

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



**LAB REPORT**  
**on**

## **COMPUTER NETWORKS**

*Submitted by*

**PRAJWAL BHAT (1BM20CS107)**

*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 **PRAJWAL BHAT (1BM20CS107)**, who is bona fide student at **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.

**M Lakshmi Neelima**  
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>
<b>1.</b>	10/11/22	Creating a topology and simulate sending a simple PDU from source to destination using hub and switch as connecting devices.	
<b>2</b>	24/11/22	Configuring IP address to Routers in Packet Tracer. Explore the following messages: Ping Responses, Destination unreachable, Request timed out, Reply.	
<b>3</b>	01/12/22	Configuring default route to the Router.	
<b>4</b>	15/12/22	Configuring DHCP within a LAN in a packet Tracer.	
<b>5</b>	08/12/22	Configuring RIP Routing Protocol in Routers.	
<b>6</b>	15/12/22	Demonstration of WEB server and DNS using Packet Tracer.	
<b>7</b>	29/12/22	Write a program for error detecting code using CRC-CCITT (16-bits).	
<b>8</b>	12/01/23	Write a program for distance vector algorithm to find suitable path for transmission.	
<b>9</b>	12/01/23	Implement Dijkstra's algorithm to compute the shortest path for a given topology.	
<b>10</b>	05/01/23	Write a program for congestion control using Leaky bucket algorithm.	
<b>11</b>	02/02/23	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.	
<b>12</b>	02/02/23	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.	

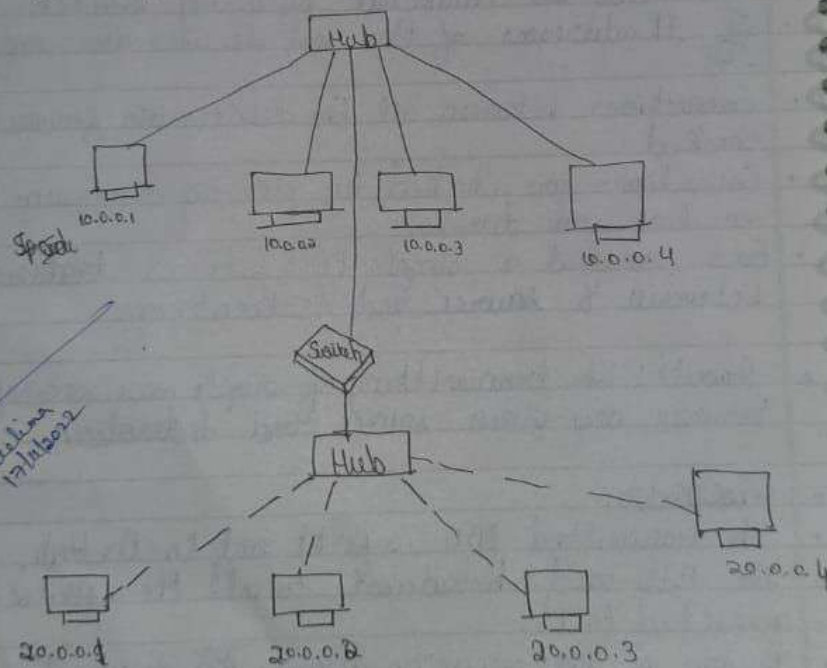
### Experiment-1

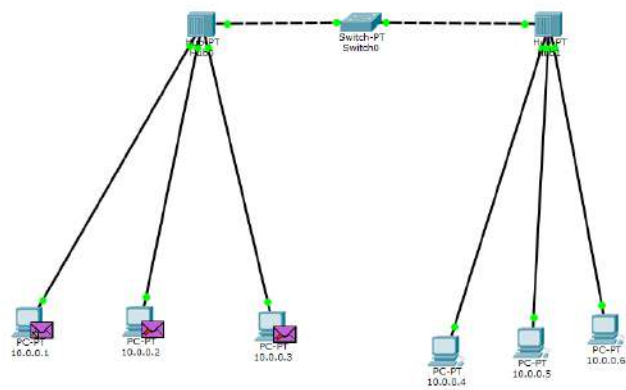
- Aim: To simulate sending a simple PDU from source to destination using hub and switch as connecting devices.
- Topology: Star topology
- Procedure:
  - End devices are connected through the hub.
  - The hubs are connected by using switch.
  - IP addresses of the end devices are configured.
  - Connections between all the devices are formed and checked.
  - Connections are checked by pinging a message between two end devices.
  - Once verified a simple PDU file is transmitted between a source and destination.
- Result: The transmitting of simple was successful between any given source and destination.
- Observation:
  - The transmitted PDU is first sent to the Hub.
  - The Hub will broadcast to all the devices connected to it.
  - If any of the receiving devices is destination device, then transmission stops, else the wait.

The switch will initially broadcast to the hubs connected to it. A ping message is sent by the destination device giving details about its mac address to the switch, which is stored for further transmissions.

The transmission is said to be successful if the PDU file returns through the network and a ✓ appears at the source.

Topology diagram:-





SimulationPanel

Event List

Vis.	Time(sec)	Last Device	At Device	Type	Info
	0.004	Hub1	10.0.0.6	ICMP	
	0.005	10.0.0.3	Hub1	ICMP	
	0.006	Hub1	Switch0	ICMP	
	0.006	Hub1	10.0.0.4	ICMP	
	0.006	Hub1	10.0.0.6	ICMP	
	0.007	Switch0	Hub0	ICMP	
	0.008	Hub0	10.0.0.1	ICMP	
	0.008	Hub0	10.0.0.2	ICMP	
	0.008	Hub0	10.0.0.3	ICMP	

Reset Simulation ☒ Constant Delay Captured I/O: 0.008 s

Play Controls: Back Auto Capture / Play Capture / Forward

Event List Filters - Visible Events

ACL Filter, ARP, BGP, CDP, DHCP, DNS, DTP, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPsec, ISAKMP, LACP, MDX, NETFLOW, NTP, OSPF, OSPFv6, PAPI, POP3, RADIUS, RDP, RDPv6, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TFTP, Telnet, UDP, VDP

Edit Filters Show All/None



- A ping is sent from end device to the connected router. Request is timed out. Hence gateway is set.
- A ping is sent from end device to a router not connected to it. We get Destination host not reachable. So we go to each router and route it to all other networks, using command  

```
ip route 30.0.0.0 255.0.0.0 20.0.0.2
```
- A ping is sent from end device to other end device.
- Result: A successful ping message is transmitted from one end device to another end device.

