CSE 322: Computer Networks Sessional

Assignment 1

File Transmission Using Socket Programming

In this assignment, you will implement a file transmission system using socket programming. You have to use JAVA programming language only for the implementation.

Task Summary:

❏ **Configure server with maximum size of all chunks allowed in the buffer.**

❏ A student has a specific IP address.

❏ **A student can Login with student ID. If the student ID is already logged in from another IP address, new login is denied by the server.** Same student ID cannot be used to login from different PC (IP address). So each student ID can be uniquely mapped to an IP address at any instance.

❏ **Client can send files to a specific student (By entering a student ID as receiver).**
   ❏ If The receiver is logged in (connected to the server), start file transmission. Otherwise, generate an error message.

   ❏ While sending a file, **client(sender) first sends the filename and file size to the server. The server checks the total size of all chunks stored in the buffer plus the new file size. If it overflows the maximum size,** the server does not allow the transmission. **Otherwise**, the server randomly generates maximum chunk size and sends a confirmation message to the client(The client can start sending the file now) with the maximum chunk size and a <u>fileID</u> to the client, which is used as the file identifier for the rest of the file transmission. So, **you need to maintain a link between the fileID and fileName and other info inside the server**.

   ❏ **Now the sender splits the file into chunks depending on the maximum size of each chunk.** Let, a file size is 1235 KB and the server allows that client maximum 100 KB for each chunk. So, client splits the file into 13 chunks(first 12 chunks with 100KB, the last chunk with 35 KB) and sends all the chunks sequentially.

   ❏ **After receiving each chunk, the server sends an acknowledgement. The sender sends the next chunk only after receiving the acknowledgement.** If the sender does not receive any acknowledgement within 30 seconds, it sends a timeout message to the

server and terminates the transmission. **After receiving the timeout message, the server should delete the chunk files for the corresponding fileID**.

❏ When the sender receives acknowledgement for the last chunk, it sends a completion message to the server. Then the server checks the file size by adding all the chunk sizes. If size of all chunks matches the initial file size mentioned by the sender, the server is done with the sender and sends a success message. Else, the server sends an error message indicating failure and deletes all the chunks.

❏ Now the server asks the receiver whether he/she wants to receive the file mentioning file name, size and the sender student ID. If the receiver is willing to receive, the server transmits the chunks to the receiver. For this task, the implementation details is left to you. **After successful transmission to the receiver, the chunks are deleted from the server.**

❏ All the communications regarding file transmission are done using the **fileID** provided by the server. Note this very carefully.

❏ **A student goes offline if he/she logs out or gets disconnected.** If the sender goes offline in the middle of a file transmission (between sender and server), discard the files from server. If he/she goes offline after the transmission to server is done, it is the responsibility of the server to deliver the file to the receiver. **What you should do when the receiver goes offline in the middle of a transmission is open-ended and left as a bonus task.**

❏ Clarification:
For testing sender and receiver using only one PC, You can integrate port number with IP address to uniquely identify a client. So, Sender and Receiver can be differentiated with port number, although they have same IP address.

Submission Deadline:
October 01, 2017 (Sunday, 10:00 PM)
Deadline will not be extended at any request. Please start early.