CSE489 PA2

Dakota Lester, Samuel Gura

I have read and understood the course academic integrity policy

-sggura

-dakotale

SERVERS USED

embankment.cse.buffalo.edu

highgate.cse.buffalo.edu

Introduction

The *Reliable Transport Protocols* assignment and report is intended to have students develop a functioning transport layer using three different types of protocols. After developing the transport layer, the protocols developed are tested in a series of simulations to compare the efficiency of each of the protocols by measuring throughput. The three protocols under testing are:

- Alternating Bit Protocol
- Go-Back-N Protocol
- Selective Repeat Protocol

The simulator to test the efficiency of each protocol will determine the data to be passed through the simulated network and determine the reliability of the medium from two different hosts. The simulated medium will vary the amount of loss and corruption of data (ranging from no loss/corruption to complete loss/corruption) under our control. The ranging values we will use for testing each of our protocols are as follows:

Included in this report is the method of implementing the timing for each of the transport layer protocols(including an in-depth explanation of the timing creation for Selective Repeat).

Timing Schemes

For each of the protocols, we developed a timing scheme to determine how long was too long for A_input() to not receive an acknowledgement from B_input(). For Alternating-Bit Protocol a constant time was used to determine the timeout. Upon testing different times, the most efficient time for the timeout was 25.0 time units. This number was reached by finding the lowest time that would not cause a premature timeout and result in duplicate packets being sent from A_timerinterrupt(). The Go-Back-N protocol used a similar method for determining the most efficient time before a timeout. Once again, different values were used to test which amount of time is best. For Go-Back-N protocol, the most efficient time was 25.0 time units. This time was the lowest time to not cause a premature timeout by A_timerinterrupt(). With the case of Selective Repeat protocol this timeout factor is much harder to think of. In our case, we did the following to address the issue of having one hardware timer and needing multiple software timers. First, we starting with declaring a vector of floats that would be able to store the values of 'get_sim_time()'. What we have is immediately after the first packet is sent, (or n) the sim time is pushed onto our vector. For our implementation, a timer interrupt would be used to mark that the first successful packet has been received. With this result, every time the timer interrupt is called a 'for' loop is run with a simple (int i = 0; i < size of the float vector; i++) and within that loop a timeout is found be either true or false. If the timeout is less than or equal to the value of our designed timeout (which in our case is RTT of 25.0) then the packet has timed out in our case and then it will be resent.