International Journal of Electrical, Electronics and Computer Engineering **3**(2): 122-131(2014)
ISSN No. (Online): 2277-2626

Improve Transport-Layer Performance by Delaying Transmissions in Data Communication Networks

Kishor Ambekar*, Narendra P. Patil* and Dinesh S. Bhadane*
**Department of Electronics & Telecommunication, Sandip Polytechnic, Nashik, (MS), India*

(Corresponding author: Kishor Ambekar)
(Received 05 October, 2014 Accepted 10 November, 2014)

ABSTRACT: Packet losses in the network have a considerable performance impact on transport-layer throughput. For reliable data transfer, lost packets require retransmissions and thus cause very long delays. This tail of the packet delay distribution causes performance problems. There are several approaches to trading off networking resources up-front to reduce long delays for some packets (e.g., forward error correction, network coding). We propose packet pacing as an alternative that changes traffic characteristics favorably by adding intentional delay in packet transmissions. This intentional delay counters the principle of best effort but can reduce the burstiness of traffic and improve overall network operation – in particular in network with small packet buffers. As a result, pacing improves transport- layer performance, providing a tradeoff example where small amounts of additional delay can significantly increase connection bandwidth. We present a Queue Length Based Pacing (QLBP) algorithm that paces network traffic using a single queue and that can be implemented with small computational and memory overhead. We present a detailed analysis on delay bounds and the quantitative impact of QLBP pacing on network traffic. Through simulation, we show how the proposed pacing technique can improve connection throughput in small-buffer networks.

Keywords: Transport layer, traffic pacing, small-buffer net- work.

I. INTRODUCTION

Many data communication networks use a layered network architecture, where each layer implements different networking protocols [1]. The separation of networking functionality into layers simplifies the design of network protocols, but also implies that the performance that can be achieved within a protocol layer is highly dependent on the performance achieved by underlying layers. The main idea is to adjust the characteristics of network traffic at the edge of the network to ensure better performance in the core of the network. Specifically, we propose to introduce intentional delay in network layer transmissions to reduce the occurrence of traffic bursts, which have detrimental effects on transport layer performance as they can lead to packet loss due to buffer overflow. Our focus is on networks with small packet buffers optical packet-switched networks, wireless networks with low-performance nodes.

A. Packet Loss in Networks

One of the most problematic events for data transmissions in the network layer is a packet loss.

The two main causes for packet loss in networks are:

(i) Bit errors in physical layer: Bit errors in the physical layer most commonly occur in wireless transmissions due to interference, but can also occur in wired links. These bit errors cause checksums in the data link layer to fail, triggering a packet drop.

(ii) Congestion in network layer: Statistical multiplexing of network traffic implies that there are no guarantees about the available bandwidth on any given link. Thus, network traffic can congest the outgoing port of a router and cause transmission buffers to fill up. If a packet arrives at such a transmission queue when no more buffer space is available, then it is dropped.

B. Delay and Bandwidth Tradeoffs

There are several possible approaches to addressing the problem of reducing the impact of packet loss on the delay in transport layer communication. Fig. 1 illustrates how some of these techniques relate. The figure shows the amount of delay incurred at the transport layer versus the amount of bandwidth used at the transport layer. The main techniques noted in this figure.

Lossy transmission: Using lossy transmission protocols (e.g., User Datagram Protocol (UDP) [3]) places the bandwidth needs and delay close to the ideal lower bounds. Marginal amounts of additional bandwidth are necessary for packet headers and additional delay is incurred due to the packetized transmission of data.

Reliable transmission: The baseline protocol for reliable transmission is the Transmission Control Protocol (TCP) [4].Compared to UDP, TCP requires more bandwidth since some packets need to be retransmitted. It also incurs additional delay due to these retransmissions.

Network coding: There are several coding techniques to reduce packet loss in networks. To reduce bit errors, error correction coding can be used [5]. To avoid packet losses, transmission information can be spread across multiple paths in the network using network coding [6]. These techniques require additional bandwidth since they rely on redundant transmission of information. They also exhibit increased delay over a lossy transmission due to the need for data reconstruction at the receiver. However, these techniques incur less delay than TCP.

Traffic pacing: Traffic pacing is based on TCP, but uses traffic conditioning techniques in the network to reduce traffic bursts. By delaying some packet transmissions, less packet losses occur and thus less retransmissions are needed. Traffic pacing incurs a small additional delay, but uses less bandwidth than TCP since fewer retransmissions are necessary.

