Washington University in St. Louis

Washington University Open Scholarship

All Computer Science and Engineering
Research

Computer Science and Engineering

Report Number: WUCSE-2007-24

2007

Price of Asynchrony: Queuing under Ideally Smooth Congestion Control

Maxim Podlesny and Sergey Gorinsky

The ability of TCP (Transmission Control Protocol) or alternative congestion control algorithms to operate successfully in networks with small link buffers has recently become a subject of intensive research. In this paper, we investigate fundamental limitations on minimum buffer requirements for any congestion control. We present an idealized protocol where all flows always transmit at their fair rates. The ideally smooth congestion control causes link queuing only due to asynchrony of flow arrivals, which is intrinsic to computer networks. Our analysis and simulations for different distributions of flow interarrival times agree that the buffer size needed for a fixed... [Read complete abstract on page 2.](#)

Follow this and additional works at: https://openscholarship.wustl.edu/cse_research

Part of the [Computer Engineering Commons](#), and the [Computer Sciences Commons](#)

Recommended Citation

Podlesny, Maxim and Gorinsky, Sergey, "Price of Asynchrony: Queuing under Ideally Smooth Congestion Control" Report Number: WUCSE-2007-24 (2007). *All Computer Science and Engineering Research*. https://openscholarship.wustl.edu/cse_research/128

Department of Computer Science & Engineering - Washington University in St. Louis
Campus Box 1045 - St. Louis, MO - 63130 - ph: (314) 935-6160.

Price of Asynchrony: Queuing under Ideally Smooth Congestion Control

Maxim Podlesny and Sergey Gorinsky

Complete Abstract:

The ability of TCP (Transmission Control Protocol) or alternative congestion control algorithms to operate successfully in networks with small link buffers has recently become a subject of intensive research. In this paper, we investigate fundamental limitations on minimum buffer requirements for any congestion control. We present an idealized protocol where all flows always transmit at their fair rates. The ideally smooth congestion control causes link queuing only due to asynchrony of flow arrivals, which is intrinsic to computer networks. Our analysis and simulations for different distributions of flow interarrival times agree that the buffer size needed for a fixed loss rate at a fully utilized link is $O(\sqrt{N})$, where N is the number of flows sharing the link. This result undermines a standpoint that small constant buffers are sufficient for links serving large numbers of flows. On the other hand, we show that fixed loss rates are achievable with constant buffers at a price of reducing average utilization of the bottleneck link. In particular, we derive analytical expressions for the maximum bottleneck link utilization that supports a fixed loss rate with arbitrarily many flows and constant buffer. We also experiment with a simple modification of RCP (Rate Control Protocol). The experiments indicate practical feasibility of compensating for larger numbers of flows by decreasing the targeted utilization of the bottleneck link.

2007-24

Price of Asynchrony: Queuing under Ideally Smooth Congestion Control

Authors: Maxim Podlesny and Sergey Gorinsky

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

Web Page: <http://www.arl.wustl.edu/~gorinsky/>

Abstract: The ability of TCP (Transmission Control Protocol) or alternative congestion control algorithms to operate successfully in networks with small link buffers has recently become a subject of intensive research. In this paper, we investigate fundamental limitations on minimum buffer requirements for any congestion control. We present an idealized protocol where all flows always transmit at their fair rates. The ideally smooth congestion control causes link queuing only due to asynchrony of flow arrivals, which is intrinsic to computer networks. Our analysis and simulations for different distributions of flow interarrival times agree that the buffer size needed for a fixed loss rate at a fully utilized link is $O(\sqrt{N})$, where N is the number of flows sharing the link. This result undermines a standpoint that small constant buffers are sufficient for links serving large numbers of flows. On the other hand, we show that fixed loss rates are achievable with constant buffers at a price of reducing average utilization of the bottleneck link. In particular, we derive analytical expressions for the maximum bottleneck link utilization that supports a fixed loss rate with arbitrarily many flows and constant buffer. We also experiment with a simple modification of RCP (Rate Control Protocol). The experiments indicate practical feasibility of compensating for larger numbers of flows by decreasing the targeted utilization of the bottleneck link.

