Lab 2: Simple Window-based Reliable Data Transfer in C/C++

Brendan Sio (SEASNET id: brendans) 804150223
Roland Zeng (SEASNET id: roland) 204150508

High Level Description

Going into this project, we found that there is a lot more we need to learn in order to be able to tackle this lab. Here’s a high level description of how we implemented it:

We divided the whole lab into four main parts as follows:

1. implement connection assuming no corruption or packet loss
    a. sender sends a packet, receiver sends an ack
        i. client sends request packet to server
        ii. server finds file
        iii. server splits it into packets
        iv. when server sends packets to client, client sends back ack
        v. when server is done, send fin packet
2. Implement Go Back N algorithm
    a. create windows
3. then introduce packet loss
    a. percentage of packets or random # of packets are lost
4. add a timer on the sender side

We utilized part of our code from lab 1 to first send a request from the client to server. After we succeeded, we proceeded divide the file into packets, and send one packet back to the client. Our packet contains an id for the sequence number, a type that identifies what kind of packet it is: request, data, or ack, a size that indicates how big the packet is, and a char data that indicates the maximum size per packet.

The input to our console from the client side is ./client *filename (packet loss probability Pi) (packet corruption probability Pc)*

The input to our console from the server side is ./server *(window size) (packet loss probability Pi) (packet corruption probability Pc)*

Our port number is defined in the code

The received file on the client side will be stored in CopiedFile.txt

Difficulties

1. We didn’t know where to start the project. We brainstormed a lot before we actually sat down and wrote code.
2. After we finish sending packets with the window size 1, it was hard for us to change our code to make it variable window size. To solve this, we decided to restart and implement the window size first before we send the first packet.

How to Run

1. Open the submitted .zip or clone from our Github repo, https://github.com/rzeng95/CS118Lab1
2. add files that you wish to send to the same directory. We added ‘hello.txt’ as an example.
3. Open 2 console, and type ./client hello.txt 0.6 0.6 (this means packet loss and corruption probability of both 0.6) in the first one. Type ./server 1 0.6 0.6 in the second console, which has the window size of 1.
4. Press enter and you should see hello.txt sent to CopiedFile.txt