VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

COMPUTER NETWORKS

Submitted by

PURVIK C (1BM21CS149)

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 **PURVIK C(1BM21CS149),** 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

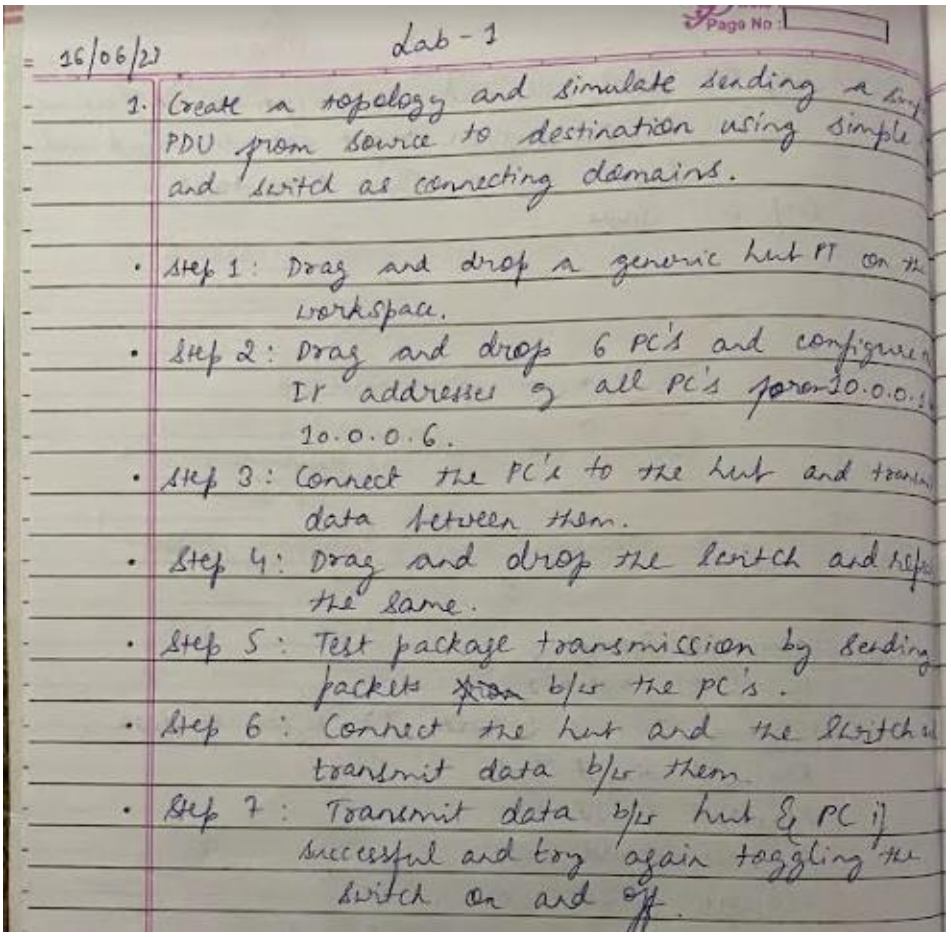
Professor and Head
Department of CSE
BMSCE, Bengaluru

`

Index

EXPERIMENT-1

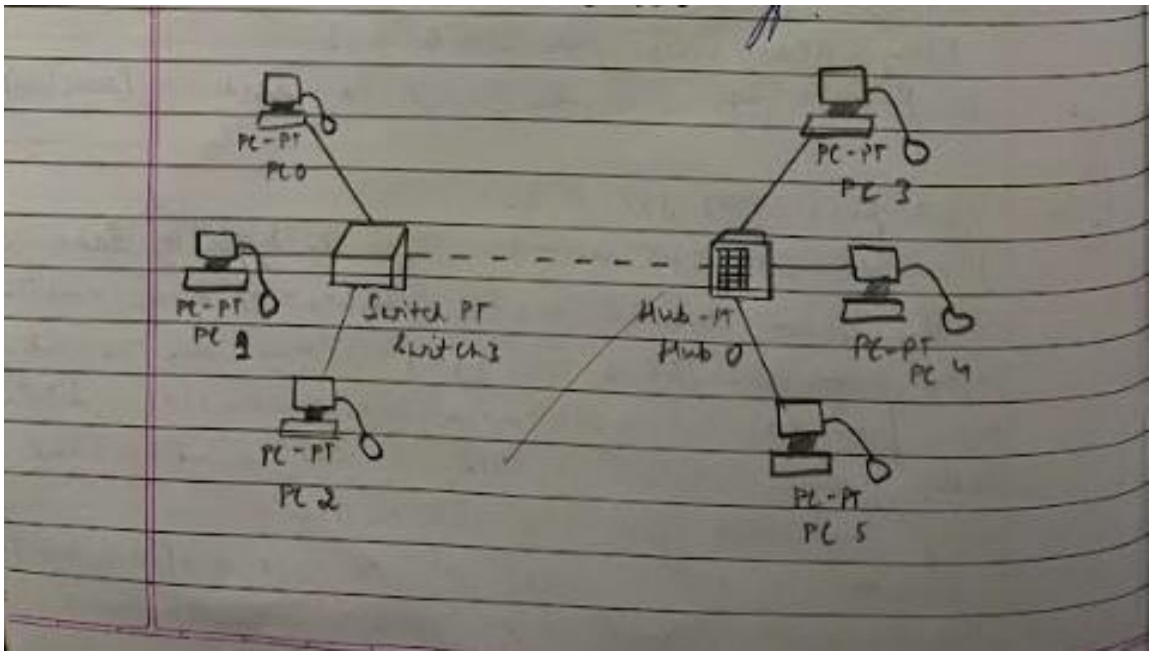
Q) Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message.



16/06/23 Lab - 1

1. Create a topology and simulate sending a simple PDU from source to destination using simple and switch as connecting domains.

- Step 1: Drag and drop a generic hub PT on the workspace.
- Step 2: Drag and drop 6 PC's and configure IP addresses of all PC's from 10.0.0.1 to 10.0.0.6.
- Step 3: Connect the PC's to the hub and transfer data between them.
- Step 4: Drag and drop the switch and repeat the same.
- Step 5: Test package transmission by sending packets b/w the PC's.
- Step 6: Connect the hub and the switch to transmit data b/w them.
- Step 7: Transmit data b/w hub & PC if successful and try again toggling the switch on and off.


PC-PT
PC 0
PC-PT
PC 1
PC-PT
PC 2
Switch PT
Switch 3
Hub-PT
Hub 0
PC-PT
PC 3
PC-PT
PC 4
PC-PT
PC 5

Command Prompt Pinging a PC

PC> ping 10.0.0.5
Pinging 10.0.0.5 with 32 bytes of data

Request from 10.0.0.5 : bytes = 32 time = 0ms TTL = 128
Request from 10.0.0.5 : bytes = 32 time = 0ms TTL = 128
Request from 10.0.0.5 : bytes = 32 time = 0ms TTL = 128
Request from 10.0.0.5 : bytes = 32 time = 0ms TTL = 128

Ping Statistics for 10.0.0.5
Packets sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milliseconds;
Minimum = 0ms, Maximum = 0ms, Average = 0ms

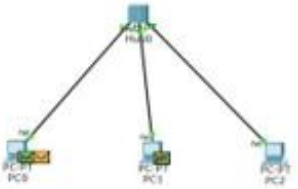
When switch is off

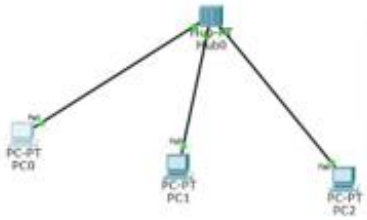
Pinging 10.0.0.5 with 32 bytes of data
Request timed out.
Request timed out
Request timed out
Request timed out

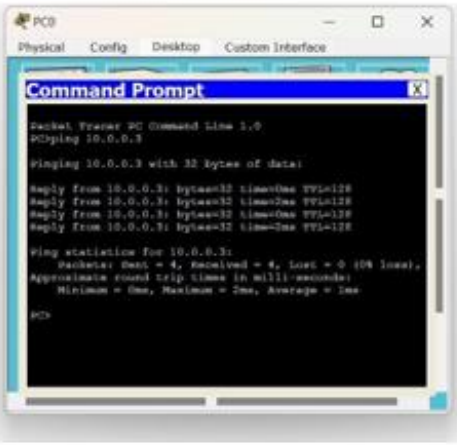
Ping Statistics for 10.0.0.4
Packets: Sent = 4, Received 0, Lost = 4 (100% loss)

TOPOLOGY & OUTPUT

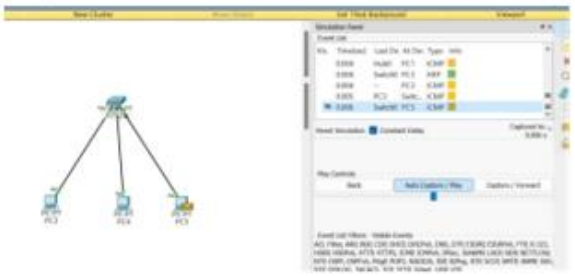
1. Hub and PCs



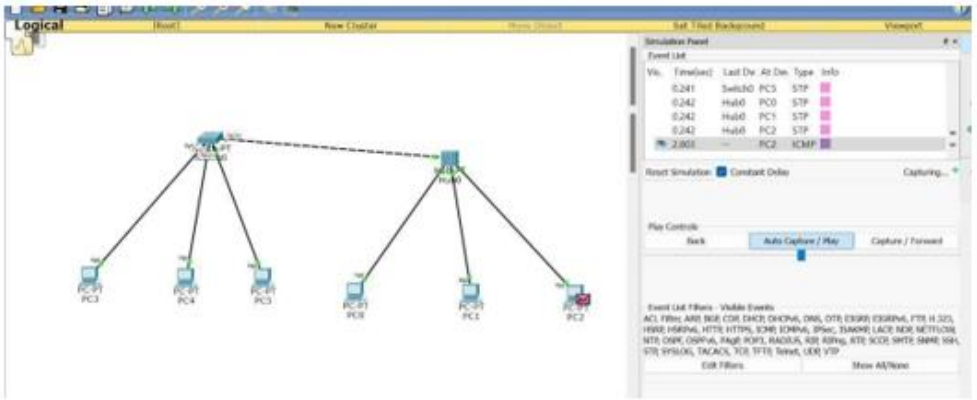


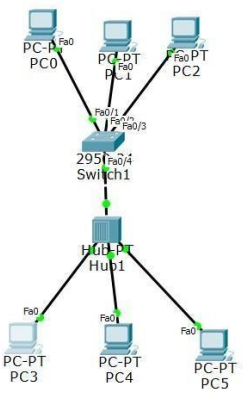


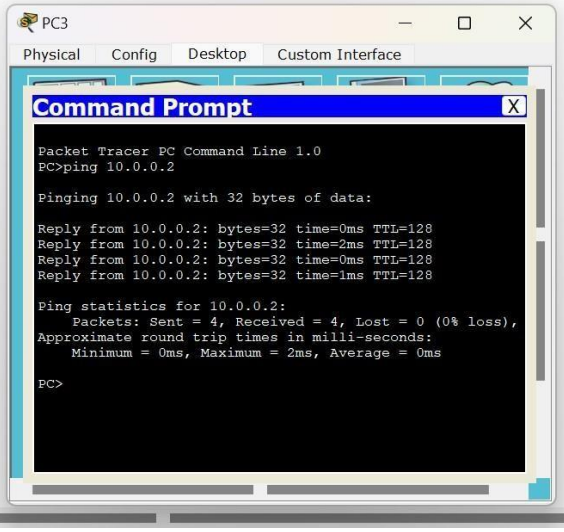
2. Switch and PCs



3. Hub,Switch and PCs



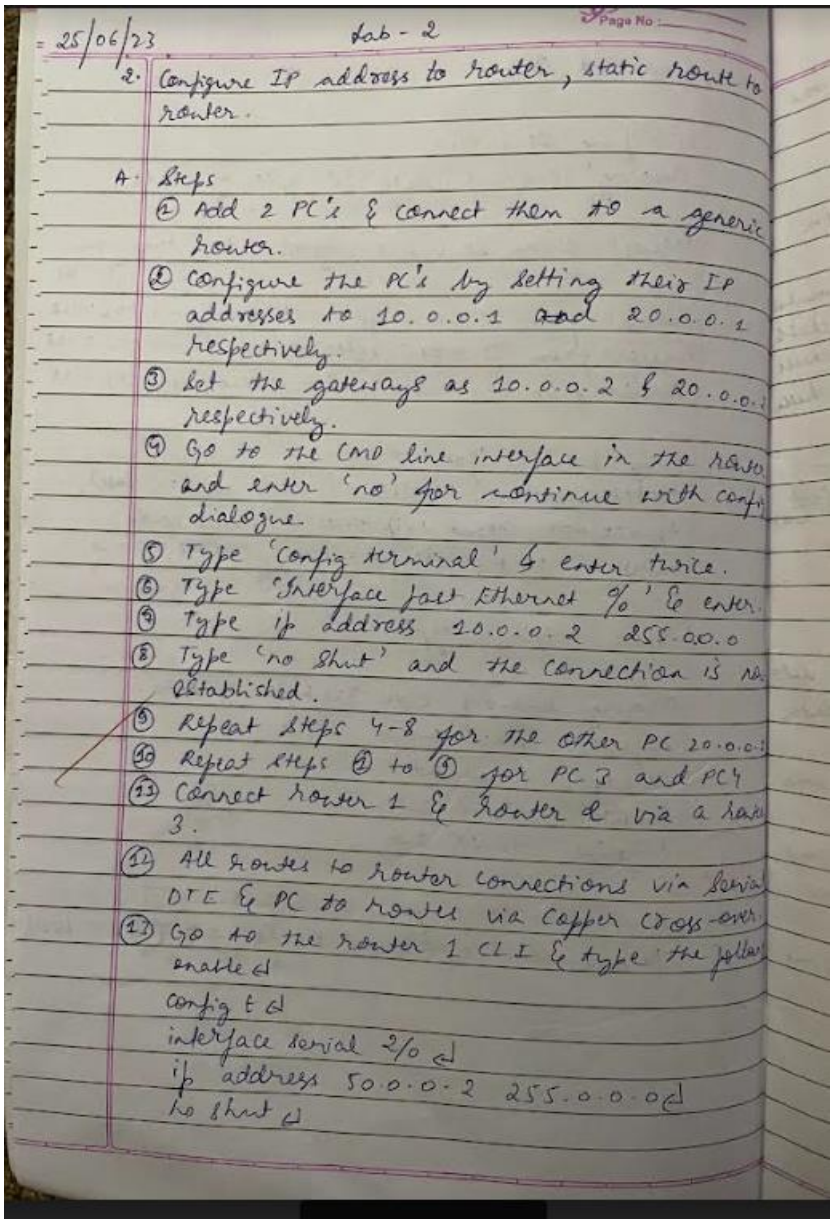




EXPERIMENT-2

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

PROGRAM 2.1



25/06/23 Lab - 2

2. Configure IP address to router, static route to router.

A. Steps

1. Add 2 PC's & connect them to a generic router.
2. Configure the PC's by setting their IP addresses to 10.0.0.1 ~~and~~ 20.0.0.1 respectively.
3. Set the gateways as 10.0.0.2 & 20.0.0.2 respectively.
4. Go to the CMD line interface in the router and enter 'no' for continue with config dialogue.
5. Type 'config terminal' & enter twice.
6. Type 'interface fast Ethernet 0/0' & enter.
7. Type ip address 10.0.0.2 255.00.0
8. Type 'no shut' and the connection is established.
9. Repeat steps 4-8 for the other PC 20.0.0.2
10. Repeat steps 2 to 9 for PC 3 and PC 4
11. Connect router 1 & router 2 via a hate 3.
12. All routers to router connections via Serial DTE & PC to router via copper cross-over.
13. Go to the router 1 CLI & type the follow

enable ↵
config t ↵
interface serial 2/0 ↵
ip address 50.0.0.2 255.0.0.0 ↵
no shut ↵

(14) Go to the router 3 CLI & type:
enable ↵
config t ↵
interface serial 2/0 ↵
ip address 50.0.0.1 255.0.0.0 ↵
no shut ↵
The connection b/w Router 1 & 3 is green now

(15) Repeat steps (13) & (14) for router 2 to 3 similarly.

(16) Go to router 1 CLI & type show ip route.
It shows only the direct connections.

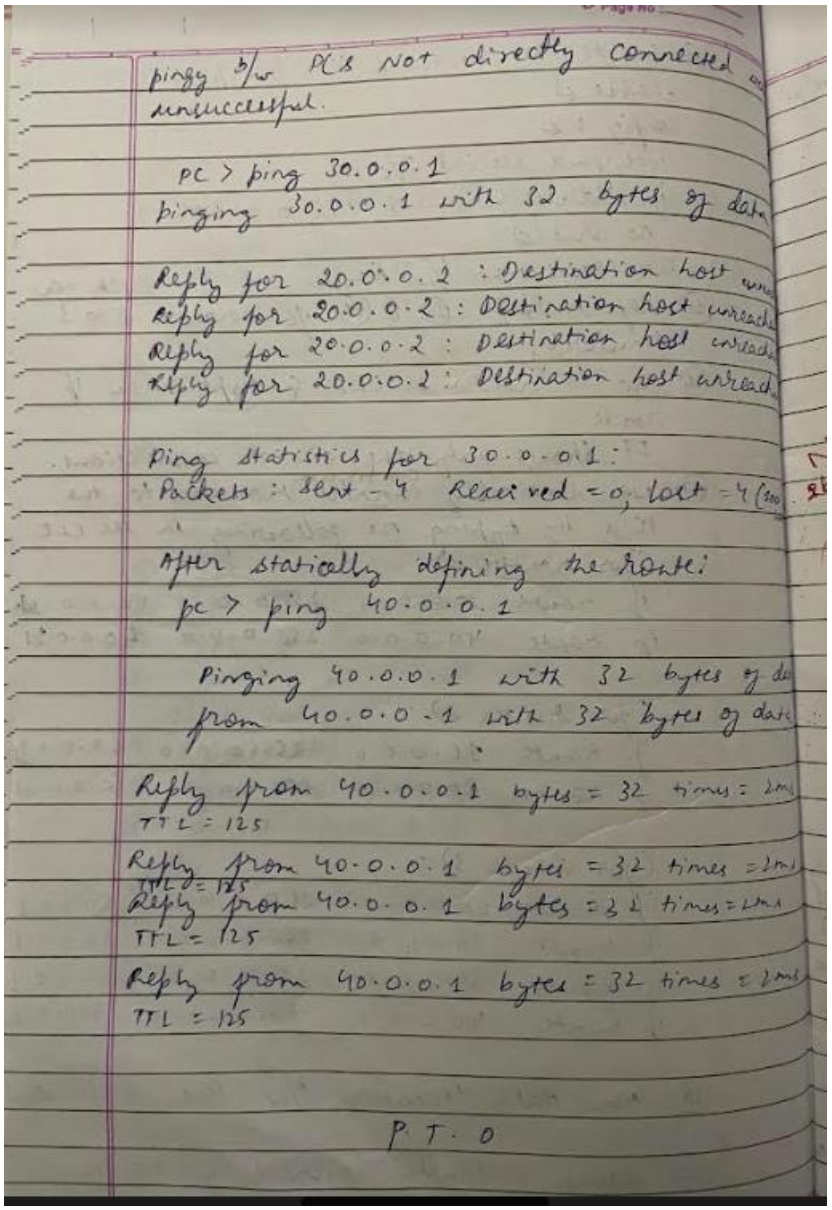
(17) We statically connect routers to the PC's by typing the following in the CLI
(for router 1):
ip route 30.0.0.0 255.0.0.0 50.0.0.1 ↵
ip route 40.0.0.0 255.0.0.0 40.0.0.1 ↵

(for router 2):
ip route 10.0.0.0 255.0.0.0 60.0.0.1 ↵
ip route 20.0.0.0 255.00.0 60.0.0.1 ↵

(for route 3):
ip route 10.0.0.0 255.0.0.0 50.0.0.1
ip route 20.0.0.0 255.0.0.0 50.0.0.1
ip route 30.0.0.0 255.0.0.0 60.0.0.1
ip route 40.0.0.0 255.0.0.0 60.0.0.1

(18) Now data transfer b/w PCs is successful

(19) Before statically connecting routes



ping b/w PCs not directly connected unsuccessful.

