o.	Name of Experiment	Submission Date	Page Number	Re
.	Introduction to packet tracer and networking device components	16/01/19	3-5	
2.	Study of basic commands of Cisco Packet Tracer	23/01/19	6-7	
3.	Configuration of static routing protocol	30/01/19	8-9	
4.	Configuration of RIP protocol.	06/02/19	11-12	
5.	Configuration of OSPF protocol.	13/02/19	14-15	
6.	Implementation of DHCP server in cisco packet tracer	06/03/19	17-19	
7.	Implementation of network address translation in cisco packet tracer	27/03/19	21-25	
8	Socket progromming using UDP Socket	03/04/19	27-30	
9.	Socket programming using TCP Socket	09/04/19	31-35	
10.	Network utilities	17/04/19	36	

LAB -1

INTRODUCTION TO PACKET TRACER AND NETWORKIND DEVICE COMPONENTS

acket Tracer is a powerful network simulator that allows user to create network topologies . It is a nnovative features help students and teachers collaborate, solve problems, and learn concepts. It llows users to create networks with almost unlimited number of devices and to experience ompensating without having to buy real Cisco switches and routers. This is a application created y Cisco. It is a user friendly Command Line Interface (CLI).

EATURES:

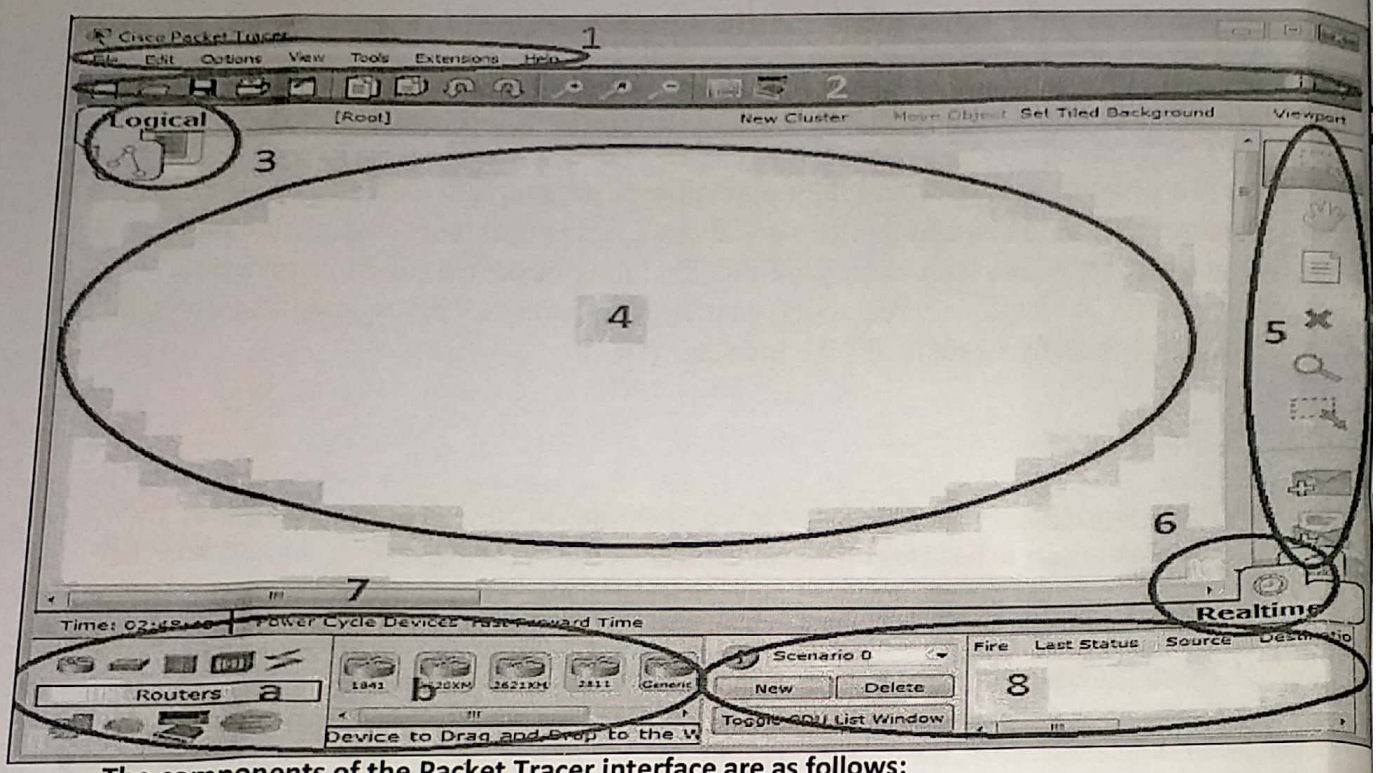
- **Packet Tracer workspace:** Cisco Packet Tracer has 2 workspaces (1) Logical and (2) Physical. The logical workspace allows users to build logical network topological by planning, connecting and clushing virtual network devices. The physical workspace provides a graphical physical dimension of logical network.
- **Packet Tracer Modes**: Cisco Packet Tracer provides 2 operating modes to visualize the behavior of a network – real time mode and simulation mode. In real time mode, the network behaves as real devices. In simulation mode, the user can see and control time intervals, the inner workings of data transfer and the propagation of data across a network.
- **Modular Devices:** Graphical representations visually simulate hardware and offer the ability to insert interface cards into modular routers and switches, which then become a part of simulation.

ETAILS:

- **Menu Bar:** This bar provides the file, edit, option, view, tools, extensions and help menus. You will find basic commands such as open, save, save as, print and preferences in these menus. You will also be able to access activity wizard from extensions menu.
- **Tool Bar:**This bar provides shortcut icons to the file and edit menu commands. This bar also provides buttons for copy, paste, undo, redo, zoom and drawing palette and custom devices dialog. On the right, you will also find the network information button, which you can use to enter a description for current network.
- **Workspace:**This area is where you will create your network, watch simulations, and view any kinds of information and statics.

