

Router Buffer Sizing for TCP Traffic and the Role of the Output/Input Capacity Ratio

Ravi S. Prasad
Cisco Systems, Inc.
ravipra@cisco.com

Constantine Dovrolis
Georgia Institute of Technology
dovrolis@cc.gatech.edu

Marina Thottan
Bell-Labs
marinat@bell-labs.com

Abstract—The issue of router buffer sizing is still open and significant. Previous work either considers open-loop traffic or only analyzes persistent TCP flows. This paper differs in two ways. First, it considers the more realistic case of non-persistent TCP flows with heavy-tailed size distribution. Second, instead of only looking at link metrics, it focuses on the impact of buffer sizing on TCP performance. Specifically, our goal is to find the buffer size that maximizes the average per-flow TCP throughput. Through a combination of testbed experiments, simulation, and analysis, we reach the following conclusions. The output/input capacity ratio at a network link largely determines the required buffer size. If that ratio is larger than one, the loss rate drops exponentially with the buffer size and the optimal buffer size is close to zero. Otherwise, if the output/input capacity ratio is lower than one, the loss rate follows a power-law reduction with the buffer size and significant buffering is needed, especially with TCP flows that are in congestion-avoidance. Smaller transfers, which are mostly in slow-start, require significantly smaller buffers. We conclude by revisiting the ongoing debate on “small versus large” buffers from a new perspective.

Index Terms—Optimal Buffer Size, Per-flow TCP Throughput, Non-persistent TCP Flows, Router Buffer Management

I. INTRODUCTION

The need for buffering is a fundamental “fact of life” for packet switching networks. Packet buffers in routers (or switches) absorb the transient bursts that naturally occur in such networks, reduce the frequency of packet drops and, especially with TCP traffic, they can avoid under-utilization when TCP connections back off due to packet losses. At the same time, though, buffers introduce delay and jitter, and they increase the router cost and power dissipation.

After several decades of research and operational experience with packet switching networks, it is surprising that we still do not know how to dimension the buffer of a router interface. As explained in §II, the basic question - *how much buffering do we need at a given router interface?* - has received hugely different answers in the last 15-20 years, such as “a few dozens of packets”, “a bandwidth-delay product”, or “a multiple of the number of large TCP flows in that link”. It cannot be that all these answers are right. It is clear that we are still missing a crucial piece of understanding, despite the apparent simplicity of the previous question.

At the same time, the issue of buffer sizing becomes increasingly important in practice. The main reason is that IP networks are maturing from just offering reachability to providing performance-centered Service-Level Agreements and delay/loss assurances. Additionally, as the popularity of

voice and video applications increases, the potentially negative effects of over-buffered or under-buffered routers become more significant.

There are mostly three new ideas in this paper. First, instead of assuming that most of the traffic consists of “persistent” TCP flows, i.e., very long transfers that are mostly in congestion-avoidance, we work with the more realistic model of *non-persistent flows that follow a heavy-tailed size distribution*. The implications of this modeling deviation are major: first, non-persistent flows do not necessarily saturate their path, second, such flows can spend much of their lifetime in slow-start, and third, the number of active flows is highly variable with time. A discussion of the differences between the traffic generated from persistent and non-persistence flows is presented in [19]. Our results show that flows which spend most of their lifetime in slow-start require significantly less buffering than flows that live mostly in congestion-avoidance.

Second, instead of only considering link-level performance metrics, such as utilization, average delay and loss probability,¹ we focus on *the performance of individual TCP flows*, and in particular, on the relation between the average throughput of a TCP flow and the buffer size in its bottleneck link. TCP accounts for more than 90% of Internet traffic, and so a TCP-centric approach to router buffer sizing would be appropriate in practice for both users and network operators. On the other hand, aggregate metrics, such as link utilization or loss probability, can hide what happens at the transport or application layers. For instance, the link may have enough buffers so that it does not suffer from under-utilization, but the per-flow TCP throughput can be abysmally low.

Third, we focus on a structural characteristic of a link (or traffic) multiplexer that has been largely ignored in the past with the exception of [8] and [12]. This characteristic is the ratio of the output/input capacities. Consider a link of output capacity C_{out} that receives traffic from N links, each of input capacity C_{in} , with $NC_{in} > C_{out}$. In the simplest case, sources are directly connected to the input links, and the input capacity C_{in} is simply the source peak rate. More generally, however, a flow can be bottlenecked at any link between the source and the output port under consideration. Then, C_{in} is the capacity of that bottleneck link. For example, consider an edge router interface with output capacity 10Mbps. Suppose that the input interfaces of that router are 100Mbps. If the traffic sources are directly connected to that router, the ratio C_{out}/C_{in} is equal

¹We use the terms loss probability and loss rate interchangeably.

to 0.1. On the other hand, if the sources are connected to the router through 1Mbps DSL links, then the ratio C_{out}/C_{in} is equal to 10.

It turns out that the ratio $\Gamma = C_{out}/C_{in}$ largely determines the relation between loss probability and buffer size, and consequently, the relation between TCP throughput and buffer size. Specifically, we propose two approximations for the relation between buffer size and loss rate, which are reasonably accurate as long as the traffic that originates from each source is heavy-tailed. If $\Gamma < 1$, the loss rate can be approximated by a power-law of the buffer size. The buffer requirement can be significant in that case, especially when we aim to maximize the throughput of TCP flows that are in congestion-avoidance (the buffer requirement for TCP flows that are in slow-start is significantly lower). On the other hand, when $\Gamma > 1$, the loss probability drops almost exponentially with the buffer size, and the optimal buffer size is extremely small (just a few packets in practice, and zero theoretically). Γ is often lower than one in the access links of server farms, where hosts with 1 or 10 Gbps interfaces feed into lower capacity edge links. On the other hand, the ratio Γ is typically higher than one at access networks, as traffic enters the high-speed core from limited capacity residential links.

We reach the previous conclusions based on a combination of experiments, simulation and analysis.² Specifically, after we discuss the previous work in §II, we present results from testbed experiments using a Riverstone router (§III). These results bring up several important issues, such as the importance of provisioning the buffer size for heavy-load conditions and the existence of an optimal buffer size that depends on the flow size. The differences between large and small flows is further discussed in §IV, where we identify two models for the throughput of TCP flows, depending on whether a flow lives mostly in slow-start (S-model) or in congestion avoidance (L-model). As a simple analytical case-study, we use these two TCP models along with the loss probability and queueing delay of a simple M/M/1/B queue to derive the optimal buffer size for this basic (but unrealistic) queue (§V).

