

Discussion Questions

1. How TCP or UDP would affect your application? Why you pick TCP over UDP? (Or vice versa)

a. TCP would affect my application by ensuring reliable transport, which means 100% of the data will be transported. TCP will also not overwhelm the receiver. In the case of my application, it will make sure the server does not become overwhelmed. TCP can also enact a throttle to prevent the sender from overwhelming the network. TCP can also negatively affect my application by not being as secure as possible. TCP also requires a connection to be established to work. At times using TCP could result in slower transmission because of a lot of traffic.

b. I picked TCP over UDP because TCP ensures 100% data transfer which I believe is very important. Without 100% data transfer the user may experience fragmented documents or other information, which could result in the user not understanding the data. TCP is also better at handling a lot of traffic because it can throttle the network, which in turn could cause the server not to crash.

2. Is your protocol stateful or stateless? How does the other type of design would affect your application? (Hint: this has nothing to do with whether you use TCP or UDP)

a. Because I used TCP my protocol is stateful. Stateful means the client and server will keep information about each other to make future connections faster and smoother. However, in this project, the server is stateless because it does not retain the information provided by the client.

b. UDP, on the other hand, is stateless. Stateless means no information will be kept between the client and the server. This type of design in the protocol may slow down future connections because the client will always have to re-provide information to the server every time it connects. If this project was stateful the server would need more storage to store all information provided by the clients. The server also may lose some of that data and therefore not recognizing certain clients which would slow down future connections.

3. Is your server program able to communicate with multiple clients at the same time?

a. My server can communicate with multiple clients at the same time because it is in a stateless protocol. Packets will be read by the order in which they arrive from the clients.