

Possible Names for protocol:

- (ಠ_ಠ)
- FTS
- True42 (42 ppl in the zoom call right now)
- FTPBB (FTP but better)
- RFC5462

*****(Any errors found in any part close the connection)*****

Step by Step Layout of protocol

For first Client -> Server message

Header Field to indicate file size

- Client -> server
- 8 bytes (unsigned long, converted to network byte order)
- Requires:
 - Big endian (network order)
 - Both server and client support files of up to 2^{64} bytes

Header Field to indicate file name

- Client -> server
- Variable byte length
- Max size is 256 bytes; so 255 bytes for characters, then one NUL ($\backslash 0$) byte. If no null byte by the 256th byte, server sends error
- Right after the null byte the data of the file begins
- Ended by null character
- Requires:
 - ASCII encoding
 - File names cannot contain (if the file name contains multiple null characters it will assume the first one was supposed to be the end of the filename)
 - . (and nothing else)
 - .. (and nothing else)
- When sent to the server, only the filename will be sent. The server will return an error if any slashes are present ($'/'$, $'\backslash'$).
- Existing files (on the server) will be overwritten
- What is sent as the file name is "bar.txt", and the server creates a file named bar.txt in the 'recvd' directory

Data Part

- Length of the data part = whatever we specified for the length of the file size
- "While the amounts of bytes read is less than expected, keep reading data"
- Leave specification if not enough data received or too much received up to implementors
- On success, server sends back to client amount of bytes read, closes connection

In general: On any errors, Server will close connection.

Repeating message for file contents

- Client -> server
- 300 bytes
- Requires:

Client should not allow that files greater than 2^{64} bytes are sent; Server should be programmed defensively against this

In general, error checking on *both* the client and server side.

Initial Header to indicate file size: 8 Bytes in network (Big Endian) order

- Requires that OS supports file sizes of up to 2^{64} bytes

Indicate type of encoding

Message from Client to Server:

File Size Indicator: 64 Bits	File Name:	Data:
------------------------------	------------	-------

Message from Server to Client:

--

Questions:

- **How should we handle file names on the server side? A: Keep a list?**
- **How should we handle two transferred files having the same name? A: Over-write**
- **Should we honor client's sent file name on server side?: A: Yes**
-