#### Observation:

- A ping doesn't cross the interface until a gateway has been set to the connected interface/router.
- Once gateway has been set, the ping will not cross over to another router as the routers are not connected to other networks and they can't know which route to take or where the next hop of the signal is to be done.
- The routers are configured with ip route where the network name, subnet mask and the next hop to reach the network is given to all the networks <sup>directly</sup> not connected to it.
- After routing, the router will know the path in which ping travels.

## Experiment-2

Aim: To configure IP addresses to Routers in Packet Tracer. Explore the following messages: Ping responses, Destination unreachable.

Topology: Star topology.

Procedure:

Two end devices are connected to a router.

The IP addresses are configured for the end devices.

The CLI of router is opened and following commands are entered:

enable

config terminal

interface Fa0/0

ip address ipaddress subnetmask

no shutdown

exit

The above commands are used to set IP address of interface of the router.

A ping is sent from end devices to the router interface to which it is connected.

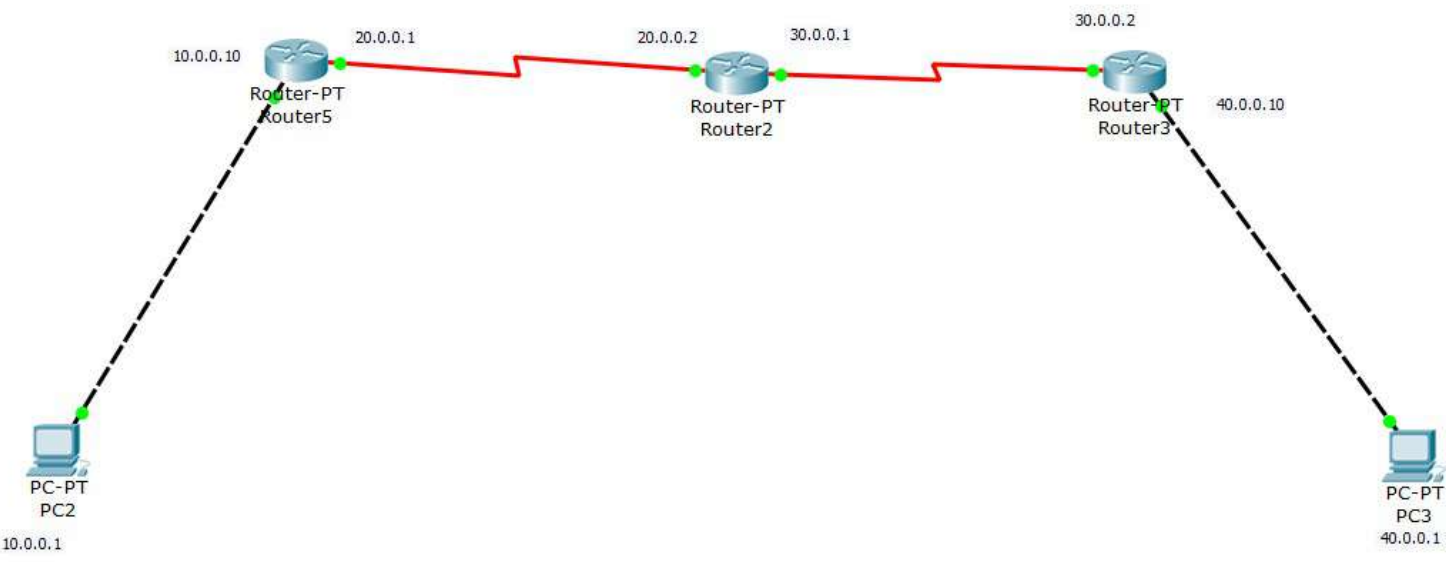
→ The ping is accepted and received.

A ping is sent. A ping is sent to the other end device.

The gateway for the end devices are set.

A ping sent from between the end devices.





Output:

1) PC > ping 20.0.0.1 [from 10.0.0.1]  
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)

2) PC > ping 30.0.0.1  
Pinging 30.0.0.1 with 32 bytes of data:  
Destination host not reachable. error

3) PC > ping 40.0.0.1  
Pinging 40.0.0.1 with 32 bytes of data:  
No Reply from 40.0.0.1: bytes=32 time=8ms TTL=125  
Reply from 40.0.0.1: bytes=32 time=8ms TTL=125  
Reply from 40.0.0.1: bytes=32 time=8ms TTL=125  
Reply from 40.0.0.1: bytes=32 time=8ms TTL=125  
Ping statistics for 40.0.0.1:  
Packets: Sent=4, Received=4, Lost=0 (0% Loss)