For more realistic queueing models, we conduct an extensive simulation study in which we examine the average queueing delay $d(B)$ and loss probability $p(B)$ as a function of the buffer size B under heavy-load conditions with TCP traffic (§VI). These results suggest two simple and parsimonious empirical models for $p(B)$. In §VI we also provide an analytical basis for the previous two models. In §VII, we use the models for $d(B)$ and $p(B)$ to derive the optimal buffer size, depending on the type of TCP flow (S-model versus L-model) and the value of Γ . We also examine the sensitivity of the TCP throughput around the optimal buffer size, when buffering is necessary (i.e., $\Gamma < 1$). We find out that throughput is a robust function of the buffer size, and that even large relative deviations from the optimal buffer size only cause minor loss in the per-flow throughput. Finally, in §VIII, we conclude by revisiting the recent debate on “large versus small buffers” based on the new insight from this work.

²All experimental data are available upon request from the authors. The simulations scripts are available on the Web. <http://www.cc.gatech.edu/~ravi/buffer.htm>.

II. RELATED WORK

Several queueing theoretic papers have analyzed the loss probability in finite buffers or the queueing tail probability in infinite buffers. Usually, however, that modeling approach considers exogenous (or open-loop) traffic models, in which the packet arrival process does not depend on the state of the queue. For instance, a paper by Kim and Shroff models the input traffic as a general Gaussian process, and derives an approximate expression for the loss probability in a finite buffer system [11].

An early experimental study by Villamizar and Song recommended that the buffer size should be equal to the *Bandwidth-Delay Product* (BDP) of that link [23]. The “delay” here refers to the RTT of a single and persistent TCP flow that attempts to saturate that link, while the “bandwidth” term refers to the capacity C of the link. That rule requires the bottleneck link to have enough buffer space so that the link can stay fully utilized while the TCP flow recovers from a loss-induced window reduction. No recommendations are given, however, for the more realistic case of multiple TCP flows with different RTTs.

The BDP rule results in a very large buffer requirement for high-capacity long-distance links. At the same time, such links are rarely saturated by a single TCP flow. Appenzeller et al. concluded that the buffer requirement at a link decreases with the square root of the number N of “large” TCP flows that go through that link [1]. According to their analysis, the buffer requirement to achieve almost full utilization is $B = (CT)/\sqrt{N}$, where T is the average RTT of the N (persistent) competing connections. The key insight behind this model is that, when the number of competing flows is sufficiently large, which is usually the case in core links, the N flows can be considered independent and non-synchronized, and so the standard deviation of the aggregate offered load (and of the queue occupancy) decreases with \sqrt{N} . An important point about this model is that it aims to keep the utilization close to 100%, without considering the resulting loss rate. As we discussed earlier, the traffic model of persistent flows is not realistic, even for core links. The number of active flows in core links can be large and relatively constant with time, but the flows constituting the aggregate traffic keep changing due to the arrival and departure of flows.

Link utilization is only one of the important factors in router buffer sizing. Loss rate, queueing delay, per flow throughput, etc. are also affected by router buffer sizing and they often lead to conflicting requirements. Dhamdhere et al. considered the buffer requirement of a Drop-Tail queue given constraints on the minimum utilization, the maximum loss-rate, and, when feasible, the maximum queueing delay [6]. They derive the minimum buffer size required to keep the link fully utilized by a set of N heterogeneous TCP flows, while keeping the loss rate and queueing delay bounded. However, the analysis of that paper is also limited by the assumption of persistent connections.

Morris was the first to consider the loss probability in the buffer sizing problem [16], [17]. That work recognized that the loss rate increases with the square of the number of competing TCP flows, and that buffering based on the BDP rule can

cause frequent TCP timeouts and unacceptable variations in the throughput of competing transfers [16]. Morris also proposed the Flow-Proportional Queueing (FPQ) mechanism, as a variation of RED, which adjusts the amount of buffering proportionally to the number of TCP flows.

Enachescu et al. showed that if the TCP sources are *paced* and have a bounded maximum window size, then the burstiness of the traffic aggregate is much smaller and a high link utilization (say 80%) can be achieved even with a buffer of a dozen packets [8]. The authors noted that explicit pacing may not be required when the access links are much slower than the core network links, providing a natural pacing. It is also interesting that their buffer sizing result is independent of the BDP, therefore the buffer requirement does not increase with upgrades in link capacity.

Recently, the ACM Computer Communications Review (CCR) has hosted a debate on buffer sizing through a sequence of letters [7], [8], [20], [24], [26]. On one side, authors in [8], [20], [26] have proposed significant reduction in the buffer requirement based in results from earlier studies [1], [8]. They argue that 100% link utilization can be attained with much smaller buffers, while large buffers cause increased delay, induce synchronization and are not feasible in all-optical routers. On the other side of the debate, authors in [7], [24] highlight the adverse impact of small buffer in terms of high loss rate and low per-flow throughput. Dhamdhare and Dovrolis argued that the recent proposals for much smaller buffer sizes can cause significant losses and performance degradation at the application layer [7]. Similar concerns are raised by Vu-Brugier et al. in [24]. That letter also reports measurements from operational links in which the buffer size was significantly reduced.

Ganjali and McKeown discussed three recent buffer sizing proposals [1], [6], [8] and argued that all these results may be applicable in different parts of the network, as they depend on various assumptions and they have different objectives [9].

Recently, Lakshmikantha et al. have reached similar conclusions with our work using different models [12]. Specifically, they show that depending on the ratio between the “edge” and “core” links capacities (which corresponds to our ratio Γ), the buffer requirement can change from $O(1)$ (just few packets) to $O(CT)$ (in the order of the BDP). They also consider non-persistent TCP flows and analyze the average flow completion time, which is equivalent to the average per-flow throughput that we focus on. They use a Poisson approximation for the case of $\Gamma > 1$ and a diffusion approximation for the case of $\Gamma < 1$. On the other hand, that work does not consider the differences in buffer sizing that can result from TCP flows that are mostly in slow-start versus congestion-avoidance. In our work we study these two cases separately, using the S-model and the L-model, respectively.

III. EXPERIMENTAL STUDY

To better understand the router buffer sizing problem in practice, we first conducted a set of experiments in a controlled testbed. The following results offer a number of interesting observations. We explain these observations through modeling and analysis in the subsequent sections.

