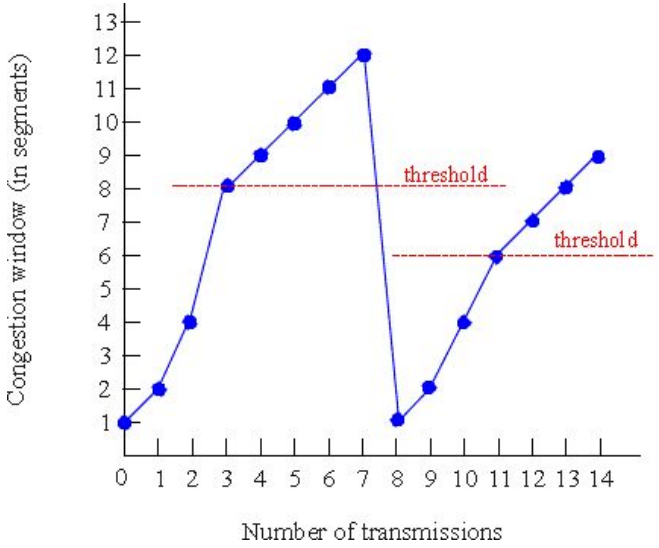
Assignment-8

TCP Congestion Control Simulation

Objective of the Assignment

The objective of this assignment is to experiment is to dynamically vary the sender's congestion window size as per the congestion control algorithm and then plot the change of CW with respect to the number of updates made to CW.



Congestion Control

Congestion control refers to techniques and mechanisms that can-

- Either prevent congestion before it happens
- Or remove congestion after it has happened

TCP Congestion Control

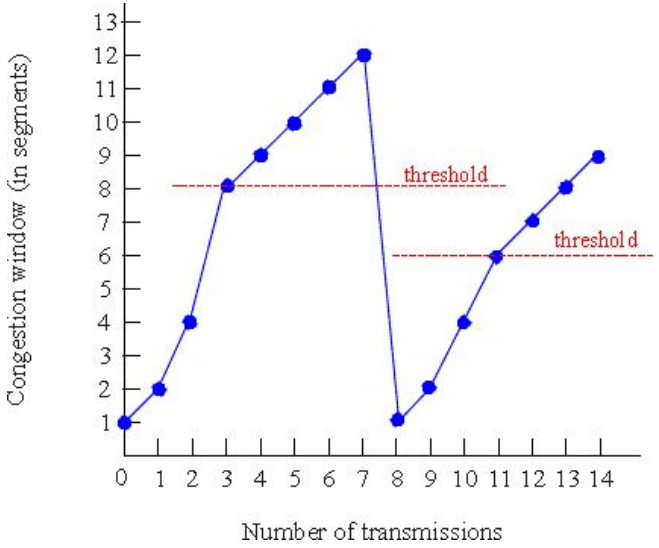
TCP reacts to congestion by reducing the sender window size.

The size of the sender window is determined by the following two factors

1. Receiver window size - Sender should not send data greater than receiver window size
2. Congestion window size - Sender should not send data greater than congestion window size.

TCP Congestion Policy

1. **Slow Start Phase**:=> the size of congestion window increases exponentially
2. **Congestion Avoidance Phase**:=> After reaching the threshold increment linearly.
3. **Congestion Detection Phase**:=> Once the congestion is detected, sender goes back to Slow Start Phase or Congestion avoidance phase.



1. Slow Start Phase

- Initially, sender sets the congestion window size

  *CWnew = Ki * MSS*

  Where, CWnew - Congestion Window

  Ki - Initial window size

  MSS - Maximum Segment size

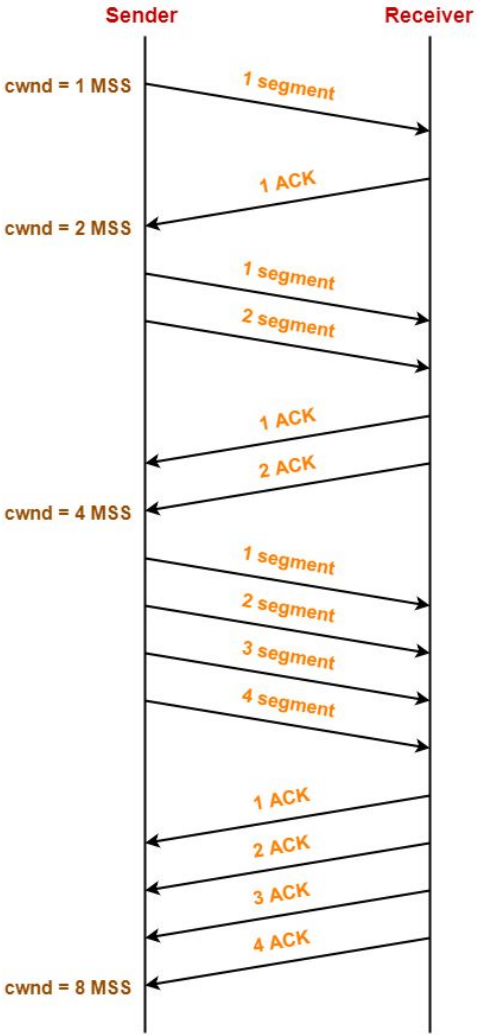
- After receiving each acknowledgment, sender increases the congestion window size

  CWnew = min(CWold + Km * MSS, RWS)

  Where, Km - multiplier of Congestion Window during exponential growth phase.

  RWS - Receiver Window Size

- In this phase, the size of congestion window increases exponentially.



