Sadat Islam
UCID: 10171897

CPSC 441 Assignment 2 – Octoput Transfer Protocol

How to Setup and Compile

1. Setting up the Server
    a. Make sure your server has the udp_server.c file as well as the text files provided on Dr. Williamsons website. These text files should be in the same folder as udp_server.c.
    b. Compile with gcc – std=gnu99 – pthread udp_server.c -o Server
    c. Run the server with ./Server
2. Setting up the Client
    a. On the client side, you just need the file udp_client.c
    b. Compile with gcc –std=gnu99 udp_client.c -o Client
    c. Run with ./Client
    d. Note: Make sure the Server is on before you start the client

Running my Project

1. Start the Server, then the client.
2. On the client you will be prompted to select a filename from the options given. Choose one of these files by typing it into the console and hitting enter.
3. The file will now be transferred to the same folder as the client. It is written in the new file called new_file.txt.

Design Choices

To add some reliability, I incorporated the use of acks, timeouts, and retransmission. After the server is notified of the filename, it will continue to push octablocks to the client. The client will return a ack once it receives the full octablock in the correct sequence. The client is expecting that the same octablock will be transmitted until it has all the octalegs in the correct order. If the client notices that it has gotten an octaleg that is from the previous octablock, it will send an ack notifying the server that it has received that segment, and to send the next octablock.

The server, after sending all 8 octalegs of an octablock, will wait for the ack that will come from the client. If the server receives the ack, it will send the next octablock if there is any remaining. If it does not receive an ack within a few seconds, it will retransmit the octablock. If after 4 retransmissions it still doesn't receive an ack, it will assume the client disconnected and wait for a new client.

UDP isn't the most reliable, and if all the acks sent from the client are lost, the server will disconnect. UDP should be used where the order of data isn't necessarily important. If order matters, TCP should be implemented instead.

Testing

My program was tested on my home computer on my home network as well as the university network. I manually turned off acks from the client side to test if the server would resend octablocks (as well as timeout after 4 failed retransmissions). I also manually discarded some octalegs so that they arrived in the incorrect order to test if the client handled this situation properly. I could successfully transmit all the test files except for the bonus test file.