Report Number: WUCSE-2007-42

2007

Lower Bounds on Queuing and Loss at Highly Multiplexed Links

Maxim Podlesny and Sergey Gorinsky

Explicit and delay-driven congestion control protocols strive to preclude overflow of link buffers by reducing transmission upon incipient congestion. In this paper, we explore fundamental limitations of any congestion control with respect to minimum queuing and loss achievable at highly multiplexed links. We present and evaluate an idealized protocol where all flows always transmit at equal rates. The ideally smooth congestion control causes link queuing only due to asynchrony of flow arrivals, which is intrinsic to computer networks. With overprovisioned buffers, our analysis and simulations for different smooth distributions of flow interarrival times agree that minimum queuing at a... Read complete abstract on page 2.

Lower Bounds on Queuing and Loss at Highly Multiplexed Links

Maxim Podlesny and Sergey Gorinsky

Complete Abstract:

Explicit and delay-driven congestion control protocols strive to preclude overflow of link buffers by reducing transmission upon incipient congestion. In this paper, we explore fundamental limitations of any congestion control with respect to minimum queuing and loss achievable at highly multiplexed links. We present and evaluate an idealized protocol where all flows always transmit at equal rates. The ideally smooth congestion control causes link queuing only due to asynchrony of flow arrivals, which is intrinsic to computer networks. With overprovisioned buffers, our analysis and simulations for different smooth distributions of flow interarrival times agree that minimum queuing at a fully utilized link is O(sqrt(N)), where N is the number of flows sharing the link. This result raises concerns about scalability of any congestion control. However, our simulations of the idealized protocol with small buffers show its surprising ability to provide bounded loss rates regardless of the number of flows. Finally, we experiment with RCP (Rate Control Protocol) to examine how existing practical protocols compare with our idealized scheme in small-buffer settings.

2007-42

Lower Bounds on Queuing and Loss at Highly Multiplexed Links

Authors: Maxim Podlesny and Sergey Gorinsky

Corresponding Author: podlesny@arl.wustl.edu, gorinsky@arl.wustl.edu

Abstract: Explicit and delay-driven congestion control protocols strive to preclude overflow of link buffers by reducing transmission upon incipient congestion. In this paper, we explore fundamental limitations of any congestion control with respect to minimum queuing and loss achievable at highly multiplexed links. We present and evaluate an idealized protocol where all flows always transmit at equal rates. The ideally smooth congestion control causes link queuing only due to asynchrony of flow arrivals, which is intrinsic to computer networks. With overprovisioned buffers, our analysis and simulations for different smooth distributions of flow interarrival times agree that minimum queuing at a fully utilized link is O(sqrt(N)), where N is the number of flows sharing the link. This result raises concerns about scalability of any congestion control. However, our simulations of the idealized protocol with small buffers show its surprising ability to provide bounded loss rates regardless of the number of flows. Finally, we experiment with RCP (Rate Control Protocol) to examine how existing practical protocols compare with our idealized scheme in small-buffer settings.

Type of Report: Other

Lower Bounds on Queuing and Loss at Highly Multiplexed Links

Maxim Podlesny and Sergey Gorinsky

Technical Report WUCSE-2007-42
Department of Computer Science and Engineering, Washington University in St. Louis
One Brookings Drive, St. Louis, MO 63130-4899, USA
{*podlesny,gorinsky*}*@arl.wustl.edu*

July 2007

***Abstract*—Explicit and delay-driven congestion control protocols strive to preclude overflow of link buffers by reducing transmission upon incipient congestion. In this paper, we explore fundamental limitations of any congestion control with respect to minimum queuing and loss achievable at highly multiplexed links. We present and evaluate an idealized protocol where all flows always transmit at equal rates. The ideally smooth congestion control causes link queuing only due to asynchrony of flow arrivals, which is intrinsic to computer networks. With overprovisioned buffers, our analysis and simulations for different smooth distributions of flow interarrival times agree that minimum queuing at a fully utilized link is $O(\sqrt{N})$, where N is the number of flows sharing the link. This result raises concerns about scalability of any congestion control. However, our simulations of the idealized protocol with small buffers show its surprising ability to provide bounded loss rates regardless of the number of flows. Finally, we experiment with RCP (Rate Control Protocol) to examine how existing practical protocols compare with our idealized scheme in small-buffer settings.**

I. Introduction

Smooth fair lossless transmission at high bitrates has been an aspiration for many explicit [1]–[6] and delay-driven [7]–[9] congestion control protocols. Smoothness of transmission is particularly important for interactive multimedia and other applications that would suffer from excessive queuing delays at network links. However, since discovery of a fair sending rate is fraught with packet bursts and other causes of link queuing, smooth transmission conflicts with another goal of responding promptly to changes in network conditions [10], [11].

Smooth transmission is also relevant to link buffer sizing, which recently attracted significant attention. In the context of TCP (Transmission Control Protocol) congestion control [12], [13], various proposals disagree on how to size the buffer with respect to the number of flows sharing the link. Arguing that aggregate oscillations of TCP traffic subside as the number of flows increases, one view maintains that a small buffer suffices for a highly multiplexed link [14]–[16], even if the buffer accommodates only up to 20 packets [17]. However, since TCP suffers from high loss rates and frequent retransmission timeouts when the network path restricts each TCP flow to less than few packets per round-trip time (RTT) [18], [19], alternative guidelines prescribe keeping the buffer size proportional to the number of flows in some network settings [20], [21], i.e., argue for even larger buffers than the traditionally recommended bitrate-delay product [22], [23]. Furthermore, it has also been argued that different congestion control is needed for networks with small link buffers [24]–[26].

The lack of agreement on many issues in congestion control has sound reasons. Whereas practical congestion control protocols tend to be multimodal and complex, precise comprehensive analysis of their performance is hard. On the other hand, experimental evaluations face scalability challenges. In particular, while even a single packet-level simulation of transient and steady states for a highly multiplexed link might consume long time, congestion control studies rarely report reliable results for more than few hundred flows [27].

In this paper, we present a model for investigating lower bounds on queuing and loss under smooth congestion control with overprovisioned and small buffers. We consider an idealized protocol where all flows always transmit at equal rates. The ideally smooth transmission does not eliminate queuing altogether because packets of different flows might overlap at a link due to asynchronous arrivals of the flows. For example, even if the constant-rate flows underutilize the link on average, a queue arises when packets from multiple flows arrive to the link simultaneously. The asynchrony of flow arrivals constitutes the chief distinction of our model from the perfect TDM (Time Division Multiplexing) which avails the link to each packet immediately upon the packet arrival. Such asynchrony is intrinsic to congestion control.

A prominent aspect of our model is its simplicity which makes analysis tractable and experiments scalable. In particular, our simulation methodology captures the steady-state queuing for N concurrent flows exactly by examining only $2N$ packets. The low overhead enables us to assess expected steady-state performance reliably by conducting extensive simulations with up to 5,000 concurrent flows and repeating each experiment 1,000 times.

