Tzu-Huai Chen
Kaito Hakamata
CSE 160 - Computer Networks

Project 4: Chat Client and Server

Design Process and Design Decisions

For the last project, we focused on utilizing the TCP commands that we created in order to set up a chat client and server. The main idea of Project 4 is giving the server names and establishing connection to each of them and making them talk to each other. We decided to centralize the commands into a chatInput command that is only called within run.py in order to organize what command is being called and to make the hello, message, whisper and print user handling easier. The actual hello, message, whisper and print user commands were based on the ones provided within the PDF. As for a simplification, we decided to store the username information within the socket information section despite being a layer violation. This was the easiest implementation that we could think of that would get the job done for this project. The naming of the servers were done within the datatransfer section utilizing the testclient command in order to rename a socket to be a certain name. The message portion focused on just printing the actual content being sent since it was being sent to everyone. The whisper portion focused on just sending the message to a specific socket so we needed to find the socket number of the given name and send the message. Lastly, the print users section we just needed to print out all of the sockets that are currently alive at the moment this command was called.

P.S. I wanted to thank you for this wonderful semester of knowledge and a fun project to work on. Despite trying the best of our abilities some tasks throughout this project were challenging and I hope for the leniency of grading, since we just want to pass this class. Thank you very much and we apologize for including this messaging within our Project 4 write up.

Discussion Questions

1. The pros of using the single transport connection is that there transferring of messages are immediate and do not need to be checked if the information was sent reliably or not. The cons of it would be the inability to reliably transfer the information. The pros of using a transport connection per command and reply will be the maintaining of reliability in data transfer. The cons will be the more steps necessary to set up as well as the more transfer information needed.
2. The features that are a good fit for the chat client and server application are the setting up a server and establishing a connection between a server and a client. The features that are not a good fit would be the closing of the servers since the current set up takes a while before the connected setup can be closed as well as the project 4 application doesn't have a close connection between server and client command. The best way we can deal with this would be creating a done chatting command to close the connections that are made.
3. The features of congestion control were not implemented but the good fit for the web server application would be able to control the flow of the information being transferred and read by the servers making sure that there is no delay as well as maintaining reliability for the data being sent. The only issue would be the more time needed to get this to work and the heavier overhead.
4. One improvement to the design would be fixing the socket from storing the username information and violating the layer rules. The fix to this issue would be creating another header file that deals with storing unique sockets and information that deal with the chat as well as deploying the application via another data transferring method besides the tcp method.