

# **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 P Asst. Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE
--	--

## Index

<b>Sl. No.</b>	<b>Date</b>	<b>Experiment Title</b>	<b>Page No.</b>
1	1-10-24	Transmission of a single PDU between two devices connected using a hub and a switch	4
2	10-10-24	Configuration of IP address of the router and explore ping command	8
3	15-10-24	Configuration of default routes to the router	11
4	22-10-24	Configuration using default and static route	14
5	29-10-24	Operation of TELNET by accessing the router placed in the server room from a PC in IT office	18
6	12-11-24	IP address configuration of host using DHCP server present within same LAN	24
7	19-11-24	Configure RIP routing protocol in routers	30
8	26-11-24	Communication between two devices using a wireless LAN Working of ARP for communication within a LAN	35
9	3-12-24	Create a virtual LAN on top on physical LAN and enable communication between them	43
10	19-12-24	Program for congestion control using Leaky Bucket Algorithm	49
11	24-12-24	Client Server Programming using TCP and UDP Sockets	52

Github Link:

[https://github.com/mohithjain/CNLab\\_CS162](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**      **Bafna Gold**  
Date: 10 Page: 1

**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**

IP address 10.0.0.1  
IP address 10.0.0.2  
IP address 10.0.0.3

**Steps to configure**

- \* Click on the PC-PT PC0 device, select config choose Fast Ethernet 0/1 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

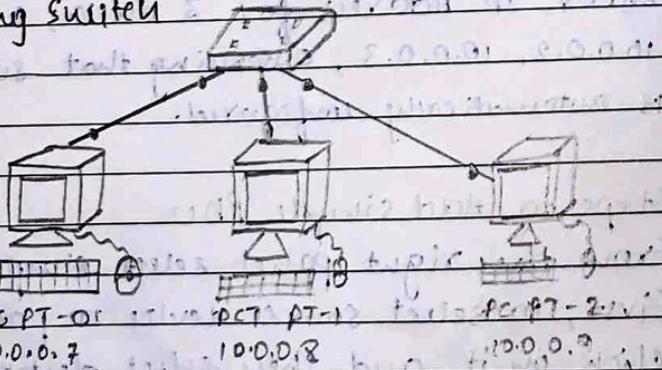
11/10/24

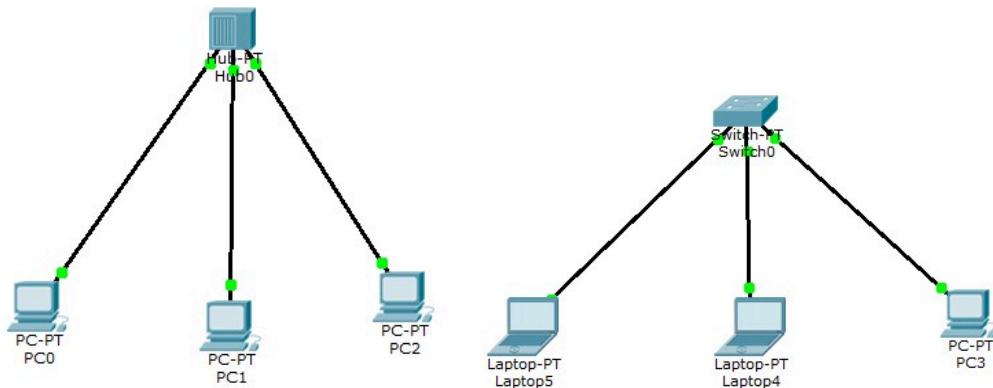
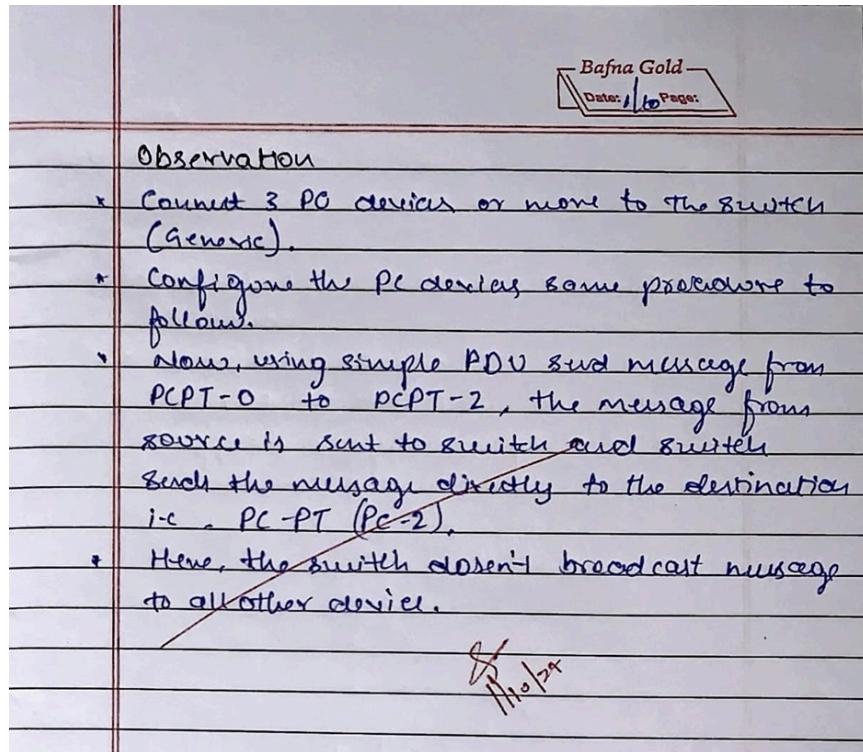
- \* Now, the PDU is successfully set.
- \* To check the simulation of PDU, switch to simulation mode from the panel.
- \* From play control in simulation mode, click on Add Capture / Play

#### 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 reject it.
- \* Now, the destination device 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 others.

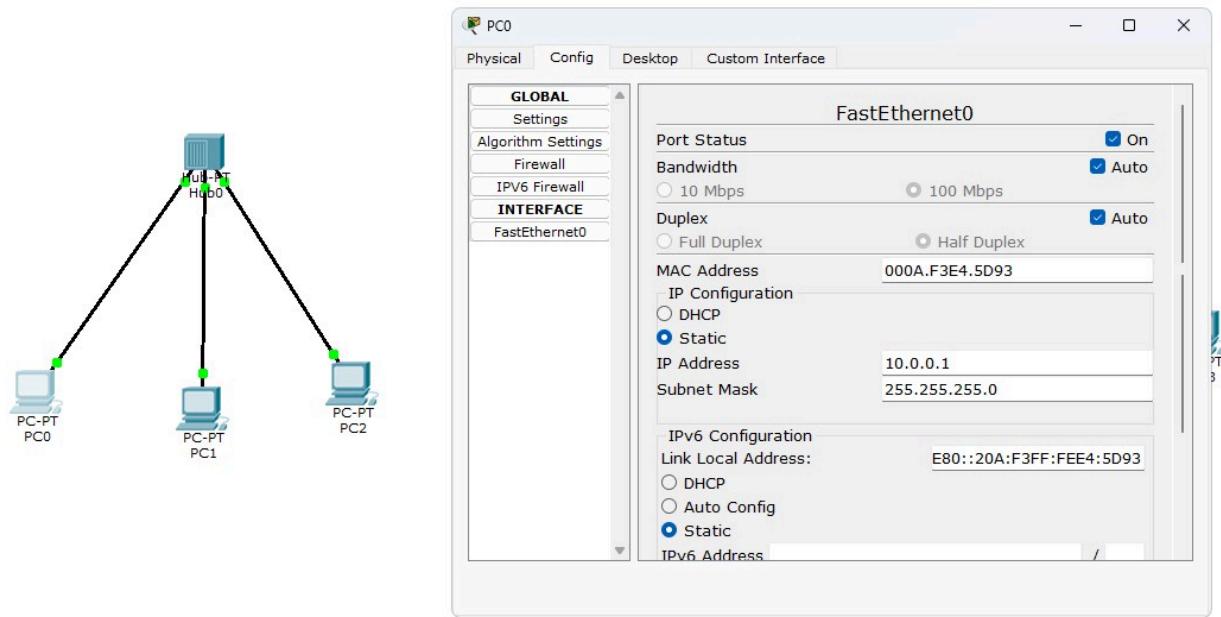
#### Using Switch



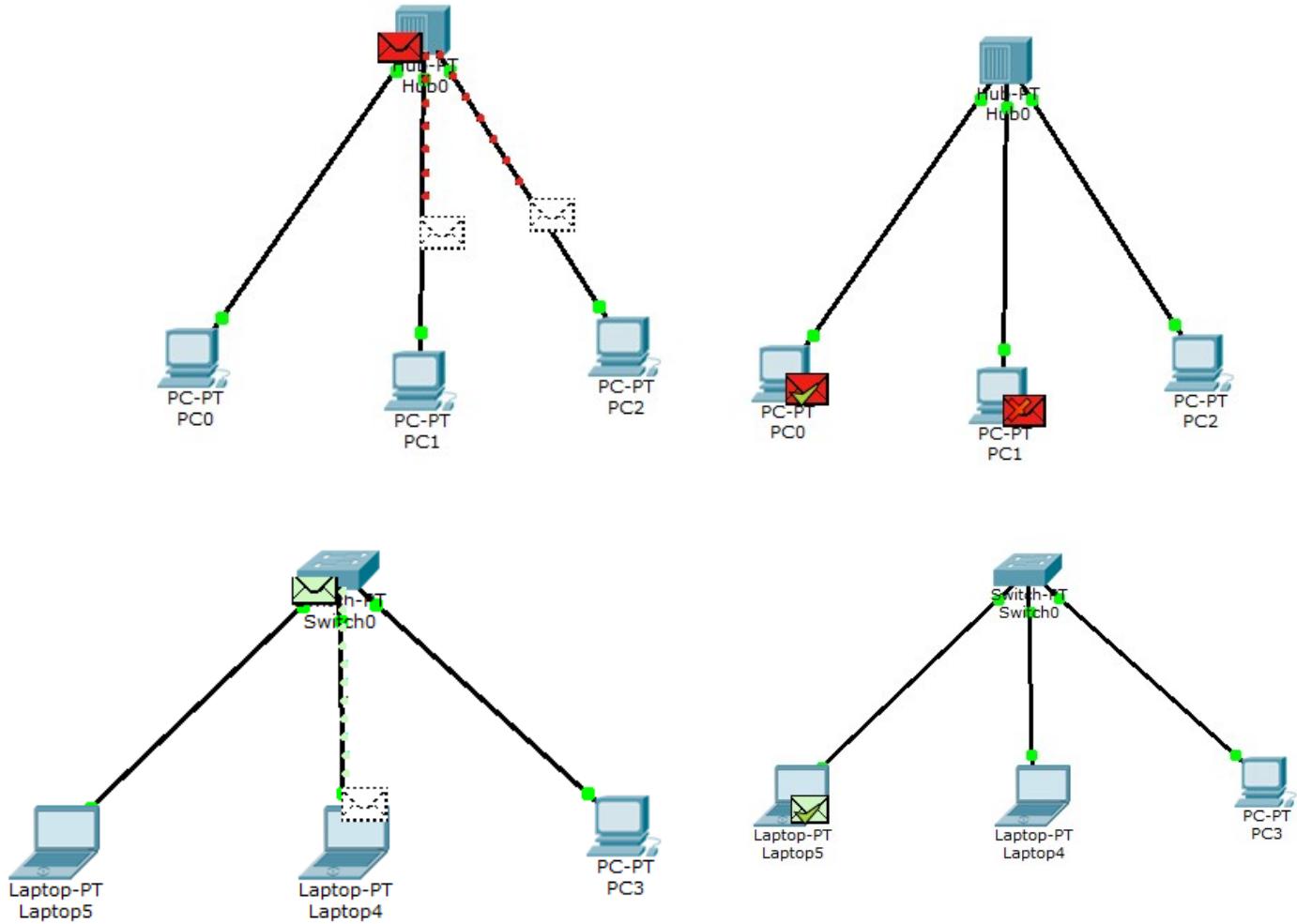


Hub (Star topology)

Switch (Star topology)



Configuration



## Computer Networks Lab 2

### Router Configuration and ping

#### Observation

**Lab - 02**

**Aim:** To demonstrate configuration of IP Address to the router & explore ping command.

**Steps:** Command line interface (CLI)

- ① enable
- ② #
- ③ config-t
- Click on interface of Router Fa0/0 or Fa0/1
- ④ interface fastethernet0/0
- ⑤ ip address 10.0.0.2 255.0.0.0
- ⑥ no shutdown

Now, Fastethernet0/0 changed state to up  
The line turns green

Similarly configure all devices

- \* Now, to configure Gateway to send message from source to destination
- \* Select the end device to config and run gateway enter the IP addresses of the router Fa0/0.
- \* Similarly, follow same for all end devices.

Now, the message is sent using ping

- \* Select end device → desktop → command prompt
- Enter
- ping destination-ipaddr (end device)
- ping 200.0.1

**Toplogy**

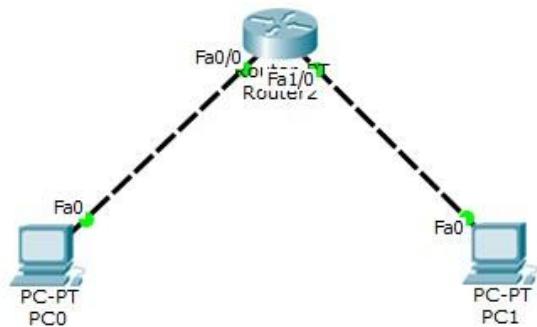
Interface	IP Address	Status
Fastethernet0/0	10.0.0.2	Up
Fastethernet1/0	20.0.0.3	Up

Device	IP Address	Gateway
PC-PT	10.0.0.1	10.0.0.2
PC0	10.0.0.2	10.0.0.1
PC-PT	20.0.0.1	20.0.0.3
PC1	20.0.0.3	20.0.0.1

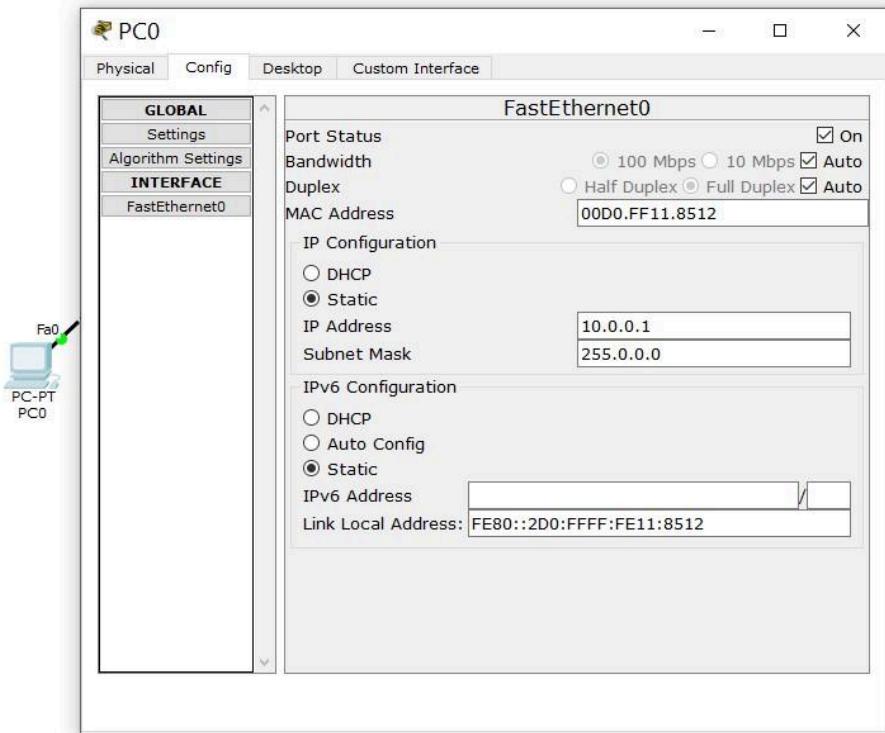
**Observation**

- \* Router is connected to two different networks with IP addresses 10.0.0.1 & 20.0.0.1.
- \* The two different networks are connected with commercial PC devices.
- \* The gateway is set to each end device; the router address is the gateway.
- \* When the message is ping and should receive to PC1 from PC0.
- \* The packets are received successfully to the PC1 through the router.

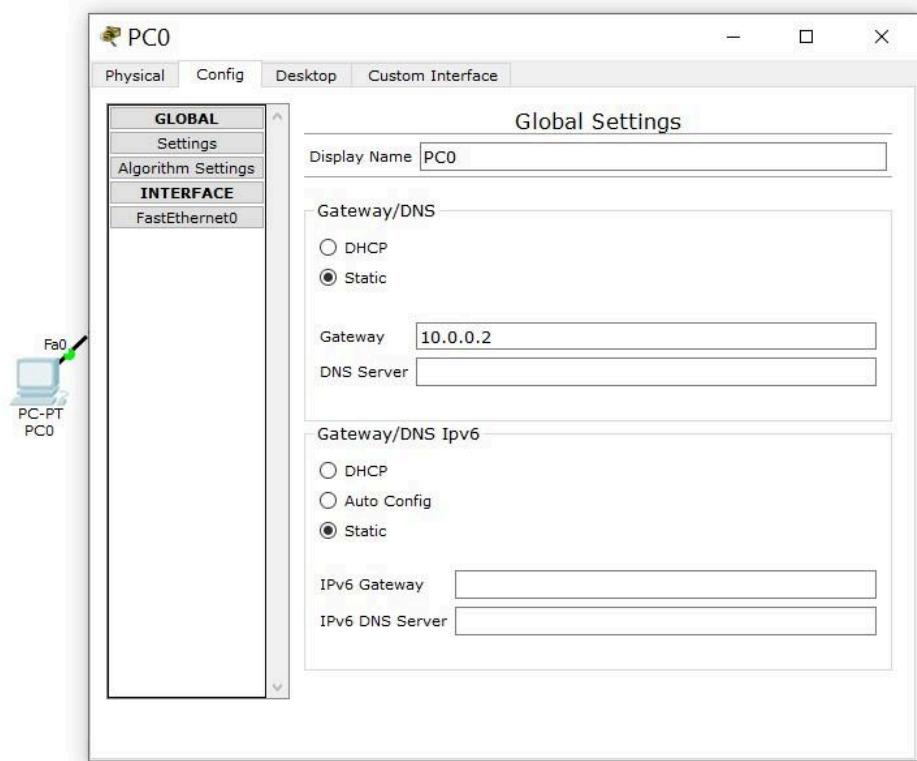
#### 1. Network Topology



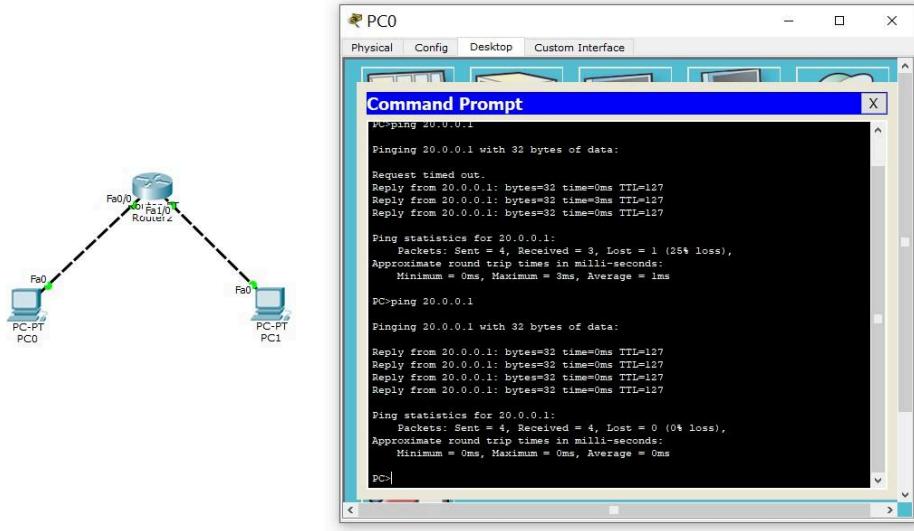
## 2. PC0 Configuration



## 3. PC0 Gateway



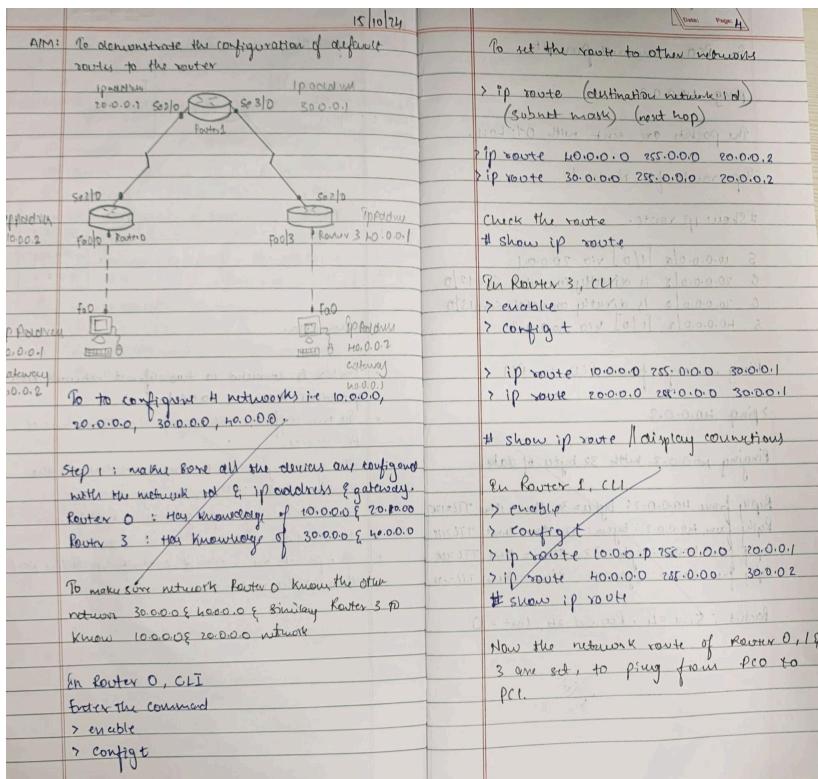
#### 4. PC0 Ping PC1



# Computer Networks Lab 3

## Configuration of default routes to Router and ping

### Observation



In PC0, command prompt  
 ping 10.0.0.2 (host) (loopback)  
 The packets are sent with 0% loss.  
 Up route for Router 1 (loopback)  
 # Show ip route

```

    S 10.0.0.0/8 [1/0] via 20.0.0.1
    C 20.0.0.0/8 is directly connected, Serial1/0
    C 30.0.0.0/8 is directly connected, Serial3/0
    S 40.0.0.0/8 [1/0] via 30.0.0.2
    
```

