VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB RECORD

Computer Network Lab (23CS5PCCON)

Submitted by

Mohith Jain (1BM22CS162)

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
Academic Year 2024-25 (odd)

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 Network (23CS5PCCON)" carried out by **Mohith Jain (1BM22CS162),** who is a bonafide student of **B.M.S. College of Engineering.** It is in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements of the above-mentioned subject and the work prescribed for the said degree.

Sneha Asst. Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE

Index

Github Link:
https://github.com/mohithjain/CNLab_CS162

Program 1

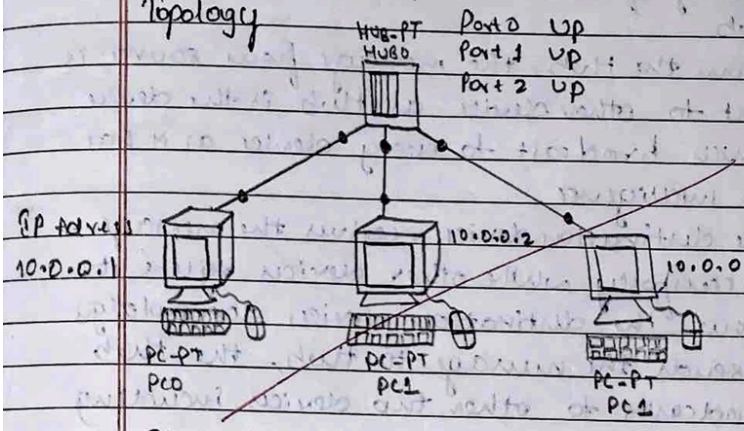
AIM: To demonstrate the transmission of a single PDU between two devices connected using a hub and a switch

Lab-01

AIM: To demonstrate the transmission of a single PDU between two devices connected using a Hub and a switch.

* Select the end devices : PC-PT PC0; add 3 device
* Select Hub-PT Hub0
* Connect the devices with the Hub using copper straight-through cable.

Topology



Steps to configure

* Click on the PC-PT P0 devices, select config Choose FastEthernet0 from Interface
* In IP configuration, check static and enter IP Address for 3 devices as 10.0.0.1, 10.0.0.2, 10.0.0.3, observing that subnet mask is automatically configured.

Steps to Add simple PDU

* From the right panel select simple PDU
* First, to select source device to pass message click on it and then select destination device

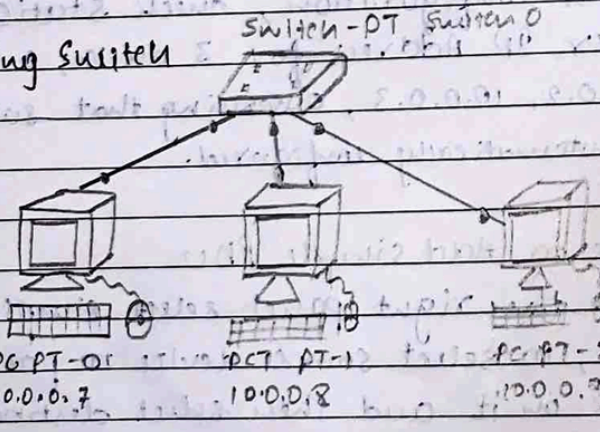
9/10/24

* Now, the PDU is successfully set.
* To check the simulation of PDU, switch to simulation mode from the panel.
* From play control in simulation mode, click on Add Capture / Play

Observation

* When the simulation mode is played, the message from source device is sent to Hub
* From the Hub, the message from source is sent to other device as Hub is the device which broadcast to every device as it has no intelligence.
* The destination device receives the message successfully while other devices rejects it.
* Now, the destination devices acknowledges it sends the message to Hub, the Hub broadcasts to other two devices including source device.
* The message is received by the source successfully and rejected by other.

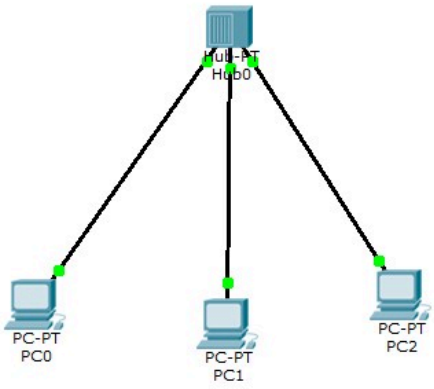
Using Switch



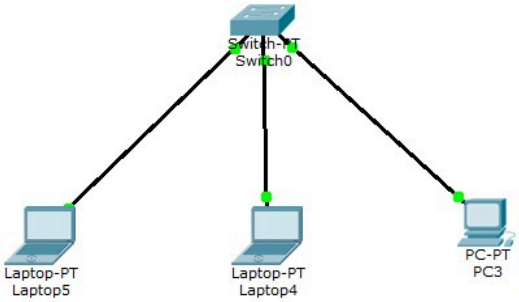
Bafna Gold
Date: 1/10 Page:

Observation

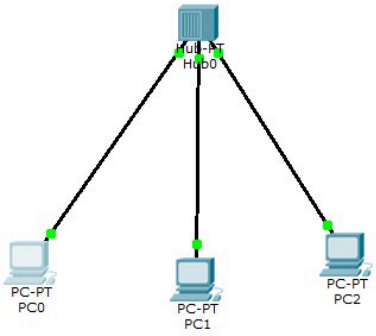
* Connect 3 PC devices or more to the switch (Generic).
* Configure the PC devices same procedure to follow.
* Now, using simple PDU send message from PCPT-0 to PCPT-2, the message from source is sent to switch and switch send the message directly to the destination i.e. PC-PT (PC-2).
* Here, the switch doesn't broadcast message to all other devices.

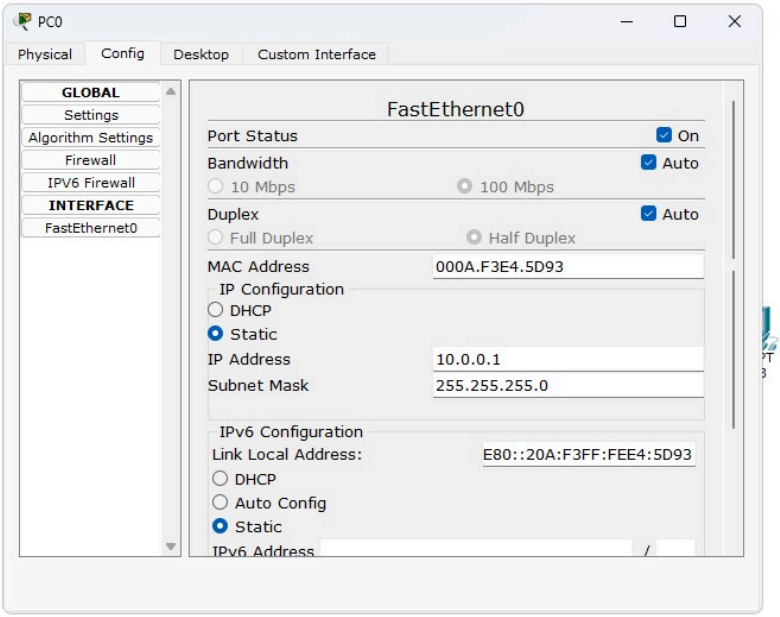


Hub (Star topology)

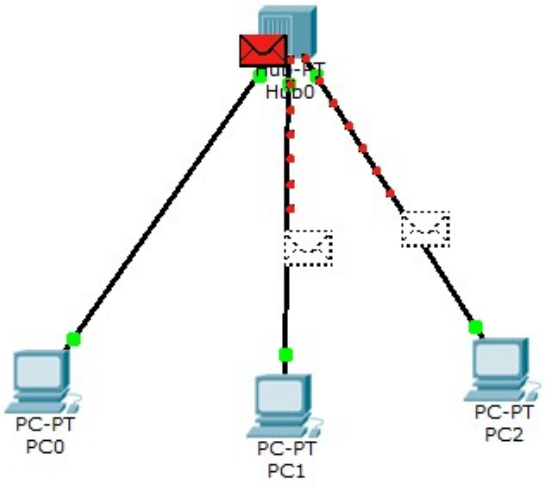


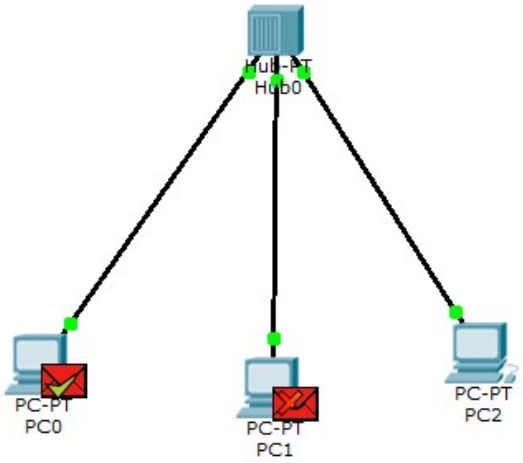
Switch (Star topology)

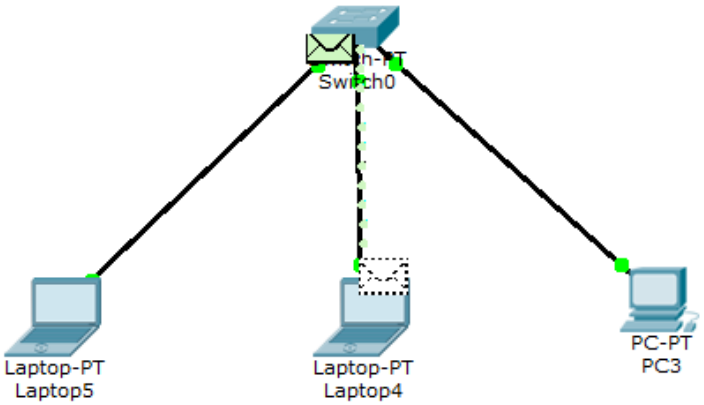


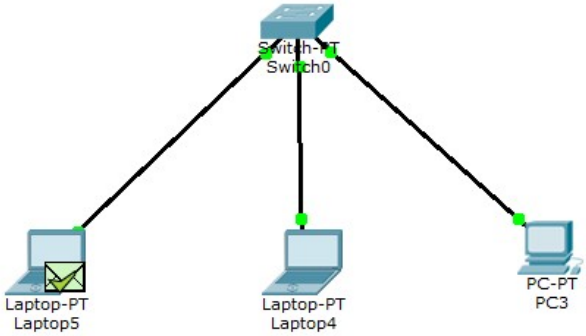


Configuration



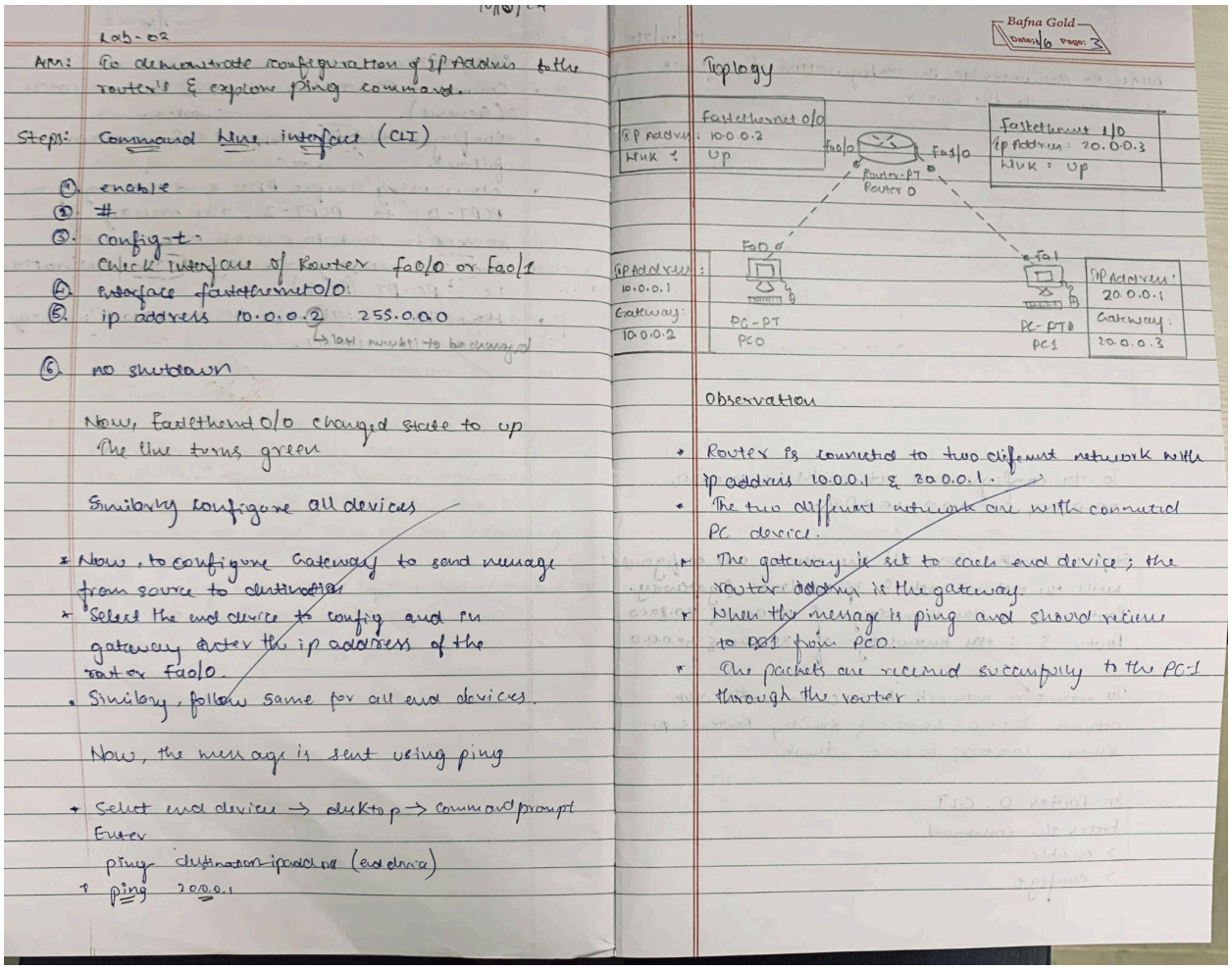




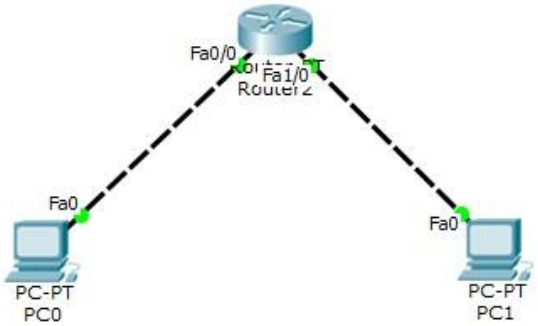


Computer Networks Lab 2
Router Configuration and ping

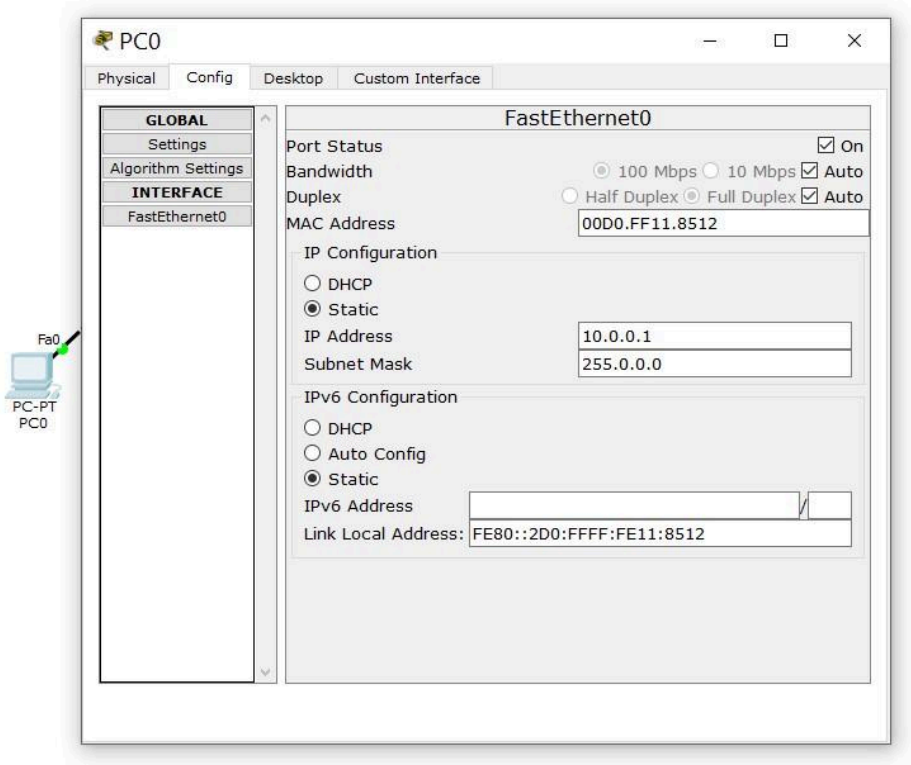
Observation



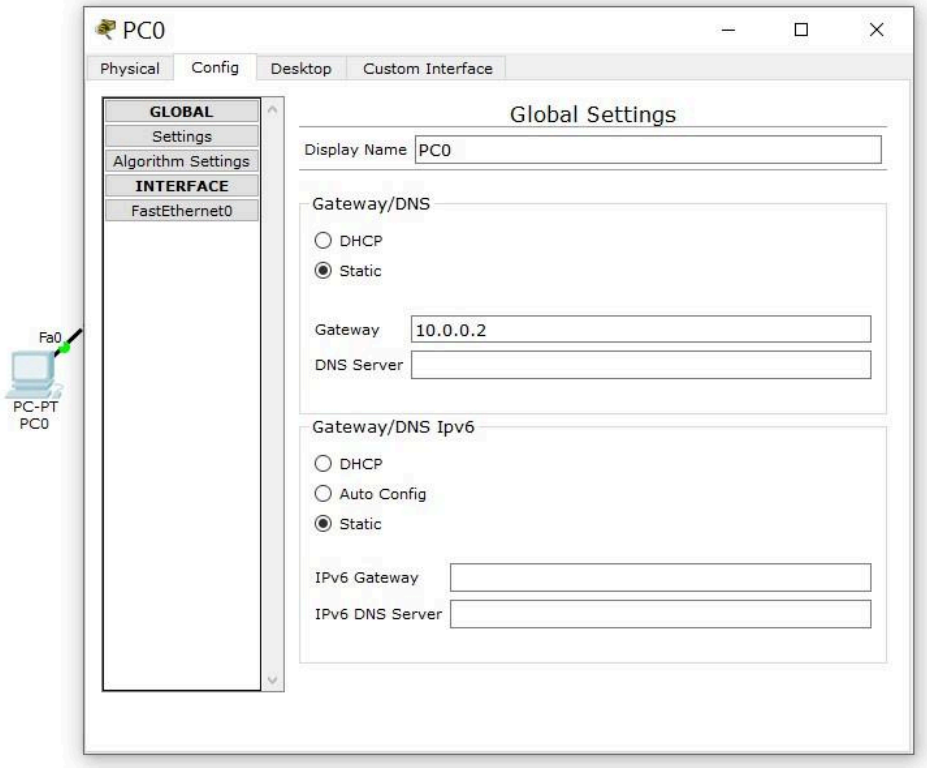
1.Network Topology



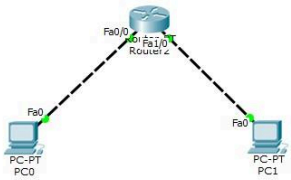
2. PC0 Configuration

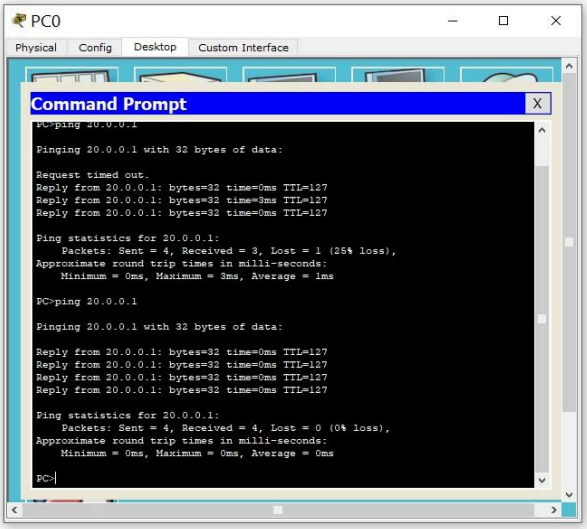


3. PC0 Gateway



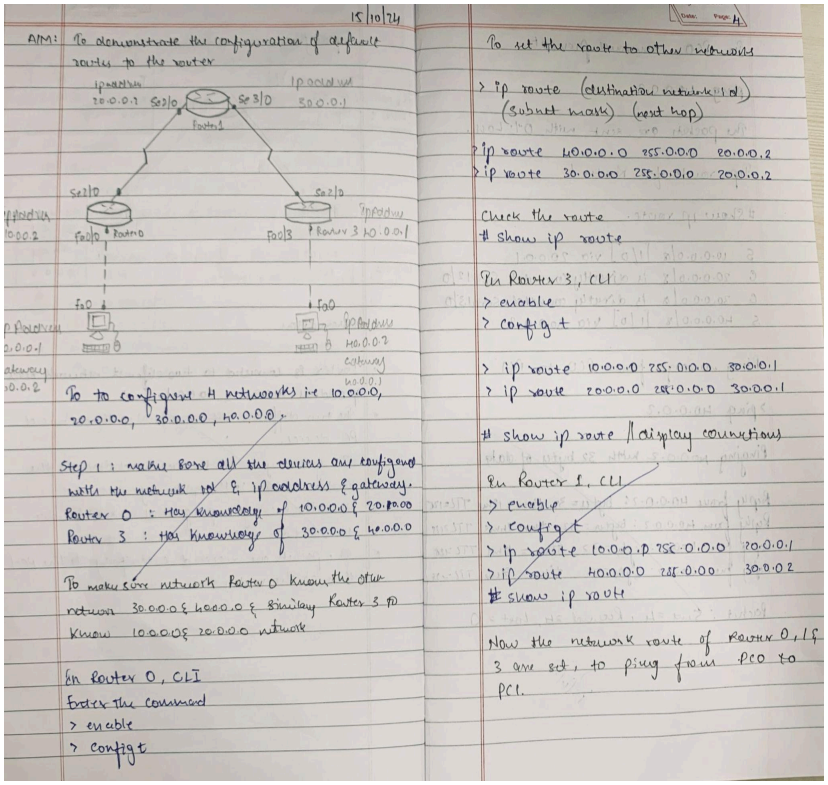
4. PC0 Ping PC1

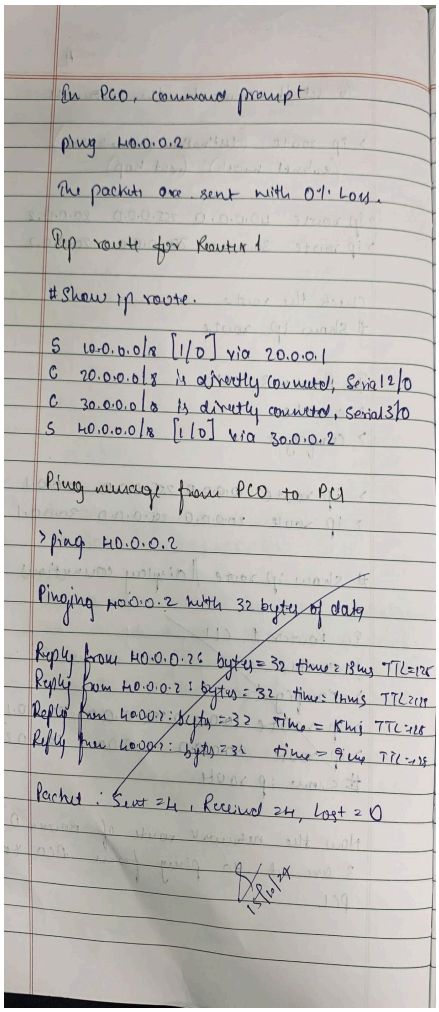




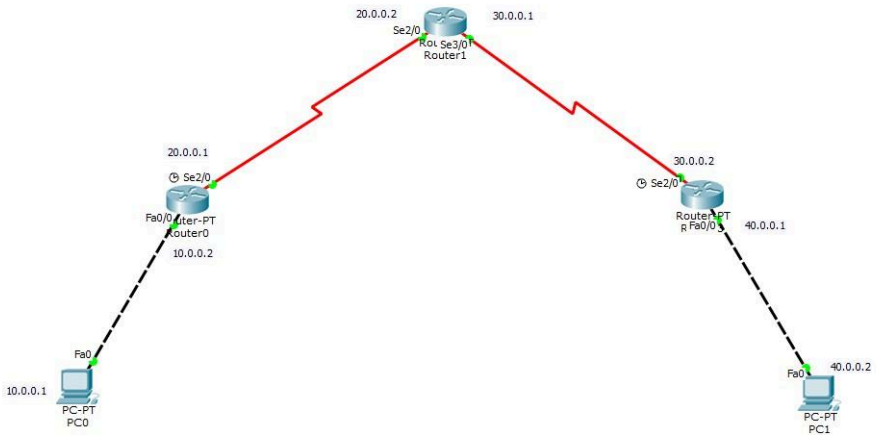
Computer Networks Lab 3
Configuration of default routes to Router and ping