With overprovisioned buffers, our analysis and simulations for different smooth distributions of flow interarrival times agree that minimum queuing at a fully utilized link is $O(\sqrt{N})$.

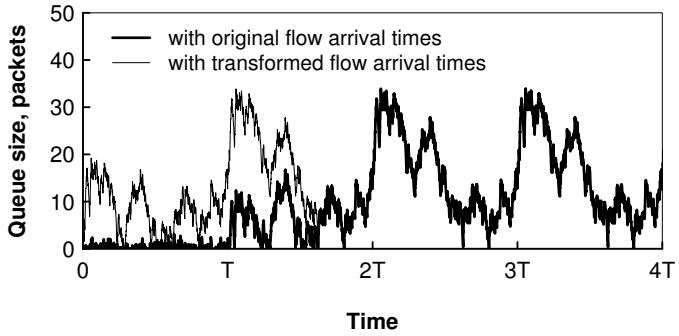


Fig. 1. Identical steady-state queuing in the overprovisioned buffer of a fully utilized link under original and transformed arrival times of $N = 1,000$ flows. The flows originally arrive according to a Poisson process with average rate $\frac{N}{2.05T}$, where $T = 80$ ms is the steady-state period. To highlight the longer duration of the transient stage under original arrival times, the rate is chosen lower than in the flow interarrival time distributions examined in this paper.

This finding implies that no congestion control protocol is able to avoid packet losses with a constant buffer and arbitrarily high number of flows. However, our studies of the idealized protocol with small buffers abate the above concerns about congestion control scalability by showing the surprising ability of our protocol to provide bounded loss rates regardless of the number of flows.

While the presented model yields fresh interesting insights into queuing and buffer sizing at highly multiplexed links, our main results are lower bounds. To examine how existing practical protocols compare with our idealized scheme, we simulate RCP (Rate Control Protocol) [2] in ns-2 [28] with small buffers and observe significantly larger loss rates. Further studies are needed to establish whether the derived lower bounds are achievable by practical congestion control protocols, which might face synchronization of flows and other extra sources of queuing.

The rest of the paper is organized as follows. Section II clarifies our model. Section III shows that neither the link bitrate nor the packet size affects the steady-state queuing under our idealized protocol. Section IV derives bounds on queuing in overprovisioned buffers. Section V supplements the analysis with extensive simulations. Section VI reports loss rates under our idealized protocol with small buffers. Section VII presents our RCP experiments. Finally, Section VIII concludes the paper with a summary of our findings.

II. Model

We model a steady-state scenario where N flows share a bottleneck link with bitrate C and FIFO (First-In First-Out) buffer. We denote arrival time of flow i as t_i, where $i = 1, \ldots, N$. Without loss of generality, we assume $t_1 = 0$. We refer to time between arrivals of flows $i-1$ and i as δ_i:

$$\delta_i = t_i - t_{i-1}. \tag{1}$$

Average utilization of the link by the flows is U, where $0 < U \leq 1$. Each flow transmits packets of size S periodically at the same constant bitrate R equal to:

Distribution name	*Mean,* μ	*Variance,* σ^2
Uniform	$\frac{D}{U}$	$\frac{1}{3}\left(\frac{D}{U}\right)^2$
Exponential	$\frac{D}{U}$	$\left(\frac{D}{U}\right)^2$
Pareto, $k = 2.1$	$\frac{D}{U}$	$\frac{1}{k(k-2)}\left(\frac{D}{U}\right)^2$

Fig. 2. Considered distributions of flow interarrival times.

$$R = \frac{U \cdot C}{N}. \tag{2}$$

Hence, subsequent packets within any flow are separated by the same time interval T:

$$T = \frac{N \cdot S}{U \cdot C} = \frac{N \cdot D}{U} \tag{3}$$

where D is packet transmission delay, i.e., the amount of time it takes to transmit one packet into the bottleneck link:

$$D = \frac{S}{C}. \tag{4}$$

The considered pattern of packet transmissions is the smoothest possible under asynchronous congestion control where distributed senders of different flows do not deliberately schedule packets to arrive to a shared link at non-overlapping times. Such smoothest congestion control is an idealized protocol because any real protocol consumes time and creates bursts in order to discover a new fair rate after a change in network conditions. Once again, our rationale for examining this idealized protocol is to uncover fundamental limitations of congestion control with respect to minimum queuing and loss achievable under any practically realizable congestion control algorithm.

With the ideally smooth congestion control, queuing arises due to asynchrony of flow arrivals and hence potential overlap of packets from different flows. After the last flow arrives, imperfect alignment of the flows creates a queue oscillation pattern that repeats with period T. Figure 1 illustrates the periodic queue oscillations in the steady state.

While the flow arrival process is clearly an important aspect of our model, two factors make flow arrivals difficult to model realistically. First, the problem of Internet load modeling is far from being settled [29]–[32]. In particular, there is no universal agreement on how to model flow arrivals in different Internet applications [33]. Second, while any practical approximation of our idealized congestion control will affect alignment of packets on the shared link, it is hard to predict this impact and reflect it accurately in our model. Our general approach to handling this uncertainty is to consider a variety of flow arrival distributions, with an emphasis on smooth distributions because we are primarily interested in lower bounds on queuing and loss.

Distribution name	*Probability $\theta(i)$ that the i-th packet encounters a queue size longer than Q*	*Lower bound Q_{min} on the queue size for*		
		top θ packets	*top 1% packets*	*top 5% packets*
Uniform	$\frac{1}{2}\left(1-\text{erf}\left(\sqrt{\frac{3}{2i}}Q\right)\right)$	$L_\theta\sqrt{\frac{2}{3}}\sqrt{N}$	$0.89\sqrt{N}$	$0.49\sqrt{N}$
Exponential	$\frac{1}{2}\left(1-\text{erf}\left(\frac{Q}{\sqrt{2i}}\right)\right)$	$L_\theta\sqrt{2}\sqrt{N}$	$1.54\sqrt{N}$	$0.85\sqrt{N}$
Pareto, $k>2$	$\frac{1}{2}\left(1-\text{erf}\left(\sqrt{\frac{k(k-2)}{2i}}Q\right)\right)$	$L_\theta\sqrt{\frac{2}{k(k-2)}}\sqrt{N}$	$\frac{1.54}{\sqrt{k(k-2)}}\sqrt{N}$	$\frac{0.85}{\sqrt{k(k-2)}}\sqrt{N}$

Fig. 3. Lower bounds on queuing for top θ packets in the overprovisioned buffer of a fully utilized link.

