VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

COMPUTER NETWORKS

Submitted by

ADNAN ANWAR (1BM21CS008)

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 **ADNAN ANWAR(1BM21CS008),** 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.

Dr. Nandini Vineeth

Assistant Professor
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak

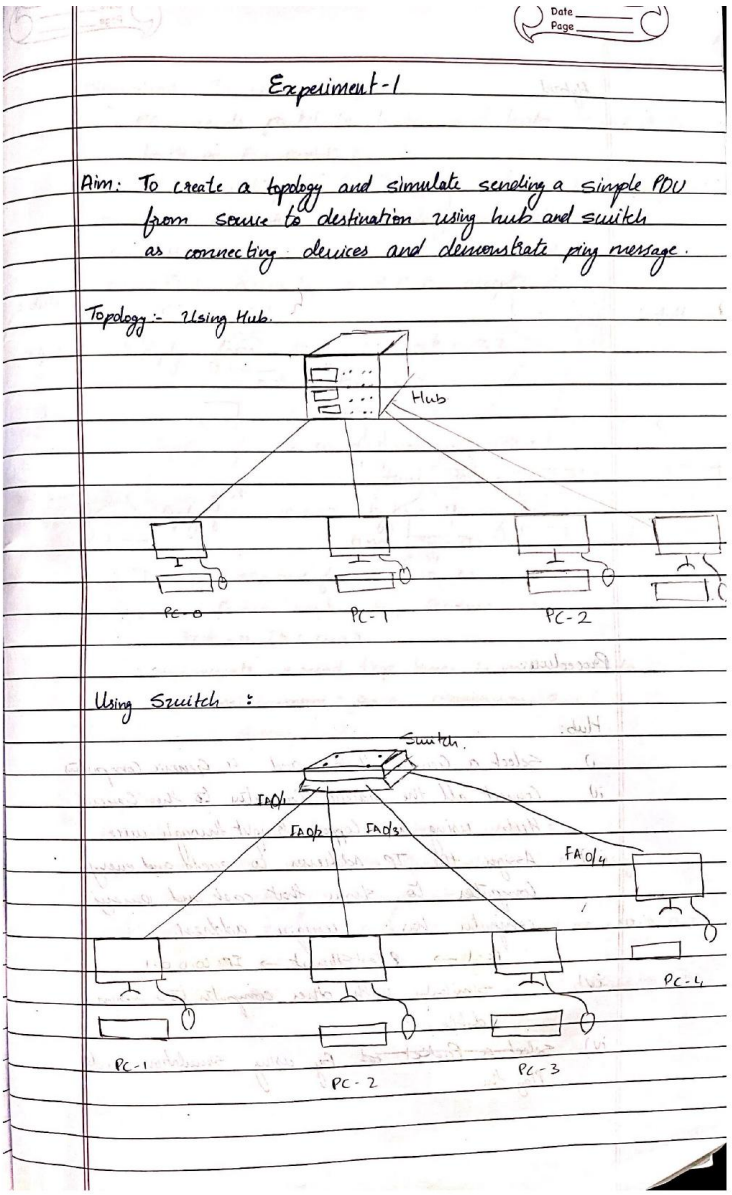
Professor and Head
Department of CSE
BMSCE, Bengaluru

`

Index

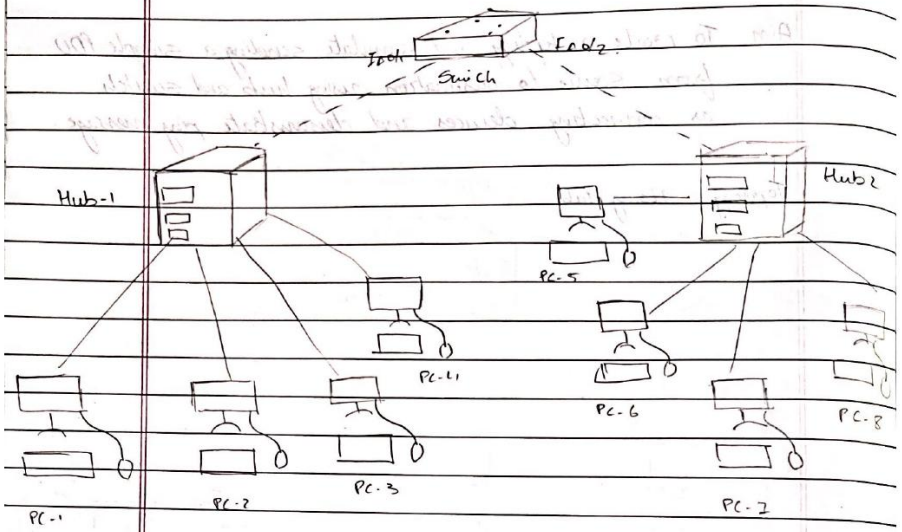
EXPERIMENT-1

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.



Scanned with CamScanner

Hybrid.



Procedure:

Hub:

i) Select a Generic Hub and 4 Generic Computers
ii) Connect all the Generic Computers to the Generic Hub using a Copper-Straight through wire.
iii) Assign the IP-Addresses to each and every Computer to show that each and every computer has a unique address.
PC-1 → IP FastEthernet → IP: 10.0.0.1
iv) Similarly with other computer too assign different
iv) ~~Select a Packet~~ By using simulation mode Play the

Observation: In Simulation mode:

- PC sends packet to hub and hub send it to both PC1 and PC2.
- PC1 dispatch the message whereas PC2 accepts it.
- PC2 send the acknowledgement packet to hub.
- hub again sends it to PC0 and PC1
- PC1 discards and PC0 accepts it.

Output:

Reply form 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 = 3ms TTL = 128

Ping statistics for 10.0.0.2.
Packet = sent = 4, Recieved = 4,
Lost = 0 (0% loss)
Approximate round trip times in milli-seconds
minimum = 0ms, Maximum = 3ms
Average = 2ms

Switch:

Procedure: i) Select a switch and 3PC's
ii) Connect the switch to the conditioned PC's using a copper straight through cut
iii) Assign the I.P address to the PC's - 10.0.0.4, 10.0.0.5, 10.0.0.5, 10.0.0.6 respectively.
iv) Select the PC's and the source and distribution to PC

Date
Page

Observation in simulation:
- PC3 sends the packet to switch and it sends to the both PC4 and PC5 in first round.
- PC4 reject and PC 5 accepts and sends acknowledge packet to both PC3 and PC5
- PC 4 discards it PC3 accepts it
- Now because of switch PC5 sends packet only to PC5

output: Reply from 10.0.0.5 bytes = 32 time=0ms TTL=128
Reply from 10.0.0.5 bytes = 32 time=0ms TTL=128
Reply from 10.0.0.05 bytes = 32 time=0ms TTL=128
Ping statistis from 10.0.0.5
Packets set 4, recieved=4 lost=0 (0% lost)
Approximate round trips in milli seconds
minimum = 0ms Maximum = 3ms, average = 22ms

Hybrid
Procedure:
- 11 PC's 2 generic Hubs and 1 switch were placed on the workspace
- 4PC's were connected to hub 0 via copper straight through cables. Remaining 7 pc's were connected to hub 1 via copper straight cable.
- All pc's were assigned address IP(10.0.0.1 to 10.0.0.11)
- Two hubs were connected to the switch via copper cross over cables which are used to connect devices on the same level.

Observation.

Hub: A hub receives data and broadcast it to all device connected to it.

Switch: A switch prevents traffic between 2 device from being shared with other devices connected to it.

Result:

PC% command line

PC> ping 10-0-0-8

Ping from 10.0.0.8 byte = 32 time = 3ms TTL=128

Ping from 10.0.0.8 byte = 32 time = 18ms TTL=128

Ping from 10.0.0.8 byte = 32 time = 10ms TTL=128

Ping statistics 10.0.0.8

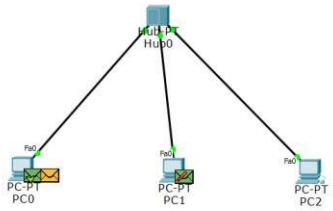
Packets sent = 4, received = 4, loss = 0 (0% loss).

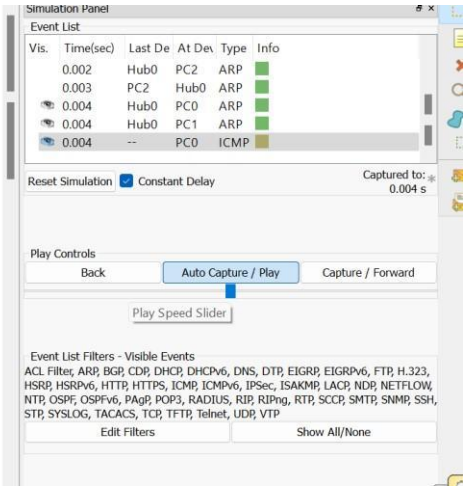
Approximate round trip time in millie seconds

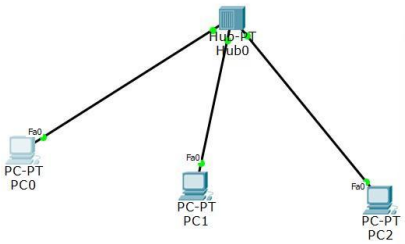
maximum = 8ms Average = 2ms Minimum = 0ms/s

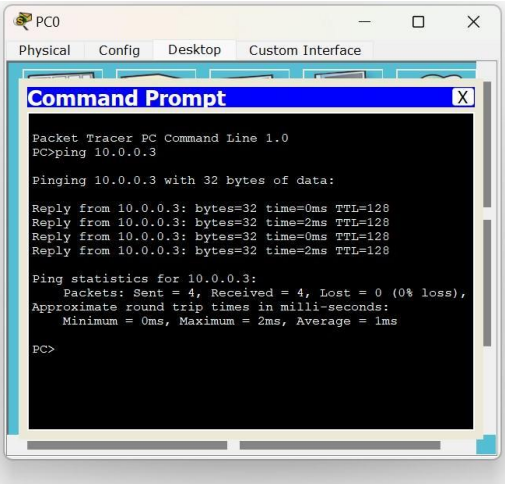
TOPOLOGY & OUTPUT

1.Hub and PCs

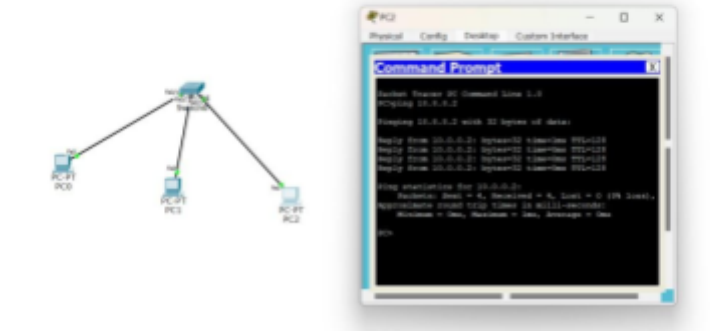




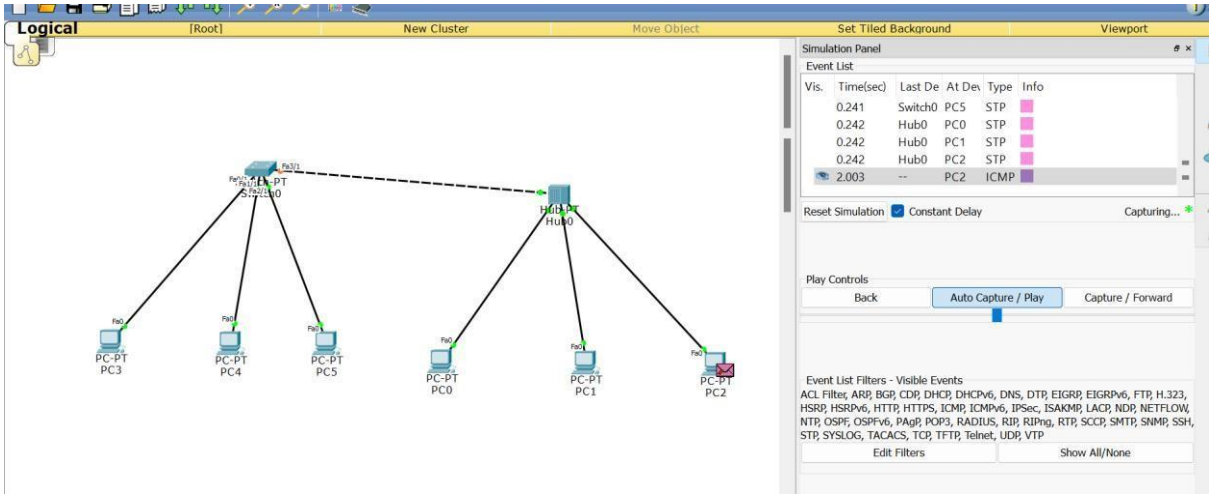


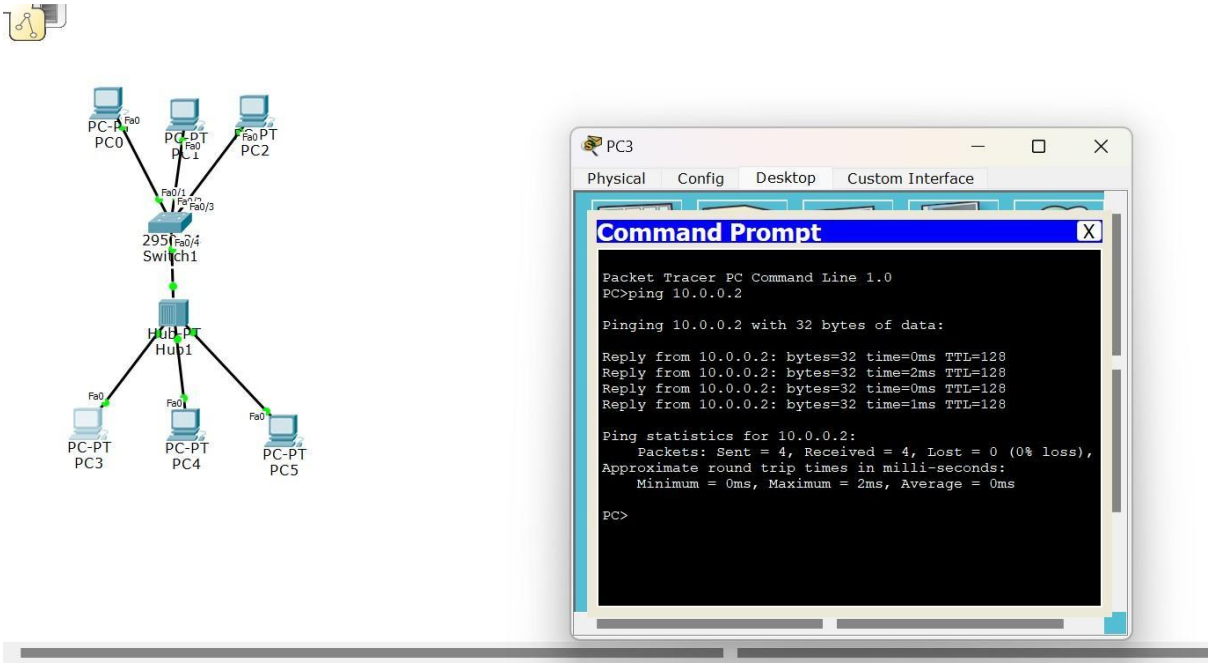


2.Switch and PCs



3.Hub,Switch and PCs

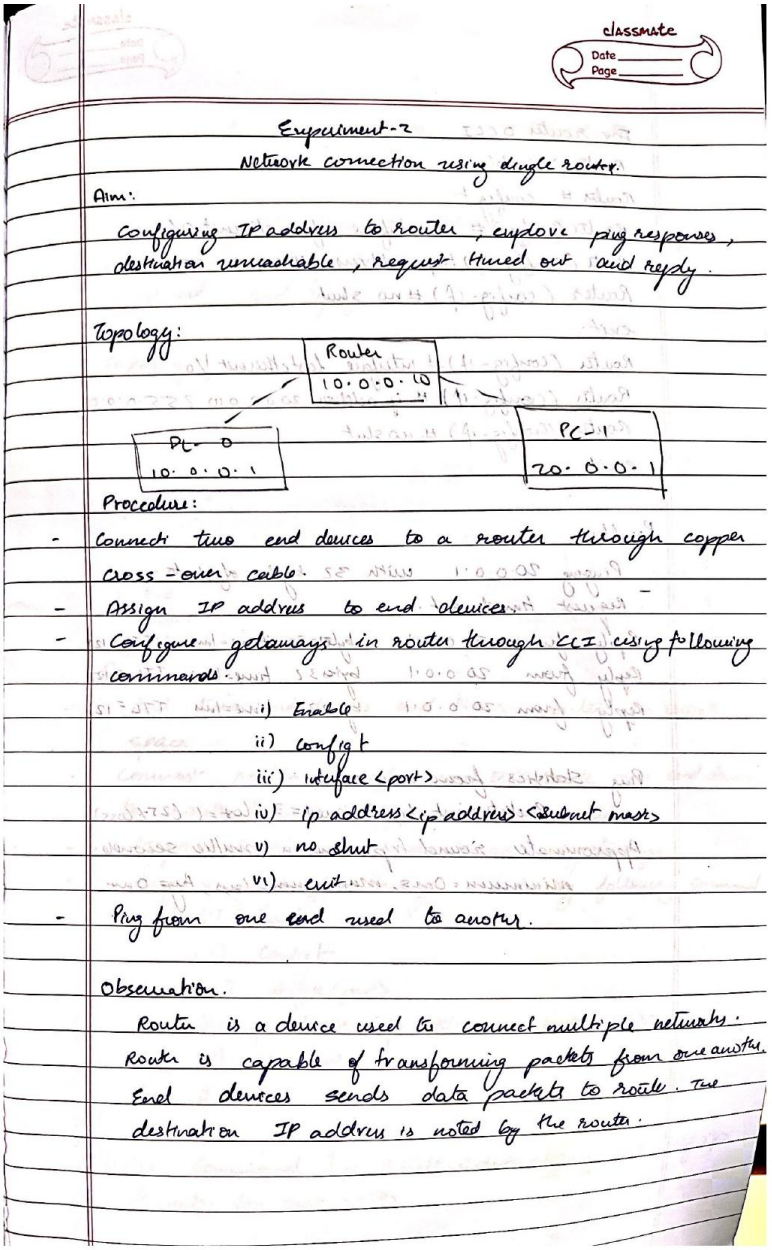



PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
Fa0
Fa0/1
Fa0/2
Fa0/3
Fa0/4
2950-24
Switch1
Hub-PT
Hub1
PC-PT
PC3
PC-PT
PC4
PC-PT
PC5
PC3
Physical
Config
Desktop
Custom Interface
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.2
Pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2: bytes=32 time=0ms 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



Experiment-2

Network connection using single router.

Aim:

Configuring IP address to router, explore ping responses, destination unreachable, request timed out and reply.

Topology:

Procedure:

- Connect two end devices to a router through copper cross-over cable.
- Assign IP address to end devices.
- Configure gateways in router through CLI using following commands.
  - i) Enable
  - ii) config t
  - iii) interface <port>
  - iv) ip address <ip address> <subnet mask>
  - v) no shut
  - vi) exit
- Ping from one end used to another.

Observation.

Router is a device used to connect multiple networks. Router is capable of transforming packets from one another. End devices sends data packets to router. The destination IP address is noted by the router.

For router CLI

Router > enable

Router # config t

Router (config) # interface fast ethernet 0/0

Router (config-if) # ip-address 10.0.0.10 255.0.0.0

Router (config-if) # no shut

exit

Router (config-if) # interface fastethernet 1/0

Router (config-if) # ip address 20.0.0.10 255.0.0.0

Router (config-if) # no shut

exit

Result:

Pinging 20.0.0.1 with 32 bytes of data:

Request timed out.