Interface Overview:

The layout of Packet Tracer is divided into several components similar to a photo editor. Match the numbering in the following screenshot with the explanations given after it:



The components of the Packet Tracer interface are as follows:

- **Area 1: Menu bar** – This is a common menu found in all software applications;it is used to open, save, print, change preferences, and so on.
- **Area 2: Main toolbar** – This bar provides shortcut icons to menu optionsthat are commonly accessed, such as open, save, zoom, undo, and redo, and on the right-hand side is an icon for entering network information for the current network.
- **Area 3: Logical/Physical workspace tabs** – These tabs allow you to togglebetween the **Logical** and **Physical** work areas.
- **Area 4: Workspace** – This is the area where topologies are created andsimulations are displayed.
- **Area 5: Common tools bar** – This toolbar provides controls for manipulatingtopologies, such as select, move layout, place note, delete, inspect, resize shape, and add simple/complex PDU.
- **Area 6: Realtime/Simulation tabs** – These tabs are used to toggle betweenthe real and simulation modes. Buttons are also provided to control the time, and to capture the packets.
- **Area 7: Network component box** – This component contains all of thenetwork and end devices available with Packet Tracer, and is further divided into two areas:
  - **Area 7a: Device-type selection box** – This area contains devicecategories

- **Area 7b: Device-specific selection box** – When a device categoryis selected, this selection box displays the different device models within that category

Area 8: User-created packet box – Users can create highly-customizedpackets to test their topology from this area, and the results are displayed as a list.

LAB 2

STUDY OF BASIC COMMANDS OF CISCO PACKET TRACER

A VLAN is a group of devices on one or more LANs that are either on same LAN network or in different LAN networks. Using VLAN, we can set up the communication between different devices in such a way that they may be assumed on same LAN Network.

VLANs are used to give same level of priorities to different devices either in same network or in different network. Since VLANs are based on logical connections instead of physical connections, hence they are extremely flexible. Switches are used to logically divide the network on different LANs. Switches are multi p bridges that allow you to create multiple broadcast domains. Each broadcast domain is like a distinct bridge within a switch.

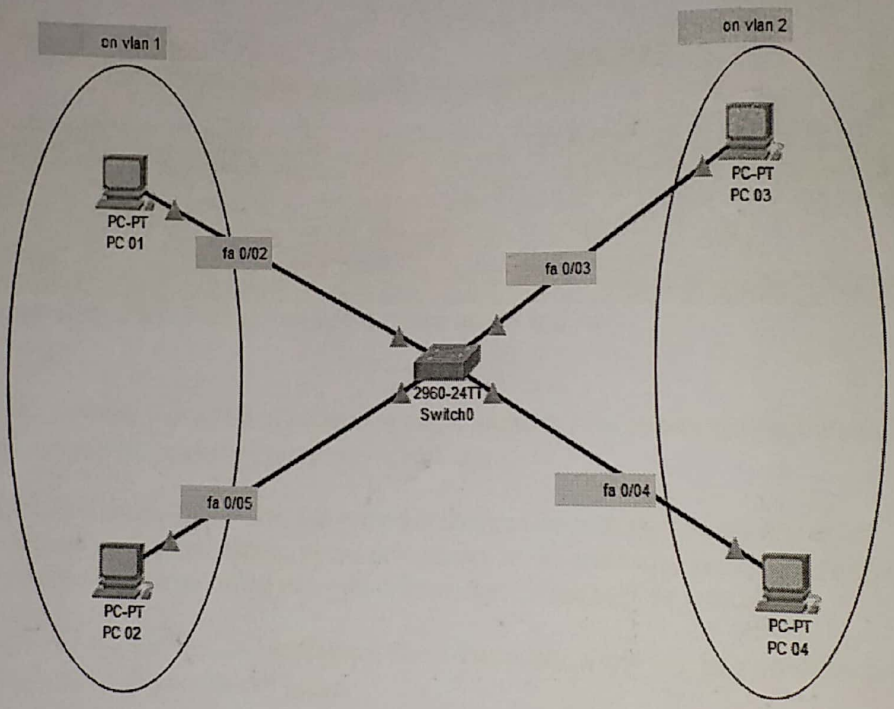


Figure showing connected PCs over different VLANs

To configure the VLAN, we need to type following commands on CLI (Command Line Interface):

- Type **enable** on the CLI to enter into the terminal.
- Type **config t** to enter into configuration mode.
- Type **vlan vlan_id** to create a VLAN with **vlan_id** as VLAN number. Note that the number ranges fro to 4094.
- Then you may give name to the created vlan by using command **name vlan_name**
- Then to change the interface state to up, use command **no shut**.
- Then exit the configuration of VLAN using **exit** as the command.
- Now, to set a PC on a new VLAN, enter into configuration mode again using **config t**.
- Now add port to newly created vlan using command **int fa0/1** (say for example).

- Now type **switchport access vlan vlan_id**. Here I add fast-ethernet port number 0/1 to the vlan just created above.
- To verify that the port was successfully added to new vlan, check using **show vlan** command. And done! You have added your port to new vlan.

The steps to configure interfaces fa 0/2 and fa 0/3 onto another vlan.

```
Switch(config)#
Switch(config)#
Switch(config)#
Switch(config)#vlan 3
Switch(config-vlan)#exit
Switch(config)#int fa 0/2
Switch(config-if)#switchport access vlan 3
Switch(config-if)#exit
Switch(config)#int fa 0/3
Switch(config-if)#switchport access vlan 3
Switch(config-if)#exit
Switch(config)#
```

ow ping PCs over same and different LANs:

```
acket Tracer PC Command Line 1.0
:\>ping 192.168.100.10

inging 192.168.100.10 with 32 bytes of data:

eply from 192.168.100.10: bytes=32 time<1ms TTL=128
eply from 192.168.100.10: bytes=32 time<1ms TTL=128
eply from 192.168.100.10: bytes=32 time=5ms TTL=128
eply from 192.168.100.10: bytes=32 time=5ms TTL=128

ing statistics for 192.168.100.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
pproximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 5ms, Average = 2ms
```

