**General Overview**

Congestion Control mechanisms of old TCP was inadequate for the rapidly growing Internet. A bottom-up rethink of the policies were needed in order to adapt the TCP stack, in an effort to make congestion the exception rather than a common case scenario.

The main observation around which the solutions presented in this paper was the "packet conservation" principle - "for a connection in equilibrium, a new packet shouldn't enter the network before the old packet leaves it". This principle is violated when connection doesn't reach equilibrium, sender injects a new packet before the old packet has exited or equilibrium cant be reached because of resource limits along the path.

The solutions presented deal with the scenarios mentioned above with following mechanisms

- *Slow Start* - To reach equilibrium (after connection start or restart), TCP uses a new congestion window (cwnd) that grows with every ACK received. 'cwnd' is set to 1 at every start or restart and it grows exponentially until a threshold is reached. This addresses the problem of reaching an equilibrium by ensuring that a connection will source data at a rate at most twice the maximum possible on the path. This automatically brings down the packet loss and the subsequent retransmission.
- *Round Trip Timing* - RTT measures governs when a packet is deemed lost and is retransmit. Accurate measurements would help reduce the retransmissions that are unnecessary. The main contribution to this area is the development of an algorithm to estimate variance in RTT. Exponential Back-off is employed in retransmission.
- *Congestion Avoidance* - Packet loss is taken a signal of network congestion and the window sizes are reduced to source data at a lower rate.

**Assumptions, are they still valid?**

As mentioned in the paper at several places, the design of some of the algorithms were applicable to the Internet Traffic and Demands of the early 90s. Some of these would require a rethink in the current scenario.

- Timeouts are due to packet loss and Packet loss is almost always due to congestion - This assumption might not be valid in the case of wireless networks.
- The congestion control scheme proposed is quadratic in 'w' (old window size). As mentioned in the paper, this might be reasonable given the window sizes of ARPANET (~16 bytes), but with window sizes of 64k, this algorithm might not be efficient.
- With current window sizes, algorithms proposed would require a rethink, may be to make the converge faster or take advantage of larger window sizes.
- Also, slow start algorithm requires the congestion window to be reduced to 1 at every start or restart - this might be an over kill.