Performance of High-Speed TCP Applications in Networks with Very Small Buffers

Ben Zhao, Arun Vishwanath and Vijay Sivaraman
School of Electrical Engineering and Telecommunications
University of New South Wales
Sydney, NSW 2052, Australia
Emails: benz@student.unsw.edu.au, {arunv@ee., vijay@}unsw.edu.au

***Abstract*— In the on-going debate on how large Internet router buffers should be, of particular interest is a recent claim that 10-50 packets of buffering suffice, permitting all-optical buffering in the Internet core. It is argued that TCP traffic, when well-paced (typically by slow access links), can achieve acceptable link utilisation with very small buffers. In this paper we investigate if this result holds in the presence of high-performance scientific applications connecting over high-speed access links. Our results show that the presence of a significant proportion of high-speed access links necessitates larger buffers at the core link. However, we show that end-host TCP pacing by such applications can alleviate the problem effectively.**

I. Introduction

Buffers in Internet routers reduce packet loss by absorbing transient bursts of traffic. They are also instrumental in keeping output links fully utilised during times of congestion. An important question concerns the sizing of these buffers. Buffer overflow leads to packet loss, adversely affecting application performance. An under-flow causes idling and wastage of link bandwidth, thereby degrading network throughput. Currently, router manufacturers determine buffer size using a rule-of-thumb commonly attributed to [1]. Specifically, the rule-of-thumb mandates a buffer size of $B = T \times C$, where T denotes the average round-trip time of a TCP flow through the router, and C the capacity of the bottleneck link. For a typical T of 250ms, a router with a C = 40Gbps link would thus require 10 Gigabits of buffering.

The above rule-of-thumb was first challenged by the Stanford research group in 2004 [2]. With minimal impact on link throughput, they showed that the buffer requirement can be reduced to $B = \frac{T \times C}{\sqrt{N}}$, where N is the number of long-lived TCP flows sharing the bottleneck link. A core Internet router today carries tens of thousands of flows, consequently this reduction in buffer size is significant. Very recently, the Stanford group have shown theoretical and experimental evidence that when TCP traffic is paced, as few as 10-50 packet buffers can provide acceptable link utilisation [3]. This creates avenues for building cost-effective all-optical routers where buffering is often an expensive and complex operation.

The study in [3] relies on packets from individual TCP sessions being well-spaced at the core link. Typical access links (e.g. from home users) today tend to be much slower than core links, and this naturally paces the traffic. However, there are many high performance scientific applications that connect over fast access links. For example, as part of our collaboration with the Australia Telescope National Facility (ATNF), we are evaluating TCP performance for reliable transfer of real-time astronomical data between Australia and Europe. The telescopes in Australia interconnect over a high-speed (Gigabit) fibre-optic network [4], which feeds into a lower-speed inter-continental core link operating at 155Mbps. In such an environment, where access links are much faster than core links, we investigate if small buffers suffice at the core link in order to maintain high utilisation. Our contributions are two-fold: first, we show that when there are sufficiently large number of scientific applications connecting over high-speed access links, the buffer requirements at core routers increase. Second, we show that TCP pacing by end-hosts of such applications eliminates the need for larger buffers at core routers.

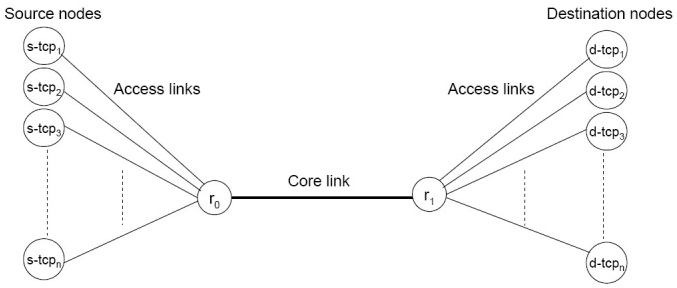

Fig. 1. ns2 simulation topology

II. Simulation setup and results

We use $ns2$ [5] to simulate TCP performance on a dumbbell topology shown in Fig. 1. The propagation delay on the access links is uniformly distributed between $[1, 25]$ms, while the core link (r_0, r_1) has a propagation delay of 50ms; the end-to-end round-trip time thus varies between 102ms and 150ms. We simulate 1000 TCP Reno flows between each source-destination pair *(s-tcp$_i$,d-tcp$_i$)*, $1 \leq i \leq 1000$, which is fairly realistic for a core link, and mitigates synchronisation issues. The buffer size at the core router r_0 is varied in packets. FIFO queue with simple drop-tail queue management is employed. TCP packet size is set to 1000 Bytes. All TCP sources start at random times between $[0, 10]$s. The simulation is performed for a period of 400s and only data in the interval $[100, 400]$s is used in the calculations.