We consider three smooth distributions of flow interarrival times: Exponential, Uniform, and Pareto. All three distributions have the same average value:

$$\mu = \frac{D}{U} = \frac{T}{N}, \tag{5}$$

i.e., the N flows are expected to arrive over a time interval which has the same duration T as the period of the steady-state queue oscillations. What distinguishes the distributions is their variances summarized in Figure 2:

- *Uniform* interarrival times are distributed uniformly between 0 and 2μ.
- *Exponential* interarrival times are generated by a Poisson process with average arrival rate $\frac{1}{\mu}$.
- *Pareto* interarrival times follow the Pareto distribution with mean μ and index $k = 2.1$. While the choice of $k = 2.1$ is due to our interest in smooth distributions, we also report results for other values of k, including $k < 2$.

For the link buffer, we examine both overprovisioned and small settings. While overprovisioned buffers are large enough to store and forward all arriving packets without loss, packet losses are possible with small buffers.

With overprovisioned buffers, the primary metric of performance for flow i is *queue size* q_i measured in packets:

$$q_i = \frac{d_i}{D} \tag{6}$$

where d_i is the queuing delay experienced by packets of the flow in the steady state.

In small-buffer settings, we quantify performance with a *loss rate* defined as a fraction of packets discarded in the steady state due to buffer overflow.

III. Link bitrate and packet size

Bottleneck link bitrate C and packet size S are two parameters of our model that affect transmission delay D. As Equation 3 and Figure 2 reveal, D scales proportionally the flow interarrival processes and period T between packets within a flow. Consequently and in conformity with Equation 6, changes in D do not modify the queue size encountered by any packet. This leads us to our first conclusion:

Observation 1: Neither the link bitrate nor the packet size affects the steady-state queuing.

An important practical implication from Observation 1 is a possibility of congestion control where a constant buffer suffices regardless of link capacities and packet sizes. In fact, RCP and other recent proposals are close to maintaining the perfect capacity scalability, i.e., independence of the steady-state queuing from the bottleneck link capacity. As we show later, the situation is different for scalability with respect to the number of flows.

IV. Analysis for Overprovisioned Buffers

We conduct a stochastic analysis of steady-state queuing in the overprovisioned buffer of a fully utilized link with N flows, where N is at least 20. While Figure 1 deliberately stretches expected arrivals of the N flows across interval $[0; 2.05T)$ in order to highlight differences between the transient and steady states, all N flows with any of the arrival distributions considered in our model are likely to arrive during interval $[0; T)$. First, we formally show that number M of flows with arrival times t_i within interval $[0; T)$ is close to N. Let ψ denote a distribution of flow interarrival times δ_j. The flow interarrival times represent arrival time t_i as:

$$t_i = \sum_{j=1}^{i} \delta_j. \tag{7}$$

Because N is large, and all δ_j are from the same distribution ψ, the Central Limit Theorem establishes that t_i follows the normal distribution with mean α and variance ω^2:

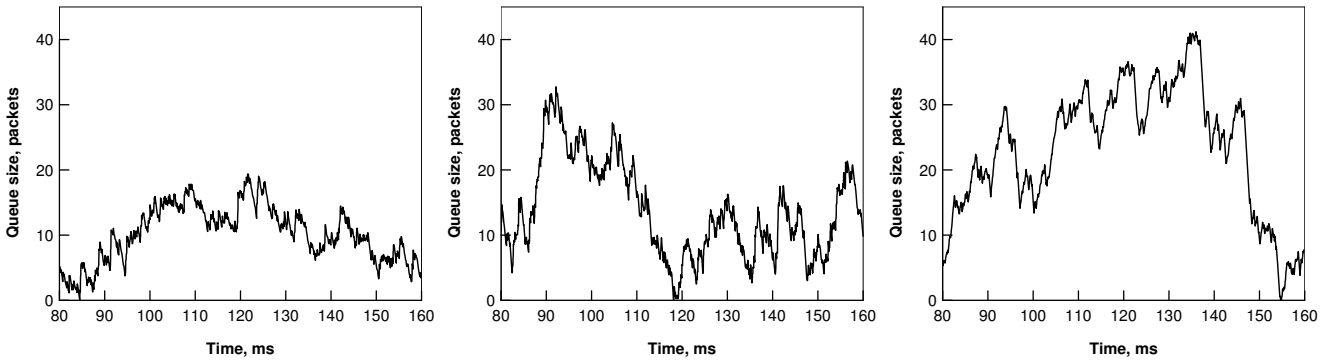
$$\alpha = i\mu \quad \text{and} \quad \omega^2 = i\sigma^2, \tag{8}$$

where μ and σ^2 are respectively the mean and variance of distribution ψ.

For the Exponential distribution of flow interarrival times, we express the probability that $M < N$, i.e., that a flow arrives at time T or later, as:

$$P[M < N] = P\left[m \geq \frac{N-M}{\sqrt{M}}\right] \tag{9}$$

where m is an auxiliary variable defined as:



(a) Uniform (b) Exponential (c) Pareto, $k = 2.1$

Fig. 4. One period of steady-state queuing in the overprovisioned buffer: N = 1,000, $U = 1$, C = 100 Mbps, and S = 1,000 bytes (T = 80 ms).

$$m = \frac{t_i - \frac{M}{N}T}{\frac{\sqrt{M}}{N}T}. \tag{10}$$

Since variable m follows the normal distribution with mean 0 and variance 1, $P[M < N] \approx 0$ whenever

$$\frac{N - M}{\sqrt{M}} \geq 3. \tag{11}$$

Under Constraint 11, $M \to N$ if $N \to \infty$.

Similar lines of reasoning for the Uniform and Pareto flow interarrival processes also show that $M \to N$ when $N \to \infty$.

Based on the above, our subsequent analysis assumes that differences between N and M are negligible, i.e., all N flows arrive within time interval $[0; T)$, and interarrival times of packets within any steady-state period of duration T conform to distribution ψ of flow interarrival times. Then, we express queue size q_i encountered by the i-th packet of the steady-state time interval $[T; 2T)$ as:

$$q_i = q_0 + i - \frac{t_i}{D} \tag{12}$$

where q_0 represents the queue size at time T, i is the number of packets arrived during time interval $[T; T + t_i)$, and $\frac{t_i}{D}$ denotes the number of packets transmitted into the link during this interval $[T; T + t_i)$. Whereas N packets that arrive during any time interval of length T consume exactly time T to be transmitted into the link, Equation 12 captures the steady-state queue size exactly. Since we are primarily interested in lower bounds, we assume $q_0 = 0$.