On same VLAN

```
C:\>ping 192.168.100.30

Pinging 192.168.100.30 with 32 bytes of data:

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

Ping statistics for 192.168.100.30:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

On different VLANs

LAB 3

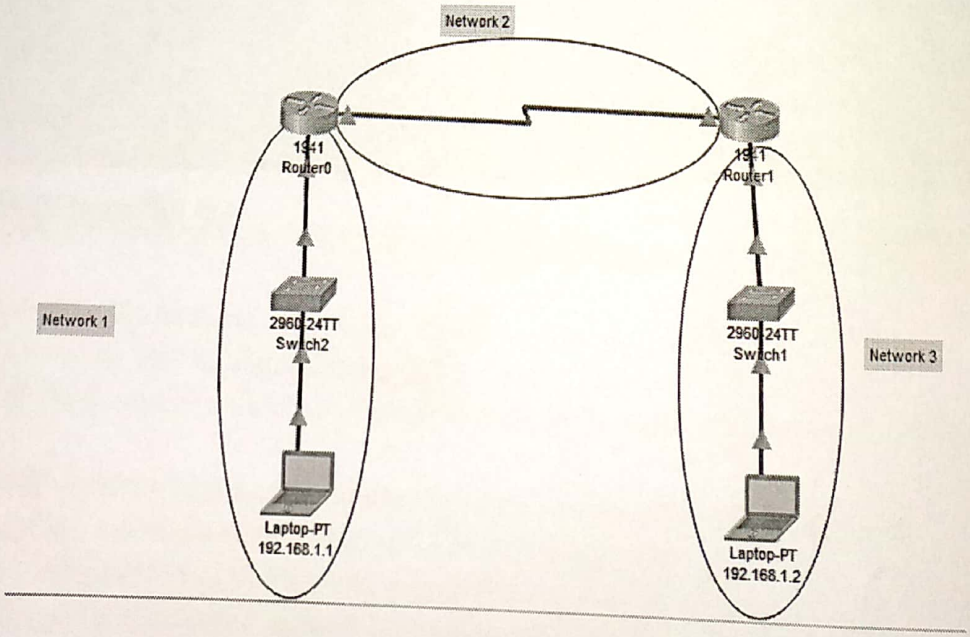
CONFIGURATION OF STATIC ROUTING PROTOCOL

Router:

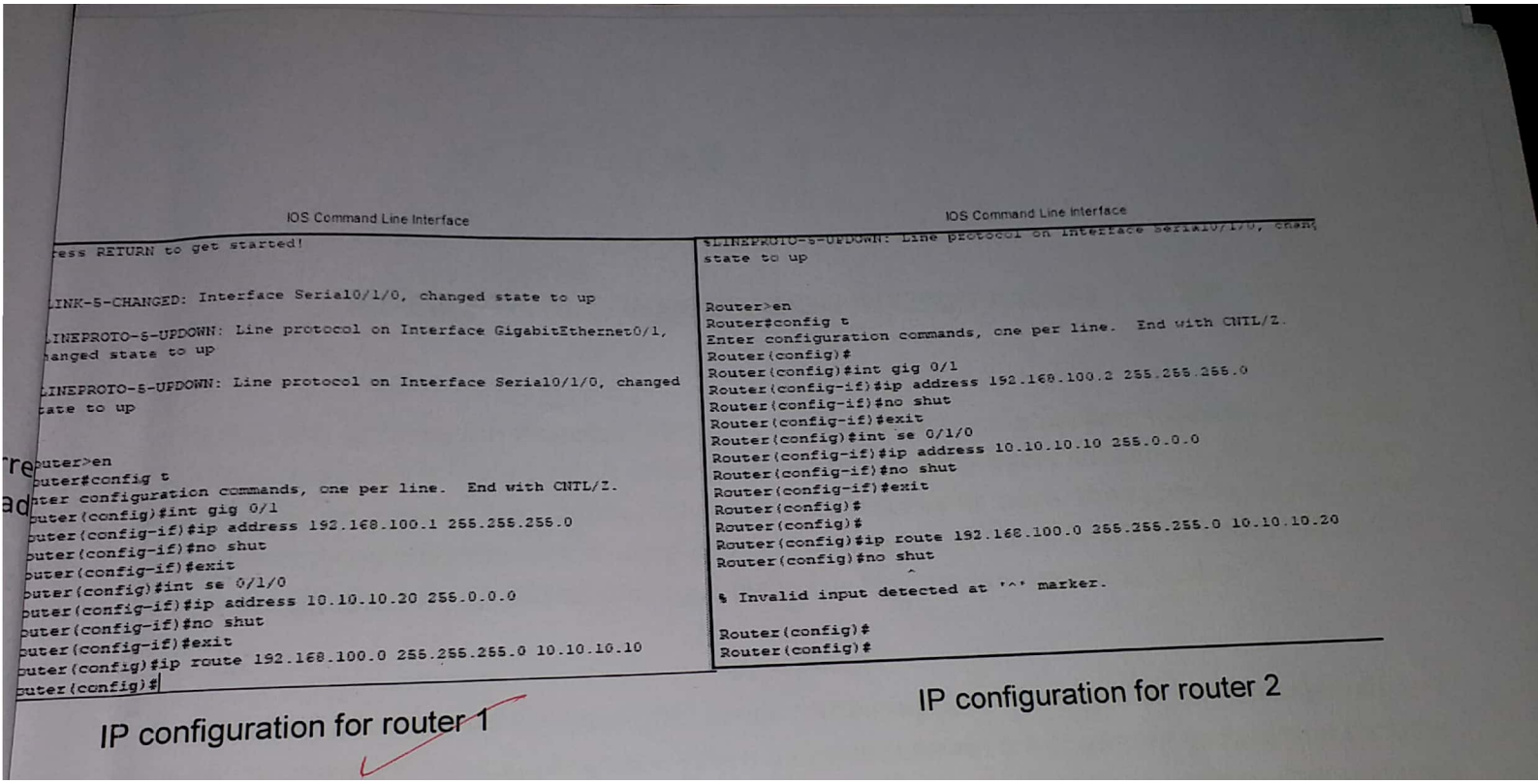
A router is a physical or virtual appliance that passes information b/w two or more networks. Routers determine whether the source and destination are on the same network or whether data must be transfer from one network type to another, which requires encapsulation the data packet with routing protocol hea information for the new network type.

Configuration of Routers:

- **Configuring Gigabit Ethernet Interfaces (Same for both routers):**
  - Type **enable** to enter into CLI.
  - Type **config t** to enter into configuration mode.
  - Type **int gig slot/port** to configure gigabit port on the router.
  - Type **ip address ip_address mask** to configure IP address to the network.
  - Type **no shut** to change the gigabit ethernet state to up.
  - Type **exit** to exit the configuration mode.
- **Configuring Static Routes:**
  - To configure static routes, perform these steps in CLI:
    - Type **config t** to enter into configuration mode.
    - Type **ip route** prefix mask {ip_address | interface_type interface_number [ip_addre
      - **Example:**
        - Router(config)# ip route 192.168.1.0 255.255.0.0 10.10.10.2.
    - Type **end** to exit the router configuration mode.



Router and different networks

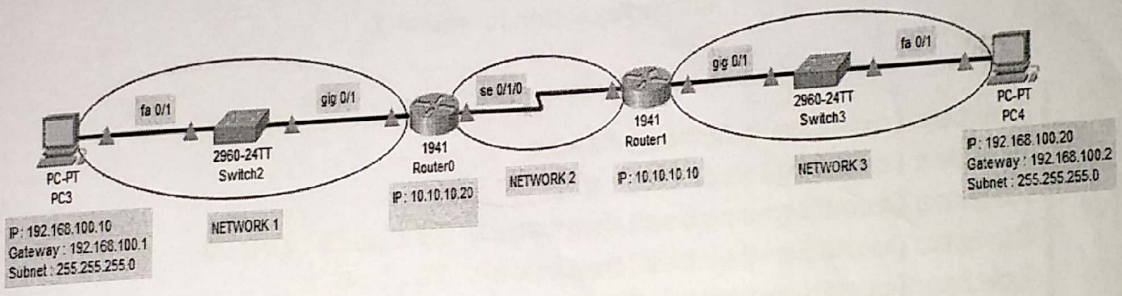


LAB - 4

RIP PROTOCOL CONFIGURATION IN CISCO PACKET TRACER

The Routing Information Protocol (RIP) uses broadcast UDP data packets to exchange routing information. Cisco software sends routing information updates every 30 seconds. If a device does not receive an update from another device for 180 seconds or more, the receiving device marks the routes served by the non routing device as unusable. If there is still no update after 240 seconds, the device removes all routing table entries for the non updating device.

The Routing Information Protocol (RIP) sends routing-update messages at regular intervals and when the network topology changes. When a device receives a RIP routing update that includes changes to an entry, the deice updates its routing table to reflect the new route. After updating its routing table, the device immediately begins transmitting RIP routing updates to inform other network devices of the change.



The network topology

CONFIGURATION OF RIP USING CLI IN CISCO PACKET TRACER:

The steps to set up the RIP is same to some point as setting a router network link explained in previous blog. The further steps to follow are given below:

- Type **no ip route network_ip network_subnet gateway_router_address**.
  - (e.g.): **no ip route 192.168.100.0 255.255.255.0 10.10.10.20**
- Then type **router rip** and hit enter.
- Then type **network_id** to add and **gateway_id** of the network.

* Do same for second router. The complete set of instructions in CLI are shown below.