A. Testbed setup

The schematic diagram of our experimental setup is shown in Figure 1. There are four hosts running servers/senders and

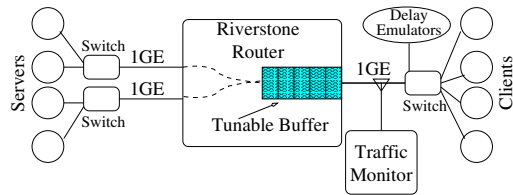


Fig. 1. Schematic diagram of the experimental testbed.

four hosts running clients/receivers, all of which are Fedora Core-5 Linux systems. Each host has two Intel Xeon CPUs running at 3.2 GHz, 2GB memory, and a DLink Gigabit PCIexpress network interface. The traffic from the four senders is aggregated on two Gig-Ethernet links before entering the router. The testbed bottleneck is the Gig-Ethernet output interface that connects the router to the distribution switch.

We use a Riverstone RS-15008 router. The switching fabric has much higher capacity than the bottleneck link, and there is no significant queuing at the input interfaces or at the fabric itself. The router has a tunable buffer size at the output line card. Specifically, we experiment with 20 buffer sizes, non-uniformly selected in the range 30KB to 38MB. With Ethernet MTU packets (1500B), the minimum buffer size is about 20 packets while the maximum buffer size is approximately 26,564 packets. We configured the output interface to use Drop-Tail queueing,³ and confirmed that the maximum queuing delay for a buffer size B is equal to B/C_{out} , where C_{out} is the capacity of the output link, as shown in Figure 2.

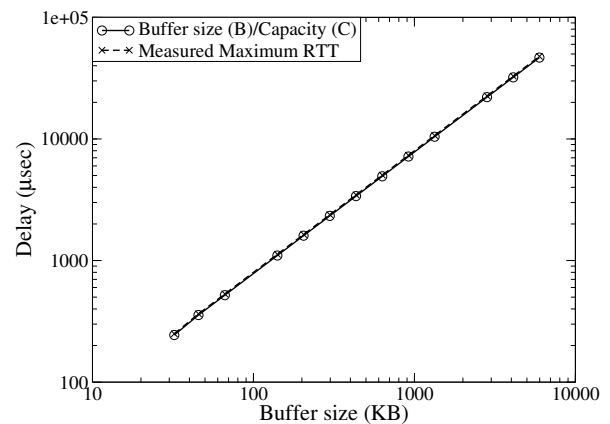


Fig. 2. The expected and observed maximum latency with drop-tail queueing at the testbed router.

Two delay emulators run NISTNet [4] to introduce propagation delays in the ACKs that flow from the clients to the servers. With this configuration, the minimum RTT of the TCP connections takes one of the following values: 30ms, 50ms, 120ms or 140ms, with a different RTT for each client machine.

³In this work, we focus exclusively on Drop-Tail queueing, as that is the norm in the Internet today.

The traffic at the output link is monitored using *tcpdump*, running on a FreeBSD 4.7 system. We record the headers of both data packets and ACKs. The packet drop rate at the monitor is 0.1%. We use these packet traces to measure link utilization and per-flow TCP throughput.

We configured the Linux end-hosts to use the TCP *Reno* stack that uses the NewReno congestion control variant with Selective Acknowledgments. The maximum advertised TCP window size is set to 13MB, so that transfers are never limited by that window. Finally, we confirmed that the path-MTU is 1500 Bytes and that the servers send maximum-sized segments.

The traffic is generated using the open-source Harpoon system [22]. We modified Harpoon so that it generates TCP traffic in a “closed-loop” flow arrival model [21]. A recent measurement work has shown that most of the traffic (60-80%) conforms to the closed-loop flow arrival model [18].⁴ In this model, a given number of “users” (running at the client hosts) performs successive TCP downloads from the servers. The size of the TCP transfers follows a given random distribution. After each download, the corresponding user stays idle for a “thinking period” that follows another random distribution. For the transfer sizes, we use a Pareto distribution with mean 80KB and shape parameter 1.5. These values are realistic, based on comparisons with actual packet traces. The think periods follow an exponential distribution with mean duration one second. The key point, here, is that the generated traffic, which resembles the aggregation of many ON-OFF sources with heavy-tailed ON periods, is Long-Range Dependent (LRD) [25]. As will be shown next, LRD nature of the traffic has major implications, because it causes significant deviations from the average offered load over long time periods.

One important property of this closed-loop flow arrival model is that it never causes overload (i.e., the offered load cannot exceed the capacity). Specifically, suppose we have U users, a flow size of S Bytes per user, a think period of T_h seconds and the flow completion time of T_t seconds: the offered load generated by the U clients is $\frac{US}{T_h+T_t}$. The offered load cannot exceed the capacity of the bottleneck link C_{out} . If that link becomes congested, the transfers take longer to complete, the term T_t increases, and the offered load remains at or below C_{out} [2]. Note that this is not the case in an open-loop flow arrival model, where new flows arrive based on an external random process (e.g., a Poisson process).

We control the offered load by emulating different numbers of users. The three experiments that we summarize in this paper, referred to as U_{1000} , U_{1200} , and U_{3000} , have $U=1000$, 1200 and 3000 users, respectively. The first two experiments do not generate enough offered load to constantly saturate the output link. The third experiment, U_{3000} , produces an offered load that is very close to the capacity (1Gbps). The run time for each experiment is 5 minutes. To avoid transient effects, we analyze the collected traces after a warm-up period of one minute.

⁴We have also experimented with open-loop TCP flow arrivals, without observing qualitative differences in the results.

B. Results

1) *Link utilization*: Figure 3 shows the average utilization ρ of the bottleneck link as a function of the buffer size in each of the three experiments. First note that the utilization curves, especially in the two experiments that do not saturate the output link, are quite noisy despite the fact that they represent 4-minute averages. Such high variability in the offered load is typical of LRD traffic and it should be expected even in longer time scales. We observe that the experiment U_{1000} can generate an average utilization of about 60-70% (with enough buffering), U_{1200} can generate a utilization of about 80-90%, while U_{3000} can saturate the link.

