Cpsc 471 project

Members:

- James peou
- Renell Miller
- Jhoshua Ampo
- Carlos Fuentes
- Thanedh Sithisombath

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

For the clients, we will have to send command messages (get, ls, put, quit). Then, the server will have to send back the status of the command. If it was successful, it will send back the message that it was successful. However, if it fails then it would send an error message.

2. How should the otherside respond to the messages?

The otherside will respond with a status message if it was successful in sending the command message or an error message that it failed to send it.

3. What sizes/formats will the messages have?

The sizes of the messages should be very small i think like close to about 20 bytes for both files. The Amount of sizes largely depends on the amount of data that is received.

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

In order to setup the file transfer channel the client would need to have send a command to the server which should include get, ls, or put. Next, the server will have to create a socket and then blind the socket like welcomeSock.bind(('', listenPort)). As this will select an port that is available to blind to. Then, we find out what the port number is and send it to the client. The server will wait and listen for a connection, once the client connects we have a data channel. All the necessary data will be transfer through the channel and once it's completed, the data sockets will close, but the control the channel for the message will still be on. The quit command will close the control channel.

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

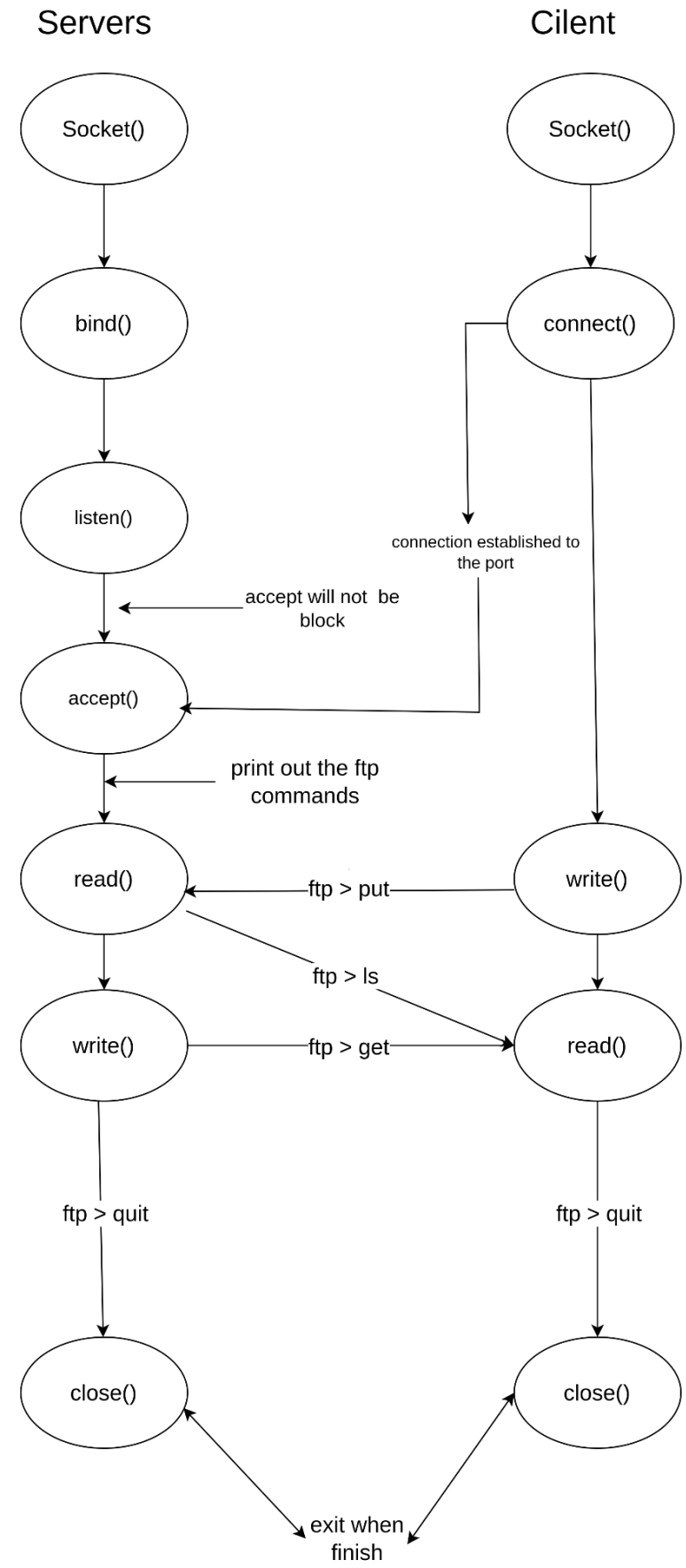
First off, the client will have to be connected to the sockets that the server is blind to. After this, the server will continue to listen to the client for its message. It will know that it

receives or sends a file when a client sends using commands like put, get or ls. Then, the server will create and send an ephemeral port to the client to connect to. Which the client will connect and then the server will send or receive data depending on the commands that it uses. The receiving end knows when to stop receiving the files by looking at the header and getting the size of the data.

6. How to avoid overflowing TCP buffers?

To avoid TCP buffers we would need to make sure to check in place to not send over the TCP buffer size. So, to check the file that is download or upload to make sure that it less than buffer size (65,536 bytes). If it happens to be over the buffer size then, the programs will not allow the transfer to occur and will give an error.

Diagram:


Servers
Cilent
Socket()
Socket()
bind()
connect()
listen()
connection established to the port
accept will not be block
accept()
print out the ftp commands
read()
ftp > put
write()
ftp > ls
write()
ftp > get
read()
ftp > quit
ftp > quit
close()
close()
exit when finish