Kenny Crowell /kcrowe01

Yuxuan Zhang/yzhang17

Comp 150

File Copy

Design Document

The beginning of each file transfer will start with a packet that includes only the file name, as well as a flag that indicates the start of a file transfer. This will be followed by data packets (set with a data flag). These data packets will contain an offset number that will be multiplied by the amount of data sent in each packet to get the correct offset to which the data should be written in the file. The server will acknowledge all of these packets and if the client does not receive this acknowledgement it will resend the packet. The client will continue to send data packets for this file until every one of them has been acknowledged, at which point it will send an end-of-file packet. At this point the server will send back a check packet with the sha1 of the file and wait for a response from the client. If the client does not respond with success or failure then the server will resend this check packet. Once the server receives a success from the client it will remove the .tmp from the end of the file and send an acknowledgement back to the client saying it is ready for the next file.

Lost packets will be resent when the server/client does not receive an ack packet. Duplicate packets will be handled differently for each type of packet. Duplicate start packets will be recognized if any start packet is received while a file is still open or being written to. Duplicate data packets will simply rewrite the data if the fileNum field matches the current open fileNum and will be dropped otherwise. Duplicate ack packets will just be treated as non-duplicates because this will have no effect. Duplicate end-of-file packets should have no effect because the current fileNum being worked on will be different than that in the packet. Duplicate check packets can simply be reused because this will not change the success or failure of the file transfer. Reordered packets also should not matter as data packets have an offset for each piece of data to write.

Packet Structure

Start Packet

Char Flag (1 Byte)

This determines the packet type

Uint32 fileNum(4 Bytes)

Each new file transmission gets a new file number

Char *filename (max 255 bytes)

Max size for a filename in linux is 255 bytes

Data Packet

Char Flag (1 Byte)

Unsigned int32 fileNum (4 Bytes)

Unsigned int32 data offset (4 Bytes)

This offset is multiplied by 501 bytes to determine where the data goes in the file

File Data (max 501 Bytes)

The rest of this packet contains data from the file being transported

Ack Packet

Char Flag (1 Byte)

Char AckFlag (1 Byte)

This tells the receiver what type of packet is being acknowledged

Unsigned int32 fileNum (4 Bytes) (if start or data packet)

Unsigned int 32 data offset (4Bytes) (if data packet)

End-of-file packet

Char Flag (1 Byte)

Unsigned int32 fileNum (4 Bytes)

Check Packet

Char Flag (1 Byte)

Unsigned int32 fileNum(4 Bytes)

Char* Sha1 (40 Bytes)

This holds the Sha1 used for the end-to-end check