PC > ping 30.0.0.1

pinging 30.0.0.1 with 32 bytes of data

Reply for 20.0.0.2 : Destination host unreachable
Reply for 20.0.0.2 : Destination host unreachable
Reply for 20.0.0.2 : Destination host unreachable
Reply for 20.0.0.2 : Destination host unreachable

Ping statistics for 30.0.0.1:
Packets : Sent = 4 Received = 0, lost = 4 (100% loss)

After statically defining the route:
PC > ping 40.0.0.1

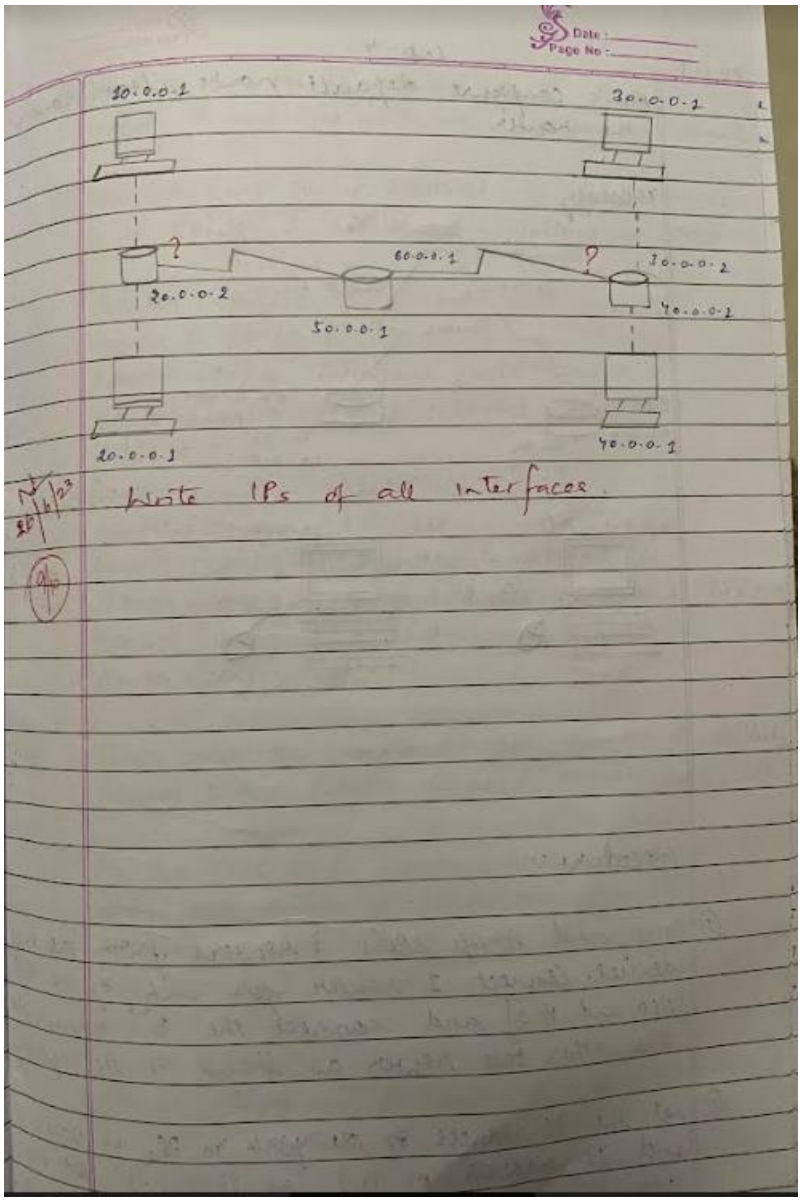
Pinging 40.0.0.1 with 32 bytes of data
from 40.0.0.1 with 32 bytes of data:

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

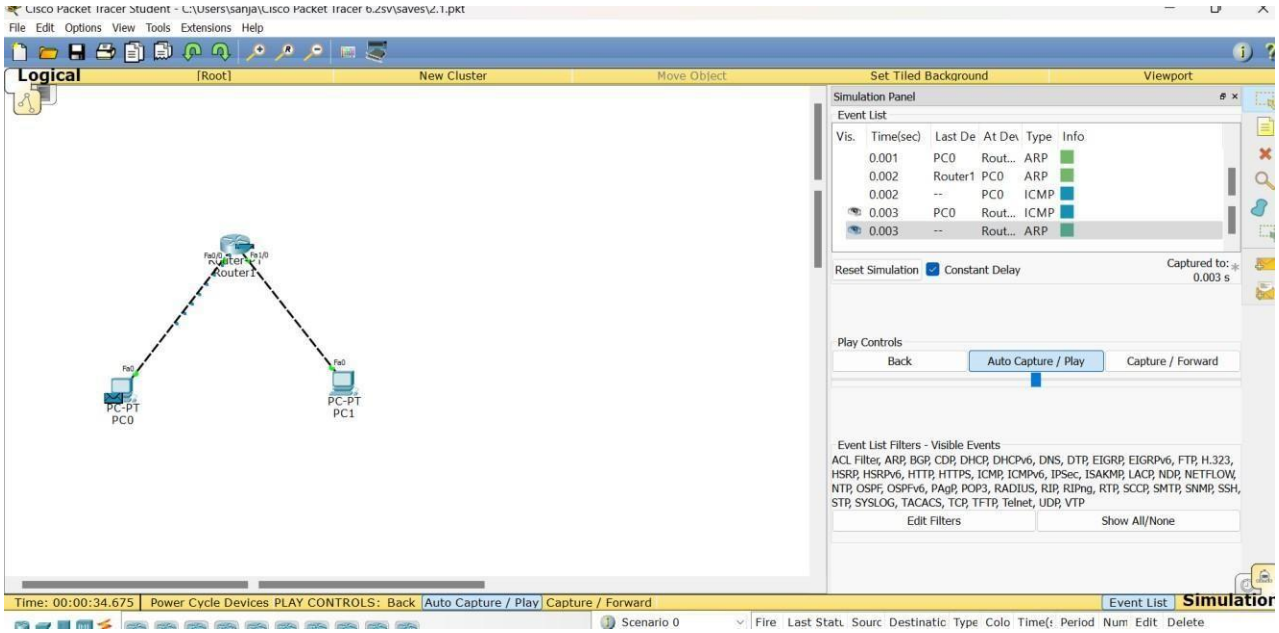
Reply from 40.0.0.1 bytes = 32 times = 2ms
TTL = 125
Reply from 40.0.0.1 bytes = 32 times = 2ms
TTL = 125

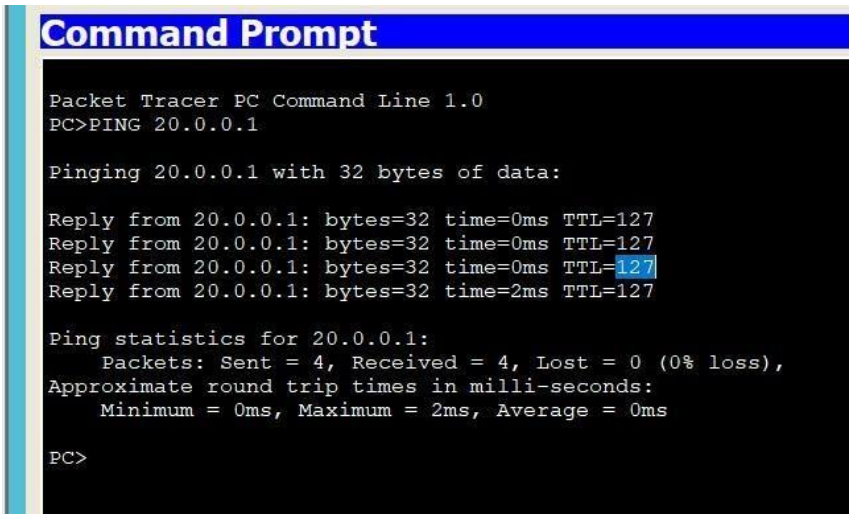
Reply from 40.0.0.1 bytes = 32 times = 2ms
TTL = 125

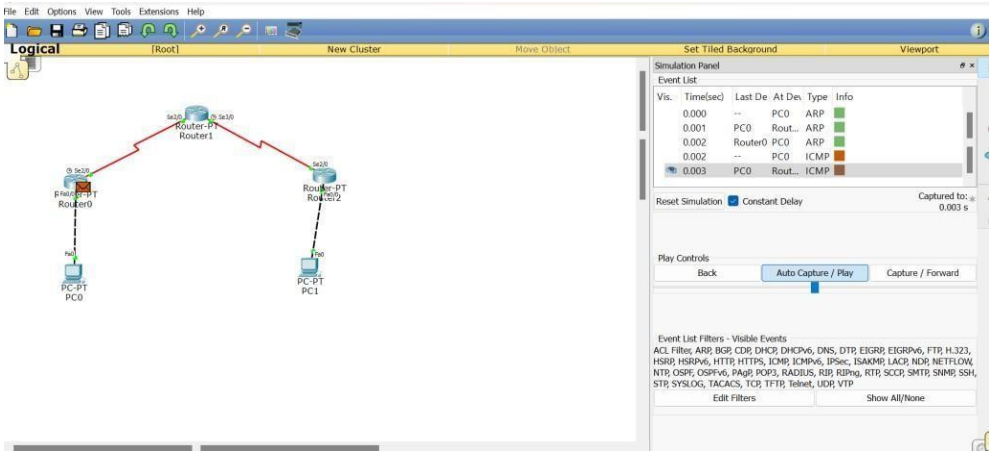
P.T.O



TOPOLOGY &OUTPUT


File Edit Options View Tools Extensions Help
Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router1
PC-PT
PC0
PC-PT
PC1
Simulation Panel
Event List
Vis. Time(sec) Last De At De Type Info
0.001 PC0 Rout... ARP
0.002 Router1 PC0 ARP
0.002 -- PC0 ICMP
0.003 PC0 Rout... ICMP
0.003 -- Rout... ARP
Reset Simulation
Constant Delay
Captured to: 0.003 s
Play Controls
Back
Auto Capture / Play
Capture / Forward
Event List Filters - Visible Events
ACL Filter, ARP, BGP, CDP, DHCP, DHCPv6, DNS, DTP, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, LACP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, RADIUS, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, VTP
Edit Filters
Show All/None
Time: 00:00:34.675
Power Cycle Devices PLAY CONTROLS: Back
Auto Capture / Play
Capture / Forward
Event List
Simulation
Scenario 0
Fire Last Statu Sourc Destinatic Type Colo Time(s Period Num Edit Delete

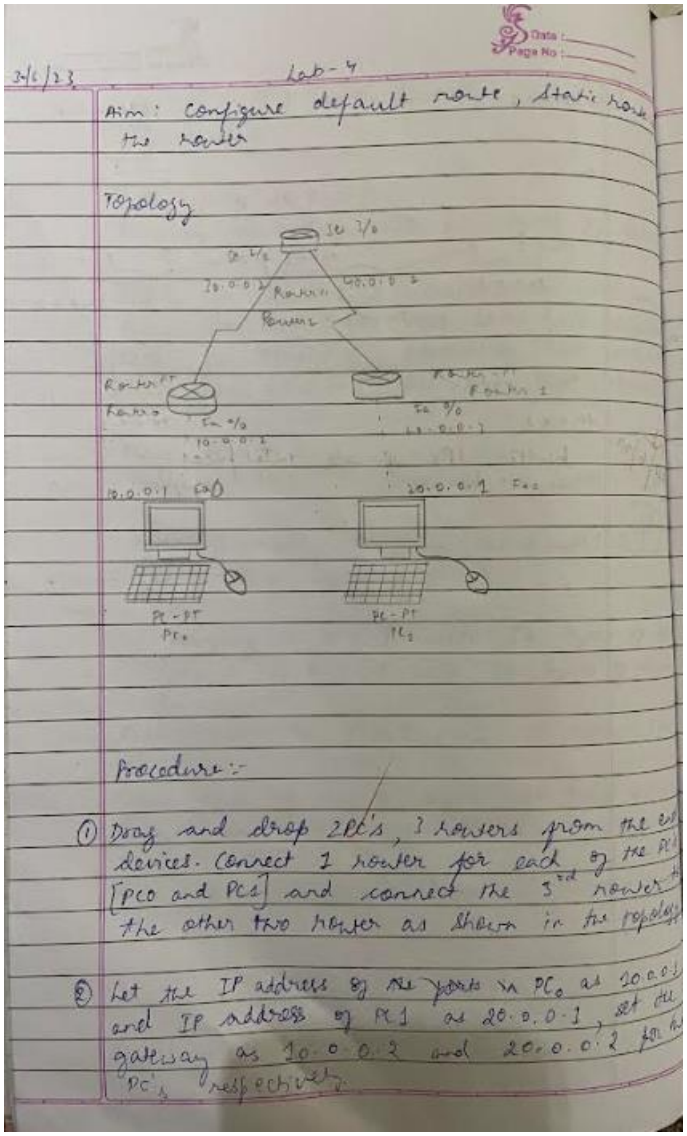

Command Prompt
Packet Tracer PC Command Line 1.0
PC>PING 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127
Reply from 20.0.0.1: bytes=32 time=2ms TTL=127
Ping statistics for 20.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 2ms, Average = 0ms
PC>

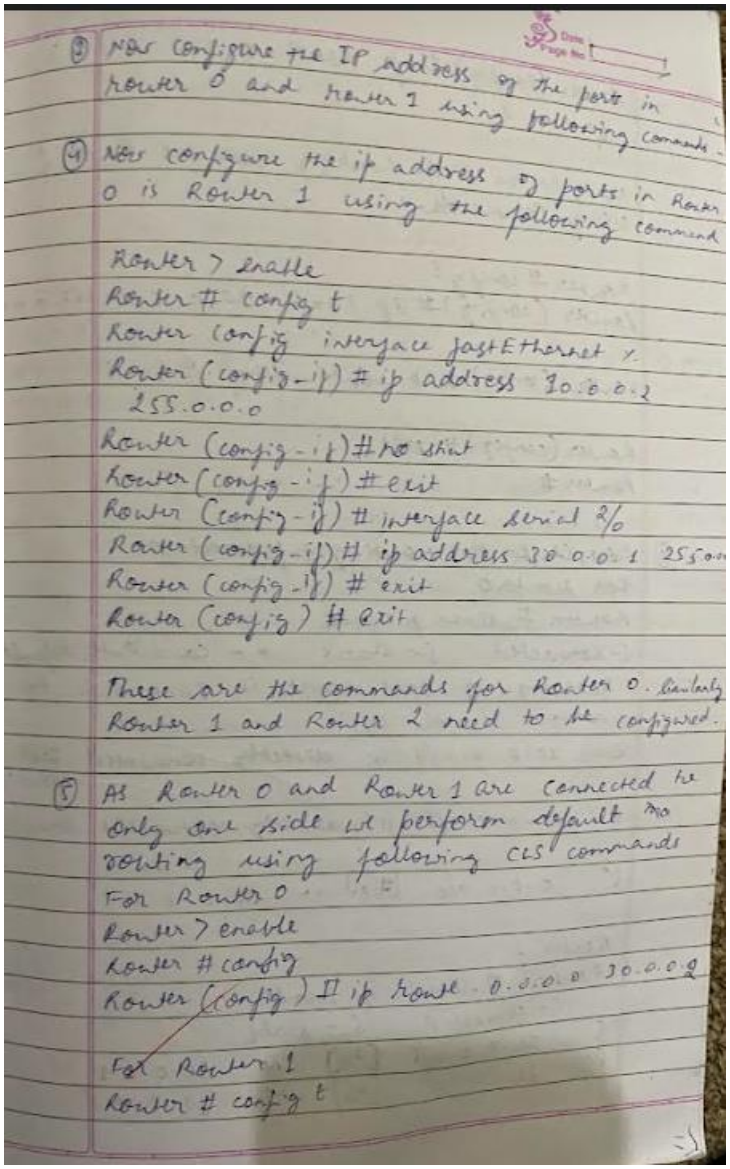



```
Packet Tracer PC Command Line 1.0
PC>40.0.0.1
Invalid Command.

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





(3) Now configure the IP address of the ports in Router 0 and Router 1 using following commands -

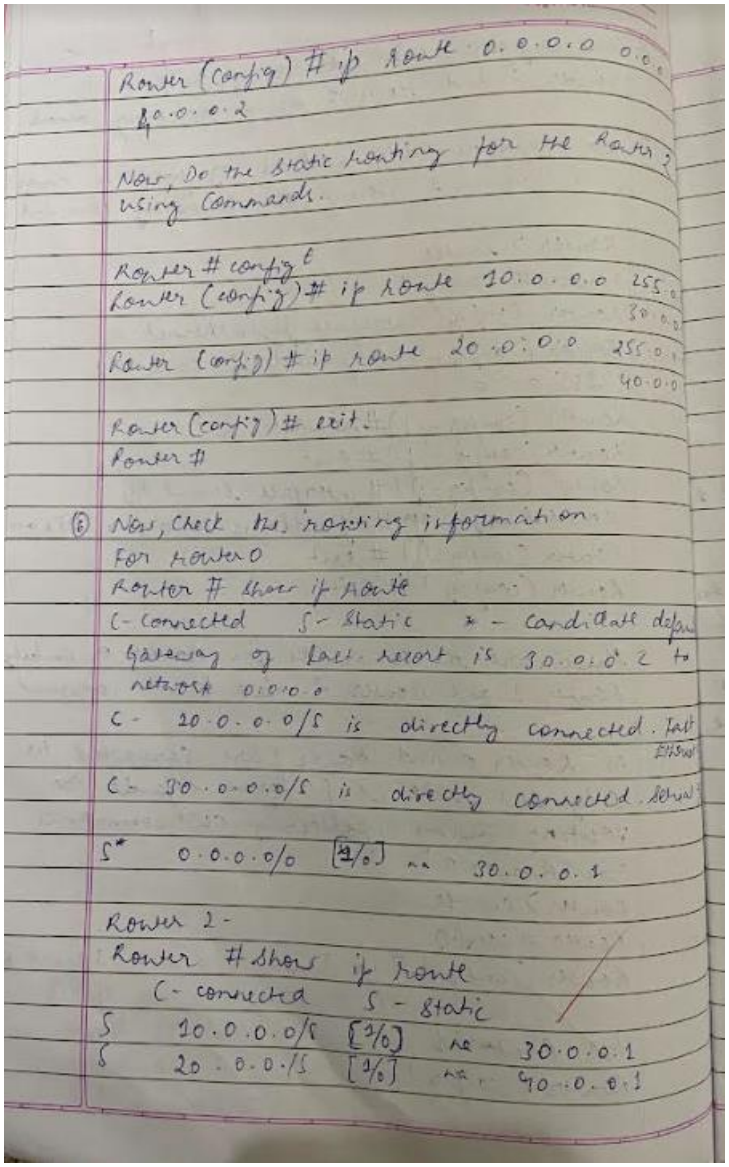
(4) Now configure the ip address of ports in Router 0 is Router 1 using the following command

