

Assignment - 6

Problem Statement -

Write a program using TCP socket for wired network for following

- a) Say Hello to each other
- b) File transfer
- c) calculator (Trigonometry)

Objective -

To learn TCP socket and implement client & server program.

Outcome -

Learn concept of TCP socket programming

Implement program for client and server interaction.

S/W and H/W Requirements -

Windows 10, 8GB RAM, i7 processor, Monitor, Keyboard, Mouse, Eclipse.

Theory -

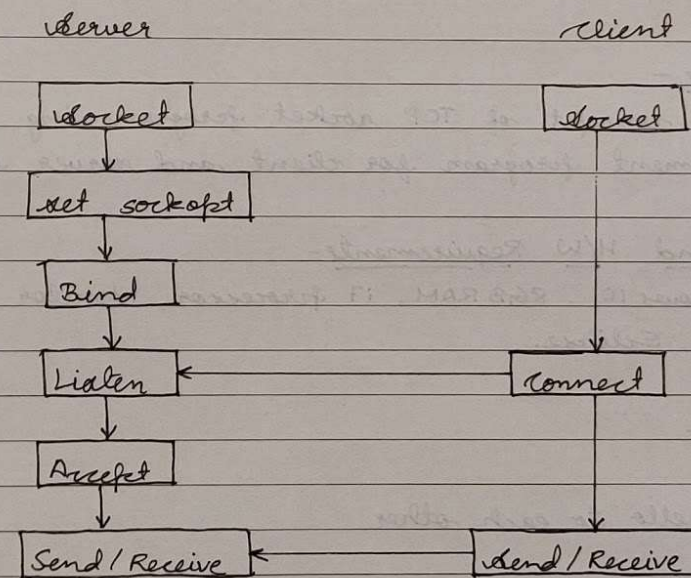
- a) say hello to each other

1. TCP Socket Programming for wired network

If we are creating a connection between client and server using TCP then it has few functionality like, TCP is suited for applications that require

high reliability and transmission time is relatively less critical. It is used by other protocols like HTTP, HTTPS, FTP, SMTP, Telnet. TCP rearranges data packets in order specified. There is absolute guarantee that the data transferred remains intact and arrives in the same order in which it was sent. TCP does flow control and requires three packets to set up a socket connection, before any user data can be sent.

The process can be broken down into following steps.



TCP Server

1. use `create()`, create TCP socket
2. use `bind()`, Bind the socket to server address.
3. use `listen()`, put the server socket in a passive mode, where it waits for client to approach the server to make connection.
4. using `accept()`, at this point, connection is established between client and server, they are ready to transfer data.
5. Go back to step 3.

TCP Client

1. create TCP socket
 2. Connect newly created client socket to server.
2. Running Socket Programs -
1. Run `server.c` file & create an output file for that in Unix or Linux.
 2. Type `g++ server.c`
 3. run by command `./a.out`
 4. After running the server just minimize the terminal, open new terminal
 5. Run `g++ client.c` & for output `./a.out`

b) File Transfer -

A TCP client initiates the communication with server which is waiting for the connection. TCP is connection oriented and UDP is connectionless, which

means that UDP sockets do not need to be connected before being used. Another difference between TCP and UDP is that there is no guarantee that a msg sent via a UDP socket will arrive at its destination, and messages can be delivered in a different order than they were sent.

The TCP listener is created and starts listening to the specified port. Again the buffer size is set to 1024 bytes. A TCP listener can pre-check to see if there are any connections pending before calling the accept TCP client method. It returns true if there are any pending connections.

- Conclusions -

Thus, we successfully implemented TCP socket program with client server interaction.