

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**  
“JnanaSangama”, Belgaum -590014, Karnataka.



**LAB REPORT**  
**on**  
**Computer Networks**

*Submitted by*

**J S MEGHANA (1BM18CS039)**

*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**  
**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 “LAB COURSE **Computer Networks**” carried out by **J S MEGHANA (1BM18CS039)**, who is bonafide student of **B. M. S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the 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

<b>Sl. No.</b>	<b>Date</b>	<b>Experiment Title</b>	<b>Page No.</b>
1.	17/11/22	<b>Hub &amp; Switch Topology</b>	<b>1-3</b>
2.	17/11/22	<b>Configuring IP address to Routers</b>	<b>4-8</b>
3.	24/11/22	<b>Configuring Static route to the Router</b>	<b>9-16</b>
4.	1/12/22	<b>Configuring Default route to the Router</b>	<b>17-25</b>
5.	8/12/22	<b>Configuring DHCP within LAN</b>	<b>26-28</b>
6.	15/12/22	<b>Configuring RIP routing protocol in Routers</b>	<b>29-32</b>
7.	15/12/22	<b>Web Server &amp; DNS Demonstration</b>	<b>33-35</b>
8.	29/12/22	<b>Write a program for error detecting code using CRC-CCITT(16-bits)</b>	<b>36-40</b>
9.	5/1/23	<b>Dijkstra's Algorithms Implementation</b>	<b>41-44</b>
10.	13/1/23	<b>Distance Vector Algorithm Implementation</b>	<b>45-48</b>
11.	30/1/23	<b>Leaky Bucket Algorithm Implementation</b>	<b>49-52</b>
12.	30/1/23	<b>TCP/IP Socket Program</b>	<b>53-56</b>
13.	30/1/23	<b>UDP Socket Program</b>	<b>57-60</b>

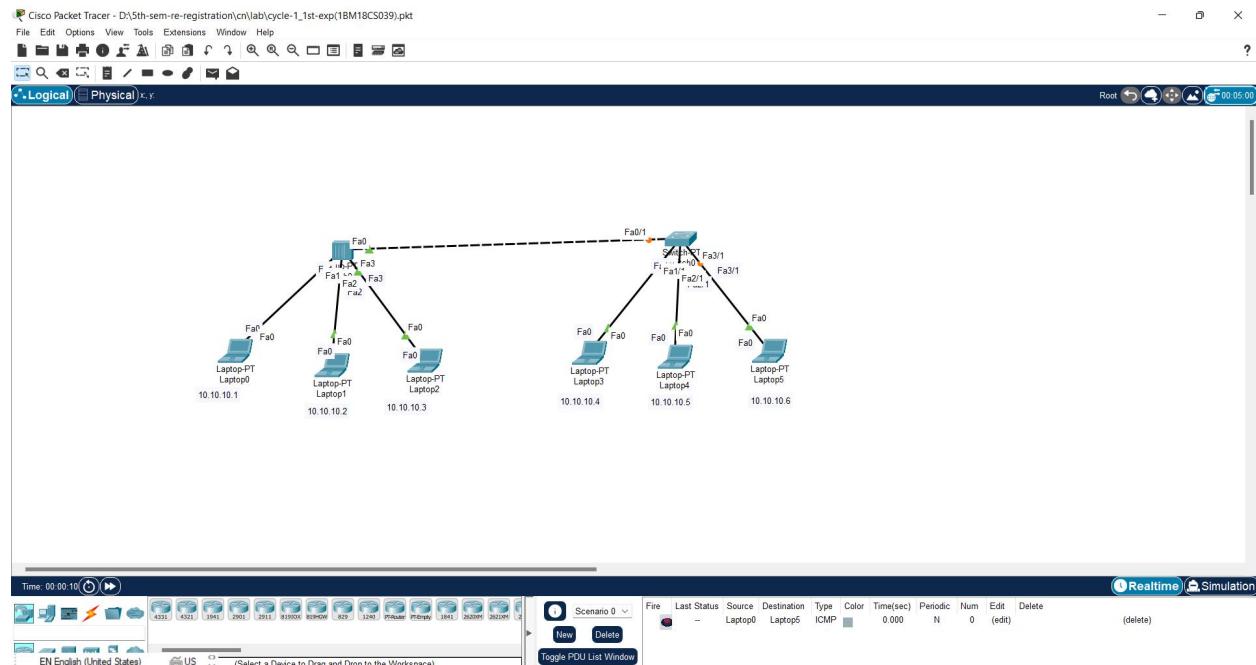
# Experiment No: 1

## Cycle-1

### Aim of the experiment:

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

### Topology:



## Procedure:

17/11/2022 Atm-1

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

Topology

TTL - Time To Live

Procedure: Hub

- 1) Hubs  $\rightarrow$  Generic (1)
- 2) End Devices  $\rightarrow$  Generic (desktop) (4 devices)
- 3) Connect the device using Blackwin
- 4) Configure the IP address as 10.0.0.1, 10.0.0.2, 10.0.0.3, 10.0.0.4 to each device.
- 5) Next in simulation mode add PDU from src  $\rightarrow$  dest & click on Auto Computer/ play.
- 6) we see that PDUs are being sent to all the devices but only to the concerned destination. only particular device output accept it rest all reject it.

Switch

1) From device type selection box, selected 4 devices by clicking end devices.

2) Connect in the similar manner as done for the Hub.

Routing

$\Rightarrow$  ping 10.0.0.4

$\Rightarrow$  pinging 10.0.0.4 with 32 bytes of data

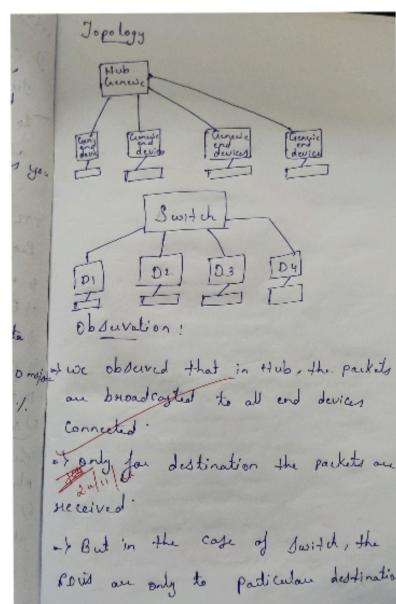
$\Rightarrow$  Reply from 10.0.0.4 bytes: 32 times + 0ms

$\Rightarrow$  Packets: Sent=4, Received=4, Lost=0%.

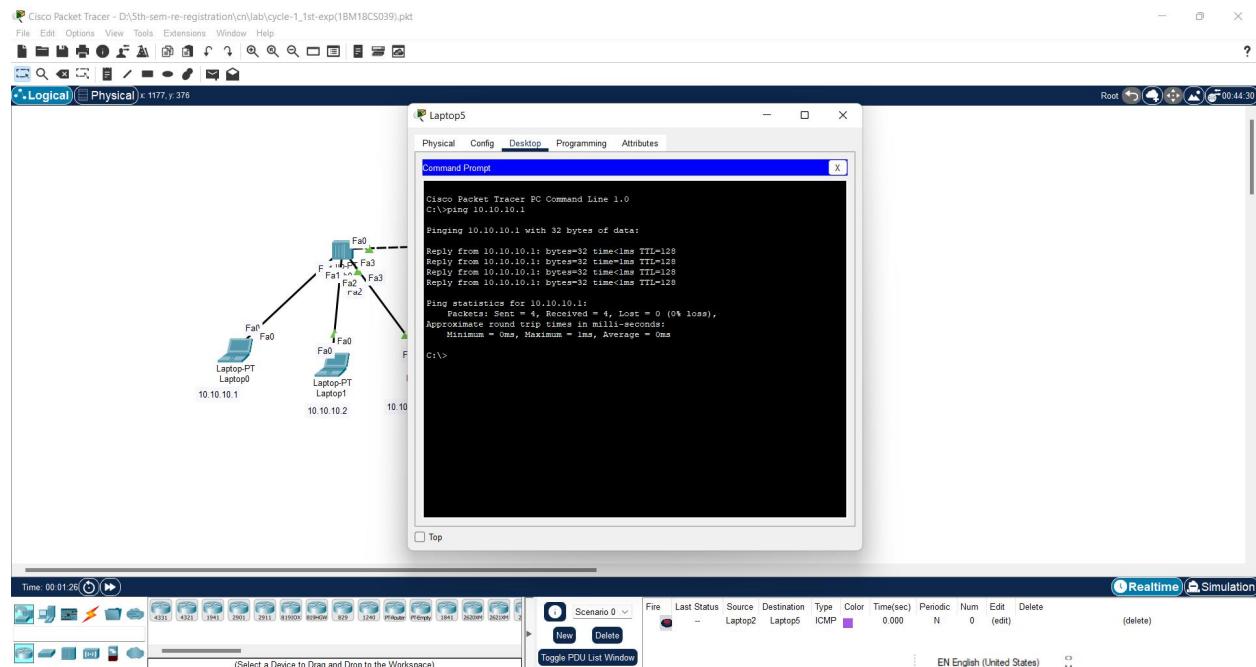
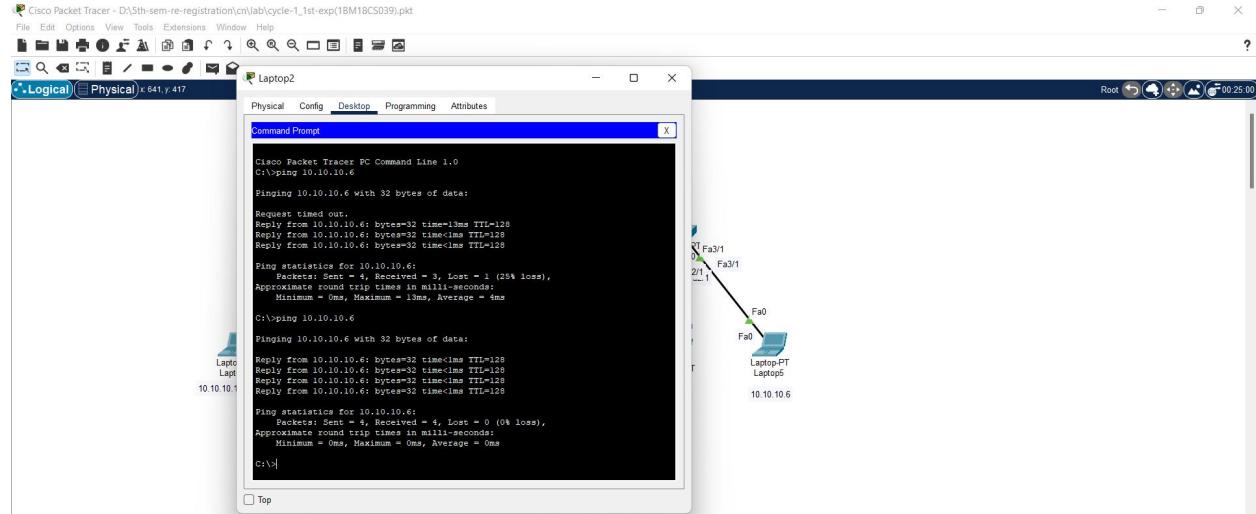
$\Rightarrow$  Round Trip time minimums and, min = 0ms, Avg = 0ms

$\Rightarrow$  In the case of hub, the packets are broadcasted to all end devices connected.

$\Rightarrow$  In the case of switch, the packets are only to particular destination.



## Snapshots of Output:



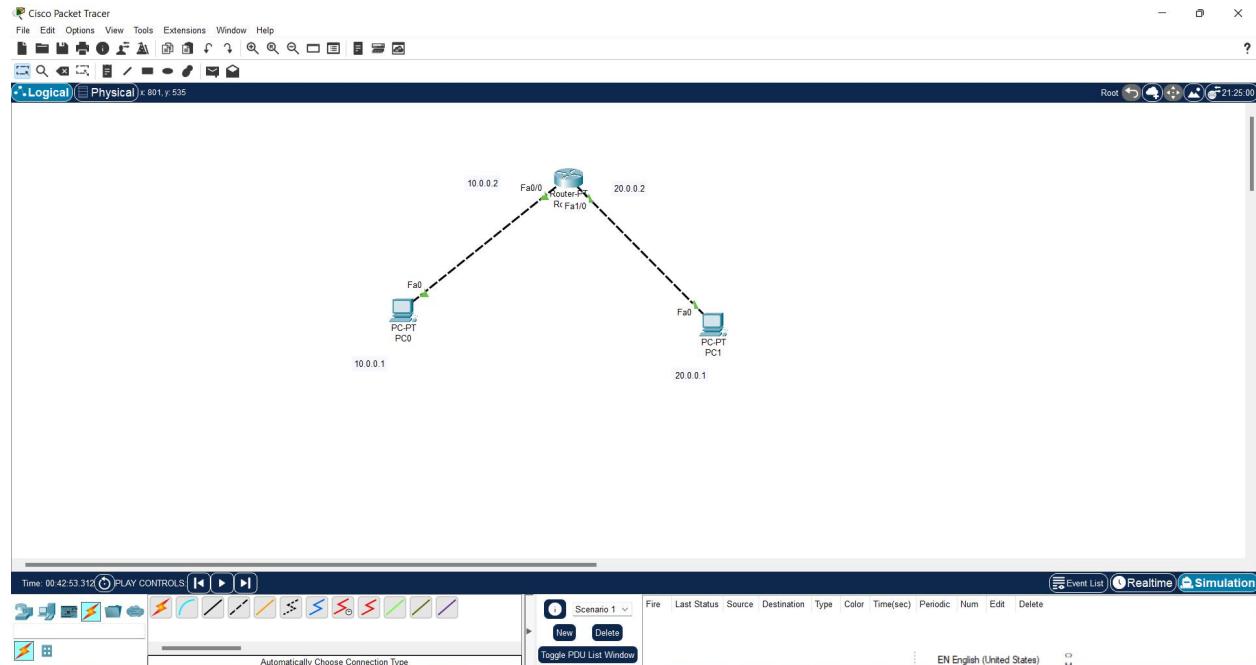
# Experiment No: 2

## Cycle-1

### Aim of the experiment:

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

### Topology:



## Procedure:



The screenshot shows a terminal window titled "Router0". The window has tabs for "Physical", "Config" (which is selected), and "Attributes". The title bar also includes standard window controls (minimize, maximize, close) and the text "IOS Command Line Interface".