Type of Report: Other

Price of Asynchrony: Queuing under Ideally Smooth Congestion Control

Maxim Podlesny and Sergey Gorinsky

Technical Report WUCSE-2007-24

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

April 2007

Abstract—The ability of TCP (Transmission Control Protocol) or alternative congestion control algorithms to operate successfully in networks with small link buffers has recently become a subject of intensive research. In this paper, we investigate fundamental limitations on minimum buffer requirements for any congestion control. We present an idealized protocol where all flows always transmit at their fair rates. The ideally smooth congestion control causes link queuing only due to asynchrony of flow arrivals, which is intrinsic to computer networks. Our analysis and simulations for different distributions of flow interarrival times agree that the buffer size needed for a fixed loss rate at a fully utilized link is $O(\sqrt{N})$, where N is the number of flows sharing the link. This result undermines a standpoint that small constant buffers are sufficient for links serving large numbers of flows. On the other hand, we show that fixed loss rates are achievable with constant buffers at a price of reducing average utilization of the bottleneck link. In particular, we derive analytical expressions for the maximum bottleneck link utilization that supports a fixed loss rate with arbitrarily many flows and constant buffer. We also experiment with a simple modification of RCP (Rate Control Protocol). The experiments indicate practical feasibility of compensating for larger numbers of flows by decreasing the targeted utilization of the bottleneck link.

I. INTRODUCTION

The problem of link buffer sizing has recently attracted significant attention. In particular, contradictory opinions were expressed on sizing the link buffer with respect to the number of flows sharing the link. One view maintains that increasing the number of TCP (Transmission Control Protocol) [2], [15] flows reduces aggregate oscillations of traffic on the bottleneck link. This view led to conclusions that small buffer sizes are sufficient for TCP to utilize efficiently links that serve large numbers of concurrent flows [3], [29], [31]. An extreme position among such conclusions is a guideline to set the link buffer size to a small constant, e.g., to hold at most 20 packets [9]. However, the above arguments for small buffer sizes are far from being universally accepted. Other researchers point out that when the network path restricts the window of a TCP flow to less than few packets, TCP congestion control causes high loss rates and suffers from frequent retransmission timeouts [6], [28]. These observations yield entirely opposite suggestions to keep the link buffer

size proportional to the number of flows in some network settings [7], [22], i.e., to make the buffer even larger than the traditionally recommended bitrate-delay product [16], [30].

While link buffer sizing has been debated mostly in the context of TCP congestion control, another approach to the problem is to examine what applications want. In particular, since long queuing at bottleneck links can severely disrupt delay-sensitive applications, a different rationale for small link buffers is their ability to ensure low queuing delay. From this perspective on link buffer sizing, different congestion control is needed to support high link utilization and low loss rates despite the curtailed buffering [12]. In fact, a new congestion control protocol E-TCP is recently proposed for networks with small link buffers [14]. This pioneering work is preceded by a wide body of related research on smooth steady-state congestion control for multimedia applications [10], [11] and large-bitrate long-propagation network topologies [8], [17], [32], [33].

Smoothness in congestion control is a challenging goal due to its natural tension with another objective of prompt response to changes in network conditions. Discovery of fair efficient transmission is fraught with packet bursts and other effects that can cause link queuing. Unlike numerous earlier attempts to design better algorithms for transmission adjustment, our paper has quite a different intention. We ask the following question: How much link queuing would occur, and how much link buffering would be needed, under an ideally smooth congestion control that always transmits at fair rates? Answering this question is useful for networking practice because of providing a lower bound on queuing under any actual algorithm for congestion control. Equivalently, the idealized scenario also sheds light on the minimum buffer size needed at a bottleneck link shared by many flows.

The ideally smooth congestion control does not eliminate queuing altogether because flows arrive asynchronously, creating a possibility that packets of different flows overlap at a link. For example, even if fair constant-rate flows underutilize their network paths on average, simultaneous arrivals of packets from multiple flows to an idle link create a queue. The asynchrony of flow arrivals is the only feature distinguishing

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 computer networks.

Through both simulations and analysis, we evaluate queuing under the ideally smooth congestion control for various types of flow arrival distributions. Our investigation shows that queuing becomes longer as the link utilization or the number of flows increases. A particularly interesting result is that the buffer size needed to limit the loss rate at a fully utilized link to a fixed value is $O(\sqrt{N})$ where N is the number of flows. This finding undermines the prior suggestions that small constant buffers are sufficient for links serving large numbers of flows. We also explore how to rectify the imperfect population scalability of congestion control by reducing the targeted utilization of the bottleneck link. In addition to extensive experimentation within our model, we also perform ns-2 [21] simulations with RCP-U, our simple modification of RCP (Rate Control Protocol) [8]. Beside the fresh perspectives on fundamental limitations of any congestion control, our paper offers justifications for common practices in network capacity planning.