```
Router > enable
Router # config t
Router (config) interface fastEthernet 1/
Router (config-if) # ip address 30.0.0.2
  255.0.0.0
Router (config-if) # no shut
Router (config-if) # exit
Router (config) # interface serial 2/0
Router (config-if) # ip address 30.0.0.1 255.0.0
Router (config-if) # exit
Router (config) # exit
```

These are the commands for Router 0. Similarly Router 1 and Router 2 need to be configured.

(5) As Router 0 and Router 1 are connected to only one side we perform default no routing using following CLI commands

```
For Router 0
Router > enable
Router # config
Router (config) # ip route 0.0.0.0 30.0.0.2

For Router 1
Router # config t
```



Router (config) # ip route 0.0.0.0 0.0.0.0 30.0.0.2

Now, Do the static routing for the Router 2 using Commands.

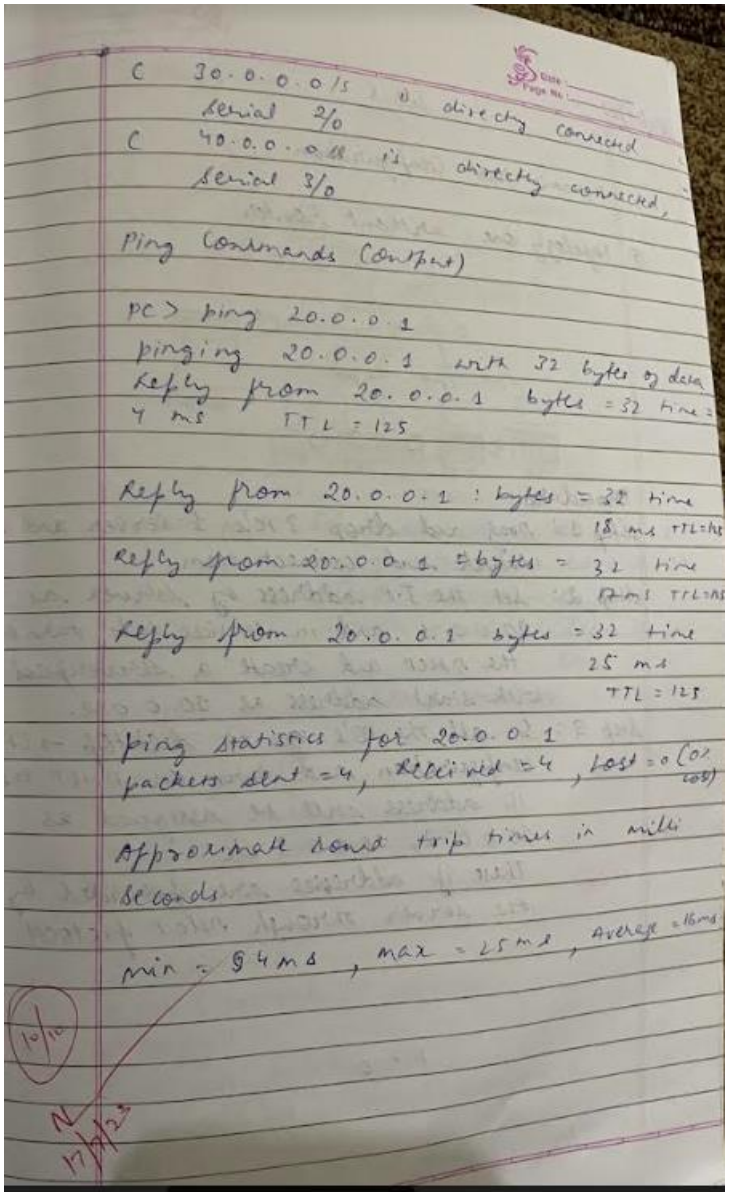
Router # config t
Router (config) # ip route 10.0.0.0 255.0.0.0 30.0.0.1

Router (config) # ip route 20.0.0.0 255.0.0.0 40.0.0.1

Router (config) # exit
Router #

(6) Now, Check the routing information
For Router 0
Router # show ip route
C - Connected S - Static * - candidate default
Gateway of last resort is 30.0.0.2 to network 0.0.0.0
C - 20.0.0.0/8 is directly connected, FastEthernet
C - 30.0.0.0/8 is directly connected, Serial
S* 0.0.0.0/0 [1/0] via 30.0.0.1

Router 2 -
Router # show ip route
C - connected S - static
S 10.0.0.0/8 [1/0] via 30.0.0.1
S 20.0.0.0/8 [1/0] via 40.0.0.1


C 30.0.0.0/8 is directly connected,
Serial 2/0
C 40.0.0.0/8 is directly connected,
Serial 3/0

Ping Commands (Output)

PC> ping 20.0.0.1

pinging 20.0.0.1 with 32 bytes of data
Reply from 20.0.0.1 bytes = 32 time = 4 ms TTL = 125

Reply from 20.0.0.1 : bytes = 32 time 18 ms TTL=

Reply from 20.0.0.1 : bytes = 32 time

Reply from 20.0.0.1 bytes = 32 time 25 ms TTL = 125

ping statistics for 20.0.0.1
packets sent = 4, Received = 4, Lost = 0 (0% loss)

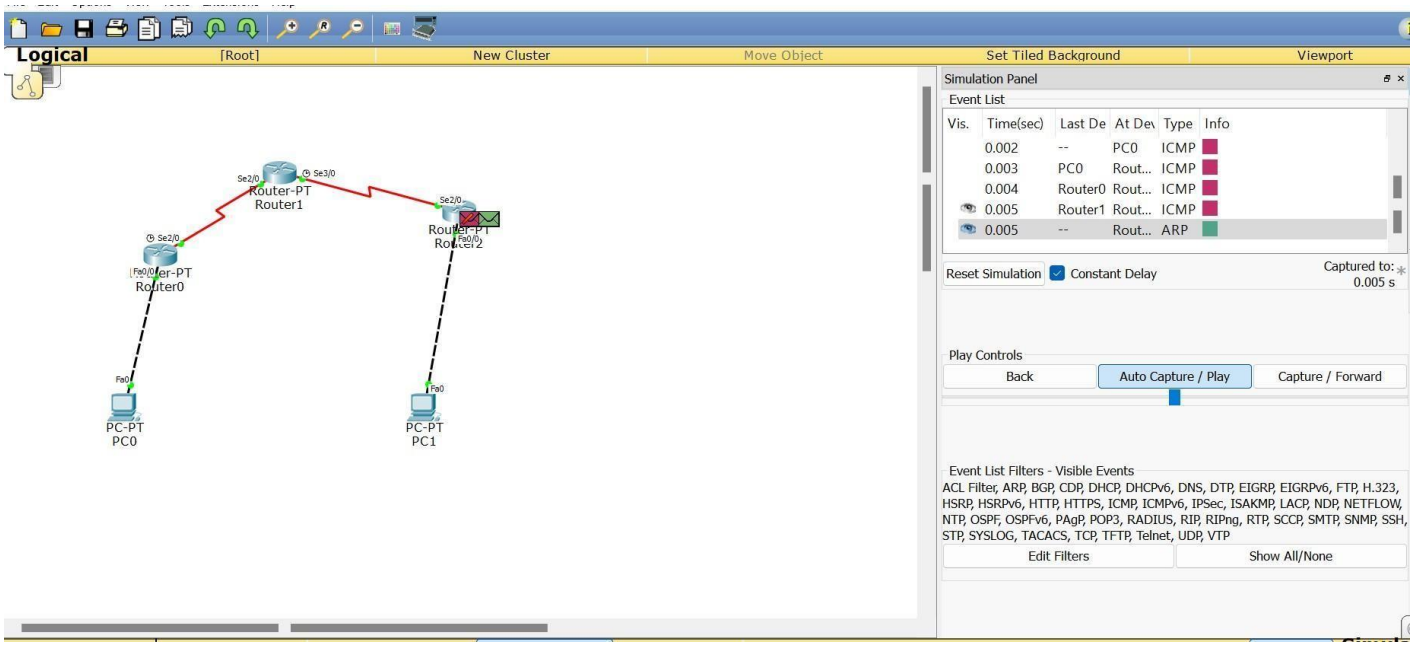
Approximate round trip times in milli seconds

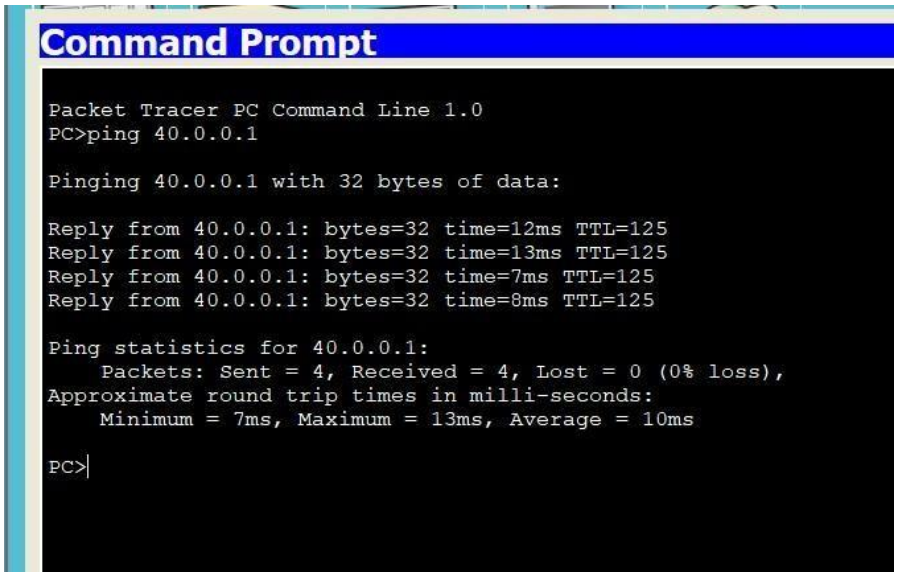
min = 94ms, max = 25ms, Average = 16ms

10/10

17/11/23

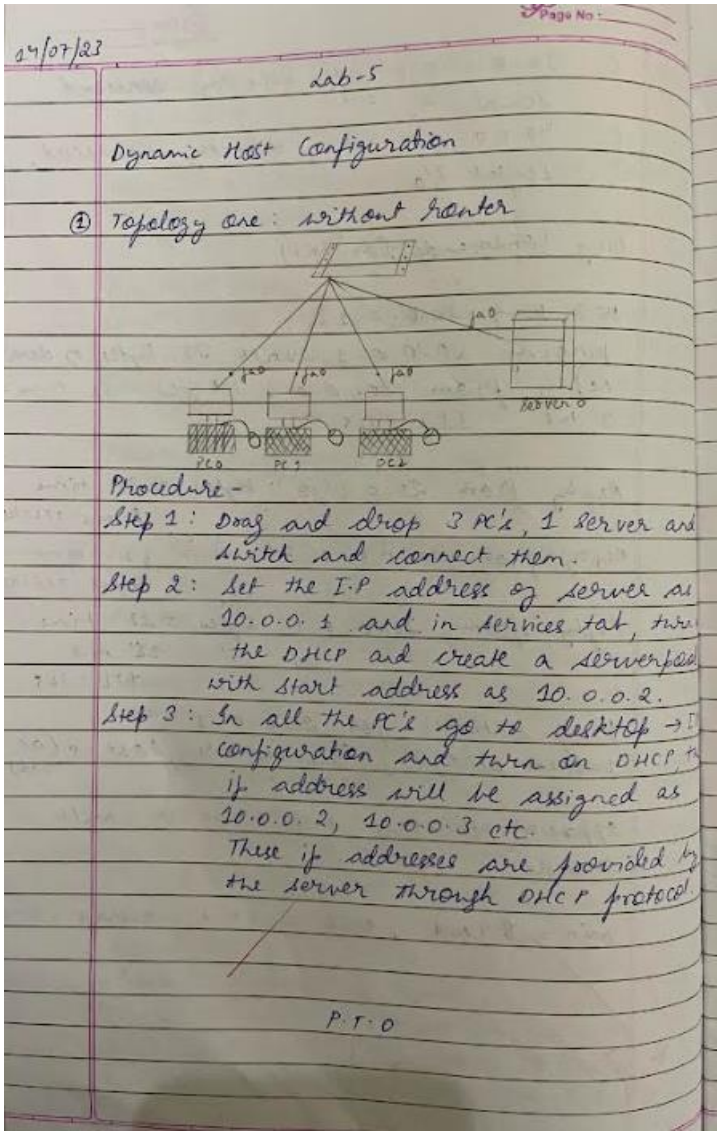
TOPOLOGY & OUTPUT


Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
Router-PT
Router1
Router-PT
Router0
Router-PT
Router2
PC-PT
PC0
PC-PT
PC1
Simulation Panel
Event List
Vis. Time(sec) Last De At De\ Type Info
0.002 -- PC0 ICMP
0.003 PC0 Rout... ICMP
0.004 Router0 Rout... ICMP
0.005 Router1 Rout... ICMP
0.005 -- Rout... ARP
Reset Simulation
Constant Delay
Captured to: 0.005 s
Play Controls
Back
Auto Capture / Play
Capture / Forward
Event List Filters - Visible Events
ACL Filter, ARP, BGP, CDP, DHCP, DHCPv6, DNS, DTP, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, LACP, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, RADIUS, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, VTP
Edit Filters
Show All/None


Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.1
Pinging 40.0.0.1 with 32 bytes of data:
Reply from 40.0.0.1: bytes=32 time=12ms TTL=125
Reply from 40.0.0.1: bytes=32 time=13ms TTL=125
Reply from 40.0.0.1: bytes=32 time=7ms TTL=125
Reply from 40.0.0.1: bytes=32 time=8ms 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 = 7ms, Maximum = 13ms, Average = 10ms
PC>

EXPERIMENT-4

Q) Configure DHCP within a LAN and outside LAN



24/07/23

Lab-5

Dynamic Host Configuration

① Topology one: without router

Procedure -

Step 1: Drag and drop 3 PC's, 1 server and switch and connect them.

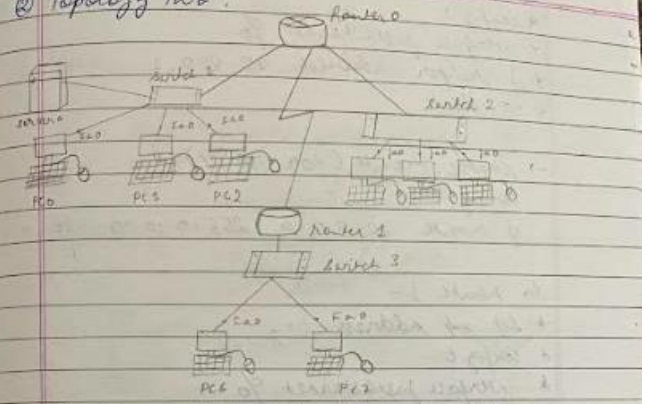
Step 2: Set the I.P address of server as 10.0.0.1 and in services tab, turn the DHCP and create a serverpool with start address as 10.0.0.2.

Step 3: In all the PC's go to desktop → IP configuration and turn on DHCP. If address will be assigned as 10.0.0.2, 10.0.0.3 etc.

These if addresses are provided by the server through DHCP protocol.

P.T.O

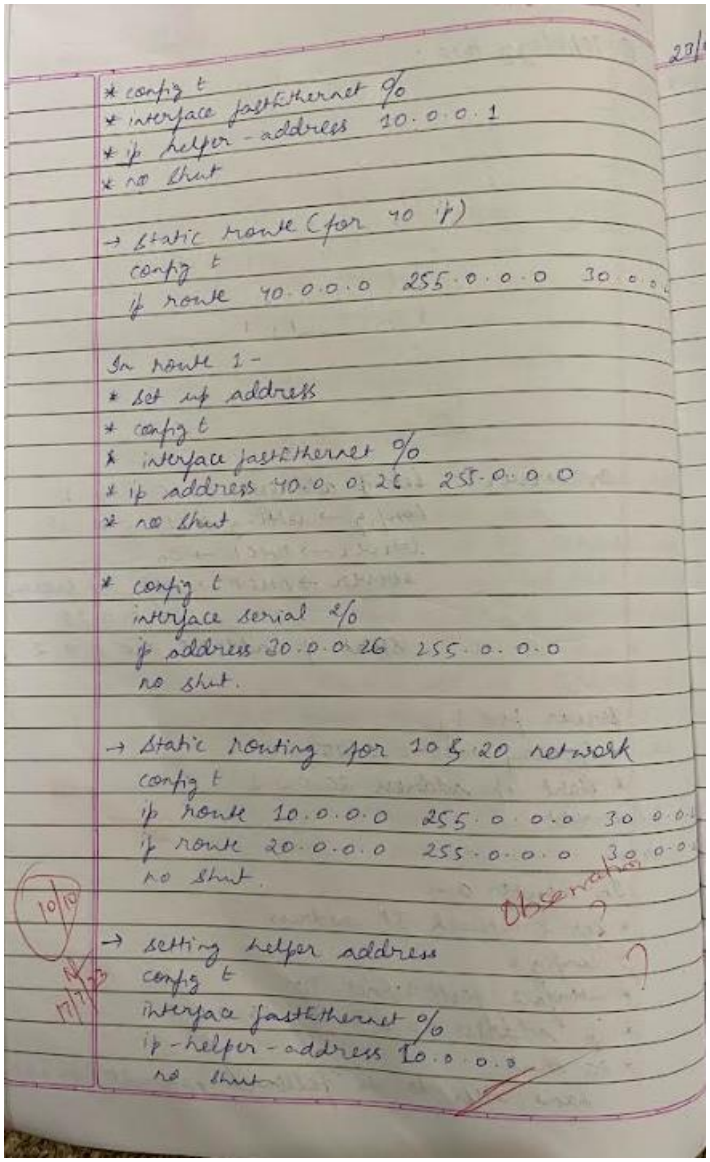
② Topology two :

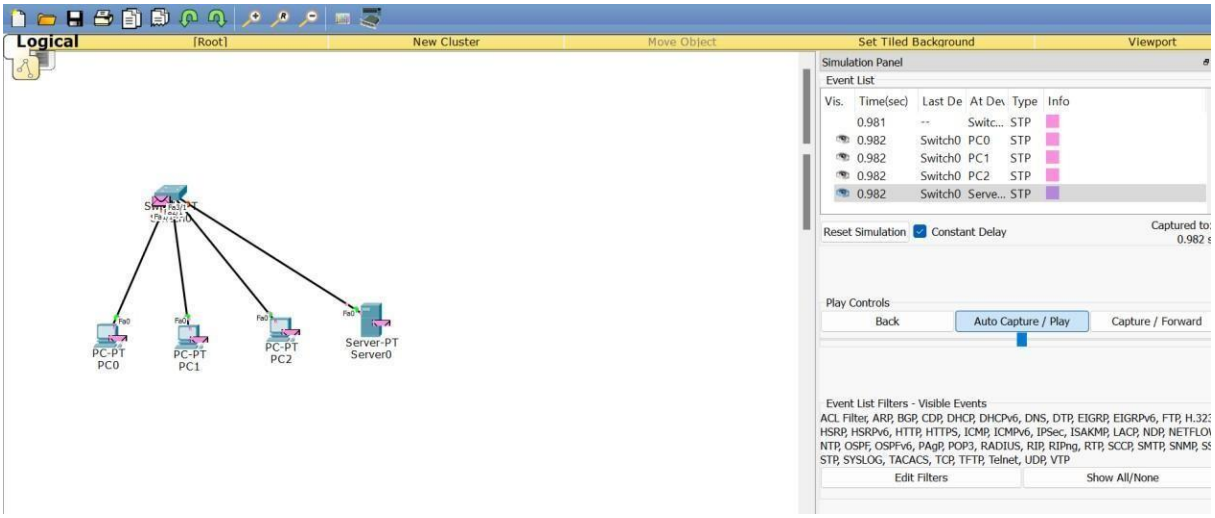


In server :- Set ip address as 10.0.0.1
Config → Setting → 10.0.0.25
Server → DHCP → On
server → DHCP → default gateway 10.0.0.25
Start ip address 10.0.0.2

Server pod 1
* Gateway → 10.0.0.25
* Start ip address 20.0.0.2
* add.

In router 0 -
* set 2 network IP address
* config t.
* interface fastEthernet 1/0
* ip address 10.0.0.25 25.0.0.0
* no shut.
same steps to be followed for 20.0.0.15


* config t
* interface fastEthernet 0/0
* ip helper-address 10.0.0.1
* no shut
→ static route (for 40 ip)
config t
ip route 40.0.0.0 255.0.0.0 30.0.0.
In router 1-
* set up address
* config t
* interface fastEthernet 0/0
* ip address 40.0.0.26 255.0.0.0
* no shut
* config t
interface serial 2/0
ip address 30.0.0.26 255.0.0.0
no shut.
→ static routing for 10 & 20 network
config t
ip route 10.0.0.0 255.0.0.0 30.0.0.
ip route 20.0.0.0 255.0.0.0 30.0.0.
no shut.
→ setting helper address
config t
interface fastEthernet 0/0
ip-helper-address 10.0.0.3
no shut


Logical
[Root]
New Cluster
Move Object
Set Tiled Background
Viewport
PC-PT PC0
PC-PT PC1
PC-PT PC2
Server-PT Server0
Simulation Panel
Event List
Vis. Time(sec) Last De At De\ Type Info
0.981 -- Switc... STP
0.982 Switch0 PC0 STP
0.982 Switch0 PC1 STP
0.982 Switch0 PC2 STP
0.982 Switch0 Serve... STP
Reset Simulation
Constant Delay
Play Controls
Back
Auto Capture / Play
Capture / Forward
Event List Filters - Visible Events
Edit Filters
Show All/None

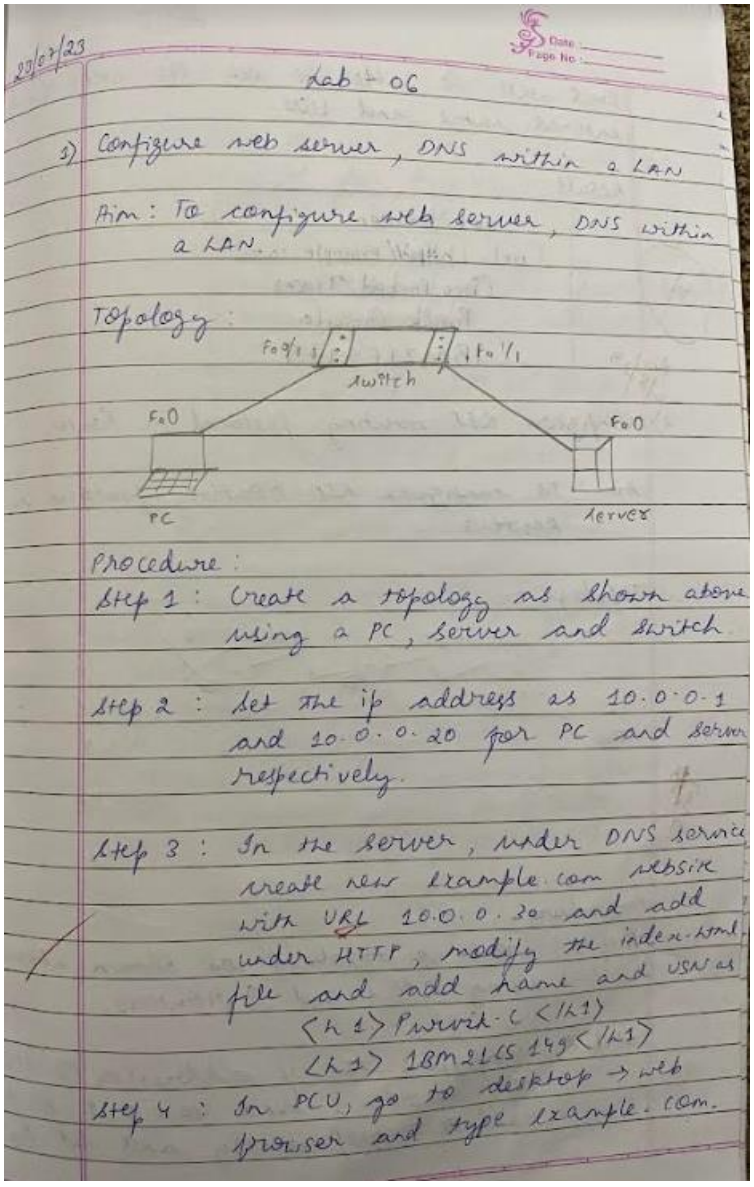
```
PC>ping 10.0.0.4