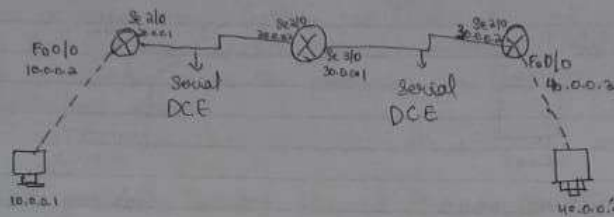
*Alexia*  
*24/11/2012*

24/11/22

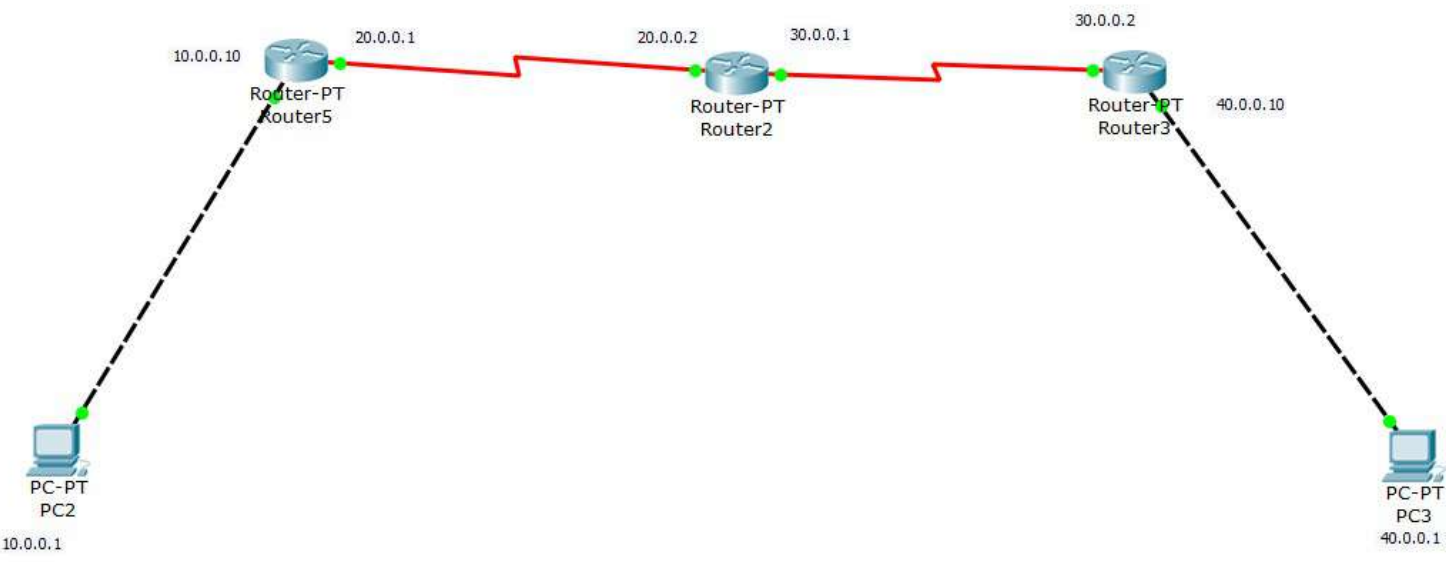
### LAB-Experiment-3

\* Aim: To configure static IP route to the routers

\* Topology:-



- Procedure:-
  - The end devices are connected to routers.
  - The routers are connected to each other by using another router.
  - The IP addresses and gateway is set for the end devices.
  - The CLI of router is opened and the interfaces and IP addresses for each interface is set.
  - The connection is valid if there is a green light on the connection.
  - Following commands are written for configuring a router:  
enable  
config terminal  
interface Ser0/0  
ip address 20.0.0.2 255.0.0.0  
no shutdown  
exit



Reply from 40.0.0.1: bytes=32 time=70ms TTL=253  
Packets: Sent=4, Received=4, Lost=0 (0% loss)

Akshay  
4/12/2022



Request timed out  
Request timed out  
Request timed out  
Ping statistics for 30.0.0.1: Packets: Sent=4, Received=0, Loss=4 (100% loss)  
2) Gateway setup route not configured.

PC > ping 30.0.0.1  
Pinging 30.0.0.1 with 32 bytes of data:  
Reply from 10.0.0.3: Destination host unreachable  
Reply from 10.0.0.3: Destination host unreachable  
Reply from 10.0.0.3: Destination host unreachable  
Reply from 10.0.0.3: Destination host unreachable  
Ping statistics for 30.0.0.1:  
Packets: Sent=4, Received=0, Loss=4 (100% loss)

3) Gateway set, up route configured:  
PC > ping 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=11ms TTL=253  
Request timed out  
Reply from 40.0.0.3: bytes=32 time=11ms TTL=253  
Ping statistics for 40.0.0.3:  
Packets: Sent=4, Received=2, Loss=2 (50% loss)

4) PC > ping 40.0.0.1  
Pinging 40.0.0.1 with 32 bytes of data:  
Reply from 40.0.0.1: bytes=32 time=20ms TTL=253  
Reply from 40.0.0.1: bytes=32 time=20ms TTL=253  
Reply from 40.0.0.1: bytes=32 time=20ms TTL=253

Router(Config)# ip route 0.0.0.0 0.0.0.0 20.0.0.1

→ Connections are checked by pinging packets from one end-device to another via interfaces.

• Observation:

→ The packets suffer a loss when the gateways of the end devices aren't set (request timed out)

→ When a default route isn't set, but gateway is set, the packets pinged from a PC will not recognize an interface router (results in destination host unreachable)

→ Only 50% of the packets are transferred to a router in the middle. The end-device will receive all the packets. ~~At~~ the middle routers will have ~~two~~ default routes, 50% of packets are sent in both direction. Hence 50% of packet is lost, because those packets aren't returned.

→ While pinging to destination end device, one of the packets sent first via the switch, the switch receives the destination IP address of required end device. The rest of the packets are sent to the required end device. Therefore the first packet will get timed out.

• Output:

1) Gateway not set / ip route not configured

ping PC > ping 20.0.0.1

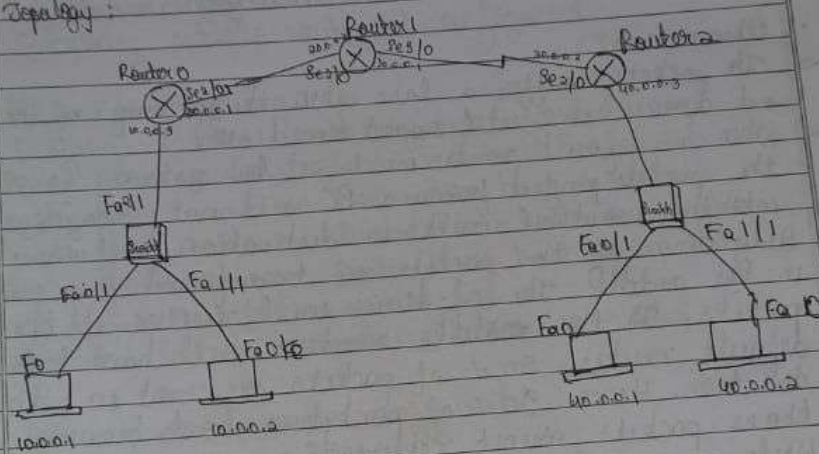
Pinging 20.0.0.1 with 32 bytes of data

Request timed out



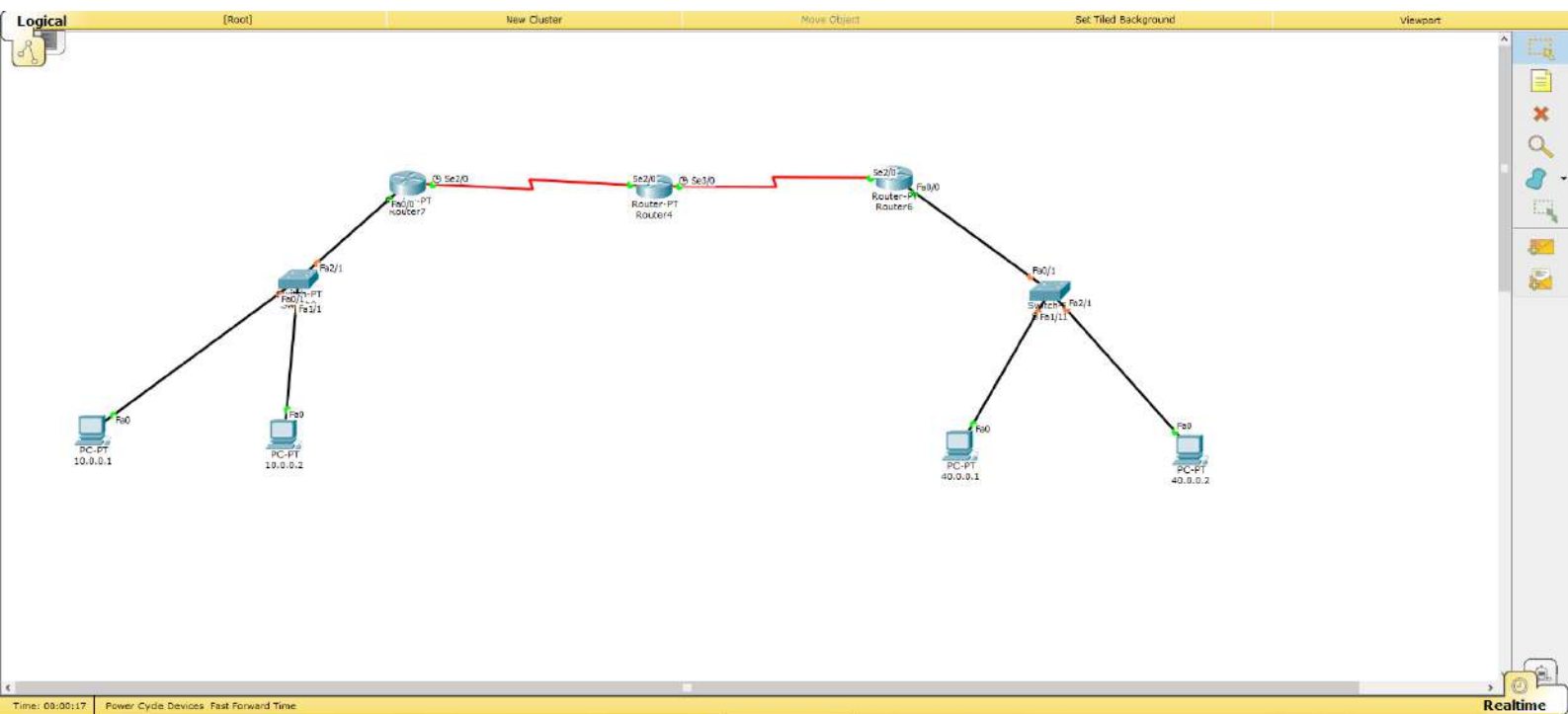
## LAB-Experiment-4

- 1. Aim: To configure default route to routers
- 2. Topology:



## Procedure:

- Construct topology using end devices, switch & routers
  - Set IP addresses of all end devices. Networks connected to same switch must have same network id.
  - The gateway address of the end devices are set.
  - Default route IP addresses of interfaces in routers are set
  - Default IP route is set by using following command.
- Router > enable  
Router # config terminal



ip address. Once it receives ip address ~~for the~~ from the destination device, the packets are sent without any loss.

- RIP is a dynamic routing protocol that uses hop count as a routing ~~technique~~ metric to find the best path between the source and destination.

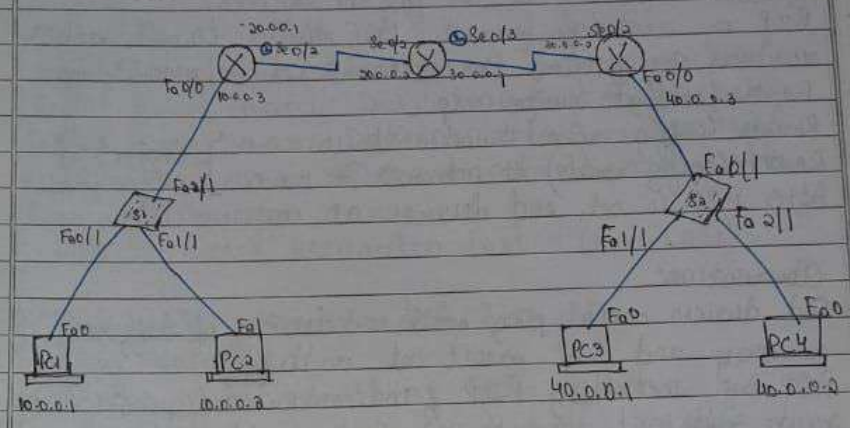
Neelam  
8/1/2022



### LAB-Experiment-5

**Aim:** Configure RIPv2 (Routing Information Protocol) in Routers

**Top. Topology:**



**Procedure:**

- Construct a network topology as shown.
- Configure the end devices and set their IP addresses.
- The routers connected to each other with a serial clock connection.
- Point to Point (PPP) protocol is used to encapsulate all network connections within routers. The routers are configured with IP addresses before this.
- Clock rate for the router is set to only those for which the serial clock connection begins.

```

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

```

- Sp route will not be set for the routers.
- RIP is used to connect the router to all other routers by giving the network id information
- Router(config)# router rip
- Router (config-router) # network 10.0.0.0 } Connected
- Router (config-router) # network 20.0.0.0 } networks
- After RIP is set, end devices can communicate.