Let $\theta(i)$ denote the probability that the i-th packet encounters a queue size longer than Q. Since t_i is normally distributed, we use Equations 8 and 12 to derive:

$$\theta(i) = P\left[q_i > Q\right] = \frac{1}{2}\left(1 - \text{erf}\left(\frac{(Q - i)D + i\mu}{\sigma\sqrt{2i}}\right)\right) \tag{13}$$

where erf is the error function, and μ and σ are parameters of interarrival time distribution ψ. Figure 2 contains particular values of μ and σ for the three distributions of our model. Taking into account these values with $U = 1$, we simplify the above expression for $\theta(i)$ as:

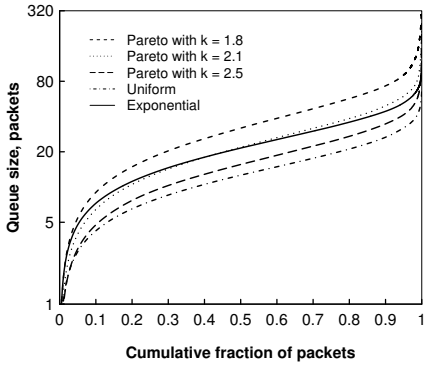


Fig. 5. Cumulative distributions of steady-state queuing in the overprovisioned buffer of a fully utilized link with 1,000 flows.

$$\theta(i) = \frac{1}{2}\left(1 - \text{erf}\left(\frac{Q}{A_\psi\sqrt{i}}\right)\right) \tag{14}$$

where A_ψ is a coefficient specific to interarrival time distribution ψ. The following A_ψ values characterize the Uniform, Exponential, and Pareto distributions respectively:

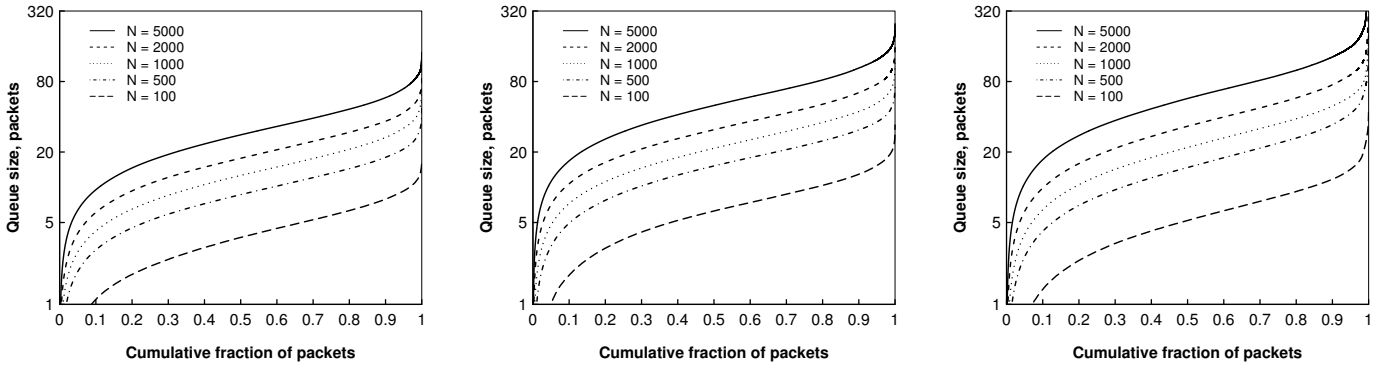
$$A_{\text{Uni}} = \sqrt{\frac{2}{3}},\ A_{\text{Exp}} = \sqrt{2},\ \text{and}\ A_{\text{Par}} = \sqrt{\frac{2}{k(k-2)}}. \tag{15}$$

Figure 3 reports $\theta(i)$ expressions for the three distributions. Using θ to represent the probability that the steady-state queue size exceeds Q, we express this probability as:

$$\theta = \frac{1}{N}\sum_{i=1}^{N}\theta(i). \tag{16}$$

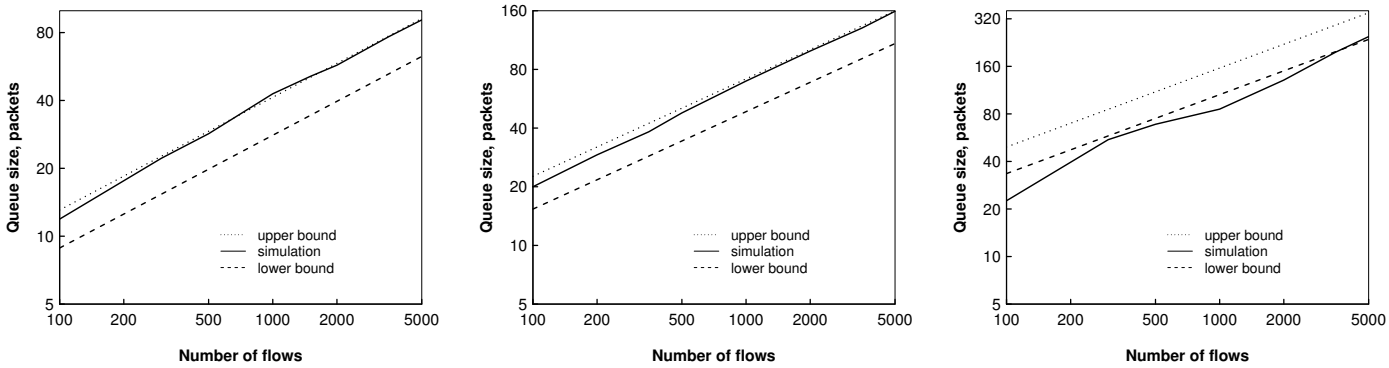
Since $\theta(i)$ is a nonnegative increasing function, we bound θ from below as follows:

$$\theta \geq \frac{1}{N}\sum_{i=\frac{2N}{3}}^{N}\theta\left(i\right) \geq \frac{\theta\left(\frac{2N}{3}\right)}{3} \geq \frac{1 - \text{erf}\left(\frac{Q}{A_\psi\sqrt{\frac{2N}{3}}}\right)}{6} \tag{17}$$



(a) Uniform (b) Exponential (c) Pareto, $k = 2.1$

Fig. 6. Cumulative distributions of steady-state queuing in the overprovisioned buffer of a fully utilized link for different numbers of flows.



(a) Uniform (b) Exponential (c) Pareto, $k = 2.1$

Fig. 7. Analytic and experimental results for the steady-state queue size encountered by top 1% packets in the overprovisioned buffer of a fully utilized link.

To derive a lower bound on queuing for top θ packets, we define L_θ to be such that

$$\text{erf}\left(L_\theta\sqrt{\frac{3}{2}}\right) = 1 - 6\theta. \tag{18}$$

$L_{1\%} \approx 1.1$ and $L_{5\%} \approx 0.6$ for top 1% and 5% packets respectively. From Inequality 17, we express lower bound $Q_{\min}$ on the queue size for top θ packets as:

$$Q_{\min} = L_\theta A_\psi \sqrt{N} \tag{19}$$

where L_θ depends only on the fraction of packets, and A_ψ is a coefficient associated with the interarrival time distribution. Figure 3 presents the lower bounds on steady-state queuing for top 1% and 5% packets under the Uniform, Exponential, and Pareto interarrival time distributions. While deriving Equation 19, we have proved the following:

Theorem 1: Minimum queuing in the overprovisioned buffer of a fully utilized link is $O(\sqrt{N})$, where N is the number of flows sharing the link.

Theorem 1 has an important practical implication that no congestion control protocol is able to avoid packet losses while utilizing the bottleneck link fully with a constant buffer and arbitrarily many flows.