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>
```

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



20/07/23

Lab - 06

1) Configure web server, DNS within a LAN

Aim: To configure web server, DNS within a LAN.

Topology:

Procedure:

Step 1: Create a topology as shown above using a PC, server and switch.

Step 2: Set the ip address as 10.0.0.1 and 10.0.0.20 for PC and server respectively.

Step 3: In the server, under DNS service create new example.com website with URL 10.0.0.20 and add under HTTP, modify the index.html file and add name and USN as
<h1> Pwrwik-(</h1>
<h1> 1BM21CS149 </h1>

Step 4: In PC0, go to desktop → web browser and type example.com.

You will be able to see the website entered name and USN.

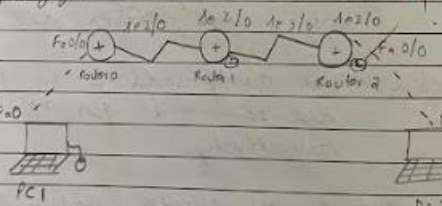
Result

web browser
url https://example.com/
Cisco Packed Tracer
Purvik Gowda
1BM21CS149

2) Configure RIP routing protocol in Router

Aim: To configure RIP routing protocol in Routers.

Topology:



Procedure:

Step 1: Create a topology as shown above using 2 PC's and 3 routers.

Step 2: Configure the IP addresses of PC's as 10.0.0.1 and 40.0.0.1 for PC1 and PC2 respectively and set the

gateways as 10.0.0.3 & 40.0.0.3

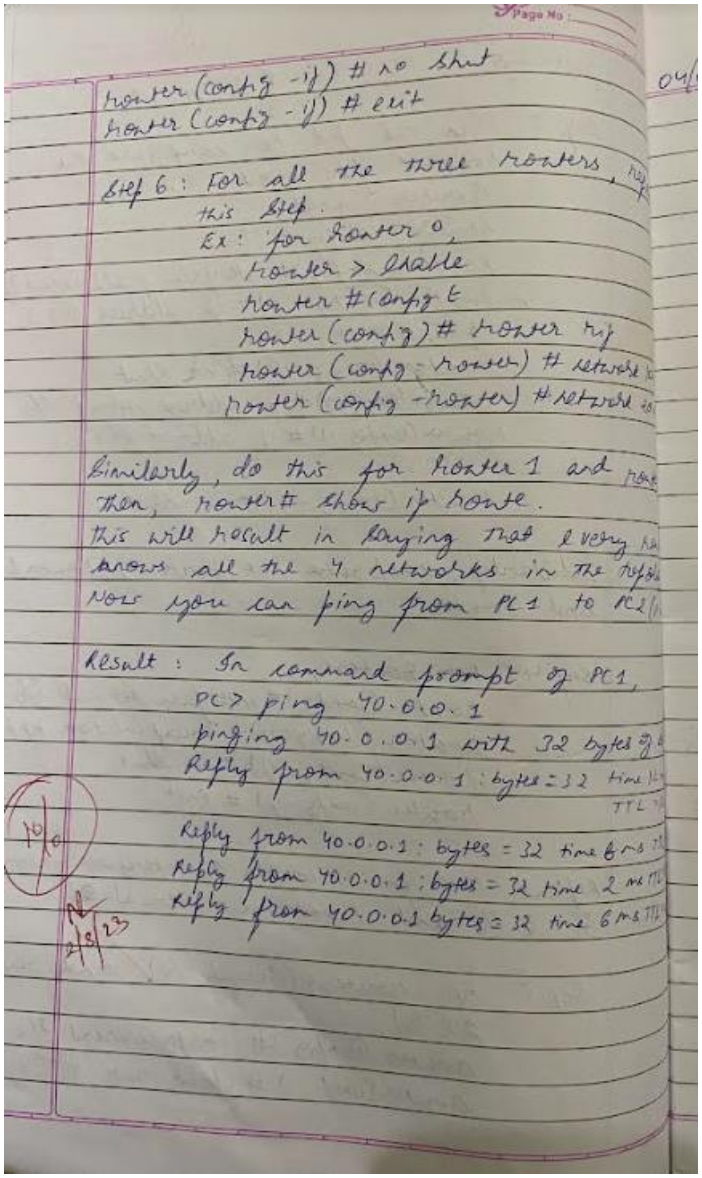
Step 3: Plan the ip's to configure the routers for router 0,
Router > enable
Router # config t
Router (config) # interface fastEthernet 0/0
router (config-if) # ip address 10.0.3
255.0.0.0
router (config-if) # no shut.
router (config) # ip address serial 2/0
router (config-if) # ip address 20.0.0.1
255.0.0.0
router (config-if) # no shut.

Similarly, configure the ports of router 1 and router 2.

Step 4: For router 0,
router (config) # interface serial 2/0
router (config-if) # encapsulation ppp
router (config-if) # no shut.
router (config-if) # exit

Repeat this for router 1 interface -> serial 2/0 & 3/0 and router 2 → serial 2/0

Step 5: For router 0 (serial 2/0) and router 1 (3/0),
router (config) # interface serial 2/0
router (config) # clock rate 64000



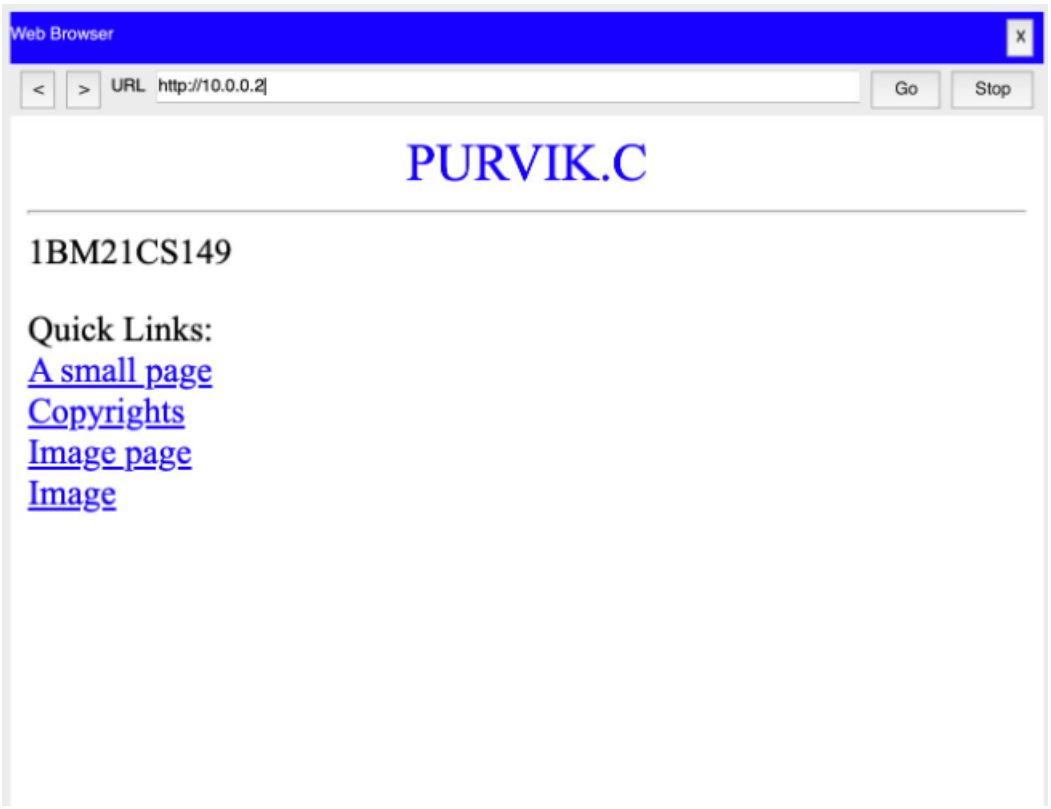
Router (config-if) # no shut
Router (config-if) # exit

Step 6: For all the three routers, [illegible] this step.
Ex: for Router 0,
Router > enable
Router # config t
Router (config) # router rip
Router (config-router) # network [illegible]
Router (config-router) # network [illegible]

Similarly, do this for Router 1 and [illegible] Then, router # show ip route.
This will result in saying that every [illegible] knows all the 4 networks in the [illegible]
Now you can ping from PC1 to PC2 [illegible]

Result: In command prompt of PC1,
PC> ping 40.0.0.1
pinging 40.0.0.1 with 32 bytes of [illegible]
Reply from 40.0.0.1 : bytes = 32 time [illegible] TTL [illegible]
Reply from 40.0.0.1 : bytes = 32 time 6 ms TTL [illegible]
Reply from 40.0.0.1 : bytes = 32 time 2 ms TTL [illegible]