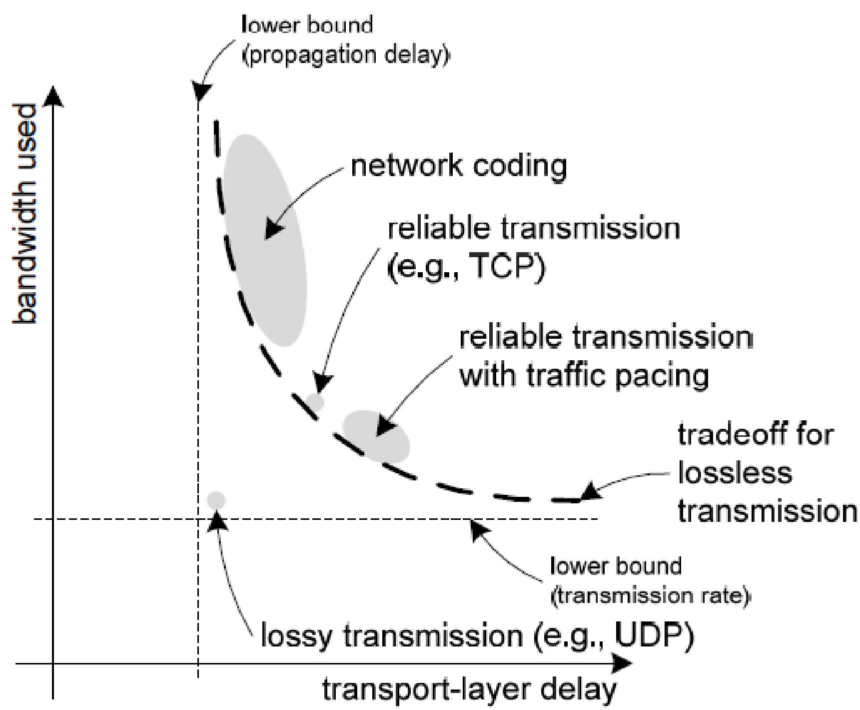

Fig. 1. Tradeoff of delay and bandwidth consumption for different lossless transmission techniques.

Overall, Figure 1 shows that there is a general tradeoff between bandwidth use and delay for lossless transmission in the transport layer.

C. Traffic Pacing in Networks

A key operational principle in the Internet is "best effort." Network resources are used when there is traffic to be sent and link schedulers on routers use "work-conserving" scheduling disciplines. This approach of not wasting opportunities to transmit packets intuitively seems to lead to the best possible network performance. In our work, we present a traffic pacing technique that can reduce the burstiness of traffic and improve the throughput of transport layer TCP connections. The design of our traffic pacing system is particularly suitable for emerging network architectures for two reasons:

The remainder of this paper is organized as follows. Sec- tion II introduces the network architecture for pacing and details on the Queue Length Based Pacing algorithm We present a novel pacing algorithm that decreases the burstiness of network traffic by delaying packets based on the length of the local packet buffer.. Analytical results present a formal analysis of QLBP that provides delay bounds and a quantitative understanding of the effect of traffic smoothing.are presented in Section III. Simulation results on the effectiveness of QLBP & improvements of transport layer performance in small-buffer networkare presented in Section IV. Section V discusses related work, and Section VI summarizes and concludes this paper.

II. QUEUE-LENGTH BASED PACING

The pacing technique that we propose in this work aims to reduce the burstiness of network traffic.