The main pane displays the following text:

```
Would you like to enter the initial configuration dialog? [yes/no]: no
Press RETURN to get started!

Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface fastethernet0/0
Router(config-if)#ip address 10.0.0.2 255.0.0.0
Router(config-if)#no shutdown

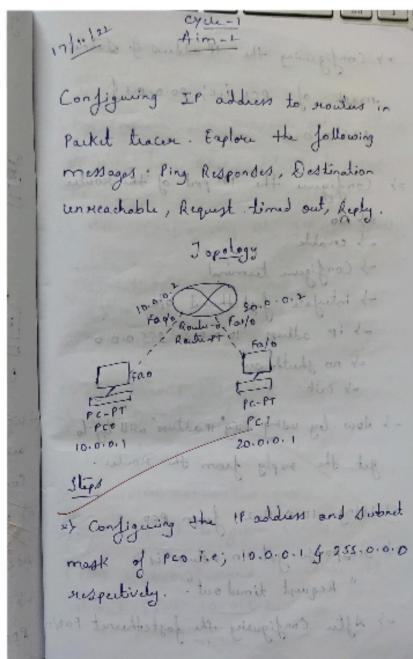
Router(config-if)#
$LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
$LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
Router(config-if)#exit
Router(config)#interface fastethernet1/0
Router(config-if)#ip address 20.0.0.2 255.0.0.0
Router(config-if)#no shutdown

Router(config-if)#
$LINK-5-CHANGED: Interface FastEthernet1/0, changed state to up
$LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/0, changed state to up
Router(config-if)#exit
Router(config)#

Router con0 is now available

Press RETURN to get started.
```

At the bottom right of the terminal window, there are "Copy" and "Paste" buttons. Below the terminal window, there is a toolbar with a "Top" button and language settings indicating "EN English (United States)".



→ Configuring the IP address & Subnet mask of PC1 i.e; 20.0.0.1 & 255.0.0.0.

→ Configure the 1st part of the router

- No shutdown
- enable
- Configure terminal
- interface fastethernet 0/0
- ip address 10.0.0.2 255.0.0.0
- no shutdown
- exit

→ Now by using "Ping" IP address will try to get the reply from the router.

→ Pinging 10.0.0.2 from PCo.

→ Response got in return is "Request timed out".

→ After Configuring the fastethernet fast0

→ Now tried pinging from PCo to router again

→ Ping, 10.0.0.2, then response got in return

or "Reply from 10.0.0.2: bytes=32 time=1ms TTL=255

→ Now Configuring the 2nd part of the router

- interface fastethernet 1/0
- ip address 10.0.0.2 255.0.0.0
- no shutdown
- exit

→ Now tried pinging 20.0.0.2 from PC1

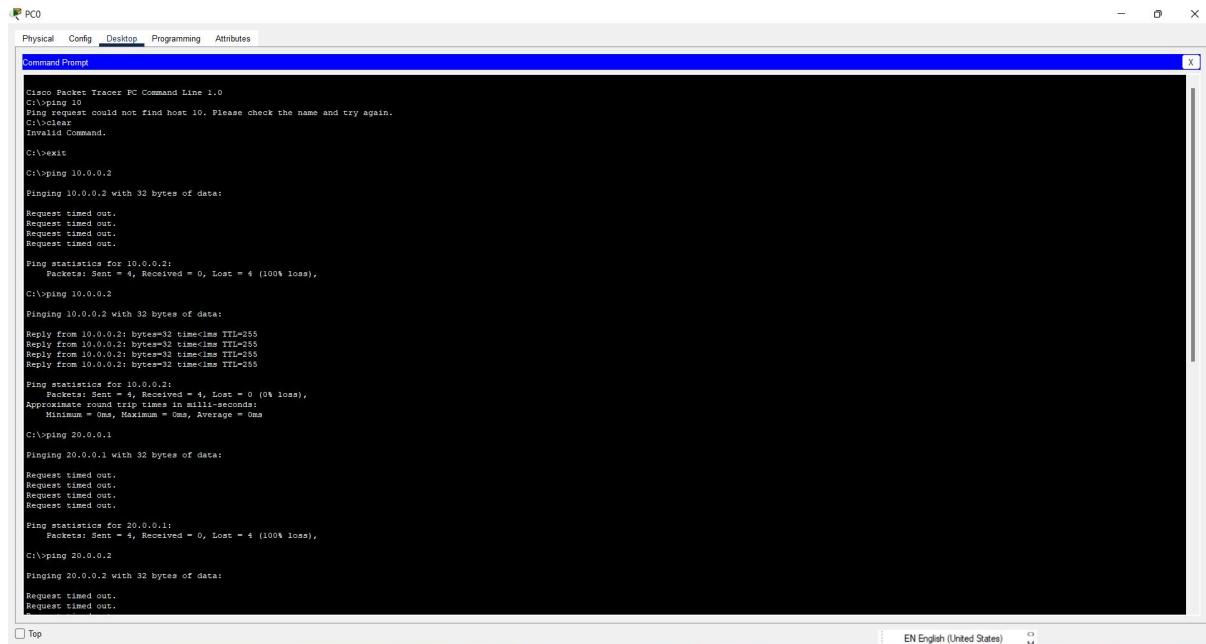
→ response got in return is "Request timed out".

"Reply from 20.0.0.2: bytes=32 time=1ms TTL=255

→ then tried to from PC1 to PCo (ping 10.0.0.1) and to router as well to 1st part (ping 10.0.0.2).

↳ For both the ping messages I got  
 "request timed out".  
 ↳ Then configured the gateway of  
 both the PCs of other ports of the  
 router for PC0 (10.0.0.2) and PC1  
 (20.0.0.2).  
 ↳ Then tried to and got the reply from  
 both the PCs and from both  
 ports.  
 ↳ After this sent a simple RDU from  
 PC0 → Router 0 → PC1 → Router 0 →  
 PC0 → Router 0 → Router 0 → PC1 → PC1 →  
 Router 0 → PC0 (acknowledgment)  
 ↳ 17/11/22 10:38 max at least out for  
 this destination at 6.13 (10.0.0.03 6.13)  
 · (10.0.0.01 ping) later at

## Snapshots of Output:



```

PC0
Physical Config Desktop Programming Attributes
Command Prompt

Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10
Ping request could not find host 10. Please check the name and try again.
C:\>clear
Invalid Command.
C:\>exit

C:\>ping 10.0.0.2
Pinging 10.0.0.2 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.0.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>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),
C:\>ping 20.0.0.2
Pinging 20.0.0.2 with 32 bytes of data:
Request timed out.
Request timed out.

```

**PC0**

Physical Config Desktop Programming Attributes

Command Prompt

```
C:\>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),
C:\>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),
C:\>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

Ping statistics for 20.0.0.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Reply from 20.0.0.1: bytes=32 time<1ms 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
C:\>
```

Top EN English (United States)

**PC1**

Physical Config Desktop Programming Attributes

Command Prompt

```
Administrator: Pocket Tester PC Command Line 1.0
C:\>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),
C:\>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

Ping statistics for 20.0.0.2:
    Packets: Sent = 4, Received = 0, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\>ping 10.0.0.2
Pinging 10.0.0.2 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\>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=255

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 0, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time<1ms TTL=127

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

Top EN English (United States)

**PC1**

Physical Config Desktop Programming Attributes

Command Prompt

```
C:\>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\>ping 10.0.0.2
Pinging 10.0.0.2 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\>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=255

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 0, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time<1ms TTL=127

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

Top EN English (United States)

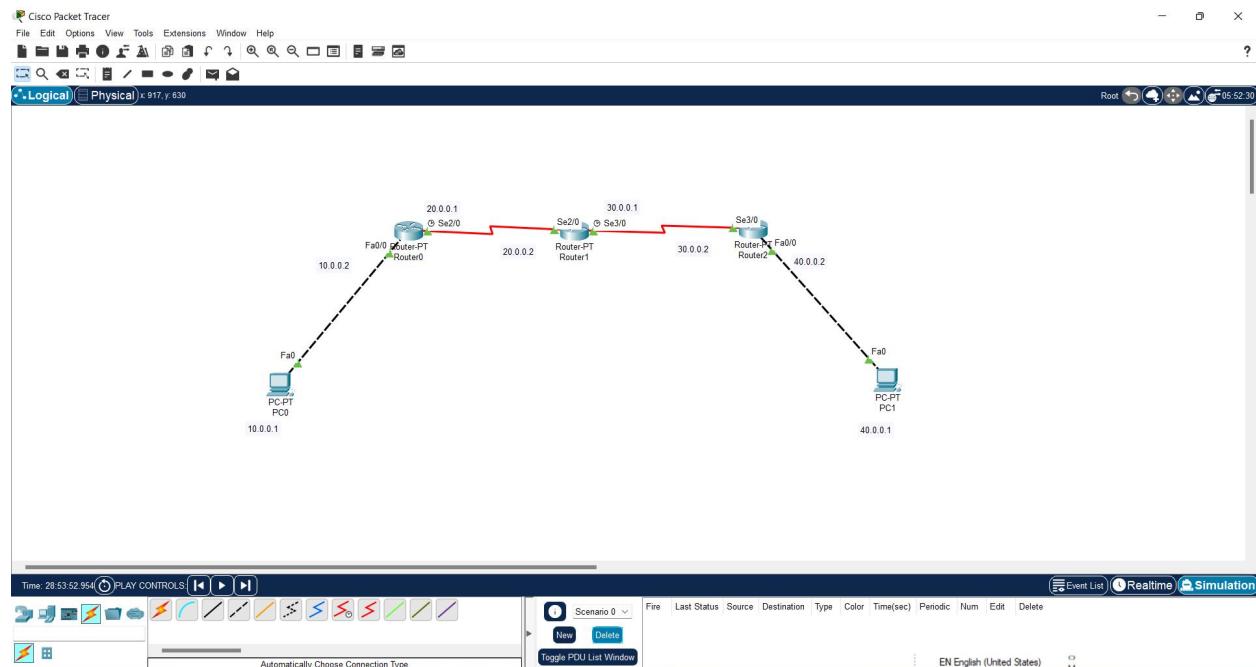
# Experiment No: 3

## Cycle-1

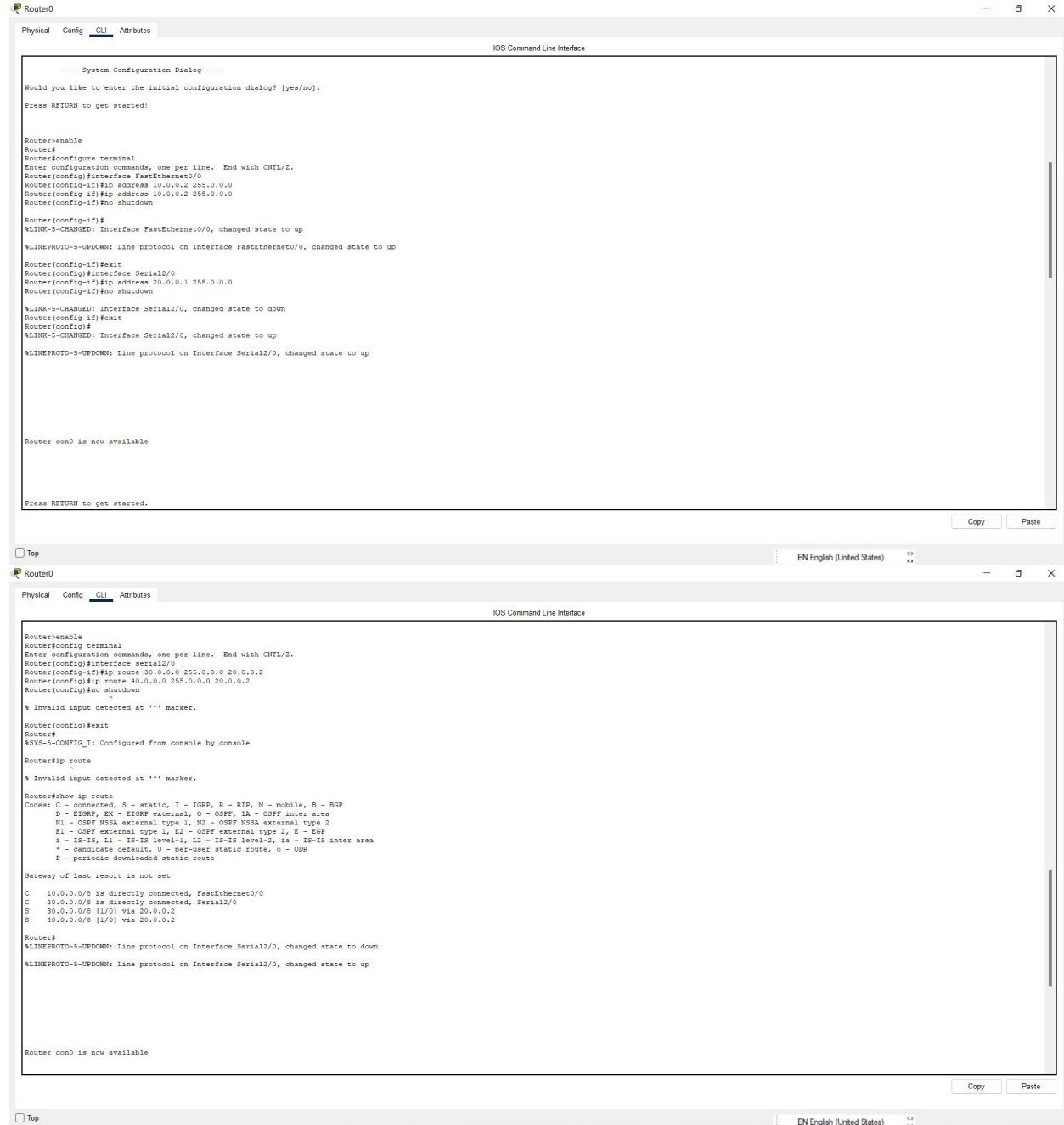
### Aim of the experiment:

Configuring Static route to the Router

### Topology:



## Procedure:



The image shows two screenshots of the Cisco IOS Command Line Interface (CLI) running on a Router. Both screenshots are titled "Router0" and show the "IOS Command Line Interface".

**Screenshot 1 (Top): Initial Configuration Dialog**

```

--- System Configuration Dialog ---
Would you like to enter the initial configuration dialog? [yes/no]:
Press RETURN to get started!

Router>enable
Router>configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 10.0.0.2 255.0.0.0
Router(config-if)#ip address 10.0.0.2 255.0.0.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial2/0, changed state to down
Router(config-if)#exit
Router(config)#
%LINK-4-CHANGED: Interface Serial2/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up

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

Router con0 is now available

Press RETURN to get started.