The rest of the paper is organized as follows. Section II clarifies our model. Section III shows that neither bottleneck link bitrate nor packet size affects the steady-state queuing under our idealized protocol; this suggests possibility of practical congestion control where a constant buffer is sufficient regardless of the bottleneck capacity and used packet sizes. Using extensive simulations, Section IV shows how the steady-state queuing depends on the number of flows and bottleneck link utilization. Section V supplements the experimental results with analysis. Section VI reports our ns-2 simulations with RCP-U. Finally, Section VII concludes the paper with a summary of its theoretical and practical contributions.

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 $i = 1, \dots, N$. Without loss of generality, we assume $t_1 = 0$. We refer to time between arrivals of flows $i - 1$ and i as δ_i :

$$\delta_i = t_i - t_{i-1}. \quad (1)$$

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

$$R = \frac{U \cdot C}{N}. \quad (2)$$

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

$$T = \frac{N \cdot S}{U \cdot C} = \frac{N \cdot D}{U} \quad (3)$$

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

$$D = \frac{S}{C}. \quad (4)$$

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. 1. Considered distributions of flow interarrival times.

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 some time, and creates some burstiness, 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 on the minimum buffer size needed for any practically realizable congestion control algorithm.

Under 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 .

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 [4], [5], [20], [23]. In particular, there is no universal agreement on how to model flow arrivals in different Internet applications [25]. 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 a larger emphasis on smooth distributions because we are primarily interested in uncovering the minimum required buffer size.

We consider the following three distributions of flow interarrival times: *Exponential* (smooth), *Uniform* (smoothest), and *Pareto* (burstiest). All three distributions have the same average value:

$$\mu = \frac{D}{U} = \frac{T}{N}, \quad (5)$$

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 1:

- *Uniform* interarrival times are distributed uniformly between 0 and 2μ .
- *Exponential* interarrival times are generated by a Poisson process with average arrival rate $\frac{1}{\mu}$.

Fig. 2. Transformation of original flow arrival times to promote simplicity of simulations: $N = 1,000$ flows arrive originally according to a Poisson process with average arrival rate $\frac{N}{2.05T}$, where $T = 80$ ms, and utilize the bottleneck link completely in the steady state, i.e., $U = 1$.

Fig. 3. Steady-state queuing during a single experiment: $N = 1,000$, $U = 1$, $C = 100$ Mbps, and $S = 1,000$ bytes ($T = 80$ ms).

Fig. 4. Cumulative distributions of steady-state queuing with different flow interarrival processes: $N = 1,000$ and $U = 1$.

- Pareto interarrival times follow the Pareto distribution with mean μ and index $k = 2.1$. In our experiments, we

also consider other Pareto distributions with both smaller (burstier) and larger (smoother) values of k .

The main metric of performance for flow i is *queue size* q_i measured in packets:

$$q_i = \frac{d_i}{D} \quad (6)$$

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

To quantify how effectively a buffer of size B handles the steady-state queuing, we define *loss rate* ρ as a fraction of packets that encounter a queue size of more than B .

III. BOTTLENECK 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 1 reveal, D scales the flow interarrival processes and period T between packets within a flow proportionally. 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 bottleneck link bitrate nor the packet size affects the steady-state queuing.

Fig. 5. Cumulative distributions of the queue size for different number of flows with $U = 1$.

Fig. 6. Dependence of steady-state queuing on the number of flows for $U = 1$.

An important practical implication from the above result is a possibility of congestion control where a constant buffer is sufficient regardless of the bottleneck capacity and used packet sizes. In fact, XCP (Explicit Control Protocol) [17], RCP (Rate Control Protocol) [8], JetMax [33], and MCP (Multimodal Control Protocol) [26], [27] are examples of recent protocols that are close to maintaining the perfect capacity scalability, i.e., independence of the steady-state queue size from the bottleneck link capacity. As we show later, the situation with population scalability, i.e., dependence on the number of flows, is fundamentally different.

IV. SIMULATIONS OF STEADY-STATE QUEUING

Although in each of the considered distributions, most flows are expected to arrive during time interval $[0; T)$, some flows might arrive later and thereby postpone the ensuing steady state. To reduce complexity of simulations, we transform the arrival times of the flows as

$$\tau_i = t_i \bmod T \quad (7)$$

where mod is the modulo operation on real numbers. The transformation preserves the steady-state queuing pattern and commences the steady state by time T .

