Jonathan Woong
804205763
CS 118
Project 2

NOTE TO READER: I have some code for timeout that I believe should work, but for some reason does not. I've been trying to find the source of the bug but no luck.

High level design of client:

Functions:

1. createSocket()
2. bindSocket()
3. getHostData()
4. sendPacket(packet)
5. performHandshake(): continually send SYN until receive SYN-ACK, then send ACK and open file to be written to.
6. handleData(packet): extract sequence number and payload size, write packet payload into file, send ACK
7. closeConnection(finPacket): extract sequence number and acknowledgment number, send FIN-ACK, send FIN, close connection and file

Algorithm:

1. Parse command line arguments
2. Set up client (create socket, bind socket, set address/port)
3. Wait for incoming packet
4. When receive packet, determine packet type
5. Handle packet
6. When receive FIN, close connection

High level design of server:

Functions:

1. createSocket()
2. bindSocket()
3. getClientData()
4. sendPacket(packet)
5. beginHandshake(): send SYN-ACK
6. completeHandshake(synAckPacket): set completedHandshake flag to true
7. divideFileIntoPackets(filename): open file for reading, calculate file size, calculate number of packets needed to send to client, divide file into payload-size chunks and fill vector with packets, close file
8. processDataAck(packet): update necessary integer trackers, check if done sending file, increase CWND based on Slow Start
9. choosePacket(): returns index of next packet to be sent
10. attemptSendData(): based on current sender window, send as many packets as allowable
11. beginClosingConnection(): send FIN
12. finishClosingConnection(): send FIN-ACK and close socket

13. getCurrentTime()

14. checkForTimeOut(): checks for timeout, if found, resent packet

15. linkSeqToIndex(): maps sequence numbers to index of packet

Algorithm:

1. Parse command line arguments

2. Set up server (create socket, bind socket, set address/port)

3. Wait for incoming packet

4. Check for timeout

5. When receive packet, determine packet type

6. Handle packet

7. When done sending file, close connection

Problems your ran into and how you solved the problems:

1. I had an issue where received packet on the client size was too always 30 bytes too big. I resolved the issue by completely rewriting my entire client code with more abstraction.

2. My implementation of timeout currently does not work, though I feel like my code shows an algorithm that can work.

Additional instructions to build your project (if your project uses some other libraries);

1. I hardcoded the server address to 10.0.0.1 for ease of testing purposes.

2. Use Vagrant

How you tested your code and why:

1. Write entire code without abstraction (one large chunk of low-level code) and test that it works. Then refactor chunks of code into functions (abstraction) and test that it works. This makes finding bugs easier.

2. I tested only by sending small.txt first. When that test succeeded every time, I moved onto large.txt. I did it this way because Professor Afanasyev said his tests will use these two files.

Contribution of each team member (up to 3 members in one team) and their UID:

1. I worked by myself):