Networks Assignment 2 Report

A.V.S Nikhil Srivatsava
cs1170353@cse.iitd.ac.in

Musunuru Saurav
cs1170352@cse.iitd.ac.in

I. Handling Ctrl-C

We used a blocking read which means that a readLine() function call will be stuck at that call until the underlying stream closes due to a socket connection close. When the stream closes, the readLine() returns null. This enables the Server to detect the closing of the stream and helps remove the user gracefully.

II. Offline User

Server maintains 4 hash tables. One is the username-socket table. Second is the username-status(Online or Offline). Third is the username-List of Pending Messages. Fourth is the username-publicKey table. First we need to take care that our KeyPair is not lost. So we store the encoded formats of the Public and Private keys offline in a file securely. Now when the user wishes to log out, he sends a message to the server and closes his sender and receiver sockets and threads also. The server on receiving the logout message retrieves the socket from the hashTable and closes it and updates the status of the user to offline and closes this thread also. Now when another user wants to send a message to the offline user, He sends the encrypted message to the server. Now the server checks the status and sees that the recipient is offline and hence pushes the message into the pendingMessages table. Now when the same user tries to register again the server sends a message to the client that this user is already registered and now makes the status online. Now the client retrieves its KeyPair from the local file and then the server handles all the pending messages in a row before listening to new messages from the sender. To make it more secure the local file can be encrypted with a publicKey of another KeyPair which is freshly generated by the server. Before logging out the publicKey is sent to the user and on coming back online the publicKey is sent to the server on which the server returns the privateKey to decrypt the file.