2. Congestion Avoidance Phase

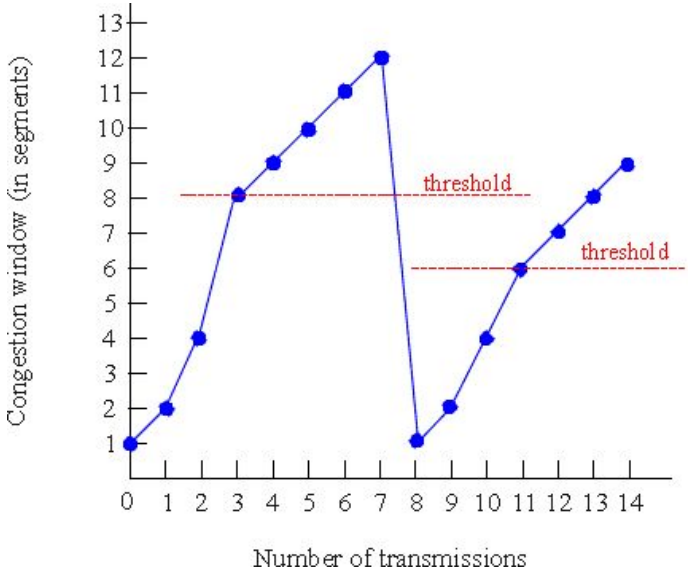
After reaching the threshold,

- Sender increases the congestion window size linearly to avoid the congestion.
- On receiving each acknowledgement, sender increments the congestion window size

$$CWnew = min(CWold + Kn * MSS * MSS/CWold, RWS)$$

Where,

Kn - multiplier of Congestion Window during linear growth phase.



3. Congestion Detection Phase

Sender detects the loss of segments/packet

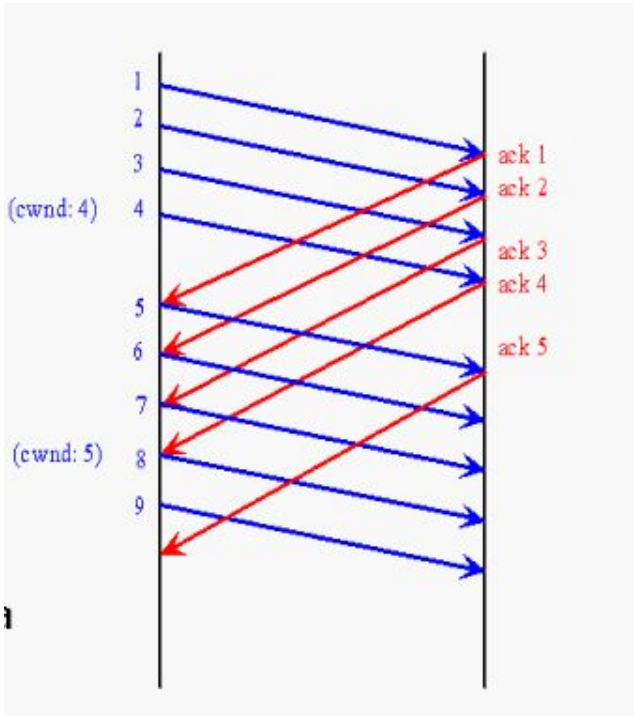
How to decide if Congestion occurs

- The idea here is to simulate the process of sending transport layer segments and receiving ACKs corresponding to each and every segment sent
- For each segment sent, we toss a coin to decide if the sender will receive an ACK corresponding to the segment or not.

  (*Programmatically, we call a built-in function in C (such as rand()) which will generate random numbers from a uniform distribution and compare the generated random number with the user supplied probability Ps. If the generated random number is less than Ps, it means ACK is lost implying packet loss. However, if generated random number is more than Ps, it means ACK successfully reaches sender.*)

- If congestion occurs (i.e., ACK does not reach sender), decrease CW multiplicatively as follows:

  CWnew = max(1, Kf * CWold) // during failure of receipt of ACK



Output

Output file - Whenever the CW value updated (either decrease the CW size during congestion or increase during the linear/exponential growth), write the value into a file. This file write operation should happen for each update to CW.

Output file format

Update Number	**CW value / size**
1	1
2	4
3	6
4	8
5	10
6	12

What to do next

You have to do this experiment for different values of

- Ki
- Km
- Kn
- Kf
- Ps

At any point of time in your simulation you should vary any one of the parameters mentioned above, while keeping the other fixed to some value.

Example:

Suppose you want to study the behavior of congestion avoidance algorithm with respect to different values of Ki, where Ki varies from 1 to 4.

1. Fix the value of Km=1.0, Kn=2.0, Kf=0.4 and Ps=0.009.
2. Fix the value of Ki to 1 and execute your program with all the values you just fixed.
3. Change value of Ki to 2 and repeat the above steps
4. Continue doing step 2 for Ki=3 and 4 respectively.
5. Now you get 4 output files corresponding to different values of Ki. Plot a graph by taking input as the 4 files and understand the behavior of congestion avoidance algorithm with different values of Ki
6. Explain your understanding in a report
7. Continue this process for all the different parameters(Km, Kn, Kf and Ps). At the end you should be getting 5 graphs.

Gnuplot

- is a portable command-line driven graphing utility for many platforms.
- can be used interactively to plot functions and data points in both two- and three-dimensional plots.
- To install execute the following commands in Linux OS

  $sudo apt-get update

  $sudo apt-get install gnuplot

- To Draw run the following commands

  user@debian:~$ gnuplot

  gnuplot> plot 'graph.txt' w l

Thank you

Questions?