Reply from 20.0.0.1 bytes=32 time=1ms TTL=127

Reply from 20.0.0.1 bytes=32 time=1ms TTL=127

Reply from 20.0.0.1 bytes=32 time=1ms TTL=127

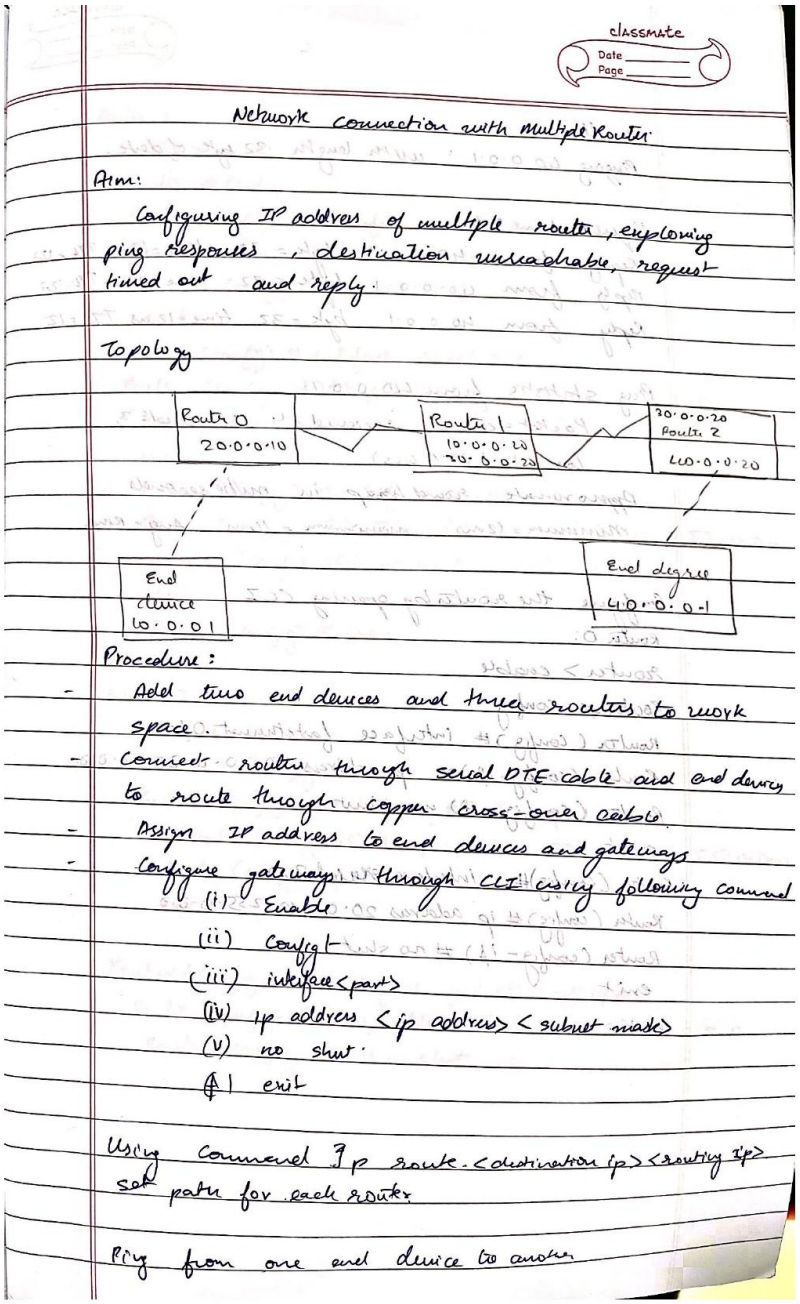
Ping statistics from 20.0.0.1

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

Approximate round trip times in millie seconds.

Minimum = 0ms, maximum = 1ms, Avg = 0ms

PROGRAM 2.2



Network connection with Multiple Router

Aim:

Configuring IP address of multiple router, exploring ping responses, destination unreachable, request timed out and reply.

Topology

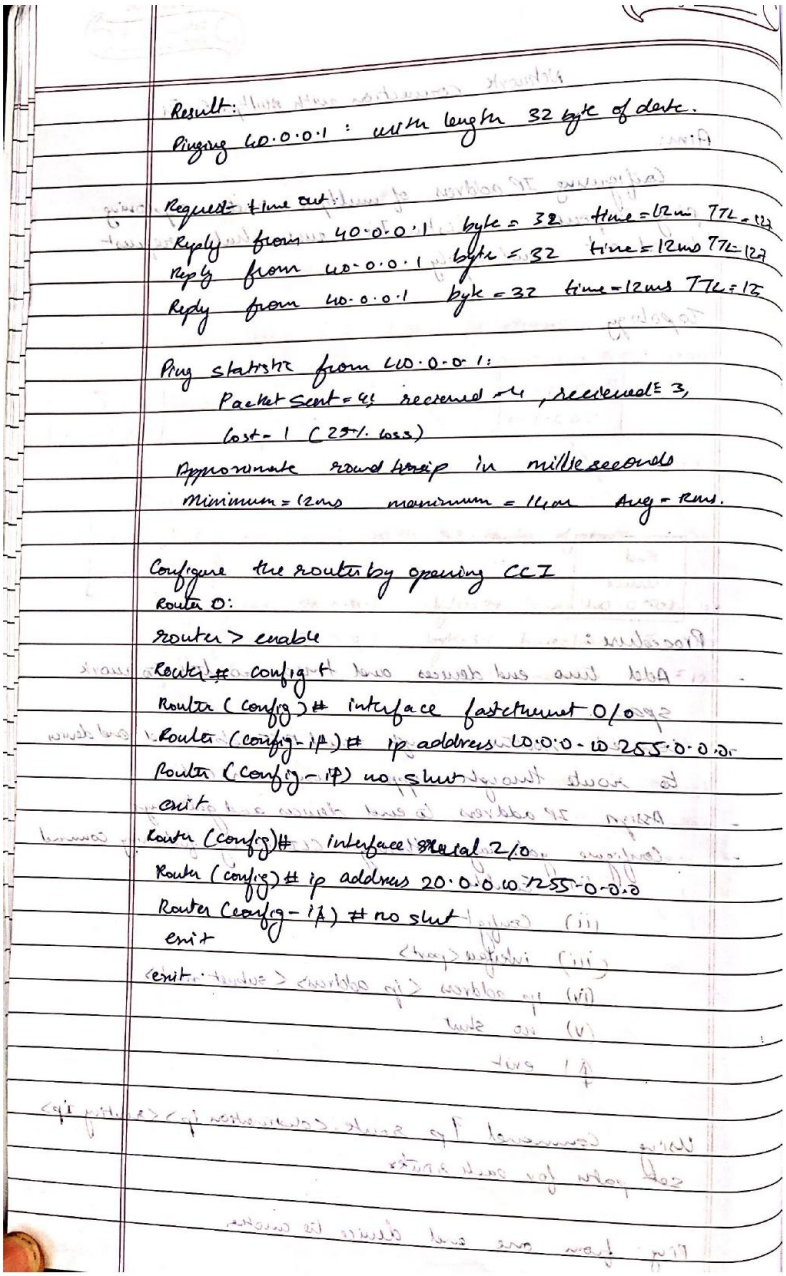
Procedure :

- Add two end devices and three routers to work space.
- Connect routers through serial DTE cable and end devices to route through copper cross-over cable.
- Assign IP address to end devices and gateways
- Configure gateways through CLI using following command
  - (i) Enable
  - (ii) Config t
  - (iii) interface <port>
  - (iv) ip address <ip address> <subnet mask>
  - (v) no shut.
  - exit

Using Command ip route <destination ip> <routing ip> set path for each router

Ping from one end device to another

Scanned with CamScanner



Result:
Pinging 40.0.0.1 : with length 32 byte of data.

Request time out
Reply from 40.0.0.1 byte = 32 time = 12ms TTL = 127
Reply from 40.0.0.1 byte = 32 time = 12ms TTL = 127
Reply from 40.0.0.1 byte = 32 time = 12ms TTL = 127

Ping statistic from 40.0.0.1:
Packet Sent = 4, received = 4, received = 3,
lost = 1 (25% loss)
Approximate round trip in millie seconds
Minimum = 12ms maximum = 14ms Avg = 12ms.

Configure the router by opening CLI
Router 0:
router > enable
Router # config t
Router (config) # interface fastethernet 0/0
Router (config-if) # ip address 40.0.0.10 255.0.0.0
Router (config-if) no shut
exit
Router (config) # interface serial 2/0
Router (config) # ip address 20.0.0.10 255.0.0.0
Router (config-if) # no shut
exit
exit

Router 1:

Router > enable

Router # config t

Router (config) # interface serial 2/0

Router (config-if) # ip address 20.0.0.20 255.0.0.0

Router (config-if) # no shut.

exit

Router (config) # interface serial 3/0

Router (config-if) # ip address 30.0.0.20 255.0.0.0

Router (config-if) # no shut.

exit

Router (config) # interface serial 3/0

Router (config-if) # ip address 30.0.0.20 255.0.0.0

Router (config-if) no shut

exit

Router (config) # exit.

Router 2:

Router > enable.

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

exit

Router (config-if) # interface fast ethernet 0/0

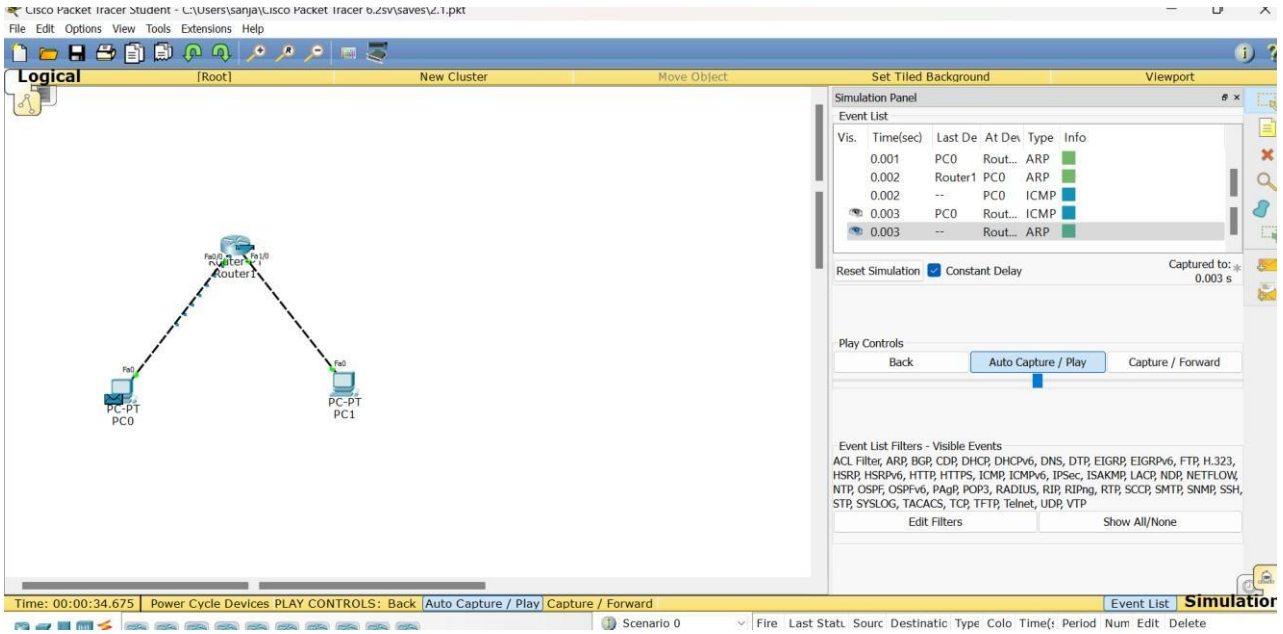
Router (config-if) # ip address 40.0.0.10 255.0.0.0

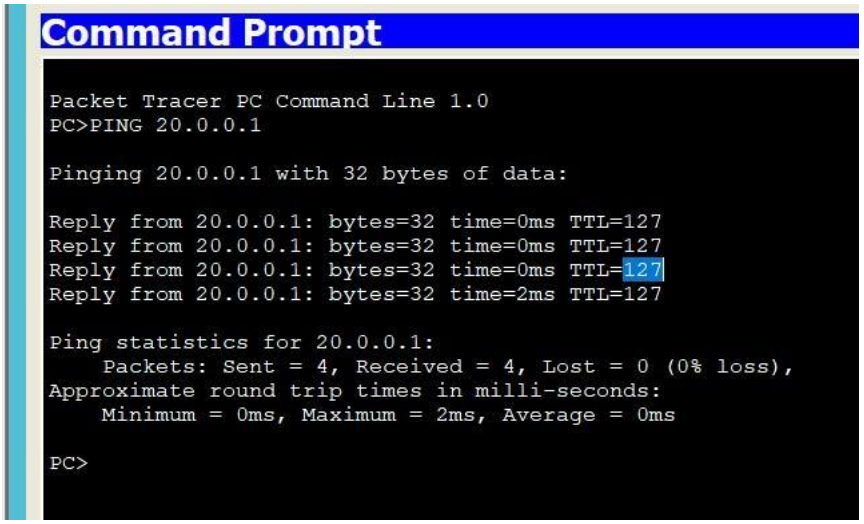
Router (config-if) # no shut

exit.

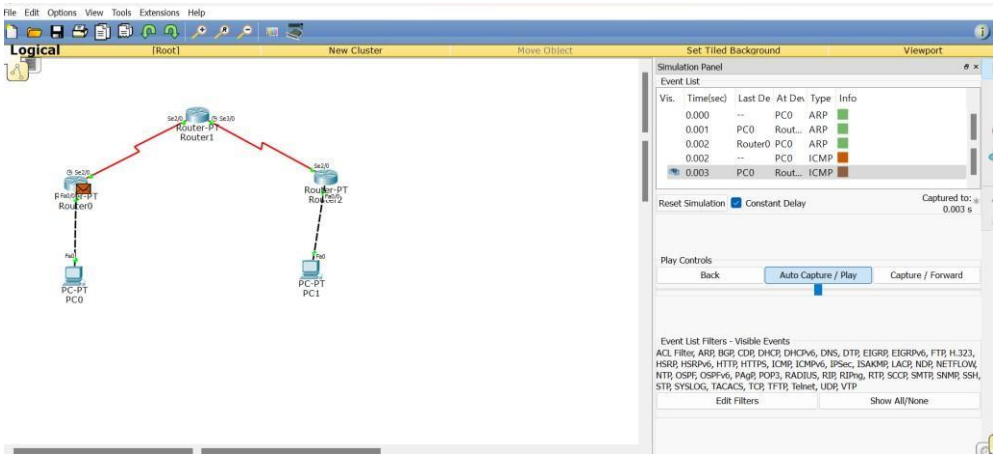
TOPOLOGY &OUTPUT

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

classmate
Date
Page

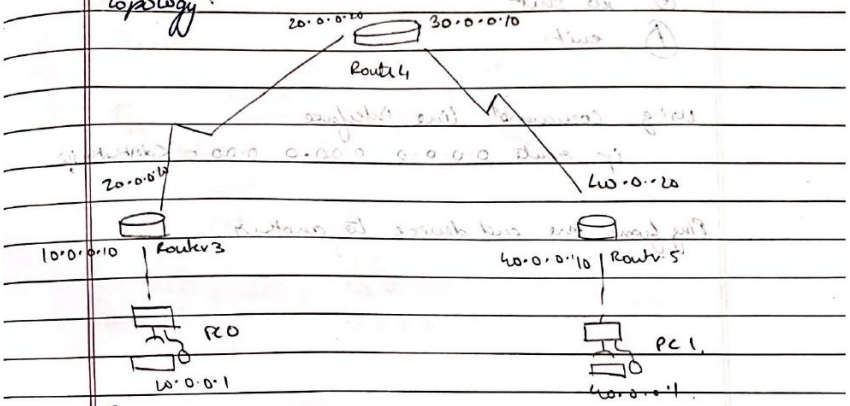
Experiment-3

Configuration of default route.

Aim:

Configure default route and static route to the route

Topology:



Procedure:

Add two end devices and three routers to workspace.

Connect routers through serial DTE cable and end devices through copper cross over cable Assign IP address to the end devices.

End device 1 : 10.0.0.1

End device 2 : 40.0.0.1

Observation:

→ Routers are seen to connect two different networks

→ If router has only one path to transverse. Default routing to send packet of any destination.

→ End devices sends the packet to the router which then redirect it to the appropriate destination.

Configuration through CLI command

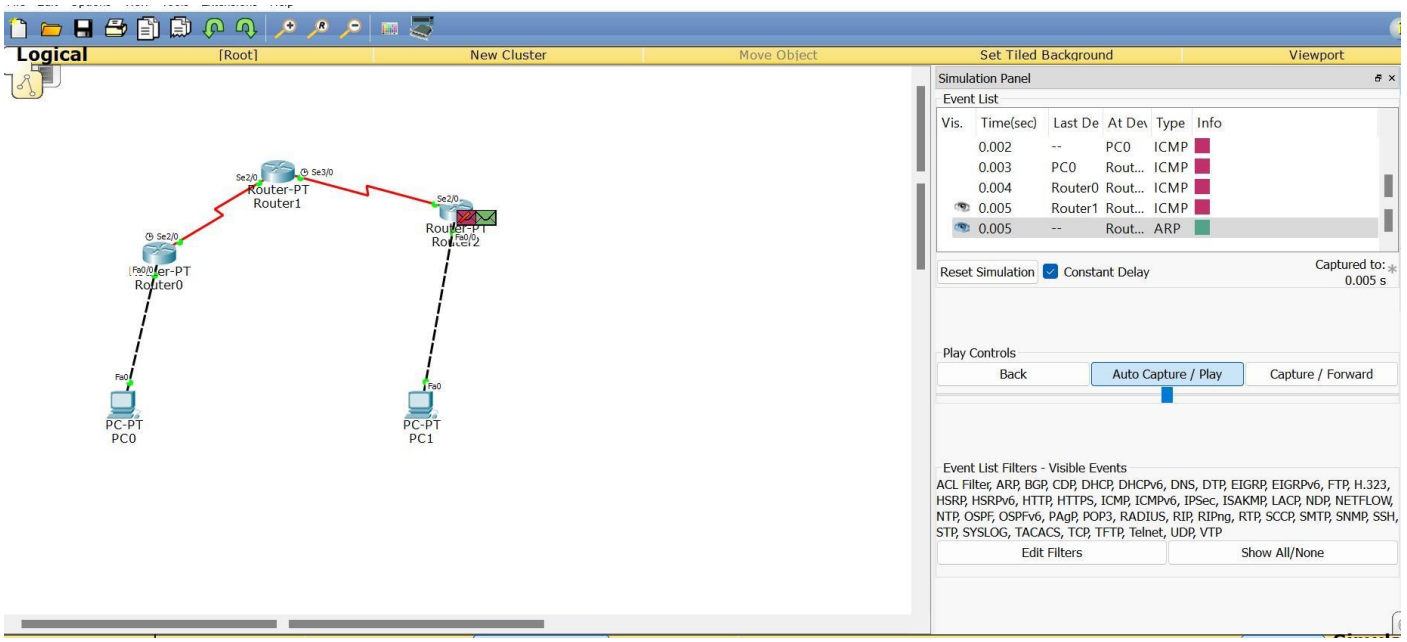
(a) enable.
(b) configt
(c) Interface <port>
(d) ip address <ip address> <subnet mask>
(e) no shut
(f) exit.

