

**VISVESVARAYA TECHNOLOGICAL
UNIVERSITY**

“JnanaSangama”, Belgaum -590014, Karnataka.



**LAB REPORT
on
COMPUTER NETWORKS**

Submitted by

SAMRITH SANJOO.S (1BM21CS185)

*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
June-2023 to September-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 **SAMRITH SANJOO.S(1BM21CS185)**, 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 academic semester June-2023 to September-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. Shyamala G
Assistant Professor
Department of CSE
BMSCE, Bengaluru

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

Index Sheet

Lab Program No.	Program Details	Page No.
1	Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping message.	5
2	Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply	12
3	Configure default route, static route to the Router	16
4	Configure DHCP within a LAN and outside LAN.,,	23
5	Configure Web Server, DNS within a LAN.	33
6	Configure RIP routing Protocol in Routers	37
7	Configure OSPF routing protocol	44
8	Demonstrate the TTL/ Life of a Packet	48
9	To construct simple LAN and understand the concept and operation of Address Resolution Protocol (ARP)	50

10	To construct a VLAN and make the PC's communicate among a VLAN	54
11	To understand the operation of TELNET by accessing the router in server room from a PC in IT office.	61

12)	To construct a WLAN and make the nodes communicate wirelessly	67
13)	Write a program for error detecting code using CRC-CCITT (16-bits).	77
14)	Write a program for congestion control using Leaky bucket algorithm.	81
15)	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.	88
16)	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.	91

Program 1

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

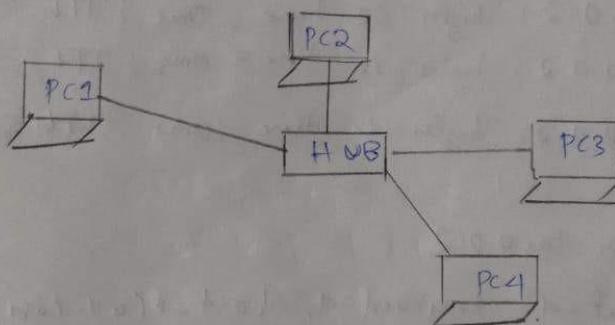
EXPERIMENT 1

O) Create a topology and simulate a simple PDU from Source to destination using hub and Switch as connecting devices and demonstrate Ping message.

AIM:

- * To create a topology using only HUB
- * To create a topology using Switch.
- * To create a topology using Switch and Hub.

→ To create a topology using Hub.



Result:

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 = 3ms	TTL = 128
Reply	from	10.0.0.4 :	bytes = 32	time = 0ms	TTL = 128
Reply	from	10.0.0.4 :	bytes = 32	time = 0ms	TTL = 128
Reply	from	10.0.0.4 :	bytes = 32	time = 0ms	TTL = 128

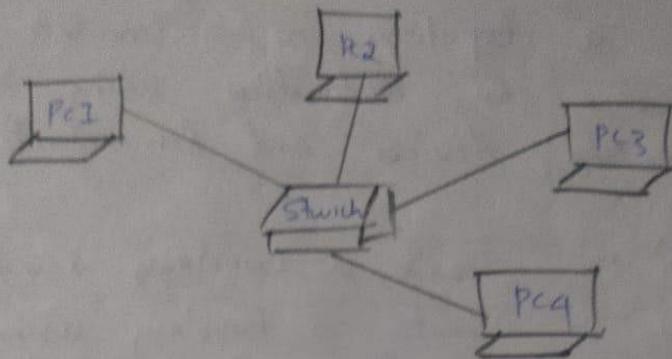
Ping Statistics for 10.0.0.4

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

Approximate Round trip times in milliseconds.

Minimum = 0ms , Maximum = 3ms , Average = 0ms .

→ To create a topology using Switch



Result $PC > ping 10.0.0.2$

Pinging 10.0.0.2 with 32 bytes of data

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

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

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

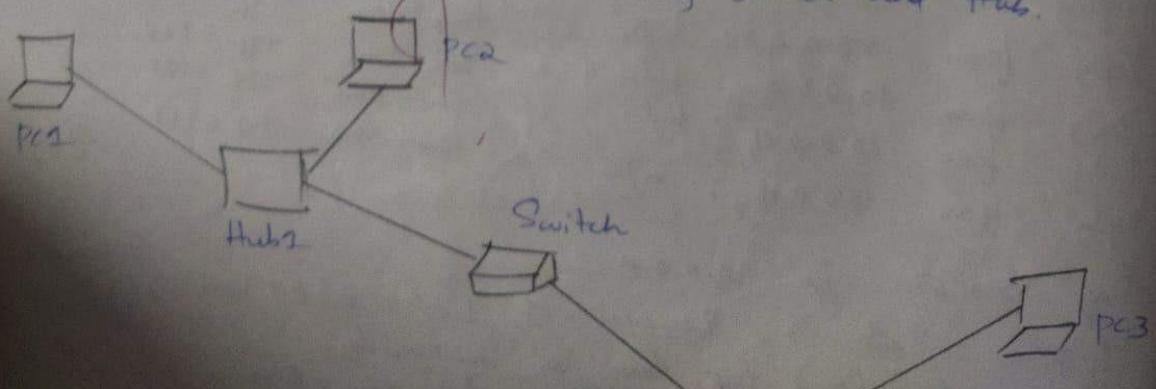
Reply from 10.0.0.2: bytes = 32 time = 0ms TTL = 128

Ping Statistics for 10.0.0.2

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milliseconds:

Minimum = 0ms, Maximum = 3ms, Average = 0ms.

→ To Create a topology using Switch and hub.



Objective :- * The Main objective of a Hub is to transmit the signal to the port, which will respond to where the signal was received.

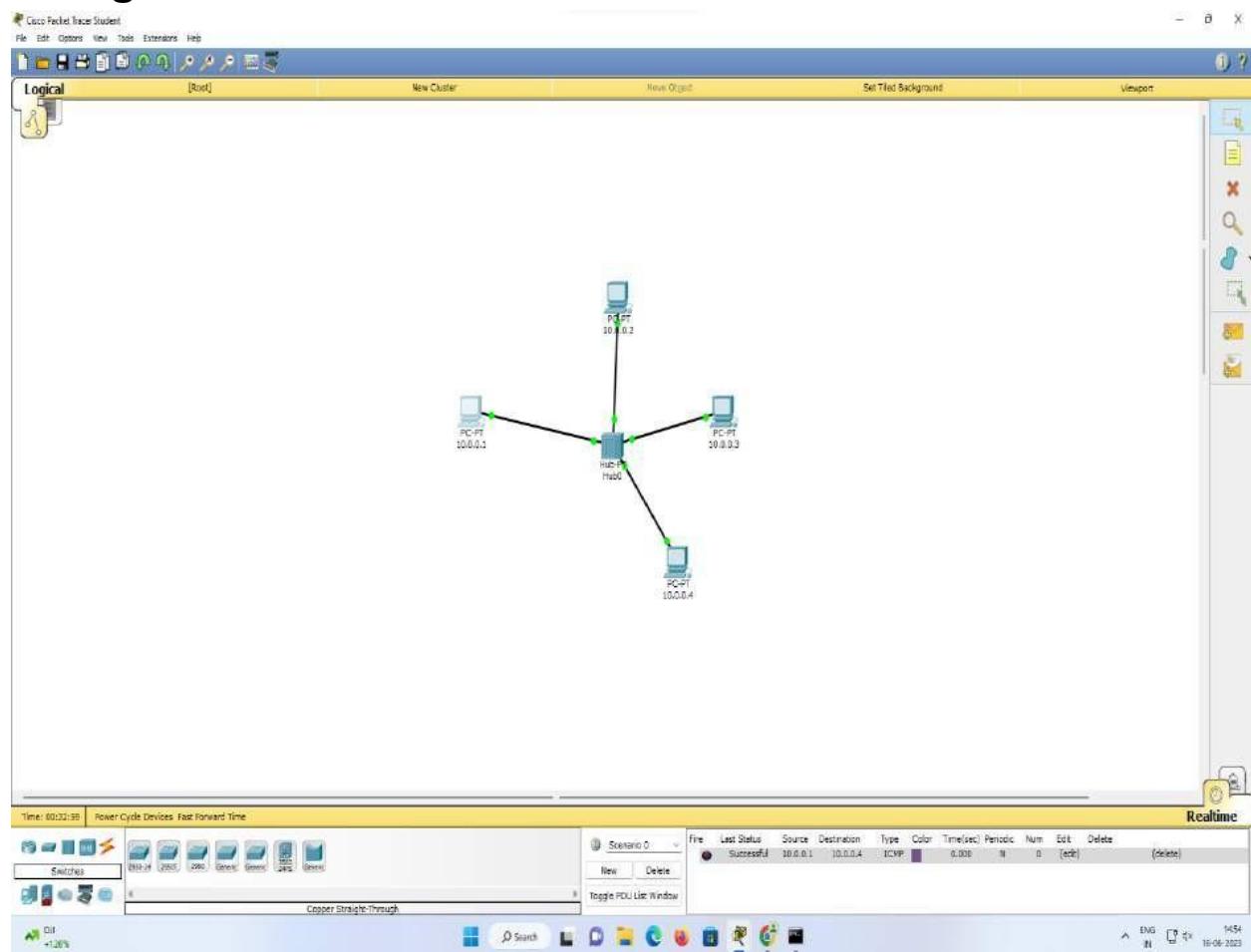
* A Switch allows you to Set up and terminate connections as needed.

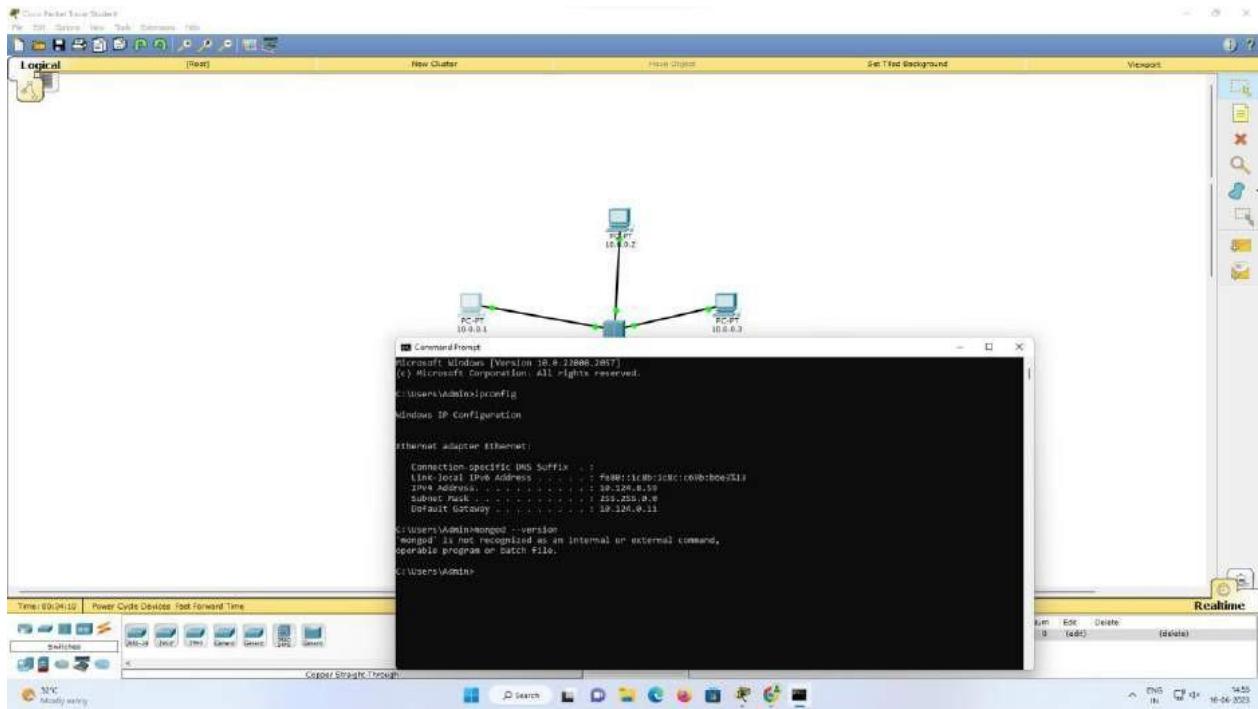
* Hubs use broadcast type transmission.

* Switch uses unicast, multicast as well as broadcast type transmission.

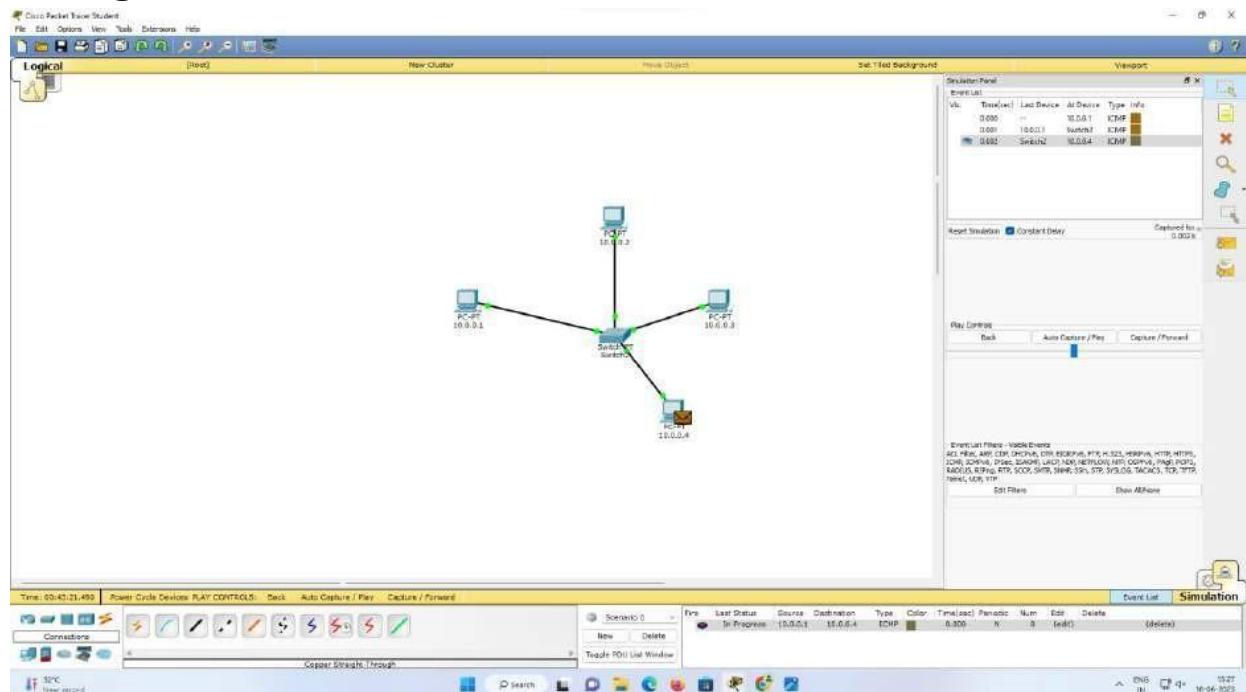
Q1
3/6/23

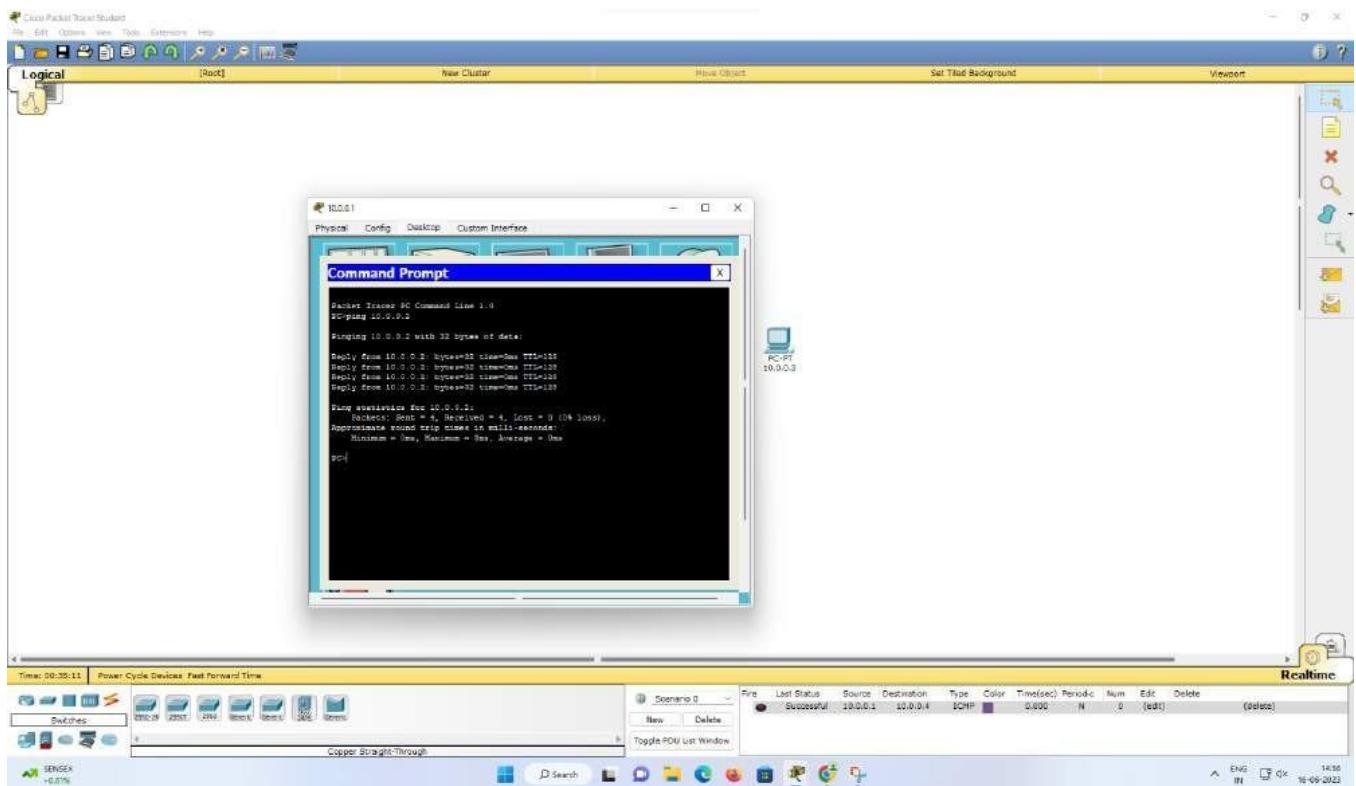
Using hub:



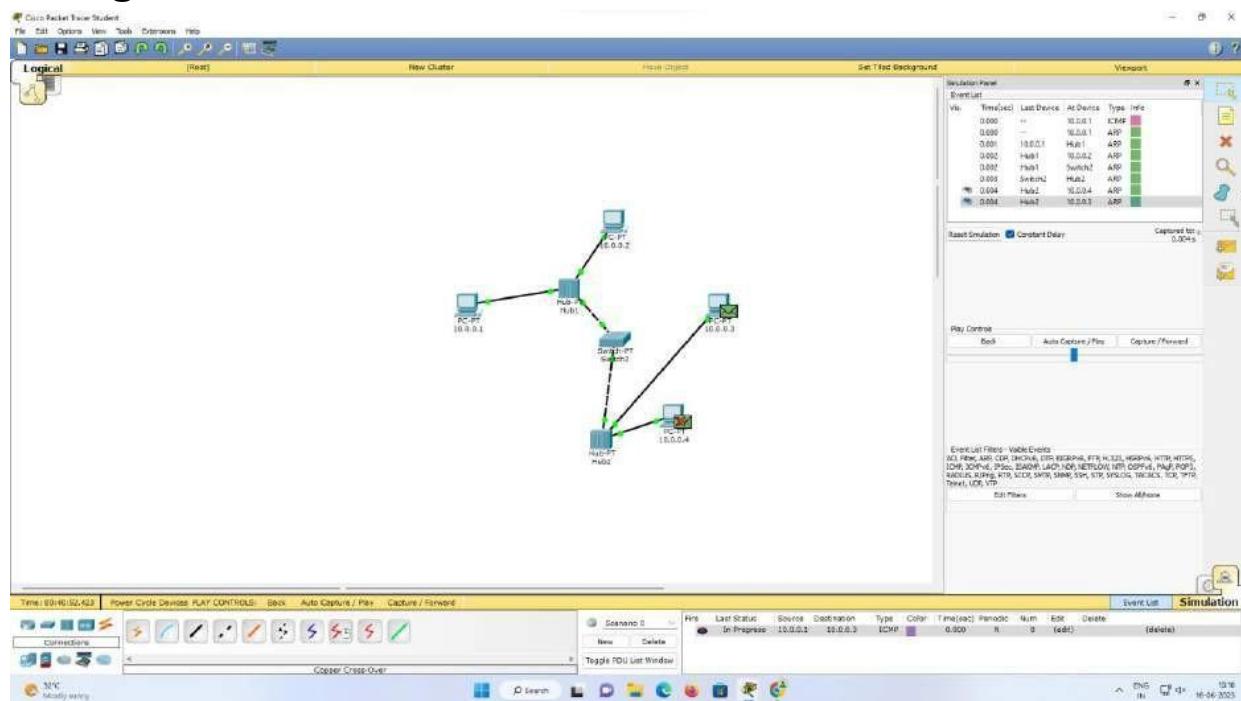


Using Switch:





Using switch and hub:



Program 2

Create a topology consisting of 2 devices connected with the help of a router.

EXPERIMENT - 2

Q1 Create a topology consisting of 2 devices connected with help of a router

AIM 1) To create a topology using one router
2) Using multiple routers

→ One only router.

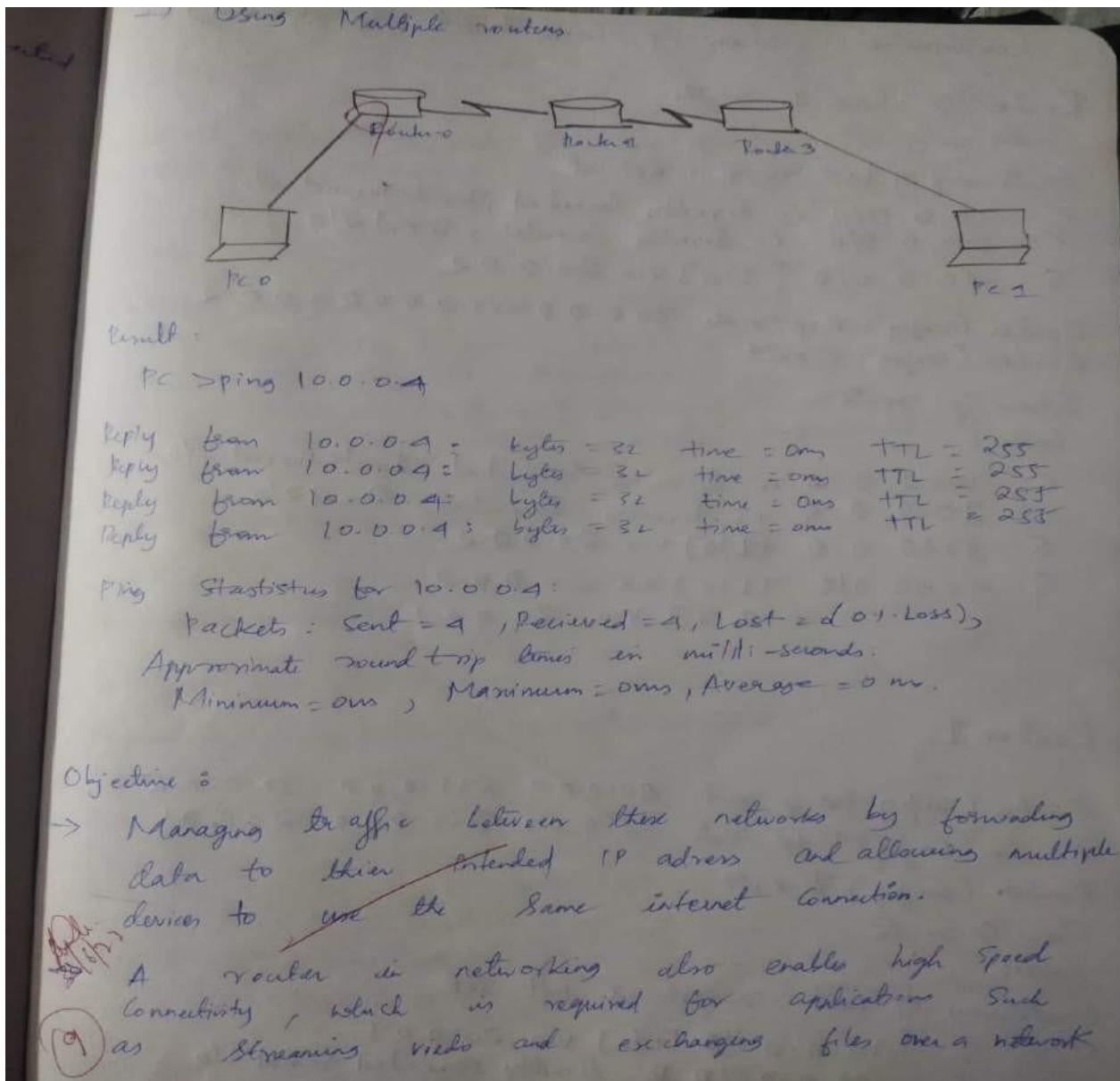
Result :

pc > ping 10.0.0.3
pinging 10.0.0.3 with 32 bytes of data.

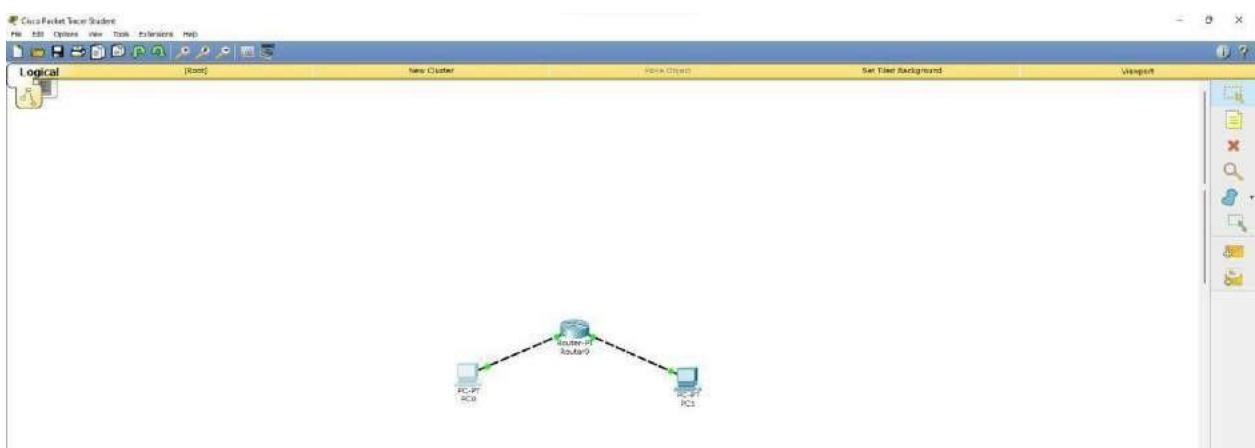
Reply from 10.0.0.3 :	bytes = 32	time = 0ms	TTL = 255
Reply from 10.0.0.3 :	bytes = 32	time = 3ms	TTL = 255
Reply from 10.0.0.3:	bytes = 32	time 0ms	TTL = 255
Reply from 10.0.0.3:	bytes = 32	time = 0ms	TTL = 255

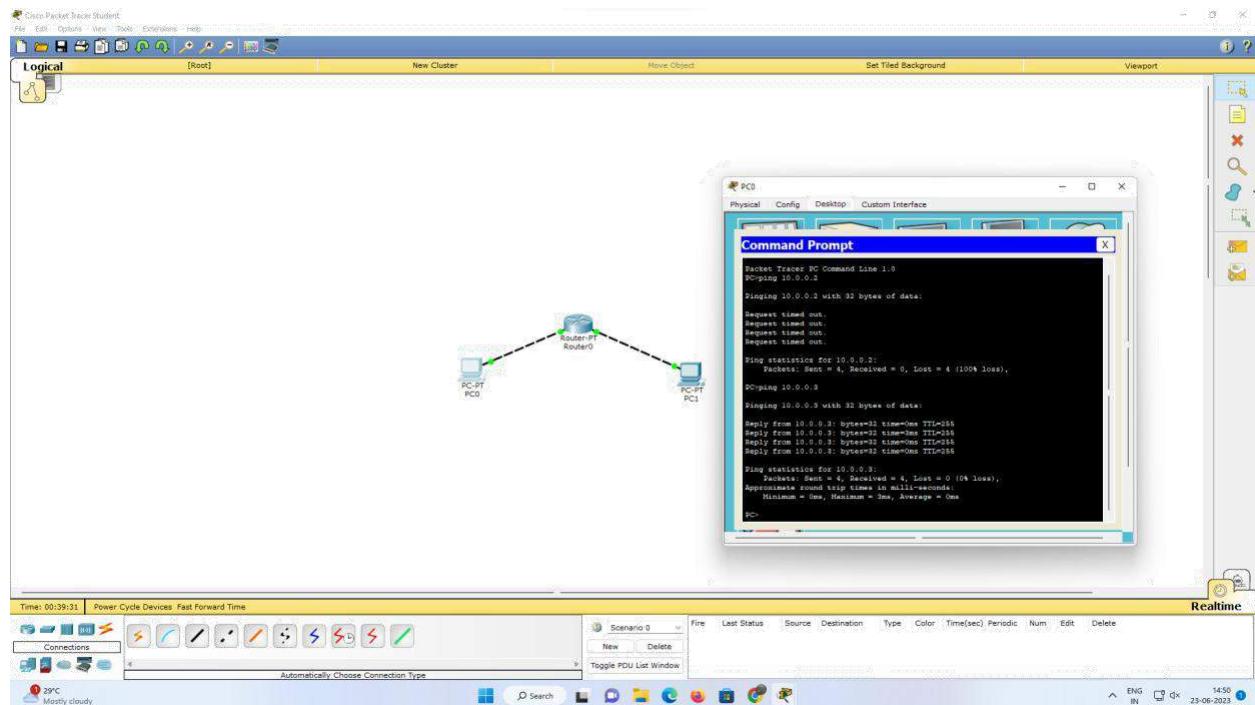
Ping Statistics for 10.0.0.3.