Reply from 40.0.0.1 bytes = 32 time 6 ms TTL [illegible]

10/10
28/9/23


Web Browser
X
<
>
URL http://10.0.0.2
Go
Stop
PURVIK.C
1BM21CS149
Quick Links:
A small page
Copyrights
Image page
Image

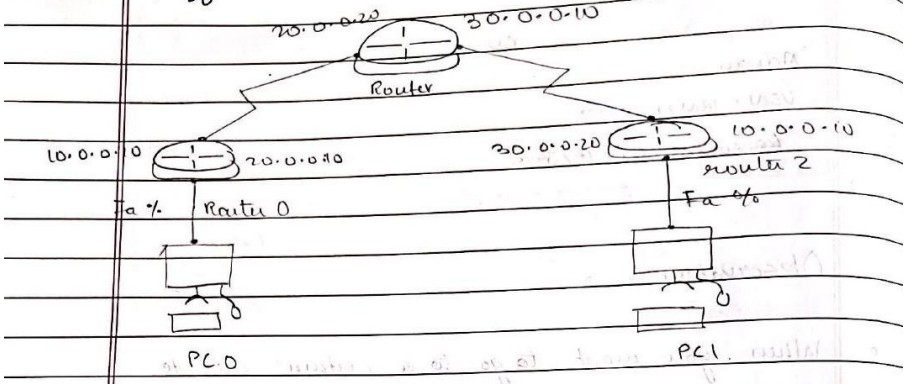
EXPERIMENT-6

Q) Configure RIP routing Protocol in Routers

Program - 6

Aim: Configure RIP routing protocol in router.

Topology:



Procedure

- Create a network using 3- routers & 2PC's
- 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 - dock rate 640
  - Step 12 — No shut.

- 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)

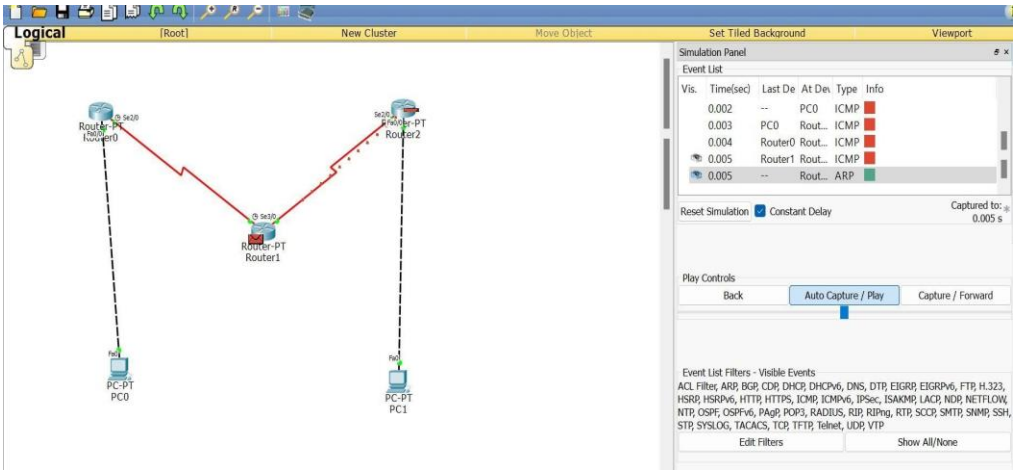
Approx round trip in ms:

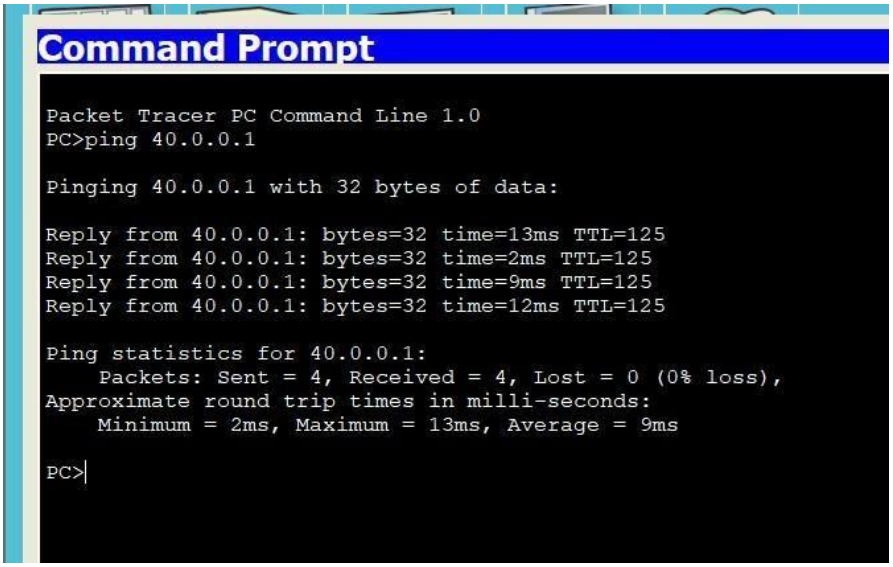
min = 5ms , max = 10ms, Avg = 7ms.

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.

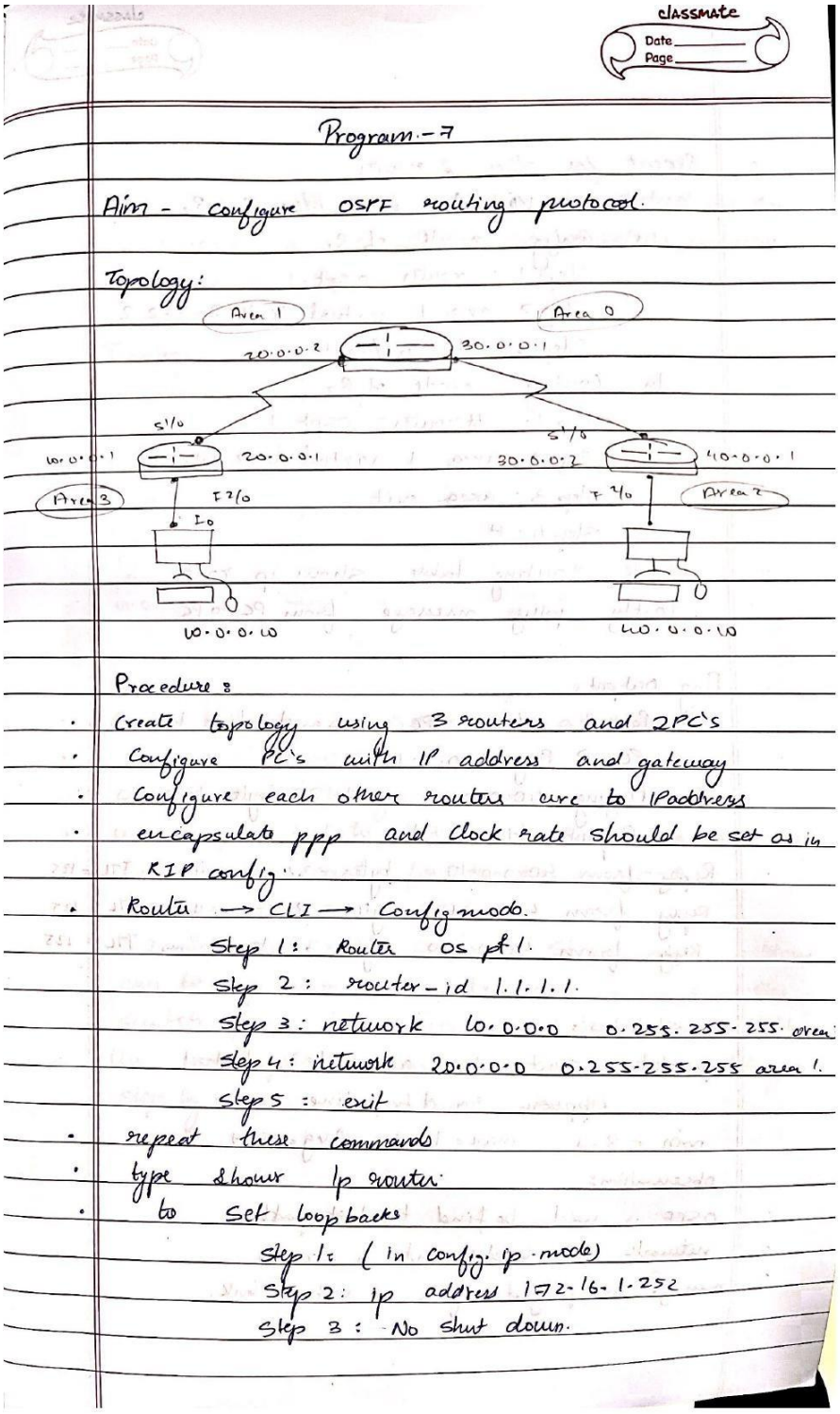
TOPOLOGY & OUTPUT





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 &how 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

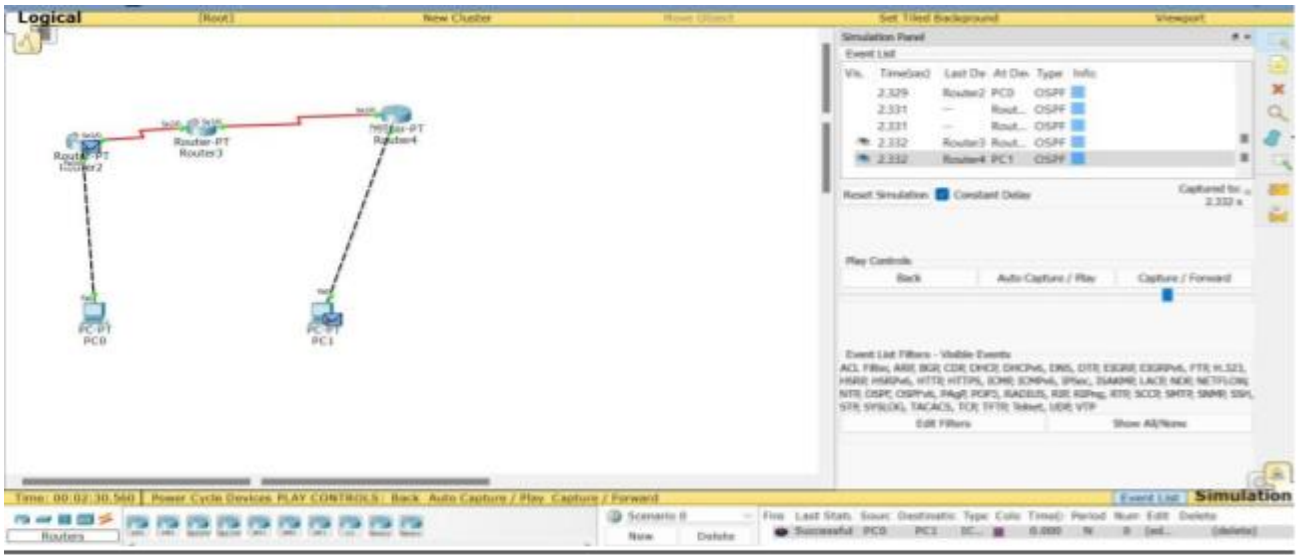
Approx round trip time

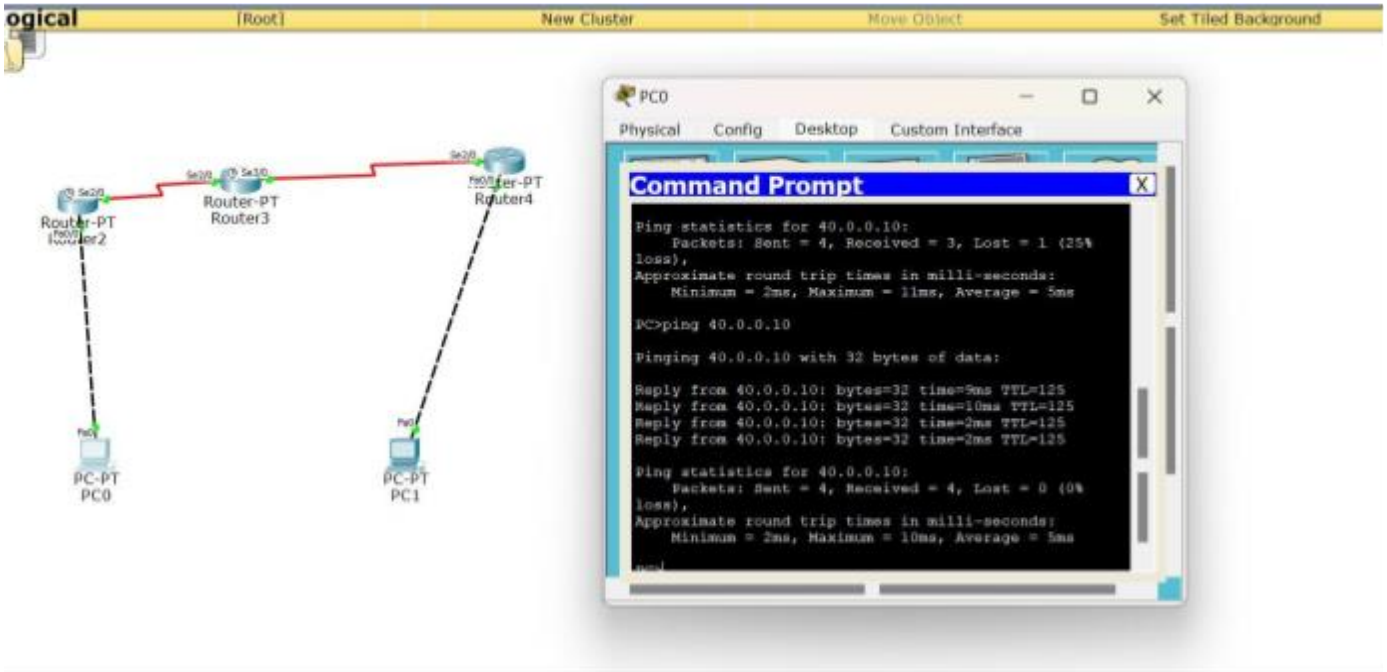
min = 8ms, max = 11ms, Avg = 10ms

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

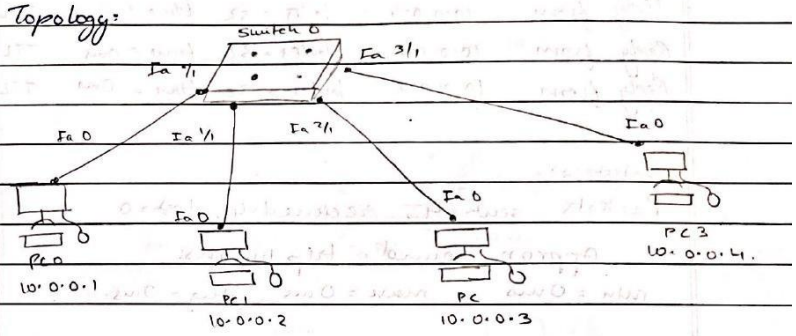
Q) To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)

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

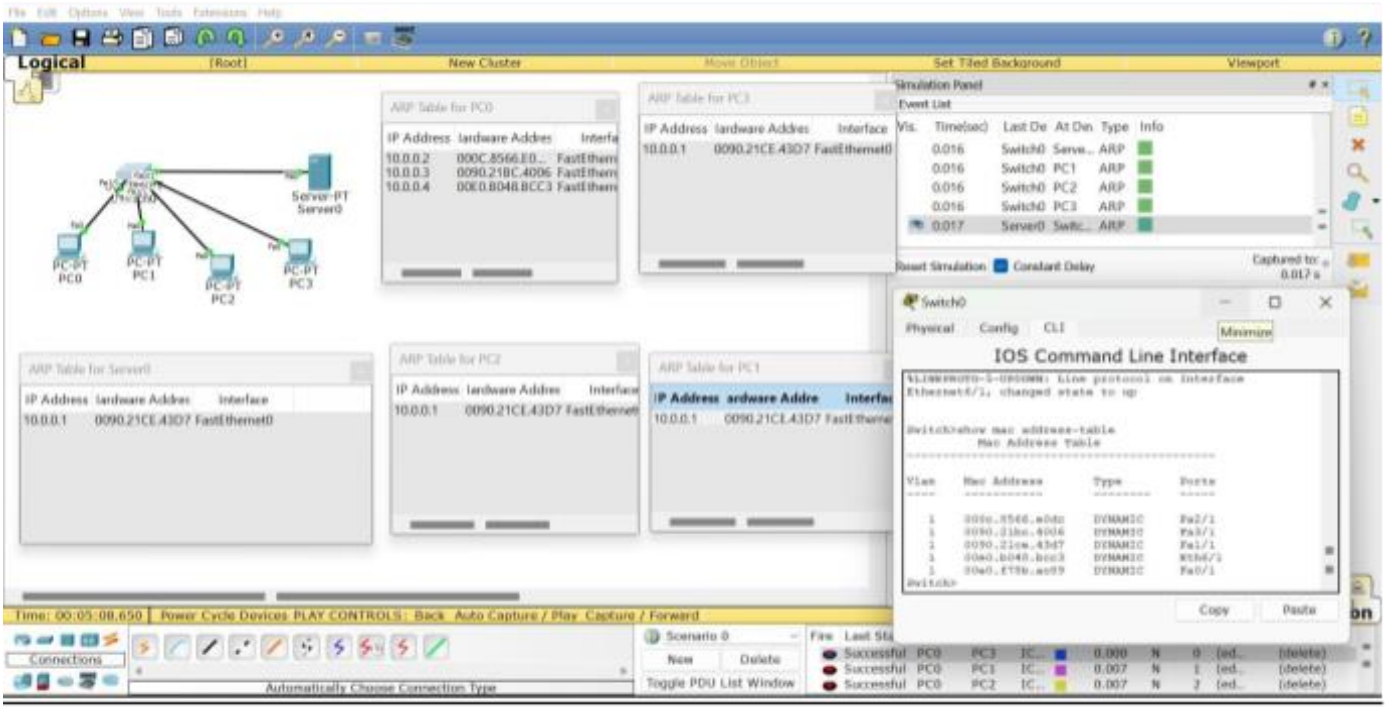
Approx round trip in ms:

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



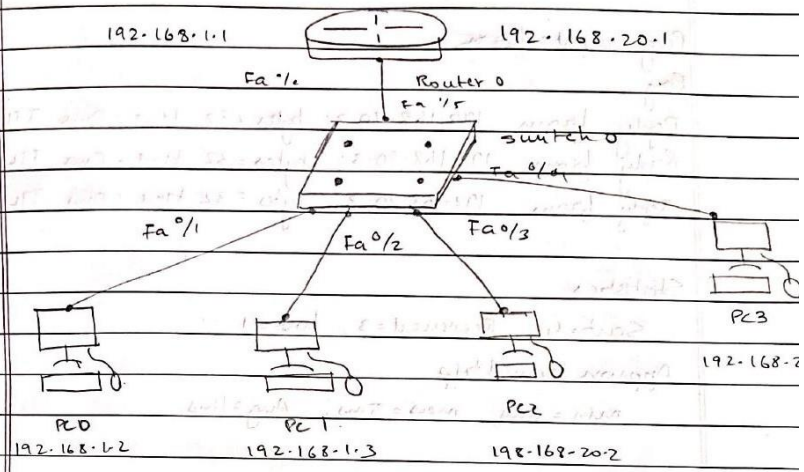
EXPERIMENT-9

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 4 PC's
- in Switch, go to config tab and select VLAN database. Give any VLAN no-like 2.
- Select interface fast ethernet 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

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

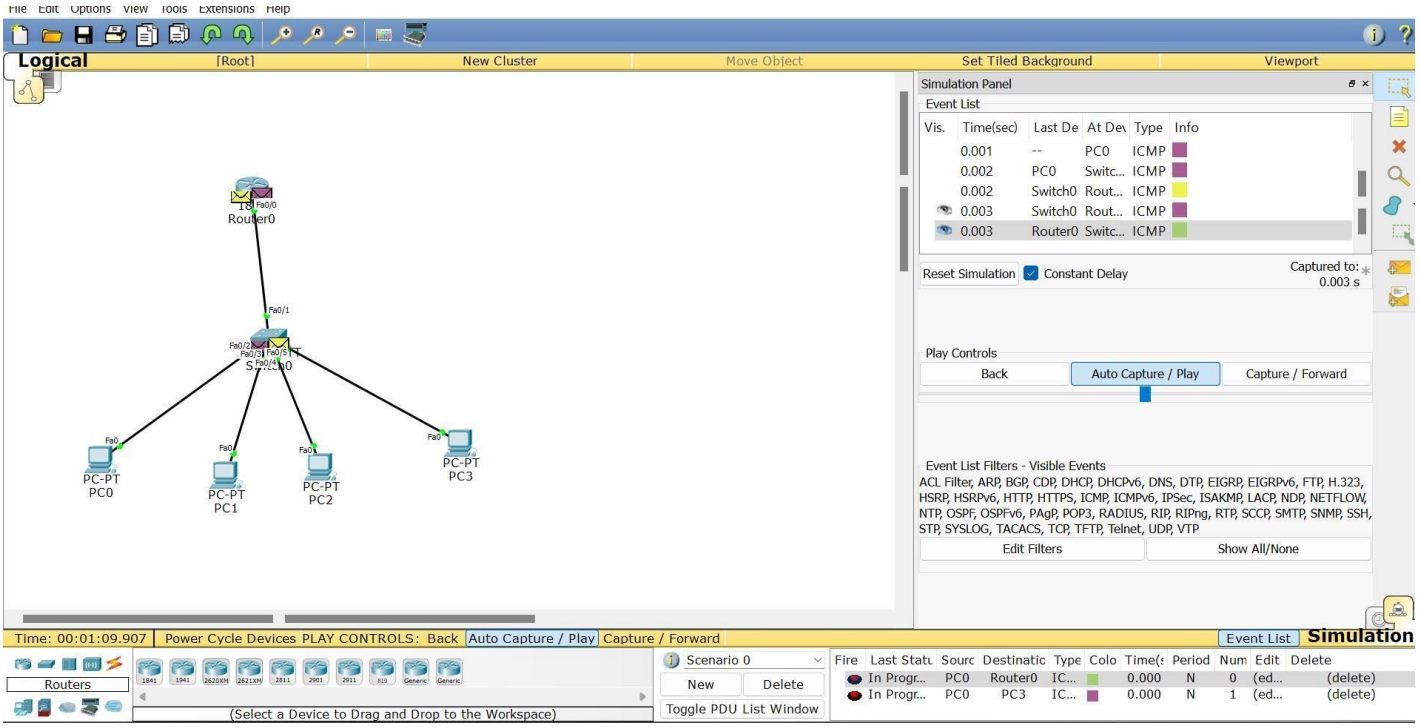
Approx round trip

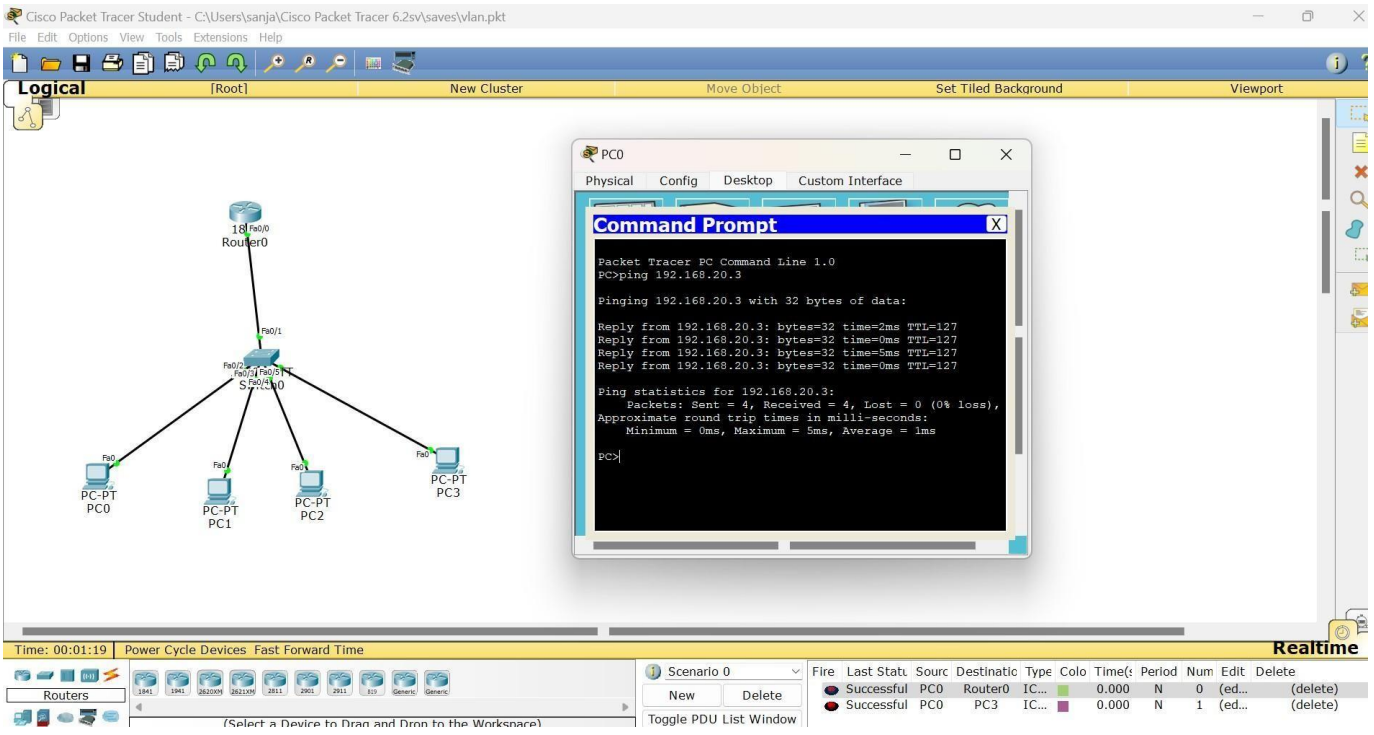
min=0ms, max=5ms, Avg=1ms.

Observation

VLAN & other VLAN connected to same switch.
VLAN's don't use IP address
Inter-VLAN routing gives as flexible tool to logically subdivide their networks.

TOPOLOGY & OUTPUT





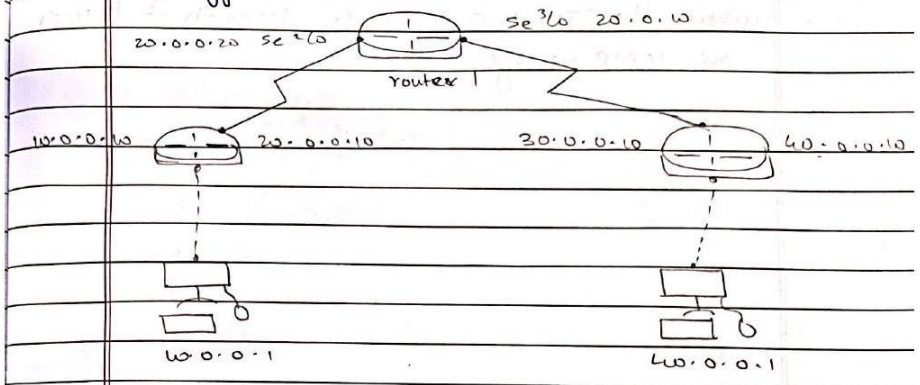
EXPERIMENT-10

Q) Demonstrate the TTL/ Life of a Packet

Program 10.

Aim: Demonstrate the TTL / life of a packet

Topology:

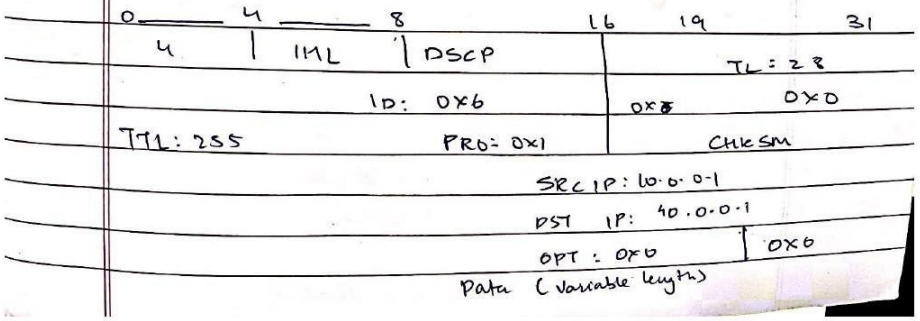


Procedure:

- 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

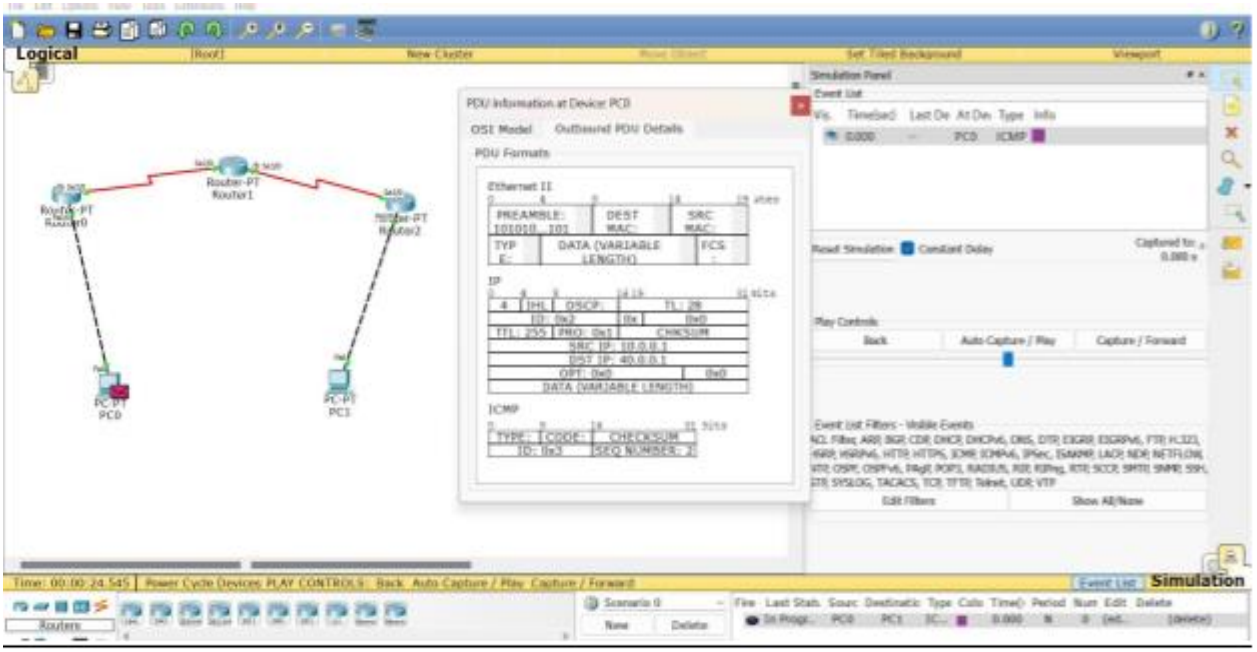


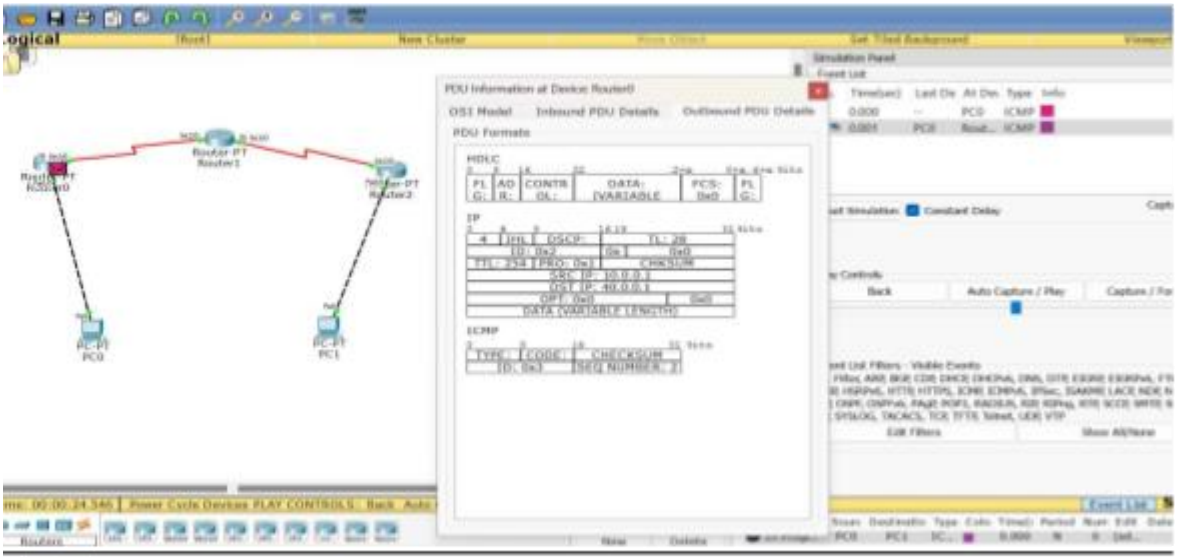
Scanned with CamScanner

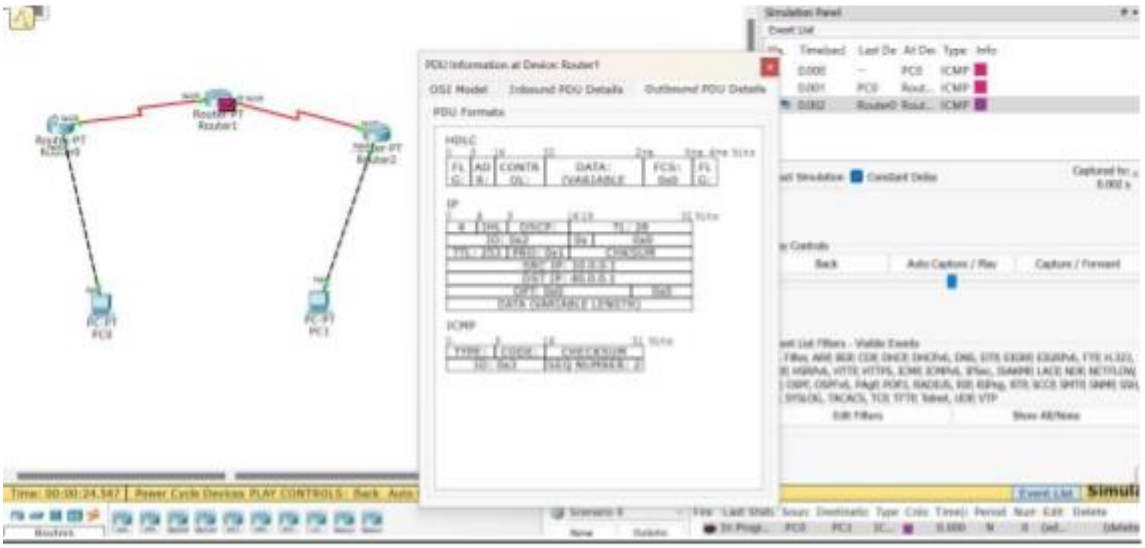
Observation:

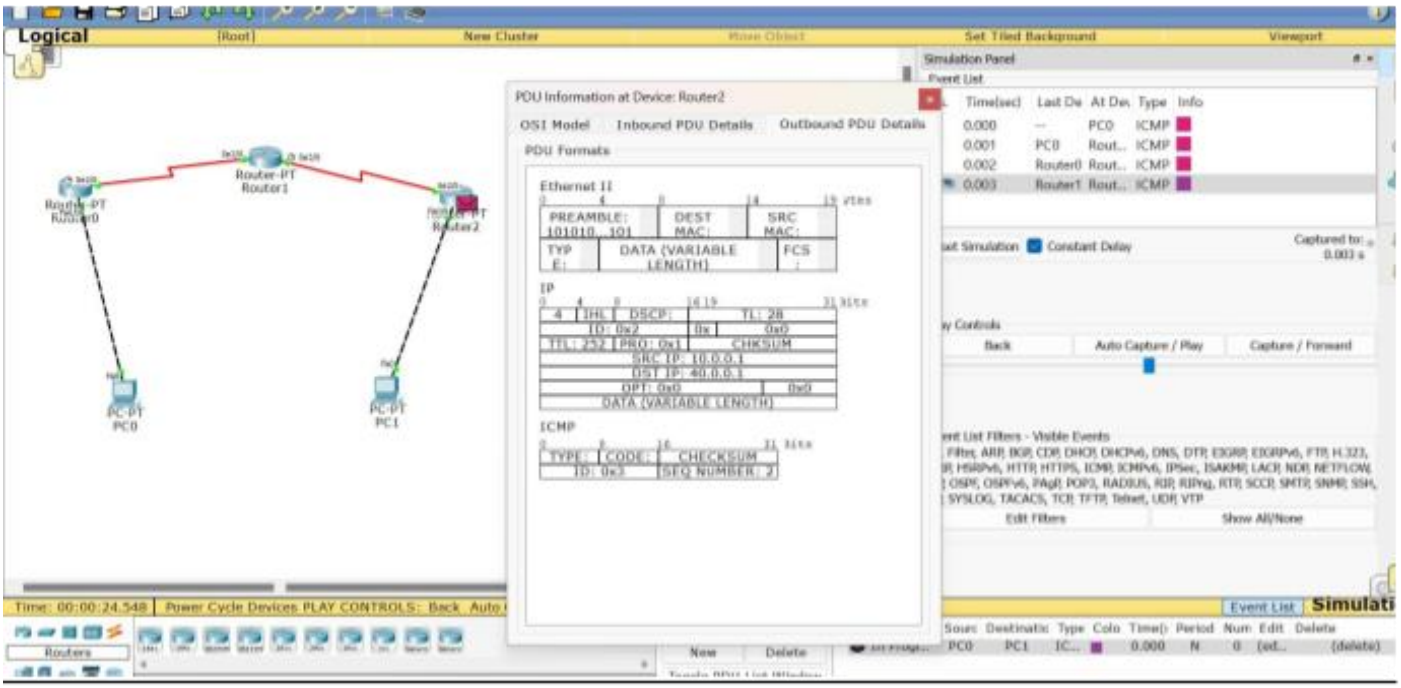
- The no of hops the packets travel before being discarded as TTL
- Router reduce TTL values by one until forwards packets
- when the TTL is 0, router discards it & s an ICMP message.

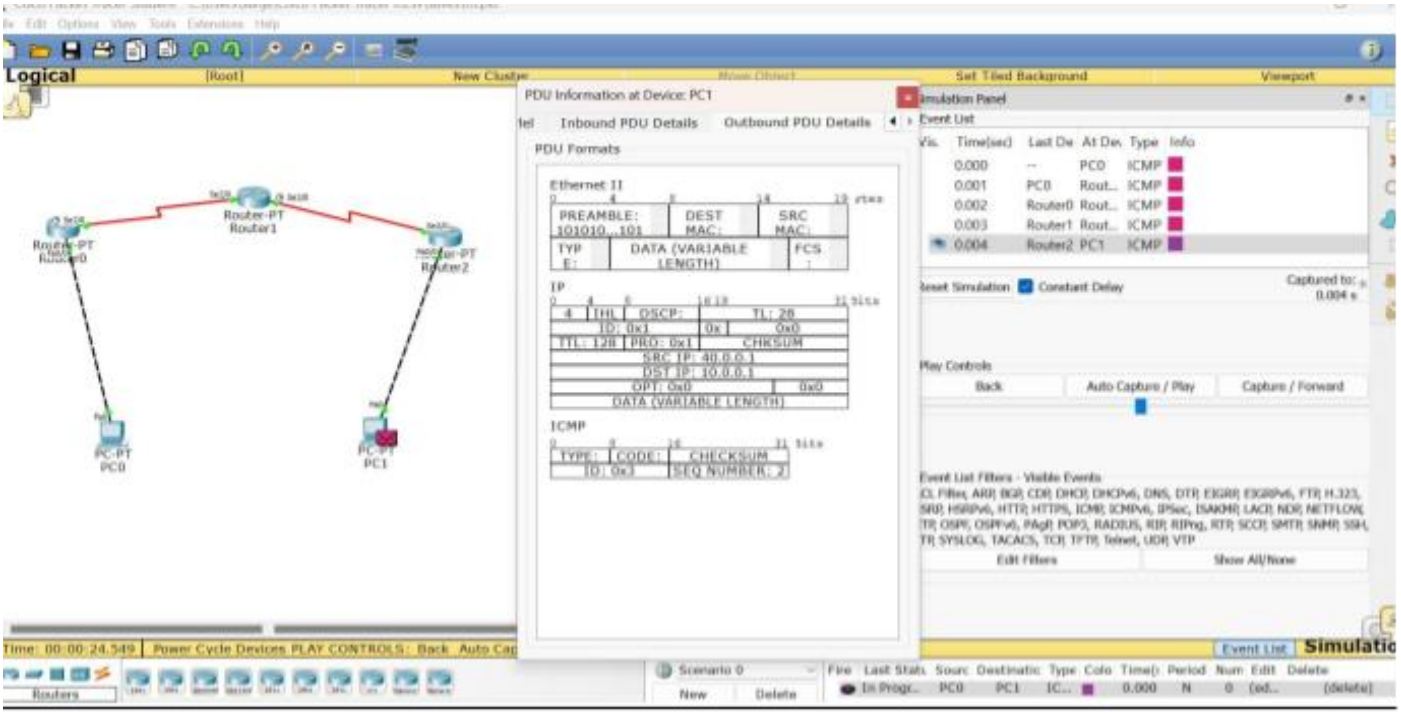
