VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

COMPUTER NETWORKS

Submitted by

BHARAT J(1BM21CS038)

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

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

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

CERTIFICATE

This is to certify that the Lab work entitled "**COMPUTER NETWORKS**" carried out by **BHARAT J(1BM21CS038),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2023. The Lab report has been approved as it satisfies the academic requirements in respect of a **Computer Networks - (22CS4PCCON)**work prescribed for the said degree.

LAKSHMI NEELIMA
Assistant Professor
Department of CSE
BMSCE, Bengaluru

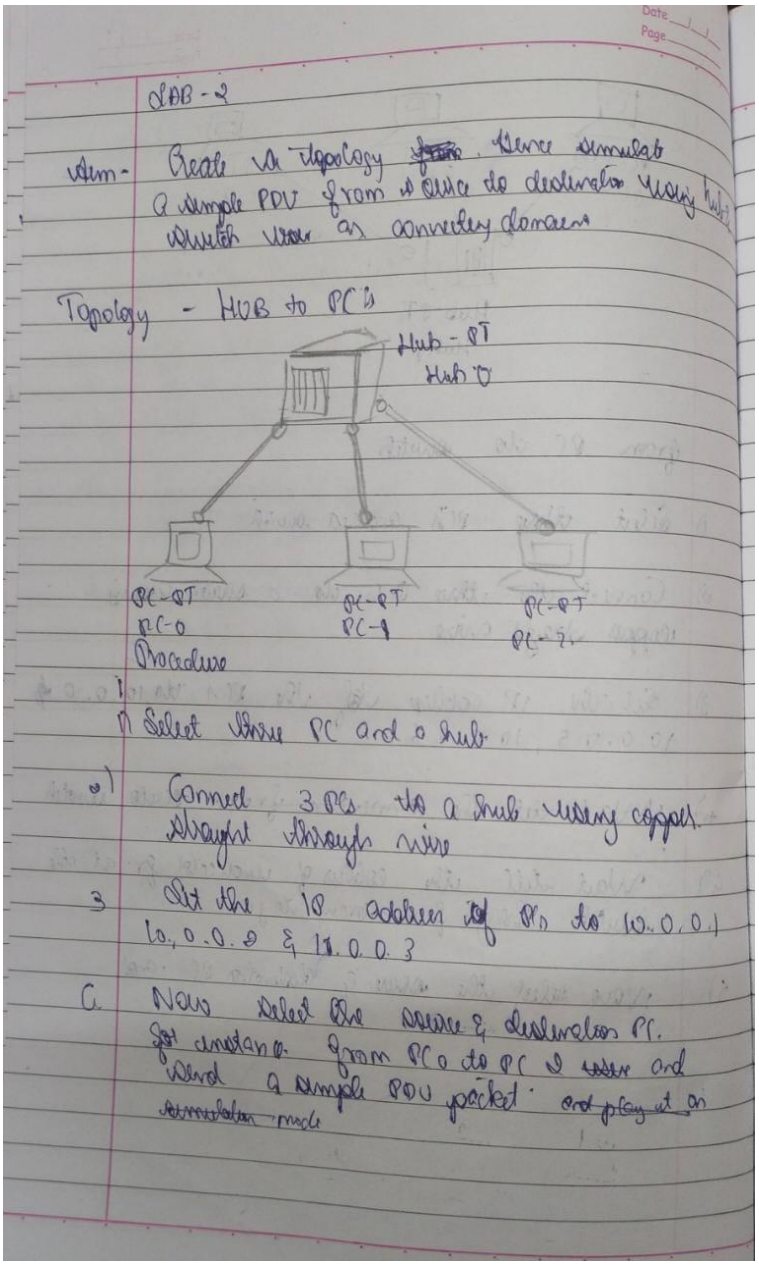
Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

`

Index

EXPERIMENT-1

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



LAB - 2

Aim - Create a topology ~~from~~. Hence simulate a simple PDU from a device to destination using hub/switch when as connecting domain

Topology - HUB to PC's

Hub - PT
Hub 0

PC-PT
PC-0

PC-PT
PC-1

PC-PT
PC-2

Procedure

1. Select three PC and a hub
2. Connect 3 PCs to a hub using copper straight through wire
3. Set the IP address of PC's to 10.0.0.1 10.0.0.2 & 10.0.0.3
4. Now select the source & destination PC. Set constant from PC 0 to PC 2 ~~wire~~ and send a simple PDU packet. ~~and play it on~~ simulation mode

Observation in simulation mode.

- PC0 sends packet to hub and hub sends it to both PC1 and PC2
- PC1 discards the message while PC2 accepts it
- PC2 sends acknowledgement packet back to hub.
- Hub again sends it to PC0 and PC1.
- PC1 discards it and PC0 accepts it

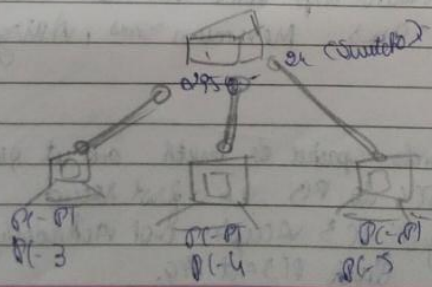
Output

~~Packet Tracer PC Command Line 1.1~~

Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
Reply from 10.0.0.2: bytes=32 time=3ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128

Ping statistics for 10.0.0.2:
Packets: Sent=4, Received=4, Lost=0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum=0ms, Maximum=3ms, Average=0ms

Topology - Switch to PC's



Procedure.

1) - Select 3 PCs and a switch
2) - Connect the PCs to the switch using copper straight wire
3) Set the IP address of the three PCs to 10.0.0.4, 10.0.0.5, 10.0.0.6.
4) Now select the source & destination PCs for instance from PC-3 to PC-5.
5) and send a PDU packet

~~Observation on simulation mode~~

~~- PC3 sends packet to switch~~

Output

Reply from 10.0.0.6: bytes=32 time=0ms, TTL=128
Reply from 10.0.0.6 bytes=32 time=0ms, TTL=128
Reply from 10.0.0.6 bytes=32 time=0ms TTL=128
Reply from 10.0.0.6 bytes=32 time=3ms TTL=128

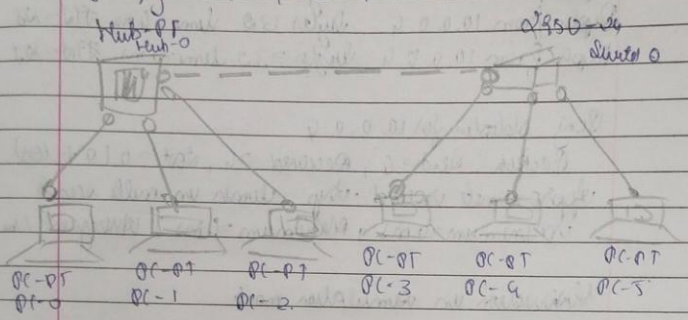
Ping statistics for 10.0.0.6
Packets: Sent=4, Received=4, Lost=0 (0% loss)
Approximate round trip times in ms
Minimum = 0ms Maximum=3ms, Average=0ms

Observation

PC3 sends packet to switch and it sends to both PC4 to PC5 use. first record PC4 reject PC5 accepts and acknowledge packet to both PC3 and PC4.

PC4 discards it and PC accepts
Now when PC3 sends packet it sends reply to PC5.

Topology - PC, Hub & Switch



Procedure:

1) Select 6 PCs, 1 a hub & a switch.
2) Connect the first three PCs and a hub with copper straight wire and the next 3 PCs and a switch with a copper straight wire.
3) Now connect the hub and switch with copper cross over wire.
4) Now send a ping packet from PC 0 to PC 4.
5) Set the IP address of PCs to 10.0.0.1, 10.0.0.2, 10.0.0.3, 10.0.0.4, 10.0.0.5, 10.0.0.6

Output

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 = 4ms 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 = 4ms, Average = 1ms

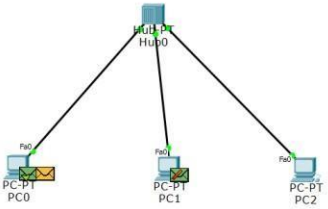
Observation in simulation mode

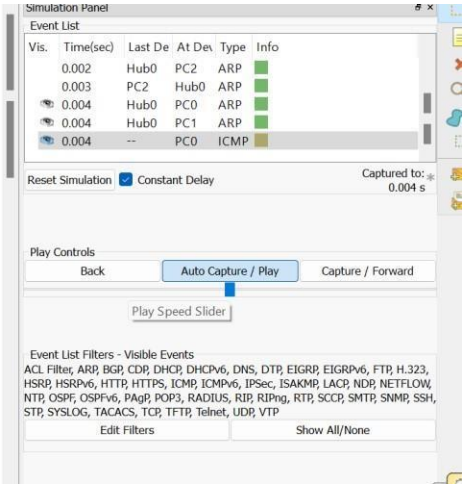
→ In simulation mode PC 0 sends packet to hub
→ Hub sends it to PC1 and PC2 and switch broadcasts it to PC3, PC4 and PC5
→ PC3 accepts and sends acknowledgement to hub through switch
→ Hub also broadcasts it to all 3 PCs
→ Only PC0 accepts it and others discard
→ In second round PC0 sends packet to hub. It's broadcast to PC1, PC2, switch
→ Now switch broadcasts it only to PC3
Thus switch is smart device

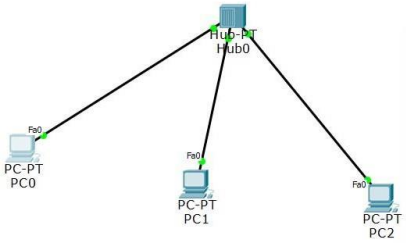
15/6/23

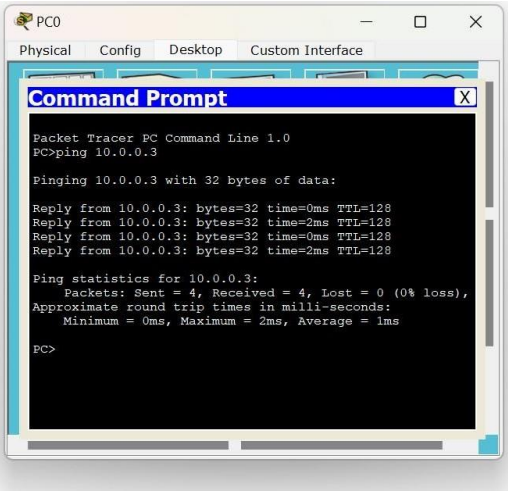
TOPOLOGY & OUTPUT

1. Hub and PCs

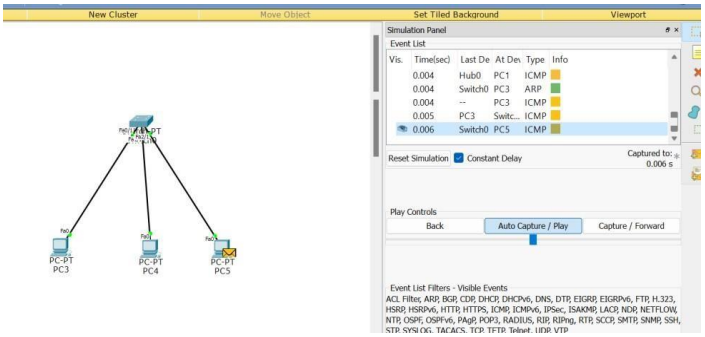


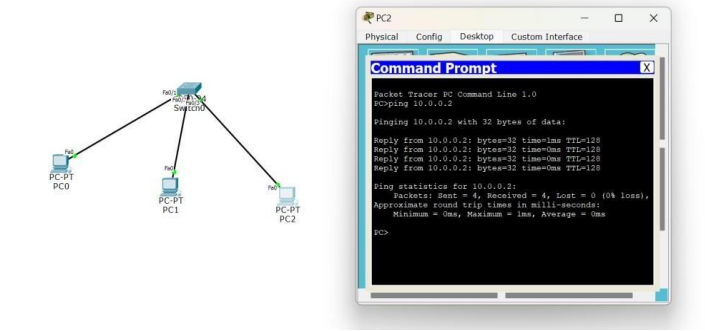




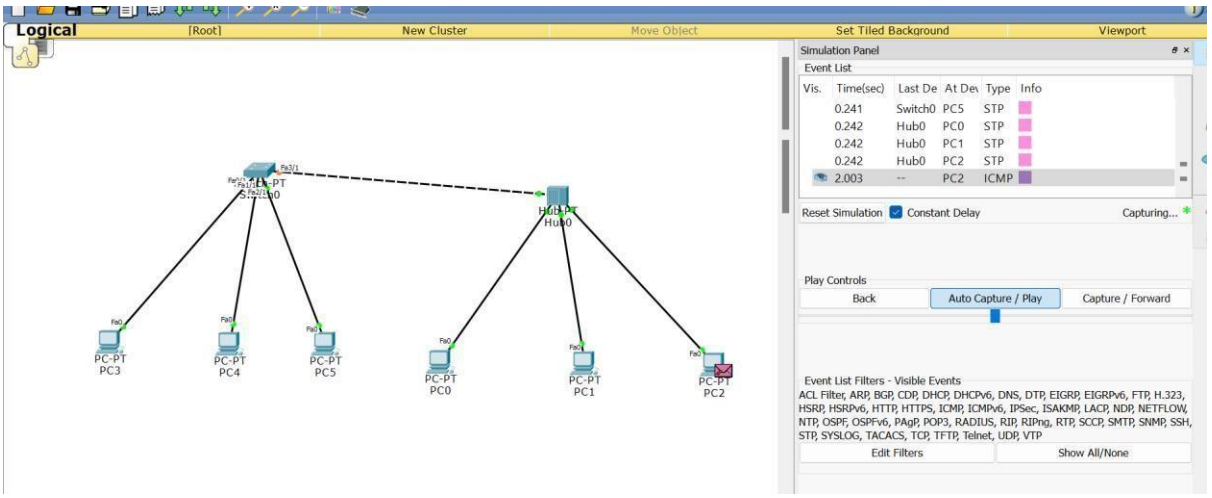


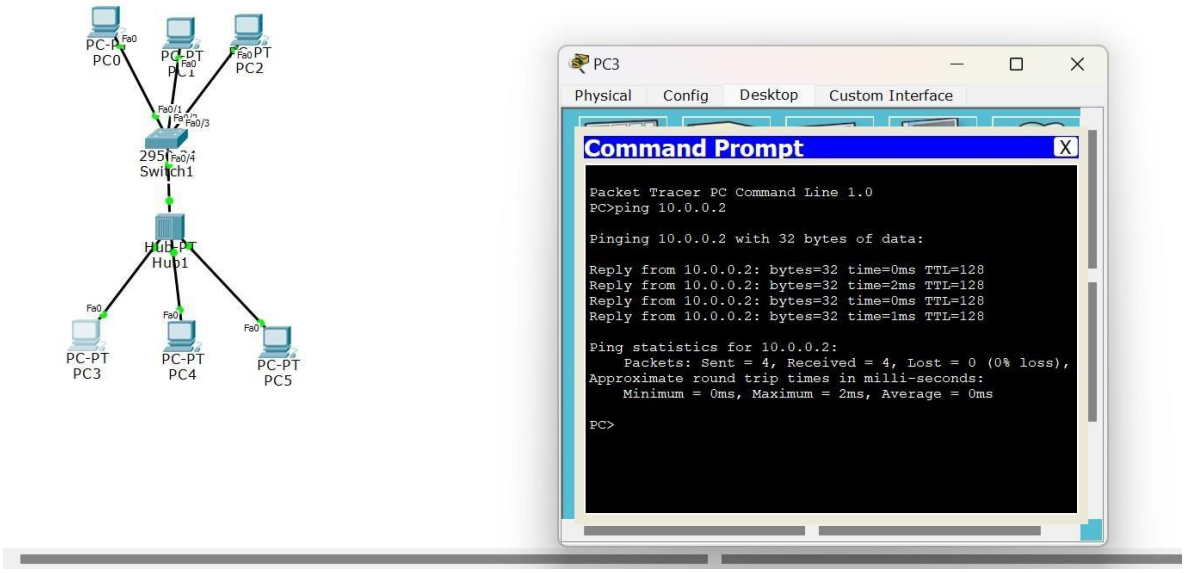
2. Switch and PCs





3. Hub,Switch and PCs

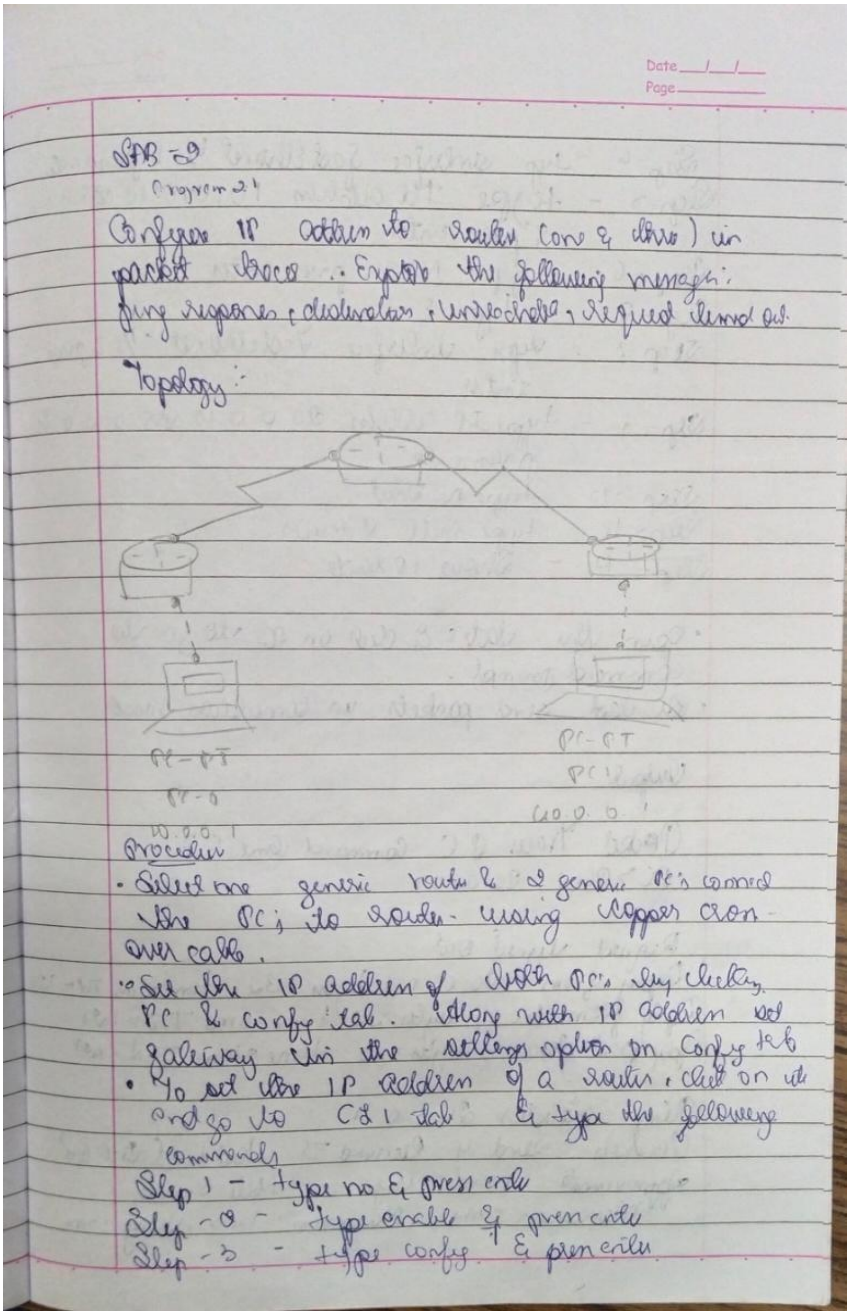



PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
Fa0/1
Fa0/3
Fa0/4
Switch1
Hub-PT
Hub1
Fa0
PC-PT
PC3
PC-PT
PC4
PC-PT
PC5
PC3
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.2
Pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
Reply from 10.0.0.2: bytes=32 time=2ms TTL=128
Reply from 10.0.0.2: bytes=32 time=0ms TTL=128
Reply from 10.0.0.2: bytes=32 time=1ms TTL=128
Ping statistics for 10.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 2ms, Average = 0ms
PC>

EXPERIMENT-2

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

PROGRAM 2.1



Date __/__/__
Page _____

EXP-2

Program 2.1

Configure IP address to router (one & three) in packet tracer. Explore the following messages: ping response, destination unreachable, request timed out.

Topology:

Procedure

- Select one generic router & 2 generic PC's connect the PC's to router using copper cross over cable.
- Set the IP address of both PC's by clicking PC & config tab. Along with IP address set gateway in the settings option on Config tab
- To set the IP address of a router, click on it and go to CLI tab & type the following commands

Step 1 - type no & press enter
Step 2 - type enable & press enter
Step 3 - type config & press enter

Step-4. type interface fastEthernet 0/0 & press enter
Step-5 - type ip address 10.0.0.10 255.0.0.0 & press enter
Step-6 : type no & press enter
Step-7 : type end
Step-8 : type interface fastethernet 1/0 & press enter
Step-9 - type IP address 20.0.0.10 255.0.0.0 & press enter
Step -10 : type no shut
Step-11 : type exit & press
Step-12 - Show IP route

- Open the lab: & click on PC to go the command prompt.
- At last send packets in simulation mode

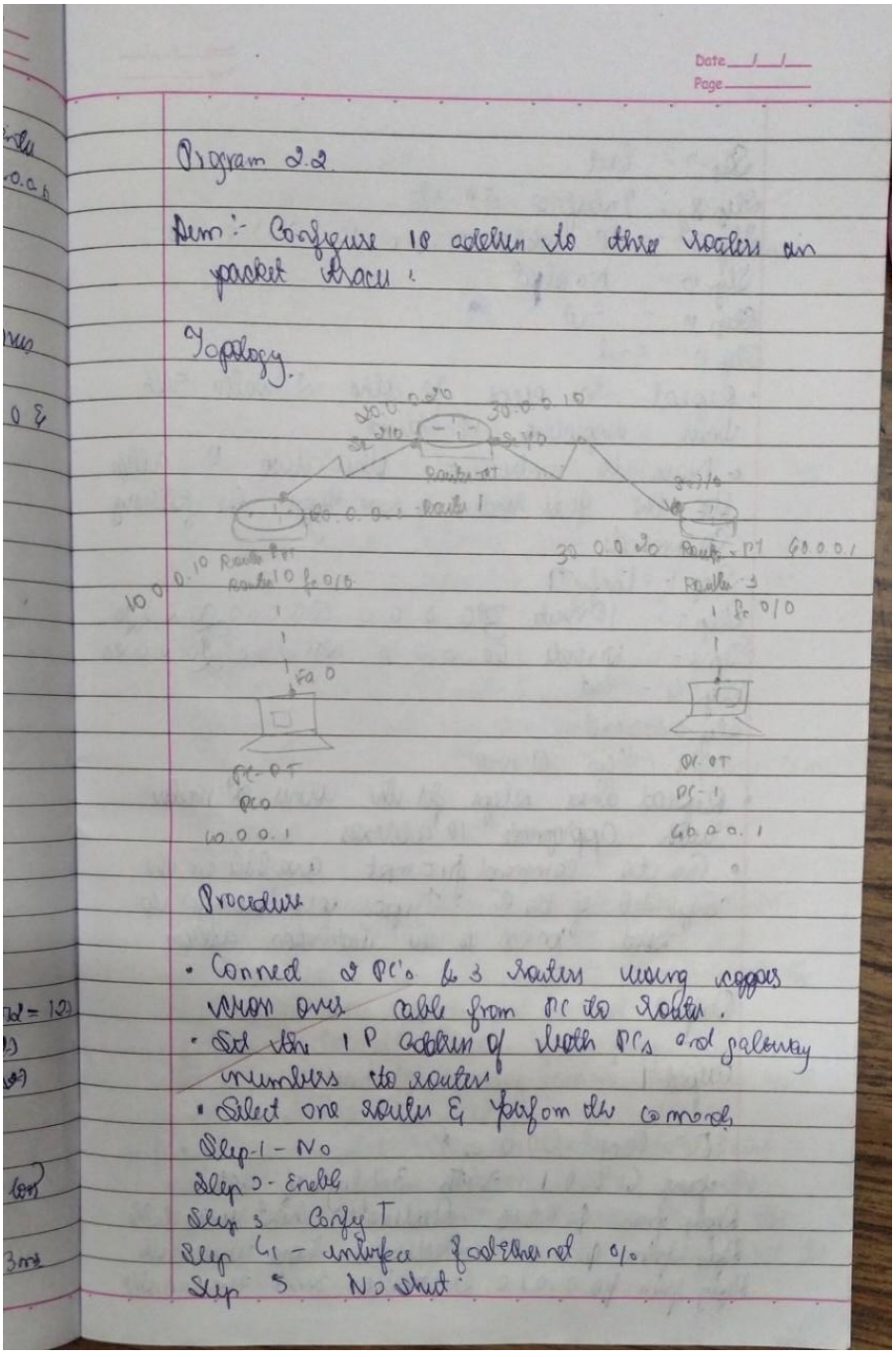
Output

Packet Tracer PC Command Line 1.0
PC > ping 20.0.0.1

Request timed out
Reply from 20.0.0.1 bytes = 32 time = 0ms TTL = 128
Reply from 20.0.0.1 bytes = 32 time = 0ms TTL = 128
Reply from 20.0.0.1 bytes = 32 time = 1ms TTL = 128

Ping statistics 20.0.0.1
Packets: Sent = 4 Received = 3 Lost = 1 (25% loss)
Approximate round trips in milli-seconds
Minimum = 0ms, Maximum = 1ms, Average = 3ms

PROGRAM 2.2



Date / /
Page

Program 2.2

Aim:- Configure IP address to three routers and packet tracer.

