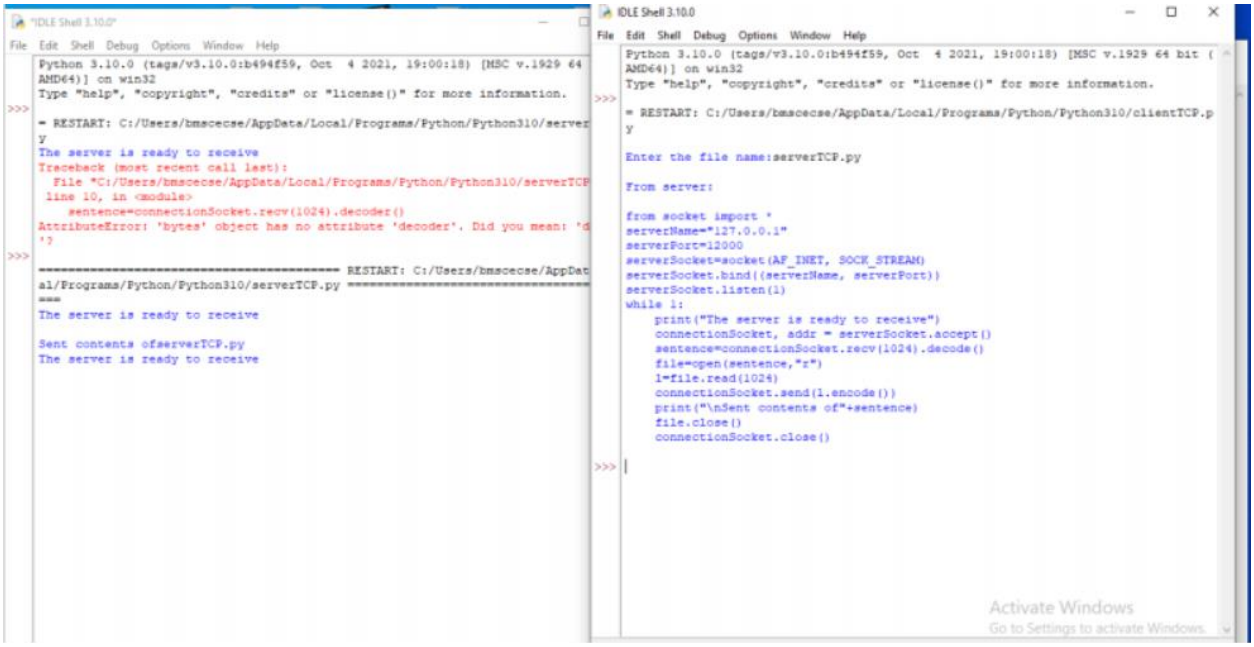
serverUDP.py - C:/Users/bmscecse/AppData/Local/Programs/Python/Python310/serverUDP....

File Edit Format Run Options Window Help

```
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")
while1:
    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)
    file.close()
```


"IDLE Shell 3.10.0"
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/bmscecse/AppData/Local/Programs/Python/Python310/clientTCP.py
y
Enter the file name:
Activate Windows
Go to Settings to activate Windows.
Type here to search
25°C Mostly cloudy
ENG
10:13
24-08-2023


"IDLE Shell 3.10.0"
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/bmscecse/AppData/Local/Programs/Python/Python310/serverUDP.py
y
The server is ready to receive
Traceback (most recent call last):
File "C:/Users/bmscecse/AppData/Local/Programs/Python/Python310/serverUDP.py", line 12, in <module>
serverSocket.sendto(bytes(con,"utf-8").clientAddress)
AttributeError: 'bytes' object has no attribute 'clientAddress'
= RESTART: C:/Users/bmscecse/AppData/Local/Programs/Python/Python310/serverUDP.py
y
The server is ready to receive
Sent contents of clientTCP.py
Activate Windows
Go to Settings to activate Windows.
Ln: 15 Col: 0
Type here to search
25°C Mostly cloudy
ENG
10:33
24-08-2023

