

CS3205: Introduction to Computer Networks

Assignment 3 [Deadline: 27th April, 2022]

Total Marks: 100 [Weightage: 10%]

Simulation of the TCP Congestion Control Algorithm

The objective of this project is to emulate the TCP congestion control algorithm. The assumptions and variables are as follows:

§ Receiver Window Size (RWS) is set to 1MB 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 1KB. Each segment has a fixed length of one MSS.

§ Go-back-N is used, but cumulative acknowledgments are not considered. For each segment, an individual timeout timer and ACK are used.

§ The congestion window is always interpreted as a multiple of MSS (1 KB).

§ The congestion threshold is always set to 50% of the current CW value. Assume the initial threshold to be RWS/2.

K_i , $1 \leq K_i \leq 4$ denotes the initial congestion window (CW). Default value is 1. The initial CW is given by:

$$CW_{\text{new}} = K_i * MSS$$

K_m , $0.5 \leq K_m \leq 2$ denotes the multiplier of the Congestion Window, during the exponential growth phase. Default value is 1. When a segment's ACK is successfully received,

$$CW_{\text{new}} = \min(CW_{\text{old}} + K_m * MSS, RWS)$$

K_n , $0.5 \leq K_n \leq 2$ denotes the multiplier of the Congestion Window, during the linear growth phase. Default value is 1. When a segment's ACK is successfully received,

$$CW_{\text{new}} = \min(CW_{\text{old}} + K_n * \frac{MSS * MSS}{CW_{\text{old}}}, RWS)$$

K_f , $0.1 \leq K_f \leq 0.5$ denotes the multiplier when a timeout occurs:

$$CW_{\text{new}} = \max(1, K_f * CW_{\text{old}})$$

P_s , $0 < P_s < 1$, denotes the probability of receiving the ACK packet for a given segment before its timeout occurs.

Running the program: The program is invoked with the following command-line parameters:

```
$. /mytcp -i <double> -m <double> -n <double> -f <double> -s  
<double> -T <int> -o outfile
```

The values correspond to K_i , K_m , K_n , K_f , P_s and the total number of segments to be sent before the emulation stops. The output (specified below) is saved in an output file. The congestion window progression is done on a slot-by-slot basis. In each "round", a set of segments are sent, in proportion to the current value of CW, i.e. $N = \text{ceiling}(CW/MSS)$. For example, if CW is equal to 4.3KB, five packets are sent. However, the CW growth is based on MSS values as explained earlier. For each segment transmitted, the ACK for this segment is received before timeout with random probability P_s . Depending on this outcome, the CW increases and decreases as described earlier.

Expected Output: Given the set of input parameters, the simulation progresses as above. The congestion window value is printed to the output file (one per line) at each CW update. A graph with x-axis being the update number and y-axis the corresponding CW value must be plotted. A technical report must be written based on the results and graphs obtained for the following parameter combinations:

$$K_i \in \{1, 4\}, K_m \in \{1, 1.5\}, K_n \in \{0.5, 1\}, K_f \in \{0.1, 0.3\}, P_s \in \{0.99, 0.9999\}$$

The report should explain how these factors influence the CW change over the duration of the session.

Submit your code (C/C++/Python) that when executed from the command line produces all the desired output in a local directory. Include a 'readme' file.

Code: 70 Points, Report: 30 Points, Total 100 points.