VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Computer Networking Lab

Submitted by

Neha L(1BM21CS405)

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 Lab**" carried out by **Neha L (1BM21CS405),** 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 **Course Title - (Course code)** work prescribed for the said degree.

Lab-Incharge : **Dr. Jyothi S Nayak**

Lakshmi Neelima
Department of CSE
BMSCE, Bengaluru

Professor and Head
Department of CSE
BMSCE, Bengaluru

Index

Cycle 1:

Program1:

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

Observation:

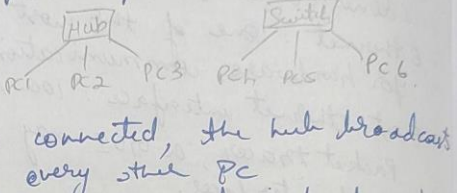
Aim: week-1 10/11/22

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

Procedure:

- connect 4 generic PC's to ~~hub~~ connecting devices
- Configure the IP addresses of all PCs
- Send a simple PDU from source PC to destination PC
- In simulation mode, press capture / Forward to see the working

Topology: Star Topology



Observation:

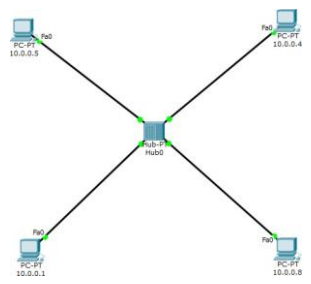
→ When the PCs are connected, the hub broadcasts the recieve PDU to every other PC. The hub recieves the PDU and simply transfers it to next layers and then broadcast it to other PC.
If the destination address matches, The PC accepts the PDU.

→ When switches are used, initially switch broadcasts the PDU to every other PC and forms the switch table.
In subsequent transfers, switch unicasts the PDU to the particullar PC with destination address using switch table.
When switch recieves a PDU, it de-capsulates the PDU check the destination address and encapsulates again and then transfer it to specific PC.

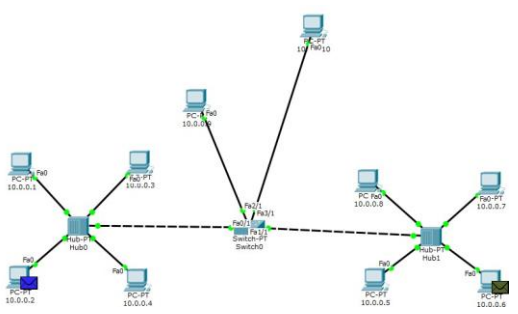
Result: Observed and analysed the working of hubs and switches.

Nadim 13/11/2022

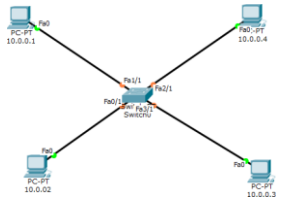
Screenshots:



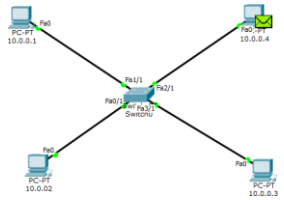
Hub Topology



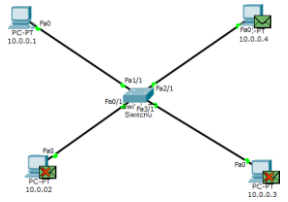
Hub-Switch Topology



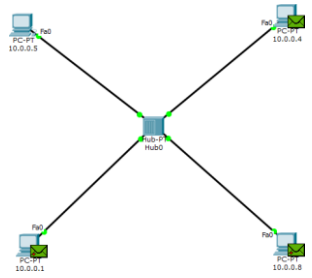
Switch Topology



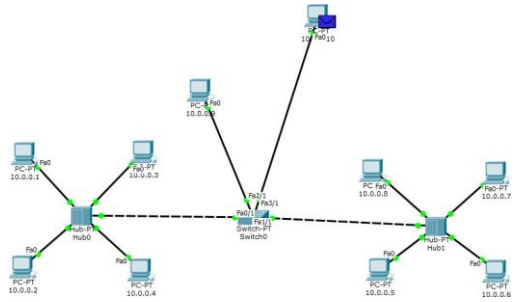
Switch after forming ARP



Switch before forming ARP



Hub broadcasting



Unicasting switch

Program 2:

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

Observation:

Week-2. 17/11/22

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

Procedure

- Connect the generic PC's to the router.
- Configure the IP addresses of router for each interface.
- Ping ~~mess~~ ~~mess~~ other IP addresses to see ~~response~~ responses.

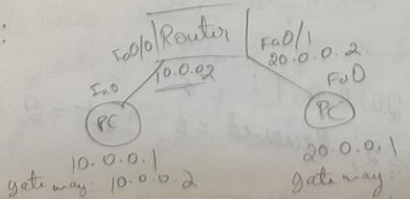
Observation:

- Configured the IP address of the router using following command

```
enable
config terminal
interface Fa0/0
ip address 10.0.0.2 255.0.0.0
no shutdown
exit
```

- After configuring, we are unable to ping the destination address of PC's of other networks. Response we are getting is request timed out as gateway is not set.
- After setting gateway for the PC's, we get the reply back from the destination on pinging.
- If we were to ping a destination of different network which does not exist we get the response as Destination host unreachable.

→ Topology:



Result:
Observed and analysed ping responses.

Outcome:

- We got the following responses when pinged
  - → request timed out: we got this reply when we didn't set a gate way for that PC
  - → reply: After ~~getting~~ setting gateway for the PC, the ping was successful and we got a reply from the destination.
  - → destination host unreachable: when ~~we tried~~ the device ~~provide~~ specified in the destination IP address doesnot exist, we get a reply back from the router interface of that PC that the specified host is unreachable.

Example: • ping 20.0.0.1

pinging 20.0.0.1 with 32 bytes of data:

Reply from 20.0.0.1 : bytes = 32 time = 0ms TTL = 127

Reply from 20.0.0.1 : bytes = 32 time = 0ms TTL = 127

Ping statistics for 20.0.0.1

Packets: Sent = 2, Recieved = 2, Lost = 0 (0% Loss)

Approximate round trip times in milli-seconds:

Minimum = 0ms, Maximum = 0ms, Average = 0ms.

Nachiket 13/11/2022

- ping 20.0.0.1

pinging 20.0.0.1 with 32 bytes of data:

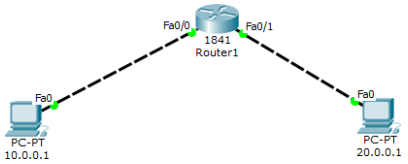
Request timed out

Request timed out.

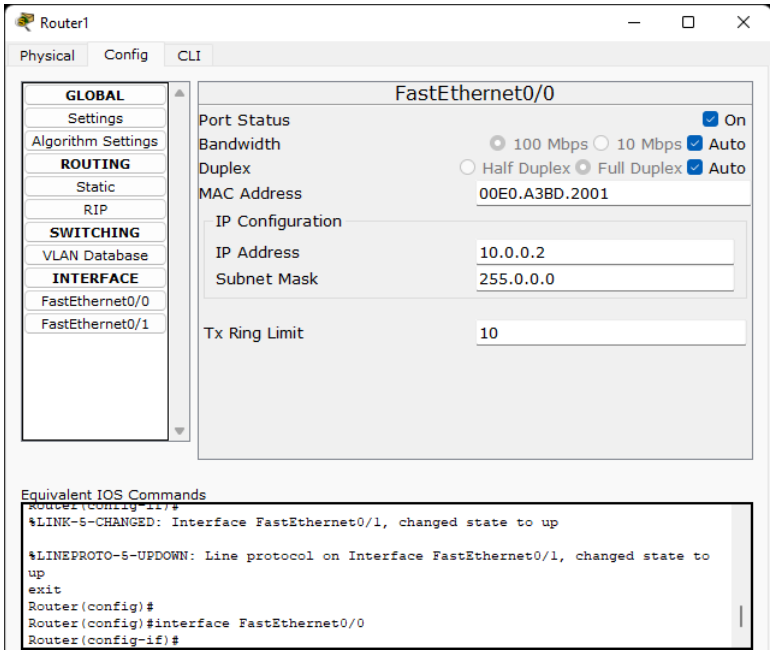
Ping statistics for 20.0.0.1:

Packages: Sent = 2, Recieved = 0, Lost = 2 (100% Loss)