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip in milli seconds:
Minimum = 0ms, Maximum = 3ms, Average = 0ms.

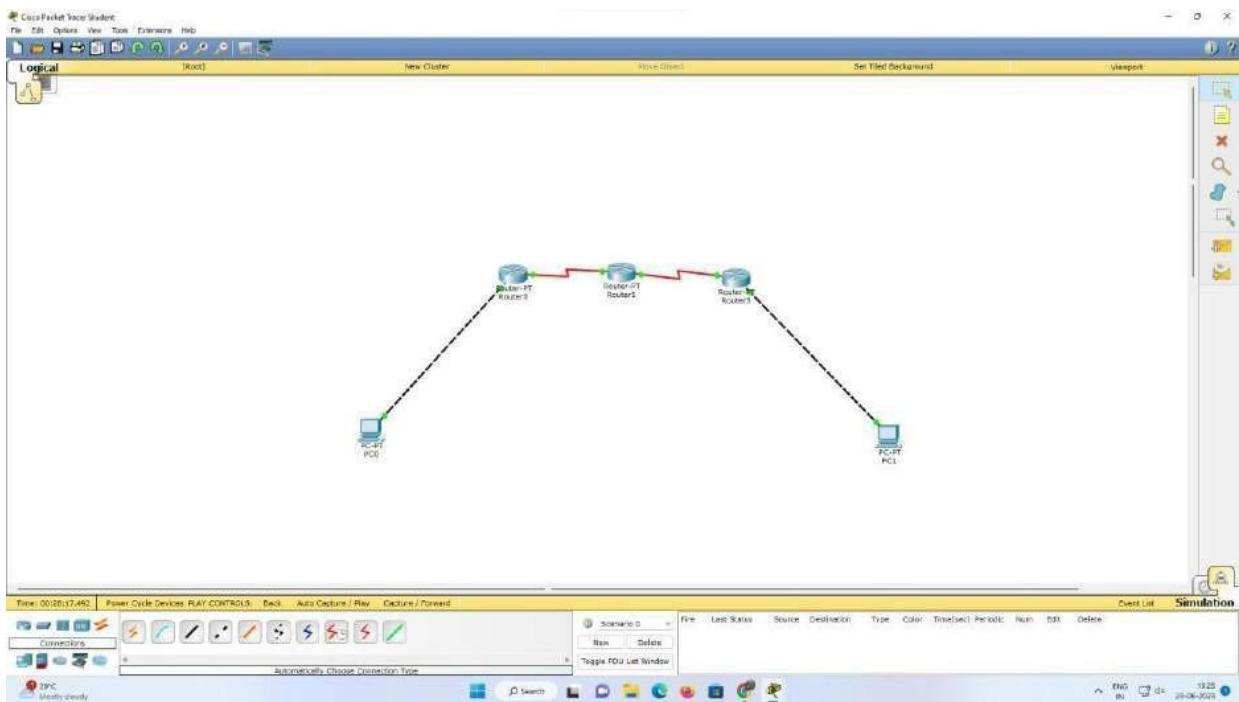


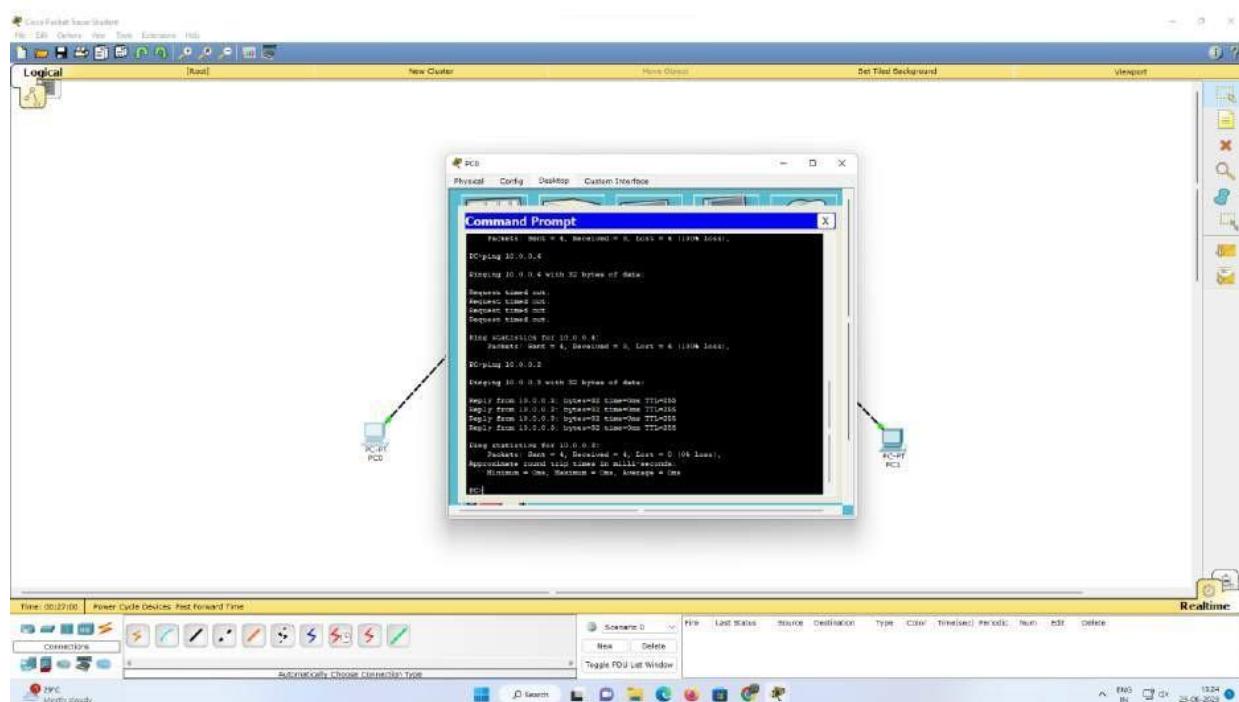
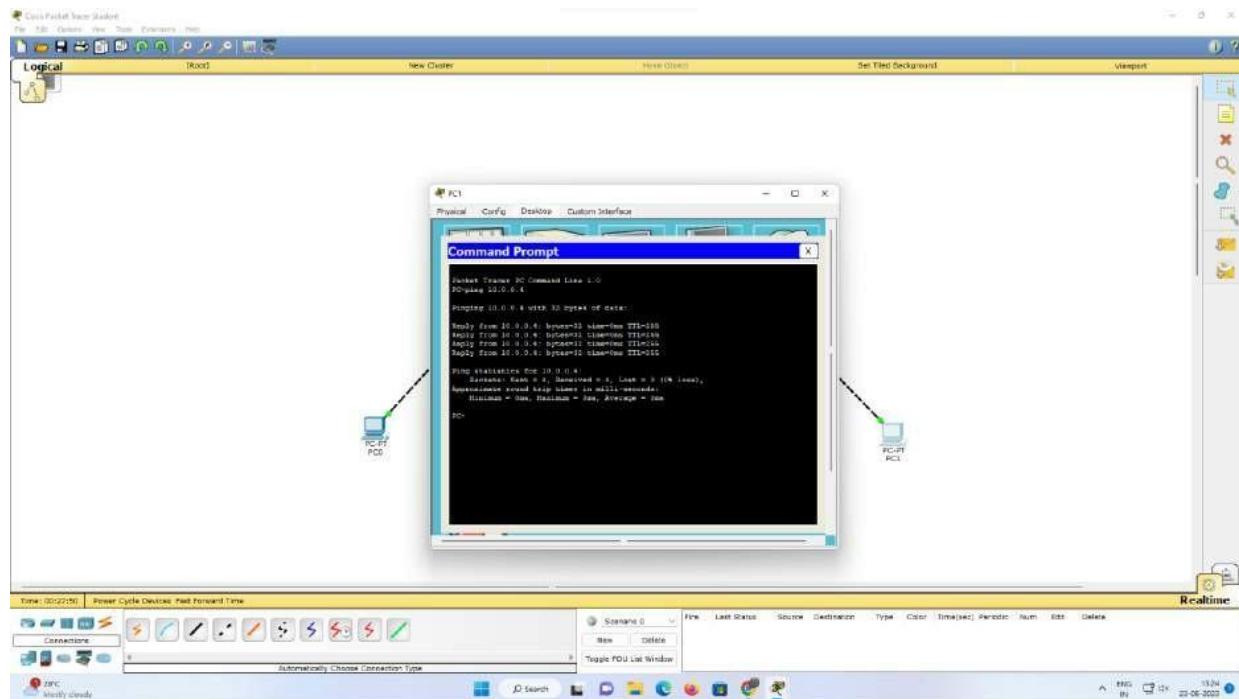
Using one router:





Using multiple routers:





Program 3

Create a topology consisting of 2 devices connected with the help of a router.

Continuation (Show ip commands and ping command)

Router > Show ip route
Codes:

Gateway of last resort is not set

C 10.0.0.10 is directly connected, FastEthernet 0/0
C 20.0.0.10 is directly connected, Serial 2/0
S 40.0.0.10 [2/0] via 20.0.0.2

Router (config) # ip route 30.0.0.0 255.0.0.0 20.0.0.2

Router (config) # exit

Show ip route
Codes:

C 100.0.0.10 is directly connected, FastEthernet 0/0
C 200.0.0.10 is directly connected, Serial 2/0
S 300.0.0.10 [2/0] via 20.0.0.2
S 400.0.0.10 [2/0] via 20.0.0.2
[2/0] via 30.0.0.1

Router I

Router (config) # ip route 40.0.0.0 255.0.0.0 30.0.0.2
Router (config) # ip route 10.0.0.0 255.0.0.0 20.0.0.1

Router (config) # exit

Show ip route
Codes:

Gateway of last resort is not set

S 10.0.0.0/0 [1/0] via 20.0.0.1
C 40.0.0.0/0 is directly connected Serial 2/0
30.0.0.0/0 is directly connected Serial 3/0

S 40.0.0.0/0 [2/0] via 30.0.0.2

Router 2

Router (config) # ip route 10.0.0.0 255.0.0.0 30.0.0.2
Router (config) # exit

Show ip route

Code: - - -

Gateway of last resort is not set

S 10.0.0.0/0 [2/0] via 30.0.0.1

S 20.0.0.0/0 [2/0] via 30.0.0.2

C 30.0.0.0/0 is directly connected, Serial2/0

C 40.0.0.1/0 is directly connected, Fastethernet 0/0

Ping Commands

PC0

Command prompt:

pc> ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

~~Request time-out~~

Reply from 40.0.0.02: bytes = 32 time = 12ms TTL = 125

Reply from 40.0.0.02: bytes = 32 time = 11ms TTL = 125

Reply from 40.0.0.02: bytes = 32 time = 10ms TTL = 125

Ping Statistics for 40.0.0.2:

packets: sent = 4, received = 3, Lost = 1 (25%)

Observation : *) The output of Show ip route command is a very powerfull tool to use at this point in our search.

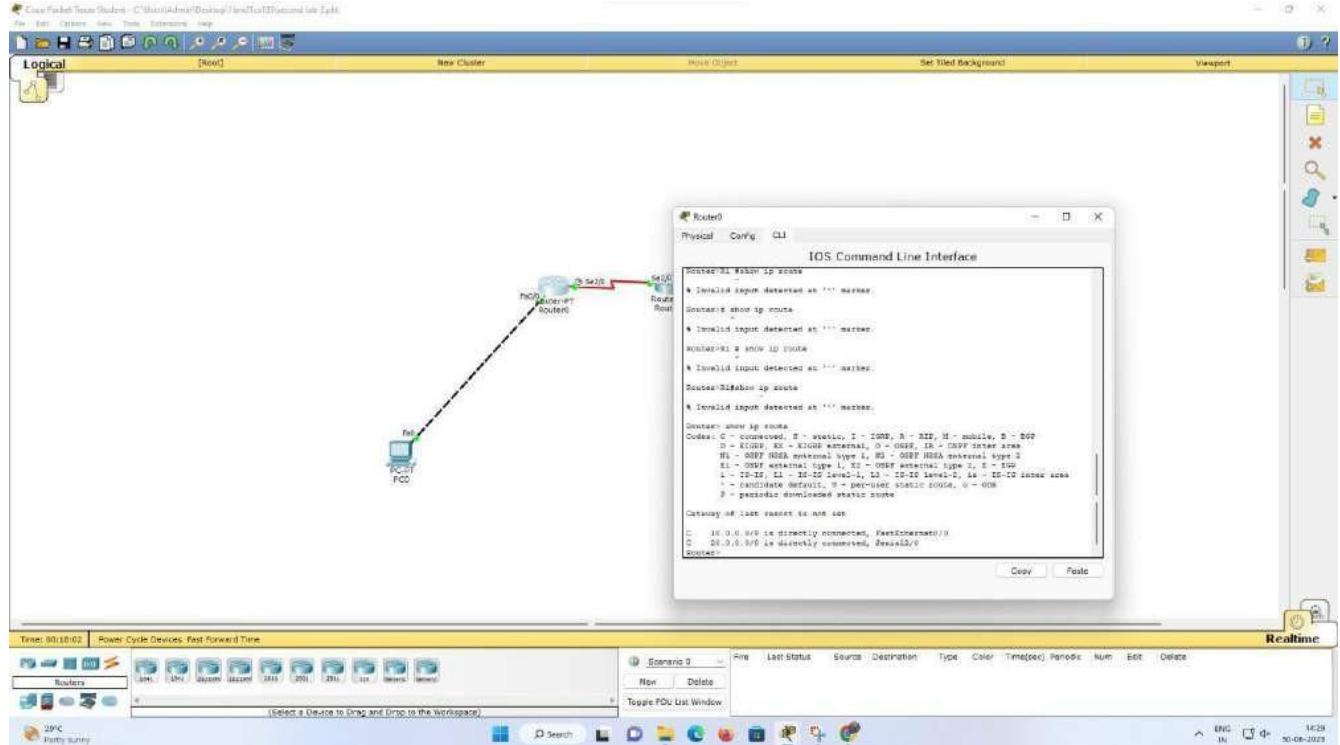
*) The output of Show ip route displays the entries in the routing table.

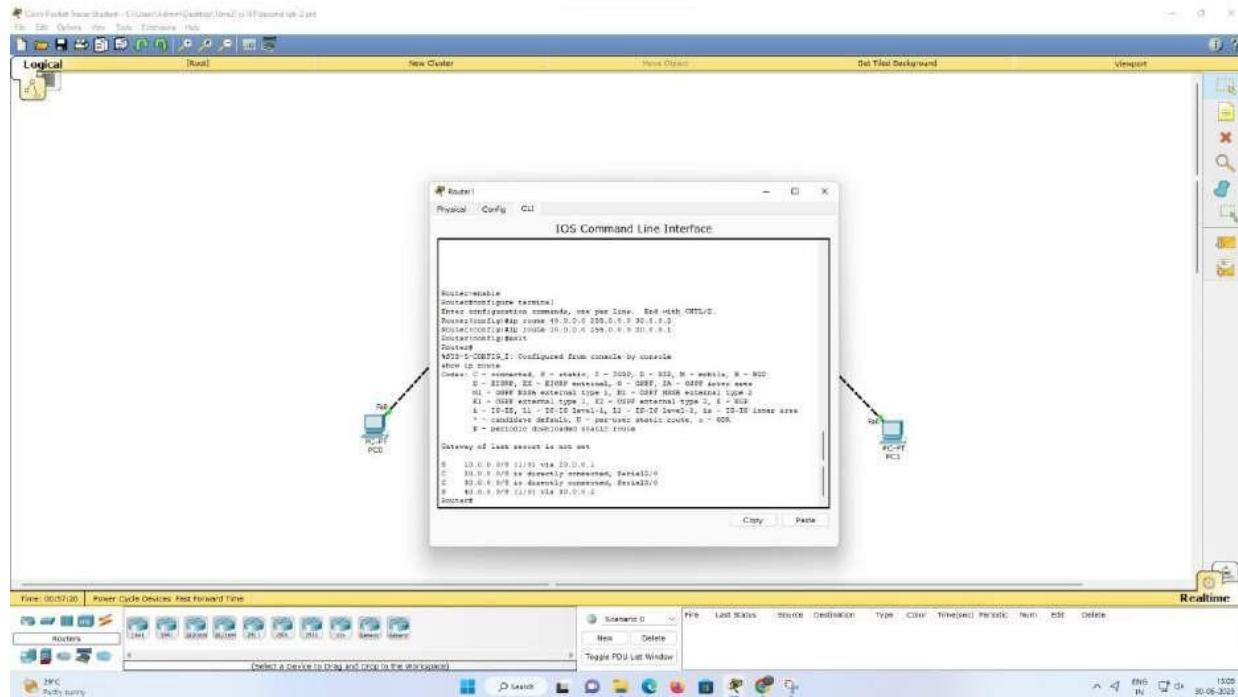
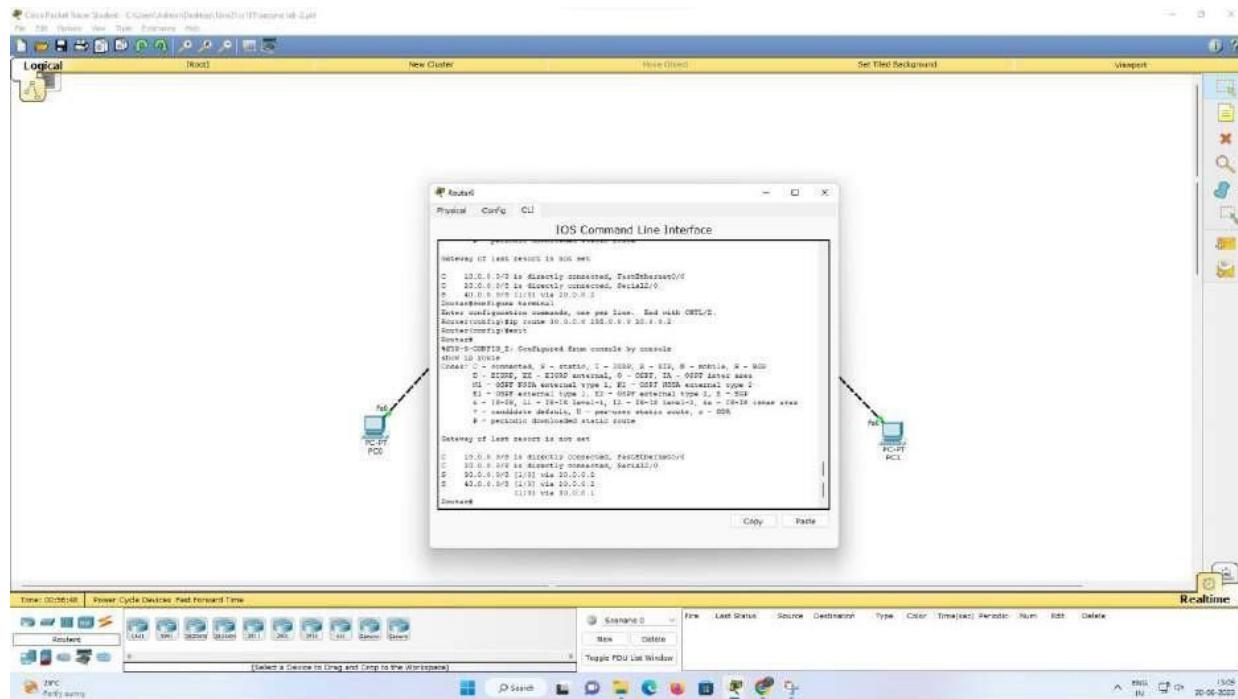
*) It also displays the gateway of last resort sometimes called as default gateway, if one configured.

*) This is a static route configured by the user that routes the IP address 0.0.0.0 (all destination) through a single host (the gateway)

Using multiple routers:

Show ip commands and ping commands



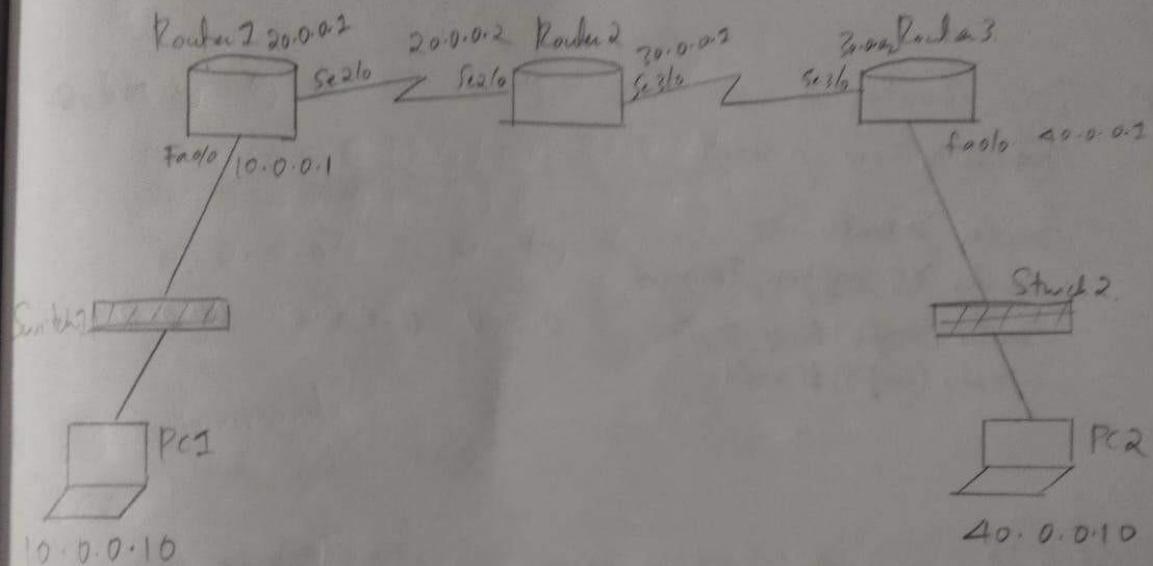


Lab 3:

Default Routing

Q) Create a topology consisting of 3 routers with Switch and configure default routing.

AIM: To create a topology of 3 routers and switch



Working Commands:

For Router 1:

Router > Show ip route

Setting default route for Router 1

Router > enable

Router # Configure terminal

Router (config) # ip route 0.0.0.0 0.0.0.0 20.0.0.2

Router (config) # exit

IP Router A

- C 20.0.0.0/8 is directly connected Se2/0
C 30.0.0.0/8 is directly connected Se3/0

IP routes

- C 20.0.0.0/8 is directly connected Se2/0
C 20.0.0.0/8 is directly connected Se3/0
S 10.0.0.0/8 [1/0] via 20.0.0.1
S 40.0.0.0/11 [0/1/0] via 30.0.0.2

IP route for Router 3

- C 30.0.0.0/8 is directly connected ,Se2/0
C 40.0.0.0/8 is directly connected ,Fa0/0
S* 0.0.0.0/0 0.0.0.0/0 [3/0] via 30.0.0.1

Pinging PC2 from PC1

PC1 > ping 40.0.0.2

Reply from 40.0.0.20 bytes = 32 time = 10ms TTL = 25

packets Sent = 4 , Received = 4 , loss = 0 (0% loss)

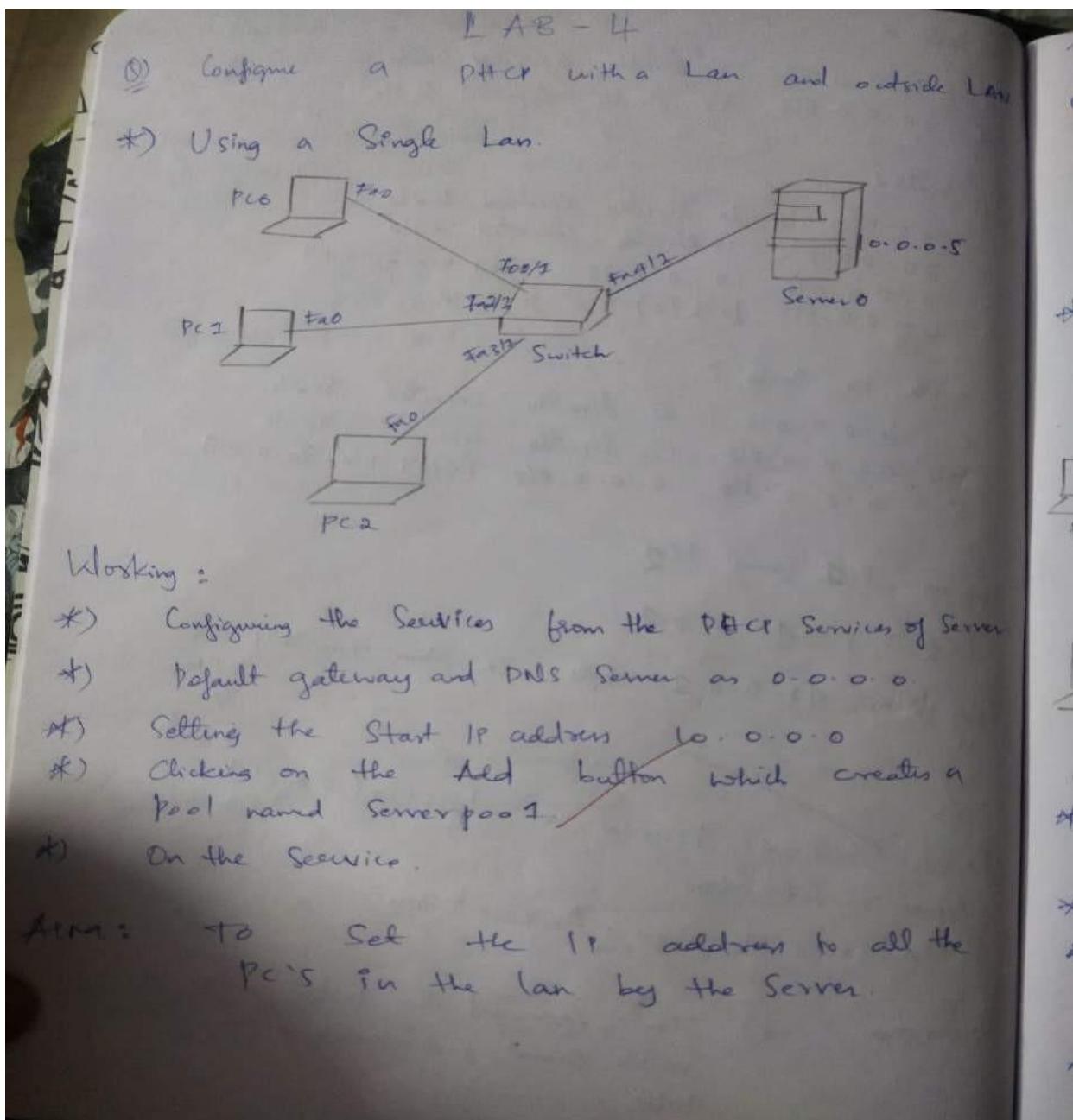
Approx round trip time

Min = 5ms Max = 10ms , Average = 7ms

Observation : * For forward the packet when the Switch cannot find a match in the routing table .