TOPOLOGY & OUTPUT








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
TTL: 28
SRC IP: 10.0.0.1
DST IP: 40.0.0.1
OPT: 0x0
DATA (VARIABLE LENGTH)
ICMP
TYPE:
CODE:
CHECKSUM
ID: 0x3
SEQ NUMBER: 2
Simulation Panel
Event List
Time(sec)
Last De
At De
Type
Info
0.000
PC0
ICMP
0.001
PC0
Rout...
ICMP
0.002
Router0
Rout...
ICMP
0.003
Router1
Rout...
ICMP
Captured to:
0.003 s
Back
Auto Capture / Play
Capture / Forward
Edit Filters
Show All/None
Event List
Simulati
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]
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:
TYP E:
DATA (VARIABLE LENGTH)
FCS
IP
TTL: 128
SRC IP: 40.0.0.1
DST IP: 10.0.0.1
OPT: 0x0
DATA (VARIABLE LENGTH)
ICMP
TYPE:
CODE:
CHECKSUM
ID: 0x3
SEQ NUMBER: 2
Simulation Panel
Event List
Time(sec)
Last De
At De
Type
Info
0.000
PC0
ICMP
0.001
PC0
Rout...
ICMP
0.002
Router0
Rout...
ICMP
0.003
Router1
Rout...
ICMP
0.004
Router2
PC1
ICMP
Captured to:
0.004 s
Back
Auto Capture / Play
Capture / Forward
Edit Filters
Show All/None
Event List
Simulatic
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
New
Delete

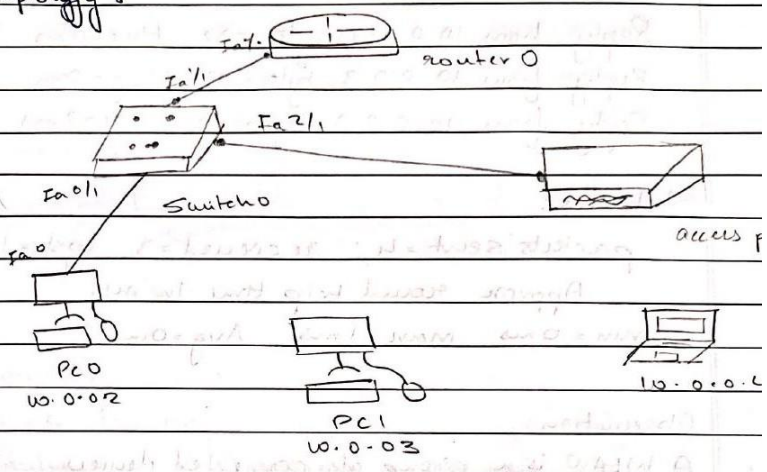
EXPERIMENT-11

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

Program-11

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

Topology:



Procedure.

- Construct topology
- Configure P0 & router 0 as normaly done
- Configure access point 1- port 1 → SSID Name-
- select LOEP & give any 10 digit hexa key - 01
- Configure PC1 & 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 byte 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

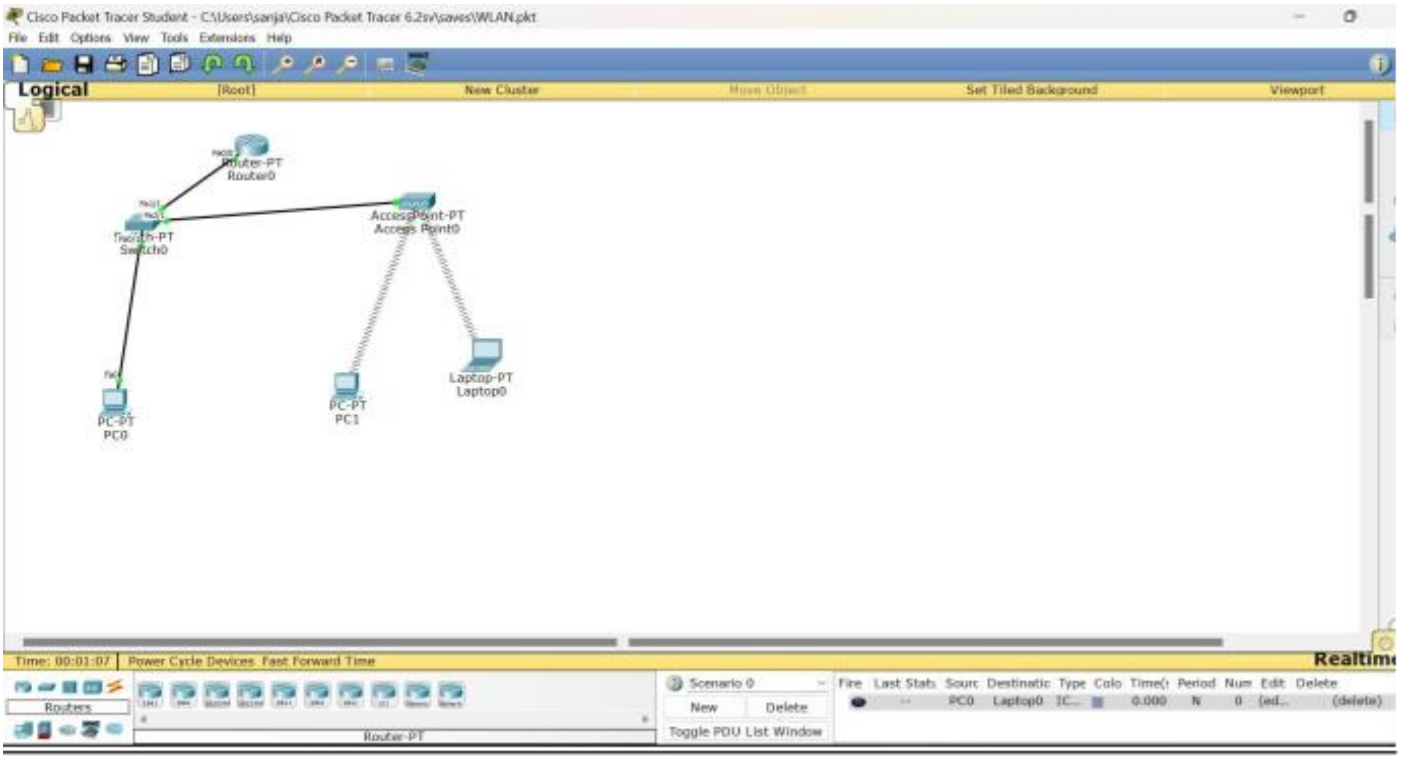
Approx round trip time in ms:

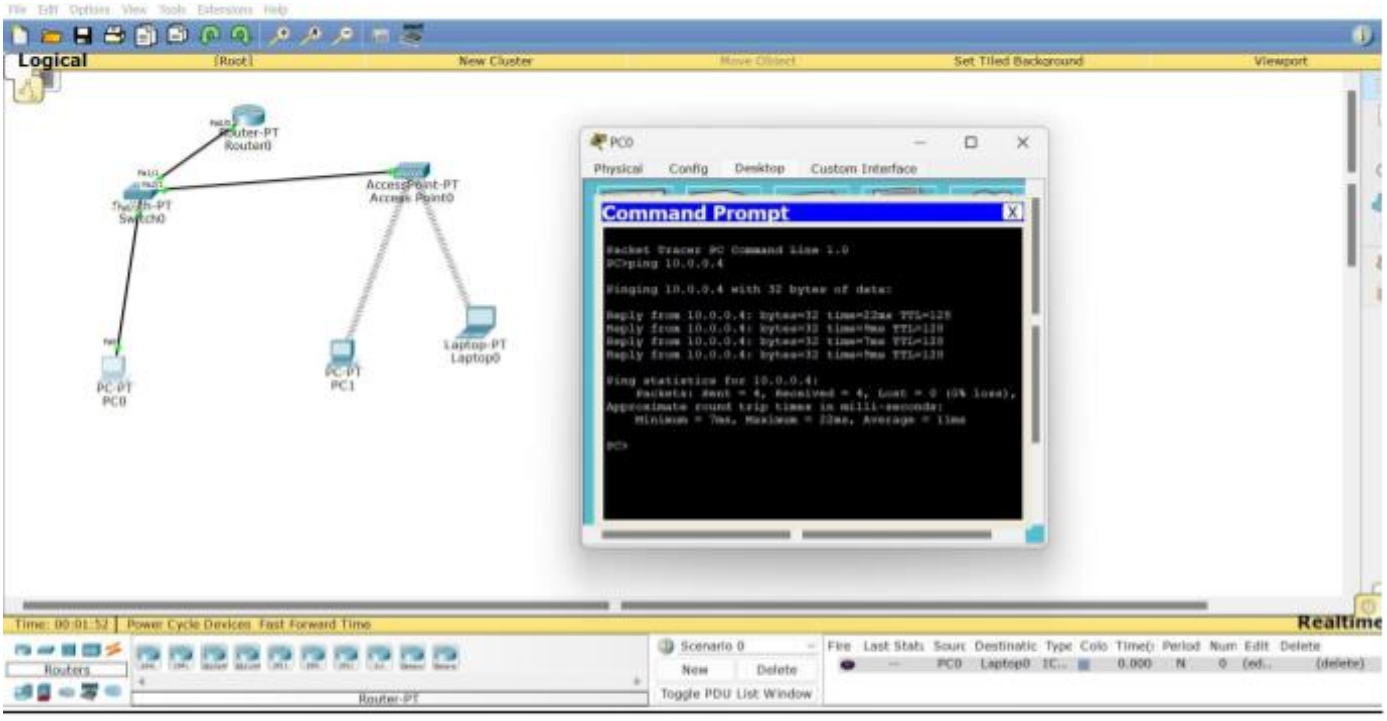
min = 0ms, max = 1ms, Avg = 0ms

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





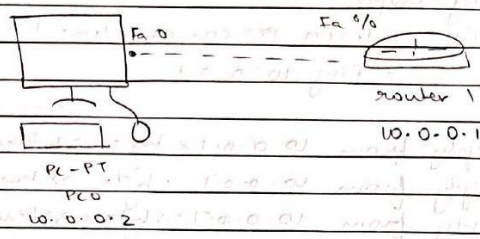
EXPERIMENT-12

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

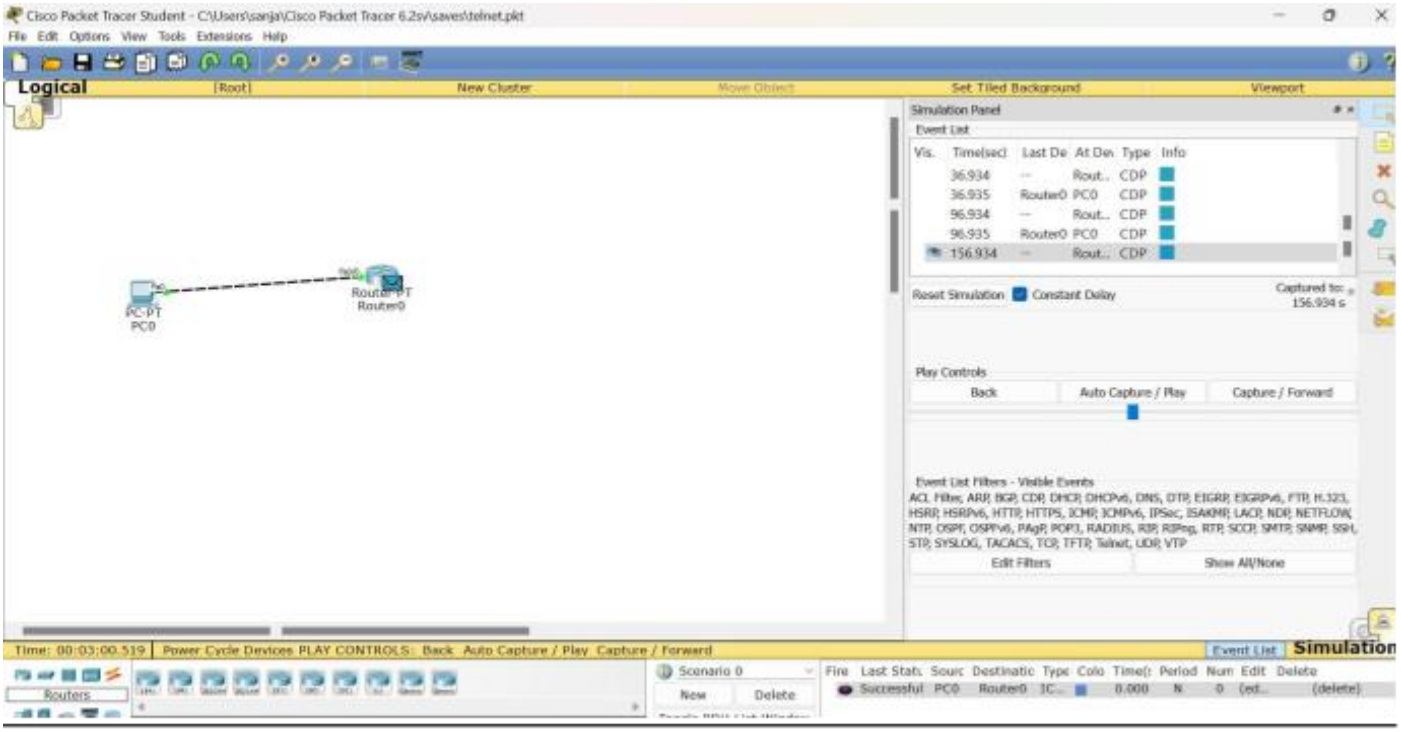
Ping Output
Packet tracer PC command line 1.0
PC> Ping 10.0.0.1

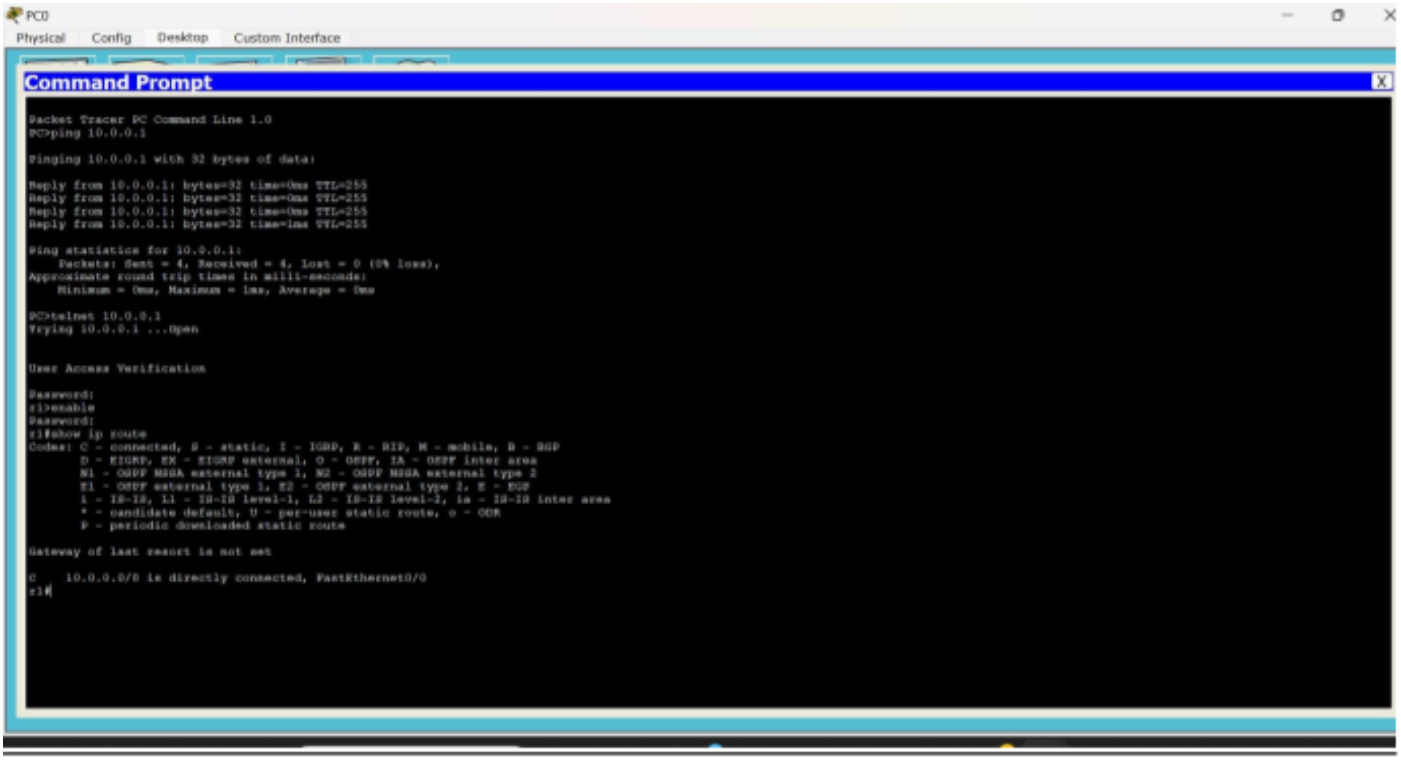
Reply from 10.0.0.1 : byte = 32 time = 0ms TTL = 255
Reply from 10.0.0.1 : byte = 32 time = 0 ms TTL = 255
Reply from 10.0.0.1 : byte = 32 time = 0ms TTL = 255
Reply from 10.0.0.1 : byte = 32 time = 0ms TTL = 255

Statistics:
Packets sent = 4, received = 4, lost = 0
Approx 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

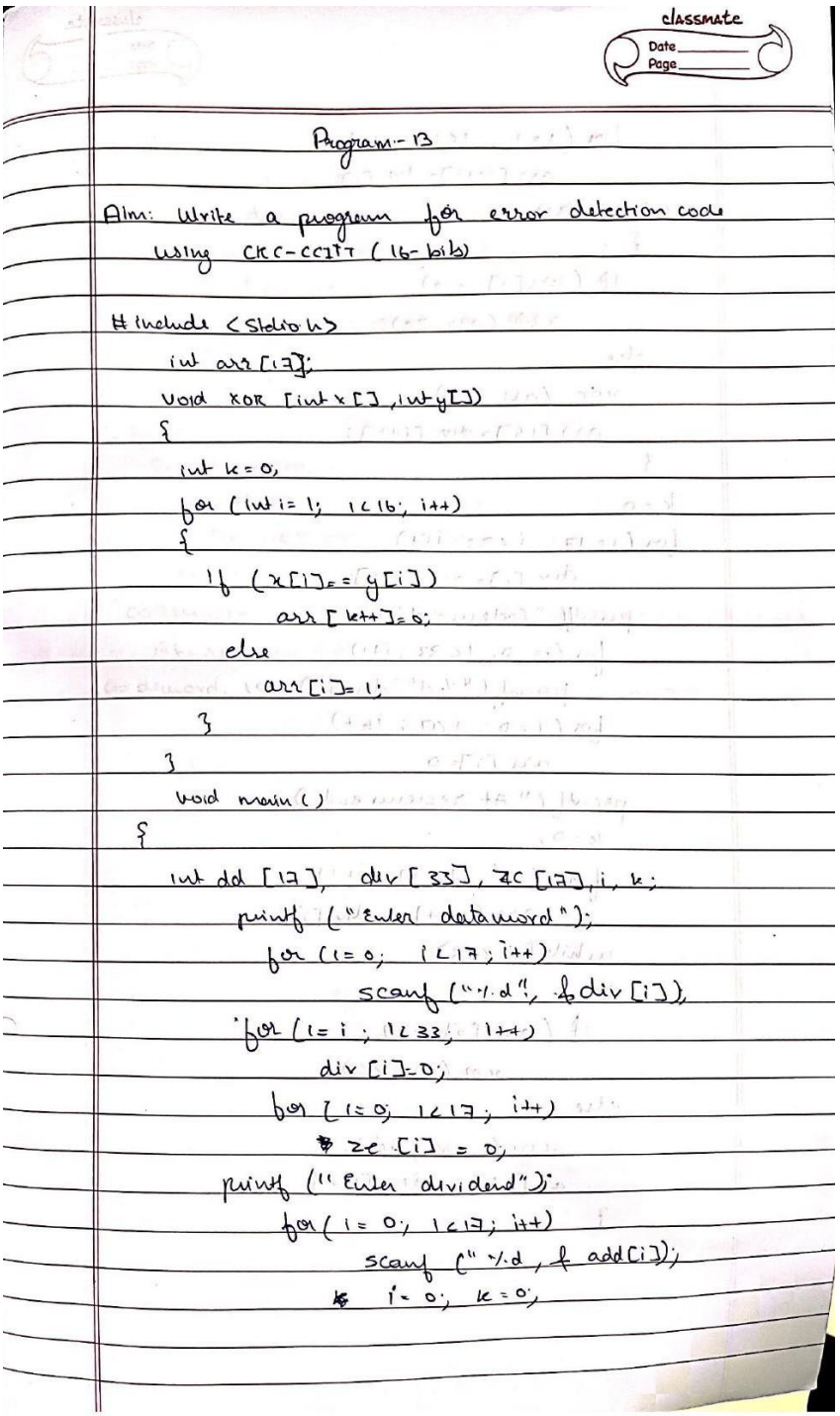
TOPOLOGY & OUTPUT





EXPERIMENT-13

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

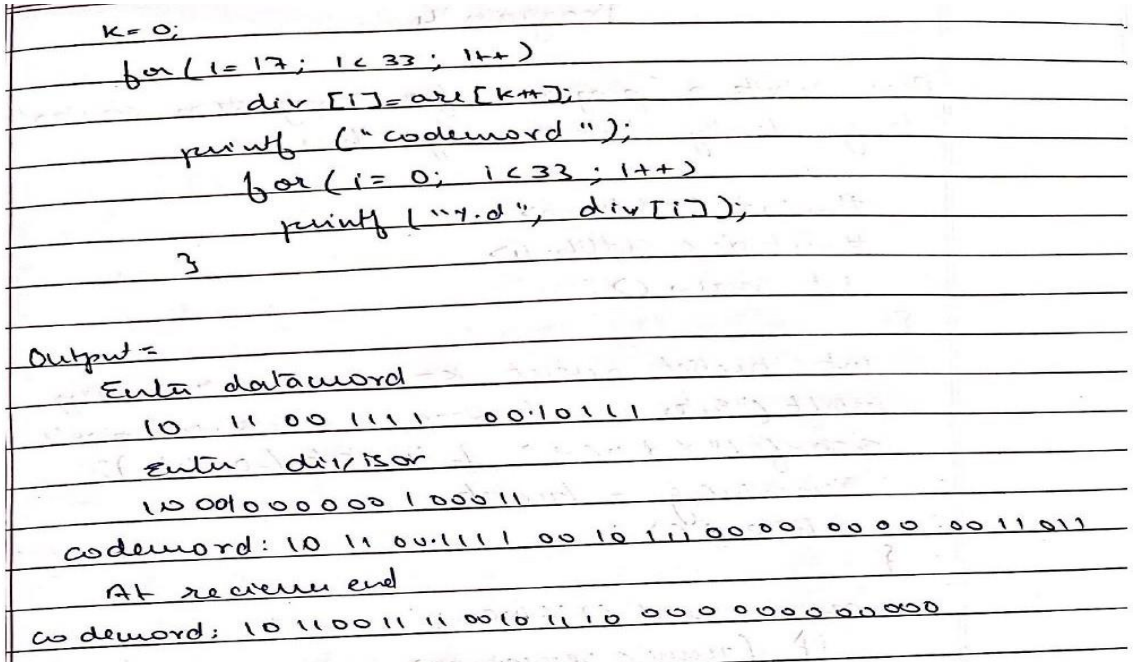


Program-13

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 dataword");
          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;
```

```
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 receiver 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++];
}
```



