

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB RECORD

Computer Network Lab (23CS5PCCON)

Submitted by

Sidhvin Vidyadhar Burli(1BM21CS211)

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 **Sidhvin Vidyadhar Burli(1BM21CS211)** 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.

Srushti C S Assistant Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE
--	--

Index

Sl. No.	Date	Experiment Title	Page No.
1	09-10-24	Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping messages.	4-8
2	09-10-24	Configure IP address to routers in packet tracer. Explore the following messages: ping responses, destination unreachable, request timed out, reply.	9-11
3	16-10-24	Configure default route, static route to the Router (Part 1).	12-15
4	23-10-24	Configure default route, static route to the Router (Part 2).	16-20
5	13-11-24	Configure DHCP within a LAN and outside LAN.	21-26
6	20-11-24	Configure RIP routing Protocol in Routers .	27-30
7	20-11-24	Demonstrate the TTL/ Life of a Packet.	31-33
8	27-11-24	Configure OSPF routing protocol.	34-37
9	18-12-24	Configure Web Server, DNS within a LAN.	38-39
10	18-12-24	To construct a simple LAN and understand the concept and operation of Address Resolution Protocol (ARP).	40-42
11	18-12-24	To understand the operation of TELNET by accessing the router in the server room from a PC in the IT office.	43-45
12	18-12-24	To construct a VLAN and make the PC's communicate among a VLAN.	46-49
13	18-12-24	To construct a WLAN and make the nodes communicate wirelessly.	50-52
14	18-12-24	Write a program for error detecting code using CRC-CCITT (16-bits).	53-54
15	18-12-24	Write a program for congestion control using Leaky bucket algorithm.	55-58
16	18-12-24	Using TCP/IP sockets, write a client-server program to make the client send the file name and the server to send back the contents of the requested file if present.	59-61

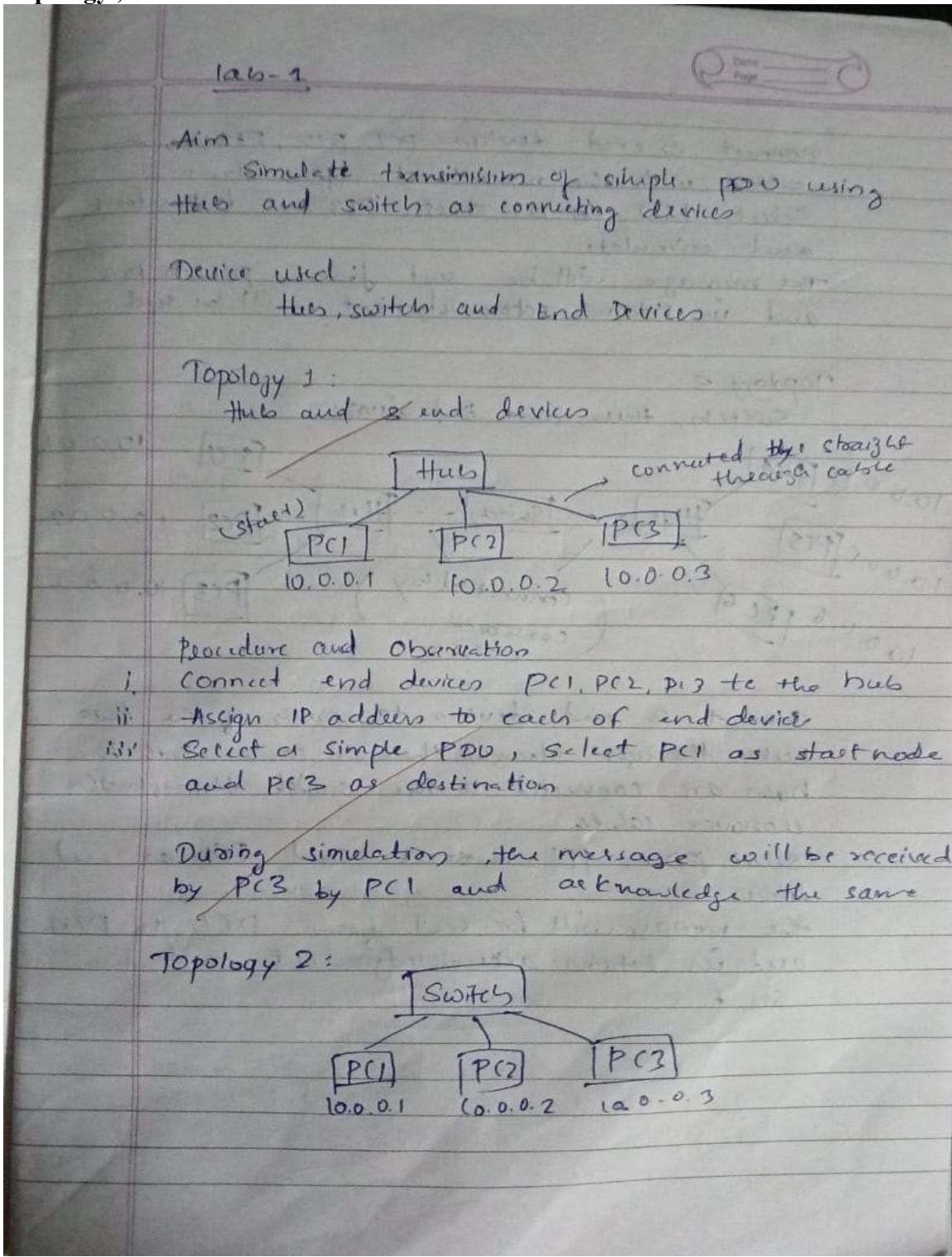
17	18-12-24	Using UDP sockets, write a client-server program to make the client send the file name and the server to send back the contents of the requested file if present.	62-64
----	----------	---	-------

Github Link: <https://github.com/SidhvinBurli/CNLAB5TH>

Program 1

Aim: Create a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices and demonstrate ping messages.

Topology , Procedure and Observation:



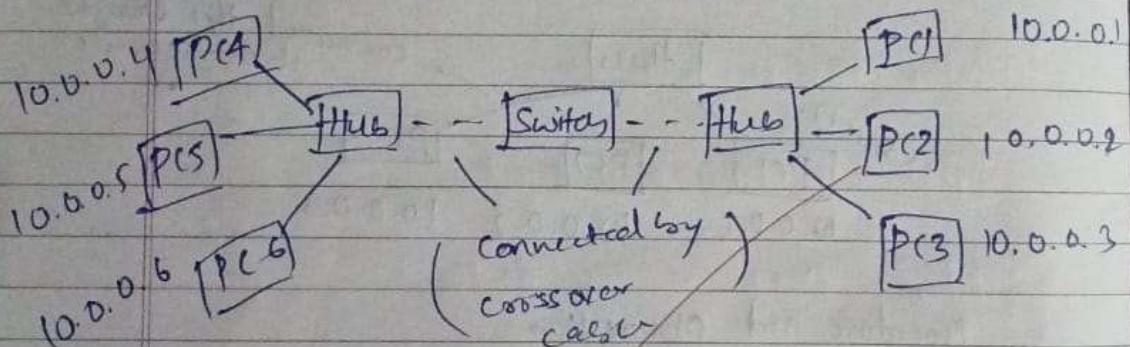
Connect 3 end devices PC1, PC2, PC3, to the switch with the mentioned IP address

Select PDU, PC1 as start PC3 as destination and stimulate

The message will be sent from PC1 to PC3 and in return acknowledgement will be sent

Topology 3:

Switch, hub and End device



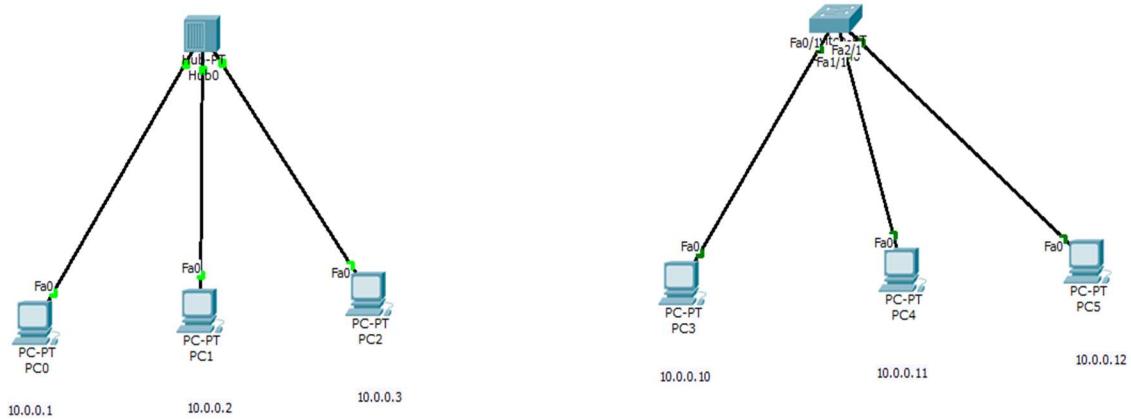
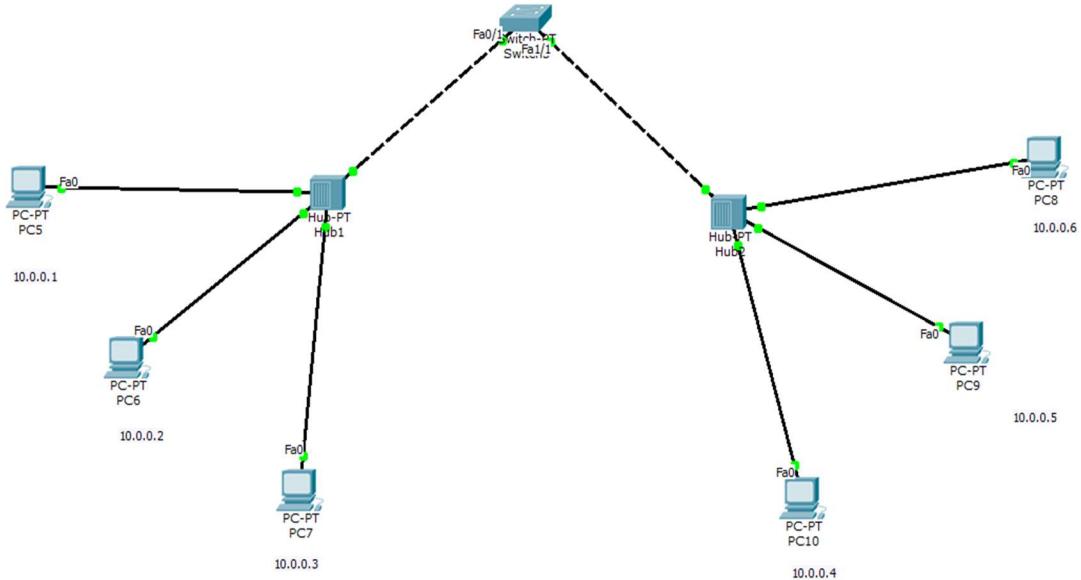
Connect 3 end devices to each of the two hubs through straight through cables. Both the hubs are connected to the switch through the crossover cables.