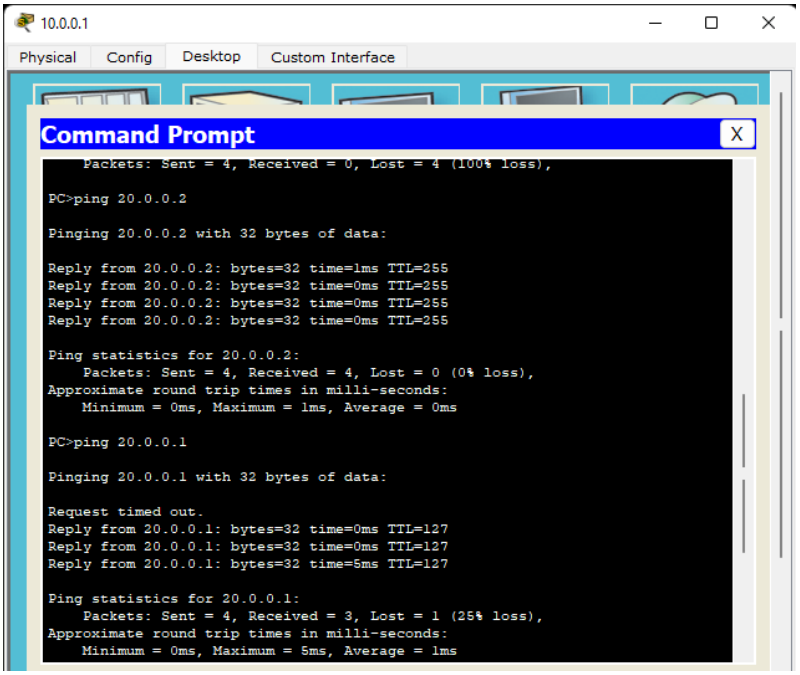
Screenshots:



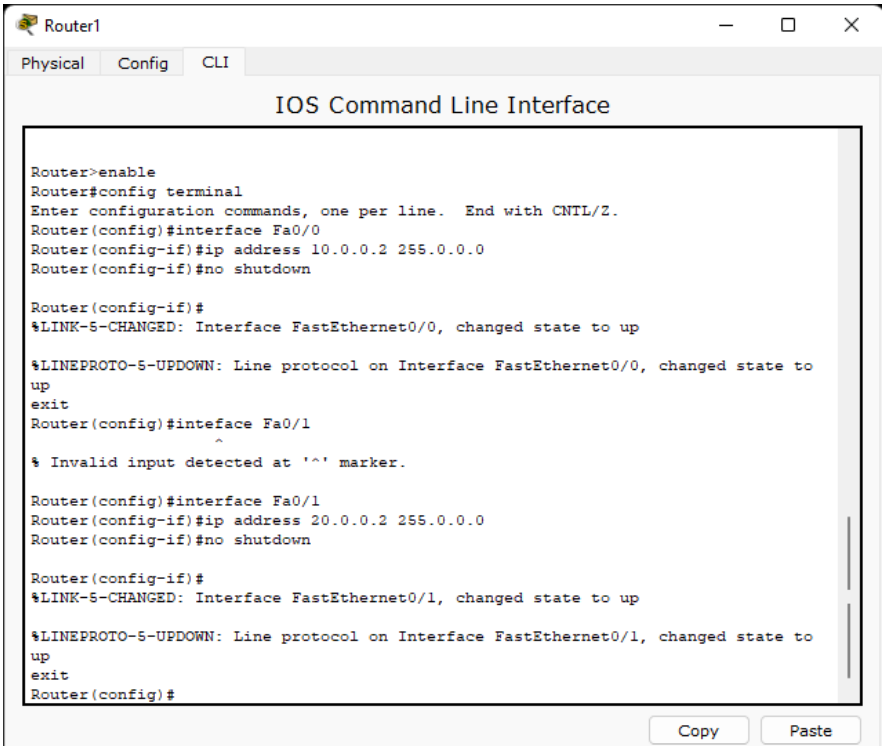
Topology



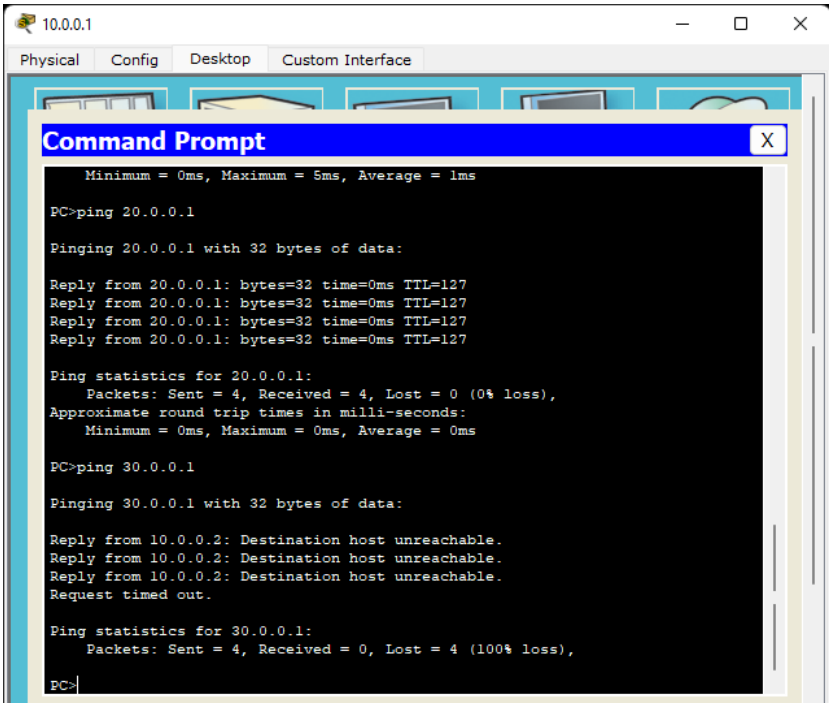
Configuring IP address for end device



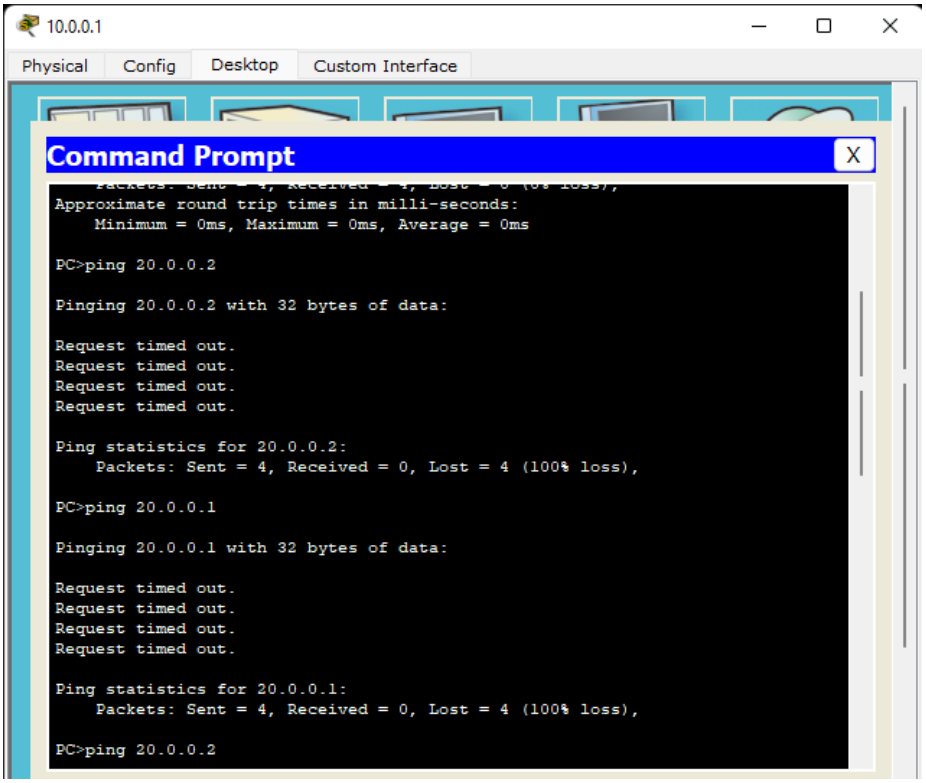
Successful ping message



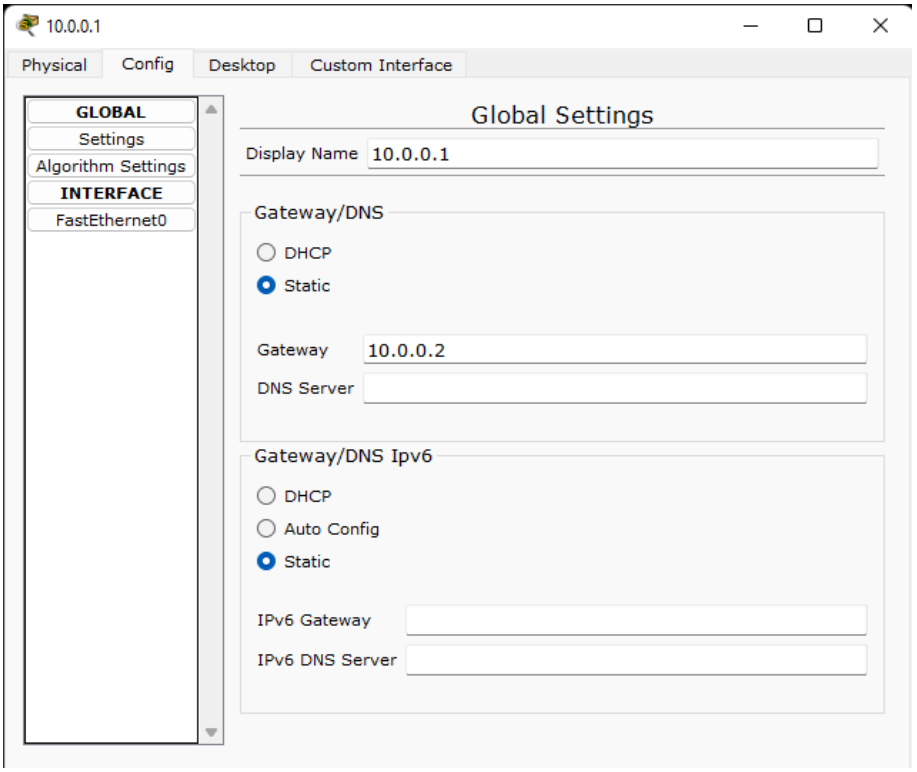
Configuring router ip address for the interface



Destination host unreachable message



Request timed out message



Configuring gateway for the end device

Program 3:

Configuring default route to the Router

Observation:

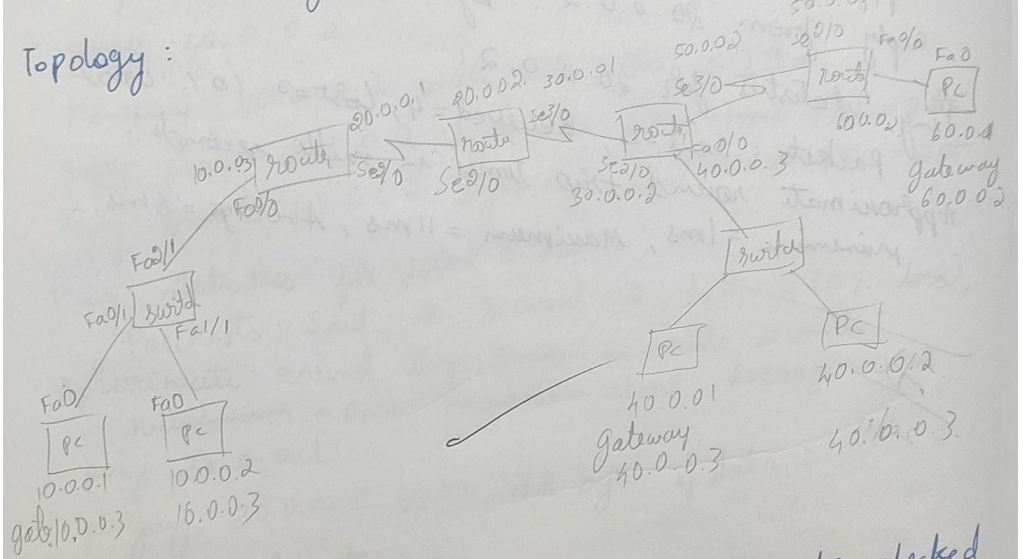
Week-4 1/12/22

Aim: Configuring default route to the Router

Procedure:

- Connect 2 end devices to 2 switches each
- Connect switches ~~routers~~ to the routers.
- Connect the 2 routers via another router
- Set the gateway for the end devices
- Set the static route for the end devices.
- Take another router connect it to the 3rd router and set a default route.

Topology:



Observation:

- The routes configured for a router can be checked using following command:

show ip route

20.0.0.0/8 is directly connected, Serial 2/0

30.0.0.0/8 is directly connected, Serial3/0

- The routes can be configured for a router using the following command

  ip route 10.0.0.0 255.0.0.0 20.0.0.1

To configure a default route:

  ip route 0.0.0.0 0.0.0.0 20.0.0.1

  any network, any subnet mask, hop

- when we ping to ~~30.0.0~~ 20.0.0.1 from 10.0.0.1 without setting gateway for 10.0.0.1 we get request timed out since the end device doesnot know which path to follow.
- After setting gateway and before setting route if we ping to 30.0.0.1 we get request timed out since the router doesnot know how to reach that network
- Before configuring the route for router 4 if we ping to 60.0.0.1 from 10.0.0.1 we get destination host unreachable because the router 3 is unaware of 60.0.0.0
- After we configure a default route for router 3 with hop as 50.0.0.1, we are ~~su~~ able to successfully ping ~~to~~ 60.0.0.1

Outcome:

Before configuring default route for router 3

ping 60.0.0.1

pinging 60.0.0.1 with 32 bytes of data:

Reply from 30.0.0.2 : Destination host unreachable.

Reply from 30.0.0.2 : Destination host unreachable

Ping statistics for 60.0.0.1 :
Packets : Sent = 4 , Received = 0, Lost = 4 (100% loss).

After configuring default route :

Ping 60.0.0.1

Pinging 60.0.0.1 with 32 bytes of data

Reply from 60.0.0.1 : bytes = 32 time = 30ms TTL = 124

Reply from 60.0.0.1 : bytes = 32 time = 9 ms TTL = 124

Ping statistics for 60.0.0.1 :
Packets : Sent = 4, Received = 4 , Lost = 0 (0% Loss),

Approximate round trip times in milli-seconds :
Min = 3ms , Max = 30ms , Avg = 14 ms.

→ Before setting gate way :

ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data :

Request timed out.

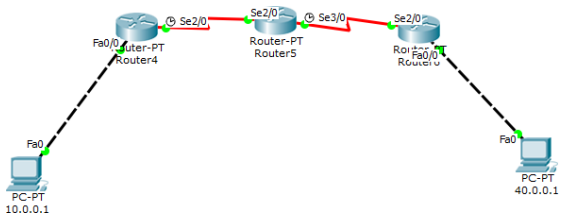
Request timed out.

Request timed out.

Ping statistics for 20.0.0.1
Packets : Sent = 4 , Received = 0 , Lost 4 (100% Loss)

1/12

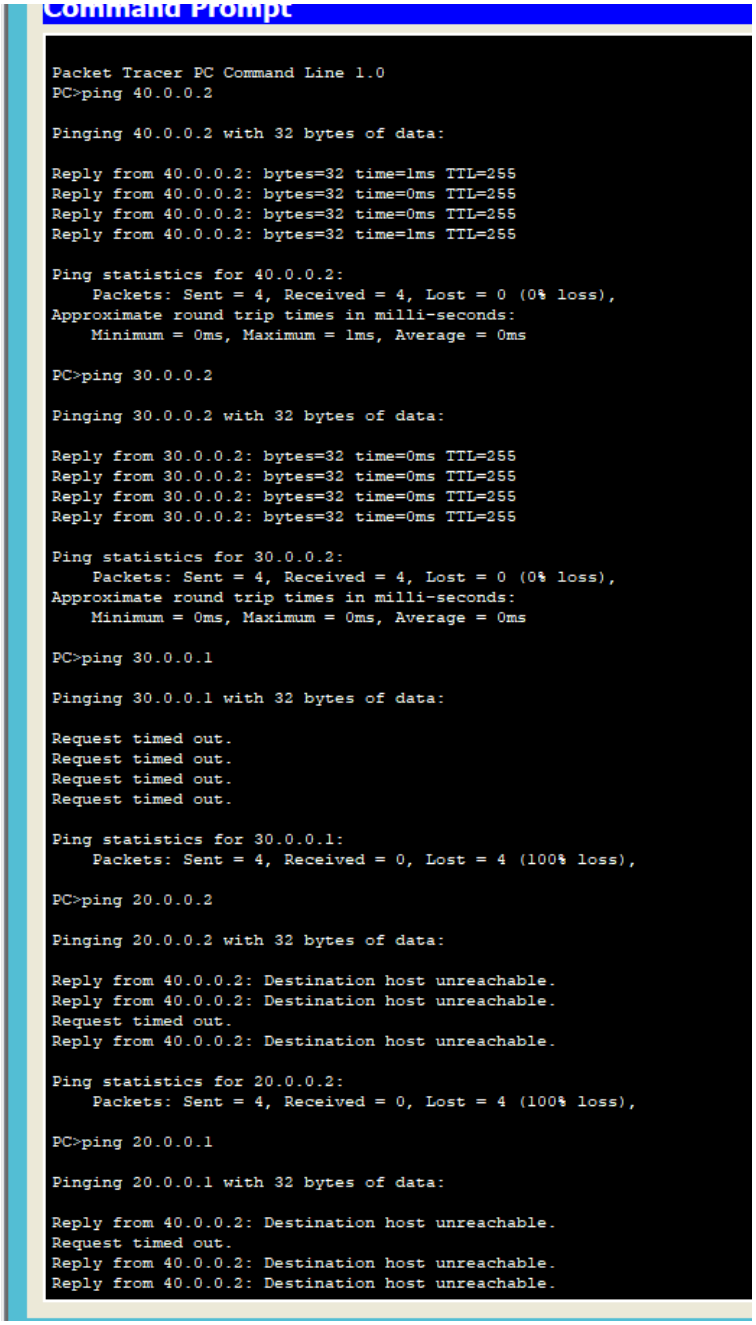
Screenshots:



Topology

```
Reply from 30.0.0.1: bytes=32 time=1ms TTL=254

Ping statistics for 30.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 9ms, Average = 4ms

PC>ping 20.0.0.2

Pinging 20.0.0.2 with 32 bytes of data:

Reply from 20.0.0.2: bytes=32 time=11ms TTL=254
Reply from 20.0.0.2: bytes=32 time=6ms TTL=254
Reply from 20.0.0.2: bytes=32 time=1ms TTL=254
Reply from 20.0.0.2: bytes=32 time=8ms TTL=254

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

PC>ping 20.0.0.1

Pinging 20.0.0.1 with 32 bytes of data:

Reply from 20.0.0.1: bytes=32 time=13ms TTL=253
Reply from 20.0.0.1: bytes=32 time=2ms TTL=253
Reply from 20.0.0.1: bytes=32 time=13ms TTL=253
Reply from 20.0.0.1: bytes=32 time=9ms TTL=253

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

PC>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

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

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

PC>ping 10.0.0.1

Pinging 10.0.0.1 with 32 bytes of data:

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

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

Successful ping messages



Request Timed out and Destination host unreachable

Router5

Physical Config CLI

IOS Command Line Interface

```
Press RETURN to get started!



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

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

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

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

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

Copy Paste

Configuring ip address for router interface

Router5

Physical Config CLI

IOS Command Line Interface

```
Router>
Router>enable
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip route 10.0.0.0 255.0.0.0 20.0.0.1
Router(config)#ip route 40.0.0.0 255.0.0.0 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 20.0.0.1
C    20.0.0.0/8 is directly connected, Serial2/0
C    30.0.0.0/8 is directly connected, Serial3/0
S    40.0.0.0/8 [1/0] via 30.0.0.2
Router#
```

Copy Paste

Configuring ip route

Program 4:

Configuring DHCP within a LAN in a packet Tracer

Observations:

Week - 6 15/12/22

Aim : Configuring DHCP

Procedure :

→ Connect 3 end devices to a switch.

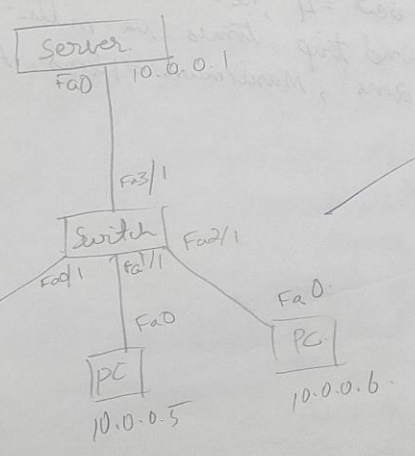
→ Connect switch to server and configure IP address of the server

→ Goto services of the server and switch on the DHCP service

→ Give a start IP address under same network in DHCP service.

→ Switch the IP configuration from static to DHCP in the end devices

Topology :



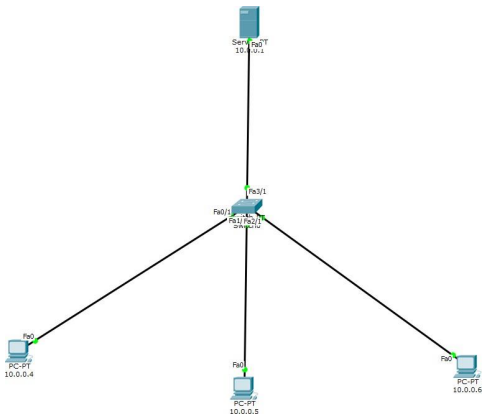
Observation :

→ Once the IP configuration mode is changed to DHCP the IP addresses of each PC are dynamically configured starting from start IP address of the server

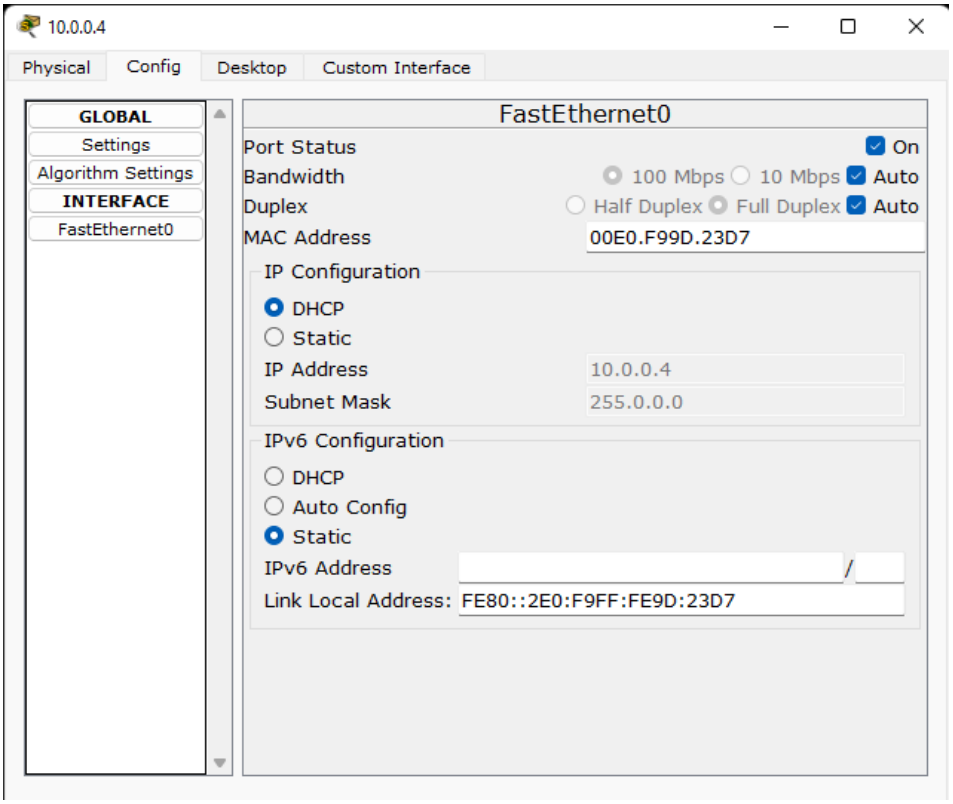
→ Result :

Dynamic Host Configuration Protocol (DHCP) is a server protocol that automatically provides the host with its IP address dynamically.

