Marko Sevo, 704833042

Shangyu Hsu, 204784850

You run the server and client just as the project description says to.

**Implementation Description**

**Header format**: the header consists of 3 uint16\_t’s representing sequence number, acknowledgement number, and the receive window size, as well as one char used to represent the type of packet it is. In the Constants.h file, you can see the different types a char can be, including SYN, SYNACK, ACK, FIN; this is a slightly different implementation than having single bits represent these values as flags, but takes up the same amount of space (8 bytes) and leads to the same functionality.

The implementation hits all of the major aspects of TCP. There is initially a three way handshake, on which the final ACK there is piggybacked the name of the file requested by the client. Once the server obtains the packet with this filename, it opens the file and divides it into packets depending on how large it is. Then, depending on the receive window size and congestion window size (whichever is smaller) it begins sending these packets, moving the sender window as the sent packets get ack’d. Using the values given in the project description, the timeout to trigger packet retransmission is 500ms. Fast retransmit is also enabled on the server’s side, which keeps track of duplicate acks sent from the client and retransmits the specific file packet when three are received.

The **extra credit** was done, implementing the TCP congestion control via slow start and congestion avoidance. This implementation changes the values of the congestion window and ssthresh accordingly as per the TCP implementation described by the textbook.

There were many difficulties faced during this project, but simply building incrementally and making sure the foundation was strong before moving onto the next functionality was the most helpful tool in getting this done. Adding extra information such as any window sizes, packet buffer contents, ack and sequence numbers, etc in the output allowed me to see exactly how the client and server were interacting, which packets were being sent, and how they were received. Whenever I ran into a problem, the biggest help would be paying cautious attention to the details and going through exactly how the programs would interact in any given scenario.

The biggest difficulty of this project came when implementing the reliability aspect, and in what order and way to send packets, keep track of if they were acknowledged, how to make sure the timeouts were correctly getting called, etc. Again, there the textbook was the biggest resource in understanding the theory of how TCP works, so that helped me structure the send and receive buffers, how I would process/read and output data to the file.