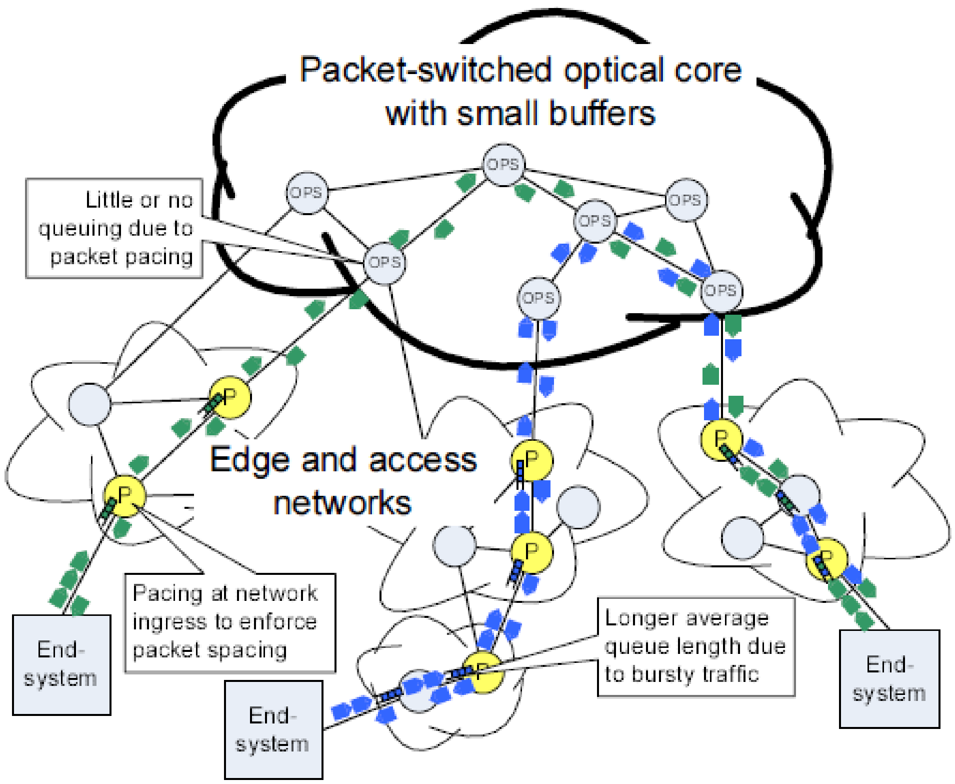

Fig. 2. Network architecture with opportunistic pacing.

A. TCP Burstiness

TCP is the most widely used transport layer protocol in the Internet. Its traffic characteristics have considerable impact on the operation of the network. The TCP protocol can pace itself due to ACK-clocking, where acknowledgments are spaced out by the bottleneck link. As a result, packets sent in the congestion avoidance phase are spaced by acknowledgement arrivals. However, as pointed out by Aggarwal *et al.* in [8], a number of factors inherent to TCP can cause burstiness in the behavior of a TCP flow, such as slow start, lost packet retransmission, ACK-compression and multiplexing (for details, see [8]). Even though the impact of retransmissions of lost packets can somehow be mitigated by enabling TCP selective acknowledgement (SACK) options [9], [10], the negative impact of ACK-compression and multiplexing might become even worse in the future Internet with much larger bandwidth.

B. Pacing Network Architecture

To reduce the burstiness of TCP traffic (and any other traffic), we propose a pacing technique that delays some packet transmissions. This pacing process can be implemented on the outgoing interfaces of routers. We envision an overall network architecture as shown in Figure 2. Pacing is deployed on several (but not necessarily all) nodes in the network.

Since pacing cannot be practically implemented on optical packet switches, it is constrained to non-optical routers. These routers have sufficiently large buffers that allow moderate traffic bursts to be absorbed and paced without packet loss.

C. Queue Length Based Pacing System

The general ideal of Queue Length Based Pacing (QLBP) is to dynamically adjust the sending rate of a queue according to the queue length, rather than to send packets at a constant rate. The structure of a QLBP system is shown in Fig. 3, and the major notation used in this paper is summarized in Table 1. The Fig. 4,5 shows a single input and output, but the concept can be applied to routers with any number of ports. A QLBP system includes a delay queue and a rate controller, and has three parameters: μ_{max}, μ_{min} and Q_{max}.

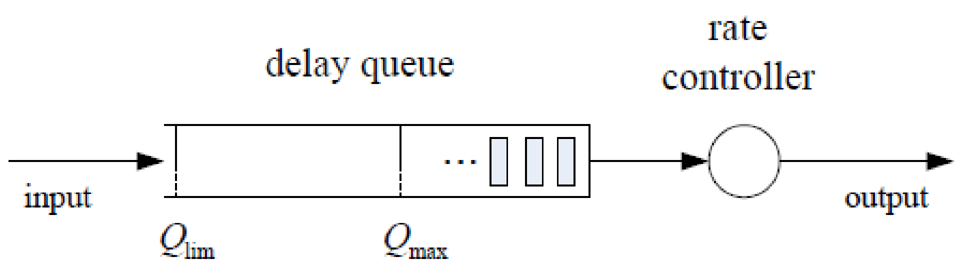

Fig. 3. QLBP system for router buffer.

TABLE I

NOTATION USED IN THIS PAPER

Defined in Section II-C	
$q(t)$	instantaneous length of the delay queue at time t
$\lambda(t)$	arrival rate of input traffic at time t
$\mu(t)$	output rate of the rate controller at time t
$\mu_{\max}$	maximum rate at which the rate controller transmits packets when pacing is enabled
$\mu_{\min}$	minimum rate at which the rate controller transmits packets when pacing is enabled
$Q_{\max}$	(pacing cutoff queue length) queue length beyond which no pacing delays are introduced by the pacer
$Q_{\lim}$	buffer size of the delay queue
C	capacity of the outgoing link
Defined in Section III-A	
d	pacing delay
d_{pacer}	delay a packet experiences when passing through a QLBP pacer
d_{FIFO}	delay a packet experiences when passing through a FIFO queue
Defined in Section III-B	
N_1	ON Poisson counter of the Markov ON-OFF modeled process
N_2	OFF Poisson counter of the Markov ON-OFF modeled process
r_1	rate of ON Poisson counter N_1
r_2	rate of OFF Poisson counter N_2
h	peak rate during ON periods

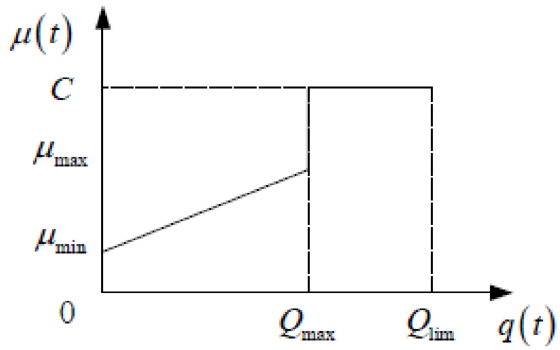

Fig. 4. Pacing rate $\mu(t)$ vs. queue length $q(t)$.