Topology.

Procedure

- Connect 3 PC's & 3 routers using copper cross over cable from PC to router.
- Set the IP address of both PCs and gateway numbers to router.
- Select one router & perform the commands.

Step-1 - No

Step 2 - Enable

Step 3 - Config T

Step 4 - interface fastethernet 0/0

Step 5 - No shut

Step-7 : Exit
Step-8 : Interface se 1/0
Step -9 - IP address 20.0.0.10 255.0.0.0
Step 10 - No shut
Step 11 - Exit
Step 12 - End
- Repeat the steps for other 2 routers with their respective IP address
- Now to introduce other device IP address to the first router, we type the following commands

Step 1- Config T
Step 2 - IP route 30.0.0.0 255.0.0.0 20.0.0.10
Step 3 - IP route 60.0.0.0 255.0.0.0 20.0.0.20
Step 4 - Exit
Step 5 - Exit
Step 6 - Show IP route
- Repeat these steps for the other 2 routers with appropriate IP addresses
- Go to command prompt available on the Config tab of the PC. Type ping message to send packets to the destination address

Output

Output-1

PC> Ping 60.0.0.1

Ping 60.0.0.1 with 32 bytes of data
Reply from 40.0.0.10 Destination host unreachable
Reply from 40.0.0.10 Destination host unreachable
Reply from 40.0.0.10 Destination host unreachable

Request timed out

Ping statistics for 10.0.0.1

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

Output - 2

PC> Ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data

Reply from 10.0.0.1 bytes=32 time=2ms TTL=125

Reply from 10.0.0.1 bytes=32 time=8ms TTL=125

Reply from 10.0.0.1 bytes=32 time=2ms TTL=125

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

Ping statistics for 10.0.0.1

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

Approximate round trip in milliseconds

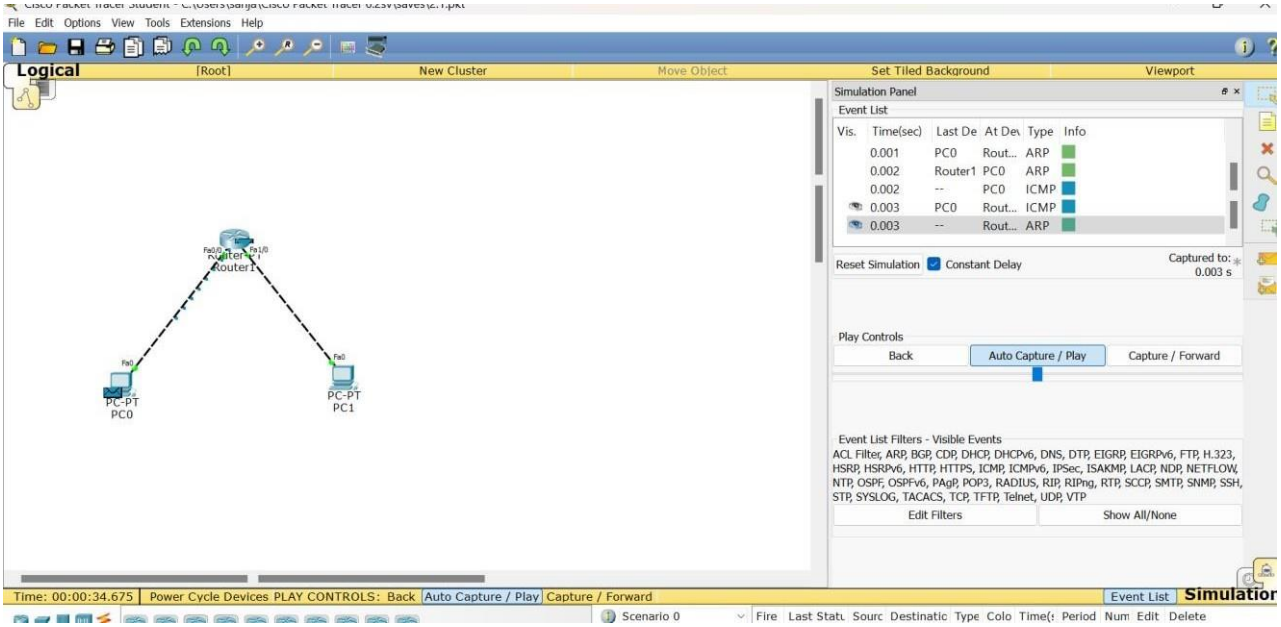
Minimum = 2ms Maximum = 8 ms Average = 3 ms.

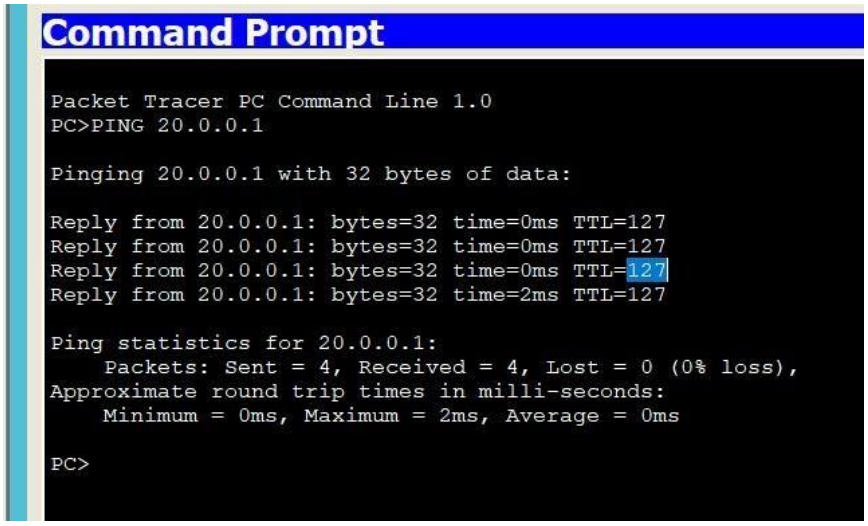
Observation

- In program 2.1 when we ping the destination address we get allocated with 32 bytes of data. In this first 8 bytes are used to learn about the routers and their address. Rest are used to send packets to destination address. If we ping again then all bytes are used for sending message. And thus well be no timed out message.
- In program 2.2 when the router didn't know about the remaining address and we ping the message we get host unreachable message. When the routers have seen or known, the message well be send successfully.

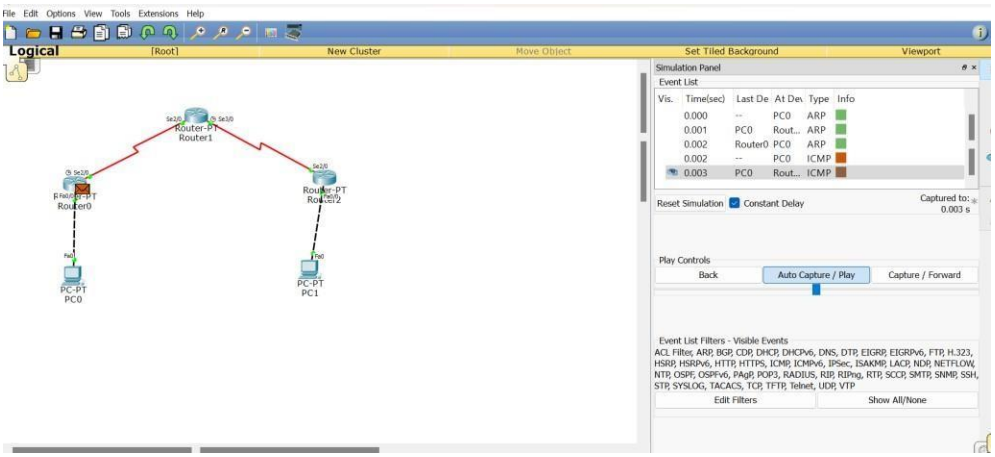
TOPOLOGY &OUTPUT

PROGRAM 2.1





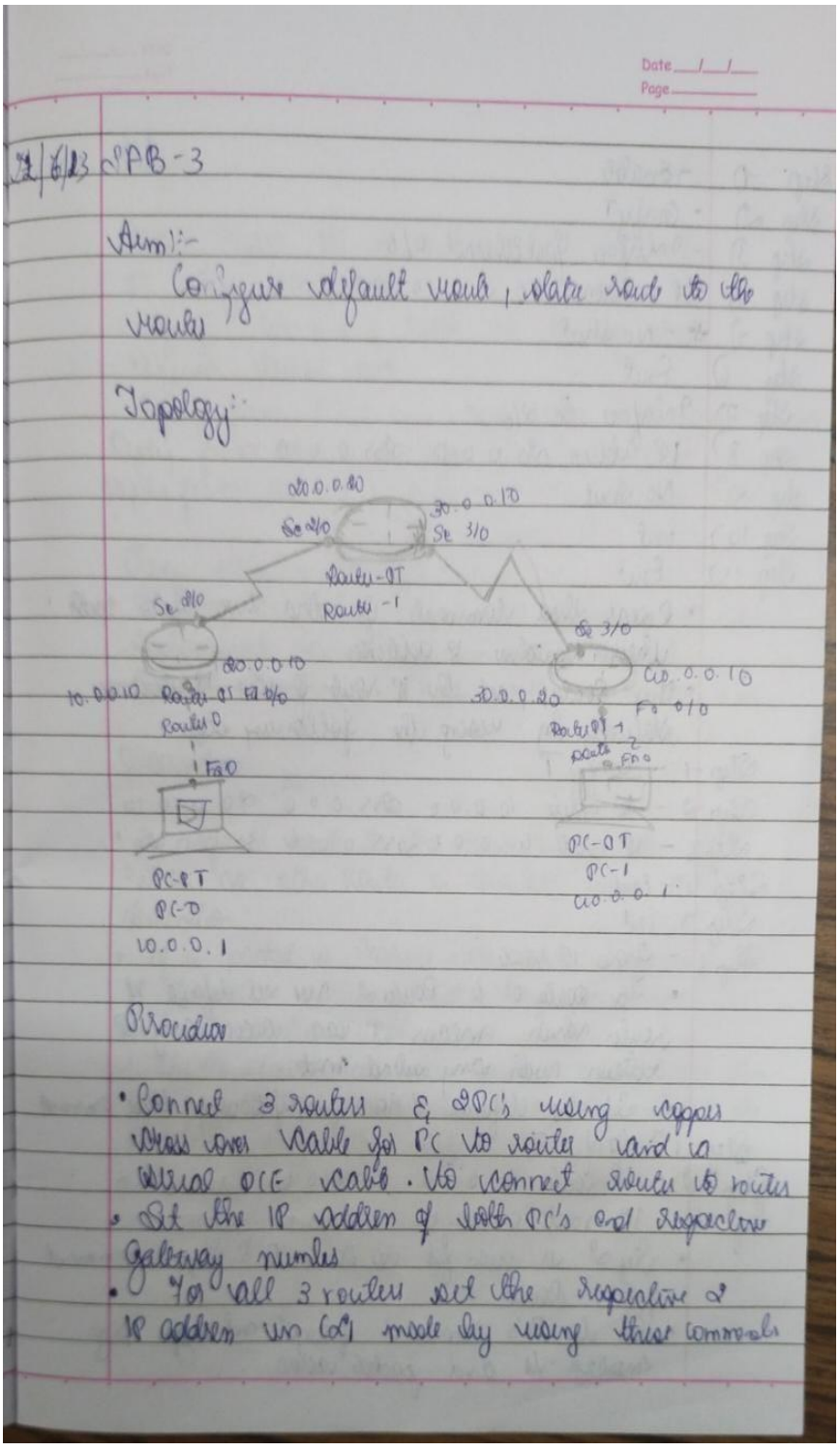
PROGRAM 2.2




```
Packet Tracer PC Command Line 1.0
PC>40.0.0.1
Invalid Command.

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=11ms TTL=125
Reply from 40.0.0.1: bytes=32 time=6ms TTL=125
Reply from 40.0.0.1: bytes=32 time=8ms 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 = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 11ms, Average = 6ms

PC>
```

EXPERIMENT-3

Q)Configure default route, static route to the Router



Date __/__/__
Page _____

Step 1) - Enable
Step 2) - Config T
Step 3) - Interface GigabitEthernet 0/0
Step 4) - IP address 10.0.0.10 255.0.0.0
Step 5) - No shut
Step 6) Exit
Step 7) Interface Se 0/0
Step 8) IP address 20.0.0.10 255.0.0.0
Step 9) No shut
Step 10) Exit
Step 11) Exit

- Repeat these commands for other two routers with their respective IP addresses
- For Router1, set the IP route of other IP addresses statically by using the following steps

Step 1 - config T
Step 2 - IP route 10.0.0.0 255.0.0.0 20.0.0.10
Step 3 - IP route 40.0.0.0 255.0.0.0 30.0.0.20
Step 4 - Exit
Step 5 - Exit
Step 6 - show IP route

- For Router 0 & Router 2 we set default IP route which means it can accept any IP addresses with any subnet mask
- Set the default IP route by following other commands

Step 1: Config T
Step 2: IP route 0.0.0.0 0.0.0.0 20.0.0.20
Step 3: IP route 0.0.0.0 0.0.0.0 30.0.0.10

- Step 2 is given for Router 0 & Step 3 command for Router 1
- Go to PC's command prompt and type ping message to and packets across

Date __/__/__
Page _____

Output

Packet Tracer PC command line 1.0
PC > Ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data
Request timed out

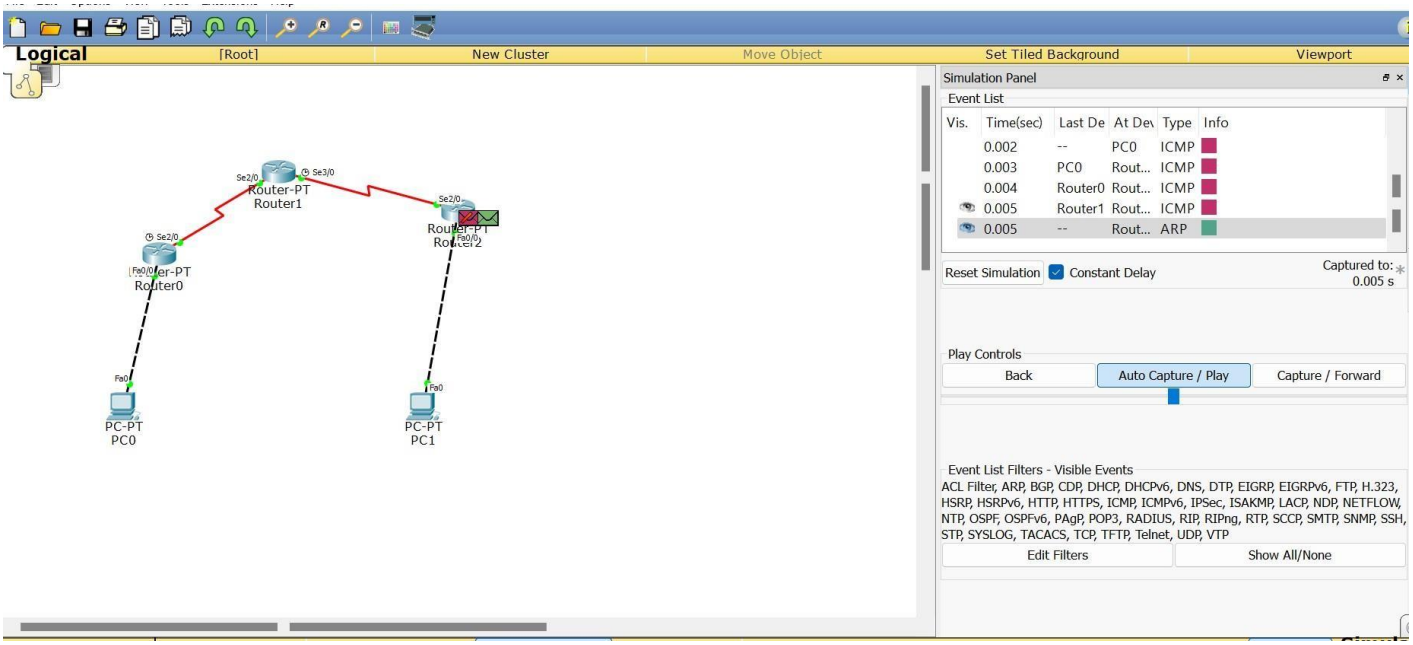
Reply from 10.0.0.1 : bytes = 32 time = 2ms TTL = 125
Reply from 10.0.0.1 : bytes = 32 time = 16ms TTL = 125
Reply from 10.0.0.1 : bytes = 32 time = 2ms TTL = 125

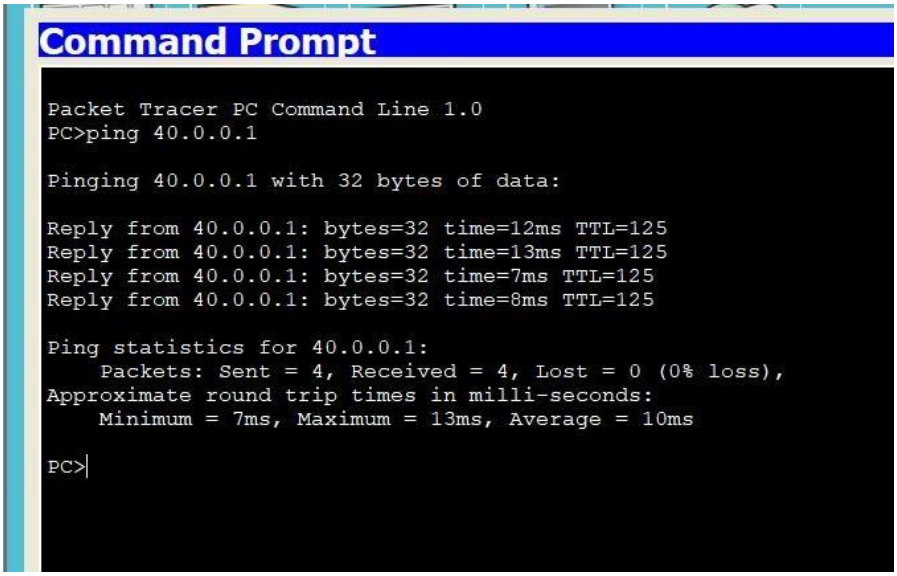
Ping statistics for 10.0.0.1
Packets : Sent = 4 , Received = 3 , Lost = 1 (25 % loss)
Approximate round trip in milli seconds
Minimum = 2ms , Maximum = 16 ms , Average = 6 ms

Observation

- A default route is the route which take effect when no other route is available for an IP address destination
- If a packet is received , the device first check the IP destination address is not local , the device check its routing table
- If the remote destination subnet is not listed then the packet is forwarded to the next hop - toward the destination using the default route
- The process repeats until the packet is delivered

TOPOLOGY & OUTPUT

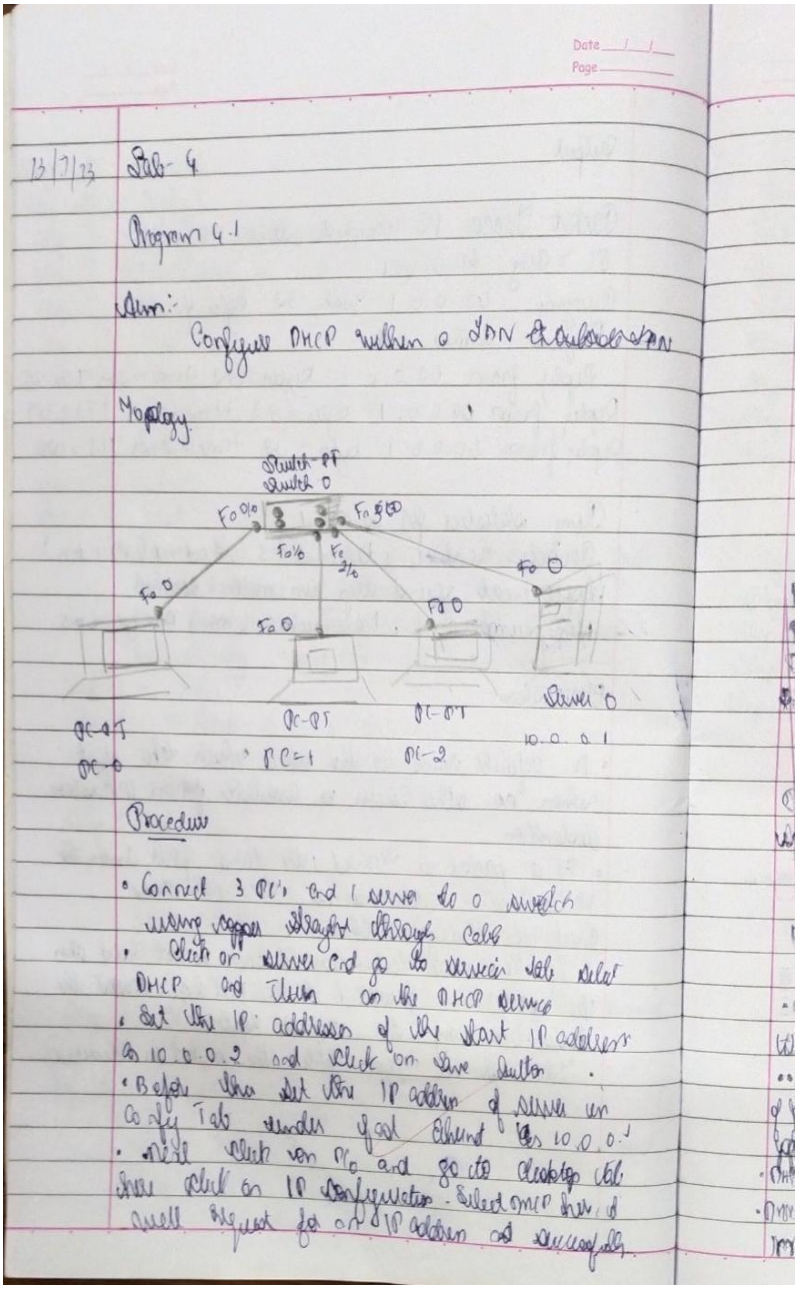




EXPERIMENT-4

Q) Configure DHCP within a LAN and outside LAN

PROGRAM 4.1



Date __/__/__
Page _____

get the DHCP request also sets the IP addres
• Repeat these steps for other 2 PC's
• To send a packet across PC's, go to PC's
Command prompt and type ping destination
IP address

Output

Packet tracer PC command line 1.0
PC > Ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data

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

Ping statistics for 10.0.0.3
Packets: Sent = 4, Received = 4, Lost = 0, (0% loss)
Approximate round trip times in milli seconds
Minimum = 0ms, Maximum = 1ms, Average = 0ms

Observation

• DHCP is used to dynamically assign an IP address to any device or node
• It is a client server protocol in which servers manage a pool of unique IP address & also about client configuration parameters.
• DHCP - enabled clients : Sends a request to DHCP server to connect to n/w.
• DHCP server responds to the client request by providing IP configuration information from address pools previously specified by n/w admin

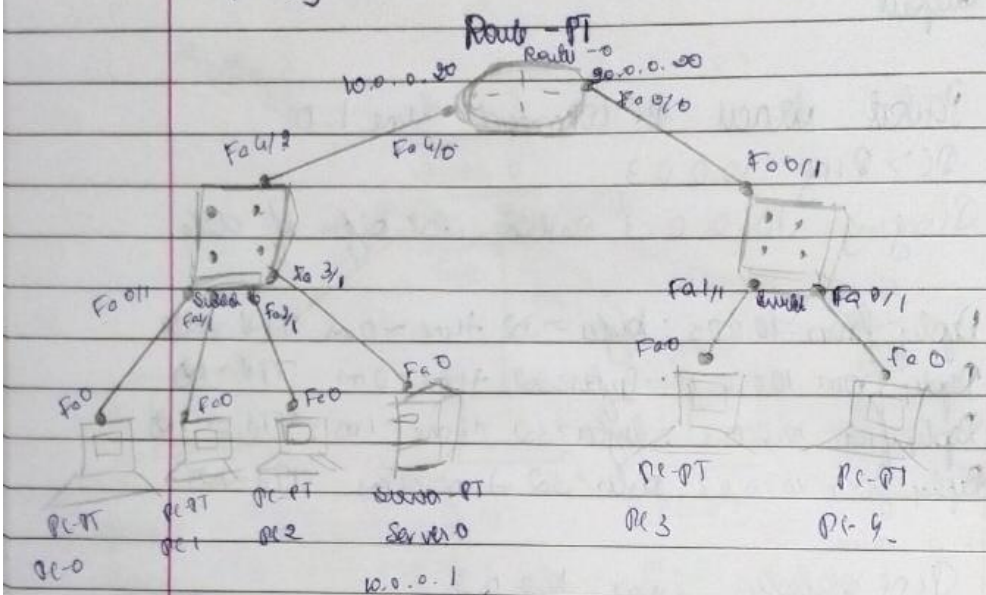
PROGRAM 4.2

Program 4.2

Aim:

Configure DHCP Outside LAN.

