**Protocol Design**

* **What kinds of messages will be exchanged across the control channel?**

Initially, the server will create a socket, prepare socket address, and bind socket address to port. While listening to the socket and waiting for client, the client will be connecting to the server with the address and port number given.

* **How should the other side respond to the messages?**

Once the client is connected to the server with the address and port number, the server will accept the connections and await commands from user which would result in the client sending the command. The server will receive that FTP command from client and begin its execution.

* **What sizes/formats will the messages have?**

The buffer size of the messages are 256 bytes; meanwhile, the max connections size is set up as 100.

* **What message exchanges have to take place in order to setup a file transfer channel?**

First off, the server will create a socket, prepare socket address, and bind socket address to port. Whilst the server is listening to the socket and waiting for client, the client will be connecting to the server with the address and port number given. Once the client is connected to the server, the server will then accept connection and awaits command from client. The client will send a command, get, which would result in the server acknowledging that a file transfer channel has been requested. The server will send file to the client, open it if exists and fill buffer with data. A while loop will then begin executing in order to write to the data channel until the buffer is completely empty. The client will then download file from server, connect to the server data channel socket and writes the command. Once the file has been opened, a file exist verification and file size will be sent in order to write file to buffer and send it through the data channel. Once a file transfer channel has been setup, the client and server will close data channel socket and be ready to quit.

* **How will the receiving side know when to start/stop receiving the file?**

The receiving side will know when to start/stop receiving the file as both the client and the server are constantly communicating with each other. The server can acknowledge whether the file has been successfully opened or not and inform the client. The client has the option to confirm that the file has been received and the server will be ready to close the data channel socket and quit.

* **How to avoid overflowing TCP buffers?**

In order to avoid overflowing the TCP buffer, a while loop was implemented. The while loop consists of comparing the received bytes with the read’s client buffer size. If it is, in fact, overflowing, an error message will be printed out, informing that the content from the file in packets were not received from client.



