Mobile Transport Layer

Popular Congestion Control Algorithms

- TCP Tahoe
- TCP New Reno
- TCP SACK
- TCP Vegas

TCP Tahoe

- Widely used
- Process of Algorithm includes – Slow-start, Congestion Avoidance (CA) and Fast Retransmit (FR)
- Slow-start and CA maintain optimal Congestion Window (CW)
- Slow-start CW exponential increase and CA very slowly
- If ACK for packet not received before timer expires, assumed to be lost and CA starts
- For CA Tahoe uses **“Additive Increase and Multiplicative Decrease”**

Contd…

- In CA, Congestion Threshold (Cthres) is set to half the previous CW
- CW starts from 1 SS (Segment Size), slow-start started till it reaches Cthres
- Then increments linearly till packet loss encountered
- For 3 dup ACKs, CA stops and FR starts
- From the value of ACK packet, the sender knows the sequence no. of next expected packet to be delivered
- Does not wait for Timer to expires, so may results in dup packet delivery

Contd…

- So for both packet loss and out of order, TCP Tahoe, retransmit the lost packet and reduces the CW size to by 2 or minimum
- Out of packet arises – load balancing mechanism and routing protocol using multipath routing
- Issue with TCP Tahoe – retransmit out of order packets and call congestion control algorithm unncessarily
- Leads to reduction in transmission rate, bandwidth wasted

TCP New Reno

- Improvisation over TCP Reno
- TCP Reno – retransmit all data packets of same segment and enters into Slow-start
- Leads to unnecessary retransmission
- In TCP New Reno, it transmit partial ACKs of packets lost, notifies lost packet sequence no.
- So minimize packet retransmission, bandwidth utilization improved

Contd…

- In Fast Recovery phase, not waiting for timer to expire, retransmit all lost packets in a window
- Best suited for mobile networks where multiple packet losses are common
- It retransmit one packet / partial ACK / RTT

TCP SACK

- Improvization over TCP Reno
- TCP Reno, retransmit packet once 3 dup ACKs is received without waiting for timer expire
- Sometimes packet may receive by that time, leading to bandwidth wastage
- TCP SACK, enables receiver to indicate out of sequence packet received
- SACK choice is used in ACK to indicate lost packets by the receiver
- SACK option does not change the nature of congestion control algorithm

Contd…

- It uses the same strategies as TCP Reno and TCP Tahoe for out of order packets
- Only difference when multiple packets are lost from a window, only that packets are retransmitted
- **However suffers performance penalty, sender has to list all the packets transmitted to find the lost packets**
- **Listing Procedure cause performance degradation**

TCP Vegas

- It extends the retransmission mechanism of TCP Reno
- Calculates RTT for each packet it sends
- For unACK packet, it checks whether difference between existing time and recorded timestamp value > time out value
- If greater it resends the packet without waiting for 3 dup ACKs
- Central Idea – Calculate retransmission time to infer available network bandwidth
- But TCP Reno, increases bandwidth till it shoots up

Contd…

- Finding available bandwidth and actual bandwidth as follows:
- Base RTT is set to minimum of all RTTs (first transmitted segment before router queue increases)
- Based on this **Available bandwidth** is computed
- Calculates **Actual sending rate** by finding no. of bytes send between for certain segment send and ACK
- Based on this **Actual bandwidth** is computed
- CW is increased when **Available bandwidth > Actual bandwidth**
- CW is reduced **Available bandwidth < Actual bandwidth**