# Data Communication Networks – Program 1

## Jason McQueen

## 2/17/2019

### Documentation

The program works as a client-server system that allows the client to transfer the contents of a text file to a server that then outputs that text into an output text file.

First, the server generates a random port number in the range of 1024-65535. Then the server creates a TCP socket, binds it, and listens for any connections. The client also creates a TCP socket, and connects to the server. Once the connection with the client side is acquired, it waits for negotiation to begin by the client sending "259" to the server. Once the server receives the message, the server sends back a reply with the randomly generated port number. Both the client and the server then close their TCP sockets. The server and client then create UDP sockets. The server then binds the UDP socket to the random port and waits for the client to start transferring the text in an infinite loop. The client starts an infinite while loop, and starts sending 4-bit packets to the server, by chopping up the text file into a 4 bit buffer and sending the buffer every 4 characters. An if statement checks for an EOF character, and once it finds it, sends an end message to the server and closes its socket. The server during all of this grabs the 4 bit messages and appends them to a large buffer. After every 4-bit message, the server sends an ACK in the form of the last four characters it received, but uppercased. Once the server receives the end message from the client, it breaks out of its infinite loop and closes its socket.

Compilers:

gcc compiler on Ubuntu, unix environment

Development and testing:

Program was heavily tested on multiple different sample files and TCP port numbers. The program as well has commented out print statements that checked return values and other relevant variables.

Sample commands that were used:

```
./server 8080
```

```
./client 8080 sample.txt
```