Analysis of High Speed TCP Protocols Using Realistic Traffic

Key words: high speed TCP, fairness analysis, realistic traffic generation.

Motivation: An uncongested 56kbps modem connection should theoretically provide ample bandwidth for a voice over IP (VOIP) phone call. Yet, as users of VOIP surely have experienced firsthand, the quality of VOIP calls even from a campus network connected directly to today's 40 gigabit/sec Internet backbone often falls short of what one would expect. This observation is counter-intuitive, given the ample bandwidth and low congestion present on the modern Internet.

Observations such as these motivate the study of network protocols, the algorithms that fundamentally regulate the usage and performance of the internet. Indeed, Douglas Comer, a networking researcher at Purdue and VP of Research at Cisco, recently noted that "building a large packet-switching network is easy; understanding the behavior of traffic in a large packet-switching network is nearly impossible" [1]. One of the most studied of these behaviors is the fact that the congestion control mechanism of the Transmission Control Protocol (TCP), the most commonly used transport protocol on the Internet, prevents connections from taking full advantage of the high bandwidth network links that exist in today's Internet. As a result, a plethora of high-speed variants of TCP have been proposed in recent years, each of which approaches the problem of inefficient use of high-bandwidth links differently, and as a result have different performance profiles.

I propose to measure performance characteristics of four variants of TCP engineered for high speed networks – TCP CUBIC, H-TCP, HSTCP, and FAST – using a novel realistic network traffic simulation approach. This analysis is both important and innovative because the performance of these protocols has not yet been evaluated in the presence of realistic network traffic that captures the characteristics of modern Internet traffic. As implementations of these high-speed TCP protocols see increased usage, a thorough understanding of their behavior in the presence of modern Internet traffic is vital to ensure proper provisioning of network resources by service providers. It is equally vital to ensure the quality of service application users and developers will need for the next generation of technologies that will operate over the Internet. Moreover, this analysis will contribute to the development of principles and best practices that are currently lacking in the networking community for evaluating protocol performance.

Research Plan. An important property of high speed TCP protocols is the ability to coexist fairly with standard TCP traffic. Most studies of high speed TCP protocols have considered the performance of long-lived flows, such as large File Transfer Protocol (FTP) transfers [2,3], in what could be called the "FTP model" of traffic simulation. Obviously, this is not representative of traffic patterns on the modern Internet, which generally consists of a large number of short, small connections. Because of this, high speed TCP protocols run the risk of unfair performance when coexisting alongside realistic internet traffic.

A model for realistic traffic simulation is the "a-b-t" model, developed at UNC-Chapel Hill [4]. In the a-b-t model, traffic is modeled as a set of "connection vectors", consisting of some amount of data transferred in the "a" direction, a "server think-time" (for example, corresponding to the time required for a web server to render the requested page), some amount transferred back in the "b" direction, followed by another "user think-time" (which could be the time the user spends reading the rendered page before making the next request). A single a-b-t sequence is known as an "epoch". Connections between computers on the Internet often take the form of one or more epochs of data transfer rather than a steady stream of data in each direction as one would see in an FTP transfer.

I will create a-b-t connection vectors from real Internet traffic (from network traces available in the research community) and use this data as my experimental network traffic to control for arbitrary differences in network traffic (repeatability and control of confounding variables is the reason traffic simulation is necessary in networking research). I plan to use the workload generator Tmix to generate TCP connections and reproduce application-level behaviors observed in the captured Internet trace. This emulation faithfully reproduces the essential behavior of the applications that generated the original captured Internet traffic without knowledge of what those original applications actually were. In [4], the author describes Tmix and demonstrates how the generated traffic displays the key characteristics of the original captured trace. This traffic will be used as background traffic during my experiments, and a set of long-lived FTP-like flows generated using one of the high-speed TCP variants will be generated on top of this background traffic. Note that the use of FTP-like flows for the high speed variants is acceptable in this case, since the performance of these high speed TCP variants in short connections is identical to that of non-high speed TCP. I will run experiments with background traffic both at and below the network's capacity, simulating congested and uncongested scenarios, respectively.

I will then perform a matrix of experiments, comparing the performance of the four high speed TCP protocols mentioned above along with a fifth non-high speed TCP variant, TCP SACK, the current "gold standard" variant for general-purpose best practices. For each of the five protocols, I will test three traffic generation techniques in both congested and uncongested scenarios. The first of these will emulate the FTP model, ignoring the epoch structure of the experimental network traffic. While unrealistic, this is a common experimental setup in the literature, and it will provide a baseline against which to compare other experimental results. For realistic traffic generation, I will simulate network traffic using the a-b-t model both with and without think times (but otherwise preserving the epoch structure). Because think times can often dominate the delay of a given connection (more than even network latency), the difference between the two methods of "realistic" traffic generation will illuminate how faithful replication of epoch structure can affect protocol performance. Rather than using a network simulator, I plan to use a physical experimental network consisting of several pairs of machines connected through a pair of routers. Each pair of machines will be responsible for replaying a given portion of the experimental traffic. Fairness will be measured with Jain's fairness index, a metric commonly used in the high speed TCP literature.

Expected Findings. My analysis should uncover emergent behaviors of high speed TCP variants when operating alongside modern internet traffic. Moreover, it should shed light on the impact that epoch structure has on protocol performance, in particular in a network with heterogeneous variants of TCP. The results will contribute important knowledge to the networking community regarding the development of standards for experimental design that reflect protocol performance on the modern Internet. In a broader sense, this work will contribute to an understanding of how to effectively provision Internet connected networks and ensure those networks perform at the level society needs in the years to come.

Works Cited

[1] Comer, Douglas. *Lessons Learned from the Internet Project*. Oct. 8, 2008. Lecture.
[2] Molnar, S., et al. *A Comprehensive TCP Fairness Analysis in High Speed Networks.* Computer Communications, 2009. p. 1460-1484.
[3] Rhee, I., and Xu, L. *CUBIC: A New TCP-Friendly High Speed TCP Variant.* Proceedings of ACM SIGOPS, 2008. p.66-74.
[4] Felix Hernandez-Campos. *Generation and Validation of Empirically-Derived TCP Application Workloads*. PhD thesis, University of North Carolina at Chapel Hill, August 2006.