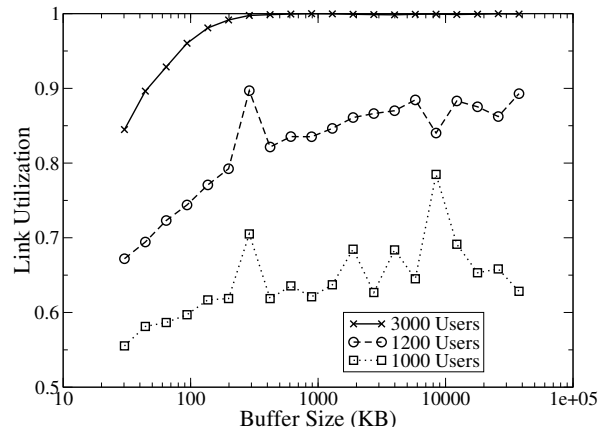


Fig. 3. Link utilization as a function of the router buffer size for U_{1000} , U_{1200} and U_{3000} .

As expected, there is a loss of utilization when the buffers are too small. Specifically, to achieve the maximum possible utilization we need a buffer size of at least 200KB in U_{3000} , and an even larger buffer in the other two experiments. The loss of utilization when there are not enough buffers has been studied in depth in previous work [1]. As we argue in the rest of this paper, however, maximizing the aggregate throughput should not be the only objective of buffer sizing.

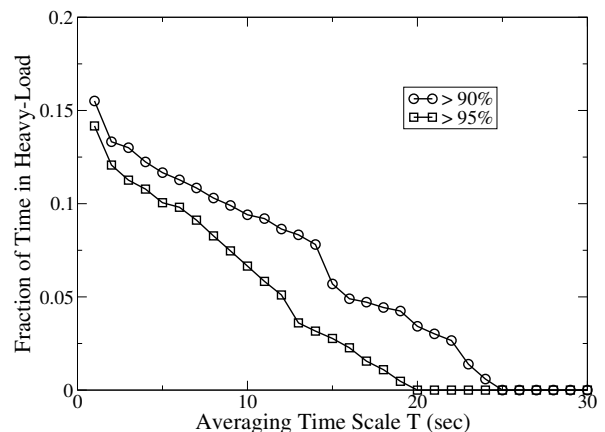


Fig. 4. Fraction of time a link is under heavy-load (i.e., more than 90% or 95% utilized) in different averaging time scales, when the long-term average utilization for the experiment duration is 68% (Experiment U_{1000} , Buffer size = 4MB).

Another important observation regarding the utilization of

the output link is that, even if the link is moderately loaded, there can be long time periods in which the link is practically congested. This is a direct consequence of the LRD nature of the Internet traffic [13]. For instance, consider one of the U_{1000} experiments in which the 4-minute average utilization is only 68%. Figure 4 shows the fraction of time in which the link utilization is higher than 90% or 95% (i.e., heavy-load conditions) when the utilization is measured in an averaging time scale of duration T . For example, with $T=10$ secs, we observe that the link is practically saturated, $\rho > 0.95$, for about 7% of the time. Congestion events lasting for several seconds are unacceptable to many Internet applications such as VoIP, interactive applications and network gaming. This example shows that *it is important that the buffer sizing process considers heavy-load conditions ($\rho \approx 1$), even when the average utilization of the link is much less than 100%*.

2) *Median per-flow throughput*: Next, we focus on the relation between per-flow throughput and router buffer size. Figures 5-7 show the median per-flow throughput for two groups of flows. One group, that we refer to as “small flows”, send about 45-50KB. The “large flows”, on the other hand, send more than 1000KB. The classification of flows as small or large is arbitrary at this point; we will discuss this crucial point in §IV.

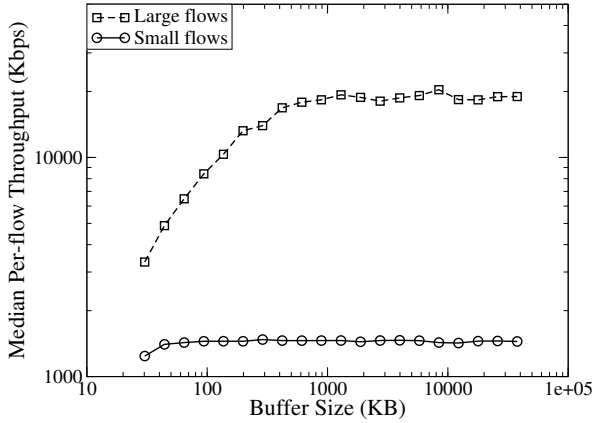


Fig. 5. Median per-flow throughput as a function of the buffer size in the U_{1000} experiments.

First, in the case of U_{1000} the median per-flow throughput generally increases with the buffer size up to a certain cutoff point. Note that the Y-axis is in log scale. The minimum buffer size that leads to the maximum per-flow throughput can be viewed as the *optimal buffer size* \hat{B} . Note that the optimal buffer size is significantly lower for small flows compared to large flows. The experiment U_{1200} gives similar results. Second, the optimal buffer size for each flow type increases as the load increases. And third, in the saturated-link experiment (U_{3000}), we also note that the median per-flow throughput of small flows first increases up to a maximum point that corresponds to the optimal buffer size \hat{B} , and it then drops to a lower value.

The above experimental results raise the following questions: What causes the difference in the optimal buffer size between small flows and large flows? Why does the per-flow

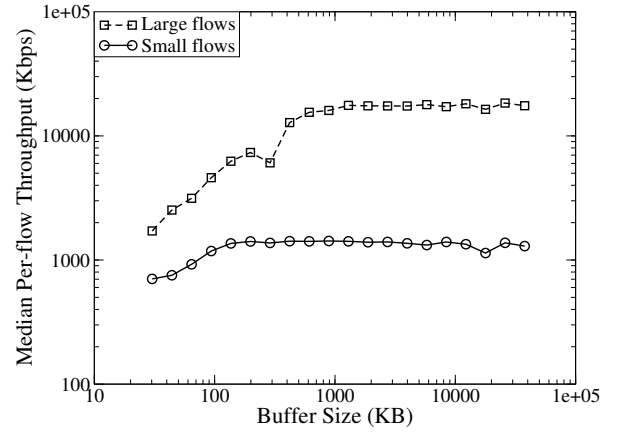


Fig. 6. Median per-flow throughput as a function of the buffer size in the U_{1200} experiments.