Program 4

Configuring DHCP with Lan



Result : In PC0

AN

① DHCP ② static ③ DHCP request Successful

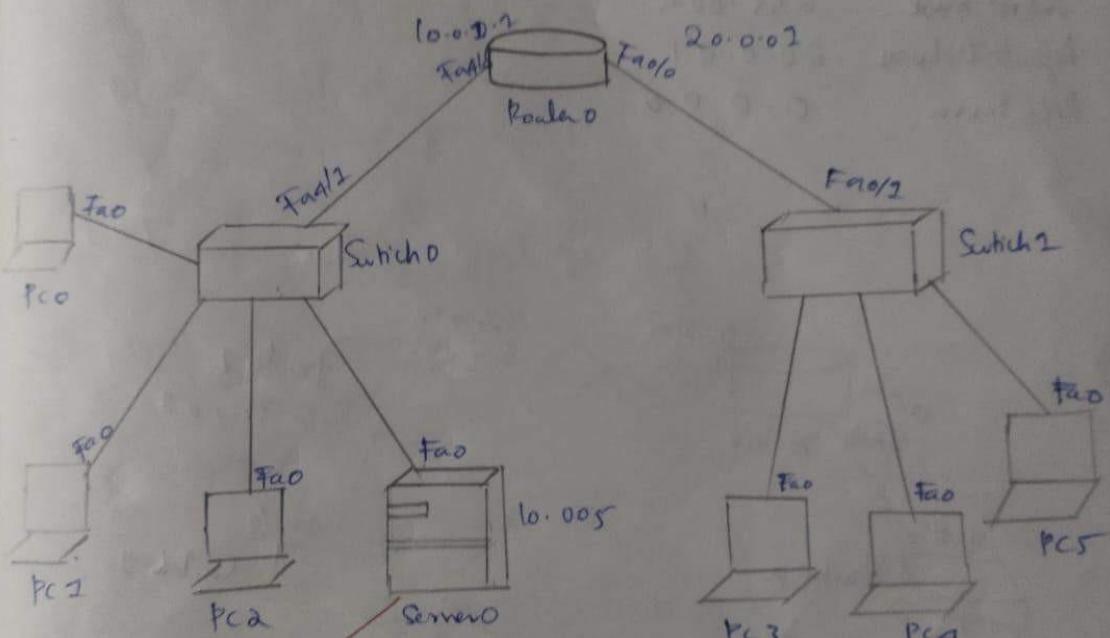
IP address 10.0.0.1

Subnet Mask 255.0.0.0

Default gateway 0.0.0.0

DNS Server 0.0.0.0

* Using 2 lan Networks and a Router.



Working:

* Extending the same topology and adding another LAN network with a Router.

* Adding another pool in Service

* Server pool 1 with Start IP Address: 20.0.0.10
and default gateway of 20.0.0.1

A) From the CLI of Router.

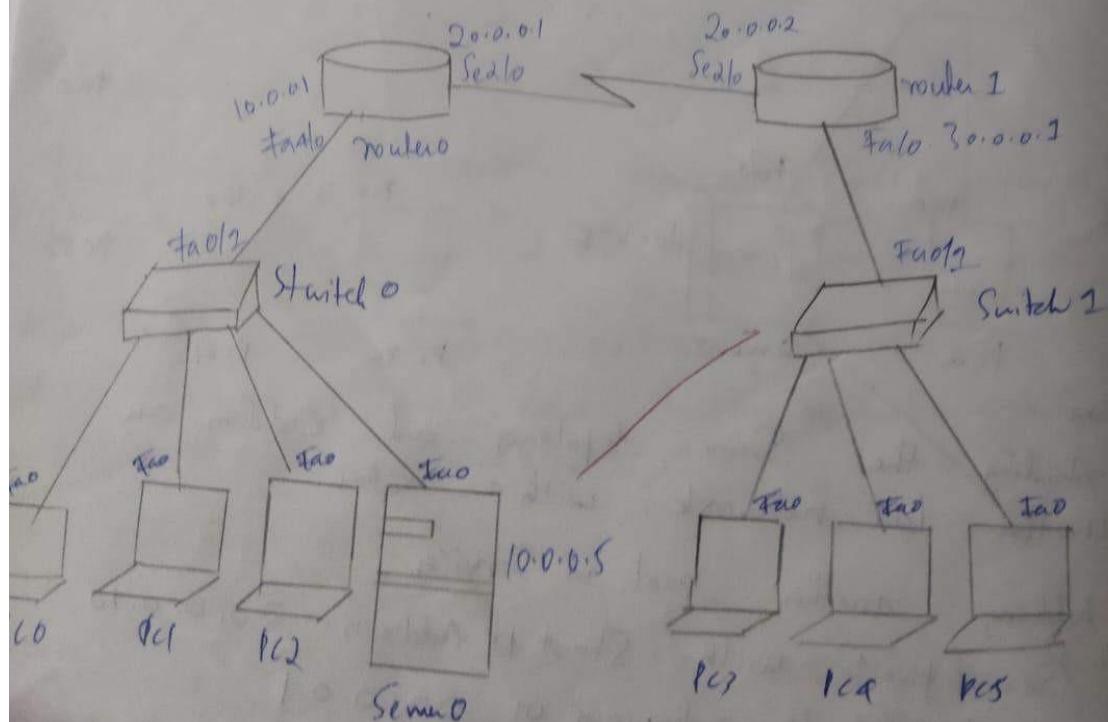
Router (config-if) # interface FastEthernet 0/0
Router (config-if) # ip helper-address 10.0.0.5
Router (config-if) # exit

Output : PC3

IP Configuration

④ DHCP ○ Static DHCP request Successful

IP Address 20.0.0.11
Subnet mask 255.0.0.0
Default Gateway 20.0.0.1
DNS Server 0.0.0.0



Procedures

*) Configure Static IP address on the Server 10.0.0.5

*) Creating a pool and save it. To do this go to Services and Select DHCP.

*) Pool Name : ServerPool

*) Default Gateway : 0.0.0.0

*) DNS Server : 0.0.0.0

*) Start IP address : 10.0.0.10

*) Subnet mask : 255.0.0.0

*) Pool Name : ServerPool1

*) Default Gateway : 10.0.0.3

*) Start IP address : 20.0.0.10

*) Subnet mask : 255.0.0.0

*) Turn on the Service:

*) Click on the 1st Router

→ ~~Router (config-if) # interface FastEthernet 0/0
Router (config-if) # ip helper address 10.0.0.5
Router (config-if) # exit~~

Click on the 2nd Router

→ Router (config-if) # interface FastEthernet
Router (config-if) # ip helper address 10.0.0.5
Router (config-if) # exit

Output : PC3

IP Configuration

② DHCP

③ Static

DHCP request Successful

IP Address : 20.0.0.11

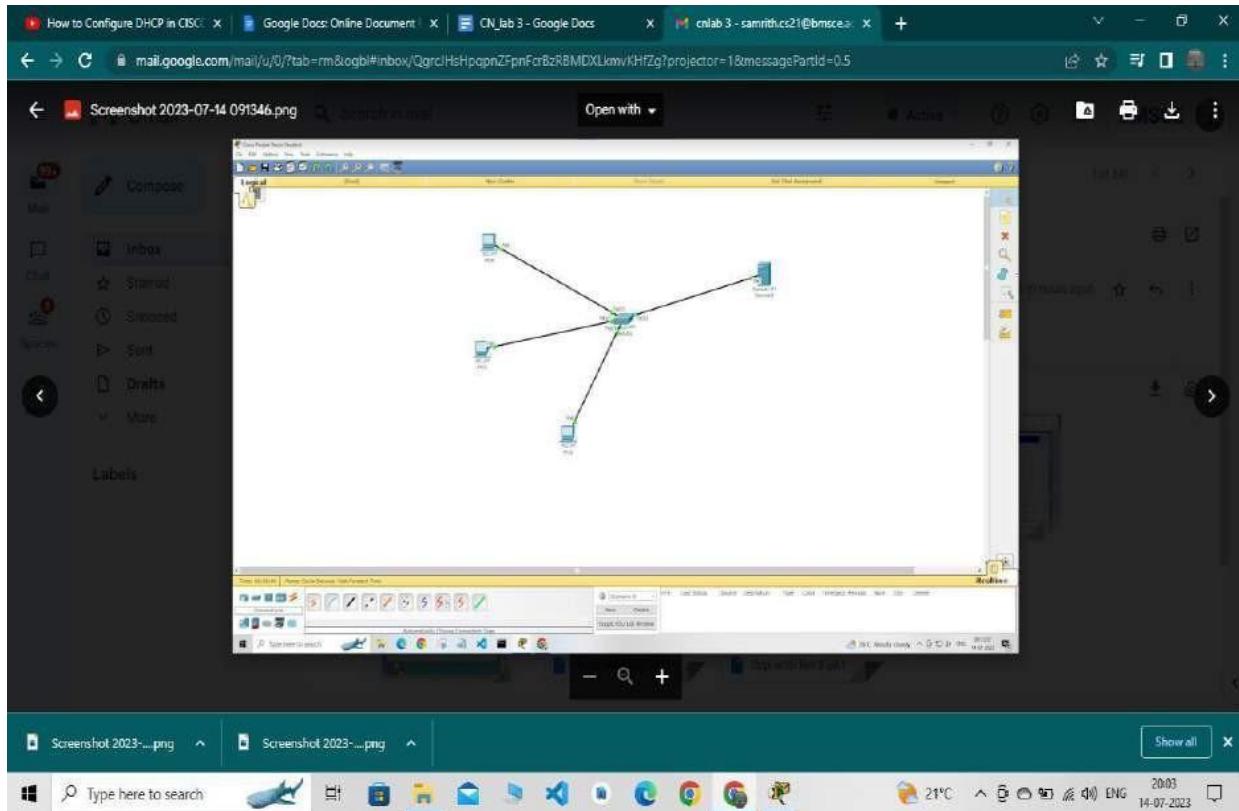
Subnet mask : 255.0.0.0

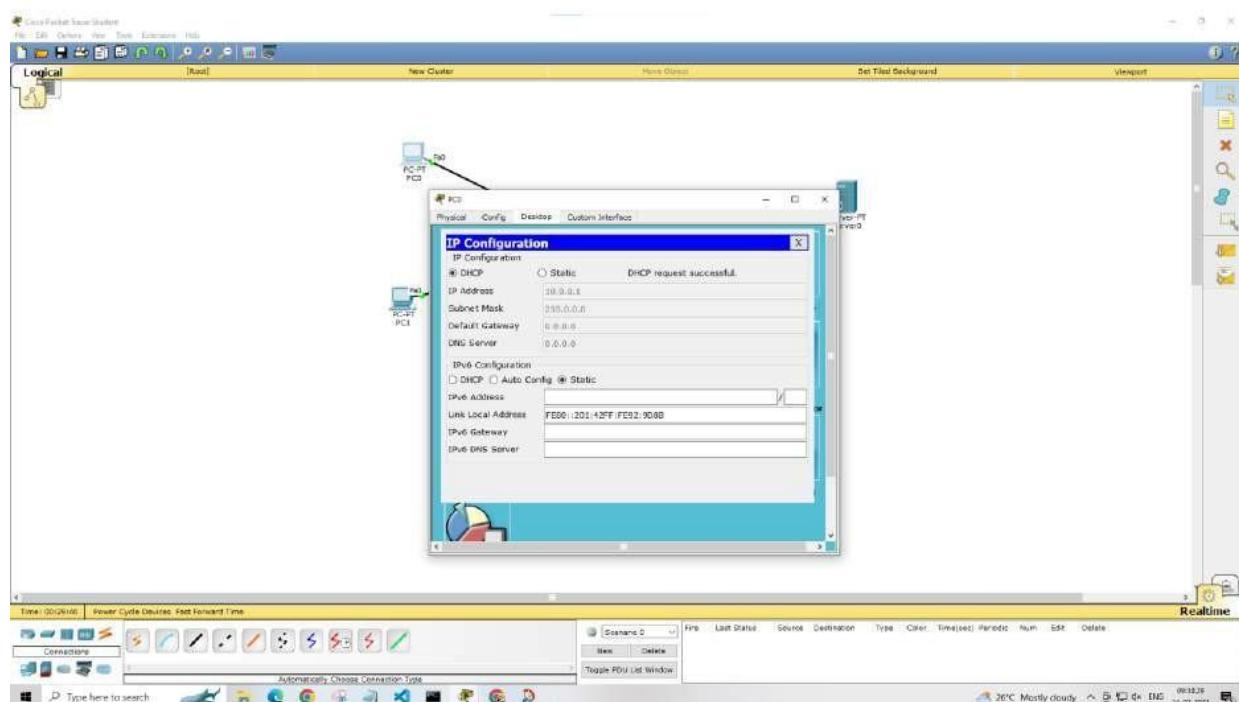
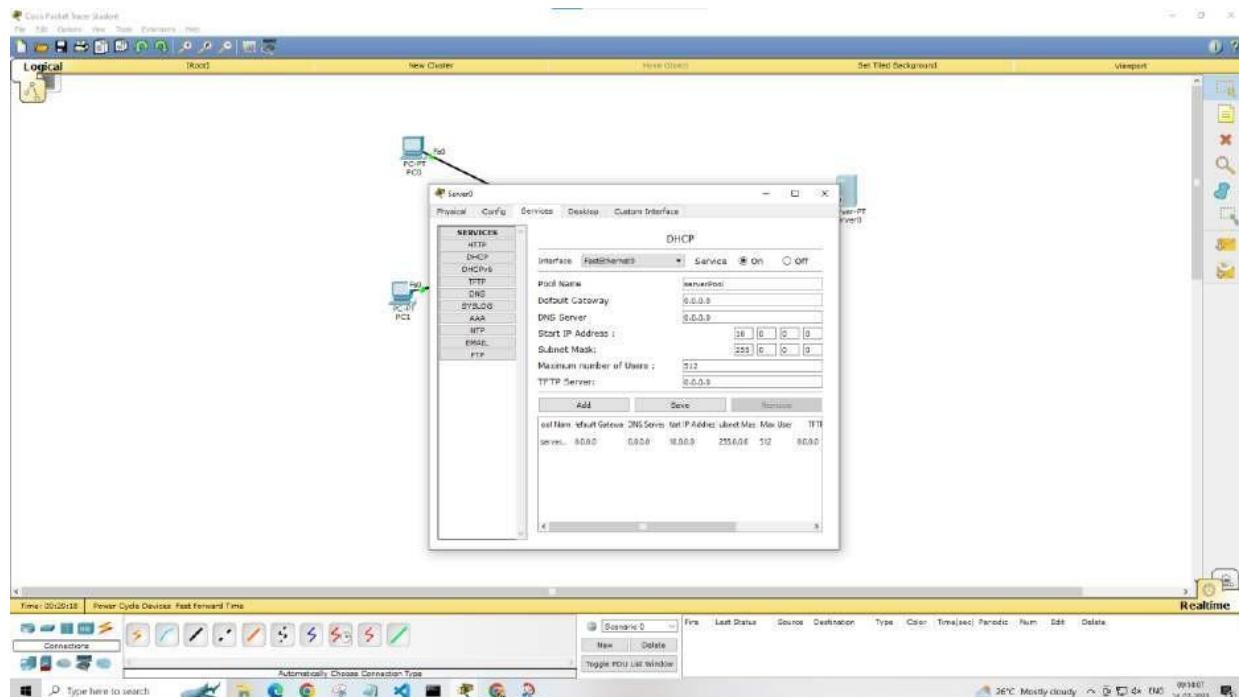
Default gateway : 20.0.0.1

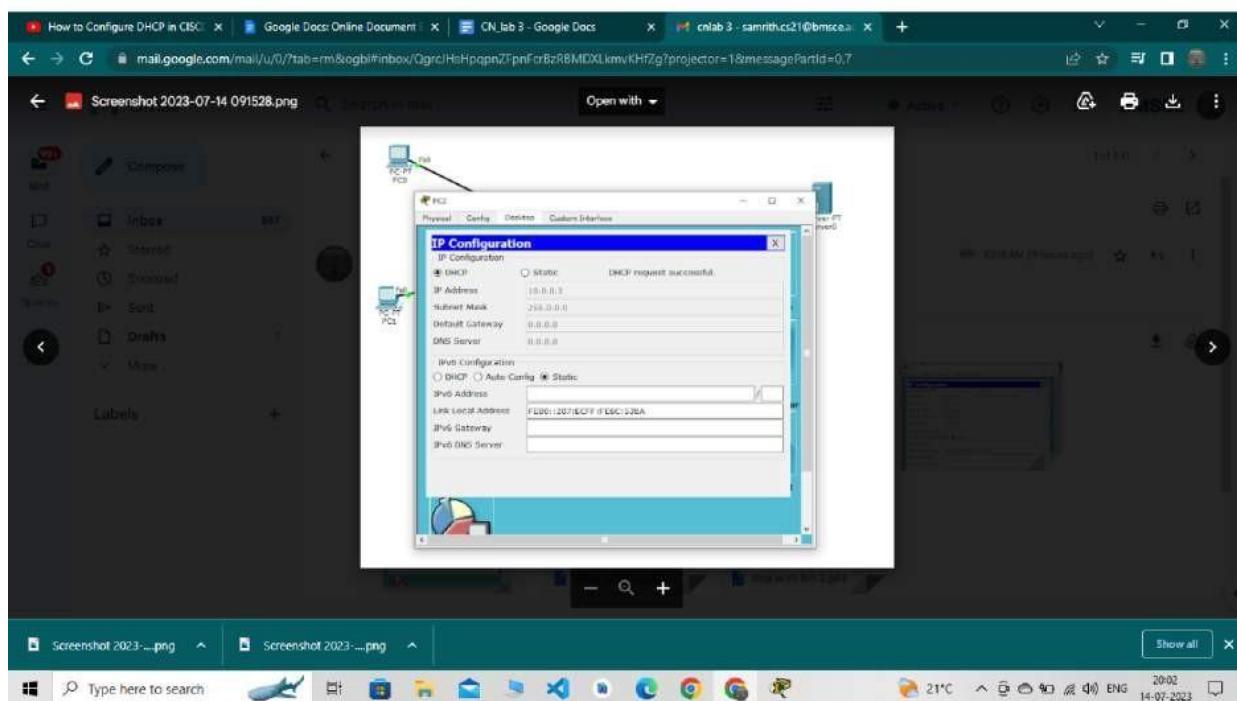
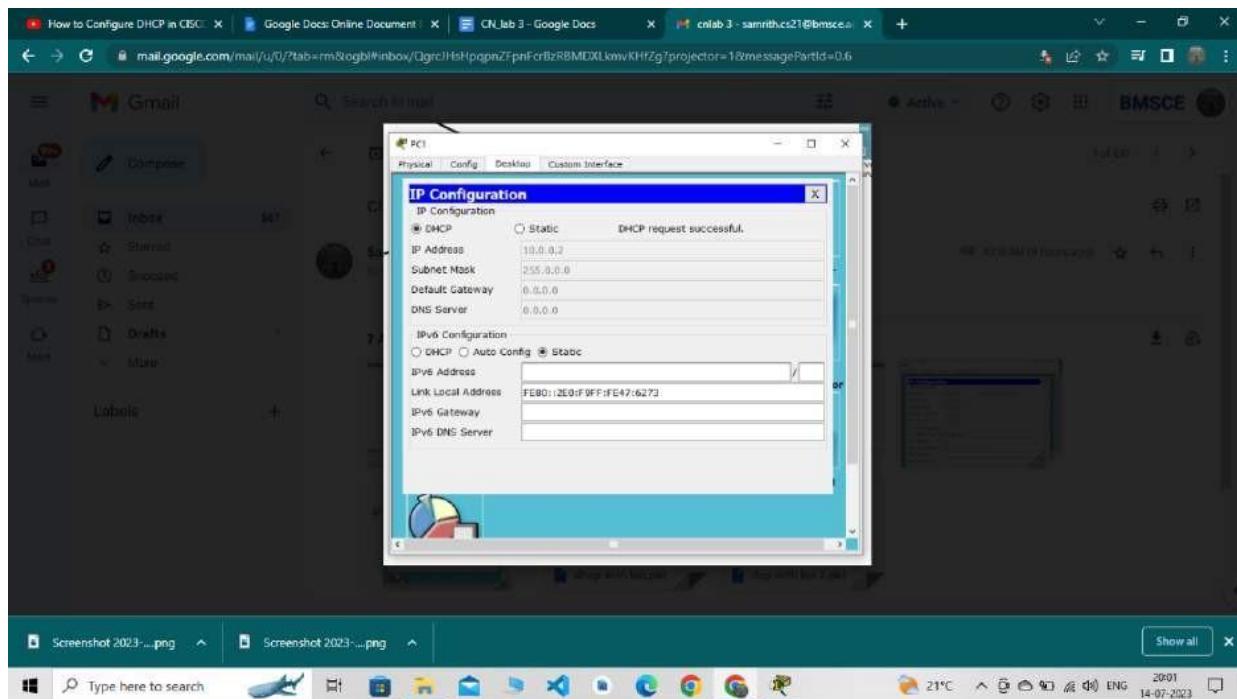
DNS Server : 0.0.0.0

- Observation :
- * The configuration of the Server by creating pool is observed.
 - * IP address is not set statically rather than uses DHCP Services.

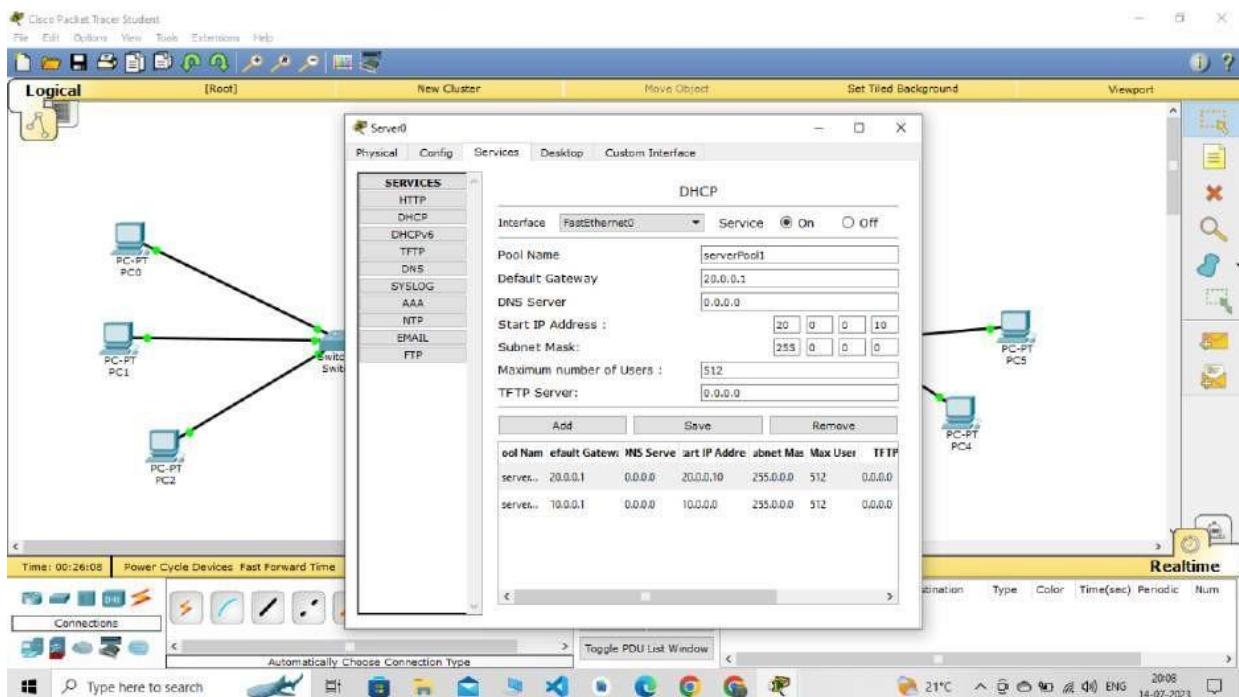
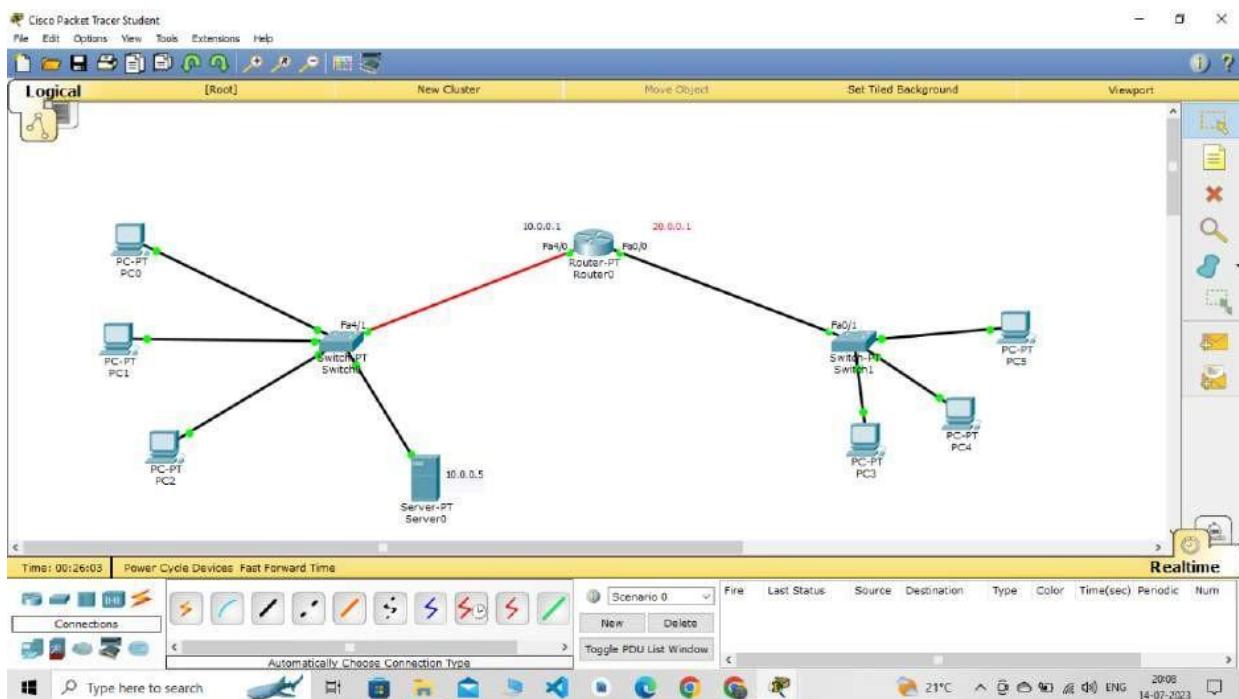
Using Single lan:

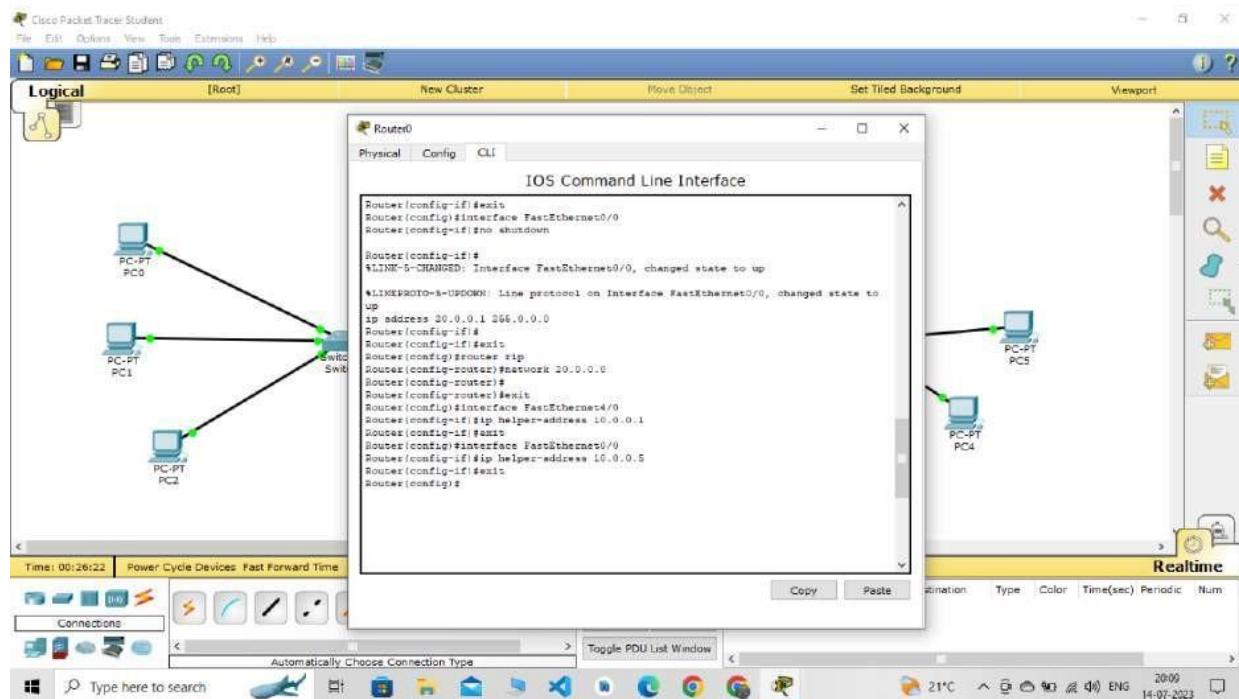
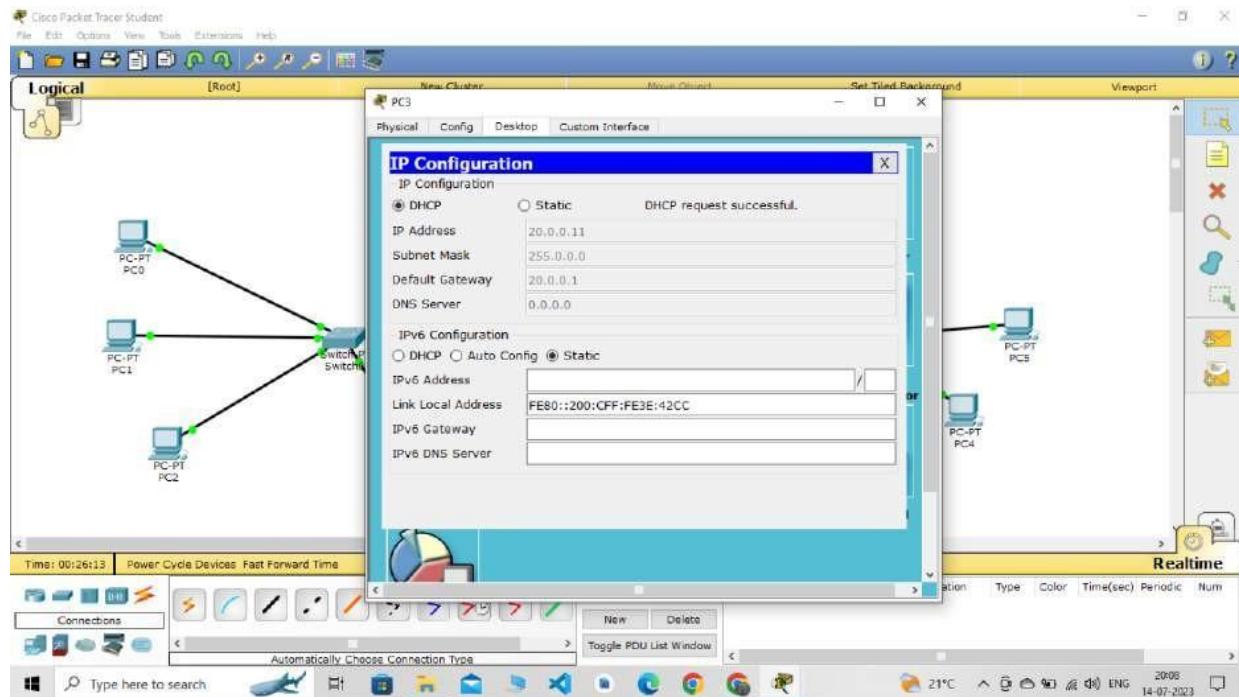






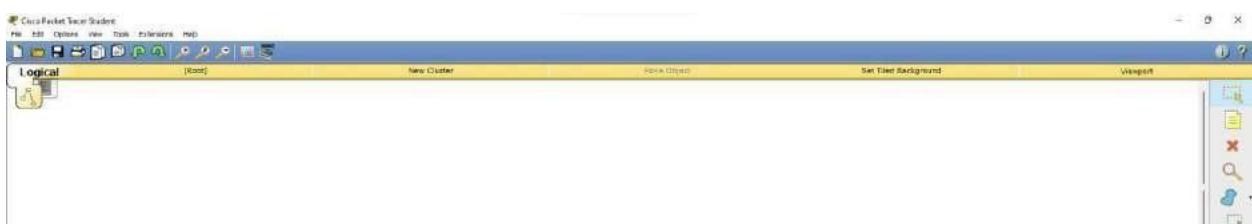
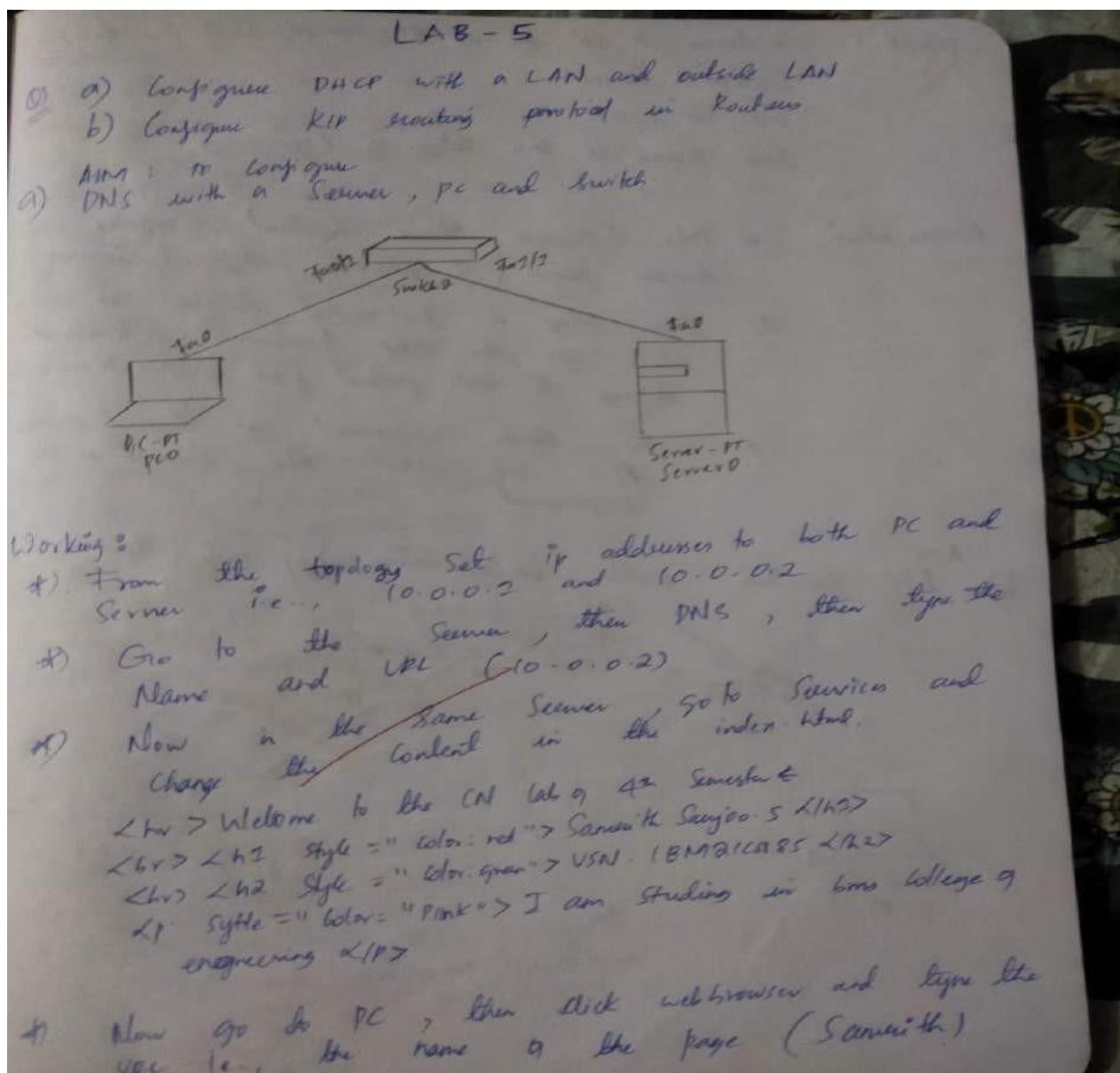
Using 2 lan networks





Program 5

Configuring DNS with a server and pc and a switch



result : Welcome to CN lab of 4th Semester

Name with Sanyoo.S

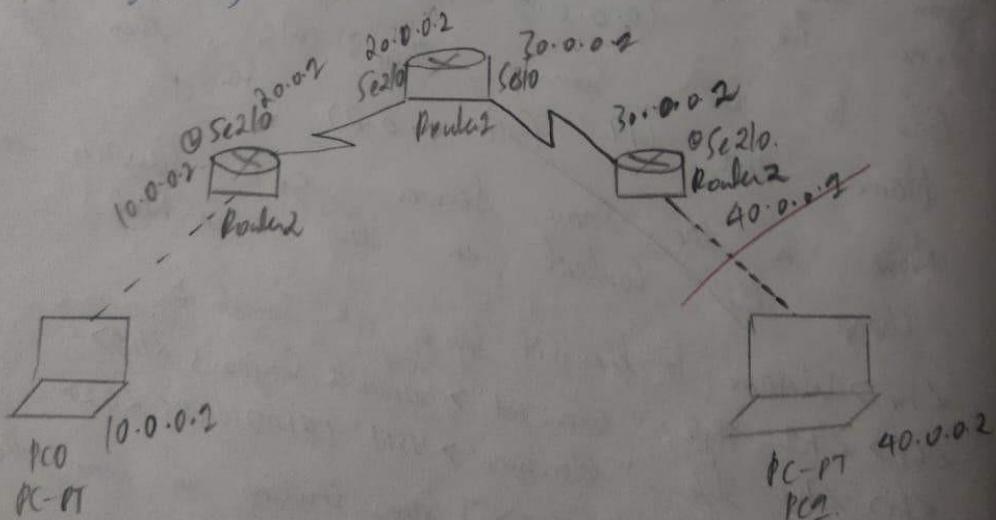
USN: 18M21CS185

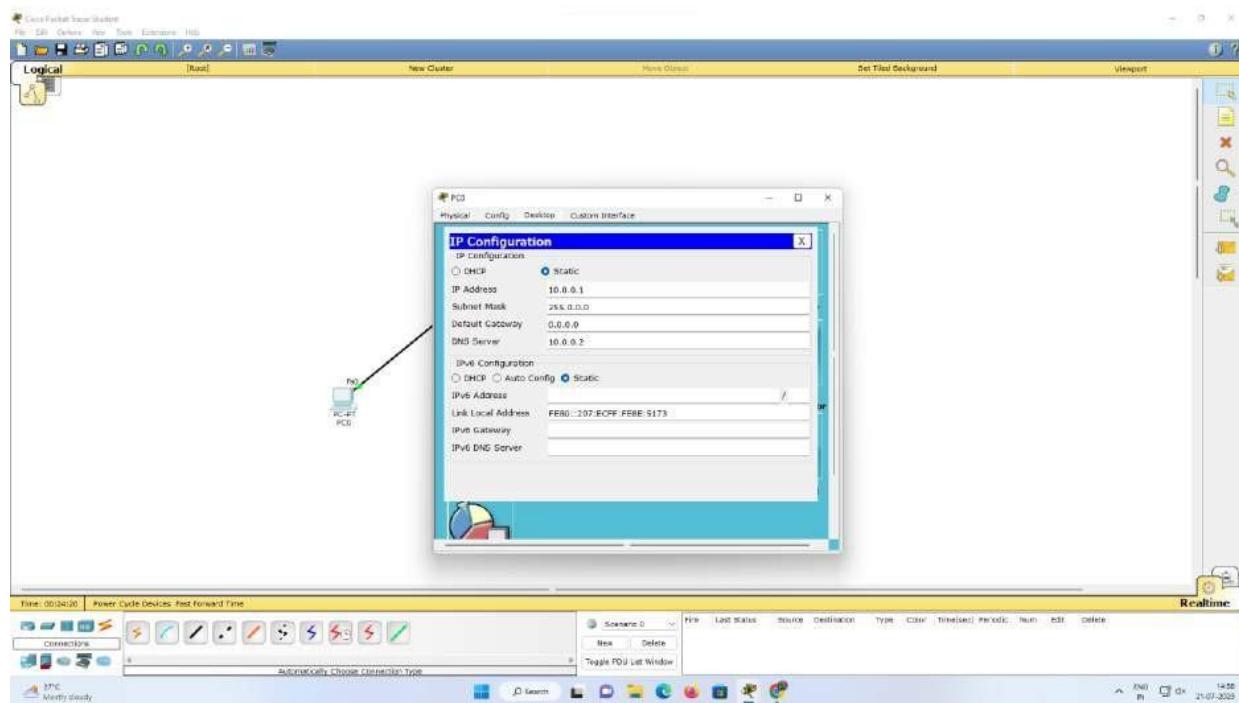
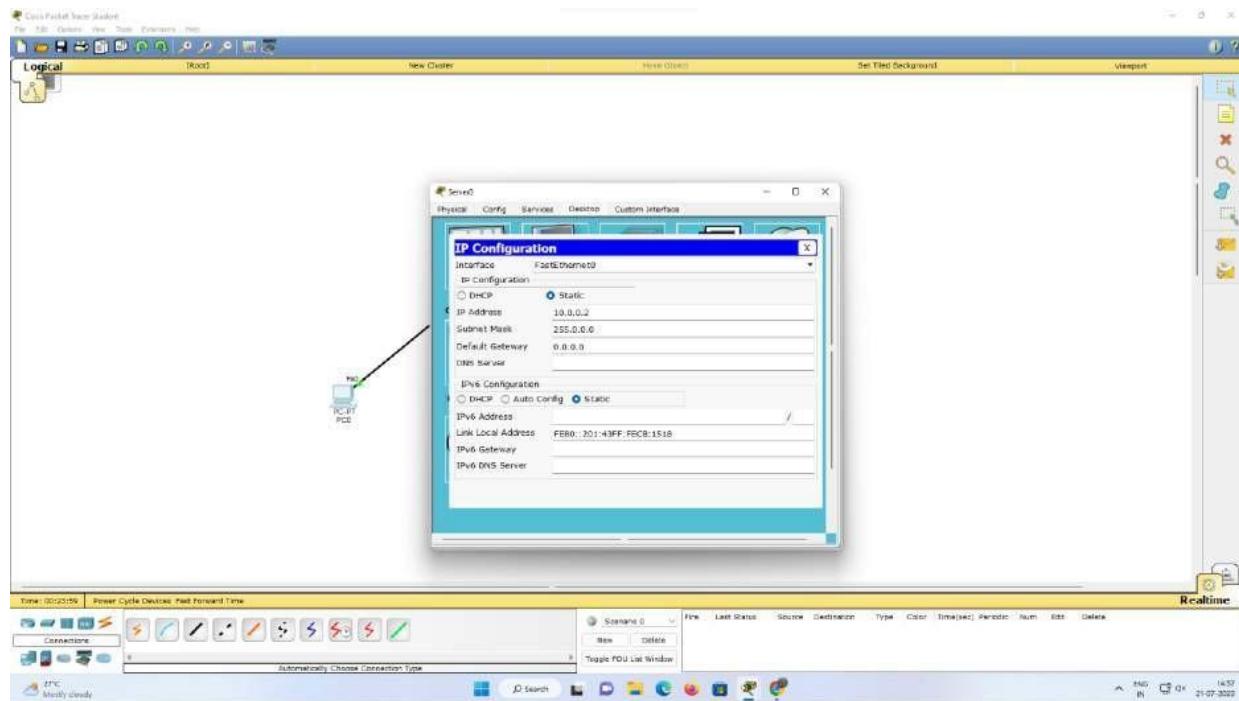
I am Studying in GMC College of Engineering.

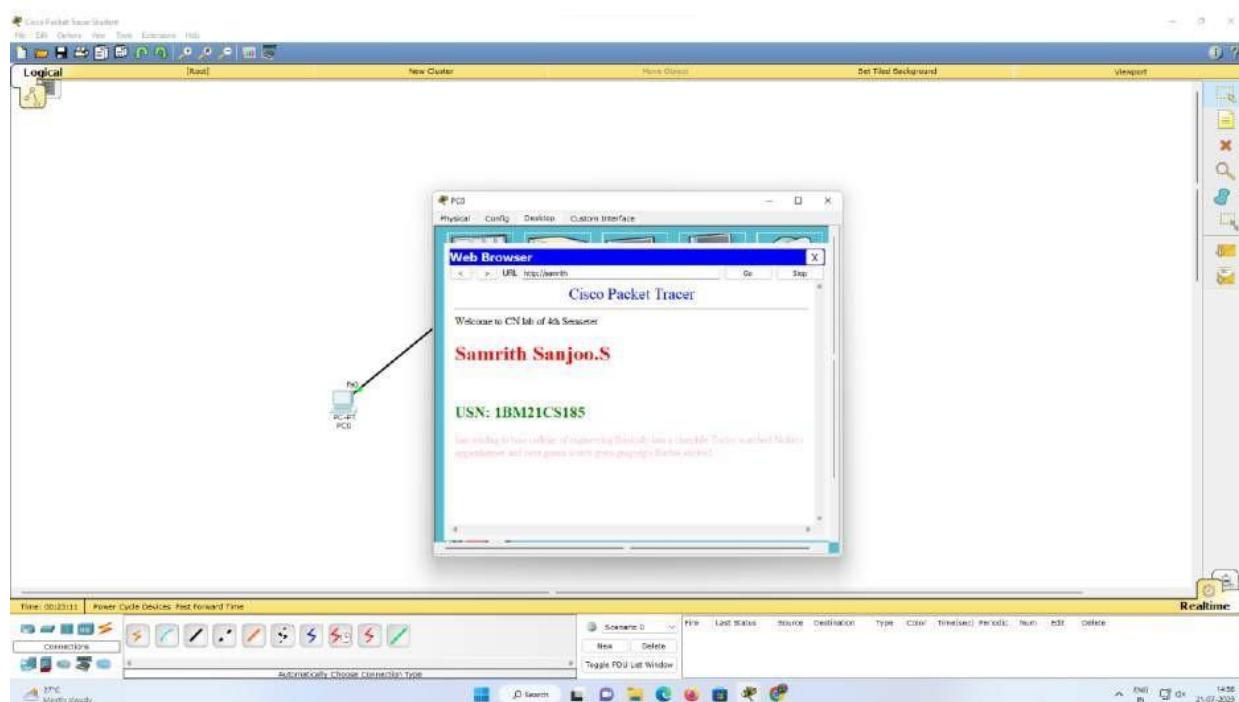
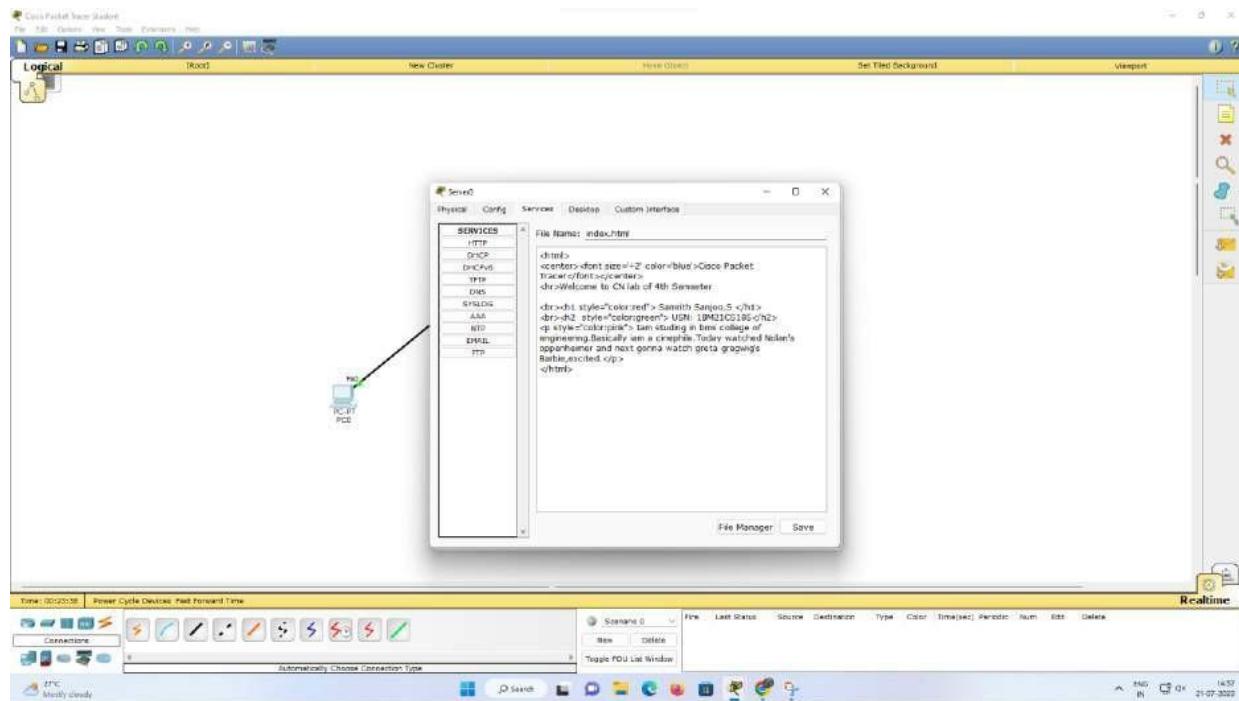
- Observation :
- a) DNS (Domain Name System) to resolve domain names, IP addresses, different servers
 - b) Can Map a distributed database in which you can map hostnames to IP addresses through the DNS protocol from a DNS server.
 - c) The above experiment is done using DNS and noted the outcome of the experiment.

AIM:

- b) RIP routing with encapsulation PPP and clock rate







Program 6

Rip routing with encapsulation ppp and clock rate:

result : Welcome to CN lab of 4th Semester

Sam with Sanyoo S

USN: 10M21CS185

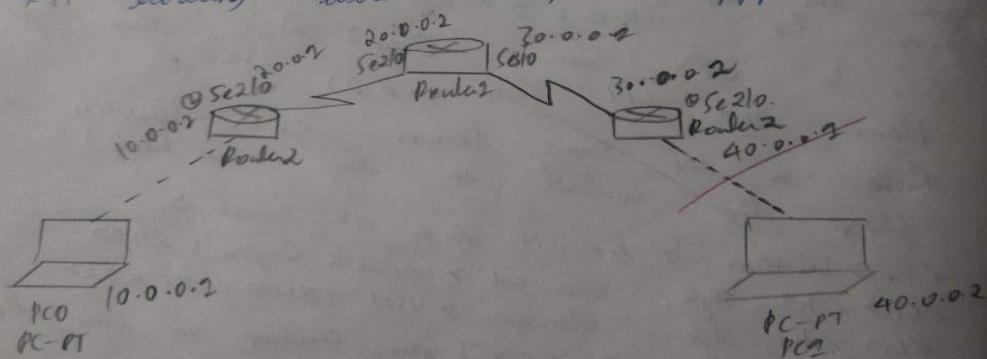
I am studying in BMC College of Engineering.

