

CPSC 471 Assignment #3

- Protocol Documentation -

**By: Kelly Siska, Shaun Feltman, Farruk Kamal,
Sara Mortazavi, and RJ McKeehan**

Design and Functionality of the Protocol(s)

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

The kinds of messages that will be exchanged across the control panel will be:

help – displays the list of commands that are usable for the program
get <file_name> - downloads the designated file name from the server
put <file_name> - uploads the designated file to the to the server
ls – this command will display a list of files that exist on the server
quit – terminates the process

How should the other side respond to the messages?

The other side should respond to the following messages by acknowledging the request, establishing a file transfer connection for which the data (in this case, the files) are sent across, carry out the request (either upload or download the file), and then once the command is completed, close the socket connection and await further instructions.

What sizes/formats will the messages have?

The sizes that the messages will have will be sent through the connection link up to the size, but not exceeding, the size of the data buffer (i.e. TCP buffer). The format that these messages will be sent through will be of type 'string'.

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

In order to setup a file transfer channel, an ephemeral port needs to be created between the client and the server. The client makes a request to the server through the file transfer system and the server responds with an accept/reject message after it identifies the fts source and destination. Once the connection is established and accepted, the request can be carried out to either get a file from the server or to upload a file to the server. The server acknowledges/rejects this request and once the transfer has been completed, the transfer connection is closed (i.e. the socket) and the server awaits for additional commands from the client.

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

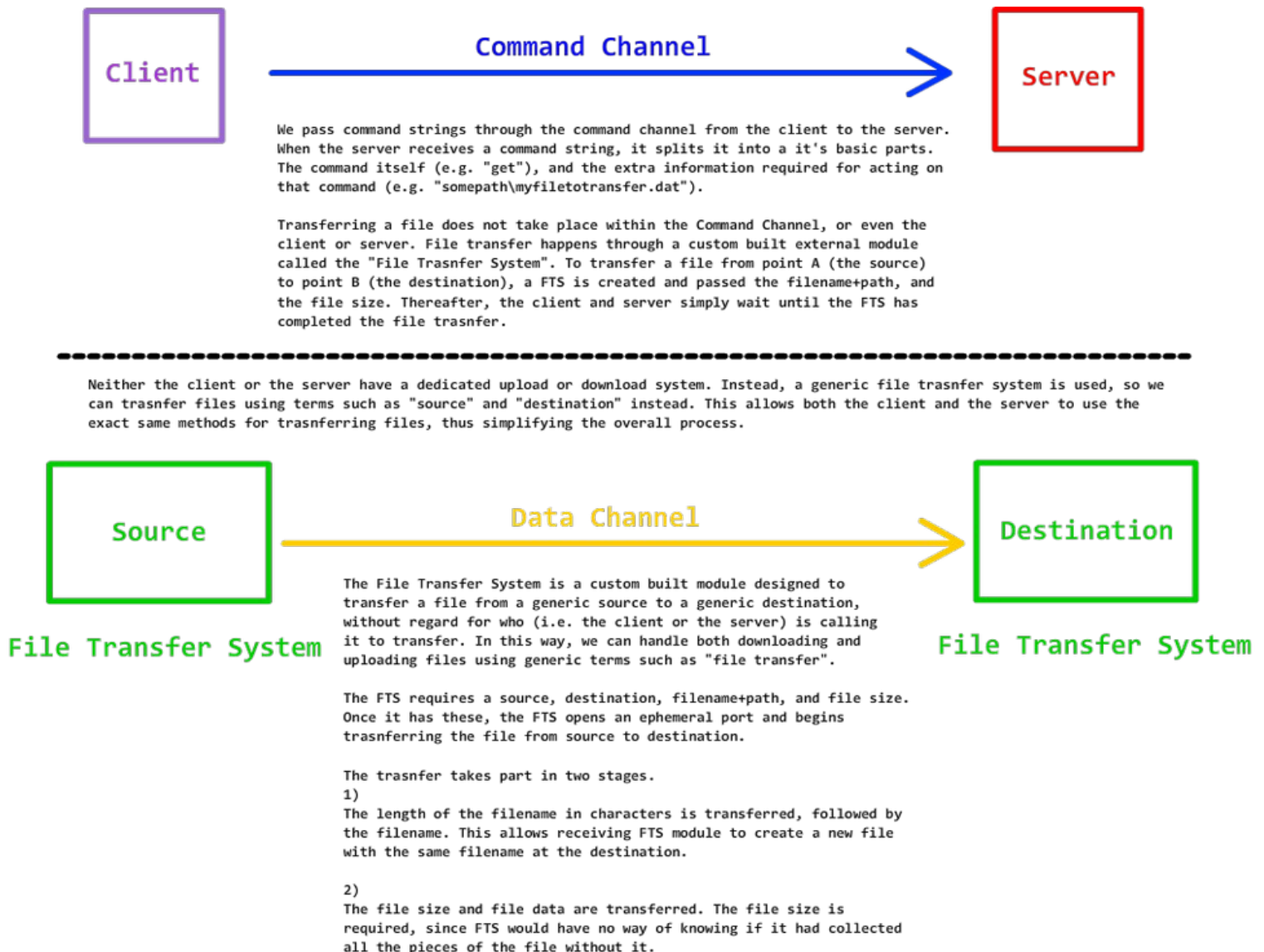
The receiving side will know when to start/stop receiving the file for, in the protocol, the size of the file to be received is specified. Therefore, when the receiving end is sent data on the file, it will know when to start receiving the file when the bytes it has received is 0, and will stop receiving the file when the number of bytes received equals the number of bytes indicated that the file size to be sent was.