Select PDU, PC1 as start and PC4 as end devices

The message will be sent from PC1 to PC4 and in return acknowledgement will be sent

Straight
11/10/24

Screen Shots:



Program 2

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

Topology , Procedure and Observation:

LAB-2

Aim : Simulate transmission of sample PDU using routers

Devices
Routers, End Device

Topology

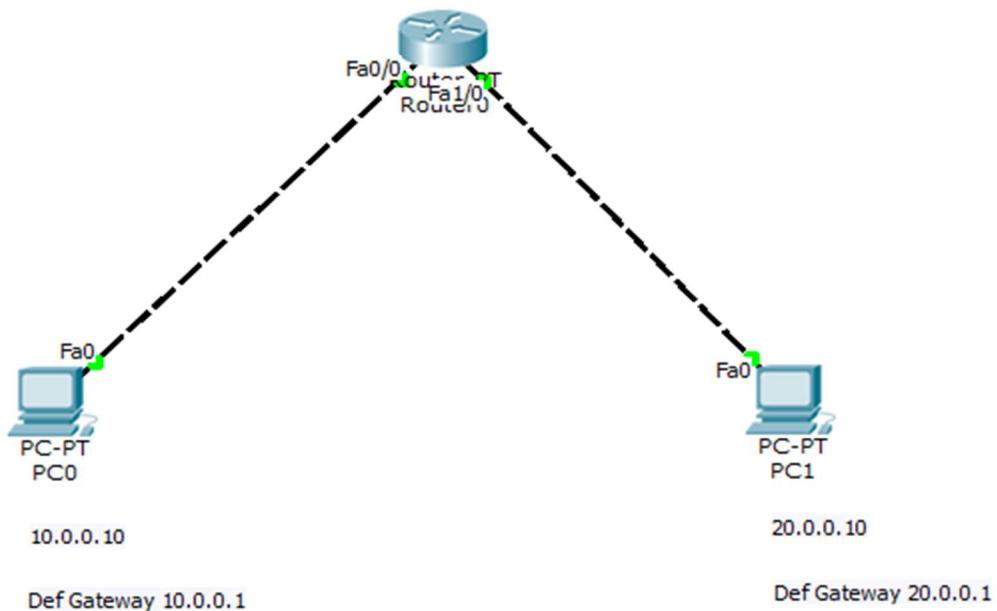
```
graph TD; Router --- PC1[PC1<br/>10.0.0.1]; Router --- PC2[PC2<br/>20.0.0.1]
```

Procedure and objectives

- * Connect the two end devices to the Router
- * Configure the Router
- * Set IP address and gateway for both of the end devices

During simulation, message sent by the PC1 is received by the PC2 and ~~not~~ in return the acknowledgement is received by the PC1

Screen Shots:



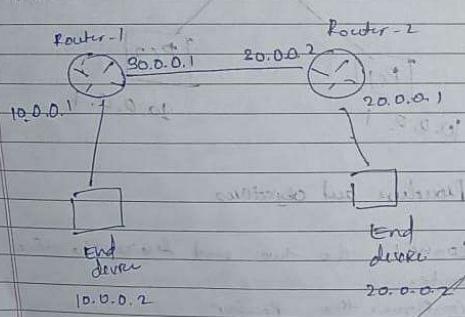
A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the output of several ping commands. The first set of pings is from PC0 to PC1 (IP 20.0.0.10), resulting in 25% loss. The second set is from PC1 to PC0 (IP 10.0.0.10), resulting in 0% loss. The third set is from PC0 to PC0, also resulting in 0% loss.

```
Pinging 20.0.0.10 with 32 bytes of data:  
Request timed out.  
Reply from 20.0.0.10: bytes=32 time=0ms TTL=127  
Reply from 20.0.0.10: bytes=32 time=0ms TTL=127  
Reply from 20.0.0.10: bytes=32 time=0ms TTL=127  
  
Ping statistics for 20.0.0.10:  
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 0ms, Maximum = 0ms, Average = 0ms  
  
PC>ping 20.0.0.10  
  
Pinging 20.0.0.10 with 32 bytes of data:  
  
Reply from 20.0.0.10: bytes=32 time=0ms TTL=127  
  
Ping statistics for 20.0.0.10:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 0ms, Maximum = 0ms, Average = 0ms  
  
PC>
```

Program 3

Aim: Configure default route, static route to the Router(Part 1).

Topology , Procedure and Observation:

<p>LAB-2</p> <p>Aim 2 configure IP address to Router in packet tracer</p> <p>Devices used : 2 Routers, 2 End devices</p> <p>Topology Routers and End devices</p>  <p>Procedure</p> <ul style="list-style-type: none"> * Select router R1 * Connect an end device PC1 to router R1 through parallel connection Ethernet 1/0 * Configure PC1 IP as 10.0.0.2 and gateway 10.0.0.1 * Similarly select router R2 and PC2 * Configure PC2 with IP address 20.0.0.2 and gateway 20.0.0.1 	<p>Now select Router R1 go to CLI and execute the following</p> <pre>Router > enable Router # configure terminal 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) exit</pre> <p>Similarly select router R2 go to CLI and execute the following</p> <pre>Router > enable Router # config terminal Router (config) # interface fastethernet 0/0 Router (config-if) # ip address 20.0.0.2 255.0.0.0 Router (config-if) # no shutdown</pre> <p>Hence the connection b/w routers and end devices has been established</p> <p>Now connect router R1 with R2 using serial cable</p> <p>Select R1 and go to CLI</p> <pre>Router (config) # interface serial 2/0 Router (config-if) # ip address 20.0.0.1 255.0.0.0 Router (config-if) # no shutdown</pre> <p>Select R2 and go to CLI</p> <pre>Router (config) # interface serial 3/0 Router (config-if) # ip address 20.0.0.2 255.0.0.0 Router (config-if) # no shutdown</pre>
--	---

Observations:

After setting up mentioned topology. Now try to ping PC2 from PC1 open command prompt and type ping 192.0.0.1

Destination host unreachable

ping 192.0.0.1 → successful.

packets sent is received by the router

hence although the routers were connected serially the end devices are unable to ping each other

~~bx
23/10/24~~

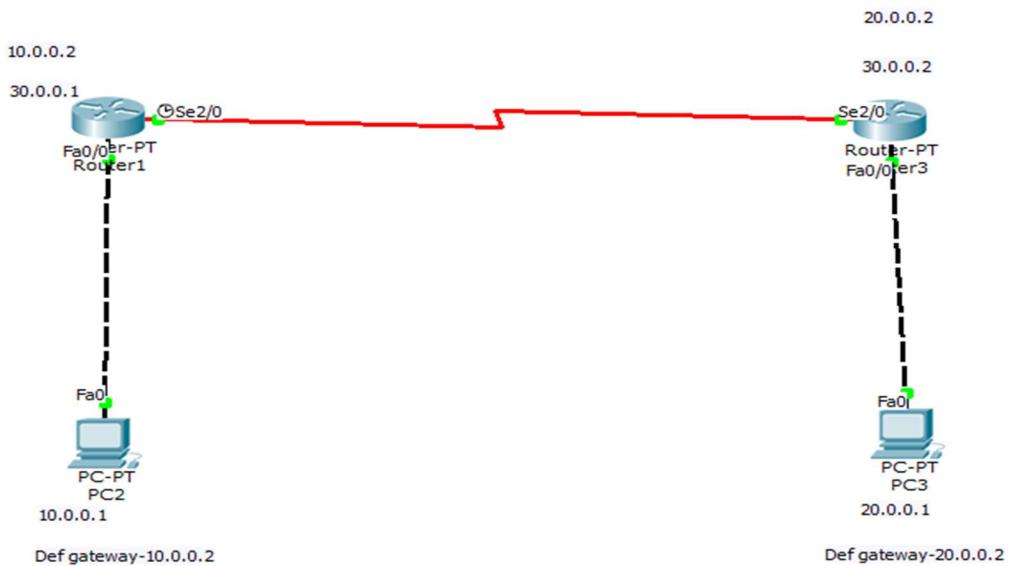
but two hosts will communicate with each other because needed config.

so now we have to change config.

configuration of bus is done
done config of 1st (pc1) host
configuration of 2nd (pc2) host
configuration of 3rd (routers) host

configuration of bus is done
done config of 1st (pc1) host
configuration of 2nd (pc2) host
configuration of 3rd (routers) host

Screen Shots:



PC2

Physical Config Desktop Custom Interface

Command Prompt X

```
Reply from 10.0.0.2: Destination host unreachable.  
Reply from 10.0.0.2: Destination host unreachable.  
Reply from 10.0.0.2: Destination host unreachable.  
  
Ping statistics for 20.0.0.1:  
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),  
  
PC>ping 20.0.0.1  
  
Pinging 20.0.0.1 with 32 bytes of data:  
  
Reply from 10.0.0.2: Destination host unreachable.  
  
Ping statistics for 20.0.0.1:  
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),  
  
PC>ping 20.0.0.2  
  
Pinging 20.0.0.2 with 32 bytes of data:  
  
Reply from 10.0.0.2: Destination host unreachable.  
  
Ping statistics for 20.0.0.2:  
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),  
  
PC>
```

Program 4

Aim: Configure default route, static route to the Router(Part 2).

Topology , Procedure and Observation:

LAB - 3

Date: 23/01/24
Page: 1

Aim : Configuration of default route, static route to the Router.

Devices used : 3 routers, 2 end devices

Topology:

Procedure:

- * Select End Device 1 (PC-1)
Config ip and set it to 10.0.0.10 and gateway to 10.0.0.1
- * Select End Device 2 (PC-2)
Config ip and set it to 40.0.0.10 and gateway to 40.0.0.1
- * Select Router 0, go to CLI and execute the following
Router > enable

```

Router # Configure terminal
Router (config) # interface fastethernet 0/0
Router (config-if) # ip address 10.0.0.1 255.0.0.0
Router (config-if) # no shutdown
Router (config-if) # exit
Router (config) # interface serial 2/0
Router (config) # ip address 20.0.0.1 255.0.0.0
Router (config) # no shutdown

```

* Select Router 2 and execute the following

```

Router > enable
Router # configure terminal
Router (config) # interface serial 2/0
Router (config) # ip address 20.0.0.2 255.0.0.0
Router (config-if) # no shutdown
Router (config-if) # interface serial 3/0
Router (config-if) # ip address 20.0.0.3 255.0.0.0
Router (config-if) # no shutdown
Router (config) # exit

```

* Select Router 3 and execute the following

```

Router > enable
Router # configure terminal
Router (config) # interface fastethernet 0/0
Router (config-if) # ip address 40.0.0.1 255.0.0.0
Router (config-if) # no shutdown
Router (config-if) # exit

```

Route (config) # interface serial 3/0
 Router (config-if) # ip address 30.0.0.2 255.0.0.0

IP Routes

* Router R0

```

Router # config terminal
Router (config) # ip route 0.0.0.0 0.0.0.0 20.0.0.2
Router # show ip routes
C 10.0.0.0/8 is directly connected
C 20.0.0.0/8 is directly connected
S 0.0.0.0/0 [1/0] via 20.0.0.2

```

* Router R1

```

Router # config terminal
Router (config) # ip route 10.0.0.10 255.0.0.0 20.0.0.1
Router # 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
C 30.0.0.8/8 is directly connected
S 40.0.0.0/8 [1/0] via 30.0.0.2

```

Observation:

After setting up the config topology,
PC1 is successful in pinging PC2

Ping to 40.0.0.10

Ping statistics for 40.0.0.10:

packets: sent=4 received=4 lost=0 (0%)
ping to 40.0.0.10 from 20.0.0.1 via en0, link layer

Open CLI of PC2

PC > ping 10.0.0.10

Pinging 10.0.0.10 with 32 bytes of data

Ping statistics for 10.0.0.10:

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

Hence, the connection between end devices is established successfully

IP Routes

Router - 2

Route # show ip route

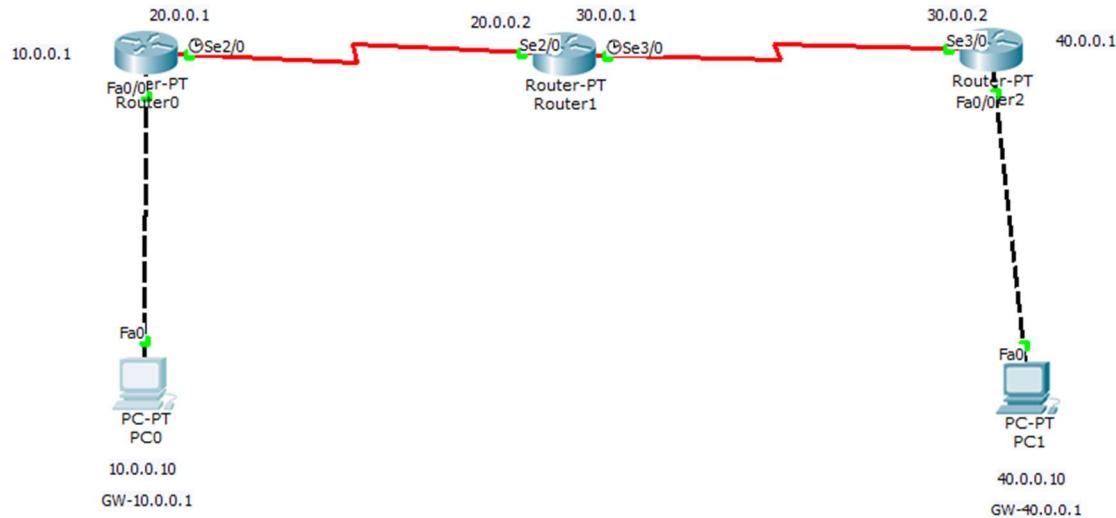
c 300.0.0/8 is directly connected

c 40.0.0.0/8 is directly connected

s 0.0.0/0 [1/0] via 30.0.0.1

85
23.10
23.0.0.0/8

Screen Shots:



```
PC> ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125
Reply from 40.0.0.10: bytes=32 time=5ms TTL=125

Ping statistics for 40.0.0.10:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 5ms, Maximum = 7ms, Average = 6ms

PC> ping 40.0.0.10

Pinging 40.0.0.10 with 32 bytes of data:

Reply from 40.0.0.10: bytes=32 time=8ms TTL=125
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125
Reply from 40.0.0.10: bytes=32 time=9ms TTL=125
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125

Ping statistics for 40.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 6ms, Maximum = 9ms, Average = 7ms
```

Program 5

Aim: Configure DHCP within a LAN and outside LAN.

