Arjun D'Cunha
CS14BTECH11039
27 March 2017

Assignment 3

Implementation of a Reliable Application Layer Protocol

The purpose of this assignment was to implement two different types of Reliable Application Layer Protocols

i) Stop and Wait Protocol : The server sends one packet at a time, and then waits for a response ack from the client. Until it gets a response or timeout, it does not send another packet. This reduces the efficiency of the server a lot.

ii) Go Back N Protocol : The server rapidly sends multiple packets on the basis of an inputted window size. and then wait for the respective acks from the client. In the case a timeout occurs, it sends all the packets from the unrelieved ack.

Design :

In both the cases, I followed the given Finite State Machines given in the book.

I created my own type of data and ack packets and used my own type of checksum value.

Each packet has a sequence number and data and the required checksum value calculated using an appropriate function.

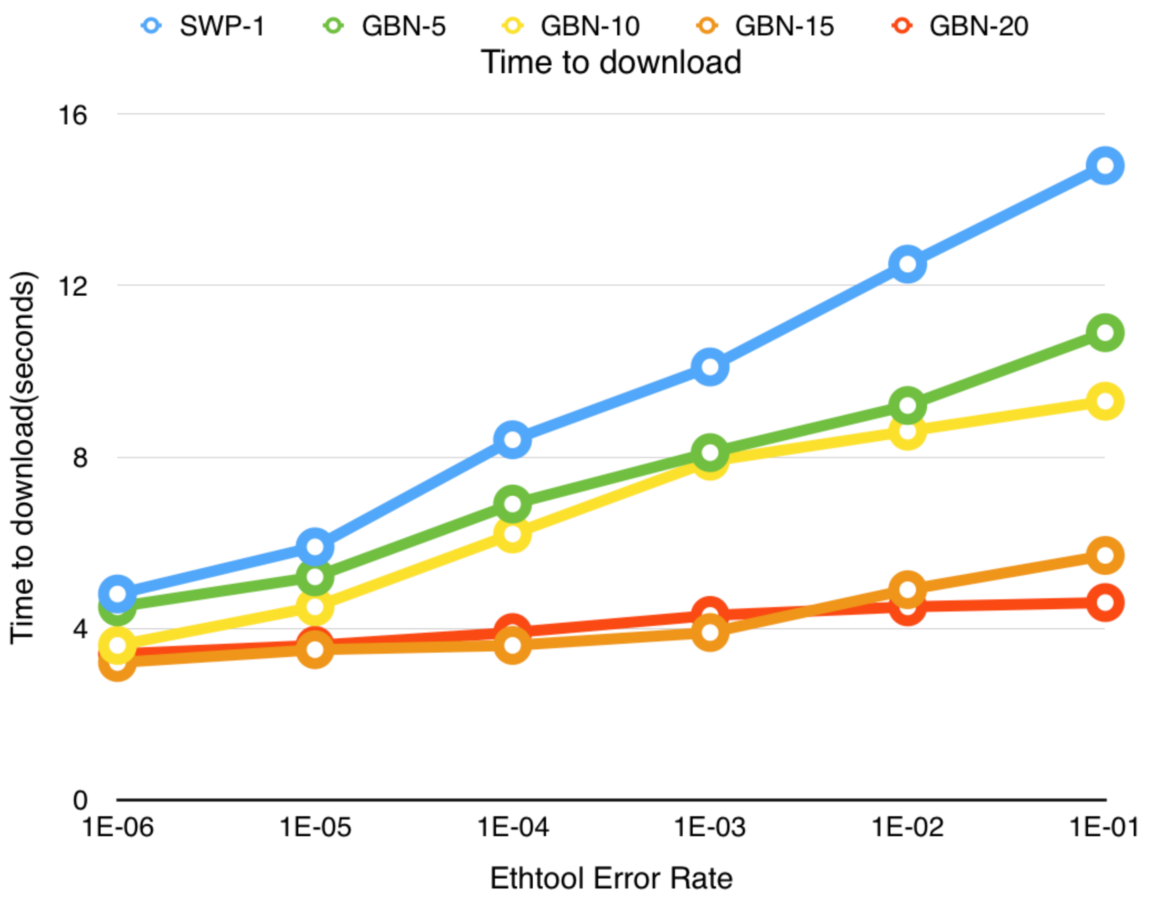
The tester needs to input the correct port numbers for the server and the client and the sample file to simulate the download.