```

**Screenshot 2 (Bottom): Static Route Configuration**

```

Router>enable
Router>config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface serial2/0
Router(config-if)#ip address 20.0.0.2 255.0.0.0 20.0.0.2
Router(config-if)#ip route 40.0.0.0 255.0.0.0 20.0.0.2
Router(config)#no shutdown

% Invalid input detected at `''' marker.

Router(config)#exit
Router#
%SYS-6-CONFIG_I: Configured from console by console

Router#ip route
% Invalid input detected at `''' marker.

Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, E - EIGRP external, O - OSPF, 1B - OSPF Intra 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
      * - candidate default, 0 - 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
C 20.0.0.0/8 is directly connected, Serial2/0
S 30.0.0.0/24 via 20.0.0.2
S 40.0.0.0/24 via 20.0.0.2

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

Router con0 is now available

```

**Router1**

Physical Config **CLI** Attributes

IOS Command Line Interface

```

Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)interface Serial1/0
Router(config-if)#ip route 10.0.0.0 255.0.0.0 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 30.0.0.2
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

Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 10.0.0.0 255.0.0.0 20.0.0.1
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 30.0.0.2
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
S  40.0.0.0/8 [1/0] via 20.0.0.1

Router>config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 10.0.0.0 255.0.0.0 20.0.0.1
Router(config)#ip route 10.0.0.0 255.0.0.0 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 30.0.0.2
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
S  40.0.0.0/8 [1/0] via 20.0.0.1

Router#

```

Top

**Router1**

Physical Config **CLI** Attributes

IOS Command Line Interface

```

P - periodic downloaded static route

Gateway of last resort is not set
S  10.0.0.0/8 [1/0] via 30.0.0.2
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
S  40.0.0.0/8 [1/0] via 20.0.0.1

Router>config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 10.0.0.0 255.0.0.0 20.0.0.1
Router(config)#ip route 10.0.0.0 255.0.0.0 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 30.0.0.2
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
S  40.0.0.0/8 [1/0] via 20.0.0.1

Router#

```

Top

EN English (United States)

**Router2**

Physical Config **CLI** Attributes

IOS Command Line Interface

```

--- System Configuration Dialog ---
Would you like to enter the initial configuration dialog? [yes/no]:
Press RETURN to get started!

Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface Serial3/0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#no shutdown

%LINK-5-CHANGED: Interface Serial3/0, changed state to down
Router(config-if)#exit
Router(config)#
Router(config)#interface fastethernet0/0
Router(config-if)#ip address 40.0.0.2 255.0.0.0
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
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

Router con0 is now available

Press RETURN to get started.

```

Top EN English (United States)

**Router2**

Physical Config **CLI** Attributes

IOS Command Line Interface

```

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

Router>config terminal
% Invalid input detected at `''' marker.

Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 10.0.0.0 255.0.0.0 30.0.0.1
Router(config)#ip route 20.0.0.0 255.0.0.0 30.0.0.1
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

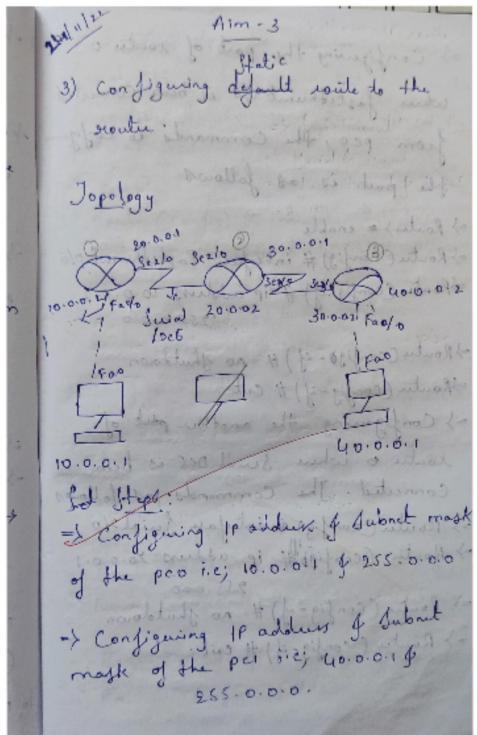
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, N1 - OSPF1, N2 - OSPF2
       IA - OSPFv3 inter-area route, N3 - OSPFv3 external type 1
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       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 30.0.0.1
S 20.0.0.0/8 [1/0] via 30.0.0.1
C 30.0.0.0/8 is directly connected, Serial3/0
C 40.0.0.0/8 is directly connected, FastEthernet0/0

Router#
Router#
Router>config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#

```

Top EN English (United States)



=> Configuring the port of router 0 where fastethernet 0/0 is been connected from pco, the commands to configure the port is as follows

=> Router > enable

=> 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

=> Configuring the another port of router 0 where serial 0/0 is been connected. The Commands as follows

=> Router (Config) # interface Serial 2/0

=> Router (Config) # ip address 20.0.0.1  
                          255.0.0.0

=> Router (Config-if) # no shutdown

=> Router (Config-if) # exit

=> Configuring the router 1 where serial 0/0 is been connected from router 0 to router. The command to configure the port as follows

=> enable

=> Router (Config) # interface Serial 2/0

=> Router (Config-if) # ip address 20.0.0.2  
                          255.0.0.0

=> Router (Config-if) # no shutdown

=> Router (Config-if) # exit

=> Configuring the another port of router 1 where serial 0/1 is been connected. The commands to configure is as follows.

=> Router > enable

=> Router (Config) # interface Serial 3/0

=> Router (Config-if) # ip address 20.0.0.1  
                          255.0.0.0

=> Router (Config-if) # no shutdown

=> Router (Config-if) # exit

-> Configuring the Router 2 where Serial S0/E connected from router 1.  
 The commands as follows  
 => Router & enable  
 => Router # config terminal  
 -> Router (config)# interface Serial 3/0  
 => Router (config-if)# ip address  
 30.0.0.2 255.0.0.0  
 => Router (config-if)# no shutdown  
 => Router (config-if)# exit  
 => Configuring the FastEthernet port  
 It is connected to one of the router 2 port & PC1.  
 The commands as follows  
 => 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  
 => After this Configured the "Default Gateway" of PC0 1.1.1; 10.0.0.2  
 => And configured the "Default Gateway" of PC1 1.1.1; 10.0.0.2  
 => Then tried pinging from PC0 to PC1  
 and PC1 to PC0.  
 -> PC0 = PC1  
 c:> ping 10.0.0.1  
 Pinging 10.0.0.1 with 32 Bytes of data:  
 reply from 10.0.0.2! Destination host  
 unreachable  
 Ping statistics for 10.0.0.1  
 packets: sent = 4, received = 0, lost = 4  
 (100% loss).

-> Then tried pinging from PC1 to PC0.  
 c:> ping 10.0.0.1  
 Replying ping to 10.0.0.1 with 32 bytes of  
 data:  
 Reply from 10.0.0.2! Destination host  
 unreachable.  
 Request timed out.  
 Ping statistics for 10.0.0.1:  
 packets: sent = 4, received = 0, lost = 4  
 (100% loss).  
 => So now Configuring all the routers  
 so that the communication can happen  
 by / through all the end devices and  
 network devices (PCs & routers respectively)

=> Configuring Router 0 for network  
 30.0.0.0 & 40.0.0.0. (as it is aware  
 of them)  
 The commands are as follows  
 (IP route Dest-N/W address, subnet mask  
 Router 0  
 not IP address).  
 => Router & enable  
 => Router # config terminal  
 => Router (config)# ip route 30.0.0.0  
 255.0.0.0 20.0.0.1  
 => Router (config)# ip route 40.0.0.0  
 255.0.0.0 20.0.0.2  
 => Router (config)# no shutdown  
 => Router (config)# exit  
 => Router # show ip route  
 c 10.0.0.0/8 is directly connected,  
 to FastEthernet 0/0  
 c 20.0.0.0/8 is directly connected,  
 to Serial 2/0.  
 via 30.0.0.0/3 [1/0] Via 20.0.0.2  
 via 40.0.0.0/3 [1/0] Via 20.0.0.2

→ Configuring Router 1 for switches:  
10.0.0.10 of 40000/0 Card 1 (back  
plane of them).

The commands are as follows:

Router 1

- Router & enable
- Router # config terminal
- Router (Config) # ip route 10.0.0.0  
355.0.0.0 30.0.0.2
- Router (Config) # ip route 10.0.0.0  
355.0.0.0 30.0.0.2
- Router (Config) # exit
- Router # show ip route
- 3 10.0.0.0/8 (C) via 30.0.0.2
- 3 30.0.0.0/8 is directly connected  
Serial 3/0
- 3 30.0.0.0/8 is directly connected  
Serial 3/0

→ Router 2 enable

→ Router # config terminal

→ Router (Config) # ip route 10.0.0.0  
355.0.0.0 30.0.0.1

→ Router (Config) # exit

→ Router # show ip route

→ 3 10.0.0.0/8 [110] via 30.0.0.1

→ 3 30.0.0.0/8 is directly connected  
Serial 3/0

→ 3 30.0.0.0/8 is directly connected  
Serial 3/0

→ Router 3 enable

→ Router # config terminal

→ Router (Config) # ip route 10.0.0.0  
355.0.0.0 30.0.0.2

→ Router (Config) # exit

→ Router # show ip route

→ 3 10.0.0.0/8 [110] via 30.0.0.2

→ 3 30.0.0.0/8 is directly connected  
Serial 3/0

- Configuring the Router 2 for  
natively 10.0.0.0 & 20.0.0.0  
(as it is not aware of them).  
The commands are
- Router # enable
- Router # config terminal
- Router (config) # ip route 10.0.0.0  
255.0.0.0 30.0.0.1
- Router (config) # ip route 20.0.0.0  
255.0.0.0 30.0.0.1
- Router (config) # exit
- Router # show ip route
  - 3 10.0.0.0/32 [10] via 30.0.0.1
  - 3 20.0.0.0/32 [10] via 30.0.0.1
  - C 30.0.0.0/32 is directly connected  
Serial 2/0
- C 40.0.0.0 is directly connected  
FastEthernet 0/0
- After configuring all the IP routes  
try pinging from PC0\_pc1 & PC0\_pc2  
The response will be as follows
- pc0 -> ping
- First ping
- City ping 10.0.0.1!
- Pinging 10.0.0.1 with 32 bytes of data:
- Request times out
- Reply from 10.0.0.1: bytes=32 time = 1ms  
TTL = 255
- Ping Statistics for 10.0.0.1:  
packets: Sent = 4, Received = 3, Lost = 1  
(25% loss),
- Approximate round trip times in milliseconds:

minimum = 0 miles, maximum = 40 miles  
 Average = 10 miles

→ found 1 second ping

→ clicking balloon

pinging balloon with 30 bytes of data

Reply from 100.0.0.1: bytes=30 time=1ms  
 T1 - 113

ping statistics for 100.0.0.1:  
 packets: sent 4, received 4, lost = 0 (0% loss)

minimum = 0 miles, maximum = 40 miles  
 Average = 10 miles

→ This time got successful reply from 10.0.0.1 with "0" loss.

→ PING -> 10.0.0.1

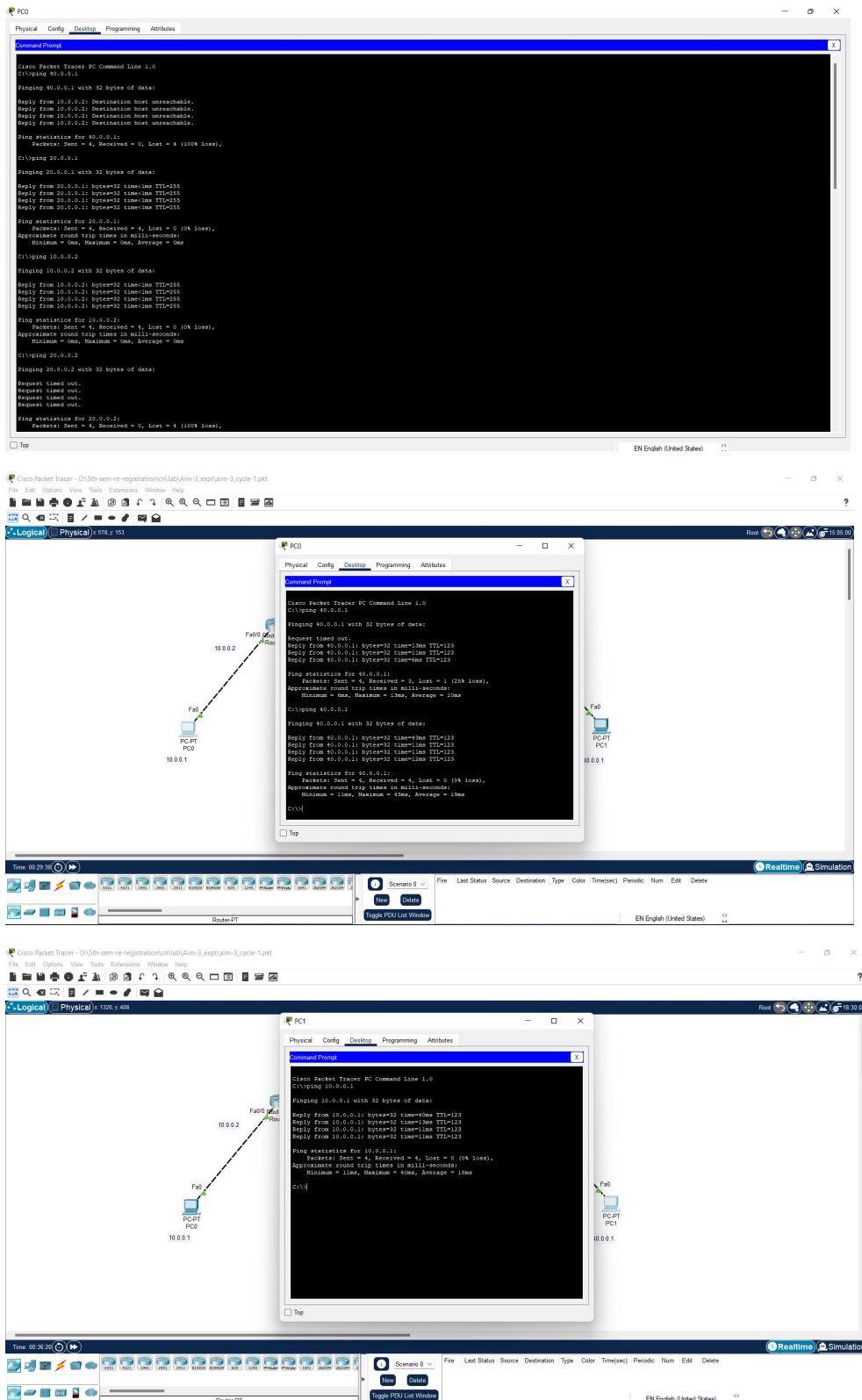
→ 10.0.0.1: bytes=30 time=1ms

Reply from 10.0.0.1: bytes=30 time=1ms  
 T1 - 113

ping statistics for 100.0.0.1:  
 packets: sent 4, received 4, lost = 0 (0% loss)