Topology , Procedure and Observation:

LAB-4

Date: 13/11/24

Design of dynamic host configuration protocol using with the network and the other networks

Devices used
8 End devices, 1 Server, 2 switch, 1 Router

Topology

The diagram illustrates a network topology. At the top center is a Router. Two lines extend downwards from it to two separate clusters of network equipment. The left cluster contains a Switch labeled "Switch 1". Three lines connect this switch to four devices: an "End device" (represented by a square with a triangle), a PLC-PT device (represented by a square with a gear-like symbol), PC4 (represented by a square with a 'C' inside), and PC5 (represented by a square with a '5' inside). The right cluster contains another Switch labeled "Switch 2". Three lines connect this switch to three PCs: PC0 (square with '0'), PC1 (square with '1'), and PC2 (square with '2'). Additionally, a line connects the Router to the second Switch, and another line connects the second Switch to the Server (represented by a square with a 'S' inside).

Procedure :

Step 1: Place our generic servers, two switches and one router along with 6 end devices

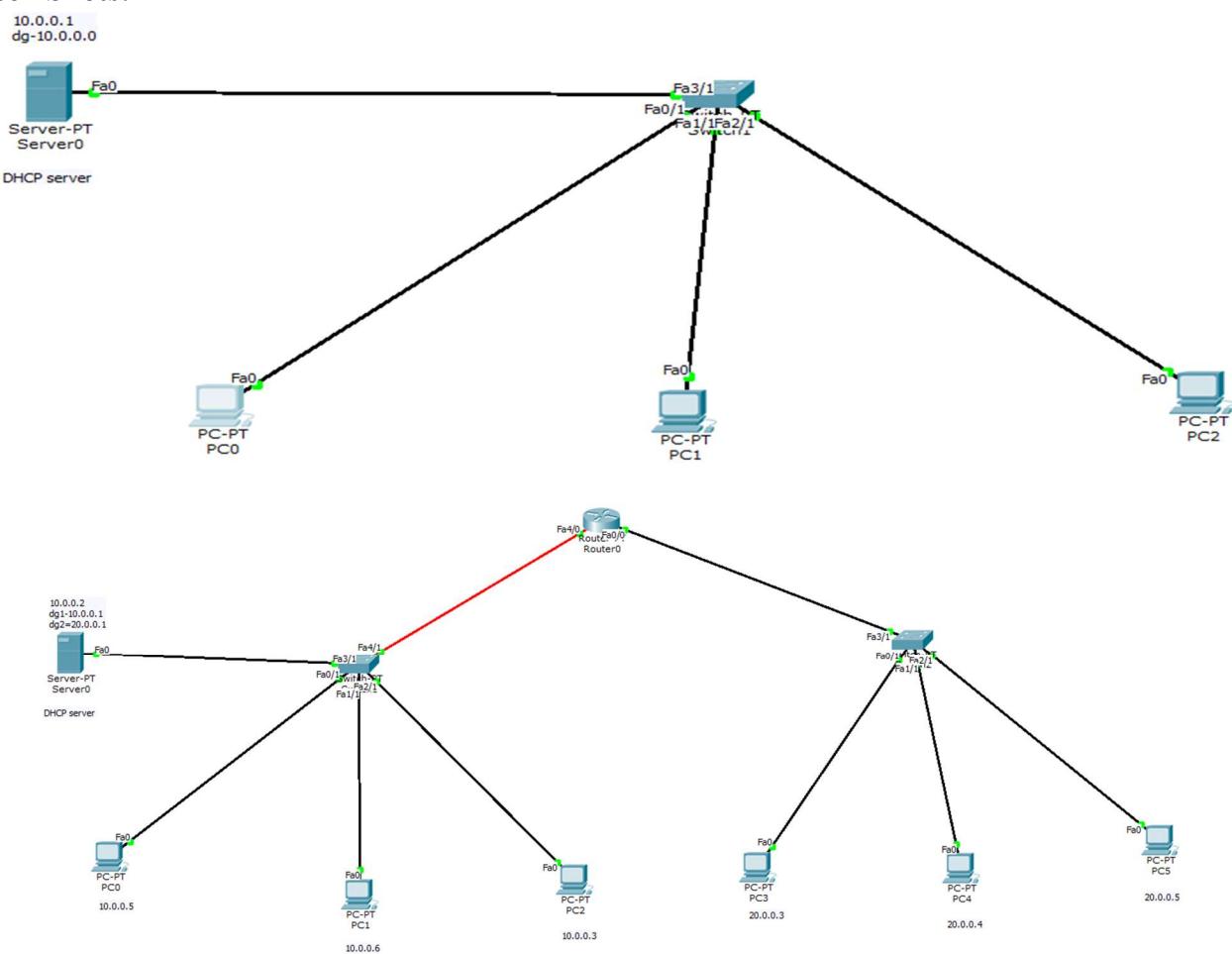
Step 2: Connect 3 PCs to switch 0 and 3 PCs to switch 1 using copper straight through

Step 3: Connect Server to switch 0 then connect switch 0 to router 0 and switch 1 to the route

Step 4: Set IP of server as 10.0.0.2 and default gateway as 10.0.0.1

<ul style="list-style-type: none"> * Configuring services for switch 0 <ul style="list-style-type: none"> Server → Services → DHCP Poolname → switch 0 Default gateway → 10.0.0.1 Start IP: 10.0.0.3 Mask number of use: 100 Add to the configuration * Configuring services for switch 1 <ul style="list-style-type: none"> Server → Services → DHCP Poolname → switch 1 Default gateway: 20.0.0.1 Start IP: 20.0.0.3 Mask no of use: 100 Add to the configuration * Route Configuration <ul style="list-style-type: none"> go to CLI, execute below for switch 0 <ul style="list-style-type: none"> > enable # config terminal # interface Fa1/0 # ip address 10.0.0.1 255.0.0.0 # ip helper address 10.0.0.2 # no shutdown # exit Config route for switch 1 <ul style="list-style-type: none"> # ip address 20.0.0.1 255.0.0.0 # ip helper address 20.0.0.2 	<p style="text-align: right;">Date _____ Page _____</p> <ul style="list-style-type: none"> * Now go to end system and select ip configuration, select DHCP. Service will automatically give ip to system * Repeat the same for other end systems <p><u>Observations:</u></p> <p>DHCP can be used to dynamically allocate IP addresses to end systems which are part of different networks.</p> <p style="text-align: center;"><i>SAC 20/1/24</i></p>
--	--

Screen Shots:



The screenshot shows a Cisco Packet Tracer interface with a window titled "Command Prompt". The window contains the following text output from a ping command:

```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:

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

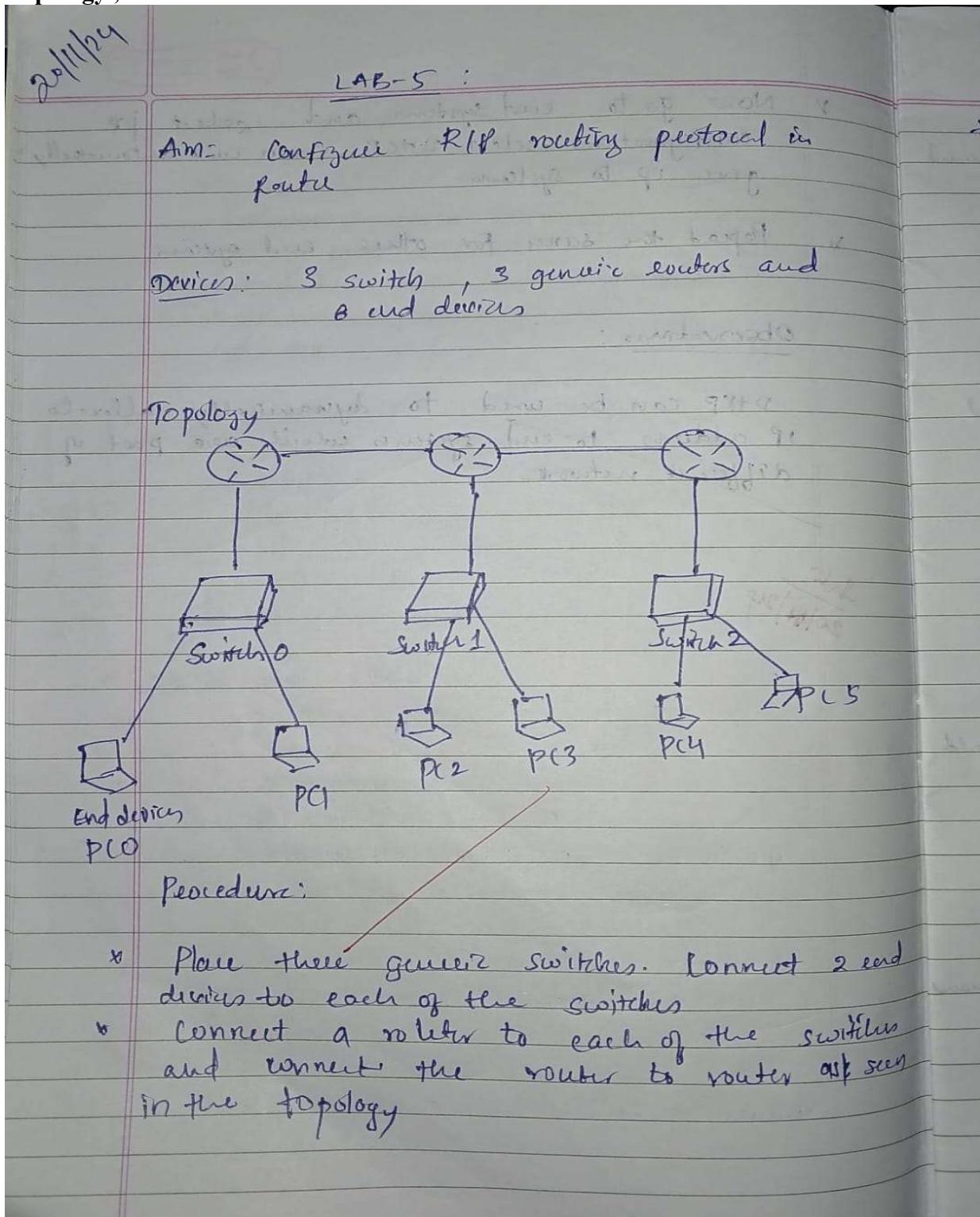
Ping statistics for 10.0.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

PC>
```

Program 6

Aim:Configure RIP routing Protocol in Routers .

Topology , Procedure and Observation:



* Set Configuration of PC0 and PC1
Set IP address of PC0 as 10.0.0.2
Set default gateway - 10.0.0.1

Set IP address of PC1 as 10.0.0.3
Set default gateway - 10.0.0.1

* Configure PC2 and PC3

Set IP of PC2 : 20.0.0.2
Default gateway : 20.0.0.1

Set IP of PC3 : 20.0.0.3

Default gateway : 20.0.0.1

* Configure PC4 and PC5

Set IP of PC4 : 30.0.0.2

Default gateway : 30.0.0.1

Set IP of PC5 : 30.0.0.3

Default gateway : 30.0.0.1

* Configure Router 0

Go to CLI and enter following command

enable

config terminal

interface fa 0/0

ip address 10.0.0.1 255.0.0.0

no shutdown

exit

Similarly configure router1 with 20.0.0.1 255.0.0.0
and router2 with 30.0.0.1 255.0.0.0

➤ Configure R0, R1

Go to CLI of R0 and enter following command

> enable

config terminal

interface Se2/0

ip address 40.0.0.1 255.0.0.0

no shut

exit

Go to CLI of R1 and enter following command

> enable

config terminal

interface Se2/0

ip address 40.0.0.2 255.0.0.0

no shut

exit

➤ Configure R1, R2

> enable

Go to CLI of R1 and enter following command

> enable

interface Se3/0

ip address 50.0.0.1 255.0.0.0

no shut

exit

Go to CLI of R2 and enter following command

config terminal

interface Se2/0

ip address 50.0.0.2 255.0.0.0

no shut

exit

Router RIP configuration

For Router 0

```
# config terminal  
# router rip  
# Network 10.0.0.0  
# Network 40.0.0.0
```

for Router 1

```
# config terminal  
# router rip  
# Network 20.0.0.0  
# Network 40.0.0.0  
# Network 50.0.0.0
```

for Router 2

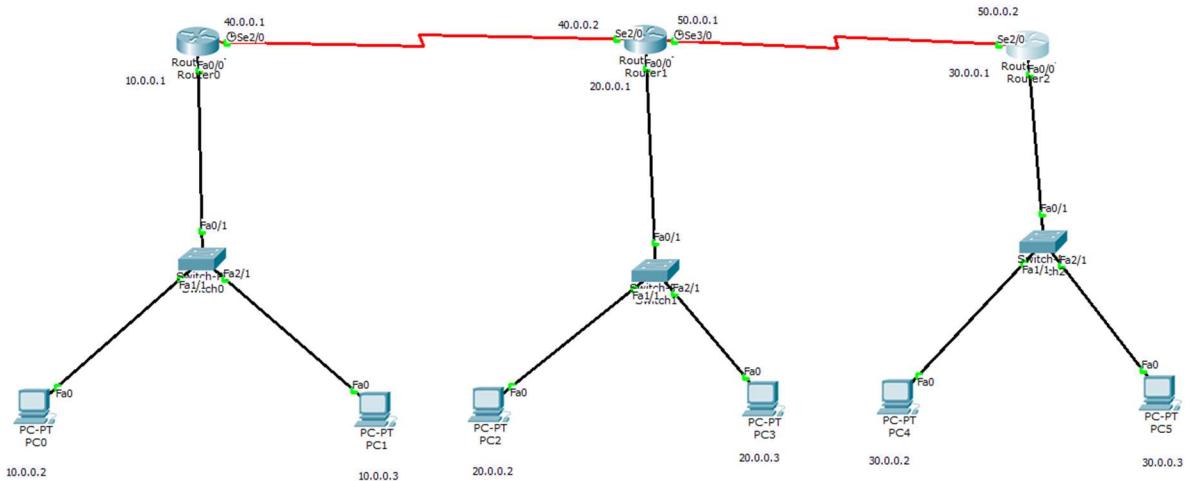
```
# config terminal  
# router rip  
# Network 50.0.0.0  
# Network 30.0.0.0
```

Observation:

After setting RIP routing protocol, we successfully ping between the end devices.

