Computer Networks 56 (2012) 3076–3086

Contents lists available at SciVerse ScienceDirect

Computer Networks

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

Periodic early detection for improved TCP performance and energy efficiency ☆

Andrea Francini *

Computing and Software Principles Research Department, Alcatel-Lucent Bell Laboratories, Mooresville, NC, USA

ARTICLE INFO

Article history:
Received 11 October 2011
Received in revised form 20 March 2012
Accepted 17 April 2012
Available online 2 May 2012

Keywords:
Buffer management
TCP
Active queue management
Random early detection
Energy efficiency
Simulation

ABSTRACT

Reducing the size of packet buffers in network equipment is a straightforward method for improving the network performance experienced by user applications and also the energy efficiency of system designs. Smaller buffers imply lower queueing delays, with faster delivery of data to receivers and shorter round-trip times for better controlling the size of TCP congestion windows. If small enough, downsized buffers can even fit in the same chips where packets are processed and scheduled, avoiding the energy cost of external memory chips and of the interfaces that drive them. On-chip buffer memories also abate packet access latencies, further contributing to system scalability and bandwidth density. Unfortunately, despite more than two decades of intense research activity on buffer management, current-day system designs still rely on the conventional bandwidth-delay product rule to set the size of their buffers. Instead of decreasing, buffer sizes keep on growing linearly with link capacities.

We draw from the limitations of the buffer management schemes that are commonly available in commercial network equipment to define Periodic Early Detection (PED), a new active queue management scheme that achieves important buffer size reductions (more than 95%) while retaining TCP throughput and fairness. We show that PED enables on-chip buffer implementations for link rates up to 100 Gbps while relieving end users from network performance disruptions of common occurrence.

1. Introduction

As link rates keep growing faster than the density and access speed of memory chips, the conventional strategy of sizing packet buffers proportionally to the link capacity is creating critical challenges to the development of future generations of network systems and to the ability of packet networks to sustain the performance requirements of all user applications.

At times of traffic congestion, large packet buffers inflate end-to-end packet delays, extending the round-trip times of TCP connections and deferring the completion of file transfers. Large buffers also hamper the scalability of system designs and contribute prominently to energy consumption. Their instantiation typically requires large memories that cannot be integrated in the same chips that process and forward the packets. Off-chip memory components and the connectivity infrastructure needed to reach them consume a substantial portion of real estate on circuit boards, taking it away from packet-handling devices. Off-chip memories also impose limitations on packet forwarding rates. Compared to the ideal case with on-chip memories only, external memories cause a substantial increase in area and power consumed per unit of forwarding capacity.

☆ Part of this paper was previously presented in A. Francini, Beyond RED: Periodic Early Detection for On-Chip Buffer Memories in Network Elements, Proceedings of the 2011 IEEE High-Performance Switching and Routing Conference (HPSR 2011) in Cartagena, Spain, July 4–6, 2011.

* Address: P.O. Box 773, Davidson, NC 28036, USA. Tel./fax: +1 336 697 3405.

E-mail address: andrea.francini@alcatel-lucent.com

1389-1286/$ - see front matter © 2012 Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.comnet.2012.04.017

Despite being harmful to application performance and bandwidth density, large packet buffers (hundreds of milliseconds) remain a fixture in system designs because they maximize the aggregate throughput of congested TCP connections using the simplest buffer management policy (Tail-Drop). UDP traffic also needs buffering for the absorption of overload conditions, but small buffers are better than large ones at addressing all UDP overload occurrences [1]. Even better is to combine a small Tail-Drop buffer with the differentiation of packet drop priorities based on UDP compliance with pre-negotiated traffic profiles [2,3]. Unfortunately, small Tail-Drop buffers are not nearly as effective at handling TCP traffic, whose elasticity implies long-lasting congestion incidents and the absence of pre-negotiated per-flow contracts.