minimum = 0 miles, maximum = 40 miles  
 Average = 10 miles

## Snapshots of Output:



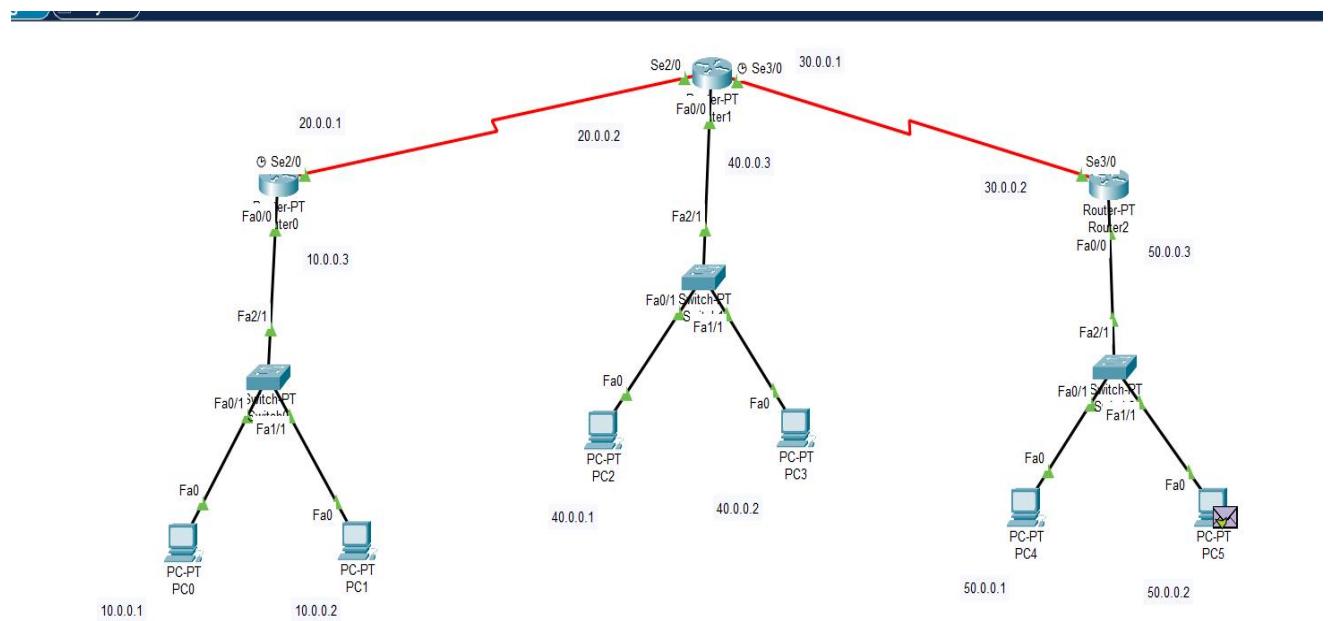
# Experiment No: 4

## Cycle-1

### Aim of the experiment:

Configuring default route to the Router

### Topology:



## Procedure:

Router1

Physical Config **CLI** Attributes

IOS Command Line Interface

```

Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 0.0.0.0 0.0.0.0 30.0.0.2
Router(config)#exit
Router#
*SYN-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 30.0.0.2 to network 0.0.0.0

C 20.0.0.0/8 is directly connected, Serial1/0
C 30.0.0.0/8 is directly connected, Serial3/0
C 40.0.0.0/8 is directly connected, FastEthernet0/0
S* 0.0.0.0/0 [1/0] via 30.0.0.2

Router#
```

Router con0 is now available

Press RETURN to get started.

Copy Paste

Router1

Physical Config **CLI** Attributes

IOS Command Line Interface

```

Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 0.0.0.0 0.0.0.0 20.0.0.1
Router(config)#exit
Router#
*SYN-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 30.0.0.2 to network 0.0.0.0

C 20.0.0.0/8 is directly connected, Serial1/0
C 30.0.0.0/8 is directly connected, Serial3/0
C 40.0.0.0/8 is directly connected, FastEthernet0/0
S* 0.0.0.0/0 [1/0] via 30.0.0.1
[1/0] via 20.0.0.1

Router#
```

Router con0 is now available

Press RETURN to get started.

Copy Paste

Top EN English (United States)

Router2

Physical Config **CLI** Attributes

IOS Command Line Interface

```

Router>enable
Router#
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface Serial3/0
Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#

```

Router con0 is now available

Press RETURN to get started.

Press RETURN to get started!

```

Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface Serial3/0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#no shutdown

```

**Copy** **Paste**

Router2

Physical Config **CLI** Attributes

IOS Command Line Interface

```

Router>enable
Router#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface Serial3/0
Router(config-if)#ip address 30.0.0.2 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface Serial3/0, changed state to up
Router(config-if)#exit
Router(config)#
%LINKPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up
Router(config)#interface fastethernet0/0
Router(config-if)#ip address 50.0.0.3 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINKPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
Router(config-if)#exit
Router(config)#

```

Router con0 is now available

Press RETURN to get started.

**Copy** **Paste**

Top ... EN English (United States) ☰

Router2

Physical Config **CLI** Attributes

IOS Command Line Interface

```

Router>
Router>enable
Router>config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 0.0.0.0 0.0.0.0 30.0.0.1
Router(config)#ip route 0.0.0.0 0.0.0.0 40.0.0.3
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router con0 is now available

Press RETURN to get started.

Router>enable
Router>config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

```

**Copy** **Paste**

Router2

Physical Config **CLI** Attributes

IOS Command Line Interface

```

Router con0 is now available

Press RETURN to get started.

Router>enable
Router>config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

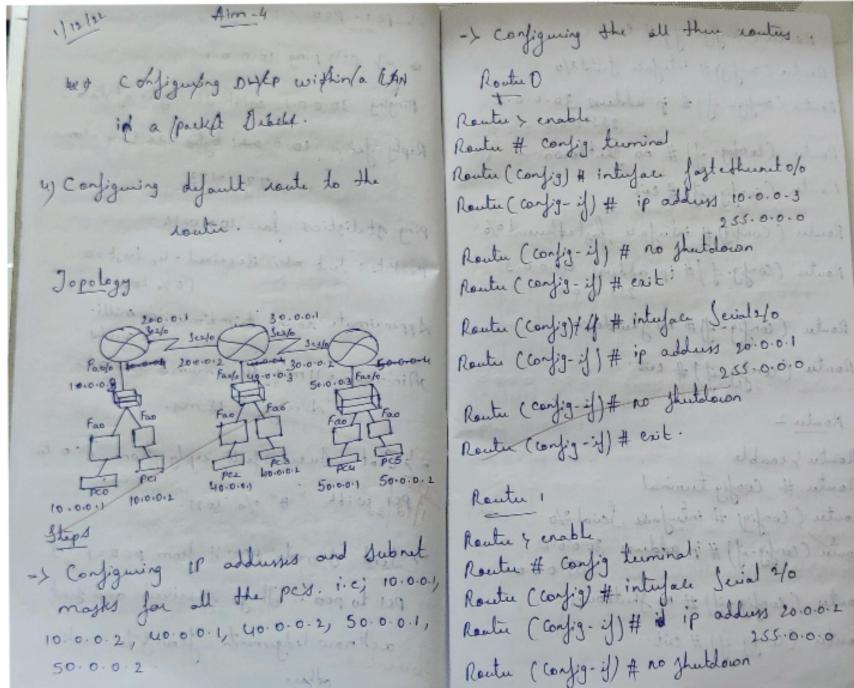
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       E1 - OSPF external type 1, E2 - OSPF external type 2
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       * - candidate default, + - per-user static route, o - ODR
       P - periodic downloaded static route
Gateway of last resort is 30.0.0.1 to network 0.0.0.0

C      30.0.0.0/8 is directly connected, Serial3/0
C      30.0.0.0/8 is directly connected, FastEthernet0/0
S*+    0.0.0.0/0 [1/0] via 30.0.0.1

Router#

```

**Copy** **Paste**



Router (config-if) # exit

Router (config) # interface serial 0/0

Router (config-if) # ip address 30.0.0.1

255.0.0.0

Router (config-if) # no shutdown

Router (config-if) # exit

Router (config) # interface fastethernet 0/0

Router (config-if) # ip address 10.0.0.3

255.0.0.0

Router (config-if) # no shutdown

Router (config-if) # exit

Router 2

Router > enable

Router # config terminal

Router (config) # interface serial 3/0

Router (config-if) # ip address 30.0.0.2

255.0.0.0

Router (config-if) # no shutdown

Router (config-if) # exit

→ Now configuring the default gateway of PCs i.e; for PC0 & PC1 (10.0.0.3)

for PC2 & PC3 (30.0.0.3) for PCs

(30.0.0.3).

→ Tried pinging from PCs to PC0 with

> ping 10.0.0.1

Pinging to 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1: Destination host unreachable.

→ Now Configuring all the routers (default route) for communication to happen from all.

<p><u>Router 0</u></p> <pre> Router &gt; enable Router # config terminal Router (config) # ip route 0.0.0.0 0.0.0.0 20.0.0.2 </pre> <p>Command is → ip route destination-network Subnet mask next-hop address</p> <p><u>Router 1</u></p> <pre> Router &gt; enable Router # config terminal Router (config) # ip route 0.0.0.0 0.0.0.0 20.0.0.1 </pre> <p><del>Router (config) # ip route 0.0.0.0 0.0.0.0 20.0.0.2</del></p> <p><del>Router (config) # exit</del></p> <p><del>Router # show ip route</del></p>	<p><u>Router 2</u></p> <pre> Router &gt; enable Router # config terminal Router (config) # ip route 0.0.0.0 0.0.0.0 30.0.0.1 </pre> <p><del>Router (config) # ip route 0.0.0.0 0.0.0.0 40.0.0.3.</del></p> <p><del>Router (config) # exit</del></p> <p><del>Router # show ip route</del></p> <p>→ New tried ping from pc1 to pc2. → ping 10.0.0.1</p> <p>pinging 10.0.0.1 with 32 bytes of data:</p> <p>Reply from 10.0.0.1 bytes=32 time=3ms TTL=125.</p> <p>Ping statistics for 10.0.0.1: packets: Sent 0/0, Received =0, Lost =0 (% loss), Approximate round trip time in milliseconds: Minimum =2ms, Maximum =25ms Average =13 ms.</p>
---	--

Observations

- Configuring default ip route ensures that the packet passes through the default route when no other route is available for an ip destination address.
- The simulation of sending a simple PDU from Source to destination (here from pc1 to pc2) shows the route taken by the ICMP packet.

~~\$ 12/22~~

~~1.0.0.1 ping~~

~~total bytes 32 time 10.0.0.1 bytes~~

~~loss = 0%, round-trip min = 10.0.0.1 max = 25ms avg = 13 ms~~

## Snapshots of Output:

PC0

Physical Config Desktop Programming Attributes

Command Prompt

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 50.0.0.2

Pinging 50.0.0.2 with 32 bytes of data:

Reply from 50.0.0.2: bytes=32 time=33ms TTL=125
Reply from 50.0.0.2: bytes=32 time=2ms TTL=125
Reply from 50.0.0.2: bytes=32 time=2ms TTL=125
Reply from 50.0.0.2: bytes=32 time=2ms TTL=125

Ping statistics for 50.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 33ms, Average = 9ms

C:\>
```

Top

PC3

Physical Config Desktop Programming Attributes

Command Prompt

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 40.0.0.3

Pinging 40.0.0.3 with 32 bytes of data:

Reply from 40.0.0.3: bytes=32 time=10ms TTL=255
Reply from 40.0.0.3: bytes=32 time<1ms TTL=255
Reply from 40.0.0.3: bytes=32 time<1ms TTL=255
Reply from 40.0.0.3: bytes=32 time<1ms TTL=255

Ping statistics for 40.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 10ms, Average = 2ms

C:\>ping 50.0.0.3

Pinging 50.0.0.3 with 32 bytes of data:

Reply from 50.0.0.3: bytes=32 time=2ms TTL=254
Reply from 50.0.0.3: bytes=32 time=2ms TTL=254
Reply from 50.0.0.3: bytes=32 time=30ms TTL=254
Reply from 50.0.0.3: bytes=32 time=30ms TTL=254

Ping statistics for 50.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 30ms, Average = 16ms

C:\>
```

Top

**PC4**

Physical Config Desktop Programming Attributes

Command Prompt

```

Cisco Packet Tracer PC Command Line 1.0
C:\pinging 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\pinging 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\pinging 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\pinging 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=1ms 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 = 1ms, Maximum = 15ms, Average = 1ms
C:\x\

 Rsp
EN English (United States)  

```

**PC5**

Physical Config **Desktop** Programming Attributes

Command Prompt

```

Cisco Packet Tracer PC Command Line 1.0
C:\pinging 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\pinging 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\pinging 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\pinging 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\pinging 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\pinging 10.0.0.1

 Top
EN English (United States)  

```

**PC5**

Physical Config Desktop Programming Attributes

Command Prompt

```

Request timed out.
Reply from 10.0.0.1: bytes=32 time=1ms TTL=128
Reply from 10.0.0.1: bytes=32 time=1ms TTL=128
Reply from 10.0.0.1: bytes=32 time=1ms TTL=128

Ping statistics for 10.0.0.1:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms
C:\pinging 40.0.0.1

C:\pinging 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:
Request timed out.
Reply from 40.0.0.1: bytes=32 time=1ms TTL=128
Reply from 40.0.0.1: bytes=32 time=1ms TTL=128
Reply from 40.0.0.1: bytes=32 time=1ms TTL=128

Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms
C:\pinging 40.0.0.3

Pinging 40.0.0.3 with 32 bytes of data:
Request timed out.
Reply from 40.0.0.3: bytes=32 time=1ms TTL=128
Reply from 40.0.0.3: bytes=32 time=1ms TTL=128
Reply from 40.0.0.3: bytes=32 time=1ms TTL=128

Ping statistics for 40.0.0.3:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms
C:\pinging 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=1ms 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 = 1ms, Maximum = 1ms, Average = 1ms
C:\x\

 Top
EN English (United States)  