✓
20/11/24

Screen Shots:



PC0

Physical Config Desktop Custom Interface

Command Prompt

```

Pinging 30.0.0.2 with 32 bytes of data:

Request timed out.
Reply from 30.0.0.2: bytes=32 time=7ms TTL=125
Reply from 30.0.0.2: bytes=32 time=6ms TTL=125
Reply from 30.0.0.2: bytes=32 time=7ms TTL=125

Ping statistics for 30.0.0.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 6ms, Maximum = 7ms, Average = 6ms

PC>ping 30.0.0.2

Pinging 30.0.0.2 with 32 bytes of data:

Reply from 30.0.0.2: bytes=32 time=4ms TTL=125
Reply from 30.0.0.2: bytes=32 time=7ms TTL=125
Reply from 30.0.0.2: bytes=32 time=7ms TTL=125
Reply from 30.0.0.2: bytes=32 time=7ms TTL=125

Ping statistics for 30.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 4ms, Maximum = 7ms, Average = 6ms

PC>
    
```

Program 7

Aim:Demonstrate the TTL/ Life of a Packet .

Topology , Procedure and Observation:

Screen Shots:

PDU Information at Device: Router0

OSI Model Inbound PDU Details Outbound PDU Details

At Device: Router0
Source: PC0
Destination: PC3

In Layers

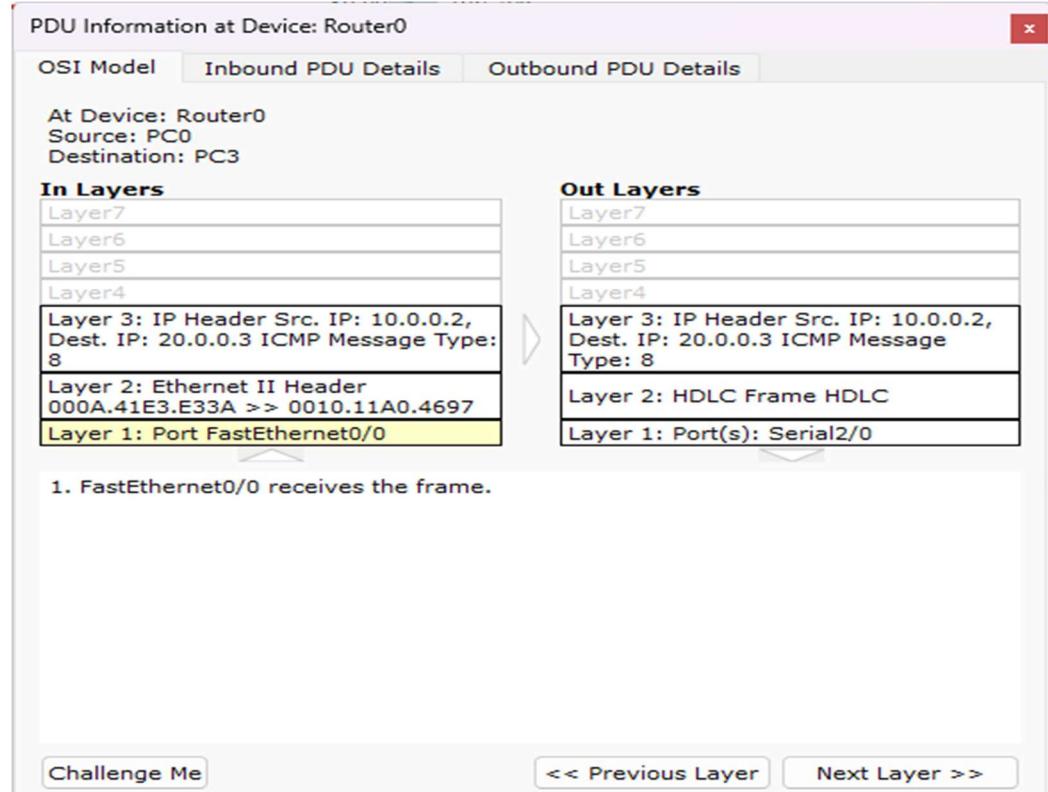
Layer7
Layer6
Layer5
Layer4
Layer 3: IP Header Src. IP: 10.0.0.2, Dest. IP: 20.0.0.3 ICMP Message Type: 8
Layer 2: Ethernet II Header 000A.41E3.E33A >> 0010.11A0.4697
Layer 1: Port FastEthernet0/0

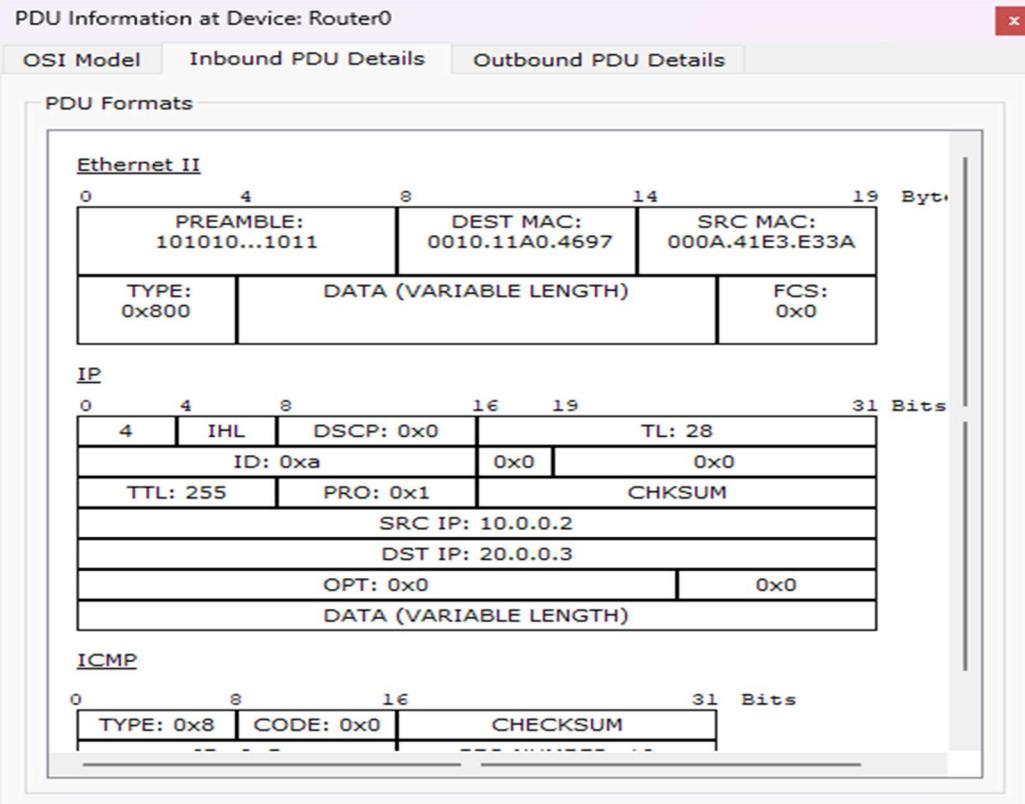
Out Layers

Layer7
Layer6
Layer5
Layer4
Layer 3: IP Header Src. IP: 10.0.0.2, Dest. IP: 20.0.0.3 ICMP Message Type: 8
Layer 2: HDLC Frame HDLC
Layer 1: Port(s): Serial2/0

1. FastEthernet0/0 receives the frame.

Challenge Me << Previous Layer Next Layer >>





PDU Information at Device: Router0

OSI Model Inbound PDU Details Outbound PDU Details

PDU Formats

HDLC

0	8	16	32	32+x	48+x	56+:
FLG: 0111 1110	ADR: 0x8f	CONTROL: 0x0	DATA: (VARIABLE LENGTH)	FCS: 0x0	FLG: 0111 1110	

IP

0	4	8	16	19	31 Bits
	4	IHL	DSCP: 0x0	TL: 28	
			ID: 0xa	0x0	0x0
		TTL: 254	PRO: 0x1	CHKSUM	
			SRC IP: 10.0.0.2		
			DST IP: 20.0.0.3		
			OPT: 0x0	0x0	
			DATA (VARIABLE LENGTH)		

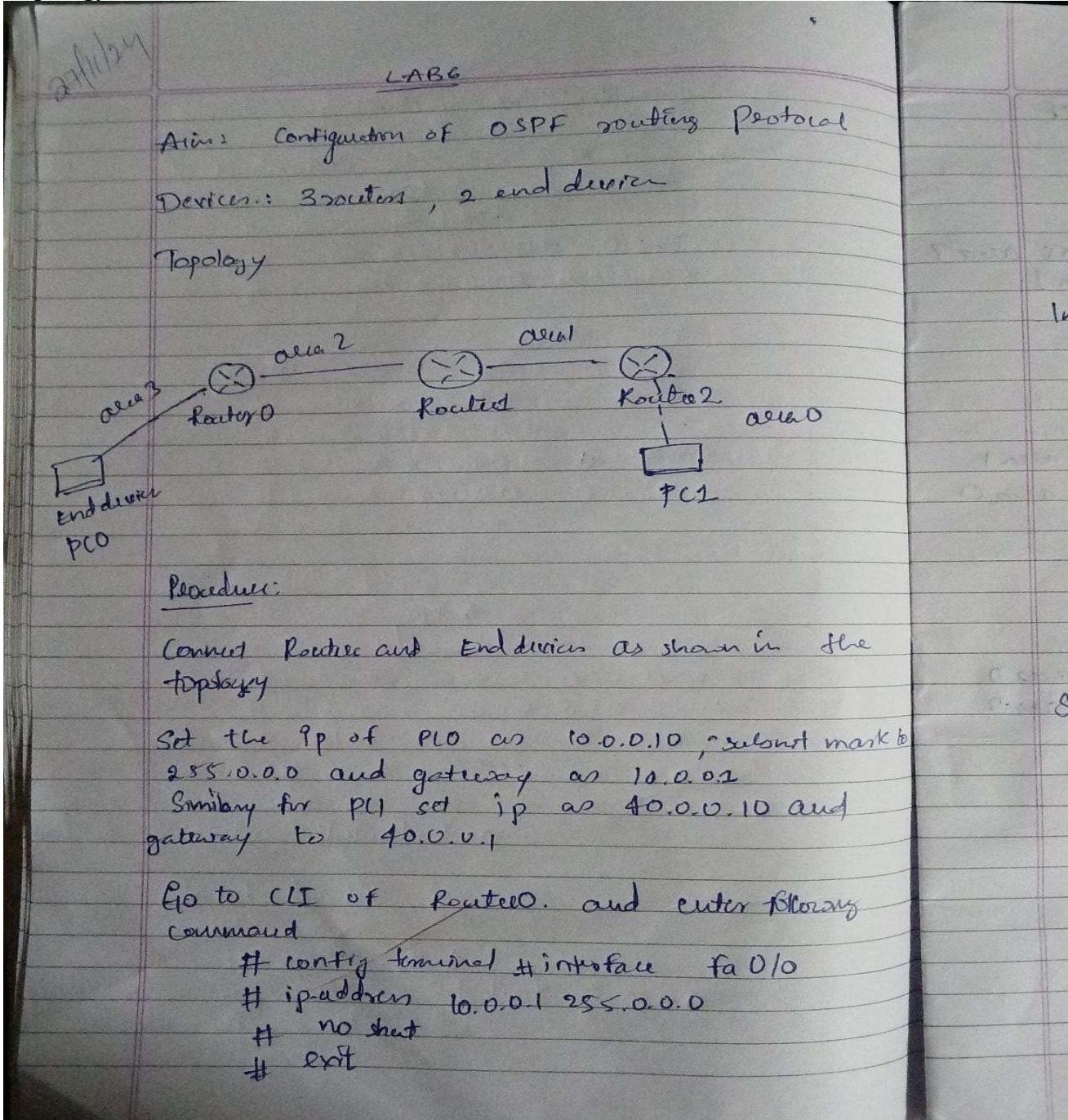
ICMP

0	8	16	31 Bits
TYPE: 0x8	CODE: 0x0	CHECKSUM	
ID: 0x5	SEQ NUMBER: 10		

Program 8

Aim: Configure OSPF routing protocol .