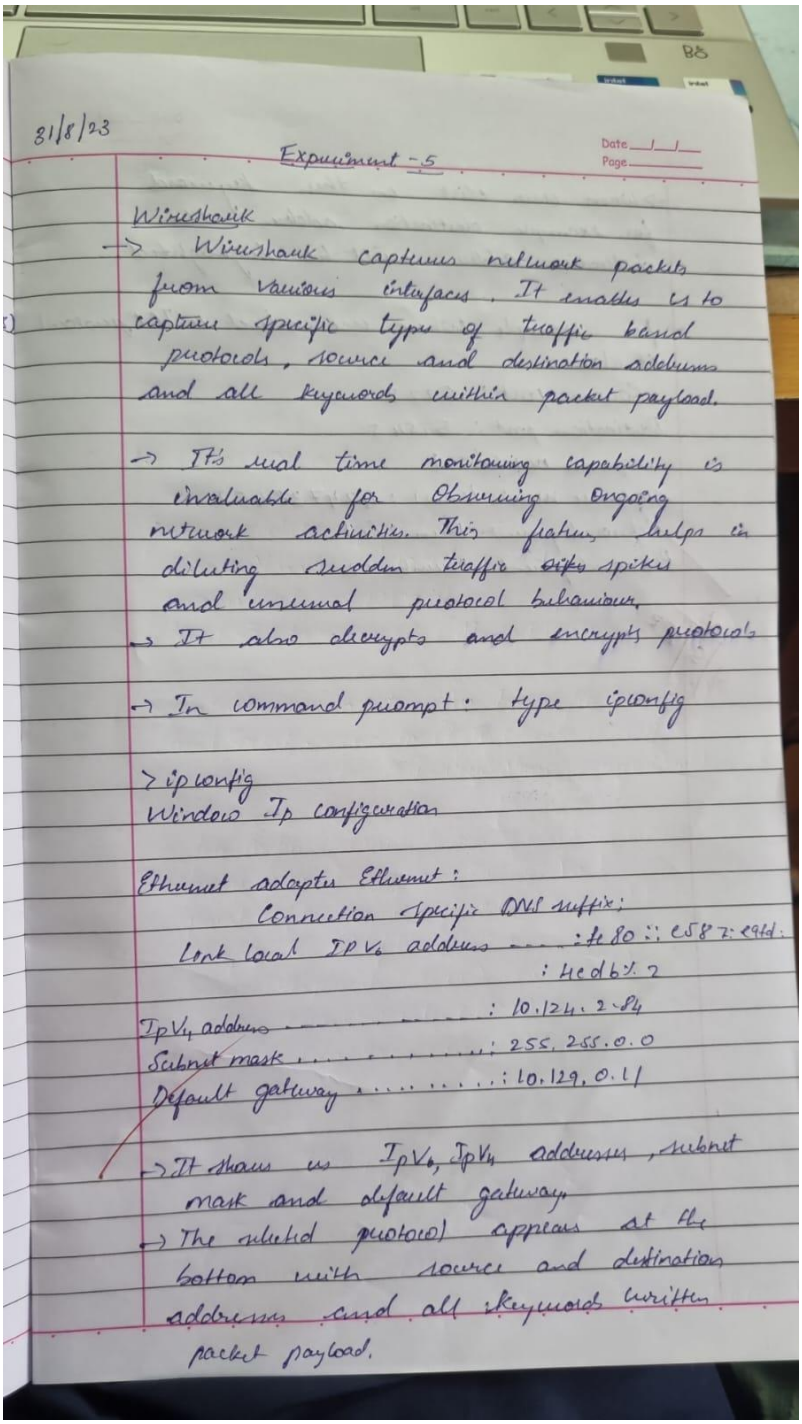

"IDLE Shell 3.10.0"
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/bmscecse/AppData/Local/Programs/Python/Python310/server
y
The server is ready to receive
Traceback (most recent call last):
File "C:/Users/bmscecse/AppData/Local/Programs/Python/Python310/serverTCP
line 10, in <module>
sentence=connectionSocket.recv(1024).decoder()
AttributeError: 'bytes' object has no attribute 'decoder'. Did you mean: 'd
'?
>>>
========== RESTART: C:/Users/bmscecse/AppDat
al/Programs/Python/Python310/serverTCP.py ==========
===
The server is ready to receive
Sent contents ofserverTCP.py
The server is ready to receive
IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/bmscecse/AppData/Local/Programs/Python/Python310/clientTCP.p
y
Enter the 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()
>>>
Activate Windows
Go to Settings to activate Windows.

CYCLE 2

AIM:

Tool Exploration -Wireshark

OBSERVATION:



31/8/23

Experiment - 5

Date / /
Page

Wireshark

→ Wireshark captures network packets from various interfaces. It enables us to capture specific types of traffic based protocols, source and destination addresses and all keywords within packet payload.

→ It's real time monitoring capability is invaluable for observing ongoing network activities. This feature helps in diluting sudden traffic ~~dips~~ spikes and unusual protocol behaviours.