#### • Observation:

- End devices cannot ping ~~each~~ and devices of different if gateway and RIP isn't set with ppp.
- ~~RIP~~ By including RIP in router configuration, ip route ~~is~~ not need to set because the networks will be connected
- If RIP or gateway isn't set, we get two types of responses; Request timed out and destination host unreachable

#### • Result:

- 1) When gateway or RIP isn't set:

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)

ii) When only gateway is set & RIP can't set (PPP is set)

> ping 30.0.0.1

Pinging 30.0.0.1 with 32 bytes of data:

Reply from 30.0.0.1: Destination host unreachable

Reply from 30.0.0.1: Destination host unreachable

Reply from 30.0.0.1: Destination host unreachable

Reply from 30.0.0.1: Destination host unreachable

Ping statistics for 30.0.0.1:

Packets: Sent=4, Received=0, Lost=4 (100% loss)

iii) When both gateway and RIP is set:

> Ping 40.0.0.1

Pinging 40.0.0.1 with 32 bytes of data:

Reply from 40.0.0.1: bytes=32 time=19ms TTL=128

Reply from 40.0.0.1: bytes=32 time=16ms TTL=128

Reply from 40.0.0.1: bytes=32 time=30ms TTL=128

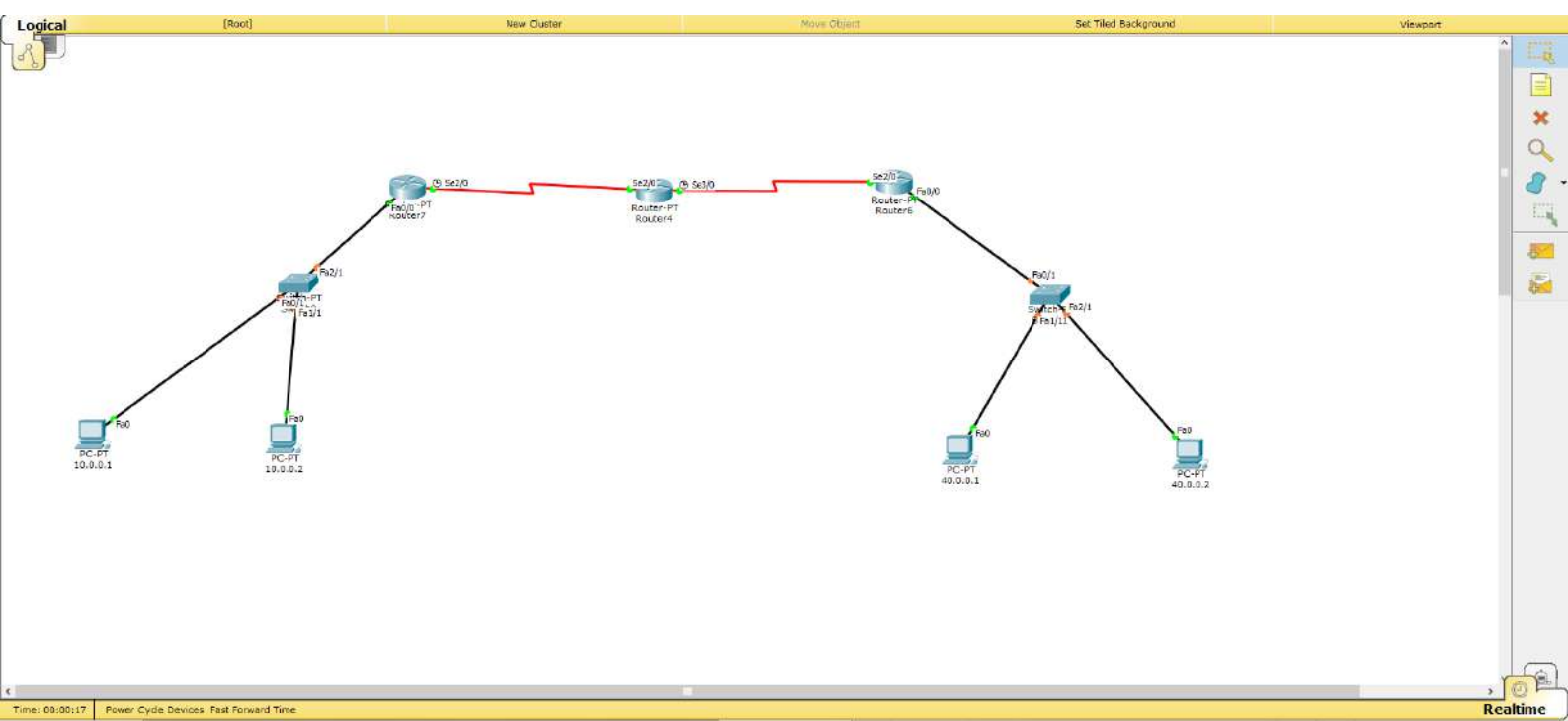
Reply from 40.0.0.1: bytes=32 time=3ms TTL=128

Ping statistics for 40.0.0.1:

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

Note: While pinging the end device for the first time after RIP configuration, the first packet will be lost because the switch will not know the device which is connected to the destination





see server:

Observation:

- When DHCP server configuration is set for end devices, a dynamic IP address is automatically set. This happens to all the end devices.
- Ping message from one device to another will have 0% loss

Result:

- 1) PC > ping 10.0.0.2  
pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time=0ms TTL=128

Reply from 10.0.0.2: bytes=32 time=0ms TTL=128

Reply from 10.0.0.2: bytes=32 time=0ms TTL=128

Reply from 10.0.0.2: bytes=32 time=0ms TTL=128

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

- 2) SERVER > ping 10.0.0.1  
pinging 10.0.0.1 with 32 bytes of data:

Reply from 10.0.0.1: bytes=32 time=0ms TTL=128

Reply from 10.0.0.1: bytes=32 time=0ms TTL=128

Reply from 10.0.0.1: bytes=32 time=0ms TTL=128

Reply from 10.0.0.1: bytes=32 time=0ms TTL=128

Ping statistics:

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

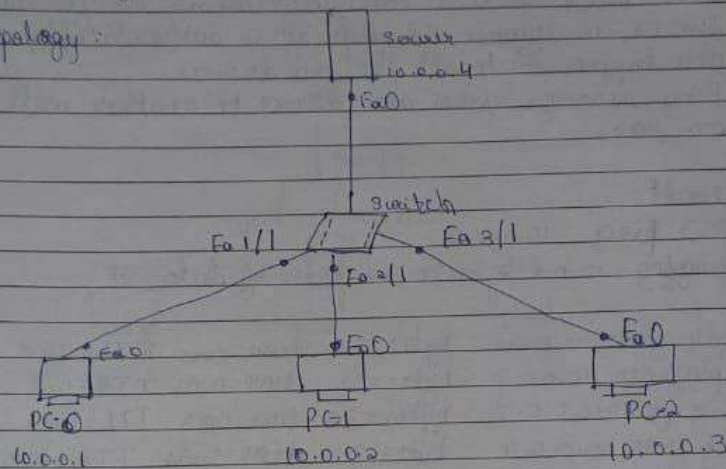
Neelima  
15/12/2022



## Lab Experiment - 6

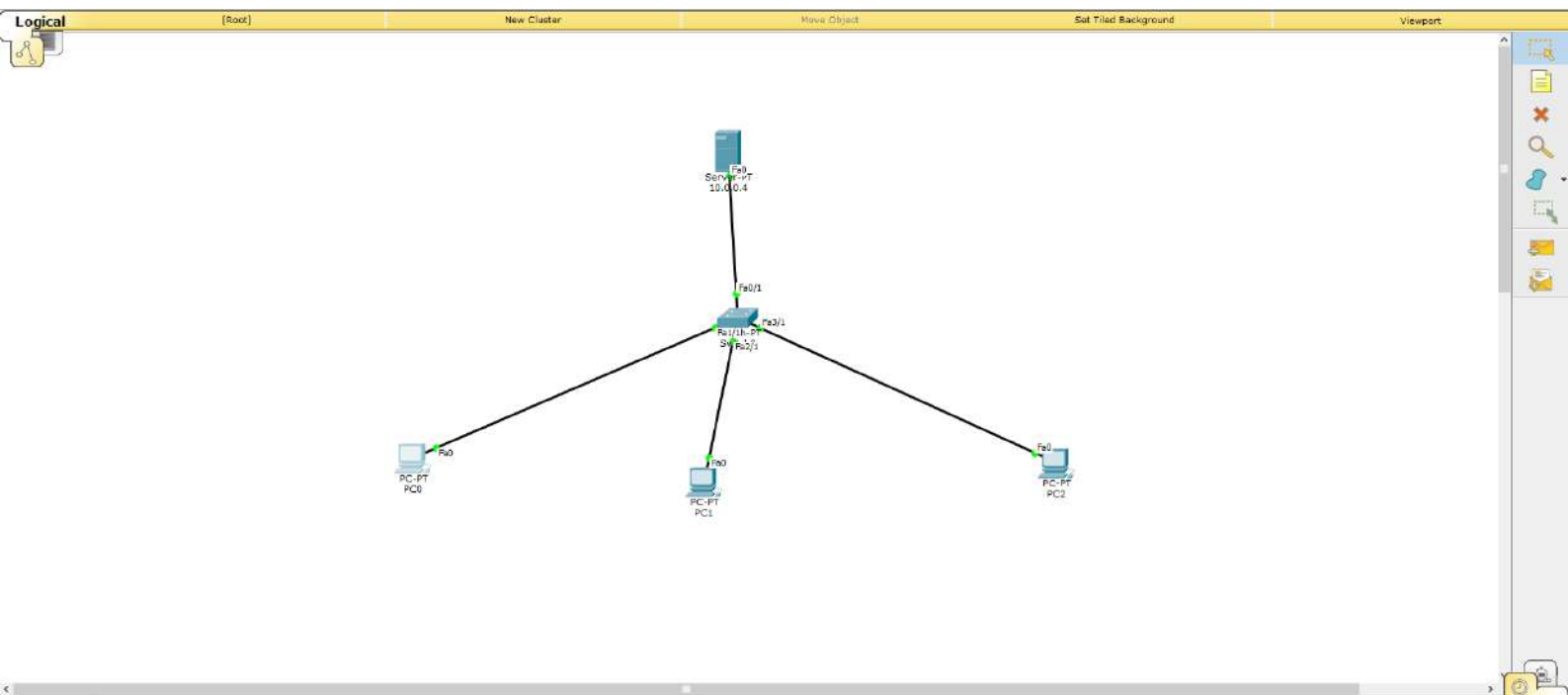
• Aim: Configure DHCP to server

• Topology:



• Procedure:

- Construct the above topology.
- Configure IP address of switch as (10.0.0.4)
- Open services on server and switch on DHCP service.
- Give a new static IP address which can't same as IP address of server.
- In end device module, switch from static to DHCP in IP configuration.
- Perform this operation to all the end devices connected to the switch.
- Ping a message from the end devices to server.



### Observation:-

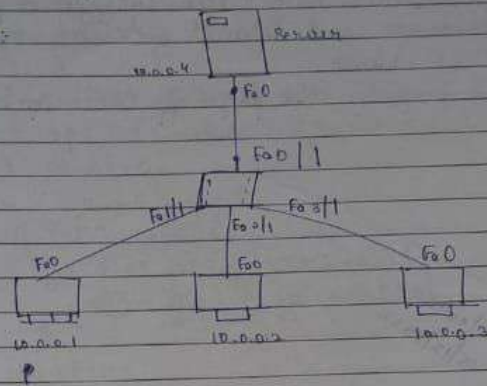
- Web browser module is opened in the end device and the set domain name ([www.kimcc.com](http://www.kimcc.com)) is entered.
- If the system/server hasn't been configured properly, i.e., set DNS server and default gateway, the 'Host Unresolved' is shown.
- If configured properly, the page of Cisco Packet Tracer is opened.