Observation

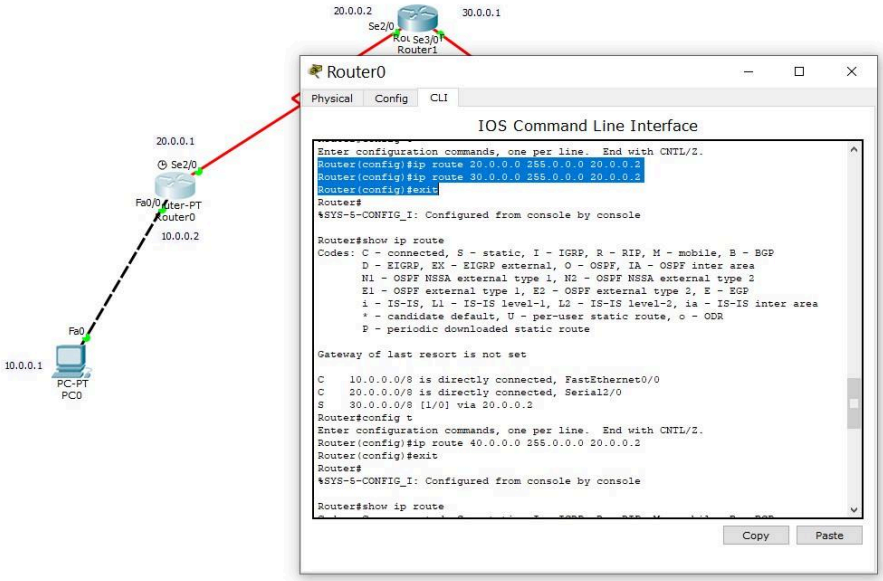




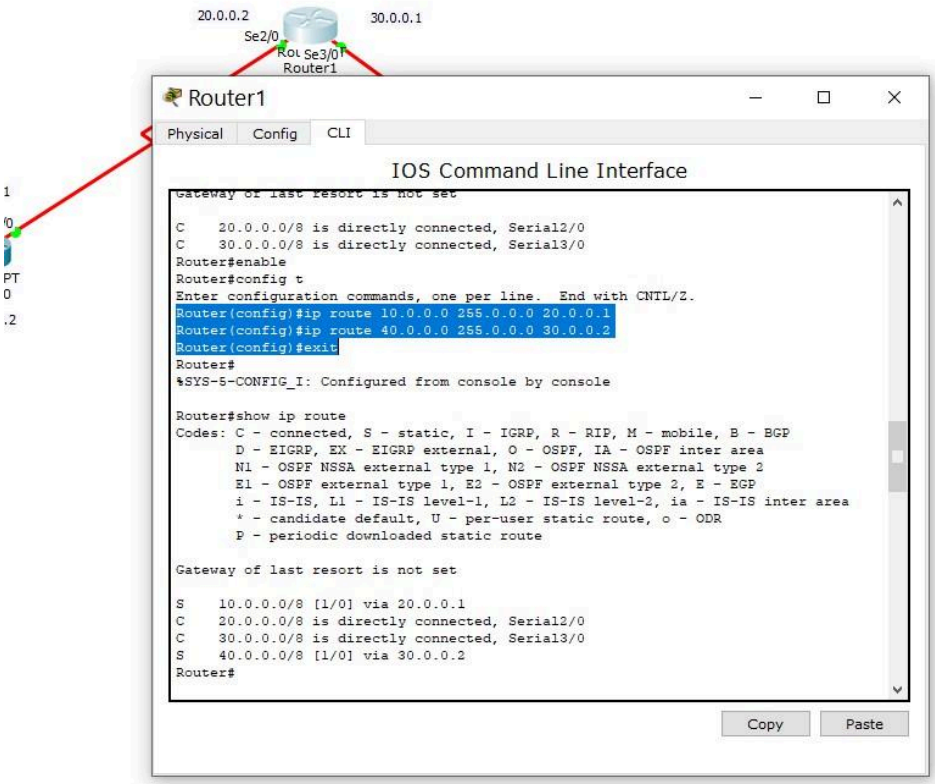
1.
2. **Topology**



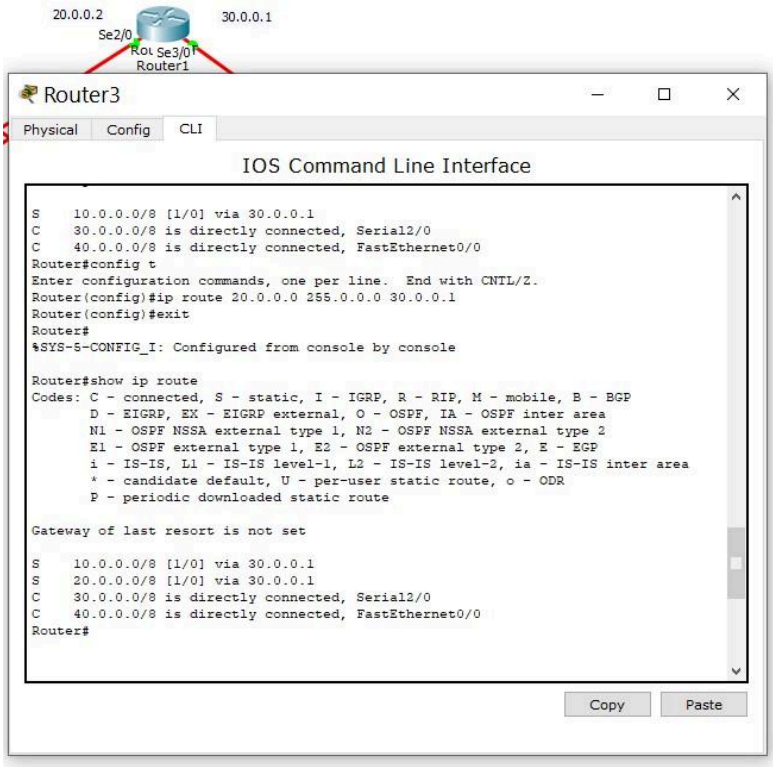
3. Router 0 ip routing



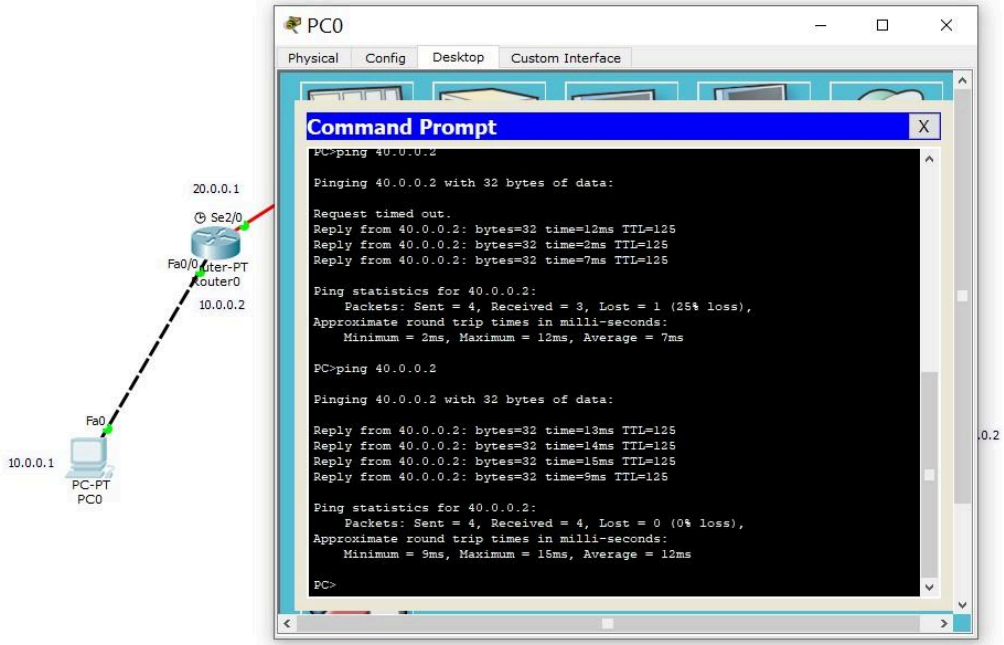
4. Router 1 ip routing



4. Router 3 ip routing



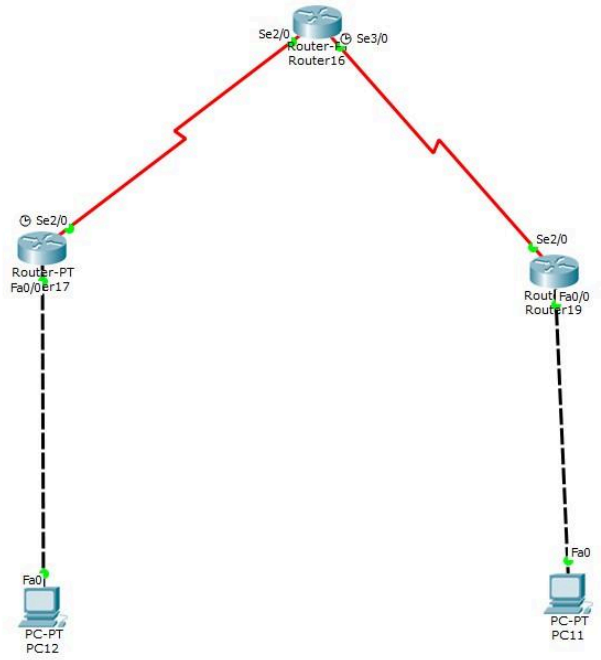
5. Ping to different network device

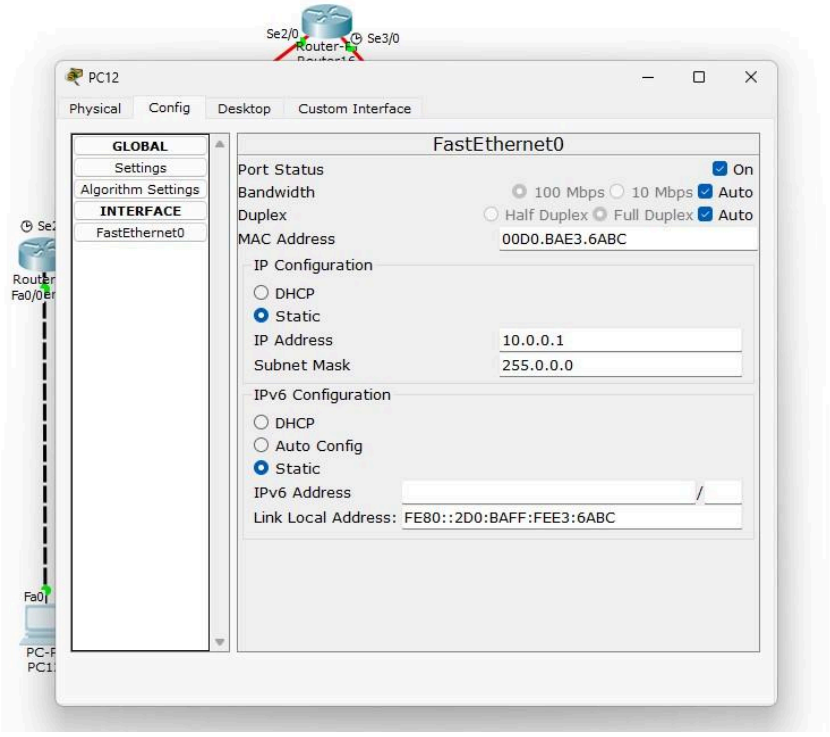


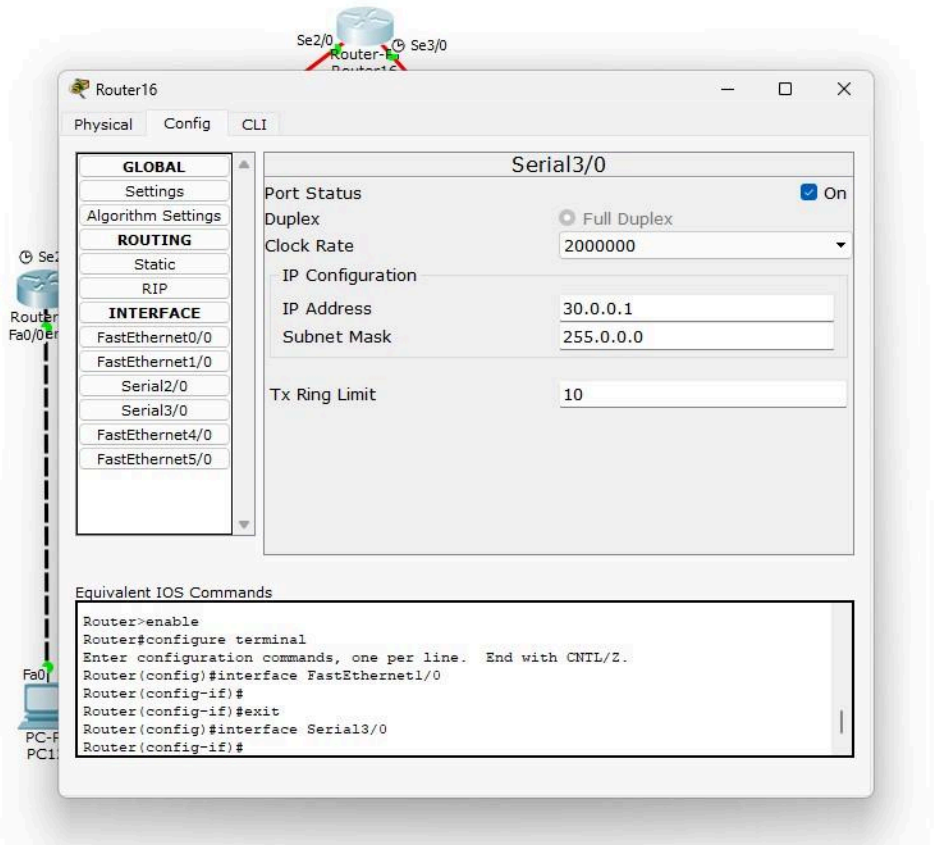
CN LAB OBSERVATION

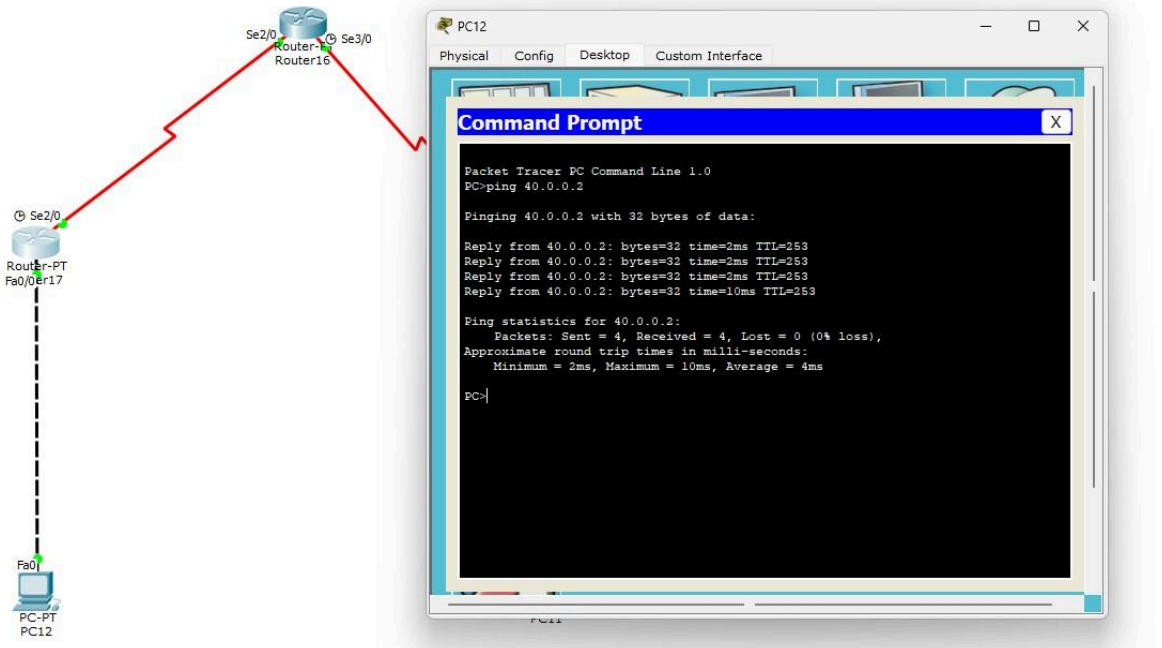
Mohith Jain 1BM22CS162

AIM : Configuration using default & static route



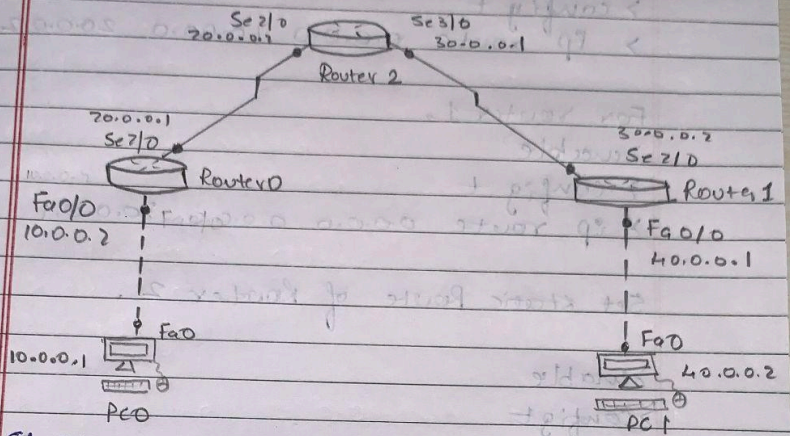



Se2/0
Se3/0
Router16
Physical
Config
CLI
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
INTERFACE
FastEthernet0/0
FastEthernet1/0
Serial2/0
Serial3/0
FastEthernet4/0
FastEthernet5/0
Serial3/0
Port Status
On
Duplex
Full Duplex
Clock Rate
2000000
IP Configuration
IP Address
30.0.0.1
Subnet Mask
255.0.0.0
Tx Ring Limit
10
Equivalent IOS Commands
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet1/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#


Se2/0
Se3/0
Router16
Se2/0
Router-PT
Fa0/0
Fa0
PC-PT
PC12
PC12
Physical
Config
Desktop
Custom Interface
Command Prompt
X
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.2
Pinging 40.0.0.2 with 32 bytes of data:
Reply from 40.0.0.2: bytes=32 time=2ms TTL=253
Reply from 40.0.0.2: bytes=32 time=2ms TTL=253
Reply from 40.0.0.2: bytes=32 time=2ms TTL=253
Reply from 40.0.0.2: bytes=32 time=10ms TTL=253
Ping statistics for 40.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 10ms, Average = 4ms
PC>

Lab-04

AIM: To demonstrate the configuration of default route and static routes.



Steps:

Configure the PC0, PC1, Router0, Router 1 and Router 2.

* Set ip address of PC0 to 10.0.0.1 & gateway to 10.0.0.2. Similarly set ip address for PC1 to 40.0.0.2 & gateway to 40.0.0.1.
* Now using CLI configure router0 & router1.
  > enable
  > Config t
  > Interface fastethernet0/0
  > ip address 10.0.0.2 255.0.0.0
  > exit
  > interface Serial 2/0
  > ip address 20.0.0.1 255.0.0.0
  > exit.
* Similary configure router 1 & router 2

Set the default Route to Router 0 & Router 1

For router 0.
> enable
> config t
> ip route 0.0.0.0 0.0.0.0 20.0.0.2

For router 1,
> enable
> config t
> ip route 0.0.0.0 0.0.0.0 30.0.0.1

Set Static Route of Router 2.

> enable
> config t
> ip route 40.0.0.0 255.0.0.0 30.0.0.2
destination network address · destination subnet mask · next hop configuration
> ip route 10.0.0.0 255.0.0.0 20.0.0.1

Observation

* The default route & static route of router has been set & configured.
* Now the network is ready for communication
* When ping from PC0 to PC1

> ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:
Reply from 40.0.0.2: bytes=32 time=18ms TTL=123
Reply from 40.0.0.2: bytes=32 time=4ms TTL=123

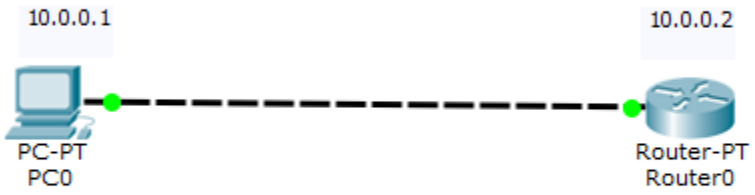
Packet sent = 4, Received = 4 Lost = 0 (0% Loss)

NAME : Mohith Jain USN: 1BM22CS162

CN LAB 5

AIM: TO UNDERSTAND THE OPERATION OF TELNET BY ACCESSING THE ROUTER PLACED IN THE SERVER ROOM FROM A PC IN IT OFFICE.

TOPOLOGY:



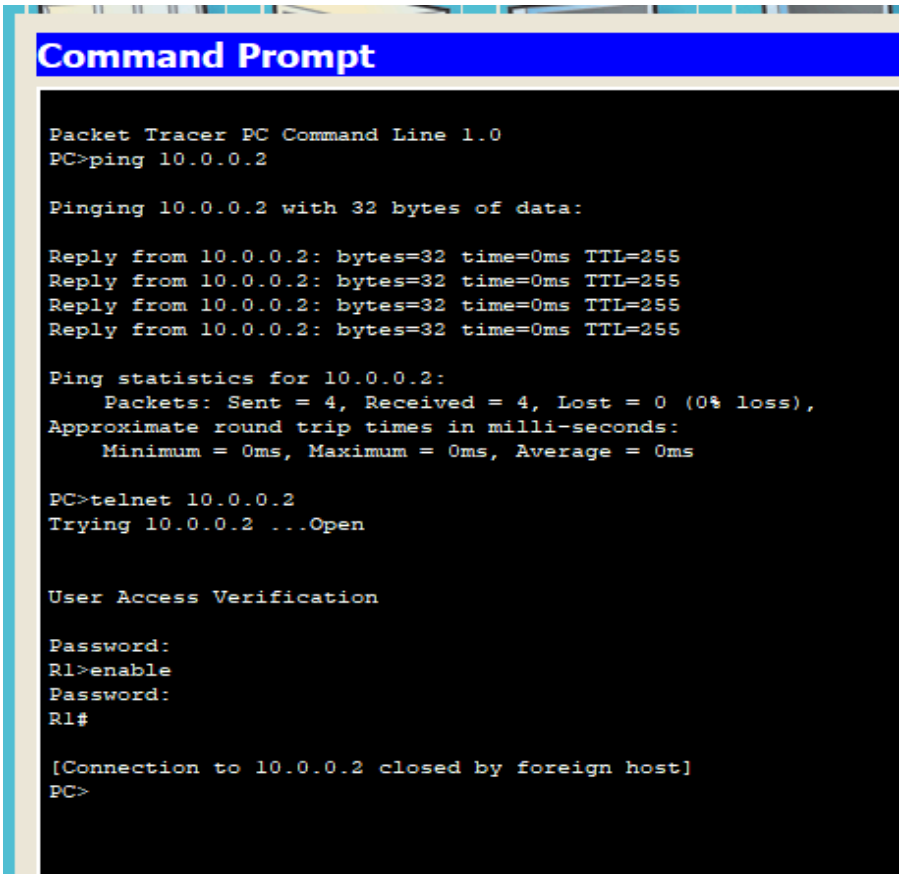
CONNECTION SETUP B/W PC and ROUTER:

```
Router>enable
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface fastethernet0/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shutdown
```

ROUTER CONFIGURATION FOR SECRET-KEY and PASSWORD:

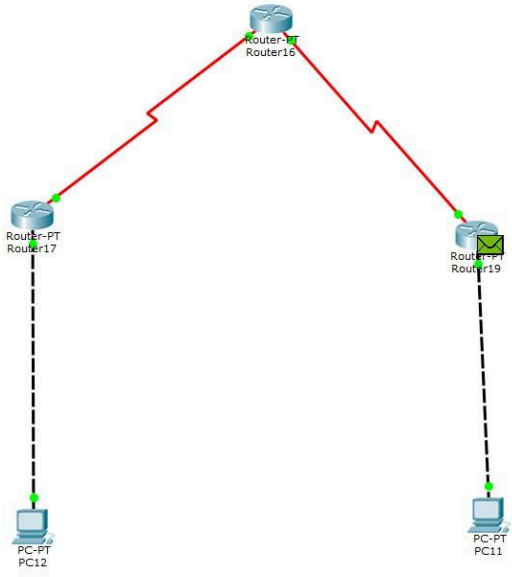
```
Router(config-if)#exit
Router(config)#interface FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 10.0.0.2 255.0.0.0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/0
Router(config-if)#exit
Router(config)#hostname R1
R1(config)#ebenable secret p0
R1(config)#line vty 0 5
R1(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
R1(config-line)#password p1
R1(config-line)#exit
R1(config)#exit
R1#
%SYS-5-CONFIG_I: Configured from console by console

R1#wr
Building configuration...
[OK]
R1#
```

USING TELNET COMMAND:

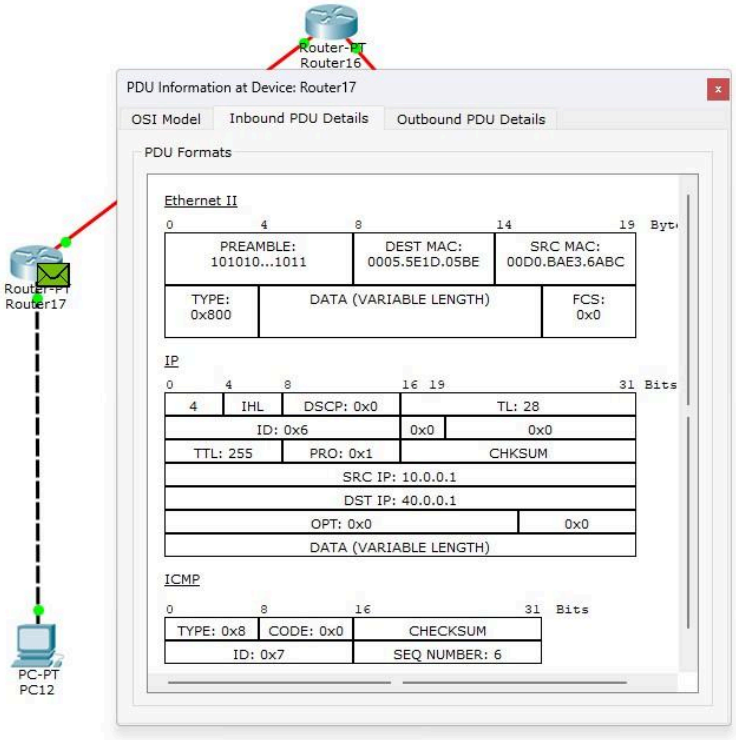


b) TTL CONCEPT

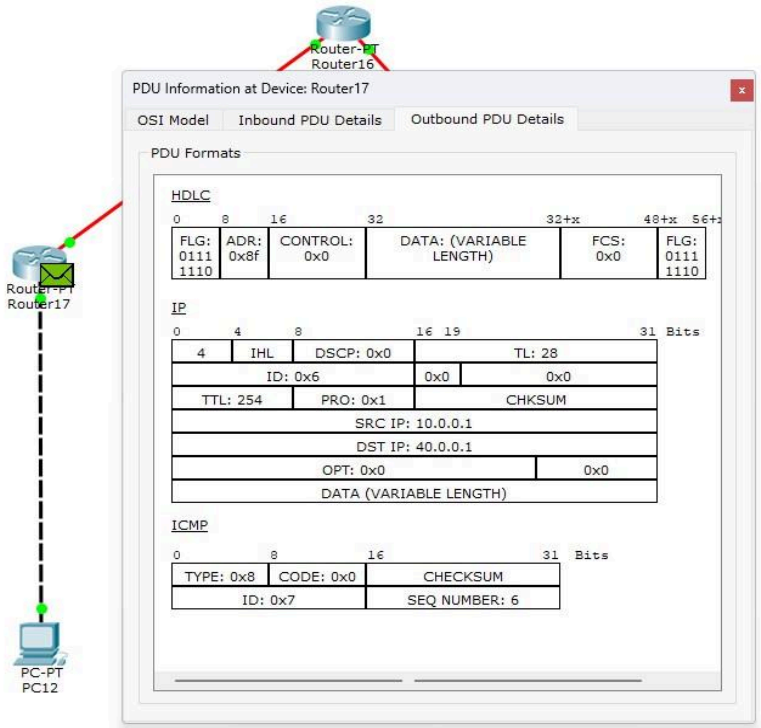
TOPOLOGY:



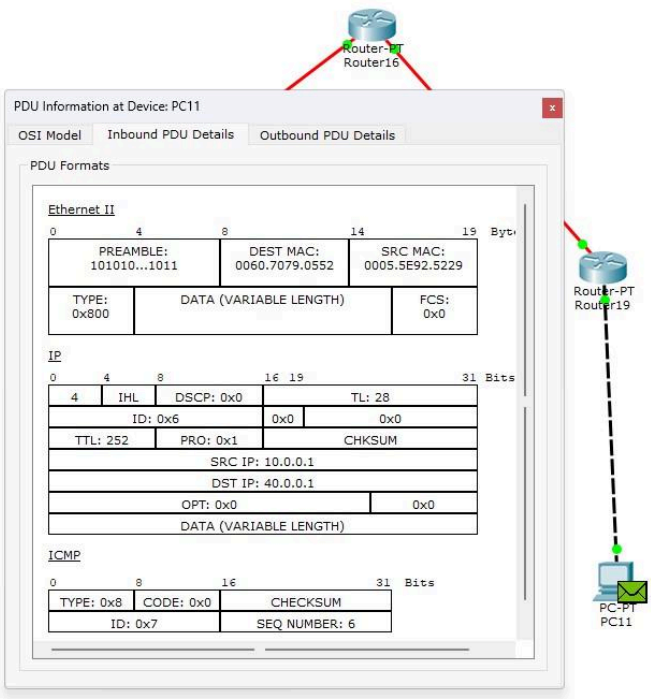
INBOUND FOR ROUTER1:



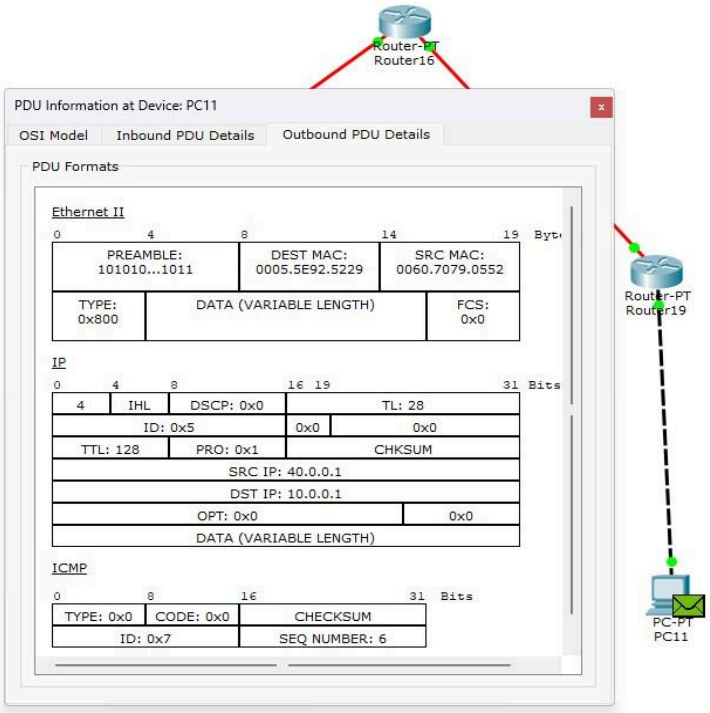
OUTBOUND FOR ROUTER1:



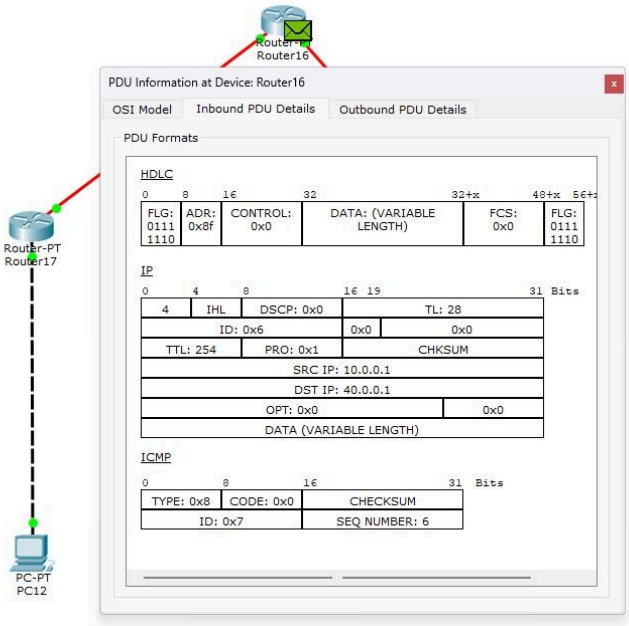
INBOUND FOR PC1:



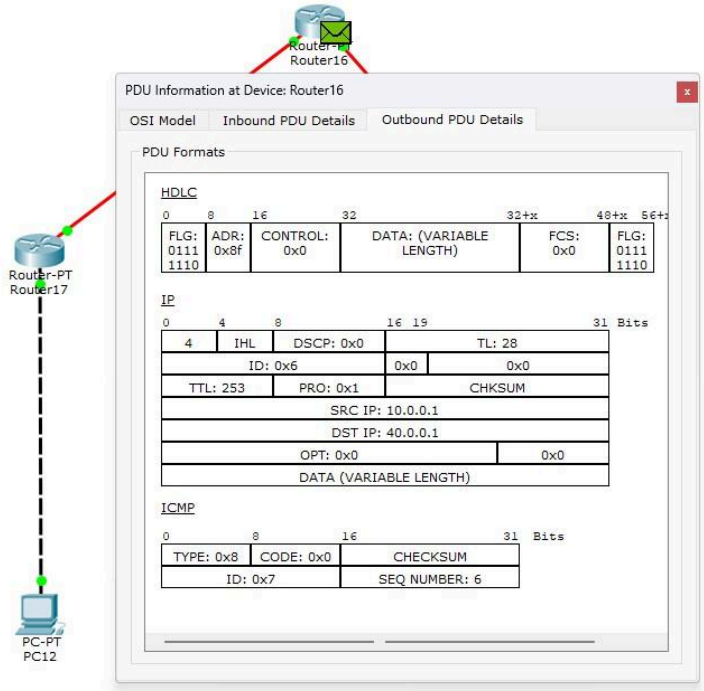
OUTBOUND FOR PC1:



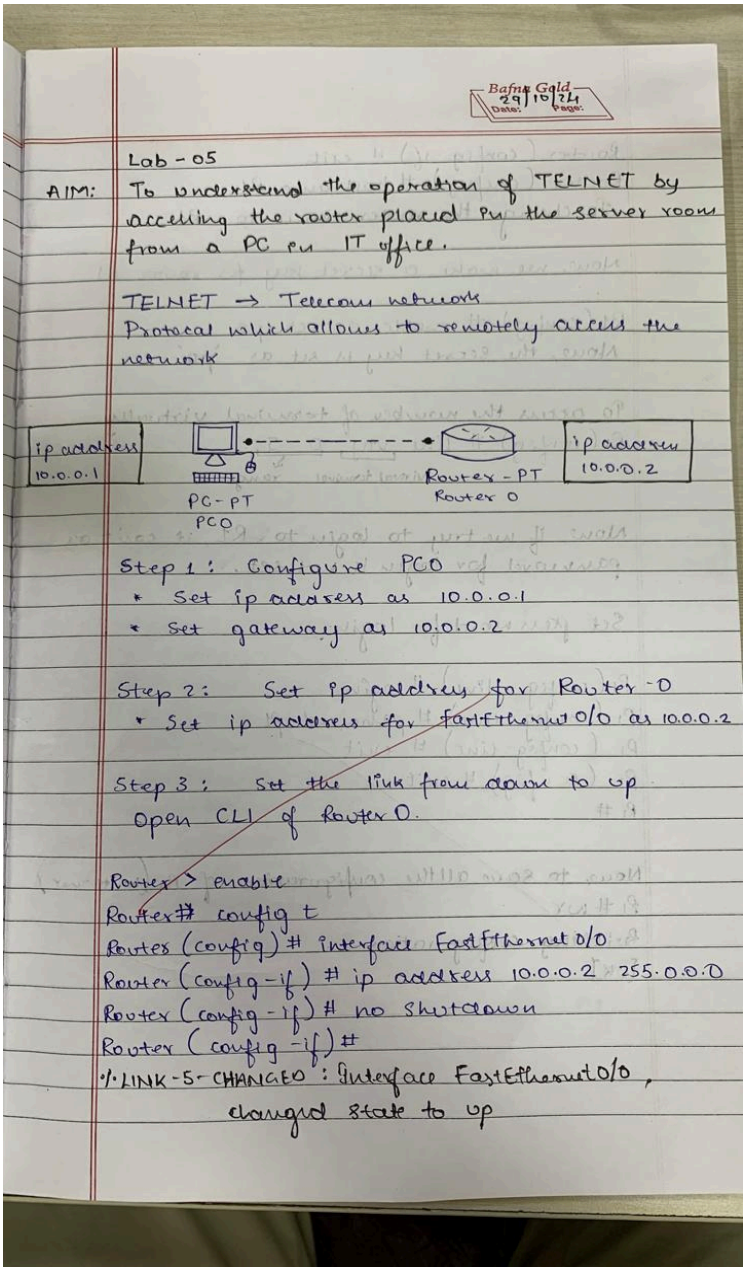
OUTBOUND FOR ROUTER1:



INBOUND FOR ROUTER1:



OBSERVATION: ACCESSING ROUTER USING TELNET:



Lab - 05

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

TELNET → Telecom network
Protocol which allows to remotely access the network

Step 1: Configure PC0
- Set ip address as 10.0.0.1
- Set gateway as 10.0.0.2

Step 2: Set ip address for Router-0
- Set ip address for FastEthernet 0/0 as 10.0.0.2

Step 3: Set the link from down to up
Open CLI of Router 0.

```
Router > enable
Router# config t
Router(config)# interface FastEthernet 0/0
Router(config-if)# ip address 10.0.0.2 255.0.0.0
Router(config-if)# no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0,
        changed state to up
```

```
Router(config-if)# exit
Router(config)# hostname R1
```
To change the router name to R1

Now, we make a secret key for router R1

```
R1(config)# enable secret p0
```
Now, the secret key is set as 'p0'

To access the number of terminal virtually
```
R1(config)# line vty 0 5
```
(vty: virtual terminal; 0 5: range)

Now, if we try to login to R1, it says as password for login is not set.

Set password for login

```
R1(config-line)# login
R1(config-line)# password p1
R1(config-line)# exit
R1(config)# exit
R1#
```

Now, to save all the configurations of R1 (write-mr)
```
R1# wr
Building configuration...
[OK]
```

Bafna Gold
Date: Page:

Step 4: Ping from PC0 to Router 0

• open command prompt in PC0.

>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2 bytes = 32 time = 0ms TTL = 255
Reply from 10.0.0.2 bytes = 32 time = 0ms TTL = 255
Reply from 10.0.0.2 bytes = 32 time = 0ms TTL = 255
Reply from 10.0.0.2 bytes = 32 time = 0ms TTL = 255

Ping statistics for 10.0.0.2:
Packets: Sent = 4, Received = 4 Lost = 0 (0% loss)

Now, use TELNET to access router R1

PC> telnet 10.0.0.2
Trying 10.0.0.2 .... Open

User Access Verification
Password: p1 // login password
R1> enable
Password: p0 // Secret key
R1#

Observation:
Using TELNET, PC0 can access router R1

29/10/24

CN LAB 6

Name : Mohith Jain USN: 1BM22CS162

AIM: TO CONFIG IP ADDRESSES OF HOST USING DHCP SERVER PRESENT WITHIN SAME LAN

OBSERVATION:

12/11/24

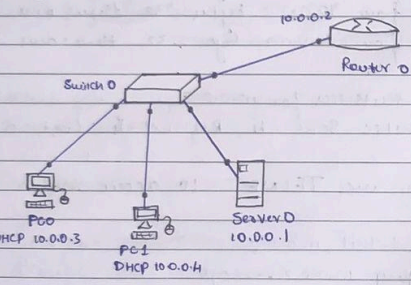
Lab-06

AIM: (A) To configure ip addresses of the Host using DHCP server present within the ~~the~~ LAN.

(B) To configure ip addresses of the Host using DHCP server present outside the ~~the~~ LAN

A: DHCP - Dynamic Host configuration Protocol - helps to do configuration of host automatically.

~~Conf~~ Topology with Network 10.0.0.0



Configuration

- Assign ip address manually to Server PT-Server 0 as 10.0.0.1
- Configure Router 0 & assign ip address 10.0.0.2 using CLI commands.
- Set gateway for the server-0 to be 10.0.0.2
- Now, select server-0, select services & click on DHCP, set service to be ON.
- Set PoolName as ServerPool1
- Default gateway to server as 10.0.0.2 & DNS server as 10.0.0.1 (temporarily) & SAVE.

Bafna Gold 12/11/24 Date: Page:

- Now DHCP server is ready & have been assigned to server 0
- Now, host devices can automatically get assigned to ip address by DHCP server.
- Select PC0 & PC1 & check DHCP to automatically assign ip address.
- Now, the network is ready to communicate.

Output

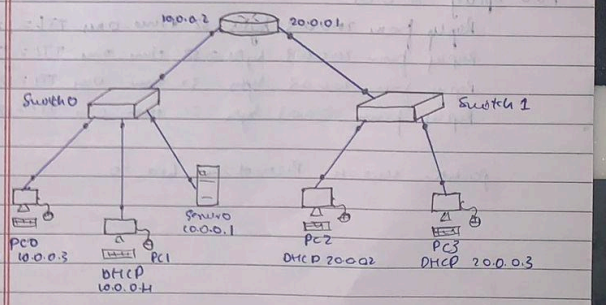
- ping 10.0.0.4

```
Pinging 10.0.0.4 with 32 bytes of data
Reply from 10.0.0.4 bytes=32 time=0ms TTL=128
Reply from 10.0.0.4 bytes=32 time=0ms TTL=128
Reply from 10.0.0.4 bytes=32 time=0ms TTL=128
Reply from 10.0.0.4 bytes=32 time=0ms TTL=128
```

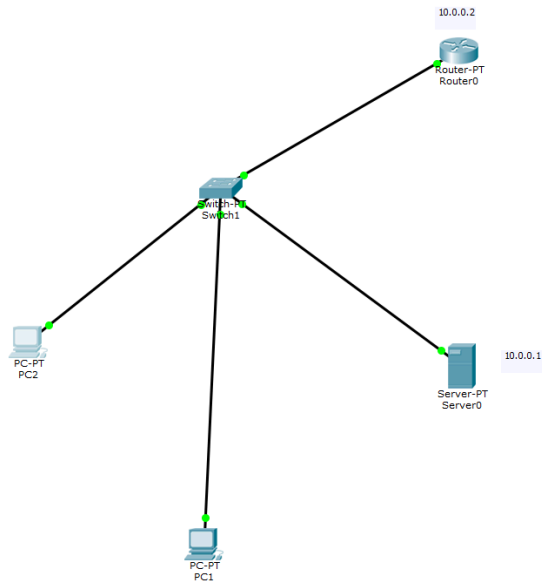
packets: sent=4 Received=4 lost=0

B) Outside the LAN

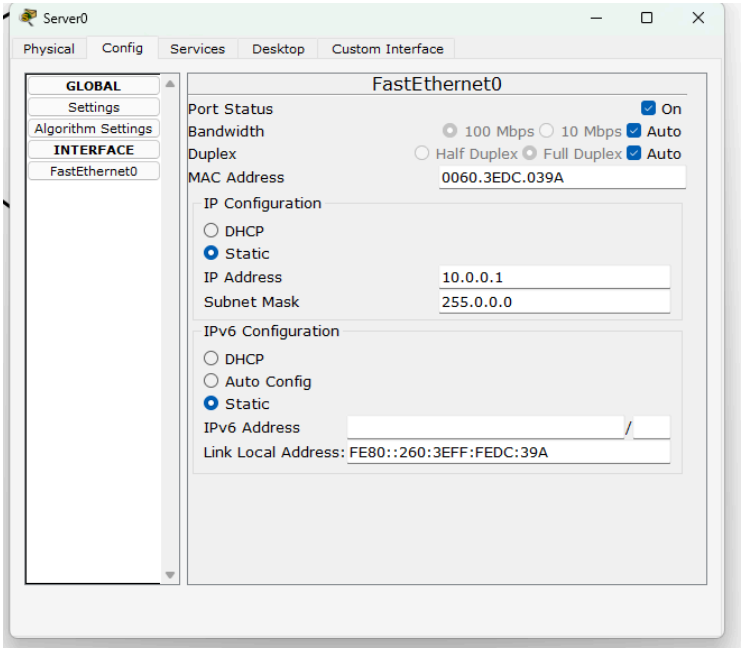
Topology



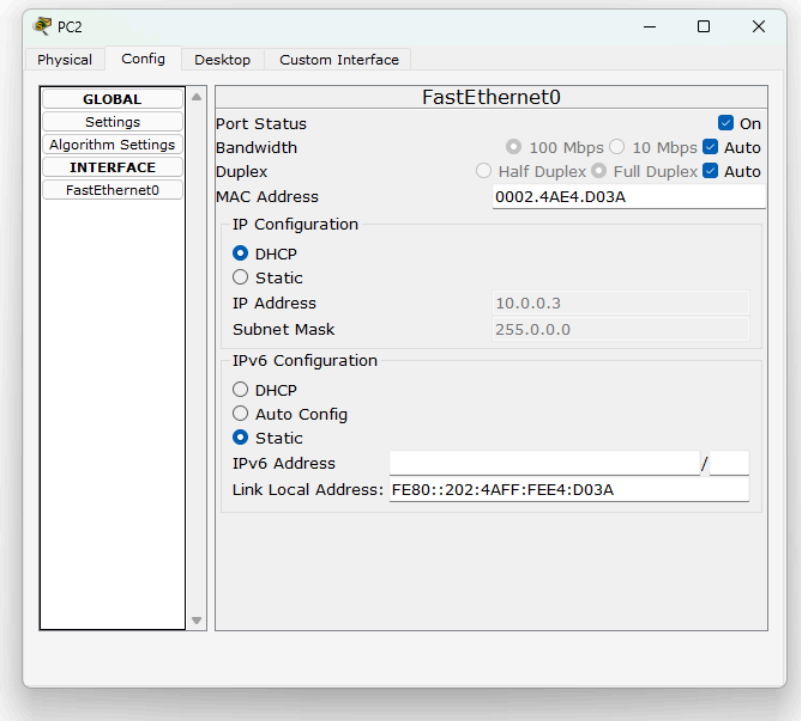
TOPOLOGY



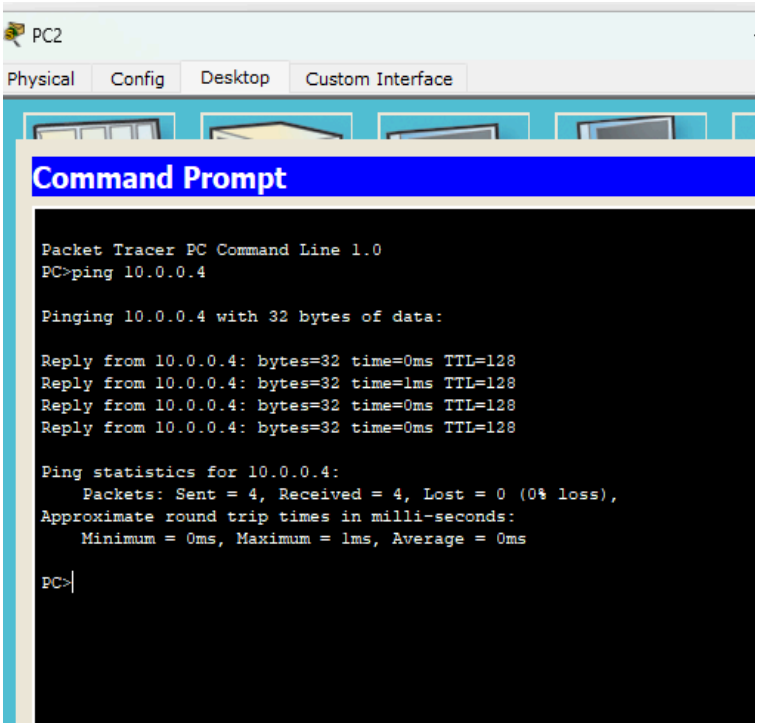
SERVER CONFIG



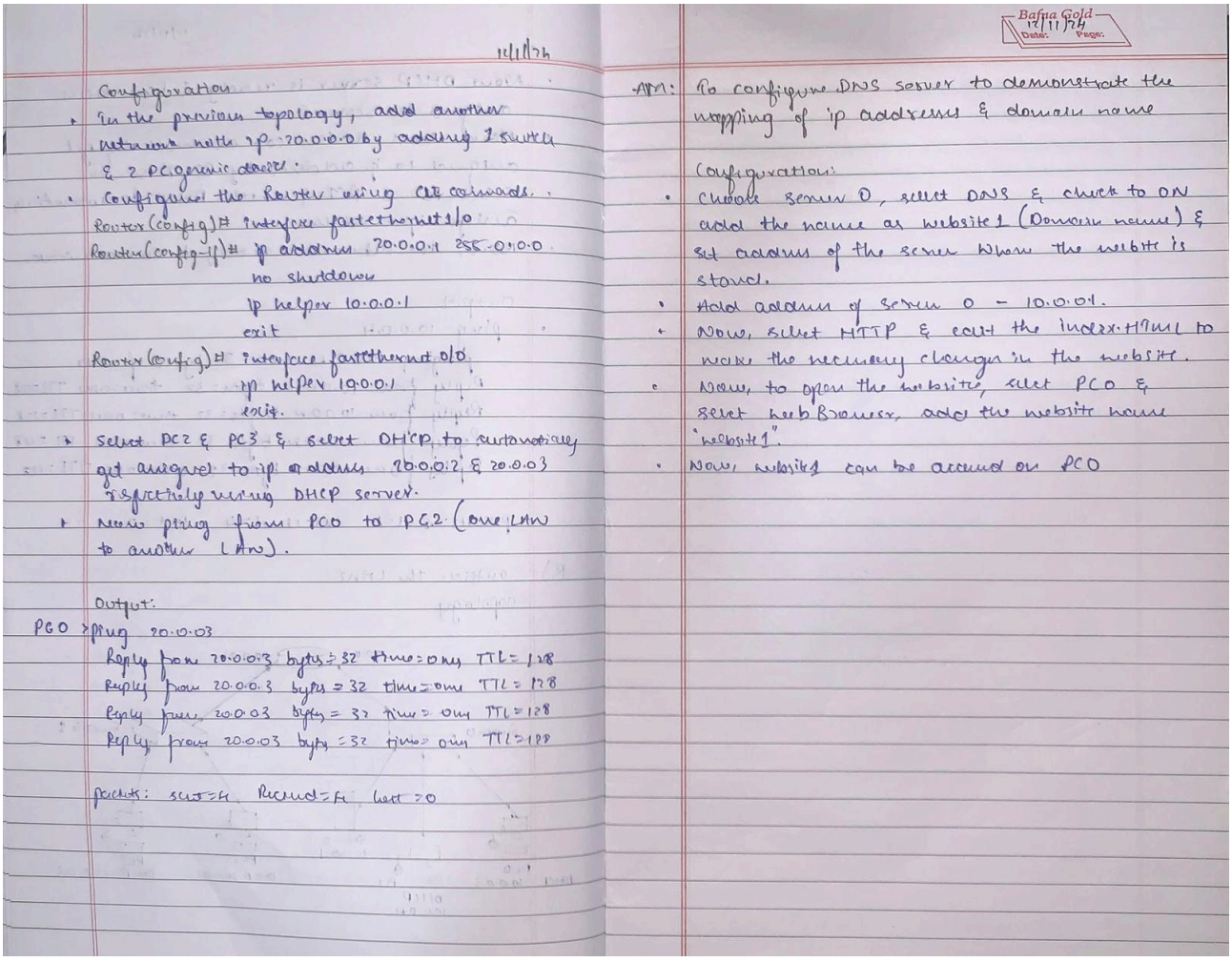
PC CONFIG



PING COMMAND



AIM: TO CONFIG IP ADDRESSES OF HOST USING DHCP SERVER PRESENT OUTSIDE SAME LAN



14/11/24

Configuration

- In the previous topology, add another network with ip 20.0.0.0 by adding 1 switch & 2 PC (generic device).
- Configure the Router using CLI commands.