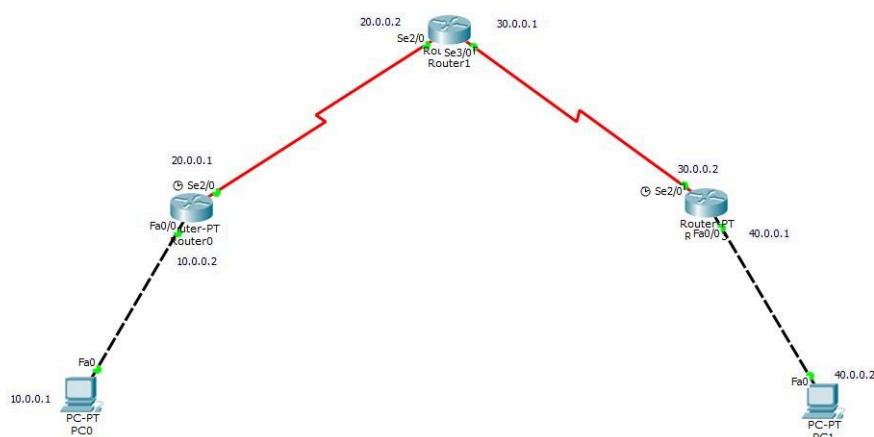
Ping message from PC0 to PC1  
 > ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data

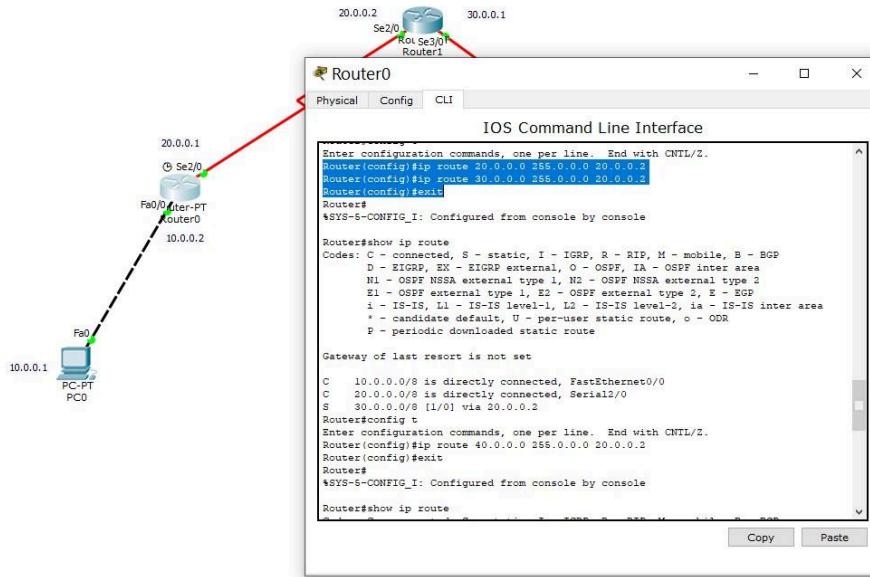
Reply from 10.0.0.2: bytes=32 time=1ms TTL=128  
 Reply from 10.0.0.2: bytes=32 time=1ms TTL=128  
 Reply from 10.0.0.2: bytes=32 time=1ms TTL=128  
 Reply from 10.0.0.2: bytes=32 time=1ms TTL=128

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

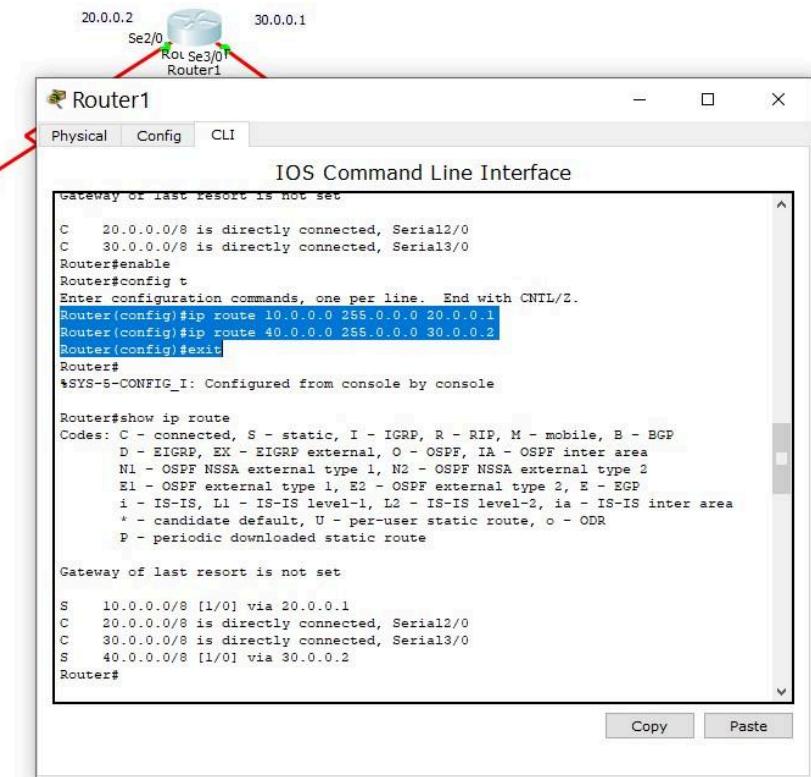
### 1. 2. Topology



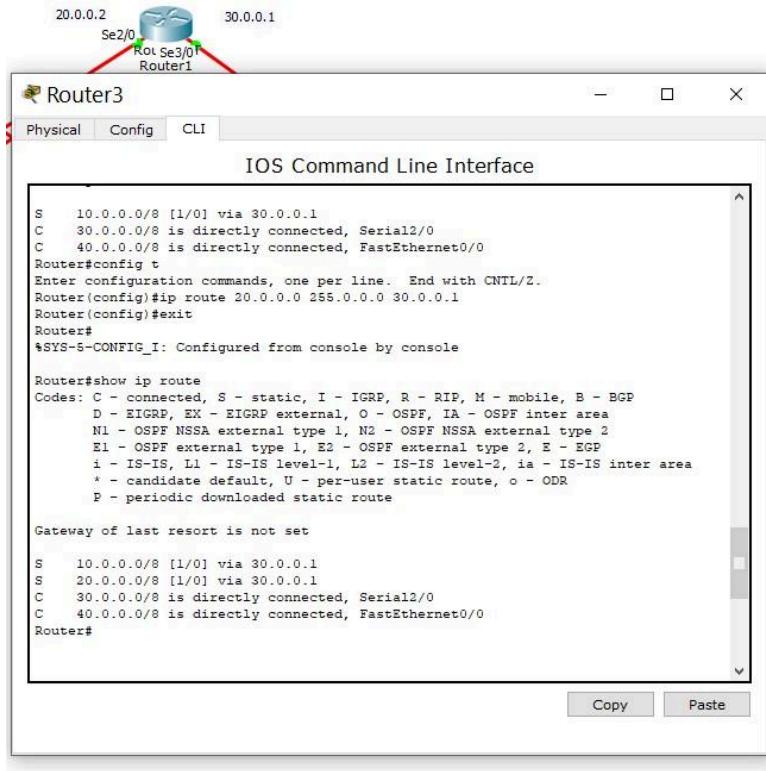
### 3. Router 0 ip routing



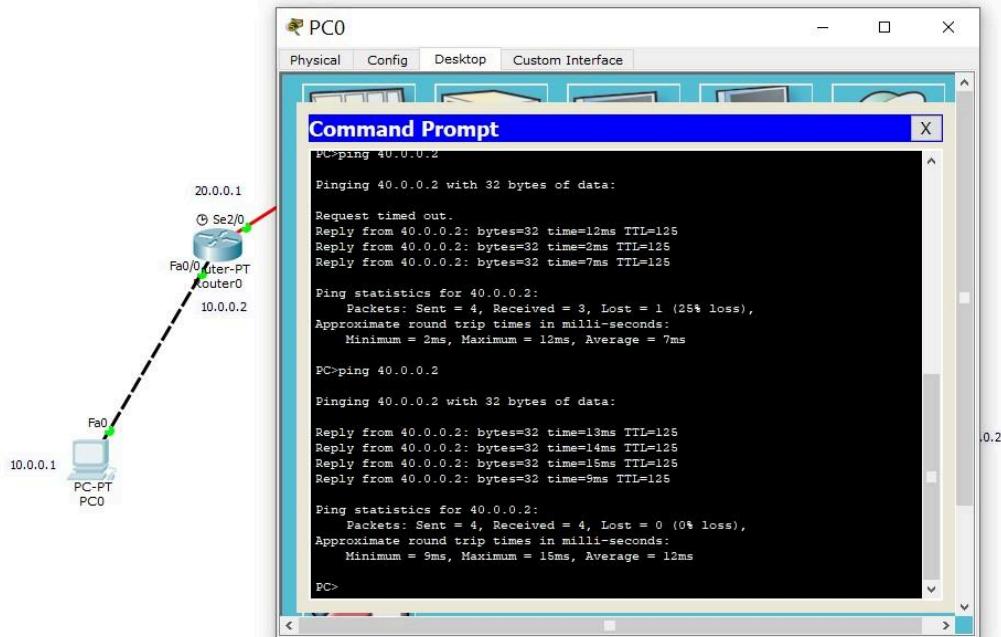
### 4. Router 1 ip routing



#### 4. Router 3 ip routing



#### 5. Ping to different network device

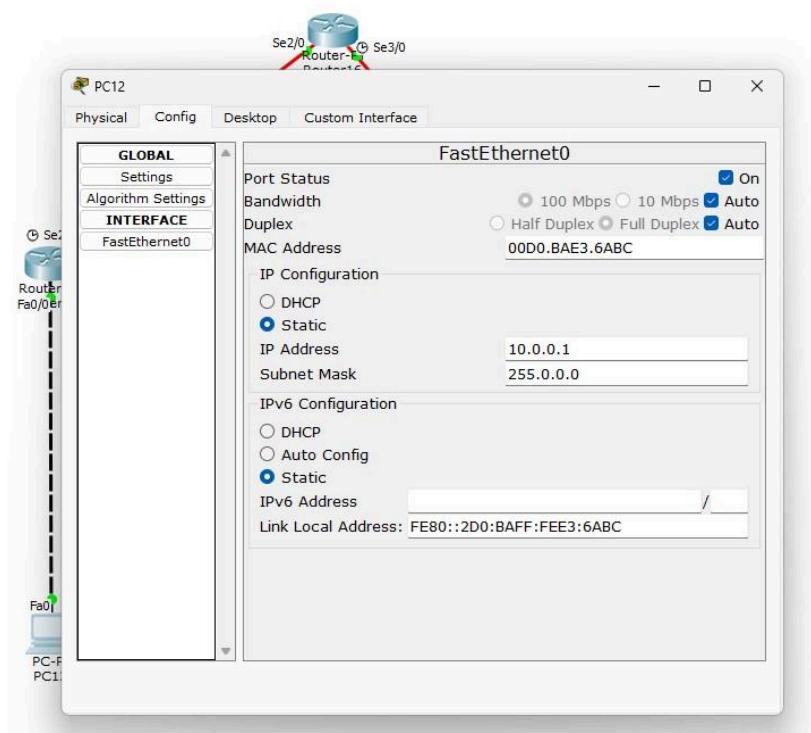
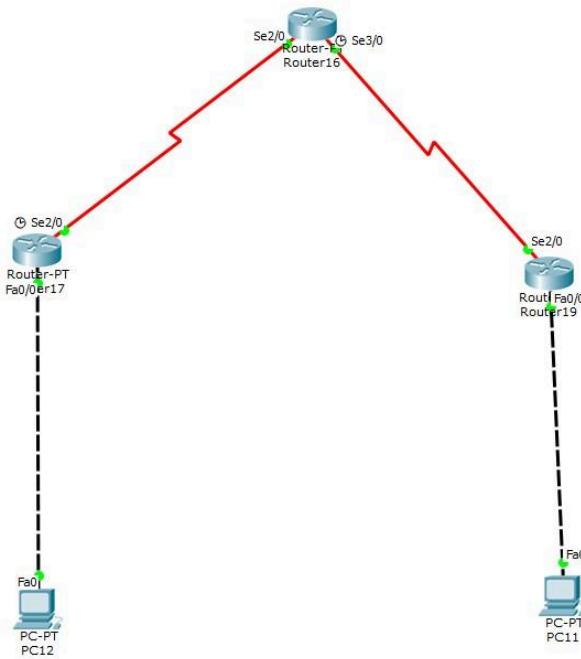


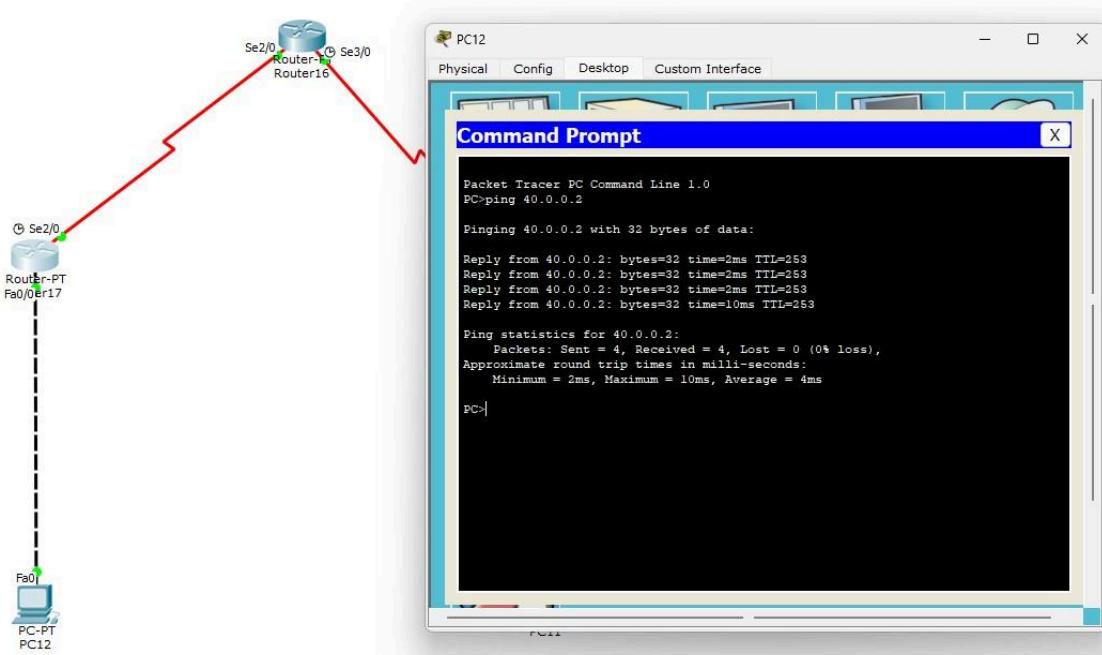
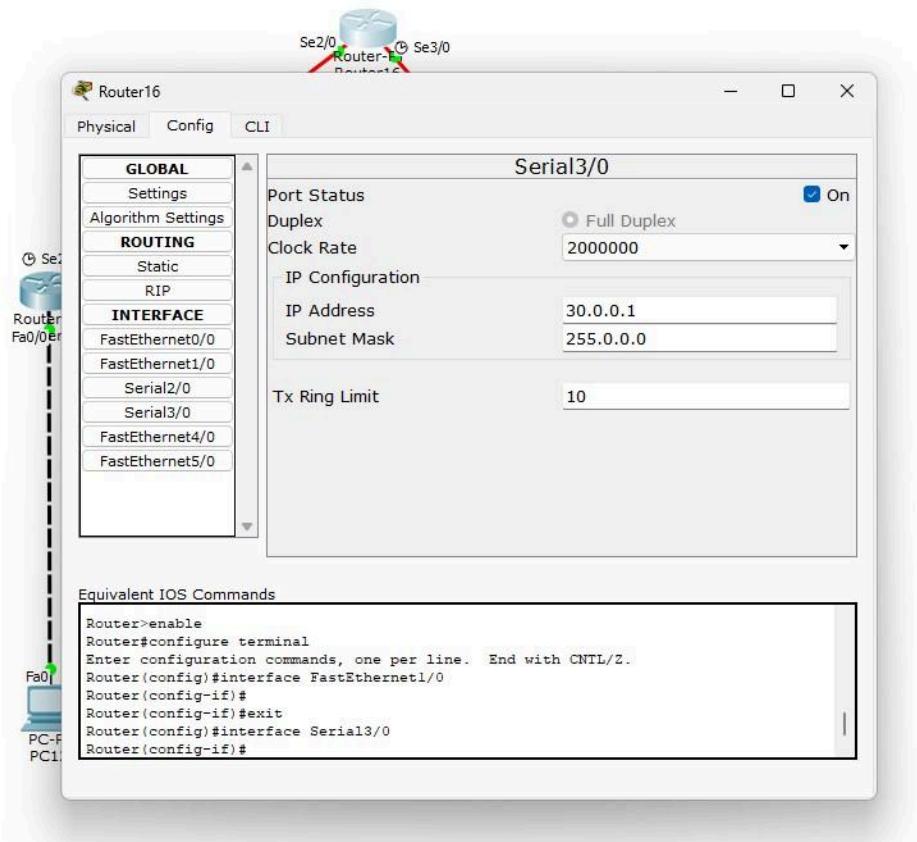
# CN LAB OBSERVATION

Mohith Jain

1BM22CS162

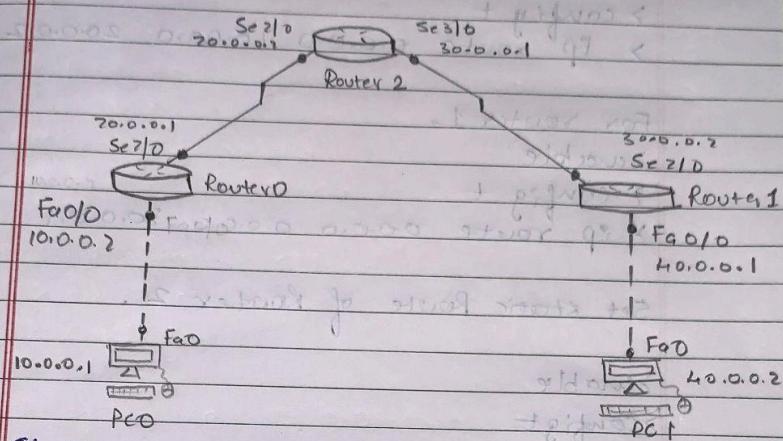
AIM : Configuration using default & static route





Lab - 04 at of study Network WPA2

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



**Steps:**

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

- \* Set ip addresses 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
- \* Similarly configure routes 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 10.0.0.0 255.0.0.0 30.0.0.2

destination subnet configuration  
network address mask

> 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 10.0.0.2

pinging 10.0.0.2 with 32 bytes of data?