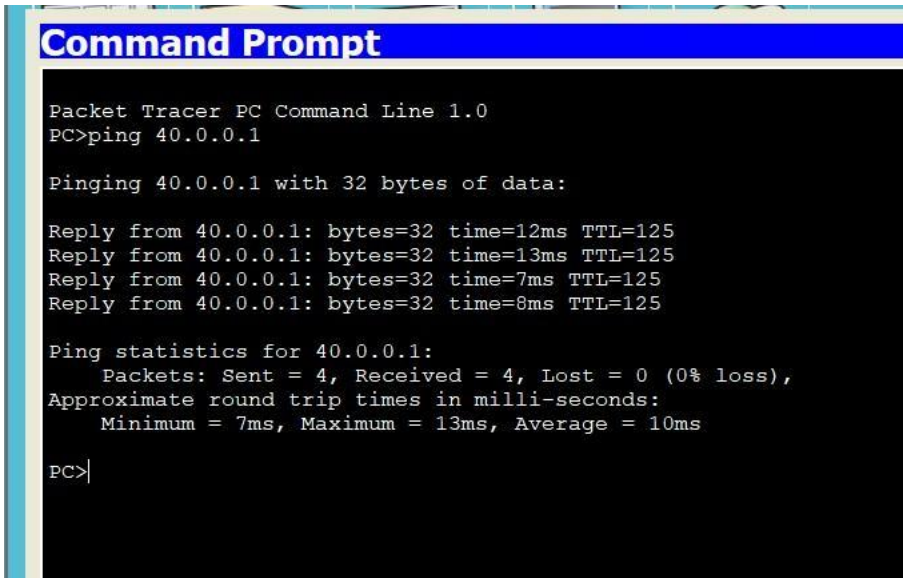
using command line interface

ip route 0.0.0.0 0.0.0.0 0.0.0.0 <destination ip>

Ping from one end device to another.

TOPOLOGY & OUTPUT

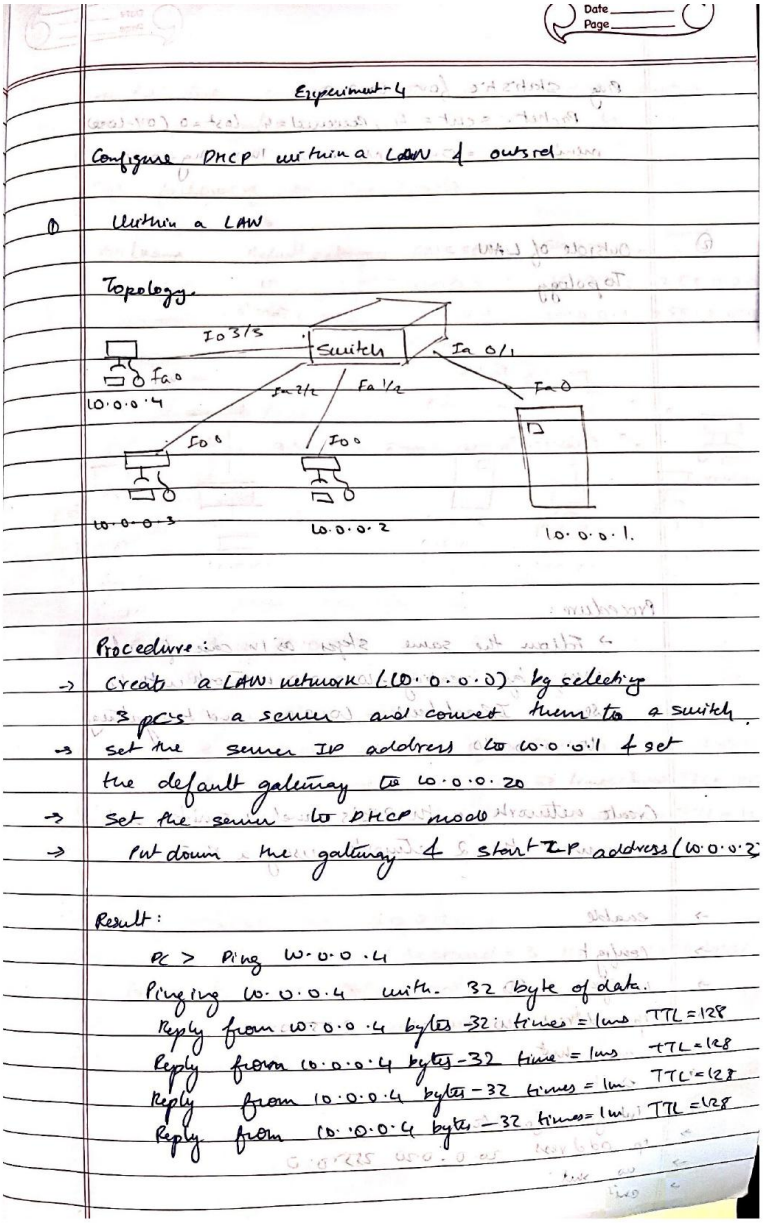




EXPERIMENT-4

Q) Configure DHCP within a LAN and outside LAN

PROGRAM 4.1



Experiment-4

Configure DHCP within a LAN & outside

1) Within a LAN

Topology.

Procedure:

→ Create a LAN network (10.0.0.0) by selecting 3 PC's a server and connect them to a switch.

→ set the server IP address to 10.0.0.1 & set the default gateway to 10.0.0.20

→ Set the server to DHCP mode

→ Put down the gateway & start IP address (10.0.0.2

Result:

PC > Ping 10.0.0.4

Pinging 10.0.0.4 with 32 byte of data.

Reply from 10.0.0.4 bytes=32 times = 1ms TTL=128

Reply from 10.0.0.4 bytes=32 time = 1ms TTL=128

Reply from 10.0.0.4 bytes=32 times = 1ms TTL=128

Reply from 10.0.0.4 bytes=32 times=1ms TTL=128

Scanned with CamScanner

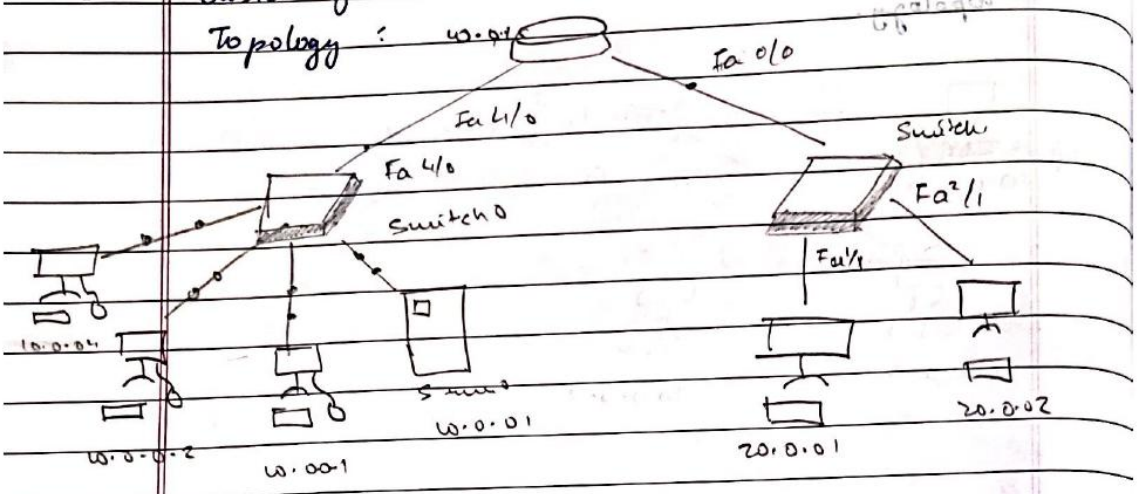
Ping statistic for 10.0.0.4

Packet. sent = 4, Received = 4, Lost = 0 (0% loss)

minimum = 0ms, Maximum = 1ms, Avg = 0ms

② Outside of LAN

Topology :



Procedure :

→ Follow the same steps as in case of Inside LAN, by creating 10.0.0.0 network with the server IP address - 10.0.0.1 and the gateway 10.0.0.10.

Create network with 2PC's and a switch and connect the 2 networks using a Router

→ enable

→ config t

→ interface 4/0

→ ip address 10.0.0.20 255.0.0.0

→ no shut

→ exit

→ interface fa 0/0

→ ip address 20.0.0.20 255.0.0.0.

→ no shut.

→ exit

PROGRAM 4.2

Go to the services of service 0 and set another DHCP pool gateway (default) to 20.0.0.10.

The following are the 2 pools

Pool Name	Default gateway	DNS server	Start IP address	Subnet mask
Server pool	10.0.0.10	0.0.0.0	10.0.0.2	255.0.0.0
Server pool	20.0.0.10	0.0.0.0	20.0.0.1	255.0.0.0

> Config t
> interface fa 0/0
> ip helper address < server ip-address >
> no shut
exit

Result

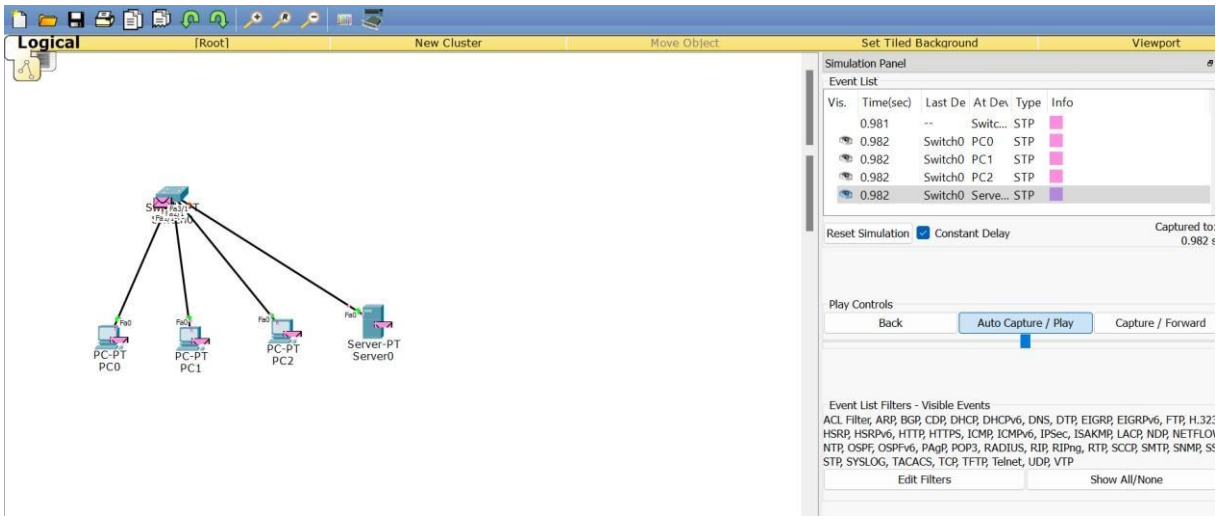
PC > Ping 20.0.0.2.

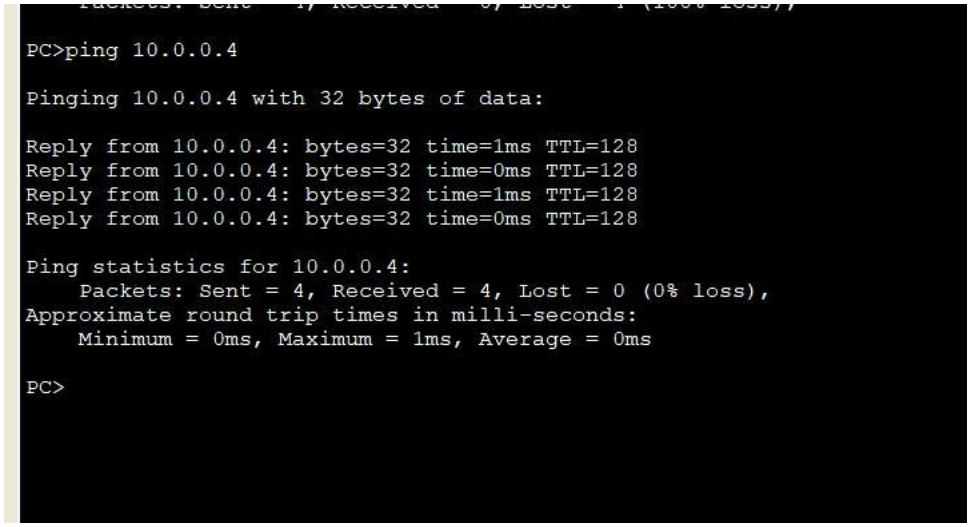
Pinging 20.0.0.2 with 32 byte of data
Request timed out
Reply from 20.0.0.2 : bytes = 32 time = 0 ms TTL = 127
Reply from 20.0.0.2 bytes = 32 time = 2 ms TTL = 127
Reply from 20.0.0.2 byte = 32 time = 1 ms TTL = 127

Ping statistics for 20.0.0.2.
Packets : sent = 4 received = 3 lost = 1 (25% loss)
Approx time (in milliseconds)
Minimum = 0 ms, maximum = 4 ms, Average = 1 ms

TOPOLOGY & OUTPUT

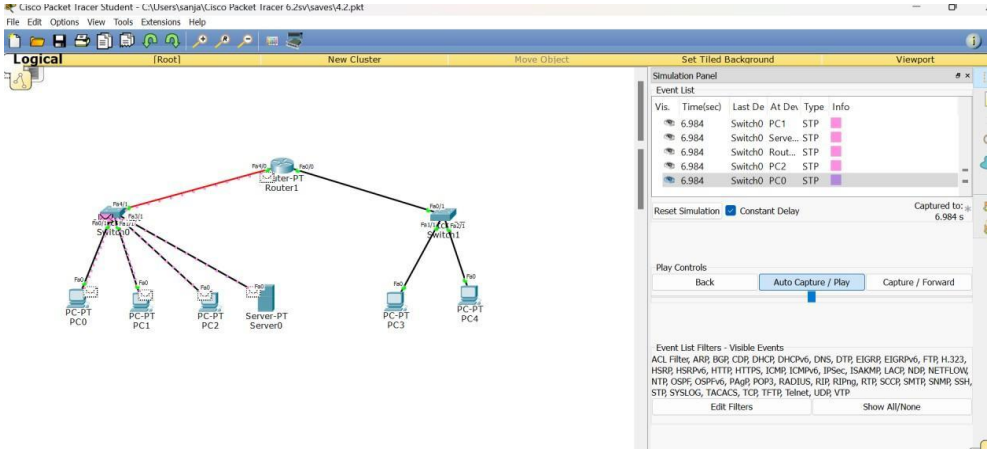
PROGRAM 4.1

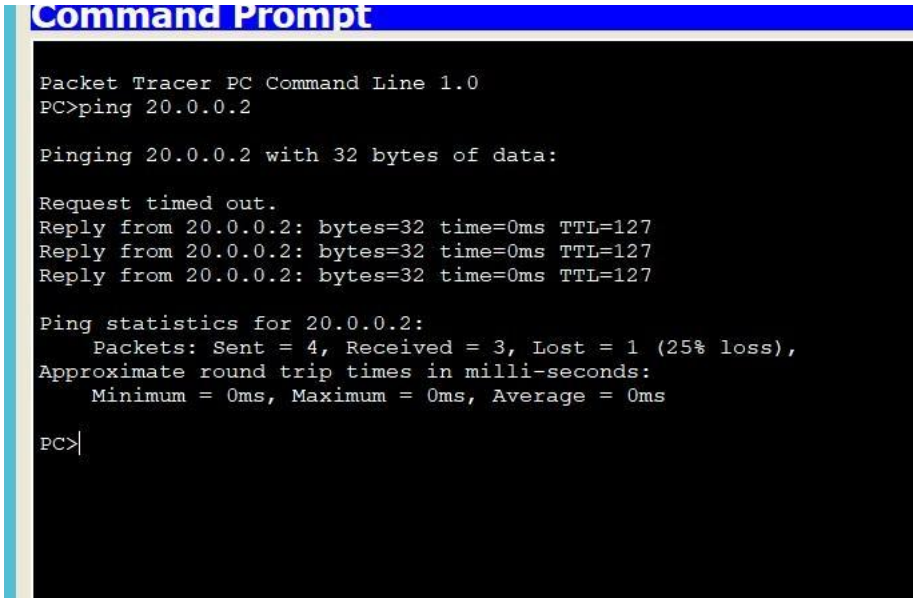




```
PC>ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:

Reply from 10.0.0.4: bytes=32 time=1ms TTL=128
Reply from 10.0.0.4: bytes=32 time=0ms TTL=128
Reply from 10.0.0.4: bytes=32 time=1ms 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 = 1ms, Average = 0ms

PC>
```

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>

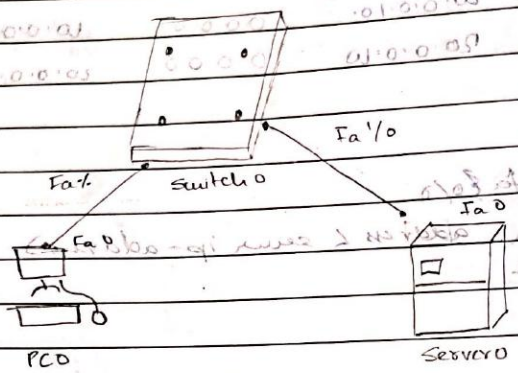
EXPERIMENT-5

Q) Configure Web Server, DNS within a LAN

Program - 5

Aim: Configure web server, DNS within a LAN.

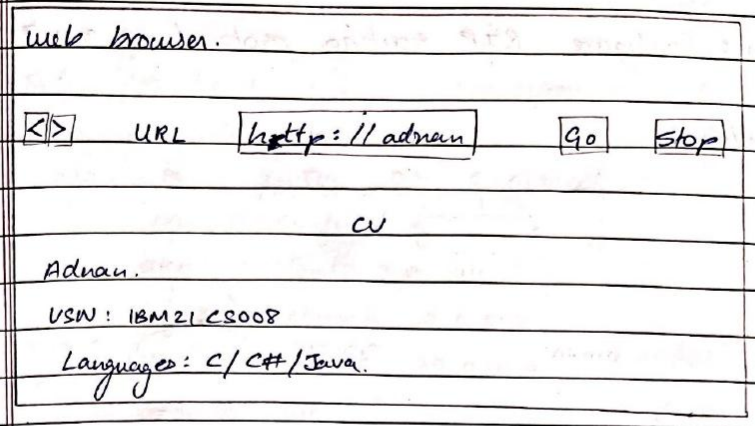
Topology:



Procedure.