Router(config)# interface fastethernet 1/0
Router(config-if)# ip address 20.0.0.1 255.0.0.0
no shutdown
ip helper 10.0.0.1
exit

Router(config)# interface fastethernet 0/0
ip helper 10.0.0.1
exit

- Select PC2 & PC3 & select DHCP to automatically get assigned to ip address 20.0.0.2 & 20.0.0.3 respectively using DHCP server.
- Now ping from PC0 to PC2 (one LAN to another LAN).

Output:

PC0 > ping 20.0.0.3

Reply from 20.0.0.3 bytes=32 time=0ms TTL=128
Reply from 20.0.0.3 bytes=32 time=0ms TTL=128
Reply from 20.0.0.3 bytes=32 time=0ms TTL=128
Reply from 20.0.0.3 bytes=32 time=0ms TTL=128

packets: sent=4 Received=4 lost=0

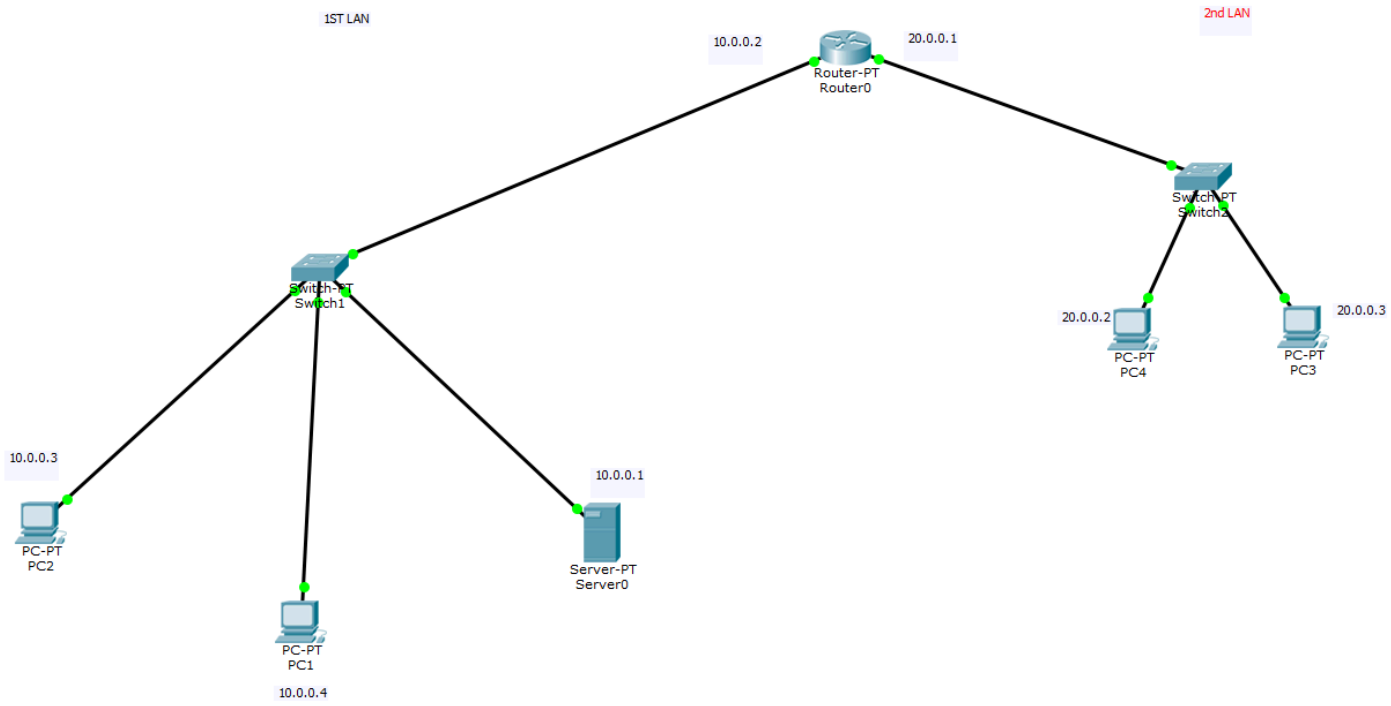
Bafna Gold 12/11/24 Date: Page:

AIM: To configure DNS server to demonstrate the mapping of ip addresses & domain name

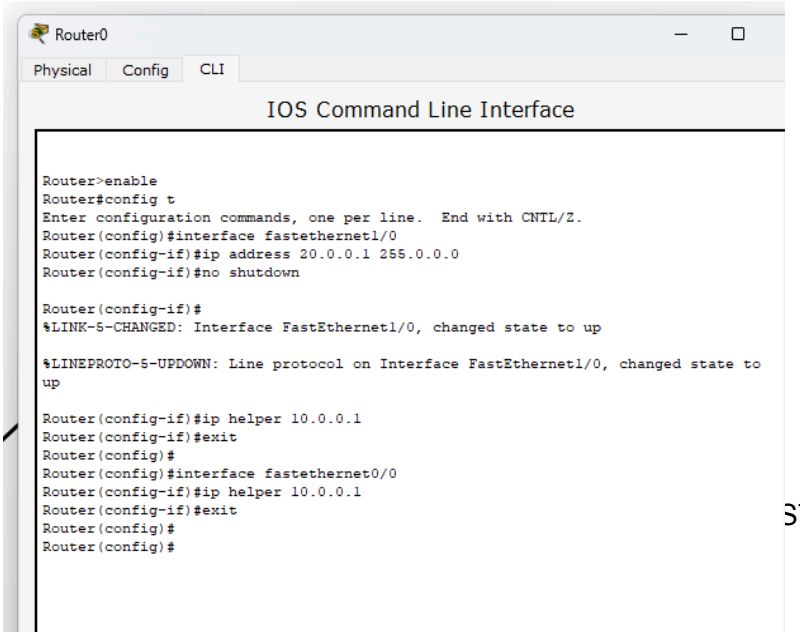
Configuration:

- Choose server 0, select DNS & check to ON add the name as website1 (Domain name) & set address of the server where the website is stored.
- Add address of server 0 - 10.0.0.1.
- Now, select HTTP & edit the index.html to make the necessary changes in the website.
- Now, to open the website, select PC0 & select Web Browser, add the website name "website1".
- Now, website1 can be accessed on PC0

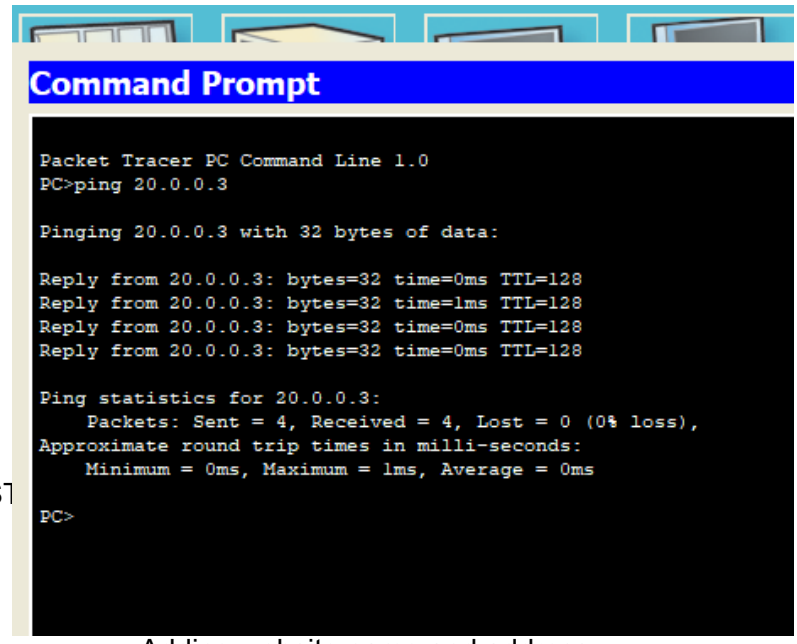
TOPOLOGY:



ROUTER CONFIG



PING COMMAND



DNS SERVER CONFIG

Adding website name and address

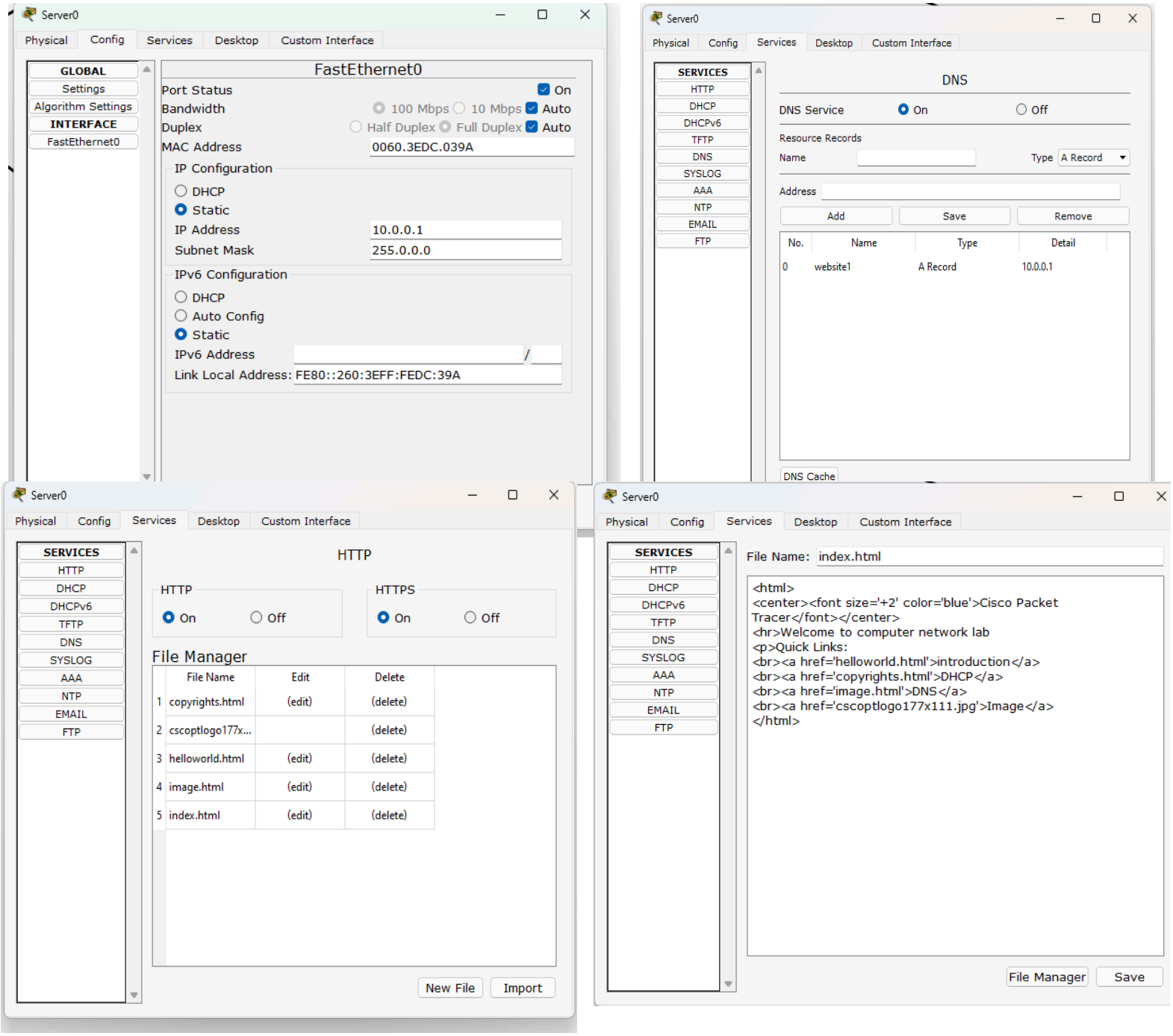


Fig: Accessing and changing index.html

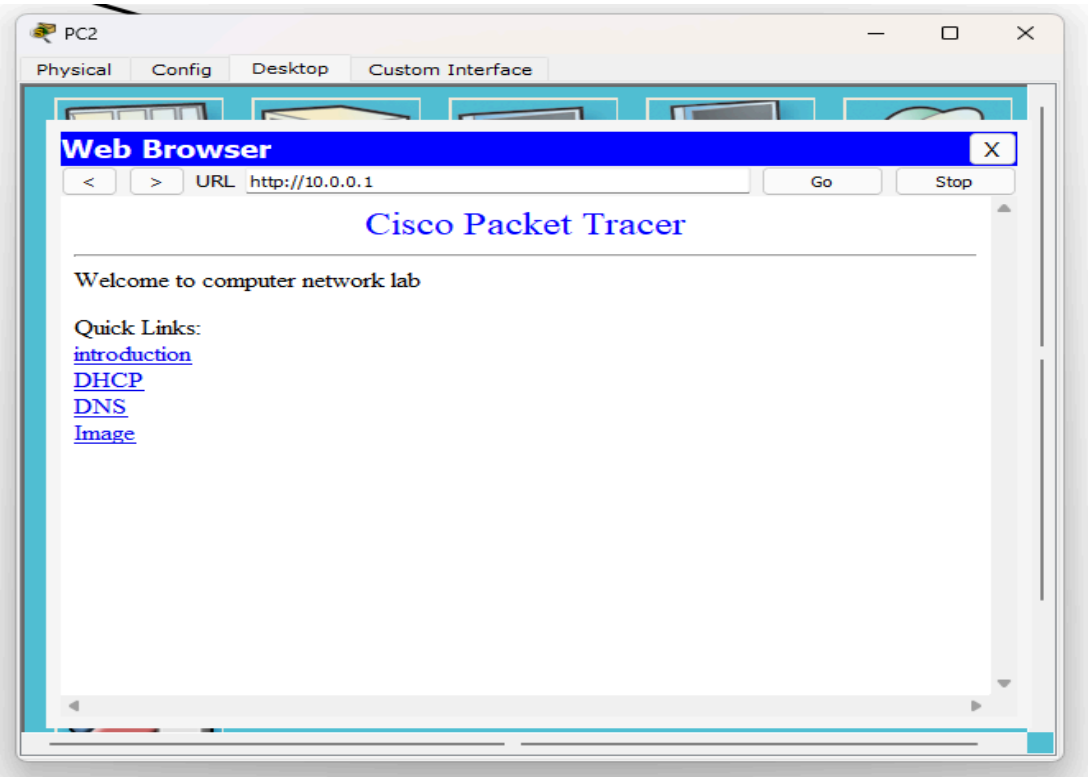


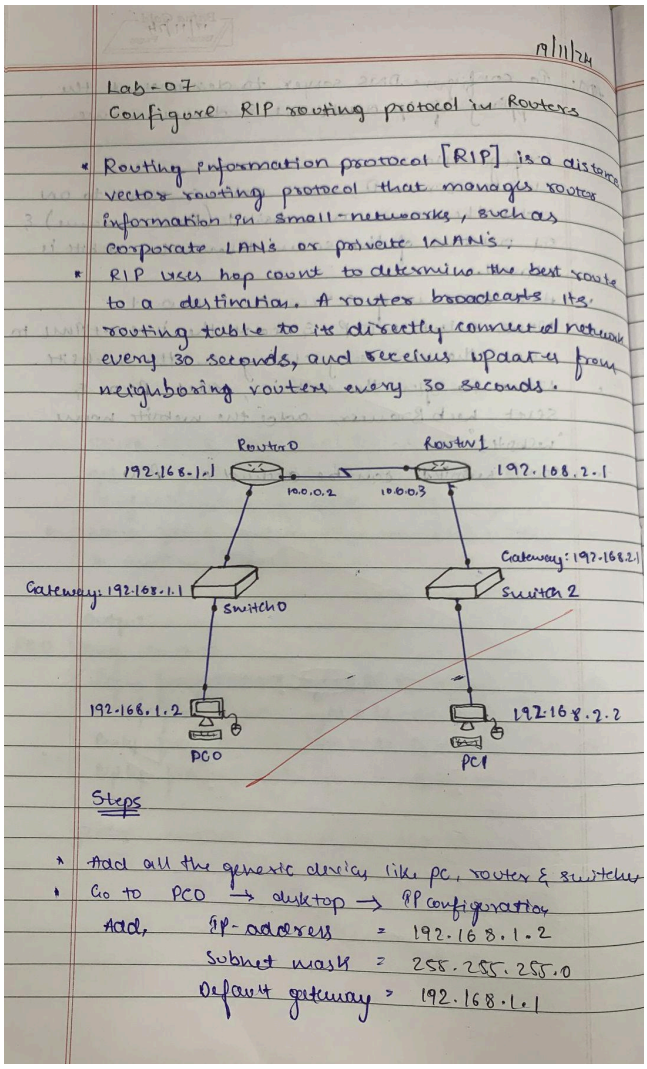
Fig : webBrowser output

NAME : MOHITH JAIN USN: 1BM22CS162

CN LAB 7

AIM: TO CONFIGURE RIP ROUTING PROTOCOL IN ROUTERS.

OBSERVATION:



Lab-07

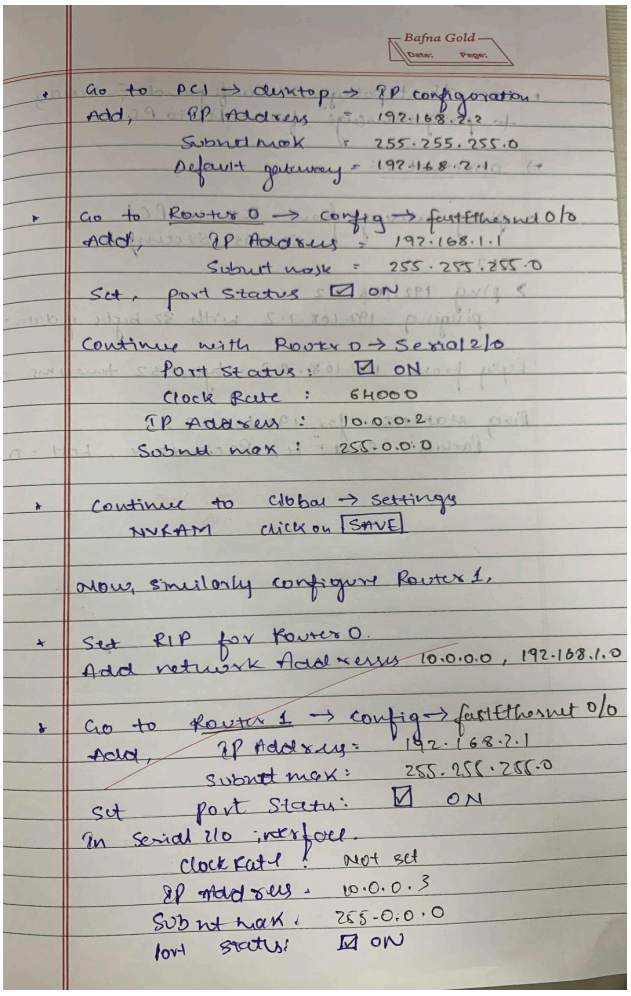
Configure RIP routing protocol in Routers

- Routing information protocol [RIP] is a distance vector routing protocol that manages router information in small-networks, such as corporate LAN's or private WANs.
- RIP uses hop count to determine the best route to a destination. A router broadcasts its routing table to its directly connected network every 30 seconds, and receives updates from neighboring routers every 30 seconds.

Steps

- Add all the generic devices like pc, router & switches
- Go to PC0 → desktop → IP configuration

Add, IP address = 192.168.1.2
Subnet mask = 255.255.255.0
Default gateway = 192.168.1.1



- Go to PC1 → desktop → IP configuration

Add, IP Address = 192.168.2.2
Subnet mask = 255.255.255.0
Default gateway = 192.168.2.1

- Go to Router 0 → config → fastEthernet 0/0

Add, IP Address = 192.168.1.1
Subnet mask = 255.255.255.0
Set, port Status ☑ ON

Continue with Router 0 → Serial 2/0
Port Status : ☑ ON
Clock Rate : 64000
IP Address : 10.0.0.2
Subnet mask : 255.0.0.0

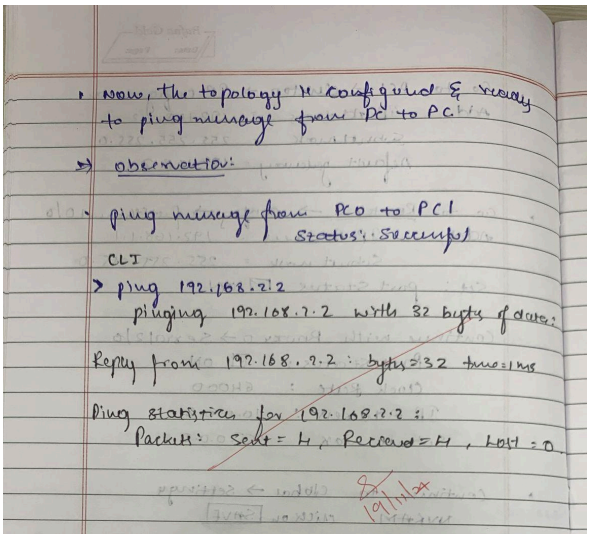
- Continue to Global → Settings
NVRAM click on [SAVE]

Now, similarly configure Router 1,

- Set RIP for Router 0.
Add network Address 10.0.0.0, 192.168.1.0

- Go to Router 1 → config → fastEthernet 0/0

Add, IP Address: 192.168.2.1
Subnet mask: 255.255.255.0
Set port Status: ☑ ON
In Serial 2/0 interface.
Clock Rate: Not set
IP address: 10.0.0.3
Subnet mask: 255.0.0.0
Port status: ☑ ON



- Now, the topology is configured & ready to ping message from PC to PC.