Topology



Procedure

- Add a Router, 2 switch & 2 PC's to a new logical network & connect the router to both switches
- Set the server IP address of server and with the help of a server set the first 3 PC's IP address through DHCP
- Now set the router IP address with the following commands manually

Step-1:- No

Step-2: Enable

Step-3 : Config T

Step-4 : Interface fastethernet 4/0

Date __/__/__
Page _____

Step 5: IP address 10.0.0.20 255.0.0.0
Step 6: No shut
Step 7: Exit
Step 8: interface fastEthernet 0/0
Step 9: IP address 20.0.0.20 255.0.0.0
Step 10: No shut
Step 11: Exit
Step 12: Show IP route Exit
Step 13: Show IP route

- Go to server and add the gateway as 10.0.0.20
- Again go to router CLI and follow these commands.

Step-14: Config T
Step-15: interface fastEthernet 0/0
Step-16: IP helper address 10.0.0.1
Step-17: No shut
Step-18: exit

- Now go to server service and add one more pool name as server pool1, start IP address as 20.0.0.2 and default gateway as 20.0.0.20 Then click add & save
- Now set the other DHCP address by going to their desktop configuration and selecting DHCP which will automatically convert all IP addresses
- Now verify the network is complete and can send packets from any PC to other by typing ping destination IP address in their respective command prompts

Output

Packet tracer PC command line 1.0
PC > Ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data:

Request timed out
Reply from 20.0.0x: bytes = 32 time = 0ms TTL = xx
Reply from 20.0.0.2 : bytes = 32 time = 0ms TTL = 127
Reply from 20.0.0.e : bytes = 32 time = 0ms TTL = 127

Ping statistics for 20.0.0.2
Packets sent = 4 Received = 3 Lost = 1 (25% loss),
Approximate round trip times in milliseconds
Minimum = 0ms Maximum = 0ms Average = 0ms

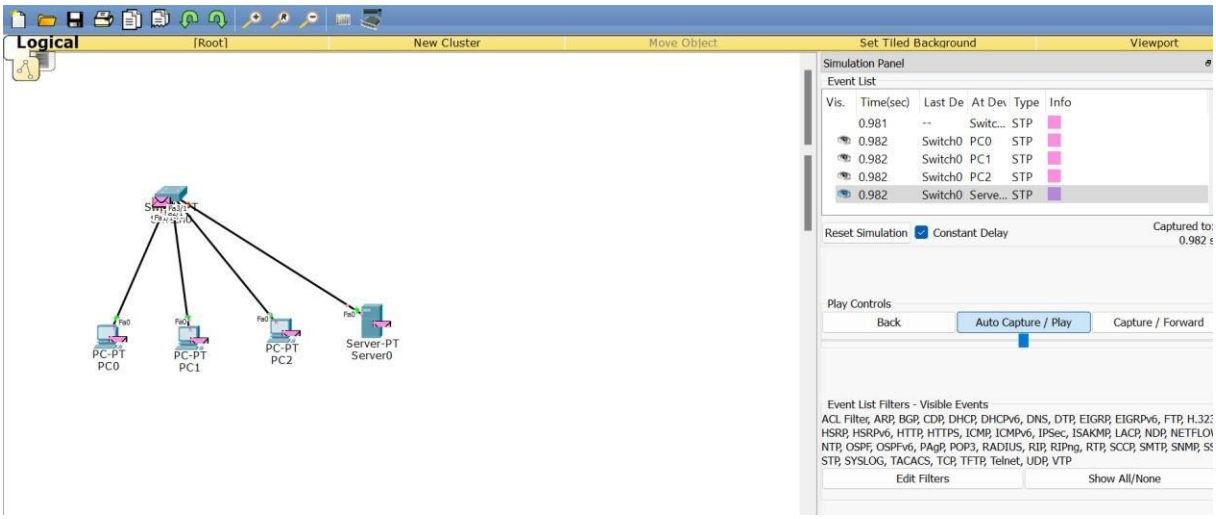
Observation

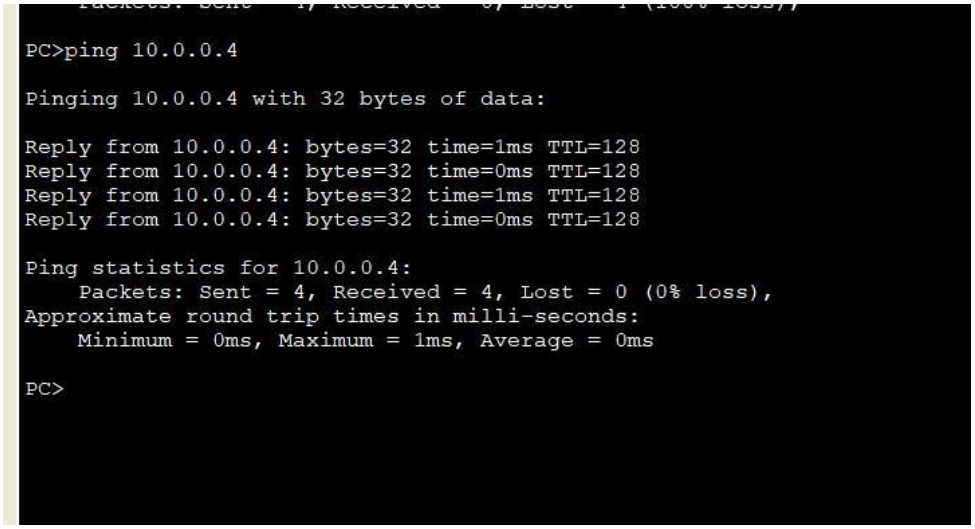
- DHCP is used to assign IP address dynamically to different devices
- To assign continuous IP address we create a serverpool where we assign the starting IP address and a default gateway number.
- For PCs under diff. switch we create a different server pool & start again. This takes care of delivering packets to correct destination IP address and also sends back to initial device

27/8/23

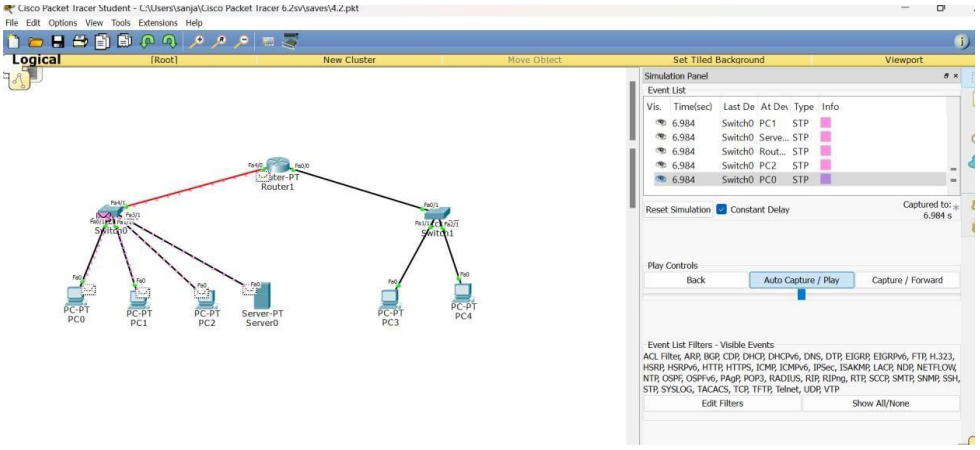
TOPOLOGY & OUTPUT

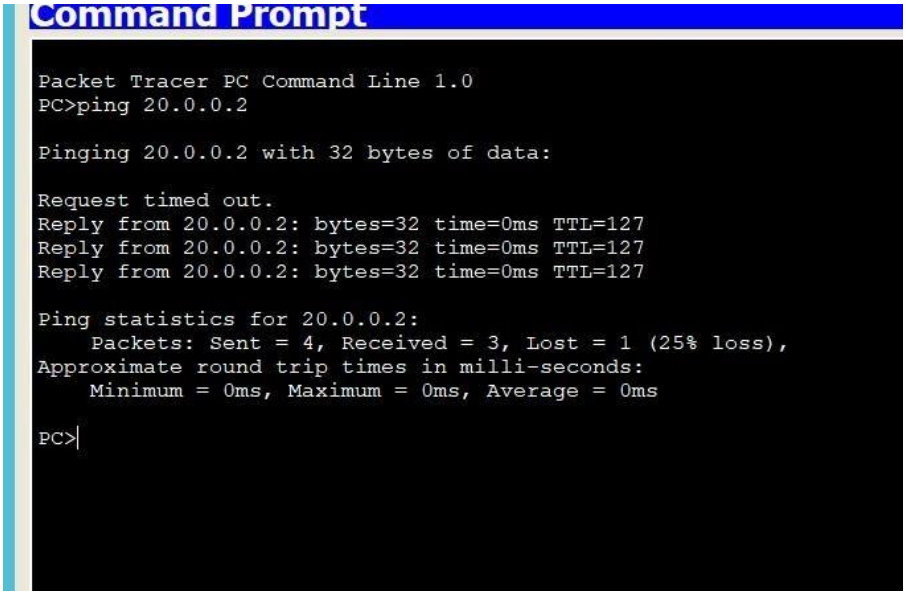
PROGRAM 4.1





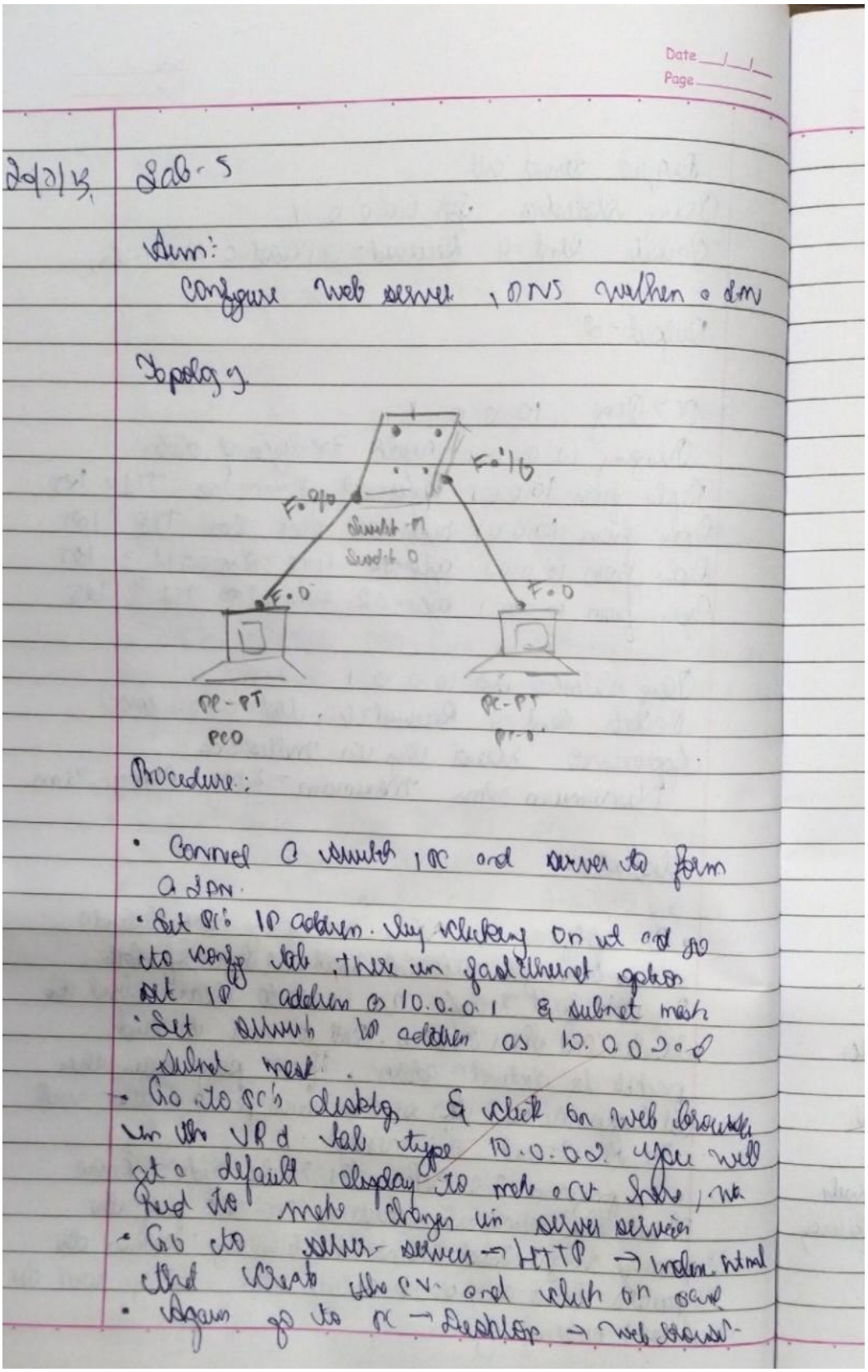
PROGRAM 4.2




Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data:
Request timed out.
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Reply from 20.0.0.2: bytes=32 time=0ms TTL=127
Ping statistics for 20.0.0.2:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>

EXPERIMENT-5

Q) Configure Web Server, DNS within a LAN



Date __/__/__
Page

20/2/15 Lab-5

Aim:
Configure web server, DNS within a LAN

Topology

Procedure:

- Connect a switch, PC and server to form a LAN.
- Set PC's IP address. By clicking on it and go to config tab. There in fastEthernet option set IP address as 10.0.0.1 & subnet mask
- Set server IP address as 10.0.0.2 & subnet mask
- Go to PC's desktop & click on web browser in the VPd tab type 10.0.0.2. You will get a default display to make a CV. Now, to find the make changes in server service
- Go to server → HTTP → index.html and create the CV and click on save
- Again go to PC → Desktop → Web browser

Date __/__/__
Page _____

and type 10.0.0.10 you will get CV or. content that is shown.

- Next go to server → services → DNS and switch on the server. Now add a domain name and type the IP address as 10.0.0.0. Then save
- Again go back to PC → Desktop → Web browser and type the given domain name. Here we can see the CV which had been added earlier

Output

Web Browser

URL [www.studentform] (Go) (Stop)

CV

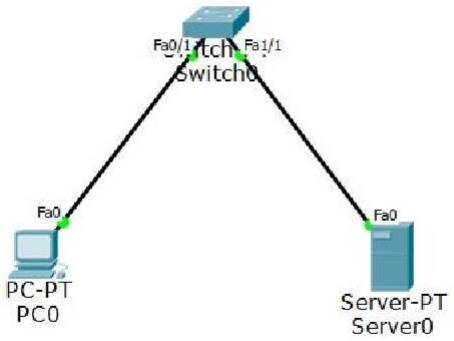
Name: Sanjana

Hobby: coding

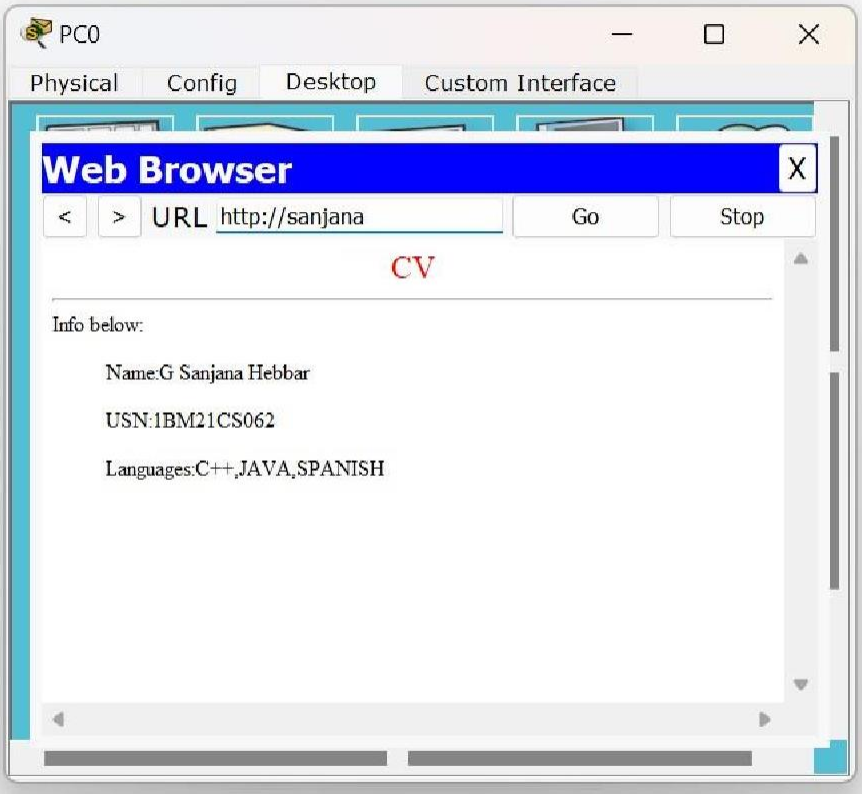
Observation

- If you wanted to go to a certain website you would open web browser and type domain name of the website or else you can also type the IP address
- Since we can't remember IP address of all website DNS server will search through its cache to find a match. IP address for the domain name & will resolve it i.e domain name to IP address of website, once that is done then computer is able to communicate with website & retrieve the page.

TOPOLOGY & OUTPUT

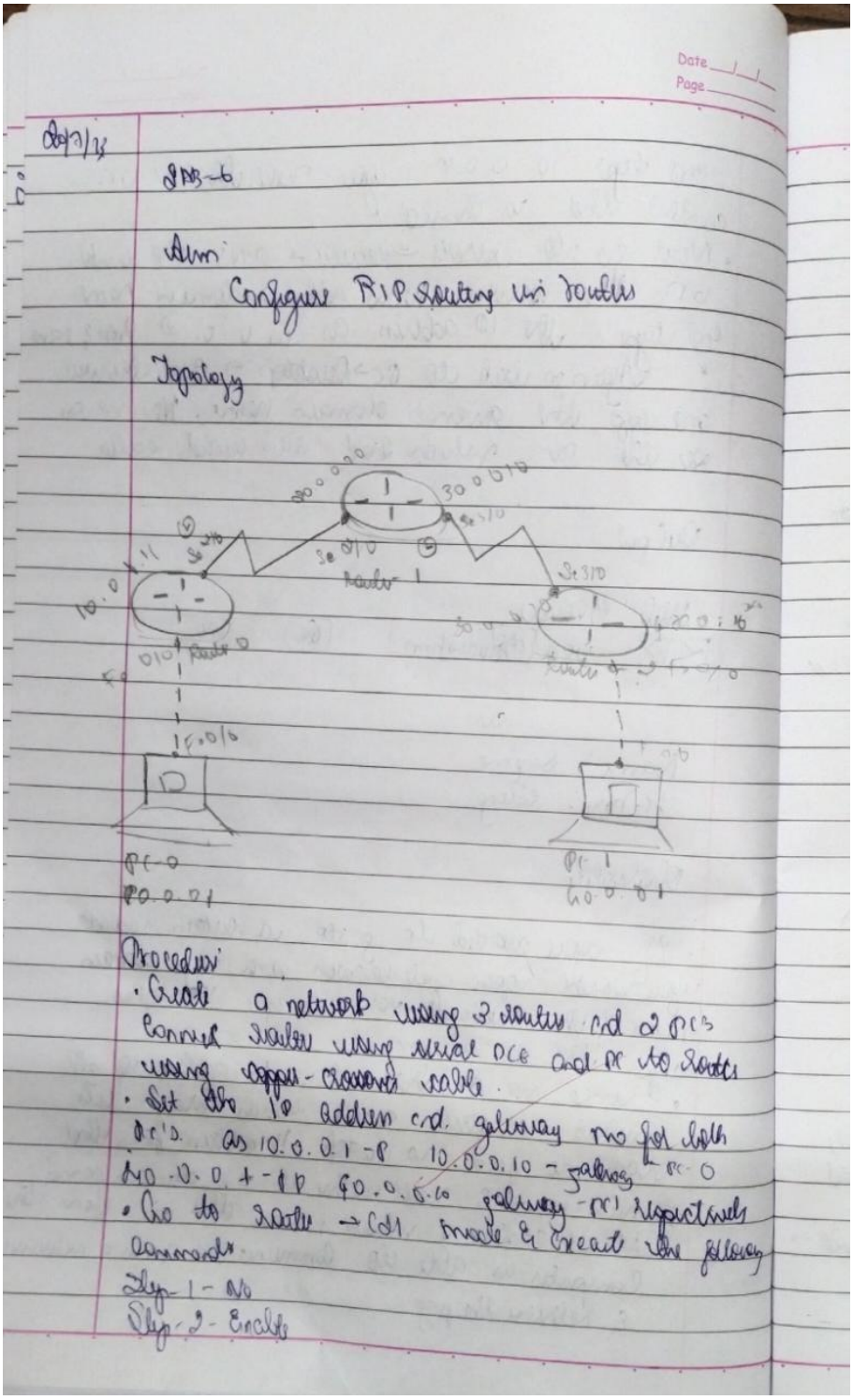


New Cluster | Move Object



EXPERIMENT-6

Q) Configure RIP routing Protocol in Routers



Date __/__/__
Page _____

Step 3 - Config T
Step 4 - Interface fastEthernet 0/0
Step 5 - IP address 10.0.0.10 255.0.0.0
Step 6 - No shut
Step 7 - Exit
Step 8 - Interface se 2/0
Step 9 - IP address 20.0.0.10 and 255.0.0.0
Step 10 - Encapsulation PPP
Step 11 - Clock rate 64000
Step 12 - No shut.
• Now for router with fastEthernet execute only until Step 9 and then No shut
• Only for Router to Router connection execute all steps, also execute the step 11 only for that router connection which has to clock symbol at start
Repeat these steps for all routers
• Again go to Router 0 → CLI mode and type these steps
• Step 1 : Config T
Step 2 : router RIP
Step 3 : Network 10.0.0.0
Step 4 : Network 20.0.0.0
Step 5 : Exit
• Repeat these steps for all routers
• At last you go to each router and type show IP route. Here the IP addresses associated with the router will be labelled as 0 and other IP addresses are labelled as R.
• Lastly go to go to PC 0 and ping a message to PC 1 using ping destination IP address command.

