CS540 - Paper Review Report # IX

Hailu Belay Kahsay - 20155624

Title: Binary Increase Congestion Control (BIC) for Fast Long-Distance Networks

Author: Lisong Xu, Khaled Harfoush, and Injong Rhee

Real time applications require access to high-bandwidth real time data, images and video captured from remote sensors, and predictable, low-latency access to this data, In early days , when TCP was used as a data transfer protocol for these networks, but, with long RTTs, TCP underutilized network bandwidth. I.e. current protocols mainly considers TCP friendliness and bandwidth scalability but not RTT fairness.

The paper discusses about RTT unfairness problem in detail, its sources: Synchronized loss, using multiple flows with different RTT delays, and proposed a new protocol called Binary Increase Congestion Control (BIC) , that satisfies criteria such as : [1] Scalability, [2] RTT fairness [3] TCP friendliness and [4] Fairness and convergence, as a solution. Synchronized loss model has been used for the theoretical analysis of RTT fairness. In addition to this, the authors have conducted an NS based simulation involving 20 high-speed connections of HSTCP with RTTs varying from 40 ms to 150 ms in the network with bottleneck link bandwidth varying from 100Mbps to 2.5 Gbps. Using synchronized loss model they have also analyzed the effect of synchronization loss on the RTT fairness of HSTCP and STCP.

The paper also talks about Binary Increase Congestion Control (BIC) protocols ,in detail, its basic idea, how it operates, it advantage over others, the algorithm, and its parts: binary search increase and additive increase.

Although BIC achieves pretty good scalability, fairness, and stability with high speed environments, I think the BIC's growth function can still be too aggressive for TCP, especially under short RTT or low speed networks. How to solve such problems? Complexity issue: since the protocol has several different phases of window control, analyzing it will add a lot of complexity? On an early packet drop, BIC may suffer in its initial ascent due to an incorrect estimate of the maximum link capacity?

Title: Equation-Based Congestion Control for Unicast Applications

Author: Sally Floyd, Mark Handley, Jitendra Padhye, Jorg Widmer

Congestion control is an important component of a transport protocol in a packet-switched network and the congestion control algorithms are responsible for detecting congestion and reacting to overloads in the Internet. TCP, the dominant transport protocol in the Internet, uses an Additive Increase Multiplicative Decrease (AIMD) congestion control algorithm. i.e. the 'sending rate' is controlled by a congestion window which is halved for every window of data containing a packet drop, and increased by roughly one packet per window of data otherwise. Using such approach for real-time applications will reduce the user-perceived quality.

This paper has three main goals: [1] To present a proposal for equation based congestion control that lays the foundation for the nearterm experimental deployment of congestion control for unicast streaming multimedia. [2] to contribute to the development and evaluation of equation-based congestion control, by addressing concerns in the some design issues such as : responsiveness to persistent congestion, avoidance of unnecessary oscillations, avoidance of the introduction of unnecessary noise, and robustness over a wide range of timescales and [3] to build a solid basis for the further development of congestion control for multicast traffic.

Detail discussions on basic idea of TCP-Friendly Rate Control(TFRC) Protocol, its design principles Such as: Do not aggressively seek out available bandwidth, Do not halve the sending rate in response to a single loss Event, other additional design principles and its features are also included in the paper.

Simulation based performance evaluation(at various timescales, with long-duration background Traffic and with ON-OFF flows as background traffic) and effects of TFRC on Queue dynamics has been conducted and it showed that TFRC is generally fair to TCP traffic across the wide range of network types and conditions.

- How to select the appropriate scheme for TFRC and TCP, when measuring RTT and RTO? i.e the different scheme used for TFRC and TCP, when measuring RTT and RTO, would drive in different behaviors between them, especially during the slow-start phase.
- The equation is a rate-based control, such that when RTT is so small (much smaller than the host OS's interrupt timer granularity), there would be a period that the sender produces zero throughput in an extreme case. This ill-reported rate can limit the TFRC's computed rate, resulting in being excessively conservative mode.
- I think TFRC is not a complete protocol; it is a congestion control mechanism only that could be implemented in a transport protocol like real-time transport protocol or in an application incorporating end-to-end congestion control at the application level.