Observation : a) DNS (Domain Name System) is used to resolve domain names, IP addresses, different servers.

b) Can Map a distributed database in which you can map hostnames to IP addresses through the DNS protocol from a DNS server.

c) The above experiment is observed using DNS and noted the outcome of the experiment.

b) AIM: RIP routing with encapsulation PPP and clock rate



Router 1 Configuration

route (config) #

interface Serial 0/0

route (config) # clock rate 64000

route (config) # encapsulation PPP

route (config) # no shutdown

route (config) # interface Serial 0/0

route (config) # route ip

route (config-route) # network 0.0.0.0

route (config) # route ip

route (config) # network 20.0.0.0

route (config) # route ip

a) Router 1

route (config) # interface Serial 0/0

route (config) # clock rate 64000

route (config) # encapsulation PPP

route (config) # no shutdown

route (config) # route ip

route (config-route) # network 20.0.0.0

route (config) # route ip

b) Router 2

route (config) # route ip

route (config-route) # network 40.0.0.0

route (config) # route ip

Output : Ping from PC1 to PC2 : 10.0.0.1 to 10.0.0.2

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 = 2ms TTL = 125

Reply from 10.0.0.2 bytes = 32 time = 2ms TTL = 125

Reply from 10.0.0.2 bytes = 32 time = 2ms TTL = 125

Reply from 10.0.0.2 bytes = 32 time = 8ms TTL = 125

Ping statistics for 10.0.0.2

Packet: Sent = 4, Received = 4, Lost = 0 (0.0% loss),

Approximate round trip time in milli-seconds:

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

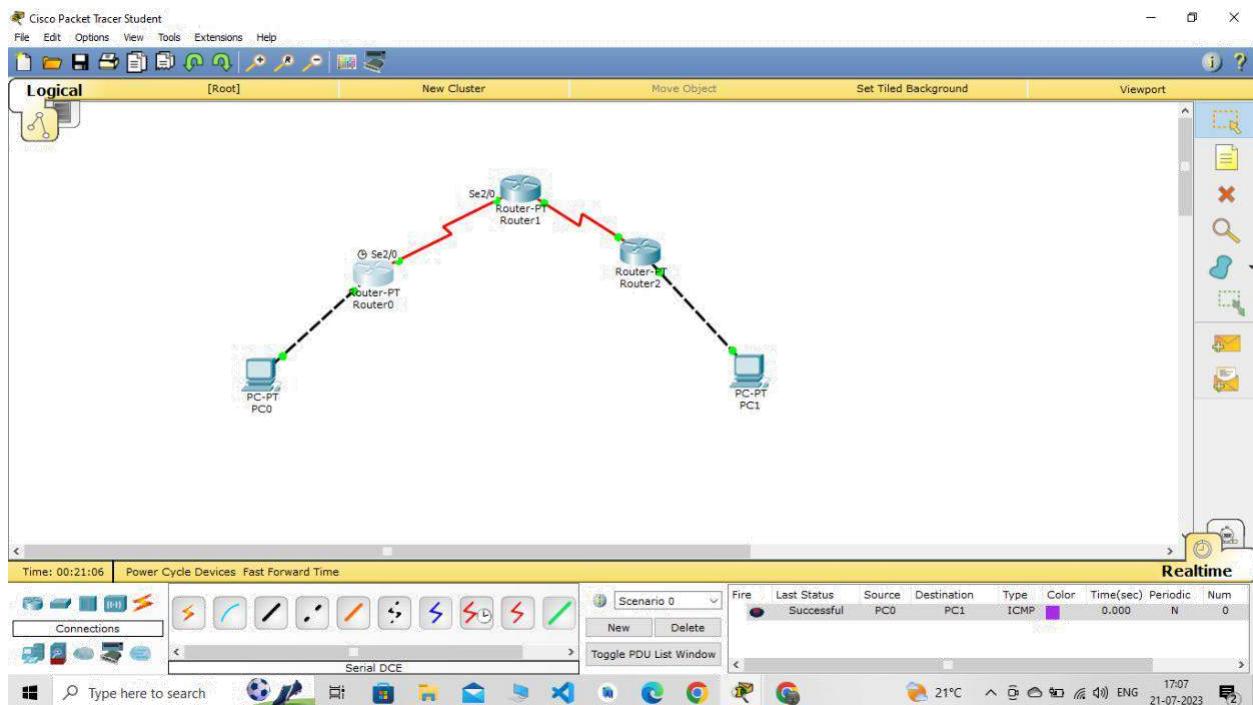
PC >

Observation :

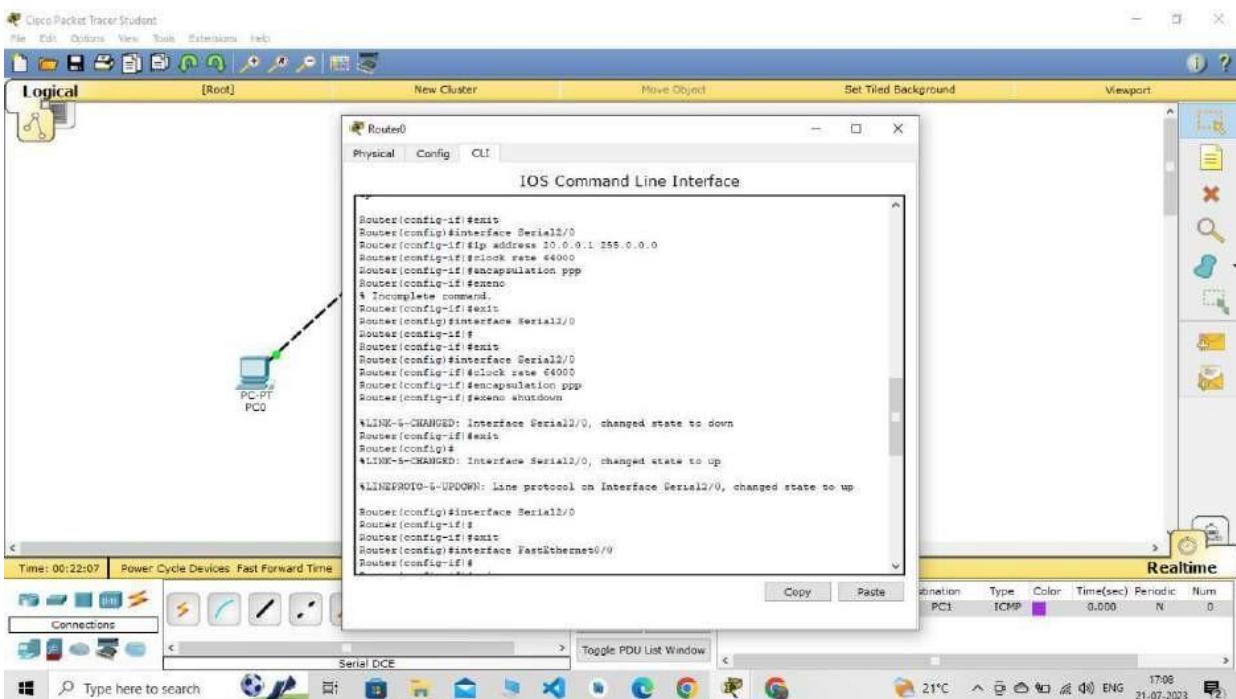
- # RIP defines how routers should share information when moving traffic among an interconnected group of local networks.

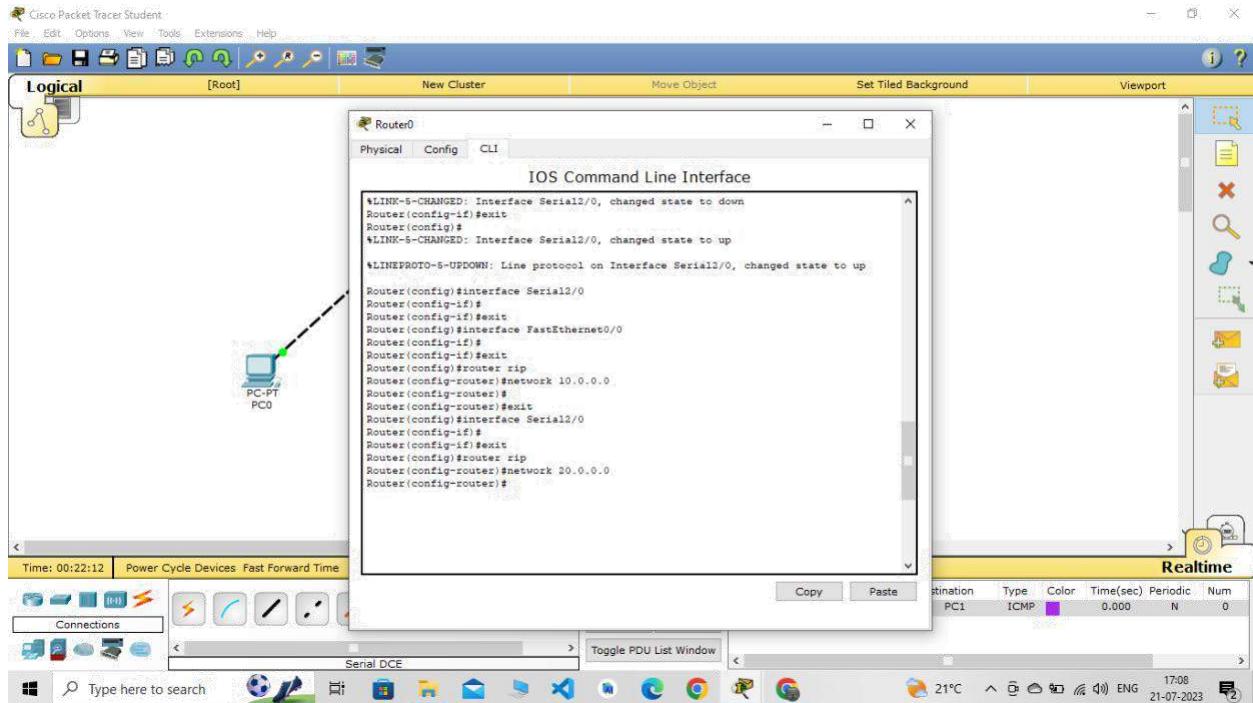
~~- # A coke experiment is done using commands such as encapsulation PPP and rip commands. and the outcomes are noted.~~

Ques

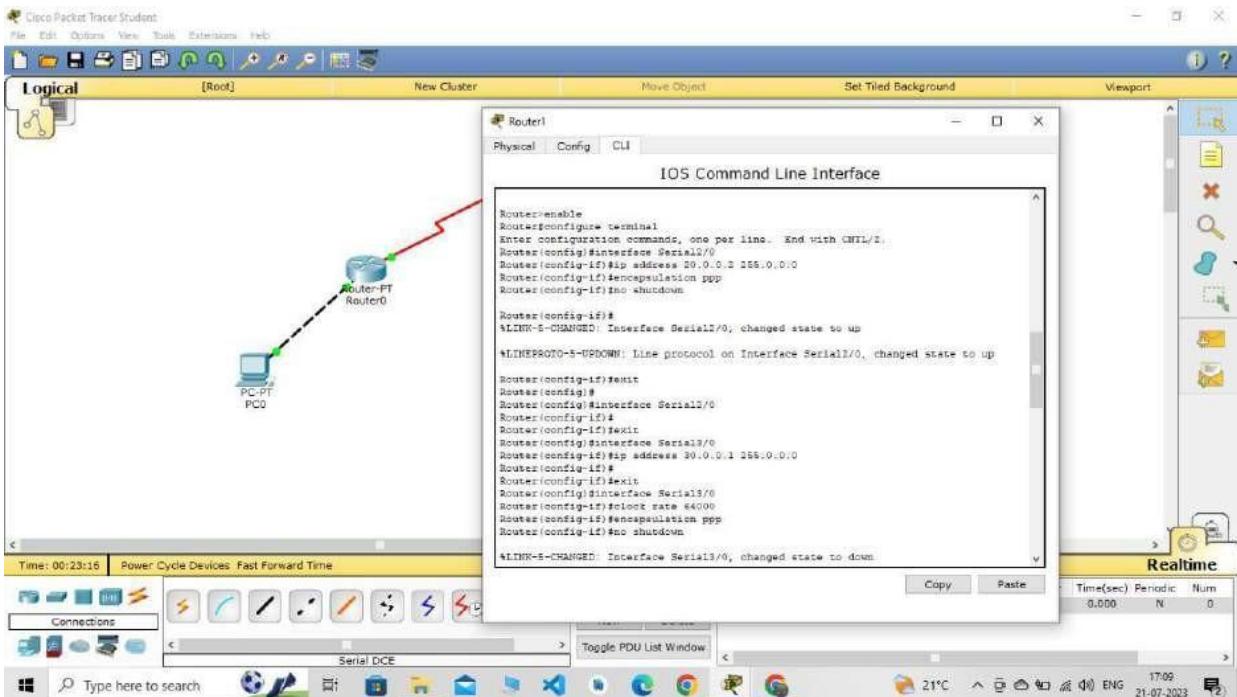


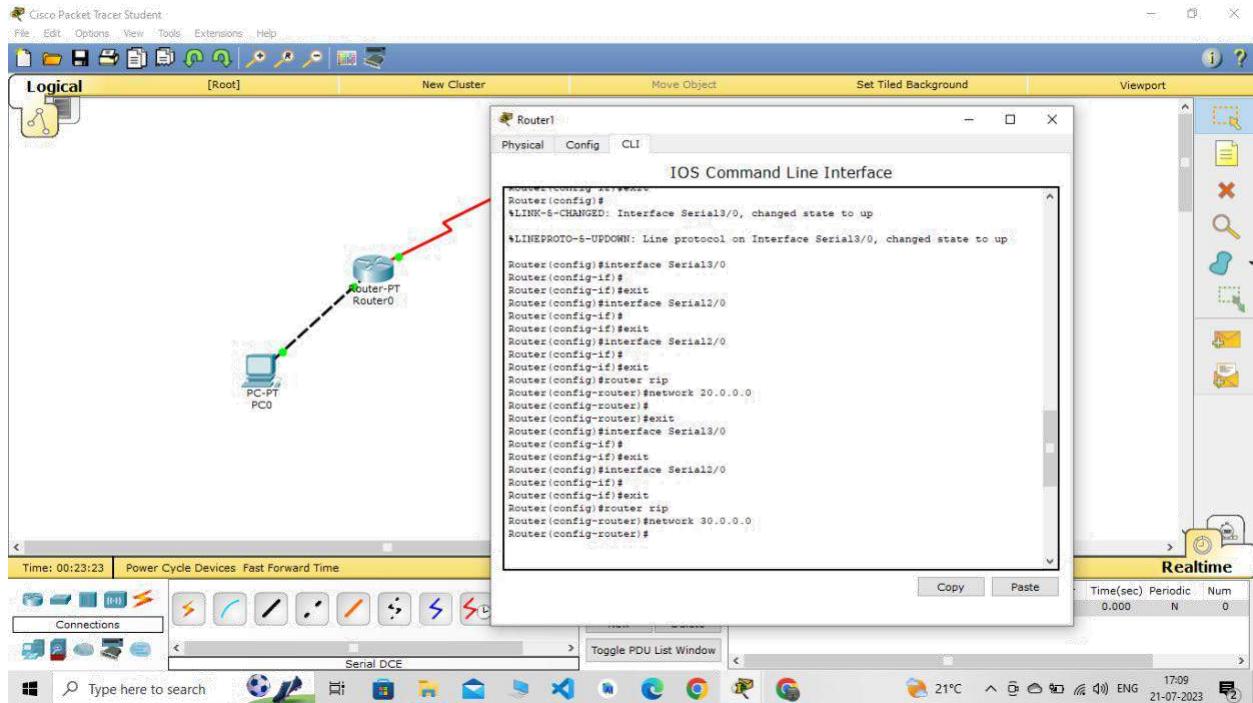
Router 1:



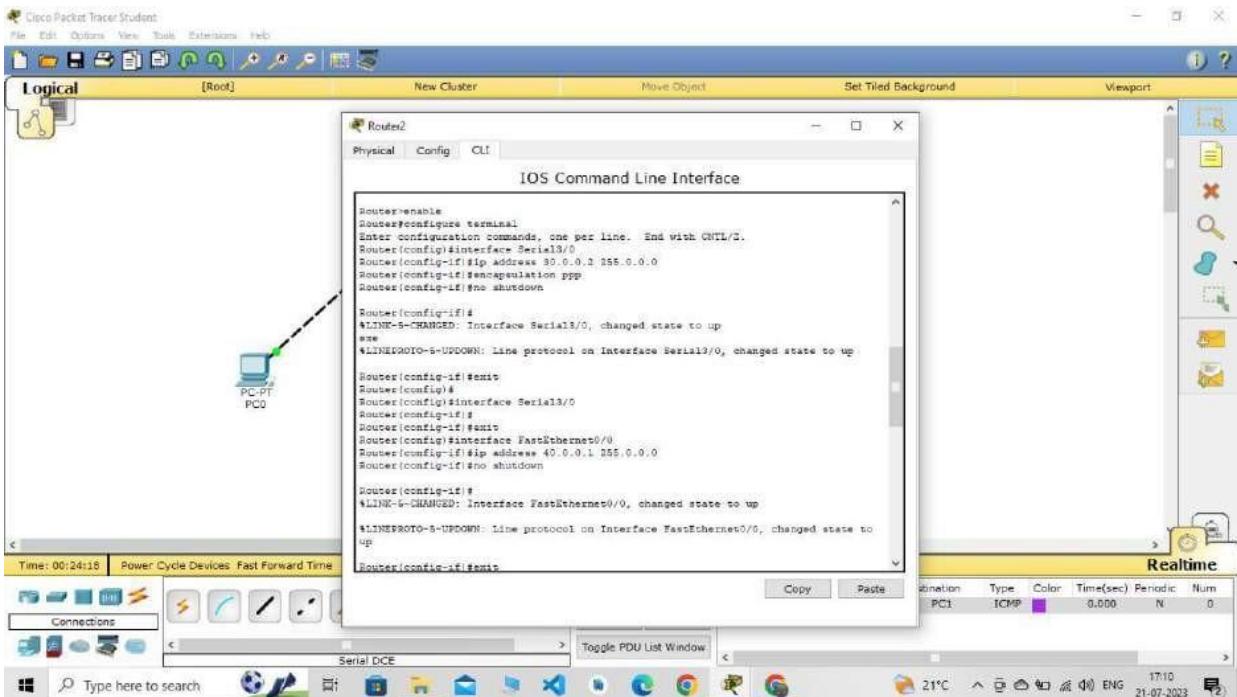


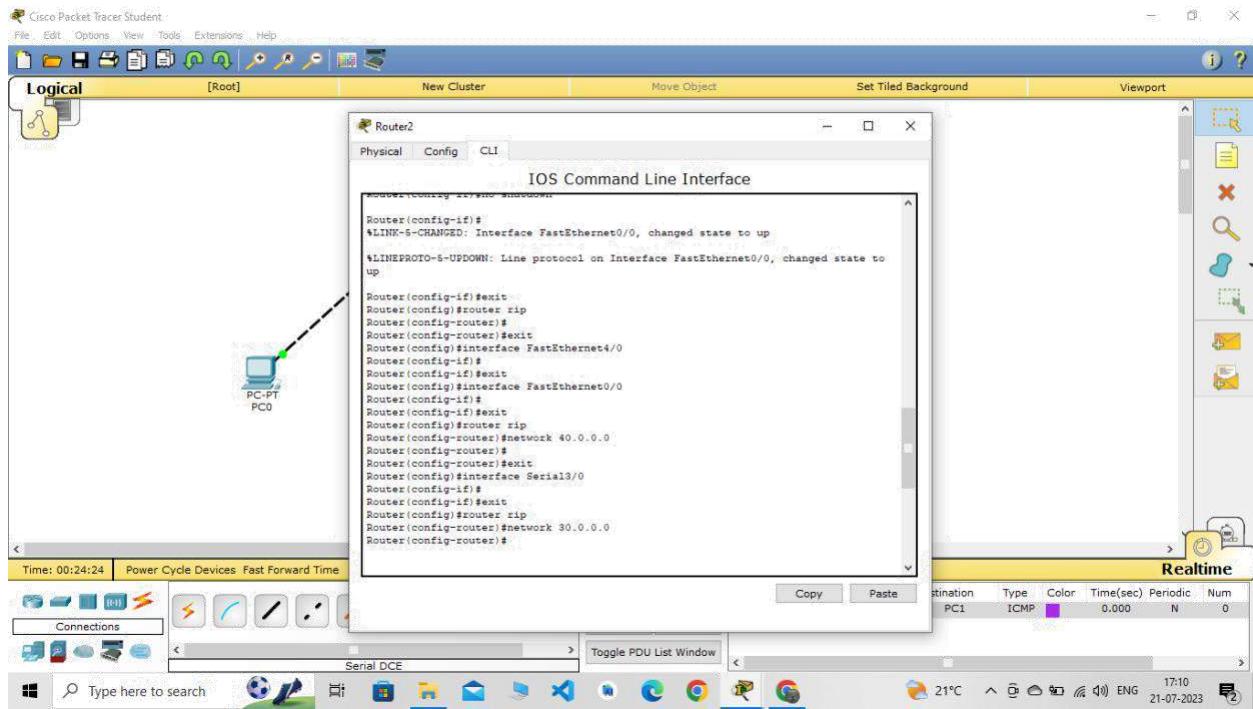
Router 2:



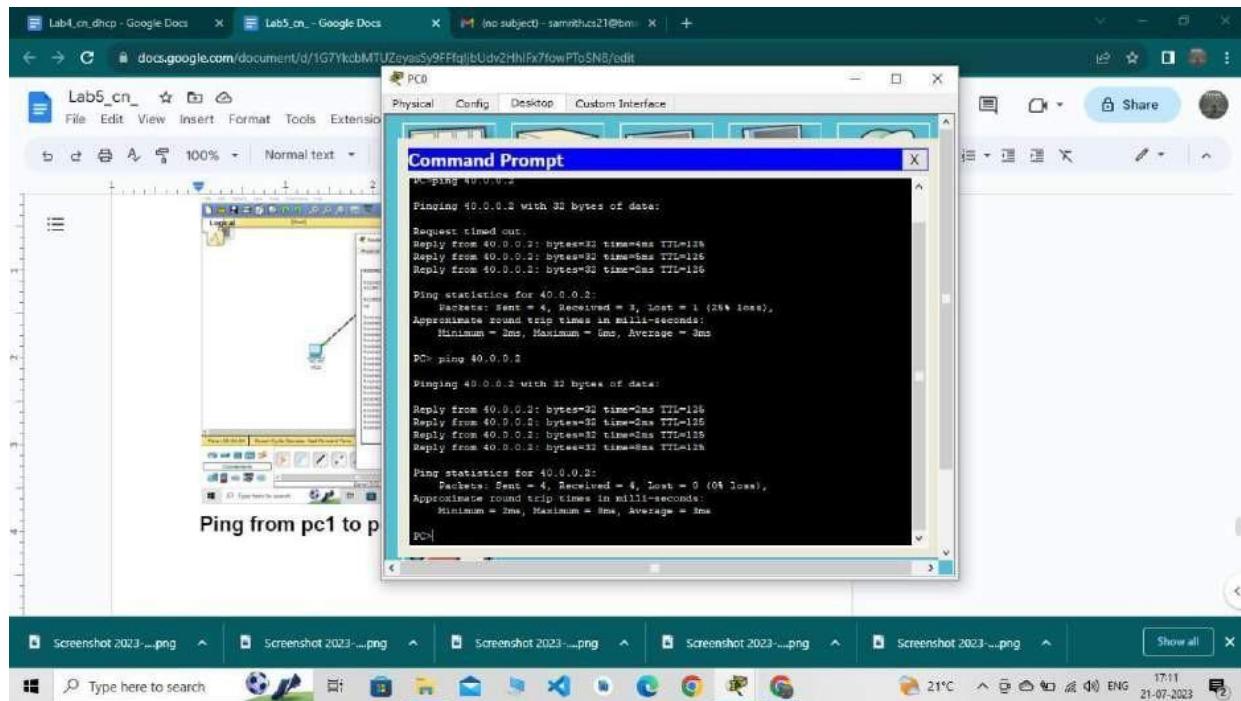


Router 3:





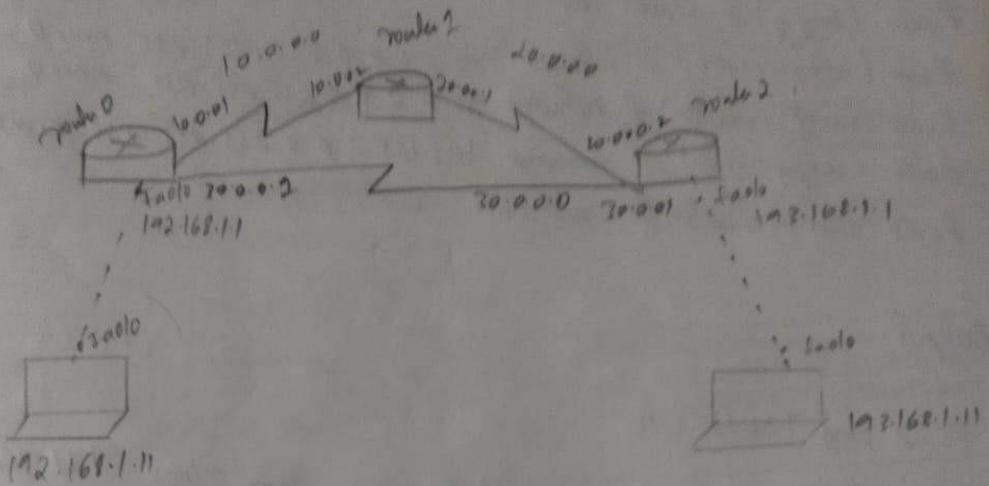
Ping from pc1 to pc2:10.0.0.1 to 40.0.0.2



Program 7

Configuring OSPF commands

Q) Configuring OSPF Protocol and its Commands



AIM: To find the shortest route using ospf command

Working: Router 0 :

Router (config)# router ospf 1

Router (config-router)# network 192.168.1.0 0.0.0.255 area 0

Router (config-router)# network 10.0.0.0 0.255.255.255 area 0

Router (config-router)# network 20.0.0.0 0.255.255.255 area 0

Router (config-router)# exit

Router 2

Router (config)# router ospf 2

Router (config-router)# network 10.0.0.0 0.255.255.255 area 0

Router (config-router)# network 20.0.0.0 0.255.255.255 area 0

Router (config-router)# exit

Router 2

Router (config) # router ospf 3

Router (config-router) # network 20.0.0.0 0.255.255.255 area 0

Router (config-router) # network 30.0.0.0 0.255.255.255 area 0.

Router (config-router) # network 193.168.1.0 0.255.255.255 area 0

Router (config-router) # exit

Output :

PC > ping 193.168.1.11

Reply from	193.168.1.11	bytes = 32	time = 1ms	TTL=126
Reply from	193.168.1.11	bytes = 32	time = 1ms	TTL=126
Reply from	193.168.1.11	bytes = 32	time = 6ms	TTL=126
Reply from	193.168.1.11	bytes = 32	time = 3ms	TTL=126

Ping statistics for 193.168.1.11

Packet : Sent = 4 , Received = 4 , Lost = 0 , (0 % loss),

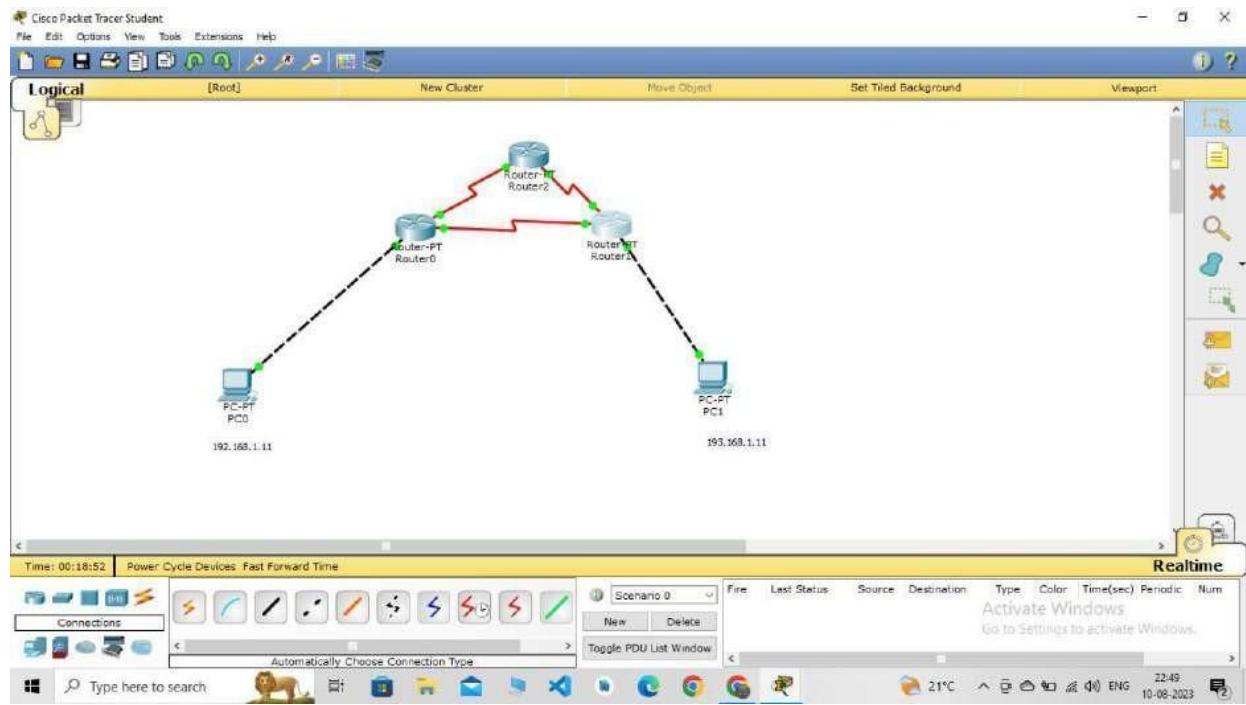
Approximate round trip times in milliseconds

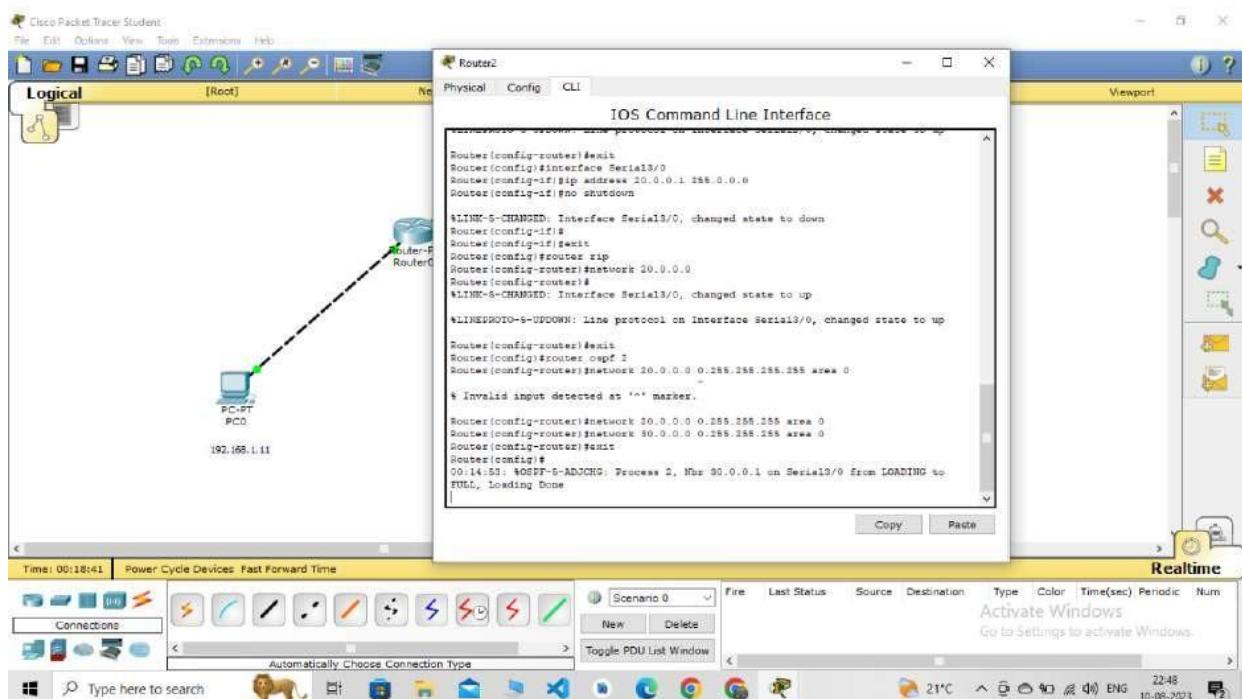
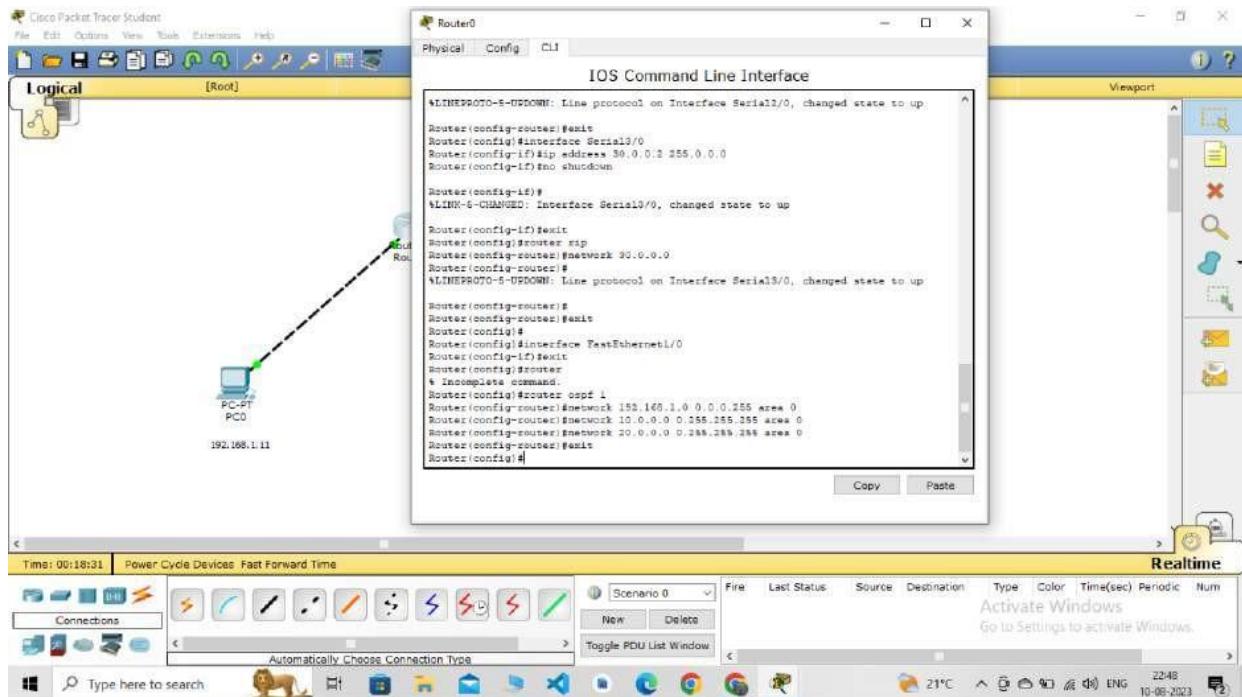
Minimum = 1ms , Maximum = 6ms , Average = 2ms .