```
Router(config-router)#
Router(config-router)#exit
Router(config)#router rip
Router(config-router)#network 192.168.200.0
Router(config-router)#network 10.0.0.0
Router(config-router)#exit
Router(config)#
```

Ctrl+F6 to exit CLI focus

RIP configuration for router 1

```
Router(config-router)#exit
Router(config)#router rip
Router(config-router)#network 192.168.100.0
Router(config-router)#network 10.0.0.0
Router(config-router)#exit
Router(config)#
```

Ctrl+F6 to exit CLI focus

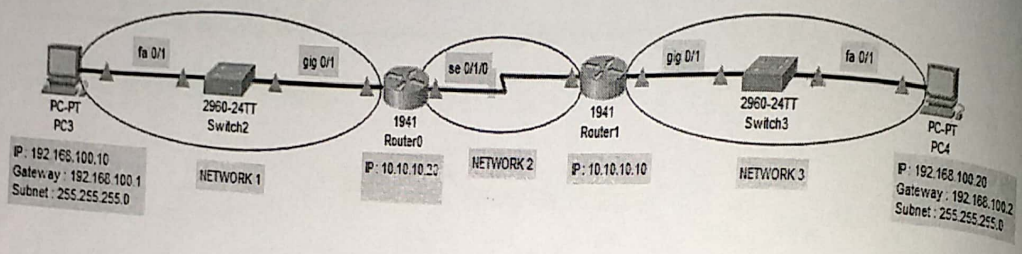
RIP configuration for router 2

LAB - 5

OSPF PROTOCOL CONFIGURATION IN CISCO PACKET TRACER

OSPF is an Interior Gateway Protocol (IGP). OSPF was designed expressly for IP networks and it supports IP subnetting and tagging of externally derived routing information. OSPF also allows packet authentication and uses IP multicast when sending and receiving packets. The OSPF protocol is a link-state routing protocol, which means that the routers exchange topology information with their nearest neighbors.

The topology information is flooded throughout the Autonomous System(AS). This picture is then used to calculate end-to-end paths through the AS, normally using **Dijkstra's algorithm.**



The network topology

Configuration of OSPF using CLI in Cisco Packet Tracer:

- The steps to set up the RIP is same to some point as setting a router network link explained in previous blog. The further steps to follow are given below:
- Type **no ip route network_ip network_subnet gateway_router_address.**
  - (e.g.): **no ip route 192.168.100.0 255.255.255.0 10.10.10.20**
- Then type **router ospf process_id**. (*here process id is any number such as 99).
- Then type **network network_ip_address wild_mask area area_number.**

(wild_mask is inverted subnet mask and area_number is process_id declared above).

* Do same for second router. The complete set of instructions in CLI are shown below.

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1/0, chang

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1/0, chang

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1/0, chang


Router>enable
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router ospf 99
Router(config-router)#network 192.168.100.0 0.0.0.255 area 99
Router(config-router)#network 10.10.10.10 0.255.255.255 area 99
Router(config-router)#exit
Router(config)#
```

Ctrl+F6 to exit CLI focus

OSPF configuration for router 1

```
%LINK-5-CHANGED: Interface Serial0/1/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1/0, chang


Router>enable
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#router ospf 89
Router(config-router)#network 192.168.200.0 0.0.0.255 89
                                                       ^
% Invalid input detected at '^' marker.

Router(config-router)#network 192.168.200.0 0.0.0.255 area 89
Router(config-router)#network 10.10.10.20 0.255.255.255 area 89
Router(config-router)#exit
Router(config)#
```

OSPF configuration for router 2

LAB 06

IMPLEMENTATION OF DHCP IN CISCO PACKET TRACER

The DHCP (dynamic host configuration protocol) is a standardized network protocol which is used in internet protocol (IP) networks. DHCP is used to assign IP automatically to the system with the help of a machine called DHCP server. A DHCP server enables computers to request IP addresses and networking parameters automatically. In the absence of a DHCP server, each computer on the network needs to statically (manually) assigned to an IP address.

Static IP addresses are usually assigned to routers, management interfaces on switches, servers and other devices in the network which do not change location either physically or logically. Static IP addresses are also used to access and manage these devices remotely.

