BEL File Transfer Protocol

**Data Format**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ACK | NACK | EOF | RDY | ASK | FIN | U | U | File Length (4 Bytes) |
| Data (up to 1452 Bytes) | | | | | | | | |

We have implemented a 5-byte header. The first byte is for flags, which determine the type of message that is being sent. EOF stands for end of file; the RDY tells the server that the client is ready to receive data. ASK is a request for a file from the server, and there will be data in the data field that is the name of the file requested. U stands for unused bits. If needed, two more flags can be implemented without changing the header. We are limiting the TCP packet sizes to 1457 Bytes per segment, which is just shy of the 1500 max. Because our checksum is 32bit, the data load must be divisible by four. The file length is the length of the file in bytes. This limits the file size to 4GB. The file length works as a secondary check to make sure the file transferred correctly.

**Checksum**

We will be using a 32bit checksum lookup table which will be a constant in both applications.

**Closing the Connection**

When the client has received all of the files that it wants from the server, it will send one message with the FIN flag, and the server will close the connection.

**Data Transfer Diagrams**

File Not Found File Transfer Error File Transfer Successful

ASK = 1

ASK = 1

ASK = 1

Checksum

EOF = 1

Checksum

EOF = 1

Data

Data

ACK = 1

ACK = 1

NACK = 1

RDY = 1

RDY = 1

ACK = 1

NACK = 1