Figure 2 illustrates queuing for original and transformed flow arrival times. The graph confirms that the transformation

limits the transient stage to time interval $[0; T)$ and captures the steady-state queuing pattern during time interval $[T; 2T)$.

Hence, our simulations examine only the queuing caused by first two packets of all flows, or $2N$ packets altogether. The first packet of flow i arrives at time τ_i in accordance with the used distribution and Equation 7. The second packet of flow i arrives at time $\tau_i + T$, and its queuing delay d_i is used to compute queue size q_i according to Equation 6.

The simplicity of our model enables us to conduct 1,000 experiments for each examined set of parameter settings and report average 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 3 illustrates typical patterns of steady-state queuing under the default parameter settings. As expected, the graphs show that the Uniform and Pareto distributions of flow interarrival times yield smoothest and burstiest queuing respectively.

Figure 4 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 queuing.

Fig. 7. Cumulative distributions of the queue size for various link utilizations with $N = 1,000$.

Fig. 8. Influence of the link utilization on steady-state queuing for $N = 1,000$.

A. Number of flows

To assess dependence of the steady-state queuing on the number of flows, we vary N in our experiments from 100 to 5,000. Figure 5 unveils that varying the number of flows preserves the qualitative profile observed for cumulative distributions of the queue size in Figure 4. Figure 5 also shows that larger values of N consistently produce longer queues. Figure 6 displays the sublinear queue growth more clearly by plotting queue size B such that 99% (95% for the dashed lines) of all packets in the steady state encounter a queue size of at most B . In particular with 1,000 flows, 99% of the packets encounter a queue size of at most 43, 70, and 86 packets under the Uniform, Exponential, and Pareto flow interarrival processes respectively. With 2,000 and 5,000 flows, queue size B for 99% of the packets increases to at most 59, 95, 131 packets and 90, 157, 247 packets respectively. The experiments indicate a fundamental impossibility of perfect population scalability of steady-state queuing under congestion control that targets to utilize the bottleneck link fully:

Observation 2: The queue size grows sublinearly with the number of flows sharing the bottleneck link.

Figure 6 has another practically useful interpretation. A line for percentile $1 - \rho$ represents buffer size B needed to maintain a loss rate of at most ρ . For instance, the line for the 95th

percentile with 1,000 flows and Pareto flow interarrival times shows that maintaining a loss rate of at most 5% requires the buffer size of 60 packets. The plot for the 99th percentile demonstrates that decreasing this loss rate to 1% requires increasing the buffer size to 86 packets.

B. Bottleneck link utilization

In the next set of experiments, we vary link utilization U from 0.05 to 1. Figure 7 confirms that changes in U do not affect the qualitative pattern for cumulative distributions of the queue size. The plots also demonstrate that increasing the link utilization intensifies the steady-state queuing. Figure 8 quantifies the superlinear growth of the queue size. In particular for utilization 50%, the buffer size needed to provide a loss rate of at most 1% is 2, 5, and 5 packets with Uniform, Exponential, and Pareto flow interarrival times respectively. Raising U to 75% and 95% increases the buffer requirement to 9, 20, 34 packets and 25, 49, 75 packets respectively. The experiments lead us to the following conclusion:

Observation 3: The queue size grows superlinearly with the link utilization.

The results offer two other interesting insights. First, extremely low queuing with utilization 50% and relatively high queuing with utilizations close to 100% justify a common

practice among network providers to operate links with average utilizations of at most 50% [24].

Second, the experiments suggest a possibility of congestion control that targets a lower link utilization to compensate for a higher number of flows and thereby achieves a fixed loss rate with a constant link buffer and arbitrarily large number of flows. Later in the paper, we conduct ns-2 [21] simulations with a congestion control protocol designed toward this goal.

V. ANALYSIS OF STEADY-STATE QUEUING

To generalize the experimental findings from Section IV, we now conduct a stochastic analysis of steady-state queuing when number N of flows is large (at least 20).

While Figure 2 deliberately stretches expected arrivals of the N flows across interval $[0; 2.05T)$ in order to highlight differences between transient and steady states as well as to illustrate how our transformation of original flow arrival times promotes simplicity of simulations, 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 original 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:

$$t_i = \sum_{j=1}^i \delta_j. \quad (8)$$

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 :

$$\alpha = i\mu \text{ and } \omega^2 = i\sigma^2, \quad (9)$$

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:

$$P[M < N] = P\left[m \geq \frac{N-M}{\sqrt{M}}\right] \quad (10)$$

where m is an auxiliary variable defined as:

$$m = \frac{t_i - \frac{M}{N}T}{\frac{\sqrt{M}}{N}T}. \quad (11)$$

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

$$\frac{N-M}{\sqrt{M}} \geq 3. \quad (12)$$

Under Constraint 12, $M \rightarrow N$ if $N \rightarrow \infty$.