```

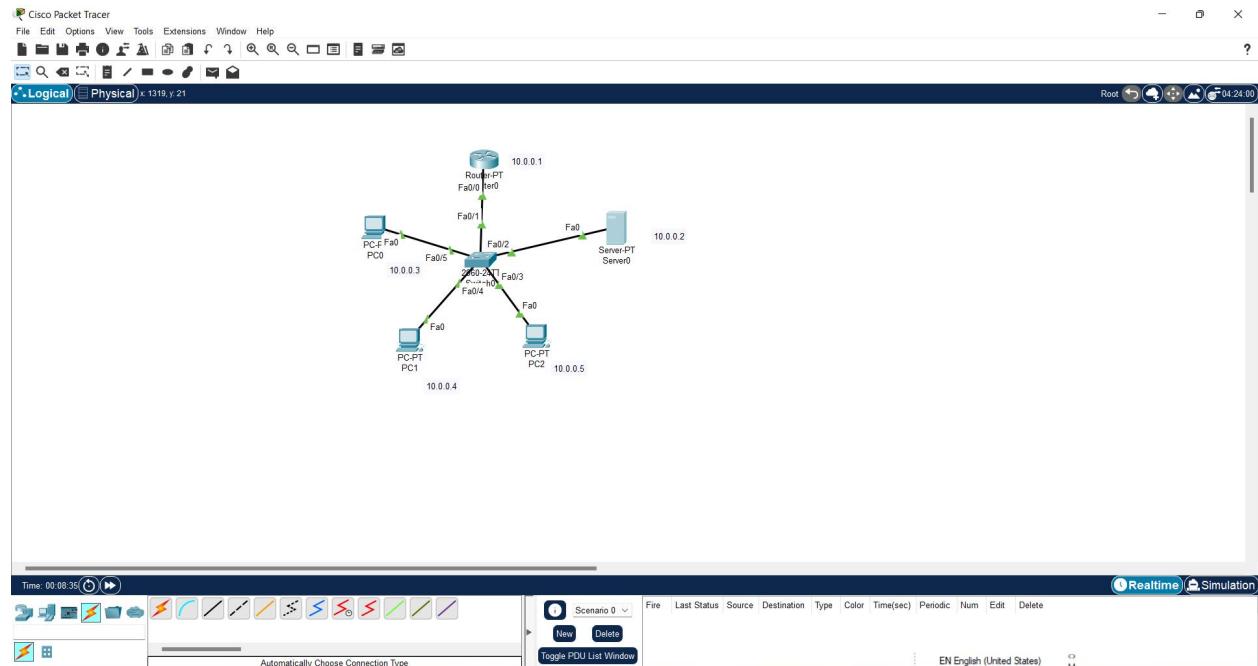
# Experiment No: 5

## Cycle-1

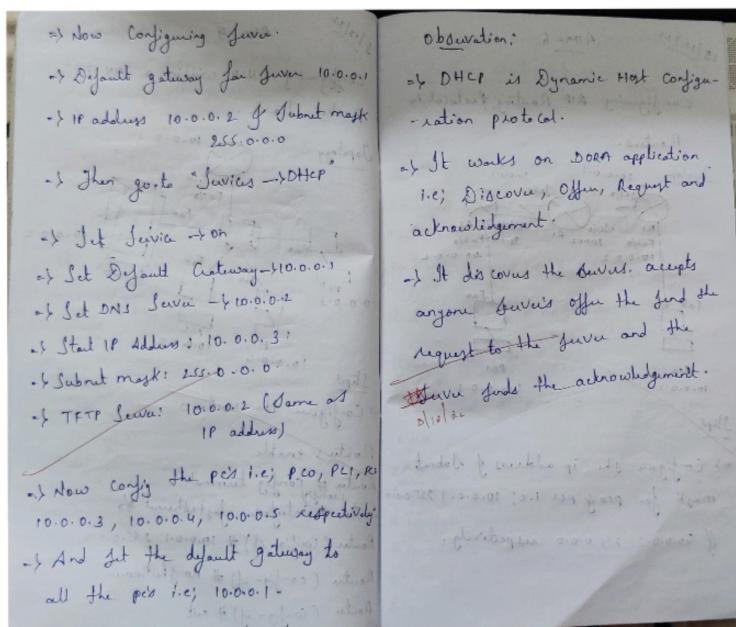
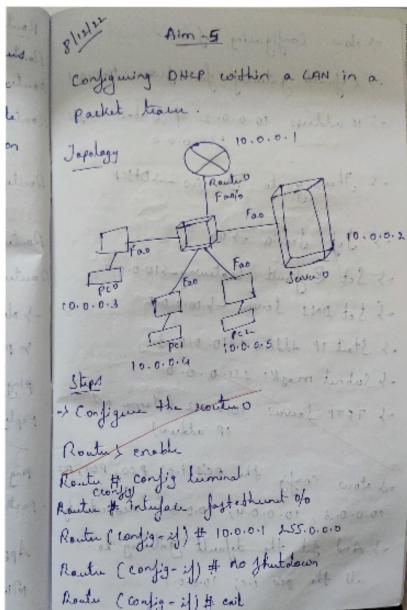
### Aim of the experiment:

Configuring DHCP within a LAN in a packet Tracer

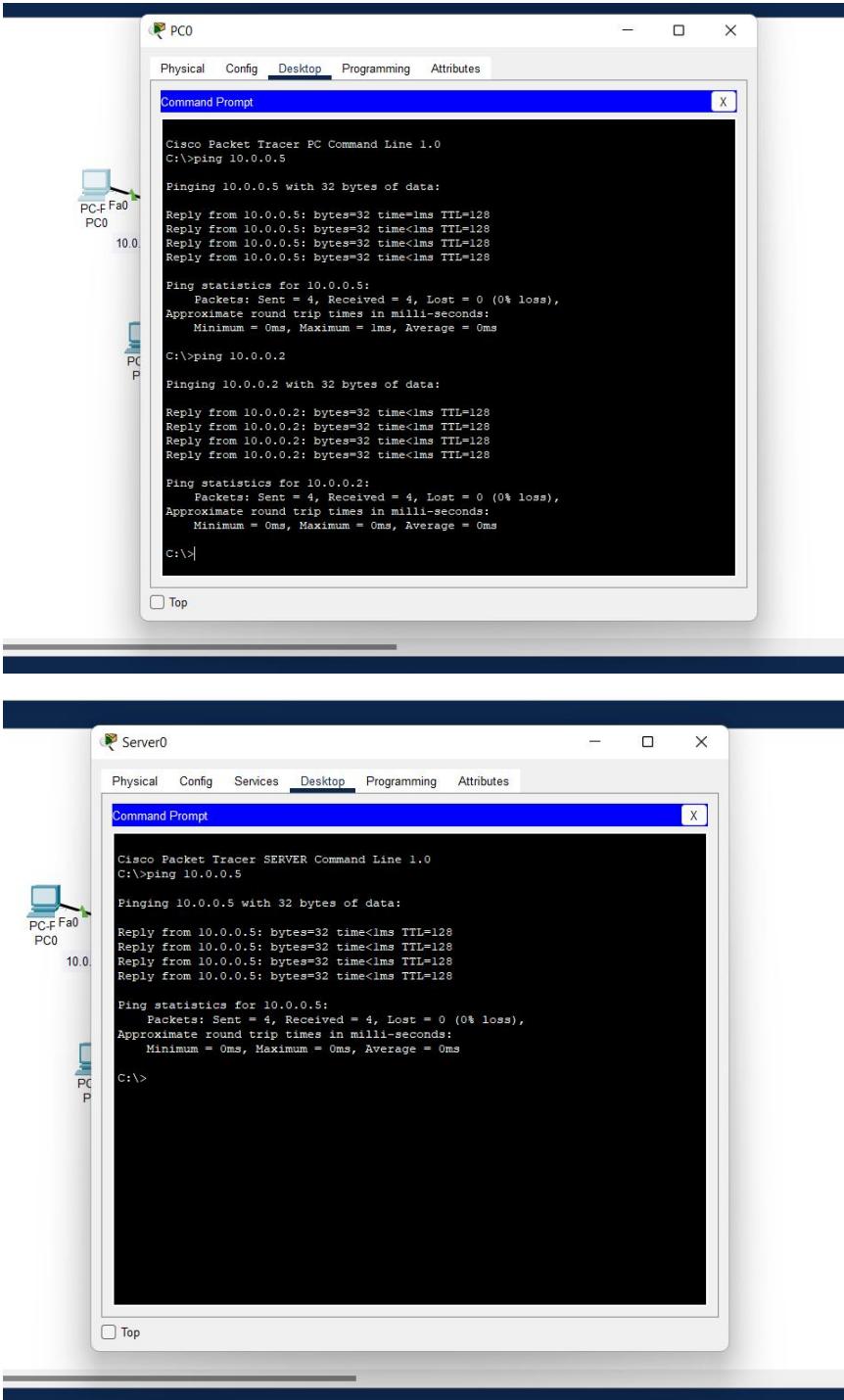
### Topology:



## Procedure:



## Snapshots of Output:



The image shows two Cisco Packet Tracer windows. The top window is titled 'PC0' and the bottom window is titled 'Server0'. Both windows have a 'Command Prompt' tab selected. The network diagram shows two hosts, 'PC0' and 'Server0', connected via a link labeled '10.0.0.0'. The PC0 host has an interface 'PC-Fa0' connected to the link. The Command Prompt window for PC0 displays the following output:

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.5

Pinging 10.0.0.5 with 32 bytes of data:
Reply from 10.0.0.5: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>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

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

C:\>
```

The Command Prompt window for Server0 displays the following output:

```
Cisco Packet Tracer SERVER Command Line 1.0
C:\>ping 10.0.0.5

Pinging 10.0.0.5 with 32 bytes of data:
Reply from 10.0.0.5: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

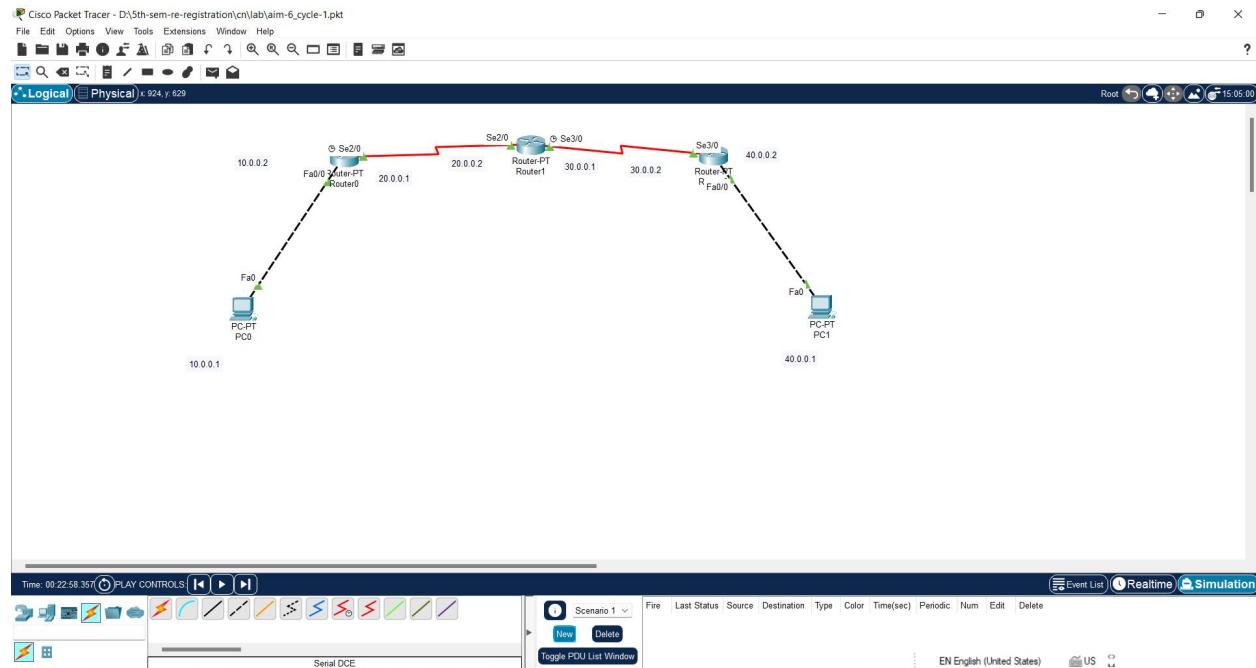
# Experiment No: 6

## Cycle-1

### Aim of the experiment:

Configuring RIP Routing Protocol in Routers

### Topology:



### **Procedure:**

Router0

Physical Config CLI Attributes

IOS Command Line Interface

```
#2100# processor: part number 0, mask 01
#2100# software, Version 3.0.0.
#2100# FastEthernet0/0 (FastEthernet0/0)
#2100# Serial0/0 (Serial0/0)
#2100# Loopback0 (Loopback0)
#2100# 32K bytes of non-volatile configuration memory.
#2100# Switch bytes of ROM configuration (Read Only).
#2100#
```

--- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: no

Press RETURN to get started!

Router>enable

Router>configure terminal

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

Router(config)#interface fastethernet0/0

Router(config-if)#ip address 10.0.0.2 255.0.0.0

Router(config-if)#no shutdown

Router(config-if)#exit

\*LINE0-CHANGED: Interface FastEthernet0/0, changed state to up

\*LINEPROTO-0-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

Router(config)#interface serial0/0

Router(config-if)#ip address 20.0.0.1 255.0.0.0

Router(config-if)#encapsulation ppp

Router(config-if)#clock rate 64000

Router(config-if)#no shutdown

\*LINE0-CHANGED: Interface Serial0/0, changed state to down

Router(config-if)#exit

Router(config)#exit

\*LINE0-CHANGED: Interface Serial0/0, changed state to up

\*LINEPROTO-0-UPDOWN: Line protocol on Interface Serial0/0, changed state to up

Router(config)#
Router(config)#router rip
Router(config-router)#network 10.0.0.0
Router(config-router)#network 20.0.0.0
Router(config-router)#exit
Router(config)#

Top

EN English (United States)

Router1

Physical Config CLI Attributes

IOS Command Line Interface

```
Would you like to enter the initial configuration dialog? [yes/no]: no

Press RETURN to get started!

Router>enable
```

Router>configure terminal

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

Router(config)#interface serial1/0

Router(config-if)#ip address 30.0.0.1 255.0.0.0

Router(config-if)#encapsulation ppp

Router(config-if)#no shutdown

Router(config-if)#
\*LINE0-CHANGED: Interface Serial1/0, changed state to up

\*LINEPROTO-0-UPDOWN: Line protocol on Interface Serial1/0, changed state to up

Router(config)#interface serial1/0

Router(config-if)#ip address 30.0.0.1 255.0.0.0

Router(config-if)#encapsulation ppp

Router(config-if)#no shutdown

\*LINE0-CHANGED: Interface Serial1/0, changed state to down

Router(config-if)#
Router(config)#exit

Router(config)#
\*LINE0-CHANGED: Interface Serial1/0, changed state to up

Router(config)#
Router(config)#interface serial1/0
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
Router(config-if)#
\*Incomplete command
Router(config-if)#clock rate 64000
Router(config-if)#no shutdown
Router(config-if)#
Router(config)#
Router(config)#router rip
Router(config-router)#network 20.0.0.0
Router(config-router)#exit
Router(config)#

Top

EN English (United States)

Router2

Physical Config CLI Attributes

IOS Command Line Interface

```
Processor board ID SPC101 (023)
#2100# processor: part number 0, mask 01
#2100# Bridging software.
#2100# Software version 3.0.0.
#2100# FastEthernet0/0 (FastEthernet0/0)
#2100# Serial0/0 (Serial0/0)
#2100# Loopback0 (Loopback0)
#2100# 32K bytes of non-volatile configuration memory.
#2100# 64480K bytes of ROM configuration (Read Only).
#2100#
```

--- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: no

Press RETURN to get started!