Topology , Procedure and Observation:



```
# interface se1/0  
# ip address 20.0.0.1 255.0.0.0  
# encapsulation ppp  
# clock rate 64000  
# no shutdown  
# exit
```

In Router R2

```
# interface se1/0  
# ip address 20.0.0.2 255.0.0.0  
# encapsulation ppp  
# no shutdown  
# exit
```

```
# interface se2/0  
# ip address 30.0.0.1 255.0.0.0  
# encapsulation ppp  
# clock rate 64000  
# exit
```

Similarly in R3

To

```
# interface se1/0  
# ip address 30.0.0.2 255.0.0.0  
# encapsulation ppp  
# no shutdown  
# exit
```

```
# interface fa2/0  
# ip address 40.0.0.1 255.0.0.0  
# no shutdown  
# exit
```

* Enabling ip routing by configuring OSPF

In Router R1

```
# router ospf 1  
# router-id 1.1.1.1  
# network 10.0.0.0 0.255.255.255 area 0  
# network 20.0.0.0 0.255.255.255 area 0  
# exit
```

In Router R2

```
# router ospf 1  
# router-id 2.2.2.2  
# network 10.0.0.0 0.255.255.255 area 0  
# network 30.0.0.0 0.255.255.255 area 0  
# exit
```

In Router R3

```
# router ospf 1  
# router-id 3.3.3.3  
# network 30.0.0.0 0.255.255.255 area 0  
# network 40.0.0.0 0.255.255.255 area 0  
# exit
```

Q) Creating Virtual-Link

In Router R1

```
#1  
# router ospf 1  
# area 0 virtual-link 2.2.2.2
```

In Router R2

```
# area 1 virtual-link 3.3.3.3  
# exit
```

for Router R3

show ip route

```
IA 20.0.0.0 /8  
IA 20.0.0.0 /8  
30.0.0.0 /8  
30.0.0.0 /8  
40.0.0.0 /8  
172.16.0.0/16
```

Observation:

Telnet can successfully ping P1 from P10

Loopback

Router 1 :

```
# interface loopback 0  
# ip add 172.16.1.252 255.255.0.0  
# no shut  
# exit
```

Router 2

```
# interface loopback 0  
# ip add 172.16.1.253 255.255.0.0
```

Router 3 :

```
# interface loopback 0  
# ip add 172.16.1.252 255.255.0.0  
# no sh  
# end
```

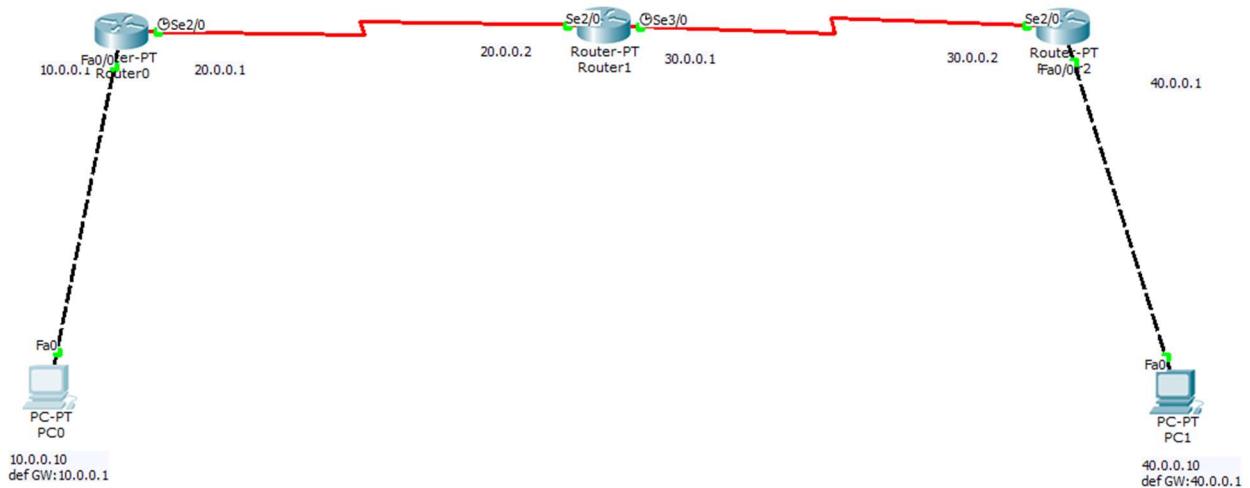
Observation : All connections are successful

```
ping 40.0.0.10  
reply from 40.0.0.10 bytes=32 time=7ms TTL=64  
reply from 40.0.0.10 bytes=32 time=7ms TTL=64  
reply from 40.0.0.10 bytes=32 time=7ms TTL=128  
reply from 40.0.0.10 bytes=32 time=7ms TTL=128
```

Ring Statistics

sent = 4 , Received = 4 , Loss = 0 (0 % loss)

Screen Shots:



PC0

Physical Config Desktop Custom Interface

Command Prompt

```
Pinging 40.0.0.10 with 32 bytes of data:  
  
Request timed out.  
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125  
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125  
Reply from 40.0.0.10: bytes=32 time=8ms TTL=125  
  
Ping statistics for 40.0.0.10:  
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 7ms, Maximum = 8ms, Average = 7ms  
  
PC>ping 40.0.0.10  
  
Pinging 40.0.0.10 with 32 bytes of data:  
  
Reply from 40.0.0.10: bytes=32 time=9ms TTL=125  
Reply from 40.0.0.10: bytes=32 time=7ms TTL=125  
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125  
Reply from 40.0.0.10: bytes=32 time=6ms TTL=125  
  
Ping statistics for 40.0.0.10:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 6ms, Maximum = 9ms, Average = 7ms  
  
PC>
```

Program 9

Aim: Configure Web Server, DNS within a LAN.

Topology , Procedure and Observation:

LAB - 8

Objective
Configure Web server, DNS within a LAN

Topology

The diagram illustrates a network topology with three main components: a Server, a Switch, and a PC. The Server is labeled "Server-PT" with IP "10.0.0.2" and has an interface "fa0". It is connected to the Switch via "fa1/1" and to the PC via "fa0". The Switch is labeled "Switch-PT" and "Switch 0" and has an interface "fa0/0". The PC is labeled "PC-PT" with IP "10.0.0.1" and has an interface "fa0". All connections are represented by straight lines.

Procedure :

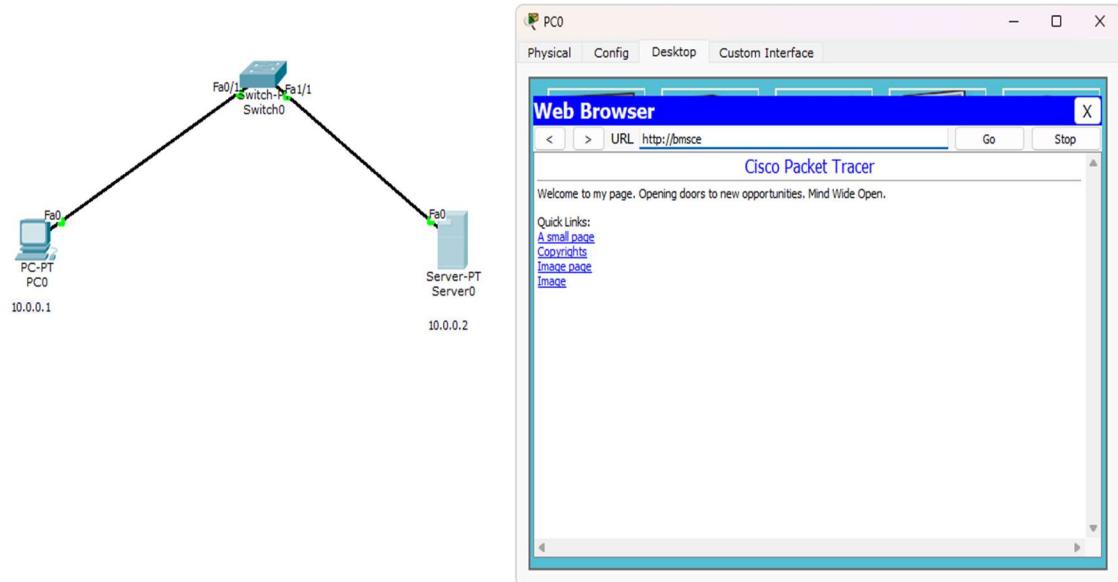
- * Place all end device , a server and a switch . Connect them using copper straight lines
- * Assign IP addresses as demonstrated in topology
- * To set IP of server
 - go to config
 - select DNS method , turn it on & add a resource
 - set IP , make sure port is on
 - Select HTTP
 - Turn on the services to on
 - Select PC0 → Desktop → Help browser
 - Enter URL specified in DNS resources

Observation:

- The server's web page could be successfully accessed from the PC by entering the address **VR2**.
- The DNS server could hence be configured within a LAN by enabling the DNS.

30/12/2021

Screen Shots:



Program 10

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

Topology , Procedure and Observation:

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

Topology:

* Procedure:

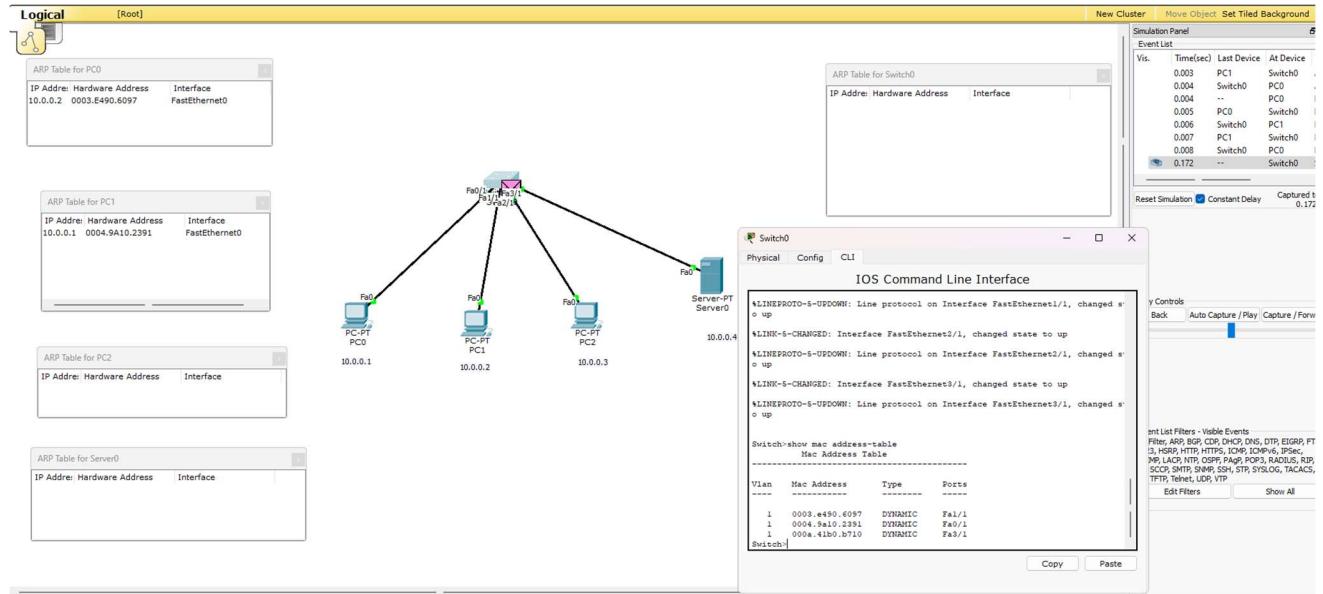
- Place three end devices, a server and switches, connect the PCs and the server to the ARP table switch using copper straight wires
- Use prospect tool to click on PC to view the ARP table
- The same can also be viewed in command prompt by using 'arp -a'.
- Goto CLI of switch and do show mac address
- Similarly obtain ARP table of server and other end devices
- Enter the simulation mode and click on capture by selecting PCI and PC2 for simple PDU

