Max Gasser
CSE 156
ASGN 2 Design Document

Design:

myclient.c

The client function first parses the arguments from the command line and stores them for later use. It then checks if certain arguments are valid such as MTU. Then the client socket is created using the command line arguments. Both files are opened/created then logic for sending datagrams is begun. The client then initializes a few values for data transmission and then sets up alarm functionality to account for transmission time outs. Then in a while loop the input file then has mtu - header_size bytes read from it. Those bytes are then sent to the echo server using sendto, after sending the datagram, the client then calls recvfrom to then receive back the bytes that were sent. Those bytes are then written to the output file. This process is repeated until the entire file has been sent and received back from the server. Once done the program terminates. To counteract the client from waiting infinitely, before each recvfrom and sendto call the client will wait a maximum of 60 seconds before timing out and exiting.

Myserver.c

The server functionally first passes the command line arguments and stores the port. A socket is then created and to be used to receive datagrams. Once the socket has been created the server then sits in an infinite while loop waiting for clients to send it packets. Once a packet is received, the server simply sends the data right back to the client who originally sent the data.

Splitting data into packets:

The method I used to split data into packets was by using fgets to read upto a maximum of mtu - header_size bytes from the file into a buffer. Therefore each datagram would always be smaller than the mtu.

Shortcomings:

The client does not have functionality for reordering packets on arrival. Throughout my own testing I have not ever had packets arrive in the incorrect order however I understand that it is entirely possible. Unfortunately, I wasn't able to figure out a method to do this.

Testing:

The following commands were ran to test my programs

```
- ./myclient 127.0.0.1 8888 25 testfiles/longfile.txt output.txt
- ./myclient 127.0.0.1 8888 1500 testfiles/hello.txt output.txt
- ./myclient 52.43.121.77 10001 1500 ./testfiles/hello.txt output.txt
- ./myclient 127.0.0.1 8080 22 testfiles/longfile.txt output2.txt and ./myclient
127.0.0.1 8080 1500 ./testfiles/hello.txt output1.txt ran simultaneously
- ./myclient 52.43.121.77 10001 24 ./testfiles/longfile.txt output.txt
```