Router>enable

Router>configure terminal

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

Router(config)#interface fastethernet0/0

Router(config-if)#ip address 40.0.0.2 255.0.0.0

Router(config-if)#no shutdown

Router(config-if)#
\*LINE0-CHANGED: Interface FastEthernet0/0, changed state to up

\*LINEPROTO-0-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up due to Dlc/Degraded

Router(config-if)#exit

Router(config)#interface serial1/0

Router(config-if)#ip address 40.0.0.2 255.0.0.0

Router(config-if)#encapsulation ppp

Router(config-if)#no shutdown

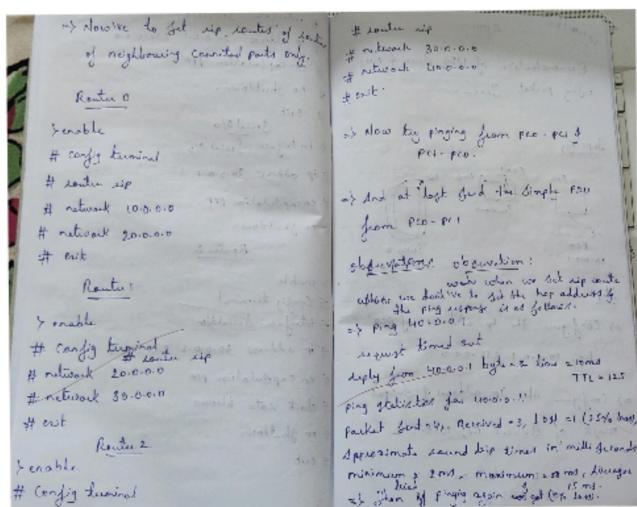
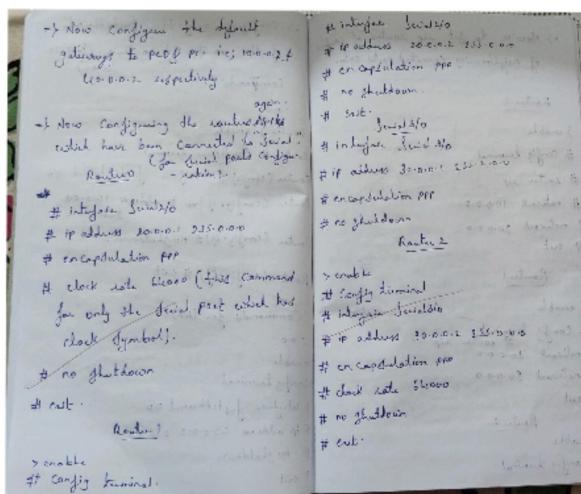
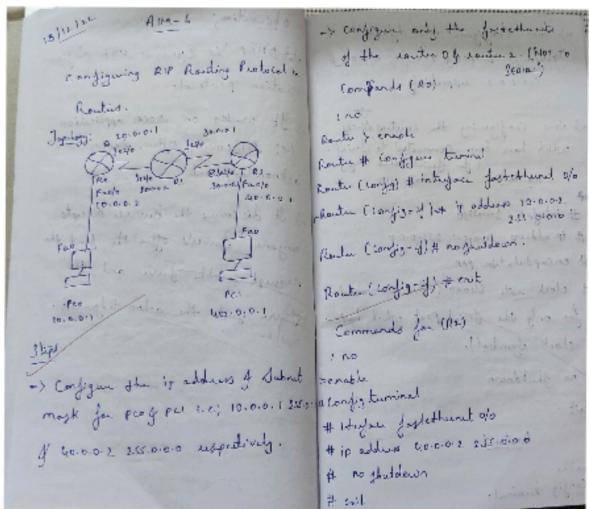
Router(config-if)#
\*LINE0-CHANGED: Interface Serial1/0, changed state to up

Router(config-if)#
\*LINE0-CHANGED: Interface Serial1/0, changed state to up

Router(config)#
Router(config)#router rip
Router(config-router)#network 30.0.0.0
Router(config-router)#network 40.0.0.0
Router(config-router)#exit
Router(config)#

Top

EN English (United States)



## Snapshots of Output:

PC0

Physical Config Desktop Programming Attributes

Command Prompt X

```
Cisco Packet Tracer PC Command Line 1.0
C:>ping 40.0.0.1
Ping request could not find host 40.0.0.1. Please check the name and try again.
C:>clear
Invalid Command.

C:>ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Request timed out.
Reply from 40.0.0.1: bytes=32 time=15ms TTL=125
Reply from 40.0.0.1: bytes=32 time=13ms TTL=125
Reply from 40.0.0.1: bytes=32 time=14ms TTL=125

Ping statistics for 40.0.0.1:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 13ms, Maximum = 15ms, Average = 14ms

C:\>
```

PC1

Physical Config Desktop Programming Attributes

Command Prompt X

```
Cisco Packet Tracer PC Command Line 1.0
C:>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1: bytes=32 time=22ms TTL=125
Reply from 10.0.0.1: bytes=32 time=22ms TTL=125
Reply from 10.0.0.1: bytes=32 time=13ms TTL=125
Reply from 10.0.0.1: bytes=32 time=21ms 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 = 13ms, Maximum = 22ms, Average = 19ms

C:\>
```

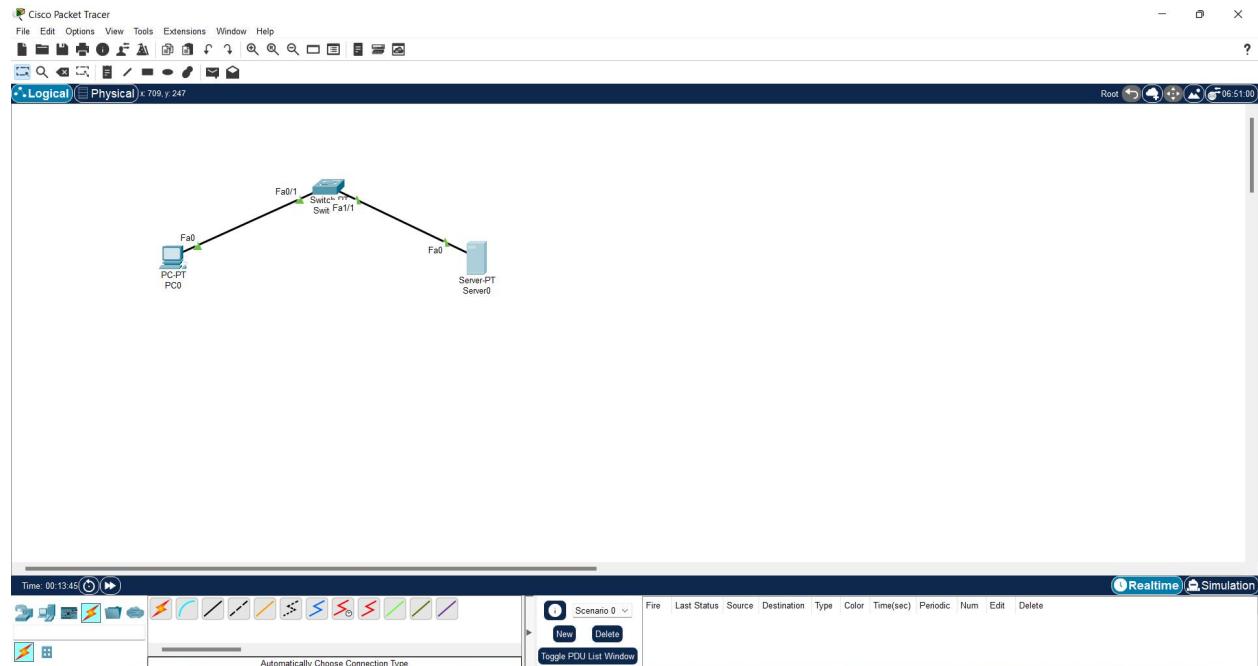
# Experiment No: 7

## Cycle-1

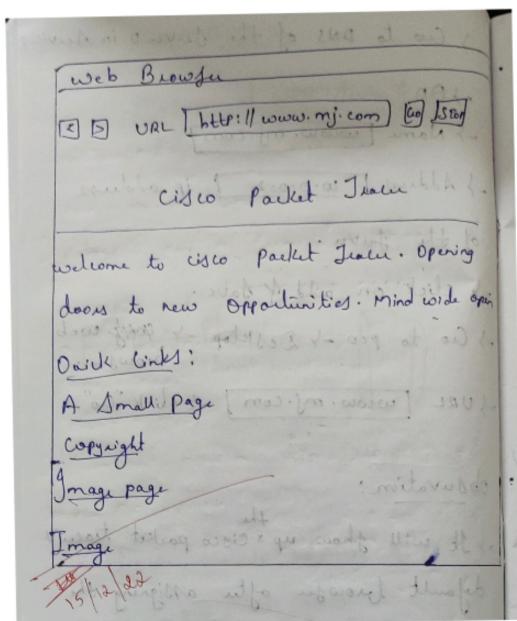
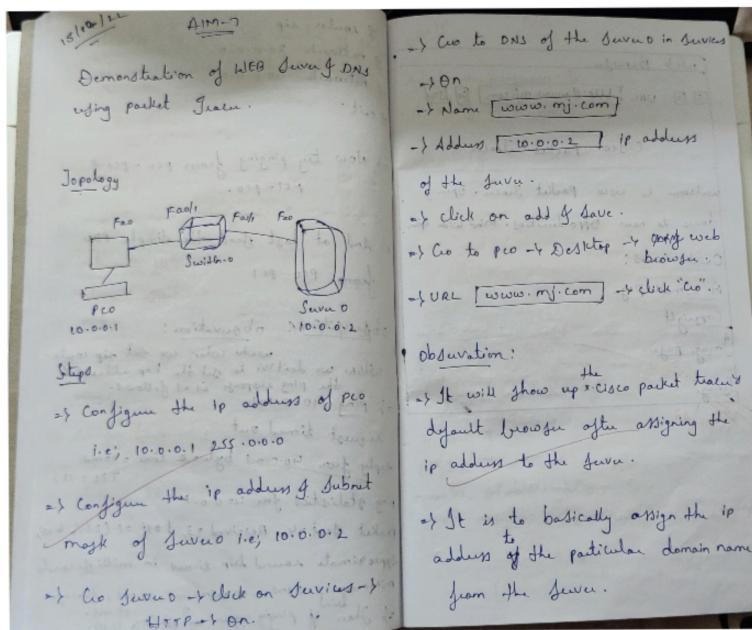
### Aim of the experiment:

Demonstration of WEB server and DNS using Packet Tracer

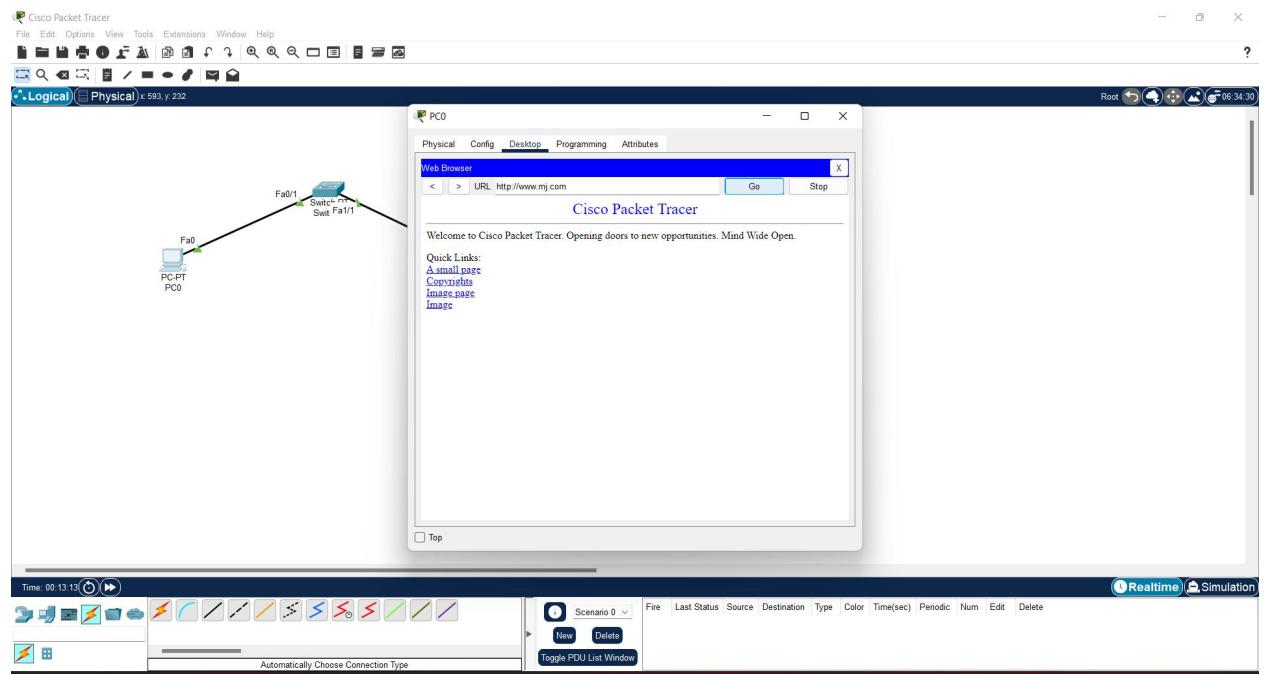
### Topology:



