Protocol Design

The packet for this protocol is designed in the following way:

zzz

y

xxxx

aaaaaaaaaaaaaaaaaaaaaaaaaa

For this packet there are 4 different fields: sequence number, packet type, payload size, and payload.

The sequence number is used to order all the different data for one transaction. A transaction, in this situation, is the number of packets needed to transfer a full file across from either the client to the server or the server to the client. This is used to make sure all packets are put in the correct order by ordering them from 1, 2, 3, …, n; the n represents the largest packet needed to pass. For this system, it can allow a maximum of packets from 0 to hex ffff which is 164 = 65,536 packets. The 4 bytes required to do that are represented as the xxxx in the design.

The packet type represents the kind of operations we want to do. All packets used to do one operation use the same number and are restricted to one of 5 numbers. The 0 = get, 1 = put, 2 = ls, 3 = quit, and 4 = initiate the second call. This packet will determine the action for the server to know what job is needed to be done. Since it is restricted to only 0-4, only one byte is used. This is represented as the single y byte in the diagram.

The payload size just states how big the payload is for this packet. For instance, if the payload is ‘abc.txt’, then it should have the value of 7 in hexadecimal for that spot. Since we are restricting the payloads to be a maximum of 1024 bytes then we only need 3 bytes to represent 0-1023 which is 163 = 4,096 at max.

Finally, the payload is just the text being sent for the packet. The on transfers of file data, this will be the entire file information. But on setups of the connections, file names, quitting the connection, and message passing, the payload will have action information describing what the system should do. For instance, we may use the payload to tell the server we want to do a ‘get’ command or we may say ‘file.txt 40’ as the name of the file with the total size of the data. Since we have a max of 1,024 bytes/packet to pass and 65,536 packets that we could use, this would mean we can accommodate 210 \* 232 = 242 bytes of data or 4 TB of information to pass. Having said that it would take a long time to pass the information.