Gloria Chan 604284011

Daniel Yao 004182464

**CS118 Project 2 Report**

Implementation description:

Header format:

* Type: tracks the type of packet being transmitted
* Last packet flag: flag to check if a packet is the last data packet
* Sequence number
* ACK number
* Data

We implemented the Go-Back-N protocol.

Our client and server go through a hand-shaking phase; the client sends the server a SYN and the server responds with a SYN-ACK. The client can then send the file request to the server.

Once the server receives the file request and opens the file, it partitions the file data into individual packets with an appended header. These packets are queued and ready to be sent.

Our Go-Back-N protocol begins after. We initialize our window and send all of the packets inside the window. Each time an ACK is received that is greater than the base of our window, we slide the window to the new value. To deal with timeouts, we set a timer each time we are transmitting new packets from a new window (one timer per window). If no ACKs are received, we will timeout (timeout time is set to 2 seconds) and the whole current window gets resent again.

When the client receives the last packet, it sends the server a FIN. If this FIN is lost by the server, then it will retransmit the last packet a maximum of 5 times. After the fifth retransmission, the server will end the connection and continue to wait for new connections.

Difficulties:

* Problems with null bytes when copying data: solved by directly writing into the packet data buffer
* Reading packet’s data, due to incorrectly set up headers
* Designing the header format/struct used to represent packets
* Closing connections (when server loses the last ACK/FIN): solved by setting up a max number of retransmissions allowed
* In general, debugging