- Connect a switch, PC, Server to form a LAN.
- Set PC's IP address by clicking and go to config tab. There in fast ethernet option, set IP address as 10.0.0.1. and subnet mask.
- Set server IP address as 10.0.0.2 and subnet mask.
- Go to PC's desktop and click on web browser. In URL type 10.0.0.2.
- To make CV, make changes in server service
- Go to server → services → HTTP → index.html.
- PC → desktop → web browser and type 10.0.0.2.
- Server → services → DNS and switch on the services.
- PC → desktop → web browser and type domain name

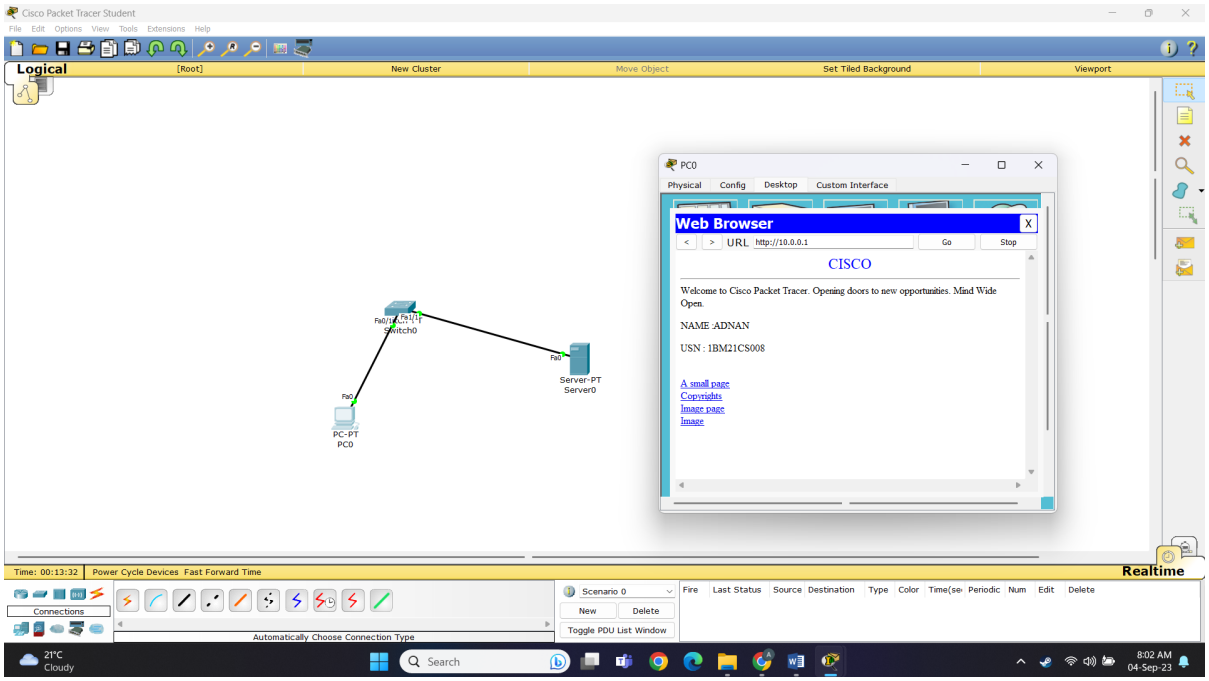
output:



Observation:

- When you want to go to a certain website, you open web browser & type domain name of that website or you can also type IP address.
- Since the IP address cannot be remembered, we resolve to find the domain and compute communi

TOPOLOGY & OUTPUT



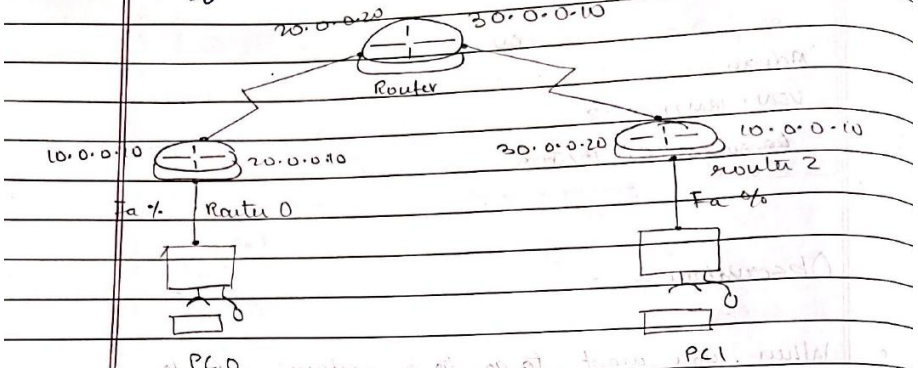
EXPERIMENT-6

Q) Configure RIP routing Protocol in Routers

Program - 6

Aim: Configure RIP routing protocol in router.

Topology:



Procedure

- Create a network using 3- routers & 2PCs
- Set IP address and gateway no for both
  - 10.0.0.1 – IP 10.0.0.10 - gate way - PC0
  - 40.0.0.1 – IP 40.0.0.10 - gateway - PC1
- Router → CLI mode.
  - Step 1- No
  - Step 2 - Enable
  - Step 3 - config T
  - Step 4 — interface fast ethernet %
  - Step 5 — IP address 10.0.0.10 255.0.0.0
  - Step 6 - No short
  - Step 7 - exit
  - step 8 - interface se %
  - Step 9 - IP address 20.0.0.10 255.0.0.0
  - Step 10 - encapsulation ppp
  - Step 11 - clock rate 64
  - Step 12 — No shut.

Date
Page

- here for router with fast ethernet execute till step 9. and type no shut.
- Only for router to router connection execute all steps also execute step 11.
- again to Router 0 → CLI mode :

  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 step for all routers
- Now go to each router and type show IP route
- Go to PC0 and Ping a msg to PC1 using ping destination IP address command.

Ping output:

Packet tracer PC command line 1.0

PC > Ping 40.0.0.1

Pinging 40.0.0.1 with 32 byte data.

Request time out

Reply from 40.0.0.1 : byte = 32 time = 8ms TTL=125
Reply from 40.0.0.1 : bytes = 32 time = 5ms TTL = 125
Reply from 40.0.0.1 : bytes = 32 time=10ms TTL= 125.

Statistics

Packets sent = 4 ; recieved = 3, lost = 1 (25% loss)

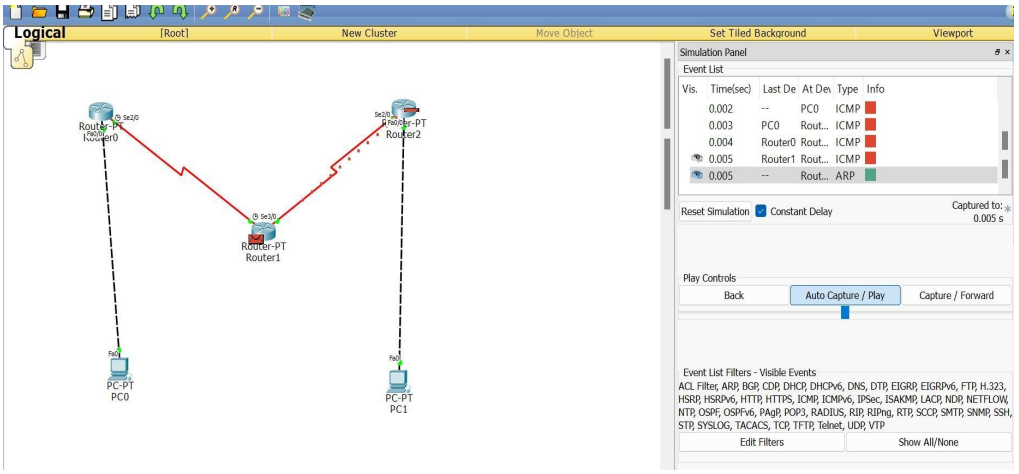
Approx round trip in ms:

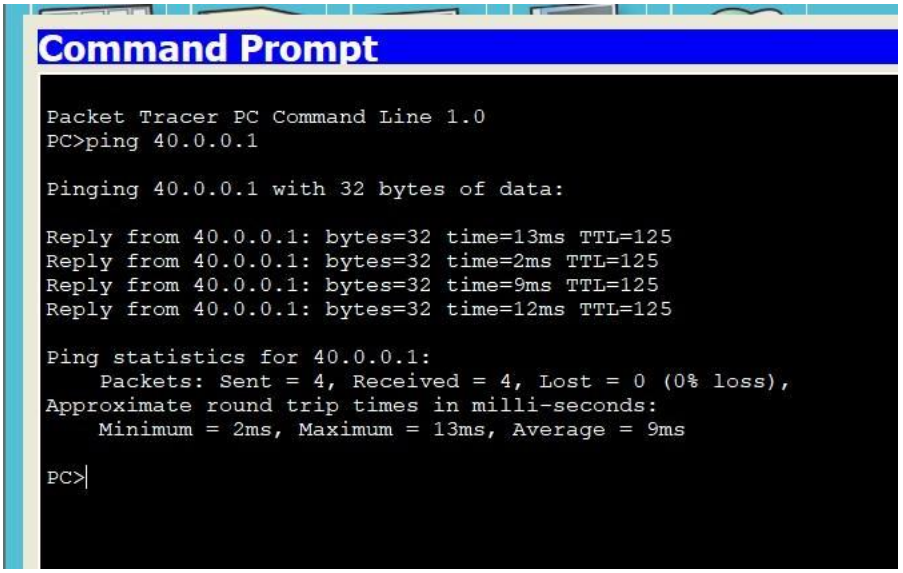
min = 5ms , max = 10ms, Avg = 7ms.

Observation

- RIP is used to hop count as a routing matric to find bus path. It is a distance - vector routing
- updates of the network are exchanged periodically
- updates of routing information are broadcasted
- full routing information are broadcasted in updates
- router always trust routing information received from neighbour routers

TOPOLOGY & OUTPUT





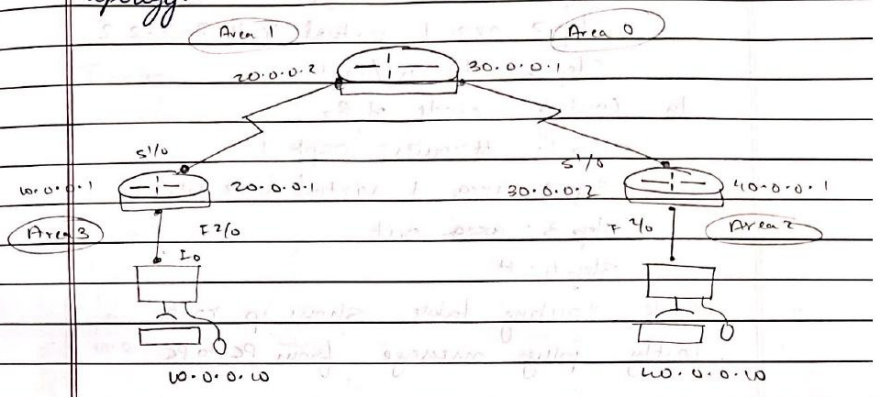
EXPERIMENT-7

Q) Configure OSPF routing protocol

Program-7

Aim - configure OSPF routing protocol.

Topology:



Procedure :

- Create topology using 3 routers and 2PC's
- Configure PC's with IP address and gateway
- Configure each other routers are to IPaddress
- encapsulate ppp and clock rate should be set as in RIP config.
- Router → CLI → Config mode.
  - Step 1: Router os pf 1.
  - Step 2: router-id 1.1.1.1.
  - Step 3: network 10.0.0.0 0.255.255.255 area
  - Step 4: network 20.0.0.0 0.255.255.255 area 1.
  - Step 5: exit
- repeat these commands
- type show ip router
- to set loop backs
  - Step 1: (in config. ip mode)
  - Step 2: ip address 172.16.1.252
  - Step 3: No shut down.

- Repeat for other 2 router.
- create a virtual link b/w R_1, R_2

In config mode of R_1

Step 1 : router ospf 1

Step 2 : area 1 virtual link 2.2.2.2

Step 3 : # enter / exit.

In config mode of R_2

Step 1 : # router ospf 1

Step 2 : area 1 virtual link 1.1.1.1

Step 3 : area exit

Step 4 : #

- check routing table, show ip route
- lastly ping message from PC to PC.

Ping output :

Packet tracer PC command - line 1.0

PC > Ping 40.0.0.10

Pinging 40.0.0.10 with 32 byte 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 = 11ms TTL = 125

Reply from 40.0.0.10 bytes = 32 time = 8ms TTL = 125

Statistics :

packets sent = 4, received = 3, lost = 1

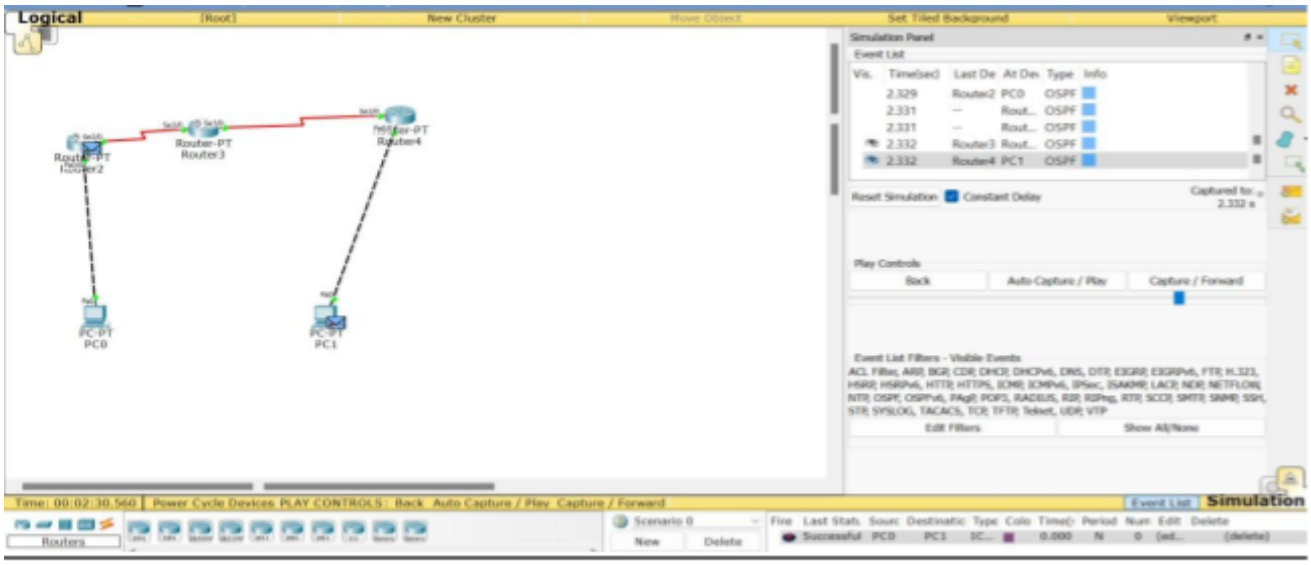
Approx round trip time

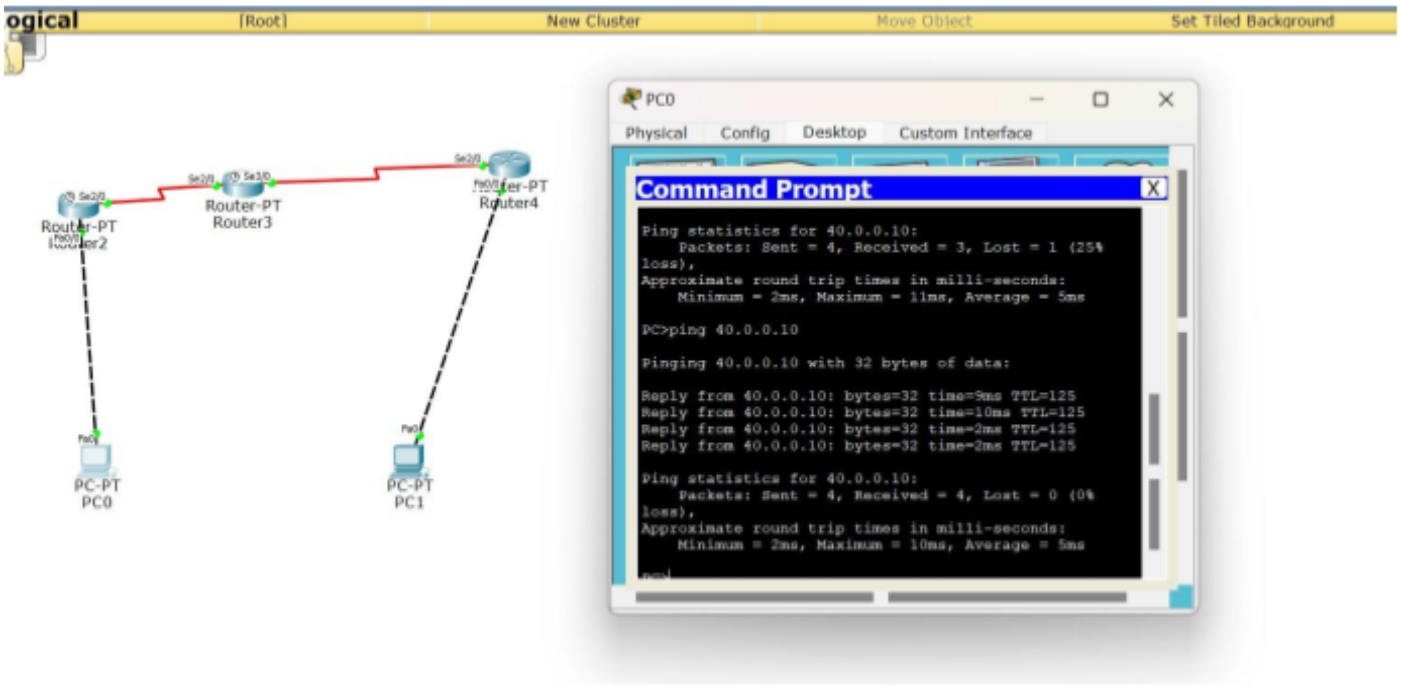
min = 8ms, max = 11ms, Avg = 10ms

Observation :

- OSPF is used to find the best path
- network is divided into 4 arrivals
- message is pinged after a virtual link.

TOPOLOGY & OUTPUT





EXPERIMENT-8

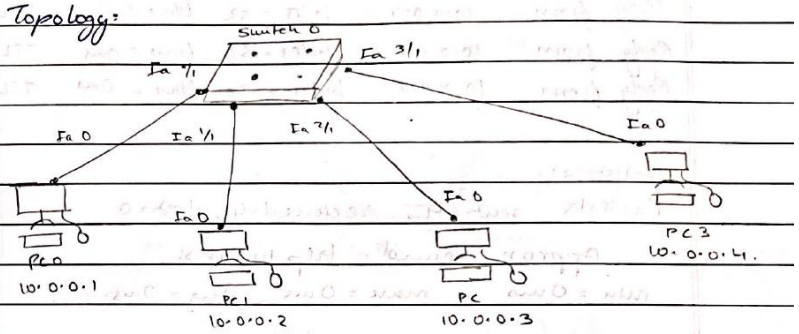
Q) To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)

classmate
Date
Page

Program - 8

Aim: To construct single LAN and understand the concept and operation of ARP (Address resolution protocol).

Topology:



Procedure:

- Create topology for 3 PC's and a server.
- assign IC address
- connect them through a switch
- use inspect tool to click on a PC to see ARP table
- Command in cmd for the same arp-a
- Initially ARP table is empty
- CLI of switch, the command- show mac address can be given on every transaction to see how the switch from transaction and build address table.
- Use the capture button in simulation panel to go step by step.