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

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 $[0; T)$. Hence, we assume that interarrival times of packets within any steady-state period of duration T conform to original distribution ψ of flow interarrival times.

A. Fully utilized link

When the link is utilized fully, N packets that arrive during any time interval of length T consume exactly time T to be transmitted into the link. Hence, any steady-state period of duration T contains a moment when the queue size equals 0. Figure 3 illustrates this property of the steady-state queuing for the Uniform, Exponential, and Pareto interarrival time distributions. Without loss of generality and for convenience of using the already established notation, we suppose that the queue size becomes 0 at time T . Then, we express queue size q_i encountered by packet i of the steady-state time interval $[T; 2T)$ as:

$$q_i = i - \frac{t_i}{D} \quad (13)$$

where 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 $[T; T + t_i)$. Due to the full utilization of the link, the right-hand side of Equation 13 is never negative and captures the queue size precisely.

We represent loss rate ρ as the probability that steady-state queue size q_i exceeds buffer size B . Since t_i is normally distributed, we use Equations 9 and 13 to derive:

$$\rho = P[q_i > B] = \frac{1}{2} \left(1 - \operatorname{erf} \left(\frac{(B-i)D + i\mu}{\sigma\sqrt{2i}} \right) \right) \quad (14)$$

where erf is the error function, and μ and σ are parameters of interarrival time distribution ψ . Figure 1 reports 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 the loss rate as:

$$\rho = \frac{1}{2} \left(1 - \operatorname{erf} \left(\frac{B}{A_\psi \sqrt{i}} \right) \right), \quad (15)$$

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:

$$A_{\text{Uni}} = \sqrt{\frac{2}{3}}, \quad A_{\text{Exp}} = \sqrt{2}, \quad \text{and} \quad A_{\text{Par}} = \sqrt{\frac{2}{k(k-2)}}. \quad (16)$$

Furthermore, since up to N packets arrive during the steady-state period, loss rate ρ is bounded from above as:

$$\rho \leq \frac{1}{2} \left(1 - \operatorname{erf} \left(\frac{B}{A_\psi \sqrt{N}} \right) \right). \quad (17)$$

For the Uniform, Exponential, and Pareto interarrival time distributions, we report respective bounds for the loss rate in Figure 9.

To determine the buffer size needed to support a loss rate of at most ρ , we define L_ρ to be such that

$$\operatorname{erf}(L_\rho) = 1 - 2\rho. \quad (18)$$

$L_{1\%} \approx 1.6$ and $L_{5\%} \approx 1.15$ for loss rates 1% and 5% respectively. From Inequality 17, we derive that minimum

Distribution name	Loss rate ρ	Minimum buffer size $B_{\min}$		
		for at most ρ	for at most $\rho = 1\%$	for at most $\rho = 5\%$
Uniform	$\frac{1}{2} \left(1 - \operatorname{erf} \left(\sqrt{\frac{3}{2N}} B \right) \right)$	$L_\rho \sqrt{\frac{2}{3}} \sqrt{N}$	$1.31 \sqrt{N}$	$0.94 \sqrt{N}$
Exponential	$\frac{1}{2} \left(1 - \operatorname{erf} \left(\frac{B}{\sqrt{2N}} \right) \right)$	$L_\rho \sqrt{2} \sqrt{N}$	$2.26 \sqrt{N}$	$1.63 \sqrt{N}$
Pareto, $k > 2$	$\frac{1}{2} \left(1 - \operatorname{erf} \left(\sqrt{\frac{k(k-2)}{2N}} B \right) \right)$	$L_\rho \sqrt{\frac{2}{k(k-2)}} \sqrt{N}$	$\frac{2.26}{\sqrt{k(k-2)}} \sqrt{N}$	$\frac{1.63}{\sqrt{k(k-2)}} \sqrt{N}$

Fig. 9. Buffer sizes needed to support a loss rate of at most ρ at a fully utilized link.

