FCN

Project 2: TCP TAHOE

Ketki Trimukhe(kt5829)

TCP Tahoe was early TCP version that took care of congestion control, flow control, corrupted packets and retransmission of lost packets. The motivation for this version was to use self-clocking and match the sending and the receiving rate of the packets thus avoiding maximum packet loses. This is done by two stages: slow start and congestion avoidance.

1. Slow Start:

In this version of TCP a congestion window is maintained. This window indicates the number of packets that can be sent at a given point of time. The cwnd(congestion window) starts slowly with the value of 1, hence this stage is called the slow start, and increases exponentially with each RTT. So the value of cwnd increases by one with every Ack for a packet received by the receiver. There are 3 ways to determine a packet loss or the need of retransmission:

1. 3 DupAcks: The receiver sends duplicate acknowledgments for a given packet that has been lost due to network congestion or the buffer overflowing at the receiver side.
2. TimeOut: If the acknowledgement packet is lost in the network and the sender does not receive and acknowledgment for a given timeout period, we assume that the packet is lost due to congestion in the network.
3. If the checksum of the packet does not match the received packet the receiver drops the packet and sends an acknowledgment for the packet expected in line.
4. Congestion Avoidance:

These actions at the receiver lead to the start of congestion avoidance stage. The moment the receiver knows a packet has been lost, the following actions take place:

1. The lost packet is retransmitted.
2. ssthresh is set to the current cwnd/2
3. cwnd=1
4. cwnd increases by 1 for every ack that is it increases exponentially with each RTT until it becomes equal to ssthresh.
5. Once the cwnd becomes equal to ssthresh it goes into the congestion avoidance stage, here the cwnd increases linearly with every RTT that is 1/cwnd with every ACK received.