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.

- Two connections uses more space to create both sockets
- Two connections uses more socket numbers
- Two connections uses two ports
- If two connections are used, it is easier for a human to parse what data is transmitted on the client and server side

2. Describe which features of your transport protocol are a good fit for the chat client and 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?

It is very awkward having to manually manage the amount of bytes that the transport protocol is able to currently send on the application layer. It would be nice if I could just write all the data into the transport layer immediately, but the buffer size is limited to 128 bytes. Also, having a linear buffer makes it so extended conversations between devices get bottlenecked by having to wait for the 128th byte to be sent, so the segment can be reset. It would be nice to fix this with a cyclical buffer.

3. Even if you did not implement the extra credit application, read its protocol specification. Describe which features of your transport protocol are a good fit for 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?

I believe this question is a legacy question, because there is no extra credit application.

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

I would like to improve my design by making it so any node could become a server on an arbitrary port. Currently, I’ve hard coded the server to be set up on node 1, and port 41, because this was the specification of the project and I didnt feel like implementing more than that given our time constraints.