Reply from 10.0.0.2: bytes=32 time=18ms TTL=2123

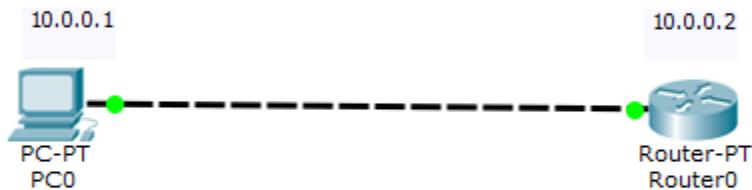
Reply from 10.0.0.2: bytes=32 time=4ms TTL=2123

Packet sent = 4, received = 4 host = 0 (0% loss)

## 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)#enable 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 pl
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:

The screenshot shows a 'Command Prompt' window with the title bar 'Command Prompt'. The window displays the following text:

```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

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

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

PC>telnet 10.0.0.2
Trying 10.0.0.2 ...Open

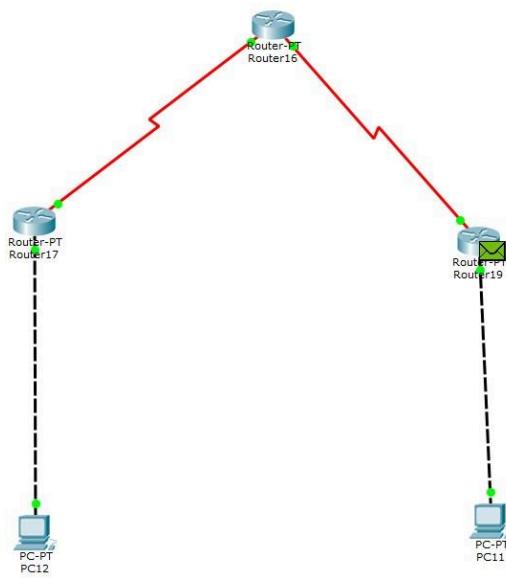
User Access Verification

Password:
R1>enable
Password:
R1#

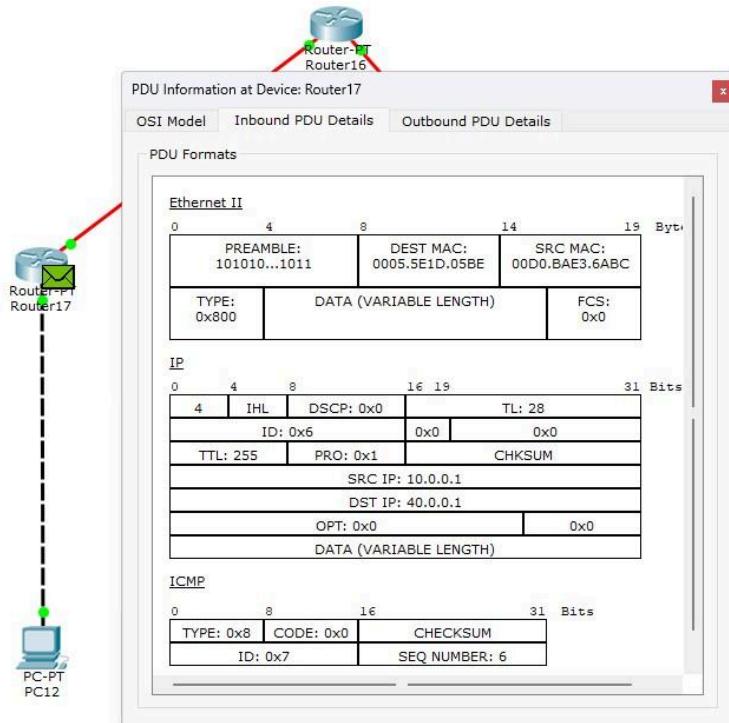
[Connection to 10.0.0.2 closed by foreign host]
PC>
```

## b) TTL CONCEPT

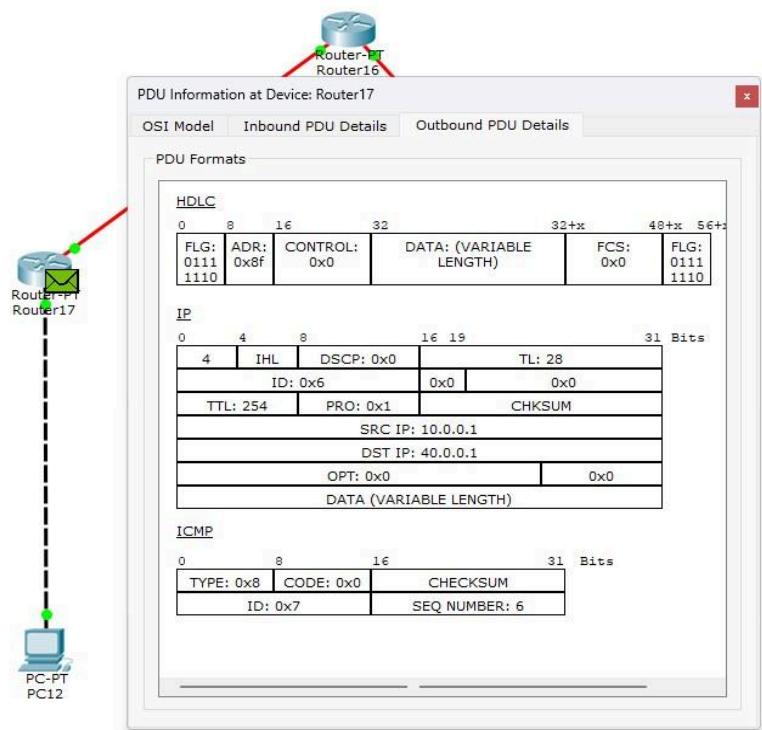
### TOPOLOGY:



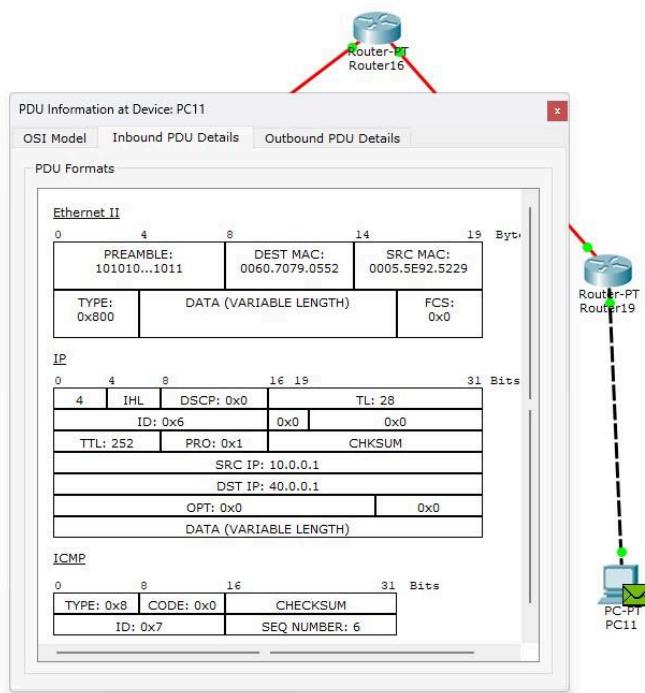
### INBOUND FOR ROUTER1:



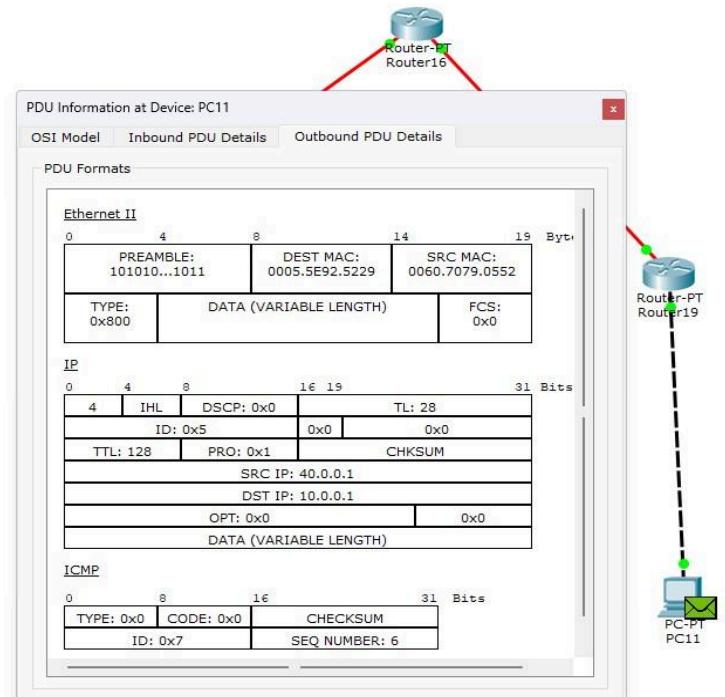
### OUTBOUND FOR ROUTER1:



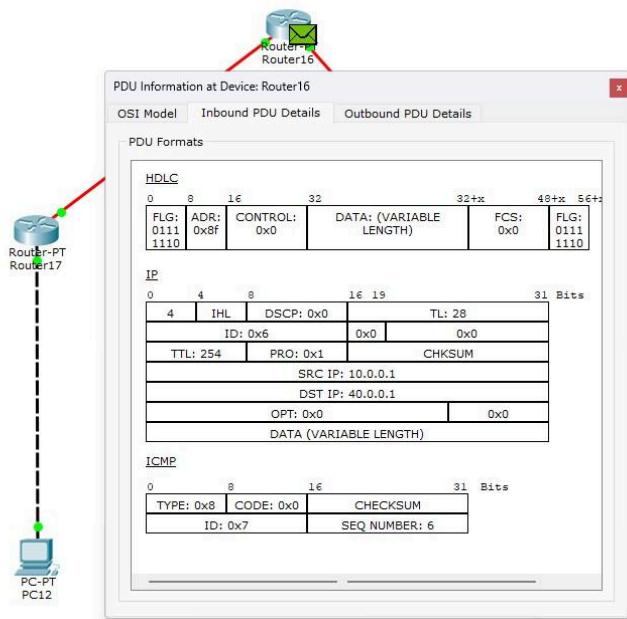
## INBOUND FOR PC1:



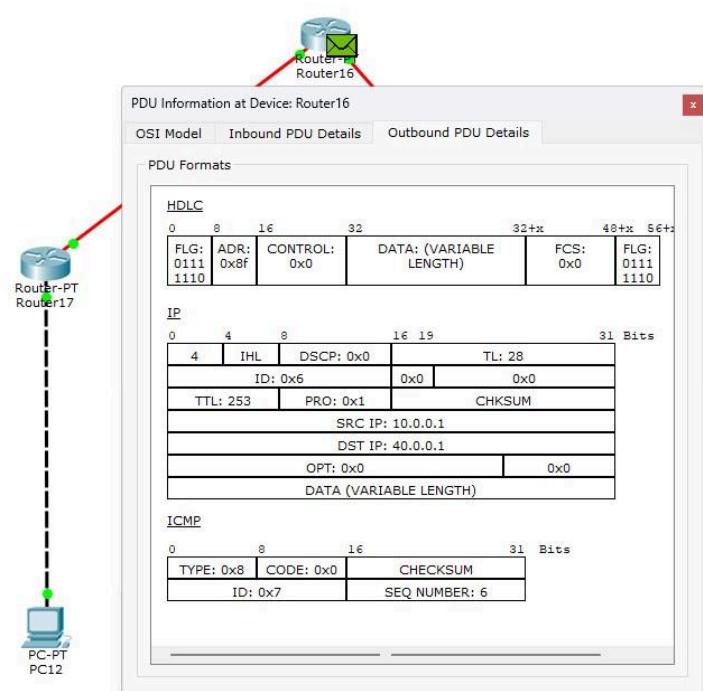
## OUTBOUND FOR PC1:



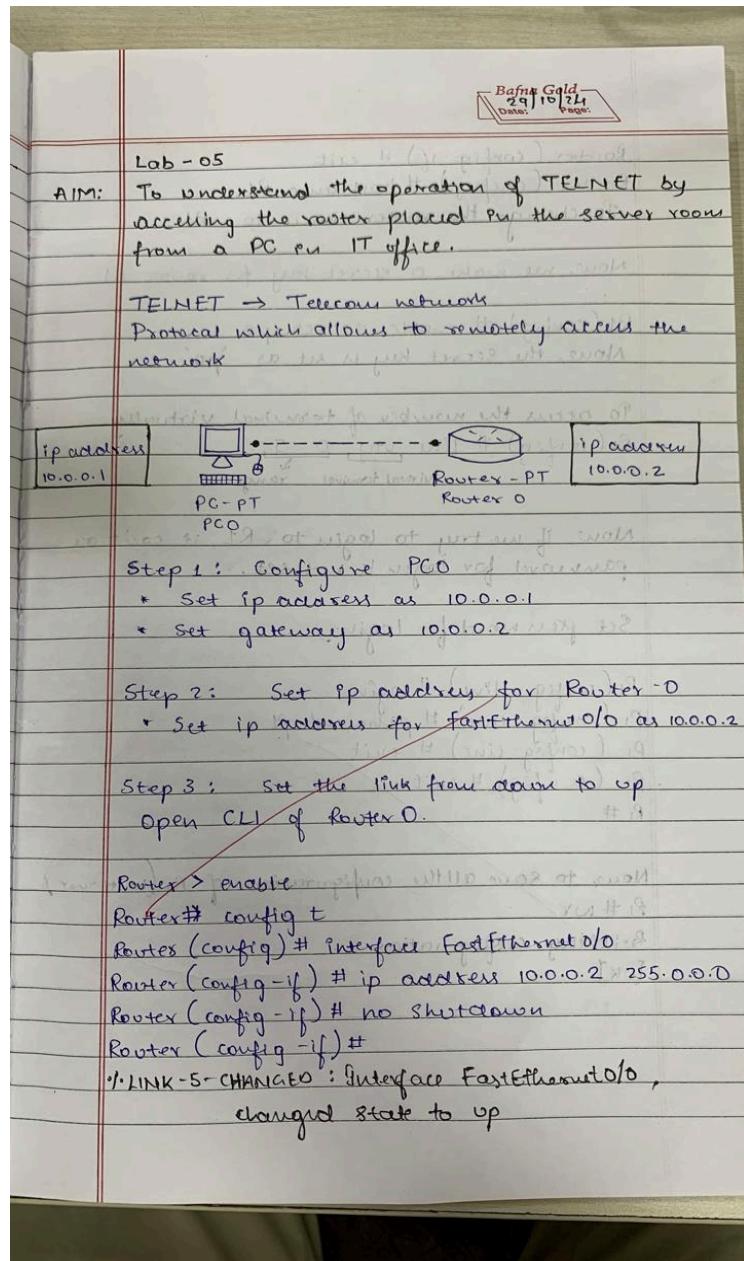
## OUTBOUND FOR ROUTER1:



## INBOUND FOR ROUTER1:



## OBSERVATION: ACCESSING ROUTER USING TELNET:



**Router (config-if) # exit** 20 - do!

**Router (config) # hostname R1** 20 - do!

To change the router name to 'R1' 20 - do!

Now, we make a secret key for router R1

**R1 (config) # enable secret po** 20 - do!

Now, the secret key is set as 'po'

To access the number of terminal virtually

**R1 (config) # line vty 0 5** 20 - do!

Virtual terminal range

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

Set password for login.

**R1 (config-line) # login** 20 - do!

**R1 (config-line) # password po** 20 - do!

**R1 (config-line) # exit** 20 - do!

**R1 (config) # exit** 20 - do!

**R1 #** 20 - do!

Now, to save all the configurations of R1 (write-er)

**R1 # wr** 20 - do!

Building configuration... [OK]

Configuration saved [OK]

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: po // Secret key

R1#

Observation:

Using TELNET, PC0 can access router R1

8/10/2\*

## CN LAB 6

Name : Mohith Jain

USN: 1BM22CS162

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

OBSERVATION:

Lab -06  
12/11/24

Bafra Gold  
Date: \_\_\_\_\_  
Page: \_\_\_\_\_

**Aim :**

- ① To configure ip addresses of the Host using DHCP server present within the same LAN
- ② To configure IP addresses of the Host using DHCP server present outside the same LAN

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

**Ring 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-D to be 10.0.0.2
- Now, select server-D, select services & click on DHCP, set service to be ON.
- Set poolName as ServerPool1
- Default gateway to Server as 10.0.0.2 & DNS as 10.0.0.1 (temporarily) & SAVE.

