

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.



LAB REPORT on

COMPUTER NETWORKS

Submitted by

V ASMA ANJUM (1BM20CS178)

in partial fulfilment 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
October-2022 to Feb-2023

**B. M. S. College of Engineering,
Bull Temple Road, Bangalore 560019**
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering



CERTIFICATE

This is to certify that the Lab work entitled "**Computer Networks**" carried out by **V ASMA ANJUM(1BM20CS178)**, who is a bonafide student of **B. M. S. College of Engineering**. It is inpartial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a **Computer Networks - (20CS5PCCON)** work prescribed for the said degree.

REKHA G S
Assistant Professor
Department of CSE
BMSCE, Bengaluru

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

Index

Sl. No.	Date	Experiment Title	Page No.
1	7/11/22	Creating a topology and simulating sending a simple PDU from source to destination using hub and switch as connecting devices.	1
2	14/11/22	Configuring IP address to Routers in Packet Tracer. Explore the following messages: Ping Responses, Destination unreachable, Request timed out, Reply	3
3	19/11/22	Configuring default route to the Router	9
4	28/11/22	Configuring DHCP within a LAN in a packet Tracer	15
5	5/12/22	Configuring RIP Routing Protocol in Routers	18
6	12/12/22	Demonstration of WEB server and DNS using Packet Tracer.	23
7	19/12/22	Write a program for error detecting code using CRC-CCITT (16-bits).	27
8	26/12/22	Write a program for a distance vector algorithm to find a suitable path for transmission.	30
9	2/1/23	Implement Dijkstra's algorithm to compute the shortest path for a given topology.	34
10	9/1/23	Write a program for congestion control using Leaky bucket algorithm.	36
11	16/1/23	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.	38
12	16/1/23	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.	41

Cycle 1:

Program1:

Creating a topology and simulating sending a simple PDU from source to destination using hub and switch as connecting devices.

Observation:

LAB - 1

SPLASH

Aim:- To create a topology & simulate sending a simple PDU from source to destination using hub & switch as connecting devices.

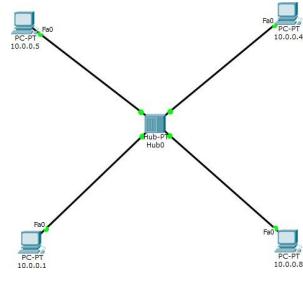
Topology (for sd wireless working in C)

Diagram shows a PC-PT connected via a bridge to a hub. The hub is connected to a server-PT. A dashed line labeled "switch" connects the hub and the server.

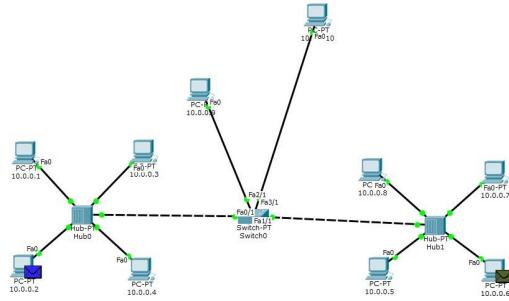
Procedures:- PCO

- Open Cisco packet tracer app
- At the left bottom, select a generic server & drop it on screen.
- Select generic node (end devices)
- We can place a note renaming it
- We can give it an IP address as 10.0.0.1.
- Then use a connection wire
- Since generic node & generic server are peers. we can use copper crosswire.
- A green dot appears at both end.
- At right end in simulation mode, use auto capture.
- Drop 2 messages at PC-0 & server
- They move & show us connection is successful

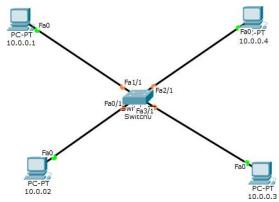
Screenshots:



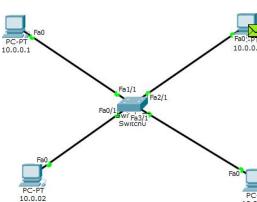
Hub Topology



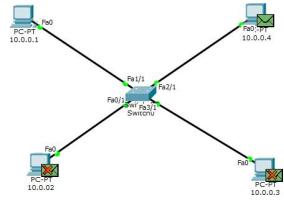
Hub-Switch Topology



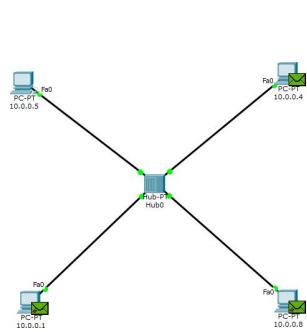
Switch Topology



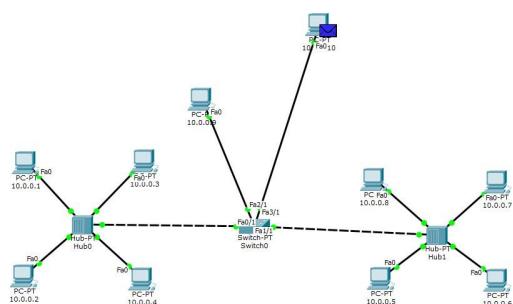
Switch after forming ARP



Switch before forming ARP



Hub broadcasting



Unicasting switch

Program 2:

Configuring IP address to Routers in Packet Tracer. Explore the following messages:
Ping Responses, Destination unreachable, Request timed out, Reply

Observation:

14/11/22

LAB:03 : Configuring IP address to Routers in Packet Traces. Explore the following messages : Ping response, destination unreachable, request timed out, reply.

Topology:-

Procedure:-

- 1) Add three routers: Router 0, Router 1 & Router 2 and end devices PCs: PC0 & PC1
- 2) Connect PC0 & Router0 & PC1 and Router2 with copper cross over wire → fastethernet 0/0.
- 3) Connect Router0 to Router1 and Router1 to Router2 using serial DCE connectors → serial 2/0 and serial 3/0 connection.

Date _____
Page _____ SPLASH

4) set all the IP addresses, subnet mask = 255.0.0.0 (for all PC's) & gateways accordingly.

PC0 → IPaddr = 10.0.0.1, gateway 10.0.0.10
Router0 gateway 1 = 10.0.0.10 & gateway 2 = 20.0.0.10

Router1 → gateway 1 = 20.0.0.20 and gateway 2 = 30.0.0.10

Router2 → gateway 1 = 30.0.0.20 and gateway 2 = 40.0.0.10

PC1 → IP addr = 40.0.0.1, gateway = 40.0.0.10

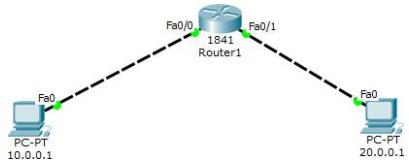
5) set up the connections b/w Router0 & PC0, Router0 & Router1, Router1 & Router2 and PC1 using CLI commands.

* CLI commands for Router0
Router > enable (password)
Router # config t
Router (config) # interface fastethernet 0/0
Router (config-if) # ip address 10.0.0.10
Router (config-if) # no shut
Router (config-if) # exit
* similarly we do the same for router1 & router2.
* for Router1 ip address is 20.0.0.20 & to interface with serial 3/0 with ip address 30.0.0.10

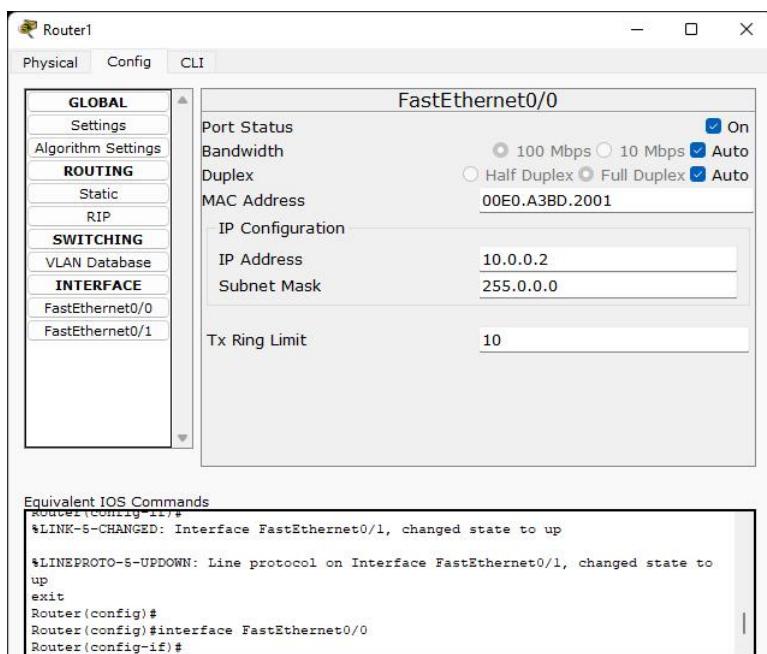
5) For router 2, interface serial 2/0
 connected with ip address 30.0.0.20. And
 then interface router 2/1 & PC1
 with fastethernet 0/0 ip address
 40.0.0.10. (Note: port number)
 6) After all the connections, the
 lights turn green, indicating
 that the connection is established
 7) After this step we teach the
 router about other routers &
 that they are indirectly
 connected (3 routers)
 * Reaching Router 1 about 30.0.0.0 &
 40.0.0.0
 Router # config
 Router (config)# ip route 30.0.0.0
 255.0.0.0 20.0.0.10
 Router (config)# ip route 40.0.0.0
 255.0.0.0 20.0.0.20
 Router (config) # exit
 Router # show ip route
 C 10.0.0.0/8 [1/0] via 20.0.0.10
 C 20.0.0.0/8 directly connected fastethernet 0/0
 S 30.0.0.0/8 [1/0] via 20.0.0.20
 S 40.0.0.0/8 [1/0] via 20.0.0.20
 * Teaching Router 2 about 10.0.0.0
 and 40.0.0.0

Router # config
 Router (config) # ip route 10.0.0.0
 255.0.0.0 20.0.0.10
 Router (config) # ip route 40.0.0.0
 255.0.0.0 30.0.0.20
 Exit
 show ip route
 S 10.0.0.0/8 [1/0] via 20.0.0.10
 C 20.0.0.0/8 directly se 2/0
 C 30.0.0.0/8 directly se 3/0
 S 40.0.0.0/8 [1/0] via 30.0.0.20
 * Reaching Router 3 about 10.0.0.0
 and 20.0.0.0
 Router (config) # ip route 10.0.0.0
 255.0.0.0 30.0.0.10
 Router (config) # ip route 20.0.0.0
 255.0.0.0 30.0.0.10
 Exit
 show ip route
 S 10.0.0.0 [1/0] via 30.0.0.10
 S 20.0.0.0 [1/0] via 30.0.0.10
 C 30.0.0.0 directly se 3/0
 C 40.0.0.0 directly fastethernet 0/0
 * Now to check if the connection
 made are correct. Select PC1 and
 ping command prompt ping to
 each of the router first and
 then at the end ping to the
 other pc. All the 4 messages will
 be sent & received with 0 loss

