Renee Helfert
John Villalvazo
CSE 160 - Fall 2020
Project 4
Due: 12/8/2020 @3pm

1. The chat client and server application as described above uses a single transport connection in each direction per client. A different design would use a transport connection per command and reply. Describe the pros and cons of these two Designs.

In the case of the using a single transport connection in either direction of the client, what can be seen as pro's would be a more reliable connection as there is only one transaction occurring per command between client and server, but we would assume that the main drawback of this implementation would be that in a scaled up version we would run into bottle necks due to only one communication being sent in either direction. Comparing this to having a transport connection in both the command and reply between client and server, a client could be contacting and utilizing the server to its total potential and handling multiple tasks at the same time as each transport command would have their own identifiers, but this speed and efficiency comes at the potential for dropped responses between client and server and could be vulnerable to attacks as the server is saturated with communications back and forth between client(s).

2. *Describe which features of your transport protocol are a good fit to the chat client an Server application, and which are not. Are the features that are not a good fit simply unnecessary, or are they problematic, and why? If problematic, how can we best deal with them?*

Based on our progress with Project 3 leading into the implementation of the chat client and server, some features that we think are beneficial to our build would be the hard coding of our name size as this allows for our code to simply have their information readily available and does not need to compute that information, but this feature could be problematic as we are using test names that are known. If this were to be deployed, the effects of these limitations could potentially be code breaking down the line, and if not that a nuisance to the user.

3. *Read through the HTTP protocol specification covered in class. Describe which features of your transport protocol are a good fit to the web server application, and which are not. Are the features that are not a good fit simply unnecessary, or are they problematic, and why? If problematic, how can we best deal with them?*

Similar to Q2, our implementation of a hard coded name size is a feature that could have some positive and negative effects if deployed on an actual web server. A pro could be if a web page is static and have nothing changing after the initial load-ins, but once a dynamic page is requested, our code would severally decrease the PLT as not only can there only be 1 fetch done between client and server, the startup times of these responses that the server returns to the client (as how HTTP response does with a normal HTTP request), then this would not be beneficial to use practically.

4. *Describe one way in which you would like to improve your design.*

We would like to be able to handle data properly and send packets in an efficient manner. Our current code does not handle any commands cleanly and that could be fixed with some time and patience.