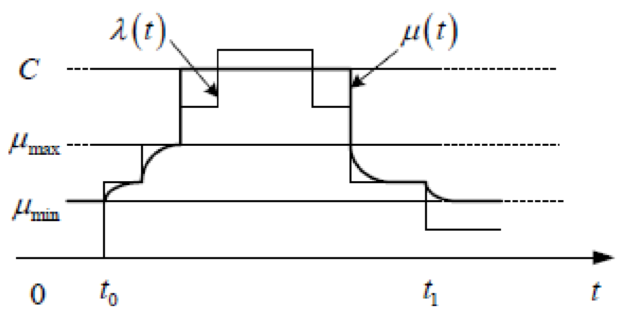

Fig. 5. Relationship between $\mu(t)$ and $\lambda(t)$.

D. Pacing Delay

One of the key aspects of any pacing algorithm is how the inter-packet pacing delay is determined. In TCP pacing the inter-packet pacing delay is roughly set to the ratio of the current RTT to the congestion window size. In the pacing scheme proposed by Sivaranman [12], the inter-packet pacing delay is calculated based on the packet arrival curve and the packet deadline curve within the same pacing interval. In QLBP, we determine this delay based on some very simple rules:

•If the pacing queue increases due to a higher input traffic rate, QLBP intentionally lowers the introduced pacing delay. This rule ensures that the link can be fully utilized under heavy load.

•For packets that arrive at a rate lower than μ_{min}, they do not get delayed. This rule ensures that pacing is only activated when packets arrive at a certain high rate. Based on these rules, we have designed the queue length dependent output rate $\mu(t)$ as follows:

$$\mu(t) = \begin{cases} \frac{\mu_{\max}-\mu_{\min}}{Q_{\max}} q(t) + \mu_{\min}, & 0 \le q(t) \le Q_{\max}, \\ C, & \text{otherwise.} \end{cases} \quad (1)$$

III. ANALYSIS

In this analysis, we show two important results: (1) the pacing delay depends on the incoming traffic rate and is upper- bounded by a constant (depending on QLBP parameters), thus limiting delay introduced by QLBP, and (2) the effectiveness of QLBP on reducing burstiness in network traffic can be quantified by evaluating the variance of the instantaneous traffic rate in the context of a fluid model.

A. Delay Guarantee

To show the bounds on delay, we first give a precise definition of pacing delay. Definition 1: For a packet, the pacing delay, denoted by d, is defined as the time difference of dpacer − dF IF O , where dpacer and dF IF O represent the delay the packet experiences when passing through a QLBP queue and an ordinary FIFO (drop-tail) queue, respectively. Theorem 1: Given parameters μ_{max}, μ_{min} and Q_{max}, for an input traffic with rate λ, the pacing delay d in steady state depends on λ and is upper bounded by a constant $Q\mu_{max}$.

B. Reduction of Traffic Burstiness

We quantitatively analyze the pacing effect of a QLBP sys- tem in two aspects: (1) how quickly a QLBP system responds to the change in the input rate, (2) how a QLBP system smoothes the input traffic by reducing the auto-covariance. Even though the modeling and analysis are established based on some simple toy traffic models, they still unveil the fundamental natures of QLBP.

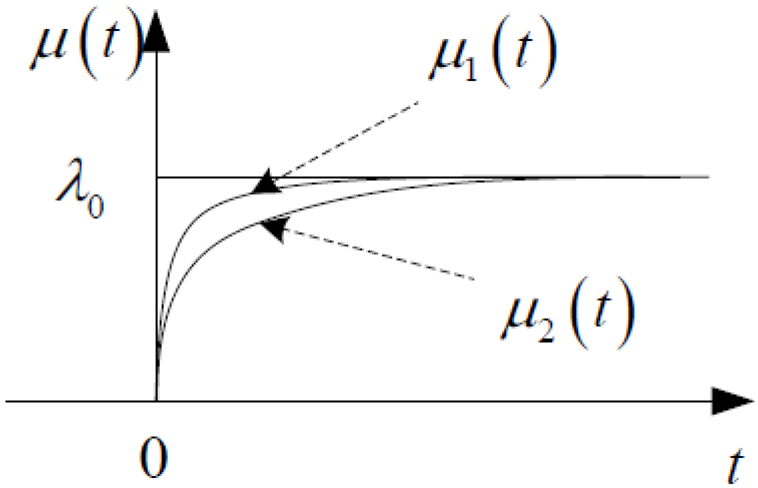

Fig. 6. Relationship between $\mu(t)$ and changes to $\lambda(t)$.