Date __/__/__
Page _____

Output

PING 40.0.0.1
Ping 40.0.0.1 with 32 byte of data

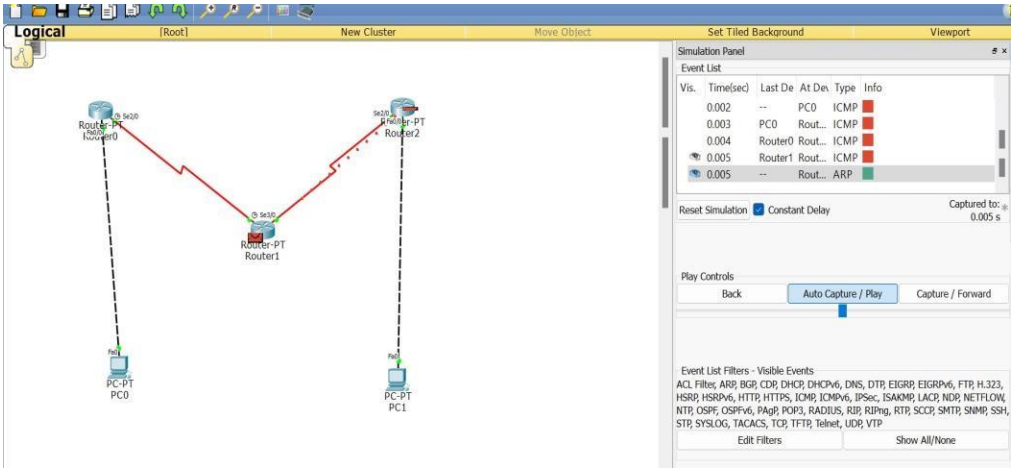
Request timed out
Reply from 40.0.0.01 bytes=32 time=8ms TTL=125
Reply from 40.0.0.1 bytes=32 time=5ms TTL=127
Reply from 40.0.0.01 bytes=32 time=10ms TTL=125

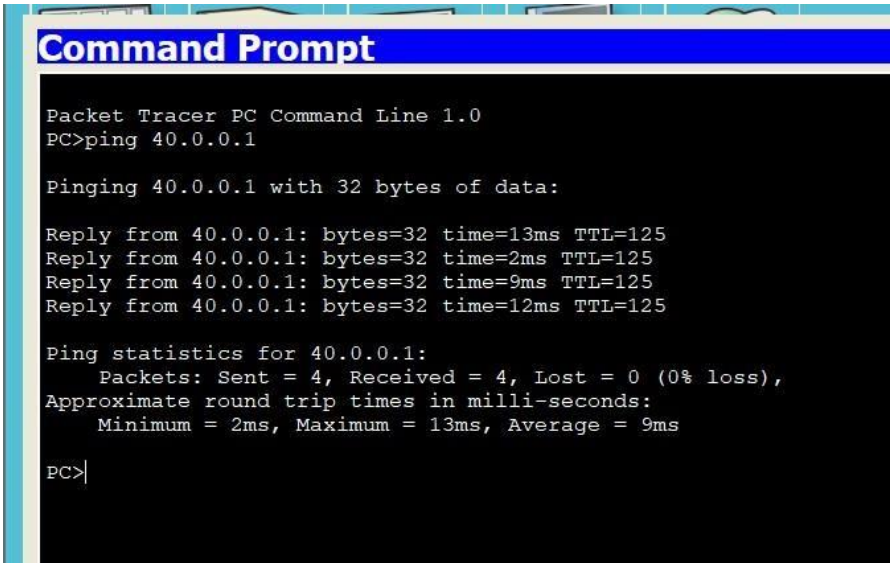
Ping statistics for 40.0.0.1
Packets Sent = 4, Received = 3, Lost = 1 (25% loss)
Approximate round trip times in milli-seconds
Minimum = 5ms Maximum = 10ms Average = 7ms

Observation

- Routing information protocol is a dynamic routing protocol that uses hop count as a routing metric to find the best path between source & destination. It is a distance vector routing protocol
- Hop count is the number of routers coming in between source & destination. The path with least hop count is selected
- Updates of the network are exchanged periodically
- Update of routing info are always broadcast
- Full routing table is sent in update
- Router always trust routing info received from neighboring routers.

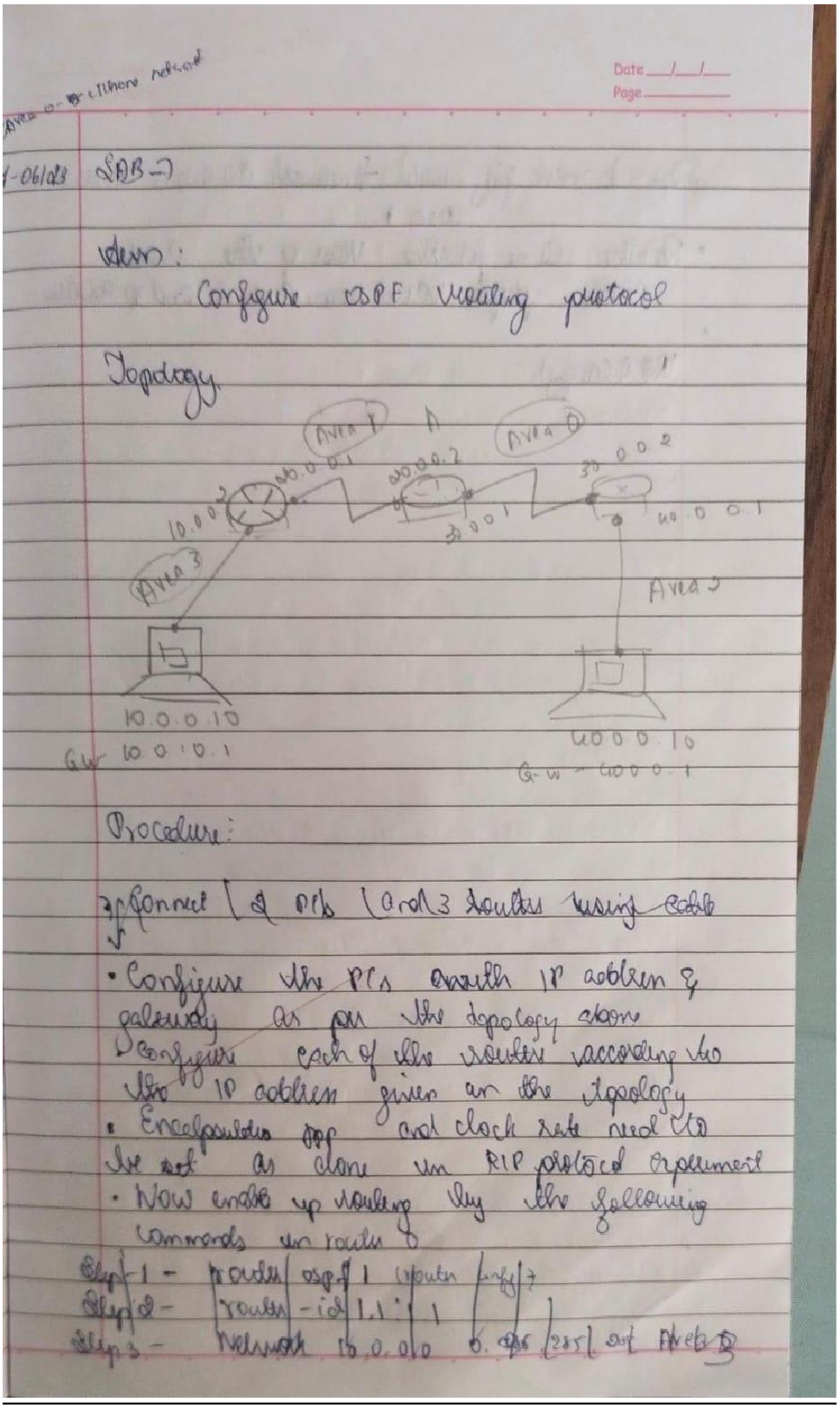
TOPOLOGY & OUTPUT





EXPERIMENT-7

Q) Configure OSPF routing protocol



Step #1 R1(config)#router ospf 1
R1(config-router)# router-id 1.1.1.1
R1(config-router)# network 10.0.0.0 0.255.255.255 area 0
R1(config-router)# network 20.0.0.0 0.255.255.255 area 0

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

R3(config)# router ospf 1
R3(config-router)# router-id 3.3.3.3
R3(config-router)# network 30.0.0.0 0.255.255.255 area 0
R3(config-router)# network 40.0.0.0 0.255.255.255 area 0

- Creating interface

R1(config-if)# interface Se2/0
R1(config-if)# interface loopback 0
R1(config-if)# ip add 172.16.1.129 255.255.255.0
R1(config-if)# no shut down

R2(config-if)# interface Se3/0
R2(config-if)# interface loopback 0
R2(config-if)# ip add 192.16.1.33 255.255.0.0
R2(config-if)# no shut down

R3(config)# interface Se3/0
R3(config-if)# interface loopback 0
R3(config-if)# ip add 172.16.1.254 255.255.0.0
R3(config-if)# no shut

Date __/__/__
Page _____

* Creating virtual link between R1 & R2

R1 (config)# router ospf 1
R1 (config-router)# area 0 virtual-link 2.2.2.2
R1 (config-router)# exit

R2 (config)# router ospf 1
R2 (config-router)# area 0 virtual-link 1.1.1.1
R2 (config-router)# exit

* Now check the connect^n

Output ping

Ping.pcs PING 10.0.0.10

Pinging 10.0.0.10 with 32 bytes of data

Reply from 10.0.0.10: bytes=32 time=1ms TTL=125
Reply from 10.0.0.10 bytes=32 time=9ms TTL=125
Reply from 10.0.0.10 bytes=32 time=9ms TTL=125
Reply from 10.0.0.10 bytes=32 time=8ms TTL=125

Ping statistics for 10.0.0.10
Packets: Sent=4 Received=4 Lost=0 (0% loss)
Approximate round trip times in milli seconds
Minimum=1ms Maximum=9ms Average=7ms

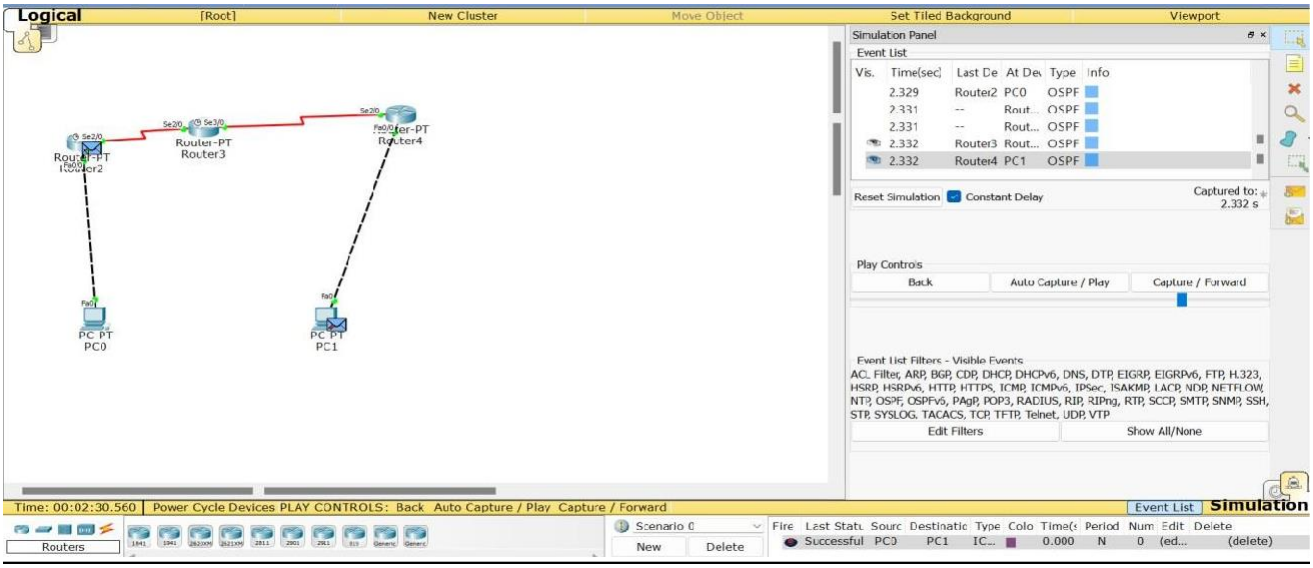
Observation- link
* OSPF is a Link state routing protocol that is used to find the best path between source and destination routers using its

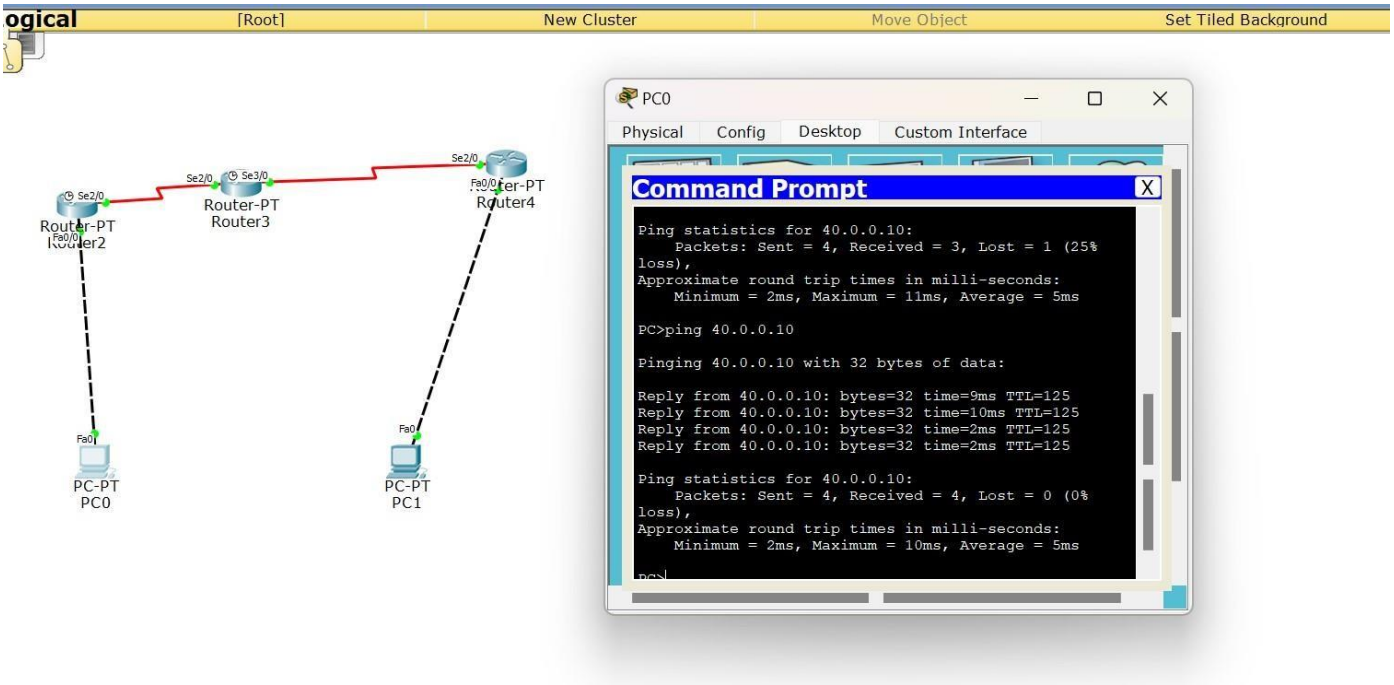
Date / /
Page

ion algorithms

- The network is divided into 4 areas where area 0 is the backbone
- After we make the virtual link between the area which is not connected to the backbone area, we can ping messages successfully

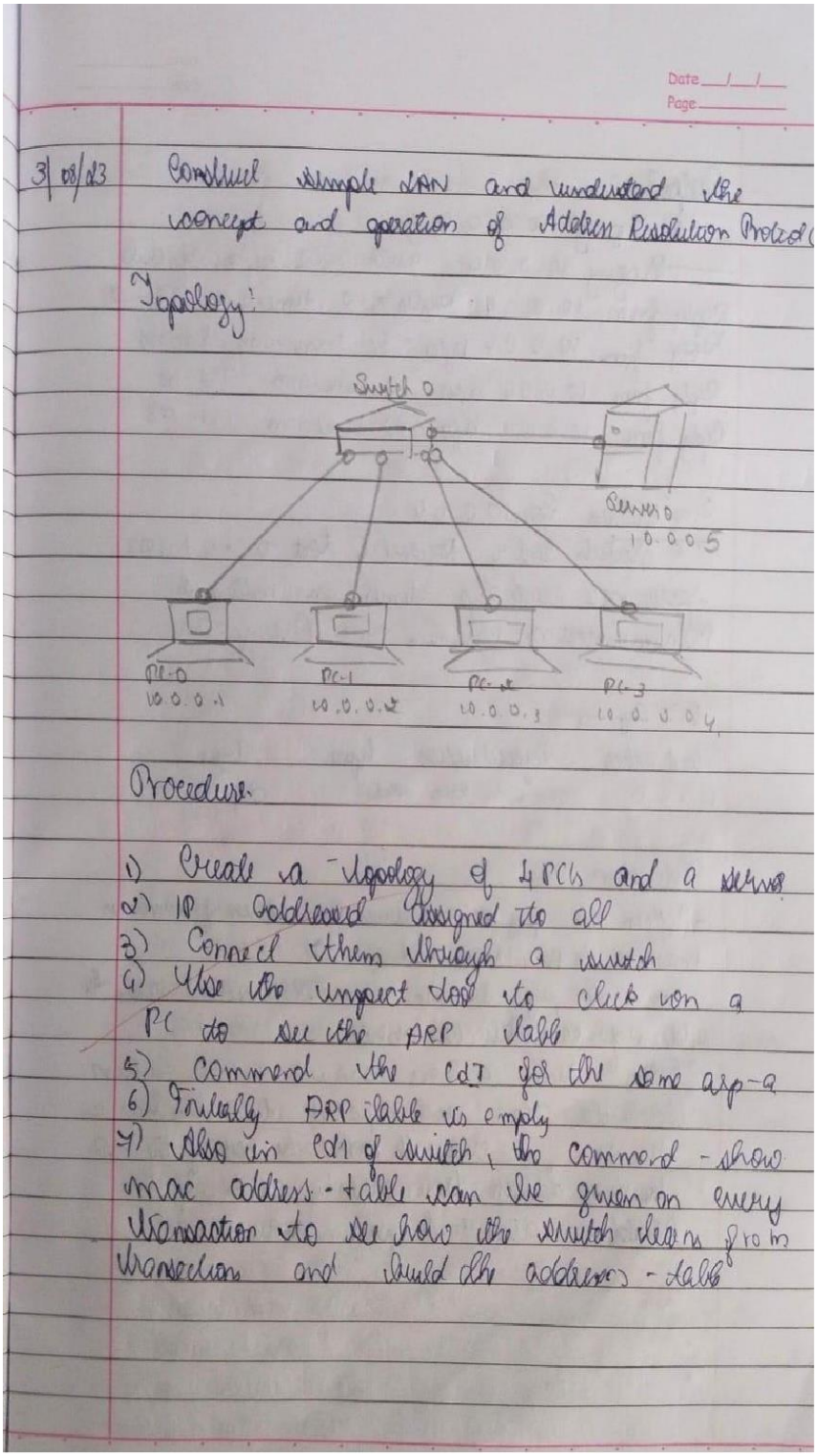
TOPOLOGY & OUTPUT





EXPERIMENT-8

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



Date
Page

Output

PC > ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data

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

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

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

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

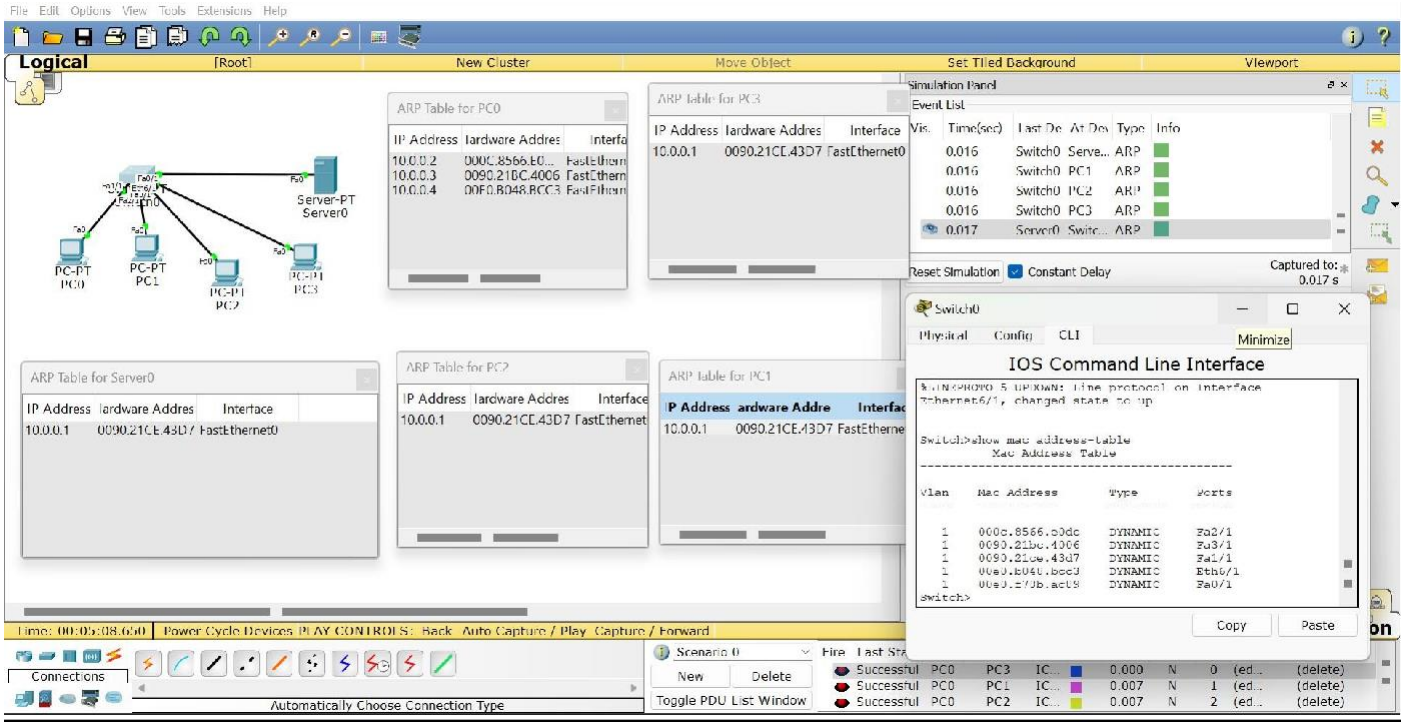
PC> arp -a

Internet address	Physical address	Type
10.0.0.4	0060.8f00.3e4d	dynamic

Observation:

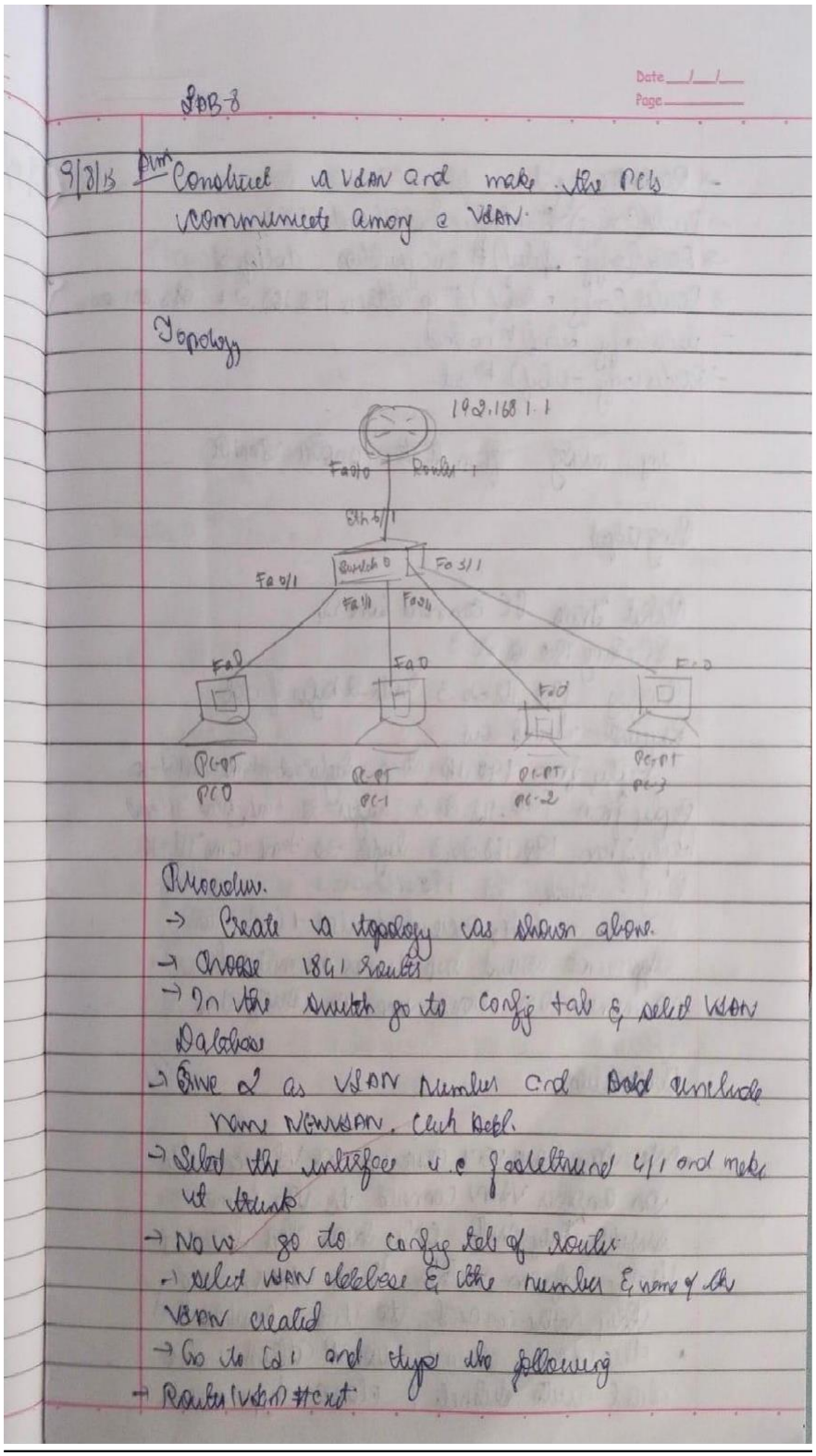
- When we ping PC and server the address of server is known to PC & vice versa.
- When we ping between other 2 PCs simultaneously the address of each other is known
- Every time a host request a MAC address in order to send a packet to another host in the LAN, it checks its ARP cache first if the IP to MAC address is already exists. If the address does not exist perform ARP

