# Networks Assignment 2

Nischith Shadagopan M N
CS18B102
*Computer Science and Engineering*
*Indian Institute of Technology Madras*

## I. Aim

The aim of this assignment is to simulate network congestion and see how the TCP congestion control protocol handles it.

## II. Introduction

Network congestion is caused because many flows share the same links. Packets can be dropped due to network congestion and this leads to re transmission of packets which further increases the congestion. Dropping of packets reduces the efficiency of the network. Hence we see the need for a congestion control protocol to help deal with these problems.

## III. Congestion Control Protocol

The congestion window size is modified as follows:

*notation*

- $MSS$ : Maximum Segment Size
- $CW$ : Congestion Window size
- $K_i$ : Initial congestion window
- $K_m$ : Multiplier of congestion window during exponential growth phase
- $K_n$ : Multiplier of congestion window during linear growth phase
- $K_f$ : Fraction to reduce when timeout

*pseudo code*

Initialize congestion threshold to $\infty$
Initialize $CW_{new} = K_i \times MSS$

- If packet received correctly i.e ACK is received:
  - If in exponential growth phase:
    $CW_{new} = \min(CW_old + K_m \times MSS, RWS)$
    If $CW_{new} \geq$ congestion threshold then go to linear growth phase.
  - Else it is in linear growth phase
    $CW_{new} = \min(CW_old + K_i \times MSS \times \frac{MSS}{CW_{old}}, RWS)$
- Else it is a time out:
  $CW_{new} = \max(1, K_f \times CW_{old}$
  congestion threshold = $\frac{CW_old}{2}$
  If $CW_{new} \geq$ congestion threshold then go to linear growth phase else go to exponential growth phase.

This work was done as part of Introduction to Computer Networks course

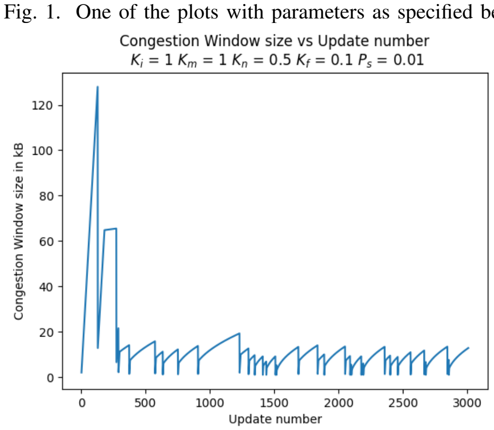
## IV. Implementational details

- Receiver window size is set to 1 MB and constant throughout.
- Sender's MSS is 1 KB.
- The sender has data to send throughout the simulation.
- Go-Back-N is used for flow control and each segment has its own timeout timer and ACK.
- The congestion window is rounded up to the nearest multiple of MSS.
- The congestion threshold is always set to 50% of the $CW$ value when there is a timeout.

No messages are actually sent in the simulation, receiving $ACK$ is modeled as a Bernoulli distribution.

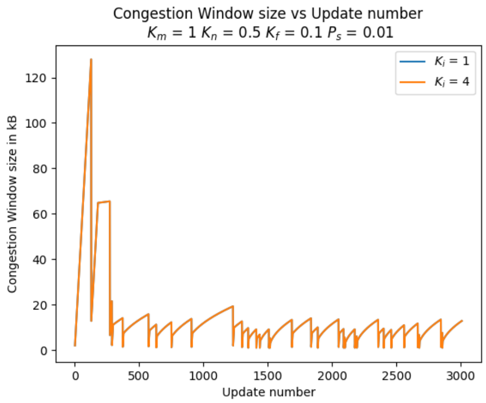
## V. Results and Observations

Fig. 1. One of the plots with parameters as specified below



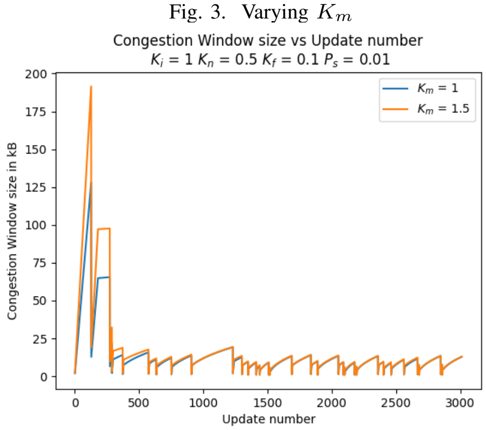

As we can see in this generic plot, initially the the threshold is set to $\infty$ and hence we have exponential growth till timeout and hence we have large peaks initially. But slowly the threshold stabilizes and hence the congestion window size also stabilizes. We can also see the exponential and linear increase phases in this graph.