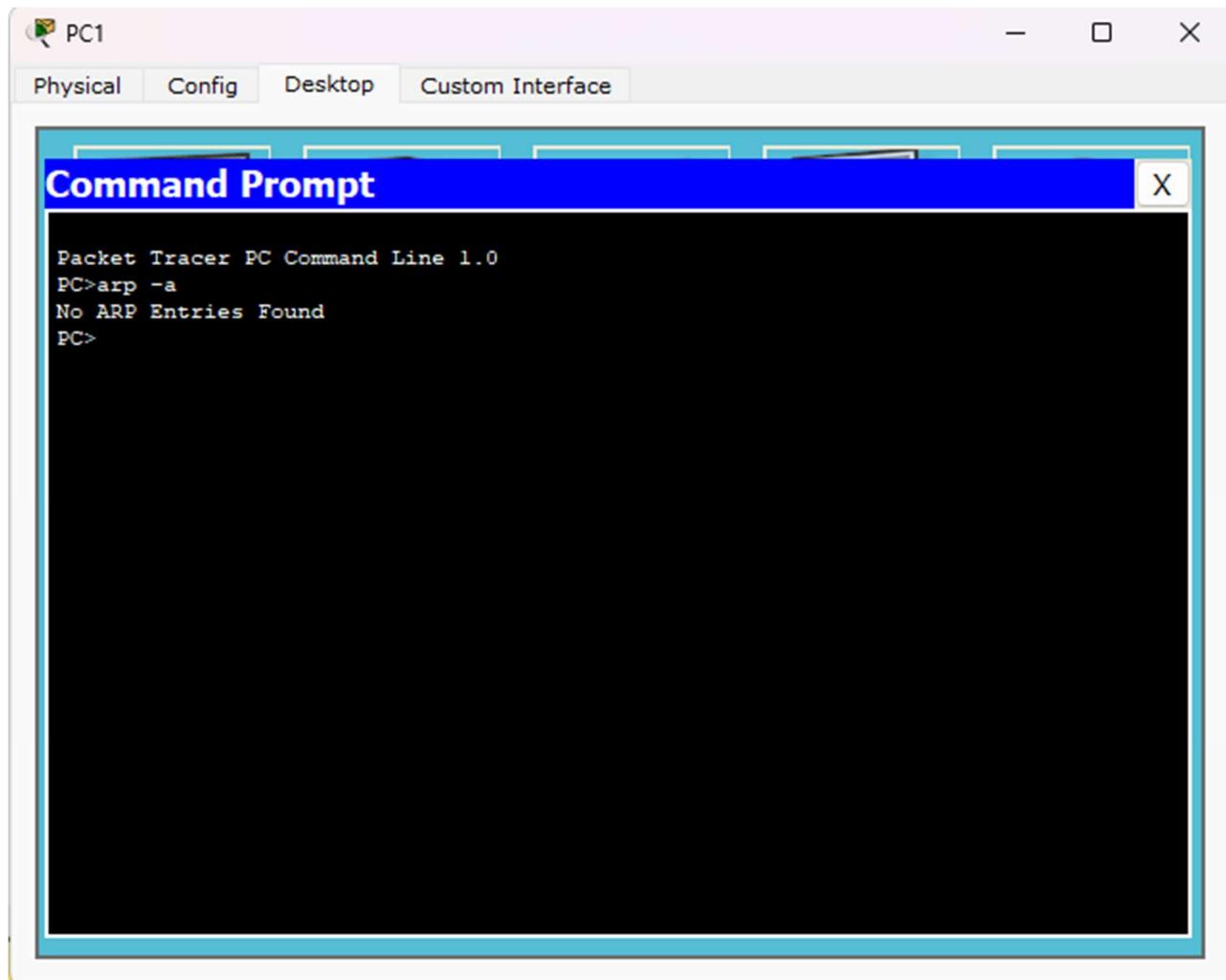
Observations:

- Initially, the ARP tables of all end devices are observed to be empty
- The MAC address table is also found to be empty
- When capture button is clicked, it is found that the ARP table is updated in PC2 with IP address of PC1 (10.0.0.2)
- Once the acknowledgement is obtained, the ARP table of PC1 is updated with IP address of PC2 (10.0.0.3)
- The Event list in the simulation panel shows the corresponding protocol used during communication

~~AK
30/12/2021~~

Screen Shots:





Program 11

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

Topology , Procedure and Observation:

3)

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

Topology

```
graph TD; Router[Router - PT] --- PC[PC - PT]; PC --- PC_IP["IP: 10.0.0.2"]; PC_IP --- Default["Default gateway: 10.0.0.2"];
```

Procedure : Place an end device and a router and connect them using a copper-crossover wire and assign the IP addresses shown in the topology.

In RouterD, do

```
> enable
# config terminal
# hostname R1
# enable secret P1
# interface fa0/0
# ip address 10.0.0.1 255.0.0.0
# no shut
# line vty 0 5
# login
# password P0
# exit
```

In PCO, do

ping 10.0.0.1

Now, to access the router CLI from PCO, do:

telnet 10.0.0.2

User Access Verification

password: p0

>enable

password: p1

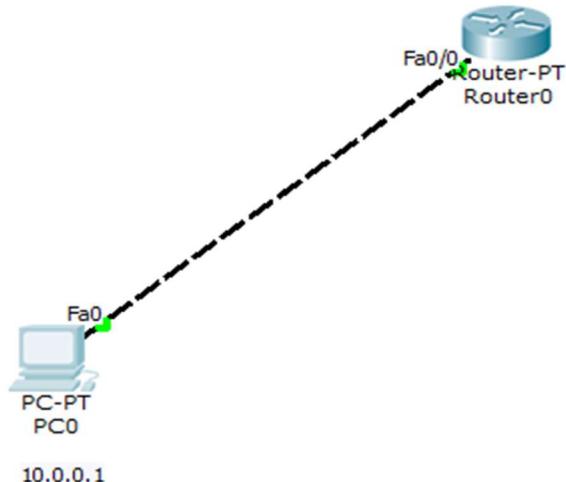
show ip route

Observation :

- * Two passwords are given while configuring the router, one being the secret key for the router and other being the password for login and commanding access.
- * The passwords entered in the router are used in reverse order here, i.e., the user has to login first to verify access via p0 and then obtain router access with secret key p1.
- * Hence, the admin in PC is able to run commands as run in router CLI and see results from PC.

30/V/79

Screen Shots:



Command Prompt

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

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time=0ms TTL=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#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
      * - candidate default, U - per-user static route, o - ODR
      P - periodic downloaded static route

Gateway of last resort is not set

C    10.0.0.0/8 is directly connected, FastEthernet0/0
R1#|
```

Program 12

Aim: To construct a VLAN and make the PC's communicate among a VLAN .

Topology , Procedure and Observation:

41.

Objective :
To construct a VLAN and make the PCs communication among a VLAN

Topology :

```
graph TD; Router[Router] --- Switch[Switch-PT]; Router --- IP1[192.168.1.1]; Router --- IP2[192.168.2.1]; Switch --- PC1[PC-PT PCL<br/>192.168.1.2]; Switch --- PC2[PC-PT PCI<br/>192.168.1.3]; Switch --- PC3[PC-PT PC2<br/>192.168.2.2]; Switch --- PC4[PC-PT PC3<br/>192.168.2.3]
```

Procedure:

- * Place 4 end devices , a switch and a router & connect end devices to switch end . The switch to the router using copper straight wires .
- * Assign IP address to end device as displayed in the topology
- * In Router do:-
 - > enable
 - # config terminal
 - # interface fa0/0
 - # ip address 192.168.1.1 255.255.255.0

```
# vlan database
# vlan 2 name cse101
# exit
# config terminal
# interface tengig Fa0/0/1
# encapsulation dot1q 2
# ip address 192.168.2.1 255.255.255.0
# no shutdown
# exit
```

In switch O, do:

choose VLAN database

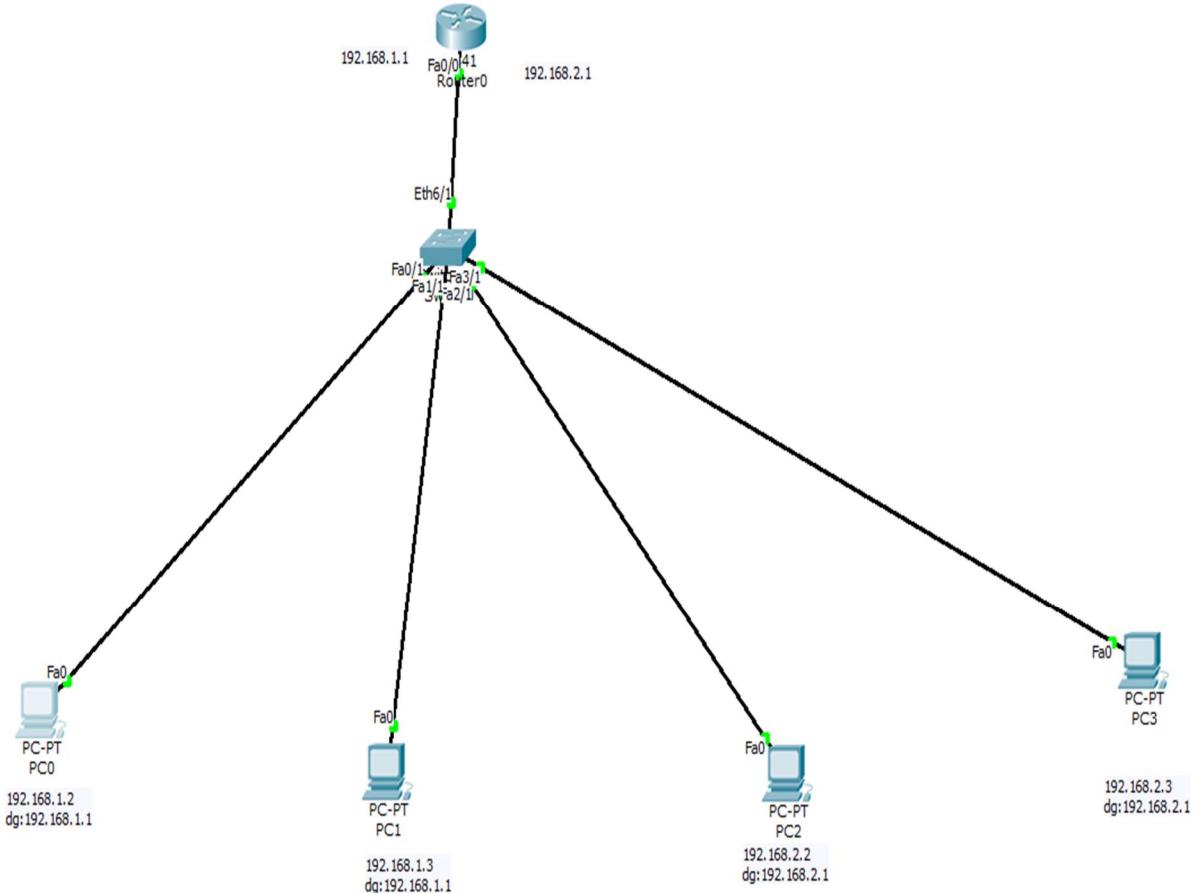
Turn port, status on for corresponding ethernet
enable trunk

Observation:

- * Proper trunk configuration is enabled to make VLAN work properly
- * VLAN trunking allows switches to forward frames from different VLANs over a single unit called trunk
- * ping messages from different PCs are observed to be working successfully henceworks

~~30/12/04~~

Screen Shots:



Command Prompt

```
Packet Tracer PC Command Line 1.0
PC>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Request timed out.
Reply from 192.168.2.2: bytes=32 time=0ms TTL=127
Reply from 192.168.2.2: bytes=32 time=0ms TTL=127
Reply from 192.168.2.2: bytes=32 time=4ms TTL=127

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 4ms, Average = 1ms

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=0ms TTL=127
Reply from 192.168.2.2: bytes=32 time=0ms TTL=127
Reply from 192.168.2.2: bytes=32 time=2ms TTL=127
Reply from 192.168.2.2: bytes=32 time=0ms TTL=127

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

PC>ping 192.168.2.3

Pinging 192.168.2.3 with 32 bytes of data:

Request timed out.
Reply from 192.168.2.3: bytes=32 time=3ms TTL=127
Reply from 192.168.2.3: bytes=32 time=2ms TTL=127
Reply from 192.168.2.3: bytes=32 time=1ms TTL=127