On the other hand, user devices such as computers, smartphones, IP phones and others are likely to change their locations either physically or logically. This means that assigning them static IP addresses would be an unviable solution. DHCP is a protocol that was invented to address these problems. With DHCP, we can assign IP address information to user nodes automatically which saves on the administrative overhead that would be involved in assigning IP addressing information to clients statically.

The image below shows the working of DHCP server

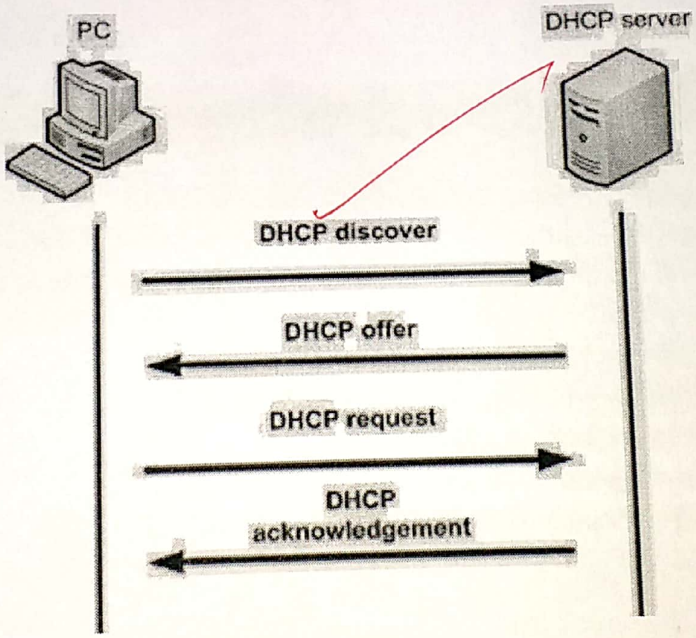


Figure 6.1

- First, the client sends the broadcast request to all DHCP servers to see if they can provide him with IP
- Then all the DHCP servers listening to the client request offers the availability of IP addresses to t client if IP addresses are available.
- Then the client asks for a IP allocation request to any of the DHCP server (usually 1st offer is accept and others are rejected).
- Then the requested DHCP server allocates the IP address to the client and client connects to t particular server for information flow.

STEPS FOR CONFIGURATION:

- Create a network topology of a PC, a switch, and a router as shown below.

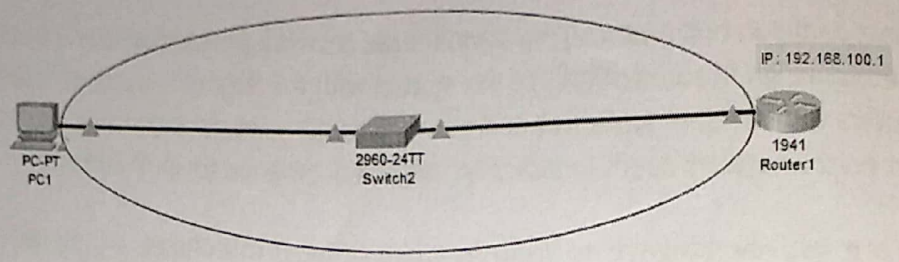


Figure 6.2

- Figure 6.2 shows the network topology for DHCP server and client connection. The client is connecte to Router, Router 1, which acts like a DHCP server whose configuration is discussed below. When cli sends a DHCP request to the router (currently as a DHCP server also), then Router, Router 1, sends a of the available IP to the router and assigns that IP to the router.

- To configure router, Router 1, as a DHCP serve, go to CLI of router and type commands shown below

```
Router>en
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int gig 0/1
Router(config-if)#ip address 192.168.100.1 255.255.255.0
Router(config-if)#no shut
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up
Router(config-if)#exit
Router(config)#ipdhcp pool IP01
Router(dhcp-config)#net 192.168.100.0 255.255.255.0
Router(dhcp-config)#default 192.168.100.1
Router(dhcp-config)#exit
Router(config)#ipdhcpexc 192.168.100.1 192.168.100.15
Router(config)#exit
Router#
```

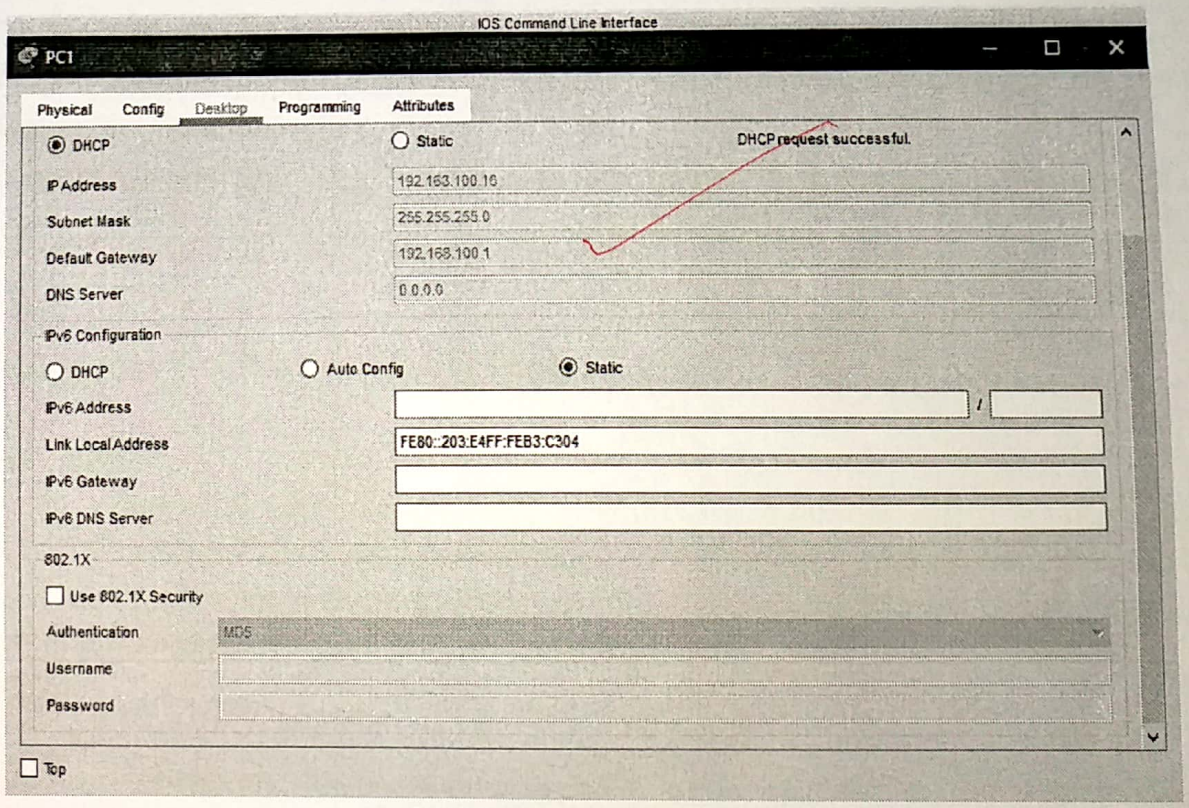
- Then go to the PC and then click on DHCP button in IP configuration.
- If everything is fine, then your PC will be issued an IP address with a message "**DHCP Req Successful".**

