

Ex No:1 Make use of various networking commands like tcpdump, netstat, ifconfig, nslookup and traceroute. Capture ping and traceroute PDUs using a network protocol analyzer and examine.

1. Ping Command

The ping command is one of the most often used networking utilities for detecting devices on a network and for troubleshooting network problems.

When you ping a device you send that device a short message, which it then sends back (**the echo**).

The general format is **ping hostname** or **ping IPaddress**.

Example **ping www.google.com** or **ping 216.58.208.68**

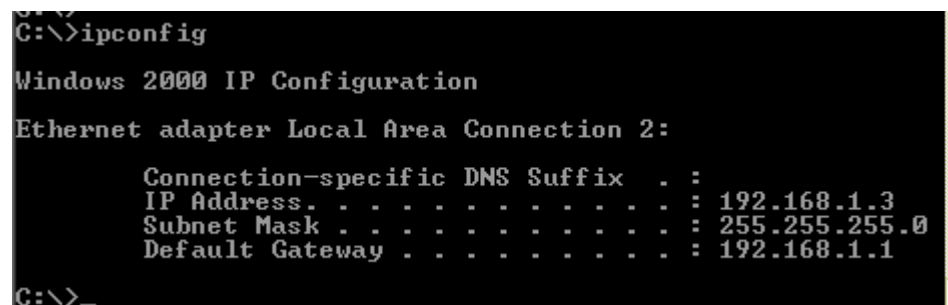
2. ipconfig Command

Another indispensable and frequently used utility that is used for finding network information about your local machine like IP addresses, DNS addresses etc

Basic Use: Finding Your IP Address and Default Gateway

Type the command ipconfig at the prompt.

The following is displayed



```
C:\>ipconfig

Windows 2000 IP Configuration

Ethernet adapter Local Area Connection 2:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . .               : 192.168.1.3
    Subnet Mask . . . . .             : 255.255.255.0
    Default Gateway . . . . .         : 192.168.1.1

C:\>
```

Ip config has a number of switches the most common are:

ipconfig /all – displays more information about the network setup on your systems including the MAC address.

ipconfig /release – release the current IP address

ipconfig /renew – renew IP address

ipconfig /? -shows help

3. NSlookup

- Used for checking DNS record entries.
- nslookup is a network administration command-line tool available for many computer operating systems.
- It is used for querying the Domain Name System (DNS) to obtain domain name or IP address mapping information
- The main use of **nslookup** is for troubleshooting DNS related problems.
- Nslookup can be use in **interactive** and **non-interactive** mode.
- To use in interactive mode **type nslookup** at the command line and hit return.
- You should get an **nslookup command prompt**.



```
C:\Users\steve>nslookup
Default Server:  bthub.home
Address:  192.168.1.254
> _
```

nslookup comand prompt

- To use in **non-interactive** mode type **nslookup options** at the command prompt.

```
C:\Users\steve>nslookup www.steves-internet-guide.com
Server:  bthub.home
Address: 192.168.1.254

Non-authoritative answer:
Name:    www.steves-internet-guide.com
Address: 82.165.119.51
```

Using Nslookup

- To illustrate the use of nslookup we are going to use it to:
 - Find the IP address of a host.
 - Find the domain name of an IP address.
 - Find mail servers for a domain.

Finding The IP Address of an Host-

To find the ip address of a host e.g. www.steves-internet-guide.com type:

nslookup www.steves-internet-guide.com

at a command prompt.

```
C:\Users\steve>nslookup www.steves-internet-guide.com
Server:  bthub.home
Address: 192.168.1.254

Non-authoritative answer:
Name:    www.steves-internet-guide.com
Address: 82.165.119.51
```

for an interactive lookup:

```

C:\Users\steve>nslookup
Default Server:  bthub.home
Address:  192.168.1.254

> www.steves-internet-guide.com
Server:  bthub.home
Address:  192.168.1.254

Non-authoritative answer:
Name:      www.steves-internet-guide.com
Address:  82.165.119.51

> _

```

Reverse Lookup IP address to domain name

Type **nslookup** IP address

```

C:\Users\steve>nslookup 82.165.119.51
Server:  bthub.home
Address:  192.168.1.254

Name:      kundenserver.de
Address:  82.165.119.51

```

Find Mail Servers for a Domain

Type **nslookup -querytype=mx domain name**

```

C:\Users\steve>nslookup -querytype=mx steves-internet-guide.com
Server:  bthub.home
Address:  192.168.1.254

Non-authoritative answer:
steves-internet-guide.com      MX preference = 10, mail exchanger = mx00.1and1.co.uk
steves-internet-guide.com      MX preference = 10, mail exchanger = mx01.1and1.co.uk

mx01.1and1.co.uk               internet address = 217.72.192.67
mx00.1and1.co.uk               internet address = 212.227.15.41

C:\Users\steve>

```

4.Netstat Command

Used for displaying information about tcp and udp connections and ports

Netstat Command Syntax

netstat [-a] [-b] [-e] [-f] [-n] [-o] [-p *protocol*] [-r] [-s] [-t] [-x] [-y] [*time_interval*] [/?]

Netstat Command List	
Option	Explanation
netstat	Execute the netstat command alone to show a relatively simple list of all active TCP connections which, for each one, will show the local IP address (your computer), the foreign IP address (the other computer or network device), along with their respective port numbers, as well as the TCP state.
-a	This switch displays active TCP connections, TCP connections with the listening state, as well as UDP ports that are being listened to.
-b	This netstat switch is very similar to the -o switch listed below, but instead of displaying the PID, will display the process's actual file name. Using -b over -o might seem like it's saving you a step or two but using it can sometimes greatly extend the time it takes netstat to fully execute.
-e	Use this switch with the netstat command to show statistics about your network connection. This data includes bytes, unicast packets, non-unicast packets, discards, errors, and unknown protocols received and sent since the connection was established.
-f	The -f switch will force the netstat command to display the Fully Qualified Domain Name (FQDN) for each foreign IP addresses when possible.
-n	Use the -n switch to prevent netstat from attempting to determine host names for foreign IP addresses. Depending on your current network connections, using this switch could considerably reduce the time it takes for netstat to fully execute.
-o	A handy option for many troubleshooting tasks, the -o switch displays the process identifier (PID) associated with each displayed connection. See the example below for more about using netstat -o .
-p	Use the -p switch to show connections or statistics only for a particular <i>protocol</i> . You can not define more than one <i>protocol</i> at once, nor can you execute netstat with -p without defining a <i>protocol</i> .
<i>protocol</i>	When specifying a <i>protocol</i> with the -p option, you can use tcp , udp , tcpv6 , or udpv6 . If you use -s with -p to view statistics by protocol, you can use icmp , ip , icmpv6 , or ipv6 in addition to the first four I mentioned.
-r	Execute netstat with -r to show the IP routing table. This is the same as using the route command to execute route print .
-s	The -s option can be used with the netstat command to show detailed statistics by protocol. You can limit the statistics shown to a particular protocol by using the -s option and specifying that <i>protocol</i> , but be sure to use -s before -p protocol when using the switches together.
-t	Use the -t switch to show the current TCP chimney offload state in place of the typically displayed TCP state.
-x	Use the -x option to show all NetworkDirect listeners, connections, and shared endpoints.
-y	The -y switch can be used to show the TCP connection template for all connection. You cannot use -y with any other netstat option.
<i>time_interval</i>	This is the time, in seconds, that you'd like the netstat command to re-execute automatically, stopping only when you use Ctrl-C to end the loop.
/?	Use the help switch to show details about the netstat command's several options.

5. tracert Command

Traceroute is a command which can show you the path a packet of information takes from your computer to one you specify. It will list all the routers it passes through until it reaches its destination, or fails to and is discarded. In addition to this, it will tell you how long each 'hop' from router to router takes

```
C:\Users\Dr. Anchitha>tracert vcet.ac.in
```

```
Tracing route to vcet.ac.in [166.62.27.146]
```

```
over a maximum of 30 hops:
```

```
  1  11 ms  7 ms  7 ms 192.168.1.1
  2 126 ms  43 ms  45 ms abts-tn-dynamic-001.48.164.122.airtelbroadband.in
[122.164.48.1]
  3  55 ms  77 ms  *    dsl-ncr-dynamic-229.114.16.125.airtelbroadband.in
[125.16.114.229]
  4  77 ms 131 ms  66 ms 182.79.237.16
  5  96 ms 104 ms 168 ms 26496.sgw.equinix.com [27.111.228.105]
  6  80 ms  98 ms 103 ms 148.72.204.1
  7  *      *      *    Request timed out.
  8  *      *      *    Request timed out.
  9  51 ms  51 ms  51 ms ip-166-62-27-146.ip.secureserver.net [166.62.27.146]
```

```
Trace complete.
```

EX.NO.2: Design a topology using PCs and Switch with configuration of IP address and Observe the flow of data from host to host by creating network traffic

Aim:

To construct simple LAN and understand the concept Switch with configuration of IP address and Observe the flow of data from host to host by creating network traffic.

Theory

The switch is a network device that is used to segment the networks into different subnetworks called subnets or LAN segments. It is responsible for filtering and forwarding the packets between LAN segments based on the MAC address.

Steps to Configure the Switch:

Step 1. Open the packet tracer desktop and take a switch (PT-Switch) from the devices.



Step 2: Configure the Host name of the switch0.