TOPOLOGY & OUTPUT



EXPERIMENT-9

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



→ Router# config t
→ Router(config)# interface fastethernet 0/0.1
→ Router(config-subif)# encapsulation dot1q 2
→ Router(config-subif)# ip address 192.168.2.1 255.255.255.0
→ Router(config-subif)# no shut
→ Router(config-subif)# end

Ping msg from PC to another VLAN PC

Ping output

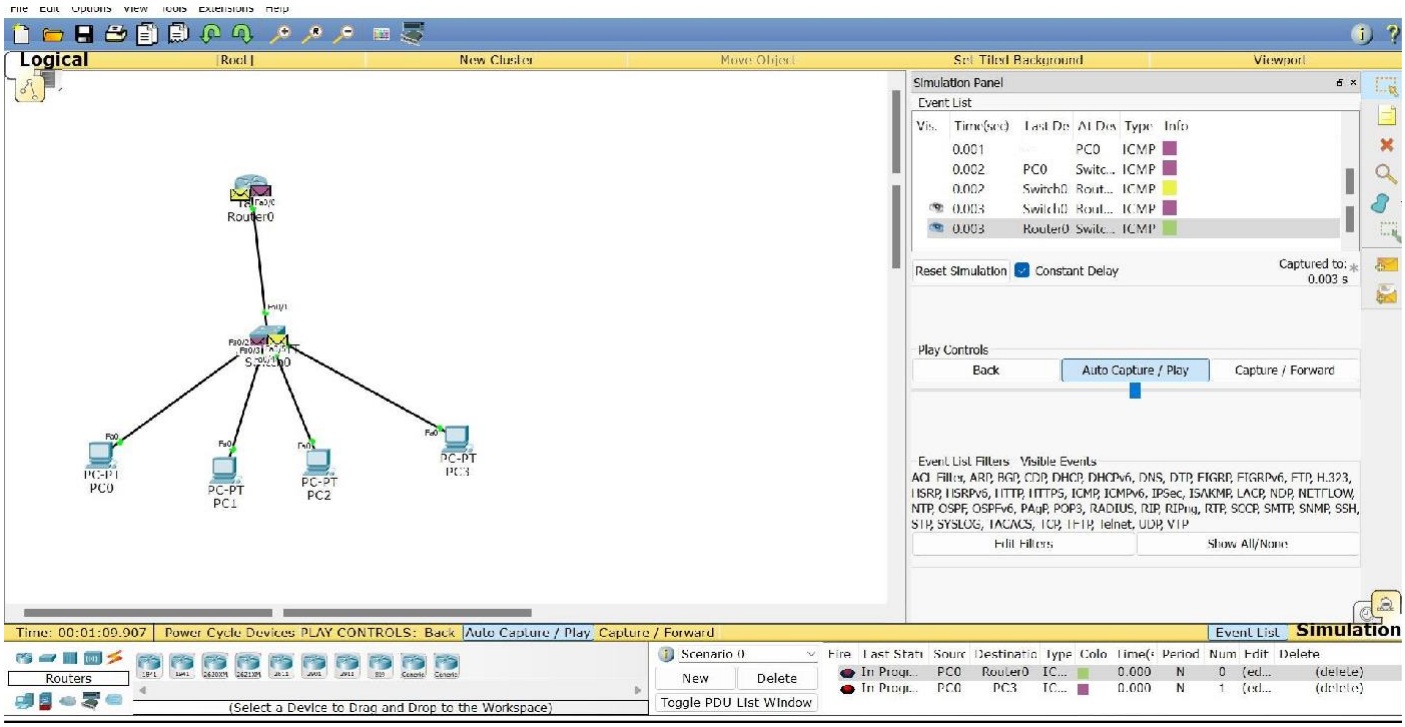
Packet Tracer PC command line 1.0
PC> Ping 192.168.20.3

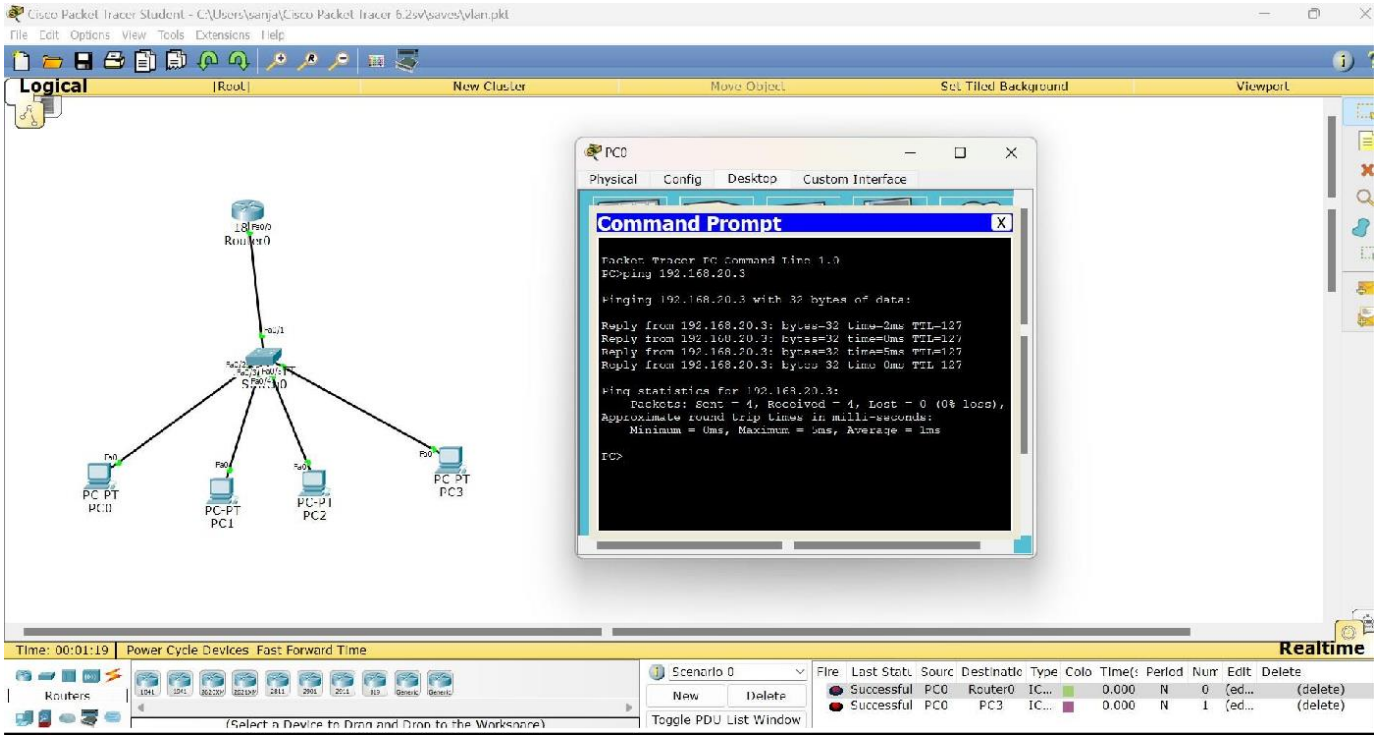
Pinging 192.168.20.3 with 32 bytes of data
Request timed out
Reply from 192.168.20.3: bytes=32 time=0ms TTL=127
Reply from 192.168.20.3: bytes=32 time=0ms TTL=127
Reply from 192.168.20.3: bytes=32 time=0ms TTL=127
Ping statistics for 192.168.20.3
Packets: Sent=4, Received=3, Lost=1 (25% loss),
Approximate round trip times in milliseconds
Minimum=0ms, Maximum=5ms, Average=1ms

Observation

- We can have same device on one VLAN & another on another VLAN connected to the same switch. They will only hear other broadcast traffic from within their VLANs, as if they were connected to two switches.
- The VLANs doesn't use IP address instead deal with subnets / class type address

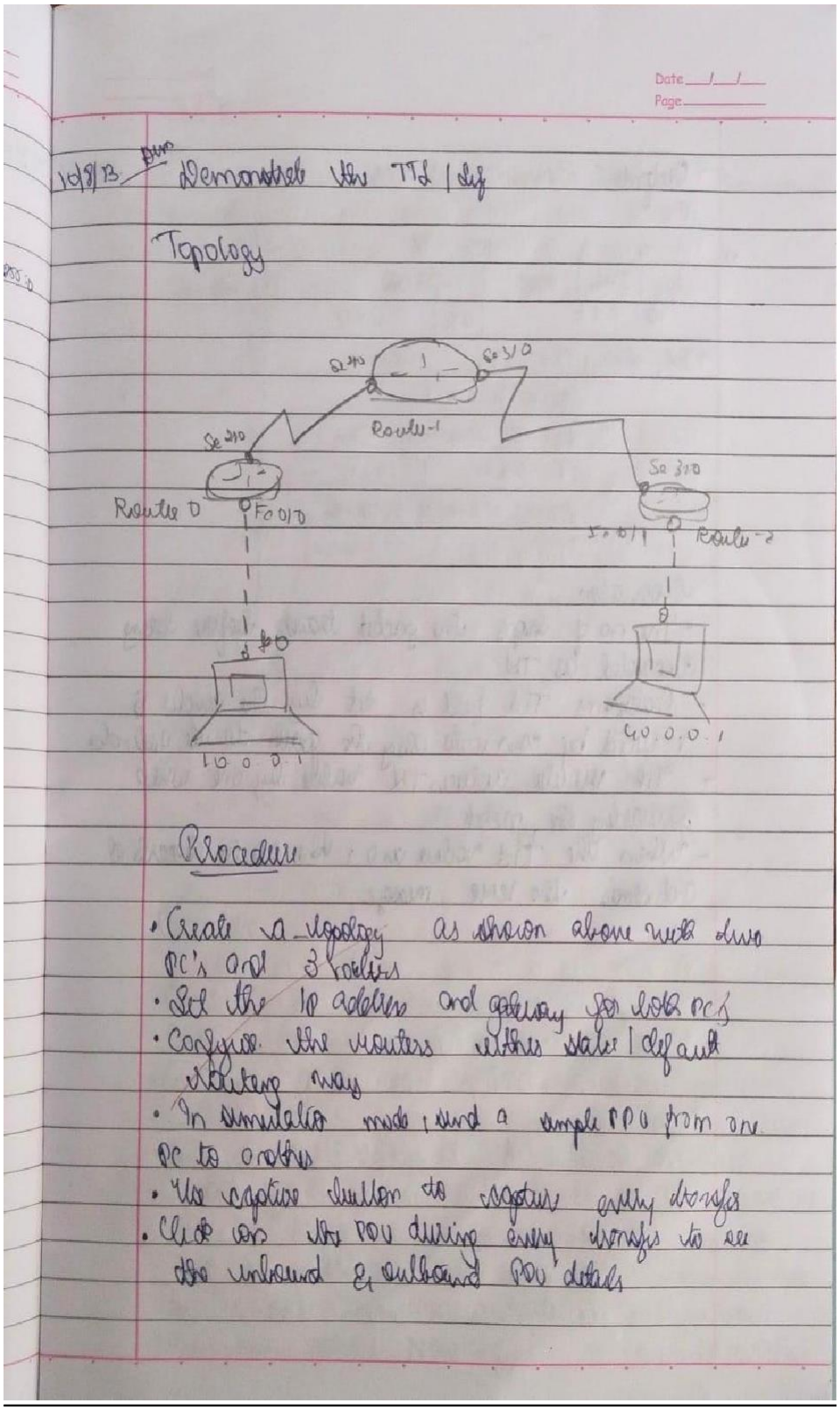
TOPOLOGY & OUTPUT





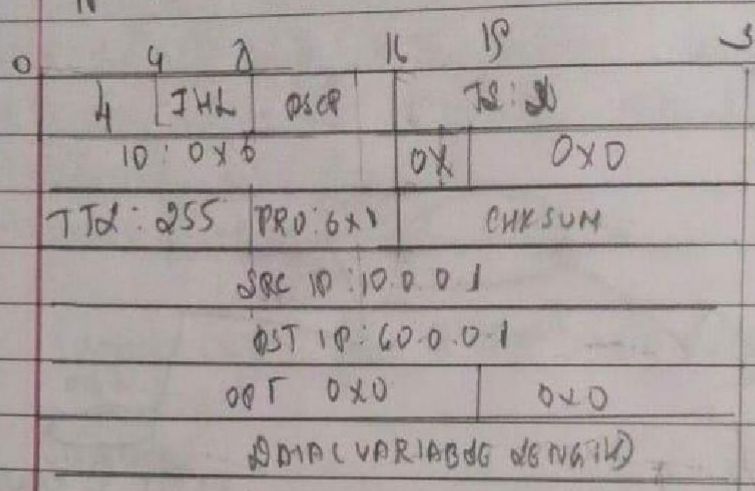
EXPERIMENT-10

Q) Demonstrate the TTL/ Life of a Packet



Output

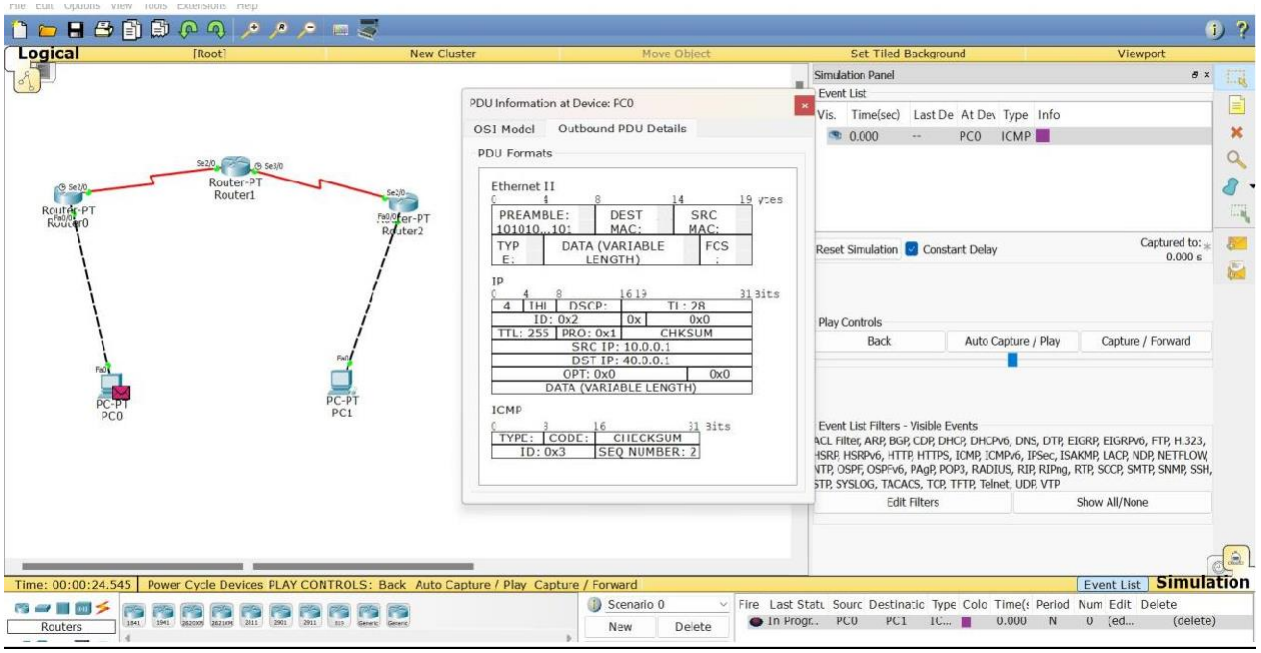
IP

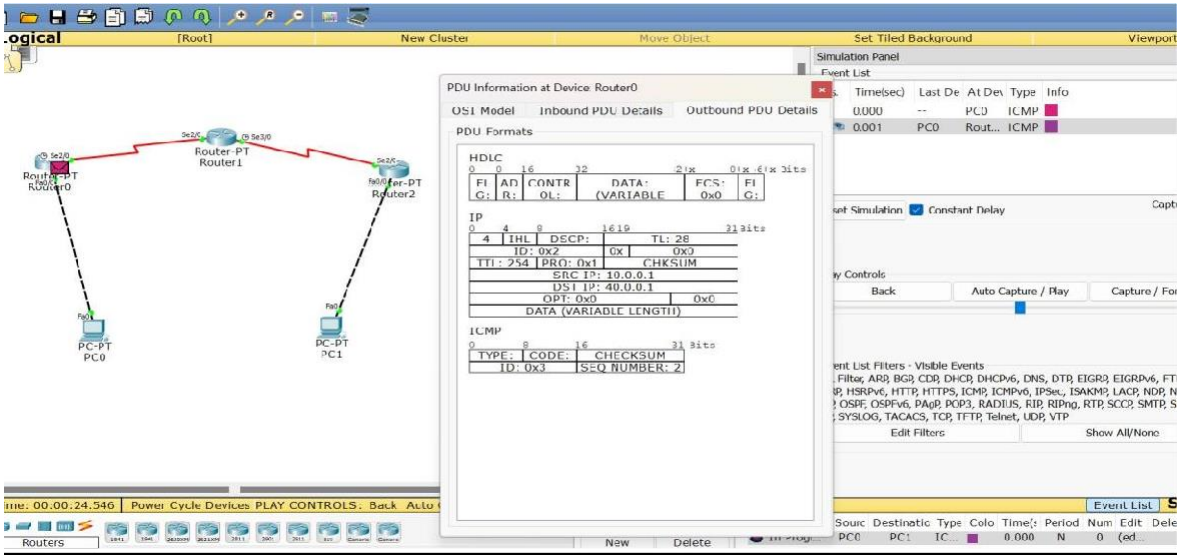


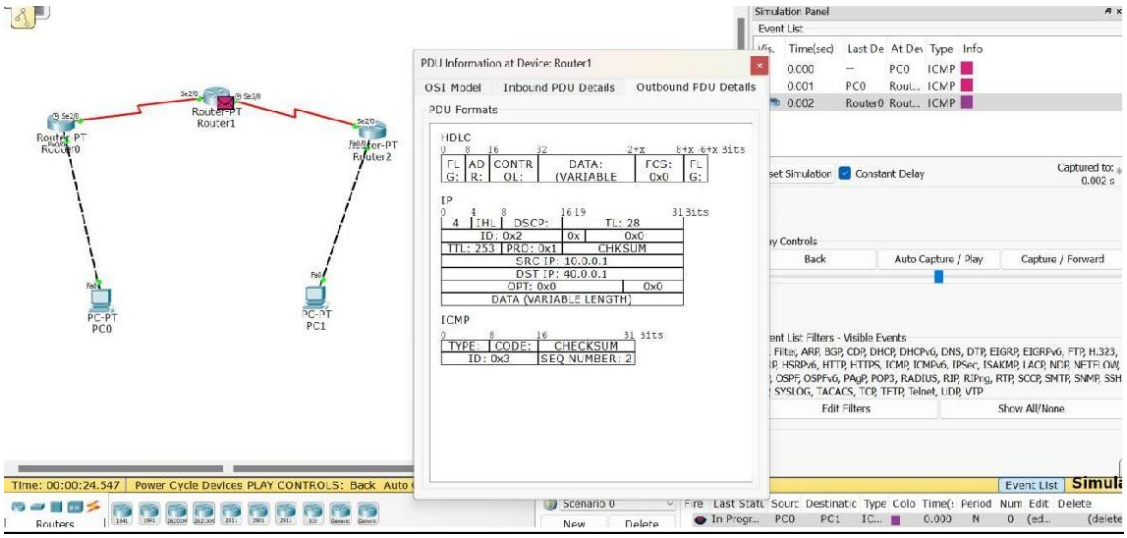
Observation:

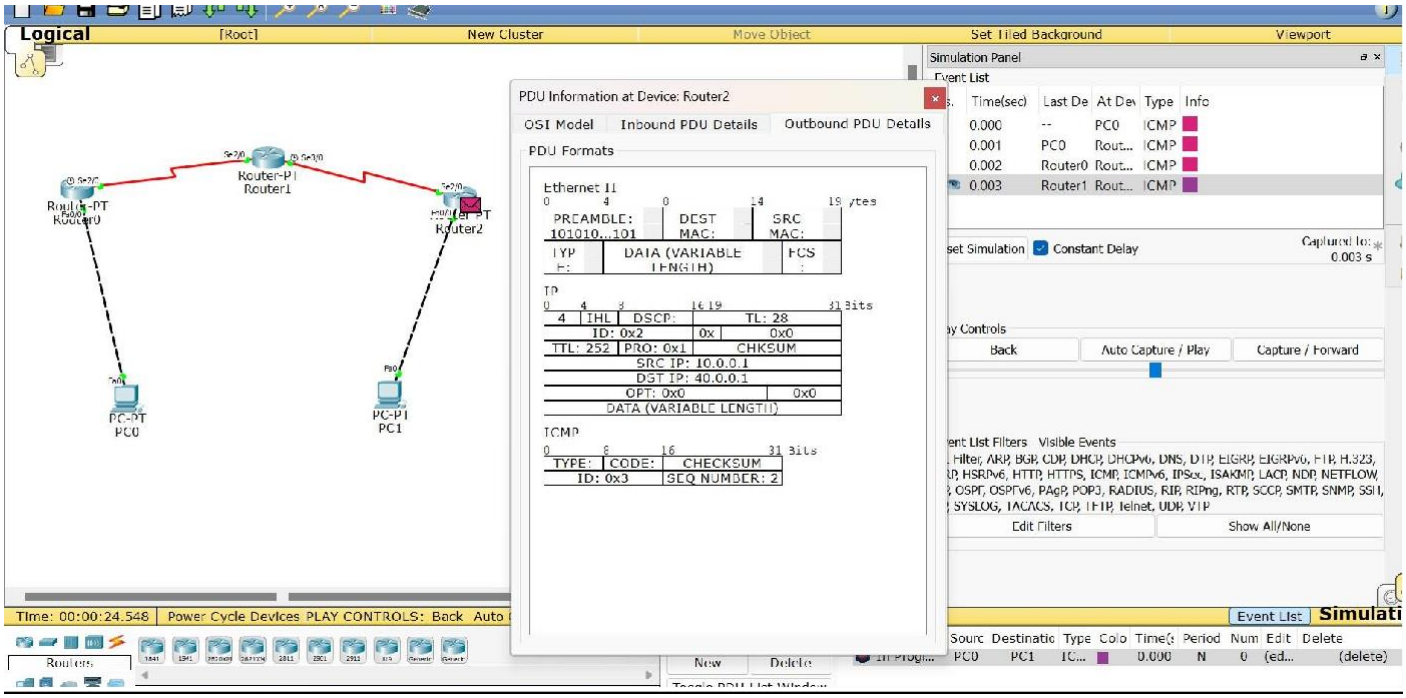
- The no of hops the packet travel before being discarded as TTL
- Datagrams TTL field is set by the sender & reduced by each router along the path to its destination
- The router reduces TTL value by one when forwarding the packet
- When the TTL value is 0, the router discards it and sends the ICMP message.

