Computer Networks 77 (2015) 103–116

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Comparing edge and host traffic pacing in small buffer networks ☆

Hassan Habibi Gharakheili [a,*], Arun Vishwanath [b], Vijay Sivaraman [a]

[a] *School of Electrical Engineering and Telecommunications, The University of New South Wales, Sydney, Australia*
[b] *IBM Research, Melbourne, Australia*

ARTICLE INFO

Article history:
Received 8 July 2014
Received in revised form 28 October 2014
Accepted 29 November 2014
Available online 17 December 2014

Keywords:
Small buffer networks
Edge pacing
Optical packet switched networks

ABSTRACT

As packet switching speeds scale to Terabits-per second and beyond, power considerations are increasingly forcing core router manufacturers to adopt all-optical and hybrid opto-electronic single-chip switching solutions. Such routers will have small buffers, typically in the range of a few tens of Kilobytes, causing potentially increased packet loss, with adverse impact on end-to-end TCP performance. We recently proposed and analysed the benefits of pacing traffic at the network edge for open-loop real-time traffic in a small buffer network. However, no detailed study of the efficacy of edge pacing on closed-loop TCP performance has been undertaken for such a network.

In this paper, we consider two pacing methods – TCP pacing at the end-hosts, and traffic pacing by the network edge – in the context of small buffer networks, and undertake a comprehensive comparison. Our contributions are threefold: First, we show via extensive simulations that under most scenarios (considering bottleneck and non-bottleneck core links, low-speed and high-speed access links, long- and short-lived TCP flows, and different variants of TCP) edge pacing performs as well or better than host pacing in terms of link utilisation (TCP throughputs) and average per-flow goodputs. Second, we provide analytical insight into the setting of the edge pacing delay parameter, showing how the efficacy of pacing relates to bottleneck buffer size. Third, we demonstrate the benefits of pacing in practical scenarios multiplexing both TCP and real-time traffic, and discuss incremental deployment of pacing, highlighting that unlike host pacing that requires adoption by a critical mass of users, edge pacing can be deployed relatively easily under service provider control to facilitate rapid migration to core networks with small buffers.

1. Introduction

As Internet traffic continues its inexorable growth, core routers are struggling to keep pace with the required switching capacity. Router scaling is primarily limited by power density – a typical rack today with a throughput of a Terabit-per second consumes tens of Kilowatts, and at current trends, scaling its capacity to Petabits-per second would require hundreds of Kilowatts of power, alongside complex cooling mechanisms. To sustain capacity growth, router manufacturers are therefore increasingly looking to photonics, including all-optical packet switching solutions and integrated single-chip systems employing hybrid optics and electronics. In order to perform energy-efficient high-speed packet forwarding, such architectures necessarily sacrifice many non-critical functionalities,

☆ This submission is an extended and improved version of our paper presented at the IFIP Networking 2013 conference [1].

* Corresponding author.

E-mail addresses: h.habibi@unsw.edu.au (H. Habibi Gharakheili), arvishwa@au.ibm.com (A. Vishwanath), vijay@unsw.edu.au (V. Sivaraman).

http://dx.doi.org/10.1016/j.comnet.2014.11.021
1389-1286/© 2014 Elsevier B.V. All rights reserved.

among them buffering of packets during periods of congestion. Recent research studies on such architectures have argued, based on theory, simulation, and experimentation, that core router buffer size can safely be reduced from Gigabytes down to Megabytes [2] or Kilobytes [3], and can even be nearly eliminated [4], though with some loss in performance. We refer the reader to our survey article [5] for a comprehensive discussion on the buffer sizing debate.

When router buffers in the network core are very small (sub-50 KB), contention and congestion at the output link can lead to high packet loss, significantly degrading end-to-end traffic performance. We have shown in [6] that real-time traffic streams can experience poor quality, and in [7] that TCP flows can have reduced throughput. Several mechanisms have been proposed for mitigating this problem, such as using wavelength conversion [8] in the core to alleviate contentions, feedback-based rate control to proactively prevent network congestion in optical burst switching networks [9,10], packet-level forward-error-correction (FEC) at edge nodes to recover from core loss [4], and traffic pacing at the edge prior to injection into the core [6,11–13]. While all these methods have their relative merits, in this paper we focus on pacing, since it is low-cost (compared to wavelength conversion), is a well-known concept (studied under various names such as rate-limiting, shaping, and smoothing), and is yet relatively unexplored in the context of small buffer networks.