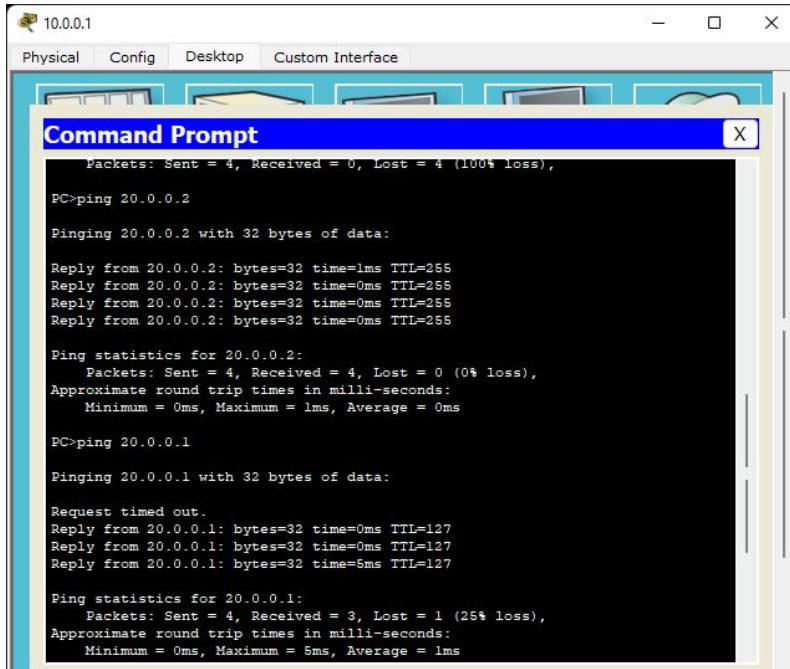
Screenshots:



Topology

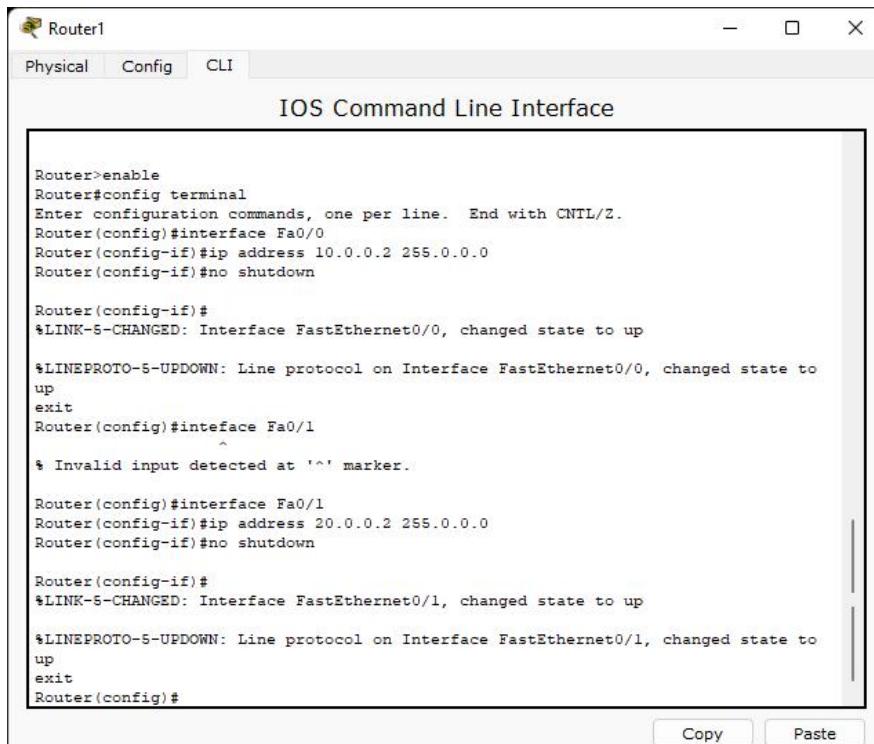


Configuring IP address for end device



```
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 20.0.0.2: bytes=32 time=1ms TTL=255  
Reply from 20.0.0.2: bytes=32 time=0ms TTL=255  
Reply from 20.0.0.2: bytes=32 time=0ms TTL=255  
Reply from 20.0.0.2: bytes=32 time=0ms TTL=255  
Ping statistics for 20.0.0.2:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 0ms, Maximum = 1ms, Average = 0ms  
PC>ping 20.0.0.1  
Pinging 20.0.0.1 with 32 bytes of data:  
Request timed out.  
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127  
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127  
Reply from 20.0.0.1: bytes=32 time=5ms TTL=127  
Ping statistics for 20.0.0.1:  
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 0ms, Maximum = 5ms, Average = 1ms
```

Successful ping message



```
Router>enable  
Router#config terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
Router(config)#interface Fa0/0  
Router(config-if)#ip address 10.0.0.2 255.0.0.0  
Router(config-if)#no shutdown  
  
Router(config-if)#  
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up  
  
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to  
up  
exit  
Router(config)#interface Fa0/1  
^  
* Invalid input detected at '^' marker.  
  
Router(config)#interface Fa0/1  
Router(config-if)#ip address 20.0.0.2 255.0.0.0  
Router(config-if)#no shutdown  
  
Router(config-if)#  
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up  
  
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to  
up  
exit  
Router(config)#

```

Configuring router ip address for the interface

10.0.0.1

Physical Config Desktop Custom Interface

Command Prompt

```
Minimum = 0ms, Maximum = 5ms, Average = 1ms
PC>ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Reply from 20.0.0.1: bytes=32 time=0ms TTL=127

Ping statistics for 20.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>ping 30.0.0.1
Pinging 30.0.0.1 with 32 bytes of data:
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.
Request timed out.

Ping statistics for 30.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC:>
```

Destination host unreachable message

10.0.0.1

Physical Config Desktop Custom Interface

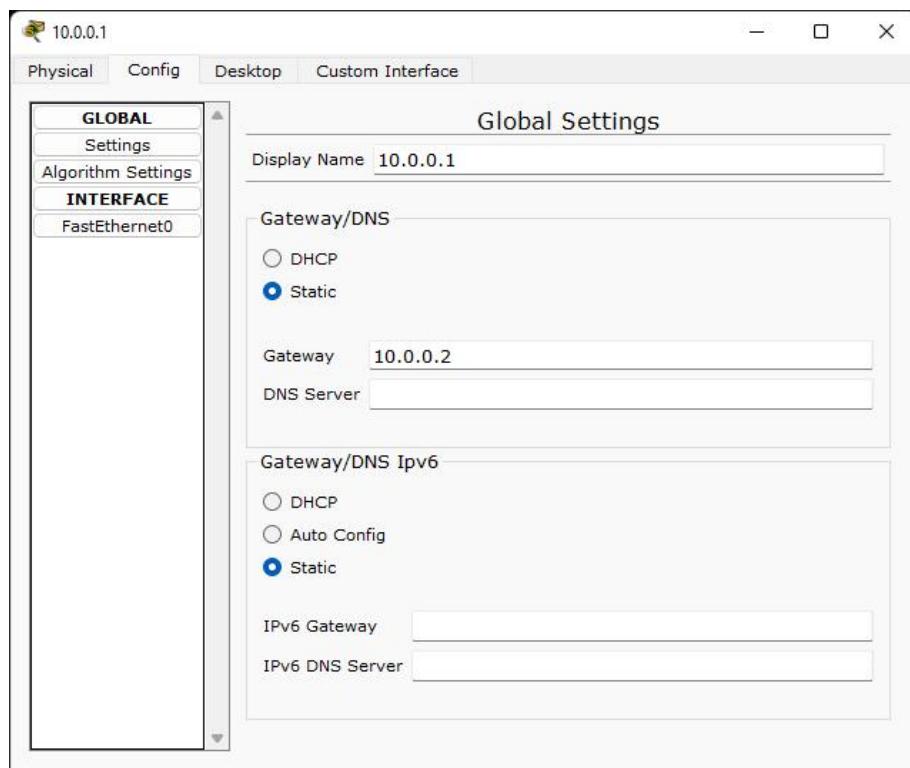
Command Prompt

```
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
PC>ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 20.0.0.2:
    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:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 20.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 20.0.0.2
```

Request timed out message

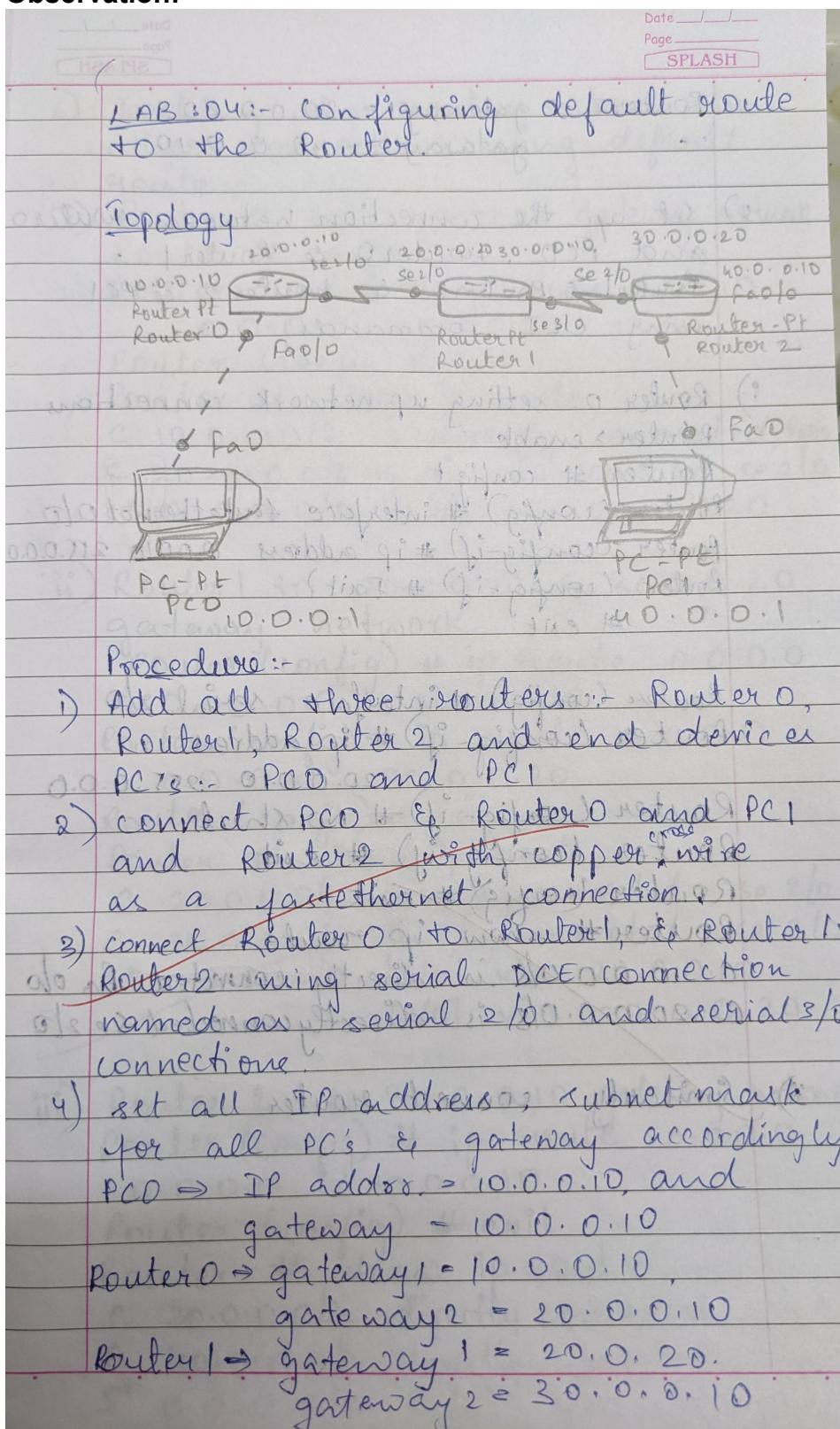


Configuring gateway for the end device

Program 3:

Configuring default route to the Router

Observation:



Router 2 \Rightarrow gateway 1 = 30.0.0.20,
gateway 2 = 40.0.0.10.