- Click on switch0 and go to Command Line Interface.
- Then change the hostname to “sh”

```
switch>
```

```
switch>en
```

```
switch#conf t
```

```
switch(config)#hostname sh
```

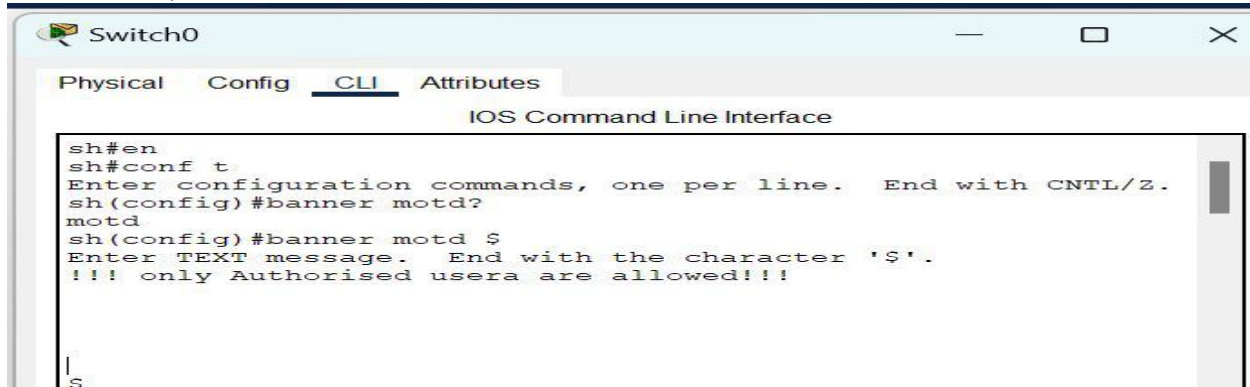
```
sh(config)#exit
```

Step 3: Set a message of the day (MOTD) banner for the users.

Command:

```
sh(config)#banner motd $
```

- Then, enter MOTD and end it with '\$' to exit.



```
Switch0
Physical Config CLI Attributes
IOS Command Line Interface
sh#en
sh#conf t
Enter configuration commands, one per line. End with CNTL/Z.
sh(config)#banner motd?
motd
sh(config)#banner motd $
Enter TEXT message. End with the character '$'.
!!! only Authorised users are allowed!!!
sh#
```

Step 4: Set up line control password and enable secret password.

To configure the Line Control password and Enable secret follow the below commands:

```
sh#conf t
```

```
sh(config)#
```

```
sh(config)#line con 0
```

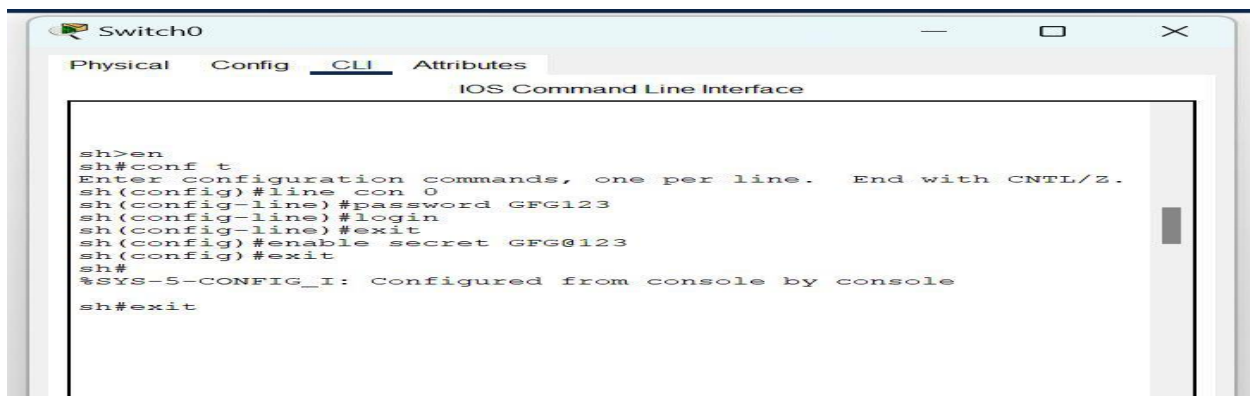
```
sh(config-line)#password GFG123
```

```
sh(config-line)#login
```

```
sh(config-line)#exit
```

```
sh(config)#enable secret GFG@123
```

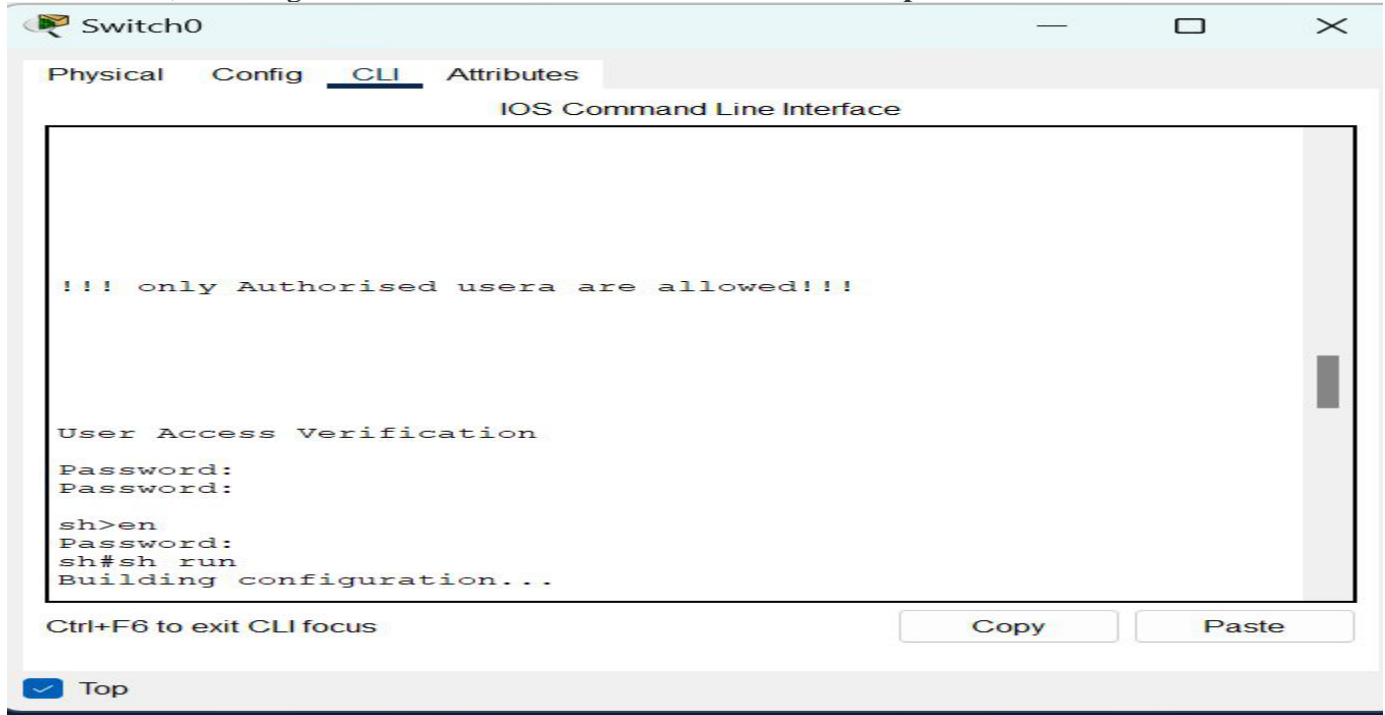
```
sh(config)#exit
```



```
Switch0
Physical Config CLI Attributes
IOS Command Line Interface
sh>en
sh#conf t
Enter configuration commands, one per line. End with CNTL/Z.
sh(config)#line con 0
sh(config-line)#password GFG123
sh(config-line)#login
sh(config-line)#exit
sh(config)#enable secret GFG@123
sh(config)#exit
sh#
%SYS-5-CONFIG_I: Configured from console by console
sh#exit
```

Step 5: Verify the password

- When you try to log in first, it will ask for the **line control password**.
- Then, to configure the terminal it will ask to **enable a secret password**.



To save the configuration use the below command:

Command:

sh#copy run startup-config

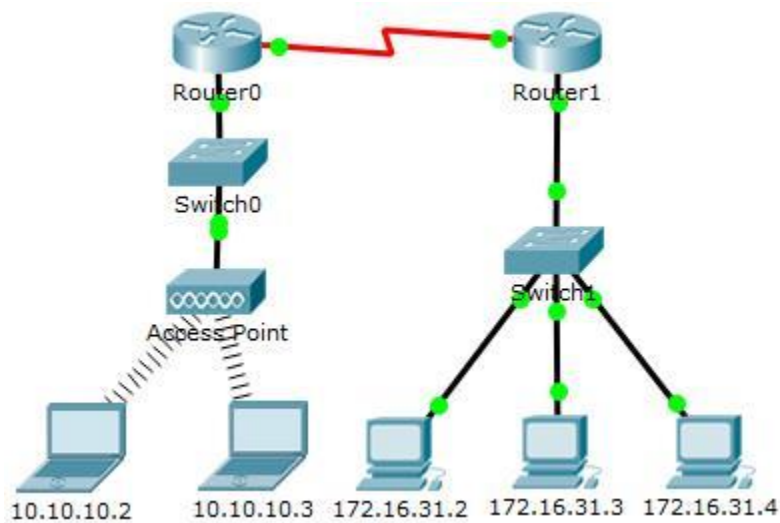
Result:

Thus the simple LAN is constructed and understood the concept Switch with configuration of IP address and Observe the flow of data from host to host by creating network traffic.