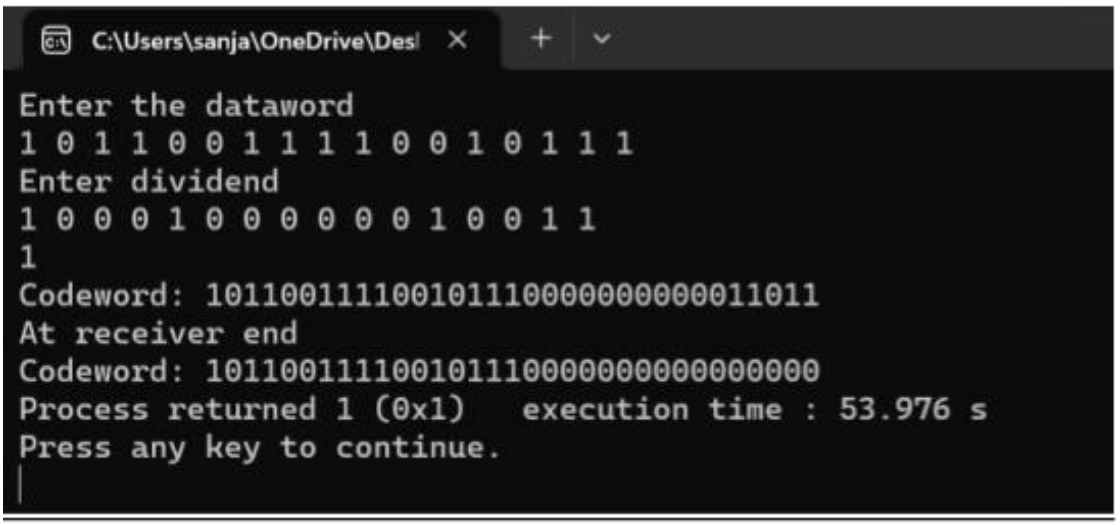
CODE-

```
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

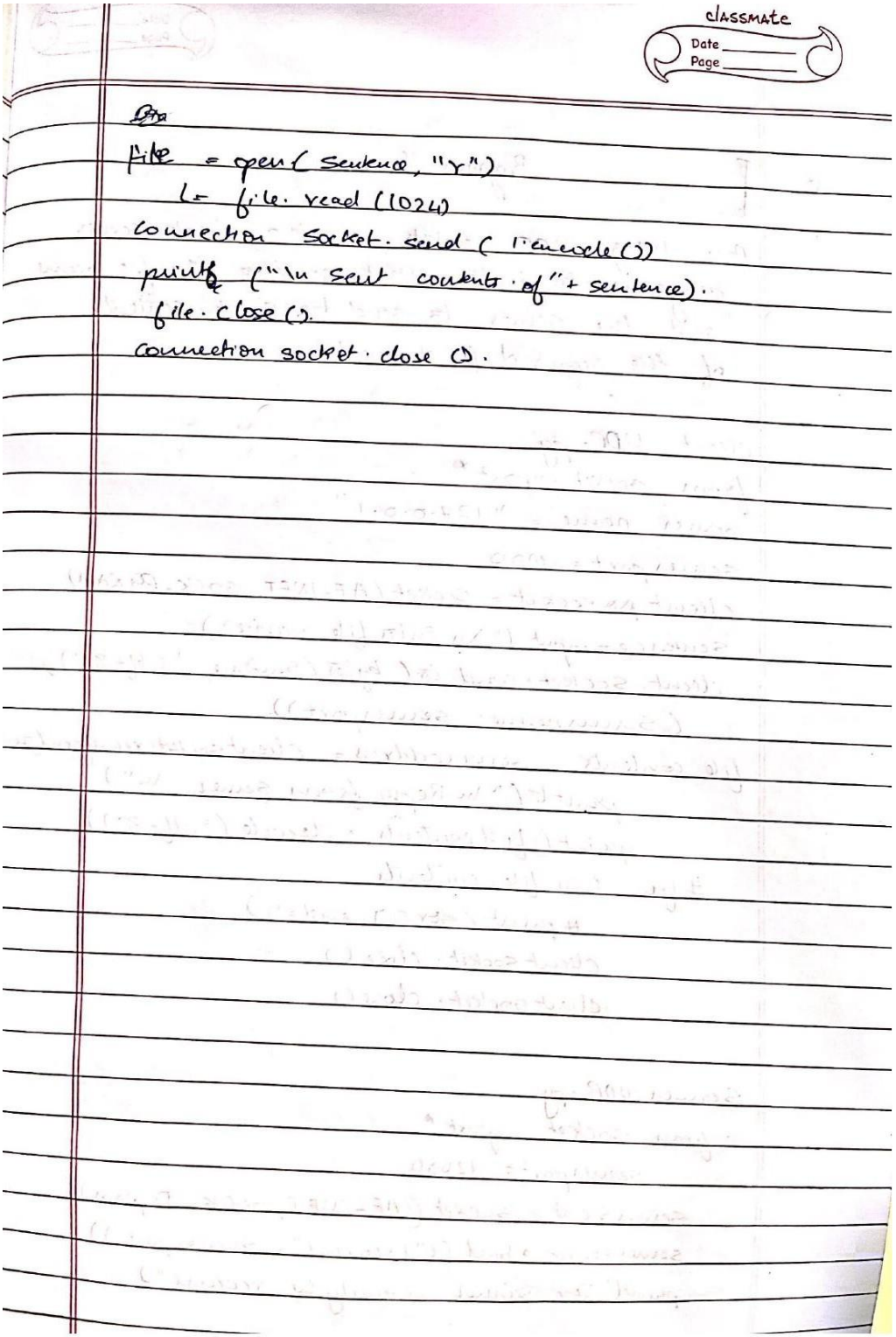
Aim: Using TCP/IP sockets, write a client-server program to make circuit sending

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
```