**Now DHCP server is ready & have been assigned to server 0**

- Now, most 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**

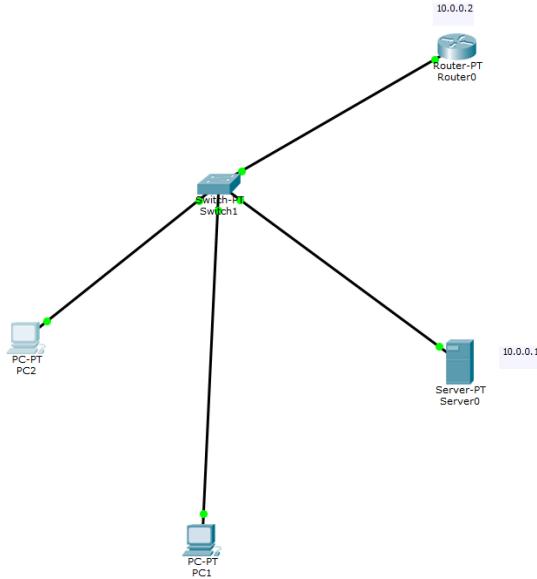
```

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

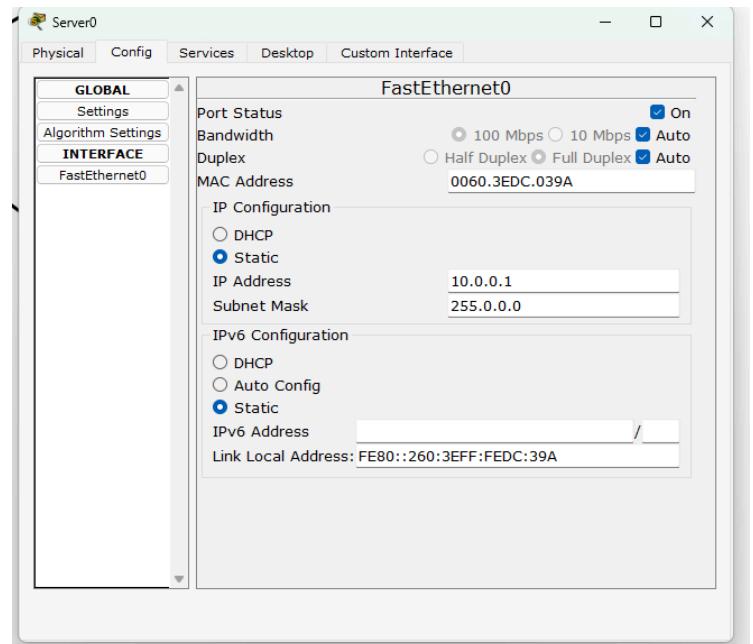
```

**B)** Outside the LAN

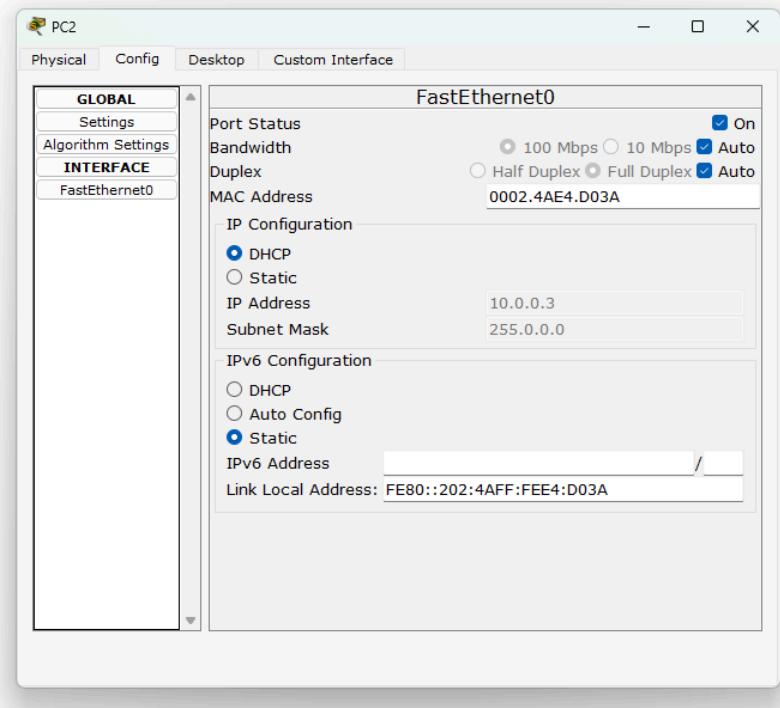
## TOPOLOGY



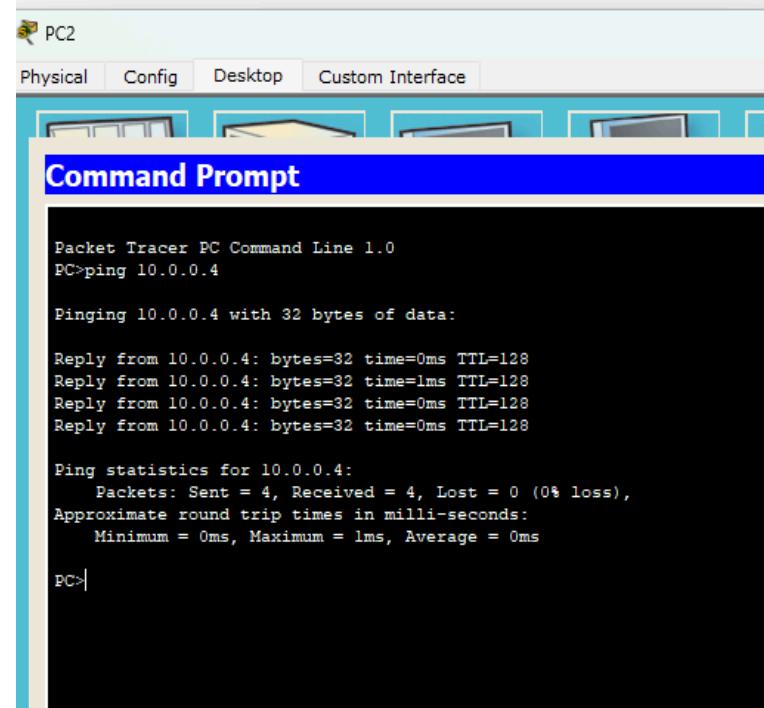
## SERVER CONFIG



## PC CONFIG



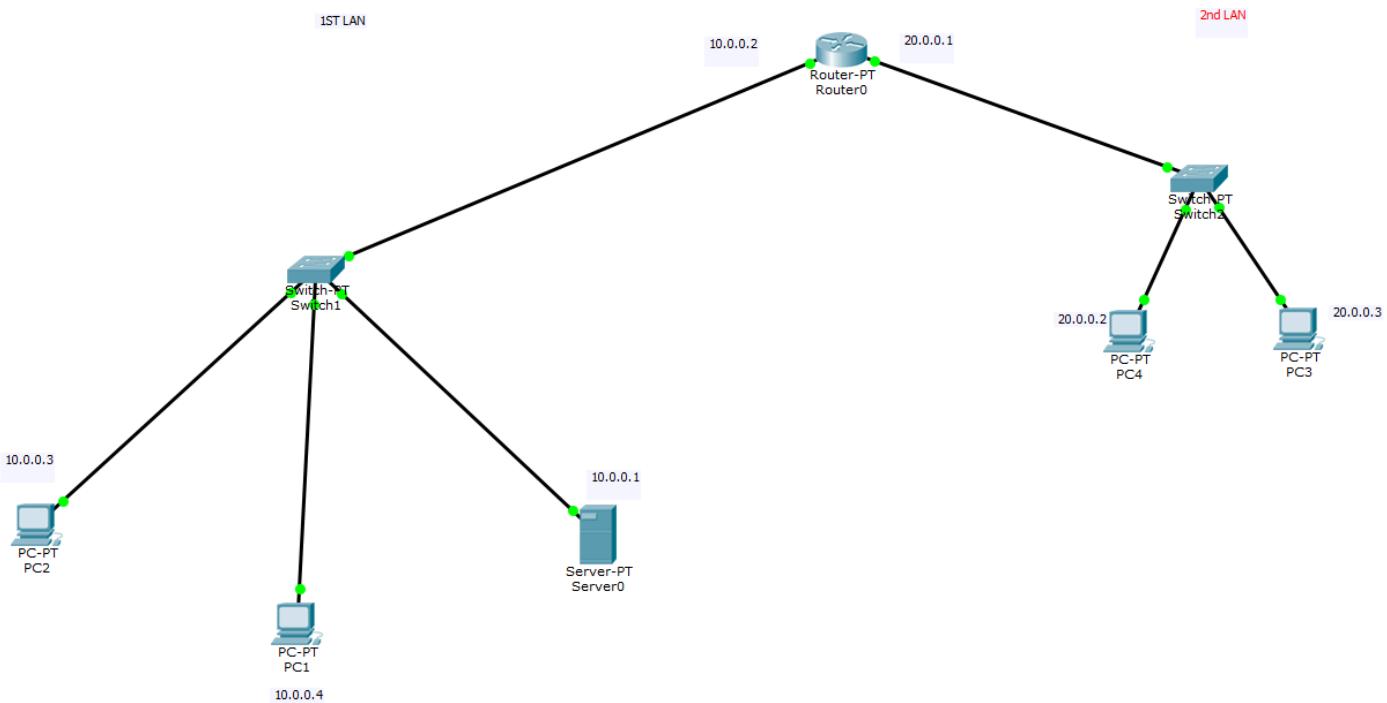
## PING COMMAND



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

<p>Configuration</p> <ul style="list-style-type: none"> <li>In the previous topology, add another return with IP 20.0.0.0 by adding 1 switch &amp; 2 PC's.</li> <li>Configure the Router using AT command.</li> </ul> <pre> Router(config)# interface fastethernet 0/0 Router(config-if)# ip address 20.0.0.1 255.0.0.0 Router(config-if)# no shutdown ip helper 10.0.0.1 exit Router(config)# interface fastethernet 0/0 ip helper 10.0.0.1 exit.     </pre> <p>→ Select PC2 &amp; PC3 &amp; select DHCP to automatically get assigned to IP addresses 10.0.0.2 &amp; 10.0.0.3 respectively using DHCP server.</p> <p>→ Now ping from PC0 to PC2 (one LAN to another LAN).</p> <p><b>Output:</b></p> <pre> PC0 &gt; ping 10.0.0.3 Reply from 10.0.0.3 bytes=32 time=0ms TTL=128     </pre> <p>Packet: sent=4 received=4 lost=0</p>	<p>Bafna Gold Date: 17/11/24 Page: 1</p> <p><b>AIM:</b> To configure DNS server to demonstrate the mapping of IP addresses &amp; domain name</p> <p><b>Configuration:</b></p> <ul style="list-style-type: none"> <li>Create Server 0, select DNS &amp; click to ON and the name as website (Domain name) &amp; set address of the server where the website is stored.</li> <li>Add address of server 0 - 10.0.0.1.</li> <li>Now, select HTTP &amp; edit the index.html to make the necessary changes in the website.</li> <li>Now, to open the website, select PC0 &amp; select Web Browser, add the website name "website".</li> <li>Now, website can be opened on PC0.</li> </ul>
---	---

## TOPOLOGY:



## ROUTER CONFIG

```
Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface fastethernet1/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#no shutdown

Router(config-if)#
*LINK-5-CHANGED: Interface FastEthernet1/0, changed state to up

*LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/0, changed state to up

Router(config-if)#ip helper 10.0.0.1
Router(config-if)#exit
Router(config)#
Router(config)#interface fastethernet0/0
Router(config-if)#ip helper 10.0.0.1
Router(config-if)#exit
Router(config)#
Router(config)#

```

## DNS SERVER CONFIG

## PING COMMAND

```
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ping 20.0.0.3

Pinging 20.0.0.3 with 32 bytes of data:

Reply from 20.0.0.3: bytes=32 time=0ms TTL=128
Reply from 20.0.0.3: bytes=32 time=1ms 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

Ping statistics for 20.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

PC>
```

Adding website name and address

**Top Left Window (FastEthernet0 Settings):**

- GLOBAL:** Port Status (On), Bandwidth (100 Mbps selected), Duplex (Full Duplex selected), MAC Address (00E0.3EDC.039A).
- INTERFACE:** FastEthernet0
  - IP Configuration:** Static IP (10.0.0.1), Subnet Mask (255.0.0.0).
  - IPv6 Configuration:** Static IPv6 Address (FE80::260:3EFF:FEDC:39A).

**Top Right Window (DNS Management):**

- SERVICES:** DNS Service (On).
- Resource Records:**

No.	Name	Type	Detail
0	website1	A Record	10.0.0.1

**Bottom Left Window (HTTP Services):**

- SERVICES:** HTTP (On), HTTPS (On).
- File Manager:**

File Name	Edit	Delete
1 copyrights.html	(edit)	(delete)
2 cscoptlogo177x...		(delete)
3 helloworld.html	(edit)	(delete)
4 image.html	(edit)	(delete)
5 index.html	(edit)	(delete)

**Bottom Right Window (File Manager):**

- SERVICES:** HTTP (selected).
- File Manager:**
  - File Name: index.html
  - Content: <html><center><font size='+2' color='blue'>Cisco Packet Tracer</font></center><br>Welcome to computer network lab<br>Quick Links:<br><a href='helloworld.html'>introduction</a><br><a href='copyrights.html'>DHCP</a><br><a href='image.html'>DNS</a><br><a href='cscoptlogo177x111.jpg'>Image</a></html>

Fig: Accessing and changing index.html

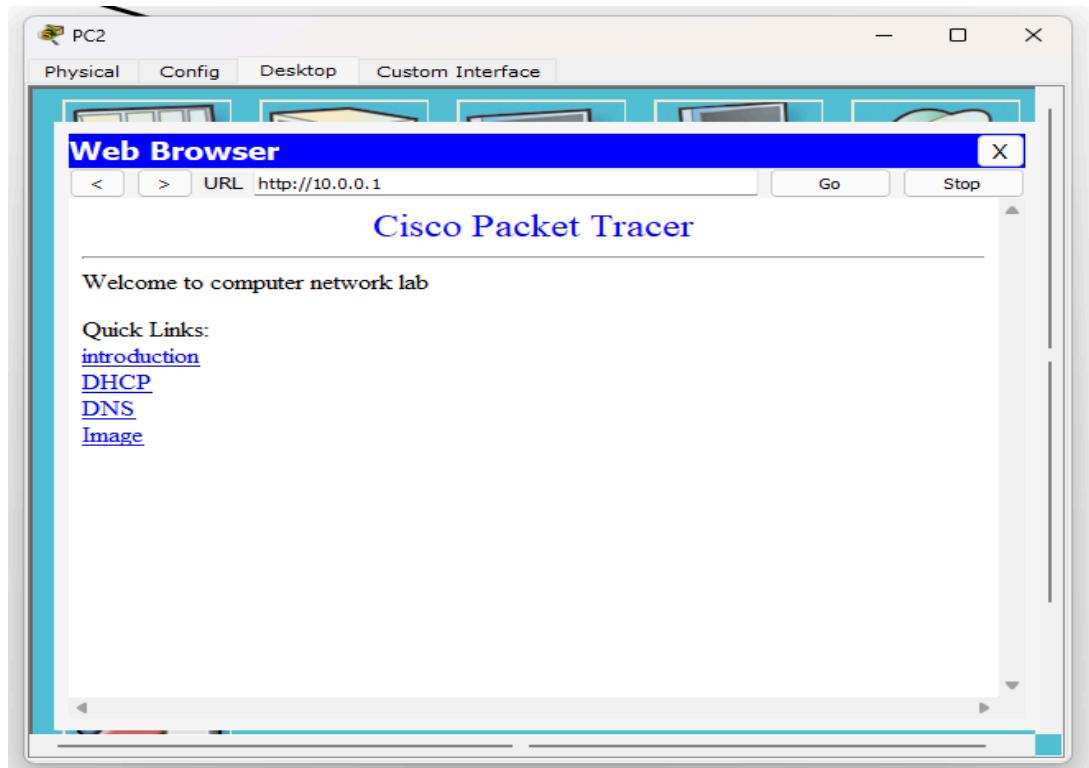


Fig : webBrowser output

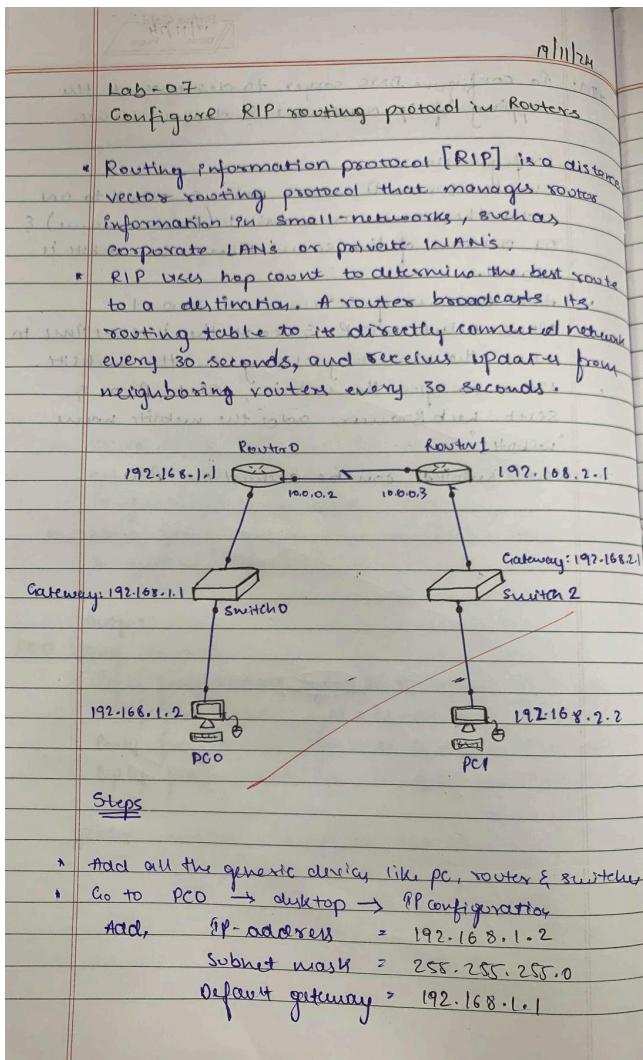
NAME : MOHITH JAIN

USN: 1BM22CS162

CN LAB 7

AIM: TO CONFIGURE RIP ROUTING PROTOCOL IN ROUTERS.