EX.NO.3: Create a Network scenario and examine dynamically learning configured Switch MAC address table and ARP Cache table using simulation tool

Aim:

To create a Network topology with suitable connecting devices and examine dynamically learning configured Switch MAC address table and ARP Cache table .



Addressing Table

Device	Interface	MAC Address	Switch Interface
Router0	Gg0/0	0001.6458.2501	G0/1
	S0/0/0	N/A	N/A
Router1	G0/0	00E0.F7B1.8901	G0/1
	S0/0/0	N/A	N/A
10.10.10.2	Wireless	0060.2F84.4AB6	F0/2
10.10.10.3	Wireless	0060.4706.572B	F0/2
172.16.31.2	F0	000C.85CC.1DA7	F0/1
172.16.31.3	F0	0060.7036.2849	F0/2

Device	Interface	MAC Address	Switch Interface
172.16.31.4	G0	0002.1640.8D75	F0/3

Step 1: Generate ARP requests by pinging 172.16.31.3 from 172.16.31.2.

- Click **172.16.31.2** and open the **Command Prompt**.
- Enter the **arp -d** command to clear the ARP table.
- Enter **Simulation** mode and enter the command **ping 172.16.31.3**. Two PDUs will be generated. The **ping** command cannot complete the ICMP packet without knowing the MAC address of the destination. So the computer sends an ARP broadcast frame to find the MAC address of the destination.
- Click **Capture/Forward** once. The ARP PDU moves **Switch1** while the ICMP PDU disappears, waiting for the ARP reply. Open the PDU and record the destination MAC address. Is this address listed in the table above? **No**
- Click **Capture/Forward** to move the PDU to the next device. How many copies of the PDU did **Switch1** make? **3**
- What is the IP address of the device that accepted the PDU? **172.16.31.3**
- Open the PDU and examine Layer 2. What happened to the source and destination MAC addresses? **Source became destination, FFFF.FFFF.FFFF turned into MAC address of 172.16.31.3**
- Click **Capture/Forward** until the PDU returns to 172.16.31.2. How many copies of the PDU did the switch make during the ARP reply? **1**

Step 2: Examine the ARP table.

- Note that the ICMP packet reappears. Open the PDU and examine the MAC addresses. Do the MAC addresses of the source and destination align with their IP addresses? **Yes**
- Switch back to **Realtime** and the ping completes.
- Click **172.16.31.2** and enter the **arp -a** command. To what IP address does the MAC address entry correspond? **172.16.31.3**
- In general, when does an end device issue an ARP request? **When it does not know the receiver's MAC address.**

Part 2: Examine a Switch MAC Address Table

Step 1: Generate additional traffic to populate the switch MAC address table.

- a. From **172.16.31.2**, enter the `ping 172.16.31.4` command.
- b. Click **10.10.10.2** and open the **Command Prompt**.
- c. Enter the `ping 10.10.10.3` command. How many replies were sent and received? **4 sent, 4 received.**

Step 2: Examine the MAC address table on the switches.

- a. Click **Switch1and** then the **CLI** tab. Enter the `show mac-address-table` command. Do the entries correspond to those in the table above? **Yes**
- b. Click **Switch0**, then the **CLI** tab. Enter the `show mac-address-table` command. Do the entries correspond to those in the table above? **Yes**
- c. Why are two MAC addresses associated with one port? **Because both devices connect to one port through the Access Point.**

Part 3: Examine the ARP Process in Remote Communications

Step 1: Generate traffic to produce ARP traffic.

- a. Click **172.16.31.2** and open the **Command Prompt**.
- b. Enter the `ping 10.10.10.1` command.
- c. Type `arp -a`. What is the IP address of the new ARP table entry? **172.16.31.1**
- d. Enter `arp -d` to clear the ARP table and switch to **Simulation** mode.
- e. Repeat the `ping to 10.10.10.1`. How many PDUs appear? **2**
- f. Click **Capture/Forward**. Click the PDU that is now at **Switch1**. What is the target destination IP destination address of the ARP request? **172.16.31.1**
- g. The destination IP address is not 10.10.10.1. Why? **The gateway address of the router interface is stored in the IPv4 configuration of the hosts. If the receiving host is not on the same network, the source uses the ARP process to determine a MAC address for the router interface serving as the gateway.**

Step 2: Examine the ARP table on Router1.

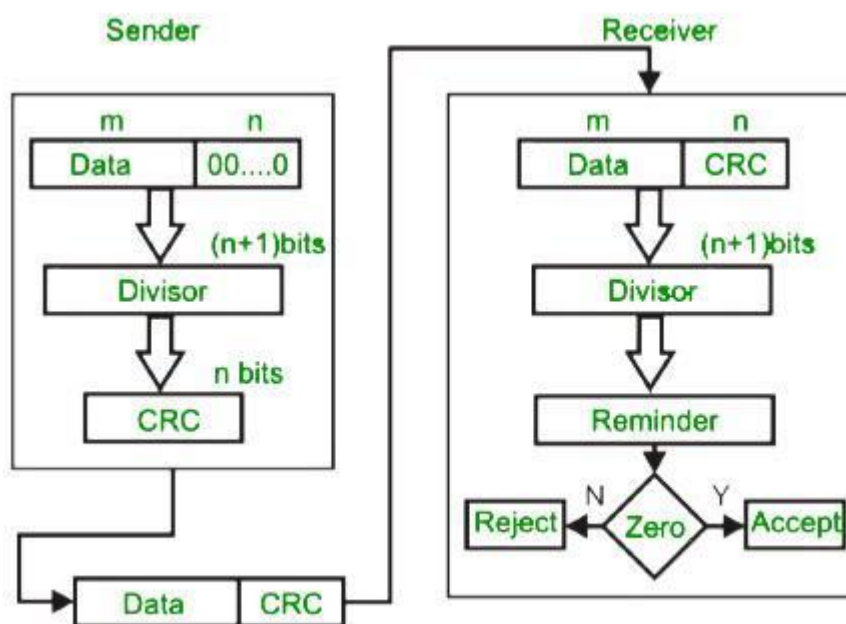
- a. Switch to **Realtime** mode. Click **Router1** and then the **CLI** tab.
- b. Enter privileged EXEC mode and then the `show mac-address-table` command. How many MAC addresses are in the table? Why? **Zero, This command means something completely different than the switch command show mac address-table.**
- c. Enter the `show arp` command. Is there an entry for **172.16.31.2**? **Yes**
- d. What happens to the first ping in a situation where the router responds to the ARP request? **It times out.**

EX.NO:04**Simulation of Error correction and detection techniques****AIM :**

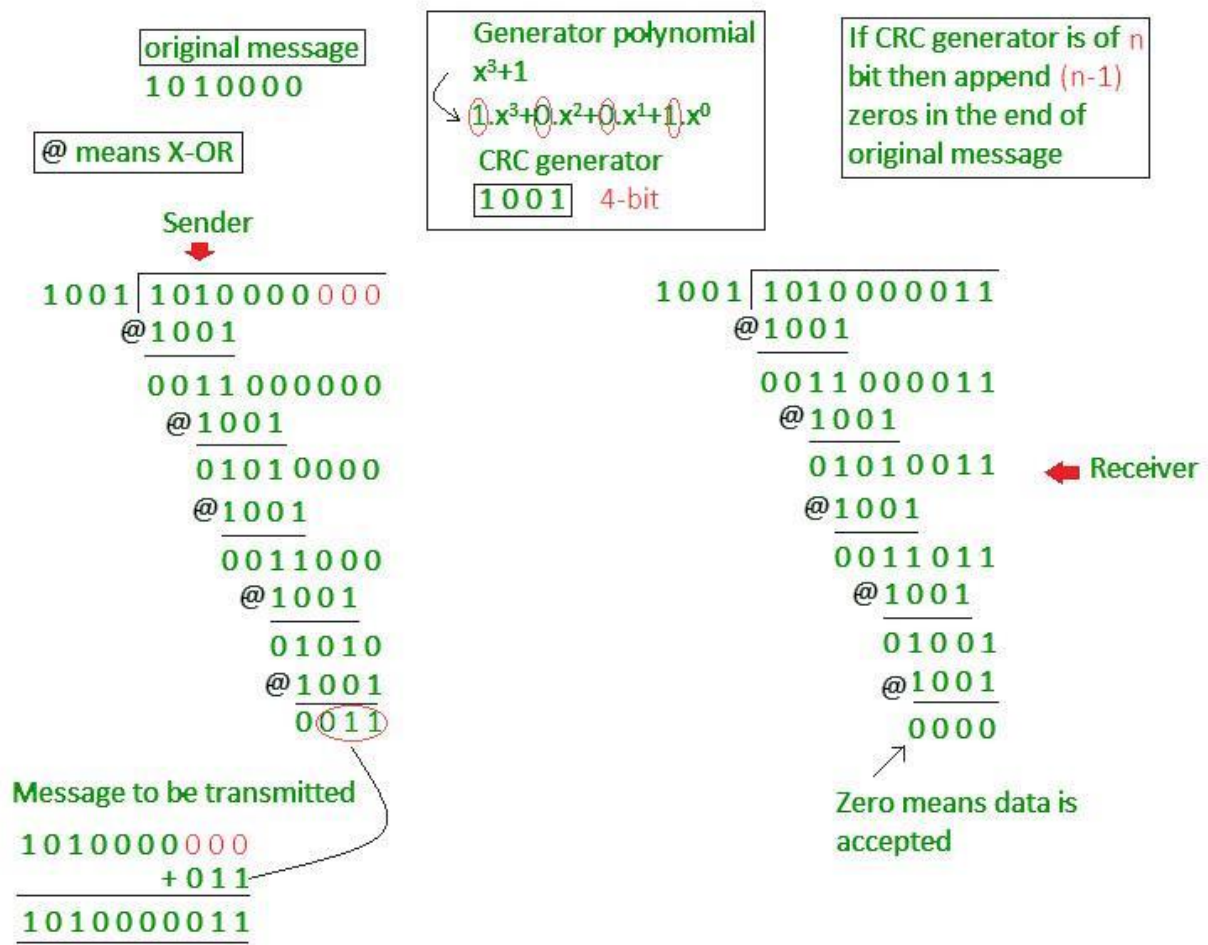
To Simulate the Error correction and detection techniques.