Fig. 10. Minimum buffer size needed to provide a fixed loss rate at a fully utilized link: theory versus simulations.

buffer size $B_{\min}$ required to support a loss rate of at most ρ with N flows is:

$$B_{\min} = L_\rho A_\psi \sqrt{N}, \quad (19)$$

where L_ρ depends only on the loss rate, and A_ψ is a coefficient associated with the interarrival time distribution. Figure 9 shows minimum buffer size $B_{\min}$ needed to support a loss rate of at most 1% and 5% with the Uniform, Exponential, and Pareto interarrival time distributions.

While deriving Equation 19, we have proved the following result that constitutes the main contribution of our paper:

Theorem 1: Minimum buffer size required to provide a fixed loss rate at a fully utilized link is $O(\sqrt{N})$, where N is the number of flows sharing the link.

Figure 10 compares theoretical predictions of the minimum buffer size in Equation 19 with our simulation results for loss rates 1% and 5%. Figures 10a and 10b reveal that the theoretical and experimental results are remarkably close for the Uniform and Exponential flow interarrival time distributions. However, Figure 10c shows that Equation 19 overestimates the needed buffer size significantly for the Pareto distribution. We attribute the differences to the heavy-tail nature of the Pareto

distribution. For the examined numbers of flows, the heavy tail of Pareto flow interarrival times undermines our analytical assumption that all N flows arrive within time interval $[0; T)$. As N grows further, we expect Equation 19 to become more accurate in estimating the minimum buffer size for the Pareto distribution.

Theorem 1 has two important practical implications. First, it establishes that no congestion control protocol is able to limit the loss rate to a fixed value while utilizing the bottleneck link fully with constant buffer. Second, our analysis offers a theoretical justification for a current practice of overprovisioning backbone links: such overprovisioning moves bottlenecks to access links where numbers of competing flows are smaller and therefore require smaller buffers to support the same loss rate at the same link utilization.

B. Underutilized bottleneck link

While our results for fully utilized links undermine the standpoint that small constant buffers are sufficient with arbitrarily many flows, this section explores how to rectify the imperfect population scalability of congestion control by reducing the targeted utilization of the bottleneck link. As

Enachescu, Ganjali, Goel, McKeown, and Roughgarden [9] point out in the context of Paced TCP [1], [18], [19], analysis of queuing at an underutilized link is a hard problem. We tackle the problem by extending our analytical method used for $U = 1$. However, since the right-hand side of Equation 13 might produce negative values when $0 < U < 1$, the extension does not inherit the exactness in representing the queue size.

The expression for the loss rate with buffer size B becomes:

$$\rho = \frac{1}{2} \left(1 - \operatorname{erf} \left(\frac{UB + (1-U)i}{A_\psi \sqrt{i}} \right) \right). \quad (20)$$

By maximizing the right-hand side of Equation 20 over i up to N , we determine that with the maximization point at $i = \min \left(N, \frac{UB}{1-U} \right)$, loss rate ρ is bounded from above as:

$$\rho \leq \begin{cases} \frac{1}{2} \left(1 - \operatorname{erf} \left(\frac{UB + (1-U)N}{A_\psi \sqrt{N}} \right) \right) & \text{if } N \leq \frac{UB}{1-U}, \\ \frac{1}{2} \left(1 - \operatorname{erf} \left(\frac{2\sqrt{(1-U)UB}}{A_\psi} \right) \right) & \text{if } N > \frac{UB}{1-U}. \end{cases} \quad (21)$$

Consequently, maximum link utilization $U_{\max}$ to support a loss rate of at most ρ with buffer size B is equal to:

$$U_{\max} = \begin{cases} \frac{1 - \frac{L_\rho A_\psi}{1 - \frac{\sqrt{N}}{B}}}{\frac{\sqrt{N}}{B}} & \text{if } N \geq B \left(\frac{2\sqrt{N}}{L_\rho A_\psi} - 1 \right), \\ \frac{1 + \sqrt{1 - \frac{(L_\rho A_\psi)^2}{B}}}{2} & \text{if } N < B \left(\frac{2\sqrt{N}}{L_\rho A_\psi} - 1 \right). \end{cases} \quad (22)$$

VI. NS-2 EXPERIMENTS WITH RCP-U

So far, we have studied link buffer sizing within our model where flows of the idealized protocol always transmit at their fair rates. Now, we use ns-2 simulator [21] to experiment with a real complete protocol for congestion control. In particular, we experiment with a simple modification of RCP (Rate Control Protocol) [8] to explore whether adjusting the targeted utilization of the bottleneck link is able to support a fixed loss rate with a small constant buffer and arbitrarily many flows. Two factors influenced us to choose RCP. First, RCP is an advance modern protocol for rate-based congestion control that uses explicit feedback from routers to transmit smoothly in the steady state. Second, RCP strives to keep the cumulative transmission rate of all flows at bottleneck bitrate C . Hence, by replacing C with $U \cdot C$ in the control algorithm, we are capable of changing RCP to aim for bottleneck link utilization U , instead of 1. We refer to this simple modification of RCP as RCP-U.

