CS 118 Project 2 - Report

Simple Window-based Reliable Data Transfer

Winter 2018

Reinaldo Daniswara, Tianyang Zhang

604840665, 404743024

Implementation Description

We created our own structure named packet_info, we put all of the flags we need inside the structure, and within each packet, there are 972 bytes of data, and the rest of 52 bytes are all header informations. The header informations includes: type of packet, sequence number, maximum packet number, fin number, error flag, timestamp, and size of data.

The sending and receiving part of server is in a while-loop. In each round of while-loop, server would send out one packet. Each message sent from the server would leave a timestamp in an array called “time_table”, server will record the time that packet sent out, then, in each loop, server will check if any packet inside current timetable is timed out. The time table has 5 slots, which is same as the window size, whenever the first slot of the timetable is ACKed by client, server would copy the timetable to the right, and free the last slot to next packet in current window.

The retransmission mechanism is based on the timetable, when current time minus the sent time is greater than limited waiting time, server will retransmit the packet to client.

After the transmission done, server will first check if the last 5 packet ACKed, if not, the server will retransmit until all packets are ACKed by client. Then, server will send FIN packet to client, and wait for the FIN_ACK from client. Client will send out FIN_ACK at most 4 times, server closes after FIN_ACK is received.

Difficulties

First, there is no linux environment available that provided by school to test the project. As a result, we need to do the implementation on virtual machine.

Doing the implementation in another virtual machine is quite time consuming. Even though it is not super complicated, but it still take a lot of time to figure out. For example, we tried to install the ubuntu virtual machine on Tianyang Zhang’s mac. However, the vm does not work for the first couple tries. Later on, once it works, it works so slow. Then, we tried on Tianyang Zhang’s windows computer, which also somehow does not work. Until eventually, we use Reinaldo’s mac, then at least it run faster, but still not smooth and slow.

Another difficulties that we have is while we debugging our code. We tried to debug on Xcode, mac terminal using gdb, and it all works perfectly fine. Our project could run smoothly. However, once we use vm ubuntu, it does not always work. Furthermore, once we tried to debug it on ubuntu vm, it works all the time, and perfectly fine. To solve this, we may use gdb for the demo because we don't have much time to work on it anymore.

Other than the environment difficulties, we feel that the projects quite challenging, yet fun. At first, we have difficulties to record the packet loss. We finally able to figure out using time table, and find the timestamp to record the packet loss.

At first we also have some problems in our code which resulting the speed of transmission is so slow. We finally figured out that there is some mistakes that make it retransmit same packets for multiple times.

We also have some offset problem at first, where after the transmission done, we received the image that really look alike with a bare eye. However, it we check using diff command, it turns out that near the end of transmission, it lost a couple bytes. Thus, after we figured out this problem, now all the packets that send are exactly same as the original.

In addition, there is also a problem when closing the connection for the server side. The problem could occurs if the server does not receive the FIN SYN and FIN ACK signal at the end. The only implementation to attack those are create another while loop to make sure those FIN are received. However, there is going to be an infinite while loop. Thus, we force them to close the connection after a 4 tries.

Suggestions

We feels like this project is going to be more fun if we can test it on seasnet Linux server. In addition to that, if this project provides a test script to test out project, it will be much more fun and interesting.