OBSERVATION:



Bafna Gold -  
Date: \_\_\_\_\_ Page: \_\_\_\_\_

\* 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 serial 2/0 interface  
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

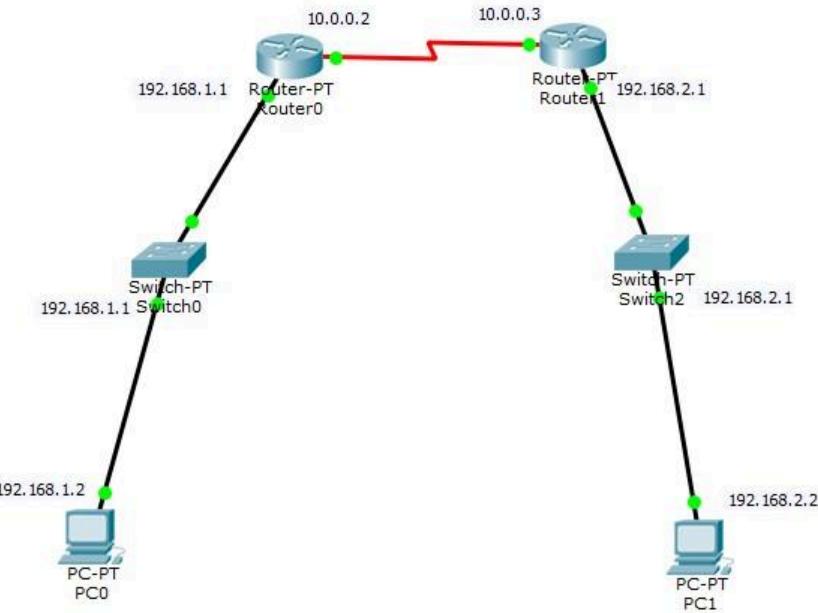
\* Go to Router 1 → config → fastEthernet 0/0  
Add, IP Address = 192.168.2.1  
Subnet mask = 255.255.255.0  
Set port status:  ON  
In serial 2/0 interface.  
Clock Rate: NOT SET  
IP Address = 10.0.0.3  
Subnet mask: 255.0.0.0  
Port status:  ON

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

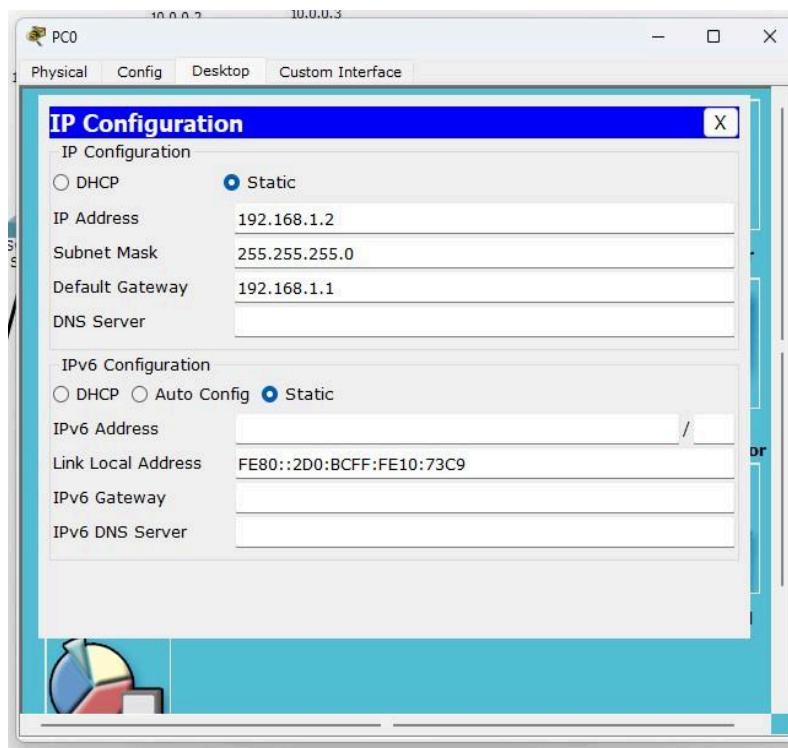
Observation:

- Ping message from PC0 to PC1  
Status: Success  
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,

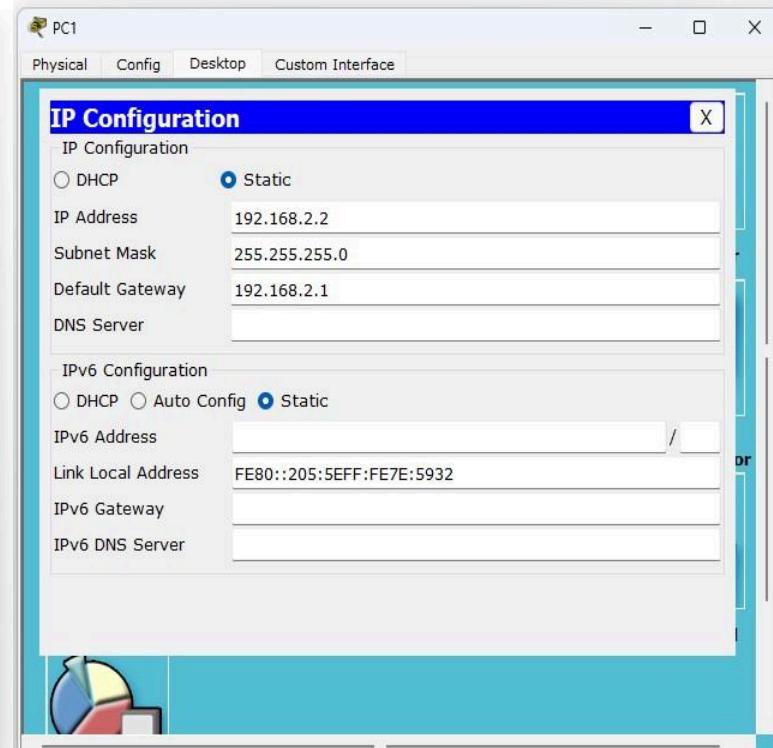
## TOPOLOGY:



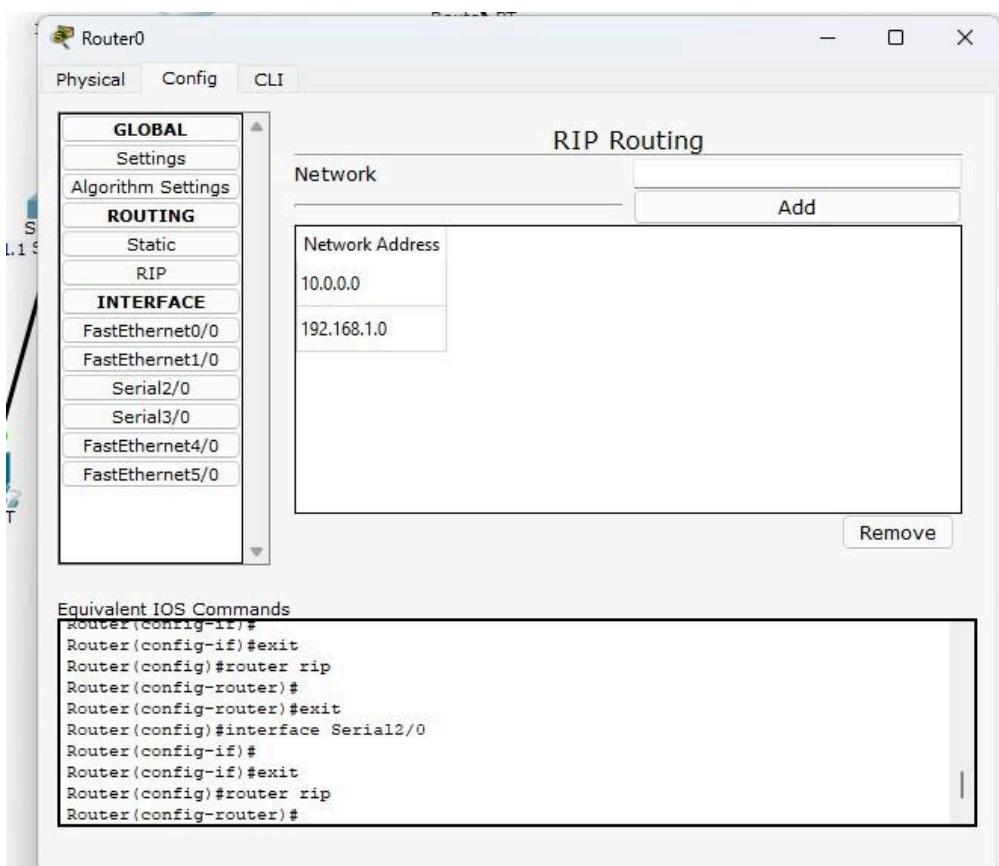
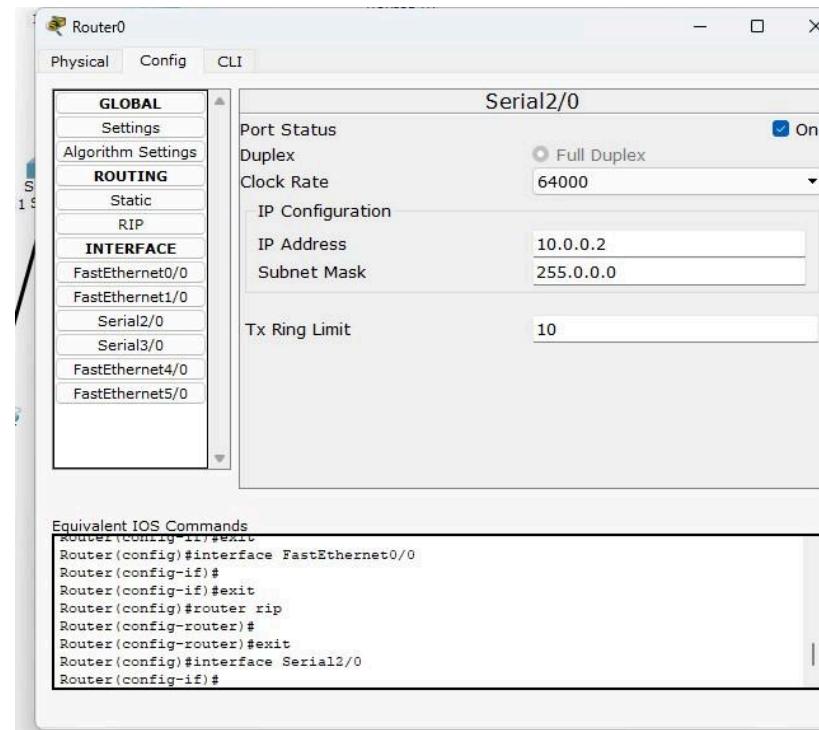
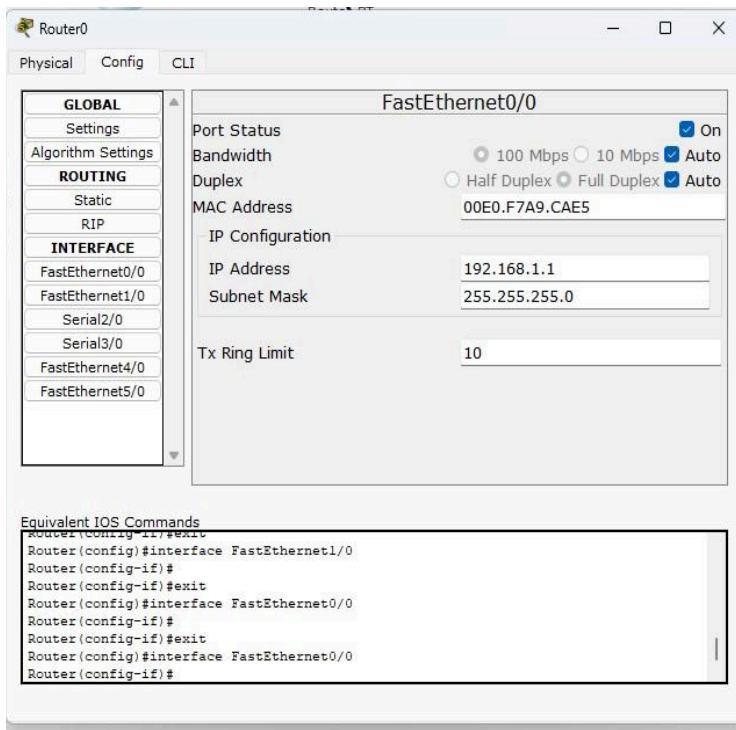
## PC\_0 CONFIGURATION:



## PC\_1 CONFIGURATION:



## ROUTER\_0 CONFIGURATION:



## ROUTER\_1 CONFIGURATION:

**FastEthernet0/0 Configuration:**

- Port Status:** On
- Bandwidth:** 100 Mbps (selected)
- Duplex:** Half Duplex (selected)
- MAC Address:** 0090.21E9.B3AA
- IP Configuration:**
  - IP Address:** 192.168.2.1
  - Subnet Mask:** 255.255.255.0
- 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 FastEthernet0/0
Router(config-if)#
```

**Serial2/0 Configuration:**

- Port Status:** On
- Duplex:** Full Duplex (selected)
- Clock Rate:** Not Set
- IP Configuration:**
  - IP Address:** 10.0.0.3
  - Subnet Mask:** 255.0.0.0
- Tx Ring Limit:** 10

**Equivalent IOS Commands:**

```
Router(config)#interface Serial2/0
Router(config-if)#exit
Router(config)#router rip
Router(config-router)#version 2
Router(config-router)#network 10.0.0.0
Router(config-router)#exit
Router(config)#exit
```

**RIP Routing Configuration:**

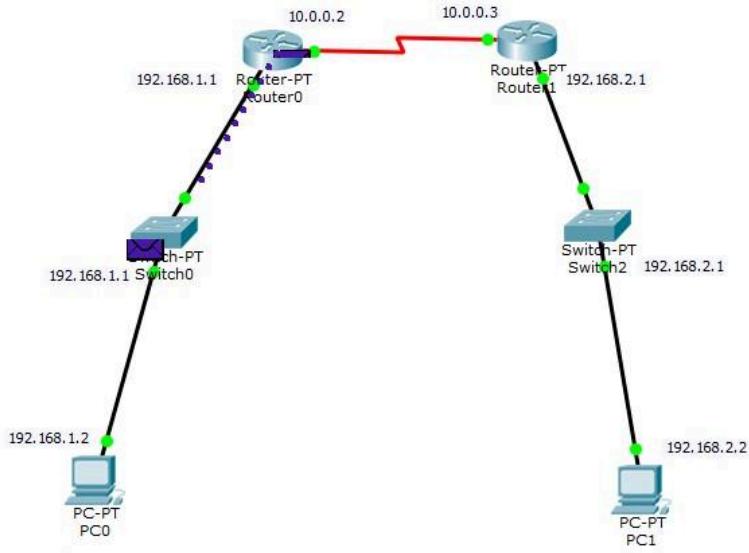
**Network:**

Network Address	Add
10.0.0.0	
192.168.2.0	

**Equivalent IOS Commands:**

```
Router(config)#router rip
Router(config-router)#version 2
Router(config-router)#network 10.0.0.0
Router(config-router)#network 192.168.2.0
Router(config-router)#exit
Router(config)#exit
```

## Packet transfer / ping



PC0

Physical Config Desktop Custom Interface

Command Prompt X

```
Packet Tracer PC Command Line 1.0
PC>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 TTL=126
Reply from 192.168.2.2: bytes=32 time=4ms TTL=126
Reply from 192.168.2.2: bytes=32 time=3ms TTL=126
Reply from 192.168.2.2: bytes=32 time=3ms TTL=126

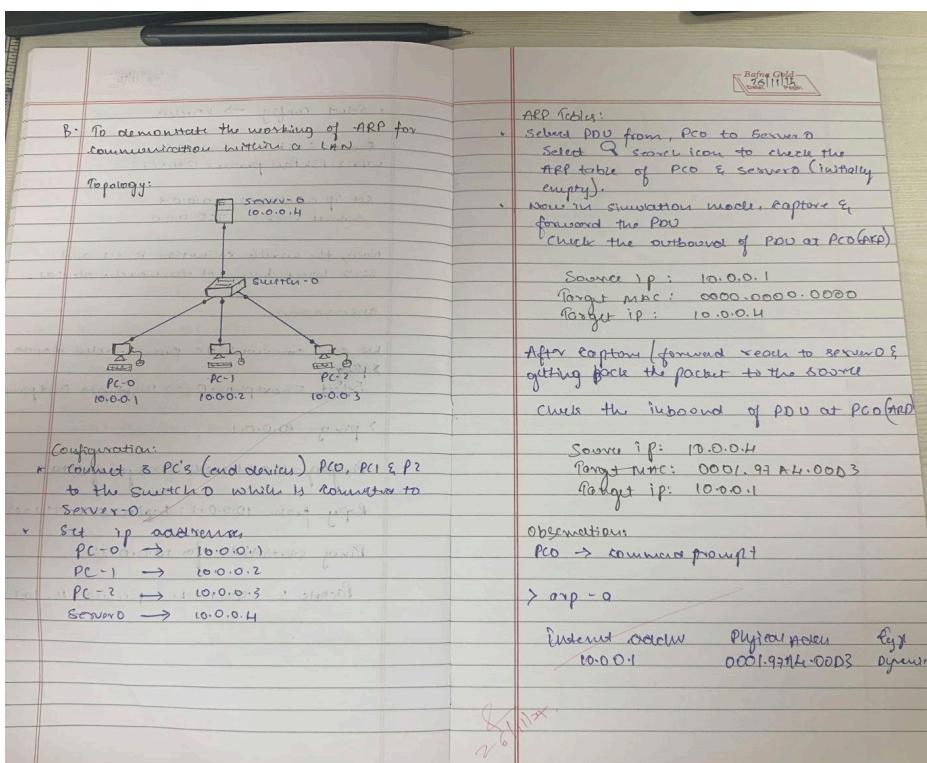
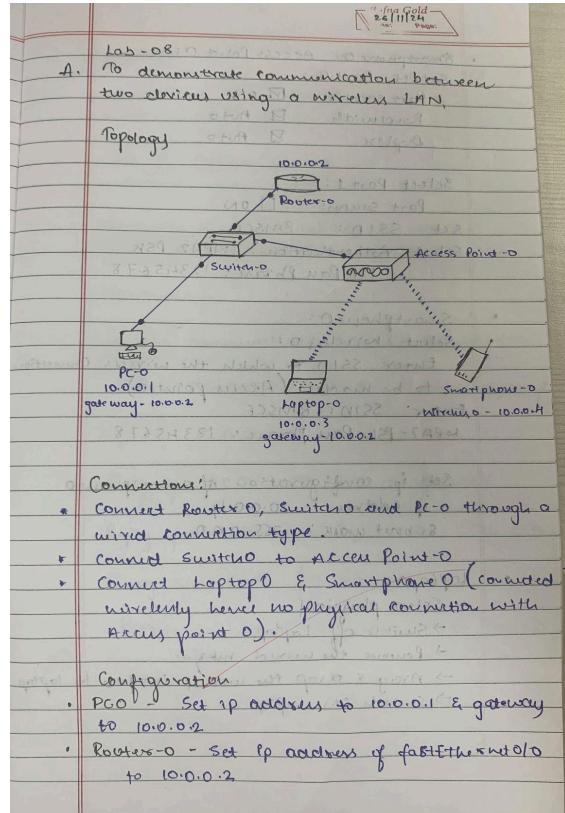
Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 4ms, Average = 2ms

