CS3205 Computer Networks Lab
Assignment 2:TCP Congestion Control

Bharath CS18B009

March 29,2021

Instructor: Prof. Siva Rama Murthy

1 Objective

The objective of this project is to emulate the TCP congestion control algorithm.

2 Introduction

Congestion is a state occuring in the network layer when the message traffic is too heavy that it slow down the network response time.Due to congestion delay occurs which causes retransmission. Here in this experiment we use an algorithm for controlling congestion.

3 Experimental Details

3.1 Experimental/Simulation Setup

• Receiver Window Size is set to 1 MB, and does not change during the entire duration of the emulation.
• The Sender always has data to send to the receiver.
• Sender's MSS is 1 KB.Each segment has a fixed length of one MSS.
• GobackN is used, but cumulative acknowledgments are not considered.For each segment, an individual timeout timer and ACK are used.
• The congestion window is always intrepreted as a multiple of MSS (1 KB).

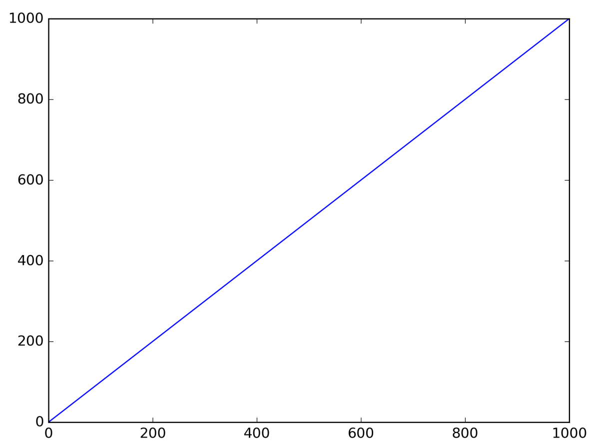
3.2 Entities Involved and Functions in each Entity

• The congestion threshold is always set 0.5 times the current CW value.
• k_i, $1k_i4$denotes the initial congestion window (CW). Default value is 1. The initial CW is given by:
$CW_{new} = k_i * MSS$

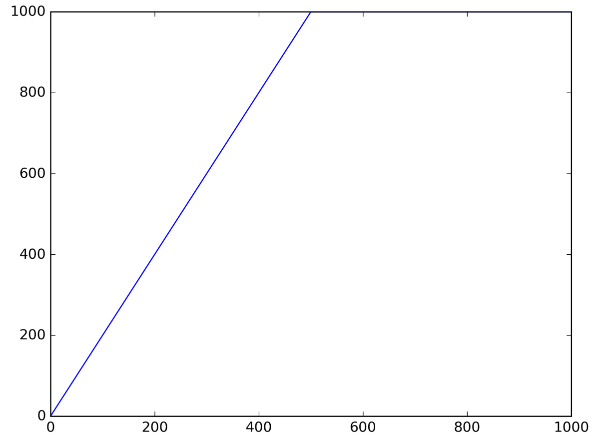
• k_m, $0.5K_m2$ denotes the multiplier of Congestion Window, during exponential growth phase. Default value is 1. When a segment's ACK is successfully received,
$CW_{new} = min(CW_{old} + k_m * MSS, RWS)$
• k_n, $0.5K_n2$ denotes the multiplier of Congestion Window, during linear growth phase. Default value is 1. When a segment's ACK is successfully received, $CW_{new} = min(CW_{old} + (k_n * MSS * MSS/CW_{old}), RWS)$
• k_f, $0.1K_f0.5$ denotes the multiplier when a timeout occurs:
$CW_{new} = max(1, k_f * CW_old)$
• p_s, $0 < Ps < 1$, denotes the probability of not receiving the ACK packet for a given segment before its timeout occurs.

4 Results and Observations

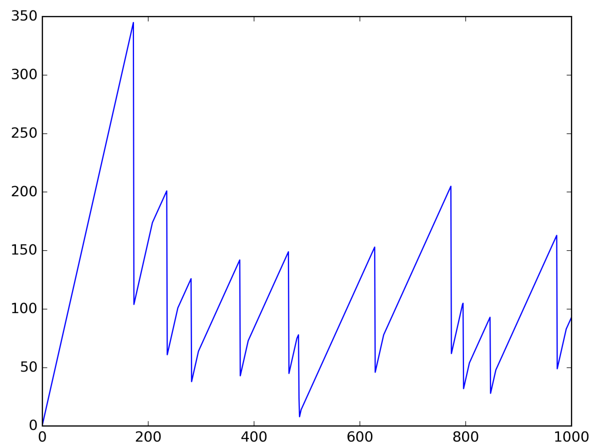
After we ran the code we obtained 32 different graphs with 32 different inputs. Here are some cases :



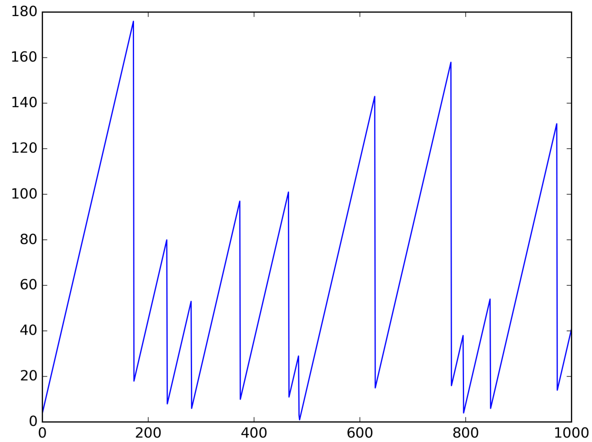
Here We have a linear graph which means there is no time out .



Here also the graph is linear until the value reaches 1000 afterwards it will be horizontal.



Here in this graph we have multiple timeouts and the graph varies as shown above.There will be slight change in slopes also after some point.



Here in this graph we have multiple timeouts and the graph varies as shown above.Here there is no change in slope for any line even if they crossed the threshold values.

5 Learnings

Congestion causes packets to be dropped on the network due to buffer overflow,therefore leads to data loss and unreliable connection.So we learned that effective congestion control is an important issue in transport layer and why we should use congestion control methods.I understood that how algorithm controls congestion.

6 Conclusions

Here from graph 1,2 we can conclude that if the p_s is low i.e. if the probability of receiving the ACK packet for a given segment before its timeout occurs is low then there will be less chances of timeout so it will be a linear graph until it reaches max values afterwards it will continue to be the max value.
In graph 3 we have multiple peaks,ups and downs here the timeout ocucred frequently so if its more then the graph will have such ups and downs.And after it reaches threshold the slope changes because of difference in equations and we approx them to an integer.
In graph 4 it will be same as graph3 but even after threshold arrives slope does not change beacause the difference will be less and we round them to integers.

References