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

Strings are exchanged along the control connection from the client to let the server know what the commands are and what port the data channel is expected to be on.

They are exchanged from server to client to inform the client if a file does not exist or any other errors.

**How does the other side respond to each type of message?**

When the server receives the command from the client it expects a command along with a port number for the data channel as well as filename if it is a ‘get’ or ‘put’ command. Once it receives that information the transfer is started between the client and server.

If a file does not exist, the client waits for another command from the user. Otherwise, if it receives a file does exist it will continue with the transfer.

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

The messages will be strings and since we are using python 3, the programmer must decode messages before sending them with a socket. Thus the messages are sent in byte code.

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

The client should randomly select an open port and let the server know what port it plans on using for the data transfer. Then they can both connect via that port.

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

Once the connection is closed on one of the sides the side receiving can assume the file has been fully transferred.

**You may want to use diagrams to model your protocol.**

**The FTP client is designed similar to this.**

The client file has a getrandomport() function to get a random unused port and sends to the server.

get <filename> <port>

Message Structure

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Size | sp | Command | sp | Filename (if required) | sp | Port |
| Message | | | | | | |



****

**Extra Credit Protocol Diagram**