- 5) Set up the connection between routers and PC0, Router0 & Router1, Router1 & Router2 & Router2 & PC1 using CLI commands.

- i) Router 0: setting up network connections

Router > enable

Router # config t

Router (config) # interface fastethernet 0/0

Router (config-if) # ip address 10.0.0.10 255.0.0.0

Router (config-if) # exit

exit

Router (config) # interface serial 1/0

Router (config-if) # ip address

20.0.0.10 255.0.0.0

Router (config-if) # no shutdown

Router (config-if) # exit

Router (config) # exit

Router # show ip route

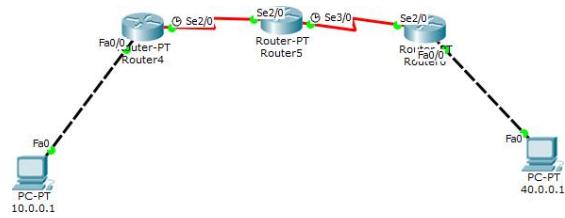
C 10.0.0.0/8 is directly connected Fa 0/0

C 20.0.0.0/8 is directly connected Se 1/0

- ii) Similarly connect router 1 to 2

- Date _____
 Page _____
- SPLASH**
- 2) Teaching the routers about the other connections using default routes.
- i) Router D to 20.0.0.20 gateway network
- Router # config
- ```
Router(config)# ip route 0.0.0.0
 0.0.0.0 20.0.0.20
```
- Router(config)# exit
- Router# show ip route
- C 10.0.0.0/8 is directly connected, fa0  
 C 20.0.0.0/8 is directly connected, se 2/0  
 S\* 0.0.0.0/0 [1/0] via 20.0.0.20
- ii) Router 1 to 20.0.0.20 via 30.0.0.20 gateway network.
- Router(config)# ip route 0.0.0.0
 0.0.0.0 20.0.0.10
- Router(config)# ip route 0.0.0.0
 0.0.0.0 30.0.0.20
- Router(config)# exit
- Router# show ip route
- C 20.0.0.0/8 directly connected, se 2/0  
 C 30.0.0.0/8 directly connected, se 3/0  
 S\* 0.0.0.0/0 [1/0] via 20.0.0.10  
 [1/0] via 30.0.0.20
- iii) Router 2 to 30.0.0.10 gateway network
- Router(config)# ip route 0.0.0.0
 0.0.0.0 30.0.0.10
- Router(config)# exit
- Router# show ip route
- C 30.0.0.0/8 directly connected, se 2/0  
 C 40.0.0.0/8 directly connected, fa 0/1  
 S\* 0.0.0.0/0 [1/0] via 30.0.0.10

## Screenshots:



Topology

```
Reply from 30.0.0.1: bytes=32 time=1ms TTL=254
Ping statistics for 30.0.0.1:
 Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
 Approximate round trip times in milli-seconds:
 Minimum = 1ms, Maximum = 9ms, Average = 4ms

PC>ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data:
Reply from 20.0.0.2: bytes=32 time=11ms TTL=254
Reply from 20.0.0.2: bytes=32 time=6ms TTL=254
Reply from 20.0.0.2: bytes=32 time=1ms TTL=254
Reply from 20.0.0.2: bytes=32 time=8ms TTL=254

Ping statistics for 20.0.0.2:
 Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
 Approximate round trip times in milli-seconds:
 Minimum = 1ms, Maximum = 11ms, Average = 6ms

PC>ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Reply from 20.0.0.1: bytes=32 time=13ms TTL=253
Reply from 20.0.0.1: bytes=32 time=2ms TTL=253
Reply from 20.0.0.1: bytes=32 time=13ms TTL=253
Reply from 20.0.0.1: bytes=32 time=9ms TTL=253

Ping statistics for 20.0.0.1:
 Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
 Approximate round trip times in milli-seconds:
 Minimum = 2ms, Maximum = 13ms, Average = 9ms

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=11ms TTL=253
Reply from 10.0.0.2: bytes=32 time=10ms TTL=253
Reply from 10.0.0.2: bytes=32 time=2ms TTL=253
Reply from 10.0.0.2: bytes=32 time=6ms TTL=253

Ping statistics for 10.0.0.2:
 Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
 Approximate round trip times in milli-seconds:
 Minimum = 2ms, Maximum = 11ms, Average = 7ms

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=8ms TTL=125
Reply from 10.0.0.1: bytes=32 time=12ms TTL=125
Reply from 10.0.0.1: bytes=32 time=8ms TTL=125
Reply from 10.0.0.1: bytes=32 time=11ms TTL=125

Ping statistics for 10.0.0.1:
 Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
 Approximate round trip times in milli-seconds:
 Minimum = 8ms, Maximum = 12ms, Average = 9ms
```

Successful ping messages

### Command Prompt

```
Packet Tracer PC Command Line 1.0
PC>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

Reply from 40.0.0.2: bytes=32 time=1ms TTL=255
Reply from 40.0.0.2: bytes=32 time=0ms TTL=255
Reply from 40.0.0.2: bytes=32 time=0ms TTL=255
Reply from 40.0.0.2: bytes=32 time=1ms TTL=255

Ping statistics for 40.0.0.2:
 Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
 Approximate round trip times in milli-seconds:
 Minimum = 0ms, Maximum = 1ms, Average = 0ms

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=0ms TTL=255

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

PC>ping 30.0.0.1

Pinging 30.0.0.1 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 30.0.0.1:
 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 40.0.0.2: Destination host unreachable.
Reply from 40.0.0.2: Destination host unreachable.
Request timed out.
Reply from 40.0.0.2: Destination host unreachable.

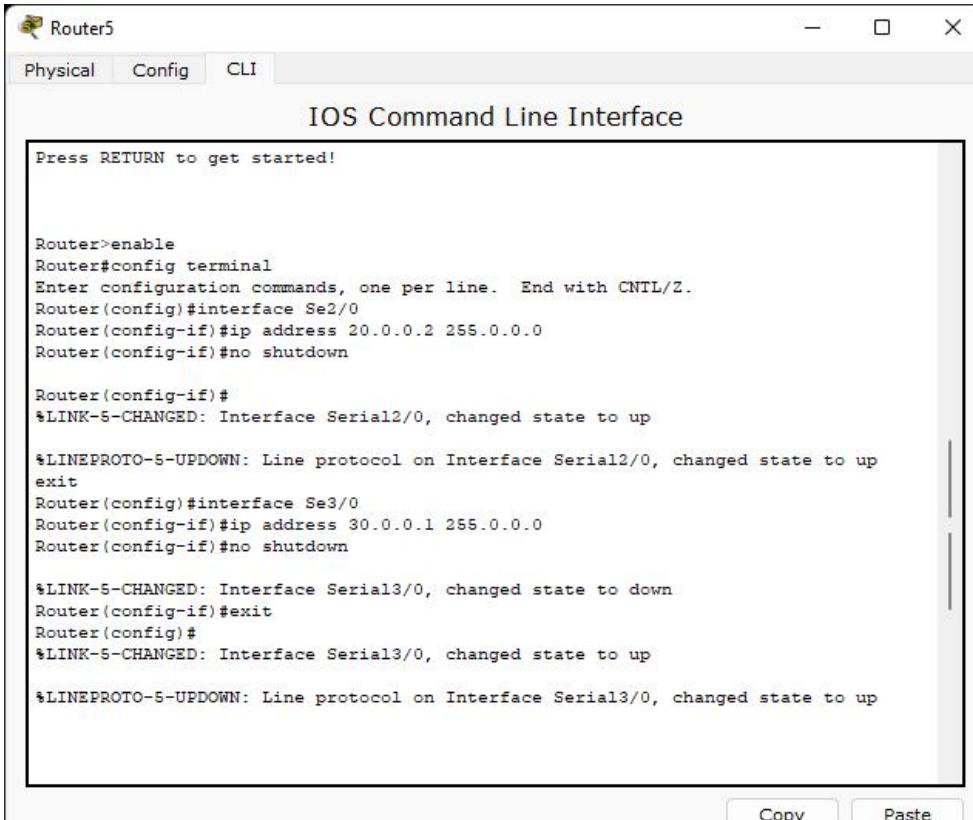
Ping statistics for 20.0.0.2:
 Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

PC>ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data:

Reply from 40.0.0.2: Destination host unreachable.
Request timed out.
Reply from 40.0.0.2: Destination host unreachable.
Reply from 40.0.0.2: Destination host unreachable.
```

Request Timed out and Destination host unreachable



Router5

Physical Config CLI

IOS Command Line Interface

```
Press RETURN to get started!

Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface Se2/0
Router(config-if)#ip address 20.0.0.2 255.0.0.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

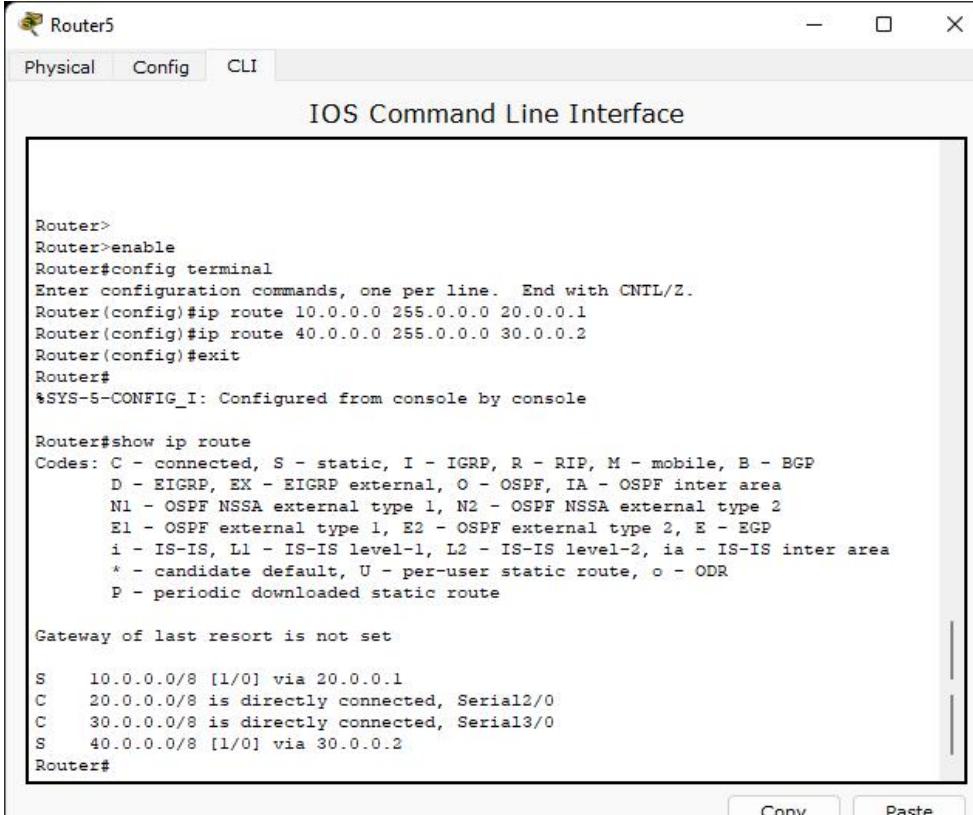
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
exit
Router(config)#interface Se3/0
Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial3/0, changed state to down
Router(config-if)#exit
Router(config)#
%LINK-5-CHANGED: Interface Serial3/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up
```