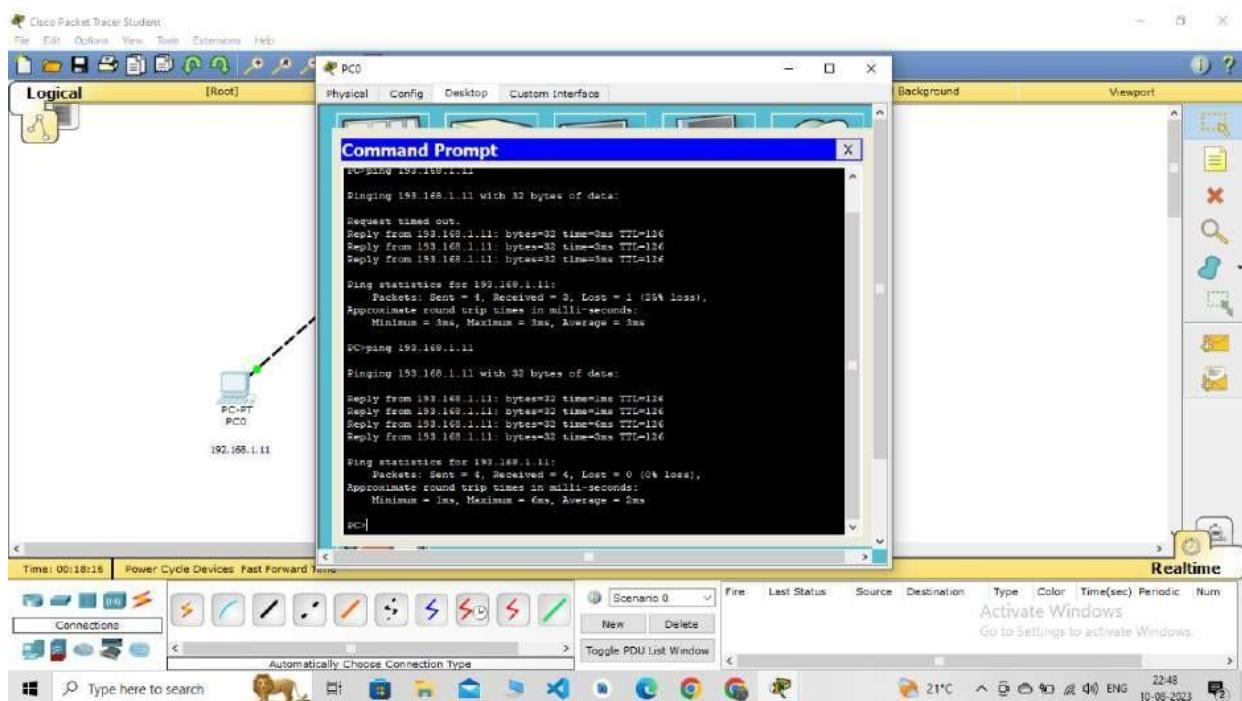
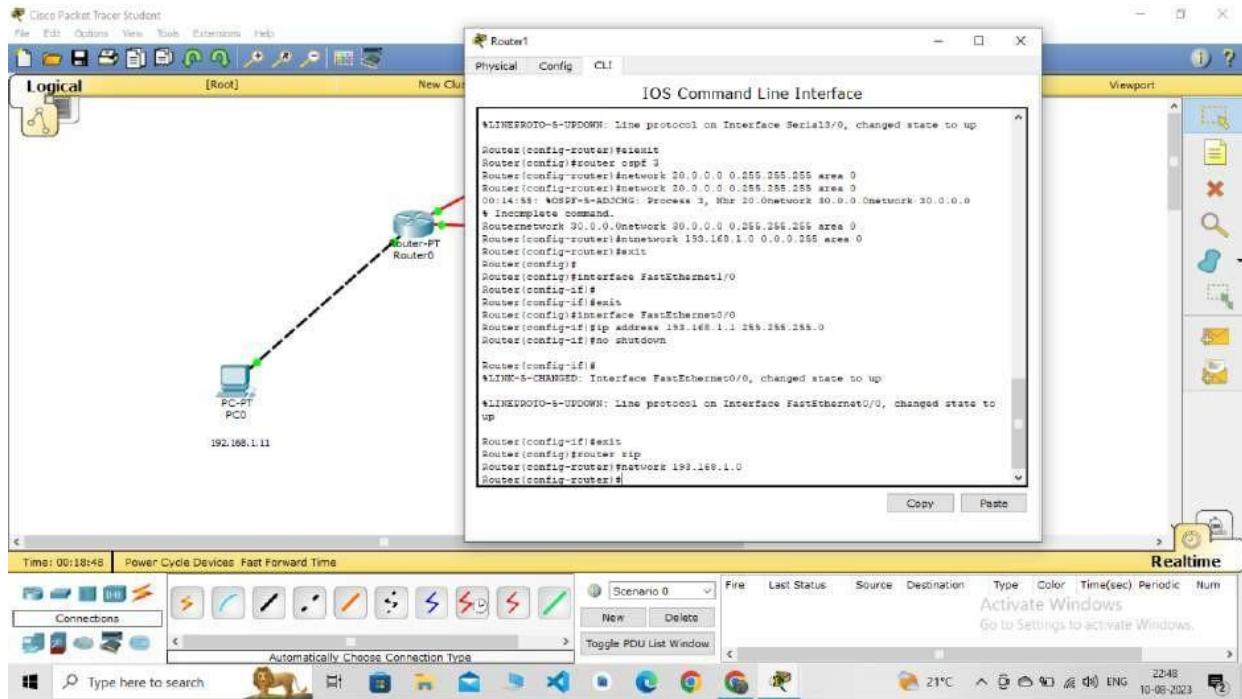
Observation

a) Utilization of areas in the network carried out by OSPF routing work well, so that the information dissemination becomes faster and more efficient.

b) Low recalculate the route in a short amount of time when the network topology changes.

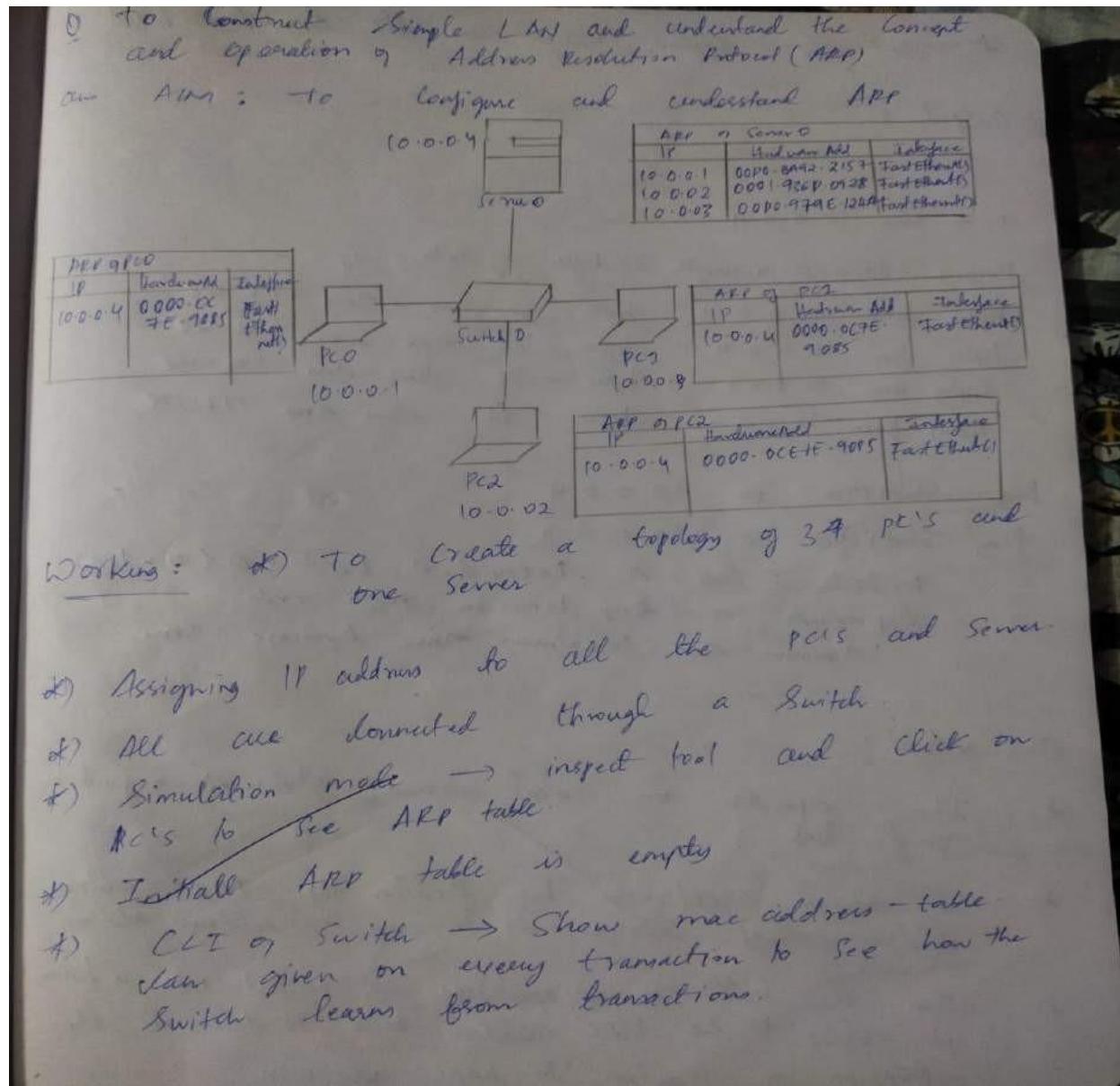






Program 9

Configuring ARP commands



d) Use the capture button in the Simulation panel

Output :

PC2

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 = 9ms TTL=128

Reply from 10.0.0.4 bytes = 32 time = 4ms TTL=128

Reply from 10.0.0.4 bytes = 32 time = 4ms TTL=128

Reply from 10.0.0.4 bytes = 32 time = 9ms TTL=128

Ping Statistics for 10.0.0.4

Ping Statistics for 10.0.0.4

Packets : Sent = 4 , Received = 4 , Lost = 0 (0.0% loss)

Approximate round trip times in milliseconds:

Minimum = 4ms , Maximum = 9ms . Average = 5ms.

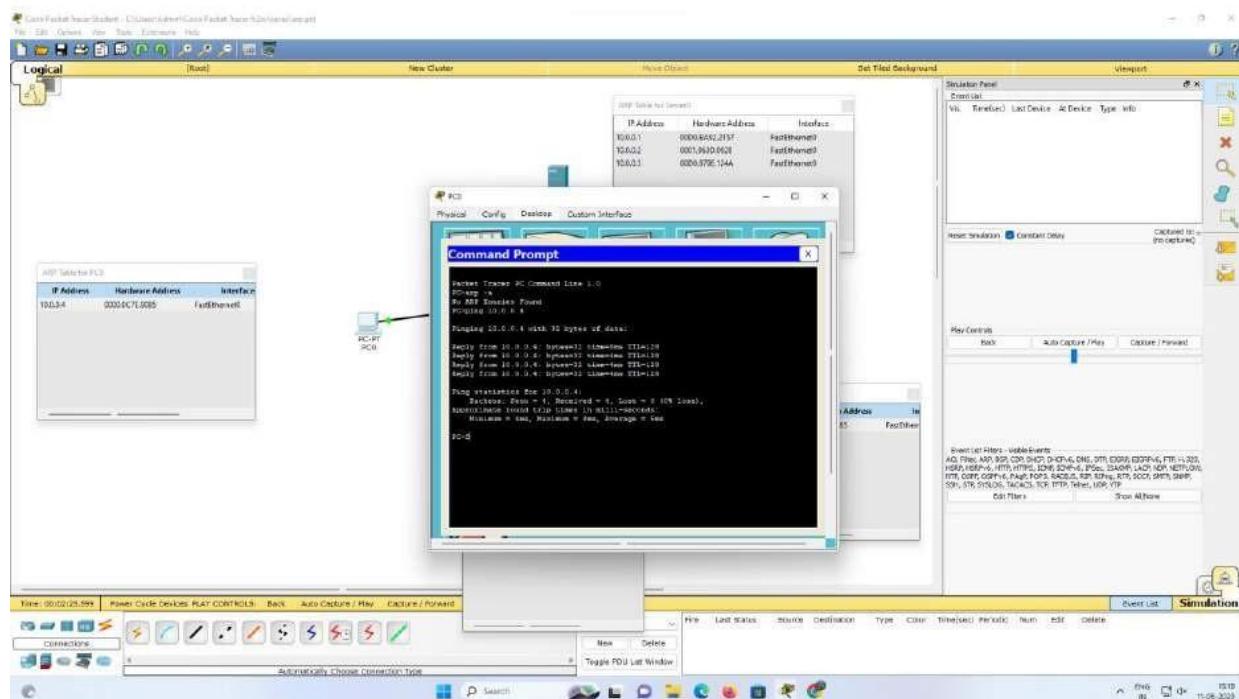
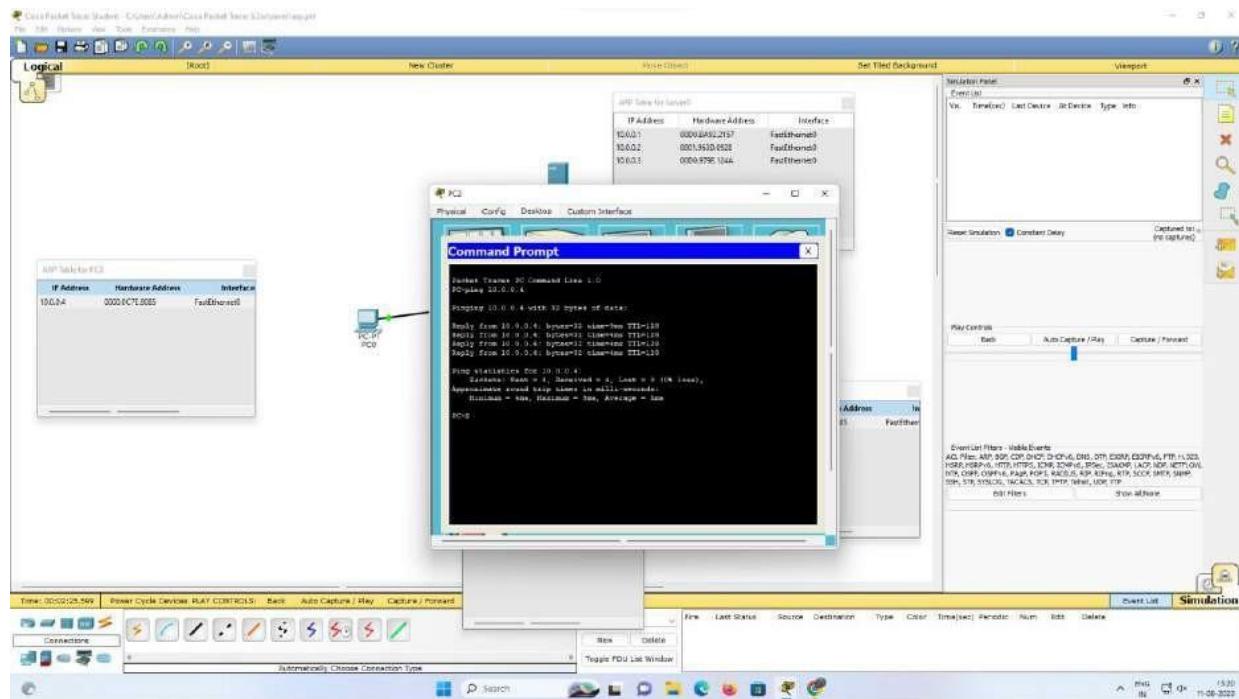
Observation :

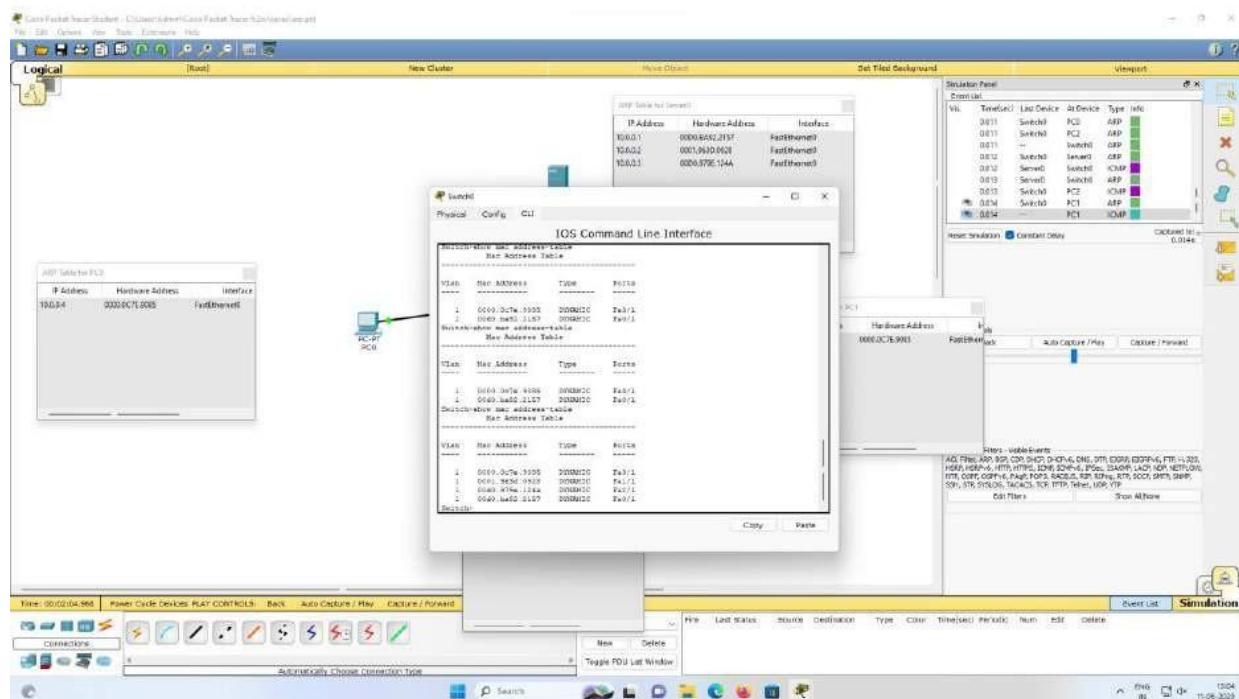
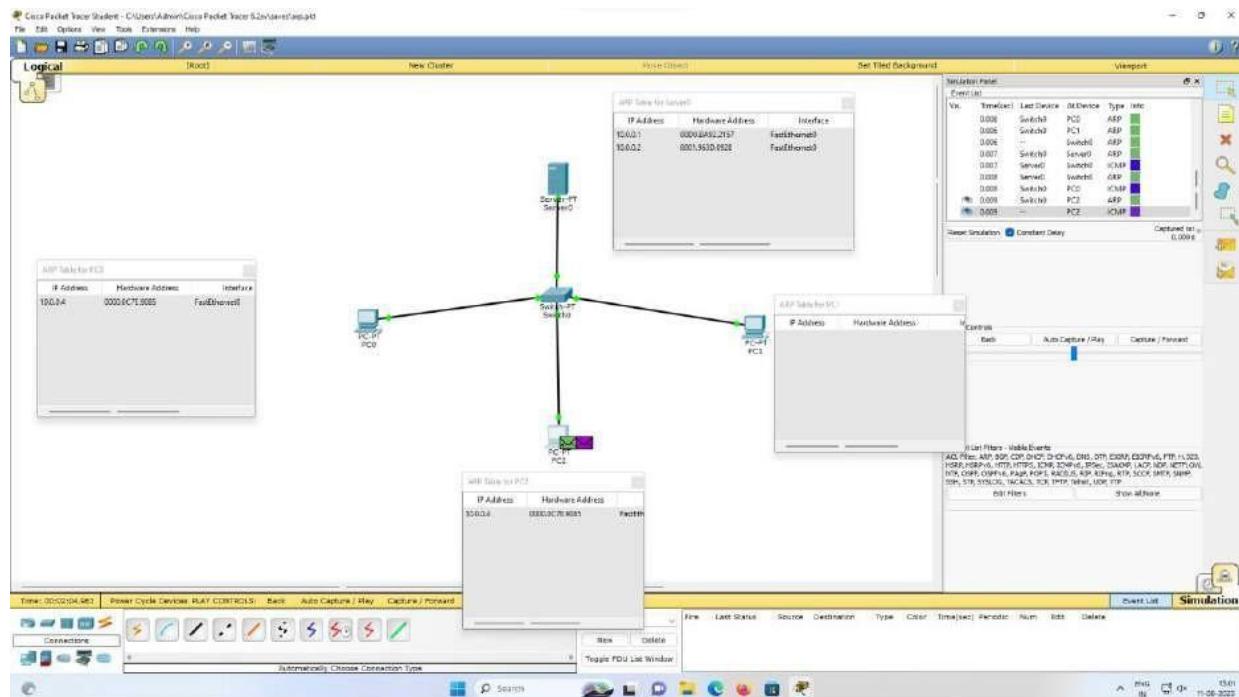
i) It operates on local area Network or point to point link.

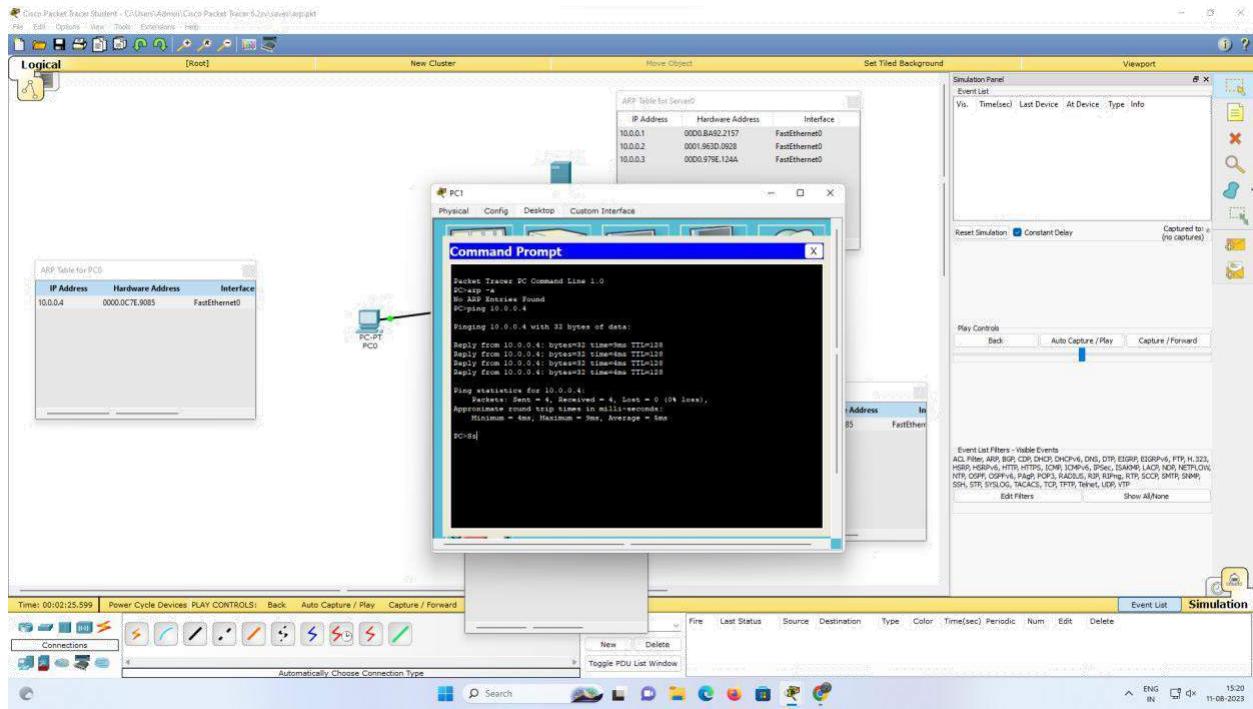
ii) to resolve address by finding the MAC address that corresponds to an IP address address.

