Project 2 Report

In this project, I was asked to implement a Reliable Data Transfer protocol.

To implement the Go-back-n protocol, below are some experiments I set up to approach it.

- Experiment 1: change the size of the array to be sent in TestClient to be 1000. In this case the MSS(Max segment size) is 100 bytes, so that the first group data will be divided to 10 segments.
    - Objective: to find out whether or not I can successfully divide data into segments since the default segment size in RDTSegment class is 100 bytes. (the default size of data in TestClient is only 10 bytes so we cannot divide them into segments)
- Experiment 2: Change the lossRate in both TestClient and TestServer to be 0.
    - Objective: Simplify the function of program, and make it easier to observe the ack number and the sequence number.
    - It helps me to successfully implement some basic functions of data transfer: send and receive data packets and acks.
- Experiment 3: Use various packet loss rates
    - Objective: To test the function of timeout scheduling and retransmission.
    - It helps to do the further implementation of GBN protocol.
- Experiment 6: Use various retransmission timeout values to test the timeout scheduling. (failed to set up since I failed to implement the timeout scheduling).
- Experiment 7: Use different computers and different port numbers.

In this project, I followed the instructions in the send function, but I failed to implement the last part: timeout scheduling, and everything stops there, I tried many ways to achieve that, for example, I tried to completely remove TimeoutHandler part and do not use thread to count timeout, I used setSoTimeout() function in Java; also I tried to use threads to count timeout instead of using Timertask class in Java, but all that failed. However, this project provides me with a great way to understand how Go-back-n works, which make me gained a lot. For example, I learned how to develop in Java(this is the first time I did a project in Java); I learned how to use IntelliJ IDE to debug and to configure; I learned to move the GBN window to the right by changing the value of base. The most interesting part and also the hardest part I found is the multithreading in Java(I still do not

how to implement timeout scheduling in thread though). The interesting part is that the run() function in the TimeoutHandler class actually need not to be called in send function. And if you called run() function directly, the run() would go onto the current call stack instead of a new call stack. I tried to use the variable timer which is a member variable in RDT, by calling timer.scheduleAtFixedRate() function to executes the TimeoutHandler thread every RTO msecs to retransmit if ack is not yet received. But it does not work well until one hour before the due time.

(you can ignore the paragraph below)
(To be honest I spent like over 40 hours on this project, it is an interesting project actually, I just realized to write the report when there is only 1 hour left before the due time...I was still trying to debug..
I'm pretty sure I can successfully implement the complete GBN protocol but time does not allow me to do so...
Finally I do learn a lot from this project although I did not finish the requirements of the project.
Thank you so much for spending time on this report. It's really a shame that I did not finish the project on time. I'm pretty sure I will spend more time on my next project :)
)