Copy Paste

### Configuring ip address for router interface



Router5

Physical Config CLI

IOS Command Line Interface

```
Router>
Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 10.0.0.0 255.0.0.1 20.0.0.1
Router(config)#ip route 40.0.0.0 255.0.0.2 30.0.0.2
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#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

S 10.0.0.0/8 [1/0] via 20.0.0.1
C 20.0.0.0/8 is directly connected, Serial2/0
C 30.0.0.0/8 is directly connected, Serial3/0
S 40.0.0.0/8 [1/0] via 30.0.0.2
Router#
```

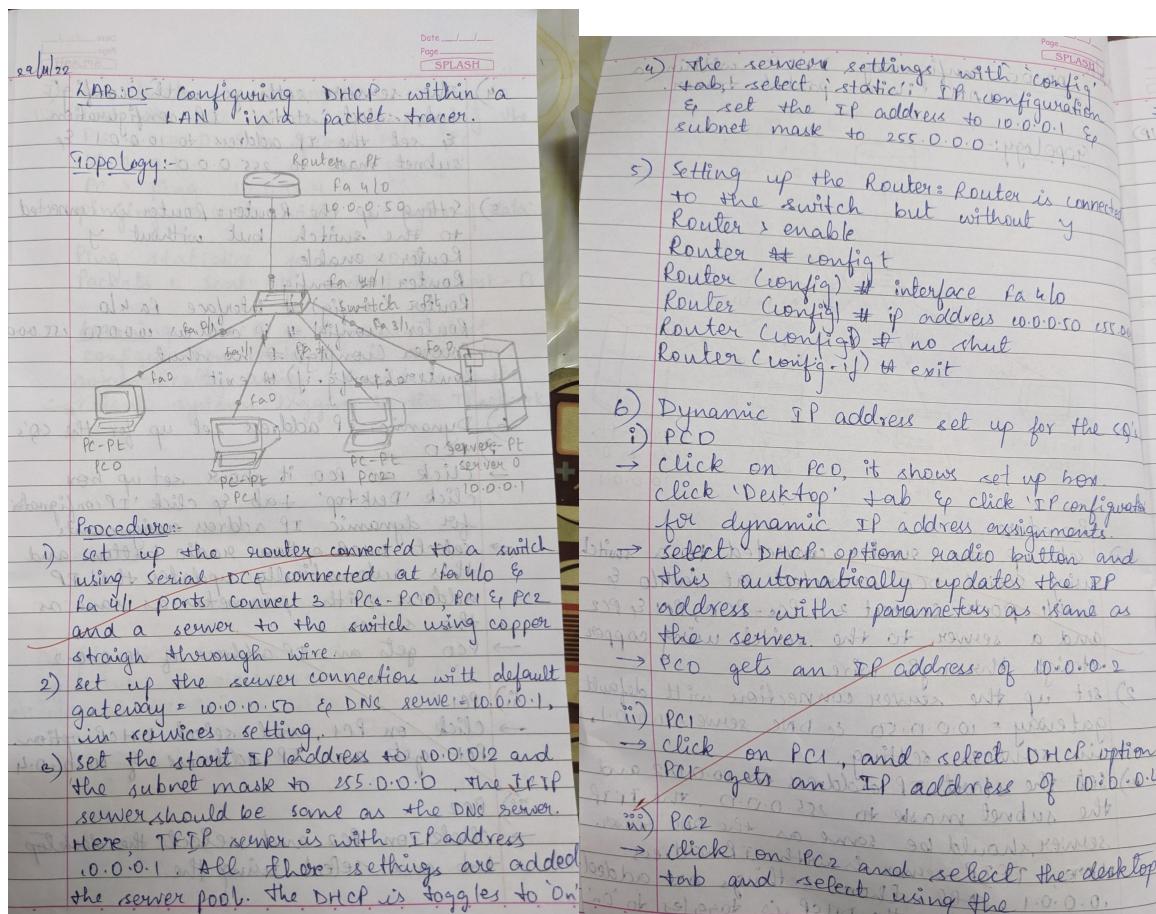
Copy Paste

### Configuring ip route

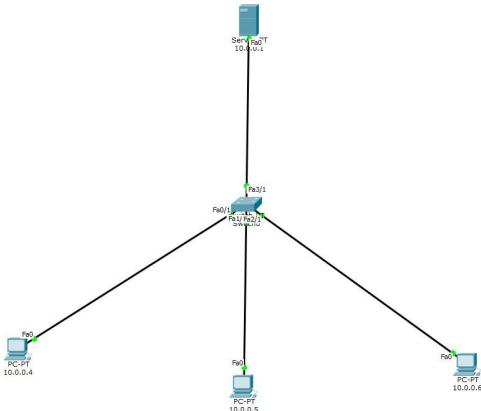
## Program 4:

### Configuring DHCP within a LAN in a packet Tracer

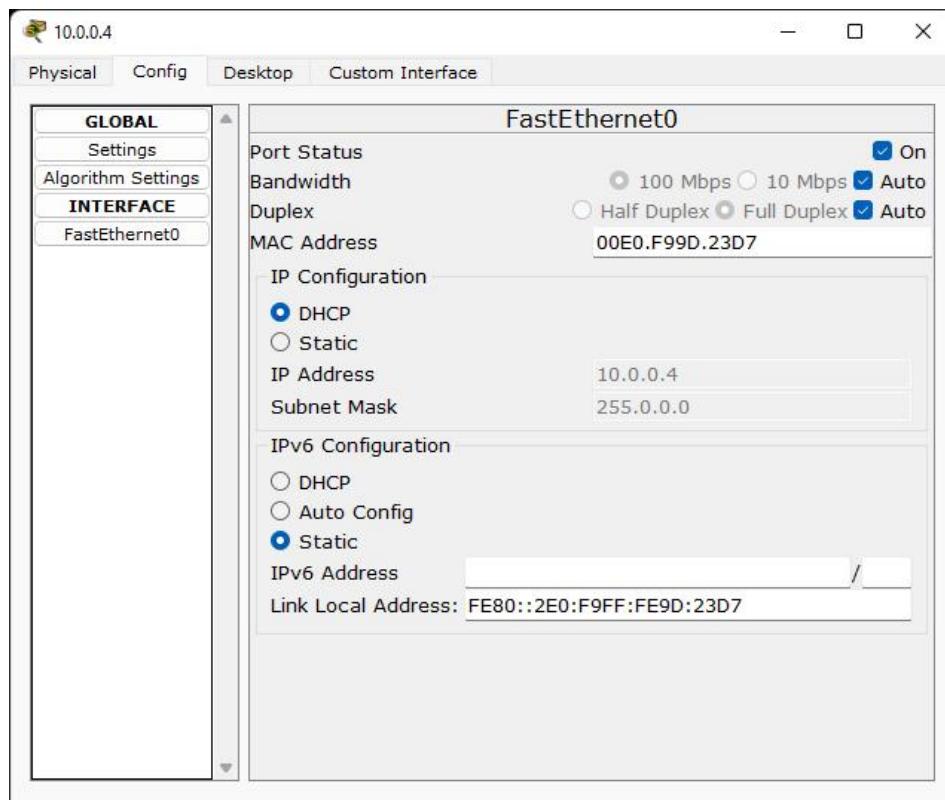
#### Observations:



## Screenshots:



Topology



DHCP for end devices

Packet Tracer PC Command Line 1.0  
PC>ping 10.0.0.6  
  
Pinging 10.0.0.6 with 32 bytes of data:  
  
Reply from 10.0.0.6: bytes=32 time=1ms TTL=128  
Reply from 10.0.0.6: bytes=32 time=0ms TTL=128  
Reply from 10.0.0.6: bytes=32 time=0ms TTL=128  
Reply from 10.0.0.6: bytes=32 time=0ms TTL=128  
  
Ping statistics for 10.0.0.6:  
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
Minimum = 0ms, Maximum = 1ms, Average = 0ms  
  
PC>

Successful ping message

**SERVICES**

- HTTP
- DHCP
- DHCPv6
- TFTP
- DNS
- SYSLOG
- AAA
- NTP
- EMAIL
- FTP

**DHCP**

| Interface                 | FastEthernet0 | Service | On | Off |
|---------------------------|---------------|---------|----|-----|
| Pool Name                 | serverPool    |         |    |     |
| Default Gateway           | 0.0.0.0       |         |    |     |
| DNS Server                | 0.0.0.0       |         |    |     |
| Start IP Address :        | 10 0 0 4      |         |    |     |
| Subnet Mask:              | 255 0 0 0     |         |    |     |
| Maximum number of Users : | 512           |         |    |     |
| TFTP Server:              | 0.0.0.0       |         |    |     |

Add Save Remove

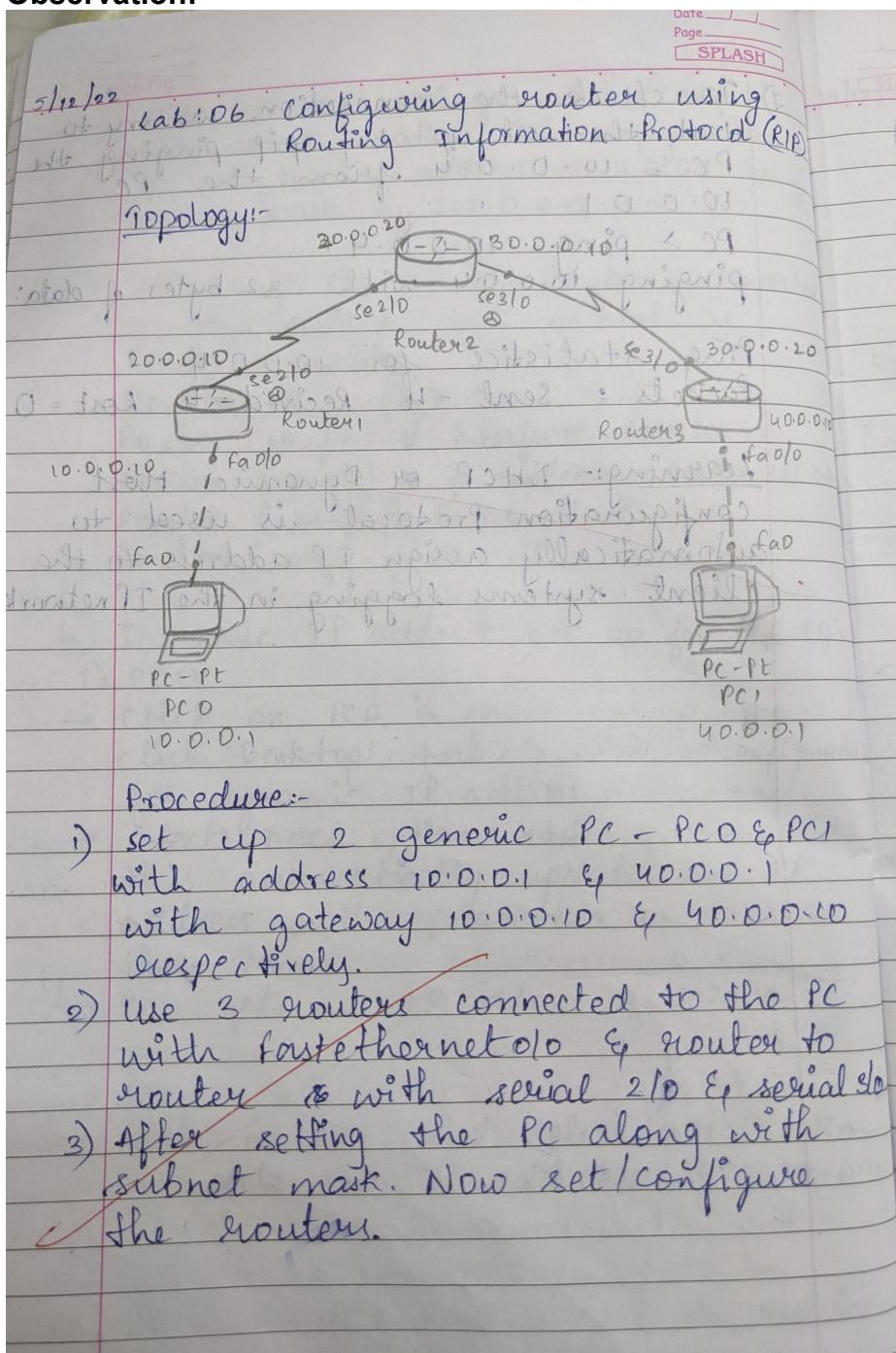
| Pool Name | Default Gateway | DNS Server | Start IP Address | Subnet Mask | Max User | TFTP    |
|-----------|-----------------|------------|------------------|-------------|----------|---------|
| server... | 0.0.0.0         | 0.0.0.0    | 10.0.0.4         | 255.0.0.0   | 512      | 0.0.0.0 |

