Congestion Control for High Bandwidth-Delay Product Networks

Dina Katabi
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139
dk@mit.edu

For the Internet to continue to thrive, its congestion control mechanism must remain effective as the network evolves. Technology trends indicate that the future Internet will have a large number of very high-bandwidth links. Less ubiquitous but still commonplace will be satellite and wireless links with high latency. These trends are problematic because TCP reacts adversely to increases in the per-flow bandwidth-delay product. This extended abstract proposes a new congestion control protocol, called XCP, that outperforms TCP in both traditional and high bandwidth-delay product environments.

The problem facing TCP as the per-flow bandwidth-delay increases is multi-fold. First, mathematical analysis of TCP congestion control reveals that, regardless of the queuing scheme, as the delay-bandwidth product increases, TCP becomes oscillatory and prone to instability. By casting the problem into a control theory framework, Low et al. [10] show that as capacity or delay increases, Random Early Discard (RED) [4], Random Early Marking (REM) [2], Proportional Integral Controller [6], and Virtual Queue [5] all eventually become oscillatory and prone to instability. They further argue that it is unlikely that *any* Active Queue Management scheme (AQM) can maintain stability over very high-capacity or large-delay links. Furthermore, Katabi and Blake [7] show that Adaptive Virtual Queue (AVQ) [9] also becomes prone to instability when the link capacity is large enough (e.g., gigabit links).

Inefficiency is another problem facing TCP in the future Internet. As the delay-bandwidth product increases, performance degrades. TCP's additive increase policy limits its ability to acquire spare bandwidth to one packet per RTT. Since the bandwidth-delay product of a single flow over very-high-bandwidth links may be many thousands of packets, TCP might waste thousands of RTTs ramping up to full utilization following a burst of congestion.

Further, the increase in link capacity does not improve the transfer delay of short flows (the majority of the flows in the Internet). Short TCP flows cannot acquire the spare bandwidth faster than "slow start" and will waste valuable RTTs ramping up even when bandwidth is available.

Additionally, since TCP's throughput is inversely proportional to the RTT, fairness too might become an issue as more flows in the Internet traverse satellite links or wireless WANs [11]. As users with substantially different RTTs compete for the same bottleneck capacity, considerable unfairness will result.

Although the full impact of large delay-bandwidth products is yet to come, we can see the seeds of these problems in the current Internet. For example, TCP over satellite links has revealed network utilization issues and TCP's undesirable bias against long RTT flows [1]. Currently, these problems are mitigated using ad hoc mechanisms such as ack spacing, split connection [1], or performance enhancing proxies [3].

To address the above problem, we have developed a novel protocol for congestion control that outperforms TCP in conventional environments, and further remains efficient, fair, and stable as the link bandwidth or the round-trip delay increases [8]. This new eXplicit Control Protocol, XCP, generalizes the Explicit Congestion Notification proposal (ECN) [12]. Instead of the one bit congestion indication used by ECN, our routers inform the senders about the degree of congestion at the bottleneck. Another new concept is *the decoupling of utilization control from fairness control*. To control utilization, the new protocol adjusts its aggressiveness according to the spare bandwidth in the network and the feedback delay. This prevents oscillations, provides stability in face of high bandwidth or large delay, and ensures efficient utilization of network resources. To control fairness, the protocol reclaims bandwidth from flows whose rate is above their fair share and reallocates it to other flows.

By putting the control state in the packets, XCP needs no per-flow state in routers and can scale to any number of flows. Further, our implementation of XCP, requires only a few CPU cycles per packet, making it practical even for high-speed routers.

Using a control theory framework motivated by previous work [9, 6, 10], we show in [8] that a fluid model of the protocol is stable for any link capacity, feedback delay, or number of sources. In contrast to the various AQM schemes where parameter values depend on the capacity, delay, or number of sources, our analysis shows how to set the parameters of the new protocol to constant values that are effective independent of the environment.

Our extensive packet-level simulations in [8] show that, regardless of the queuing scheme, TCP's performance degrades significantly as either capacity or delay increases. In contrast, the new protocol achieves high utilization, small queues, and almost no drops, independent of capacity or delay. Even in conventional environments, the simulations show that our protocol exhibits better fairness, higher utilization, and smaller queue size, with almost no packet drops. Further, it maintains good performance in dynamic environments with many short web-like flows, and has no bias against long RTT flows. A unique characteristic of the new protocol is its ability to operate with almost zero drop rate (less than 10^{-6}).

Although our main goal is to solve TCP's limitations in high-bandwidth large-delay environments, our design has several additional advantages. First, decoupling fairness control from utilization control provides a flexible framework for integrating differential bandwidth allocations. For example, allocating bandwidth to senders according to their priorities or the price they pay requires changing only the fairness controller and does not affect the efficiency or the congestion characteristics [8].

Second, the new protocol facilitates distinguishing error losses from congestion losses, which makes it useful for wireless environments. In XCP, drops caused by congestion are highly uncommon (e.g., less than one in a million packets in simulations). Further, since the protocol uses explicit and precise congestion feedback, a congestion drop is likely to be preceded by an explicit feedback that tells the source to decrease its congestion window. Losses that are preceded and followed by an explicit increase feedback are likely error losses.

In [8], we describe XCP in more detail, and evaluate its performance in various scenarios.

References

[1] M. Allman, D. Glover, and L. Sanchez. Enhancing tcp over satellite channels using standard mechanisms, Jan. 1999.

[2] S. Athuraliya, V. H. Li, S. H. Low, and Q. Yin. Rem: Active queue management. *IEEE Network*, 2001.

[3] J. Border, M. Kojo, J. Griner, and G. Montenegro. Performance enhancing proxies, Nov. 2000.

[4] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. In *IEEE/ACM Transactions on Networking, 1(4):397–413*, Aug. 1993.

[5] R. Gibbens and F. Kelly. Distributed connection acceptance control for a connectionless network. In *Proc. of the 16th Intl. Telegraffic Congress*, June 1999.

[6] C. Hollot, V. Misra, D. Towsley, , and W. Gong. On designing improved controllers for aqm routers supporting tcp flows. In *Proc. of IEEE INFOCOM*, Apr. 2001.

[7] D. Katabi and C. Blake. A note on the stability requirements of adaptive virtual queue, 2002. MIT Technichal Memo, 2002.

[8] D. Katabi, M. Handley, and C. Rohrs. Internet congestion control for future high bandwidth-delay product environments, 2002.

[9] S. Kunniyur and R. Srikant. Analysis and design of an adaptive virtual queue. In *Proc. of ACM SIGCOMM*, 2001.

[10] S. H. Low, F. Paganini, J. Wang, S. Adlakha, and J. C. Doyle. Dynamics of tcp/aqm and a scalable control. In *Proc. of IEEE INFOCOM*, June 2002.

[11] G. Montenegro, S. Dawkins, M. Kojo, V. Magret, and N. Vaidya. Long thin networks, Jan. 2000.

[12] K. K. Ramakrishnan and S. Floyd. Proposal to add explicit congestion notification (ecn) to ip, Jan. 1999.