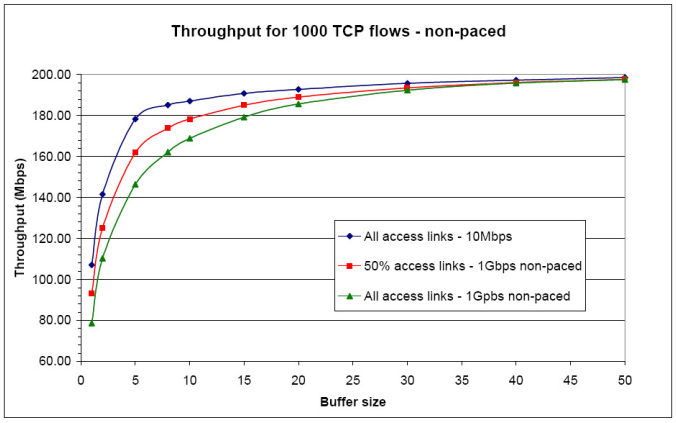

Fig. 2. Throughput for non-paced TCP flows

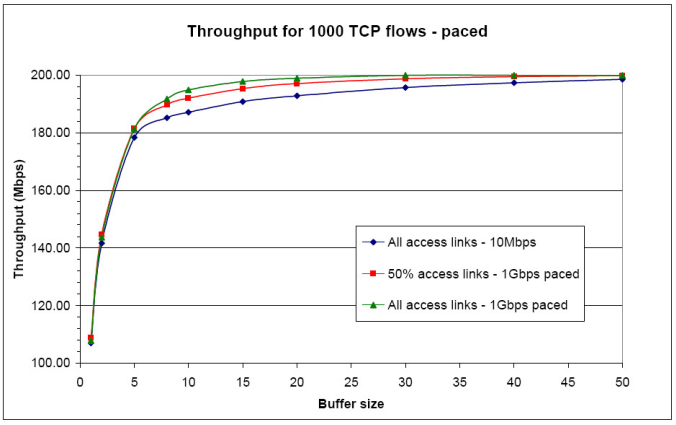

Fig. 3. Throughput for paced TCP flows

Fig. 2 shows the aggregate end-to-end throughput for the TCP flows as a function of core buffer size when the core link is 200Mbps. The top curve shows that when the access links operate at 10Mbps (thus naturally pacing TCP traffic into the core), as low as 5 packets of buffering suffice for TCP throughput to reach nearly 90% of the core link rate. This verifies that when access links are slow, only a small amount of buffering is needed to realise acceptable link utilisation. We now change the access speed to 1Gbps for half the TCP flows; these flows now correspond to high-performance scientific applications. The middle curve in the same figure shows that to obtain 90% TCP throughput, the core requires roughly 10 packets of buffering, a two-fold increase in buffer requirement. When all flows feed the core over high-speed access links, the buffer requirement to realise 90% throughput increases to 15, three-fold the amount required when all flows arrive over slow access links. For all-optical buffering, this extra buffer requirement translates into heavy cost penalties.

Having established that high-speed access links can significantly increase buffering requirements at the core, we evaluate the efficacy of TCP pacing by the end-host (generating the scientific application traffic) in mitigating this effect. We repeated the above set of simulations with TCP modified to pace traffic at end-hosts (the ns2 patch was taken from [6]). Fig. 3 plots the resulting TCP throughput as a function of buffer size, and demonstrates clearly the benefits of TCP pacing. To obtain 90% throughput, 5 packets of buffering suffice irrespective of the mix of traffic from low-speed and high-speed access links. In fact, in the region of 10 to 20 packets of buffering, TCP throughput increases by around 4% when all flows from high-speed access links are paced compared to when the flows arrive on low-speed access links. This demonstrates that pacing is extremely effective in reducing core buffer requirements, and in turn the cost of such, potentially all-optical, core routers.

We point out that there is no general agreement today on the overall benefits of TCP pacing. It is observed in [7] that pacing can severely undermine performance (measured in terms of throughput, fairness and latency) in many cases due to synchronised TCP behaviour, and also when mixed with non-paced TCP flows. However, their results were based on a small number of TCP flows (about 20-50) and larger buffer sizes. More recent studies such as [8]–[10] show that pacing has a positive impact (for TCP, XCP, and real-time traffic) in realistic scenarios, consistent with our observations in this paper.

III. Conclusion and Future Work

In this paper, we have studied the impact of high-speed applications (feeding traffic over fast access links) on buffer requirements at core routers. We showed that in the presence of several high-speed applications, much larger buffers are required at core routers to realise acceptable throughput. To mitigate this problem, we evaluated the pacing of TCP traffic from these high-speed applications, and showed pacing to be extremely effective in reducing core buffer requirements. In the future, our aim is to undertake an experimental study of TCP pacing in a realistic network operated by the Australia Telescope National Facility.

References

[1] C. Villamizar and C. Song, "High Performance TCP in ANSNet," *ACM SIGCOMM Computer Communications Review*, vol. 24, no. 5, pp. 4560, 1994.

[2] G. Appenzeller, I. Keslassy and N. McKeown, "Sizing Router Buffers," *Proc. ACM SIGCOMM*, Aug-Sep 2004.

[3] M. Enachescu, Y. Ganjali, A. Goel, N. McKeown and T. Roughgarden, "Routers with Very Small Buffers," *Proc. IEEE INFOCOM*, Apr 2006.

[4] The Australian Square Kilometre Array Pathfinder, Australia Telescope National Facility, CSIRO. `http://www.atnf.csiro.au/projects/askap/`

[5] The Network Simulator - ns-2, `http://www.isi.edu/nsnam/ns/`

[6] A TCP Pacing Implementation for NS2, `http://www.cs.caltech.edu/~weixl/technical/ns2pacing/index.html`

[7] A. Aggarwal, S. Savage and T. Anderson, "Understanding the Performance of TCP Pacing," *Proc. IEEE INFOCOM*, Mar 2000.

[8] D. X. Wei, P. Cao and S. H. Low, "TCP Pacing Revisited", `http://www.cs.caltech.edu/~weixl/research/summary/infocom2006.pdf`

[9] O. Alparslan, S. Arakawa and M. Murata, "Performance of Paced and Non-Paced Transmission Control Algorithms in Small Buffered Networks," *Proc. IEEE Symposium on Computers and Communications*, Jun 2006.

[10] V. Sivaraman, H. ElGindy, D. Moreland and D. Ostry, "Packet Pacing in Short Buffer Optical Packet Switched Networks," *Proc. IEEE INFOCOM*, Apr 2006.