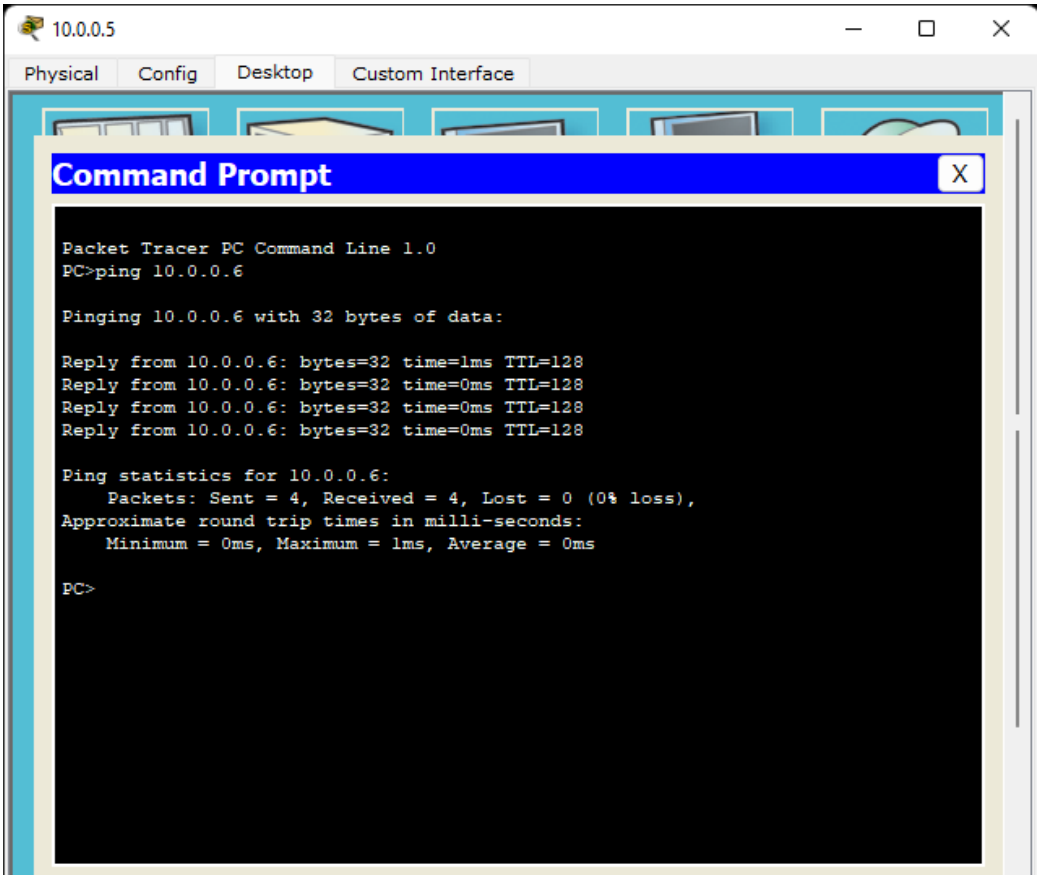
Screenshots:



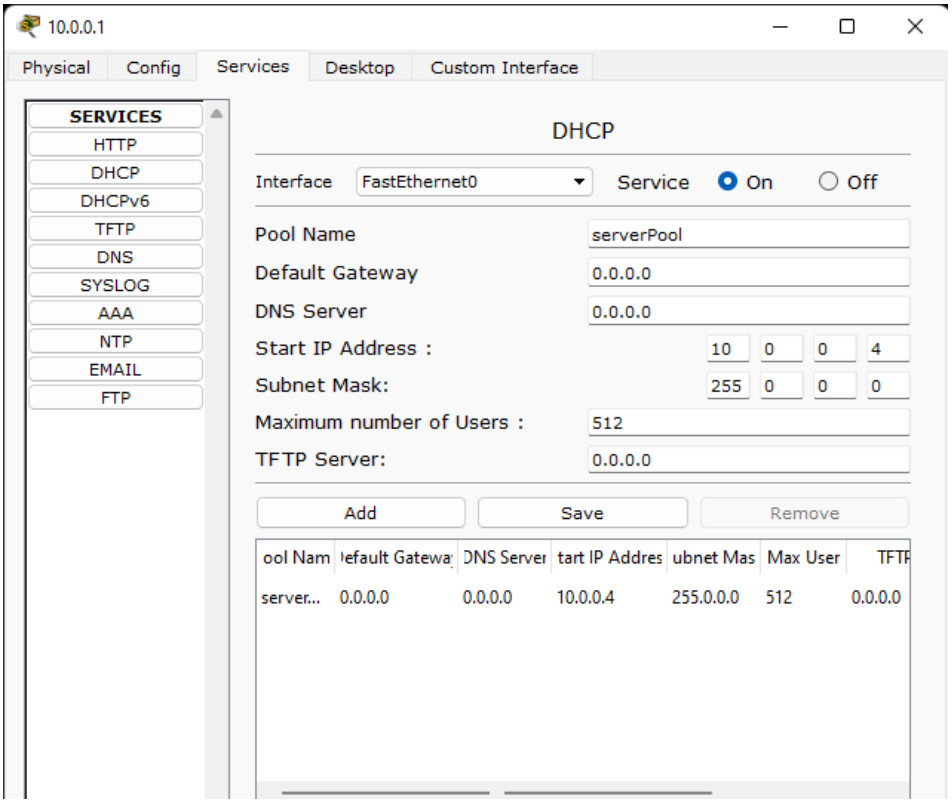
Topology



DHCP for end devices



Successful ping message



DHCP pool for the server

Program 5:

Configuring RIP Routing Protocol in Routers

Observation:

week-5 8/12/22

Aim: Configuring RIP routing Protocol in routers.

Procedure:

- Connect 2 end devices to 2 switches each and configure the ip address
- Then connect the 2 switches to 2 routers. And connect the routers via another router.
- Configure ip address of router-switch interface.
- Configure the ip address of router to router interface using point-to-point protocol.

```
enable
config terminal
interface Se 2/0
ip address 20.0.0.2 255.0.0.0
encapsulation PPP
clock rate 64000
no shut down
```

If the above is done for router 1, then no need of setting clock rate at router 2 interface.

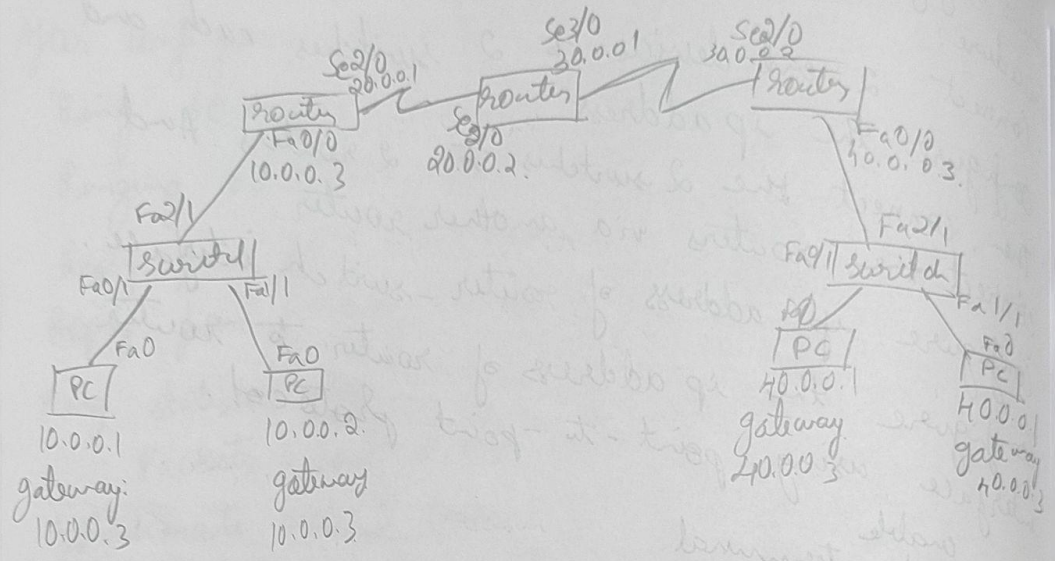
- Then we ~~hav~~ have to configure default routes to each router.

```
ip route 0.0.0.0 0.0.0.0 20.0.0.2
```

- Configure the rip protocol for each router using the following command. In config mode,

```
router rip
network 10.0.0.0
network 20.0.0.0
```

Topology



Observation:

→ We use PPP while configuring ip address of interface of 2 routers.
PPP is point-to-point protocol of the data link layer that is used to transmit multiprotocol data between two directly connected devices.
encapsulation PPP ⇒ This command encapsulates the datagram before it is transmitted and specifies the physical layer to transmit to.

→ RIP (router information protocol) is a distance vector protocol that uses hop count as its primary metric. It prevents routing loops by implementing a limit on the number of hops allowed in a path from source to destination.