Theory:**Cyclic redundancy check (CRC)**

- Unlike checksum scheme, which is based on addition, CRC is based on binary division.
- In CRC, a sequence of redundant bits, called cyclic redundancy check bits, are appended to the end of data unit so that the resulting data unit becomes exactly divisible by a second, predetermined binary number.
- At the destination, the incoming data unit is divided by the same number. If at this step there is no remainder, the data unit is assumed to be correct and is therefore accepted.
- A remainder indicates that the data unit has been damaged in transit and therefore must be rejected.



Example :



:

Procedure:

1. Open the editor and type the program for error detection
2. Get the input in the form of bits.
3. Append the redundancy bits.
4. Divide the appended data using a divisor polynomial.
5. The resulting data should be transmitted to the receiver.
6. At the receiver the received data is entered.
7. The same process is repeated at the receiver.
8. If the remainder is zero there is no error otherwise there is some error in the received bits
9. Run the program.

Program:

```

import java.io.*;

class CRC
{
    public static void main(String args[]) throws IOException

```

```

{
    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
    System.out.println("Enter Generator:");
    String gen = br.readLine();
    System.out.println("Enter Data:");
    String data = br.readLine();
    String code = data;
    while(code.length() < (data.length() + gen.length() - 1))
        code = code + "0";
    code = data + div(code,gen);
    System.out.println("The transmitted Code Word is: " + code);
    System.out.println("Please enter the received Code Word: ");
    String rec = br.readLine();
    if(Integer.parseInt(div(rec,gen)) == 0)
        System.out.println("The received code word contains no errors.");
    else
        System.out.println("The received code word contains errors.");
}

static String div(String num1,String num2)
{
    int pointer = num2.length();
    String result = num1.substring(0, pointer);
    String remainder = "";
    for(int i = 0; i < num2.length(); i++)
    {
        if(result.charAt(i) == num2.charAt(i))
            remainder += "0";
        else
            remainder += "1";
    }
}

```

```

    }
    while(pointer < num1.length())
    {
        if(remainder.charAt(0) == '0')
        {
            remainder = remainder.substring(1, remainder.length());
            remainder = remainder + String.valueOf(num1.charAt(pointer));
            pointer++;
        }
        result = remainder;
        remainder = "";
        for(int i = 0; i < num2.length(); i++)
        {
            if(result.charAt(i) == num2.charAt(i))
                remainder += "0";
            else
                remainder += "1";
        }
    }
    return remainder.substring(1,remainder.length());
}
}

```

Result:

Thus the error detection and error correction is implemented successfully.

EX.NO.5: Create a Network Scenario and assign subnet IP Addresses to various Network Devices and Verify the Connectivity using simulation tool

Aim:

To Create a Network Topology and assign subnet IP Addresses to various Network Devices and Verify the Connectivity.

Theory

A subnet, or subnetwork, is a part of a larger network. Subnets are a logical part of an IP network into multiple, smaller network components. The Internet Protocol (IP) is the method for transmitting data from one computer to another over the internet network. Each computer, or host, on the internet, has at least one IP address as a unique identifier.

Steps to Configure and Verify Three Router Connections in Cisco Packet Tracer:

Step 1: First, open the Cisco packet tracer desktop and select the devices given below:

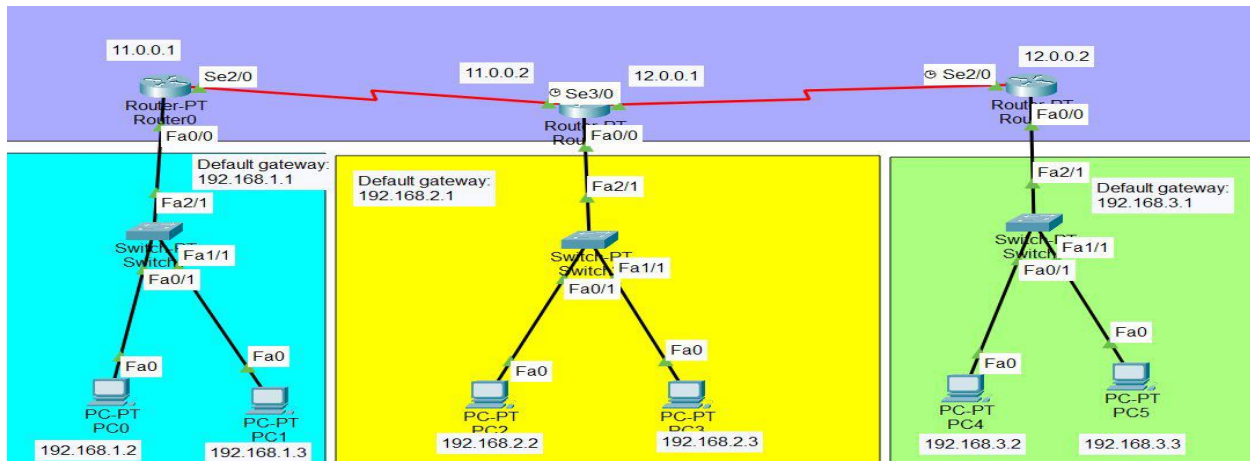
S.NO	Device	Model-Name	Qty.
1.	PC	pc	6
2.	Switch	PT-Switch	3
3.	Router	PT-Router	3

IP Addressing Table for PCs

S.NO	Device	IPv4 Address	Subnet Mask	Default-Gateway
1.	pc0	192.168.1.2	255.255.255.0	192.168.1.1
2.	pc1	192.168.1.3	255.255.255.0	192.168.1.1
3.	pc2	192.168.2.2	255.255.255.0	192.168.2.1
4.	pc3	192.168.2.3	255.255.255.0	192.168.2.1
5.	pc4	192.168.3.2	255.255.255.0	192.168.3.1

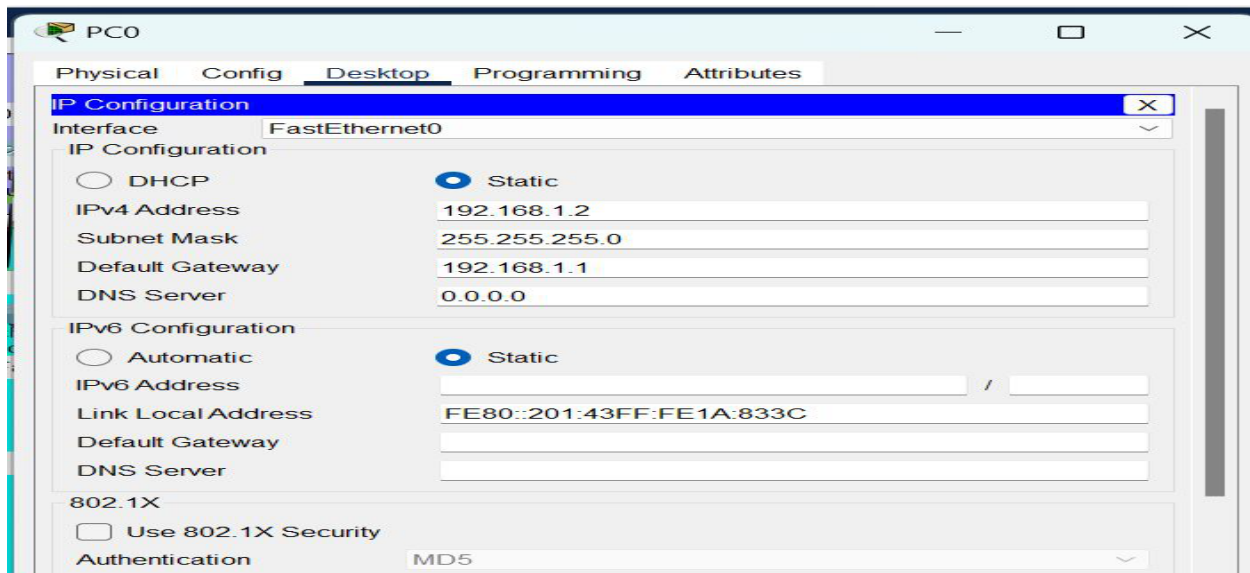
S.NO	Device	IPv4 Address	Subnet Mask	Default-Gateway
6.	pc5	192.168.3.3	255.255.255.0	192.168.3.1

- Then, create a network topology as shown below the image.
- Use an Automatic connecting cable to connect the devices with others.



Step 2: Configure the PCs (hosts) with IPv4 address and Subnet Mask according to the IP addressing table given above.

- To assign an IP address in PC0, click on PC0.
- Then, go to desktop and then IP configuration and there you will IPv4 configuration.
- Fill IPv4 address and subnet mask.



Assigning IP address using the ipconfig command.

- Or we can also assign an IP address with the help of a command.