The larger α, the faster μ(t) converges to λ(t), as shown in Figure 6. Under the same initial condition, μ1(t) with a larger αconverges to λ0 faster than μ2(t) does.Reduction of Auto-covariance: Next we propose a fluid model that describes the dynamics of the QLBP system. Our goal is to provide insights into how the QLBP system smoothes traffic in term of reducing auto-covariance of net- work traffic rate.In this case, once the queue becomes nonempty, it remains so, though it may be very arbitrarily close to zero.

C. Parameter Selection

Given a QLBP system of (Qlim, C), an important question that remains to be answered is how the parameters in QLBP are chosen. We formulate it as an optimization problem,

min B = F (λ(t), μmax, μmin,Qmax),

IV. SIMULATION RESULTS

The reduction of burstiness in network traffic translates into increased throughput performance for TCP traffic. In this section, we present results from a QLBP prototype implementation on the Open Network Laboratory (ONL) [21]. We also show results from simulation using larger-scale network configurations in ns-2 [22]. These results (1) show the pacing effect of QLBP on TCP and UDP flows, (2) validate the adaptive pacing delay introduced by QLBP, (3) quantitatively evaluate QLBP effectiveness on reducing burstiness of traffic in terms of the variance of the instantaneous traffic rate, (4) compare QLBP performance with TCP pacing in improving link utilization, and (5) show that the end-to-end delay distribution of paced traffic has a smaller tail.

A. Impact of QLBP on Single TCP and UDP Flows

This set of experiments is conducted using prototype implementation of QLBP in the Open Network Laboratory. More details on this implementation of QLBP can be found in [23]. A QLBP pacer is implemented as an ONL plugin and applied at the ingress port of router 1. A TCP or UDP flow is transmitted between the sender and the receiver.

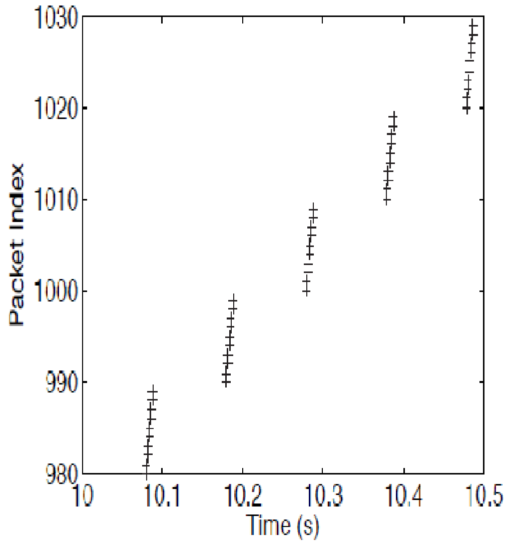

Fig.7. Arrival of TCP packets without pacing.

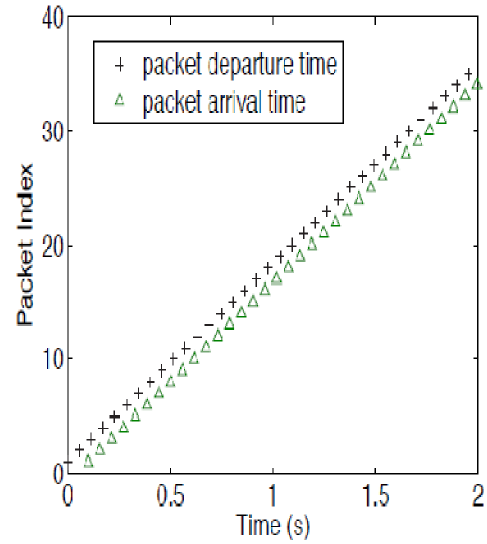

Fig. 8. Arrival of departure time of 200 Kbps CBR traffic.

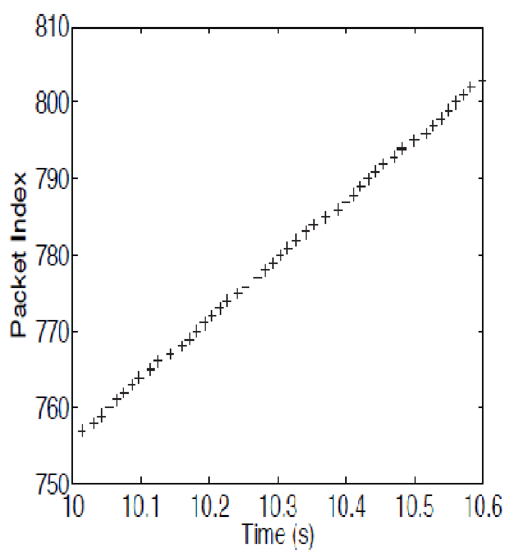

Fig. 9. Arrival process of TCP packets with pacing.