Ping Output:

PC > ping 10.0.0.4

pinging 10.0.0.4

Reply from 10.0.0.4 : bytes = 32 time = 0ms TTL = 12
Reply from 10.0.0.4 : bytes = 32 time = 0ms TTL = 12
Reply from 10.0.0.4 : bytes = 32 time = 0ms TTL = 12
Reply from 10.0.0.4 : bytes = 32 time = 0ms TTL = 12

Statistics:

Packets sent = 4, received = 4, lost = 0

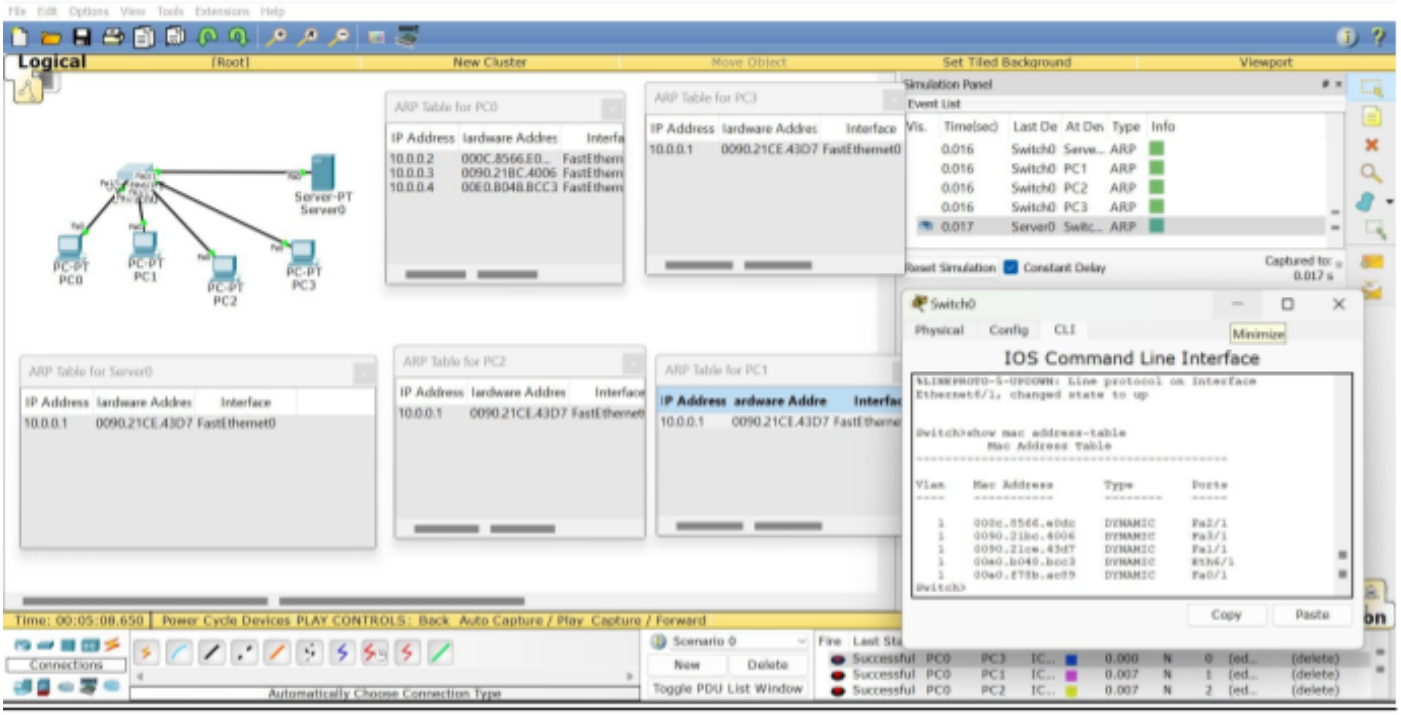
Approx round trip in ms:

min = 0ms , max = 0ms , Avg = 0ms.

Observation:

- when we ping b/w other 2 PC's, the address of PC's is known.
- Everytime a host request a mac address, a packet is sent to another host in LAN, which checks ARP cache to see IP address

TOPOLOGY & OUTPUT



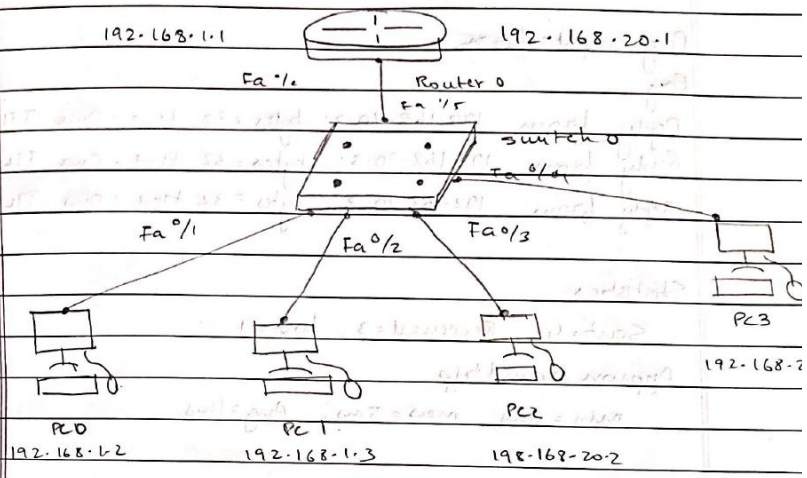
EXPERIMENT-9

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

Program - 9

Aim: To construct a VLAN and make a PC communication among VLAN

Topology:



Procedure:

- Create topology choose 1841 router 2960-24TT switch here
- Set the Ip address of router and 4PC's
- in Switch, go to config tab and select VLAN database. Give any VLAN no-like 2.
- Select interface fastethernet 4/1
- Select switch under 2nd interface.
- Router → Config tab → VLAN database → VLAN name.
- Go to router → CLI
  - Step 1: Config T
  - Step 2: interface.
  - Step 3: No shut

Scanned with CamScanner

Step 5: configT
Step 6: interface fa 0/0.1
Step 7: encapsulate
Step 8: ip address
Step 9: No shut
Step 10: exit

Ping output:

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

Statistics:

Sent=4, Received=3, lost=1

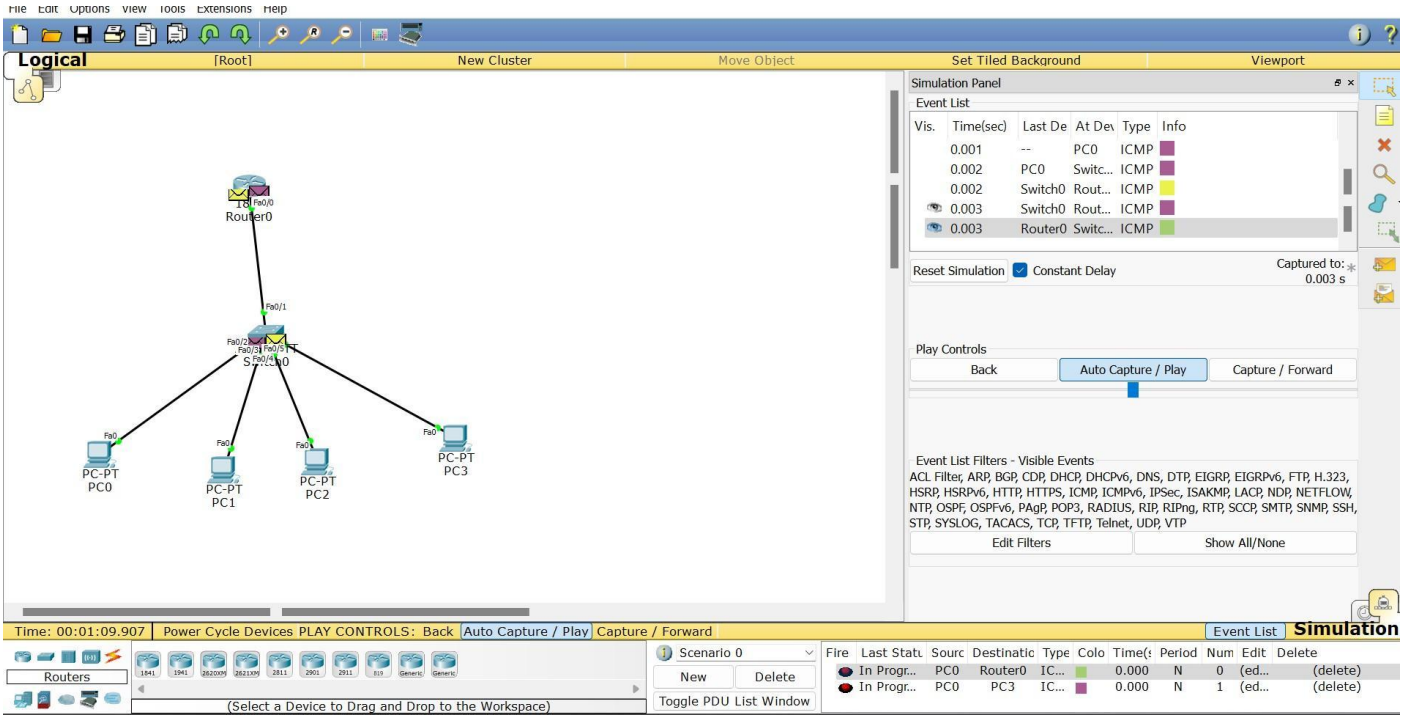
Approx round trip

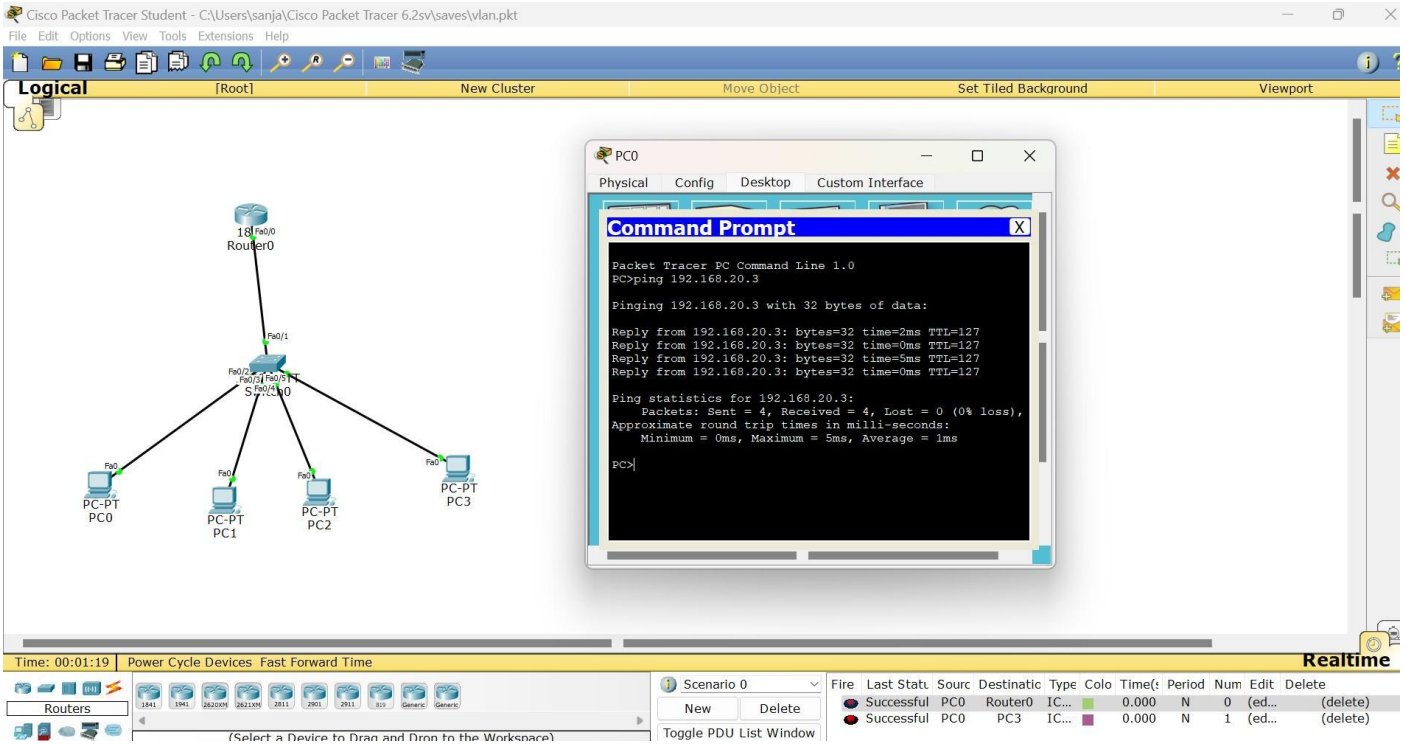
min=0ms, max=5ms, Avg=1ms.

Observation

VLAN & other VLAN connected to same switch.
VLAN's dont use IP address
Inter-VLAN routing gives as flexible tool to logically subdivide their networks.

TOPOLOGY & OUTPUT





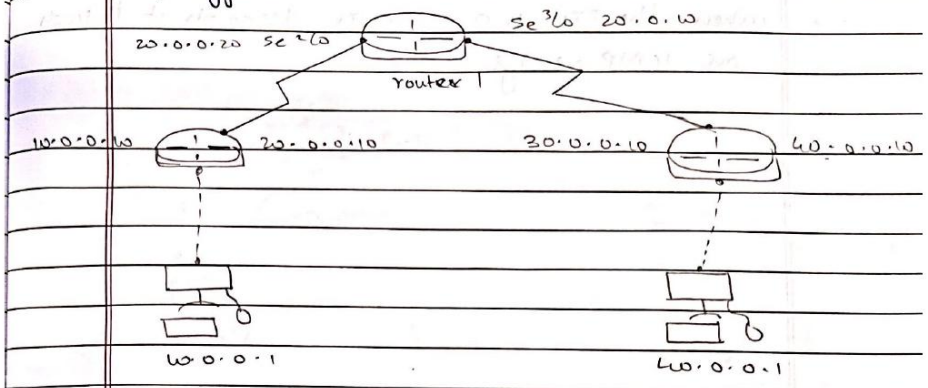
EXPERIMENT-10

Q) Demonstrate the TTL/ Life of a Packet

Program 10.

Aim: Demonstrate the TTL / life of a packet

Topology:

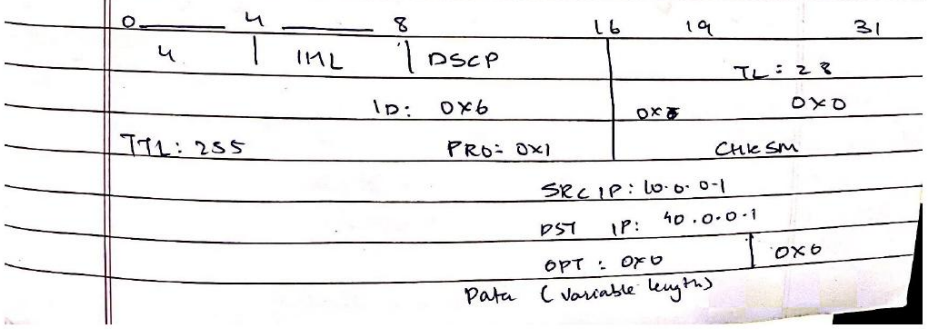


Procedure.

- Create a topology
- Set the IP address and gateway for both PC's
- Configure the routers either as static / dynamics
- In simulation mode send a simple PDU from one PC to another
- Use capture button
- Click on the PDU during every transfer.

Output:

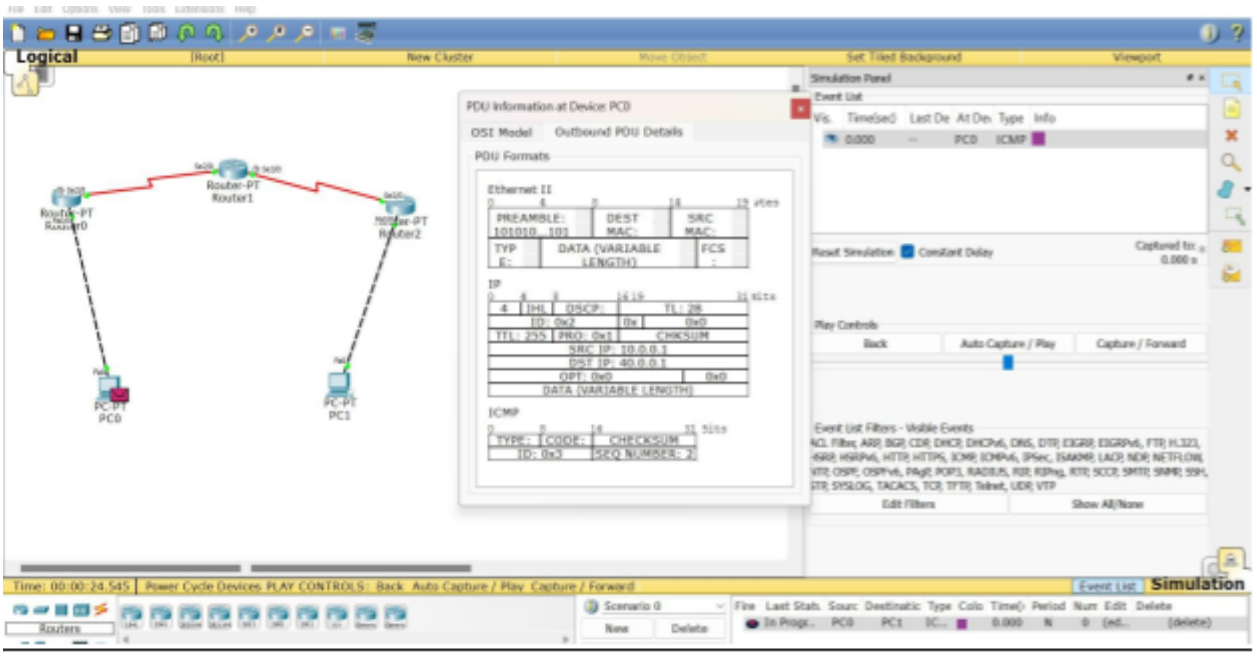
IP

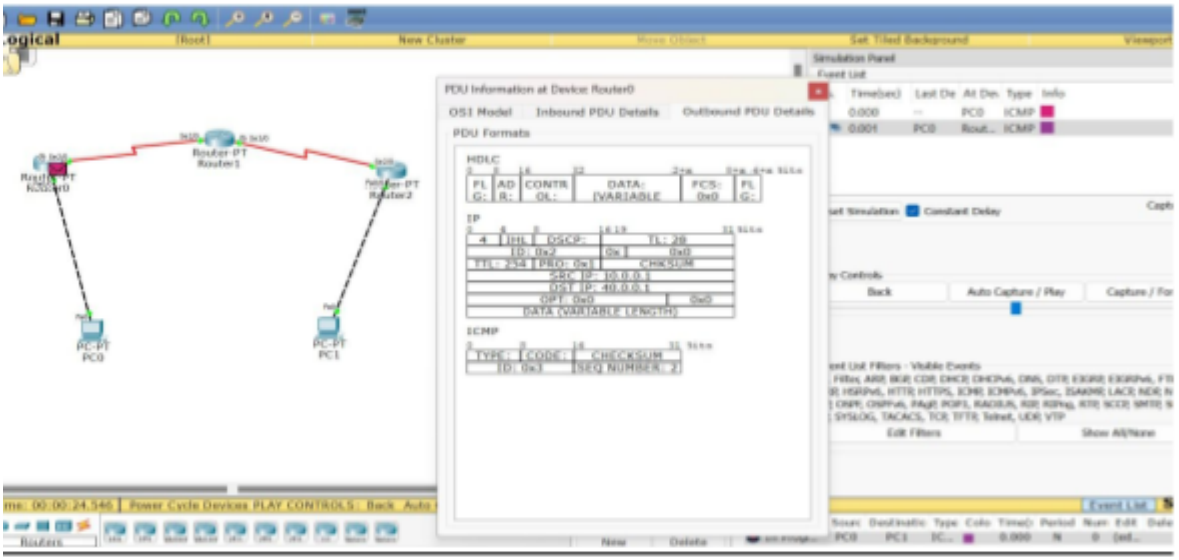


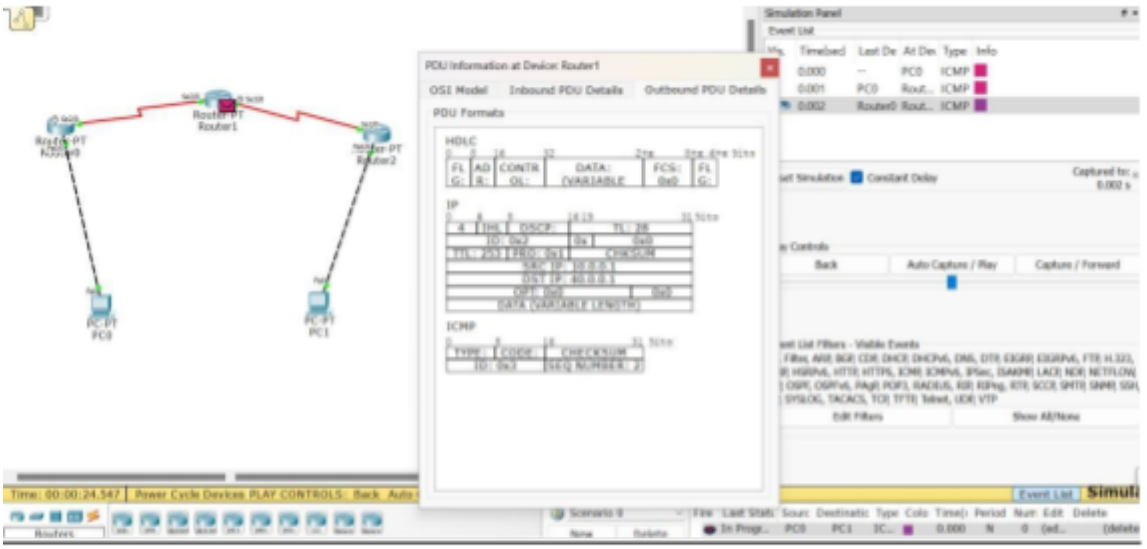
Observation:

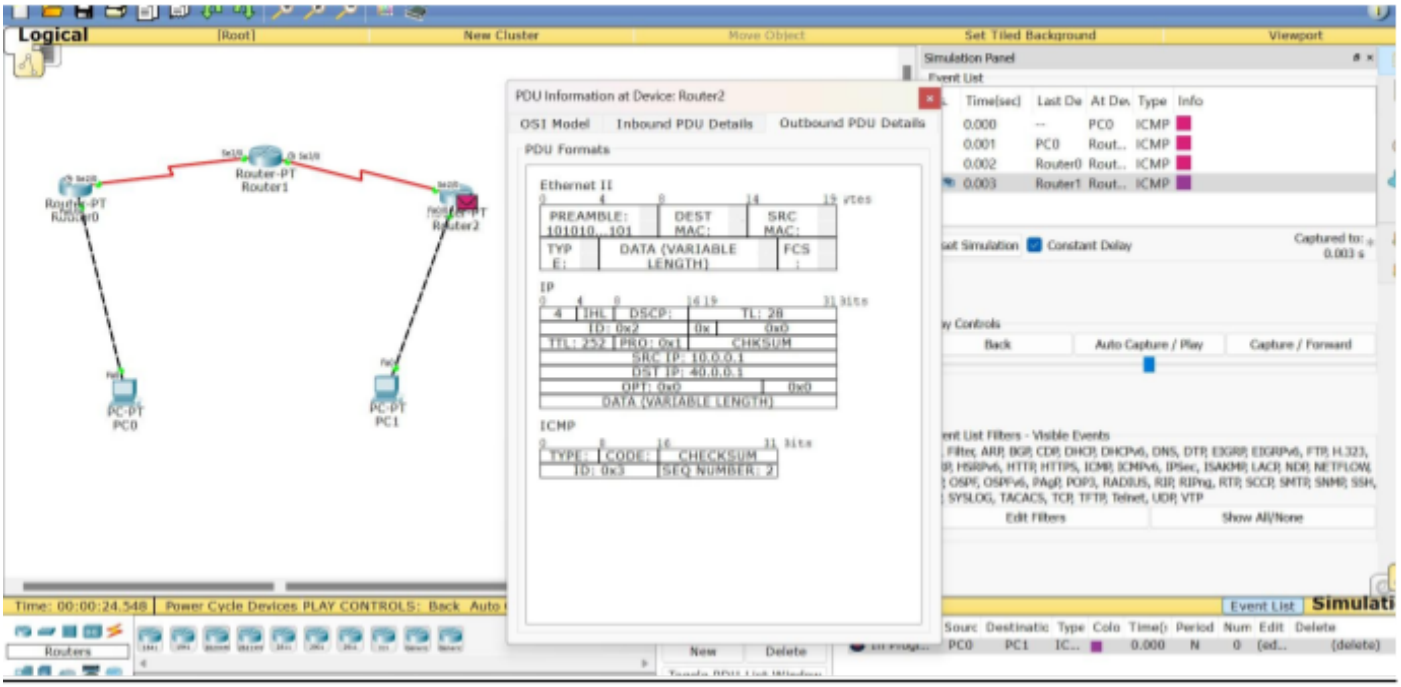
- The no of hops the packets travel before being discarded as TTL
- Router reduce TTL values by one until forwards packets
- when the TTL is 0, router discards it & s an ICMP message.

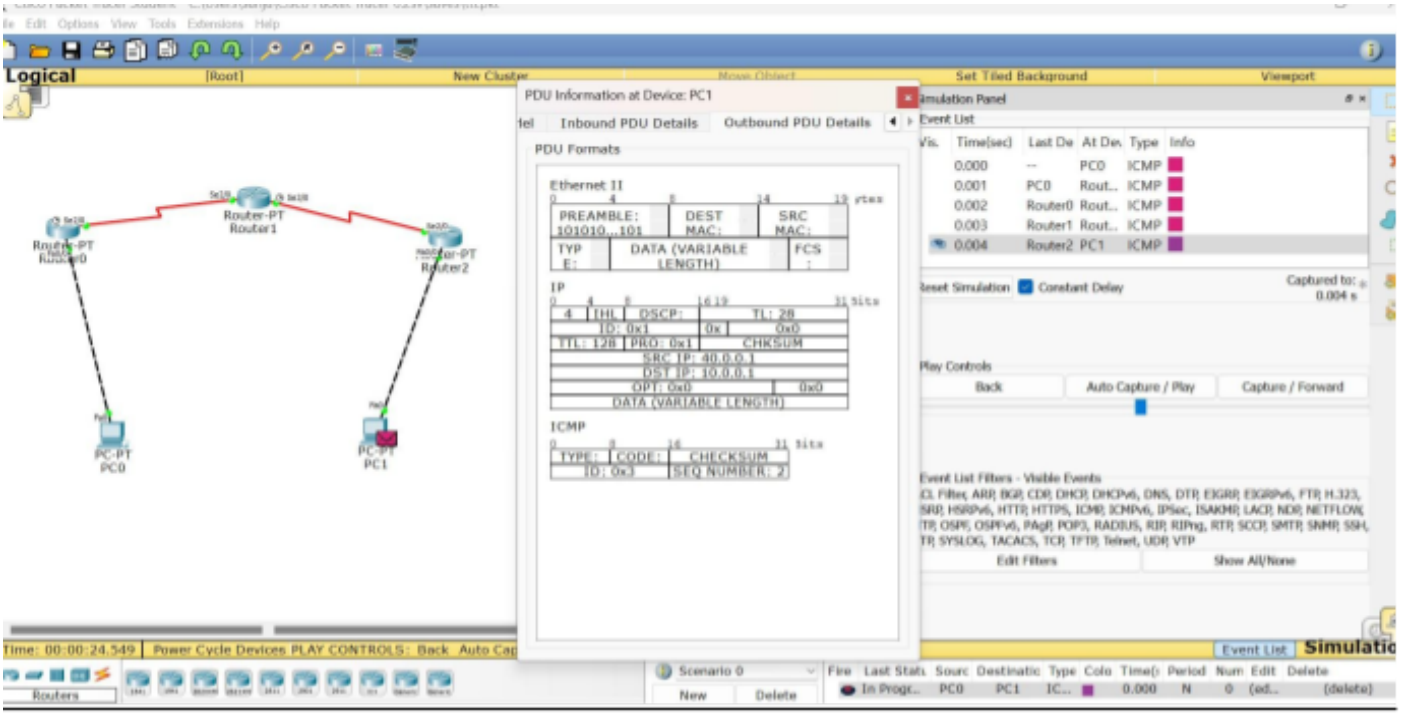
TOPOLOGY & OUTPUT








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:
TYP E:
DATA (VARIABLE LENGTH)
FCS
IP
ICMP
TYPE:
CODE:
CHECKSUM
ID: 0x3
SEQ NUMBER: 2
Simulation Panel
Event List
Time(sec)
Last De
At Dev
Type
Info
0.000
0.001
0.002
0.003
Router-PT Router1
Router-PT Router0
Router-PT Router2
PC-PT PC0
PC-PT PC1
Time: 00:00:24.548
Power Cycle Devices PLAY CONTROLS: Back
Routers
New
Delete


Logical
[Root]
PDU Information at Device: PC1
Inbound PDU Details
Outbound PDU Details
PDU Formats
Ethernet II
PREAMBLE: 101010...101
DEST MAC:
SRC MAC:
TYP E:
DATA (VARIABLE LENGTH)
FCS
IP
ICMP
TYPE:
CODE:
CHECKSUM
ID: 0x3
SEQ NUMBER: 2
Simulation Panel
Event List
Time(sec)
Last De
At Dev
Type
Info
0.000
0.001
0.002
0.003
0.004
Router-PT Router1
Router-PT Router0
Router-PT Router2
PC-PT PC0
PC-PT PC1
Time: 00:00:24.549
Power Cycle Devices PLAY CONTROLS: Back
Routers
Scenario 0
New
Delete

EXPERIMENT-11

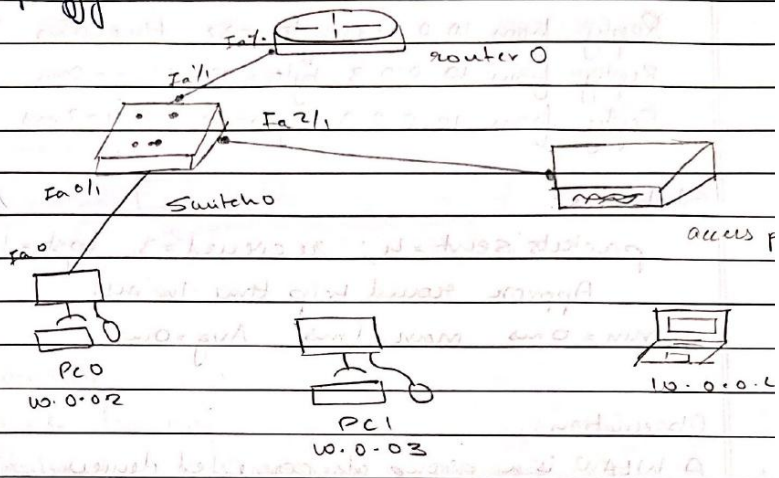
Q) To construct a WLAN and make the nodes communicate wirelessly

Date
Page

Program-11

Aim: To construct a WLAN and make nodes communication wirelessly.

Topology:



Procedure.

- Construct topology
- Configure P0 of router 0 as normaly done
- Configure access point 1- port 1 → SSID Name-
- select LOEP & give any 10 digit hex key - 01
- Configure PC1 of laptop
- Switch off device, drag the existing PT-HOST-NM- to the component list in LHS. drag WMP300N wireless interface
- 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 time out;

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

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

Reply from 10.0.0.3 bytes=32 time=2ms TTL

Statistics:

packets sent = 4, received = 3, lost = 1

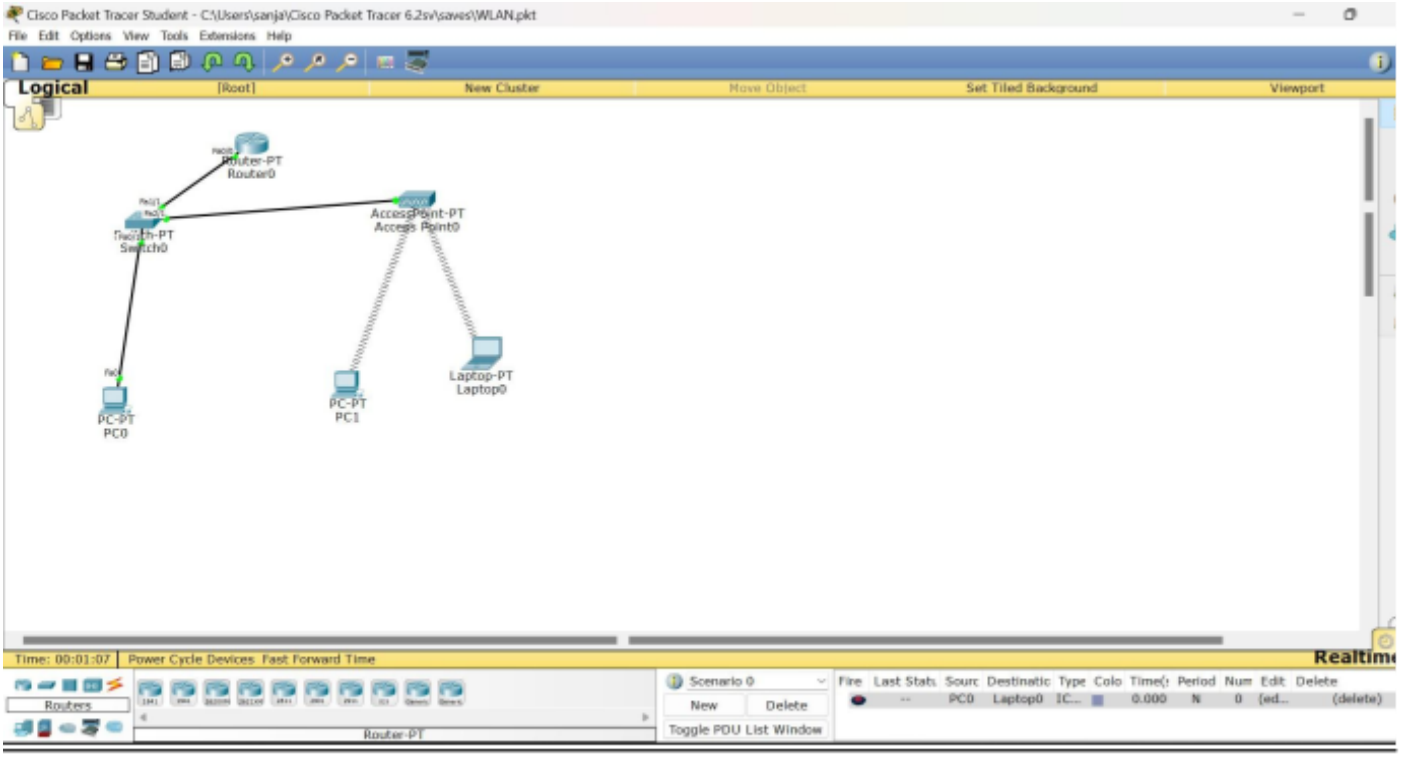
Approx round trip time in ms:

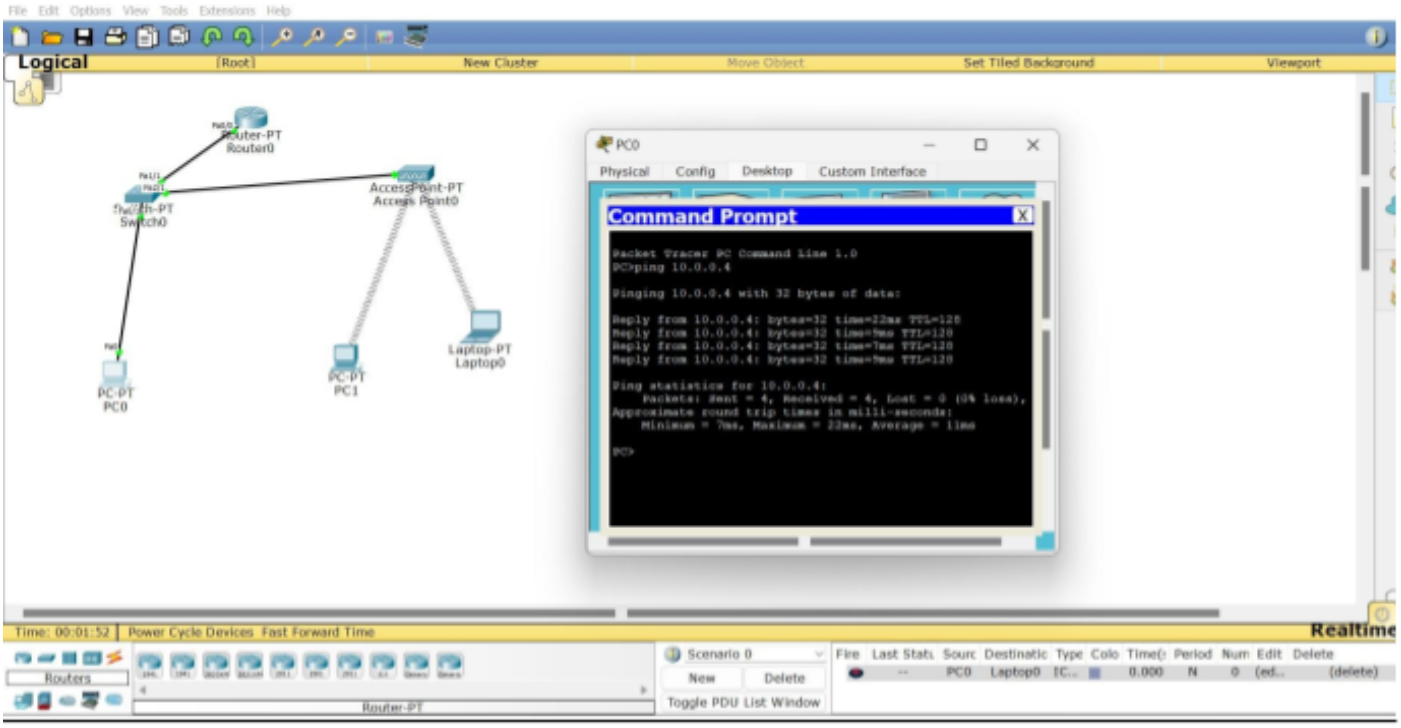
min = 0ms, max = 1ms, Avg = 0ms

Observation:

- A WLAN is a group of connected devices
- Data sent in packets contain layers with l
- The access point in the base station se as a hub to which other station are c

TOPOLOGY & OUTPUT





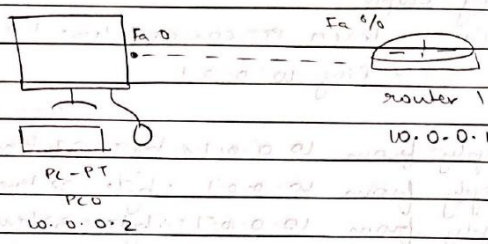
EXPERIMENT-12

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

Program - 12

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

Topology:



Procedure

- Create topology
- Configure the IP address & gateway for PC0
- Configure router
  - Step 1: enable
  - Step 2: config t
  - Step 3: host name r1
  - Step 4: Enable secret p1
  - Step 5: interface fastethernet 0/0
  - Step 6: ip address 10.0.0.1 255.0.0.0
  - Step 7: No shut
  - Step 8: line vty 05
  - Step 9: login
  - Step 10: password p0
  - Step 11: exit, exit
  - Step 12: wr

Page

Ping message to router
Password for user Acess verification is P0
Password for enable is P1.
Accessing router GLI from PC
show ip route