Ping statistics for 192.168.2.3:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 3ms, Average = 2ms

PC>ping 192.168.2.3

Pinging 192.168.2.3 with 32 bytes of data:

Reply from 192.168.2.3: bytes=32 time=0ms TTL=127
Reply from 192.168.2.3: bytes=32 time=0ms TTL=127
Reply from 192.168.2.3: bytes=32 time=2ms TTL=127
Reply from 192.168.2.3: bytes=32 time=0ms TTL=127

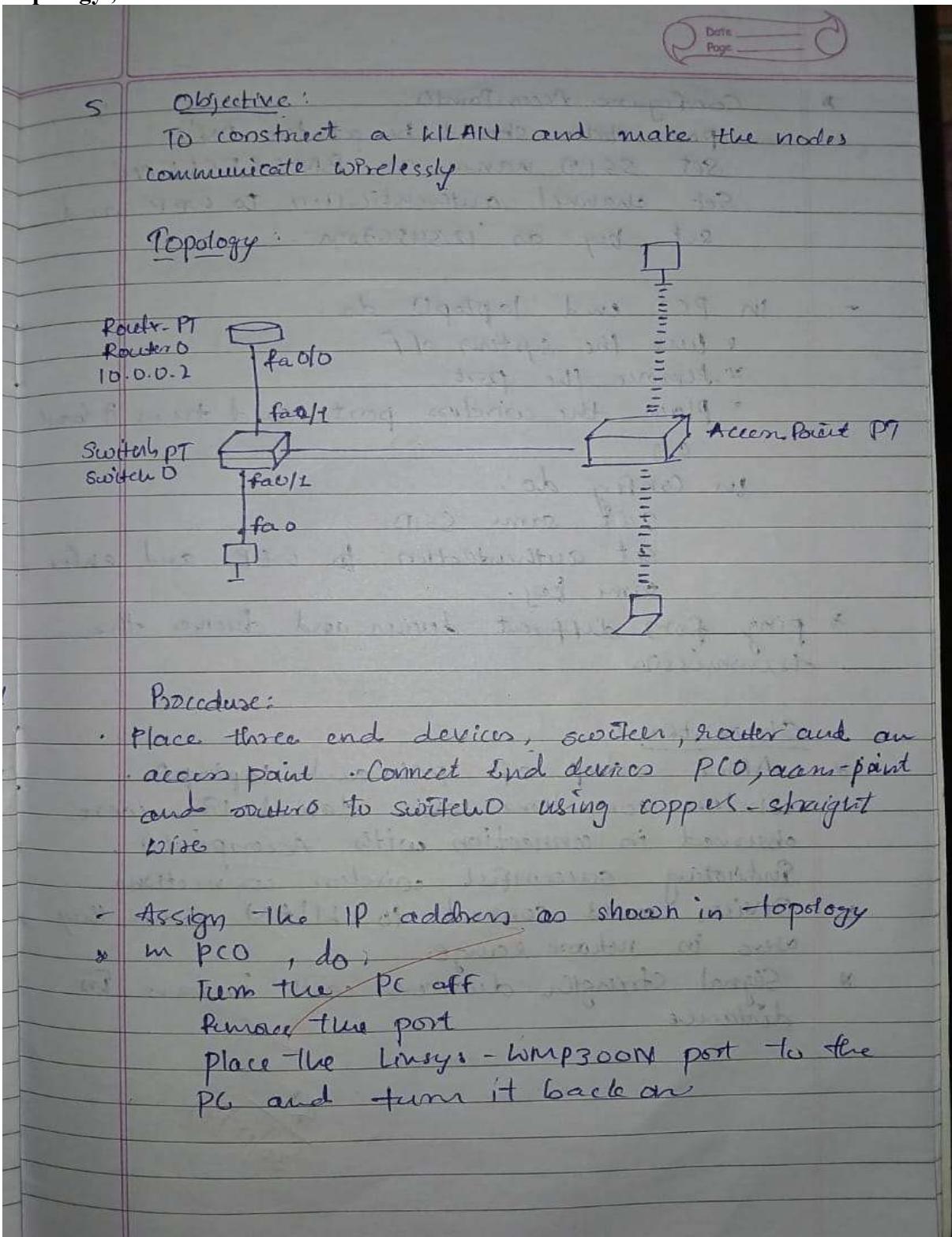
Ping statistics for 192.168.2.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms

PC>
```

Program 13

Aim: To construct a WLAN and make the nodes communicate wirelessly.

Topology , Procedure and Observation:



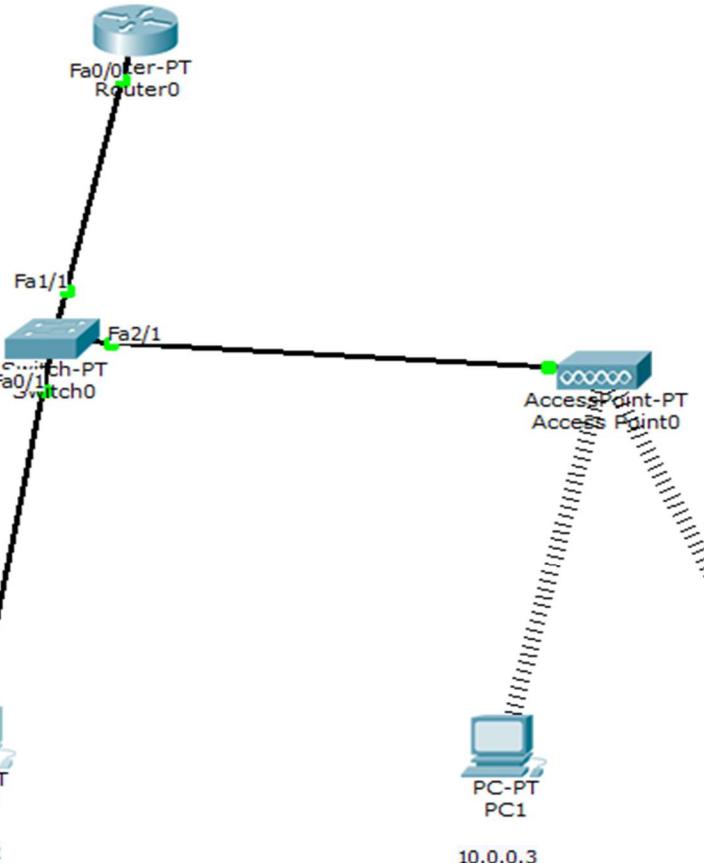
- * Configure AccessPoint0
 - Port status should be set for 'ON'
 - Set SSID name as 'BMSCECSECN'
 - Set channel authentication to WEP and set key as '1234567890'
- In PCI and laptop0 do :
 - * turn the system off
 - * remove the port
 - * Place the wireless port and turn it back on
- In config do :
 - Set same SSID
 - Set authentication to WEP and enter same key.
- * ping from different devices and observe the transmission

Observations

- * After setup of PCI and laptop0, wireless connections with dashed line were observed in connection with AccessPoint0, indicating successful wireless connections.
- * Devices could connect to WLAN since they were in network range
- * Signal strength decreases with increase in distance

25
30/12/2019

Screen Shots:



A screenshot of the "Command Prompt" window from Packet Tracer. The window title is "PC0" and the tab selected is "Physical". The command line shows the user performing pings to three different hosts.

```
Packet Tracer PC Command Line 1.0
PC>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:
Reply from 10.0.0.3: bytes=32 time=22ms TTL=128
Reply from 10.0.0.3: bytes=32 time=6ms TTL=128
Reply from 10.0.0.3: bytes=32 time=3ms TTL=128
Reply from 10.0.0.3: bytes=32 time=7ms TTL=128

Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 3ms, Maximum = 22ms, Average = 9ms

PC>ping 10.0.0.4

Pinging 10.0.0.4 with 32 bytes of data:
Reply from 10.0.0.4: bytes=32 time=19ms TTL=128
Reply from 10.0.0.4: bytes=32 time=5ms TTL=128
Reply from 10.0.0.4: bytes=32 time=6ms TTL=128
Reply from 10.0.0.4: bytes=32 time=7ms TTL=128

Ping statistics for 10.0.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 5ms, Maximum = 19ms, Average = 9ms

PC>
```

PART-B

Program 14

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

Code and Output:

Part-B :

```
#include <iostream>
#include <string.h>
using namespace std;

Pnt crc (char *ip, char *op, char *poly, int mode)
{
    strcpy(op, ip);
    if (mode)
        for (int i=1; i< strlen(poly); i++)
            strcat(op, "0");

    for (Pnt r=0; r< strlen(op); i++)
        if (op[i] == '1')
            for (int j=0; j< strlen(poly); j++)
                if (op[i+j] == poly[j])
                    op[i+j] = '0';
                else
                    op[i+j] = '1';

    for (int i=0; i< strlen(op); i++)
        if (op[i] == '1')
            return 1;
    return 0;
}
```

```

int main() {
    char ip[50], op[50], recv[50];
    char poly[] = "1000100000100001";
    cin >> pp;
    crc(ip, op, poly, 1);
    cout << op << endl;
    cin >> recv;
    if (crc(recv, op, poly, 0))
        cout << "No error" << endl;
    else
        cout << "Error in trans." << endl;
    return 0;
}

```

Output:

Enter input message in binary:

111101

~~The transmitted message is 111101101011100111010~~

Enter the received message in binary

111101

No Error in data

~~85
27/25~~

Program 15

Write a program for congestion control using Leaky bucket algorithm.

Code and Output:

Date _____
Page _____

white programs for congestion control using
Leaky bucket algorithm

```
#include <iostream>
#include <string.h>
#include <stdio.h>
using namespace std;
#include <stdlib.h>
#include <vector.h>
#define NOF_PACKETS 10
int rand (int a) {
    int m = (random() % 10) % a;
    return m == 0 ? 1 : m;
}
int main() {
    int pack_sz[NOF], i, clk, b_size, d_rate,
        p_sz_m=0, p_sz_ip_time, up;
    for (int i=0; i<NOF; ++i)
        printf("Input packet[%d]: %d bytes\n", i, pack_sz[i]);
    packet_sz[i] = rand(6)*10;
    for (i=0; i<NOF; i++) {
        cout << "packet[" << i << "] : " << d_rate << endl;
        cout << "Enter output rate: ";
        cin >> d_rate;
        cout << "Enter bucket size: ";
        cin >> b_size;
        for (i=0; i<NOF; i++) {
            if ((packet_sz[i] + p_sz_m) > b_size)
                if (packet_sz[i] > b_size)
```

printf("Priority is greater than
Bucket capacity")

else else

printf("Bucket capacity exceeded -
packet reflected");

else

p_sz_rm += packet_sz[i];

printf("In. packet size /d", packet_sz[i]);

printf("Bytes remaining /d", p_sz_rm);

p_time = rand(4)*10

printf("Time left for trans. /d units", p_time);

for (clk=10; clk <= p_time; clk=10)

q

sleep();

if (p_sz_rm)

if (p_sz_rm <= 0.5 * clk)

Op = p_sz_rm, p_sz_rm = 0;

else

Op = 0.5 * clk, p_sz_rm = 0.5 * clk;

printf("packet size /d trans", op);

printf("rem. bytes /d", p_sz_rm);

else

printf("Time rem. /d p_time - clk");

printf("no pkts to transmit")

3. Using TCP/IP sockets, write client-server program

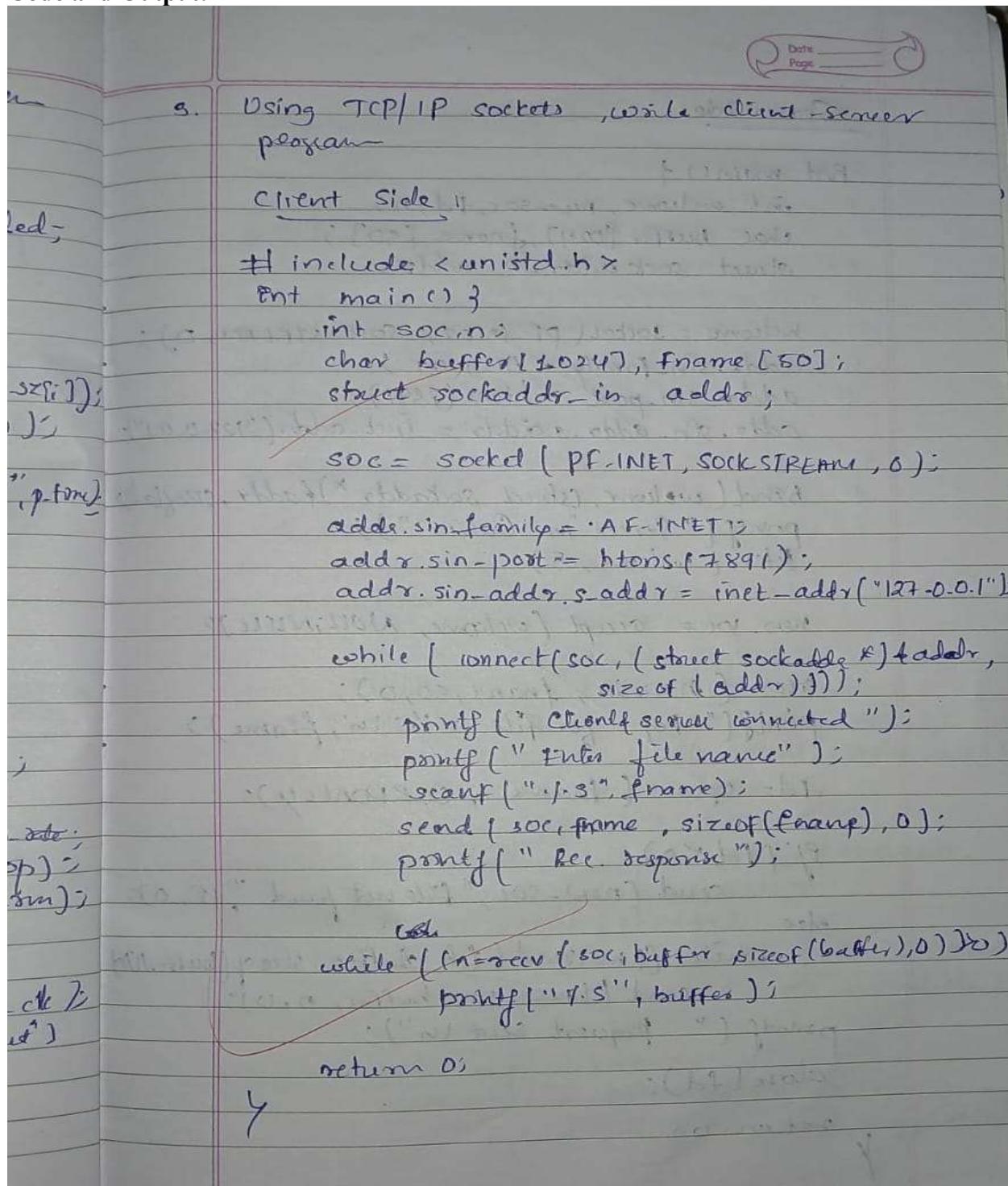
Client Side