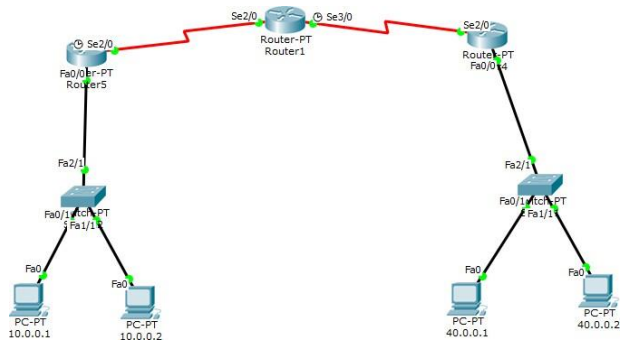
Outcome :
we get successful after configuring routes to the routers and setting gateway to end devices.

ping 40.0.0.2
pinging 40.0.0.2 with 32 bytes of data:
Request timed out.
Reply from 40.0.0.2 : bytes = 32 time = 12ms TTL = 125
Reply from 40.0.0.2 : bytes = 32 time = 2ms TTL = 125
Reply from 40.0.0.2 : bytes = 32 time = 12ms TTL = 125

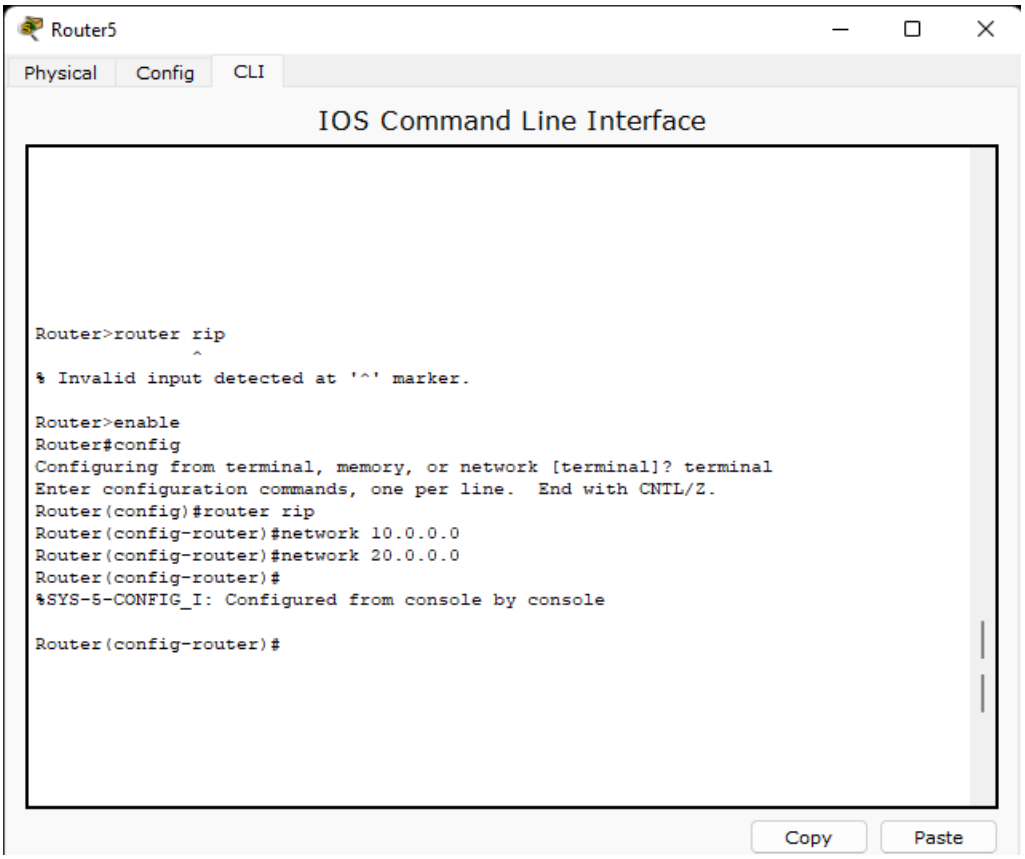
Ping statistics for 40.0.0.2 :
Packages : sent = 4 , Recieved = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds :
Minimum = 2ms , Maximum = 12ms , Average = 8ms

8/12

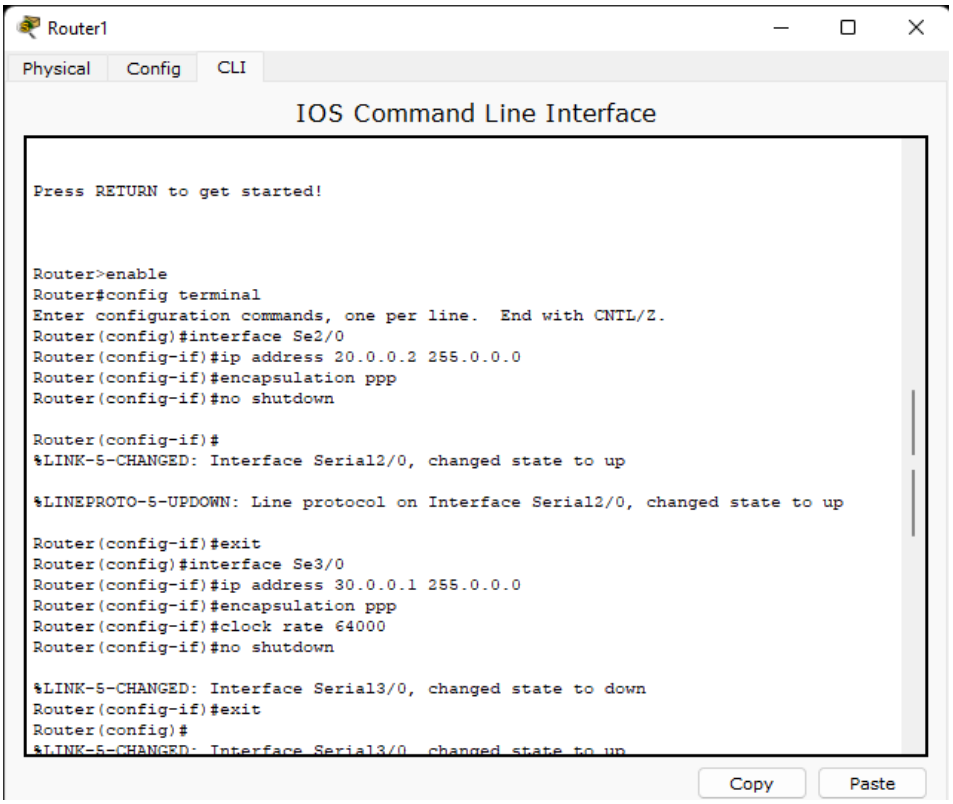
Screenshots:



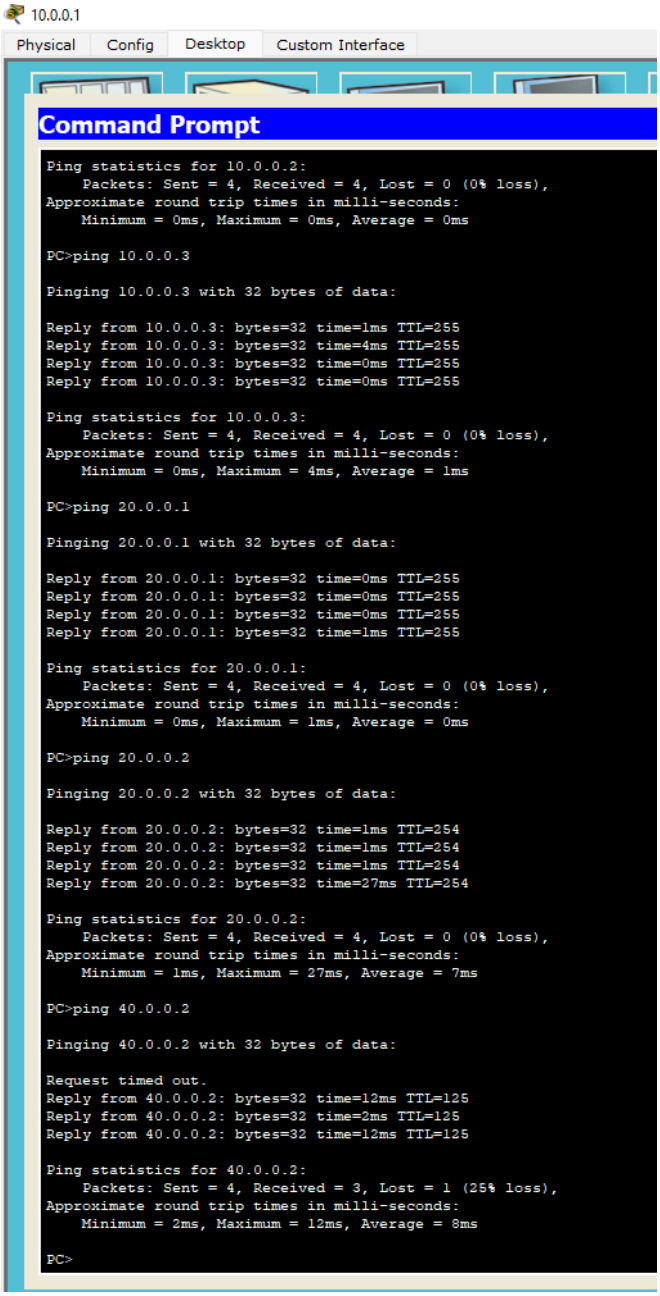
Topology



Configuring RIP for the router



Configuring ip address for the router interface



Successful ping messages

Program 7:

Demonstration of WEB server and DNS using Packet Tracer.