NAPSHOTS:

```
Press RETURN to get started!


Router>en
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int gig 0/1
Router(config-if)#ip address 192.168.100.1 255.255.255.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up

Router(config-if)#exit
Router(config)#ip dhcp pool IP01
Router(dhcp-config)#net 192.168.100.0 255.255.255.0
Router(dhcp-config)#default 192.168.100.1
Router(dhcp-config)#exit
Router(config)#ip dhcp exc 192.168.100.1 192.168.100.15
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#
```

Ctrl+F6 to exit CLI focus

DHCP configuration in CLI



DHCP success Message

LAB 07

Implementation of network address translation in cisco packet tracer

Network address translation (**NAT**) is a method of remapping one IP address space into another by modifying network address information in the IP header of packets while they are in transit across a traffic routing device. The main use of NAT is to limit the number of public IP addresses an organization or company must use, for both economy and security purposes.

NAT inside and outside addresses :

Inside refers to the addresses which must be translated. Outside refers to the addresses which are not in control of an organization. These are the network Addresses in which the translation of the addresses will be done.

- **Inside local address –** An IP address that is assigned to a host on the Inside (local) network. The address is probably not a IP address assigned by the service provider i.e., these are private IP address. This is the inside host seen from the inside network.
- **Inside global address –** IP address that represents one or more inside local IP addresses to the outside world. This is the inside host as seen from the outside network.
- **Outside local address –** This is the actual IP address of the destination host in the local network after translation.
- **Outside global address –** This is the outside host as seen form the outside network. It is the IP address of the outside destination host before translation.

Network Address Translation (NAT) Types:

- **Static NAT:**
  In this, a single unregistered (Private) IP address is mapped with a legally registered (Public) IP address i.e. one-to-one mapping between local and global address. This is generally used for Web hosting. These are not used in organizations as there are many devices who will need Internet access and to provide Internet access, public IP address is needed.

  Suppose, if there are 3000 devices who needs access to Internet, the organization have to buy 3000 public addresses that will be very costly.

- **Dynamic NAT:**
  In this type of NAT, an unregistered IP address is translated into a registered (Public) IP address from a pool of public IP address. If the IP address of pool are not free, then the packet will be dropped as only fixed number of private IP address can be translated to public addresses.

Advantages of NAT:

- NAT conserves legally registered IP addresses .
- It provides privacy as the device IP address, sending and receiving the traffic, will be hidden.
- Eliminates address renumbering when a network evolves.

Disadvantage of NAT:

- Translation results in switching path delays.
- Certain applications will not function while NAT is enabled.
- Complicates tunneling protocols such as IPsec.

CONFIGURATION:

TOPOLOGY:

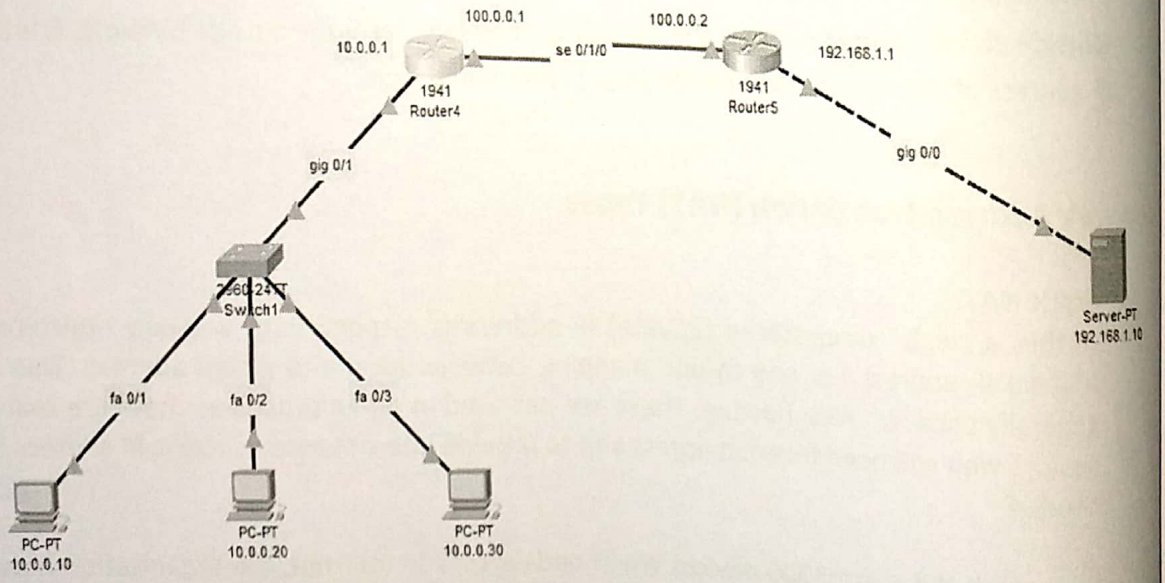


Figure 7.1

Create a n/w topology as shown in Figure 7.1. I have taken 3 PCs with IP address as **10.0.0.10, 10.0.0.20, 10.0.0.30**. These PCs are connected to a switch which is connected to a router **Router4** with **gig 0/1.** The interface IP address of **gig 0/1** is **10.0.0.1.** The **Router4** is connected to **Router5** via a serial ports **se 0/1/0** in both the routers.