## Procedure:



## Snapshots of Output:



## Cycle-2

### Program-1

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

#### Code:

```
import java.util.*;
public class crc{
    public static int n;
    public static void main(String[] args){
        Scanner in=new Scanner(System.in);
        crc ob=new crc();
        String code, copy, rec,zero="0000000000000000";
        System.out.print("Enter poly: ");
        code=in.nextLine();
        System.out.println("Generating polynomial: 10001000000100001");
        n=code.length();
        copy=code;
        code+=zero;
        System.out.println("Modified poly: "+code);
        code=ob.divide(code);
        System.out.println("CheckSum: "+code.substring(n));
        copy=copy.substring(0,n)+code.substring(n);
        System.out.println("Final Codeword: "+copy);
```

```

System.out.print("Test Error detection 0(yes) 1(no)? : ");

int choice = in.nextInt();

if(choice == 0){

    System.out.print("Enter position on error: ");

    int errorPos = in.nextInt();

    if(copy.charAt(errorPos) == '1')

        copy = copy.substring(0,errorPos) + "0" + copy.substring(errorPos+1);

    else

        copy = copy.substring(0,errorPos) + "1" + copy.substring(errorPos+1);

    System.out.println("Errorneous data: "+copy);

    System.out.println("Error detected");

}

else

    System.out.println("No Error detection");

}

public String divide(String s){

    int i,j;

    char x;

    String div="10001000000100001";

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

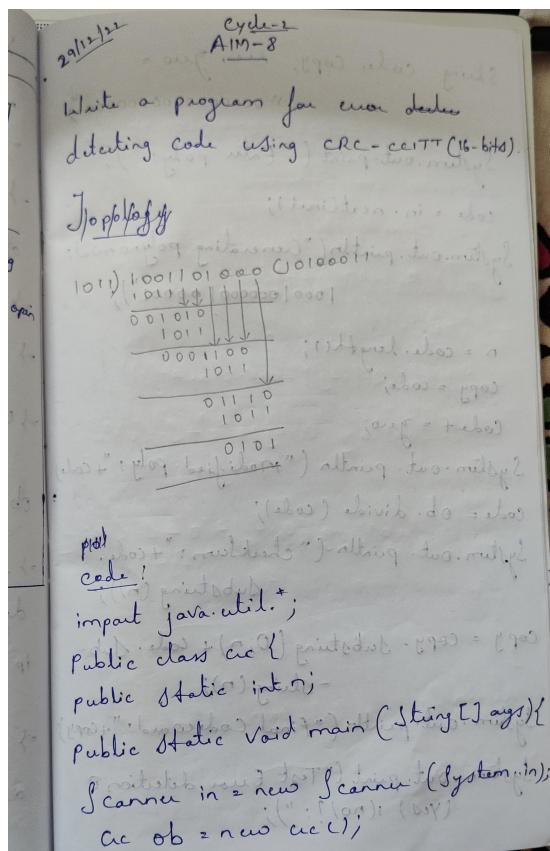
        x=s.charAt(i);

        for(j=0;j<17;j++){

            if(x=='1'){


```

```
if(s.charAt(i+j)!=div.charAt(j))  
    s=s.substring(0,i+j)+"1"+s.substring(i+j+1);  
  
else  
  
    s=s.substring(0,i+j)+"0"+s.substring(i+j+1);  
  
}  
  
}  
  
}  
  
return s;  
  
}
```



```

String code, copy, gen = "000000000000";
System.out.print("Enter poly:");
code = in.nextLine();
System.out.println("Generating Polynomial:");
System.out.println("1000100000100001");
n = code.length();
copy = code;
Code += gen;
System.out.println("Modified Poly:" + code);
code = ob.divide(code);
System.out.println("Checksum:" + code);
SubString(n);
copy = copy.substring(0, n) + code.substring(n);
System.out.println("Final Codeword:" + copy);
System.out.print("Test Error detection? ");
System.out.print("(yes) : (no)? : ");
}
}
public String divide(String s){

```

```

int i,j;
char x;
String div = "1000100000100001";
for (i=0; i<n; i++) {
    x = s.charAt(i);
    for (j=0; j<17; j++) {
        if (x == '1') {
            if (S.charAt(i+j) == div.charAt(j)) {
                S = S.substring(0, i+j) + "0" + S.substring(i+j+1);
            } else {
                S = S.substring(0, i+j) + "1" + S.substring(i+j+1);
            }
        }
    }
    return S;
}

```

Observation:

Output:  
Enter poly: 1010101010101010  
Generating polynomial: 1000100000100001  
Modified poly: 1010101010101010000000000000  
Checksum: 110011000010101  
Final Codeword: 10101010101010111001000  
Test error detection. 0 (yes) : 1 (no)? : 0  
Enter position on error (0-16): 10  
Errorous data: 00101010101010111001  
000010101  
Error detected.

## **Output – Screen shot:**

Online Java Compiler

Programiz Online Java Compiler

Shutterstock Free Trial - Get images, video, music & easy to use design tools with one subscription.

Main.java

```
49     System.out.println("No Error detection");
50 }
51
52+ public String divide(String s){
53     int i,j;
54     char x;
55     String div="1000100000100001";
56
57+     for(i=0;i<n;i++){
58         x=s.charAt(i);
59
60+         for(j=0;j<17;j++){
61             if(x=='1'){
62                 if(s.charAt(i+j)!=div.charAt(j))
63                     s=s.substring(0,i+j)+"1"+s.substring(i+j+1);
64                 else
65                     s=s.substring(0,i+j)+"0"+s.substring(i+j+1);
66             }
67         }
68     }
69
70     return s;
71 }
72 }
```

Run Output

```
java -cp /tmp/fqnE0J8eUc crc
Enter poly: 1010101010101010
Generating polynomial: 10001000000100001
Modified poly: 10100101010101000000000000000000
CheckSum: 1110011000010101
Final Codeword: 10101010101010101110011000010101
Test Error detection 0(yes) 1(no)? : 0
Enter position on error: 0
Errorneous data: 00101010101010101110011000010101
Error detected
```

Interactive Java Course

ADS VIA CARBON

Clear

## Cycle-2

### Program-2

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

#### Code:

```
#include <limits.h>
#include <stdbool.h>
#include <stdio.h>
#define V 9

int minDistance(int dist[], bool sptSet[])
{
    int min = INT_MAX, min_index;
    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[])
{
    printf("Vertex \t\t Distance from Source\n");
    for (int i = 0; i < V; i++)
        printf("%d \t\t\t %d\n", i, dist[i]);
}
```

```

void dijkstra(int graph[V][V], int src)
{
    int dist[V];
    bool sptSet[V];
    for (int i = 0; i < V; i++)
        dist[i] = INT_MAX, 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]
                && dist[u] != INT_MAX
                && dist[u] + graph[u][v] < dist[v])
                dist[v] = dist[u] + graph[u][v];
    }
    printSolution(dist);
}

int main()
{
    int graph[V][V] = { { 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, 0, 2, 0, 1, 6 },
{ 8, 11, 0, 0, 0, 0, 1, 0, 7 },
{ 0, 0, 2, 0, 0, 0, 6, 7, 0 } };
```

dijkstra(graph, 0);

return 0;

}

cycle-2  
4m-9

```
Implement Dijkstraa algorithm to
compute the shortest path for a
given topology
code:
#include <limits.h>
#include <stdio.h>
#include <stdlib.h>
#define V 9
int minDistance(int dist[], bool sptst)
{
    int min = INT_MAX, min_index;
    for (int v = 0; v < V; v++)
        if (sptst[v] == false) if (dist[v] < min)
            min = dist[v], min_index = v;
    return min_index;
}

void printSolution (int dist[])
{
    printf ("Vertex \t Distance from
    Source \n");
    for (int i=0; i<V; i++)
        printf ("%d \t %d \n", i, dist[i]);
    cout << endl;
}

void dijkstra (int graph[V][V], int src)
{
    int dist[V];
    bool sptst[V];
    for (int i=0; i<V; i++)
        dist[i] = INT_MAX, sptst[i] = false;
    dist[src] = 0;
    for (int count = 0; Count < V - 1; count++)
        {
            int u = minDistance (dist, sptst);
            sptst[u] = true;
            for (int v = 0; v < V; v++)
                if (!sptst[v] && graph[u][v] != 0)
                    if (dist[u] + graph[u][v] < dist[v])
                        dist[v] = dist[u] + graph[u][v];
}
```

for (int v=0; v<V; v++)
if (!sptst[v] && graph[u][v] != 0)
 dist[v] = dist[u] + graph[u][v];
 if (dist[v] < INT\_MAX && graph[u][v] != 0)
 graph[u][v] = 0;
int main()
{
 int graph [V][V]={{0,4,0,0,0,0,0,0,0},
 {4,0,5,0,0,0,0,0,0},
 {0,5,0,7,0,0,0,0,2},
 {0,0,7,0,9,0,0,0,0},
 {0,0,0,9,0,10,0,0,0},
 {0,0,4,0,9,0,2,0,0},
 {0,0,0,0,0,2,0,1,4},
 {0,0,0,0,0,0,1,4,0},
 {0,0,0,0,0,0,0,2,0},
 {0,0,2,0,0,0,0,6,10}};
 dijkstra (graph[0]);
 return 0;
}

Vertex	Distance from Source
0	0
1	4
2	12
3	18
4	22
5	16
6	13
7	8
8	14

## **Output – Screen shot:**

The screenshot shows the OnlineGDB IDE interface. On the left, there's a sidebar with links like 'My Projects', 'Classroom', 'Learn Programming', 'Programming Questions', 'Jobs', 'Sign Up', and 'Login'. A blue banner at the bottom left encourages users to share their opinions for rewards. The main workspace has tabs for 'Run', 'Debug', 'Stop', 'Share', 'Save', and 'Beautify'. The code editor contains a C program named 'main.c' that includes headers for limits.h, stdbool.h, and stdio.h. Below the code is a table titled 'Distance from Source' with vertices 0 through 8 and their corresponding distances. The terminal window at the bottom shows the program has finished with exit code 0 and prompts the user to press ENTER to exit.

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

```
...Program finished with exit code 0
Press ENTER to exit console.
```

## Cycle-2

### Program-3

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

#### Code:

```
#include<stdio.h>

struct node
{
    unsigned dist[20];
    unsigned from[20];
}rt[10];

int main()
{
    int costmat[20][20];
    int nodes,i,j,k,count=0;
    printf("\nEnter the number of nodes : ");
    scanf("%d",&nodes);
    printf("\nEnter the cost matrix :\n");
    for(i=0;i<nodes;i++)
    {
        for(j=0;j<nodes;j++)
        {
            scanf("%d",&costmat[i][j]);
            costmat[i][i]=0;
            rt[i].dist[j]=costmat[i][j];
        }
    }
}
```

```

    rt[i].from[j]=j;
}

}

do
{
    count=0;
    for(i=0;i<nodes;i++)
        for(j=0;j<nodes;j++)
            for(k=0;k<nodes;k++)
                if(rt[i].dist[j]>costmat[i][k]+rt[k].dist[j])
                {
                    rt[i].dist[j]=rt[i].dist[k]+rt[k].dist[j];
                    rt[i].from[j]=k;
                    count++;
                }
}while(count!=0);
for(i=0;i<nodes;i++)
{
    printf("\n\n For router %d\n",i+1);
    for(j=0;j<nodes;j++)
    {
        printf("\t\nnode %d via %d Distance %d ",j+1,rt[i].from[j]+1,rt[i].dist[j]);
    }
}
printf("\n\n");
getch(); }
```

graph LR

wire a program for distance  
vertex algorithm to find shortest  
path for transmission

**Algorithm**

```

    Distance vertex finding()
    {
        minBridge (Create initial vertex for
                    node)
        D[empty] = 0
        for [y = 1 to n]
        {
            if [y is a neighbor]
                D[y] = minBridge[y]
            else
                D[y] = ∞
        }
    }

```

Find vertex (0,0,0,...,0)  
as neighbor.  
Find the vertex with the  
shortest distance from a column  
nearest (nearest vertex)  
will (for a vertex Due from a neighbor  
choose to do any change to do  
this)

for [x = 1 to n]
 for [y = 1 to n]
 if [y is a neighbor]
 if [D[x] + weight[x,y] < D[y]]
 D[y] = D[x] + weight[x,y]

code

```

#include <stdio.h>
struct node
{
    unsigned dist[10];
    unsigned from[10];
    int count;
    int cost;
    int nodes, i, j, k, l, count;
    printf ("Enter the no. of nodes: ");
    scanf ("%d", &nodes);
    printf ("Enter the cost matrix: ");
    for (i=0; i<nodes; i++)
    {
        for (j=0; j<nodes; j++)
        {
            printf ("%d ", cost[i][j]);
        }
        printf ("\n");
    }
}

```

costmat (100\*100)
 {
 int i, j, k, l, count;
 int costmat[10][10];
 int nodes, i, j, k, l, count;
 printf ("Enter the cost matrix: ");
 for (i=0; i<nodes; i++)
 {
 for (j=0; j<nodes; j++)
 {
 printf ("%d ", costmat[i][j]);
 }
 printf ("\n");
 }
 }

```

while (Count != 0)
{
    for (i=0; i<nodes; i++)
    {
        printf ("%d ", dist[i]);
    }
    printf ("\n");
    for (i=0; i<nodes; i++)
    {
        printf (" %d via 1 Distance: %d\n", i, dist[i]);
        printf (" %d via 2 Distance: %d\n", i, dist[i]);
        printf (" %d via 3 Distance: %d\n", i, dist[i]);
    }
}

```

Output:

Enter the number of nodes: 3  
Enter the cost matrix:  
0 27 31  
27 0 2  
31 2 0

## Output – Screen shot:

The screenshot shows a web-based IDE interface for OnlineGDB. The left sidebar has links for 'My Projects', 'Classroom', 'Learn Programming', 'Programming Questions', 'Jobs', 'Sign Up', and 'Login'. A blue banner at the bottom left says 'GOT AN OPINION? SHARE AND GET REWARDED. @Rakuten AIP'. The main area has tabs for 'Run', 'Debug', 'Stop', 'Save', and 'Beautify'. The code editor shows 'main.c' with the following content:

```
25+ {  
26+     count=0;  
27+     for(i=0;i<nodes;i++)  
28+         for(j=0;j<nodes;j++)
```

The terminal window below shows the output of the program. It first asks for the number of nodes and then the cost matrix. The output then details the shortest paths from each node to all others:

```
Enter the number of nodes : 3  
Enter the cost matrix :  
0 2 3  
2 0 7  
7 1 0  
  
For router 1  
node 1 via 1 Distance 0  
node 2 via 2 Distance 2  
node 3 via 3 Distance 3  
  
For router 2  
node 1 via 1 Distance 2  
node 2 via 2 Distance 0  
node 3 via 1 Distance 5  
  
For router 3  
node 1 via 2 Distance 3  
node 2 via 2 Distance 1  
node 3 via 3 Distance 0
```

At the bottom, there are links for 'About', 'FAQ', 'Blog', 'Terms of Use', 'Contact Us', 'GDB Tutorial', 'Credits', 'Privacy', and copyright information '© 2016 - 2023 GDB Online'. The language setting is set to 'EN English (United States)'.