Although our primary interest is in lower bounds, a similar line of reasoning allows us to bound θ from above:

$$\theta \le \theta(N) \le \frac{1 - \text{erf}\left(\frac{Q}{A_\psi\sqrt{N}}\right)}{2}. \tag{20}$$

Defining E_θ to be such that

$$\text{erf}(E_\theta) = 1 - 2\theta \tag{21}$$

with $E_{1\%} \approx 1.6$ and $E_{5\%} \approx 1.15$ for top 1% and 5% packets respectively, we derive upper bound $Q_{\max}$ on the queue size for top θ packets as:

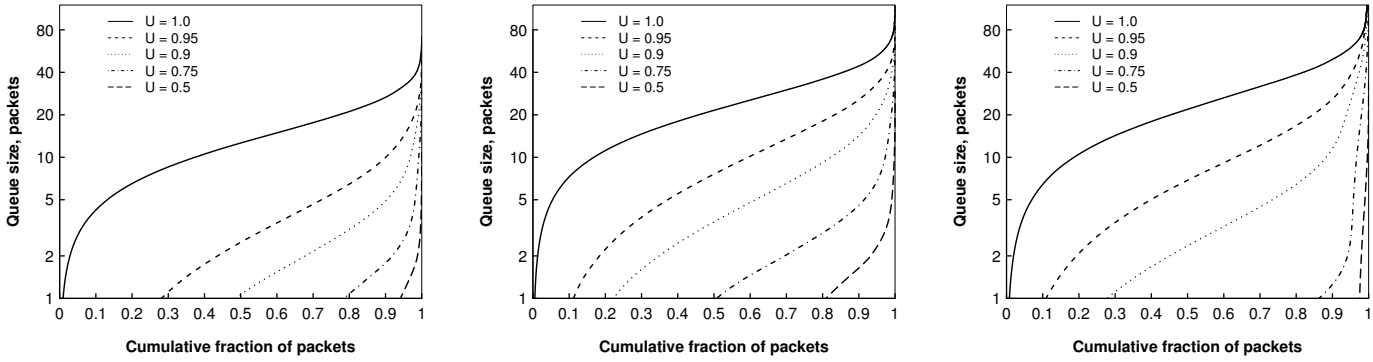
$$Q_{\max} = E_\theta A_\psi \sqrt{N}. \tag{22}$$

V. Simulations for Overprovisioned Buffers

To validate the above analysis for overprovisioned buffers, we conduct simulations within our model. The simplicity of the model enables extensive simulations with firm results. To improve efficiency of the simulations even further without sacrificing any accuracy, we transform the arrival times of flows as

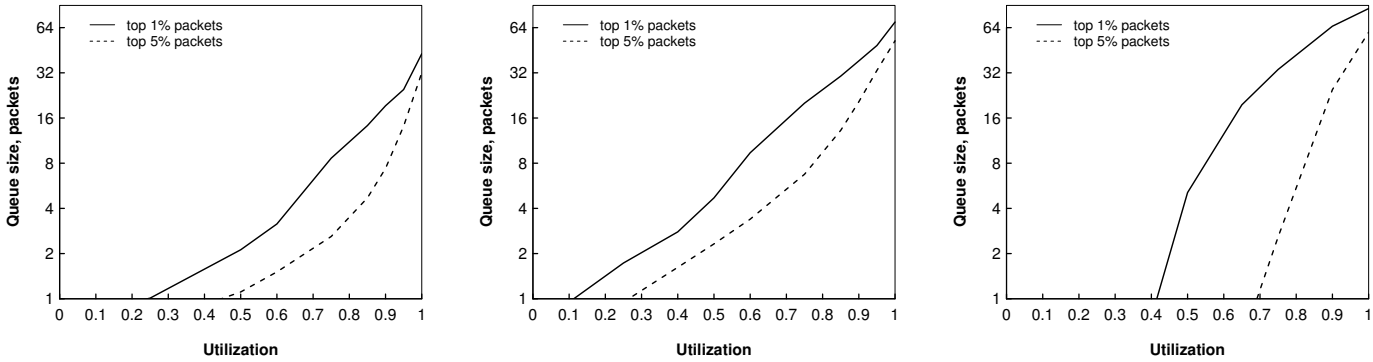
$$\tau_i = t_i \bmod T \tag{23}$$

where mod is the modulo operation on real numbers. The transformation compresses the transient stage into time interval



(a) Uniform (b) Exponential (c) Pareto, $k = 2.1$

Fig. 8. Cumulative distributions of the steady-state queue size in the overprovisioned buffer for various link utilizations and 1,000 flows.



(a) Uniform (b) Exponential (c) Pareto, $k = 2.1$

Fig. 9. Impact of the bottleneck link utilization on the steady-state queuing for the overprovisioned buffer and 1,000 flows.

$[0;T)$ and guarantees that the steady state ensues already at time T. With the transformed arrival times, a simulation run needs to examine only $2N$ packets, or 2 packets per flow. The first packet of flow i arrives at time τ_i in accordance with the used distribution and Equation 23. The second packet of flow i arrives at time $\tau_i + T$, and its queuing delay d_i is used to compute steady-state queue size q_i according to Equation 6.

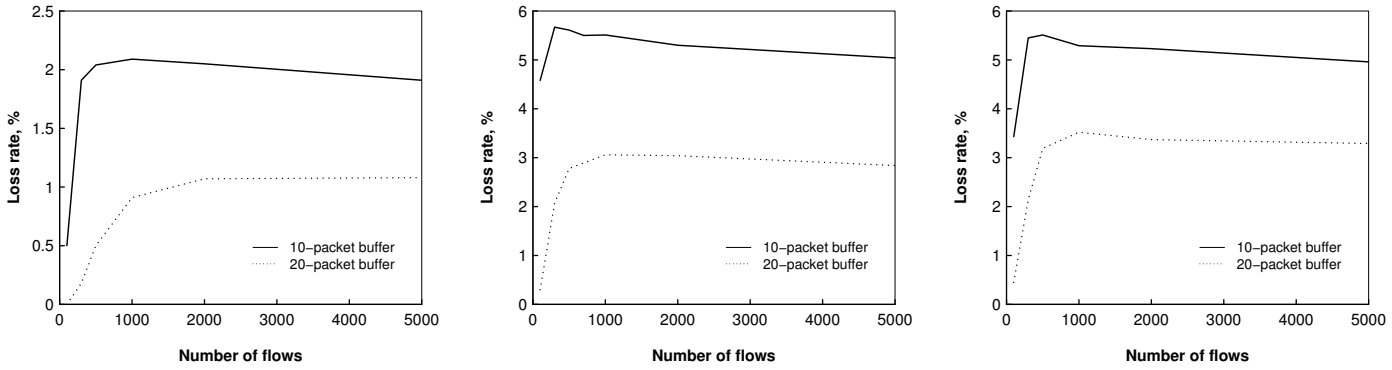
Note that unlike the analysis which makes simplifying assumptions, our simulation methodology captures steady-state queuing in the model exactly. Figure 1 illustrates queuing with original and transformed flow arrival times. The plot confirms that the transformed arrival times yield the same queuing during time interval $[T;2T)$ as the steady-state queuing that emerges with the original arrival times only after time T.