DHCP pool for the server

## Program 5:

### Configuring RIP Routing Protocol in Routers

#### Observation:



perform Router1 configuration and work  
 # enable o -> router  
 # config t for interface to  
 # interface for 0/0 downstream  
 # ip address 10.0.0.10 255.0.0.0  
 # no shut fix it  
 # exit  
 # interface serial 2/0 when  
 # config rj45 baud 0.0.0.0  
 # interface use 2/0 upstream with  
 # ip address 20.0.0.10 255.0.0.0  
 # encapsulation PPP  
 # clock rate 64000 or even  
 # dialup shutdown off the connection  
 .  
 .  
 .  
 for Router1 run the above comm. on  
 20.0.0.20 network without clock  
 rate 6400 because serial 2/0  
 holds the clock at 1.0.0.0N  
 .  
 .  
 .  
 for Router2 run above comm on  
 30.0.0.20 with clock rate  
 (64000) command.  
 .  
 .  
 .  
 Run normal ip (Router0) command  
 to set up IP of it for interface  
 10.0.0.10 with 255.0.0.0 subnet mask  
 no need to do this in Router1 because 912  
 can work through Router1 without affecting  
 .  
 .  
 .

Date / /  
Page / /  
SPLASH

Now for dynamically teaching Router-0  
 Router rip  
 # network 10.0.0.0  
 # network 12.0.0.0  
 # exit

Similarly for router 01 the network will be 20.0.0.0 & 30.0.0.0 and for router 2 the networks will be 40.0.0.0.

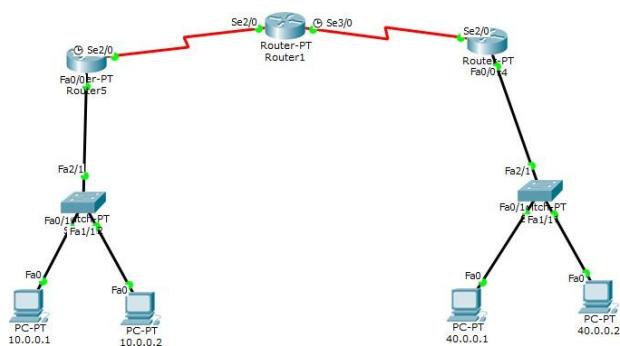
Now ran show ip route it will show all the direct & via connections b/w the devices.

Now when you ping the PC's to ip no. 0.0.0.1 the message will be sent & received.

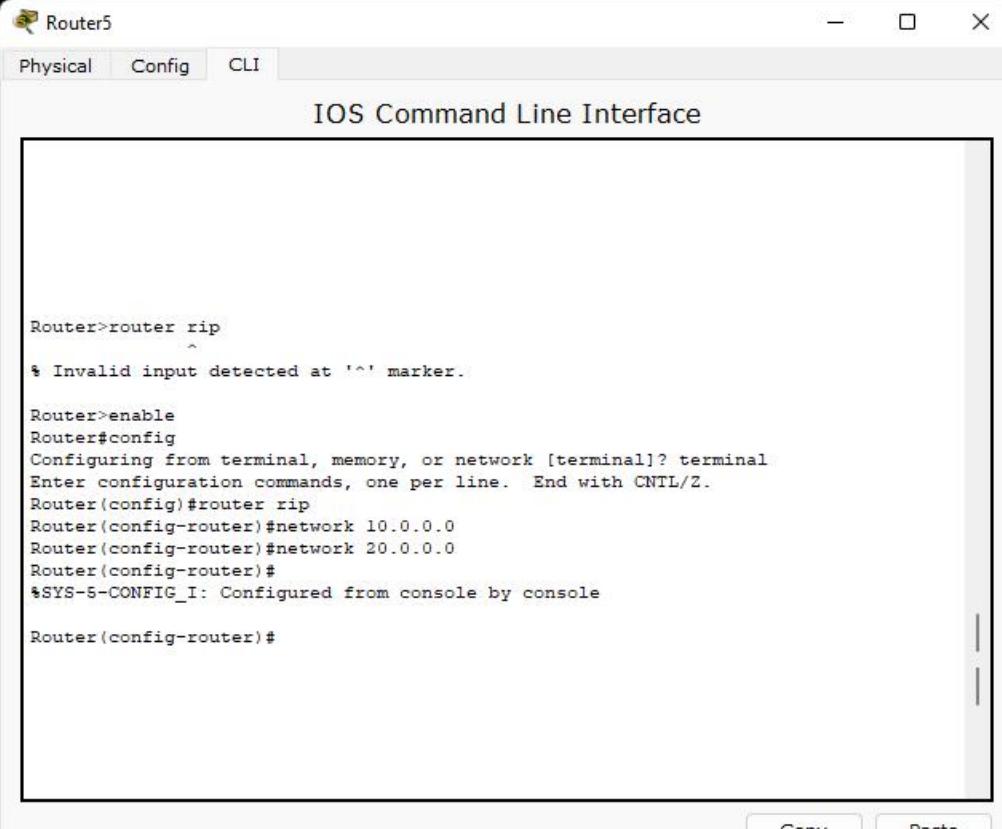
~~Learning:~~

- Routing Information Protocol (RIP) is a dynamic routing protocol that uses hop count as a routing metric to find a best path between source & destination network.
- RIP version 1 which is known as classful routing protocol because it does not send info. of subnet mask. RIP v2 is classless routing protocol since it sends info. of subnet mask.

## Screenshots:



Topology



Router5

Physical Config CLI

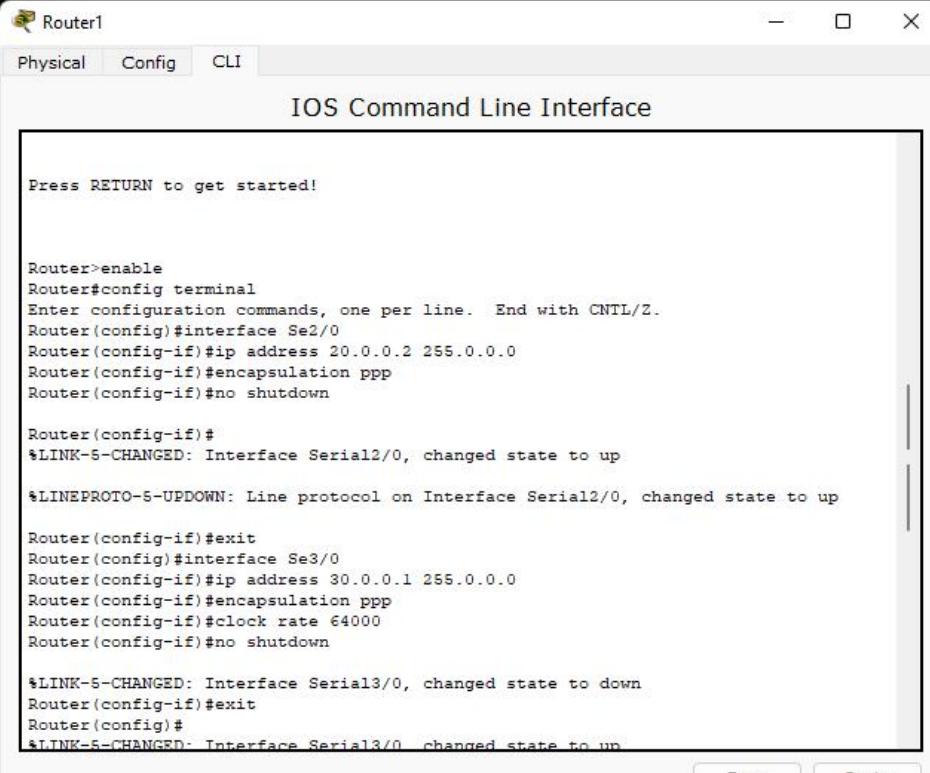
IOS Command Line Interface

```
Router>router rip
 ^
% Invalid input detected at '^' marker.

Router>enable
Router#config
Configuring from terminal, memory, or network [terminal]? terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 10.0.0.0
Router(config-router)#network 20.0.0.0
Router(config-router)#
*SYS-5-CONFIG_I: Configured from console by console
Router(config-router)#

Copy Paste
```

### Configuring RIP for the router



Router1

Physical Config CLI

IOS Command Line Interface

```
Press RETURN to get started!

Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface Se2/0
Router(config-if)#ip address 20.0.0.2 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up

Router(config-if)#exit
Router(config)#interface Se3/0
Router(config-if)#ip address 30.0.0.1 255.0.0.0
Router(config-if)#encapsulation ppp
Router(config-if)#clock rate 64000
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial3/0, changed state to down
Router(config-if)#exit
Router(config)#
%LINK-5-CHANGED: Interface Serial3/0 changed state to up

Copy Paste
```