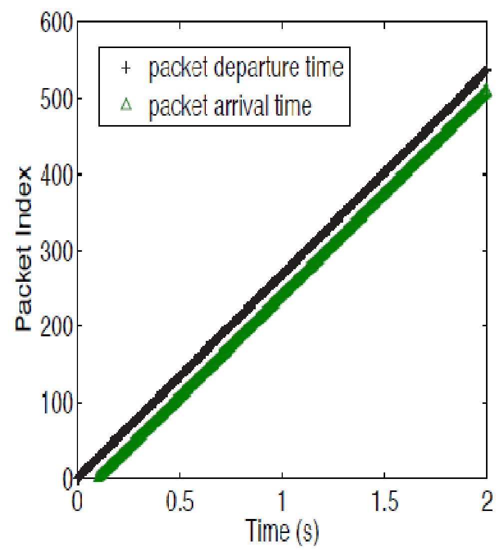

Fig. 10. Arrival of departure time of 3Kbps CBR traffic.

B. Adaptive Pacing Delay

In this 2 experiment, we send CBR traffic through a QLBP pacer and examine the pacing queue length *Qp* and the pacing delay *Dp*. Figure 8 shows the topology. A CBR traffic with rate λ flows from node 0 to node 2. A QLBP pacer is placed at node 1 to pace the traffic towards node 2. The parameters are set as follows. *BW*1 = *BW*2 = 15Mbps, and *Delay*1 = *Delay*2 = 10ms. μmax = 10Mbps, μmin = 2Mbps, *Q*max = 10pkts and *Q*lim = 1000pkts. UDP packet size is 1000 Bytes.

C. Pacing Effectiveness

We are interested in how QLBP affects traffic burstiness. The metric of concern in this ns-2 experiment is the coefficient of variation of the traffic rate, which is used in [12] to measure the extent to which traffic is bursty.

D. Improvement on Link Utilization

In this sub-section we investigate the impact of short-term burstiness on a non-bottleneck link in terms of link utilization. This set of experiments is used in [7] to show the performance improvement of TCP pacing in small buffer networks.

V. RELATED WORK

The impacts of small buffers on transport-layer network performance have been studied in the context of real-time traffic and TCP traffic [7], [11], [12], [26]–[28]. Interestingly, the results of these studies are not conclusive. On one hand, it has been shown that small buffers significantly degrade network performance with ordinary TCP sessions by causing packet drop more frequently. Enachescu et al. [7] showed that a 80% workload consisting of long-lived TCP sessions only achieves a 20% link utilization when the buffer size of the shared link is 10 packets. Sivaramman etal. [12] demonstrated that “a 10Gbps optical packet switching (OPS) node with 10 to 20 packets can experience significant losses even at low (40%) to moderate (60% for long-range dependent or 80% for short-range dependent) traffic loads.” On the other hand, theoretical analyses and empirical results show that small buffers are feasible for core routers through which tens of thousands of TCP sessions flow [7], [11], [26]– [28]. Enachescu et al. [7] argued that $O(\log W)$ buffers are sufficient for high throughput, where W is congestion window size of each flow, and router buffer can even be reduced to a few dozen packets if a small amount of link utilization is sacrificed. Gu et al. [11] demonstrated that more than 90% link utilization is achievable in a 1–10 Gbps bottleneck link with a buffer of 20 packets. Lakshmikantha et al. [28] further showed that $O(1)$ buffer sizes (20 packets) are sufficient for good performance with no loss of link utilization when considering the impact of file arrivals and departures. We note that all high performance results are achieved only when TCP sessions are paced by either some rate-control mechanism (i.e., TCP pacing) or access links with capacities much slower than the bottleneck link. The main concern with the small buffer core networks is the high packet loss probability due to the small buffer size and the bursty behavior of TCP. Several techniques are proposed to lower the drop probability in small buffer networks by smoothing network traffic. Packet pacing finds its roots in the explicit rate control non-TCP protocols, which send data at a fixed rate irrespective of the receipt of acknowledgments [29], [30]. Pacing was used in the TCP context to correct the compression of acknowledgements due to cross traffic [31], to avoid slow start [32], [33], after packet loss [34], or when an idle connection resumes [35]. Aggarwal et al. [8] concluded that pacing improves throughput in some cases but in general decreases performance. The poor performance of pacing is attributed mostly to “synchronized drops” and packet delays being misinterpreted as congestion. In addition to TCP pacing, there have been several proposals for resolving packet drops in small buffer networks [12], [36]– [39]. The work by Alparslan et al. [36] shares a very similar idea with our, i.e., turning the pacing rate based on the buffer occupancy, and the effect of the pacing is evaluated in a largescale hypothetic network. The work by Sivaraman et al. [12] stems from previous works on traffic conditioners for video transmission, called traffic conditioning *off-line* [40]. They proposed an on-line version of traffic conditioner based on this traffic conditioning *off-line*. The approaches in [37]–[39] rely on the global network-wide coordinated scheduling. Unlike the above pacing-based approaches, Vishwanath et al. proposed to recover lost packets by using the packet-level forward error correction (FEC) scheme [41]. Their codingbased approach works based on an observation that “loss at core links is due to contention, not congestion.” Through simulation they show the efficiency of the FEC-based approach