Observation:

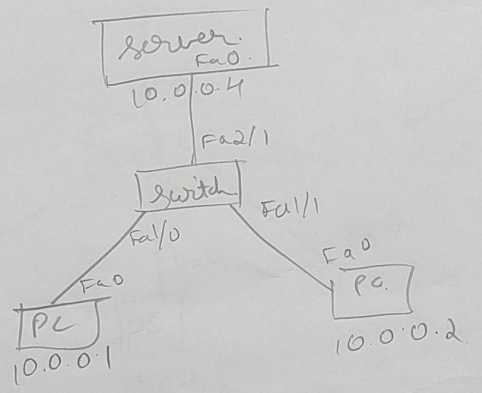
Week - 6 15/12/22

Aim: Demonstration of WEB server and DNS using Packet Tracer

Procedure:

→ Connect 2 end devices to a switch
→ Connect that switch to a server
→ Configure the IP address of the server
→ Configure the IP address of the end devices statically
→ Go to services of the server.
→ Switch on the HTTP service.
→ Switch on the DNS service.
→ Add a domain name like "www.bmsce.com" and the ip address of the server.

Topology:



Observation:

→ When we type the domain name that we added earlier in DNS service in the web browser of the end device we get the response in form of a website.

→ when we run the command nslookup www.bmsce.com in the command prompt, we get following response.

Server: [255.255.255.255]

Address: 255.255.255.255

Non-authoritative answer:

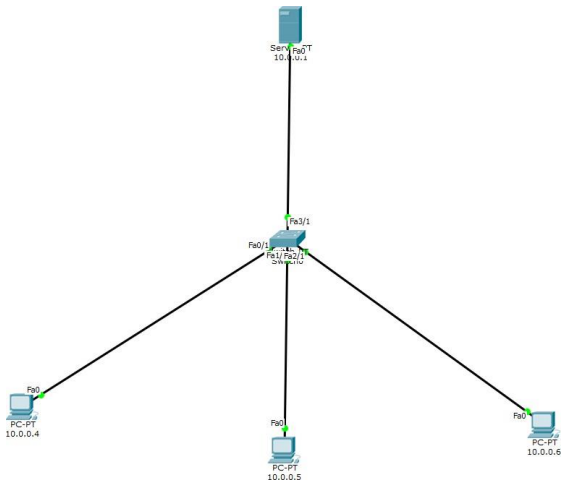
Name: www.bmsce.com

Address: 10.0.0.4

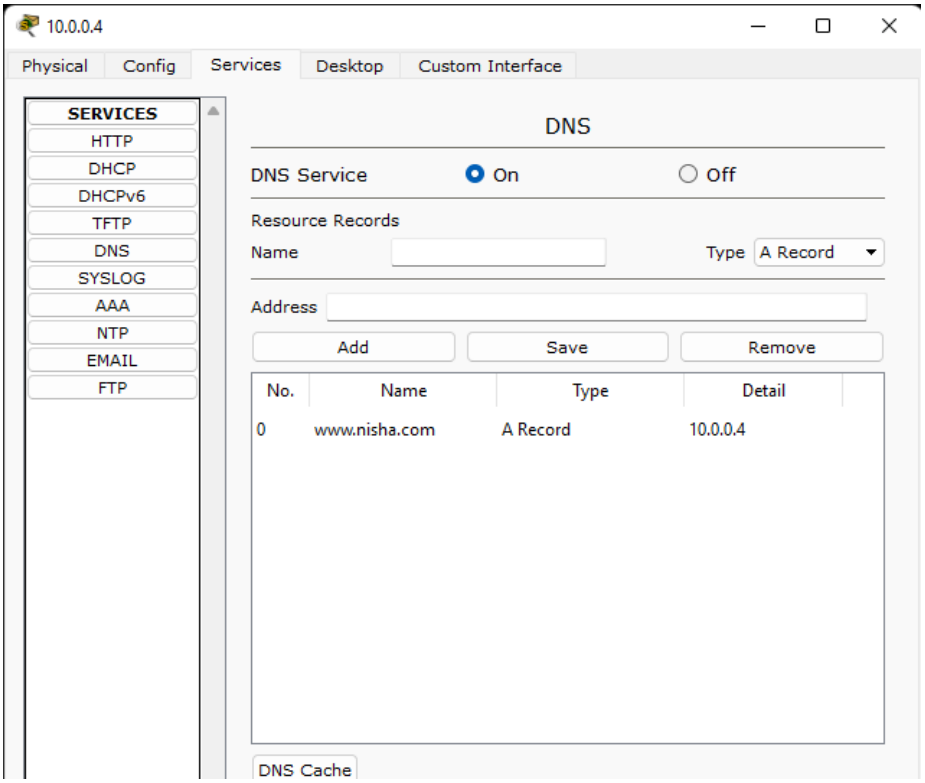
Result:

→ The Domain Name System (DNS) server is a server that is specifically used for matching website hostnames (like example.com) to their corresponding internet protocol or IP address

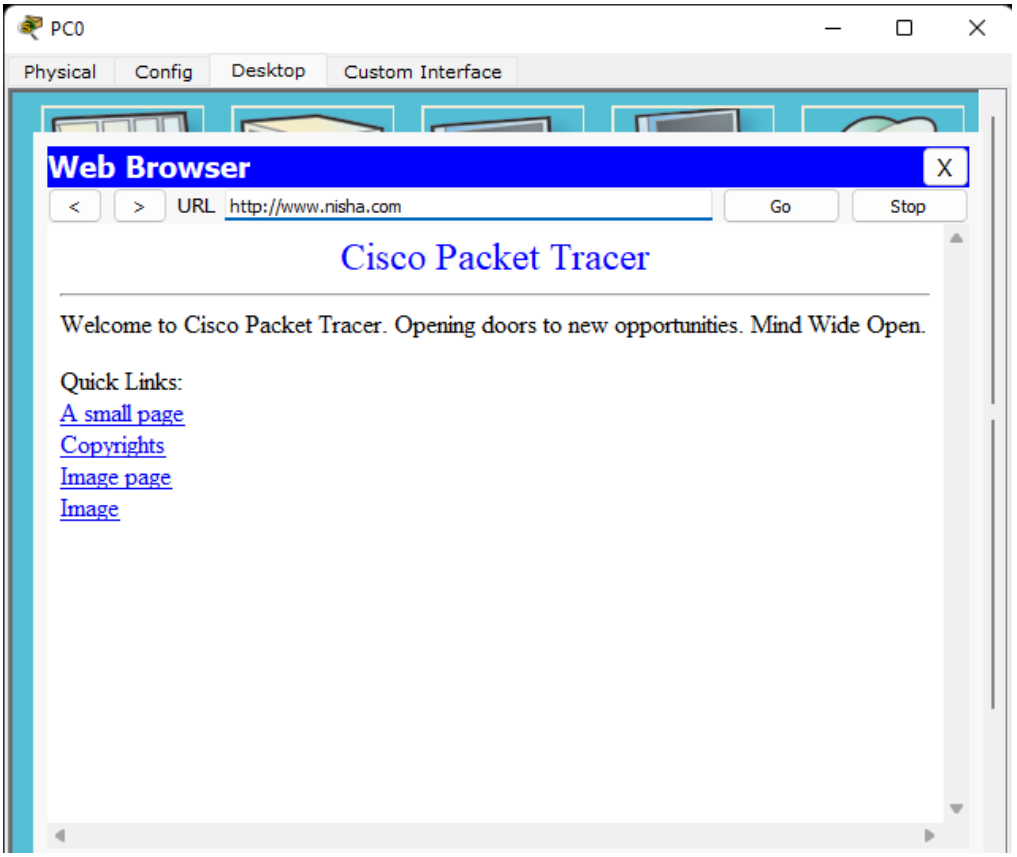
Screenshots:



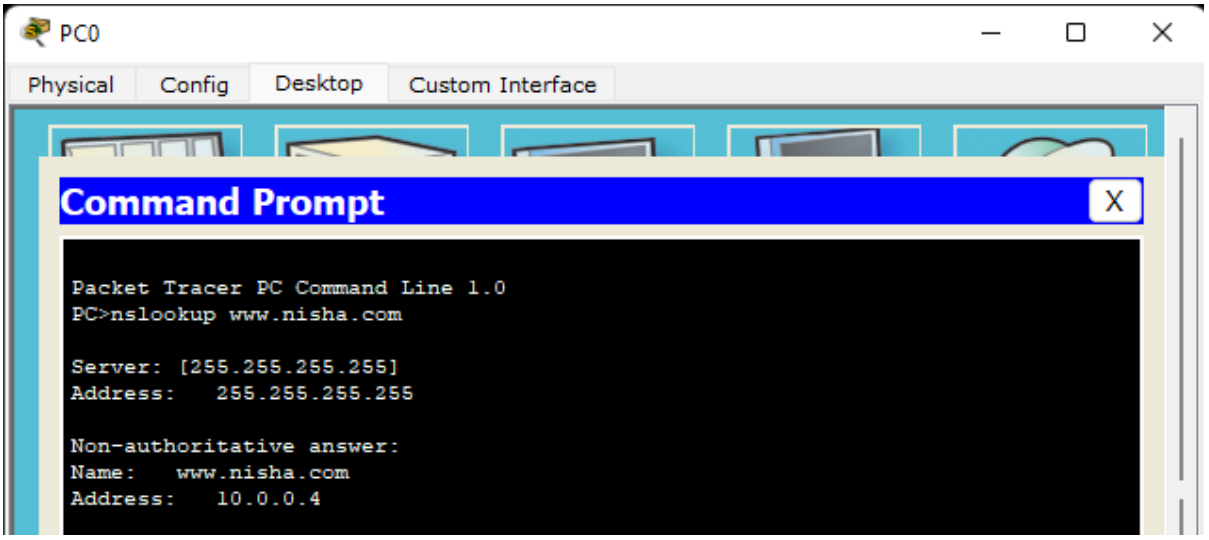
Topology



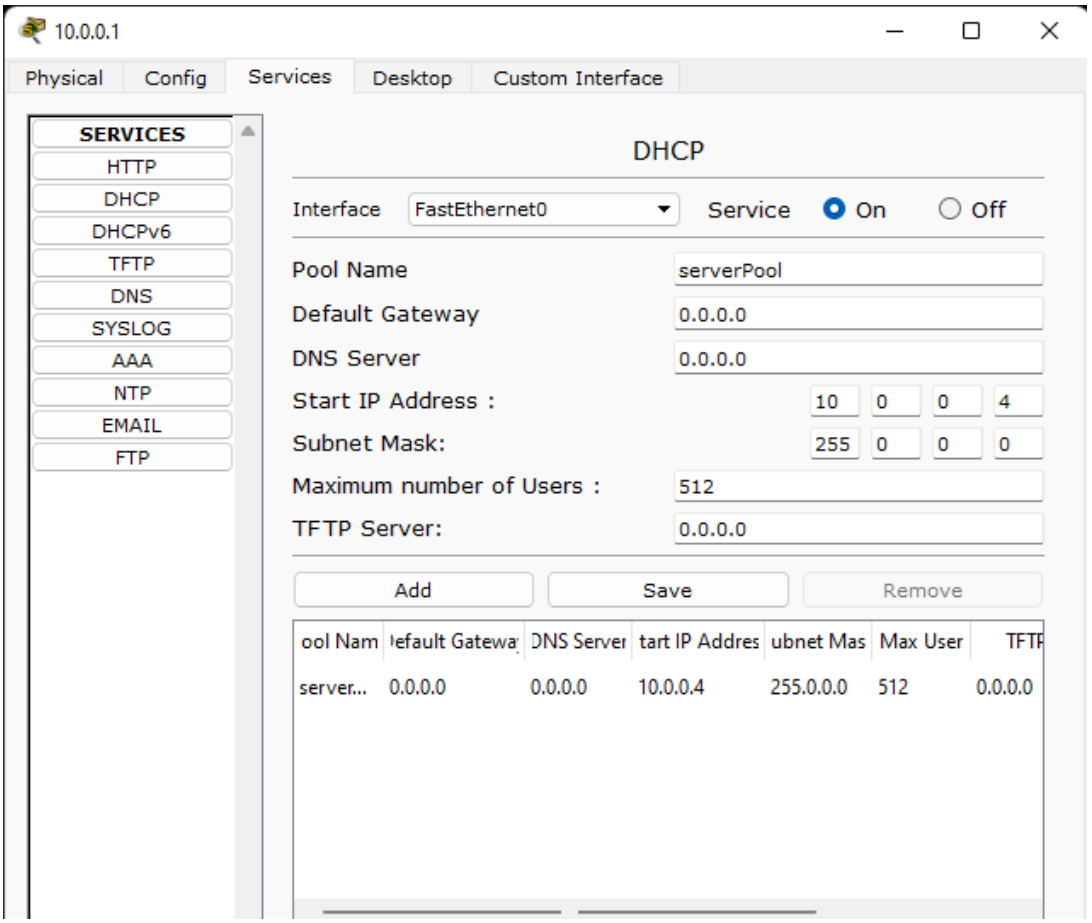
Adding web server



Accessing the web server from the end device



Nslookup for the web server created



Configuring DHCP

Cycle 2:
Program1:
Write a program for error detecting code using CRC-CCITT (16-bits).

Code:

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

		for(int i=0;i<16;i++) //adding zeros to the string
```

```
		{
			a[len++]=0;
		}

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

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

Output:

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

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

Program 2:

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

Code:

```
import java.util.*;

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

    public static void main(String args[]) {

        Scanner sc = new Scanner(System.in);
        System.out.println("Please enter the number of Vertices: ");
        v = sc.nextInt();

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

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

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

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

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

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

        dvr_calc_disp("The new Routing Tables are: ");
        sc.close();
    }
    static void dvr_calc_disp(String message) {
        System.out.println();
        init_tables();
        update_tables();
        System.out.println(message);
        print_tables();
        System.out.println();
    }

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

```
                }
            }
        }
    }

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

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

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

}
```

Output:

```
Please enter the number of Vertices:
4
Please enter the number of Edges:
5
Please enter data for Edge 1:
Source: 1
Destination: 2
Cost: 1
Please enter data for Edge 2:
Source: 1
Destination: 3
Cost: 3
Please enter data for Edge 3:
Source: 2
Destination: 3
Cost: 1
Please enter data for Edge 4:
Source: 2
Destination: 4
Cost: 1
Please enter data for Edge 5:
Source: 3
Destination: 4
Cost: 4

The initial Routing Tables are:
Dist: 0    Dist: 1    Dist: 2    Dist: 2
Dist: 1    Dist: 0    Dist: 1    Dist: 1
Dist: 2    Dist: 1    Dist: 0    Dist: 2
Dist: 2    Dist: 1    Dist: 2    Dist: 0

Please enter the Source Node for the edge whose cost has changed: 2
Please enter the Destination Node for the edge whose cost has changed: 4
Please enter the new cost: 10

The new Routing Tables are:
Dist: 0    Dist: 1    Dist: 2    Dist: 6
Dist: 1    Dist: 0    Dist: 1    Dist: 5
Dist: 2    Dist: 1    Dist: 0    Dist: 4
Dist: 6    Dist: 5    Dist: 4    Dist: 0
```

Program 3:

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

Code:

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

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

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

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

Output:

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

Program 4:

Write a program for congestion control using Leaky bucket algorithm.

Code:

```
import java.util.*;

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

Output:

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

Program 5:

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

Code:

Client:

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

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

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

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

      String fname = keyRead.readLine();

      OutputStream ostream = sock.getOutputStream();

      PrintWriter pwrite = new PrintWriter(ostream, true);

      pwrite.println(fname);

      InputStream istream = sock.getInputStream();

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

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

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

Server:

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

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

        Socket sock = sersock.accept();

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

        InputStream istream = sock.getInputStream();

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

        String fname = fileRead.readLine();

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

        OutputStream ostream = sock.getOutputStream();

        PrintWriter pwrite = new PrintWriter(ostream, true);

        String str;

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

            pwrite.println(str);

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

Output:

TCP Client:

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

TCP Server:

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

Program 6:

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

Code:

Client:

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

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

      DatagramSocket clientSocket = new DatagramSocket();

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

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

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

      sendData = sentence.getBytes();

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

      clientSocket.send(sendPacket);

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

      clientSocket.receive(receivePacket);

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

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

      clientSocket.close();
   }
```

```
}
```

Server:

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

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

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

         serverSocket.receive(receivePacket);

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

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

         InetAddress IPAddress = receivePacket.getAddress();

         int port = receivePacket.getPort();

         String capitalizedSentence = sentence.toUpperCase();

         sendData = capitalizedSentence.getBytes();

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

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

Output:

UDP Client:

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

UDP Server:

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