Aladdin  
15/12/2022

## Lab Experiment-7

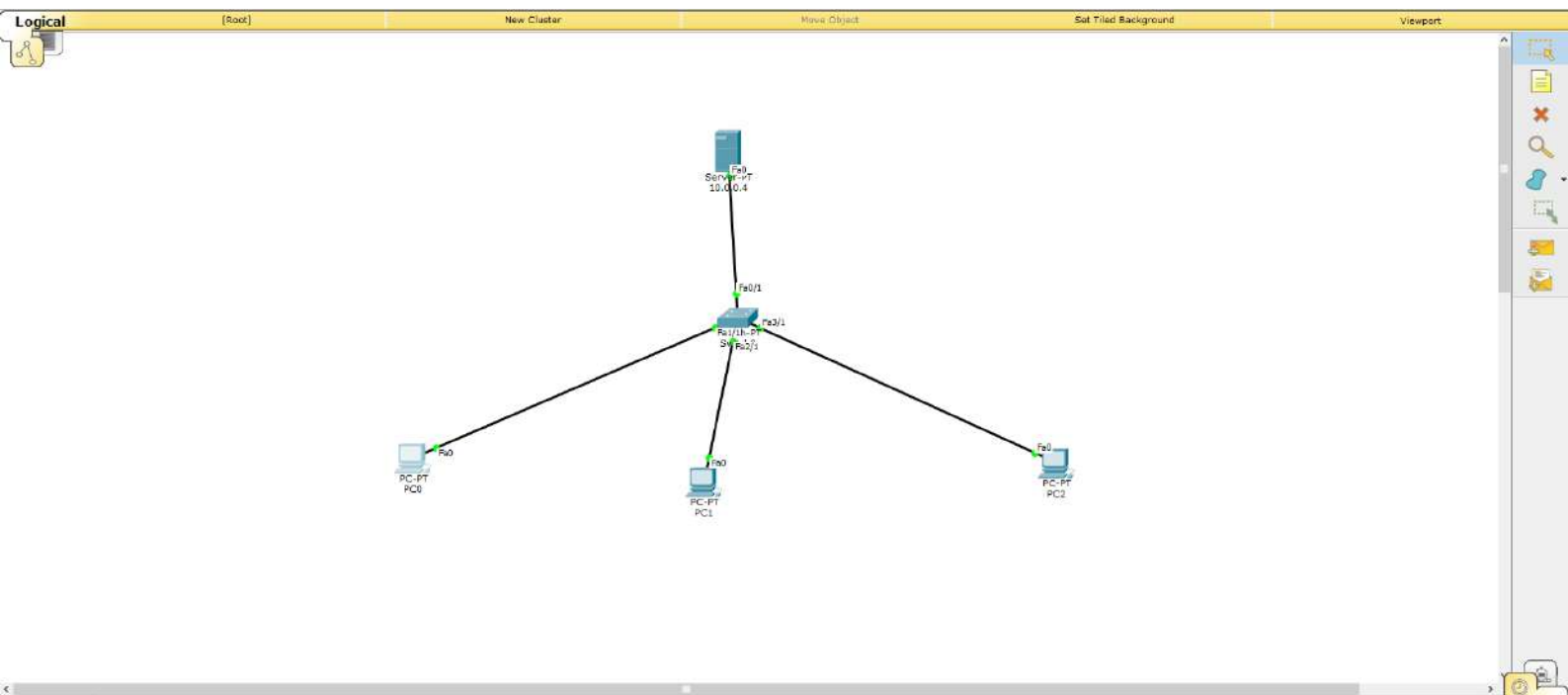
- Aim- To configure Web server and DNS server.

- Topology:



- Procedure:

- Construct the topology shown.
- Configure IP address of server and open services.
- IP address DHCP setting is switched on through which end devices get a dynamic IP address.
- End devices are configured.
- In services module, select DNS and switch on the DNS service.
- Enter a domain name (www.bmccac.com) and set address to IP address of server.
- Add and save the domain.
- In DHCP services, change the default gateway and DNS server to IP address of server.





# LAB Experiment 8

Write a program for error detection using CRC 16-bit

```
import java.util.*;
class CRC {
    void div (int a[], int k) {
        int gp[] = {1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1};
        int count = 0;
        for (i = 0; i < k; i++) {
            if (a[i] == gp[i]) {
                for (int j = i; j < 17; j++) {
                    a[j] = gp[j] ^ gp[count++];
                }
            }
            count++;
        }
    }
}
```

```
public static void main (String args[]) {
    int a[] = new int[50];
    int b[] = new int[50];
    int len, k;
    CRC ob = new CRC();
    System.out.println ("Enter data word");
    Scanner sc = new Scanner (System.in);
    len = sc.nextInt();
    int flag = 0;
    for (int i = 0; i < len; i++) {

```

```

    a[i] = sc.nextInt();
}
for (int i=0; i<16; i++) {
    a[i] = 0;
}
x = len-16;
for (int i=0; i<len; i++) {
    b[i] = a[i];
}
ob.ans(a, b);
for (int i=0; i<len; i++) {
    a[i] = a[i] ^ b[i];
}
System.out.print(a[i] + " ");
}

for (int i=0; i<len; i++)
    a[i] = scan.nextInt();
ob.ans(a, k);
for (int i=0; i<len; i++) {
    if (a[i] != 0) {
        flag = 1;
        break;
    }
}

if (flag == 1) {
    System.out.println("Error");
} else {
    System.out.println("No error");
}
}
}

```

Output:

Enter length of data frame:

7

Enter message:

10 11101

10 11101 100010 110 1011000

10 11101 100 010 110 1011000

No error

21/11/23

```
C:\Users\deept\OneDrive\Desktop>java CRC.java
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 1 1 0 1 0 0 1 0 1 0 0 0 0
error in data
```



```

else
    if (g[u] < g[v])
        g[u] = g[v];
        p[u] = p[v];
        g[u] = 0;
    if (g[u] > g[v])
        g[v] = g[u];
        p[v] = p[u];
        g[v] = 0;
    if (g[u] == g[v])
        p[u] = p[v];
    count = 1;
    while (