Fig. 2. Varying $K_i$



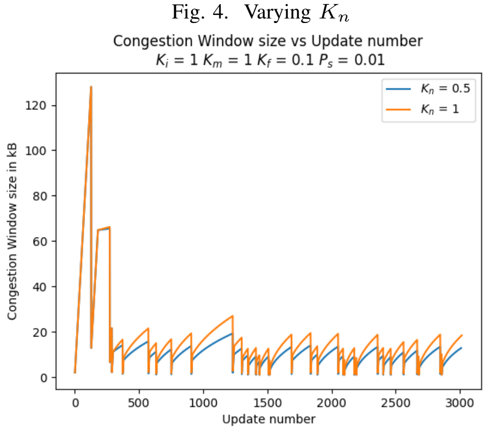

As we can see here, there is not much change when we change $K_i$. This is because $K_i$ is just the initial congestion window value and this small difference cannot be seen in the graph due to Y axis scale.

Fig. 3. Varying $K_m$



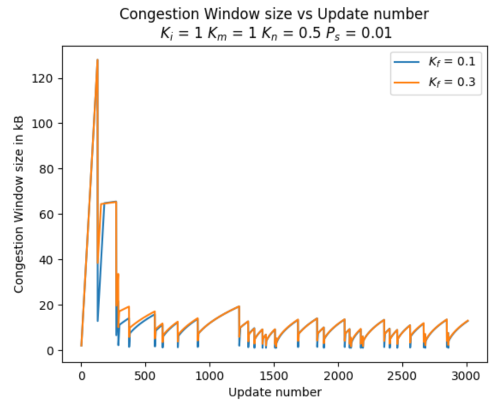

When $K_m$ is increased, in the exponential growth phase we reach higher values as expected.

Fig. 4. Varying $K_n$



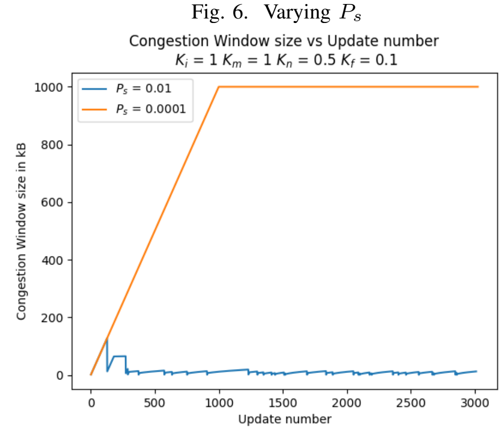

When $K_n$ is increased, in the linear growth phase we reach higher values as expected.

Fig. 5. Varying $K_f$




When $K_f$ is decreased, whenever there is a timeout, the congestion window value is reduced more.

Fig. 6. Varying $P_s$




When we decrease $P_s$, timeout becomes so rare that initially in the exponential growth phase, the congestion window size increases to a very large value and saturates at the Receiver window size.

## VI. Learning Outcomes

- Clear understanding of modified TCP congestion control protocol
- Saw how useful the TCP congestion protocol is in practice through simulations
- Learnt to write script files and python plotting
- Learnt to document my work thoroughly

## VII. Additional Thoughts

This work can be extended to include multiple acknowledgements instead of each packet having its own timer. We can also try to find for which value of $K$s, we can get the highest average congestion window size.

## VIII. Conclusion

We can see that the TCP congestion control protocol works well in simulations. It is able to provide a stable congestion free flow in the network. It also responds quickly to changing condition, there is a sharp decrease when there is a timeout to prevent congestion. It later grows quickly after the timeout also preventing under utilization of the network.

## REFERENCES

[1] Chigullapally Sriharsha, Assignment 2 problem statement pdf and slides.