We conduct the ns-2 simulations in a single-bottleneck dumbbell topology where the core bottleneck link has a 10-packet buffer, 100-Mbps bitrate, and 30-ms propagation delay. The packet size is uniform for all flows and equals 1000 bytes. The flows arrive according to a Poisson process. Round-trip time for each flow is 90 ms. We vary the number of RCP-U flows from 50 to 500 and the targeted utilization from 0.5 to 1. Due to poor scalability of ns-2 packet-level simulations [13], we are able to conduct only one experiment

Fig. 11. Steady-state loss rates under RCP-U in ns-2.

for each set of parameter values. Consequently, the steady-state loss rates reported in Figure 11 are quite noisy. However, they confirm the general tendency of lower losses with lower targeted utilizations. The RCP-U simulations also indicate practical feasibility to keep the loss rate below a fixed value by decreasing the targeted utilization as the number of flows increases. While the experimental loss rates are significantly higher than the theoretical minimums derived for our idealized protocol, we conclude that modern congestion control protocols still have large headroom for improving their steady-state queuing behavior.

VII. CONCLUSION

While link buffer sizing is debated mostly in the context of TCP congestion control, our paper established the minimum buffer size needed by any congestion control to keep the loss rate below a fixed value. We conducted analysis and simulations in an idealized protocol where all flows always transmit at their fair rates. The ideally smooth congestion control causes link queuing only due to asynchrony of flow arrivals, which is intrinsic to computer networks. While it remains uncertain how to model Internet flow arrivals realistically, we evaluated our idealized protocol for three distributions of flow interarrival times, with an emphasis on smooth distributions: Exponential (smooth), Uniform (smoothest), and Pareto (burstiest).

The main contribution of our paper is Theorem 1 stating that minimum buffer size $B_{\min}$ required to provide a fixed loss rate at a fully utilized link is $O(\sqrt{N})$, where N is the number of flows sharing the link. Results of our extensive simulations with the Uniform and Exponential flow interarrival time distributions are remarkably close to the analytically derived expression $B_{\min} = L_\rho A_\psi \sqrt{N}$ where L_ρ depends only on the loss rate bound ρ , and A_ψ is a coefficient specific to interarrival time distribution ψ . Our study of fully utilized links undermines the standpoint that small constant buffers are sufficient with arbitrarily many flows.

On the other hand, we also show that the imperfect population scalability of congestion control can be rectified by

reducing the targeted utilization of the bottleneck link. In particular, we derived analytical expressions for the maximum bottleneck link utilization that supports a fixed loss rate with arbitrarily many flows and constant buffer. We also conducted ns-2 experiments with RCP-U, a simple modification of RCP. These simulations indicated practical feasibility of compensating for larger numbers of competing flows by decreasing the targeted utilization of the bottleneck link. The RCP-U experiments also suggested that modern congestion control protocols still have large headroom for improving their steady-state queuing behavior.

In addition to the fresh perspective on fundamental limitations of any congestion control, our paper offered theoretical justifications for common practices in network capacity planning. Specifically, our results for steady-state queuing support the practice to operate links with average utilizations of at most 50%. Besides, the uncovered dependence of the minimum buffer size on the number of flows justifies the practice of overprovisioning backbone links and moving bottlenecks to access links.

To the best of our knowledge, the presented study is the first of a kind. In our future work, we plan to expand and solidify the reported results, through both analysis and especially experimentation.