The bandwidth-delay product (BDP) rule [4] commonly drives the sizing of Tail-Drop buffers for TCP traffic: in front of a link of capacity C, the buffer size should be $B = C \cdot \bar{\vartheta}$, where $\bar{\vartheta}$ is the average round-trip time (RTT) estimated over the set of TCP flows that share the link. The BDP rule generally succeeds in avoiding queue underflow conditions, and therefore reductions of link utilization, when back-to-back packet losses occur at a congested buffer. However, the rule does not guarantee the throughput of individual TCP flows and does not enforce inter-flow fairness. BDP buffers that handle long-range TCP connections are commonly sized for RTT values in the 200 ms − 300 ms range. When one such buffer along the network path of a TCP flow is congested, its queueing delay adds a substantial contribution to the RTT experienced by the flow, decreasing its steady-state throughput in proportion to the ratio between the RTT values measured with and without congestion. As the number of congested BDP buffers along the same path increases, the consequences for the throughput of the TCP flow are catastrophic, because the effective RTT that defines the share of the flow at one bottleneck is augmented by the delay added by all the other bottlenecks. While the frequency of occurrence of multi-bottleneck congestion episodes may be low for the typical end user, few such episodes in the course of one day can produce major dissatisfaction, because their perceived effects are heavy and may take minutes to subside.

Network operators may not easily find evidence of the performance degradation suffered by TCP flows that encounter large buffers in their multi-bottleneck data paths. The placement of bottlenecks in different administrative domains, or simply the scale of the traffic volume handled by each network node, can make it impractical for monitoring tools to maintain flow-level statistics that reliably reproduce the end-user experience. As a result, the bufferbloat issue keeps lingering [5] while interest in more effective buffer management is fading. Indeed, after recognizing the performance limitations of solutions based on miniature Tail-Drop buffers [6–8] and concluding that active queue management (AQM) techniques like Random Early Detection (RED) [9,10] are promising on paper but inadequate for broad practical applications [11], the networking community has lately been focusing its efforts on re-designing the endpoint behavior so that it better adjusts to the ongoing capacity expansion in host and access links [12,13] and keeps the TCP performance insensitive to the buffer management flaws exposed by the network [14,15]. Growing energy-efficiency and system-scalability concerns now move our attention back onto buffer management enhancements that can reconcile TCP traffic with small buffers.

We find important merits in RED [9], especially in some of the many modifications that have been proposed to amend it [1,16], but also critical flaws. The algorithmic steps that we devise to address those flaws provide the backbone of a new AQM scheme that we call Periodic Early Detection (PED) [17]. Like RED, PED derives its packet drop decisions from the monitoring of the average queue length (AQL). However, PED differs radically from RED in the handling of the packet drop rate. While RED continuously adjusts the drop rate to the evolution of the queue length, PED applies drop rate adjustments only at the end of fixed time intervals, and only if the current rate is clearly off-range. By imposing a fixed packet drop rate, PED aligns with the conclusion presented in [11] that RED's stability improves as the slope of its packet drop probability function decreases. PED's distinctive new feature originates from the assumption that the queue is healthily loaded (neither empty nor overflowing) if the set of TCP flows in additive increase mode (slowly expanding their congestion windows) balances the set of TCP flows that are recovering from recent packet losses. Such a balance is the product of an *ideal packet drop rate* $\varphi(t)$ that evolves over time with the composition of the traffic mix and with the sequence of packet loss assignments to individual TCP flows. PED's periodic adjustments of the drop rate compensate for the infeasibility of tracking $\varphi(t)$ without error.

Our simulations show that PED retains full link utilization with 32 MB of buffer memory in front of a 40 Gbps link, enabling on-chip buffer implementations with embedded DRAM (eDRAM) technology that was available in 2004 [6]. We argue that on-chip implementations should be possible today also for the 80 MB eDRAM that we consider ideal for a 100 Gbps link. Since more than 20% of the power consumed by a typical high-end router line card [18] is due exclusively to the off-chip placement of packet buffers [19], PED's contribution to energy efficiency can be indeed substantial. Energy gains can be even higher (up to 40%) in the packet processing boards of access nodes [20].

The buffer size reductions enabled by PED bring tremendous benefits to multi-bottleneck TCP flows: we present a simulation experiment derived from a production network, where the throughput of multi-bottleneck flows increases by three orders of magnitude when PED buffers replace conventional Tail-Drop buffers. We remark that the deployment of PED in network buffers does not conflict with the new types of TCP endpoint behavior that have recently gained popularity in the industry [12,14] and actually brings major benefits to them, in the form of faster data deliveries and network performance that is more predictable and stable.

We organize the paper as follows. In Section 2 we describe the RED algorithm in its basic formulation and discuss RED enhancements that have been proposed in the literature. We specify the PED algorithm in Section 3