PC>
```

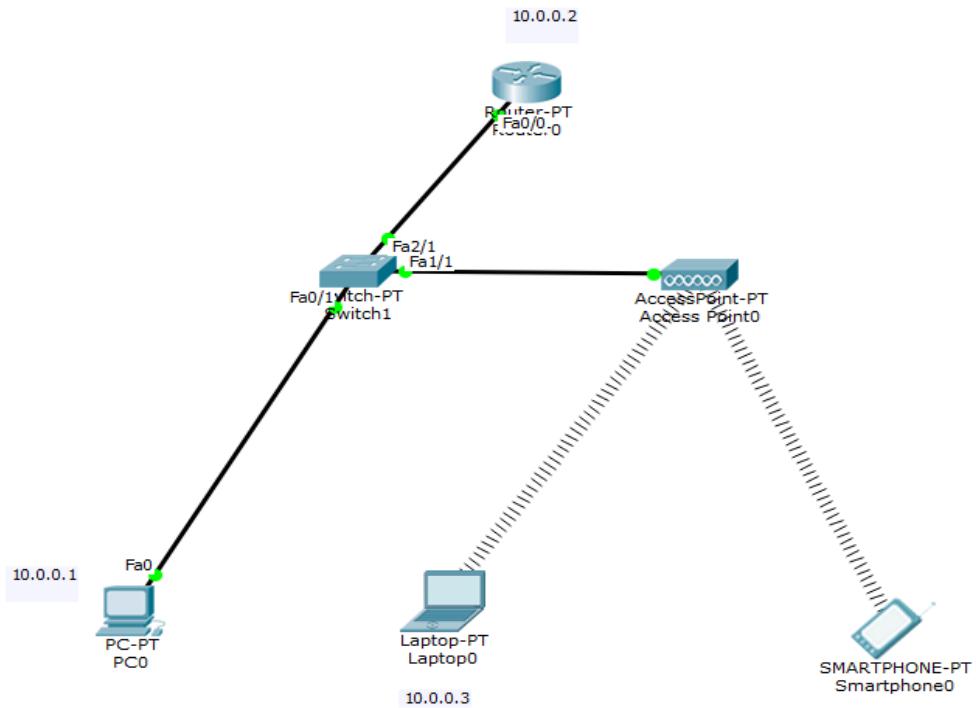
## CN LAB 8

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

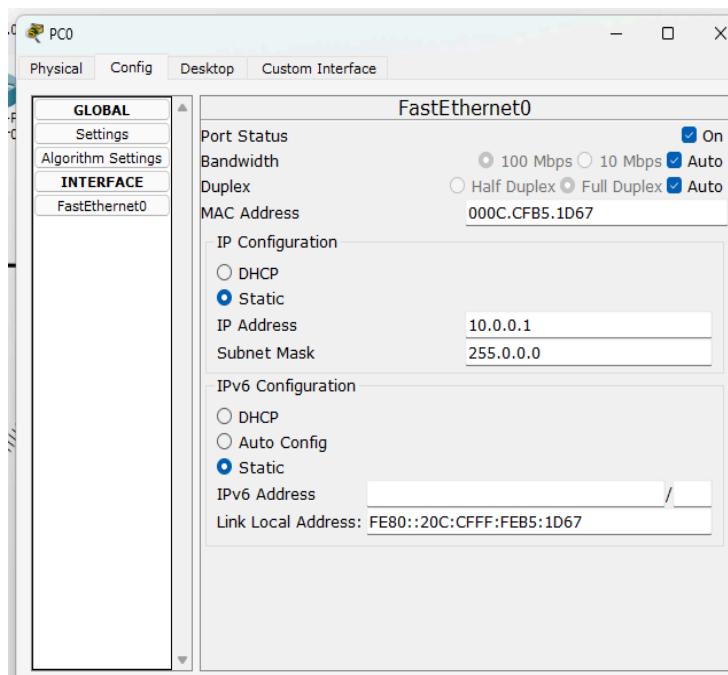
## OBSERVATION:



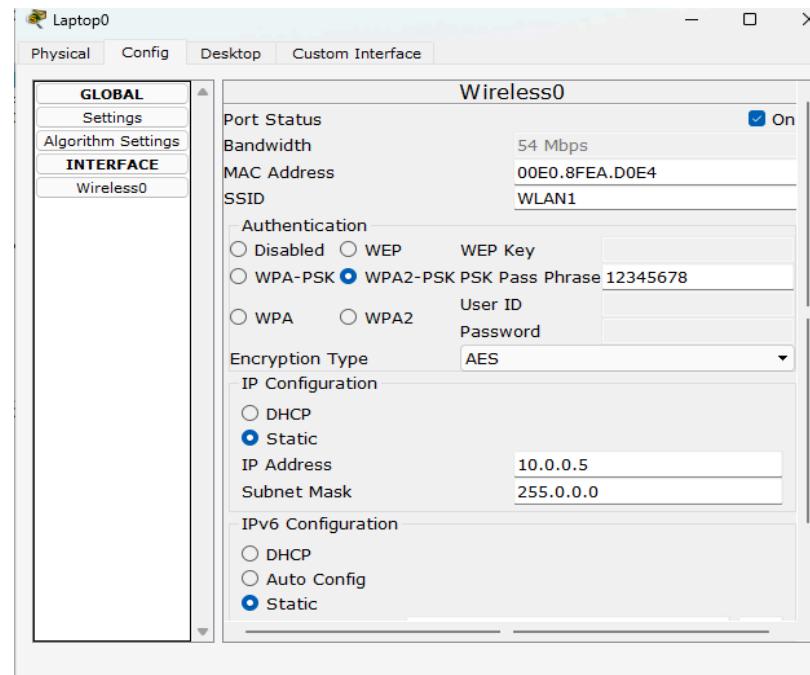
## TOPOLOGY:



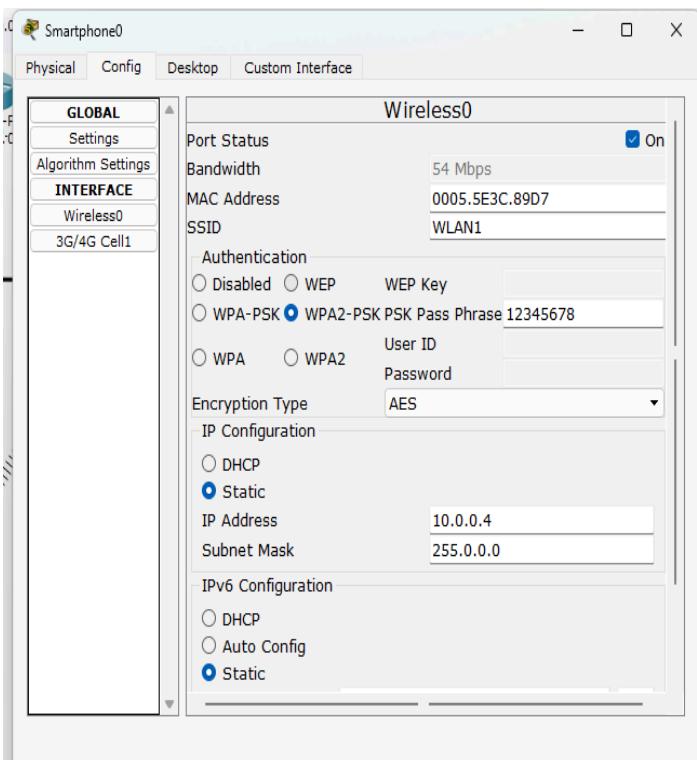
## PC0 CONFIGURATION:



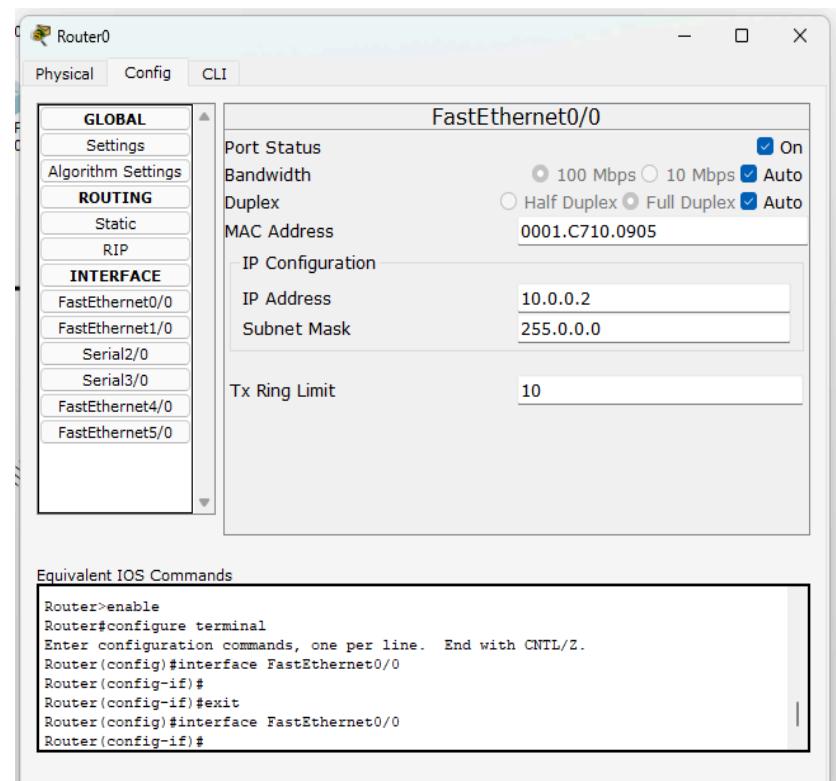
## LAPTOP CONFIGURATION:



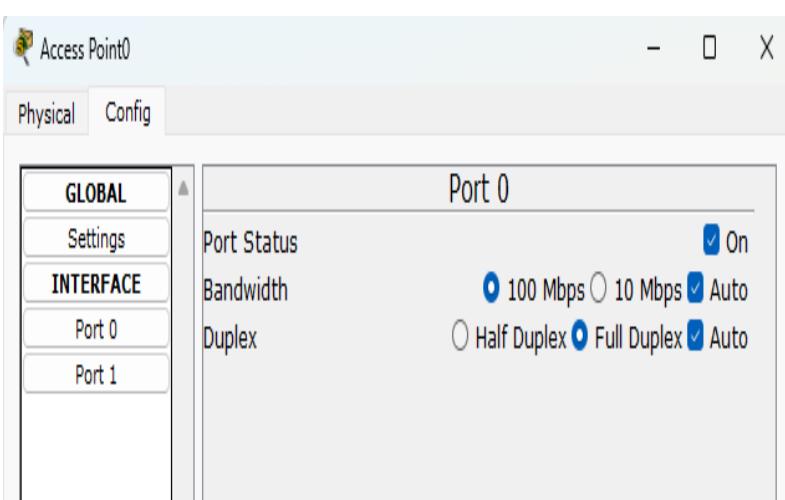
### SMART-PHONE CONFIGURATION:



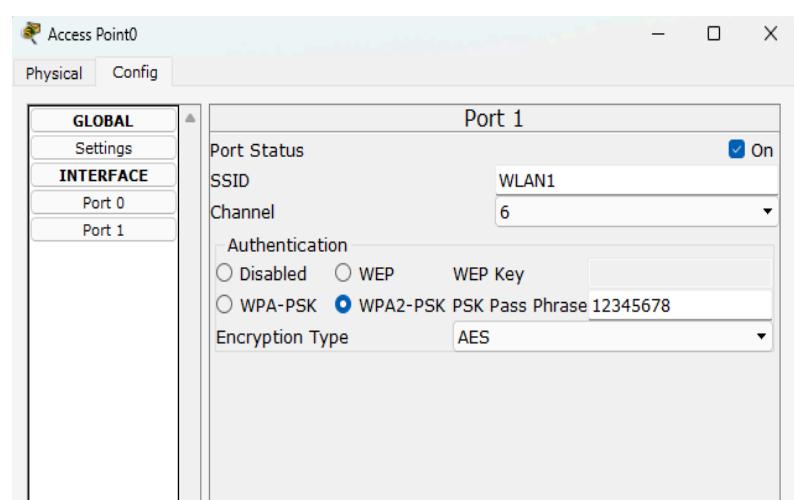
### ROUTER0 CONFIGURATION:



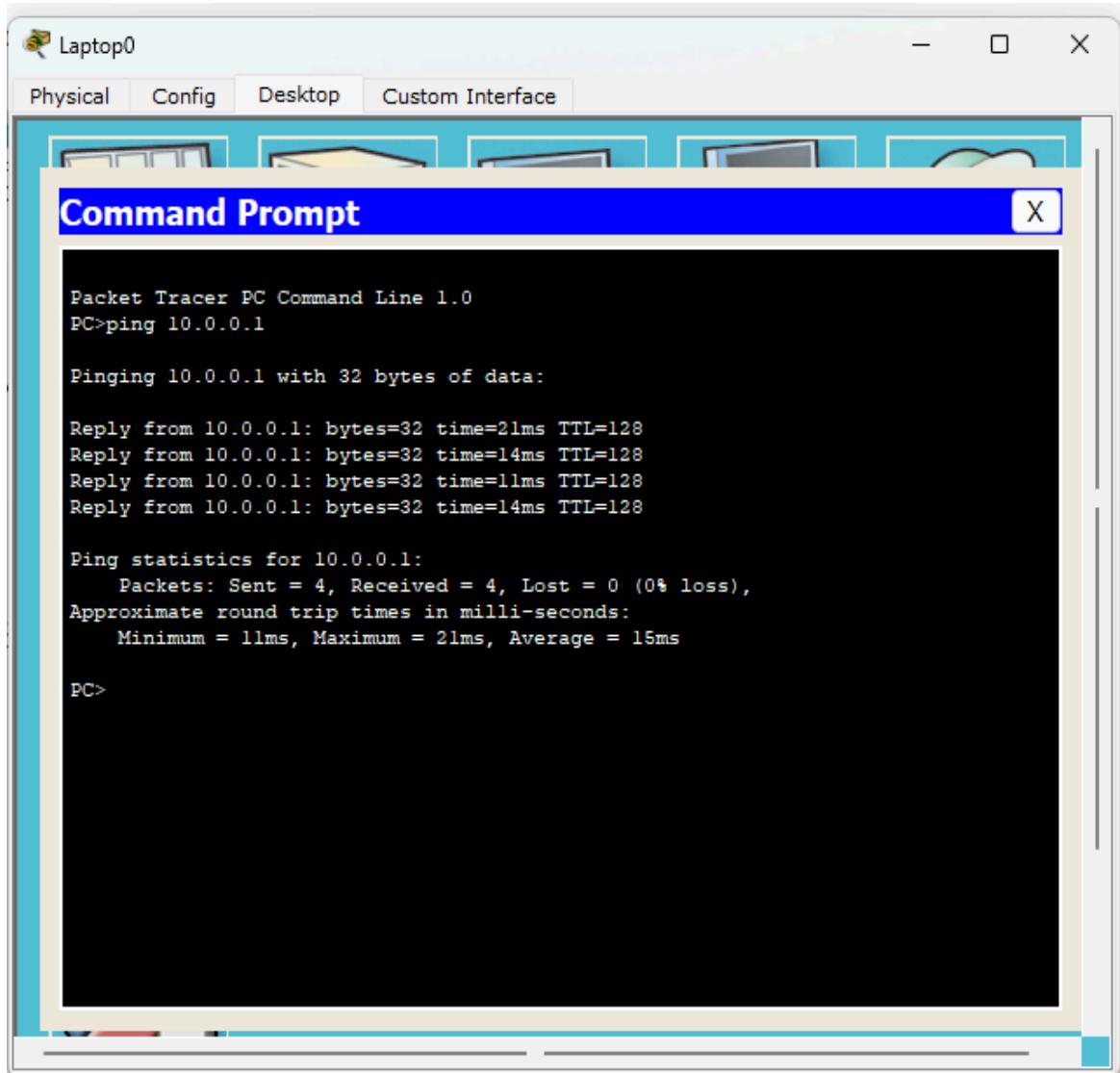
### ACCESS-POINT PORT0



### ACCESS-POINT PORT1



## PING FROM LAPTOP TO PC



Laptop0

Physical Config Desktop Custom Interface

Command Prompt X

```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

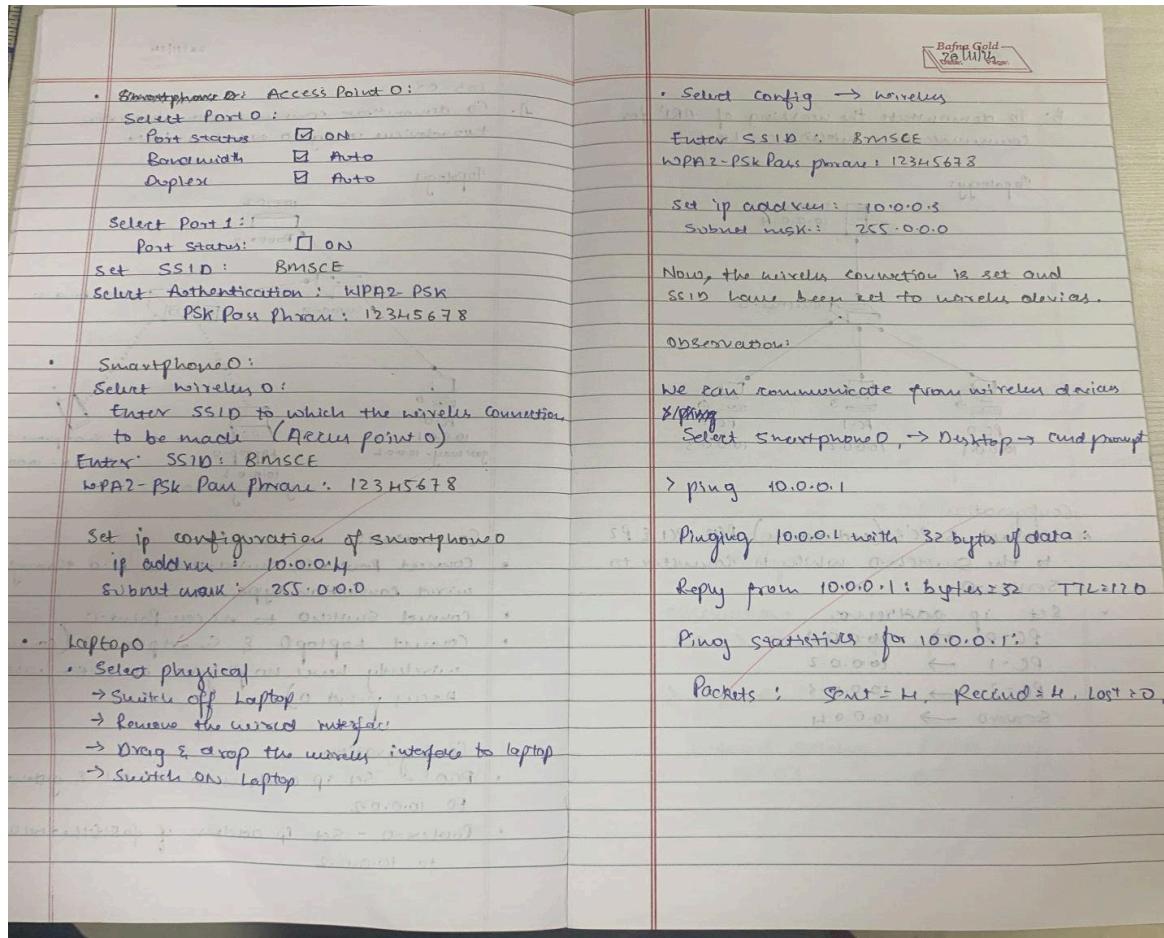
Reply from 10.0.0.1: bytes=32 time=21ms TTL=128
Reply from 10.0.0.1: bytes=32 time=14ms TTL=128
Reply from 10.0.0.1: bytes=32 time=11ms TTL=128
Reply from 10.0.0.1: bytes=32 time=14ms TTL=128

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 11ms, Maximum = 21ms, Average = 15ms

