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

Server Side: Asks for the name of the file to be transferred.

Client Side: Asks for a host name

• How should the other side respond to the messages?

Server Side: Generates a hostname that will be used by the client side.

Client Side: Generates a connection successful message after the correct hostname is inputted.

• What sizes/formats will the messages have?

The data will be sent in 1024 byte sizes.

• What message exchanges have to take place in order to set up a file transfer channel?

After we start up our server, it will generate a host name. Copy the host name. Next, start up the client. The client will ask for the host name. Paste the copied host name into this section and click enter. On the server side, it will ask which file we want to transfer. There are two files available to send, Krabby.jpg and Bikini\_Bottom.jpg. Type in one of those file names in order to start the file transfer.

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

The receiving side will receive the name of the file it will be receiving. The file is then sent in chunks. A temporary number, numBytes, will be used to measure if all of the file has been sent. As the number of bytes sent increases, so does numBytes. It checks the number of bytes the size of the file is vs. the amount of bytes sent. While the numBytes is less than the size of the file, it will keep receiving the file. A file transfer complete output will be generated when the file has been received fully. The transferred file will be in the ‘rec’ folder.

• How to avoid overflowing TCP buffers?

The data will be sent out chunks, in our case 1024 bytes. This way we control how many bytes are sent out at a time to prevent overflow.