The interface IP address for **Router4** at **se 0/1/0** is **100.0.0.1.** The interface IP address for **Router5** at **se 0/1/0** is **100.0.0.2. Router5** is connected to a server whose IP address is **192.168.1.10** via **gig 0/0** port whose interface IP address is **192.168.1.1.**

Now run the following commands as below to setup **static NAT.**

ON ROUTER 4:

```
Router>en
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int gig 0/1
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shut
Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up
Router(config-if)#int se 0/1/0
Router(config-if)#ip address 100.0.0.1 255.0.0.0
Router(config-if)#no shut

%LINK-5-CHANGED: Interface Serial0/1/0, changed state to down
Router(config-if)#
Router(config-if)#
%LINK-5-CHANGED: Interface Serial0/1/0, changed state to up

Router(config-if)#exit
Router(config)#ip nat inside source static 10.0.0.10 50.0.0.10
Router(config)#int gig 0/1
Router(config-if)#ip nat inside
Router(config-if)#exit
Router(config)#int se 0/1/0
Router(config-if)#ip nat outside
Router(config-if)#exit
Router(config)#ip nat inside source static 10.0.0.20 50.0.0.20
Router(config)#ip nat inside source static 10.0.0.30 50.0.0.30
Router(config)#exit
Router(config)#ip route 200.0.0.0 255.255.255.0 100.0.0.2
Router(config)#exit
```

ON ROUTER 5:

```
Router>en
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int gig 0/0
Router(config-if)#ip address 192.168.1.1 255.255.255.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up
Router(config-if)#int se 0/1/0
Router(config-if)#ip address 100.0.0.2 255.0.0.0
Router(config-if)#no shut

Router(config-if)#
%LINK-5-CHANGED: Interface Serial0/1/0, changed state to up

Router(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1/0, changed state to up

Router(config-if)#ip nat inside source static 192.168.1.10 200.0.0.10
Router(config)#int gig 0/0
Router(config-if)#ip nat inside
Router(config-if)#exit
Router(config)#int se 0/1/0
Router(config-if)#ip nat outside
Router(config-if)#exit
Router(config)#ip route 50.0.0.0 255.0.0.0 100.0.0.1
Router(config)#exit
```

NAT Successful:

```
C:\>ping 200.0.0.10

Pinging 200.0.0.10 with 32 bytes of data:

Request timed out.
Reply from 200.0.0.10: bytes=32 time=10ms TTL=126
Reply from 200.0.0.10: bytes=32 time=3ms TTL=126
Reply from 200.0.0.10: bytes=32 time=4ms TTL=126

Ping statistics for 200.0.0.10:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 10ms, Average = 5ms

C:\>ping 192.168.1.10

Pinging 192.168.1.10 with 32 bytes of data:

Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.
Request timed out.
Reply from 10.0.0.1: Destination host unreachable.

Ping statistics for 192.168.1.10:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>
```

LAB – 08

Socket Programming using UDP Socket

Socket:

Sockets allow communication between two different processes on the same or different machines. A Unix Socket is used in a client-server application framework. A server is a process that performs some functions on request from a client. The client in socket programming must know two information:

- IP Address of Server, and
- Port number.

Java DatagramSocket class:

Java DatagramSocket class represents a connection-less socket for sending and receiving datagram packets.

A datagram is basically an information but there is no guarantee of its content, arrival or arrival time.

Commonly used Constructors of DatagramSocket class:

- **DatagramSocket() throws SocketEeption:** it creates a datagram socket and binds it with the available Port Number on the localhost machine.
- **DatagramSocket(int port) throws SocketEeption:** it creates a datagram socket and binds it with the given Port Number.
- **DatagramSocket(int port, InetAddress address) throws SocketEeption:** it creates a datagram socket and binds it with the specified port number and host address.

Java Program for Server side using UDP:

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.SocketException;

public class udpBaseServer_2 {
        public static void main(String[] args) throws IOException {
                DatagramSocket ds = new DatagramSocket(1234);
                byte[] receive = new byte[65535];
                DatagramPacket DpReceive = null;
                while (true) {
                        DpReceive = new DatagramPacket(receive, receive.length);
                        ds.receive(DpReceive);
                        System.out.println("Client:-" + data(receive));
                        if (data(receive).toString().equals("bye")) {
                                System.out.println("Client sent bye.....EXITING");
                                break;
                        }
                        receive = new byte[65535];
                }
        }
        public static StringBuilder data(byte[] a) {
                if (a == null)
                        return null;
                StringBuilder ret = new StringBuilder();
```

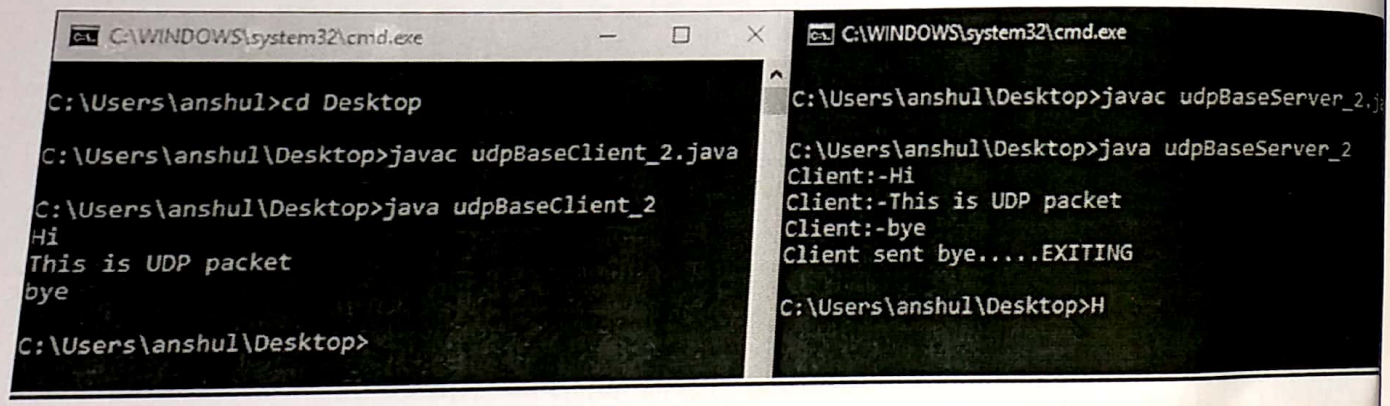
```
        int i = 0;
        while (a[i] != 0) {
            ret.append((char) a[i]);
            i++;
        }
        return ret;
    }
}
```

Java Program for Client side using UDP:

```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;

public class udpBaseClient_2 {
    public static void main(String args[]) throws IOException {
        Scanner sc = new Scanner(System.in);
        DatagramSocket ds = new DatagramSocket();
        InetAddress ip = InetAddress.getLocalHost();
        byte buf[] = null;
        while (true) {
            String inp = sc.nextLine();
            buf = inp.getBytes();
```