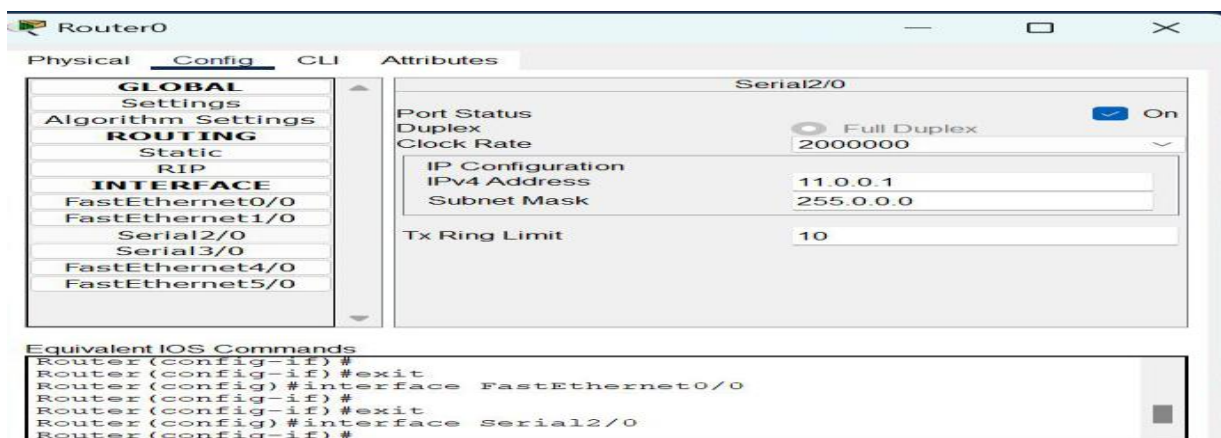
- Go to the command terminal of the PC.
- Then, type ipconfig <IPv4 address><subnet mask><default gateway>(if needed)
 - Example: ipconfig 192.168.1.2 255.255.255.0 192.168.1.1
- Repeat the same procedure with other PCs to configure them thoroughly.

Step 3: Configure router with IP address and subnet mask.

IP Addressing Table Router

S.NO	Device	Interface	IPv4 Address	Subnet mask
1.	router0	FastEthernet0/0	192.168.1.1	255.255.255.0
		Serial2/0	11.0.0.1	255.0.0.0
2.	router1	Serial 2/0	11.0.0.2	255.0.0.0
		Serial 3/0	12.0.0.1	255.0.0.0
3.	router 3	FastEthernet0/0	192.168.3.1	255.255.255.0
		Serial2/0	12.0.0.2	255.0.0.0

- To assign an IP address in router0, click on router0.
- Then, go to config and then Interfaces.
- Then, configure the IP address in FastEthernet and serial ports according to IP addressing Table.
- Fill IPv4 address and subnet mask.



- Repeat the same procedure with other routers to configure them thoroughly.
- Step 4:** After configuring all of the devices we need to assign the routes to the routers.

To assign static routes to the particular router:

- First, click on router0 then Go to CLI.
 - Then type the commands and IP information given below.
- CLI command : ip route <network id> <subnet mask><next hop>

Static Routes for Router0 are given below:

```
Router(config)#ip route 192.168.2.0 255.255.255.0 11.0.0.2
```

```
Router(config)#ip route 11.0.0.0 255.0.0.0 11.0.0.2
```

```
Router(config)#ip route 192.168.3.0 255.255.255.0 11.0.0.2
```

```
Router(config)#ip route 12.0.0.0 255.0.0.0 11.0.0.2
```

Static Routes for Router1 are given below:

```
Router(config)#ip route 192.168.1.0 255.255.255.0 11.0.0.1
```

```
Router(config)#ip route 11.0.0.0 255.0.0.0 11.0.0.1
```

```
Router(config)#ip route 192.168.3.0 255.255.255.0 12.0.0.2
```

```
Router(config)#ip route 12.0.0.0 255.0.0.0 12.0.0.2
```

Static Routes for Router2 are given below:

```
Router(config)#ip route 192.168.1.0 255.255.255.0 12.0.0.1
```

```
Router(config)#ip route 11.0.0.0 255.0.0.0 12.0.0.1
```

```
Router(config)#ip route 12.0.0.0 255.0.0.0 12.0.0.1
```

```
Router(config)#ip route 192.168.2.0 255.255.255.0 12.0.0.1
```

Step 5: Verifying the network by pinging the IP address of any PC. We will use the ping command to do so.

- First, click on PC0 then Go to the command prompt
- Then type ping <IP address of targeted node>
- As we can see in the below image we are getting replies which means the connection is working very fine

Example : ping 192.168.2.2

Result:

Thus the Network Topology is created and assigned the subnet IP Addresses to various Network Devices and Verify the Connectivity.

EX.NO.6: Create a Network scenario and examine dynamically learning configured Switch MAC address table and ARP Cache table using simulation tool.

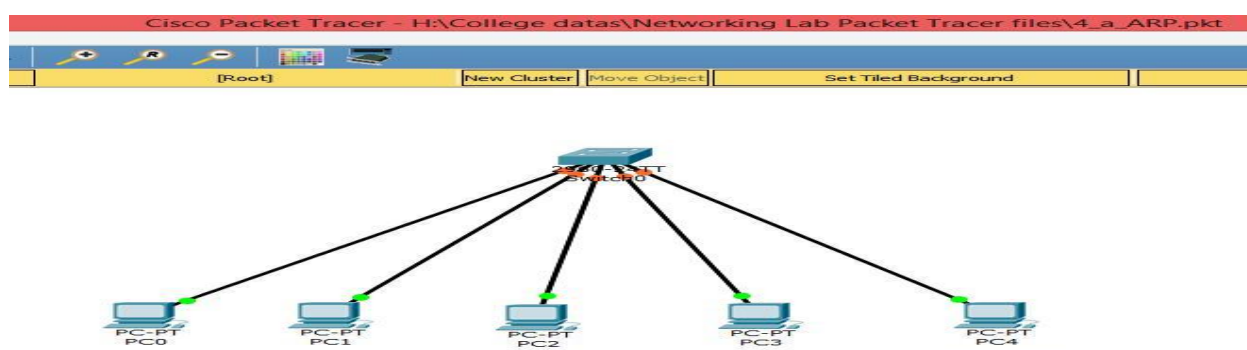
Aim:

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

Theory

ARP (Address Resolution Protocol) is a network protocol used to find out the hardware (MAC) address of a device from an IP address. It is used when a device wants to communicate with some other device on a local network (for example on an Ethernet network that requires physical addresses to be known before sending packets). The sending device uses ARP to translate IP addresses to MAC addresses. The device sends an ARP request message containing the IP address of the receiving device. All devices on a local network segment see the message, but only the device that has that IP address responds with the ARP reply message containing its MAC address. The sending device now has enough information to send the packet to the receiving device.

Network Topology Diagram for ARP



Input Details for ARP

PC0	PC1	PC2	PC3	PC4
IP Address : 10.0.0.1	IP Address : 10.0.0.2	IP Address : 10.0.0.3	IP Address : 10.0.0.4	IP Address : 10.0.0.5
Subnet	Subnet	Subnet	Subnet	Subnet
Mask : 255.255.25	Mask : 255.255.25	Mask : 255.255.25	Mask: 255.255.25	Mask : 255.255.25

OUTPUT:

ARP CATCH TABLE OF PC1 (IP: 10.0.0.2):

C:\>arp -a

Internet Address	Physical Address	Type
10.0.0.1	0001.42c1.0547	dynamic
10.0.0.3	0001.6402.dab3	dynamic
10.0.0.4	0001.43e2.332b	dynamic
10.0.0.5	0001.9665.3174	dynamic

SWITCH MAC ADDRESS TABLE:

```
Switch>
Switch>SHOW MAC ADDRESS-TABLE
      Mac Address Table
-----
Vlan    Mac Address      Type        Ports
----    -
1       0001.42c1.0547    DYNAMIC     Fa0/1
1       0001.43e2.332b    DYNAMIC     Fa0/4
1       0001.6402.dab3    DYNAMIC     Fa0/3
1       0001.9665.3174    DYNAMIC     Fa0/5
1       0060.70c9.ba88    DYNAMIC     Fa0/2
```

Result:

Thus, constructed a simple LAN and understand the concept and operation of ARP and got the ARP Cache of given layout.