For each examined set of parameter settings, we perform 1,000 experiments and report average steady-state queue sizes with high certainty. Unless explicitly stated otherwise, the parameters take the following default values: $N = 1{,}000$, $U = 1$, $C = 100$ Mbps, and $S = 1{,}000$ bytes. In these settings, period T equals 80 ms. Figure 4 illustrates typical patterns of steady-state queuing under the default parameter settings.

Figure 5 plots cumulative distributions of the queue size for different flow interarrival processes, including two additional Pareto distributions with smaller index $k = 1.8$ and larger index $k = 2.5$. All five distributions of flow interarrival times exhibit qualitatively similar profiles of steady-state queuing: while the queue size rises persistently across the spectrum, top percentiles experience sharp increases in the queue size. Queuing is the smoothest under the Uniform distribution. As intended, our main Pareto distribution (with $k = 2.1$) also produces smooth queuing: in comparison to the Exponential distribution, queue sizes are larger for top percentiles but lower for bottom percentiles.

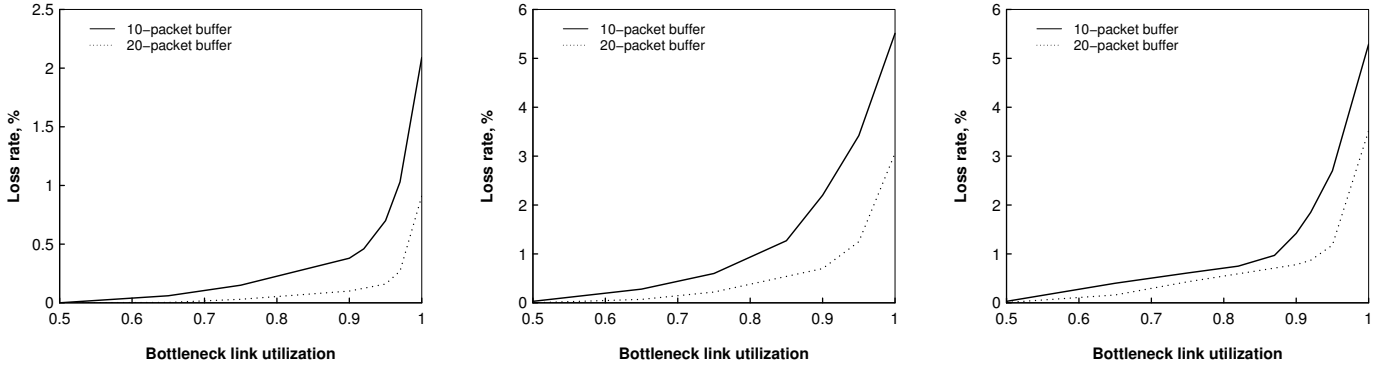
To evaluate the dependence of the steady-state queuing on the number of flows, we vary N in our experiments from 100 to 5,000. Figure 6 unveils that varying the number of flows preserves the qualitative profile observed for cumulative distributions of the queue size in Figure 5. Figure 6 also shows that larger values of N consistently produce longer queues. In particular, while bottom 5% packets experience no queuing at all with 100 flows, queue sizes for this percentile can be as large as 10 packets with 5,000 flows.

Since function $\theta(i)$ rises quickly toward its maximum $\theta(N)$, we expect the minimum steady-state queue size for top θ packets to be much closer to upper bound $Q_{\max}$ than lower bound $Q_{\min}$, which are given in Equations 22 and 19 respectively. To check this expectation, Figure 7 reports the analytical and experimental results for the queue size encountered by top



(a) Uniform (b) Exponential (c) Pareto, $k = 2.1$

Fig. 10. Steady-state loss rates with different numbers of flows and small buffers at fully utilized links.



(a) Uniform (b) Exponential (c) Pareto, $k = 2.1$

Fig. 11. Impact of the bottleneck link utilization on the steady-state loss rates with 1,000 flows and small buffers.

1% packets. For the Uniform and Exponential flow interarrival time distributions, the simulation results are remarkably close to the upper bound. For one subrange of N under the Uniform distribution, the steady-state queue size exceeds $Q_{\max}$ slightly because our derivation of the bounds assumes $q_0 = 0$ and thus underestimates the steady-state queuing. With the Pareto distribution, the simulated queue sizes are mostly below the span predicted by our bounds. We are unsure why our analysis overestimates queuing under the Pareto distribution and will explore this issue in future work. However, our experimental results for all three flow interarrival time distributions are generally consistent with the theoretical conclusion that minimum queuing in the overprovisioned buffer of a fully utilized link is $O(\sqrt{N})$.

The $O(\sqrt{N})$ dependence means that no congestion control protocol is able to avoid packet losses at a fully utilized link with a constant buffer and arbitrarily many flows. The result also justifies the current practice of overprovisioning backbone links. Such overprovisioning moves bottlenecks to access links where flows are less numerous and thus induce shorter queues.

Now, we explore whether reduced utilization of the bottleneck link mitigates the above concerns about scalability with respect to the number of flows. In the next set of experiments, U varies from 0.05 to 1. Figure 8 demonstrates that decreasing the utilization subdues the steady-state queuing substantially. For instance, bottom 80% packets experience no queuing at all with 50% utilization. Figure 9 quantifies the superlinear reductions of the queue size as U decreases. With 95% utilization, the minimum queue size for top 1% packets is 25, 49, 75 packets under the Uniform, Exponential, Pareto flow interarrival times respectively. Decreasing U to 75% and further to 50% reduces the queue size to 9, 20, 34 packets and 2, 5, 5 packets respectively. The graphs also reveal that utilization decreases help most dramatically under the Pareto distribution. Our experimental results for the link utilization justify the common practice of operating network links with average utilization of at most 50% [34].

VI. Loss Rates with Small Buffers

Our derived and validated $O(\sqrt{N})$ lower bound on achievable queuing reveals that no congestion control protocol is able to avoid packet losses with a constant buffer at a fully utilized link if the number of flows is arbitrarily large. In this section, we evaluate how our ideally smooth congestion control performs with small buffers. Figure 10 plots steady-state loss rates for fully utilized links that have 10-packet and

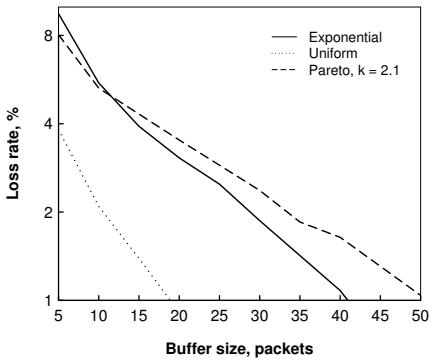


Fig. 12. Impact of the buffer size on the steady-state loss rates at a fully utilized link with 1,000 flows.

