Jongwoo Kim (503-787-749)

Bryan Lau (404-015-053)

CS 118 Project 2 (Reliable Data Transfer Protocol)

Header format

We used the standard source port and destination port in the header, each as 16 bit unsigned integers. We added a sequence number and ack number field, both 32 bit unsigned integers to help with reliable data transfer. We also have a 16 bit unsigned integer length to tell us how much data is contained in the packet, as well as a 16 bit flag field, where we could customize each packet for its function.

The flags we chose to use were the standard SYN, SYNACK, FIN, and FINACK to start. We then added an ACK for ack'ing packets. For corruption, we opted to use a COR flag for simplicity, which would be set based on a percentage. Lastly, we are using an EOF and EOFACK flag to denote when a file is finished sending.

Messages

We start off a connection by firing a SYN message, which is responded by a SYNACK message. The SYN message sent by the client also contains the filename of the desired file (because of some troubles we encountered).

The sender then starts transmits regular data packets, which are ACK'ed by the recipient. When the last packet of the file is transmitted, it is also an EOF message to tell the recipient that we're done with sending packets. The client also sends an EOFACK message to ACK the fact that we're done.

Finally, FIN and FINACK messages are exchanged and the pseudo-connection is closed.

Timeouts and Corruptions

Timeout and corruption rates will be determined by the user when starting the sender executable. The default rate is 0%, and the parameter takes in integers between 0 and 100, which will be compared against with a random number generator to determine whether or not a packet is dropped or corrupted.

We are allowing up to 10 timeouts per packet. In the case that a packet or its ACK times out 10 times, both sides will eventually give up on the transmission. However, if the network complies and successfully sends a correct packet before 10 is reached, then the counter is reset and the transmission continues. The amount of timeouts and corrupted packets should not affect the overall integrity of the transmitted file, given that transmission is successful.

Protocols

We are using the Go-Back-N protocol for reliable data transfer. Upon a timeout, all windows are paused and retransmitted.