VI. SUMMARY

Our work presents a novel view on the tradeoff between link bandwidth and packet delay. Instead of using an error correction or network coding approach where more bandwidth is used to avoid packet losses, we proposed to delay packet transmissions to reduce the burstiness of traffic and thus reduce packet losses in small-buffer networks. We present Queue Length Based Pacing, which is a pacing technique that uses a single pacing queue on router ports and adapts its sending rate based on the amount of traffic that is buffered at that port. Our analysis shows that pacing delay due to QLBP is bounded andthat the variance of the instantaneous traffic rate is reduced. We show the effectiveness of QLBP through a prototype implementation and simulation. Specifically, we show that TCP connections in a small-buffer network with QLBP pacing achieve higher link utilization than in non-paced networks. Therefore, we believe that QLBP is an effective approach to improving the operation of networks and improving the effective bandwidth of connections at the cost of only small amounts of additional delay.

REFERENCES

[1] *International Standard ISO/IEC 7498-1*, 2nd ed., International Organization for Standardization / International Electrotechnical Commission, Geneve, Switzerland, Nov. 1994.

[2] M. Shifrin and I. Keslassy, “Small-buffer networks,” *Computer Networks*, vol. 53, no. 14, pp. 2552–2565, Sep. 2009.

[3] J. Postel, “User Datagram Protocol,” Information Sciences Institute, RFC 768, Aug. 1980.

[4] , “Transmission Control Protocol,” Information Sciences Institute, RFC 793, Sep. 1981.

[5] T. K. Moon, *Error Correction Coding: Mathematical Methods and Algorithms*. Wiley-Interscience, Jun. 2005.

[6] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, “Network information flow,” *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[7] M. Enachescu, Y. Ganjali, A. Goel, N. McKeown, and T. Roughgarden, “Routers with very small buffers,” in *Proc. Twentyfifth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2006)*, Barcelona, Spain, Apr. 2006.

[8] A. Aggarwal, S. Savage, and T. Anderson, “Understanding the performance of TCP pacing,” in *Proc. IEEE INFOCOM 2000*, Tel Aviv, Israel, Mar. 2000, pp. 1157–1165.

[9] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, “RFC 2018: Tcp selective acknowledgement options,” Apr. 1996.

[10] S. Floyd, J. Mahdavi, M. Mathis, and A. Romanow, “RFC 2883: An extension to the selective acknowledgement (SACK) option for tcp,” Jul. 2000.

[11] Y. Gu, D. Towsely, C. V. Hollot, and H. Zhang, “Congestion control for small buffer high speed networks,” in *Proc. IEEE INFOCOM 07*, Anchorage, Alaska, May 2007, pp. 1037–1045.

[12] V. Sivaraman, H. Elgindy, D. Moreland, and D. Ostry, “Packet pacing in short buffer optical packet switched networks,” in *Proc. IEEE INFOCOM 06*, Spain, Apr. 2006.

[13] H. Heffes and D.Lucantoni, “A markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance,” in *IEEE J. Sel. Areas Commun.*, Sep. 1986, pp. 856–868.

[14] I. Nikolaidis and I. Akyildiz, “Source characterization and statistical multiplexing in ATM networks,” Georgia Tech., Technical Report GITCC 92-24, 1992.

[15] R. W. Brockett, W. Gong, and Y. Guo, “Stochastic analysis for fluid queueing systems,” in *IEEE CDC*, Dec. 1999.

[16] Y. Huang, Y. Liu, W. Gong, and D. Towsley, “Two-level Stochastic Fluid Tandem Queuing Model for Burst Impact Analysis,” in *IEEE CDC*, Dec. 2007.

[17] Y. Wu, W. Gong, and D. Towsley, “Analysis of abstract simulation via stochastic differential equation models,” in *IEEE CDC ’03*, Dec 2003.