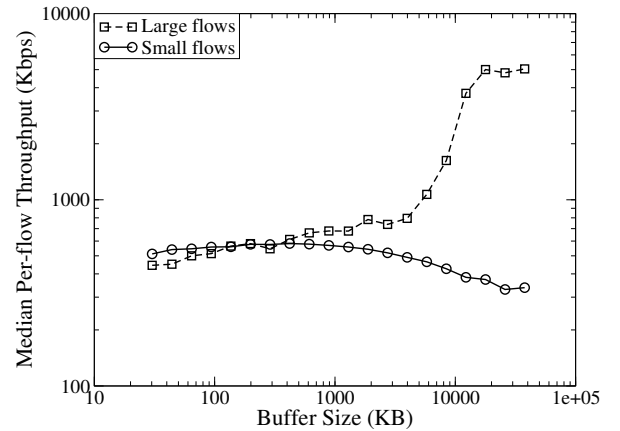


Fig. 7. Median per-flow throughput as a function of the buffer size in the U_{3000} experiments.

throughput increase up to a certain point as we increase the buffer size? Why does it drop after that point, at least for small flows? And more generally, *what does the optimal buffer size depend on?* We will answer these questions in the following sections.

3) *Variability of per-flow throughput*: Here, we present distributions of the per-flow TCP throughput for different buffer sizes. These distributions show how the buffer size affects the variability of the per-flow TCP throughput for small flows and for large flows. Specifically, Figure 8 shows the CDFs of the per-flow throughput for four different buffer sizes in the U_{3000} experiment.

Observe that *for small flows, as the buffer size increases, the variability of the per-flow throughput drops significantly*. The coefficient of variation for small flows *decreases* from 0.94 when $B=30$ KB to 0.25 when $B=38$ MB. The reason is that, as the buffer size increases the packet loss rate decreases, and an increasing fraction of small flows complete in slow-start. Without losses, however, the throughput of a TCP flow is only limited (without much statistical variation) by the flow’s RTT. So, as the buffer size increases, the variation range of the per-flow throughput for small flows decreases.

On the other hand, *for large flows, as the buffer size increases, the variability of the per-flow throughput increases*.

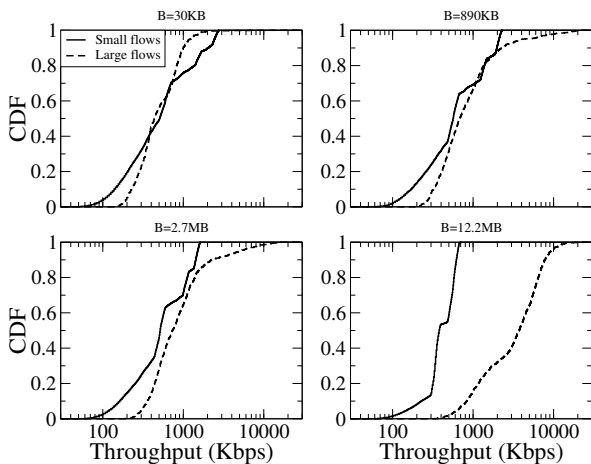


Fig. 8. CDFs of the per-flow throughput for four buffer sizes in the U_{3000} experiments.

Specifically, the coefficient of variation *increases* from 0.75 when $B=30\text{KB}$ to 1.06 when $B=38\text{MB}$. The reason is that a long flow may encounter packet losses even when the loss rate is very low. The long flows that experience losses switch from slow-start to congestion avoidance or retransmission timeouts, and receive a throughput that is both lower and less predictable than throughput of the long flows that do not experience any losses and complete in slow-start. As the buffer size increases, and the loss rate decreases, the throughput difference between the two classes of flows increases, increasing the variation range of the per-flow throughput for large flows.

IV. TWO TCP THROUGHPUT MODELS

The experimental results show that there are significant differences in the per-flow throughput between large and small flows. Intuitively, one would expect that this may have something to do with how TCP congestion control works. It is well known that TCP has two distinct modes of increasing the congestion window: either exponentially during slow-start, or linearly during congestion-avoidance. We also expect that most small flows complete their transfers, or send most of their packets, during slow-start, while most large flows switch to congestion-avoidance at some earlier point.

We first analyze the results of the U_{3000} experiments to understand the relation between per-flow throughput and flow size. Figures 9, and 10 show this relation for two extreme values of the buffer size B : 30KB, and 38MB. Each of the points in these graphs is the average throughput of all flows in a given flow size bin. The bin width increases exponentially with the flow size (note that the x-axis is in logarithmic scale).

These graphs show that the average throughput increases with the flow size. Then, for the small buffer, the average throughput tends towards a constant value as the flow size increases (but with high variance). How can we explain and model these two distinct regimes, an increasing one followed by a constant?

One may first think that the increasing segment of these curves can be modeled based on TCP's slow-start behavior. Specifically, consider a flow of size s bytes, or $M(s)$ segments,

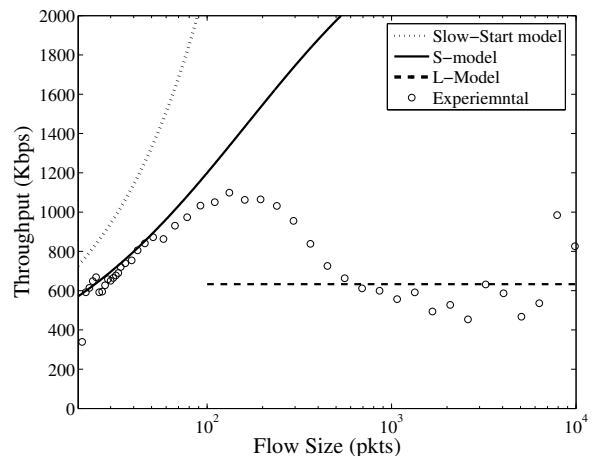


Fig. 9. Average per-flow throughput as a function of flow size for buffer size $B=30\text{KB}$.

with RTT T . If an ACK is generated for every new received segment (which is the case in the Linux 2.6.15 stack that we use), then the throughput of a flow that completes during slow-start can be approximated as $R_{ss}(s) = s/[T D(s)]$, where

$$D(s) = 1 + \lceil \log_2(M(s)/2) \rceil \quad (1)$$

is the number of RTTs required to transfer $M(s)$ segments during slow-start when the initial window is two segments and an additional RTT is needed for connection establishment. As shown in Figure 9, however, the slow-start model significantly overestimates the TCP throughput in the increasing phase of the curve.

