CSCI/CSIS 632 – Data Communications and Networking
Project 3 (Group Project – 2 students per group)
Due November 11, 2021

Background

In Project 2, you have learnt how to write multithreaded *client/server* programs with TCP sockets. The *client* program sends three registration packets, receives acknowledgement and waits to receive data packets. The *server* program accepts a connection from a *client*. If the connection is from a new *client*, it creates a *join handler* which registers the *client* by putting *client* information in a table. The *server* also has a *multicaster* thread which checks the table periodically. If the *multicaster* finds registered *clients* in the table, it creates data packets and sends them to the registered *clients*.

Project Description

In this project, finally you will implement a *chat server*. The *chat server* will have different chat rooms. Each chat room is represented by a multicast group. When a *client* joins the *chat server*, it will specify the name of the multicast group. After the *client* has received **Confirmation** packet from *server*, it will generate **Data** packets. The *client* program will get the information for **Data** packets from the user. The user will type something at the command line and press "ENTER" (to denote the end of data). The *client* program will generate **Data** packets with this data and send it to the *server*. The *server* after receiving the **Data** packets from the *client*, will check the multicast group of the *client* and multicast the data to all members of that Group. All members of that Group will receive the **Data** packet and print it on the screen. Both the *server* and *client* program will run infinitely. The *server* program will wait to receive **Data** packets from the registered *clients* or will wait to accept connections from new *clients*. The *client* program will wait to receive **Data** packets from the *server* or will get data from the user for forming **Data** packets.

The *server* should be able to handle multiple *clients* and multiple groups. You can assume that a *client* will be a member of one multicast group at any of point of time. Each multicast group is represented by an integer.

The *client* will tell the *server* the name of the multicast group during registration (as part of the registration packet). The registration table will store the information of multicast group for each *client*.

The number of thread required for *server* and *client* programs depends on your design.

You need to print out the following:

1) each request that is sent to the *server*
2) each request sent by the *client*
3) each response sent by the *server*
4) each response received by the *client*

What to upload in Canvas –

a. A softcopy of *server* program source file
b. A softcopy of *client* program source file
c. A softcopy of script file showing the output of *server* program
d. A softcopy of script file showing the output of *client* program.

Other Details – You are allowed to work in groups for this project. Maximum two students are allowed per group. Consultation in any form with other groups of the class is strictly prohibited. Your program should be documented thoroughly and 20% of the project grade will be for documentation.

Good Luck!