## Cycle-2

### Program-4

Write a program for congestion control using Leaky bucket algorithm.

#### Code:

```
#include<bits/stdc++.h>
#include<unistd.h>
using namespace std;
#define bucketSize 500

void bucketInput(int a,int b)
{
    if(a > bucketSize)
        cout<<"\n\t\tBucket overflow";
    else{
        sleep(5);
        while(a > b){
            cout<<"\n\t\t" << b << " bytes outputted.";
            a-=b;
            sleep(5);
        }
    }
}
```

```

if(a > 0)
    cout<<"\n\t\tLast "<<a<<" bytes sent\t";
    cout<<"\n\t\tBucket output successful";
}

int main()
{
    int op,pktSize;
    cout<<"Enter output rate : ";
    cin>>op;
    for(int i=1;i<=5;i++)
    {
        sleep(rand()%10);
        pktSize=rand()%700;
        cout<<"\nPacket no "<<i<<"\tPacket size = "<<pktSize;
        bucketInput(pktSize,op);
    }
    cout<<endl;
    return 0;
}

```

cycle - 2

~~30/1/23~~  
4m - 10

write a program for Congestion Control using leaky bucket algorithm.

```

#include <bits/stdc++.h>
#include <iostream.h>
using namespace std;
#define bucketSize 500

void bucketInput(int a, int b)
{
    if (a > bucketSize)
        cout << "Leaky Bucket overflow";
    else
        Sleep(5);
    while (a > b)
    {
        cout << "Input " << b << " bytes outputted";
        a -= b;
        Sleep(5);
    }
    if (a == 0)
}

```

```

cout << "Input Last " << a << " bytes sent";
cout << "Leaky Bucket Output Successful";
}

int main()
{
    int op, pktsize;
    cout << "Enter Output rate : ";
    cin >> op;
    for (int i = 1; i <= 5; i++)
    {
        Sleep((rand() % 10));
        pktsize = rand() % 200;
        cout << "In packet no " << i << " In packet size = " << pktsize;
    }
    cout << endl;
    return 0;
}

```

Output:

Enter output rate : 50

Packet no 1 packet size = 186  
 50 bytes outputted  
 50 bytes outputted  
 50 bytes outputted  
 50 bytes outputted  
 last 36 bytes sent  
 Bucket output successful

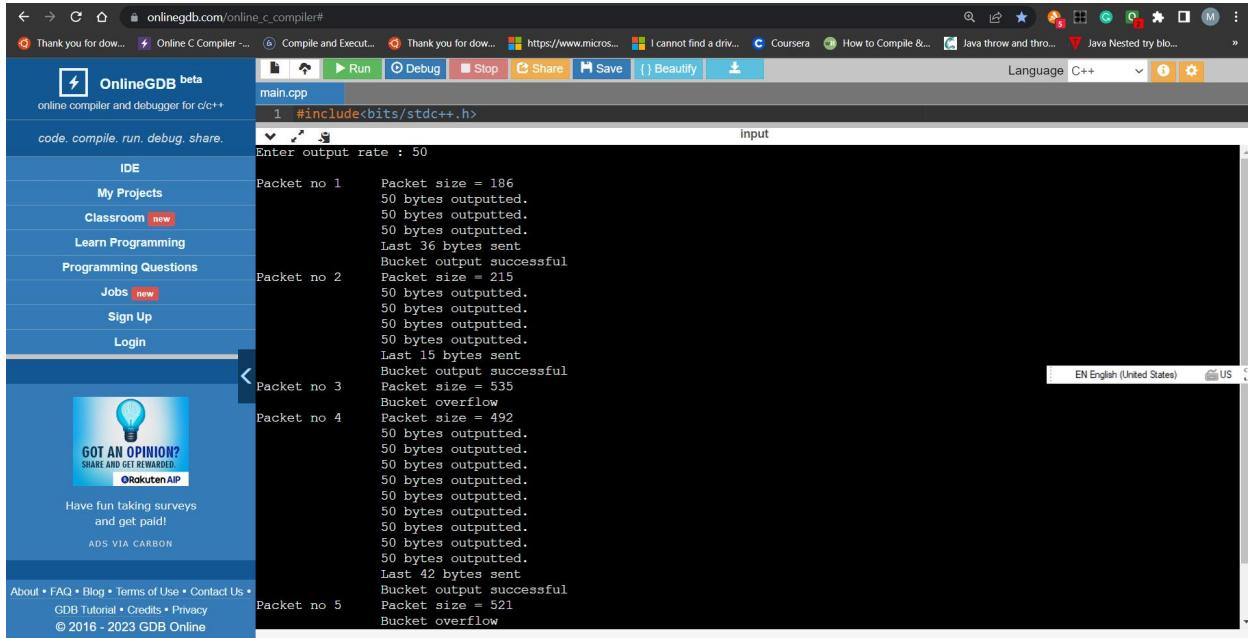
Packet no 2 packet size = 215  
 50 bytes outputted  
 " " "  
 last 15 bytes sent  
 Bucket out successful

Packet no 3 packet size = 535  
 Bucket overflow

Packet no 4 packet size = 492  
 50 bytes outputted  
 " " "  
 " " "

Last 42 bytes sent  
 Leaky Bucket Output Successful  
 Packet no 5 pack size = 521  
 Bucket Overflow

## Output – Screen shot:



The screenshot shows a web-based IDE interface for OnlineGDB. The URL in the address bar is `onlinedb.com/online_c_compiler#`. The main window displays a C++ code editor with the file `main.cpp` containing the following code:

```
1 #include<bits/stdc++.h>
```

Below the code editor is a terminal-like window titled "Input" which shows the following output:

```
Enter output rate : 50
Packet no 1    Packet size = 186
50 bytes outputted.
50 bytes outputted.
50 bytes outputted.
Last 36 bytes sent
Bucket output successful
Packet no 2    Packet size = 215
50 bytes outputted.
50 bytes outputted.
50 bytes outputted.
50 bytes outputted.
Last 15 bytes sent
Bucket output successful
Packet no 3    Packet size = 535
Bucket overflow
Packet no 4    Packet size = 492
50 bytes outputted.
Last 42 bytes sent
Bucket output successful
Packet no 5    Packet size = 521
Bucket overflow
```

The left sidebar of the IDE includes links for "My Projects", "Classroom", "Learn Programming", "Programming Questions", "Jobs", "Sign Up", and "Login". There is also an advertisement for "RokuZen AIP" and links for "About", "FAQ", "Blog", "Terms of Use", "Contact Us", "GDB Tutorial", "Credits", and "Privacy". The bottom right corner of the terminal window shows language settings: "EN English (United States)" and "US".

## Cycle-2

### Program-5

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.

#### Code:

##### Server Code:

```
import socket  
  
serverName = '127.0.0.1'  
  
serverPort = 12345  
  
#create  
  
server_socket = socket.socket(socket.AF_INET,socket.SOCK_STREAM)  
  
#bind  
  
server_socket.bind((serverName, serverPort ))  
  
#listen  
  
server_socket.listen(5)  
  
while True:  
  
    print("Server waiting for connection")  
  
    client_socket, addr = server_socket.accept()  
  
    print("Client connected from",addr)  
  
    sentence = client_socket.recv(1024).decode()  
  
    file = open(sentence, "r")  
  
    l = file.read(1024)
```

```
client_socket.send(l.encode())
file.close()
client_socket.close()
```

### **Client Code:**

```
import socket

serverName = '127.0.0.1'
serverPort = 12345

client_socket = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
client_socket.connect((serverName,serverPort))
sentence = input("Enter file name: ")

client_socket.send(sentence.encode())
filecontents = client_socket.recv(1024).decode()
print ('From Server:', filecontents)
client_socket.close()
```

30/1/12

Cycle - 2

Aim - 12

Using Tcp/Ip Sockets, write a client Java program to make client sending the file name and the server to send back the contents of the requested file if present.

Server code:

```
import Socket  
import java.net.  
ServerName = '127.0.0.1'  
ServerPort = 12345  
Server_Socket = Socket.Socket(Socket.AF_INET,  
                               Socket.SOCK_STREAM)  
  
Server_Socket.listen(5)  
  
while True:  
    print("Server waiting for connection")  
    client_Socket, addr = Server_Socket.  
                           accept()
```

```
print("client connected from", addr)  
Sentence = client_Socket.recv(1024).decode()  
file = open(Sentence, 'r')  
l = file.read(1024)  
client_Socket.send(l.encode())  
file.close()  
client_Socket.close()
```

Client code:

```
import Socket  
import java.net.  
ServerName = '127.0.0.1'  
ServerPort = 12345  
client_Socket = Socket.Socket(Socket.AF_INET,  
                             Socket.SOCK_STREAM)  
client_Socket.connect((ServerName, ServerPort))  
  
Sentence = input("Enter file name: ")  
client_Socket.send(Sentence.encode())  
filecontents = client_Socket.recv(1024).decode()  
print("From Server:", filecontents)  
client_Socket.close()
```

Output:

Server side:

```
python tcp-ip-server.py  
Server waiting for connection  
client connected from ('127.0.0.1', 56450)
```

Client side:

```
python tcp-ip-client.py
```

Enter file name: tcp-ip-example.txt

From Server: This is an example file for  
tcp/ip. (socket)visu - debot - 12345 - 0.0.0.1

## Output – Screen shot:

The image shows two Microsoft Windows Command Prompt windows side-by-side. The left window, titled 'Command Prompt', shows the execution of a Python client program. The right window, titled 'Command Prompt - python tcp\_ip\_server.py', shows the execution of a Python server program. Both windows display the command entered, the output of the program, and the file system status.

**Left Window (Client):**

```
Microsoft Windows [Version 10.0.22000.1455]
(c) Microsoft Corporation. All rights reserved.

C:\Users\megha>d:
D:>cd 5th-sem-re-registration\cn\lab
D:\5th-sem-re-registration\cn\lab>python tcp_ip_client.py
Enter file name: tcp-ip-example.txt
From Server: This is an example file for TCP/IP program.

D:\5th-sem-re-registration\cn\lab>
```

**Right Window (Server):**

```
Command Prompt - python tcp_ip_server.py
30-Jan-23 11:59 PM      570 tcp_ip_server.py
31-Jan-23 12:36 AM      457 udp-client.py
31-Jan-23 12:37 AM      40 udp-example.txt
31-Jan-23 12:37 AM      569 udp-server.py
          13 File(s)   314,948 bytes
          3 Dir(s)  467,937,742,848 bytes free

D:\5th-sem-re-registration\cn\lab>python tcp_ip_server.py
Server waiting for connection
Client connected from ('127.0.0.1', 56450)
Server waiting for connection
```

## Cycle-2

### Program-6

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:

##### Server Code:

```
import socket
```

```
serverName = '127.0.0.1'
```

```
serverPort = 12345
```

```
# Create a datagram socket
```

```
UDPServerSocket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

```
# Bind to address and ip
```

```
UDPServerSocket.bind((serverName, serverPort))
```

```
print("UDP server up and listening")
```

```
# Listen for incoming datagrams
```

```
while (True):
```

```
    sentence, clientAddress = UDPServerSocket.recvfrom(2048)
```

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

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

### **Client Code:**

```
import socket

serverName = '127.0.0.1'
serverPort = 12345

# Create a UDP socket at client side
UDPClientSocket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Send to server using created UDP socket
sentence = input("Enter file name: ")

UDPClientSocket.sendto(bytes(sentence,"utf-8"),(serverName, serverPort))
filecontents,serverAddress = UDPClientSocket.recvfrom(2048)
print ('From Server:', filecontents)

UDPClientSocket.close()
```

30/1/23      Cycle - 2  
 Aim - 13

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.

Server code:  
 import socket  
 ServerName = '127.0.0.1'  
 ServerPort = 12345  
 UDPServerSocket = socket.socket(socket.AF\_INET,  
 socket.SOCK\_DGRAM)

```

UDPServerSocket.bind((ServerName, ServerPort))
print("UDP Server up and listening")
while True:
    sentence, clientAddress = UDPServerSocket.recvfrom(1024)
  
```

Output:  
Server side:  
 python udp-server.py  
 UDP Server up and listening  
 Sent back to client: This is an example file  
 for UDP program.

Client code:  
 import socket  
 ServerName = '127.0.0.1'  
 ServerPort = 12345  
 UDPclientSocket = socket.socket(socket.AF\_INET,  
 socket.SOCK\_DGRAM)

```

Sentence = input("Enter file name: ")
UDPclientSocket.sendto(Sentence.encode("utf-8"), (ServerName, ServerPort))
filecontents, ServerAddress = UDPclientSocket.recvfrom(1024)
print("From Server:", filecontents)
UDPclientSocket.close()
  
```

Client side:  
 python udp-client.py  
 Enter file name: udp-example.txt  
 From Server: b'This is an example file  
 for UDP program'

## Output – Screen shot:

The screenshot shows two Microsoft Windows Command Prompt windows side-by-side.

**Left Window (C:\ Command Prompt):**

```
Microsoft Windows [Version 10.0.22000.1455]
(c) Microsoft Corporation. All rights reserved.

C:\Users\megha>d:
D:\>cd 5th-sem-re-registration
D:\5th-sem-re-registration>cd cn
D:\5th-sem-re-registration\cn>cd lab
D:\5th-sem-re-registration\cn\lab>python udp-client.py
Enter file name: udp-example.txt
From Server: b'This is an example file for UDP program.'
```

**Right Window (D:\ Command Prompt - python udp-server.py):**

```
30-Jan-23 02:27 PM      800 leaky-bucket.c
31-Jan-23 12:00 AM      43 tcp-ip-example.txt
30-Jan-23 11:57 PM      368 tcp_ip_client.py
30-Jan-23 11:59 PM      570 tcp_ip_server.py
31-Jan-23 12:36 AM      457 udp-client.py
31-Jan-23 12:37 AM      40 udp-example.txt
31-Jan-23 12:37 AM      569 udp-server.py
13 File(s)   314,948 bytes
3 Dir(s)   467,937,742,848 bytes free

D:\5th-sem-re-registration\cn\lab>python udp-server.py
UDP server up and listening
sent back to client: This is an example file for UDP program.
```

At the bottom right of the screen, there is a language and region settings icon showing "EN English (United States)" and "US".