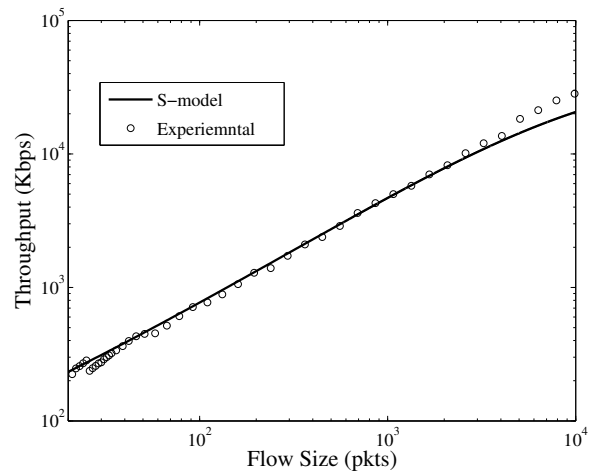


Fig. 10. Average per-flow throughput as a function of flow size for buffer size $B=38\text{MB}$.

A more detailed analysis of many flows in the “small size” range, revealed that a significant fraction of them are subject to one or more packet losses. Even though it is true that they usually send most of their packets during slow-start, they often also enter congestion-avoidance before completing. An exact analysis of such flows is difficult and it results in complex expressions (see [15] for instance). For our purposes, we need a simple model that can capture the increasing segment of

the average per-flow throughput with reasonable accuracy, and that can be used to derive the optimal buffer size. Therefore, we identified a simple empirical model that fits the increasing segment of the observed throughput values fairly well over a wide range of buffer sizes.

We refer to this empirical model as the *S-model*. According to the S-model, the average throughput of a flow with size s bytes is

$$R_S(s) = \frac{s}{T[D(s) + vpM(s)]} \quad (2)$$

where T is the flow's RTT, p is the packet loss rate, $D(s)$ is as defined earlier, and v is the number of additional RTTs that each retransmitted packet introduces. In the version of Linux that we use, which relies on SACKs, each dropped packet is usually recovered with Fast-Retransmit in one RTT, and so we set $v=1$.⁵

In Figures 9-10, we plot the S-model using the average RTT and loss rate observed in each experiment. Note that the S-model is an excellent approximation to the observed average per-flow throughput up to a certain flow size, which depends on the buffer size. Actually, in the case of the maximum buffer size (Figure 10), the S-model fits very well almost all flow sizes. The reason is that, with that buffer size, the loss rate is very low and so almost all flows, including the largest ones that send more than 10,000 packets, complete during slow-start.

In the case of the two lower buffer sizes, note that the experimental average per-flow throughput curves tend towards a size-independent value as the flow size increases beyond the scope of the S-model. In that range, flows send most of their packets during congestion avoidance. There are several models for that TCP regime. We choose to use the simplest, which is the well-known "square-root model" of [14], so that the derivations of the following sections are tractable. According to that model, which we refer to as the *L-model*, the average throughput for a flow in congestion avoidance is:

$$R_L = \frac{km}{T\sqrt{p}} \quad (3)$$

where m is the flow's Maximum Segment Size (MSS). Here k is a constant that depends on the exact variant of TCP [14] (we set $k=1.22$).

Figure 9 shows that the L-model gives a reasonable approximation for the average throughput of large flows. The variance is high, however, and the model applies only as long as the corresponding flows send most of their packets in congestion-avoidance.

One might expect that there is a specific size threshold that separates the scope of the S-model and L-model. Note, however, that this threshold would also depend on the buffer size, because the latter controls the packet loss probability. It is the loss probability, together with the flow size, that determines whether a flow will send most its packets in slow-start or congestion-avoidance. In general, the scope of the S-model expands towards larger flow sizes as we increase the buffer size, because the loss rate decreases and more larger flows

complete during slow-start. This is an interesting observation with significant implications on how we think about TCP "mice versus elephants". It is common that large TCP flows, say more than a few tens of KB, are viewed as "elephants" and they are modeled in congestion-avoidance. Slow-start, on the other hand, is viewed as important only for flows that send up to a few tens of packets. As the previous results show, however, *the mapping of small flows to slow-start and large flows to congestion-avoidance may be misleading, especially with larger buffer sizes.*

Finally, we attempted to find a quantitative criterion that can classify TCP flows as either following the S-model or the L-model. The best classifier, among many that we experimented with, is *the number of congestion events that a flow experiences*. A congestion event here is defined as one or more packet losses that are separated from other losses by at least two RTTs. Flows that saw at most four congestion events are reasonably close to the S-model, while flows that experienced five or more congestion events are closer to the L-model. It should be mentioned, however, that there is also a "grey region" of flow sizes that fall between the S-model and L-model and that cannot be approximated by either model. In the rest of the paper we ignore those flows and work entirely with the S-model and L-model,⁶ assuming that the former captures flows that sent most of their traffic in slow-start, while the latter captures flows that experienced more than four congestion events.

V. A SIMPLE CASE-STUDY

In the previous section, we identified two models that express the per-flow TCP throughput as a function of the loss probability and RTT that the flow experiences in its path. In this section, we consider a TCP flow of size s that goes through a single bottleneck link. The link has capacity C and B packet buffers. Our goal is to first derive the throughput $R(B)$ of the flow as a function of the buffer size at the bottleneck link, and then to calculate the buffer size that maximizes the throughput $R(B)$. To do so, we need to know the loss probability $p(B)$ and average queueing delay $d(B)$ as a function of B . As a simple case-study, even if it is not realistic, we consider the $M/M/1/B$ queueing model. Further, we focus on heavy-load conditions, when the link utilization is close to 100% for the two reasons we explained in §III: first, a closed-loop flow arrival model cannot generate overload, and second, the LRD nature of the traffic implies that there will be significant time periods of heavy-load even if the long-term average utilization is much less than 100%.

In the $M/M/1/B$ model, the loss probability is given by, $p(\rho, B) = (1 - \rho)\rho^B / (1 - \rho^{B+1})$. In the heavy-load regime, as ρ tends to 1, the loss probability becomes simply inversely proportional to the number of packet buffers $p(B) = 1/B$. The average queueing delay, in the heavy-load regime, becomes $d(B) = B/(2C)$. The average RTT of the TCP flow we consider can then be written as $T = T_o + B/2C$, where T_o

⁵Based on *ns-2* simulations, we found that the S-model also approximates TCP Reno quite well.