How to avoid overflowing TCP buffers?

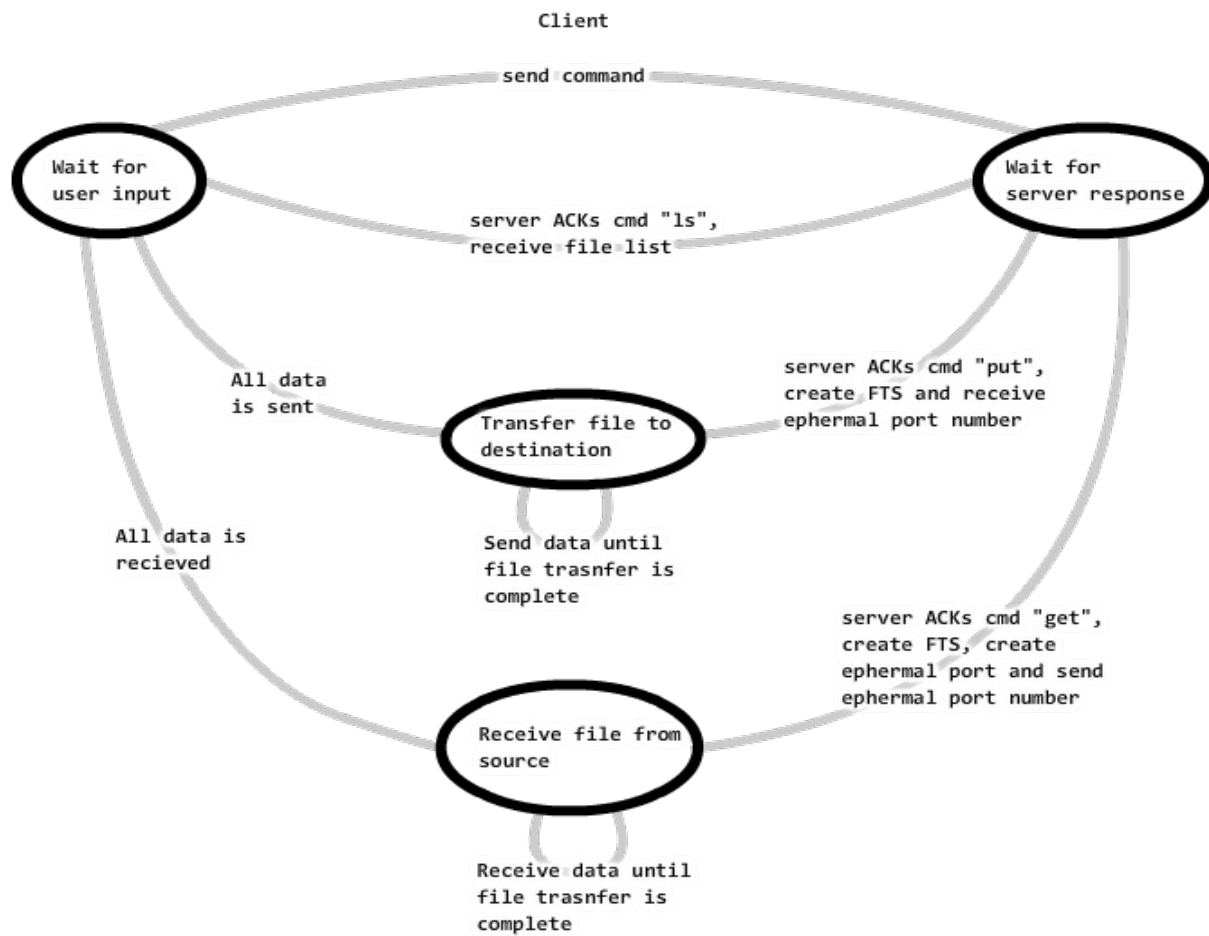
In order to avoid overflowing the TCP buffers, before the file to be sent is transmitted to the receiver, the file is broken down into discrete byte-sized chunks with each chunk representing a portion of the file. These chunks will then be sent out to the receiving end, where the data will then be re-assembled back into the total size of the file. The reason why we wish to break down the file into these “chunks” is due to the idea that it is easier for the TCP buffers to handle files and avoid overflow if the size of the file is essentially reduced. We simulate this by breaking down the total size of the file and sending the portions through the link to the receiving end, as opposed to trying to send through the entire bulk of the file where, if the file is enormously large, it becomes more difficult for the TCP buffers to handle and thus, overflowing could occur.

The following diagrams below are used to help illustrate the design of the protocol(s):

*** Protocol Design of the Command Channel and the Data Channel**



*** Diagram Illustrating the Client setup**



*** Diagram Illustrating the Server setup**