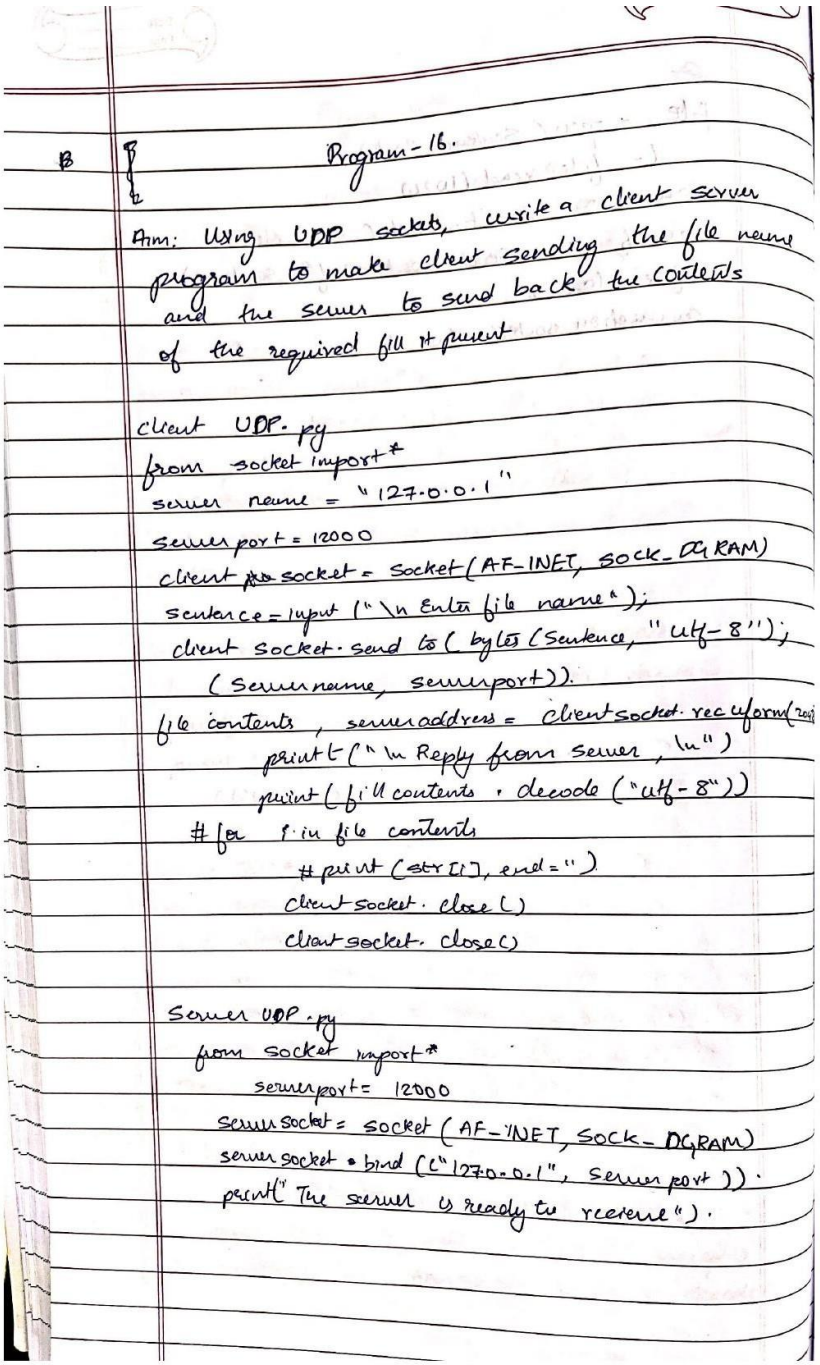
IDLE Shell 3.9.13

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.send to( bytes (Sentence, "utf-8");
   (Servername, serverport)).
file contents, serveraddress = clientsocket.recvfrom(200
      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, client address = server socket.recvfrom(2048)
    sentence = sentence.decode ("Utf-8").
    file = open (sentence, "r")
    con = file.read (2048)
server socket.send to ( bytes (con, "utf-8") client address)
    print ("In sent content of ', end = ")
        print(sentence)
    # for (i in sentence)
            # print (str(i), end = ").
        file.close().
```

CODE-

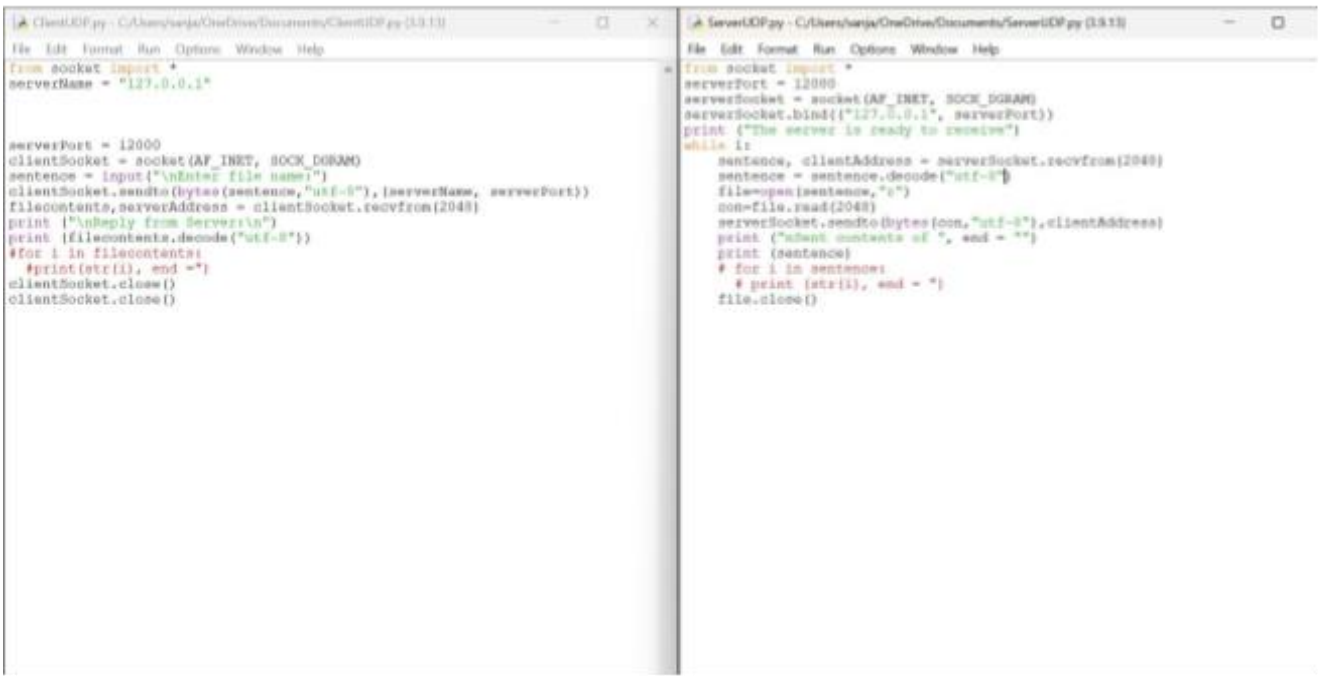
ClientUDP.py

```
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

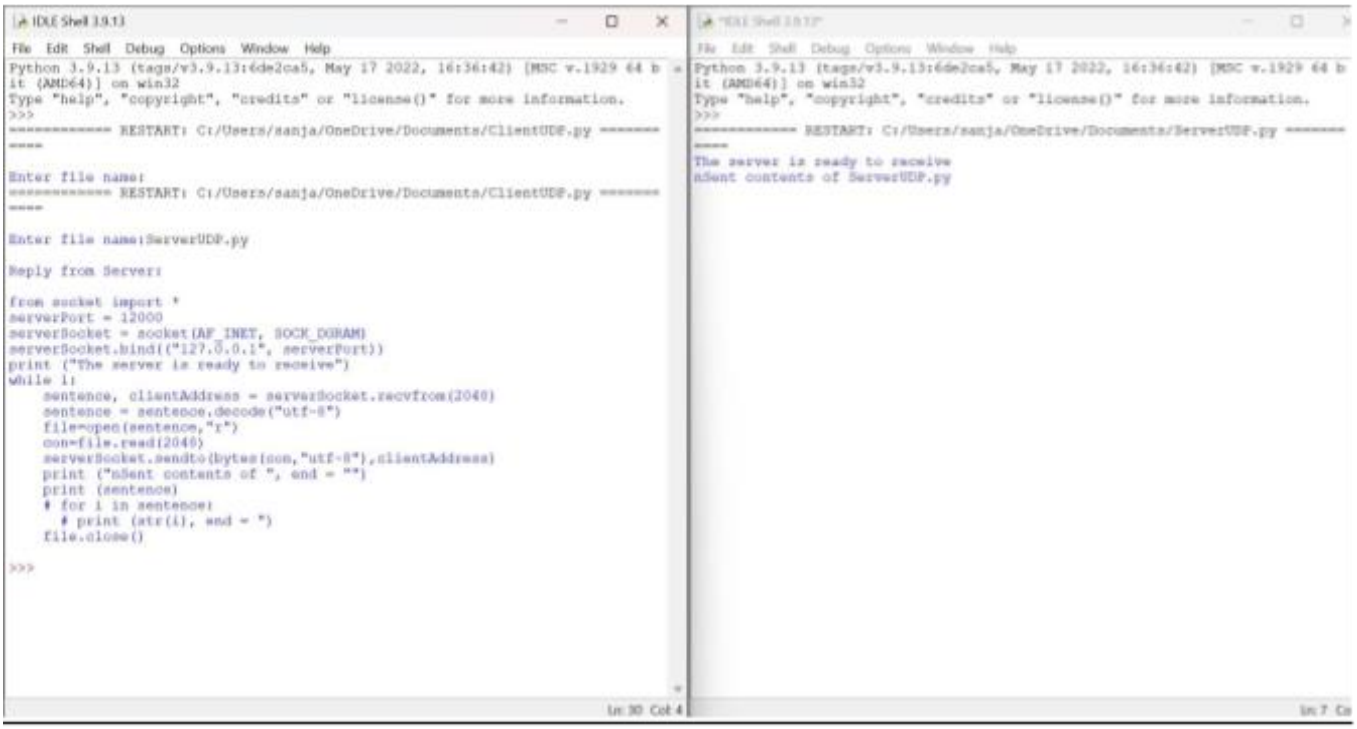
```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 1:
   sentence, clientAddress = serverSocket.recvfrom(2048)
   sentence = sentence.decode("utf-8")
   file=open(sentence,"r")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



```
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 bit (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()

>>>
```

```
Python 3.9.13 (tags/v3.9.13:6de2ca5, May 17 2022, 16:36:42) [MSC v.1929 64 bit (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
Sent contents of ServerUDP.py
```