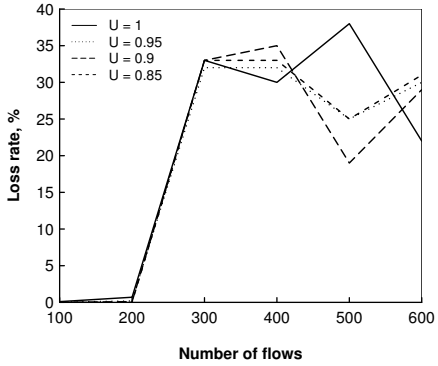


Fig. 13. Steady-state loss rates under RCP with various link utilizations and 10-packet buffer.

20-packet buffers. The number of flows in these experiments varies from 100 to 5,000. As N starts to grow, the loss rates increase quickly. Surprisingly, the loss rates stabilize and even decrease slightly after N grows beyond 1,000. The bounded nature of the loss rates opens an exciting prospect that a practical congestion control protocol might also be able to bound loss rates at fully utilized links with small buffers.

The loss rates reported in Figure 10 are higher with the Exponential and Pareto flow interarrival time distributions and stabilize with both distributions around 3% and 5% for the 10-packet and 20-packet buffer respectively. If one deems these specific values as too high, reducing the link utilization helps once again. Figure 11 shows prompt dramatic reductions in the steady-state loss rates as the link utilization reduces from 100% to 50%.

Another avenue for reducing the minimum achievable loss rate is to increase the link buffer size. Figure 12 quantifies how the steady-state loss rates decrease under our idealized protocol as the buffer size grows from 5 to 50 packets. In particular, reduction of the loss rates to 1% requires about 20, 40, and 50 packets with the Uniform, Exponential, and Pareto flow interarrival time distributions respectively.

VII. Comparison with Practical Protocols

To examine how modern practical protocols compare with our idealized scheme, we conduct ns-2 [28] simulations for RCP [2], an explicit congestion control protocol that strives to transmit smoothly in the steady state. The core bottleneck link of a simulated single-bottleneck dumbbell topology has a 10-packet buffer, 200-Mbps bitrate, and 30-ms propagation delay. RTT for each flow is 90 ms. All flows arrive according to a Poisson process and use packets sized to 1000 bytes. Based on 10 runs for each experimental setting, Figure 13 reports the steady-state loss rates for various link utilizations as the the number of flows increases from 100 to 600. The loss-rate profiles are qualitatively similar to the reported for the ideally smooth scheme: the loss rates are low for small N, then undergo prompt dramatic increases, and remain almost stable for larger N. However, the actual values of the stabilized loss rates are significantly larger under RCP, around 30%. Interestingly, reducing the targeted utilization from 100% to 85% does not help RCP to lessen the losses.

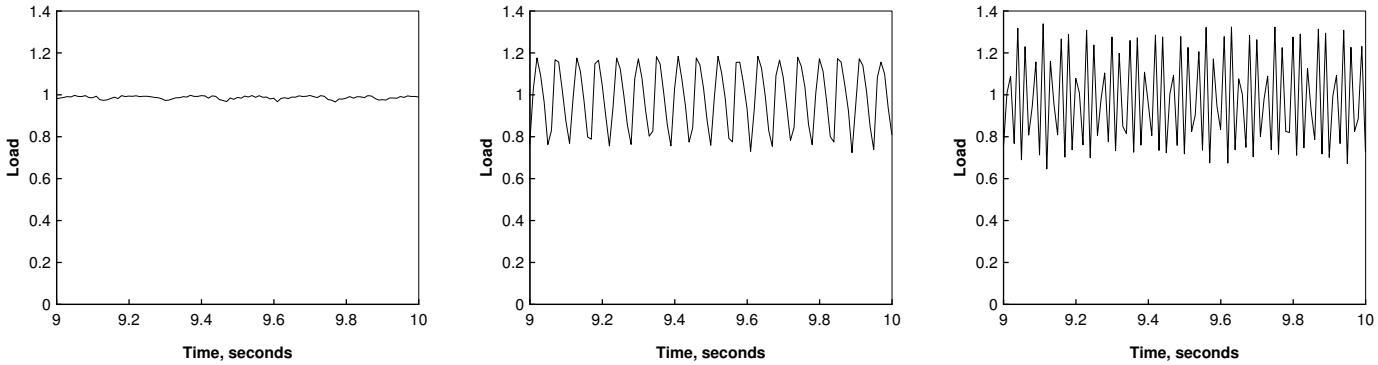
We trace the large loss rates at highly multiplexed links to unstable load under RCP in the steady state. While the default RCP strives to maintain the load of 1 (i.e., 100% utulization and no queuing at the bottleneck link), Figure 14 shows that as the number of flows increases from 100 to 300 to 600, the actual load on the bottleneck link starts to oscillate widely, causing frequent overflows of the small 10-packet buffer. Our experiments suggest that modern congestion control protocols still have large headroom for improving their steady-state performance with small buffers.

VIII. Conclusion

In this paper, we presented an idealized congestion control protocol where all flows always transmit at equal rates. The ideally smooth transmission causes link queuing only due to asynchrony of flow arrivals, which is intrinsic to computer networks. While realistic modeling of Internet flow arrivals is still an open problem, we considered three smooth distributions of flow interarrival times.

The practical utility of our model is in exposing lower bounds on queuing and loss achievable at highly multiplexed links by any congestion control protocol. In particular, we established that the minimum steady-state queue size experienced by a fixed fraction of packets in the overprovisioned buffer of a fully utilized link is $O(\sqrt{N})$, where N is the number of flows sharing the link. The $O(\sqrt{N})$ lower bound implies that no congestion control protocol is able to avoid packet losses at a fully utilized link with a constant buffer and arbitrarily many flows.

While a prominent aspect of our model is its simplicity, our simulation methodology captured the steady-state queuing for N concurrent flows exactly by examining only $2N$ packets. The low overhead enabled us to assess the steady-state performance with high certainty through extensive experiments with up to 5,000 concurrent flows and 1,000 runs per experimental setting. For overprovisioned buffers, the simulation results are generally consistent with our theoretical conclusion that minimum queuing at fully utilized links is $O(\sqrt{N})$. The



a) with 100 flows (b) with 300 flows (c) with 600 flows

Fig. 14. Instability of the steady-state link load under RCP that strives to fully utilize the bottleneck link with the 10-packet buffer.

simulations also showed that reducing the bottleneck link utilization alleviates the concerns about scalability with respect to the number of flows.

The experiments for fully utilized links with small buffers revealed bounded steady-state loss rates under our ideally smooth congestion control. This result opened an exciting prospect that a practical protocol might also be able to bound loss rates. To examine how modern protocols compare with our idealized scheme, we simulated RCP in ns-2, observed significantly larger steady-state loss rates at highly multiplexed links, and traced the large loss rates to instability of load under RCP. Our experiments suggested that modern congestion control protocols still have large headroom for improving their steady-state performance with small buffers.