EX.NO.7: Create a Network scenario with multiple routers and configure using RIP Routing in simulation tool

Aim:

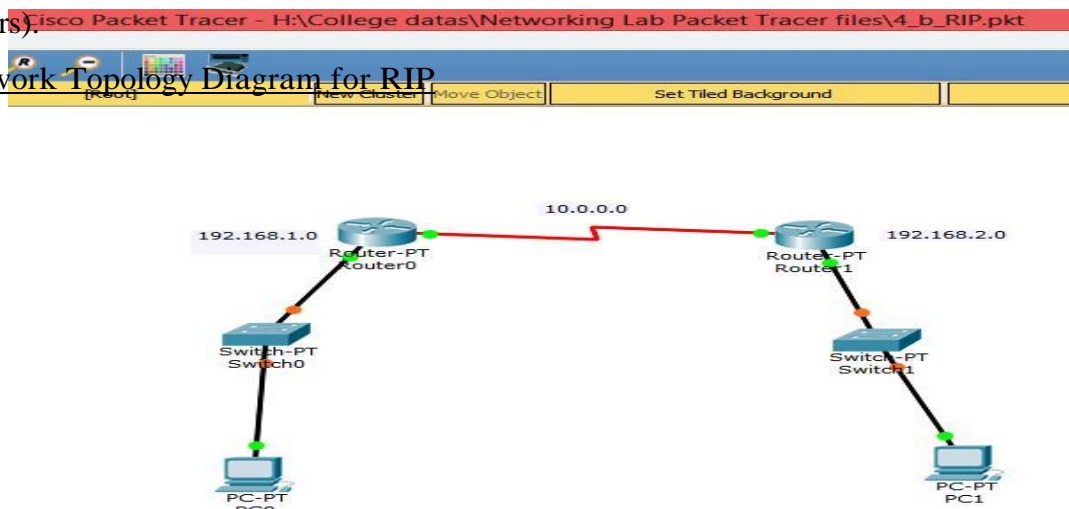
To understand the concept and operation of Routing Information Protocol (RIP)

Theory

RIP (Routing Information Protocol) is one of the oldest distance vector routing protocols. It is usually used on small networks because it is very simple to configure and maintain, but lacks some advanced features of routing protocols like OSPF or EIGRP. Two versions of the protocol exists: version 1 and version 2. Both versions use hop count as a metric and have the administrative distance of 120. RIP version 2 is capable of advertising subnet masks and uses multicast to send routing updates, while version 1 doesn't advertise subnet masks and uses broadcast for updates. Version 2 is backwards compatible with version 1.

RIPv2 sends the entire routing table every 30 seconds, which can consume a lot of bandwidth. RIPv2 uses multicast address of 224.0.0.9 to send routing updates, supports authentication and triggered updates (updates that are sent when a change in the network occurs).

Network Topology Diagram for RIP



Input Details for RIP

PC0	PC1	Router 0	Router 1
IP Address : 192.168.1.2 Gate way : 192.168.1.1	IP Address: 192.168.2.2 Gate way : 192.168.2.1	<u>Fast Ethernet 0/0</u> IP Address: 192.168.1.1 <u>Serial 2/0</u> : 10.0.0.1 at 6400 clock rate	<u>Fast Ethernet 0/0</u> IP Address : 192.168.2.1 <u>Serial 2/0</u> : 10.0.0.2 no clock rate

OUTPUT:

RIP (PINGING FROM PC0 TO PC1):

C:\>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:

Reply from 192.168.2.2: bytes=32 time=11ms TTL=126

Reply from 192.168.2.2: bytes=32 time=12ms TTL=126

Reply from 192.168.2.2: bytes=32 time=13ms TTL=126

Reply from 192.168.2.2: bytes=32 time=11ms TTL=126

Ping statistics for 192.168.2.2:

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

Approximate round trip times in milli-seconds:

Minimum = 11ms, Maximum = 13ms, Average = 11ms

Result:

Thus, understand the concept and operation of RIP and pinged from PC in are networks to PC to another network.

EX:NO.7:Create a Network scenario with multiple routers and configure using OSPF Routing in simulation tool

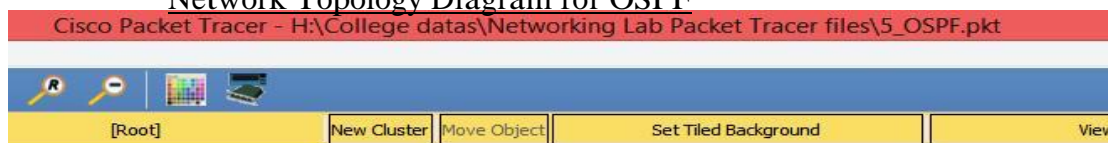
Aim:

To construct multiple router networks and understand the operation of Open shortest Path First (OSPF) Protocol.

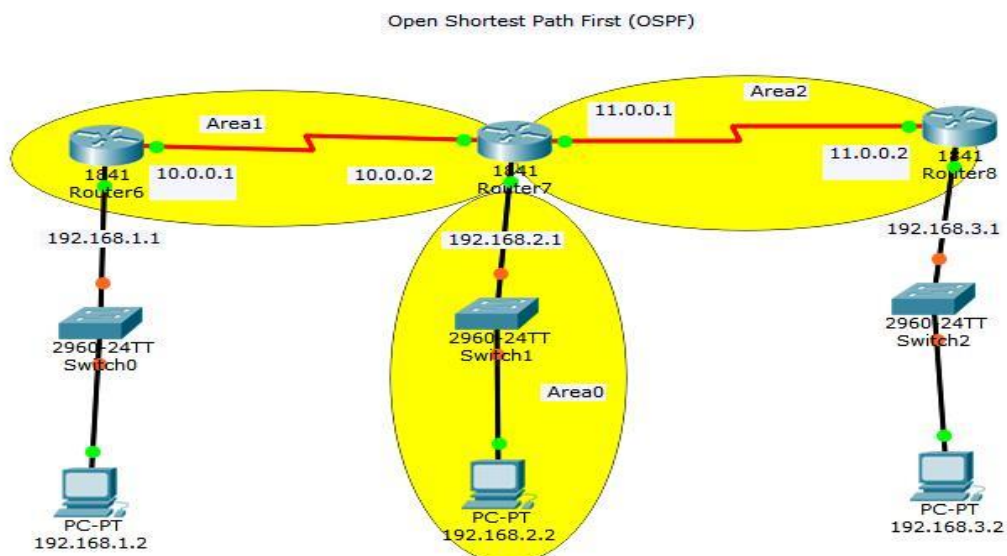
Theory

The OSPF routing protocol has largely replaced the older Routing Information Protocol (RIP) in corporate networks. Using OSPF, a router that learns of a change to a routing table (when it is reconfigured by network staff, for example) or detects a change in the network immediately multicasts the information to all other OSPF hosts in the network so they will all have the same routing table information. Unlike RIP, which requires routers to send the entire routing table to neighbors every 30 seconds, OSPF sends only the part that has changed and only when a change has taken place. When routes change -- sometimes due to equipment failure -- the time it takes OSPF routers to find a new path between endpoints with no loops (which is called "open") and that minimizes the length of the path is called the convergence time.

Network Topology Diagram for OSPF



Input Details for OSPF



Input Details for OSPF

PC0	PC1	PC2
IP Address : 192.168.1.2 Gate way	IP Address : 192.168.2.2 Gate way	IP Address : 192.168.3.2 Gate way
fa 0/0 : IP 192.168.1.1 Address : 192.168.1.1	fa 0/0 : IP 192.168.2.1 Address : 192.168.2.1	Fa 0/0 : IP 192.168.3.1 Address : 192.168.3.1
Serial 0/0/0 : 10.0.0.1 @ 2000000 clock rate Serial 0/0/1 : -	Serial 0/0/0 : 10.0.0.2 Serial 0/0/1 : - @ 2000000 clock rate	Serial 0/0/0 : 10.0.0.2 @ Se 0/0/1 : 11.0.0.1

ROUTER0 CLI:

Router#en

Router#config

Configuring from terminal, memory, or network [terminal]?

Enter configuration commands, one per line. End with

CNTL/Z. Router(config)#router ospf 1

Router(config-router)#network 192.168.1.0 0.0.0.255 area

1 Router(config-router)#network 10.0.0.0 0.255.255.255

area 1 Router(config-router)#exit

00:19:21: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.2.1 on Serial0/0/0 from LOADING to FULL, Loading Done

ROUTER1 CLI:

Router(config)#router

ospf 2

Router(config-router)#network 192.168.2.0 0.0.0.255 area 0

Router(config-router)#network 10.0.0.0 0.255.255.255 area 1

00:19:07: %OSPF-5-ADJCHG: Process 2, Nbr 192.168.1.1 on Serial0/0/0 from LOADING to FULL, Loading Done

Router(config-router)#network 11.0.0.0 0.255.255.255

area 2 Router(config-router)#exit

00:25:52: %OSPF-5-ADJCHG: Process 2, Nbr 192.168.3.1 on Serial0/0/1 from LOADING to FULL, Loading Done

ROUTER2 CLI:

Router>en

Router#config

Configuring from terminal, memory, or network [terminal]? Enter configuration commands, one per line. End with CNTL/Z.

Router(config)#router ospf 1

Router(config-router)#network 192.168.3.0 0.0.0.255 area 2

Router(config-router)#network 11.0.0.0 0.255.255.255 area 2

00:25:19: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.2.1 on Serial0/0/0 from LOADING to FULL, Loading Done

Router(config)#exit

OUTPUT:

ROUTER0:

Router>en

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

C 10.0.0.0/8 is directly connected, Serial0/0/0

O IA 11.0.0.0/8 [110/128] via 10.0.0.2, 00:04:43,

Serial0/0/0 C 192.168.1.0/24 is directly connected, FastEthernet0/0

O IA 192.168.2.0/24 [110/65] via 10.0.0.2, 00:07:42,

Serial0/0/0 O IA 192.168.3.0/24 [110/129] via 10.0.0.2, 00:00:53, Serial0/0/0

ROUTER1:

C 10.0.0.0/8 is directly connected,
Serial0/0/0 C 11.0.0.0/8 is directly
connected, Serial0/0/1
O 192.168.1.0/24 [110/65] via 10.0.0.1, 00:04:50,
Serial0/0/0 C 192.168.2.0/24 is directly connected,
FastEthernet0/0
O 192.168.3.0/24 [110/65] via 11.0.0.2, 00:04:45, Serial0/0/1

ROUTER2:

O IA 10.0.0.0/8 [110/128] via 11.0.0.1, 00:06:55,
Serial0/0/0 C 11.0.0.0/8 is directly connected, Serial0/0/0
O IA 192.168.1.0/24 [110/129] via 11.0.0.1, 00:06:45,
Serial0/0/0 O IA 192.168.2.0/24 [110/65] via 11.0.0.1,
00:06:55, Serial0/0/0 C 192.168.3.0/24 is directly connected,
FastEthernet0/0

Result:

Thus, understand the concept and operation of OSPF and obtained the routing table and observe transfer data packets in real and simulation time.