```
                DatagramPacket DpSend =
                        new DatagramPacket(buf, buf.length, ip, 1234);
                ds.send(DpSend);
                if (inp.equals("bye"))
                        break;
            }
        }
    }
```

Snapshots:



Client **Server**

LAB – 09

Socket Programming using TCP Socket

Socket:

Sockets allow communication between two different processes on the same or different machines. A Unix Socket is used in a client-server application framework. A server is a process that performs some functions on request from a client. The client in socket programming must know two information:

- IP Address of Server, and
- Port number.

Server Side Programming in Java:

Method	Description
1) public Socket accept()	returns the socket and establish a connection between server and client.
2) public synchronized void close()	closes the server socket.

Java Program for server side:

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

public class Server
{
    //initialize socket and input stream
    private Socket          socket   = null;
    private ServerSocket    server   = null;
```

```
    private DataInputStream in       =  null;

    // constructor with port
    public Server(int port)
    {
        // starts server and waits for a connection
        try
        {
            server = new ServerSocket(port);
            System.out.println("Server started");

            System.out.println("Waiting for a client ...");

            socket = server.accept();
            System.out.println("Client accepted");

            // takes input from the client socket
            in = new DataInputStream(
                new BufferedInputStream(socket.getInputStream()));

            String line = "";

            // reads message from client until "Over" is sent
            while (!line.equals("Over"))
            {
                try
                {
                    line = in.readUTF();
                    System.out.println(line);

                }
                catch(IOException i)
                {
                    System.out.println(i);
                }
            }
            System.out.println("Closing connection");

            // close connection
            socket.close();
            in.close();
        }
        catch(IOException i)
        {
            System.out.println(i);
        }
    }

    public static void main(String args[])
    {
        Server server = new Server(5000);
    }
}
```

Client side programming in Java:

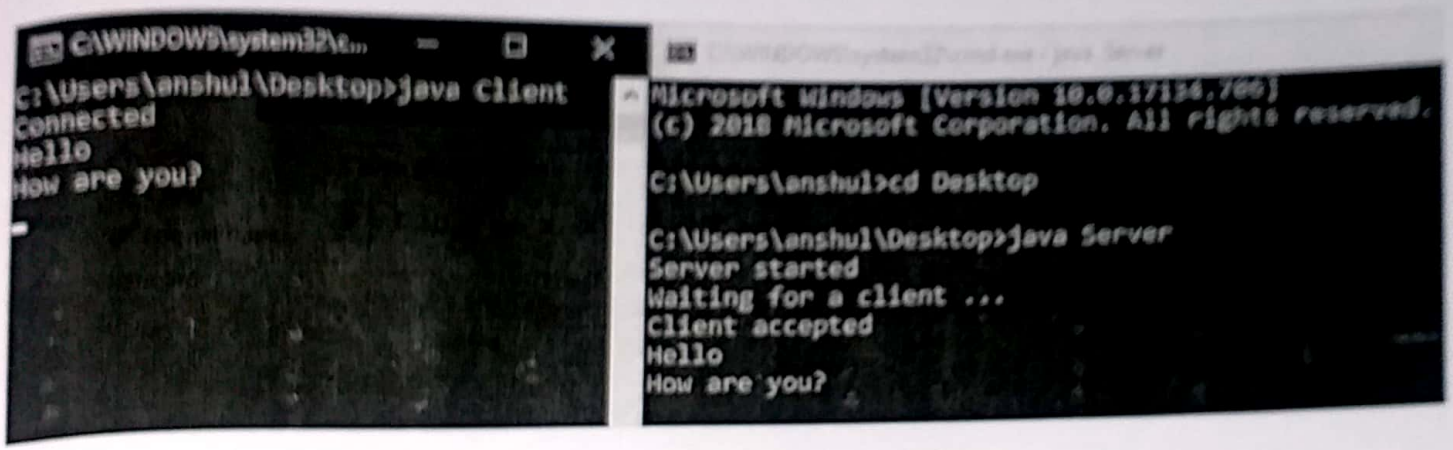
Method	Description
1) public InputStream getInputStream()	returns the InputStream attached with this socket.
2) public OutputStream getOutputStream()	returns the OutputStream attached with this socket.
3) public synchronized void close()	closes this socket

Java Program for client side:

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

public class Client
{
    // initialize socket and input output streams
    private Socket socket          = null;
    private DataInputStream  input   = null;
    private DataOutputStream out     = null;

    // constructor to put ip address and port
    public Client(String address, int port)
    {
        // establish a connection
        try
        {
            socket = new Socket(address, port);
            System.out.println("Connected");

            // takes input from terminal
            input  = new DataInputStream(System.in);

            // sends output to the socket
            out    = new DataOutputStream(socket.getOutputStream());
        }
        catch(UnknownHostException u)
        {
            System.out.println(u);
        }
        catch(IOException i)
        {
            System.out.println(i);
        }
```

```
            // string to read message from input
            String line = "";

            // keep reading until "Over" is input
            while (!line.equals("Over"))
            {
                try
                {
                    line = input.readLine();
                    out.writeUTF(line);
                }
                catch(IOException i)
                {
                    System.out.println(i);
                }
            }

            // close the connection
            try
            {
                input.close();
                out.close();
                socket.close();
            }
            catch(IOException i)
            {
                System.out.println(i);
            }
        }

        public static void main(String args[])
        {
            Client client = new Client("127.0.0.1", 5000);
        }
    }
```

Snapshots:



Client

Server

LAB - 10

Network Utilities

1. Ping:

Ping is a computer network administration software utility used to test the reachability of a host on an Internet Protocol network. It is available for virtually all operating systems that have networking capability, including most embedded network administration software.

2. Netstat:

In computing, netstat is a command-line network utility tool that displays network connections for the Transmission Control Protocol, routing tables, and a number of network interface and network protocol statistics.

3. Ipconfig:

In computing, ipconfig is a console application of some operating systems that displays all current TCP/IP network configuration values and can modify Dynamic Host Configuration Protocol and Domain Name System settings.

4. Ifconfig:

Ifconfig is a system administration utility in Unix-like operating systems for network interface configuration. The utility is a command-line interface tool and is also used in the system startup scripts of many operating systems.

5. ARP:

The Address Resolution Protocol is a communication protocol used for discovering the link layer address, such as a MAC address, associated with a given internet layer address, typically an IPv4 address. This mapping is a critical function in the Internet protocol suite.

6. Trace-route:

In computing, traceroute and tracert are computer network diagnostic commands for displaying the route and measuring transit delays of packets across an Internet Protocol network.