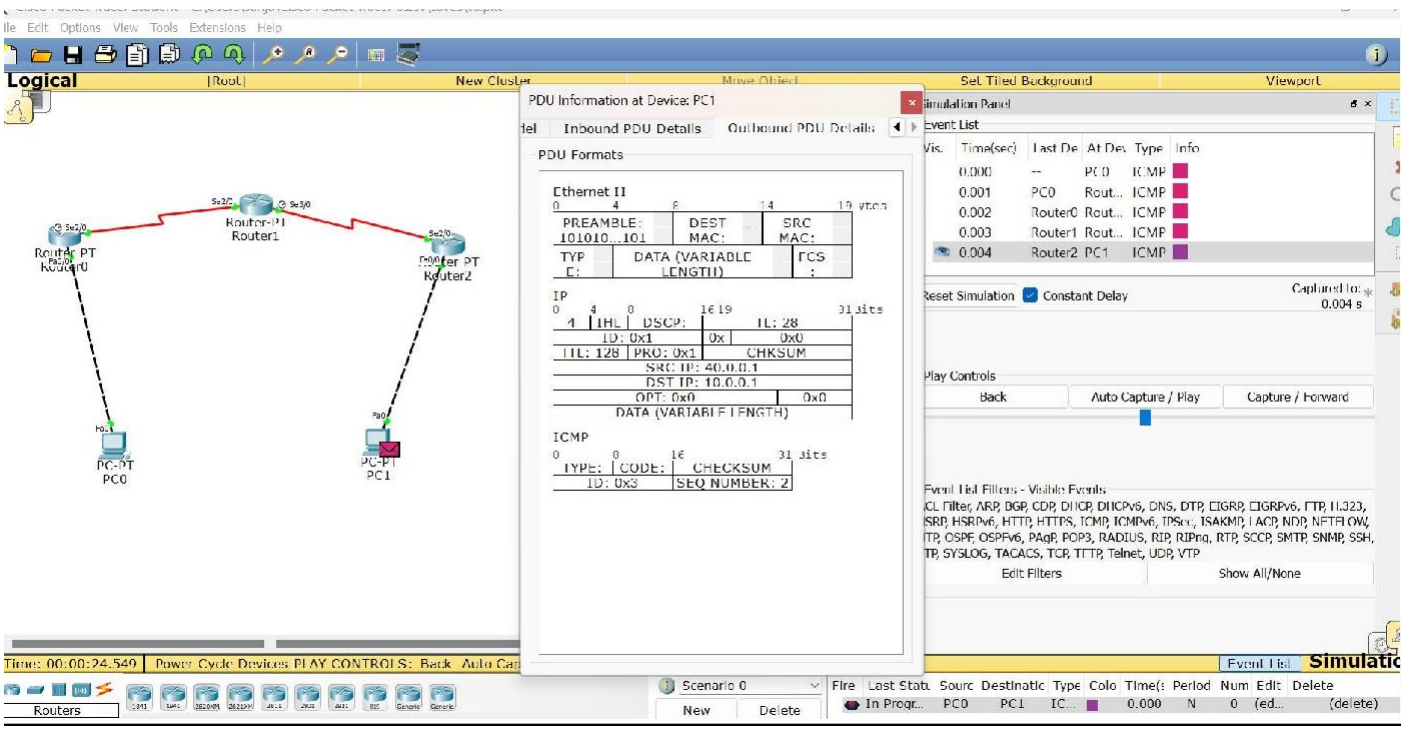
TOPOLOGY & OUTPUT





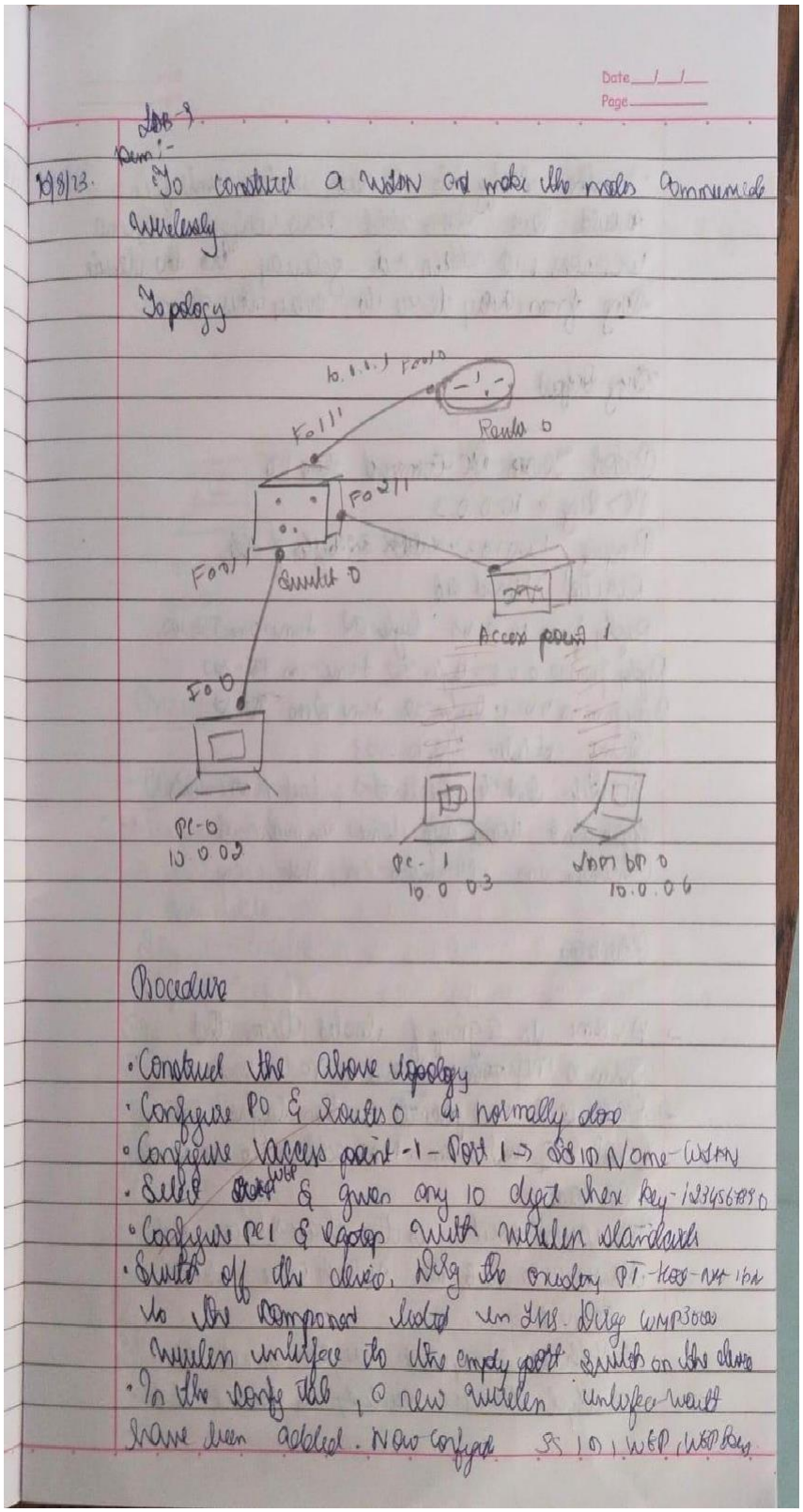



Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
PDU Information at Device: Router2
OSI Model
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
PREAMBLE: 101010...101
DEST MAC:
SRC MAC:
DATA (VARIABLE LENGTH)
IP
ICMP
TYPE:
CODE:
CHECKSUM
ID: 0x3
SEQ NUMBER: 2
Simulation Panel
Event List
Time(sec)
Last De
At Dev
Type
Info
Router-PT Router0
Router-PT Router1
Router-PT Router2
PC-PT PC0
PC-PT PC1
Time: 00:00:24.548
Power Cycle Devices
PLAY CONTROLS:
Back
Auto
Routers
New
Delete
Event List
Simulati


Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
PDU Information at Device: PC1
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
PREAMBLE: 101010...101
DEST MAC:
SRC MAC:
DATA (VARIABLE LENGTH)
IP
ICMP
TYPE:
CODE:
CHECKSUM
ID: 0x3
SEQ NUMBER: 2
Simulation Panel
Event List
Time(sec)
Last De
At Dev
Type
Info
Router-PT Router0
Router-PT Router1
Router-PT Router2
PC-PT PC0
PC-PT PC1
Time: 00:00:24.549
Power Cycle Devices
PLAY CONTROLS:
Routers
Scenario 0
New
Delete
Event List
Simulatio

EXPERIMENT-11

Q) To construct a WLAN and make the nodes communicate wirelessly



Date / /
Page

- In the config tab a new wireless interface would have been added. Now configure SSID, two WEP keys, IP address and gateway to the devices
- Ping from every device to every other device

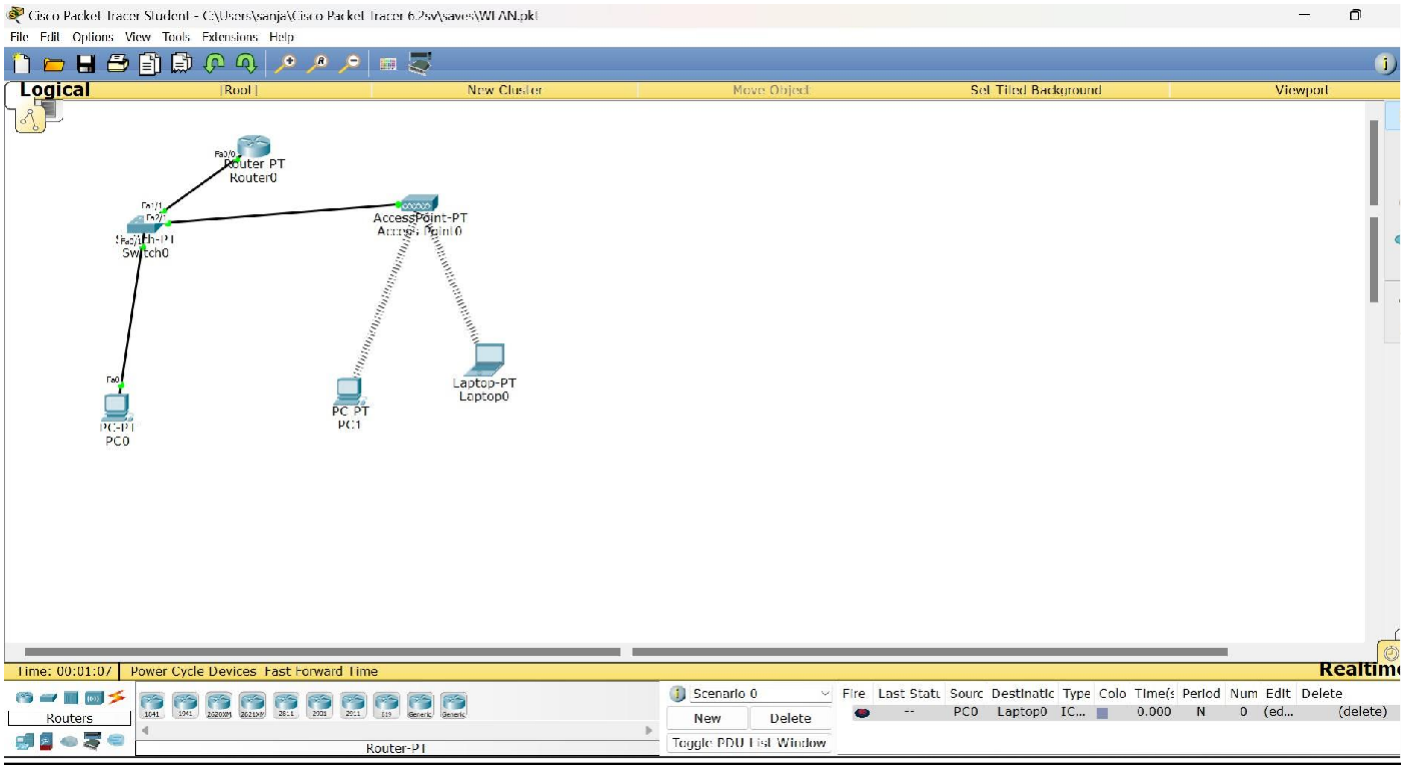
Ping output

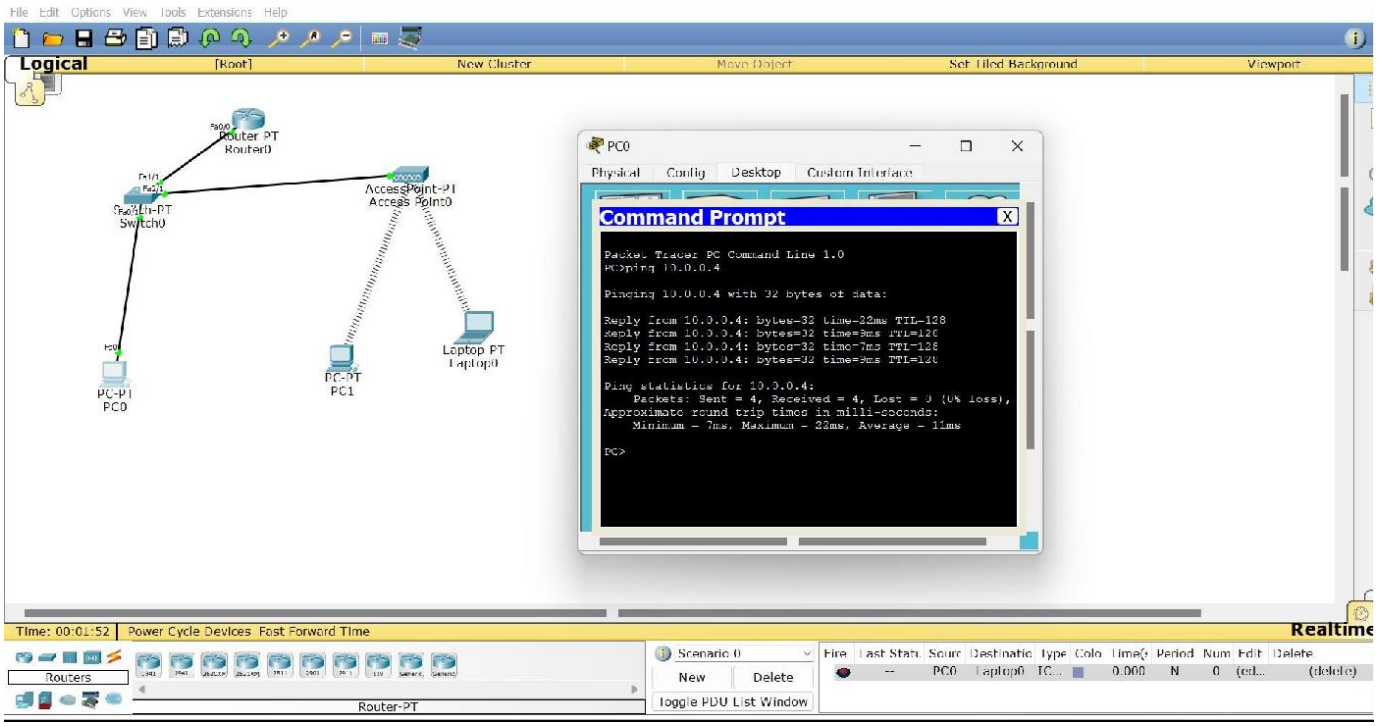
Packet Tracer PC Command Line 1.0
PC> Ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data
Request timed out
Reply from 10.0.0.3 : bytes=32 time=0ms TTL=128
Reply from 10.0.0.3 : bytes=32 time=0ms TTL=128
Reply from 10.0.0.3 : bytes=32 time=0ms TTL=128
Ping statistics for 10.0.0.3
Packets: Sent=4, Received=3, Lost=1 (25% loss)
Approximate round trip times in milliseconds
Minimum=0ms, Maximum=1ms, Average=0ms

Observation

→ Network is a group of linked devices that form a network based on radio transmission
→ Data sent in packets contain layers with labels & instructions. MAC address like envelopes for routing
→ The access point is the transmitter that serves as a hub to which other devices connect
- With one access point we can connect to multiple devices wirelessly & transmit data

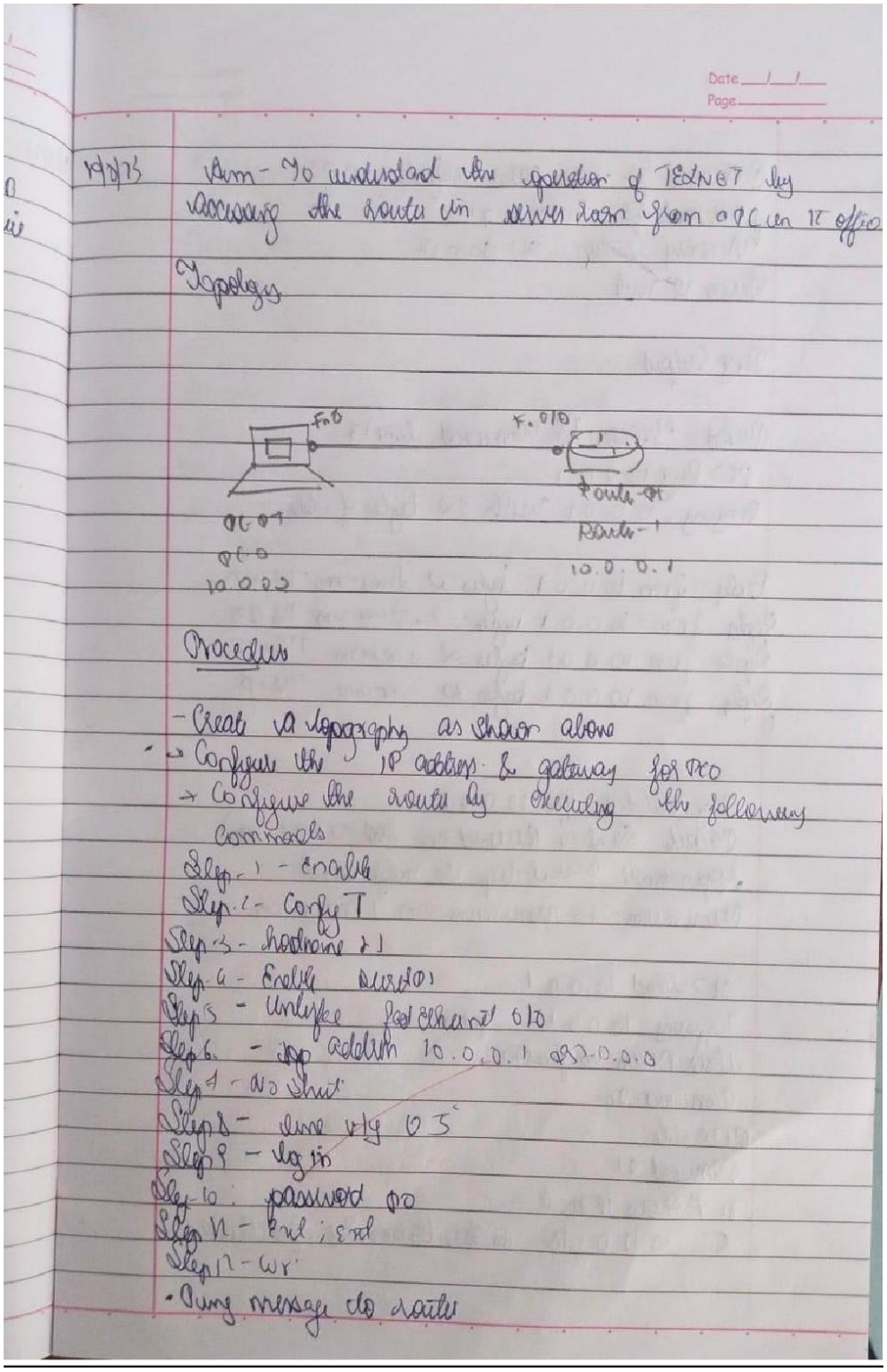
TOPOLOGY & OUTPUT





EXPERIMENT-12

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



Password for user exec verification on R0.
password for enable on R1
Access route R1 from R2.
Show ip route

Ping Output

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=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 in milli seconds
Minimum=0ms, Maximum=0ms, Average=0ms

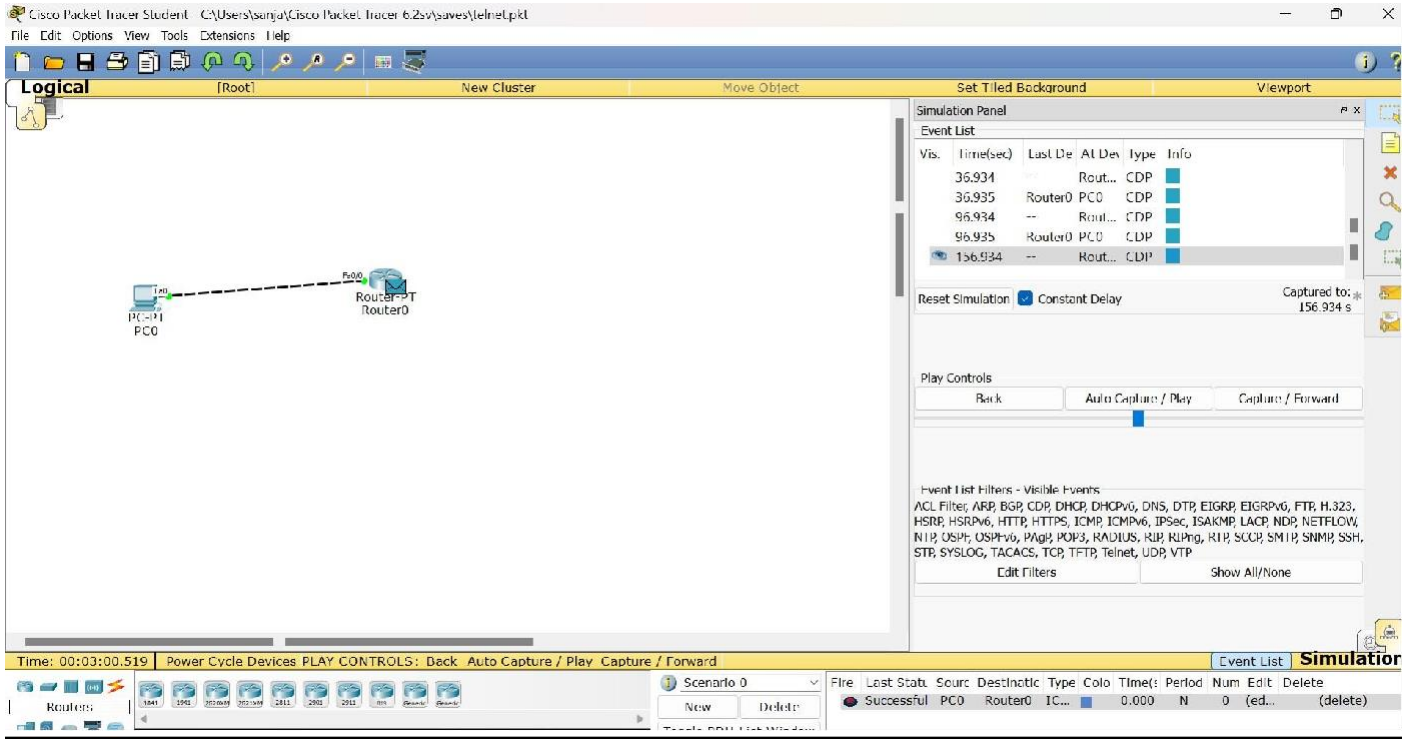
R2> telnet 10.0.0.1
Trying 10.0.0.1 ... open
User Access Verification
Password:
R1>enable
Password:
R1# show ip route
C 10.0.0.0/8 is directly connected, FastEthernet 0/0