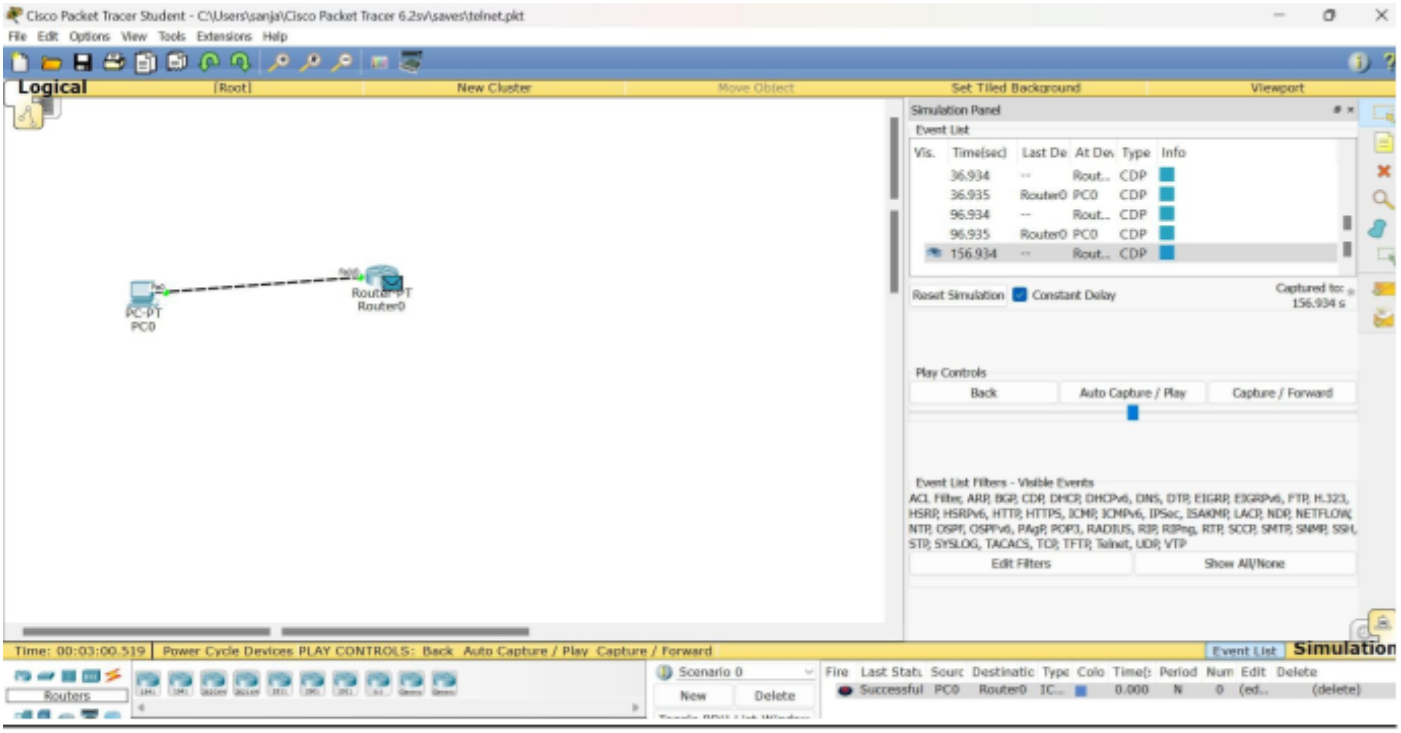
Ping Output
Packet tracer PC command line 1.0
PC> Ping 10.0.0.1

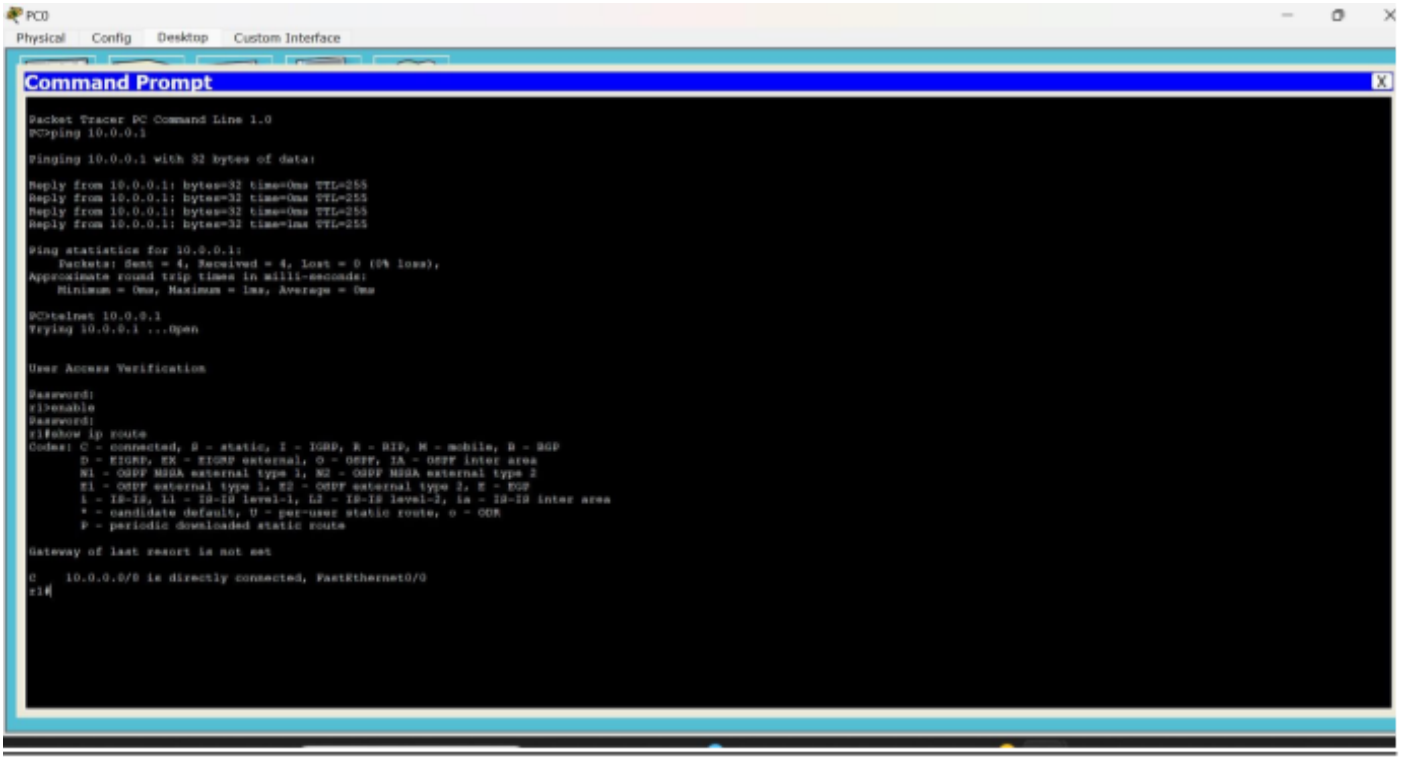
Reply from 10.0.0.1: byte = 32 time = 0ms TTL = 255
Reply from 10.0.0.1: byte = 32 time = 0 ms TTL = 255
Reply from 10.0.0.1: byte = 32 time = 0ms TTL = 255
Reply from 10.0.0.1: byte = 32 time = 0ms TTL = 255

Statistics:
Packets sent = 4, received = 4, lost = 0
Appron round trip time in ms:
min = 0ms, max = 0ms, Avg = 0ms

PC > telnet 10.0.0.1
Typing 10.0.0.1 ... : open
user Acess verification
Password: P0
P1 > enable
Password : P1
r1 # show ip route

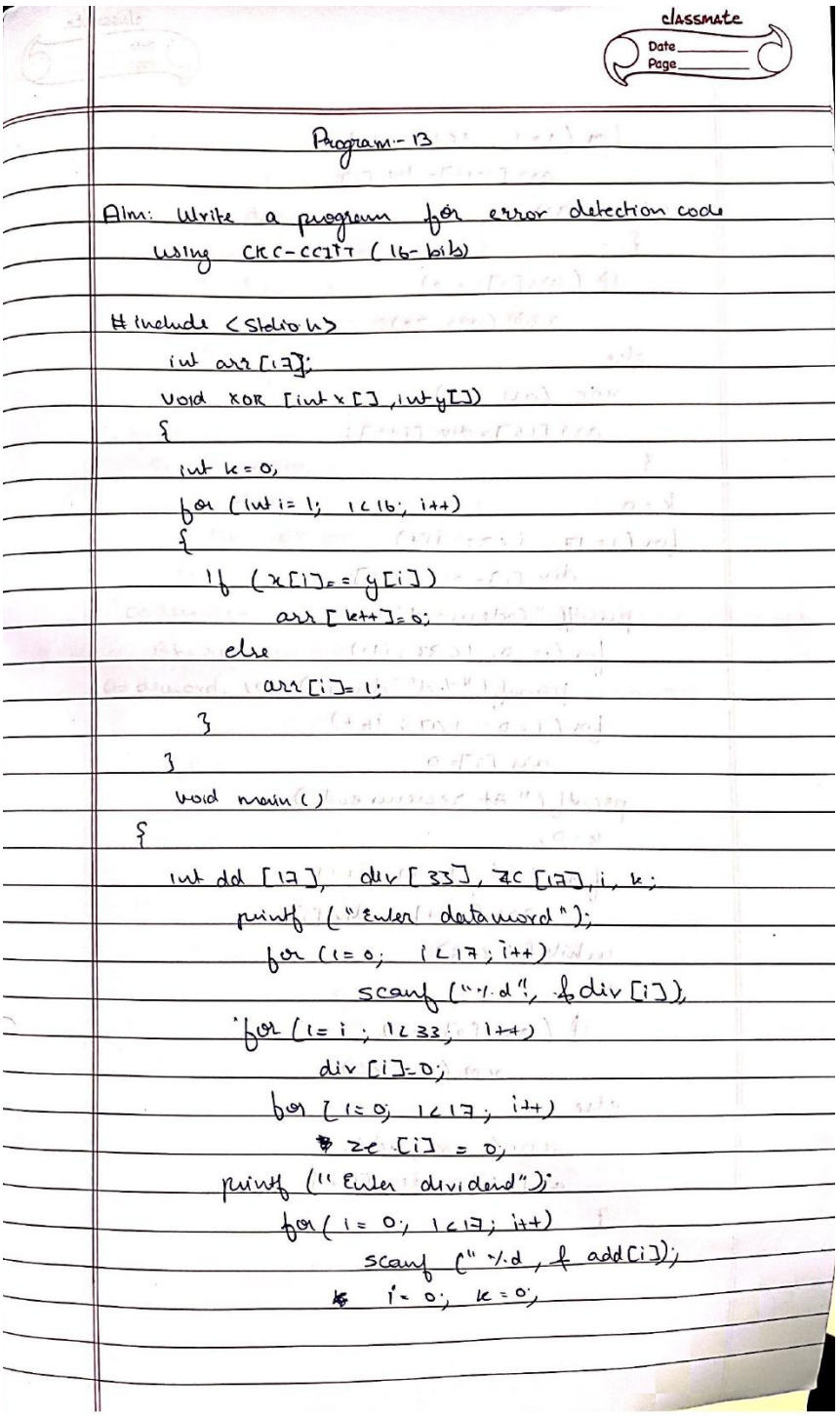
TOPOLOGY & OUTPUT





EXPERIMENT-13

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



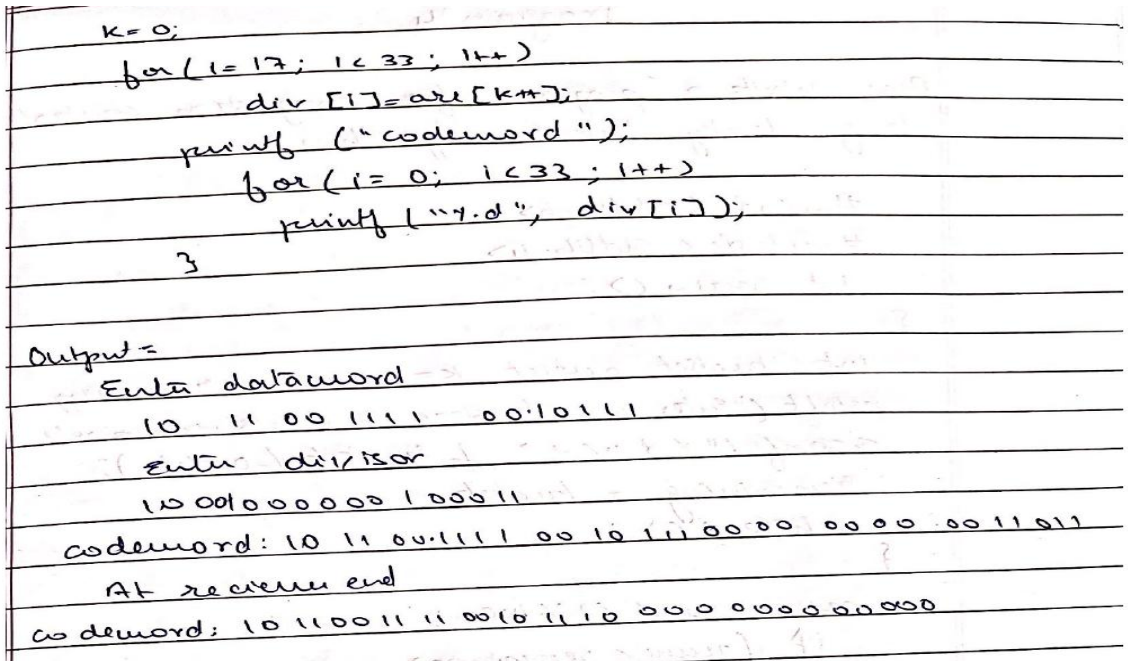
Program:- 13

Aim: Write a program for error detection code using CRC-CCITT (16-bits)

```
#include <stdio.h>
int arr[17];
void XOR [int x[], int y[])
{
 int k = 0,
 for (int i = 1; i<16; i++)
 {
  if (x[i] == y[i])
     arr [k++] = 0;
  else
     arr [i] = 1;
  }
 }
 void main()
{
 int dd [17], div [33], zc [17], i, k;
    printf ("Enter data word");
     for (i=0; i<17; i++)
          scanf ("%d", &div [i]);
    for (i = i ; i<33; i++)
        div [i] = 0;
     for (i=0; i<17; i++)
       zc [i] = 0;
    printf ("Enter dividend");
     for (i = 0; i<17; i++)
          scanf ("%d, &add[i]);
       i = 0; k = 0;
```

Scanned with CamScanner

```
for(i=i ; i<17; i++)
    arr[k++]= div [i];
while(i<33)
{
  if (arr[0]= = 0)
      xor (arr, ze);
else
  xor (arr, dd);
   arr[16]= div [i++];
}
k=0
for(i=17; i<33; i++)
      div [i]= arr[k++];
  printf("Codeword");
    for(i= 0; i<33 ; i++)
      printf("%d", div[i]);
    for(i=0; i<17 ; i++)
       arr[i]=0
  printf("At reciever end");
    k=0;
    for (i=i ; i<17; i++)
       arr[k++] = div [i];
  while [i<33)
  {
    if (arr [0]= = 0)
        xor (arr, ze);
else
    xor ( arr, dd );
    ar [16] = div [i++];
}
```



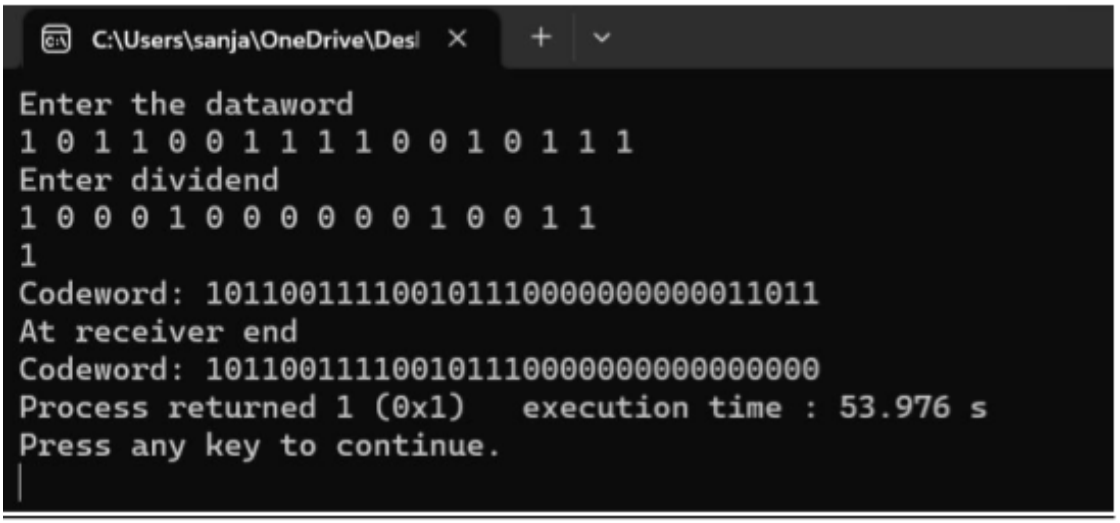
CODE-

```
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

Program 14.

Aim: Write a program for congestion control using leaky bucket algorithm.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
int bucket, outlet, k=1, num, remaining;
printf("Enter bucket size and outstream size");
scanf("%d %d", &bucket, &outlets);
remaining = buckets;
while (k);
{
num = rand() %1000;
if (num < remaining)
{
remaining = remaining - num;
printf("Packets of %d bytes are accepted", num)
}
else
{
printf("Packets of %d bytes is discarded", num)
}
if (buckets - remaining > outlets)
{
remaining += outlets;
}
else
remaining = buckets;
```

```
while (remaining < buckets).
{
  if (buckets - remaining > outlets)
  {
    remaining + = outlets;
  }
  else {
    remaining = buckets;
    printf (" Remaining bytes %d ", remaining);
  }
  return 0;
}
```

Output:

Enter bucket size and outstream size

1000 200

Packets of 41 bytes are accepted

Remaining bytes :1000

If you want to stop press 0, otherwise 1.

1

packets of 467 bytes are accepted

Remaining bytes 733

If you want to stop press 0, otherwise 1

1

packets of 334 bytes are accepted

Remaining bytes = 599.