[18] C. V. Hollot, Y. Liu, V. Misra, and D. Towsley, “Unresponsive Flows and AQM Performance,” in *Proc. IEEE INFOCOM*, Apr 2003.

[19] W. Willinger, M. Taqqu, R. Sherman, and D. Wilson, “Self-similarity through highvariability: statistical analysis of ethernet lan traffic at the source level,” pp. 100–113, Aug. 1995.

[20] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson, “Selfsimilarity through high-variability: Statistical analysis of ethernet lan traffic at the source level,” *IEEE/ACM Trans. Netw.*, vol. 5, pp. 71–86, 1997.

[21] J. DeHart, F. Kuhns, J. Parwatikar, J. Turner, C. Wiseman, and K. Wong, “The open network laboratory: a resource for networking research and education,” *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 5, pp. 75–78, Oct. 2005.

[22] The network simulator - ns-2, http://www.isi.edu/nsnam/ns/.

[23] Y. Cai, S. Hanay, and T. Wolf, “Practical packet pacing in small-buffer networks,” in *ICC ’09*, Dresden, Germany, Jun. 2009.

[24] M. C. Weigle, P. Adurthi, F. H.-C. K. Jeffay, and F. D. Smith, “Tmix: A tool for generating realistic tcp application workloads in ns-2,” *SIGCOMM Computer Communication Review*, vol. 36, no. 3, pp. 67–76, 2006.

[25] inbound.cvec and outbound.cvec, http://www.cs.odu.edu/ netsim/ TrafGen/Traces-tmix-ccr06.

[26] D. Wischik and N. McKeown, “Part I: Buffer sizes for core routers,” A*CM SIGCOMM Comput Commun Rev*, pp. 75–78, Jul. 2005.

[27] G. Raina, D. Towsely, and D. Wischik, “Part II: Control theory for buffer sizing,” *ACM SIGCOMM Comput Commun Rev*, pp. 79–82, Jul. 2005.

[28] A. Lakshmikantha, R. Srikant, and C. Beck, “Impact of File Arrivals and Departures on Buffer Sizing in Core Routers,” in *Proc. IEEE INFOCOM*, 2008.

[29] D. D. Clark, M. M. Lambert, and L. Zhang, “NETBLT: A high throughput transport protocol,” *ACM SIGCOMM Comp. Comm. Rev.*, vol. 17, pp. 353–359, Aug. 1987.

[30] F. Bonomi and K. Fendick, “The rate based flow control framework for the available bit rate ATM service,” *IEEE Network*, pp. 25–39, 1998.

[31] L. Zhang, S. Shenker, and D. D. Clark, "Observations on the dynamics of a congestion control algorithm: the effects of two way traffic," in *Proc. ACM SIGCOMM 91*, Zurich, Switzerland, Sep. 1991, pp. 133– 147.

[32] M. Aron and P. Druschel, "TCP: Improving startup dynamics by adaptive timers and congestion control," Rice University, Technical Report TR98-318, 1998.

[33] V. N. Padmanabhan and R. H. Katz, "TCP Fast Start: A technique for speeding up web transfers," in *Proc. IEEE GLOBECOMM*, Sydney, Australia, Nov. 1998.

[34] J. Hoe, "Start-up dynamics of TCP's congestion control and avoidance schemes," Masterthesis, MIT, Jun. 1995.

[35] V. Visweswaraiah and J. Heidermann, "Improving restart of idle TCP connections," University of Southern California, Technical Report TR97-661, 1997.

[36] O. Alparslan, S. Arakawa, and M. Murata, "Node pacing for optical packet switching," in *Proc. Photonics in Switching, 2008*, Sapporo, Aug. 2008.

[37] J. Naor, A. Rosen, and G. Scalosub, "Online time-constrained scheduling in linear networks," in *Proc. IEEE INFOCOM 05*, Miami, FL, Mar. 2005.

[38] M. Adler, S. Khanna, R. Rajaraman, and A. Rosen, "Time-constrained scheduling of weighted packets on trees and meshes," *Algorithmica*, vol. 36, no. 2, pp. 123–152, 2003.

[39] M. Adler, A. L. Rosenberg, R. K. Sitaram, and W. Unger, "Scheduling time-constrained communication in linear networks," *Theoretical Comp. Sc*, vol. 35, no. 6, pp. 559–623, 2002.

[40] J. D. Salehi, Z. Zhang, J. Kurose, and D. Towsley, "Supporting stored video: Reducing rate variability and end-to-end resource requirements through optimal smoothing," *IEEE/ACM Trans. Netw.*,, vol. 6, no. 4, pp. 397–410, 1998.

[41] A. Vishwanath, V. Sivaraman, M. Thottan, and C. Dovrolis, "Enabling a bufferless core network using edge-to-edge packet-level fec," in *Proc. IEEE INFOCOM 10*, San Diego, CA, Mar. 2010.