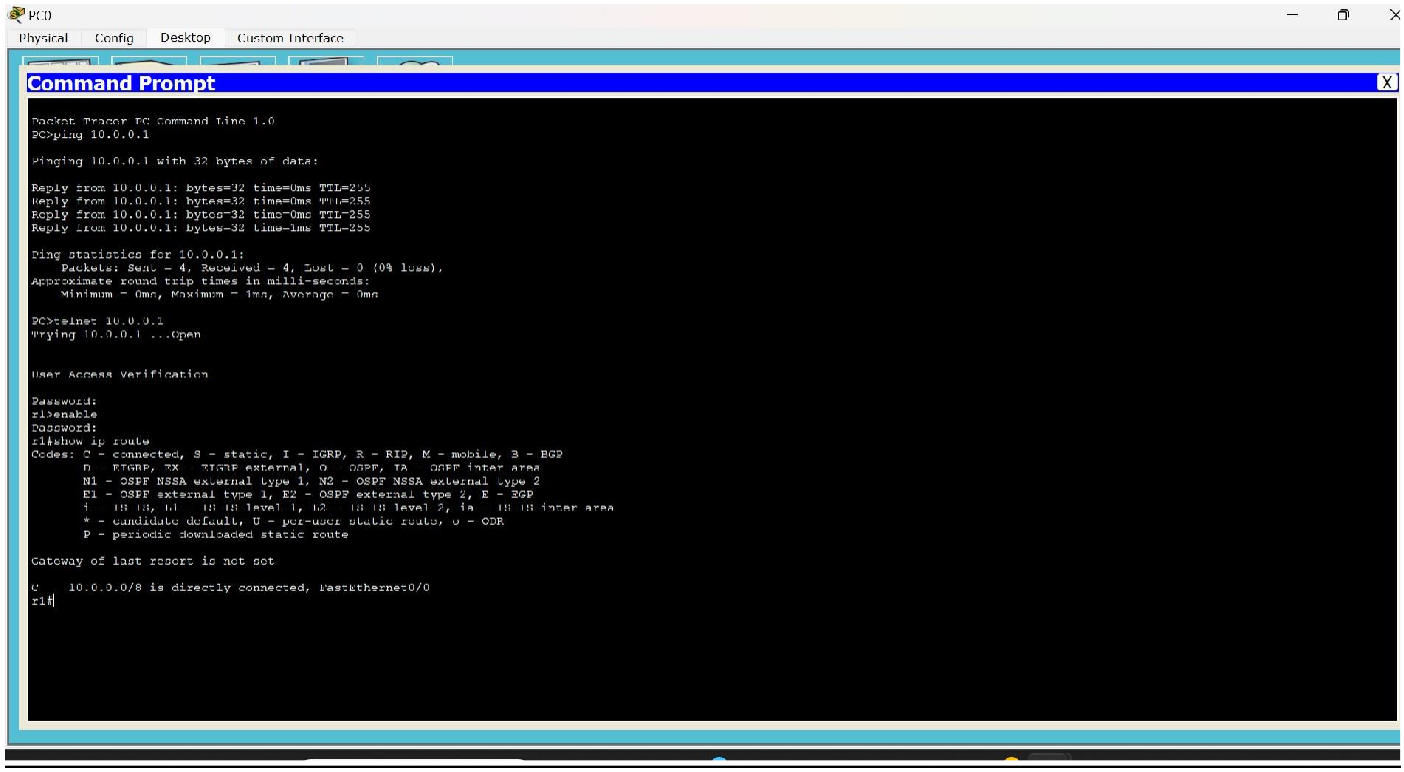
Date __/__/__
Page _____

Observation

- Telnet stands for Teletype Network. It is a type of protocol that enables one computer to connect to the local computer
- Used as standard TCP/IP protocol for virtual terminal services provided by ISO
- During TELNET operation, whatever is being performed on the remote computer will be displayed by the local computer
- Operates on a client/server principle

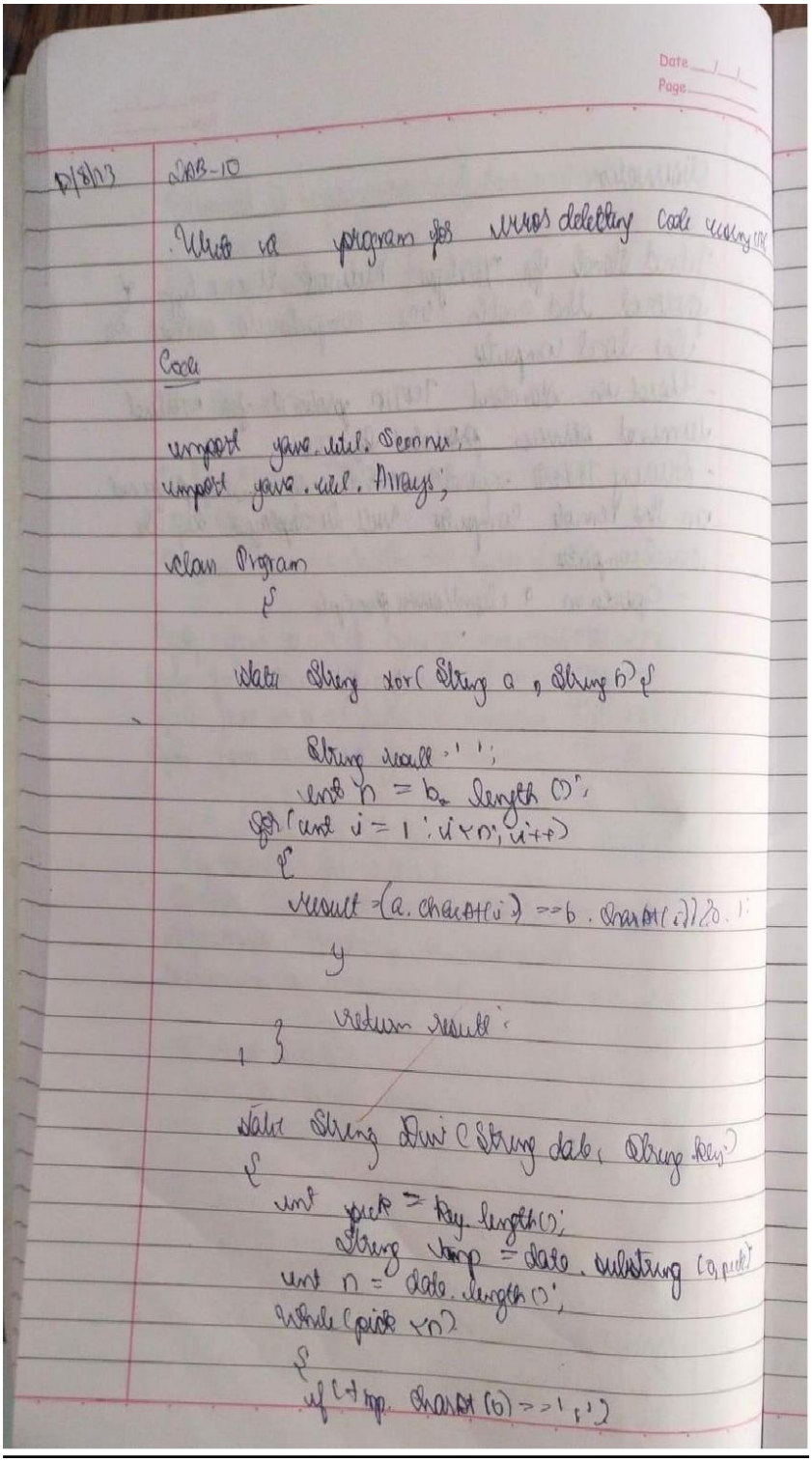
TOPOLOGY & OUTPUT



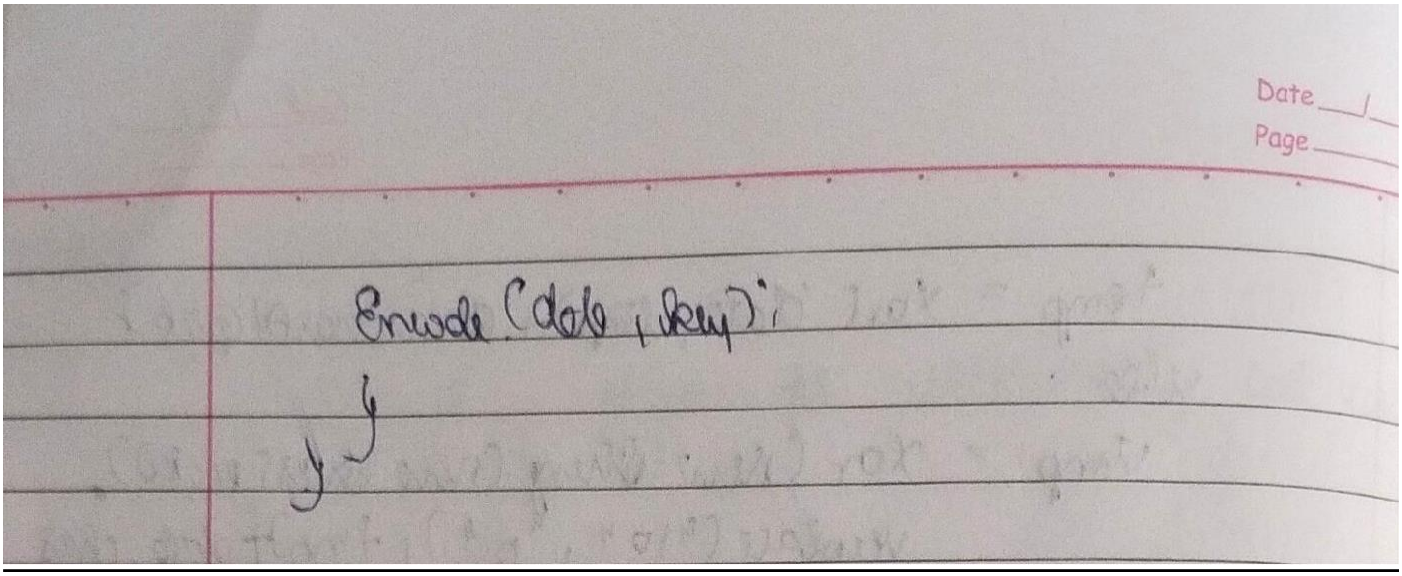


EXPERIMENT-13

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



```
tmp = xor('data', tmp) + data.charAt(pick);
else
tmp = xor(new String(new char[pick]).
      replace('\0', '0'), tmp) + data.char
      -At(pick);
      pick++;
    }

if (tmp.charAt(0) == '1')
    tmp = xor(key, tmp);
else
    tmp = xor(new String(new char[pick]).
        replace('\0', '0'), tmp);
    return tmp;
}

static void Encode(String data, String key)
{
int lkey = key.length();
String ad = (data + new String(new char[lkey-1])
        .replace('\0', '0'));
String remainder = Mod2Div(ad, key);
String codeword = data + remainder;
System.out.println("Remainder : "+remainder);
System.out.println("Encoded Data (Data+Remainder)
    : "+ codeword + "\n");
}

public static void main(String args[]){
Scanner s = new Scanner(System.in);
System.out.println("Enter dataword & key");
String data = s.next();
String key = s.next();
```



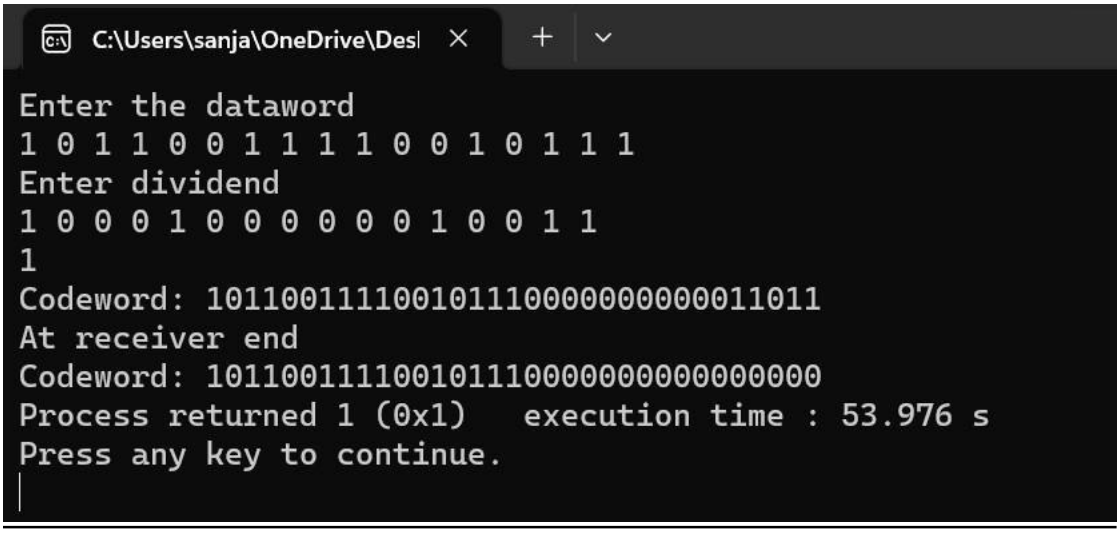
CODE-

```
import java.util.Scanner;
import java.util.Arrays;

class Program {
   static String Xor(String a, String b) {
      String result = "";
      int n = b.length();
      for (int i = 1; i < n; i++) {
         result=(a.charAt(i) == b.charAt(i))?0:1;

      }
      return result;
   }

   static String Div(String data, String key) {
      int pick = key.length();
      String tmp = data.substring(0, pick);
```

```
    int n = data.length();
    while (pick < n) {
        if (tmp.charAt(0) == '1')
            tmp = Xor(data, tmp) + data.charAt(pick);
        else
            tmp = Xor(new String(new char[pick]).replace("\0", "0"), tmp) + data.charAt(pick);
        pick += 1;
    }
    if (tmp.charAt(0) == '1')
        tmp = Xor(divisor, tmp);
    else
        tmp = Xor(new String(new char[pick]).replace("\0", "0"), tmp);
    return tmp;
}

static void Encode(String data, String key) {
    int lkey = key.length();
    String appended_data = (data + new String(new char[lkey - 1]).replace("\0", "0"));
    String remainder = Mod2Div(appended_data, key);
    String codeword = data + remainder;
    System.out.println("Remainder : " + remainder);
    System.out.println("Encoded Data (Data + Remainder) :" + codeword + "\n");
}

public static void main(String[] args) {
    Scanner s = new Scanner(System.in);
```

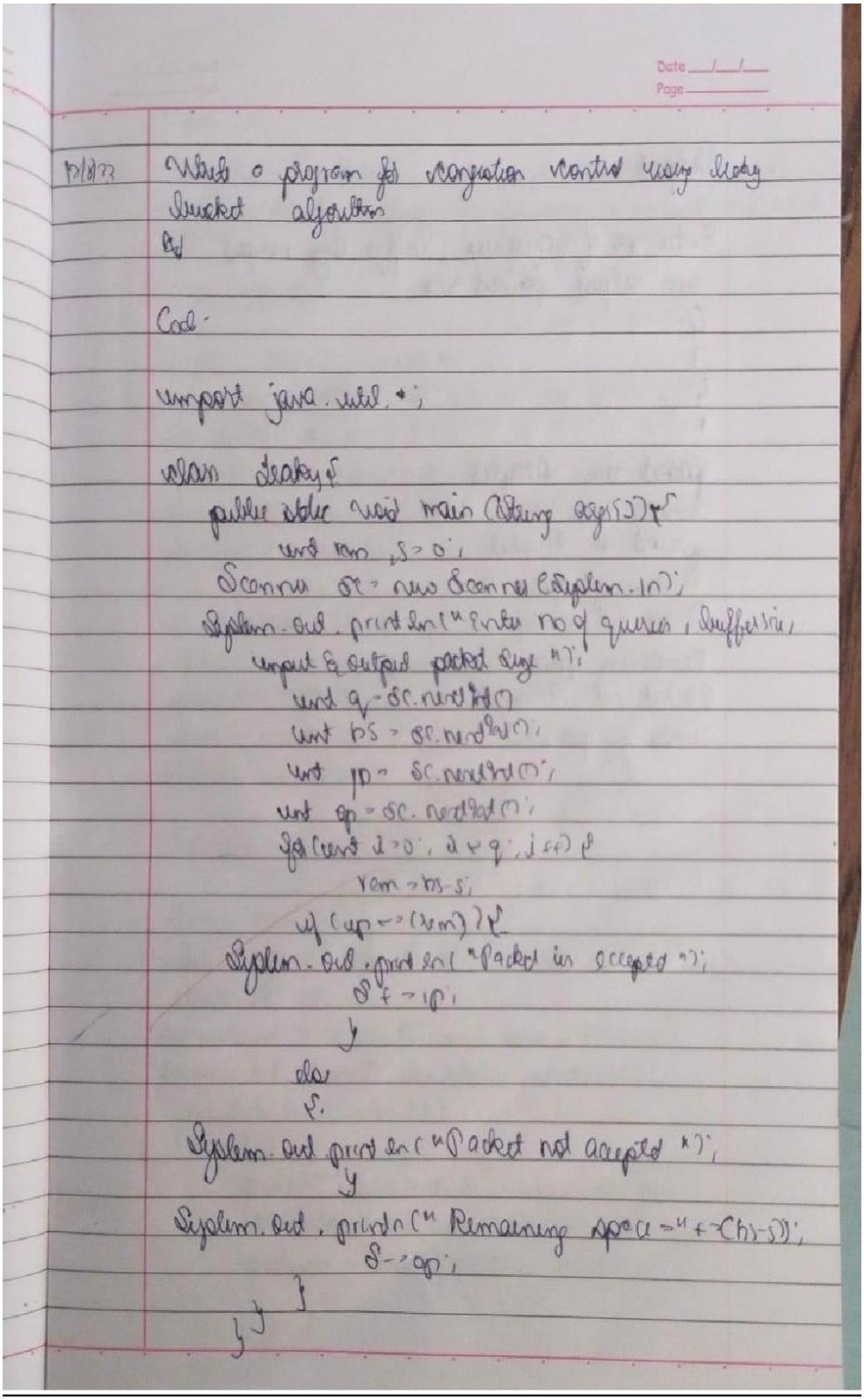
```
        System.out.println("enter dataword and key");
        String data = s.next();
        String key = s.next();

        EncodeData(data, key);
    }
}
```

OUTPUT



EXPERIMENT-14

Q) Write a program for congestion control using Leaky bucket algorithm



```
Date __/__/__
Page _____

Write a program for congestion control using leaky bucket algorithm

Code:

import java.util.*;

class Leaky {
    public static void main(String args[]) {
        int rem, s = 0;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter no of queues, buffer size,
            input & output packet size");
        int q = sc.nextInt();
        int bs = sc.nextInt();
        int ip = sc.nextInt();
        int op = sc.nextInt();
        for(int i=0; i<q; i++) {
            rem = bs - s;
            if(ip <= (rem)) {
                System.out.println("Packet is accepted");
                s = ip;
            }
            else
                System.out.println("Packet not accepted");
            System.out.println("Remaining space = " + (bs-s));
            s -= op;
        }
    }
}
```

CODE-

```
import java.util.*;
 class Leakybucket {
    public static void main(String[] args)
    {

       int  rem;
     Scanner sc=new Scanner(System.in);
     int s= 0;
      System.out.println("enter no of queries,buffer size,input and output packet size ");
      int q=sc.nextInt();
        int bs=sc.nextInt();
           int ip=sc.nextInt();
              int op=sc.nextInt();
       for (int i = 0; i < q; i++) {
           rem=bs-s;
           if (ip <= (rem)) {
               System.out.println("packet is accepted");
           s+=ip;
           }
           else {
               System.out.println("Packet not accepted ");
           }
           System.out

.println("remaining space="+(bs-s));
           s -= op;
       }
```

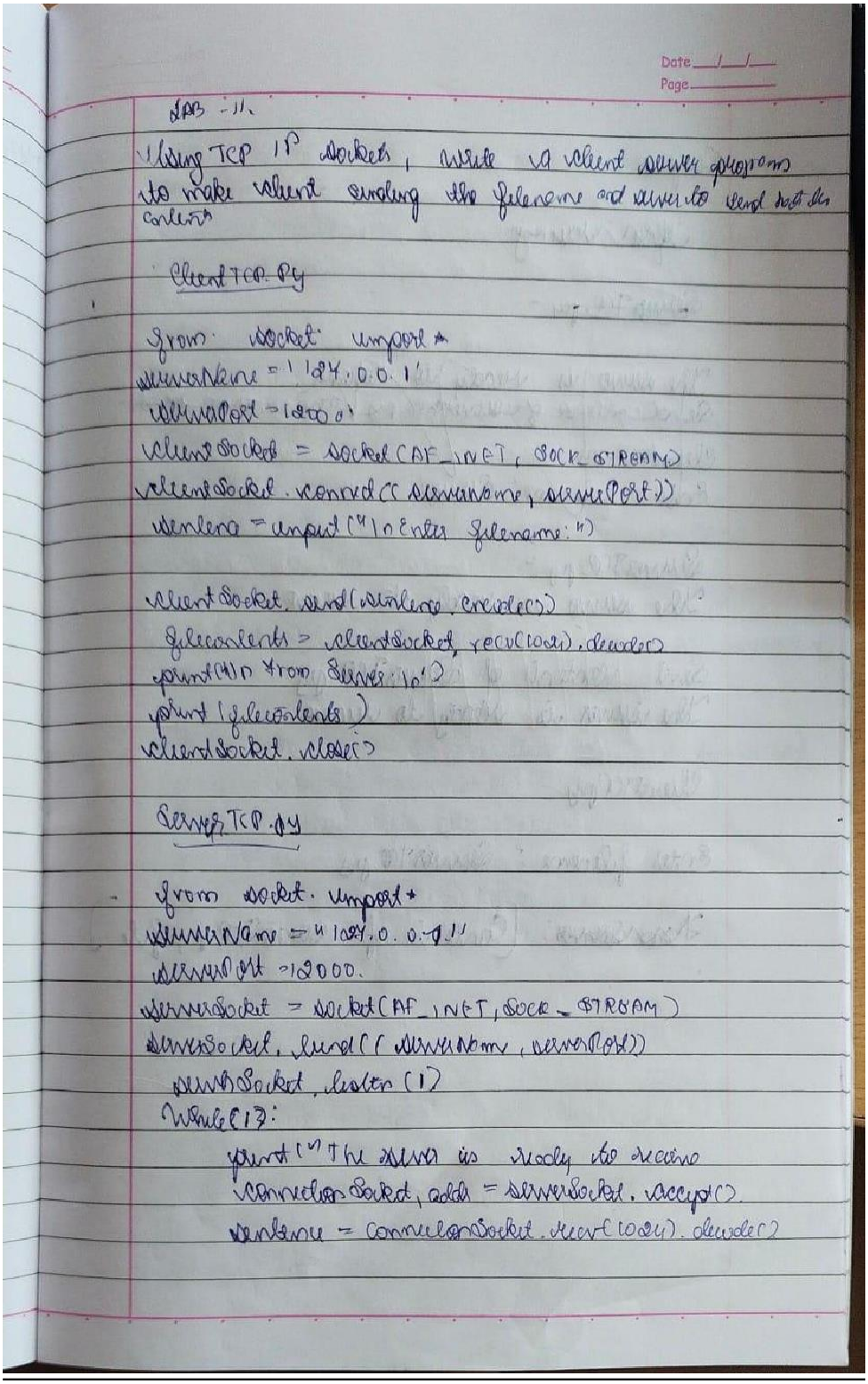
```
    }
}
```

OUTPUT

```
PS C:\Users\sanja> cd C:\Users\sanja\OneDrive\Documents
PS C:\Users\sanja\OneDrive\Documents> javac Leakybucket.java
PS C:\Users\sanja\OneDrive\Documents> java Leakybucket
enter no of queries,buffer size,input and output packet size
4 10
6
1
packet is accepted
remaining space=4
Packet not accepted
remaining space=5
packet is accepted
remaining space=0
Packet not accepted
remaining space=1
PS C:\Users\sanja\OneDrive\Documents>
```

EXPERIMENT-15

Q) 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.



Date __/__/__
Page _____

Prob - 11.