### Configuring ip address for the router interface

10.0.0.1

Physical Config Desktop Custom Interface

### Command Prompt

```
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>ping 10.0.0.3

Pinging 10.0.0.3 with 32 bytes of data:

Reply from 10.0.0.3: bytes=32 time=1ms TTL=255
Reply from 10.0.0.3: bytes=32 time=4ms TTL=255
Reply from 10.0.0.3: bytes=32 time=0ms TTL=255
Reply from 10.0.0.3: bytes=32 time=0ms TTL=255

Ping statistics for 10.0.0.3:
 Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
 Approximate round trip times in milli-seconds:
 Minimum = 0ms, Maximum = 4ms, Average = 1ms

PC>ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data:

Reply from 20.0.0.1: bytes=32 time=0ms TTL=255
Reply from 20.0.0.1: bytes=32 time=0ms TTL=255
Reply from 20.0.0.1: bytes=32 time=0ms TTL=255
Reply from 20.0.0.1: bytes=32 time=1ms TTL=255

Ping statistics for 20.0.0.1:
 Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
 Approximate round trip times in milli-seconds:
 Minimum = 0ms, Maximum = 1ms, Average = 0ms

PC>ping 20.0.0.2

Pinging 20.0.0.2 with 32 bytes of data:

Reply from 20.0.0.2: bytes=32 time=1ms TTL=254
Reply from 20.0.0.2: bytes=32 time=1ms TTL=254
Reply from 20.0.0.2: bytes=32 time=1ms TTL=254
Reply from 20.0.0.2: bytes=32 time=27ms TTL=254

Ping statistics for 20.0.0.2:
 Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
 Approximate round trip times in milli-seconds:
 Minimum = 1ms, Maximum = 27ms, Average = 7ms

PC>ping 40.0.0.2

Pinging 40.0.0.2 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.2: bytes=32 time=12ms TTL=125
Reply from 40.0.0.2: bytes=32 time=2ms TTL=125
Reply from 40.0.0.2: bytes=32 time=12ms TTL=125

Ping statistics for 40.0.0.2:
 Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
 Approximate round trip times in milli-seconds:
 Minimum = 2ms, Maximum = 12ms, Average = 8ms

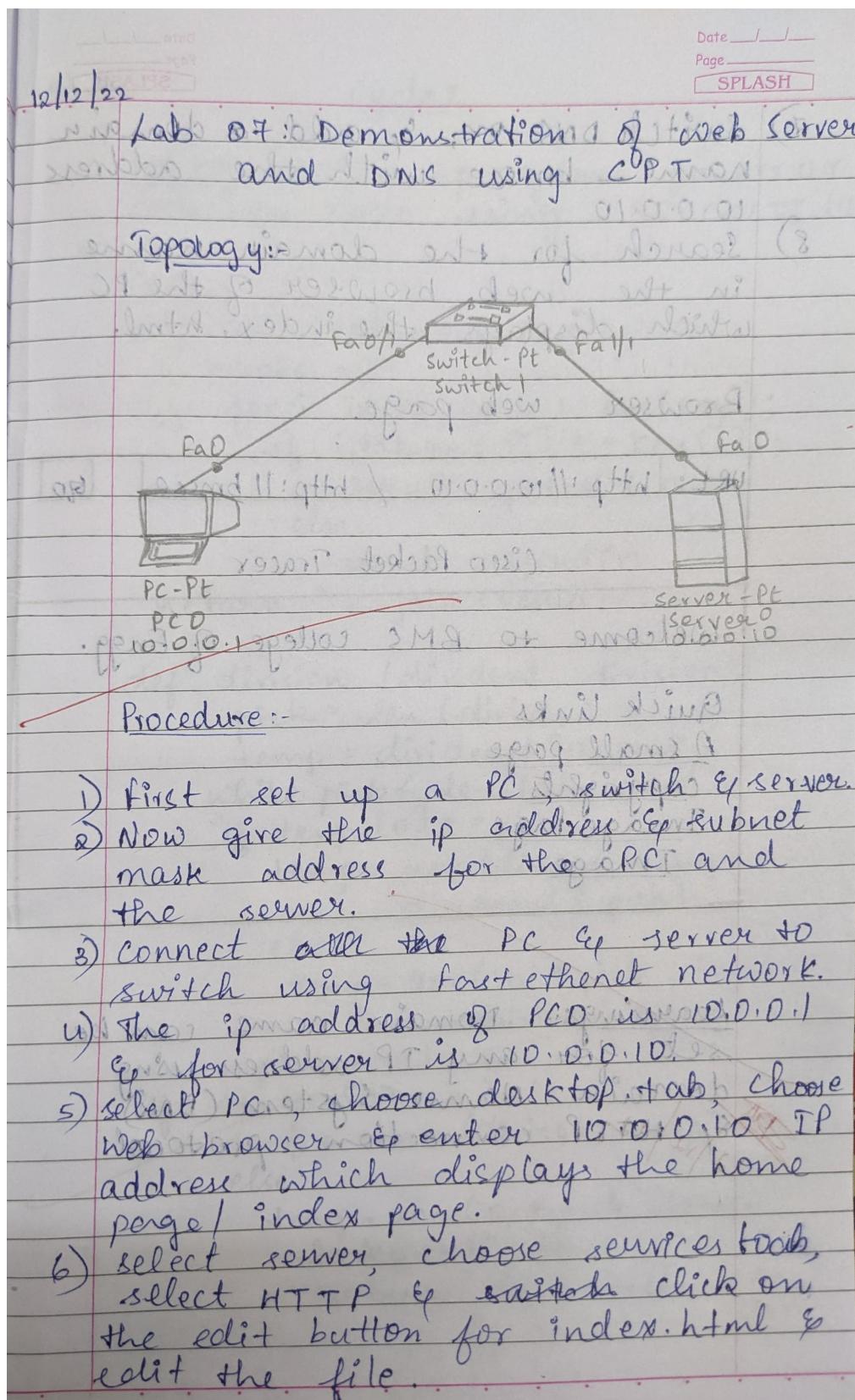
PC>
```

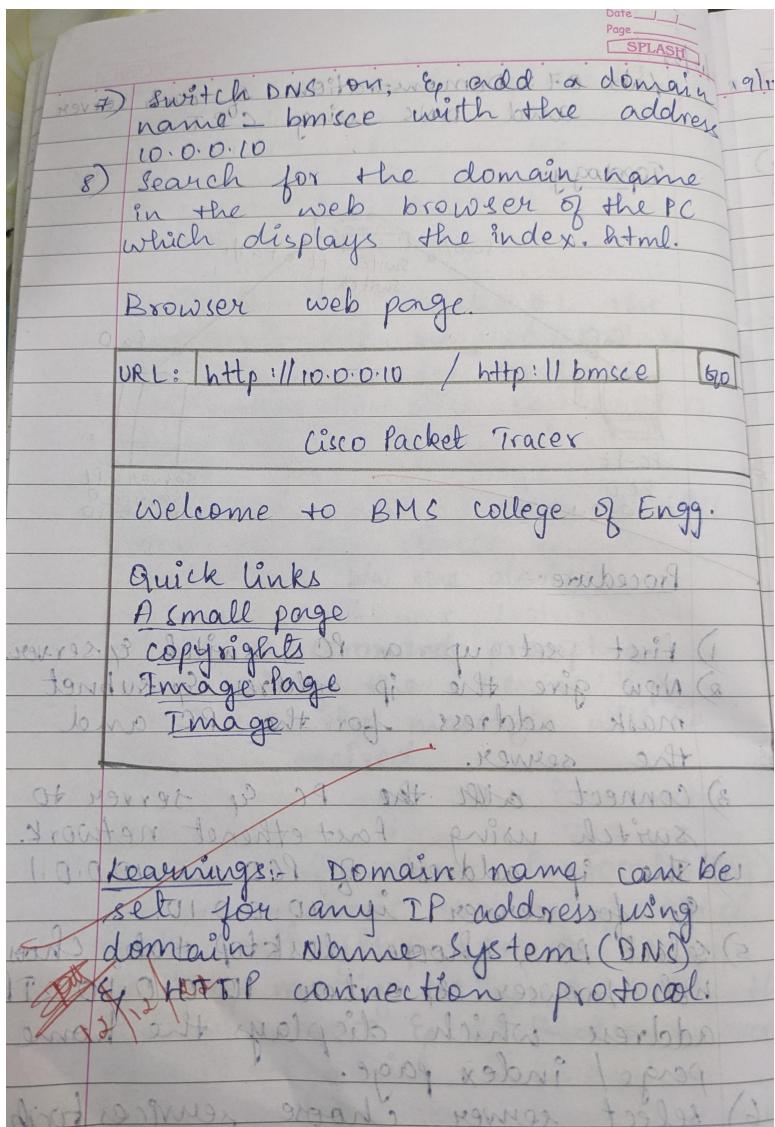
Successful ping messages

## Program 7:

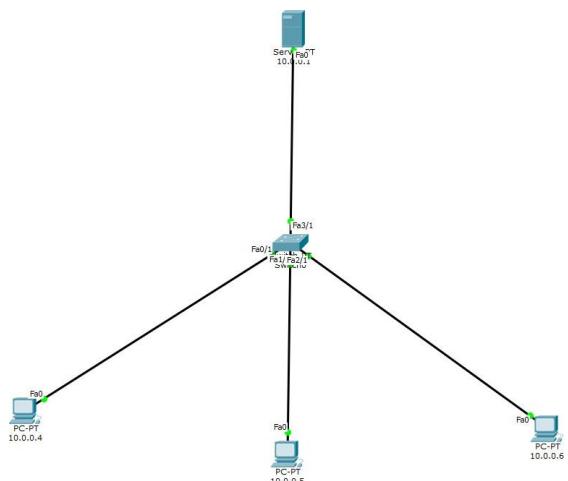
Demonstration of WEB server and DNS using Packet Tracer.