EX.NO.8: Create a Network scenario and generate the network traffic to examine the TCP/UDP communication using Simulation tool.

Aim:

To Generate Network Traffic and Examine the Functionality of the TCP and UDP Protocols.

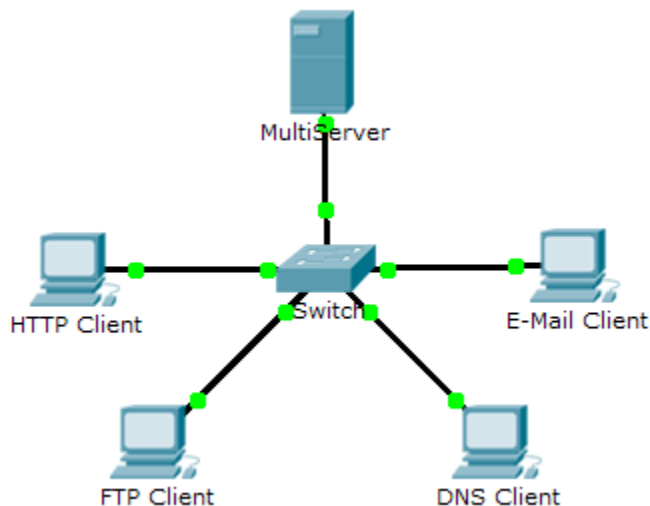
Theory

This simulation activity is intended to provide a foundation for understanding the TCP and UDP in detail. Simulation mode provides the ability to view the functionality of the different protocols.

As data moves through the network, it is broken down into smaller pieces and identified in some fashion so that the pieces can be put back together. Each of these pieces is assigned a specific name (protocol data unit [PDU]) and associated with a specific layer. Packet Tracer Simulation mode enables the user to view each of the protocols and the associated PDU. The steps outlined below lead the user through the process of requesting services using various applications available on a client PC.

This activity provides an opportunity to explore the functionality of the TCP and UDP protocols, multiplexing and the function of port numbers in determining which local application requested the data or is sending the data.

Network Topology Diagram for TCP/UDP protocols



Part 1: Generate Network Traffic in Simulation Mode

Part 2: Examine the Functionality of the TCP and UDP Protocols
Background

Part 1: Generate Network Traffic in Simulation Mode

Step 1: Generate traffic to populate Address Resolution Protocol (ARP) tables.

Perform the following tasks task to reduce the amount of network traffic viewed in the simulation.

- a. Click **MultiServer** and click the **Desktop** tab > **Command Prompt**.
- b. Enter the **ping 192.168.1.255** command. This will take a few seconds as every device on the network responds to **MultiServer**.
- c. Close the **MultiServer** window.

Step 2: Generate web (HTTP) traffic.

- a. Switch to Simulation mode.
- b. Click **HTTP Client** and click the **Desktop** tab > **Web Browser**.
- c. In the URL field, enter **192.168.1.254** and click **Go**. Envelopes (PDUs) will appear in the simulation window.
- d. Minimize, but do not close, the **HTTP Client** configuration window.

Step 3: Generate FTP traffic.

- a. Click **FTP Client** and click the **Desktop** tab > **Command Prompt**.
- b. Enter the **ftp 192.168.1.254** command. PDUs will appear in the simulation window.
- c. Minimize, but do not close, the **FTP Client** configuration window.

Step 4: Generate DNS traffic.

- a. Click **DNS Client** and click the **Desktop** tab > **Command Prompt**.
- b. Enter the **nslookup multiserver.pt.ptu** command. A PDU will appear in the simulation window.
- c. Minimize, but do not close, the **DNS Client** configuration window.

Step 5: Generate Email traffic.

- a. Click **E-Mail Client** and click the **Desktop** tab > **E Mail** tool.
- b. Click **Compose** and enter the following information:
 1. **To:** user@multiserver.pt.ptu
 2. **Subject:** Personalize the subject line
 3. **E-Mail Body:** Personalize the Email
- c. Click **Send**.
- d. Minimize, but do not close, the **E-Mail Client** configuration window.

Step 6: Verify that the traffic is generated and ready for simulation.

Every client computer should have PDUs listed in the Simulation Panel.

Part 2: Examine Functionality of the TCP and UDP Protocols

Step 1: Examine multiplexing as all of the traffic crosses the network.

You will now use the **Capture/Forward** button and the **Back** button in the Simulation Panel.

- a. Click **Capture/Forward** once. All of the PDUs are transferred to the switch.
- b. Click **Capture/Forward** again. Some of the PDUs disappear. What do you think happened to them?
 - **They are stored in the switch.**

- c. Click **Capture/Forward** six times. All clients should have received a reply. Note that only one PDU can cross a wire in each direction at any given time. What is this called?
 - **Multiplexing.**
- d. A variety of PDUs appears in the event list in the upper right pane of the simulation window. Why are they so many different colors?
 - **They represent different protocols.**
- e. Click **Back** eight times. This should reset the simulation.

Note: Do not click **Reset Simulation** any time during this activity; if you do, you will need to repeat the steps in Part 1.

Step 2: Examine HTTP traffic as the clients communicate with the server.

- a. Filter the traffic that is currently displayed to display only **HTTP** and **TCP** PDUs filter the traffic that is currently displayed:
 1. Click **Edit Filters** and toggle the **Show All/None** check box.
 2. Select **HTTP** and **TCP**. Click anywhere outside of the Edit Filters box to hide it. The Visible Events should now display only **HTTP** and **TCP** PDUs.
- b. Click **Capture/Forward**. Hold your mouse above each PDU until you find one that originates from **HTTP Client**. Click the PDU envelope to open it.
- c. Click the **Inbound PDU Details** tab and scroll down to the last section. What is the section labeled?
 - **TCP**
 - Are these communications considered to be reliable?
 - **Yes.**
- d. Record the **SRC PORT**, **DEST PORT**, **SEQUENCE NUM**, and **ACK NUM** values. What is written in the field to the left of the **WINDOW** field?
 - **1025 (could be different), 80, 0, 0 SYN**
- e. Close the PDU and click **Capture/Forward** until a PDU returns to the **HTTP Client** with a checkmark.
- f. Click the PDU envelope and select **Inbound PDU Details**. How are the port and sequence numbers different than before?
 - **80, 1025, 0, 1. SYN+ACK. The source and destination ports are reversed, and the acknowledgement number is 1. The SYN has changed to SYN+ACK.**
- g. There is a second PDU of a different color, which **HTTP Client** has prepared to send to **MultiServer**. This is the beginning of the HTTP communication. Click this second PDU envelope and select **Outbound PDU Details**.
- h. What information is now listed in the TCP section? How are the port and sequence numbers different from the previous two PDUs?
 - **1025, 80, 1, 1. PSH+ACK The source and destination ports are reversed, and both sequence and acknowledgement numbers are 1.**
- i. Click **Back** until the simulation is reset.

Step 3: Examine FTP traffic as the clients communicate with the server.

- a. In the Simulation Panel, change **Edit Filters** to display only **FTP** and **TCP**.
- b. Click **Capture/Forward**. Hold your cursor above each PDU until you find one that originates from **FTP Client**. Click that PDU envelope to open it.
- c. Click the **Inbound PDU Details** tab and scroll down to the last section. What is the section labeled?

- **TCP**
- Are these communications considered to be reliable?
- **Yes.**
- d. Record the **SRC PORT**, **DEST PORT**, **SEQUENCE NUM**, and **ACK NUM** values. What is written in the field to the left of the **WINDOW** field?
 - **1025, 21, 0, 0. SYN**
- e. Close the PDU and click **Capture/Forward** until a PDU returns to the **FTP Client** with a checkmark.
- f. Click the PDU envelope and select **Inbound PDU Details**. How are the port and sequence numbers different than before?
 - **21, 1025, 0, 1. SYN+ACK. The source and destination ports are reversed, and the acknowledgement number is 1.**
- g. Click the **Outbound PDU Details** tab. How are the port and sequence numbers different from the previous two results?
 - **1025, 21, 1, 1. ACK. The source and destination ports are reversed, and both sequence and acknowledgement numbers are 1.**
- h. Close the PDU and click **Capture/Forward** until a second PDU returns to the **FTP Client**. The PDU is a different color.
- i. Open the PDU and select **Inbound PDU Details**. Scroll down past the TCP section. What is the message from the server?
 - **May say either "Username ok, need password" or "Welcome to PT Ftp server"**
- j. Click **Back** until the simulation is reset.

Step 4: Examine DNS traffic as the clients communicate with the server.

- a. In the Simulation Panel, change **Edit Filters** to display only **DNS** and **UDP**.
- b. Click the PDU envelope to open it.
- c. Click the **Inbound PDU Details** tab and scroll down to the last section. What is the section labeled?
 - **UDP**
 - Are these communications considered to be reliable?
 - **No**
- d. Record the **SRC PORT** and **DEST PORT** values. Why is there no sequence and acknowledgement number?
 - **1025, 53. Because UDP does not need to establish a reliable connection.**
- e. Close the **PDU** and click **Capture/Forward** until a PDU returns to the **DNS Client** with a checkmark.
- f. Click the PDU envelope and select **Inbound PDU Details**. How are the port and sequence numbers different than before?
 - **53, 1025. The source and destination ports are reversed.**
- g. What is the last section of the **PDU** called?
 - **DNS ANSWER.**
- h. Click **Back** until the simulation is reset.

