CSCI551 Lab3 Report1

Zheyu Shen

November 2019

1 Program Structure and Design

In stop and window version, I add several attributes to maintain the seqno, ackno, retrasmition information, the last send segment, the last received segment and fin information. What's more, I add three help function to detect corrupted segment, send ack segment, fin segment and data segment. In ctcp_read function, I mainly deal with three situation, which is empty data, EOF and normal data. In ctcp_receive function, I mainly deal with five situations, which is corrupted segment, duplicated segment, ACK segment, FIN segment and normal data segment. And in ctcp_output function, only output the last received segment and send ACK for it. In ctcp_timer function, I mainly deal with retransmition and whether to tear down.

In sliding window version, I add two linked list to maintain unack segments and unoutput segments. And I also add one more help function to caculate the data len in air. And the other function maintain the same design with the previous one. The only different is to change the last send segment to unack linked list and change the last receive segment to unoutput linked list.

2 Implementation

In stop and window version, it is not so hard to implement. But it gives me much understand of how cTCP works and how to set the headers.

In sliding window version, the most trick thing is that I assume segment in unack linked list and unoutput linked list is sorted by seqno. Therefore, when receive ack segment, I can remove segment in unack linked list from front. But I have to insert received segment in order, which is the most complecated part. I have to consider four kinds of situation, which is unoutput is empty, insert segment in front, insert in middle and insert in tail.

3 Testing

In ctcp_tests.py, my ctcp cannot pass the sliding window test and different send/receive window test. And my ctcp could pass all other test in ctcp_tests.py, Dumbbell topology and Mini-Internet topology.

However, if I use sliding window version ctcp to test, it can pass all test in ctcp_tests.py without bidirectional transfer, even if I can transfer massage manually, And it will fail in Dumbbell topology and Mini-Internet topology test.

4 Remaining Bugs

Submitted version is stop and wait only, because sliding window version can transmit large binary file with 10% delay and 10% dropout manually and correctly, but cannot pass the test2 and test3. I will try to find the reason after I submit.