⁶The flows in the "grey region" contribute to less than 15% of bytes transferred in our experiments.

is the RTT of the flow excluding the queuing delays in the bottleneck link.

We can now substitute the previous expressions for the loss rate and RTT in the throughput equations for the S-model and L-model, (2) and (3), to derive the average throughput $R(B)$ as a function of the buffer size:

$$R_S(B) = \frac{s}{(T_o + \frac{B}{2C})[D(s) + \frac{vM(s)}{B}]} \quad (4)$$

$$R_L(B) = \frac{\sqrt{B}km}{(T_o + \frac{B}{2C})} \quad (5)$$

Figure 11 shows the throughput $R(B)$ for the S-model and the

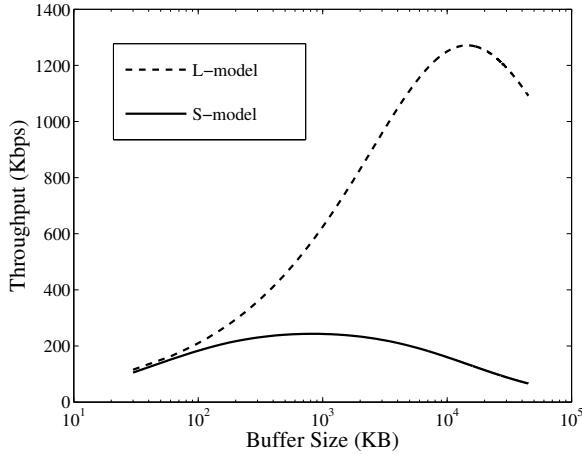


Fig. 11. Average throughput as a function of the router buffer size when the loss rate and the average queuing delay are given by the $M/M/1/B$ equations in the heavy-load regime. The bandwidth delay product here is 3750 KB.

L-model, in the case of a link with $C = 1\text{Gbps}$ and of a flow with $T_o = 60\text{ms}$ and $s=30\text{pkts}=45\text{KB}$. Note that both TCP models have an optimal buffer size \hat{B} at which the throughput is maximized.

The initial throughput increase as we increase B can be attributed to the significant reduction in the loss probability. Near the optimal buffer size, the gain in throughput due to loss rate reduction is offset by an increase in the queuing delay. Beyond the optimal buffer size the effect of the increasing queuing delays dominates, and the throughput is reduced in both the L-model and S-model. Further, note the optimal buffer size is much lower in the S-model case.

It is straightforward to derive the optimal buffer size \hat{B}_S and \hat{B}_L for the S-model and the L-model, respectively:

$$\hat{B}_S = \sqrt{\frac{2vM(s)}{D(s)}CT_o} \quad (6)$$

$$\hat{B}_L = 2CT_o \quad (7)$$

Interestingly, the optimal buffer size for the L-model is simply twice the bandwidth-delay product (BDP). On the other hand, the optimal buffer size for the S-model increases with the square-root of the BDP. This explains why the smaller flows that we considered in the experimental results have a lower optimal buffer size than the larger flows. For example, the optimal buffer size at a 1Gbps link with $T_o=60\text{ms}$ (BDP:

$CT_o=7.5\text{MB}$) is, first according to the S-model, 225KB for $s=10\text{KB}$, 450KB for $s=100\text{KB}$, and 1.125MB for $s=1\text{MB}$. According to the L-model, on the other hand, the optimal buffer size is $2CT_o$, which is equal to 15MB!

Clearly, the optimal buffer size at a network link heavily depends on whether the link is optimized for smaller flows that typically send most of their traffic in slow-start, or for bulk transfers that mostly live in congestion avoidance. It is interesting that, from the network operator's perspective, it would be better if all flows followed the S-model so that routers could also have much smaller buffer requirements.

VI. DELAY AND LOSS MODELS IN HEAVY LOAD

In the previous section, we derived closed-form expressions for the per-flow throughput $R(B)$ as a function of the buffer size for the simplistic case of the $M/M/1/B$ model. Of course in reality packets do not arrive based on a Poisson process and they do not have exponentially distributed sizes. Instead, the packet interarrival process exhibits significant correlations and burstiness even in highly multiplexed traffic [10], [13].

In this section, we aim to address the following question: *In the heavy-load regime ($\rho \approx 1$), are there simple functional forms for $p(B)$ and $d(B)$ that are reasonably accurate for LRD TCP traffic across a wide range of output/input capacity ratios and degrees of statistical multiplexing?* Given that the exact expressions for $p(B)$ and $d(B)$ could depend on several parameters that describe the input traffic and multiplexer characteristics, here we focus on “functional forms”, i.e., on general expressions for these two functions, without attempting to derive the exact dependencies between the involved parameters and $p(B)$ or $d(B)$. For instance, a functional form for the loss rate could be of the form $p(B) = aB^{-b}$, for some unknown parameters a and b . Recall that the reason we focus on the heavy-load regime is due to the LRD nature of the traffic: even if the long-term utilization is moderate, there will be significant time periods where the utilization will be close to 100%.

The mathematical analysis of queues with finite buffers is notoriously hard, even for simple traffic models. For instance, there is no closed-form expression for the loss rate in the simple case of the $M/D/1/B$ model [3]. Even asymptotic analysis (as B tends to infinity) is hard for arbitrary load conditions and general traffic models. On the other hand, it is often the case that good empirical approximations do exist in the heavy-load regime. For instance, see the Allen-Cunneen formula for the average queuing delay in the $G/G/1$ model [3].

The approach that we follow in this section is largely empirical and it is based, first, on extensive simulations, and second, on analytical reasoning. In particular, we examine whether we can approximate $p(B)$ and $d(B)$ by parsimonious functional forms in heavy-load conditions. The main conclusions of the following study are summarized as follows. 1) The queuing delay $d(B)$ can be approximated as linearly increasing with B (up to a certain cutoff point that depends on the maximum offered load), and 2) the loss rate $p(B)$ can be approximated as decreasing exponentially with B (i.e., $p(B) \approx ae^{-bB}$) or

as a power-law of B (i.e., $p(B) \approx aB^{-b}$), depending on the output/input capacity ratio. Next, § VI-A shows the simulation results that led us to these conclusions and § VI-B provides an analytical basis for these models and for the conditions under which they hold.