```
#include <unistd.h>
int main()
{
    int soc, n;
    char buffer[1024], fname[50];
    struct sockaddr_in addr;
    soc = socket(PF_INET, SOCK_STREAM, 0);
    addr.sin_family = AF_INET;
    addr.sin_port = htons(7891);
    addr.sin_addr.s_addr = inet_addr("127.0.0.1");
    while (connect(soc, (struct sockaddr*)&addr,
                   sizeof(addr))) {
        printf("Client server connected");
        printf("Enter file name");
        scanf("./%s", fname);
        send(soc, fname, sizeof(fname), 0);
        printf("Rec. response");
        while ((n = recv(soc, buffer, sizeof(buffer), 0)) > 0)
            printf("%c", buffer);
    }
    return 0;
}
```

Program 16

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

Code and Output:



Server Side:

int main()

int welcome, new_soc, fd, n;

char buffer[1024], fname[50];
struct sockaddr_in addr;

welcome = socket(PF_INET, SOCK_STREAM, 0);

addr.sin_family = AF_INET;

addr.sin_port = htons(7891);

addr.sin_addr.s_addr = Inet_addr("127.0.0.1");

bind(welcome, (struct sockaddr *) &addr, sizeof(addr));

printf("Server in Online");

listen(welcome, 5);

new_soc = accept(welcome, NULL, NULL);

recv(&new_soc, fname, 50, 0);

printf("Req. for file : %s\n", fname);

fd = open(fname, O_RDONLY);

if(fd < 0)

send(new_soc, "file not found ", 15, 0);

else

while((n = read(fd, buffer, sizeof(buffer)))

send(&new_soc, buffer, n, 0);

printf("Request sent\n");

close(fd);

return 0;

Output:

Server is online

Requesting for file: test.txt

Request sent

Client is connected to server

Enter file name: test.txt

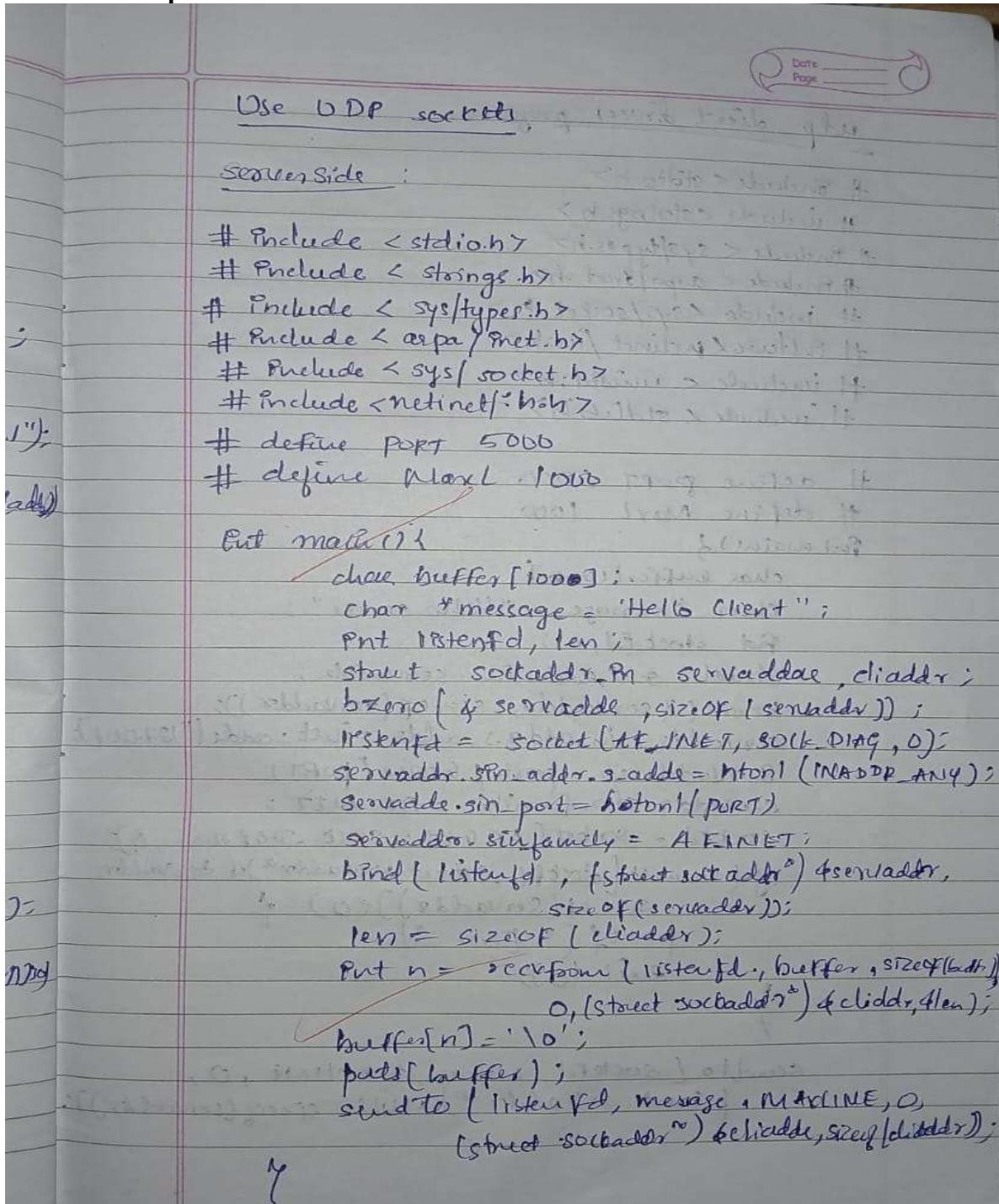
Received response

Hello World

Program 17

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

Code and Output:



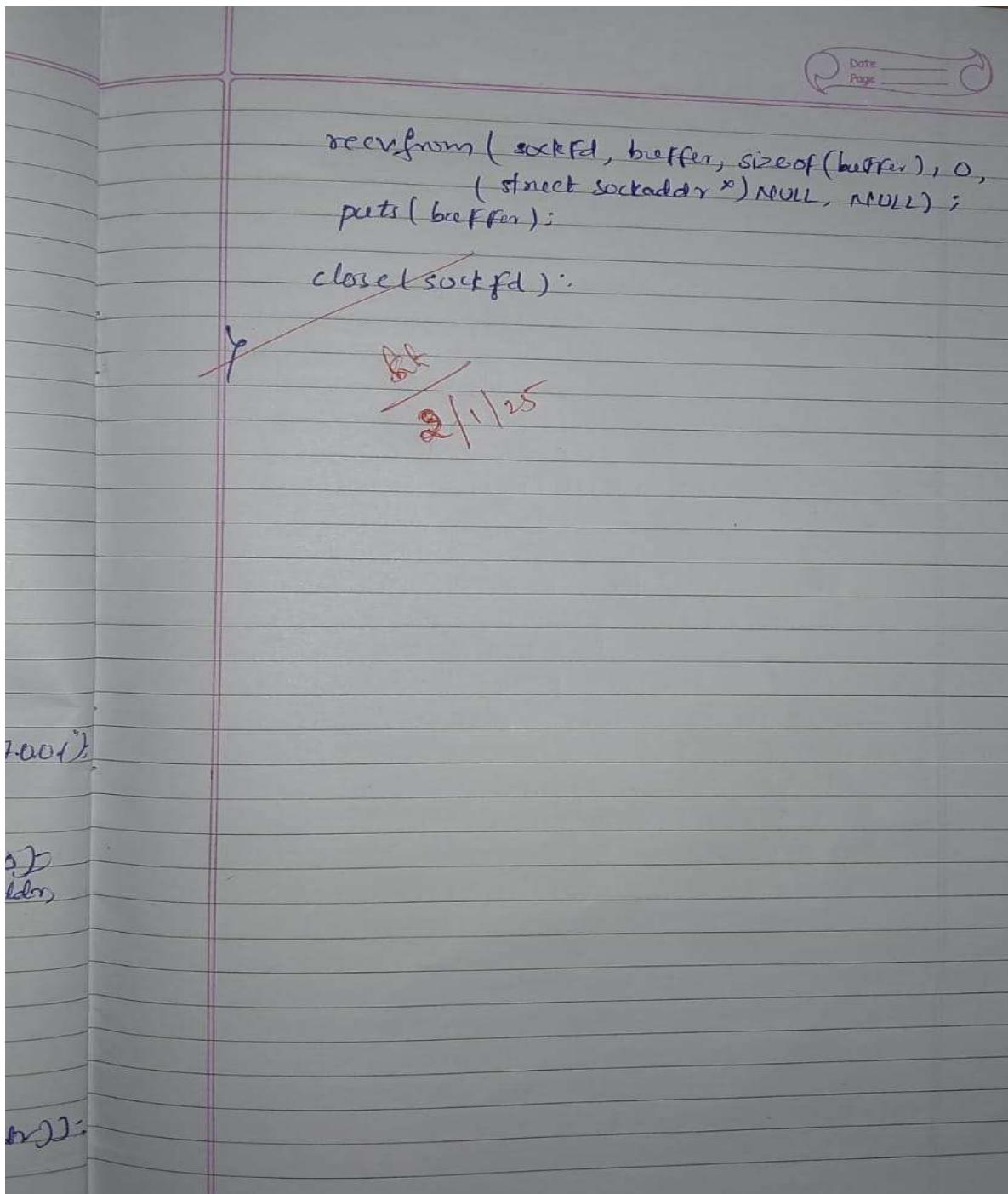
udp client driver program

```
#include <stdio.h>
#include <strings.h>
#include <sys/types.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <wininet.h>
#include <winstd.h>
#include <stdlib.h>

#define PORT 5000
#define MAXL 1000

int main()
{
    char buffer[1000];
    char *message = "Hello server";
    int sockfd, n;
    struct sockaddr_in servaddr;
    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");
    servaddr.sin_port = htons(PORT);
    servaddr.sin_family = AF_INET;
    sockfd = socket(AF_INET, SOCK_DGRAM, 0);
    if (connect(sockfd, (struct sockaddr*)&servaddr,
                sizeof(Servaddr)) < 0)
        printf("Error\n");
    exit(0);

    sendto(sockfd, message, MAXLINE, 0,
           (struct sockaddr*)&servaddr, sizeof(servaddr));
}
```



Server Output:
Server is online
Hello Server

Client Output:
Hello Client