Indian Institute of Engineering Science and Technology, Shibpur
Department of Computer Science and Technology
Computer Network Lab (CS 3272)

Assignment 3: Chat application development using TCP Socket

Time: 2 weeks

Write all the code using C/C++ programming languages. If you use C++, try avoiding any library support of Socket and Thread (if any).

Chat Application:

Problem-1:

Develop a simple TCP Client-Server application where the client app sends a predefined text message "Hello World!!" to the server app in a user-defined port of your choice. Upon receiving that message, the server app forwards the same message to the client app. Both server and client applications print the message.

Once you complete this, extend the program to support chatting. The client app sends the first message it wishes to. Upon receiving that message, the server app prints the received message. Then, it sends the response message to the client. This message exchange continues until one party (server or client) says "Bye" and closes the chat session. After receiving the "Bye" message, another party closes the chat session.

Note that in this way of chatting, one end cannot send more than one message at a time. On sending one message, it has to wait for the response from another party, and then only it can send the next message.

You should run the server application on your host computer (laptop/desktop), and the client application should run on the departmental servers Hamsa (10.2.1.40) or Hanau (10.2.1.41). You should log in to the Hamsa/Hanau using your respective login Id.

Problem-2:

This problem aims to get rid of the imitation stated in Problem-1. The chatting mechanism should be flexible, so sending and receiving messages should be concurrent. In other words, sending and receiving messages should not block each other. Users may send no or multiple messages before receiving a response from another party.

It can be achieved using the Process or Thread concept, where "reading a text from a user and sending" job can be given to a Process or a Thread. Again, the "receiving a text message and printing it in the console" job can be given to another Process or a Thread. In this assignment, you are encouraged to use the Thread concept instead of Process.

You are suggested to make sure that the termination of the chat session happens *gracefully!!*

Note: Once problems 1 and 2 are completed, then attempt the following problem –

Problem-3:

Develop a TCP chat server application where multiple client applications participate in the chat. For example, if there are four clients, Client #1 has conversations with Client #2, and Client #3 has conversations with Client #4 simultaneously. The server application acts as a mediator that forwards the message to the intended client and not a party of chatting.