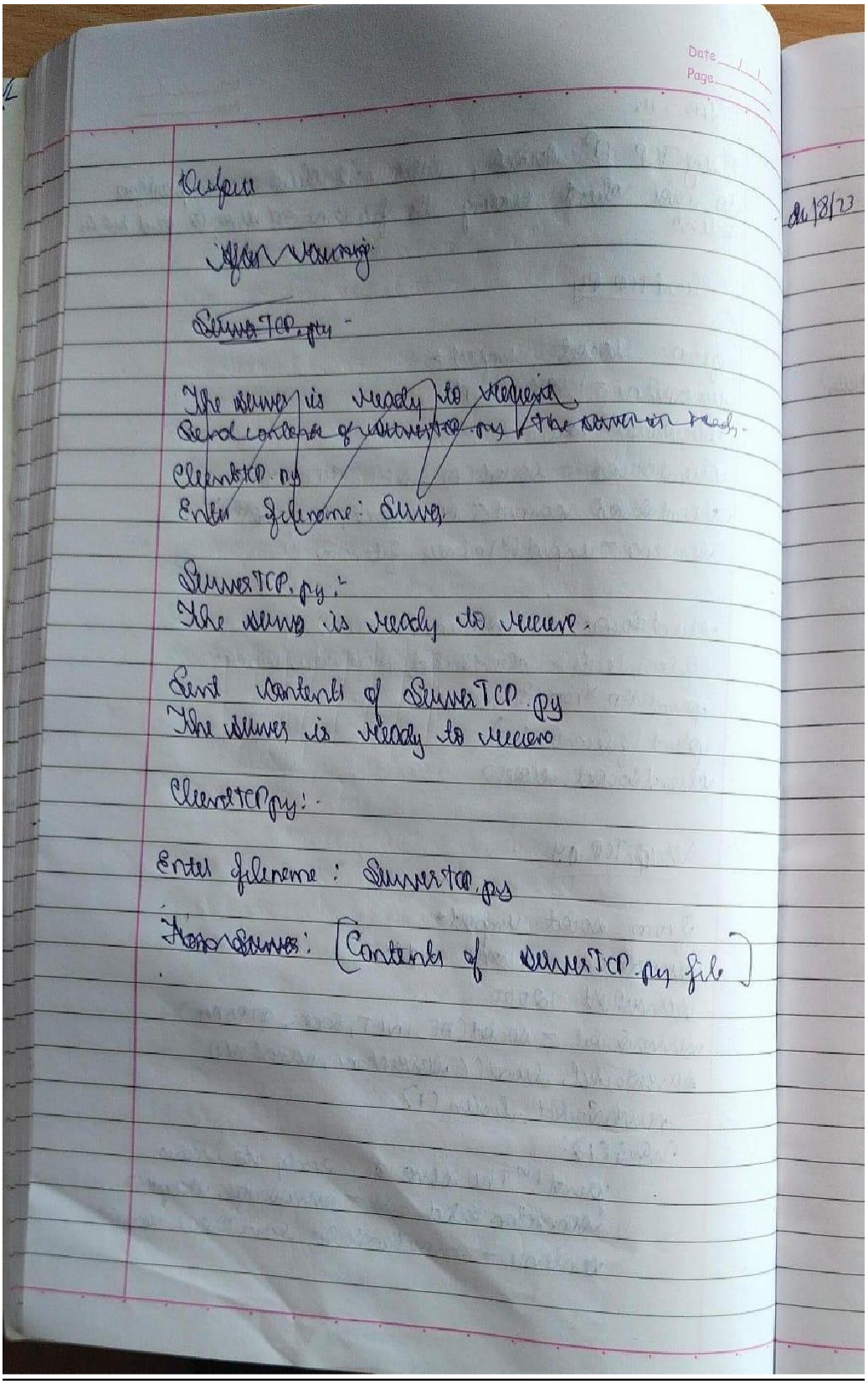
Using TCP IP sockets, write a client server program to make client sending the filename and server to send back the contents

Client TCP.py

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

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

Server TCP.py

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



Output

After running:

ServerTCP.py:-

The server is ready to receive
Send contents of ServerTCP.py
ClientTCP.py
Enter filename: Server

ServerTCP.py:-
The server is ready to receive.

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

ClientTCP.py:-

Enter filename: ServerTCP.py

From Server: [Contents of ServerTCP.py file]

04/8/23

CODE-

ClientTCP.py

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

clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print ("\nFrom Server:\n")
print(filecontents)
clientSocket.close()
```

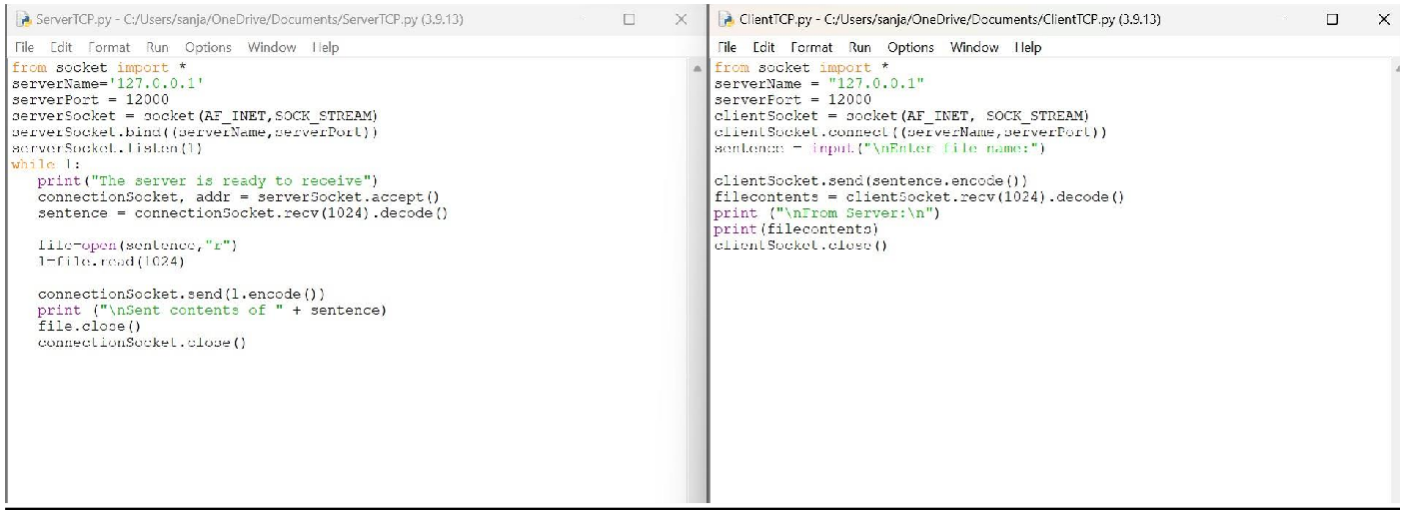
ServerTCP.py

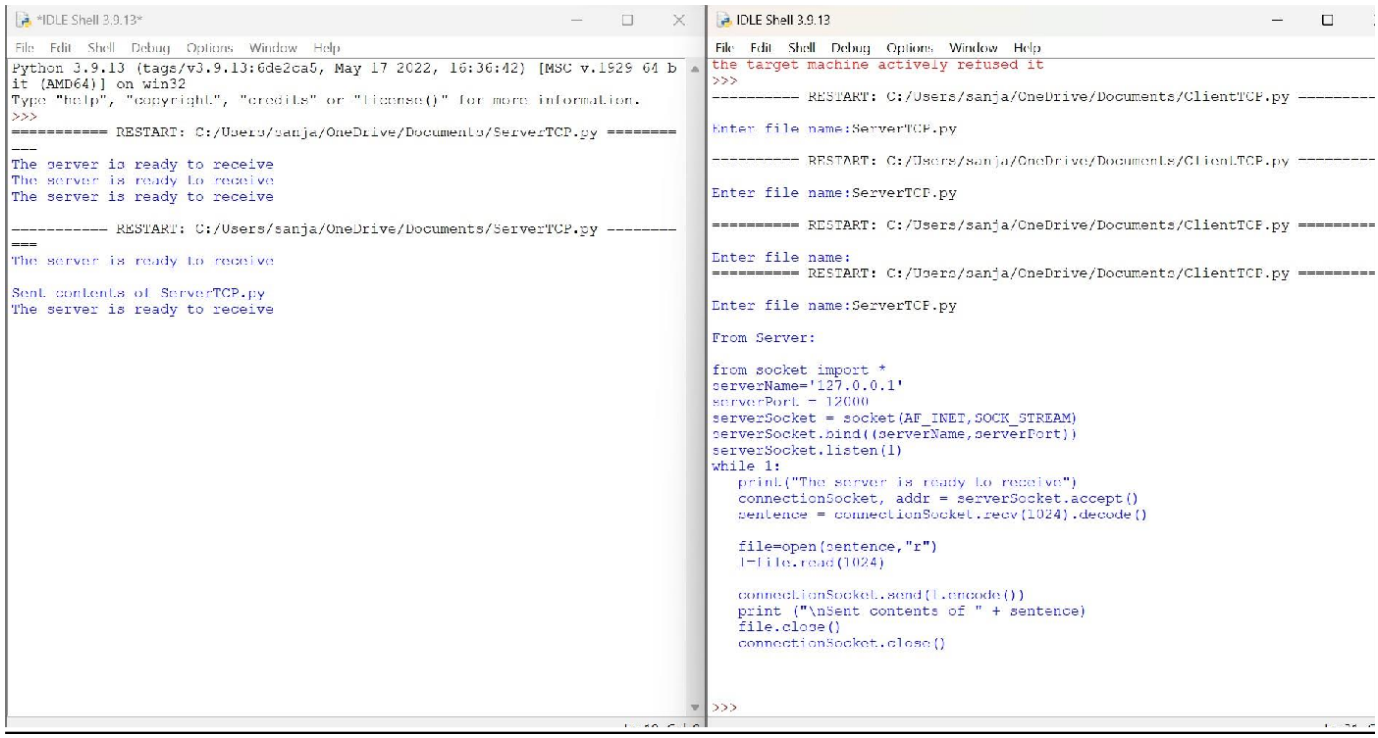
```
from socket import *

serverName='127.0.0.1'

serverPort = 12000

serverSocket = socket(AF_INET,SOCK_STREAM)

serverSocket.bind((serverName,serverPort))

serverSocket.listen(1)

while 1:

  print("The server is ready to receive")

  connectionSocket, addr = serverSocket.accept()

  sentence = connectionSocket.recv(1024).decode()


  file=open(sentence,"r")

  l=file.read(1024)


  connectionSocket.send(l.encode())

  print ("\nSent contents of " + sentence)
```

file.close()

connectionSocket.close()

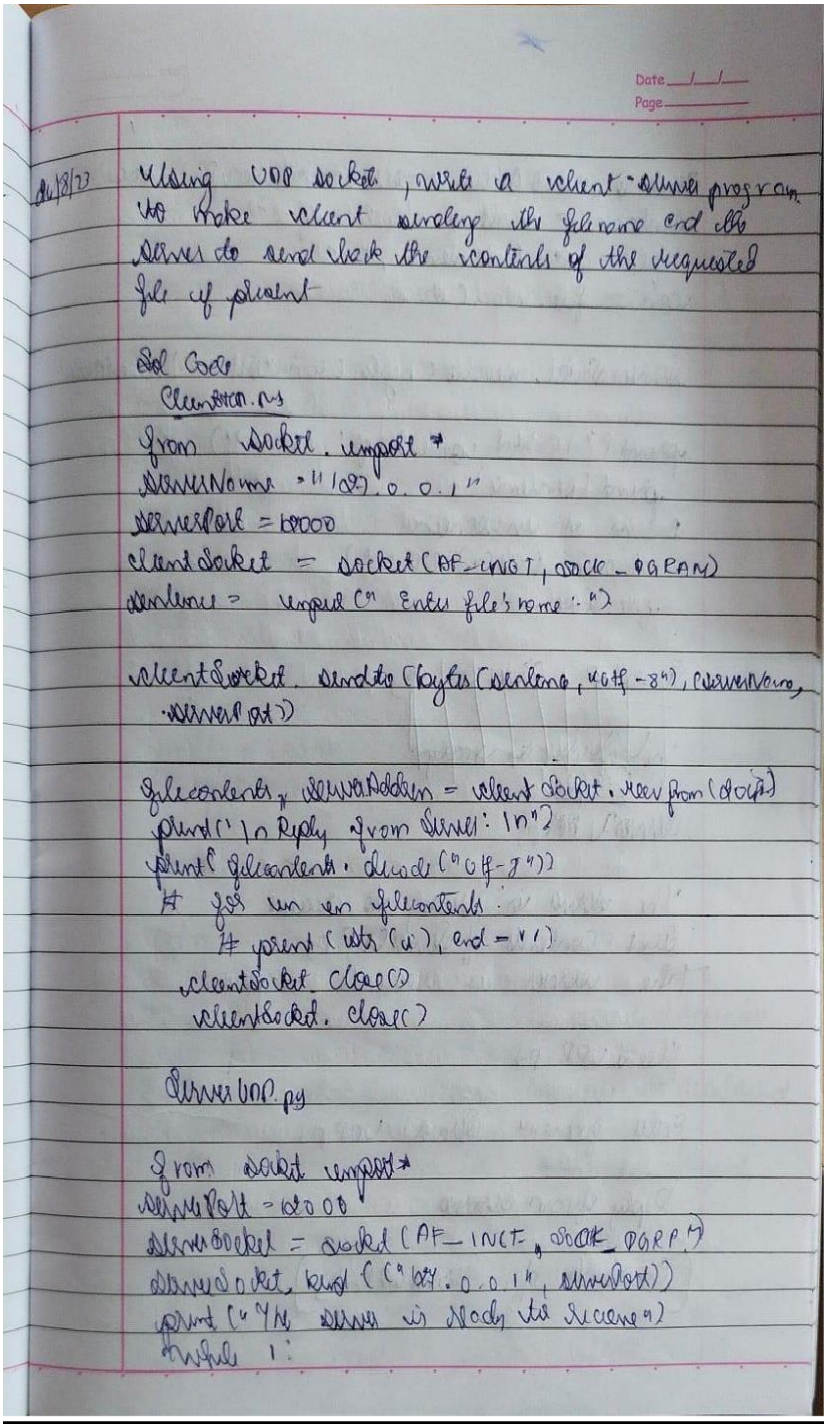
OUTPUT





EXPERIMENT-16

Q) 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



Date / /
Page

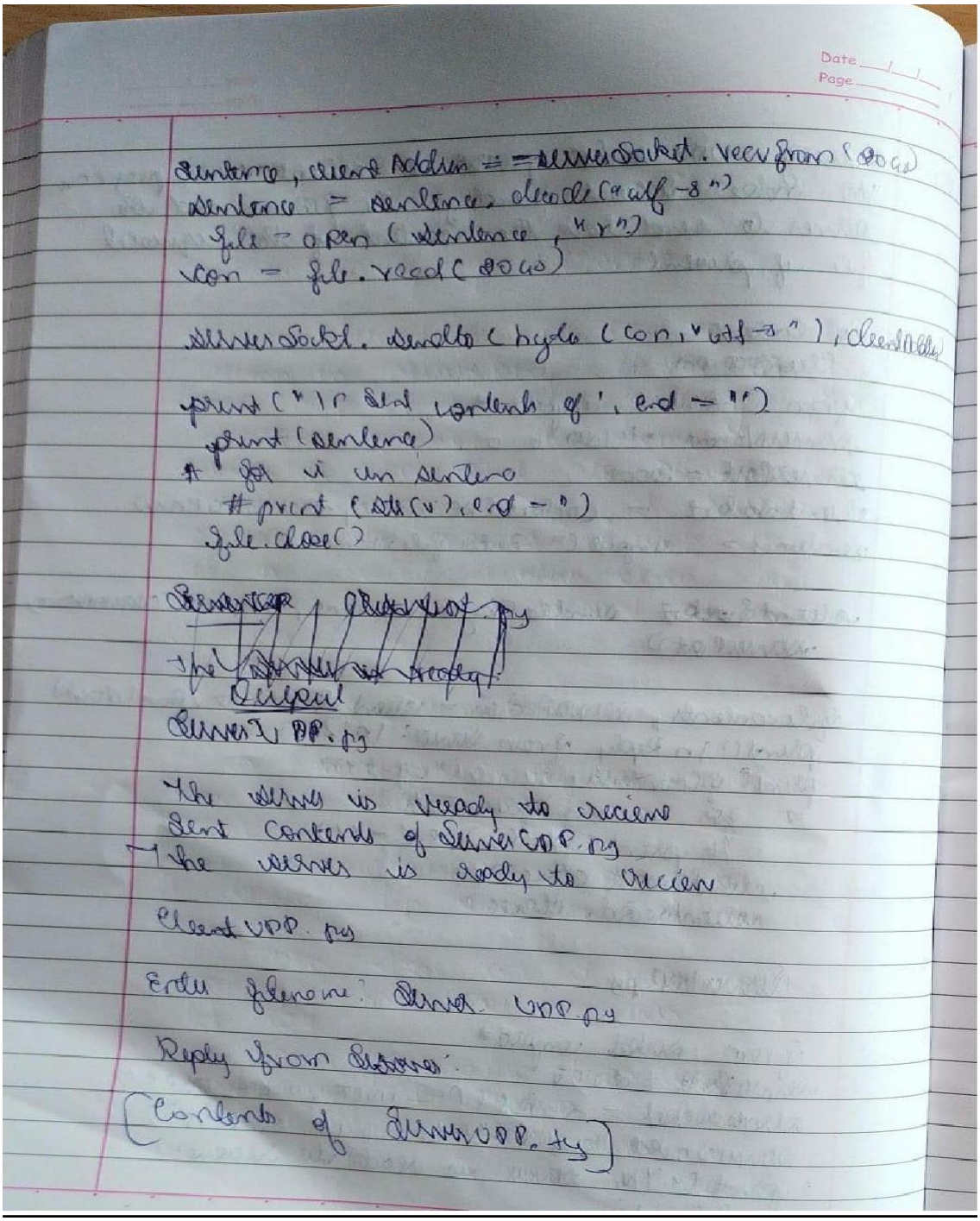
Using UDP socket, write a client-server program to make client sending the filename and the server to send back the contents of the requested file if present

Client.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("Enter file's 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()
```

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:
```



CODE-

ClientUDP.py

```
from socket import *
serverName = "127.0.0.1"
```

```
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("\nEnter file name:")


clientSocket.sendto(bytes(sentence,"utf-8"),(serverName, serverPort))
filecontents,serverAddress = clientSocket.recvfrom(2048)
print ("\nReply from Server:\n")
print (filecontents.decode("utf-8"))
#for i in filecontents:
  #print(str(i), end =")
clientSocket.close()
clientSocket.close()
```

ServerUDP.py

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 1:
   sentence, clientAddress = serverSocket.recvfrom(2048)
   sentence = sentence.decode("utf-8")
   file=open(sentence,"r")
   con=file.read(2048)
   serverSocket.sendto(bytes(con,"utf-8"),clientAddress)
   print ("nSent contents of ", end = "")
   print (sentence)
   # for i in sentence:
    # print (str(i), end = ")
   file.close()
```

OUTPUT

ClientUDP.py - C:/Users/sanja/OneDrive/Documents/ClientUDP.py (3.9.13)

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"))
#for i in filecontents:
  #print(str(i), end ="")
clientSocket.close()
clientSocket.close()
```

ServerUDP.py - C:/Users/sanja/OneDrive/Documents/ServerUDP.py (3.9.13)

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")
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()
```

IDLE Shell 3.9.13

File Edit Shell Debug Options Window Help

```
Python 3.9.13 (tags/v3.9.13:6de2ca5, May 17 2022, 16:36:42) [MSC v.1929 64 b
it (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/sanja/OneDrive/Documents/ClientUDP.py =======
====

Enter file name:
========== RESTART: C:/Users/sanja/OneDrive/Documents/ClientUDP.py =======
====

Enter file name:ServerUDP.py

Reply from Server:

from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file=open(sentence,"r")
    con=file.read(2048)
    serverSocket.sendto(bytes(con,"utf-8"),clientAddress)
    print ("nSent contents of ", end = "")
    print (sentence)
    # for i in sentence:
      # print (str(i), end = ")
    file.close()

>>>
```

Ln: 30 Col: 4

IDLE Shell 3.9.13

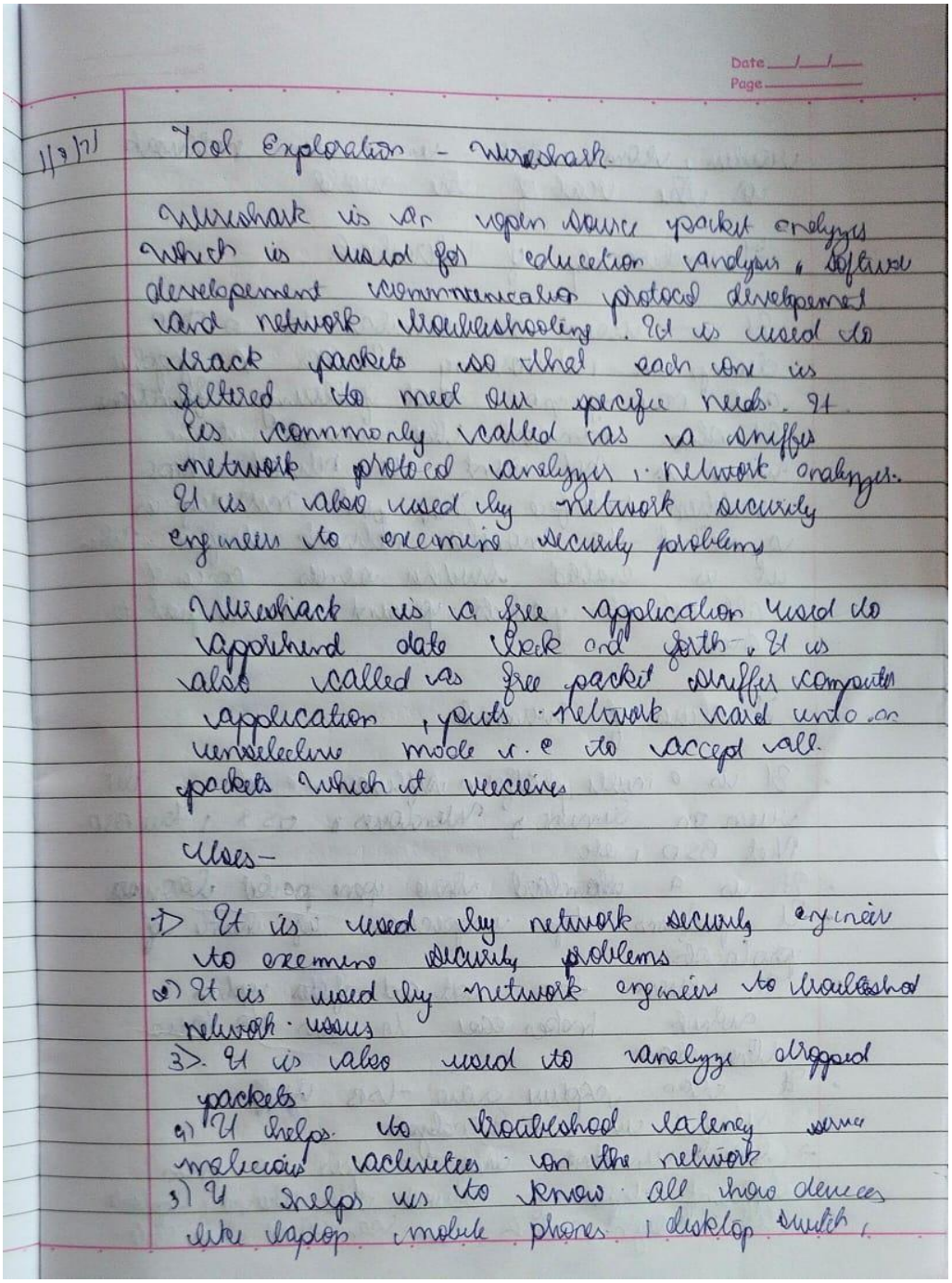
File Edit Shell Debug Options Window Help

```
Python 3.9.13 (tags/v3.9.13:6de2ca5, May 17 2022, 16:36:42) [MSC v.1929 64 b
it (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/sanja/OneDrive/Documents/ServerUDP.py =======
====
The server is ready to receive
nSent contents of ServerUDP.py
```

Ln: 7 Co

EXPERIMENT-17

Q) Tool Exploration -Wireshark



Date __/__/__
Page

1/9/21

Tool Exploration - Wireshark

Wireshark is an open source packet analyzer which is used for education analysis, software development communication protocol development and network troubleshooting. It is used to track packets so that each one is filtered to meet our specific needs. It is commonly called as a sniffer network protocol analyzer, network analyzer. It is also used by network security engineers to examine security problems

Wireshark is a free application used to apprehend data back and forth. It is also called as free packet sniffer computer application, puts network card into an unselective mode i.e to accept all packets which it receives

Uses -

1) It is used by network security engineer to examine security problems
2) It is used by network engineers to troubleshoot network issues
3) It is also used to analyze dropped packets
4) It helps to troubleshoot latency issues malicious activities on the network
5) It helps us to know all those devices like laptop, mobile phones, desktop switch,

readers, communicate on a lan network or the rest of the world

Functionality of Wireshark

It is used as similar to a TCP dump on networking. It has a graphical and non-graphical filtering functions. It also monitors the unicast traffic which is not sent to network's hyper-sadelen interface. The port mirroring is a method to monitor network traffic. When it is enabled switches sends copies of all network packets present on one port to another.

Features of wireshark.

- It is a multi platform software i.e it can run on Linux, Windows, OS X, Free BSD, Net BSD, etc
- It is a standard three pane packet browser.
- It performs deep inspection of hunt of protocols
- It even has sort and filter option which helps user to view also know the data
- It can capture raw USB traffic.
  - Useful in IP analysis
  - Also involves live analysis i.e from different types of network like ethernet, Bluetooth through which raw can read live data