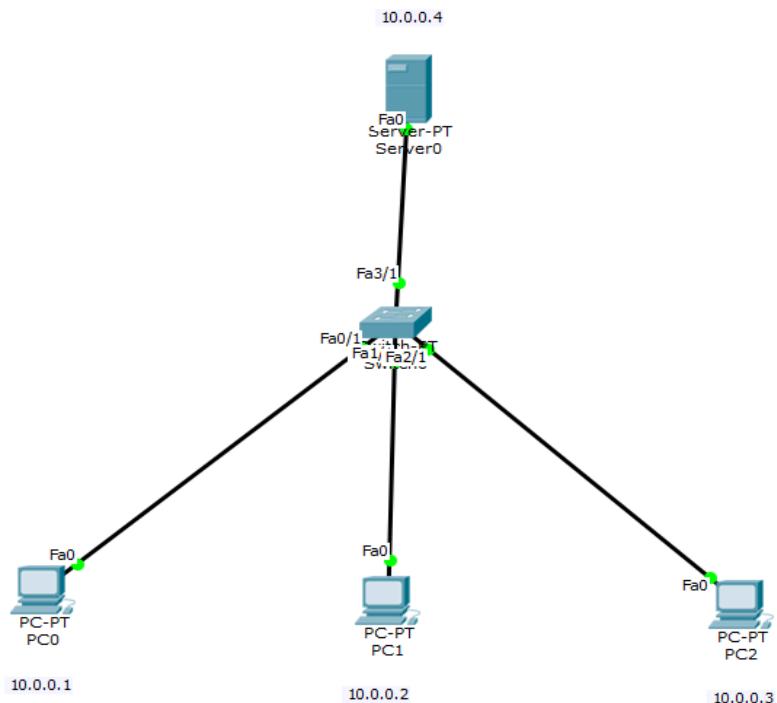
PC>
```

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

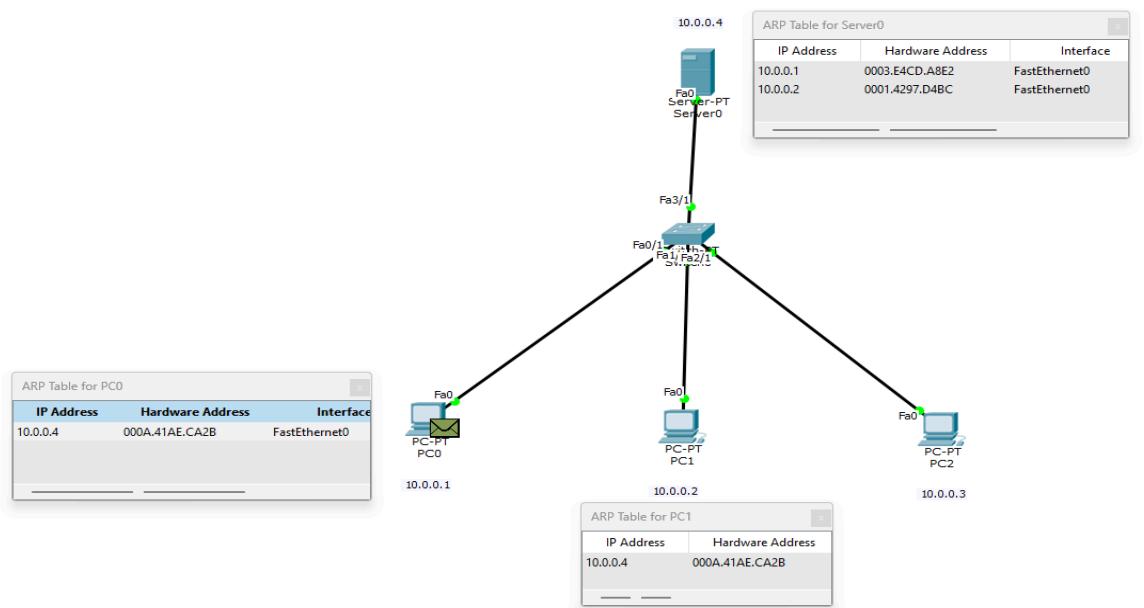
### OBSERVATION:



## TOPOLOGY:



## ARP TABLES:



## PC0 INBOUND

PDU Information at Device: PC0

OSI Model Inbound PDU Details

PDU Formats

Ethernet II

0	4	8	14	19 Bytes
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

Ethernet II

0	4	8	14	19 Bytes
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	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	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

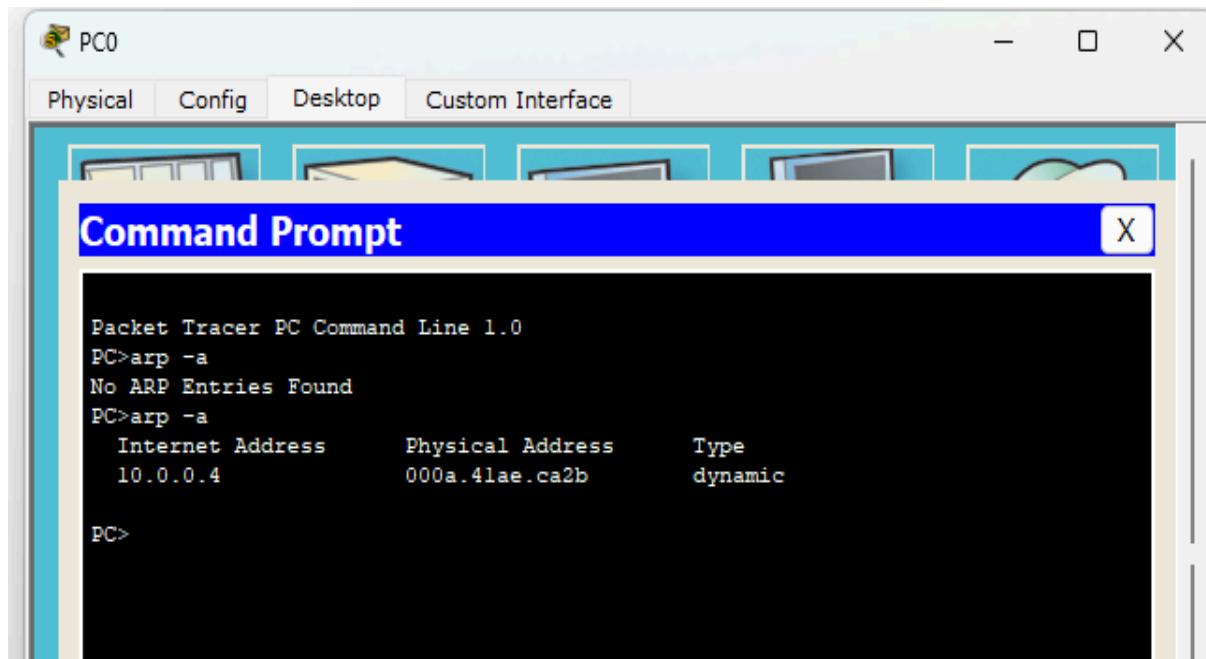
IP

0	4	8	16	19	31 Bits
4	IHL	DSCP: 0x0	TL: 28		
ID: 0x7	0x0	0x0	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: 0x8	CODE: 0x0	CHECKSUM	
ID: 0x3		SEQ NUMBER: 2	

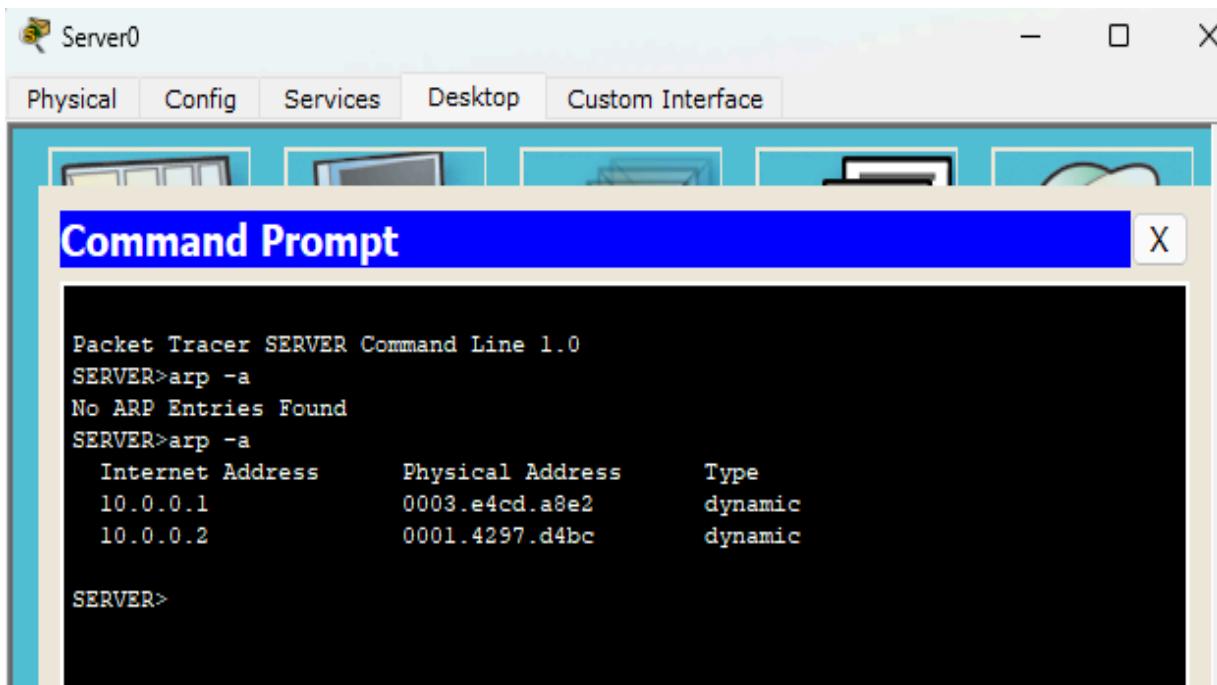
### PING FROM PC0



A screenshot of the Packet Tracer software interface for a host named "PC0". The window title is "PC0". The tab bar at the top includes "Physical", "Config", "Desktop", and "Custom Interface", with "Physical" being the active tab. Below the tabs is a toolbar with icons for different network components. A "Command Prompt" window is open, titled "Command Prompt". The command line shows the output of the "arp -a" command:

```
Packet Tracer PC Command Line 1.0
PC>arp -a
No ARP Entries Found
PC>arp -a
  Internet Address      Physical Address      Type
  10.0.0.4                000a.41ae.ca2b    dynamic
PC>
```

### PING FROM SERVER0



A screenshot of the Packet Tracer software interface for a host named "Server0". The window title is "Server0". The tab bar at the top includes "Physical", "Config", "Services", "Desktop", and "Custom Interface", with "Physical" being the active tab. Below the tabs is a toolbar with icons for different network components. A "Command Prompt" window is open, titled "Command Prompt". The command line shows the output of the "arp -a" command:

```
Packet Tracer SERVER Command Line 1.0
SERVER>arp -a
No ARP Entries Found
SERVER>arp -a
  Internet Address      Physical Address      Type
  10.0.0.1                0003.e4cd.a8e2    dynamic
  10.0.0.2                0001.4297.d4bc    dynamic
SERVER>
```

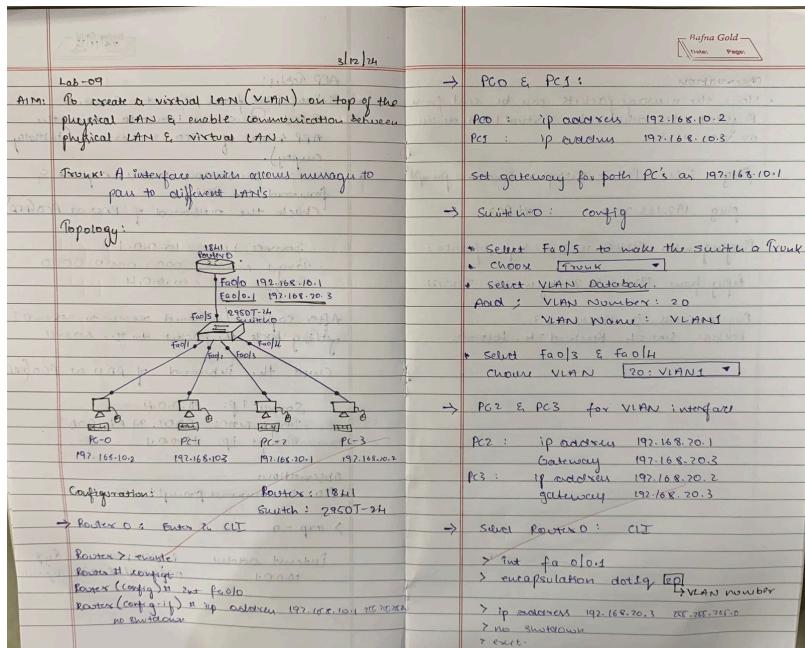
NAME : Mohith Jain

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:



**Observation**

- Now the message packets can be sent from Physical LAN devices to virtual LAN devices or vice-versa.
- ping Select PC0 > distop > command prompt

ping 192.168.20.2 # PG3

Plugging with 192.168.20.2 32 bytes of data:

Reply from 192.168.20.2: bytes=32 TTL=127

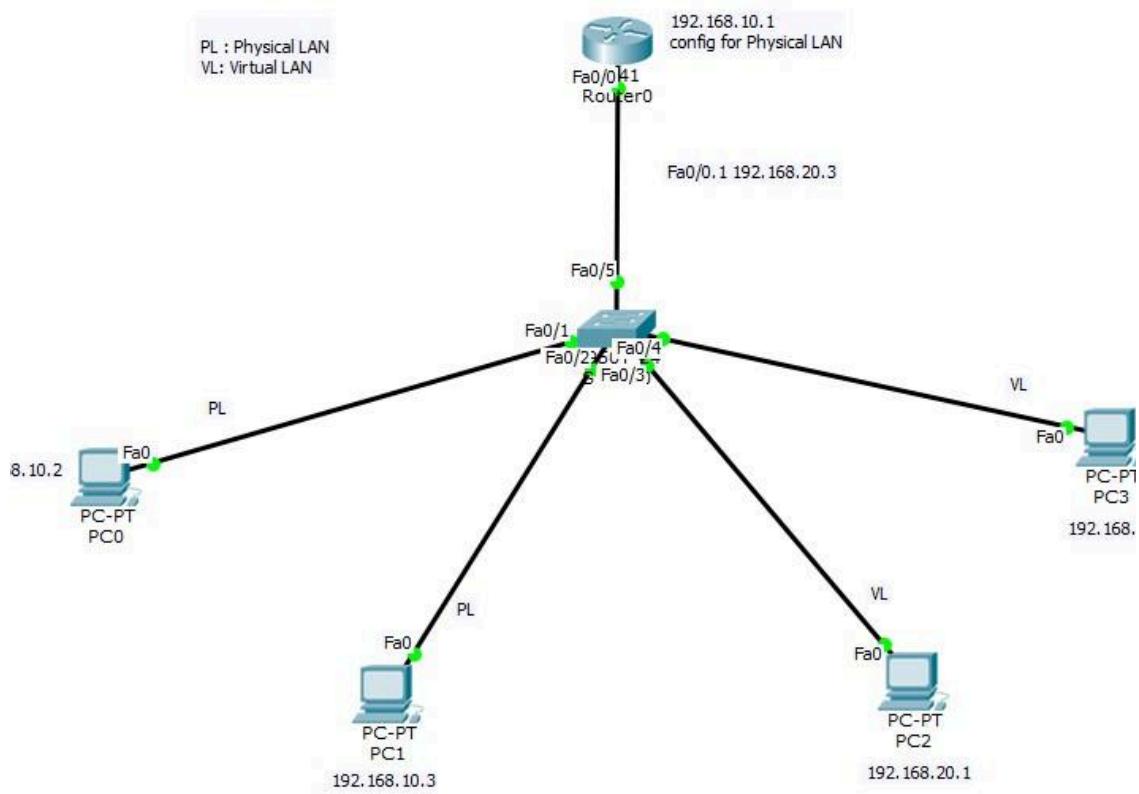
Ping statistics for 192.168.20.2

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

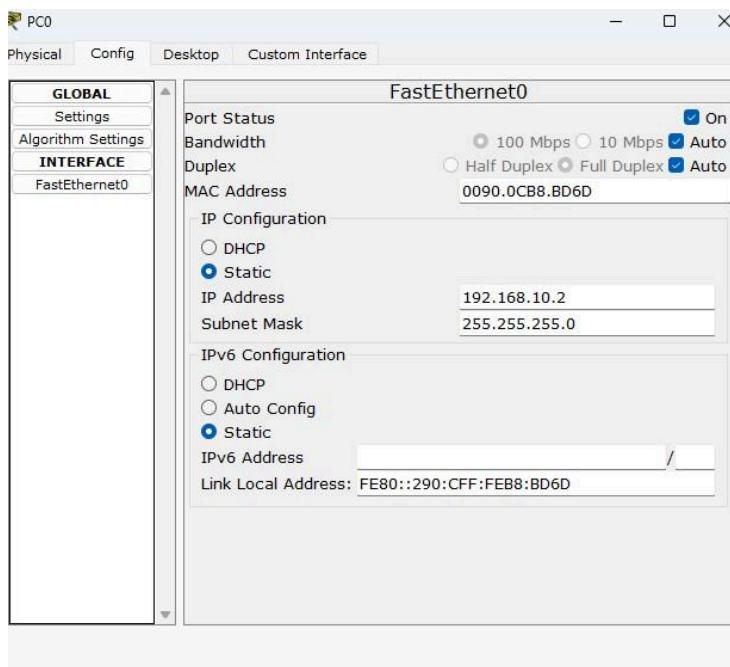
Approximate round trip times in ms:

Min = 0.000	Max = 0.000	Avg = 0.000
-------------	-------------	-------------

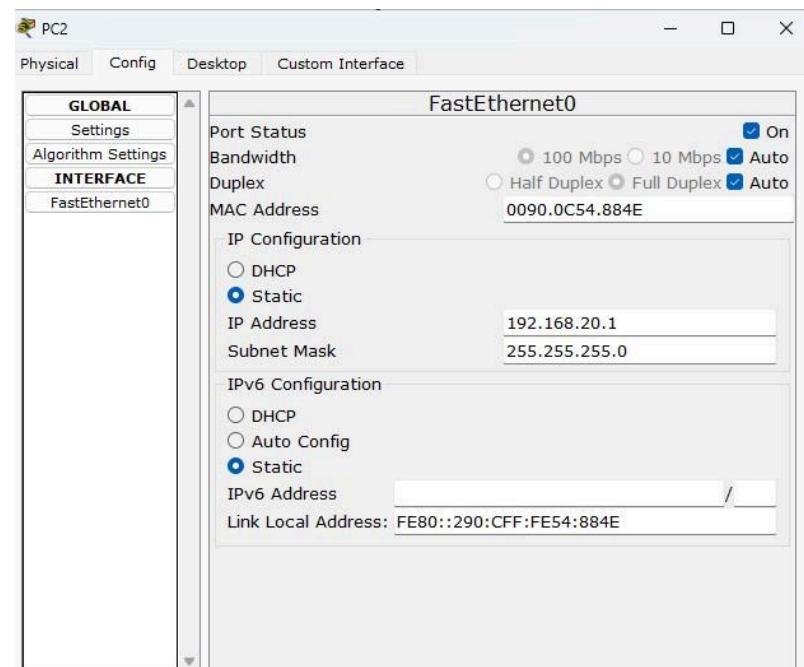
## TOPOLOGY:



## PC0 CONFIGURATION:



## PC2 CONFIGURATION:



## PC2 Gateway CONFIGURATION:

The screenshot shows the 'Global Settings' configuration window for PC2. The 'Display Name' is set to 'PC2'. Under 'Gateway/DNS', 'Static' is selected, with 'Gateway' set to 192.168.20.3 and 'DNS Server' set to 0.0.0.0. There are sections for 'Gateway/DNS Ipv6' with 'Static' selected, and fields for 'IPv6 Gateway' and 'IPv6 DNS Server'.

## ROUTER0 CONFIGURATION:

The screenshot shows the 'IOS Command Line Interface' for Router0. It displays the configuration commands entered via the terminal. The session starts with 'enable', then enters configuration mode with 'config t'. It configures interface fa0/0 with IP address 192.168.10.1 and subnet mask 255.255.255.0. It then exits configuration mode and returns to privileged EXEC mode. The terminal also shows interface status changes: 'LINK-5-CHANGED' for interface fa0/0 changing state to up, and 'LINEPROTO-5-UPDOWN' for the line protocol on interface fa0/0 changing state to up.

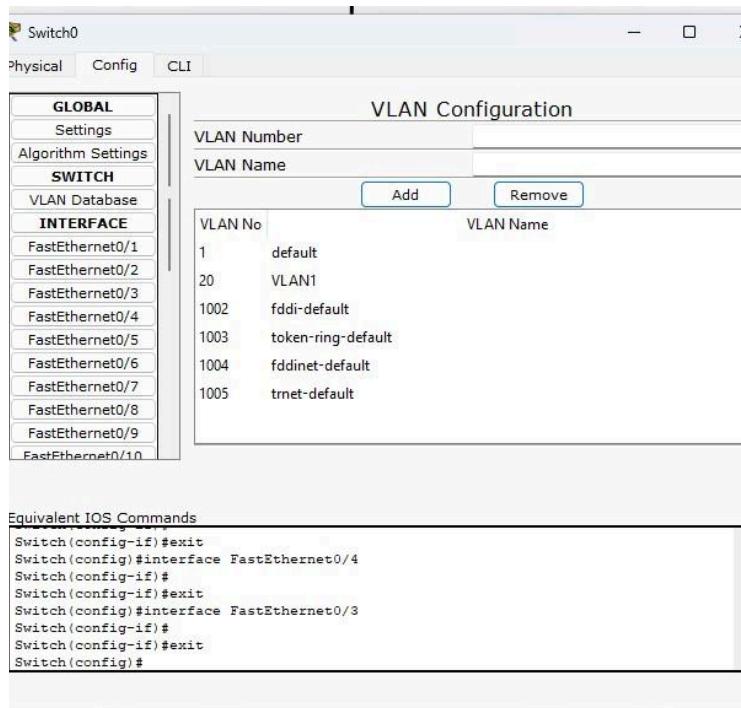
## SWITCH Fa 0/3 configuration

The screenshot shows the configuration for FastEthernet0/3 on Switch0. The port is set to 'Access' mode, VLAN 20, and Tx Ring Limit is 10. Other settings include Port Status (On), Bandwidth (100 Mbps), Duplex (Full Duplex), and Auto negotiation. The left sidebar lists other interfaces: FastEthernet0/1 through 10.

## SWITCH Fa 0/5 configuration

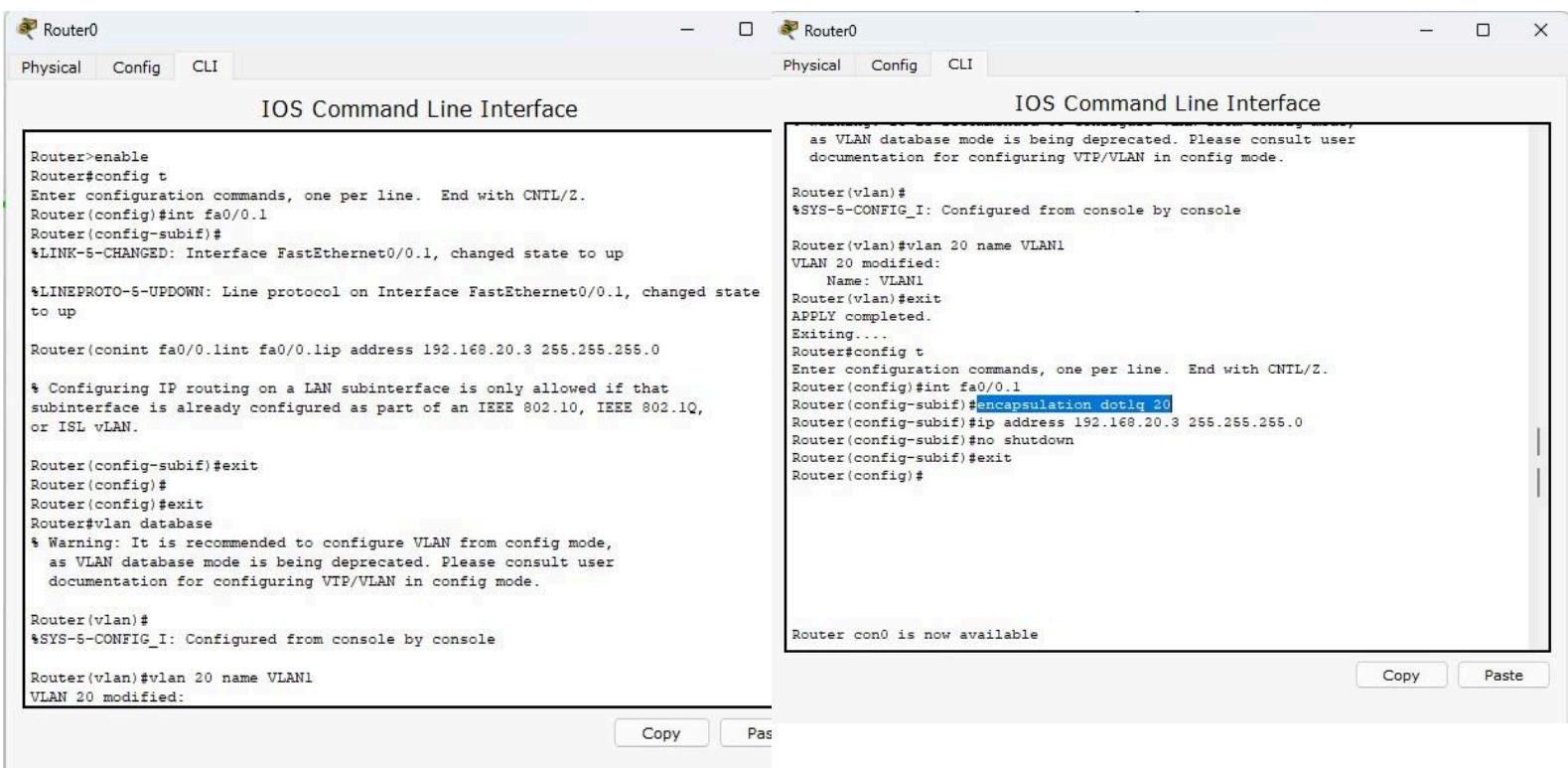
The screenshot shows the configuration for FastEthernet0/5 on Switch0. The port is set to 'Trunk' mode, VLAN 2-1001, and Tx Ring Limit is 10. Other settings include Port Status (On), Bandwidth (100 Mbps), Duplex (Full Duplex), and Auto negotiation. The left sidebar lists other interfaces: FastEthernet0/1 through 10.

## SWITCH VLAN Database



Creation of Fa 0/0.1

## Encapsulating Fa 0/0.1



## Ping message packet from PC0 to PC2

The screenshot shows the Cisco Packet Tracer PC Command Line interface. The window title is "Command Prompt". The text output is as follows:

```
Packet Tracer PC Command Line 1.0
PC>ping 192.168.20.2

Pinging 192.168.20.2 with 32 bytes of data:

Request timed out.
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127

Ping statistics for 192.168.20.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

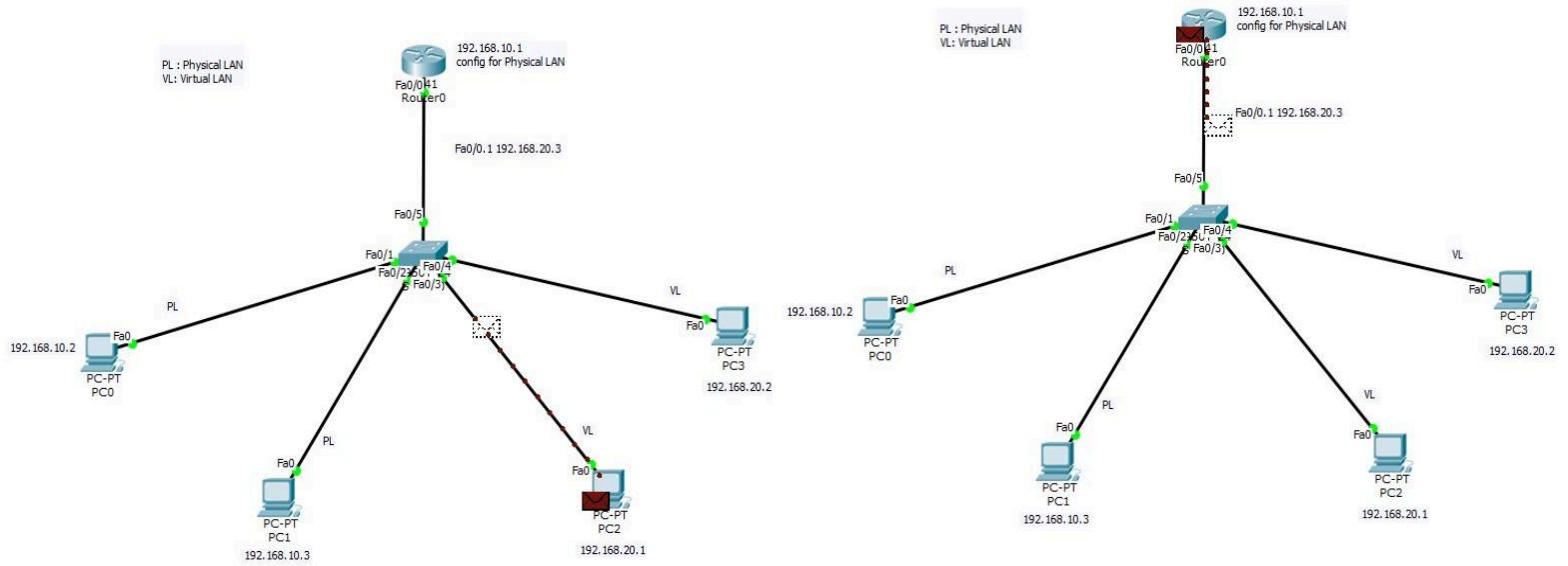
PC>ping 192.168.20.2

Pinging 192.168.20.2 with 32 bytes of data:

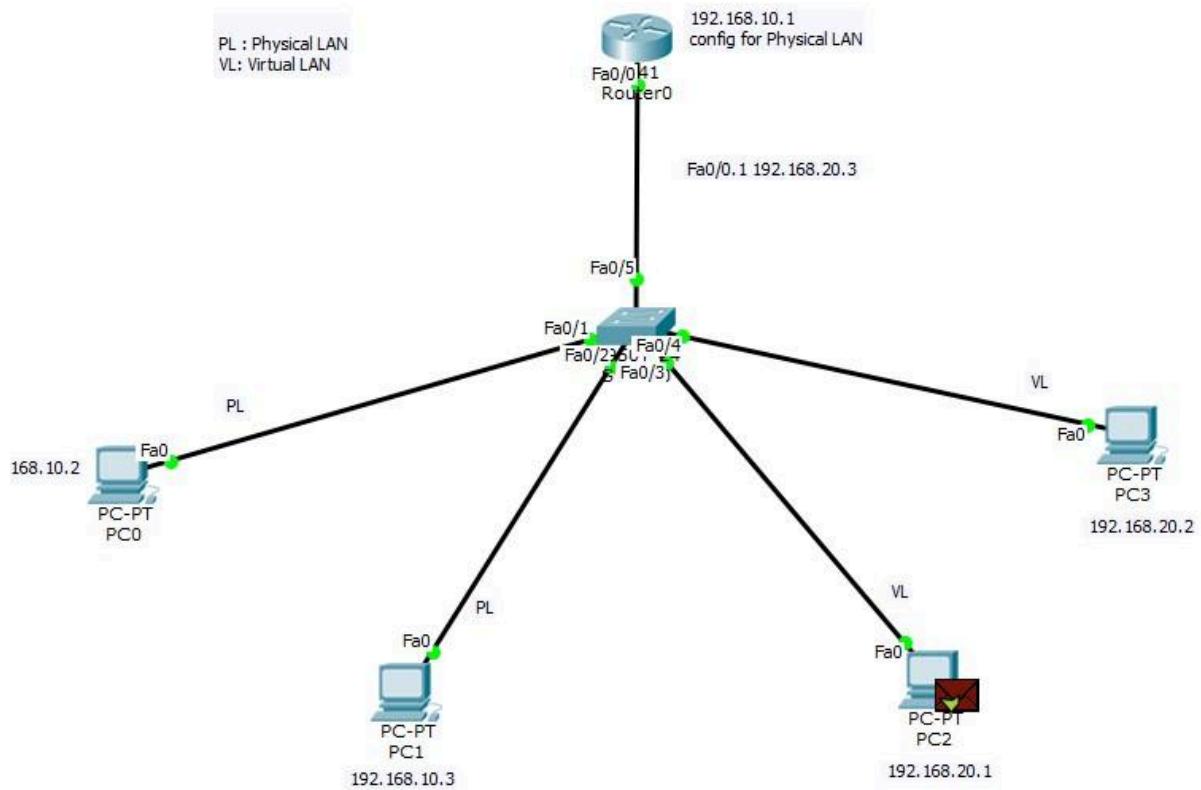
Reply from 192.168.20.2: bytes=32 time=1ms TTL=127
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127
Reply from 192.168.20.2: bytes=32 time=0ms TTL=127

Ping statistics for 192.168.20.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

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). Decrease counter by size of packet p.
3. Reset counter & go to step ①

#### Code:

```
#include <iostream>
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 buffer  
    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 = 1

Buffer size = 9 out of bucket size = 10

? (buffer size => 9 & 10, 9 & 10, 10 & 10)

10 & 10, 9 & 10, 9 & 9 = 4

9 & 9, 9 & 9

9 & 9, 10 & 10 => 9 & 10 => "lossy" >> 100%

"lossy" & 10 & 10 => "lossy"

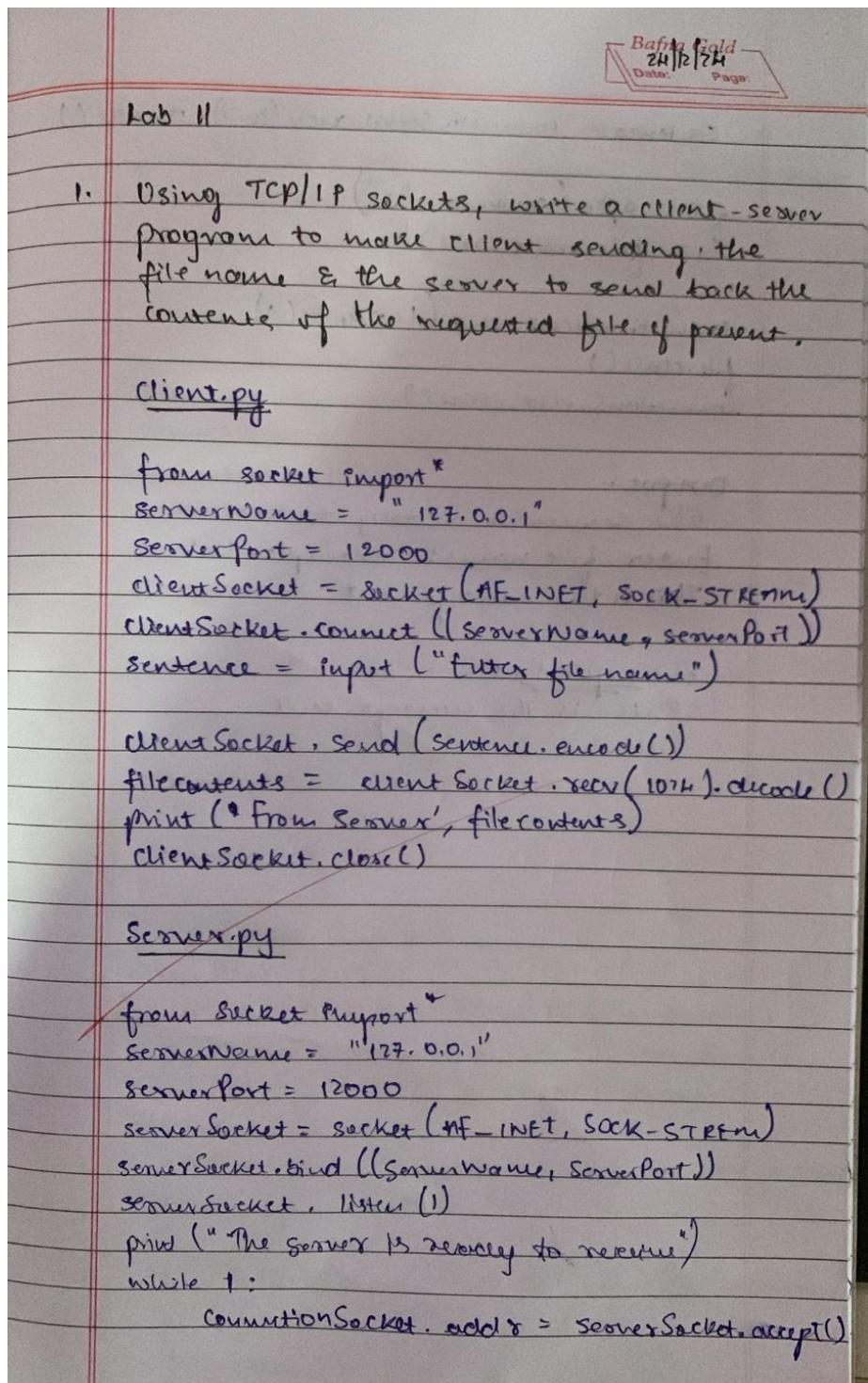
9 & 9, 9 & 9 = - "lossy"

9 & 9, 9 & 9 (9 & 9) => "lossy"

(DRAFT)

## 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.



Sentence = connectionSocket.recv(1024).decode()  
 file = open(sentence, "r")  
 l = file.read(1024)  
 connectionSocket.send(l.encode())  
 file.close()  
 connectionSocket.close()

Output:  
 The server is ready to receive  
 Enter file name: example.txt  
 This is the message sent  
 <example.txt>  
 This is the message sent

## Code:

### **ClientTCP.py**

```

from socket import *
serverName = "127.0.0.1";
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input("Enter file name: ")
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print ("From Server:")
print(filecontents)
clientSocket.close()
  
```

### **ServerTCP.py**

```

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

## Output

```
▶ 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)

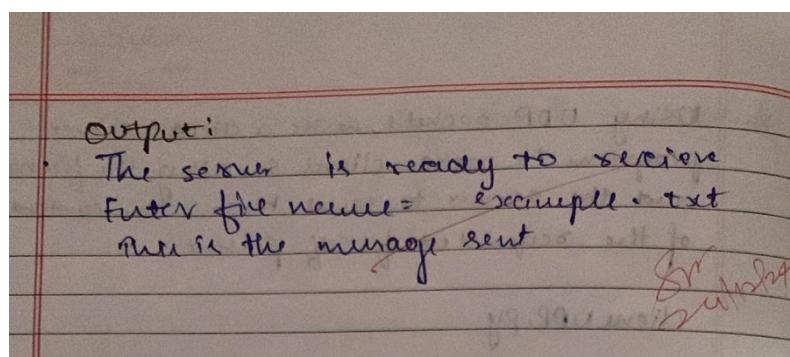
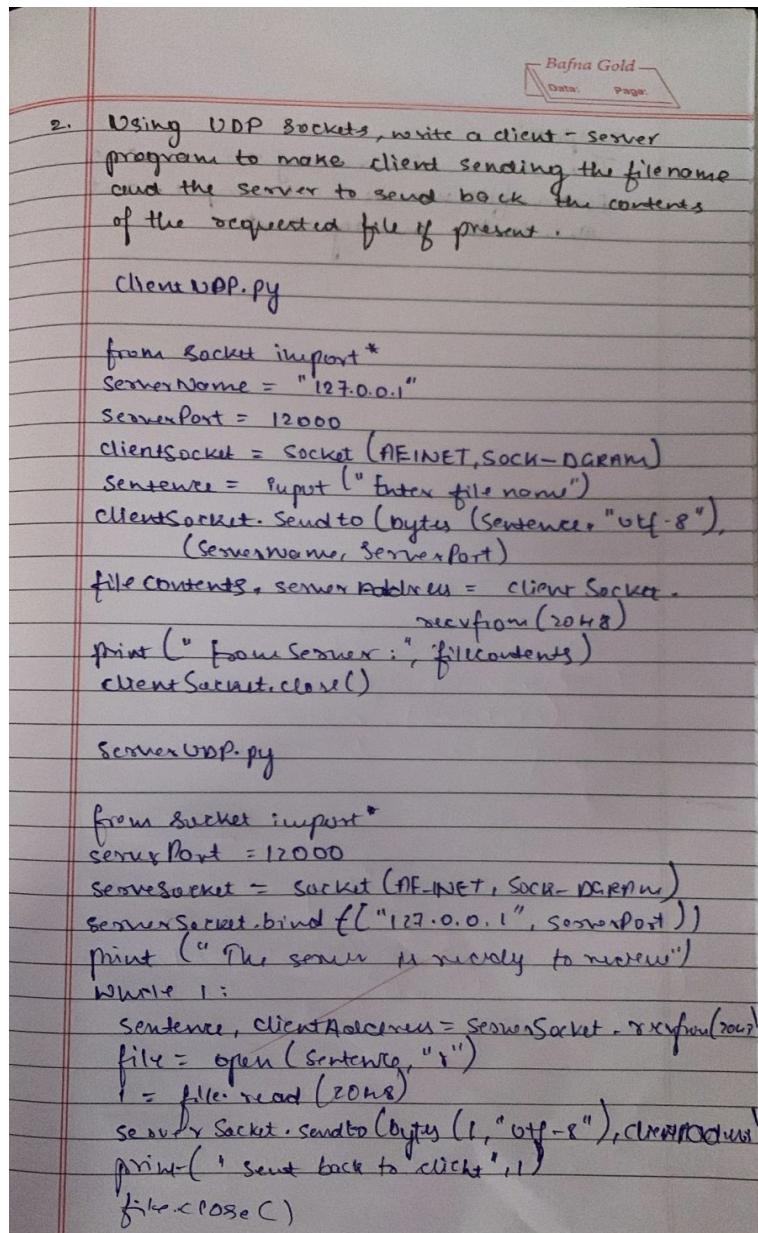
finally:
    # Terminate the server process
    server_process.terminate()
    print("\nServer process terminated.")
```

```
平淡 Starting server.py...
平淡 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.



Code:

**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 ("Reply from Server:")
print (filecontents.decode("utf-8"))
# for i in filecontents:
# print(str(i), end = '')
clientSocket.close()
clientSocket.close()
```

**ServerUDP.py**

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

## Output

```
▶ 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
```