CODE-

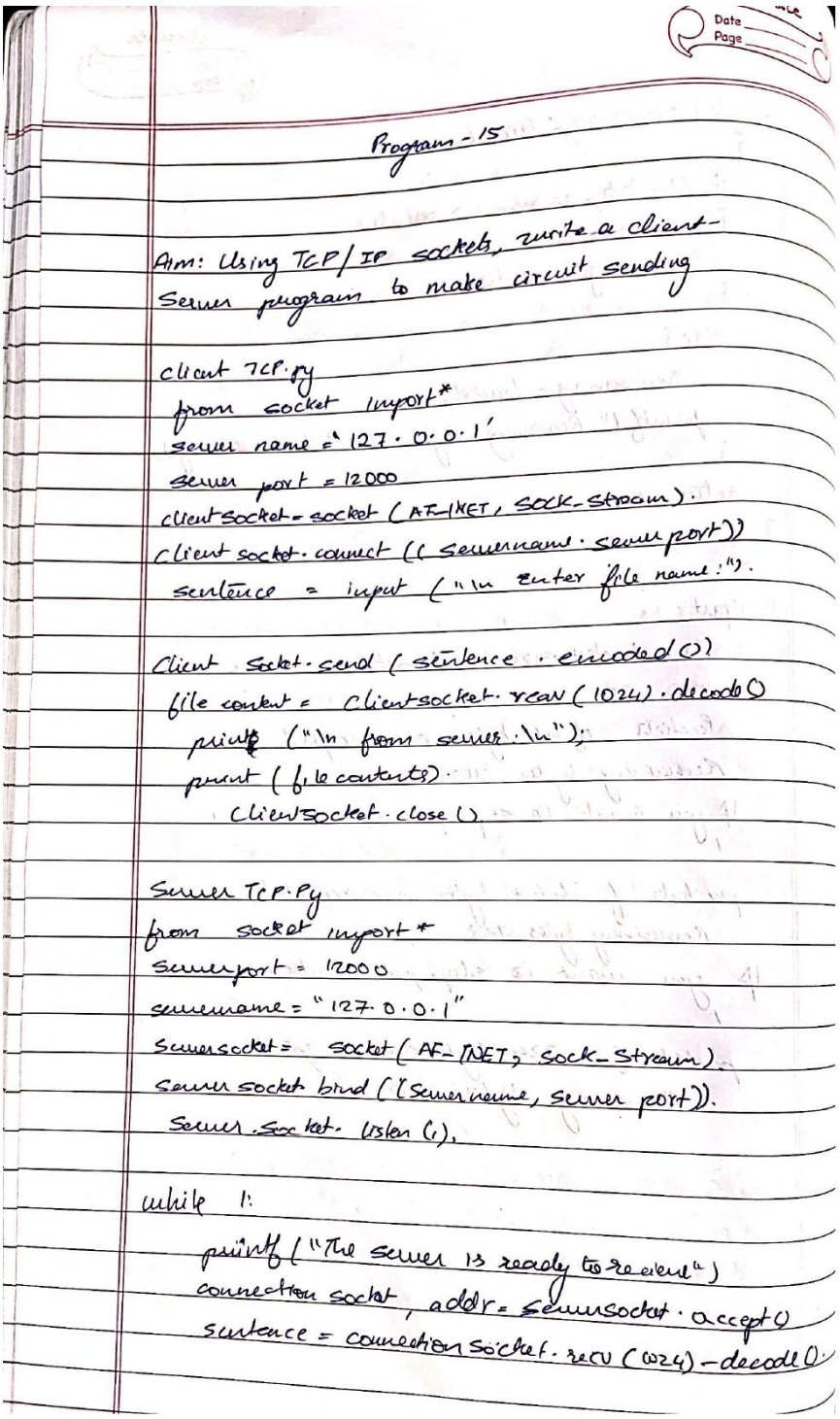
```
import java.util.*;
 class Leakybucket {
    public static void main(String[] args)
    {

       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.



Program - 15

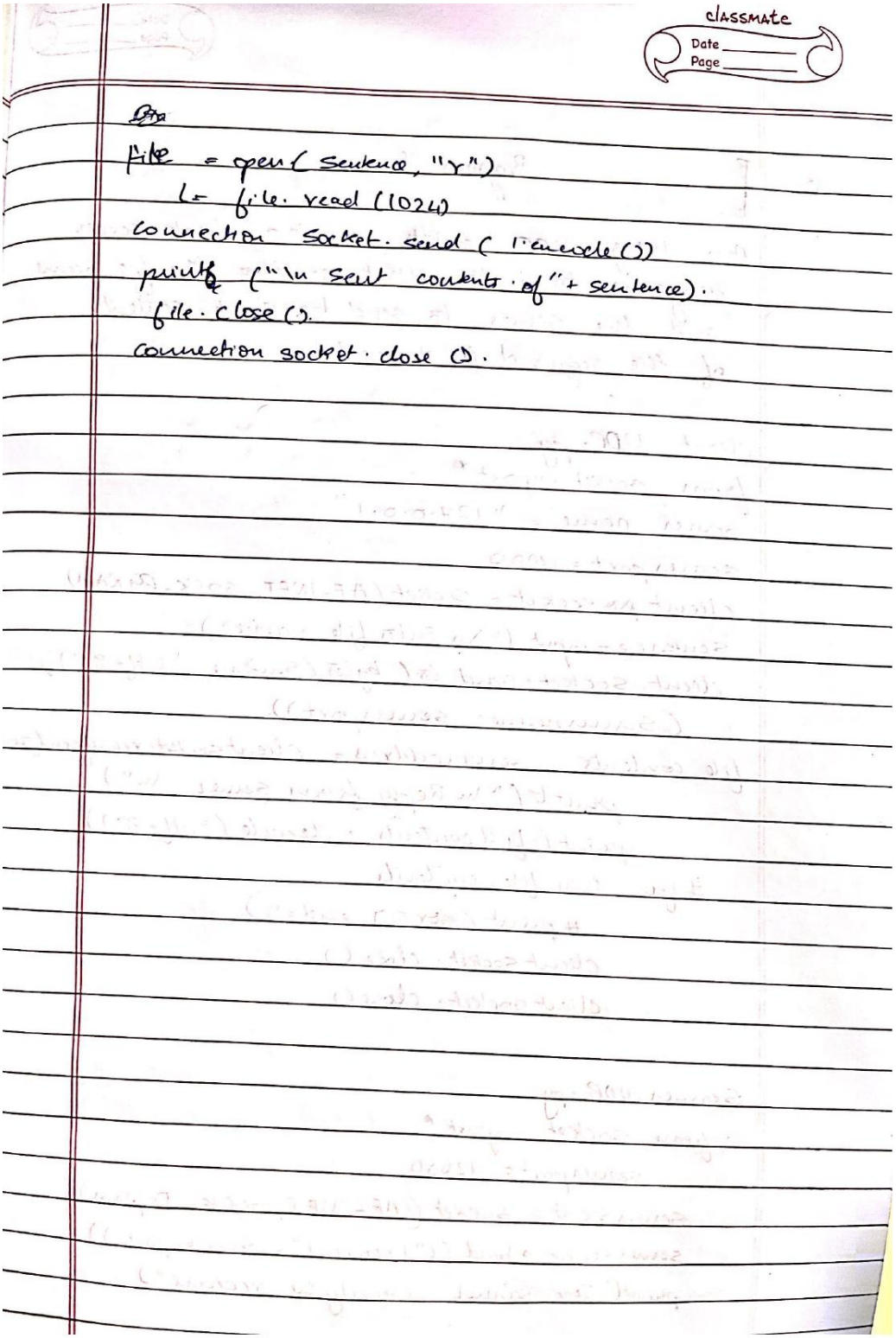
Aim: Using TCP/IP sockets, write a client-server program to make circuit sending

client TCP.py

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

Client Socket.send (sentence.encode())
file content = clientsocket.recv (1024).decode()
print ("\n from server:\n");
print (file contents)
    clientSocket.close()
```

Server TCP.Py

```
from socket import *
serverport = 12000
servername = "127.0.0.1"
Serversocket = socket (AF_INET, SOCK_Stream).
Server socket bind ((Servername, server port)).
Server.Socket. listen (1),

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



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

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

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/sanja/OneDrive/Documents/ClientTCP.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_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()
```

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/ServerTCP.py ========
===
The server is ready to receive
The server is ready to receive
The server is ready to receive

========== RESTART: C:/Users/sanja/OneDrive/Documents/ServerTCP.py ========
===
The server is ready to receive

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

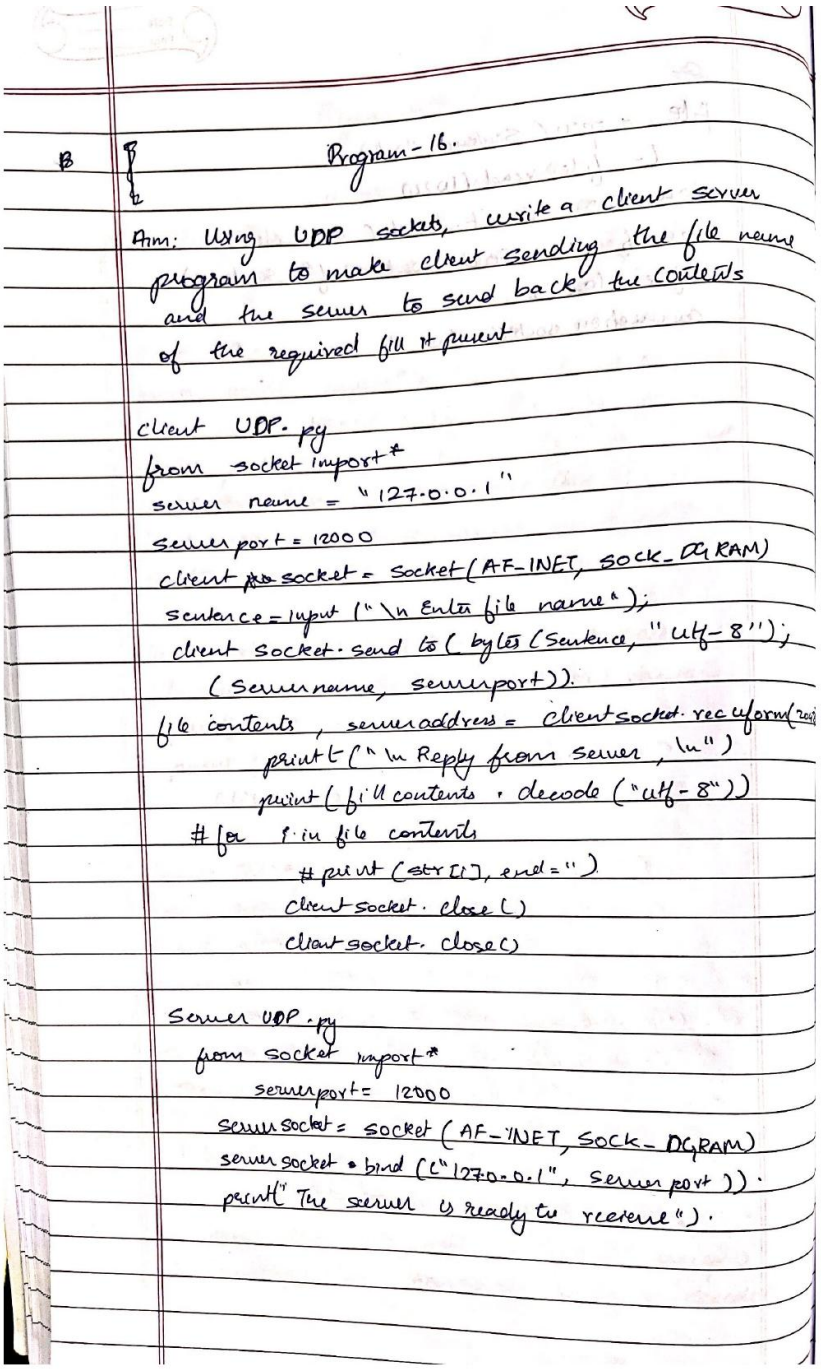
IDLE Shell 3.9.13

File Edit Shell Debug Options Window Help

```
the target machine actively refused it
>>>
========== RESTART: C:/Users/sanja/OneDrive/Documents/ClientTCP.py =========

Enter file name:ServerTCP.py

========== RESTART: C:/Users/sanja/OneDrive/Documents/ClientTCP.py =========

Enter file name:ServerTCP.py

========== RESTART: C:/Users/sanja/OneDrive/Documents/ClientTCP.py =========

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

Enter file name:ServerTCP.py

From Server:

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

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

   connectionSocket.send(l.encode())
   print ("\nSent contents of " + sentence)
   file.close()
   connectionSocket.close()


>>>
```

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



Program-16.

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 required file if present

client UDP.py

```
from socket import *
server name = "127.0.0.1"
server port = 12000
client socket = socket(AF_INET, SOCK_DGRAM)
sentence = input("\n Enter file name");
client socket.sendto(bytes(Sentence, "utf-8"));
(Servername, serverport)).
file contents, serveraddress = clientsocket.recvfrom(
print("\n Reply from server, \n")
print(file contents.decode("utf-8"))
# for i in file contents
# print(str[i], end="")
client socket.close()
client socket.close()
```

Server UDP.py

```
from socket import *
serverport = 12000
Server socket = socket(AF_INET, SOCK_DGRAM)
server socket.bind(("127.0.0.1", Server port)).
print("The server is ready to receive").
```

```
while (1)
    sentence, clientaddress = serversocket.recvfrom(2048)
    sentence = sentence.decode("Utf-8").
    file = open(sentence, "n")
    con = file.read(2048)
serversocket.sendto(bytes(con, "utf-8") clientaddress)
    print(`In sent content of ', end = '')
        print(sentence)
    # for (i in sentence)
            # print(str(i), end = '').
        file.close().
```

CODE-

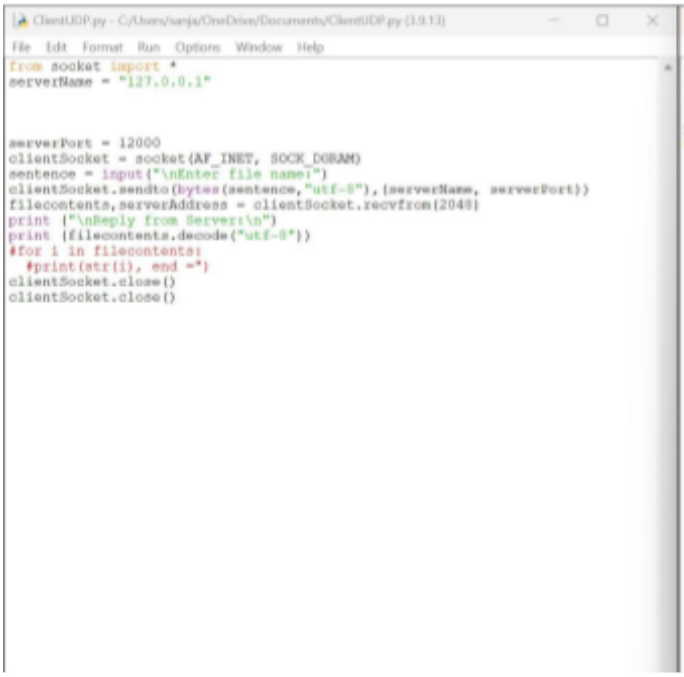
ClientUDP.py

```
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

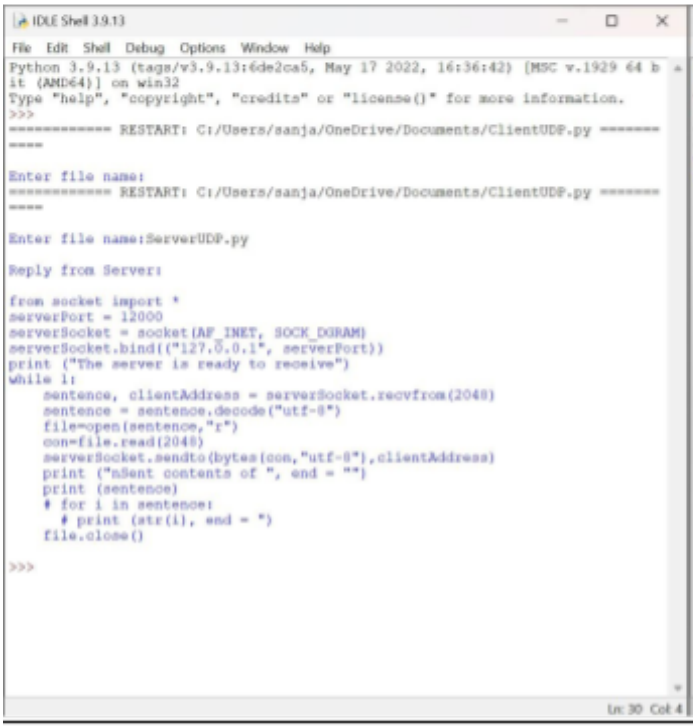
```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 1:
   sentence, clientAddress = serverSocket.recvfrom(2048)
   sentence = sentence.decode("utf-8")
   file=open(sentence,"r")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



```
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()
```

```
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()
```

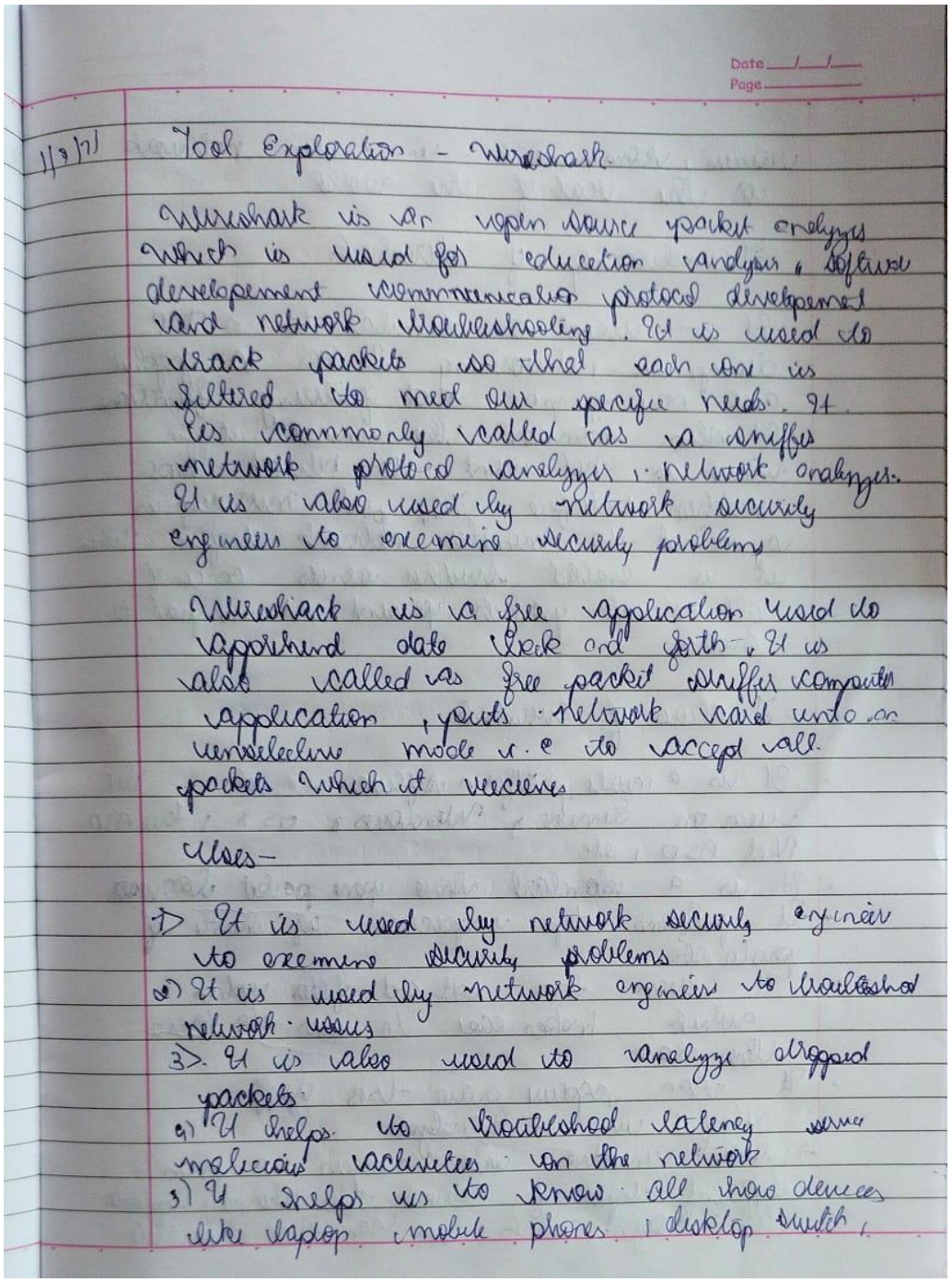


```
File  Edit  Shell  Debug  Options  Window  Help
Python 3.9.13 (tags/v3.9.13:6de2ca5, May 17 2022, 16:36:42) [MSC v.1929 64 b
it (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/sanja/OneDrive/Documents/ClientUDP.py =======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

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()

>>>
```

```
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
```

EXPERIMENT-17

Q) Tool Exploration -Wireshark



11/9/21

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 promiscuous 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,

Date / /
Page

wireless 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 NIC. (adapter interface). The port mirroring is a method to monitor network traffic. When it is enabled switches sends copies of all network packets present at one port to another.

Features of wireshark.

- It is a multi platform software i.e it can run on Linux, Windows, OS X, FreeBSD, NetBSD, etc
- It is a standard three pane packet browser.
- It performs deep inspection of hundreds 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 wireshark does analysis i.e from different types of network like ethernet, loopback etc through which we can read live data