All the codes submitted have been coded and tested on MAC OS El-Capitan.

As ethtool does not exist for MAC-OS, I used my friends linux system to run the required tests.

All performance was tested on a Linux machine with Linux Ethtool Tester to simulate the required environment.

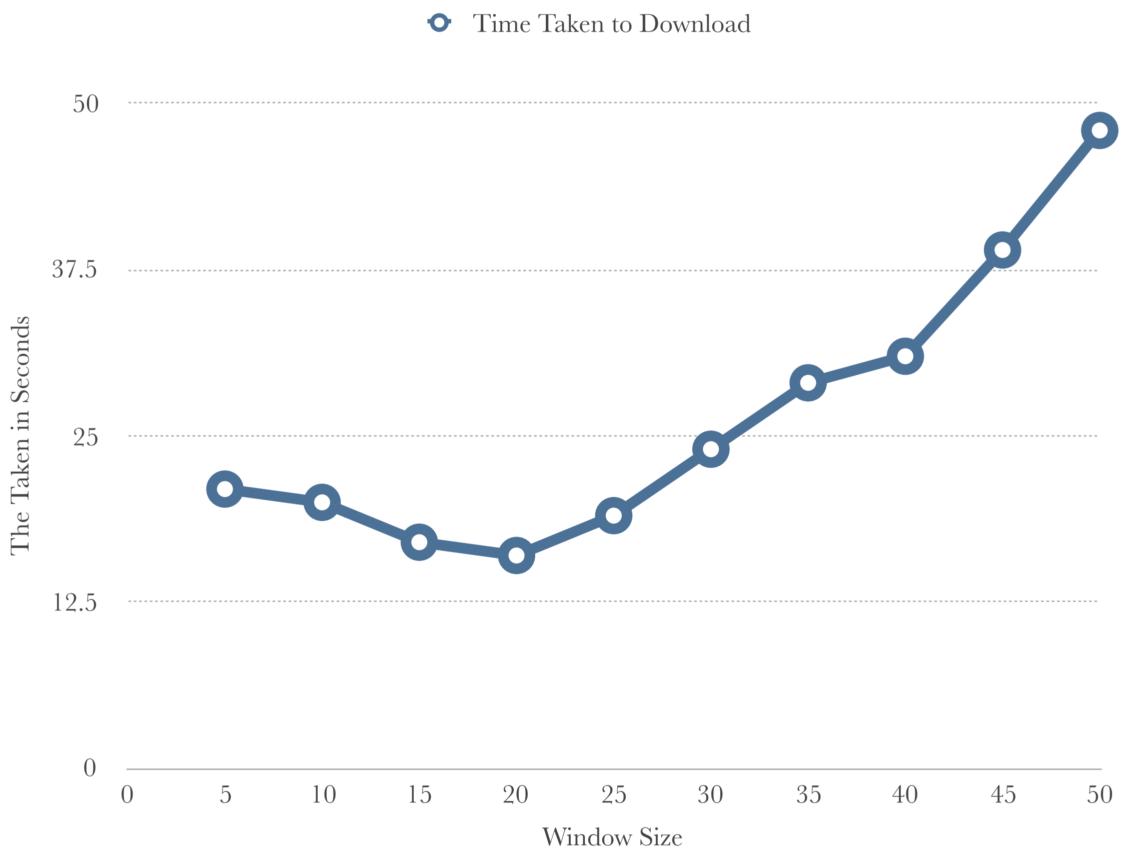
Performance Evaluation a)



Conclusion:

We can observer that as the window size keeps increasing, the time taken to simulate the download reduces considerably. However, varying the error rate on large window sizes does not affect the download time to a large extent.