Beside the fresh perspective on fundamental limitations of any congestion control in terms of minimum queuing and loss, our paper justified common practices in network capacity planning. Specifically, our results support the practices of operating network links with average utilizations of at most 50% and overprovisioning backbone links in order to move bottlenecks to access links.

ACKNOWLEDGMENT

We are thankful to Ken Calvert, David Yau, and Jorg Liebeherr for inviting us to present a precursor of this work as poster "Price of Asynchrony: Queuing under Ideally Smooth Congestion Control" at ICNP (International Conference on Network Protocols) in October 2007.

REFERENCES

[1] D. Katabi, M. Handley, and C. Rohrs, "Congestion Control for High Bandwidth-Delay Product Networks," in *Proceedings ACM SIGCOMM 2002*, August 2002.

[2] N. Dukkipati, M. Kobayashi, R. Zhang-Shen, and N. McKeown, "Processor Sharing Flows in the Internet," in *Proceedings International Workshop on Quality of Service (IWQoS 2005)*, June 2005.

[3] Y. Zhang, D. Leonard, and D. Loguinov, "JetMax: Scalable Max-Min Congestion Control for High-Speed Heterogeneous Networks," in *Proceedings IEEE INFOCOM 2006*, April 2006.

[4] Y. Xia, L. Subramanian, I. Stoica, and S. Kalyanaraman, "One More Bit Is Enough," in *Proceedings ACM SIGCOMM 2005*, August 2005.

[5] M. Podlesny and S. Gorinsky, "Multimodal Congestion Control for Low Stable-State Queuing," in *Proceedings IEEE INFOCOM 2007*, May 2007.

[6] ——, "MCP: Few Bits for Fairing and Small Queues in the Stable State," in *Proceedings IEEE Symposium on Computers and Communications (ISCC 2007)*, July 2007.

[7] R. Jain, "A Delay-Based Approach for Congestion Avoidance in Interconnected Heterogeneous Computer Networks," *ACM Computer Communications Review*, vol. 19, no. 5, pp. 56–71, October 1989.

[8] L. Brakmo, S. O'Malley, and L. Peterson, "TCP Vegas: New Techniques for Congestion Detection and Avoidance," in *Proceedings ACM SIGCOMM 1994*, August 1994.

[9] R. King, R. Baraniuk, and R. Riedi, "TCP-Africa: An Adaptive and Fair Rapid Increase Rule for Scalable TCP," in *Proceedings IEEE INFOCOM 2005*, March 2005.

[10] S. Floyd, M. Handley, and J. Padhye, "A Comparison of Equation-Based and AIMD Congestion Control," *www.aciri.org/tfrc/*, May 2000.

[11] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-Based Congestion Control for Unicast Applications," in *Proceedings ACM SIGCOMM 2000*, August 2000.

[12] V. Jacobson, "Congestion Avoidance and Control," in *Proceedings ACM SIGCOMM 1988*, August 1988.

[13] M. Allman, V. Paxson, and W. Stevens, "TCP Congestion Control," RFC 2581, April 1999.

[14] G. Appenzeller, I. Keslassy, and N. McKeown, "Sizing Router Buffers," in *Proceedings ACM SIGCOMM 2004*, September 2004.

[15] D. Wischik and N. McKeown, "Part I: Buffer Sizes for Core Routers," *ACM Computer Communication Review*, vol. 35, no. 3, pp. 75–78, July 2005.

[16] G. Raina, D. Towsley, and D. Wischik, "Part II: Control Theory for Buffer Sizing," *ACM Computer Communication Review*, vol. 35, no. 3, pp. 79–82, July 2005.

[17] M. Enachescu, Y. Ganjali, A. Goel, N. McKeown, and T. Roughgarden, "Routers with Very Small Buffers," in *Proceedings IEEE INFOCOM 2006*, April 2006.

[18] L. Qiu, Y. Zhang, and S. Keshav, "Understanding the Performance of Many TCP Flows," *Computer Networks*, vol. 37, no. 3-4, pp. 277–306, November 2001.

[19] A. Dhamdhere and C. Dovrolis, "Open Issues in Router Buffer Sizing," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 87–92, January 2006.

[20] R. Morris, "Scalable TCP Congestion Control," in *Proceedings IEEE INFOCOM 2000*, March 2000.

[21] A. Dhamdhere, H. Jiang, and C. Dovrolis, "Buffer Sizing for Congested Internet Links," in *Proceedings IEEE INFOCOM 2005*, March 2005.

[22] V. Jacobson, "Modified TCP Congestion Control Algorithm," End2end-interest mailing list, April 1990.

[23] C. Villamizar and C. Song, "High Performance TCP in the ANSNET," *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 5, pp. 45–60, November 1994.

[24] S. Gorinsky, A. Kantawala, and J. Turner, "Link Buffer Sizing: A New Look at the Old Problem," in *Proceedings IEEE Symposium on Computers and Communications (ISCC 2005)*, June 2005.

[25] Y. Gu, D. Towsley, C. Hollot, and H. Zhang, "Congestion Control for Small Buffer High Speed Networks," in *Proceedings IEEE INFOCOM 2007*, May 2007.

[26] D. Y. Eun and X. Wang, "Achieving 100% Throughput in TCP/AQM under Aggressive Packet Marking with Small Buffer," *IEEE/ACM Transactions on Networking*, to appear.

[27] S. Gorinsky, A. Kantawala, and J. Turner, "Simulation Perspectives on Link Buffer Sizing," *Simulation: Transactions of the Society for Modeling and Simulation International*, to appear in 2007.

[28] S. McCanne and S. Floyd, *ns Network Simulator. http://www.isi.edu/nsnam/ns/.*

[29] M. Crovella and A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 835–846, December 1997.

[30] W. Leland, M. Taqq, W. Willinger, and D. Wilson, "On the Self Similar Nature of Ethernet Traffic," in *Proceedings ACM SIGCOMM 1993*, September 1993.

[31] D. Chakraborty, A. Ashir, T. Suganuma, G. M. Keeni, T. Roy, and N. Shiratori, "Self-similar and Fractal Nature of Internet Traffic," *International Journal of Network Management*, vol. 14, no. 2, pp. 119–129, March 2004.

[32] C. Nuzman, I. Saniee, W. Sweldens, and A. Weiss, "A Compound Model for TCP Connection Arrivals for LAN and WAN Applications," *Computer Networks*, vol. 40, no. 3, pp. 319–337, October 2002.

[33] V. Paxson and S. Floyd, "Wide-Area Traffic: The Failure of Poisson Modeling," *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, pp. 226–244, June 1995.

[34] A. Odlyzko, "Data Networks are Lightly Utilized, and will Stay that Way," *Review of Network Economics*, vol. 2, no. 3, September 2003.