```

Enter source: 1

Vertex 1	→ cost = 0	parent = 1
Vertex 2	→ cost = 6	parent = 2
Vertex 3	→ cost = 5	parent = 2
Vertex 4	→ cost = 9	parent = 3
Vertex 5	→ cost = 8	parent = 3

No negative weight cycle

12/1/2023

```

return flag;
}

```

```

void main() {
    int V, edge[20][20], GC[20][20], i, j, R=0;
    printf("Enter no. of vertices:");
    scanf("%d", &V);
    printf("Enter graph: \n");
    for (i=0; i<V; i++) {
        for (j=0; j<V; j++) {
            scanf("%d", &GC[i][j]);
            if (GC[i][j] != 0) {
                edge[R][i] = i;
                edge[R][j] = j;
            }
        }
        if (BellmanFord(G, V, k, edge))
            printf("No negative weight cycle \n");
        else
            printf("A Negative weight cycle exists \n");
        R++;
    }
}

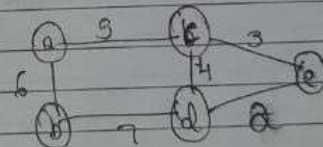
```

Output:

Enter no. of vertices: 5

Enter graph in matrix form:

0	6	3	10000	10000
6	0	10000	7	100000
3	10000	0	4	3
10000	7	4	0	2
10000	10000	3	2	0



12/1/23

Lab experiment - 10

```
#include <stdio.h>
#include <stdlib.h>
int Bellman_Ford (int G[20][20], int V, int e, int edge[20][20])
{
    int i, u, v, k, d[20], p[20], o, flag=1;
    for(i=0; i<V; i++) {
        d[i] = 1000;
        p[i] = -1;
    }
    printf("Enter source: ");
    scanf("%d", &o);
    d[o] = 0;
    for(i=0; i<V-1; i++) {
        for(k=0; k<e; k++) {
            u = edge[k][0], v = edge[k][1];
            if (distance[u] + G[u][v] < distance[v]) {
                distance[v] = distance[u] + G[u][v];
                p[v] = u;
            }
        }
    }
    for(k=0; k<e; k++) {
        u = edge[k][0], v = edge[k][1];
        if (d[u] + G[u][v] < d[v])
            flag = 0;
    }
    if (flag) {
        for(i=0; i<V; i++) {
            printf("Vertex %d -> cost = %d, parent = (%d,%d)\n",
                i+1, d[i], p[i]);
        }
    }
}
```

```
Enter the number of vertices: 4
Enter the source vertex of the graph: 1

Enter no. of edges: 5

For edge 1=>
Enter source vertex :1
Enter destination vertex :2
Enter weight :4

For edge 2=>
Enter source vertex :1
Enter destination vertex :3
Enter weight :5

For edge 3=>
Enter source vertex :3
Enter destination vertex :2
Enter weight :7

For edge 4=>
Enter source vertex :2
Enter destination vertex :4
Enter weight :7

For edge 5=>
Enter source vertex :4
Enter destination vertex :3
Enter weight :-15

NEGATIVE CYCLE PRESENT..!!
```



```

j = i;
do {
    j = pred[j];
    printf("%d <= %d", j, i);
} while (j != source);
}
}

```

Output:

Enter no. of vertices: 5  
 Enter adjacency matrix:

0	6	5	0	0
6	0	0	7	0
5	0	0	4	3
0	7	4	0	2
0	0	3	2	0

Enter starting node: 1

Distance of 0 = 6

Path 0  $\leftarrow$  1

Distance of 2 = 11

Path = 2  $\leftarrow$  0  $\leftarrow$  1

Distance of 3 = 7

Path = 3  $\leftarrow$  1

Distance of 4 = 9

Path = 4  $\leftarrow$  3  $\leftarrow$  1

*N/A*  
 12/12/2023

## Lab Experiment -11

```

#include <stdio.h>
#include <conio.h>
#define INFINITY 9999;
#define MAX 10;

void dijkstra(int G[MAX][MAX], int n, int snode);
int main() {
    int G[MAX][MAX], i, j, n, u;
    printf("Enter no. of vertices: ");
    scanf("%d", &n);
    printf("Enter adjacency matrix: \n");
    for(i=0; i<n; i++) {
        for(j=0; j<n; j++) {
            scanf("%d", &G[i][j]);
        }
    }
    printf("Enter starting node: ");
    scanf("%d", &u);
    dijkstra(G, n, u);
    return 0;
}

```

```

void dijkstra(int G[MAX][MAX], int n, int snode) {
    int C[MAX][MAX], d[MAX], p[MAX];
    int u[MAX], count, mindis, minnode, i, j;
    for(i=0; i<n; i++) {
        for(j=0; j<n; j++) {
            if (G[i][j] == 0)
                C[i][j] = INFINITY;
        }
    }
}

```

```

else
    c[i][q] = q[i][q];
    for (i = 0; i < n; i++) {
        d[i] = c[onode][i];
        p[i] = onode;
        v[i] = 0;
    }
}
d[onode] = 0;
v[onode] = 1;
count = 1;
while (count < n-1) {
    mindis = INFINITY;
    for (i = 0; i < n; i++) {
        if (d[i] < mindis && !v[i]) {
            mindis = d[i];
            nextnode = i;
        }
    }
    v[nextnode] = 1;
    for (i = 0; i < n; i++) {
        if (!v[i]) {
            if (mindis + c[nextnode][i] < d[i]) {
                d[i] = mindis + c[nextnode][i];
                p[i] = nextnode;
            }
        }
    }
    count++;
}
for (i = 0; i < n; i++) {
    if (i != snode) {
        printf("In Distance of %d d = %d, i = %d\n", d[i], i, d[i]);
        printf("Path = %d, i = %d\n", i, i);
    }
}

```

Enter the no. of vertices: 5

Enter the adjacency matrix:

0 3 1 0 0

3 0 7 5 1

1 7 0 2 0

0 5 2 0 7

0 1 0 7 0

Enter the starting node: 0

Distance of 1 = 3

Path = 1 <- 0

Distance of 2 = 1

Path = 2 <- 0

Distance of 3 = 3

Path = 3 <- 2 <- 0

Distance of 4 = 4

Path = 4 <- 1 <- 0