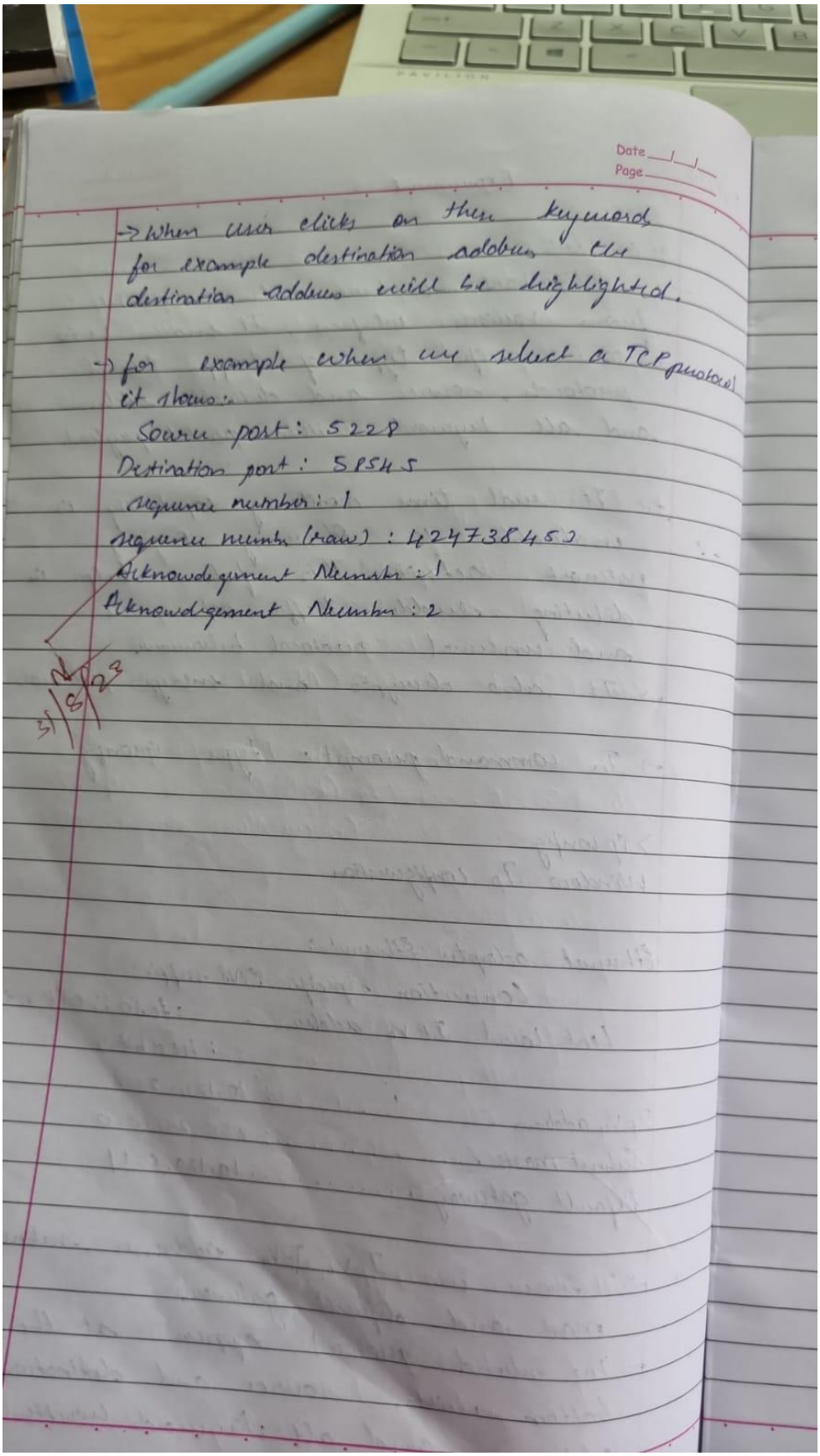
→ It also decrypts and encrypts protocols

→ In command prompt: type ipconfig

> ipconfig
Windows Ip configuration

Ethernet adapter Ethernet:
Connection Specific DNS suffix:
Link local IPV6 address . . . . : fe80::e587:e94d:
: 4edb%7
IpV4 address . . . . . : 10.124.2.84
Subnet mask . . . . . : 255.255.0.0
Default gateway . . . . . : 10.129.0.1

→ It shows us IpV6, IpV4 addresses, subnet mask and default gateway.
→ The selected protocol appears at the bottom with source and destination addresses and all keywords written packet payload.


Date __/__/__
Page

→ When user clicks on these keywords for example destination address the destination address will be highlighted.

→ for example when we select a TCP protocol it shows:

Source port: 5228
Destination port: 58545
Sequence number: 1
Sequence number (raw): 424738453
Acknowledgement Number: 1
Acknowledgement Number: 2