Step 5: Examine email traffic as the clients communicate with the server.

- a. In the Simulation Panel, change **Edit Filters** to display only **POP3**, **SMTP** and **TCP**.
- b. Click **Capture/Forward**. Hold your cursor above each PDU until you find one that originates from **E-mail Client**. Click that PDU envelope to open it.

- c. Click the **Inbound PDU Details** tab and scroll down to the last section. What transport layer protocol does email traffic use?
 - **TCP**
 - Are these communications considered to be reliable?
 - **Yes.**
- d. Record the **SRC PORT**, **DEST PORT**, **SEQUENCE NUM**, and **ACK NUM** values. What is written in the field to the left of the **WINDOW** field?
 - **1025, 25, 0, 0. SYN**
- e. Close the **PDU** and click **Capture/Forward** until a **PDU** returns to the **E-Mail Client** with a checkmark.
- f. Click the **PDU** envelope and select **Inbound PDU Details**. How are the port and sequence numbers different than before?
 - **25, 1025, 0, 1. SYN+ACK. The source and destination ports are reversed, and the acknowledgement number is 1.**
- g. Click the **Outbound PDU Details** tab. How are the port and sequence numbers different from the previous two results?
 - **1025, 25, 1, 1. ACK. The source and destination ports are reversed, and both sequence and acknowledgement numbers are 1. ACK**
- h. There is a second **PDU** of a different color that **HTTP Client** has prepared to send to **MultiServer**. This is the beginning of the email communication. Click this second **PDU** envelope and select **Outbound PDU Details**.
- i. How are the port and sequence numbers different from the previous two **PDU**s?
 - **1025, 25, 1, 1. PSH+ACK. The source and destination ports are reversed, and both sequence and acknowledgement numbers are 1.**
- j. What email protocol is associated with TCP port 25? What protocol is associated with TCP port 110?
 - **SMTP. POP3.**
- k. Click **Back** until the simulation is reset.

Step 6: Examine the use of port numbers from the server.

- a. To see TCP active sessions, perform the following steps in quick succession:
 1. Switch back to **Realtime** mode.
 2. Click **MultiServer** and click the **Desktop** tab > **Command Prompt**.
- b. Enter the **netstat** command. What protocols are listed in the left column? TCP
 - What port numbers are being used by the server? **Answers will vary, but students may see all three: 21, 25, 80. They should certainly see 21**
- c. What states are the sessions in?
 - **Answer will vary. Possible states include CLOSED, ESTABLISHED, LAST_ACK**
- d. Repeat the **netstat** command several times until you see only one session still ESTABLISHED. For which service is this connection still open? **FTP**
 - Why doesn't this session close like the other three? (Hint: Check the minimized clients)
 - **The server is waiting for a password from the client.**

EXP: 9a
FILE TRANSFER IN CLIENT & SERVER

AIM

To Perform File Transfer in Client & Server Using TCP/IP.

ALGORITHM

CLIENT SIDE

1. Start.
2. Establish a connection between the Client and Server.
3. Socketss=new Socket(InetAddress.getLocalHost(),1100);
4. Implement a client that can send two requests.
 - i) To get a file from the server.
 - ii) To put or send a file to the server.
5. After getting approval from the server ,the client either get file from the server or send file to the server.

SERVER SIDE

1. Start.
2. Implement a server socket that listens to a particular port number.
3. Server reads the filename and sends the data stored in the file for the 'get' request.
4. It reads the data from the input stream and writes it to a file in the server for the 'put' instruction.
5. Exit upon client's request.
6. Stop.

PROGRAM

CLIENT SIDE

```
import java.net.*;
import java.io.*;
public class FileClient
{
public static void main (String [] args ) throws IOException
{
    int filesize=6022386; // filesize temporary hardcoded
    long start = System.currentTimeMillis();
    int bytesRead;
    int current = 0;
    // localhost for testing
    Socket sock = new Socket("127.0.0.1",13267);
```

```

System.out.println("Connecting...");
// receive file
byte [] mybytearray = new byte [filesize];
InputStream is = sock.getInputStream();
FileOutputStream fos = new FileOutputStream("source-copy.pdf");
BufferedOutputStream bos = new BufferedOutputStream(fos);
bytesRead = is.read(mybytearray,0,mybytearray.length);
current = bytesRead;
// thanks to A. Cádiz for the bug fix
do
{
bytesRead =
is.read(mybytearray, current, (mybytearray.length-current));
if(bytesRead >= 0)
current += bytesRead;
} while(bytesRead > -1);
bos.write(mybytearray, 0 , current);
bos.flush();
long end = System.currentTimeMillis();
System.out.println(end-start);
bos.close();
sock.close();
}}

```

SERVER SIDE

```

import java.net.*; import java.io.*;
public class FileServer {
public static void main (String [] args ) throws IOException
{
ServerSocket servsock = new ServerSocket(13267);
while (true)
{
System.out.println("Waiting...");
Socket sock = servsock.accept();
System.out.println("Accepted connection : " + sock);
File myFile = new File ("source.pdf");
byte [] mybytearray = new byte [(int)myFile.length()];
FileInputStream fis = new FileInputStream(myFile);
BufferedInputStream bis = new BufferedInputStream(fis);
bis.read(mybytearray,0,mybytearray.length);
OutputStream os = sock.getOutputStream();
System.out.println("Sending...");

```

```
os.write(mybytearray,0,mybytearray.length);  
os.flush();  
sock.close(); } } }
```

OUTPUT

SERVER OUTPUT

```
C:\Program Files\Java\jdk1.6.0\bin>javac FServer.java  
C:\Program Files\Java\jdk1.6.0\bin>java FServer
```

Waiting for clients...

Connection Established Client wants file:network.txt

CLIENT OUTPUT

```
C:\Program Files\Java\jdk1.6.0\bin>javac FClient.java  
C:\Program Files\Java\jdk1.6.0\bin>java FClient
```

Connection request.....Connected

Enter the filename: network.txt

Computer networks: A computer network, often simply referred to as a network, is a collection of computers and devices connected by communications channels that facilitates communications among users and allows users to share resources with other user

RESULT

Thus the File transfer Operation is done & executed successfully.

EXP: 9b

CLIENT-SERVER APPLICATION FOR CHAT

AIM

To write a client-server application for chat using TCP

ALGORITHM

CLIENT

1. Start the program
2. Include necessary package in java
3. To create a socket in client to server.
4. The client establishes a connection to the server.
5. The client accept the connection and to send the data from client to server.
6. The client communicates the server to send the end of the message
7. Stop the program .

SERVER

1. Start the program
2. Include necessary package in java
3. To create a socket in server to client
4. The server establishes a connection to the client.
5. The server accept the connection and to send the data from server to client and vice versa
6. The server communicate the client to send the end of the message.
7. Stop the program .

PROGRAM

TCPserver1.java

```
import java.net.*;
import java.io.*;
public class TCPserver1
{
public static void main(String arg[])
{
ServerSocket s=null;
String line;
DataInputStream is=null,is1=null;
PrintStream os=null;
Socket c=null;
try
{
```

```

s=new ServerSocket(9999);
}
catch(IOException e)
{
System.out.println(e) ;
}
try
{
c=s.accept();
is=new DataInputStream(c.getInputStream());
is1=new DataInputStream(System.in);
os=new PrintStream(c.getOutputStream());
do
{
line=is.readLine();
System.out.println("Client:"+line);
System.out.println("Server:");
line=is1.readLine();
os.println(line);
}
while(line.equalsIgnoreCase("quit")==false);
is.close(); os.close()
;
}
catch(IOException e)
{
System.out.println(e) ;
} } }

```

TCPclient1.java

```

import java.net.*;
import java.io.*;
public class TCPclient1 {
public static void main(String arg[])
{
Socket c=null; String line;
DataInputStream is,is1;
PrintStream os;
try {
c=new Socket("10.0.200.36",9999);
}

```

```

catch(IOException e)
{
System.out.println(e) ;
}
try
{
os=new PrintStream(c.getOutputStream());
is=new DataInputStream(System.in);
is1=new DataInputStream(c.getInputStream())      ;
do
{
System.out.println("Client:");
line=is.readLine();
os.println(line);
System.out.println("Server:" + is1.readLine());
}
while(line.equalsIgnoreCase("quit")==false);
is1.close() os.close(); ;
}
catch(IOException e)
{
System.out.println("Socket Closed!Message Passing is over");
}}

```

OUTPUT:

SERVER

C:\Program Files\Java\jdk1.5.0\bin>javac TCPserver1.java

Note: TCPserver1.java uses or overrides a deprecated API.

Note: Recompile with -deprecation for details.

C:\Program Files\Java\jdk1.5.0\bin>java TCPserver1

Client: Hai Server

Server: Hai Client

Client: How are you

Server: Fine

Client: quit

Server: quit

CLIENT

C:\Program Files\Java\jdk1.5.0\bin>javac TCPclient1.java

Note: TCPclient1.java uses or overrides a deprecated API.

Note: Recompile with -deprecation for details.

C:\Program Files\Java\jdk1.5.0\bin>java TCPclient1

Client: Hai Server

Server: Hai Client

Client:How are you

Server: Fine

Client:quit

Server: quit

RESULT

Thus the above program a client-server application for chat using TCP / IP was executed and successfully.