→ observation:

- ping message from PC0 to PC1
Status: Successful

CLI

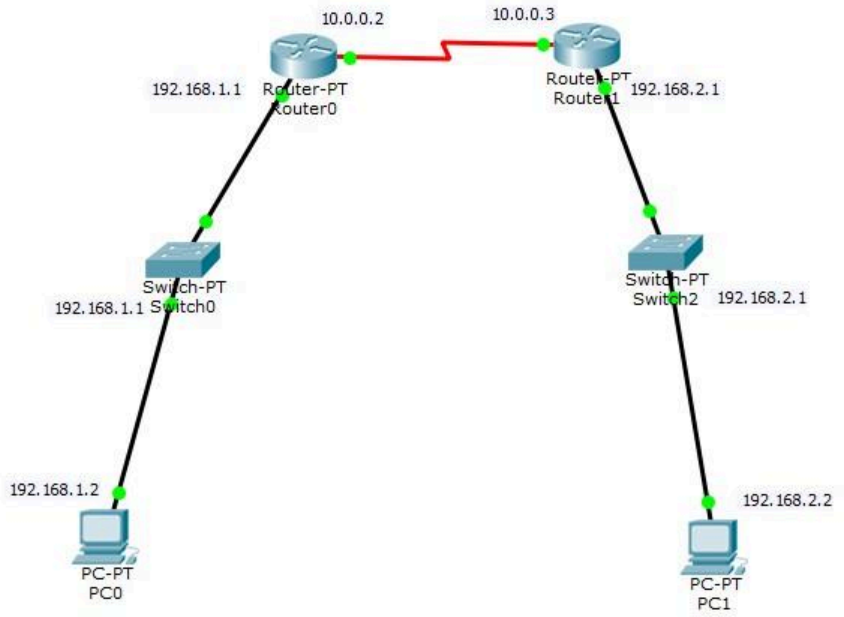
> ping 192.168.2.2

pinging 192.168.2.2 with 32 bytes of data:

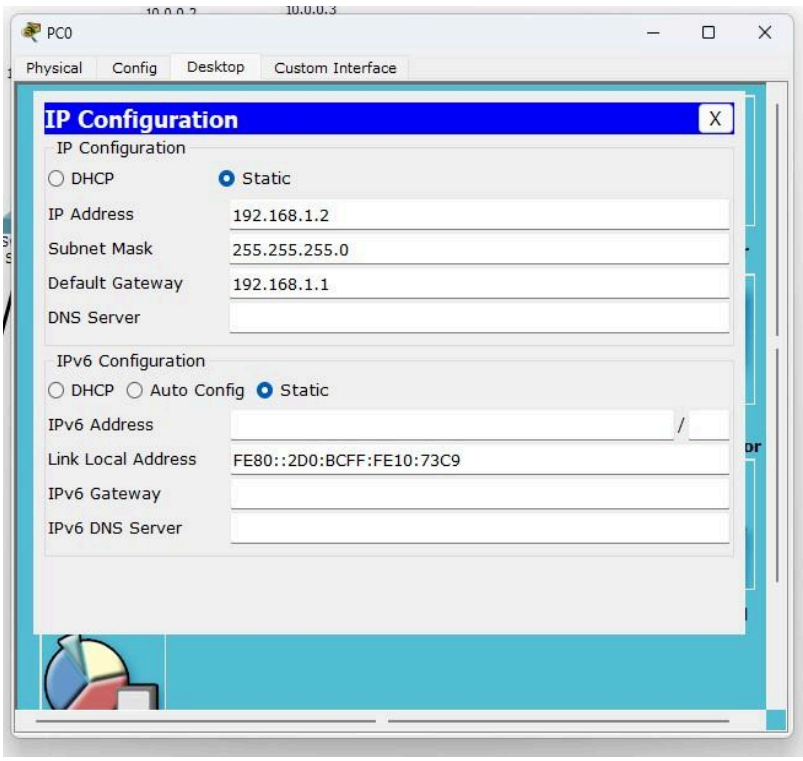
Reply from 192.168.2.2 : bytes=32 time=1ms

Ping statistics for 192.168.2.2 :
Packets: Sent = 4, Received = 4, Lost = 0

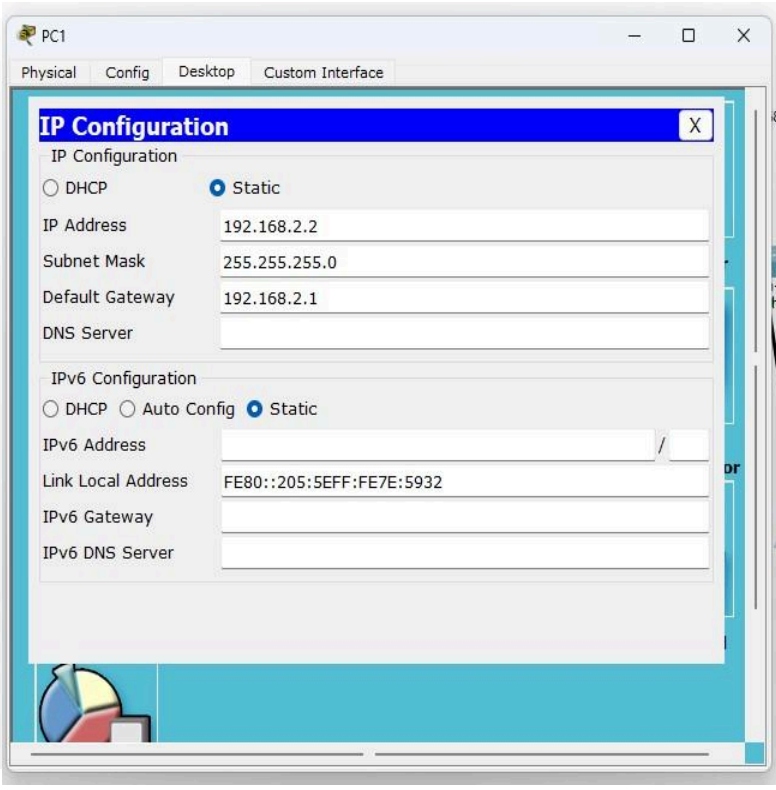
TOPOLOGY:



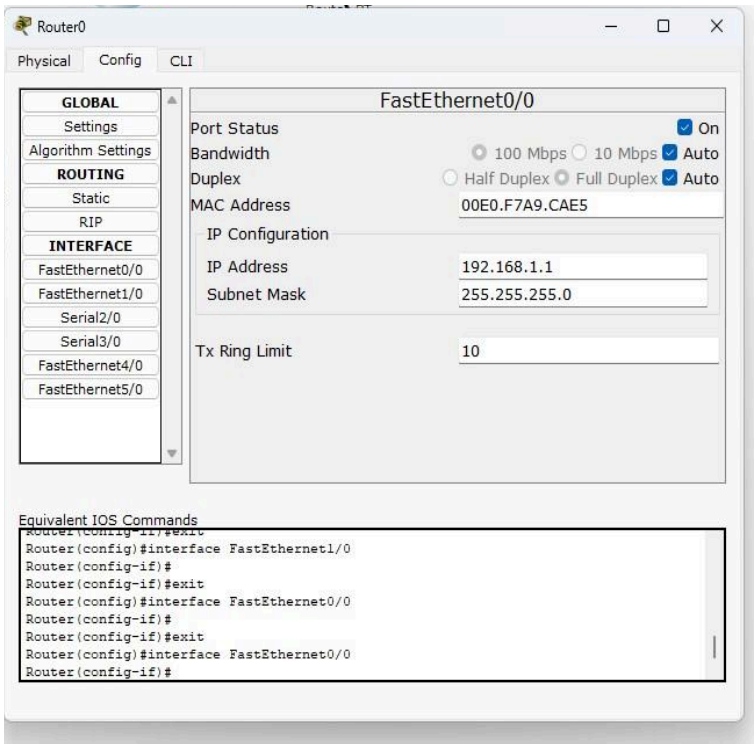
PC_0 CONFIGURATION:

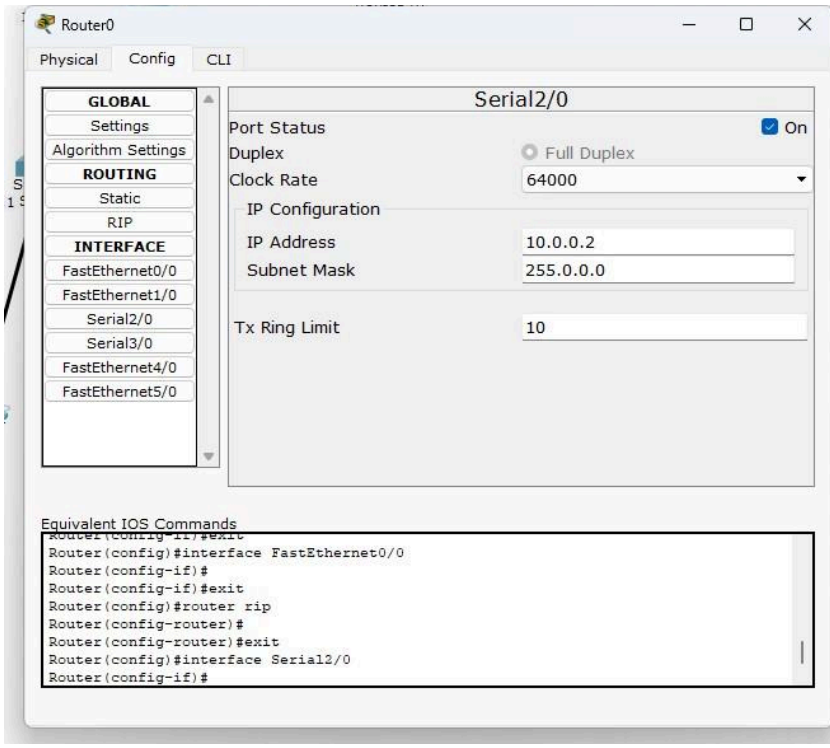


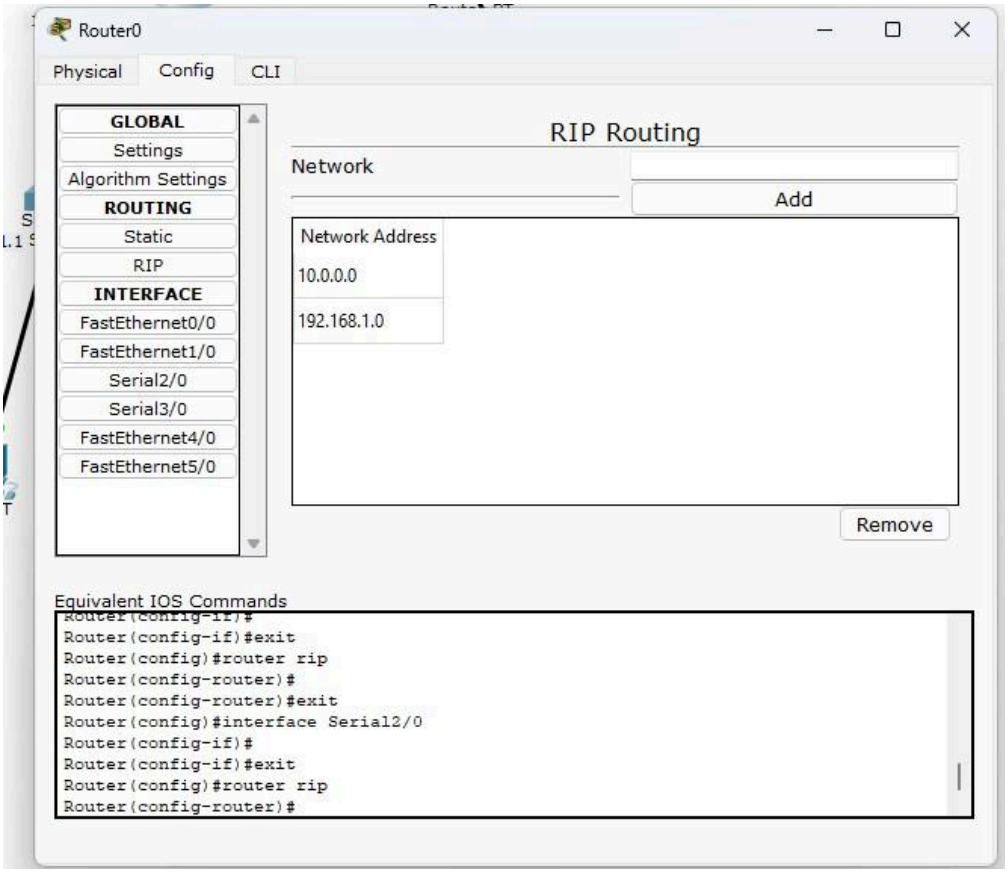
PC_1 CONFIGURATION:



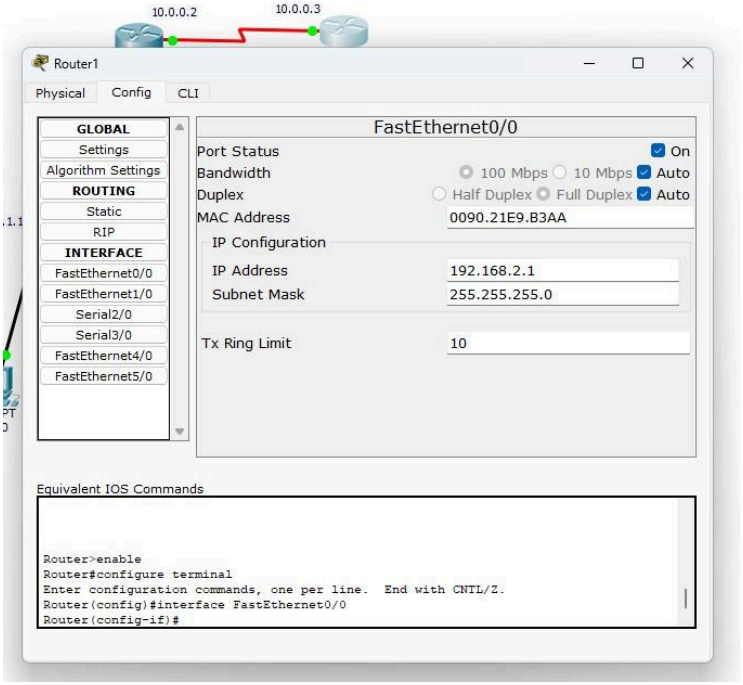
ROUTER_0 CONFIGURATION:

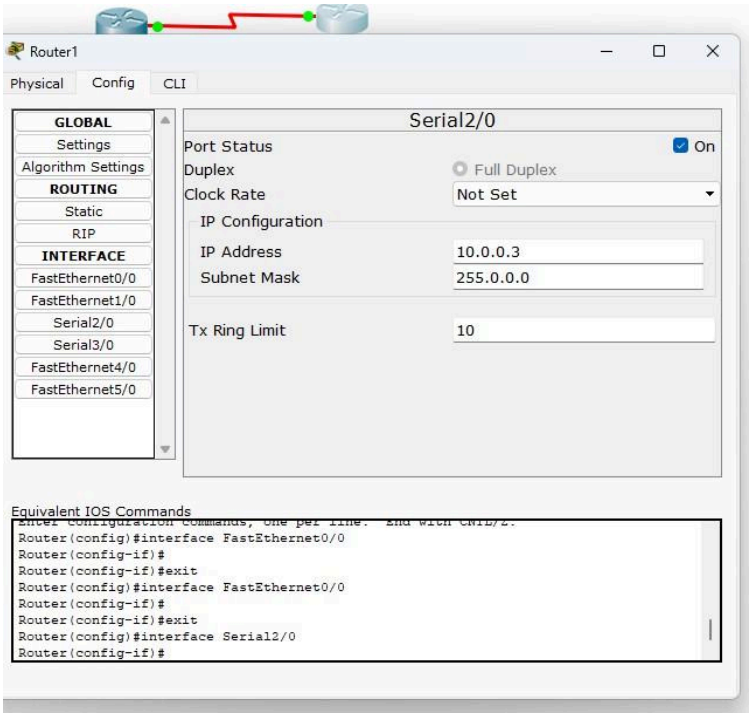


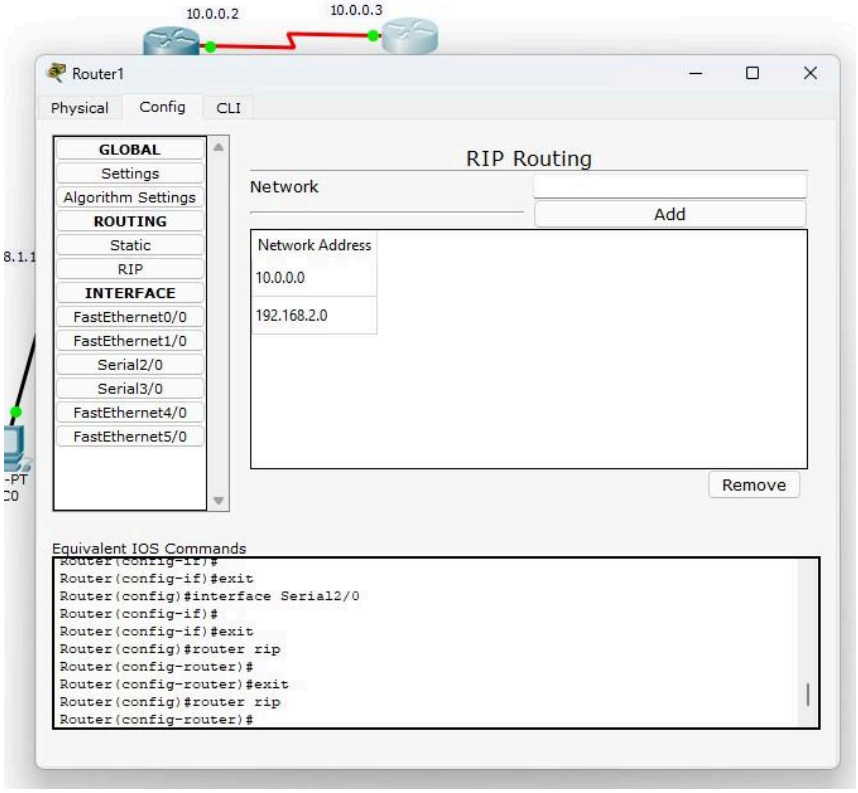




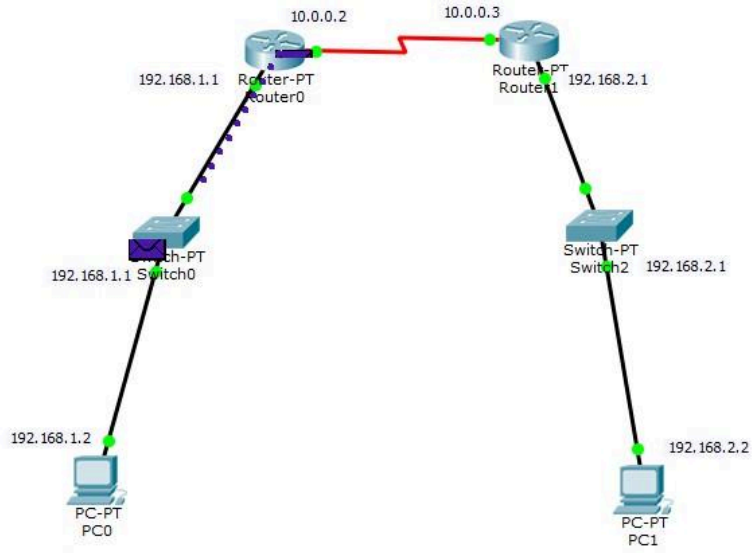
ROUTER_1 CONFIGURATION:

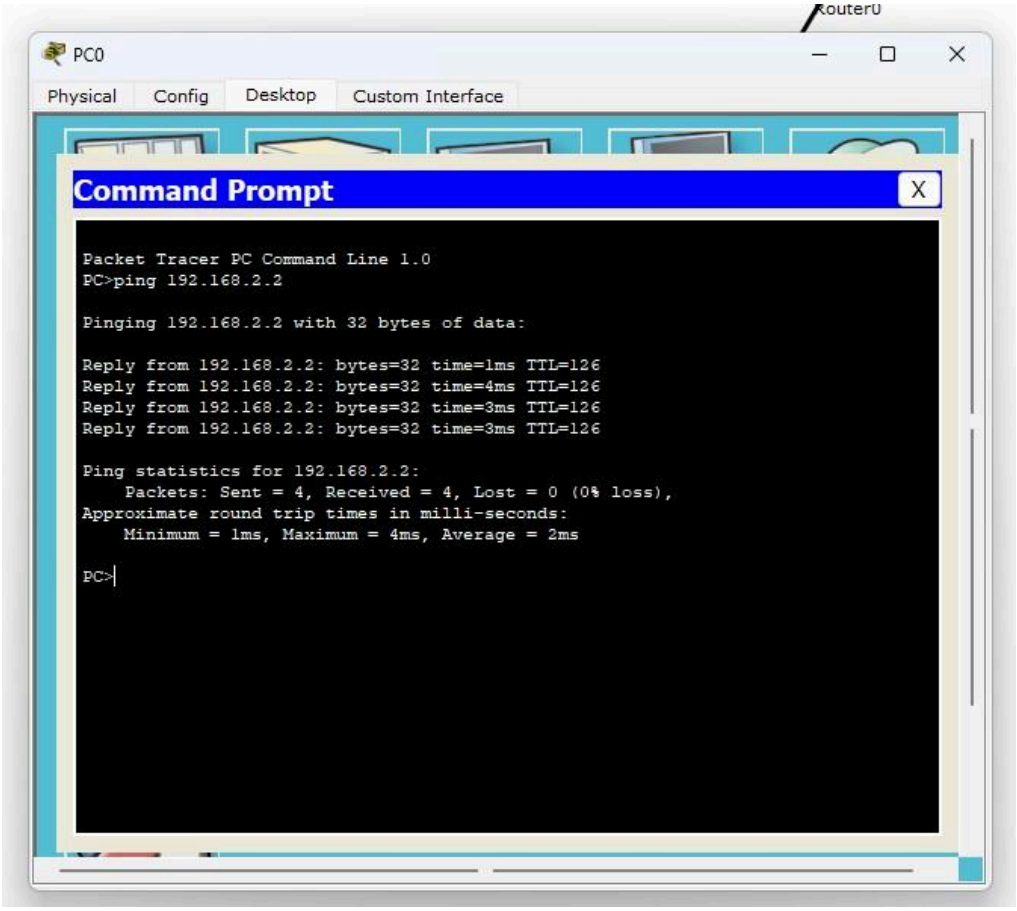






Packet transfer / ping



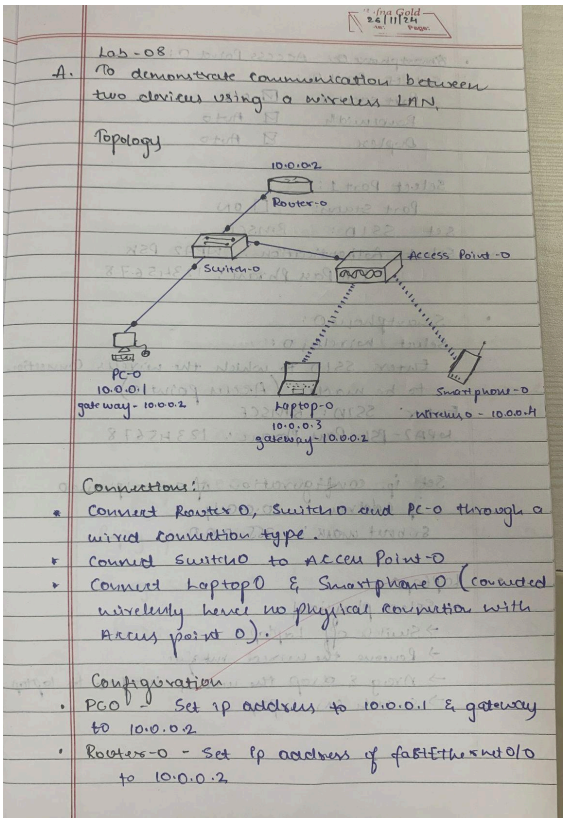


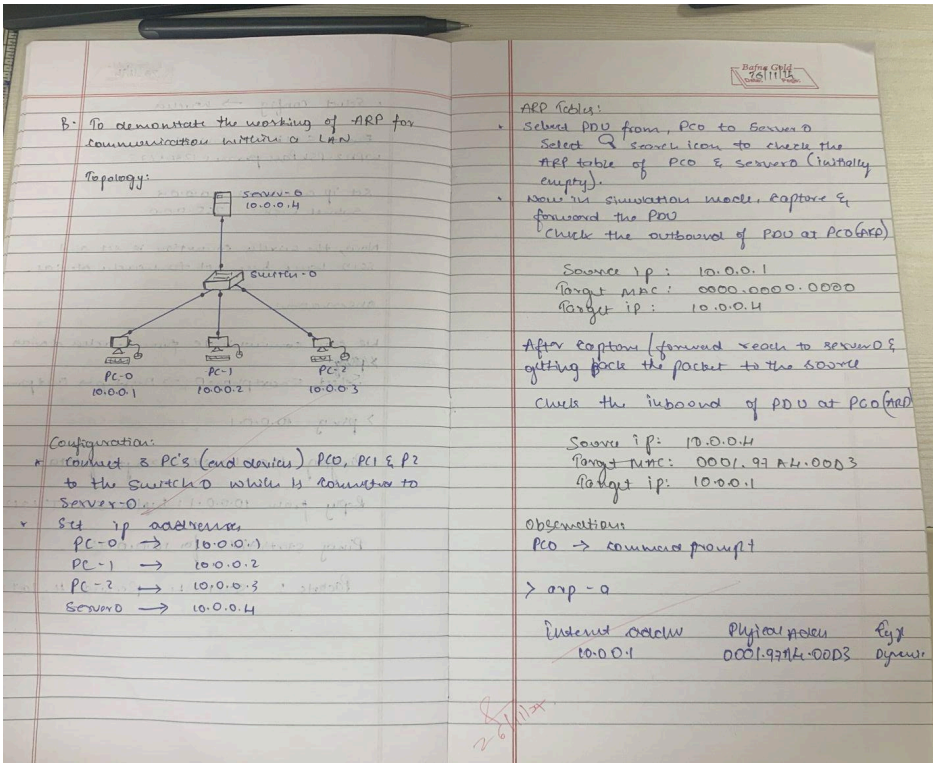
NAME : Mohith Jain USN: 1BM22CS162

CN LAB 8

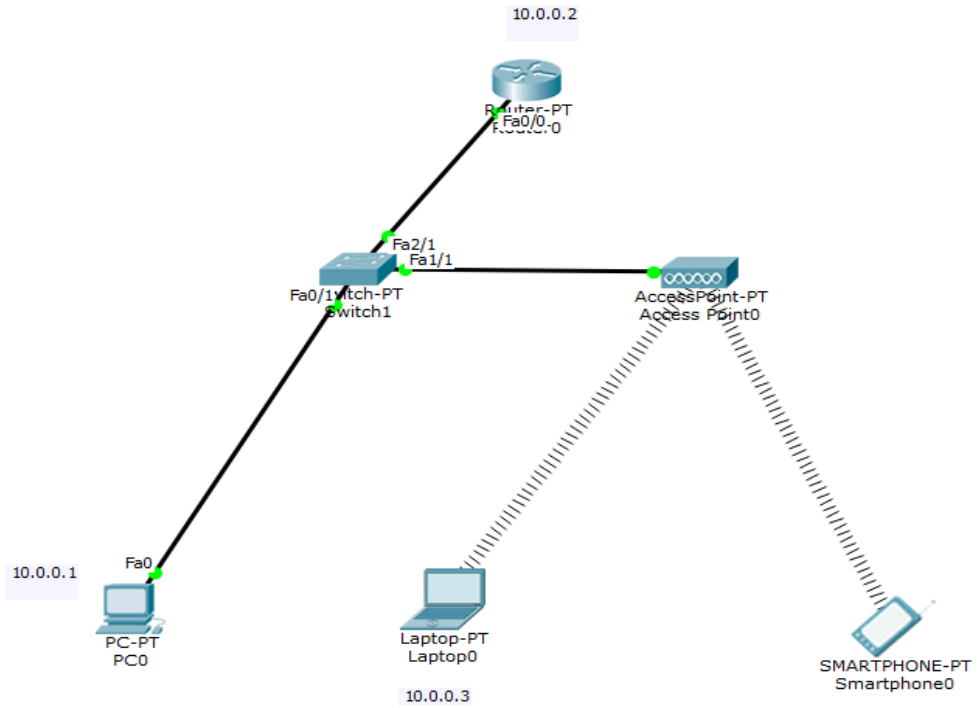
AIM: TO DEMONSTRATE COMMUNICATION B/W TWO DEVICES USING A WIRELESS LAN

OBSERVATION:

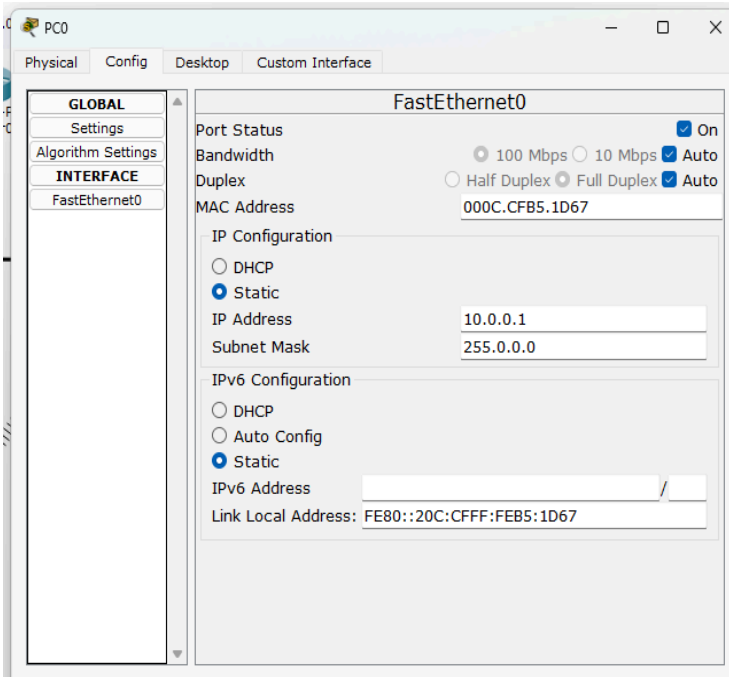




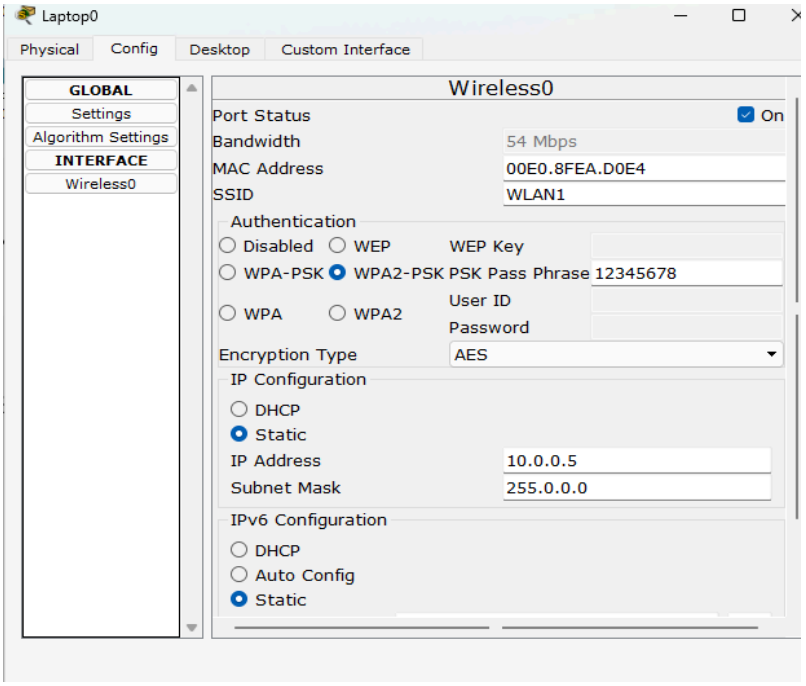
TOPOLOGY:



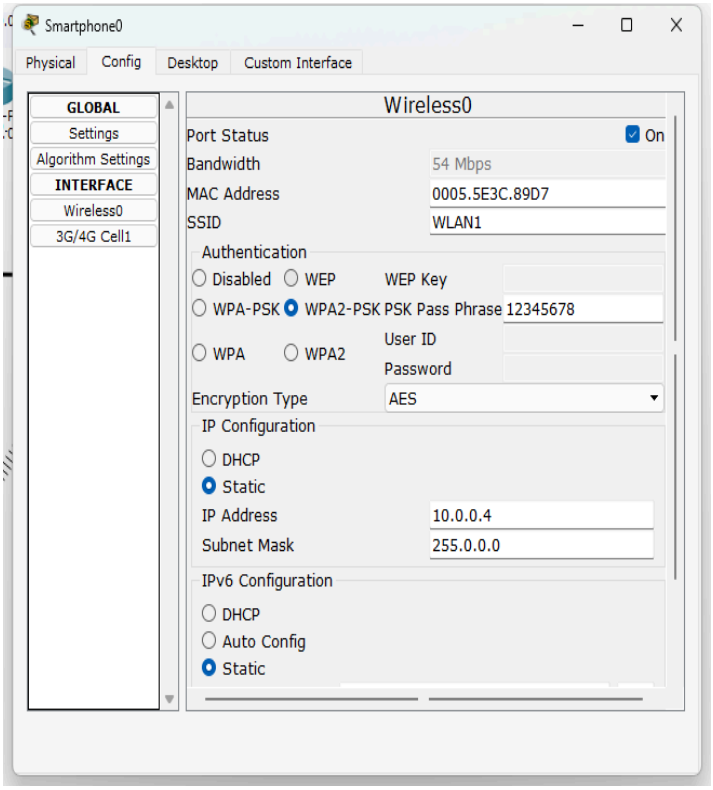
PC0 CONFIGURATION:



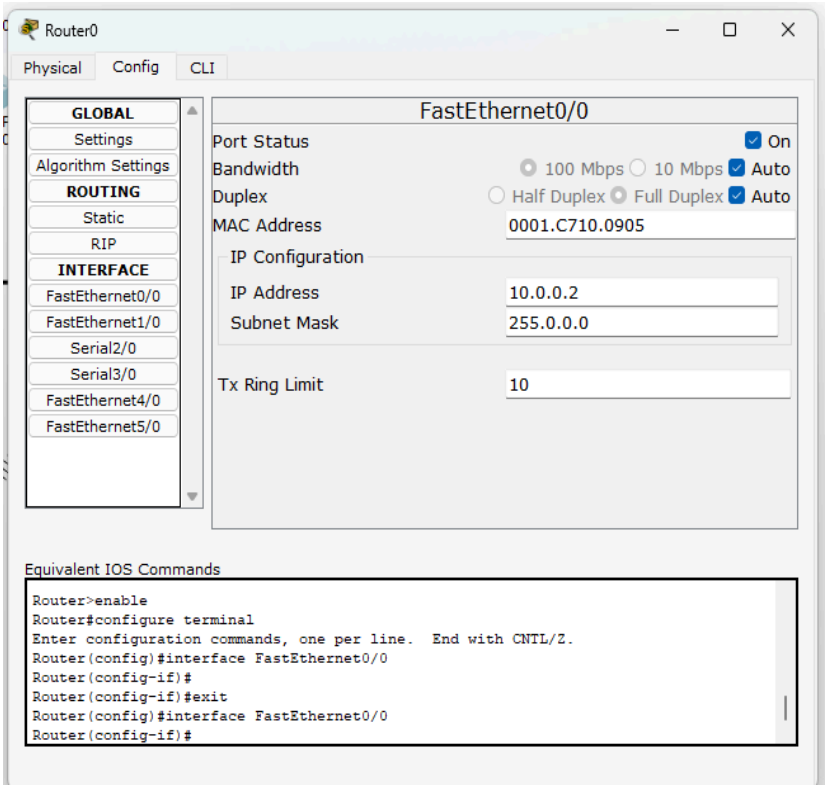
LAPTOP CONFIGURATION:



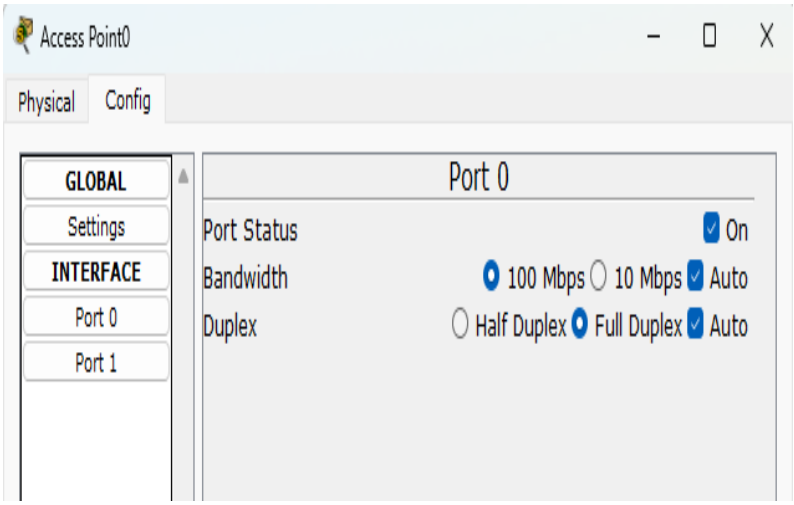
SMART-PHONE CONFIGURATION:



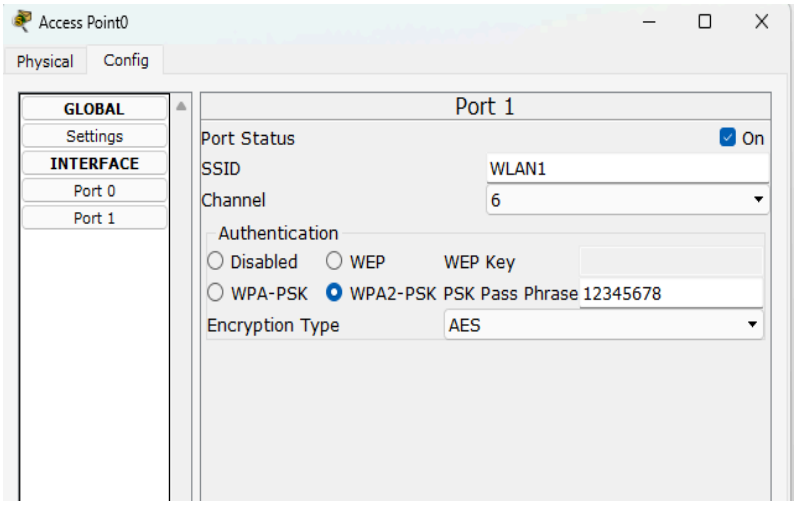
ROUTER0 CONFIGURATION:



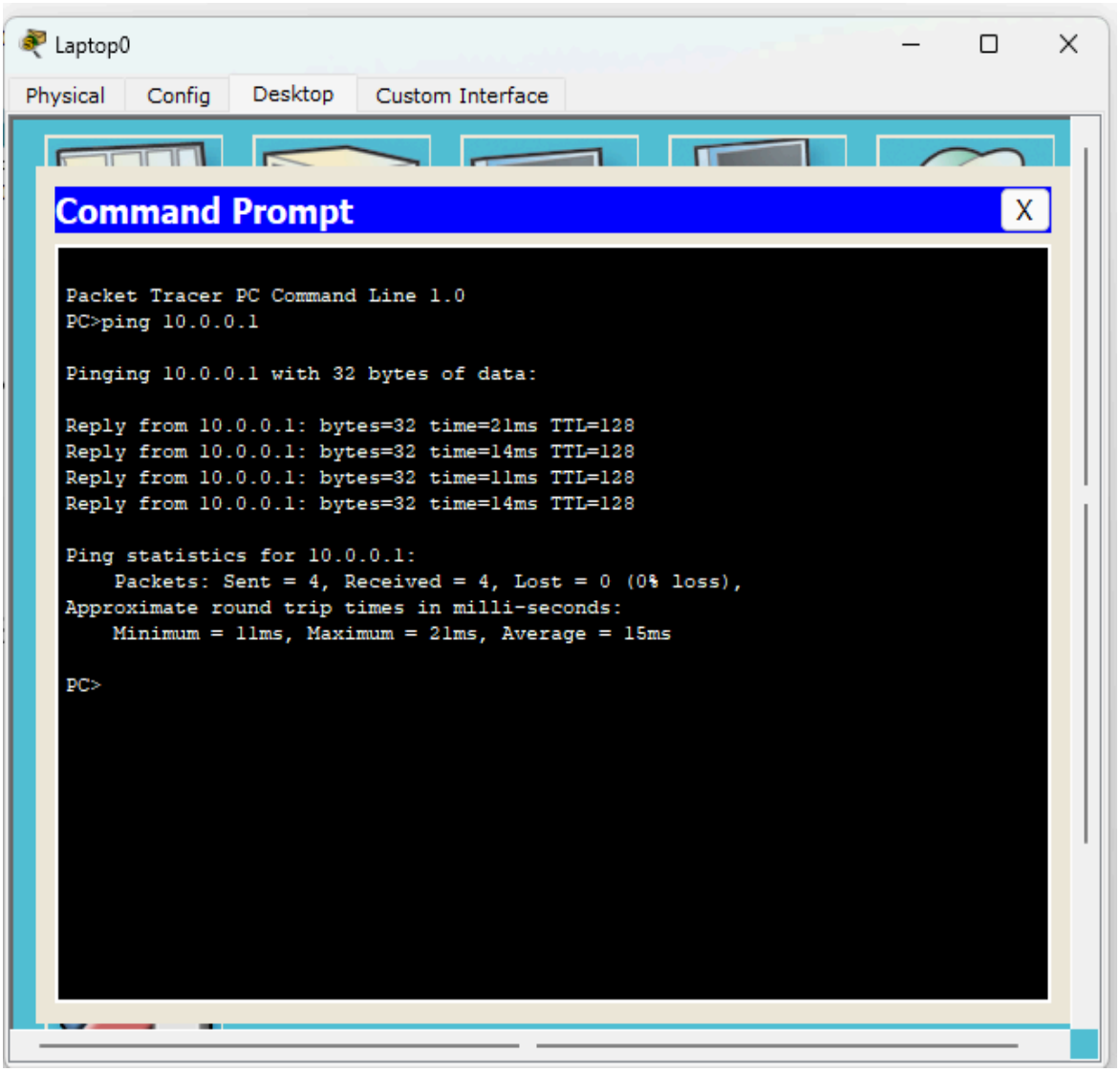
ACCESS-POINT PORT0



ACCESS-POINT PORT1

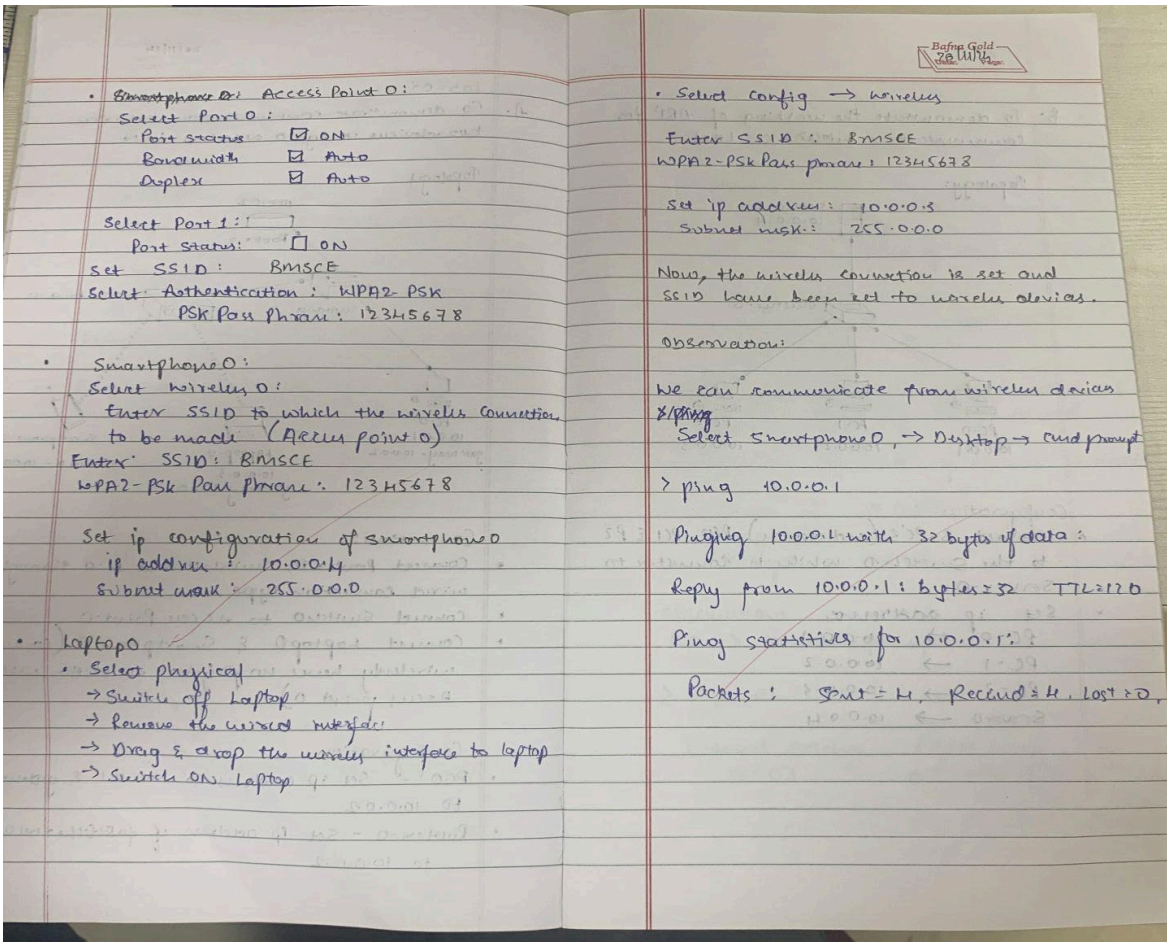


PING FROM LAPTOP TO PC



AIM: TO DEMONSTRATE THE WORKING OF ARP FOR COMMUNICATION WITHIN A LAN

OBSERVATION:



- ~~Smartphone 0~~: Access Point 0:
  Select Port 0:
  Port status ☑ ON
  Bandwidth ☑ Auto
  Duplex ☑ Auto

  Select Port 1:
  Port Status: ☐ ON
  Set SSID: BMSCE
  Select Authentication: WPA2-PSK
  PSK Pass Phrase: 12345678

- Smartphone 0:
  Select wireless 0:
  Enter SSID to which the wireless connection to be made (Access point 0)
  Enter SSID: BMSCE
  WPA2-PSK Pass Phrase: 12345678

  Set ip configuration of smartphone 0
  ip address: 10.0.0.4
  Subnet mask: 255.0.0.0

- Laptop 0
  - Select physical
    → Switch off Laptop
    → Remove the wired interface
    → Drag & drop the wireless interface to laptop
    → Switch ON Laptop

- Select config → wireless
  Enter SSID: BMSCE
  WPA2-PSK Pass phrase: 12345678

  Set ip address: 10.0.0.5
  Subnet mask: 255.0.0.0

  Now, the wireless connection is set and SSID have been set to wireless devices.

  Observation:

  We can communicate from wireless devices ~~s/ping~~
  Select Smartphone 0, → Desktop → cmd prompt

  > ping 10.0.0.1

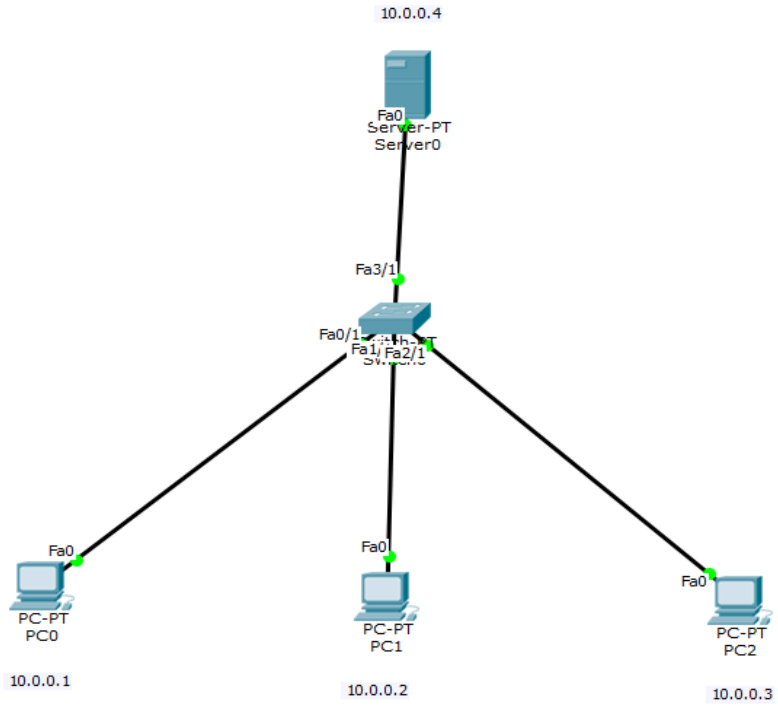
  Pinging 10.0.0.1 with 32 bytes of data:

  Reply from 10.0.0.1: bytes=32 TTL=128

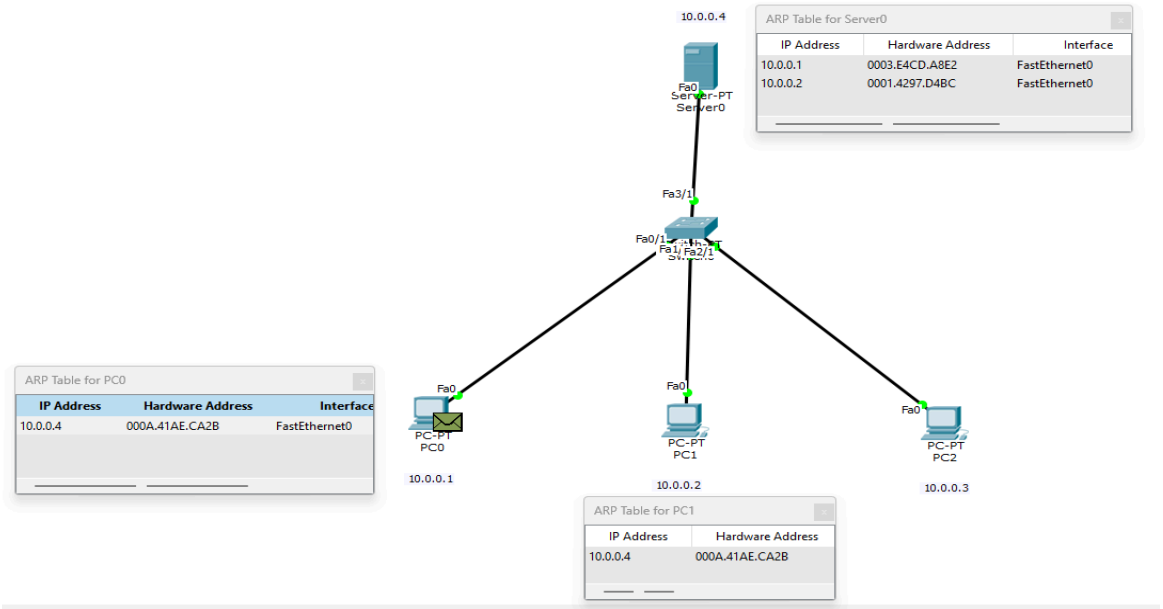
  Ping statistics for 10.0.0.1:

  Packets: Sent = 4, Received = 4, Lost = 0,

TOPOLOGY:



ARP TABLES:



PC0 INBOUND

PDU Information at Device: PC0

OSI Model | Inbound PDU Details

PDU Formats

Ethernet II

0	4	8	14	19 Byt
PREAMBLE: 101010...1011		DEST MAC: 0003.E4CD.A8E2	SRC MAC: 000A.41AE.CA2B	
TYPE: 0x800	DATA (VARIABLE LENGTH)		FCS: 0x0	

IP

0	4	8	16 19	31 Bits
4	IHL	DSCP: 0x0	TL: 28	
ID: 0x6		0x0	0x0	
TTL: 128		PRO: 0x1	CHKSUM	
SRC IP: 10.0.0.4				
DST IP: 10.0.0.1				
OPT: 0x0			0x0	
DATA (VARIABLE LENGTH)				

ICMP

0	8	16	31 Bits
TYPE: 0x0	CODE: 0x0	CHECKSUM	
ID: 0x6		SEQ NUMBER: 5	

PC0 OUTBOUND

PDU Information at Device: PC0

OSI Model | Outbound PDU Details

PDU Formats

0	4	8	14	19 Byt
PREAMBLE: 101010...1011		DEST MAC: 000A.41AE.CA2B	SRC MAC: 0003.E4CD.A8E2	
TYPE: 0x800	DATA (VARIABLE LENGTH)		FCS: 0x0	

IP

0	4	8	16 19	31 Bits
4	IHL	DSCP: 0x0	TL: 28	
ID: 0x5		0x0	0x0	
TTL: 255		PRO: 0x1	CHKSUM	
SRC IP: 10.0.0.1				
DST IP: 10.0.0.4				
OPT: 0x0			0x0	
DATA (VARIABLE LENGTH)				

ICMP

0	8	16	31 Bits
TYPE: 0x8	CODE: 0x0	CHECKSUM	
ID: 0x6		SEQ NUMBER: 5	

PC1 INBOUND

PDU Information at Device: Server0

OSI Model | Inbound PDU Details | Outbound PDU Details

PDU Formats

Ethernet II

0	4	8	14	19 Bytes
PREAMBLE: 101010...1011		DEST MAC: 000A.41AE.CA2B	SRC MAC: 0001.4297.D4BC	
TYPE: 0x800	DATA (VARIABLE LENGTH)		FCS: 0x0	

IP

0	4	8	16 19	31 Bits
4	IHL	DSCP: 0x0	TL: 28	
ID: 0x2		0x0	0x0	
TTL: 255		PRO: 0x1	CHKSUM	
SRC IP: 10.0.0.2				
DST IP: 10.0.0.4				
OPT: 0x0			0x0	
DATA (VARIABLE LENGTH)				

ICMP

0	8	16	31 Bits
TYPE: 0x8	CODE: 0x0	CHECKSUM	
ID: 0x3		SEQ NUMBER: 2	

PC1 OUTBOUND

PDU Information at Device: Server0

OSI Model | Inbound PDU Details | Outbound PDU Details

PDU Formats

Ethernet II

0	4	8	14	19 Bytes
PREAMBLE: 101010...1011		DEST MAC: 0001.4297.D4BC	SRC MAC: 000A.41AE.CA2B	
TYPE: 0x800	DATA (VARIABLE LENGTH)		FCS: 0x0	