Traffic can be paced in various parts of the network: by end-hosts as part of their TCP stack (host pacing), by the access link connecting the user to the network (link pacing), or by the edge node that connects into the core network (edge pacing). *Host pacing* (also known as TCP pacing) modifies the end-user client TCP stack to spread the transmission of packets from the TCP window over the round-trip-time (RTT) of the connection. Many researchers have studied host pacing over the past decade [14–16], and the general belief is that host pacing can, under most circumstances, improve overall TCP throughput. However, deploying host pacing has been stymied by the fact that the network operator does not have control over user devices to enforce pacing, and hosts that pace their TCP transmissions can be unfairly penalised over hosts that do not [15].

Link pacing relies on the access link being of much lower capacity than links deeper in the network, ensuring that packets belonging to any single flow are spaced apart when they arrive at the core link. This has been leveraged by works such as [3] to argue that neither can a single flow contribute bursty traffic to the core node, nor are many flows likely to synchronise to create bursts, and hence loss is contained. Though this argument applies to typical home users, entities such as enterprises, universities, and data centres are often serviced with high-speed links capable of generating bursty traffic that does not fit this assumption, necessitating explicit mechanisms (at the host or edge) to reduce burstiness.

Edge pacing relies on explicit smoothing of traffic by edge nodes prior to injection into the small buffer core network. In [6] we proposed a method that adjusts traffic release rate to maximise smoothness, subject to a given upper bound on packet delay. We proved the optimality of our scheme, analysed its burstiness and loss performance, and evaluated its impact for open-loop real-time traffic, though not for closed-loop TCP. A similar (though sub-optimal) pacing method was proposed in [11,12] to vary the edge traffic release rate based on queue backlog. However, the impact of edge pacing on TCP performance was only cursorily studied, and no appropriate guidelines on parameter settings were provided.

Our goal in this paper is to undertake a comprehensive comparison between host and edge pacing in the context of small buffer core networks, by evaluating their impact on end-to-end TCP performance. We seek to gain insights into the network and traffic characteristics that influence their efficacy, the parameter settings that maximise their benefits, and deployment strategy that make them practical in real networks. Our specific contributions are:

- We show using extensive simulations of various scenarios, considering small-buffered bottleneck and non-bottleneck links, low-speed and high-speed access links, short- and long-lived flows, different number of flows, and different variants of TCP, that edge pacing achieves as good or better performance than host pacing in terms of link throughput and per-flow goodput.
- We develop an analytical model that sheds light into the selection of the edge pacing delay parameter that maximises TCP throughput for different bottleneck link buffer sizes.
- We present evidence that edge pacing is beneficial in practical networks that multiplex both TCP and real-time traffic, and argue that the benefits of edge pacing can be easily realised under tight operator control in real networks, unlike host pacing that requires a critical mass of uptake by end-users for it to be effective.

Our intention is to show network operators that from a performance, configuration and deployment point-of-view, edge pacing presents an attractive alternative to host pacing as a mechanism for enabling scalable and energy-efficient core networks having small-buffers in the near future.

The rest of this paper is organised as follows: Section 2 gives requisite background on traffic pacing studies. In Section 3 we present comprehensive simulation studies comparing the performance of host and edge pacing, and in Section 4 we develop analytical insights into appropriate parameter settings. Section 5 shows that edge pacing benefits both TCP and real-time traffic, Section 6 discusses the deployment strategy for pacing, and the paper is concluded in Section 7.

2. Background and related work

It is well-known that TCP traffic is bursty at short-time scales [17] because of its self-clocking mechanism and queueing of packets at the bottleneck link. Bursty traffic is largely undesirable since it causes large queueing delays, higher packet loss, and degradation in end-to-end throughput. As a result, several researchers have proposed to pace