### Observation:

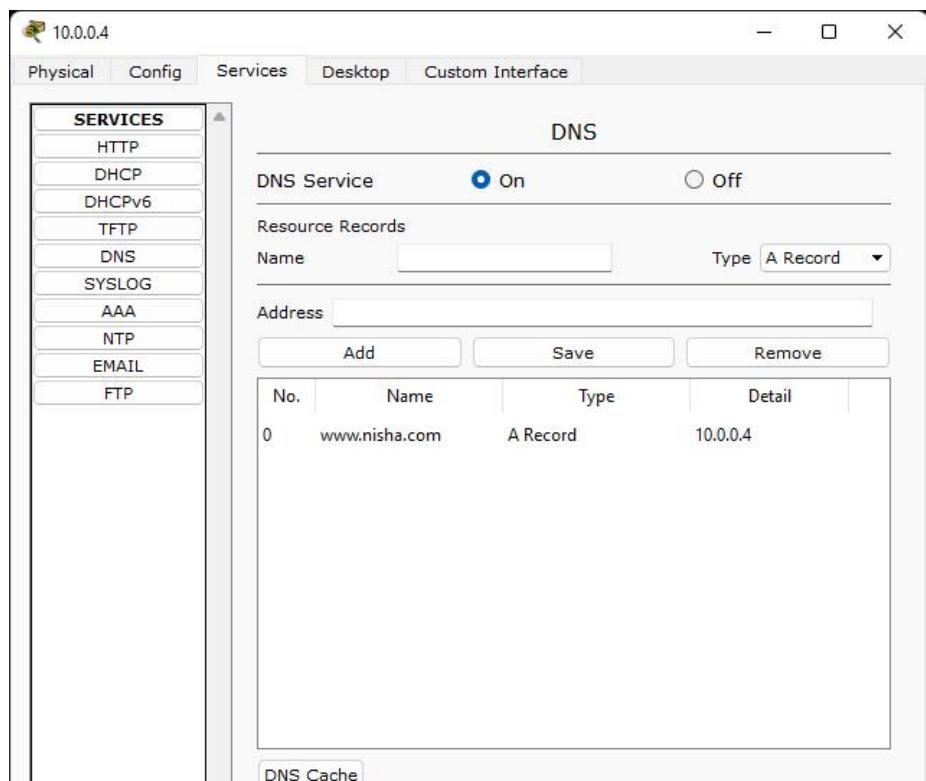




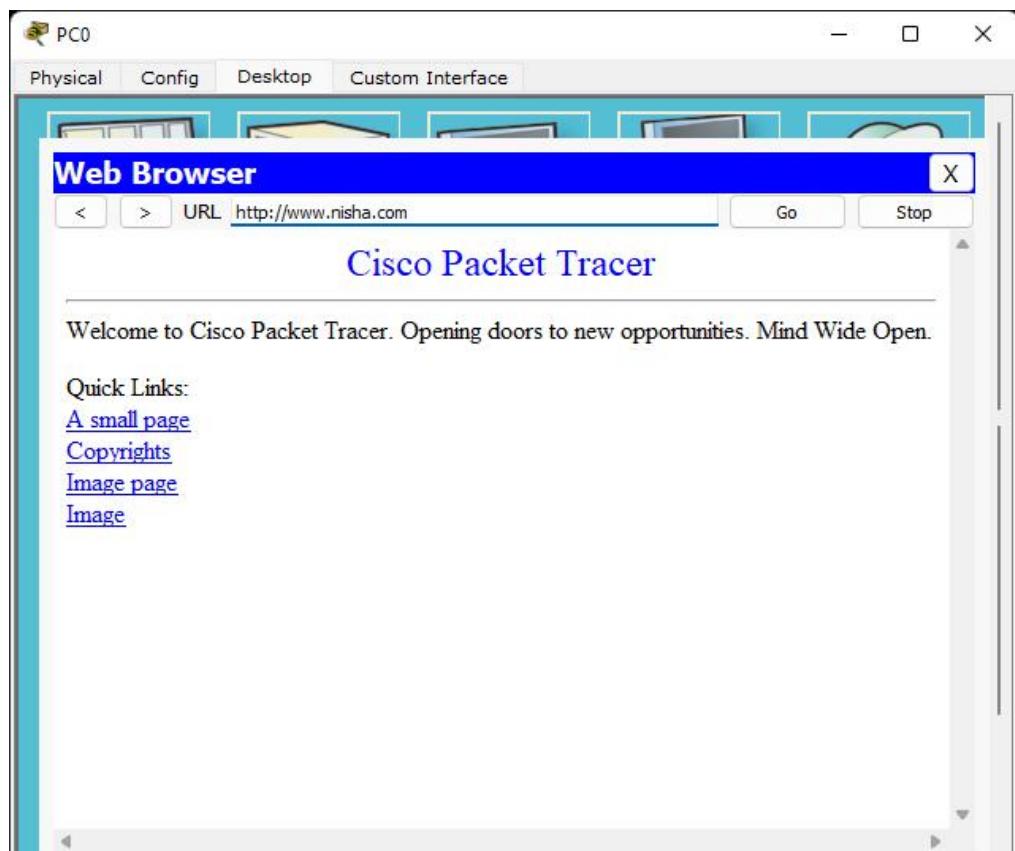
## Screenshots:



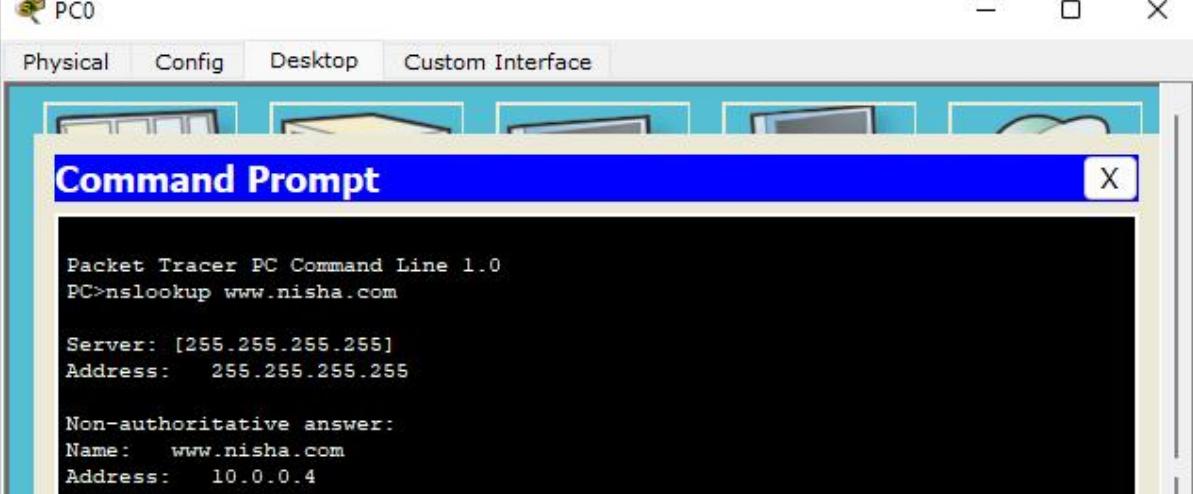
Topology



## Adding web server



## Accessing the web server from the end device

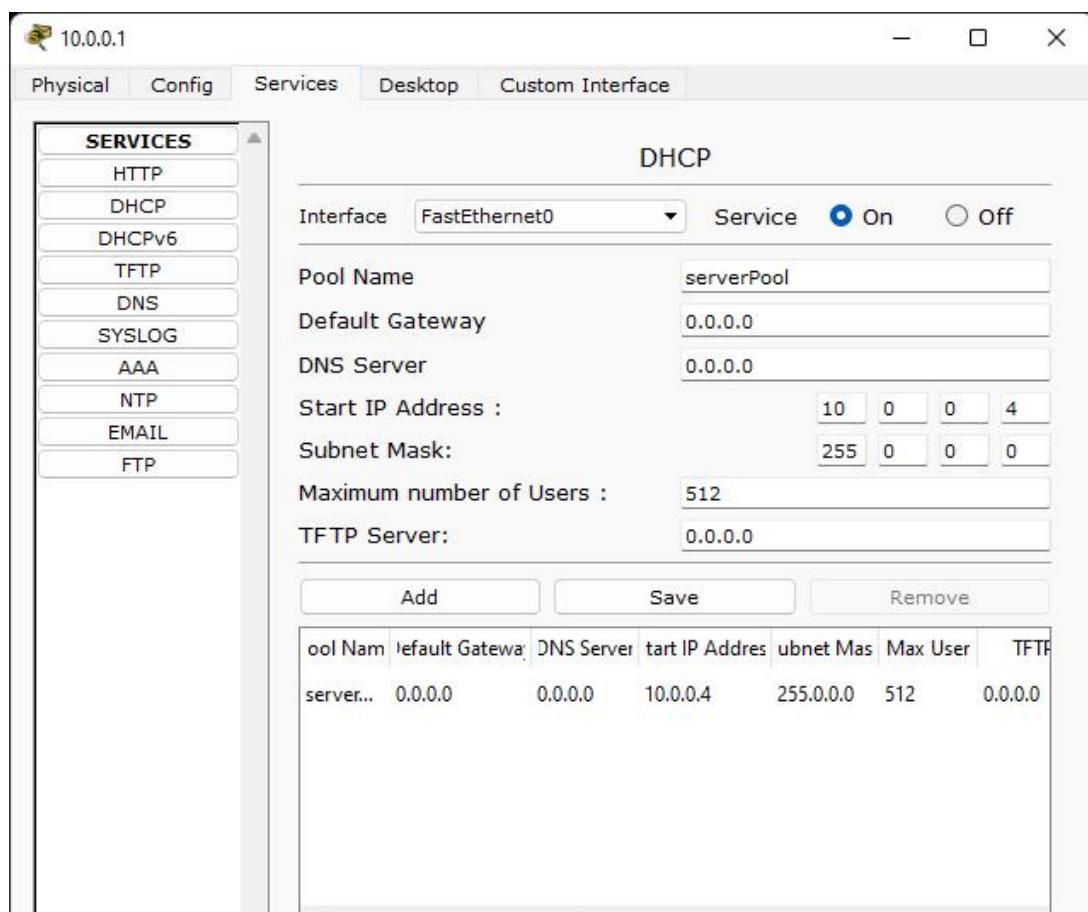


```
Packet Tracer PC Command Line 1.0
PC>nslookup www.nisha.com

Server: [255.255.255.255]
Address: 255.255.255.255

Non-authoritative answer:
Name: www.nisha.com
Address: 10.0.0.4
```

Nslookup for the web server created



Configuring DHCP

## Cycle 2:

### Program1:

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

#### Code:

```
import java.util.*;
class PRO7
{
 void div(int a[],int k)
 { int gp[]={1,0,0,0,1,0,0,0,0,0,1,0,0,0,0,1}; //generating polynomial =
X^16 + x^12 + x^5 + 1
 int count=0;
 for(int i=0;i<k;i++)
 {
 if(a[i]==gp[0])
 {
 for(int j=i;j<17+i;j++)
 {
 a[j]=a[j]^gp[count++];
 }
 count=0;
 }
 }
 }
 public static void main(String args[])
 {
 int a[]=new int[50];
 int b[]=new int[50];
 int len,k;
 PRO7 ob=new PRO7(); //creating an object of class PRO7
 System.out.println("Enter the length of Data Frame:");
 Scanner scan=new Scanner(System.in); //Creating an object to invoke
Scanner Function to read objects
 len=scan.nextInt(); //reads the length of Data or Message to be sent
 int flag=0; //indication for the data generated and received
are same or not.
 System.out.println("Enter the Message:");
 for(int i=0;i<len;i++) //iteration to accept input (the data / Message).
 {
 a[i]=scan.nextInt();
 }
 for(int i=0;i<16;i++) //adding zeros to the string
```

```

{
 a[len++]=0;
}

k=len-16; //retreieving the original data word length

for(int i=0;i<len;i++)
{
 b[i]=a[i];
}
ob.div(a,k); //calling an function to use CRC-CCITT 16 bits
for(int i=0;i<len;i++)
a[i]=a[i]^b[i]; //produces data transmion bits
System.out.println("Data to be transmitted: "); //prints data to be
transmitted
for(int i=0;i<len;i++)
{
 System.out.print(a[i]+" ");
}
System.out.println();
System.out.println("Enter the Reveived Data: "); //Prompt enter the
data received
for(int i=0;i<len;i++)
{
 a[i]=scan.nextInt();
}
ob.div(a, k); //checkes with CRC-CCITT 16 bit. "Note not compare "
for(int i=0;i<len;i++)
{
 if(a[i]!=0)
 {
 flag=1;
 break;
 }
}
if(flag==1) //prints weather received data is correct or not.
System.out.println("error in data");
else
System.out.println("no error");
scan.close();
}
}

```

**Output:**

```
Enter the length of Data Frame:
7
Enter the Message:
1 0 1 0 1 0 1
Data to be transmitted:
1 0 1 0 1 0 1 0 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0
Enter the Reveived Data:
1 0 1 0 1 0 1 0 1 0 0 1 0 1 0 0 1 0 1 0 0 0 0
error in data
```

```
Enter the length of Data Frame:
7
Enter the Message:
1 0 1 0 1 0 1
Data to be transmitted:
1 0 1 0 1 0 1 0 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0
Enter the Reveived Data:
1 0 1 0 1 0 1 0 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0
no error
```

## Program 2:

Write a program for a distance vector algorithm to find a suitable path for transmission.