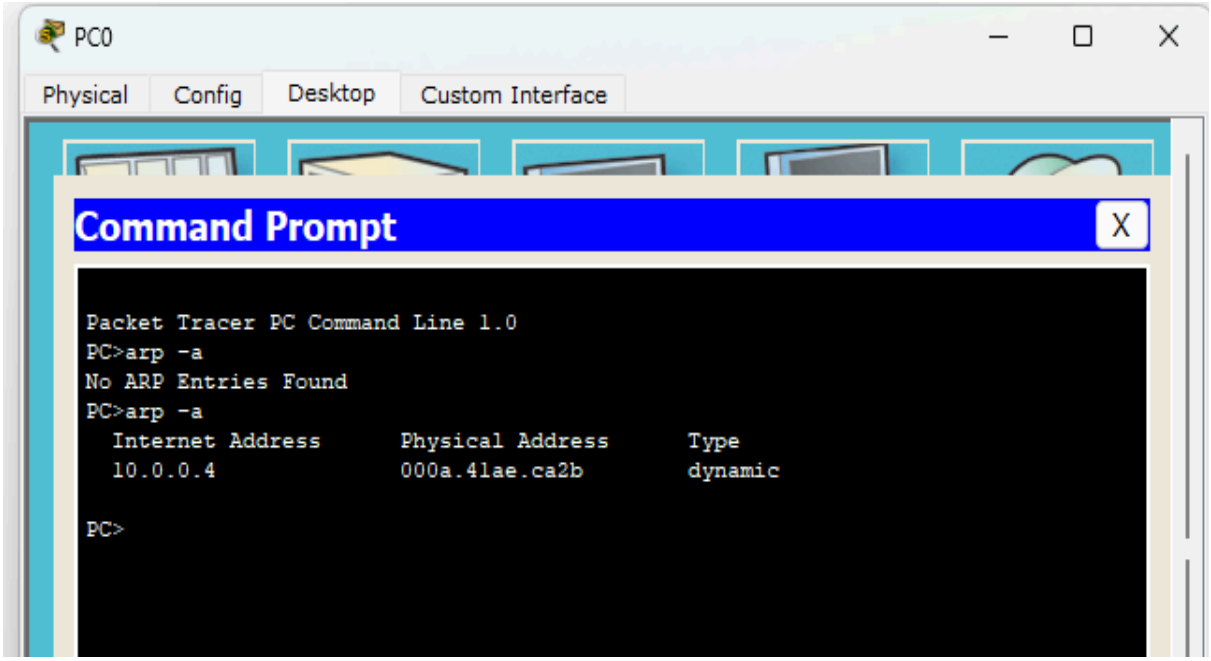
IP

0	4	8	16 19	31 Bits
4	IHL	DSCP: 0x0	TL: 28	
ID: 0x7		0x0	0x0	
TTL: 128		PRO: 0x1	CHKSUM	
SRC IP: 10.0.0.4				
DST IP: 10.0.0.2				
OPT: 0x0			0x0	
DATA (VARIABLE LENGTH)				

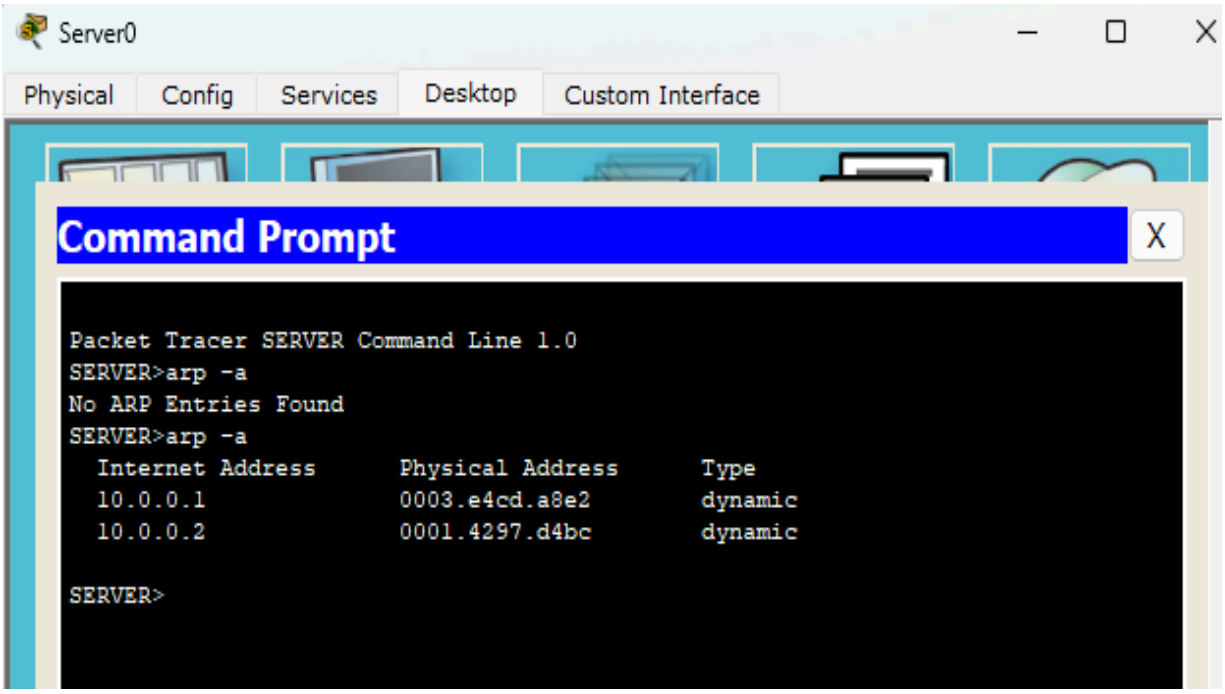
ICMP

0	8	16	31 Bits
TYPE: 0x0	CODE: 0x0	CHECKSUM	
ID: 0x3		SEQ NUMBER: 2	

PING FROM PC0



PING FROM SERVER0

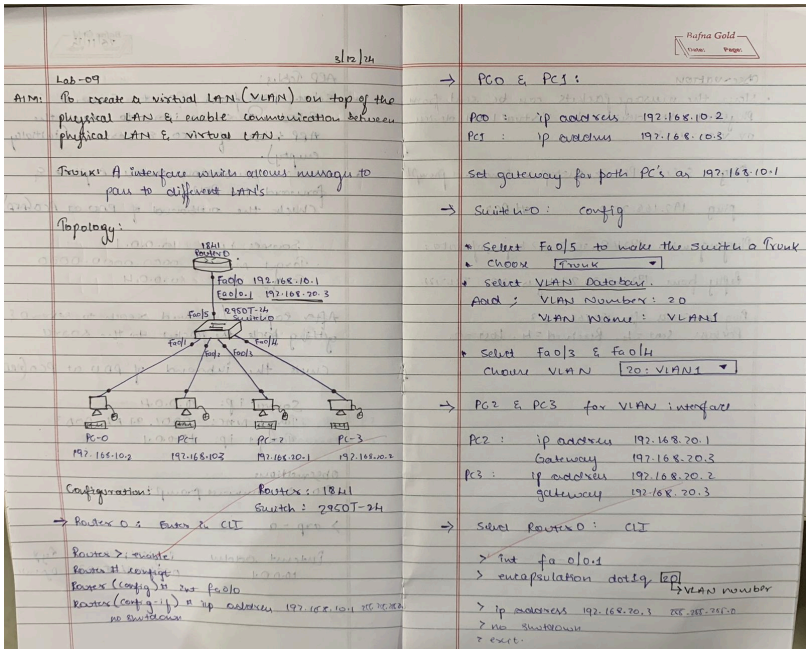


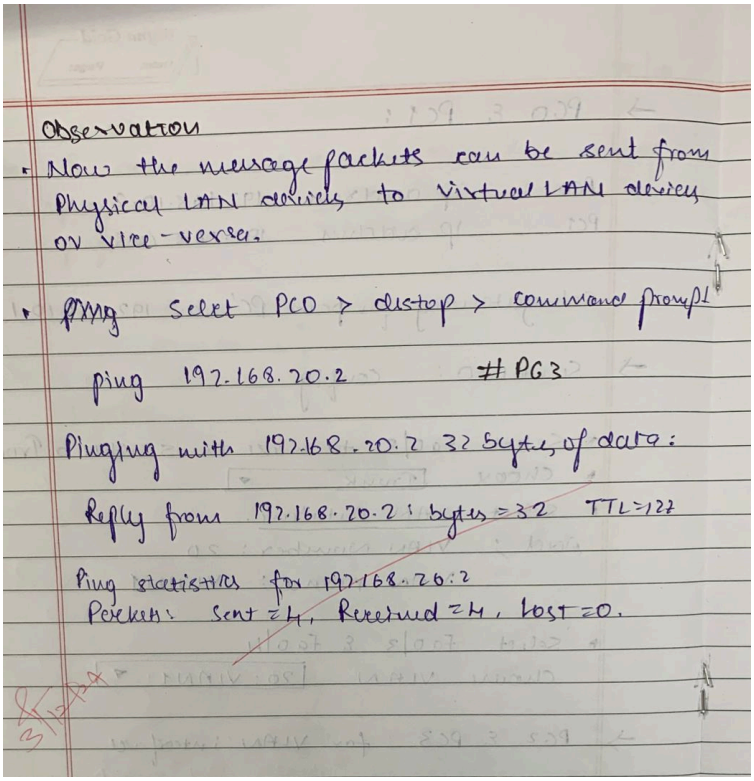
NAME : Mohith Jain USN: 1BM22CS162

CN LAB 9

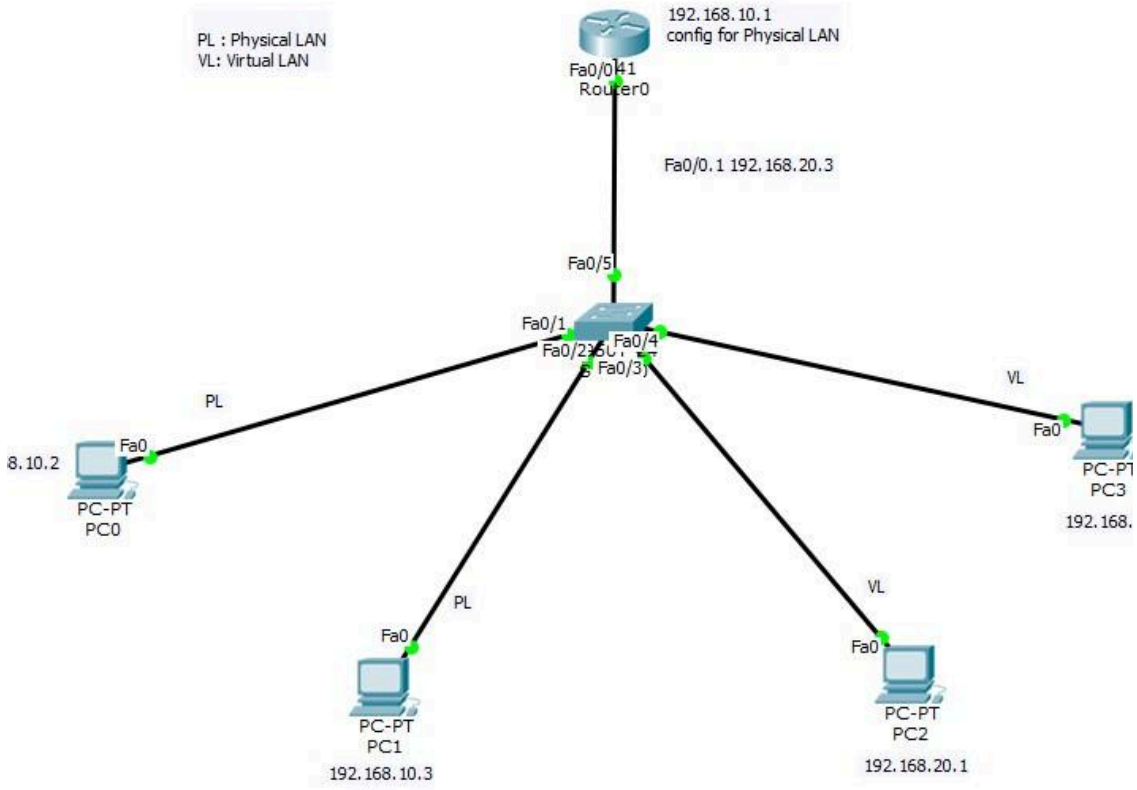
AIM: TO CREATE A VIRTUAL LAN ON TOP OF THE PHYSICAL LAN AND ENABLE COMMUNICATION B/W PHYSICAL LAN AND VIRTUAL LAN

OBSERVATION:

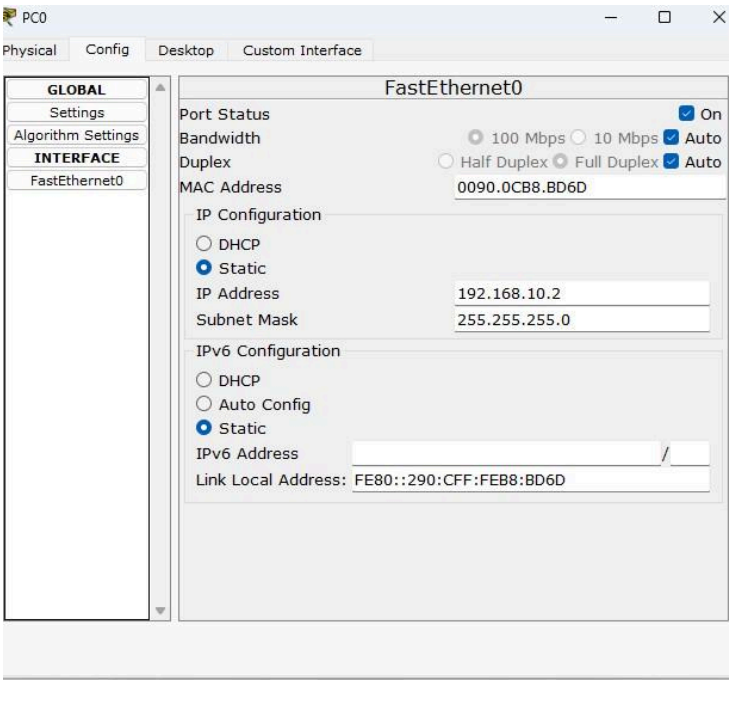




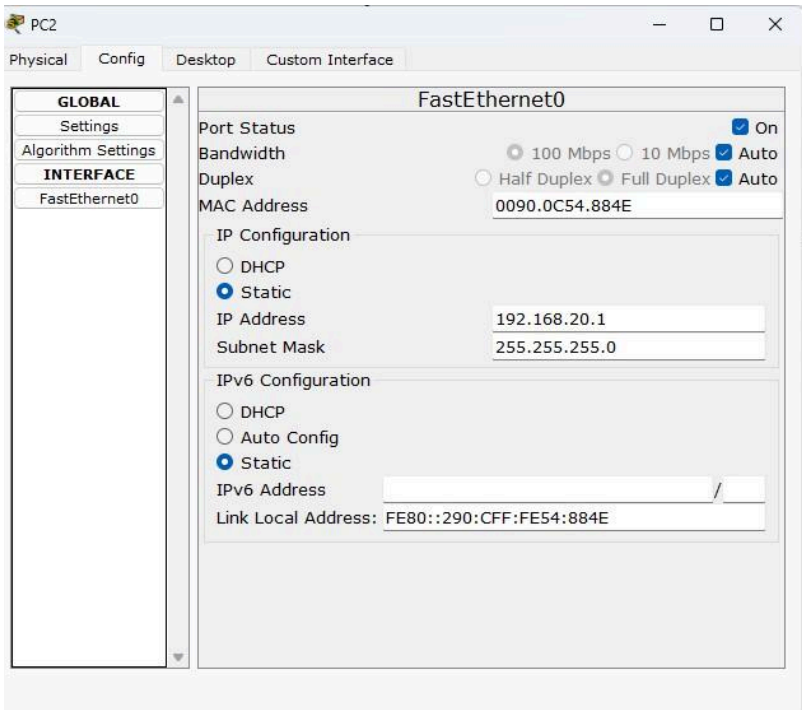
TOPOLOGY:



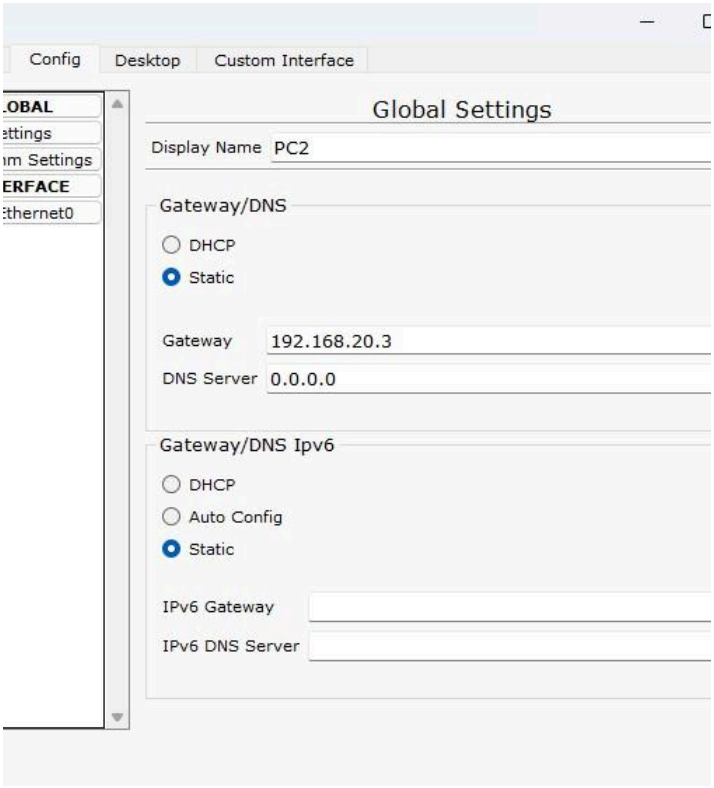
PC0 CONFIGURATION:



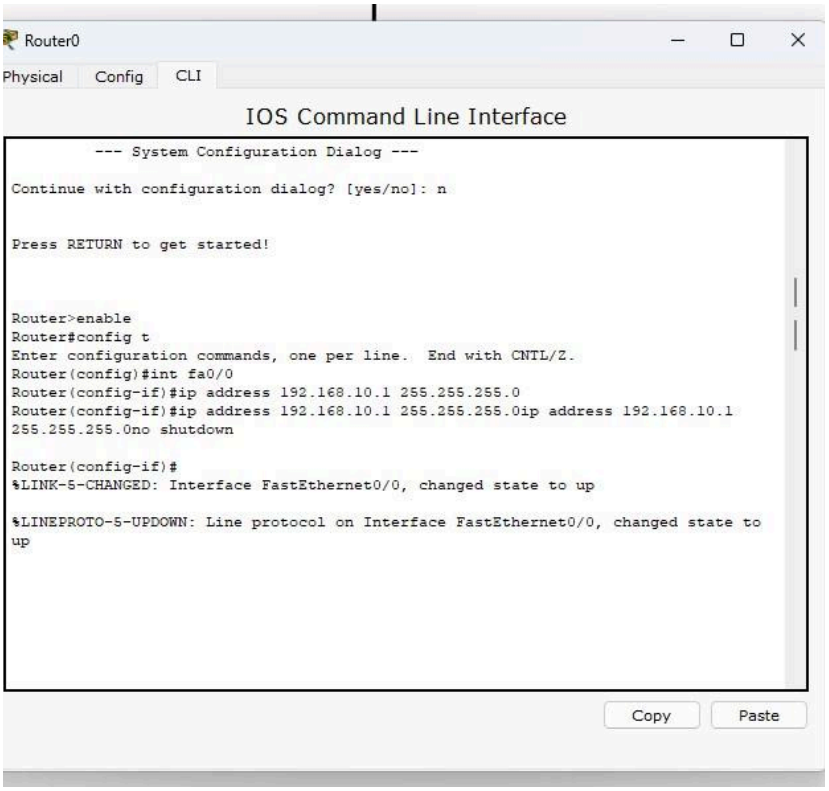
PC2 CONFIGURATION:



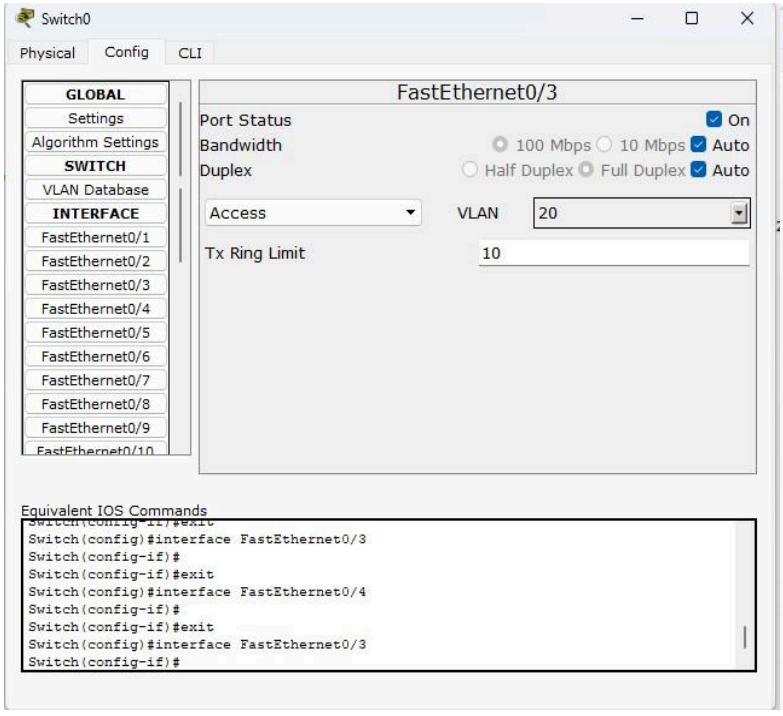
PC2 Gateway CONFIGURATION:



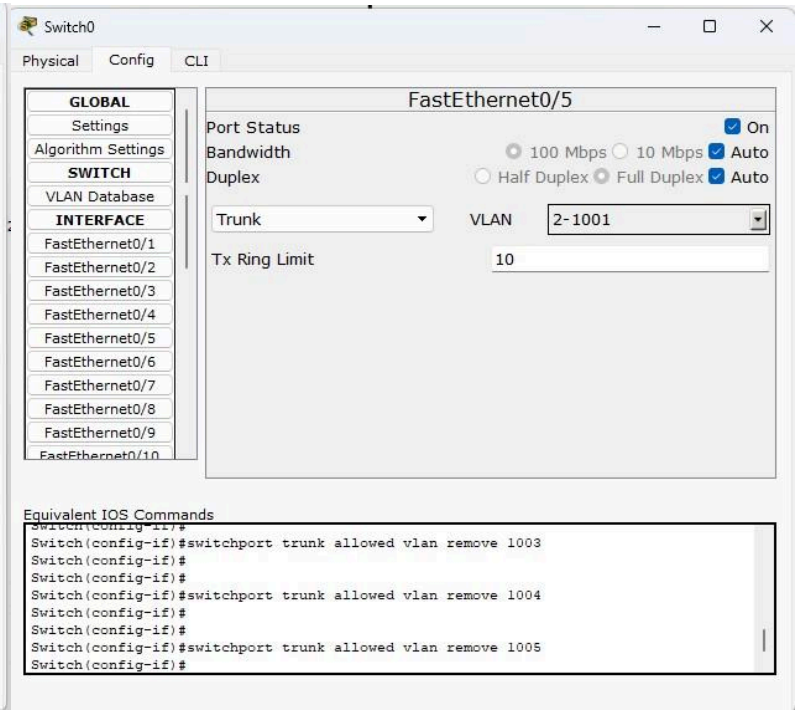
ROUTER0 CONFIGURATION:



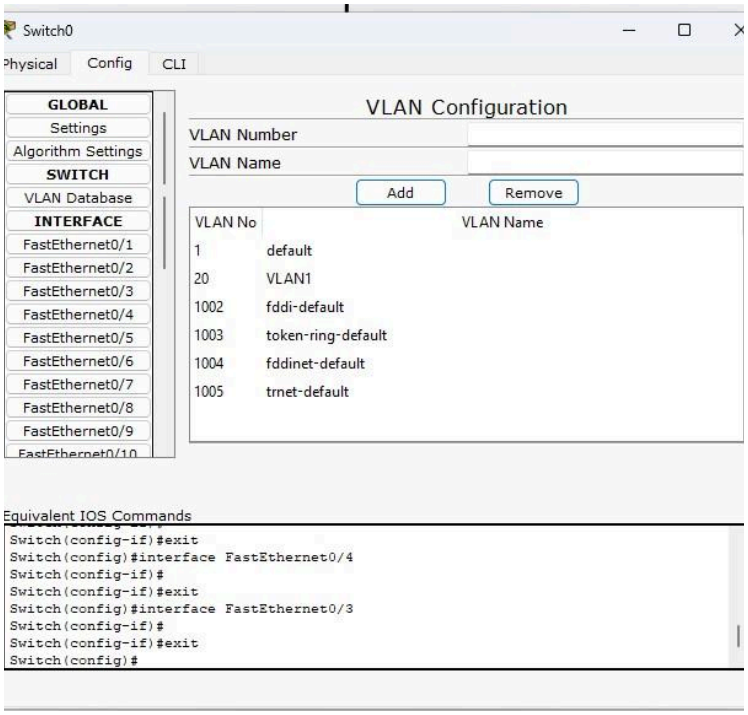
SWITCH Fa 0/3 configuration



SWITCH Fa 0/5 configuration

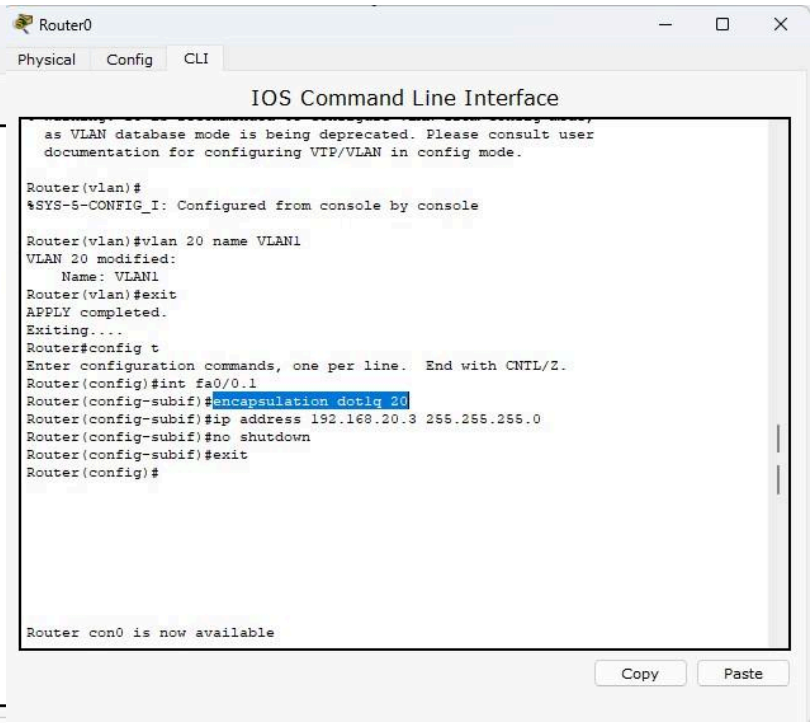


SWITCH VLAN Database

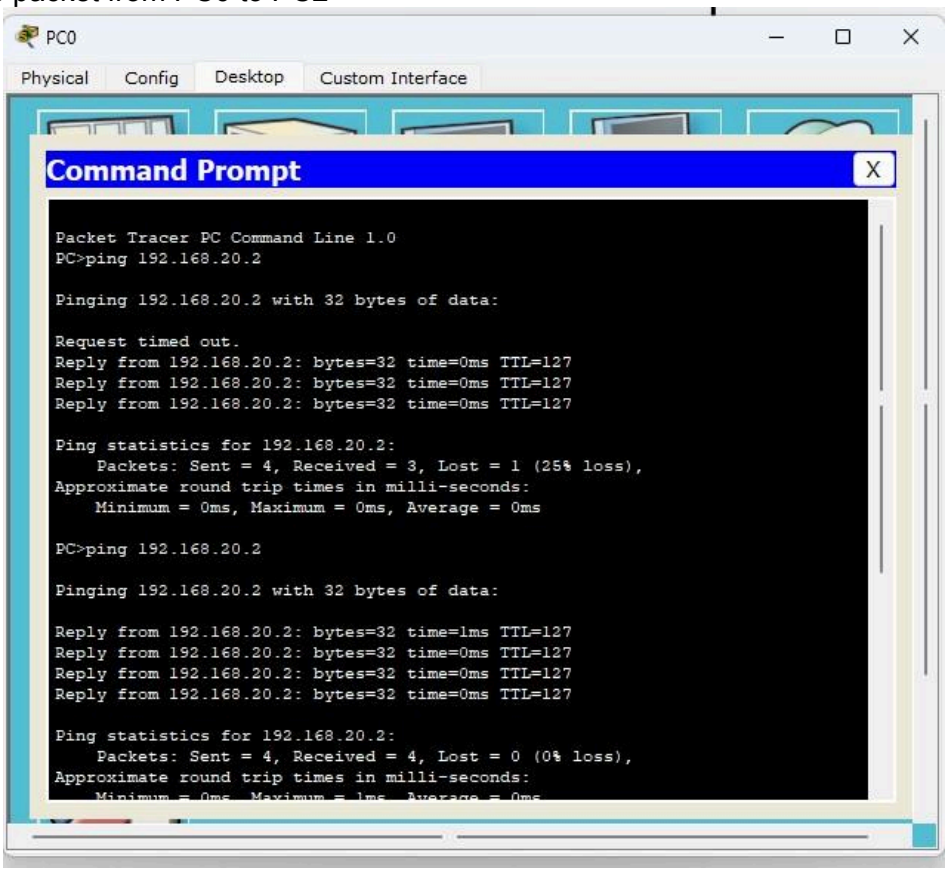


Creation of Fa 0/0.1

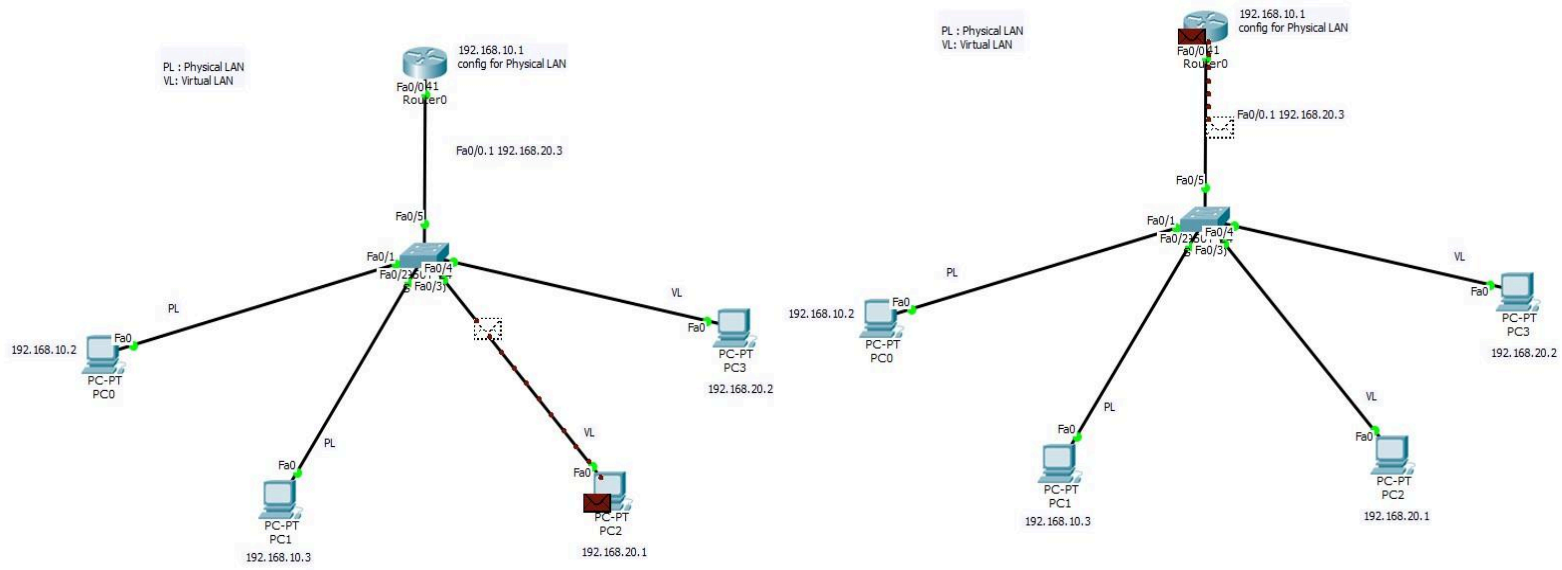
Encapsulating Fa 0/0.1



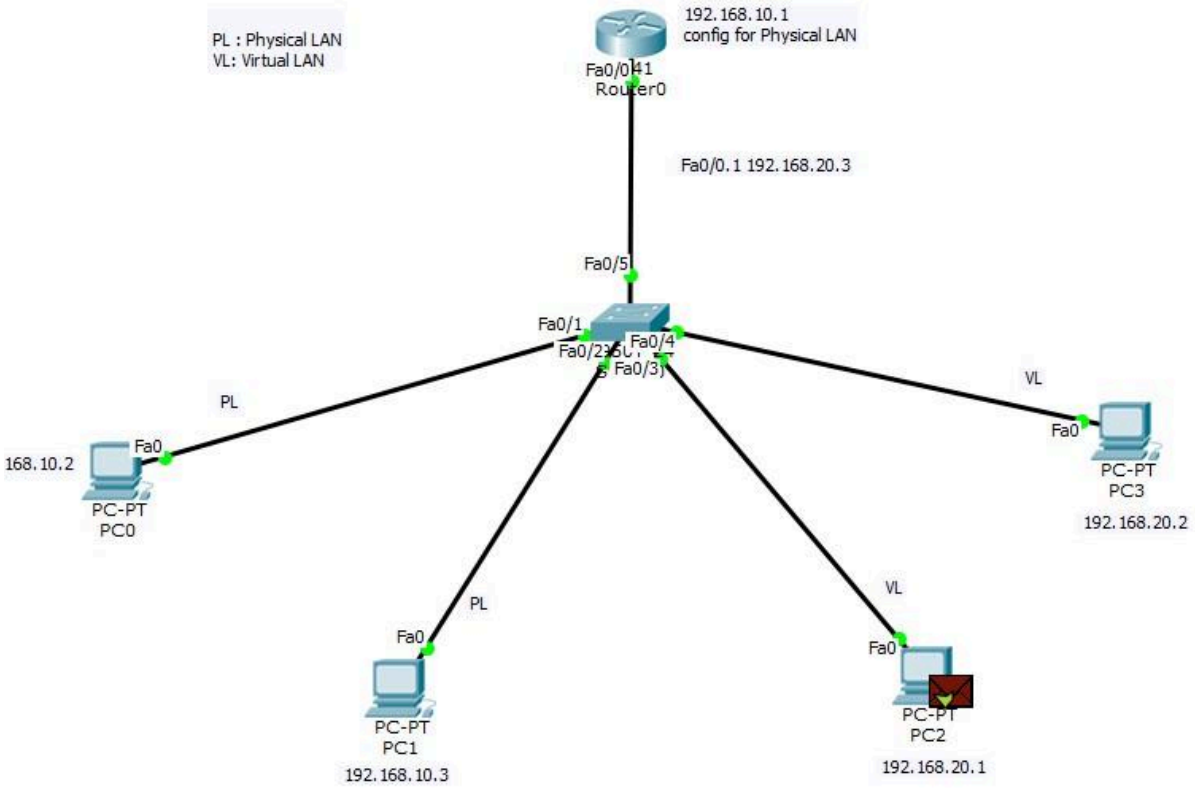
Ping message packet from PC0 to PC2



PDU



PDU Received



17/12/24

Lab 10.

Write a program for congestion control using Leaky bucket Algorithm.

* Leaky Bucket algorithm is a congestion control algorithm which uses Traffic Shaping mechanism to control the amount and rate of traffic sent to network.
* It helps to regulate rate of data transmission & reduces congestion.

* 2 types of traffic shaping algorithm
  1. Leaky Bucket 2. Token Bucket.

Algorithm for variable length packets

1. Initialize: a counter to n at tick of clock.
2. Repeat until n is smaller than packet size of packet at head of queue.
   a). pop of packet out of head of queue.
   b). Send packet P into network.
   c). Decrement counter by size of packet P.
3. Reset counter & go to step 1

Code:

```
#include <bits/stdc++.h>
using namespace std;
int main() {

int no_of_queries, storage=0, input_pkt_size, bucket_size,
output_pkt_size;
```

```
cout << "Enter the number of queries:";
cin >> no_of_queries;

cout << "Enter the bucket size:";
cin >> bucket_size;

cout << "Enter the input packet size:";
cin >> input_pkt_size;

cout << "Enter the output packet size:";
cin >> output_pkt_size;

for (int i=0; i<no_of_queries; i++) {

int size_left = bucket_size - storage;

if (input_pkt_size <= size_left) {
  storage += input_pkt_size;
} else {
  cout << "Packet loss =" << input_pkt_size << endl;
}

cout << "Buffer size" << storage << "out of bucket
  size" << bucket_size << endl;
storage -= output_pkt_size;
if (storage < 0) storage = 0;
}

return 0;
}
```

17/12/24

Output:

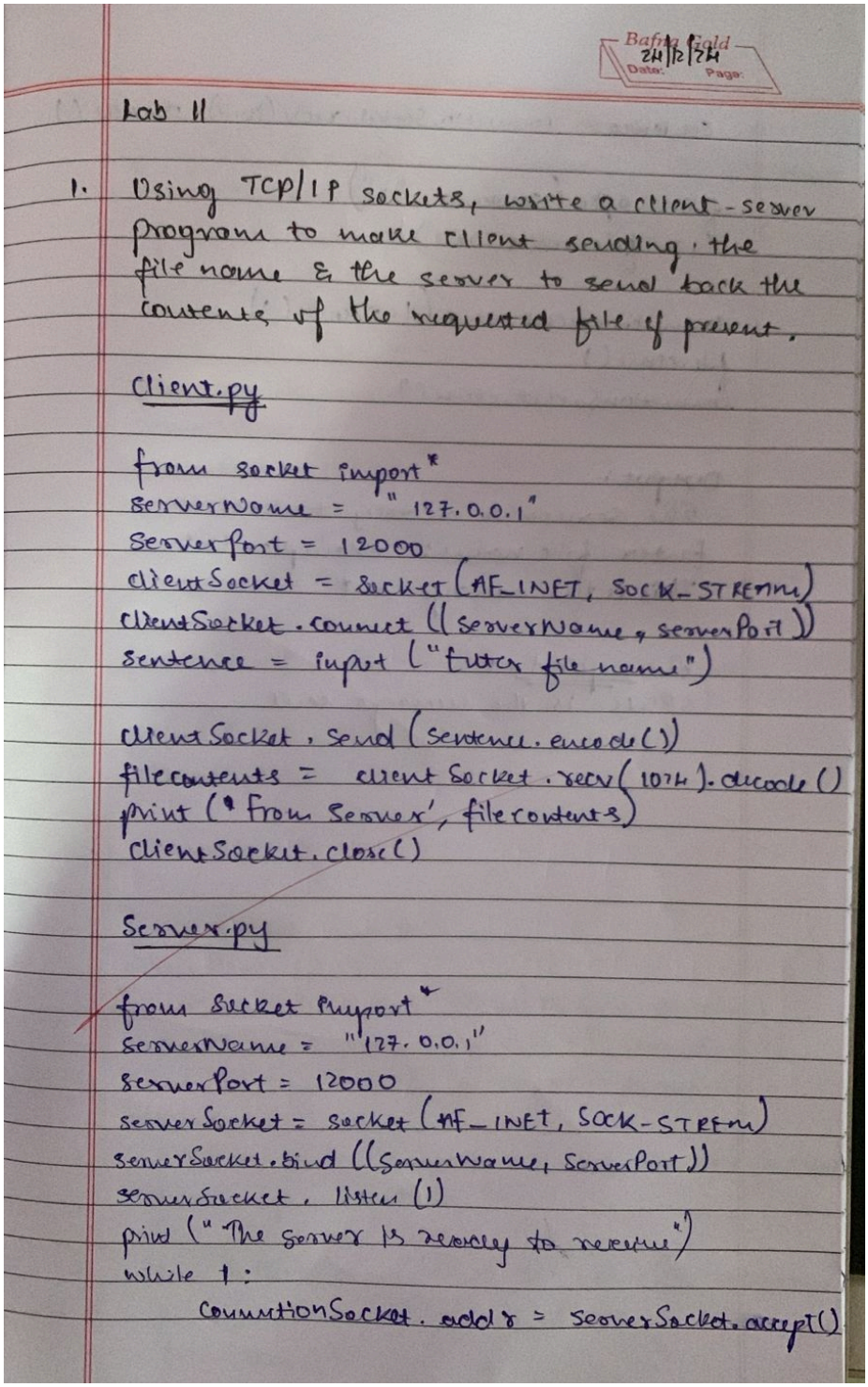
Enter number of queries: 4
Enter bucket size: 10
Enter input bucket size: 4
Enter output bucket size: 1

Buffer size = 4 out of bucket size = 10
Buffer size = 7 out of bucket size = 10
Buffer size = 10 out of bucket size = 10

Packet loss = 4
Buffer size = 9 out of bucket size = 10

CN Lab 11

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



Lab 11

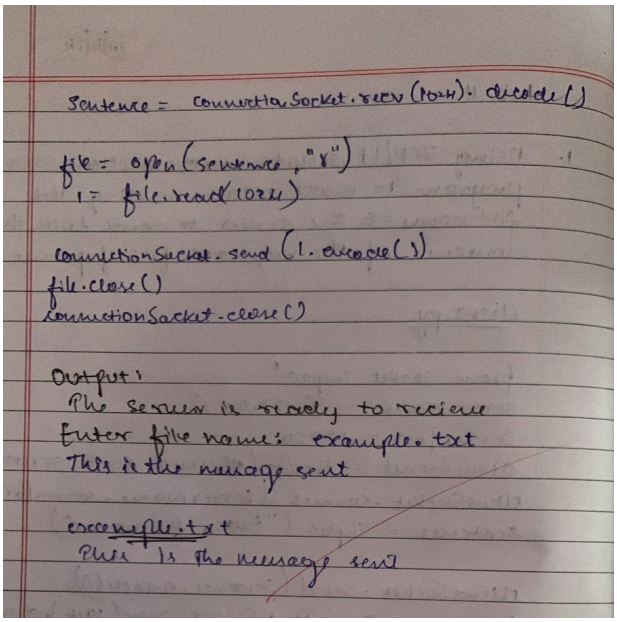
1. Using TCP/IP sockets, write a client-server program to make client sending the file name & the server to send back the contents of the requested file if present.

Client.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_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()
```

Server.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
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()
```



Code:

ClientTCP.py

```
from socket import *
serverName = &#39;127.0.0.1&#39;
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input(&quot;\nEnter file name: &quot;)
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print (&#39;\nFrom Server:\n&#39;)
print(filecontents)
clientSocket.close()
```

ServerTCP.py

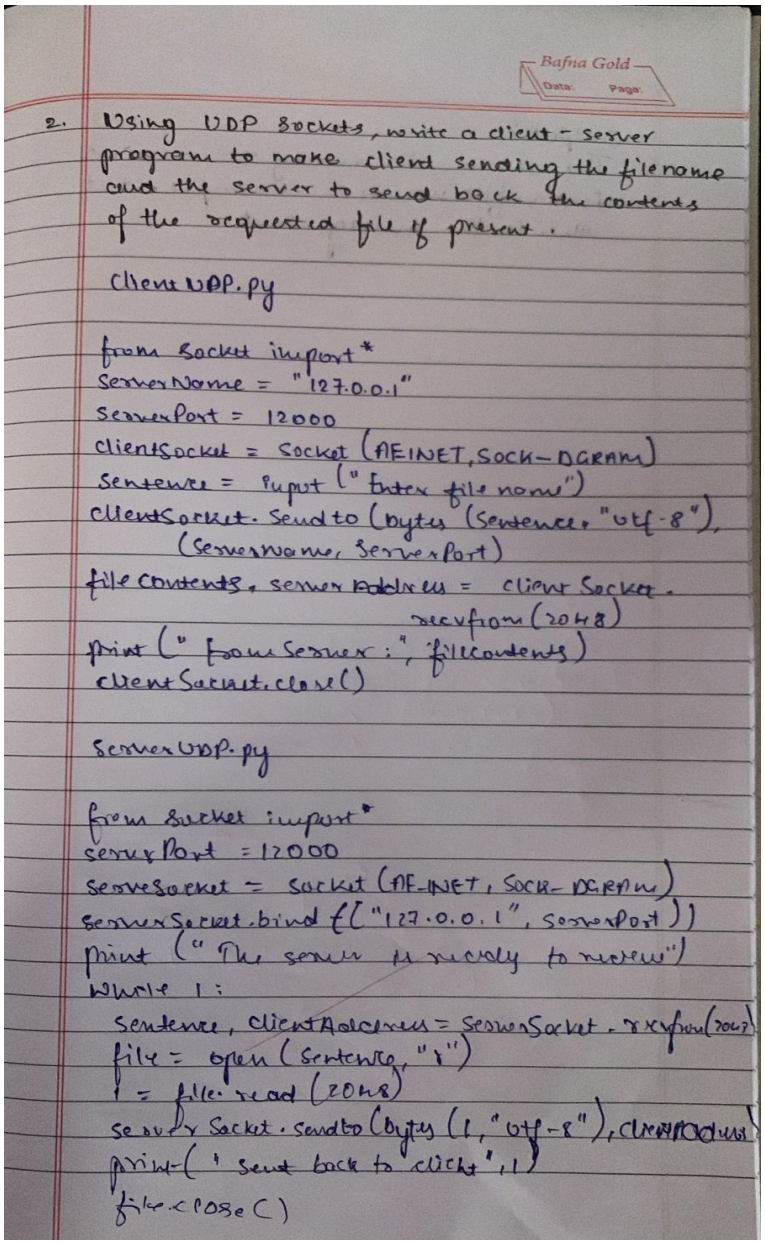
```
from socket import *
serverName=&quot;127.0.0.1&quot;
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
print (&quot;The server is ready to receive&quot;)
connectionSocket, addr = serverSocket.accept()
sentence = connectionSocket.recv(1024).decode()
file=open(sentence,&quot;r&quot;)
l=file.read(1024)
connectionSocket.send(l.encode())
print (&#39;\nSent contents of &#39; + sentence)
file.close()
connectionSocket.close()
```

Output

```python
import os
import subprocess
import time

# Folder and file details
folder_path = "/content/drive/MyDrive/CN_tcp_udp"  # Change this to the full path of your folder
requested_file = "example.txt"  # File to request from the server

# Verify the folder exists
if not os.path.isdir(folder_path):
    raise FileNotFoundError(f"Folder '{folder_path}' not found. Ensure the drive is mounted.")

# Change the working directory
os.chdir(folder_path)

# Function to run a script and capture output
def run_script(script, args=None):
    process = subprocess.Popen(
        ["python", script] + (args or []),
        stdout=subprocess.PIPE,
        stderr=subprocess.PIPE
    )
    stdout, stderr = process.communicate()
    return stdout.decode(), stderr.decode()

try:
    # Start the server
    print("Starting server.py...")
    server_process = subprocess.Popen(["python", "server.py"])

    # Allow the server to initialize
    time.sleep(3)

    # Run the client
    print("Starting client.py...")
    client_stdout, client_stderr = run_script("client.py", [requested_file])

    # Display client output
    print("\nClient Output:")
    print(client_stdout)
    if client_stderr:
        print("\nClient Errors:")
        print(client_stderr)

finally:
    # Terminate the server process
    server_process.terminate()
    print("\nServer process terminated.")
```

```
Starting server.py...
Starting client.py...

Client Output:
From Server: This is the file


Server process terminated.
```

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



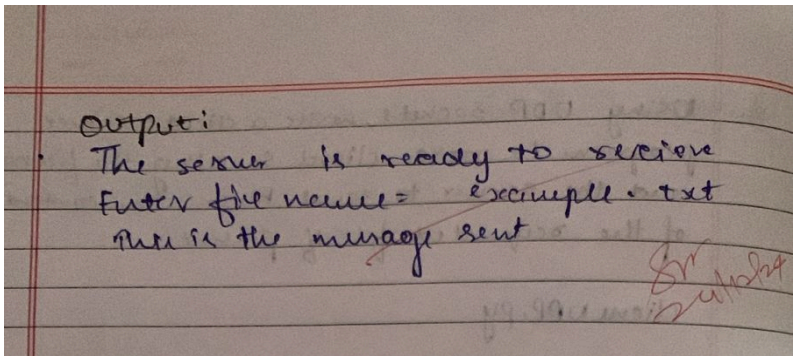
2. Using UDP sockets, write a client - server program to make client sending the filename and the server to send back the contents of the requested file if present.

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()
```

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()
```



Output:

```
The server is ready to recieve
Enter file name: example.txt
This is the message sent
```

Code:

ClientUDP.py

```
from socket import *
serverName = &amp;quot;127.0.0.1&amp;quot;
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input(&amp;quot;\nEnter file name: &amp;quot;)
clientSocket.sendto(bytes(sentence,&amp;quot;utf-8&amp;quot;),(serverName, serverPort))
filecontents,serverAddress = clientSocket.recvfrom(2048)
print (&amp;#39;\nReply from Server:\n&amp;#39;)
print (filecontents.decode(&amp;quot;utf-8&amp;quot;))
# for i in filecontents:
# print(str(i), end = &amp;#39;&amp;#39;)
clientSocket.close()
clientSocket.close()
```

ServerUDP.py

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind((&amp;quot;127.0.0.1&amp;quot;, serverPort))
print (&amp;quot;The server is ready to receive&amp;quot;)
while 1:
sentence, clientAddress = serverSocket.recvfrom(2048)
sentence = sentence.decode(&amp;quot;utf-8&amp;quot;)
file=open(sentence,&amp;quot;r&amp;quot;)
con=file.read(2048)
serverSocket.sendto(bytes(con,&amp;quot;utf-8&amp;quot;),clientAddress)
print (&amp;#39;\nSent contents of &amp;#39;, end = &amp;#39; &amp;#39;)
print (sentence)
# for i in sentence:
# print (str(i), end = &amp;#39;&amp;#39;)
file.close()
```

Output

```python
import socket
import threading

# Server Code
def start_server():
    serverPort = 12000
    serverSocket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    serverSocket.bind(("127.0.0.1", serverPort))
    print("The server is ready to receive")

    while True:
        sentence, clientAddress = serverSocket.recvfrom(2048)
        filename = sentence.decode("utf-8")

        try:
            # Try to open the file and send its contents
            with open(filename, "r") as file:
                file_contents = file.read(2048)
            serverSocket.sendto(file_contents.encode("utf-8"), clientAddress)
            print(f"Sent back to client: {file_contents}")
        except FileNotFoundError:
            # If file not found, notify the client
            serverSocket.sendto("File not found.".encode("utf-8"), clientAddress)
            print(f"File '{filename}' not found.")
        except Exception as e:
            serverSocket.sendto(f"Error: {str(e)}".encode("utf-8"), clientAddress)
            print(f"Error processing file '{filename}': {str(e)}")

# Client Code
def start_client(filename):
    serverName = "127.0.0.1"
    serverPort = 12000
    clientSocket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

    # Send the file name to the server
    clientSocket.sendto(filename.encode("utf-8"), (serverName, serverPort))

    # Receive the file contents or error message from the server
    filecontents, serverAddress = clientSocket.recvfrom(2048)
    print(f"From Server: {filecontents.decode('utf-8')}")
    clientSocket.close()

# Run the server in a separate thread
server_thread = threading.Thread(target=start_server)
server_thread.daemon = True  # This makes the server thread stop when the main program exits
server_thread.start()

# Simulate the client-side file request
requested_file = "example.txt"  # Ensure this file exists in your working directory for testing
start_client(requested_file)
```

```
The server is ready to receive
From Server: This is the file
Sent back to client: This is the file
```