Dylan Brown
CSE 3461
200148626
Lab 2 Readme

Compilation:

Simply compile the program with the "**make**" command in the terminal.

Running the program:

Sender:

1. CD into the directory for the lab
2. Run **./sender PORTNO PROB**
3. Make sure PORTNO corresponds to a port number, like 5447
4. Make sure PROB is a value between 0 and 1, like .4
    a. Corresponds to the probability of the DATA packet being sent

Receiver:

1. CD into the directory for the lab
2. Run **./receiver localhost PORTNO filename PROB**
3. For PORTNO, make sure it's the same as sender, like 5447
4. Make sure filename exists in the directory, like "largeimage.jpg"
5. Make sure PROB is a value between 0 and 1, like .4
    a. Corresponds to the probability of the ACK packet being sent

This server was initially built as a UDP FTP server, capable of transferring files between the sender and the receiver. The receiver would initially request a file, the sender would open up the file, and send it to the receiver within 1KB packets, with a 7 byte header containing information like File Size, Sequence Number, Data or ACK type of packet, and the maximum Sequence Number. There are header files linking both the sender and receiver to another corresponding C file, containing all the functions written for packet delivery. The timeout is 1 second, which is relatively lengthy. The timeout is set via setsockopt. Other options explored were using a current time, from time.h, and a start time. And when current time - start time > timeout, the kill the while loop. However, setsockopt was a simpler implementation.

The biggest issues with this program were trying to successfully convert the TCP server from prior to a UDP server. Along with that, transferring data in packets with header information, and correctly parsing the header. Another major issue was getting the timeout to work correctly, and recognizing a duplicate packet.