When a host is encapsulating a packet into a frame it refers to the MAC address table to determine the destination MAC address.





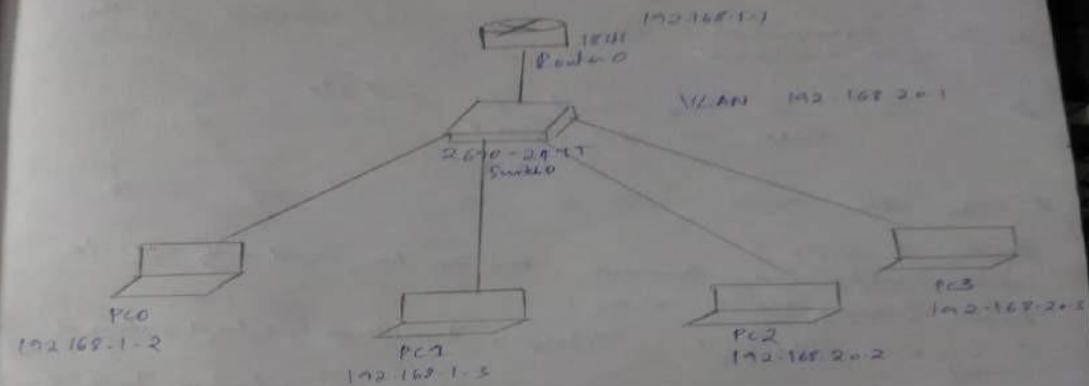




Program 10

VLAN EXPIREMENT

- Q) To construct a VLAN and make PCs communicate among
Ans. To configure and understand VLAN connections



Working

- A Router and 4 PCs are connected with a Switch.
 - The following IPs addresses are given to corresponding PCs and routers.
 - Click on Switch and VLAN configuration.
- VLAN Number : 20
 VLAN Name : LAB1

* Router #

VLAN Number 20
VLAN NAME lab2

* The corresponding entry will be added in the VLAN Database.

* Click on FastEthernet 0/3 and 0/4
VLAN to 20 (lab 2)

* In the router.

Router # config t

Enter configuration commands, one per line. End with CNTL/Z

Router (config) # int f 0/0.1

1. LINK-5 CHANGED : Interface FastEthernet.0/0.1, changed up.

1. LINEPROTO-5-UPDOWN : Line protocol on Interface FastETH
Changed State to up.

Router (config-subif) # encapsulation dot1q 20

Router (config-subif) # ip address 192.168.20.1 255.255.255

Router (config-subif) # exit

PC > ping 192.168.20.2.

Pinging 192.168.20.2 with 32 bytes of data:

Reply from 192.168.20.2 bytes=32 time=9ms TTL=128

Reply from 192.168.20.2 bytes=32 time=4ms TTL=128

Reply from 192.168.20.2 bytes=32 time=4ms TTL=128

Reply from 192.168.20.2 bytes=32 time=4ms TTL=128

Ping Statistics for 192.168.20.2

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

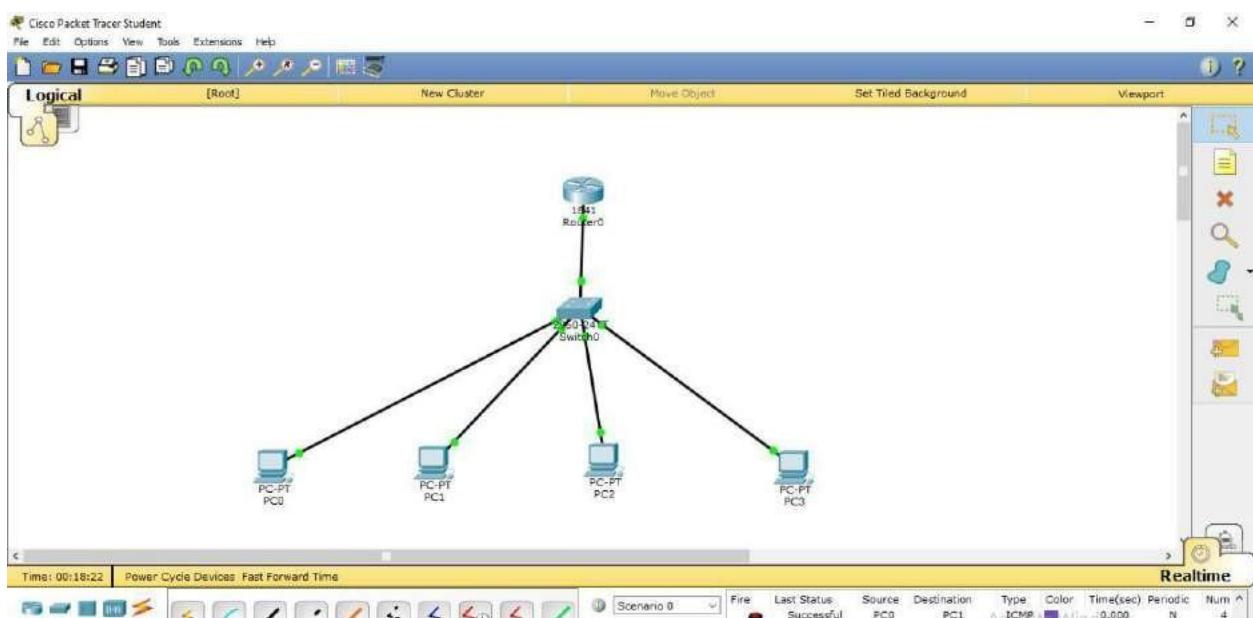
Approximate round trip times in milliseconds:

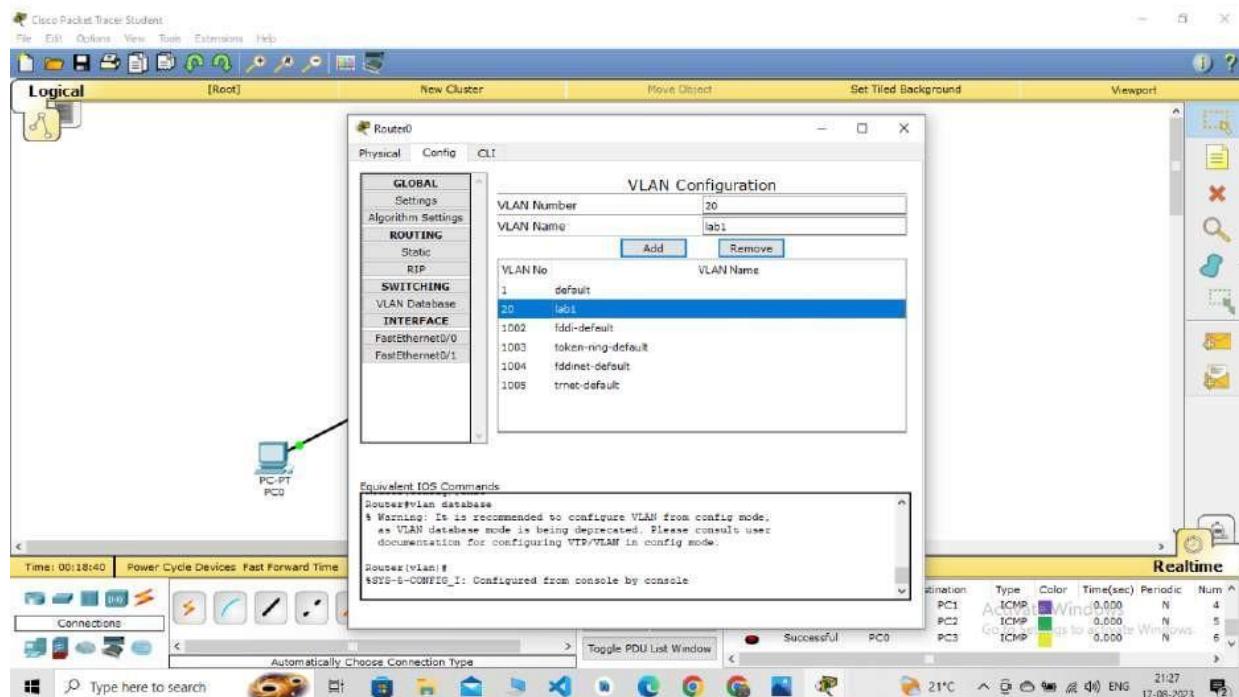
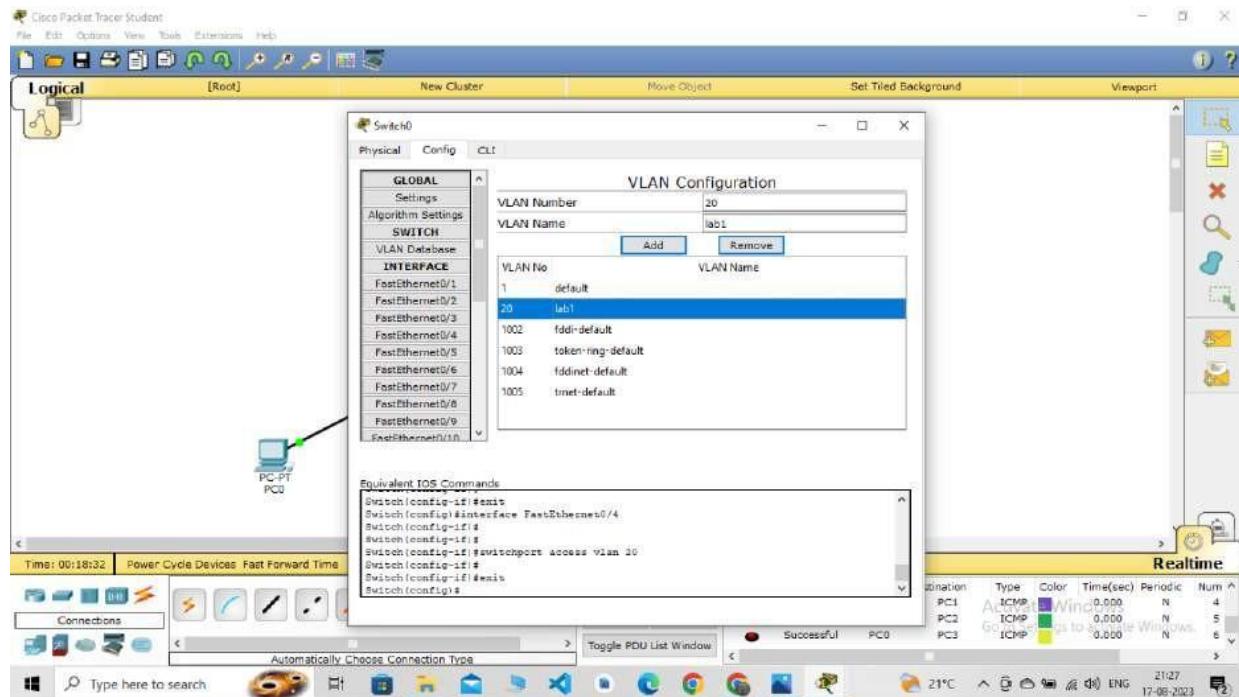
Minimum = 4ms, Maximum = 9ms, Average = 5ms.

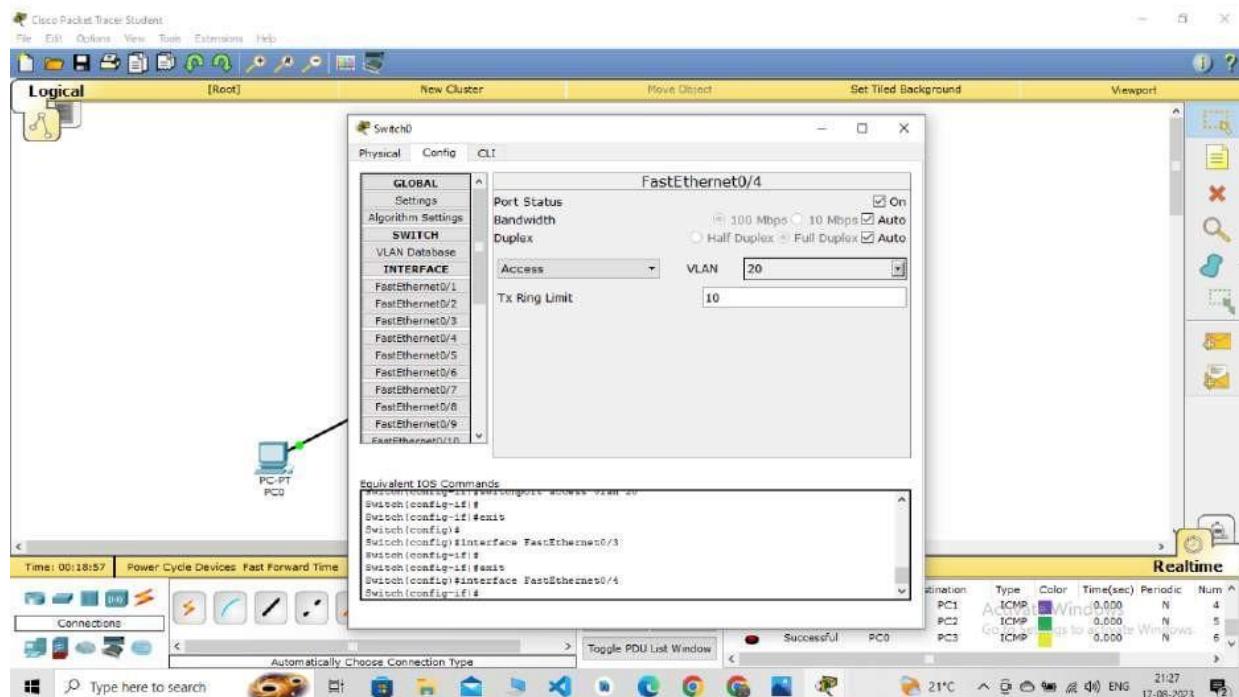
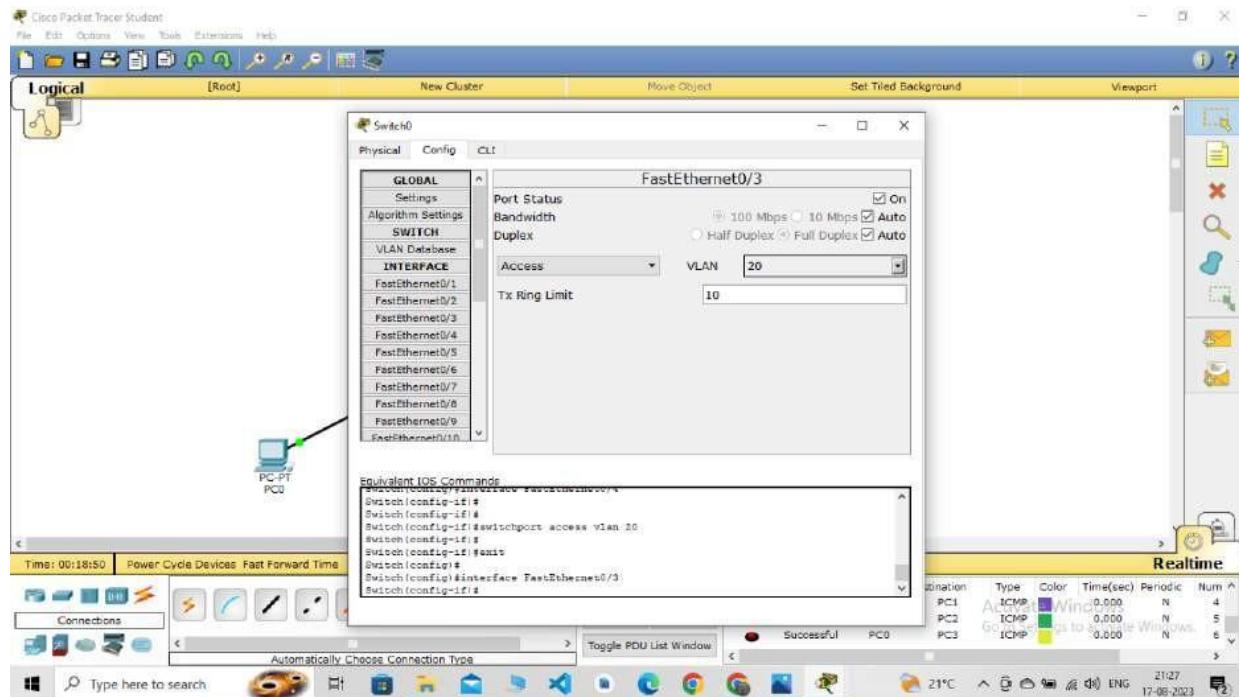
Observations :-

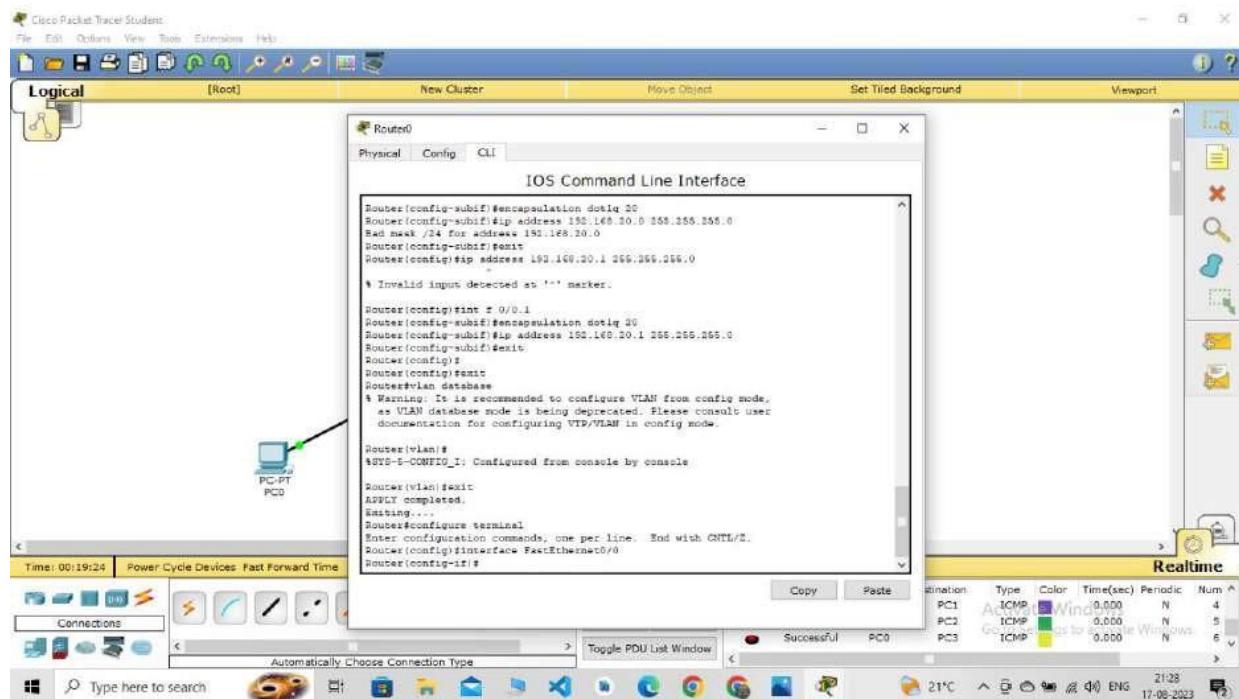
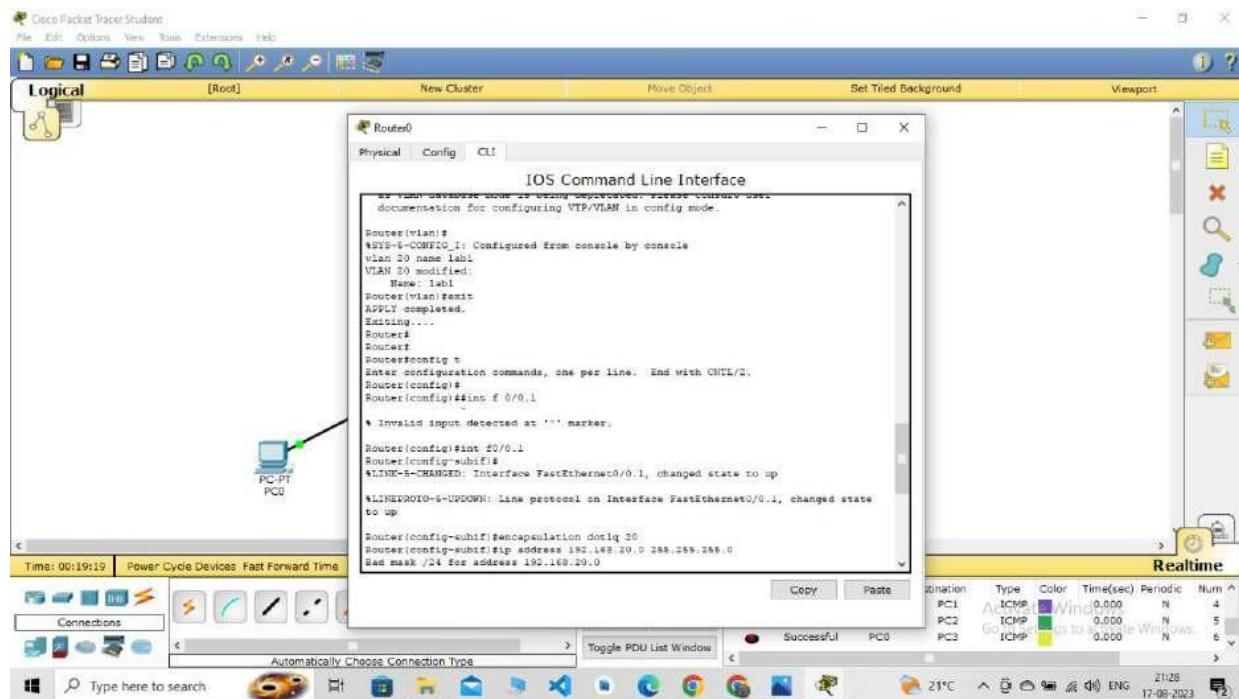
- a) VLAN reduces the incidence of collisions and decreases the no. of network resources wasted by acting as LAN Segments
- * to divide the network into separate logical areas at layer 2 level
- b) Hosts can be kept separated in a VLAN
- b) The above VLAN experiment is done and noted its observations and results.

By
Rajiv









Program 11

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

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



Working :

- a) Above topology is set up with a PC and 1941 router.
- b) From the command point of the router (CLI)
 - # ipaddr 192.168.1.1 255.255.255.0
 - # Router (config-if) # line vty 0 5
 - Router (config-line) # login
 - * Login disabled on line 132, until 'password' is set
 - * Login disabled on line 133, until 'password' is set
 - * Login disabled on line 134, until 'password' is set
 - * Login disabled on line 135, until 'password' is set
 - * Login disabled on line 136, until 'password' is set
 - * Login disabled on line 137, until 'password' is set

Router (Config) # exit
Router (Config) # enable password admin@31
Router (Config) # exit
Router #
- S85-S-Config-T: Configured from Console key console

Output :

PC > ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:

Reply from 192.168.1.1	bytes = 32	time = 0ms	TTL = 255
Reply from 192.168.1.1	bytes = 32	time = 0ms	TTL = 255
Reply from 192.168.1.1	bytes = 32	time = 0ms	TTL = 255
Reply from 192.168.1.1	bytes = 32	time = 0ms	TTL = 255

Ping statistics for 192.168.1.1

Packets: Sent = 4, Received = 4, Loss = 0 (0% loss).

Approximate round-trip times in milliseconds:

Minimum round-trip = 0ms, Maximum = 0ms, Average = 0ms.

PC > telnet 192.168.1.1

Trying 192.168.1.1... OPEN

VSys Access Verification

Password : admin@31 (unstated)

Table II Shows 17 routes

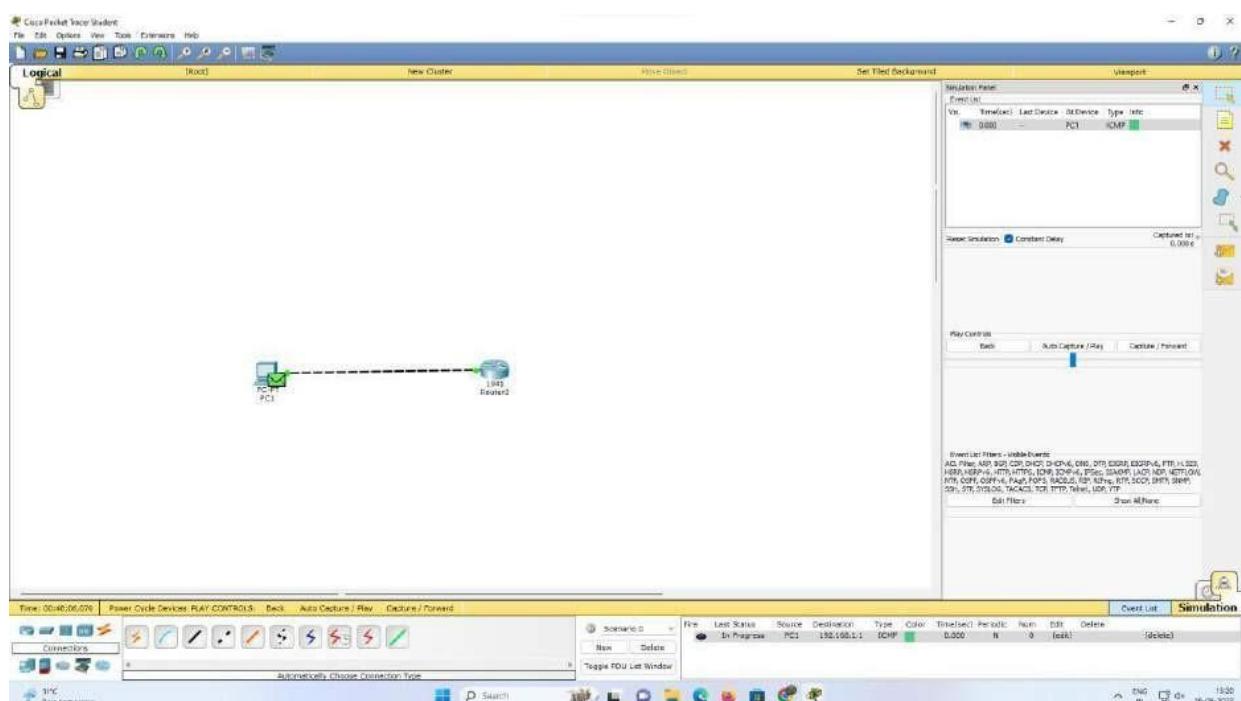
Color L -
P -
NT -
C -

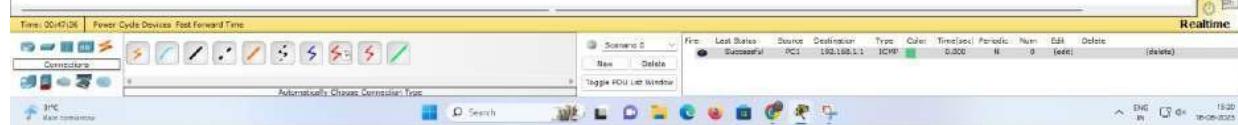
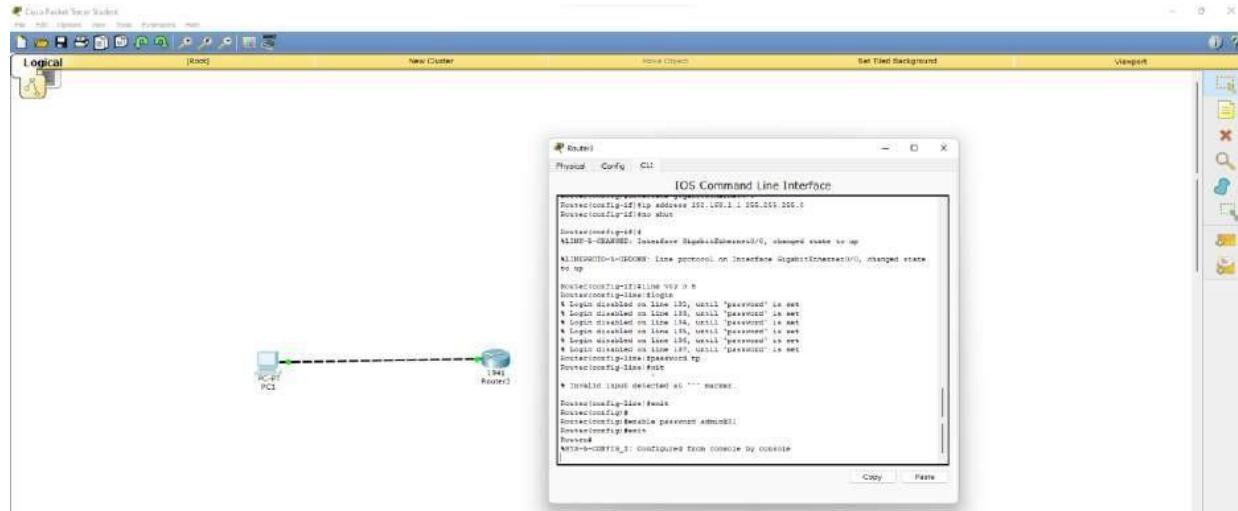
area 1 -

