CS540 - Paper Review Report # VIII

Hailu Belay Kahsay - 20155624

Title: Congestion Avoidance and Control

Author: Van Jacobson, Michael J. Karels

This paper introduces about congestion problems, with some statistical data, their causes, where they appear most and the fundamental technologies/techniques used to avoid and control TCP congestion, most of which are still widely used in today's IPv4 network. Advancements in congestion control algorithms put into the 4BSD TCP: round-trip-time variance estimation, exponential retransmit timer backoff, slow-start, more aggressive receiver ack policy, dynamic window sizing on congestion, Karn's clamped retransmit backoff and fast retransmit is also discussed in the paper. In the paper it is stated that packet conservation principle, that states for a connection to be in equilibrium and for a packet flow to be conservative a new packet isn't put into the network until an old packet leaves, is proposed.

This paper also discusses about the three problems that can cause packet conservation to fail: [1] The connection doesn't get to equilibrium, [2] A sender injects a new packet before an old packet has exited and [3]The equilibrium can't be reached because of resource limits along the path, with their mitigation techniques: slow-start, round-trip timing, and additive increase and multiplicative decrease of the window size respectively.

After presenting some congestion control and avoidance schemes and explaining the principles behind each scheme, the paper strengthens its explanations with graphs and computations.

Even though the paper addresses the three problems that can lead to congestion, there are still some issues: How to avoid or control congestion happened due to the waiting time of slow-start for time-out before it can restart? When congestion happens, numerous hosts will lose packets and all of them will drop the window size dramatically at the same time.How to effectively utilize network bandwidth during this time? Since TCP uses dropping packets to indicate congestion, misbehaved sources can cause congestions all the time and affect the whole network performance. How to handle such scenarios?

Title: **Simulation-based Comparisons of Tahoe, Reno, and SACK TCP**

Author: Kevin Fall and Sally Floyd

TCP is a reliable connection oriented end-to-end protocol, that contains within itself, mechanisms for ensuring reliability by requiring the receiver to send an acknowledgement for the segments that it receives. But, possibly small percentage of packets are lost in the route, either due to network error or due to the fact that there is congestion in the network and the routers are dropping packets. Mostly, packet losses due to network loss are minimal and most of the packet losses are due to buffer overflows at the router. Thus, TCP has to react to a packet loss and take action to reduce congestion. TCP ensures reliability by starting a timer whenever it sends a segment. If it does not receive an acknowledgement from the receiver within the 'time-out' interval then it retransmits the segment.

The paper starts by analyzing and comparing(simulation based) different congestion control and avoidance mechanisms which have been proposed for TCP/IP protocols, namely: Tahoe, Reno, New-Reno, TCP Vegas and SACK. It is shown that New-Reno TCP, a modified version of TCP without SACK, avoids some of Reno TCP's performance problems when multiple packets are dropped from a window of data. Whereas, SACK TCP, a conservative extension of Reno TCP modified to use the SACK option, works around the problems faced by TCP RENO and TCP New-Reno, namely detection of multiple lost packets, and retransmission.

The paper also introduces the above mentioned congestion control and retransmission algorithms: basic principle, how it works, unique features(s), pros and cons, when to use it, etc... After presenting about the congestion control and avoidance algorithms, the authors conducted simulations for four scenarios, with from one to four packets dropped from a window of data by running each set of scenarios for Tahoe, Reno, NewReno, and SACK TCP. The authors then measure and represent their result using graphical comparisons.

The paper has explored the fundamental restrictions imposed by the lack of selective acknowledgments in TCP, and have examined a TCP implementation that incorporates selective acknowledgments into Reno TCP while making minimal changes to TCP's underlying congestion control algorithms. It finally compare and contrast the different congestion control and avoidance algorithms by clearly specifying advantage and disadvantage of each algorithm.