Page No.
Date

Assignment - 6

* Title: TCP Socket Programming

* Problem statement:
Write a program using TCP socket for wired network for following
a) Say Hello to each other (for all students)
b) File Transfer (for all students)
c) Calculator (Arithmetic) (50% students)
d) Calculator (Trigonometery) (50% students)
Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.

* Objective:
- To learn about TCP Socket Programming
- File Transfer using TCP.

* Outcomes:
Students will be able to:
- Implement TCP Socket Programming.

* Theory:
- The TCP is one of the main protocols of the Internet protocol suite.
- TCP is connected oriented and a connection between client and server is established before data can be sent.

Teacher's Sign.: _____

Page No.
Date

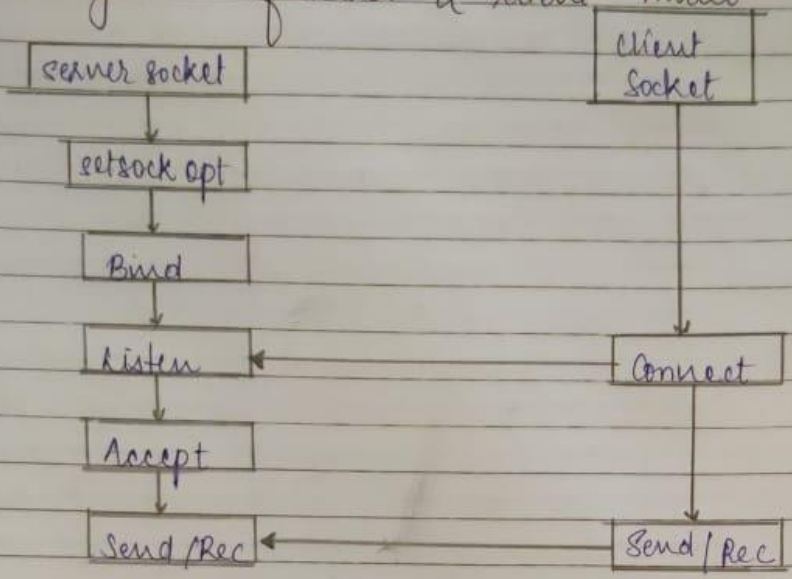
- The server must be listening for connection requests form clients before a connection is established
- TCP is dominantly used for the web, i.e., for the HTTP protocol, and local later HTTP/2, while not used by latest standard HTTP/3.

- TCP ports:
  - TCP uses port numbers to identify sending and receiving application end-points on a host, often called Internet protocols sockets.
  - Each side of a TCP connection has an associated 16-bit unsigned port number (0-65535) reserved by the sending or receiving application.
  - Arriving TCP packets are identified on belonging to a specific TCP connection has an by its sockets, that is, the combination of source host address, and destination port
  - Port numbers are categorized into three basic categories: well-known, registered, and dynamic / private.

Teacher's Sign.: _____

Page No.
Date

State diagram of server & client model.



* Conclusion:

Thus we studied & implemented Java program to demonstrate TCP socket programming for wired network.

Teacher's Sign.: _____

CODE:

Client.java:

```
import java.net.*;
import java.io.*;
import java.util.Scanner;
public class Client
{
    // initialize socket and input output streams
    private Socket socket = null;
    private Scanner sc = null;
    private DataOutputStream out = null;
    private DataInputStream in=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
            sc = new Scanner(System.in);
            in=new DataInputStream(socket.getInputStream());
// sends output to the socket
            out = new DataOutputStream(socket.getOutputStream());
        }
        catch(UnknownHostException u)
        {
            System.out.println(u);
        }
        catch(IOException i)
        {
            System.out.println(i);
        }
        int choice=1;
        do {
            try {
                System.out.println("Enter \n1.Chat \n2.File Transfer
\n3.Trigonometric Calculator \n0.Exit:");
                choice = sc.nextInt();
                out.writeByte(choice);
                switch (choice) {
// string to read message from input
                    case 1:
                        String line = "";
// keep reading until "Over" is input
                        while (!line.equals("Over")) {
                            line = sc.nextLine();
                            out.writeUTF(line);
                            System.out.println(in.readUTF());
                        }
                        break;
                    case 2:
                        System.out.println("Please enter file type:" );
                        sc.nextLine();
```

```
                        String ext=sc.nextLine();
                        out.writeUTF(ext);
                        System.out.println("Please enter file path:");
                        String file = sc.nextLine();
                        FileInputStream fis = new FileInputStream(file);
                        File file1=new File(file);
                        out.writeByte(file.length());
                        byte[] buffer = new byte[4096];
                        while (fis.read(buffer) > 0) {
                            out.write(buffer);
                        }
                        fis.close();
                        System.out.println("file transfer complete");
                        break;
                    case 3:
                        System.out.println("Enter angle in degrees:");
                        double theta=sc.nextDouble();
                        out.writeDouble(theta);
                        System.out.println("Enter \n1.Sin \n2.Cos \n3.Tan:");
                        int ch=sc.nextInt();
                        out.writeByte(ch);
                        System.out.println(in.readUTF());
                }
            } catch (IOException i) {
                System.out.println(i);
            }
        }while(choice!=0);
// close the connection
        try
        {
            sc.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);
    }
}
```

Server.java:

```
import java.net.*;
import java.io.*;
import java.util.Scanner;
public class Server
{
    //initialize socket and input stream
    private Socket socket = null;
    private ServerSocket server = null;
    private DataInputStream in = null;
    private DataOutputStream out=null;
    private Scanner sc=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(socket.getInputStream());
            out=new DataOutputStream(socket.getOutputStream());
            sc=new Scanner(System.in);
            int choice=1;
            do {
                try {
                    choice = in.read();
                    switch (choice) {
                        case 1:
                            String line = "";
// reads message from client until "Over" is sent
                            while (!line.equals("Over")) {
                                line = in.readUTF();
                                System.out.println(line);
                                out.writeUTF(sc.nextLine());
                            }
                            break;
                        case 2:
                            String ext=in.readUTF();
                            FileOutputStream fos = new
FileOutputStream("D:/testfile."+ext);
                            byte[] buffer = new byte[4096];
                            int filesize = in.read(); // Send file size in
separate msg
                            int read = 0;
                            int totalRead = 0;
                            int remaining = filesize;
                            while((read = in.read(buffer, 0,
Math.min(buffer.length, remaining))) > 0) {
                                totalRead += read;
                                remaining -= read;
                                System.out.println("read " + totalRead + "
bytes.");
                                fos.write(buffer, 0, read);
                            }
                            fos.close();
                            break;
                        case 3:
                            double theta=in.readDouble();
                            int ch=in.read();
                            switch(ch){
                                case 1:
                                    double radTheta= Math.toRadians(theta);

out.writeUTF(String.valueOf(Math.sin(radTheta)));
                                    break;
                                case 2:
                                    radTheta= Math.toRadians(theta);
```

```
out.writeUTF(String.valueOf(Math.cos(radTheta)));
                                    break;
                                case 3:
                                    radTheta= Math.toRadians(theta);

out.writeUTF(String.valueOf(Math.cos(radTheta)));
                                }
                            }
                        } catch (IOException i) {
                            System.out.println(i);
                        }
                    }while(choice!=0);
                    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);
            }
}
```

Output:

Server:

Server started
Waiting for a client ...
Client accepted

hello
hi
how are you?
great
thats good
Over
Okay
read 1 bytes.

Client:

Client Connected
Enter
1.Chat
2.File Transfer
3.Trigonometric Calculator
0.Exit:
1
hello

```
hi
how are you?
great
thats good
Over
okay
Enter
1.Chat
2.File Transfer
3.Trigonometric Calculator
0.Exit:
3
Enter angle in degrees:
45
Enter
1.Sin
2.Cos
3.Tan:
2
0.7071067811865476
Enter
1.Chat
2.File Transfer
3.Trigonometric Calculator
0.Exit:
2
Please enter file type:
txt
Please enter file path:
D:/Assignment_A6.txt
File transfer complete
Enter
1.Chat
2.File Transfer
3.Trigonometric Calculator
0.Exit:
0

Process finished with exit code 0
```

```
}while(choice!=0);
// close the connection
try
{
sc.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);
}
}
```

Process finished with exit code 0