REFERENCES

- [1] A. Aggarwal, S. Savage, and T. Anderson. Understanding the Performance of TCP Pacing. In *Proceedings IEEE INFOCOM 2000*, March 2000.
- [2] M. Allman, V. Paxson, and W. Stevens. TCP Congestion Control. RFC 2581, April 1999.
- [3] G. Appenzeller, I. Keslassy, and N. McKeown. Sizing Router Buffers. In *Proceedings ACM SIGCOMM 2004*, August 2004.
- [4] D. Chakraborty, A. Ashir, T. Suganuma, G. Mansfield Keeni, T. Roy, and N. Shiratori. Self-similar and Fractal Nature of Internet Traffic. *International Journal of Network Management*, 14(2):119–129, March 2004.
- [5] M. Crovella and A. Bestavros. Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes. In *Proceedings ACM SIGMETRICS 1996*, May 1996.
- [6] A. Dhamdhere and C. Dovrolis. Open Issues in Router Buffer Sizing. *ACM SIGCOMM Computer Communication Review*, 36(1):87–92, January 2006.
- [7] A. Dhamdhere, H. Jiang, and C. Dovrolis. Buffer Sizing for Congested Internet Links. In *Proceedings IEEE INFOCOM 2005*, March 2005.
- [8] N. Dukkkipati, M. Kobayashi, and N. McKeown. Processor Sharing Flows in the Internet. *Thirteenth International Workshop on Quality of Service (IWQoS)*, June 2005.
- [9] M. Enachescu, Y. Ganjali, A. Goel, N. McKeown, and T. Roughgarden. Routers with Very Small Buffers. In *Proceedings IEEE INFOCOM 2006*, April 2006.
- [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 1988*, May 2000.
- [12] 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.
- [13] 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.
- [14] Y. Gu, D. Towsley, C. Holot, and H. Zhang. Congestion Control for Small Buffer High Speed Networks. In *Proceedings IEEE INFOCOM 2007*, May 2007.
- [15] V. Jacobson. Congestion Avoidance and Control. In *Proceedings ACM SIGCOMM 1988*, August 1988.
- [16] V. Jacobson. Modified TCP Congestion Control Algorithm. End2end-interest mailing list, April 1990.
- [17] D. Katabi, M. Handley, and C. Rohrs. Congestion Control for High Bandwidth-Delay Product Networks. In *Proceedings ACM SIGCOMM 2002*, August 2002.
- [18] J. Kulik, R. Coulter, D. Rockwell, and C. Partridge. Paced TCP for High Delay-Bandwidth Networks. In *Proceedings IEEE Globecom 1999*, December 1999.
- [19] J. Kulik, R. Coulter, D. Rockwell, and C. Partridge. A Simulation Study of Paced TCP. Technical Report CR-2000-209416, BBN, gltrs.grc.nasa.gov/reports/2000/CR-2000-209416.pdf, January 2000.
- [20] W. Leland, M. Taq, W. Willinger, and D. Wilson. On the Self Similar Nature of Ethernet Traffic. In *Proceedings ACM SIGCOMM 1993*, September 1993.
- [21] S. McCanne and S. Floyd. *ns Network Simulator*. <http://www.isi.edu/nsnam/ns/>.
- [22] R. Morris. Scalable TCP Congestion Control. In *Proceedings IEEE INFOCOM 2000*, March 2000.
- [23] C. Nuzman, I. Saniee, W. Sweldens, and A. Weiss. A Compound Model for TCP Connection Arrivals. In *Proceedings ITC Workshop 2000*, September 2000.
- [24] A. Odlyzko. Data Networks are Lightly Utilized, and will Stay that Way. *Review of Network Economics*, 2(3), September 2003.
- [25] V. Paxson and S. Floyd. Wide-area Traffic: the Failure of Poisson Modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, June 1995.
- [26] M. Podlesny and S. Gorinsky. MCP: Few Bits for Fairing and Small Queues in the Stable State. In *Proceedings IEEE Symposium on Computers and Communications (ISCC 2007)*, July 2007.
- [27] M. Podlesny and S. Gorinsky. Multimodal Congestion Control for Low Stable-State Queuing. In *Proceedings IEEE INFOCOM Minisymposium 2007*, May 2007.
- [28] L. Qiu, Y. Zhang, and S. Keshav. Understanding the Performance of Many TCP Flows. *Computer Networks*, 37(3-4):277–306, November 2001.
- [29] G. Raina, D. Towsley, and D. Wischik. Part II: Control Theory for Buffer Sizing. *ACM SIGCOMM Computer Communication Review*, 35(3):79–82, July 2005.
- [30] C. Villamizar and C. Song. High Performance TCP in the ANSNET. *ACM SIGCOMM Computer Communication Review*, 24(5):45–60, November 1994.
- [31] D. Wischik and N. McKeown. Part I: Buffer Sizes for Core Routers. *ACM SIGCOMM Computer Communication Review*, 35(3):75–78, July 2005.
- [32] Y. Xia, L. Subramanian, I. Stoica, and S. Kalyanaraman. One More Bit Is Enough. In *Proceedings ACM SIGCOMM 2005*, August 2005.
- [33] 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.