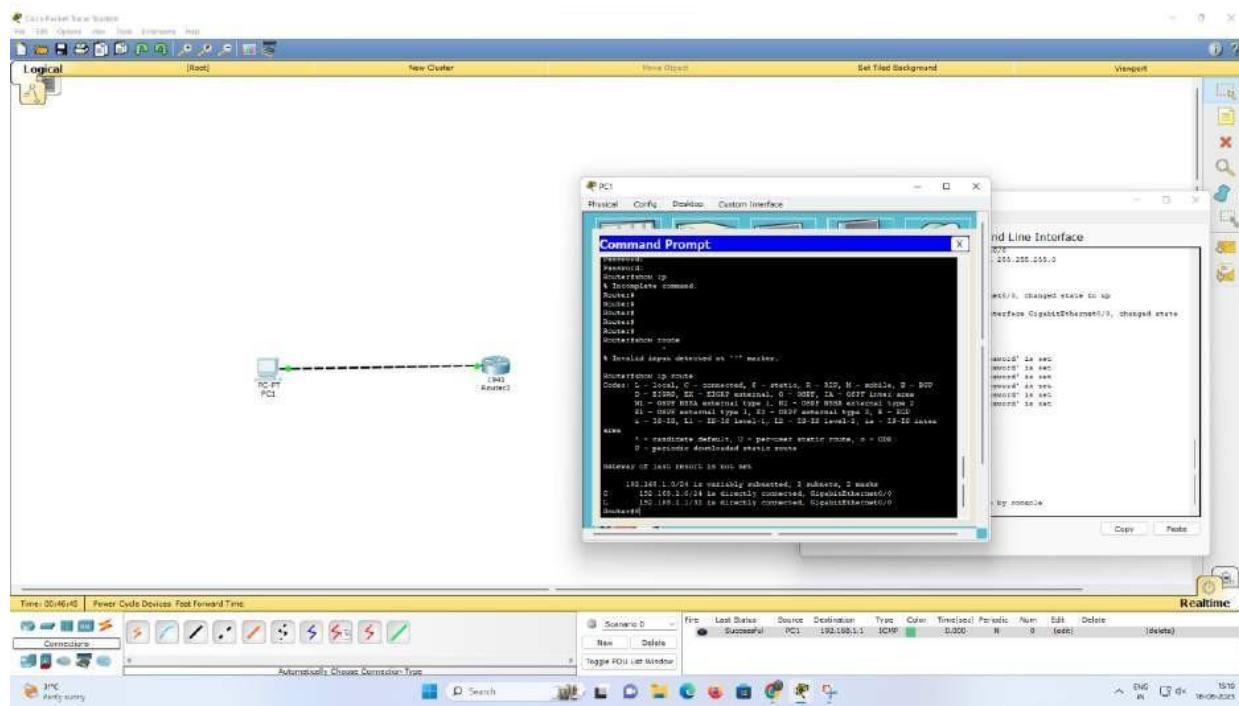
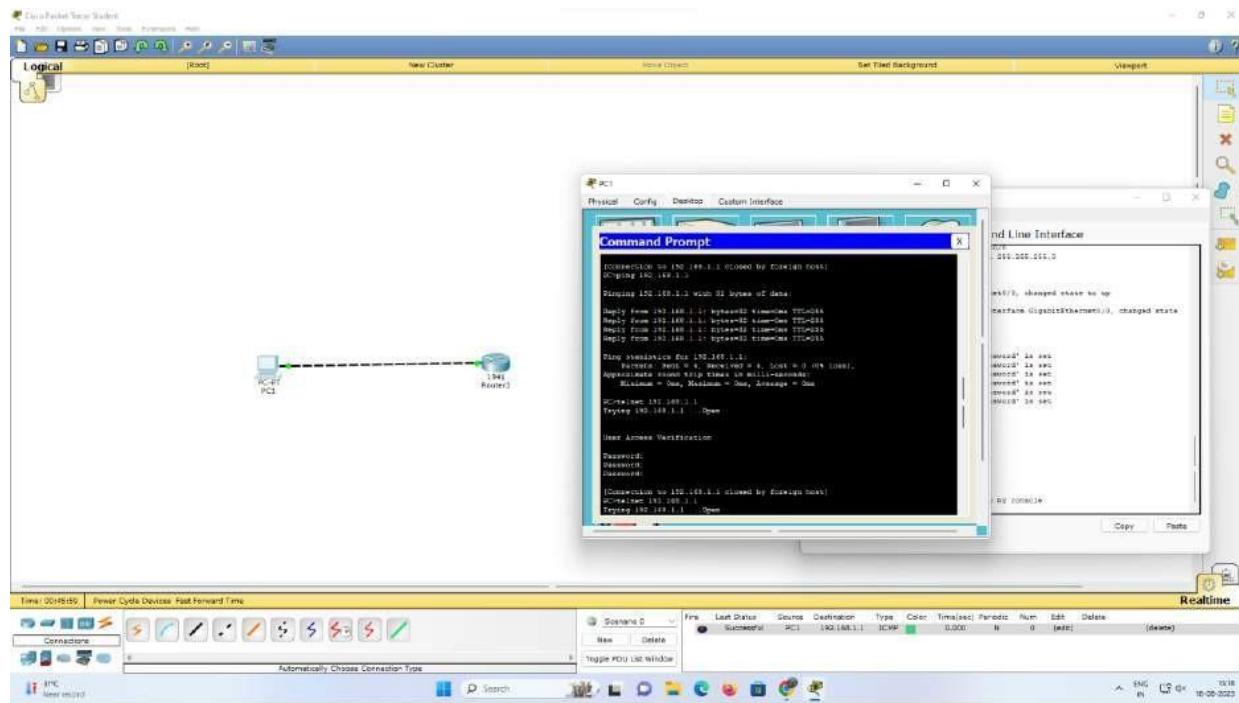
Gateway or last resort is not set

C 192.168.1.0/24 is Varies Subnet, mask
C 192.168.1.0/24 is directly Connected, GigabitEthernet 0/0
L 192.168.1.1/32 is directly connected, GigabitEthernet 0/0.

Router II



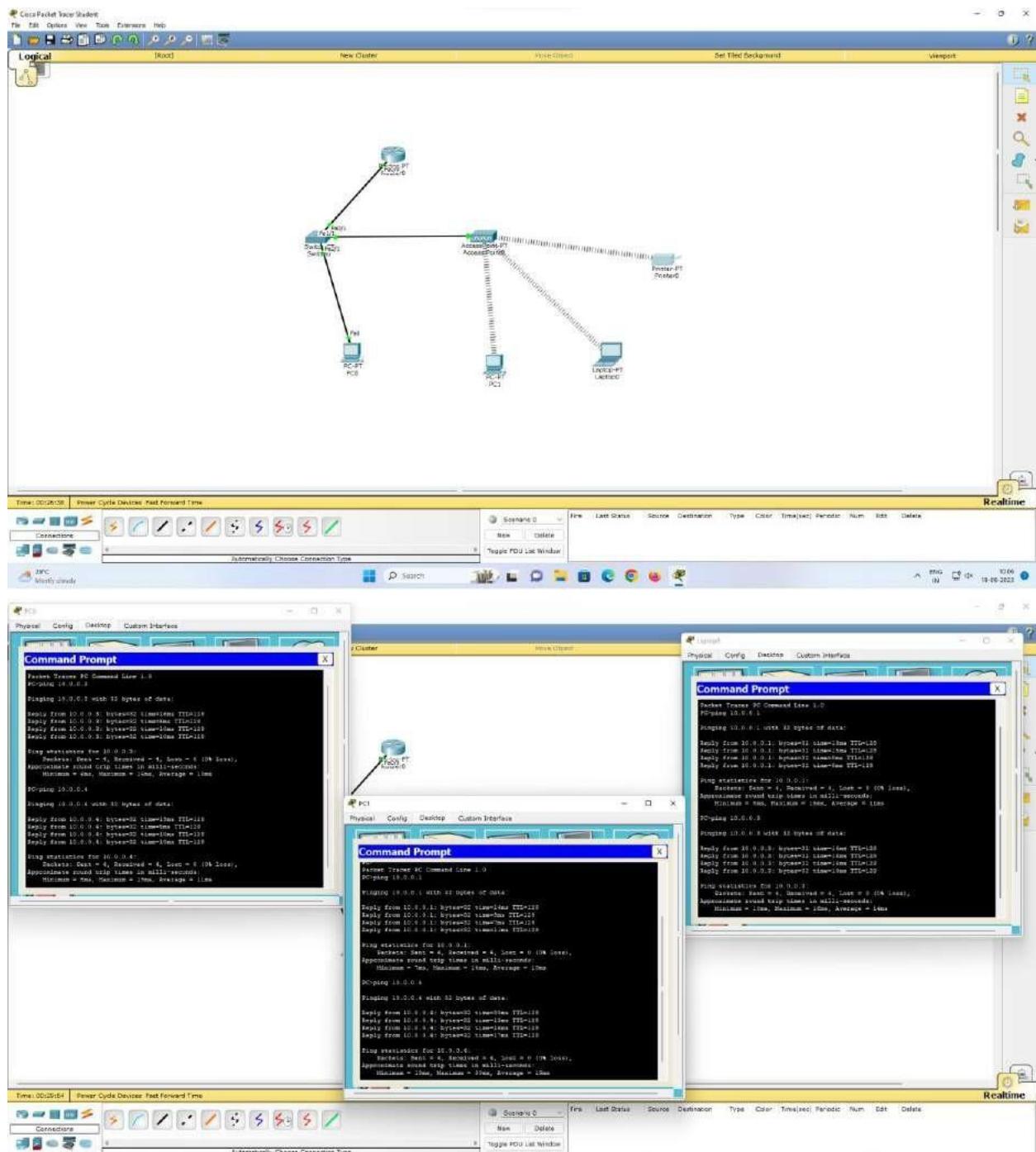




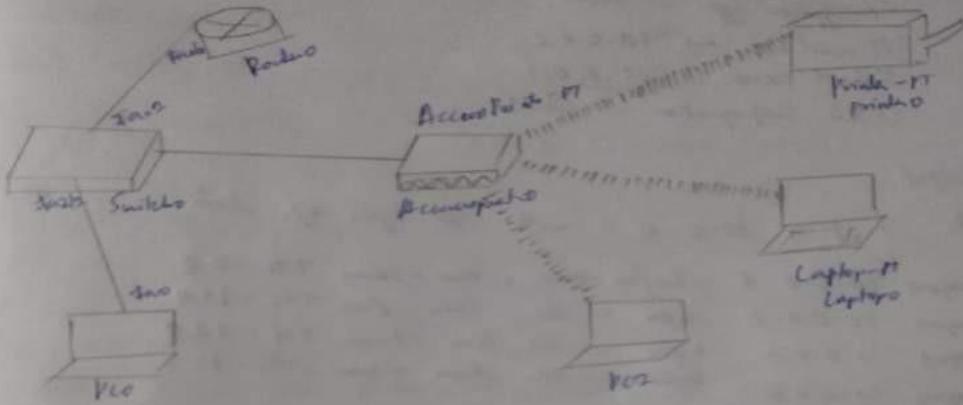
To construct a WLAN and make the nodes communicate wirelessly

Program 12

Wlan



To construct a WLAN and make the nodes communicate wirelessly -



Working b) The node topology is set up using AccessPoint, Router, PC's Laptop, Printer

c) By default Laptop has classic Ethernet Card. To inject wireless network, we should have wireless interface card

d) Wireless Interface Card (WIFC 300N) without use power on the Laptop again

d) Go to PC

Authentication → WEP

Encryption type → Static

IP address → 10.0.0.3

Subnet Mask → 255.0.0.0

IPv6 Configuration → Static

Output:

PC > ping 10.0.0.3 with 32 bytes of data.

Pinging 10.0.0.3 , bytes = 32 , time = 16ms , TTL = 128

Pinging 10.0.0.3 , bytes = 32 , time = 6ms , TTL = 128

Pinging 10.0.0.3 , bytes = 32 , time = 10ms , TTL = 128

Pinging 10.0.0.3 , bytes = 32 , time = 10ms , TTL = 128

Ping statistics for 10.0.0.3

Packets Sent = 4 , Received = 4 , loss = 0 (0.0% loss)

Approximate round-trip times in milli-Seconds

Minimum = 0ms , Maximum = 0ms , Average = 0ms.

Program 13

Q) Write a program for error detecting code using CRC - CCIT (16-bit)

```
#include <stdio.h>

char m[50], g[80], r[50], v[50], temp[50];
Void Caltrans(int);
Void CRC(int);
Void Calram();
Void Shift();
int main()
{
    int n, i=0;
    char ch, flag=0;
    printf(" Enter the frame bits : ");
    while ((ch=getchar(stdin)) != '\n');
    m[i++]=ch;
    n=i;
    for (i=0; i<16; i++)
        m[n+i] = '0';
    m[n] = '1';
    printf("\n Message after appending 16 zeros : %s", m);
    for (i=0; i<16; i++)
        g[i] = '0';
    g[0] = g[4] = g[11] = g[16] = '1';
    g[17] = '0';
    printf("\n generator : %s", g);
    CRC(n);
    printf("\n with quotient : %s", v);
}
```

```

caltrans(h),
printf("In transmitted frames : %d + s", m);
printf(" In ENCLA transmitted frame : ");
scanf("%s", s);
printf("CRC checking \n");
crc(n);
printf("In Inlat remained %s", r);
for(i=0; i<16; i++)
if(r[i] != s[i]);
    flag = 1;
else
    continue;
if(flag == 2)
    printf("Error during transmission");
else
    printf("In received frame is correct");
}

```

Void crc (it n)

```

int i, j;
for(i=0; i<n; i++)
temp[i] = m[i];
for(i=0; i<16; i++)
r[i] = m[i];
printf("In intermediate remained \n");
for(i=0; i<n-16; i++)
if(r[i] == 17) {

```

```

    v[i7] = '2';
    Cal_ram();
}
else {
    v[i] = '0';
    Shift();
}
v[16] = m[17+i];
r[17] = '10';

printf ("\n remainder : - .d : s", i+1, r);
for (j=0; j <= 17; j++)
temp[j] = r[j];
}
v[n-16] = '0';
}

```

Output :

Enter the frame bits : 1011

Message after appending 16 zeros : 101100000000000000

Generator : 10001000000100001

intermediate remainders.

remainder 1 : 011100000010000010

remainder 2 : 111000000100000100

remainder 3 : 11010000101001010.

remainder 4 : 1011000101101011

quotient : 1011

transmitted frame : 10111101100010110101

File generated from : <http://www.pdflint.com>

let `marks` = `00000000000000000000000000000000`

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

```
#include<stdio.h>

char
m[50],g[50],r[50],q[50],temp[50];
void caltrans(int); void crc(int);

void calram();
void shiftl();

int main()
{
    int n,i=0; char
ch,flag=0;
printf("Enter the frame bits:");
while((ch=getc(stdin)) != '\n')
m[i++]=ch;
n=i;
for(i=0;i<16;i++)
m[n++]='0';
m[n]='\0';

printf("Message after appending 16
zeros:%s",m); for(i=0;i<=16;i++)
g[i]='0';
```

```
g[0]=g[4]=g[11]=g[16]='1';g[17]='\0';

printf("\n\ngenerator:%s\n",g);

crc(n);

printf("\n\nquotient:%s",q);

caltrans(n);
```

```

printf("\ntransmitted frame:%s",m);

printf("\nEnter transmitted frame:");
scanf("\n%s",m);

printf("CRC checking\n");

crc(n);

printf("\n\nlast remainder:%s",r);

for(i=0;i<16;i++)

if(r[i]!='0')

flag=1;

else

continue;

if(flag==1)

printf("Error during transmission");

else

printf("\n\nReceived frame is correct");

}

void crc(int n)

{

int i,j;

for(i=0;i<n;i++)

temp[i]=m[i];

for(i=0;i<16;i++)

r[i]=m[i];

printf("\nintermediate

remainder\n"); for(i=0;i<n-16;i++)

{



if(r[0]=='1')

```

```

{
    q[i]='1';
    calram();
}

else
{
    q[i]='0';
    shiftl();
}

r[16]=m[17+i];
;
r[17]='\0';
printf("\nremainder %d:%s",i+1,r);
for(j=0;j<=17;j++)

```

```

temp[j]=r[j];

}

q[n-16]='\0';

}

void calram()

{

int i,j;

for(i=1;i<=16;i+

+)

r[i-1]=((int)temp[i]-48)^((int)g[i]-48)+48;

}

void shiftl()

{

int i;

for(i=1;i<=16;i

++)

r[i-1]=r[i];

}

void caltrans(int n)

{

int i,k=0;

for(i=n-16;i<n;i++)

m[i]=((int)m[i]-48)^((int)r[k++]-48)+48;

m[i]='\0';

}

```

```
Enter the frame bits:1011
Message after appending 16 zeros:10110000000000000000
generator:1000100000100001

intermediate remainder

remainder 1:01110000001000010
remainder 2:11100000010000100
remainder 3:11010000101001010
remainder 4:1011000101101011

quotient:1011
transmitted frame:10111011000101101011
Enter transmitted freme:10111011000101101011
CRC checking

intermediate remainder

remainder 1:01100110000011000
remainder 2:11001100000110001
remainder 3:10001000000100001
remainder 4:0000000000000000

last remainder:0000000000000000

Received freme is correct
```

Program 14

**Write a program for congestion control using
Leaky
bucket algorithm.**

Q) Write a program for Congestion Control using leaky bucket algorithm.

```
#include <Stdio.h>
#include <Stddb.h>
#include <Conio.h>

Void main()
{
    int i, packets[10], content = 0, newContent, time, clk, bcksize, operate;
    for (i=0; i<5; i++)
    {
        packets[i] = rand() % 10;
    }
    if (packets[5] == 0) --i;
    printf ("In Enter the output rate of the bucket : \n");
    Scanf ("%d", &operate);
    printf ("Enter BucketSize\n");
    Scanf ("%d", &bcksize);
    for (i=0; i<5; ++i)
    {
        if ((packet[i] + content) > bcksize)
        {
            if (packets[i] > bcksize)
                printf ("In Incoming packet size .d greater than the size of\n"
                        "the bucket\n");
            newContent = packets[i];
            content += newContent;
            printf ("Incoming packet .d\n", newContent);
            printf ("In transmission left : .d\n", content);
        }
    }
}
```

```

time = rand (0.0, 1.0)
printf("In Next packet will come at %d\n", time);
for (clk=0; clk < time + d_content > 0; ++clk) {
    printf("In left time %d ", (time - clk));
    if (content) {
        printf("In Transmitted\n");
        if (content < operate) content = 0; else
            content = content - operate;
        printf("In Bytes remaining : %d\n", content);
    }
    printf("In No packets to send\n");
}

```

Output:

Enter slp rate of bucket: 4

Enter BucketSize
5

Incoming P: 2

Transmission left = 2

Next packet will come at 8

Left time: 8

Transmitted

Bytes remaining: 0

Incoming packet size 7 greater than the size the buffer

Incoming Packet: 4

To transmission: 4

Next packet will come at 8

left time 8

Transmitted

Bytes Remaining: 0

Incoming packet size 7 greater than the size of the bucket

Incoming packet: 4

Transmission left: 9

Next packet will come at 8

left time 8

Transmitted

Bytes remaining: 0

Incoming packet size 9 greater than the size of the bucket

Incoming: 4

Transmission left: 4

Next packet will come at: 2

left time 2

Transmitted

Bytes remaining: 0

**Write a program for congestion control using
Leaky
bucket algorithm.**

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
```

```
void main() {
```

```

int i,packets[10],content=0,newcontent,time,clk,bcktsize,ostrate;

for(i=0;i<5;i+
+)

{ packets[i]=rand()
%10; if(packets[i]==0)
--i;

}

printf("\n Enter output rate of the bucket: \n");
scanf("%d",&ostrate); printf("\

n Enter Bucketsize\n");
scanf("%d",&bcktsize);
for(i=0;i<5;++i)

{

if((packets[i]+content)>bcktsize)

{

if(packets[i]>bcktsize)

printf("\n Incoming packet size %d greater than the size of the
bucket\ n",packets[i]);

else printf("\n bucket
size exceeded\n");

}

else

{

newcontent=packets[i]; content+=newcontent;
printf("\n Incoming Packet :
%d\n",newcontent); printf("\n Transmission
left : %d\n",content); time=rand()%10;

printf("\n Next packet will come at %d\n",time);

for(clk=0;clk<time && content>0;++clk)
{

```

```
printf("\n Left time %d", (time-
clk)) ; if(content)

{
printf("\n Transmitted\n");

if(content<oprate) content=0; else
content=content-oprate;
printf("\n    Bytes remaining   :
%d\n", content);

}
else printf("\n No packets to
send\n") ; }
```

```
}
```

```
Enter output rate of the bucket:  
4  
  
Enter Bucketsize  
5  
  
Incoming Packet : 1  
  
Transmission left : 1  
  
Next packet will come at 8  
  
Left time 8  
Transmitted  
  
Bytes remaining : 0  
  
Incoming packet size 7 greater than the size of the bucket  
  
Incoming Packet : 4  
  
Transmission left : 4  
  
Next packet will come at 8  
  
Left time 8  
Transmitted
```

Activate \
Go to Settings

Program 15

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

Q1 Using TCP/IP Sockets , to write a client -
server program to make the Client send the
file name and the Server back the contents
of the requested file if present

Code :

```
Servertcp.py from  
Socket import *  
ServerName = 'LAPTOP-HATRKFO6'  
ServerPort = 12530 ServerSocket = socket(AF_INET, SOCK_STREAM)  
ServerSocket.bind((ServerName, ServerPort))  
ServerSocket.listen(2).print("The Server is ready to receive")  
while True:  
    ConnectionSocket, addr = ServerSocket.accept()  
    Sentence =  
        ConnectionSocket.recv(1024).decode()  
        file = open(Sentence, "r").read(1024)  
        ConnectionSocket.send(file.encode()) file.close()  
        ConnectionSocket.close()
```

Clienttcp.py from Socket import *

```
ServerName = 'LAPTOP-HATRKFO6'  
ServerPort = 12530 ClientSocket = socket(AF_INET, SOCK_STREAM)  
ClientSocket.connect((ServerName, ServerPort))  
Sentence = input("Enter File Name")  
ClientSocket.send(Sentence.encode()).filecontents = ClientSocket.recv(1024)
```

(From Server: fileContent) ClientSocket.close()

Output:

The Server is ready to receive
Enter file name servertop.py
from Server: from socket import *
ServerName = 'LAPTOP-HATREKFO6'
ServerPort = 12530
ServerSocket = Socket (AF - INET, SOCK - STREAM)
ServerSocket . bind ((ServerName, ServerPort))
ServerSocket . listen (5)
print ("The Server is ready to receive")

Block 1:

ConnectionSocket, addr = ServerSocket . accept ()
Sentence = ConnectionSocket . recv (1024) . decode()
file = open (Sentence, "r")
l = file . read (1024)
ConnectionSocket . send (l . encode ())
file . close ()
~~ConnectionSocket . close ()~~

```
Code: servertcp.py from socket import * serverName='LAPTOP-HATTRKFO6' serverPort = 12530 serverSocket = socket(AF_INET,SOCK_STREAM) serverSocket.bind((serverName,serverPort)) serverSocket.listen(1) print ("The server is ready to receive") while 1: connectionSocket, addr = serverSocket.accept() sentence = connectionSocket.recv(1024).decode() file=open(sentence,"r") l=file.read(1024) connectionSocket.send(l.encode()) file.close() connectionSocket.close() clienttcp.py from socket import * serverName = 'LAPTOPHATTRKFO6' serverPort = 12530 clientSocket = socket(AF_INET, SOCK_STREAM) clientSocket.connect((serverName,serverPort)) sentence = input("Enter file name") clientSocket.send(sentence.encode()) filecontents = clientSocket.recv(1024).decode() print ('From Server:', filecontents) clientSocket.close()
```

Output Screenshot

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

```
Enter file nameservertcp.py
From Server: from socket import *
serverName='LAPTOP-HATTRKF06'
serverPort = 12530
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
print ("The server is ready to receive")
while 1:
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    file=open(sentence,"r")
    l=file.read(1024)
    connectionSocket.send(l.encode())
    file.close()
    connectionSocket.close()
```

Program 16

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

Q1 Using UDP sockets, write a client server program to make client send the file name and server has to send back the contents of the requested file if present.

Code:

```
Serverdp.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)  
    file = open(sentence, "r") l = file.read(2048)  
    ServerSocket.sendto(l.encode("utf-8"), ClientAddress)  
    print("Send back to Client", l) file.close()
```

Clientdp.py

```
from Socket import * serverName = "127.0.0.1"  
ServerPort = 12000 ClientSocket = Socket(AF_INET, SOCK_DGRAM)  
sentence = input("Enter file Name")  
ClientSocket.sendto(sentence.encode("utf-8"), (serverName, ServerPort))  
fileContents, ClientAddress = ClientSocket.recvfrom(2048) print("From  
Server:", fileContents) ClientSocket.close()
```

Output: The server is ready to receive
Send back to Client from back import *

ServerPort = 12000

ServerSocket = Socket(AF_INET, SOCK_DGRAM)

Server
print
while

Py

```
ServerSocket.bind ("127.0.0.1"; SomePort))  
print ("The server is ready to receive")
```

While 1:

```
Sentence, ClientAddress = ServerSocket.recvfrom(2048)
```

```
File = open(Sentence, "r")
```

```
l = file.read(2048)
```

```
ServerSocket.sendto(bytes(l, "utf-8"), ClientAddress)
```

```
print("Sent back to client", l)
```

```
file.close()
```

Done

```

Code: 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) file=open(sentence,"r") l=file.read(2048)
serverSocket.sendto(bytes(l,"utf-8"),clientAddress) print("sent back to
client",l) file.close() clientudp.py from socket import * serverName =
"127.0.0.1" serverPort = 12000 clientSocket = socket(AF_INET,
SOCK_DGRAM) sentence = input("Enter file name")
clientSocket.sendto(bytes(sentence,"utf-8"),(serverName, serverPort))
filecontents,serverAddress = clientSocket.recvfrom(2048) print ('From
Server:', filecontents) clientSocket.close()

```

Output Screenshot

```

The server is ready to receive
sent back to client 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)

    file=open(sentence,"r")
    l=file.read(2048)

    serverSocket.sendto(bytes(l,"utf-8"),clientAddress)
    print("sent back to client",l)
    file.close()

```

```

Enter file nameserverudp.py
From Server: b'from socket import *\nserverPort = 12000\nserverSocket = socket(AF_INET, SOCK_DGRAM)\nserverSocket.bind(("127.0.0.1",
serverPort))\nprint ("The server is ready to receive")\nwhile 1:\n    sentence,clientAddress = serverSocket.recvfrom(2048)\n    \n
    file=open(sentence,"r")\n    l=file.read(2048)\n    \n    serverSocket.sendto(bytes(l,"utf-8"),clientAddress)\n    print("sent ba
ck to client",l)\n    file.close()'

```