Performance Evaluation c)



Conclusion:

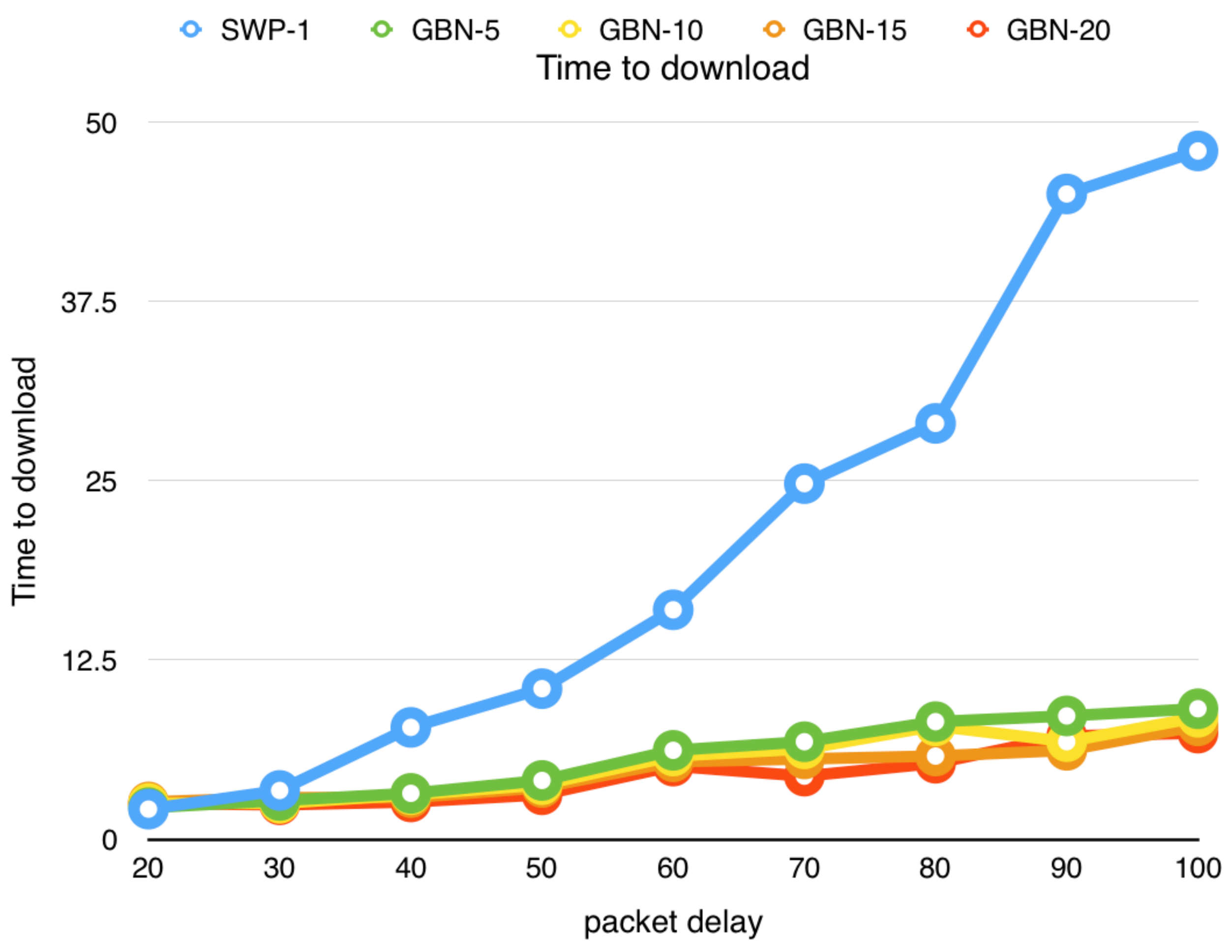
We see that the time to download first decreases and then increases.

This is because in a smaller window, very few packets are sent and a lot of time is wasted due to timeout retransmissions. In addition, having larger windows create problems where if a packet is lost, a lot of time is wasted in realising it and then retransmitting the packets. Hence an average window size is perfect.

Loss Probability:

As the probability of the loss of a packet increases, due to the Go Back N protocol, the average transfer time also increases.This is because for every increase in error rate, the number of packets to be retransmitted also increases, thus increasing the average transfer time.

Performance Evaluation b)



Initially, increasing the window size gets somewhat better performance, however, as the window size keeps increasing the performance does not differ too much. But, in the case of SWP, for a larger packet delay, the time taken to download increases a lot since we need to wait for each packet's ack before sending the next packet. This is not affect much later on since the rtt calculation was moving based on last rtt found.