### Code:

```
import java.util.*;
```

```
class DVR {
 static int graph[][], via[], rt[][], v, e;

 public static void main(String args[])
 { Scanner sc = new
 Scanner(System.in);
 System.out.println("Please enter the number of Vertices: ");
 v = sc.nextInt();

 System.out.println("Please enter the number of Edges: ");
 e = sc.nextInt();

 graph = new int[v][v];
 via = new int[v][v];
 rt = new int[v][v];
 for (int i = 0; i < v; i++)
 for (int j = 0; j < v; j++)
 if (i == j)
 graph[i][j] = 0;
 else
 graph[i][j] = 9999;
 }

 for (int i = 0; i < e; i++) {
 System.out.println("Please enter data for Edge " + (i + 1) + ":");
 System.out.print("Source: ");
 int s = sc.nextInt();
 s--;
 System.out.print("Destination: ");
 int d = sc.nextInt();
 d--;
 System.out.print("Cost: ");
```

```

 int c = sc.nextInt();
 graph[s][d] = c;
 graph[d][s] = c;
 }

 dvr_calc_disp("The initial Routing Tables are: ");

 System.out.print("Please enter the Source Node for the edge whose cost has
changed: ");
 int s = sc.nextInt();
 s--;
 System.out.print("Please enter the Destination Node for the edge whose cost
has changed: ");
 int d = sc.nextInt();
 d--;
 System.out.print("Please enter the new cost: ");
 int c = sc.nextInt();
 graph[s][d] = c;
 graph[d][s] = c;

 dvr_calc_disp("The new Routing Tables are: ");
 sc.close();
}

static void dvr_calc_disp(String message)
{System.out.println();
init_tables();
update_tables();
System.out.println(message);
print_tables();
System.out.println();
}

static void update_table(int source)
{for (int i = 0; i < v; i++) {
 if (graph[source][i] != 9999)
 {int dist = graph[source][i];
 for (int j = 0; j < v; j++) {
 int inter_dist = rt[i][j];
 if (via[i][j] == source)
 inter_dist = 9999;
 if (dist + inter_dist < rt[source][j])
 {rt[source][j] = dist + inter_dist;
 via[source][j] = i;
 }
 }
 }
}

```

```

 }
 }
}

static void update_tables()
{
 int k = 0;
 for (int i = 0; i < 4 * v; i++)
 {update_table(k);
 k++;
 if (k == v)
 k = 0;
 }
}

static void init_tables()
{
 for (int i = 0; i < v; i++)
 {
 for (int j = 0; j < v; j++)
 {if (i == j) {
 rt[i][j] = 0;
 via[i][j] = i;
 } else {
 rt[i][j] = 9999;
 via[i][j] = 100;
 }
 }
}
}

static void print_tables()
{
 for (int i = 0; i < v; i++)
 {
 for (int j = 0; j < v; j++)
 {System.out.print("Dist: " + rt[i][j] + " ");
 }
 System.out.println();
 }
}
}

```

**Output:**

```
Please enter the number of Vertices:
```

```
4
```

```
Please enter the number of Edges:
```

```
5
```

```
Please enter data for Edge 1:
```

```
Source: 1
```

```
Destination: 2
```

```
Cost: 1
```

```
Please enter data for Edge 2:
```

```
Source: 1
```

```
Destination: 3
```

```
Cost: 3
```

```
Please enter data for Edge 3:
```

```
Source: 2
```

```
Destination: 3
```

```
Cost: 1
```

```
Please enter data for Edge 4:
```

```
Source: 2
```

```
Destination: 4
```

```
Cost: 1
```

```
Please enter data for Edge 5:
```

```
Source: 3
```

```
Destination: 4
```

```
Cost: 4
```

```
The initial Routing Tables are:
```

```
Dist: 0 Dist: 1 Dist: 2 Dist: 2
```

```
Dist: 1 Dist: 0 Dist: 1 Dist: 1
```

```
Dist: 2 Dist: 1 Dist: 0 Dist: 2
```

```
Dist: 2 Dist: 1 Dist: 2 Dist: 0
```

```
Please enter the Source Node for the edge whose cost has changed: 2
```

```
Please enter the Destination Node for the edge whose cost has changed: 4
```

```
Please enter the new cost: 10
```

```
The new Routing Tables are:
```

```
Dist: 0 Dist: 1 Dist: 2 Dist: 6
```

```
Dist: 1 Dist: 0 Dist: 1 Dist: 5
```

```
Dist: 2 Dist: 1 Dist: 0 Dist: 4
```

```
Dist: 6 Dist: 5 Dist: 4 Dist: 0
```

### Program 3:

Implement Dijkstra's algorithm to compute the shortest path for a given topology.

#### Code:

```
class ShortestPath {
 static final int V = 9;
 int minDistance(int dist[], Boolean sptSet[])
 {
 int min = Integer.MAX_VALUE, min_index = -1;

 for (int v = 0; v < V; v++)
 if (sptSet[v] == false && dist[v] <= min)
 {min = dist[v];
 min_index = v;
 }

 return min_index;
 }
 void printSolution(int dist[])
 {
 System.out.println(
 "Vertex \t Distance from Source");
 for (int i = 0; i < V; i++)
 System.out.println(i + " \t " + dist[i]);
 }
 void dijkstra(int graph[][], int src)
 {
 int dist[] = new int[V];
 Boolean sptSet[] = new Boolean[V];
 for (int i = 0; i < V; i++) {
 dist[i] = Integer.MAX_VALUE;
 sptSet[i] = false;
 }
 dist[src] = 0;
 for (int count = 0; count < V - 1; count++)
 {int u = minDistance(dist, sptSet);
 sptSet[u] = true;
 for (int v = 0; v < V; v++)
 if (!sptSet[v] && graph[u][v] != 0
 && dist[u] != Integer.MAX_VALUE
 && dist[u] + graph[u][v] < dist[v])
 dist[v] = dist[u] + graph[u][v];
 }
 }
}
```

```

 }
 printSolution(dist);
}
public static void main(String[] args)
{
 int graph[][]
 = new int[][] { { 0, 4, 0, 0, 0, 0, 0, 8, 0 },
 { 4, 0, 8, 0, 0, 0, 0, 11, 0 },
 { 0, 8, 0, 7, 0, 4, 0, 0, 2 },
 { 0, 0, 7, 0, 9, 14, 0, 0, 0 },
 { 0, 0, 0, 9, 0, 10, 0, 0, 0 },
 { 0, 0, 4, 14, 10, 0, 2, 0, 0 },
 { 0, 0, 0, 0, 2, 0, 1, 6 },
 { 8, 11, 0, 0, 0, 0, 1, 0, 7 },
 { 0, 0, 2, 0, 0, 0, 6, 7, 0 } };
 ShortestPath t = new ShortestPath();
 t.dijkstra(graph, 0);
}
}

```

**Output:**

| Vertex | Distance from Source |
|--------|----------------------|
| 0      | 0                    |
| 1      | 4                    |
| 2      | 12                   |
| 3      | 19                   |
| 4      | 21                   |
| 5      | 11                   |
| 6      | 9                    |
| 7      | 8                    |
| 8      | 14                   |

## Program 4:

Write a program for congestion control using Leaky bucket algorithm.

### Code:

```
import java.util.*;

class Leakybucket {
 public static void main(String[] args)
 {
 int no_of_queries, storage, output_pkt_size;
 int input_pkt_size, bucket_size, size_left, loss, sent;
 Scanner sc = new Scanner(System.in);
 storage = 0;
 loss = 0;
 sent = 0;
 System.out.println("Enter number of packets you are sending: ");
 no_of_queries = sc.nextInt();
 System.out.println("Enter the buffer size: ");
 bucket_size = sc.nextInt();
 for (int i = 0; i < no_of_queries; i++)
 {size_left = bucket_size - storage;
 System.out.println("Enter input packet size: ");
 input_pkt_size = sc.nextInt();
 System.out.println("Enter output packet size: ");
 output_pkt_size = sc.nextInt();
 if (input_pkt_size <= (size_left))
 {sent += 1;
 storage += input_pkt_size;
 }
 else {
 loss += 1;
 }
 System.out.println("Buffer size= " + storage
 + " out of bucket size= "
 + bucket_size);
 storage -= output_pkt_size;
 }
 System.out.println("Packets Lost: "+loss);
 System.out.println("Packets Sent: "+sent);
 sc.close();
 }
}
```

**Output:**

```
Enter number of packets you are sending:
3
Enter the buffer size:
7
Enter input packet size:
3
Enter output packet size:
1
Buffer size= 3 out of bucket size= 7
Enter input packet size:
6
Enter output packet size:
1
Buffer size= 2 out of bucket size= 7
Enter input packet size:
6
Enter output packet size:
1
Buffer size= 7 out of bucket size= 7
Packets Lost: 1
Packets Sent: 2
```

## **Program 5:**

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:**

**Client:**

```
import java.net.*;
import java.io.*;

class TCPC {
 public static void main(String[] args) throws Exception
 {Socket sock = new Socket("127.0.01", 4000);

 System.out.println("Enter the filename");

 BufferedReader keyRead = new BufferedReader(new
InputStreamReader(System.in));

 String fname = keyRead.readLine();

 OutputStream ostream = sock.getOutputStream();

 PrintWriter pwrite = new PrintWriter(ostream, true);

 pwrite.println(fname);

 InputStream istream = sock.getInputStream();

 BufferedReader socketRead = new BufferedReader(new
InputStreamReader(istream));

 String str;
 while ((str = socketRead.readLine()) != null)
 {System.out.println(str);
 }

 pwrite.close();
 socketRead.close();
 keyRead.close();
 sock.close();
}
}
```

**Server:**

```
import java.net.*;
import java.io.*;

class TCPS {
 public static void main(String[] args) throws Exception
 {ServerSocket sersock = new ServerSocket(4000);
 System.out.println("Server ready for connection");

 Socket sock = sersock.accept();

 System.out.println("Connection Is successful and waiting for chatting");

 InputStream istream = sock.getInputStream();

 BufferedReader fileRead = new BufferedReader(new
InputStreamReader(istream));

 String fname = fileRead.readLine();

 BufferedReader ContentRead = new BufferedReader(new FileReader(fname));

 OutputStream ostream = sock.getOutputStream();

 PrintWriter pwrite = new PrintWriter(ostream, true);

 String str;

 while ((str = ContentRead.readLine()) != null)

 {pwrite.println(str);

 }

 sock.close();
 sersock.close();
 pwrite.close();
 fileRead.close();
 ContentRead.close();
 }

}
```

**Output:**

**TCP Client:**

```
Enter the filename
sample.txt
sample file for execution
```

**TCP Server:**

```
Server ready for connection
Connection Is successful and waiting for chatting
```

## **Program 6:**

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:**

**Client:**

```
import java.io.*;
import java.net.*;
import java.net.InetAddress;

class UDPCClient {
 public static void main(String[] args) throws Exception
 { BufferedReader inFromUser = new
 BufferedReader(new
 InputStreamReader(System.in));

 DatagramSocket clientSocket = new DatagramSocket();

 InetAddress IPAddress = InetAddress.getByName("localhost");

 byte[] sendData = new byte[1024];
 byte[] receiveData = new byte[1024];

 System.out.println("Enter the sting to be converted in to Upper case");
 String sentence = inFromUser.readLine();

 sendData = sentence.getBytes();

 DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, IPAddress, 9876);

 clientSocket.send(sendPacket);

 DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);

 clientSocket.receive(receivePacket);

 String modifiedSentence = new String(receivePacket.getData());

 System.out.println("FROM SERVER:" + modifiedSentence);

 clientSocket.close();
 }
}
```

```
}
```

**Server:**

```
import java.net.*;
import java.net.InetAddress;

class UDPServer {
 public static void main(String args[]) throws Exception
 { DatagramSocket serverSocket = new
 DatagramSocket(9876);byte[] receiveData = new byte[1024];
 byte[] sendData = new byte[1024];
 while (true) {
 System.out.println("Server is Up");

 DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);

 serverSocket.receive(receivePacket);

 String sentence = new String(receivePacket.getData());

 System.out.println("RECEIVED:" + sentence);

 InetAddress IPAddress = receivePacket.getAddress();

 int port = receivePacket.getPort();

 String capitalizedSentence = sentence.toUpperCase();

 sendData = capitalizedSentence.getBytes();

 DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, IPAddress, port);

 serverSocket.send(sendPacket);
 }
 }
}
```

**Output:**

**UDP Client:**

```
Enter the sting to be converted in to Upper case
cn lab program for udp socket
FROM SERVER:CN LAB PROGRAM FOR UDP SOCKET
```

**UDP Server:**

```
Server is Up
RECEIVED:cn lab program for udp socket
Server is Up
```