A. Simulation results

Figure 12 shows our $ns(2)$ simulation setup. There are

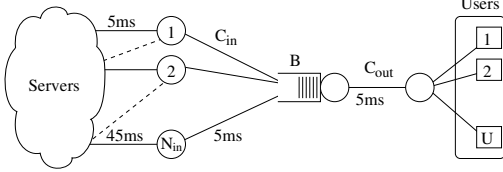


Fig. 12. Simulation setup

N_{in} input links, each with capacity C_{in} , feeding an output link that has capacity C_{out} and buffer size B . There are $\max(20, N_{in})$ servers that are connected to the input links with propagation delays that vary between 5ms and 45ms. The round-trip propagation delay T_o in this setup varies between 30ms and 110ms, with a harmonic mean of 60ms. There are U users in the system that create TCP transfers through the output link. Each user follows the closed-loop flow generation model, selecting a random server for each transfer. The transfer sizes follow a Pareto distribution with mean 80KB and shape parameter 1.5.

Of course, if $N_{in}C_{in} < C_{out}$ then there is no reason for buffering at the output link. So, we focus on the case that $N_{in} > \Gamma = C_{out}/C_{in}$. Also, U is set to a point that the offered load is always enough to saturate the output link, as long as B is sufficiently large. Because of the closed-loop nature of the traffic, the output link is saturated, but it is not overloaded. The simulation parameters are listed in Table I. Note that these simulation parameters can capture a wide variety of traffic multiplexers. A residential or office access link used by a small number of people can be well represented by $N_{in} = 2$, $U = 5$ and $\Gamma = 0.1$. Similarly, the parameter setting $N_{in} = 1000$, $U = 25$ and $\Gamma = 10$ can model the upstream link of a DSLAM packet multiplexer.

N_{in}	U	$\Gamma = C_{out}/C_{in}$	C_{out}	C_{in}	BDP
2	5	0.1	2.5Mbps	25 Mbps	6 pkts
20	5	0.1	2.5Mbps	25 Mbps	6 pkts
2	100	0.1	50Mbps	500 Mbps	125 pkts
20	100	0.1	50Mbps	500 Mbps	125 pkts
1000	25	10	10Mbps	1 Mbps	25 pkts
20	25	10	10Mbps	1 Mbps	25 pkts
1000	500	10	100Mbps	10 Mbps	250 pkts
20	500	10	100Mbps	10 Mbps	250 pkts

TABLE I
SIMULATION PARAMETERS

Due to space constraints, we only show a few typical results. Figures 13 and 15 show the loss rate $p(B)$ for a value Γ that is less than and larger than one, respectively.

Notice that the loss rate decreases in the case $\Gamma < 1$ almost linearly in a log-log plot (Figure 13), which means that the

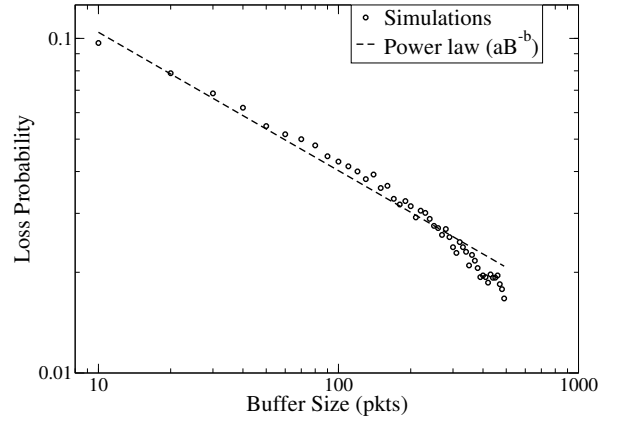


Fig. 13. Loss probability as a function of buffer size for low Γ ($N_{in}=20$, $U=100$, $\Gamma=0.1$, BDP=125 pkts).

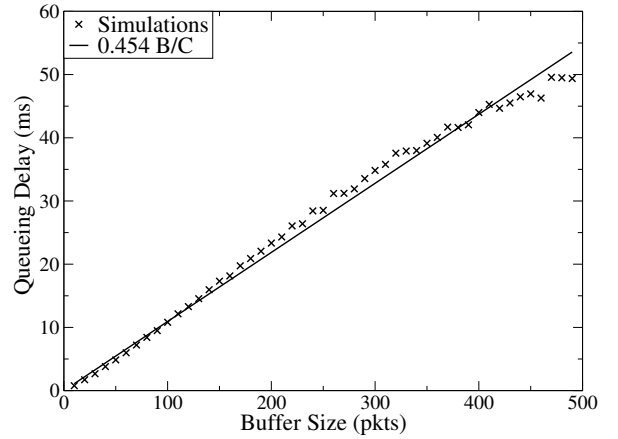


Fig. 14. Queueing delay as a function of buffer size for low Γ ($N_{in}=20$, $U=100$, $\Gamma=0.1$, BDP=125 pkts).

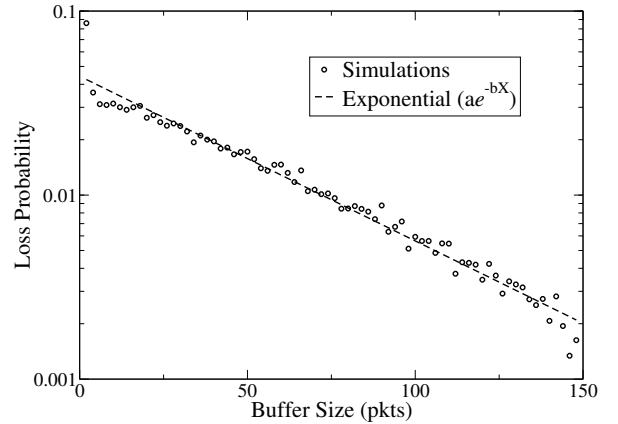


Fig. 15. Loss probability as a function of buffer size for high Γ ($N_{in}=1000$, $U=25$, $\Gamma=10$, BDP=25 pkts).

loss rate can be approximated by a power-law functional form, $p(B) = aB^{-b}$. On the other hand, Figure 15 shows the loss rate when $\Gamma > 1$. Here, the decrease is almost linear in a linear-log plot, which means that the loss rate can be modeled by an exponential functional form, $p(B) = ae^{-bB}$.

In terms of the average queueing delay, Figures 14 and 16 show that $d(B)$ increases almost linearly with B , up to a

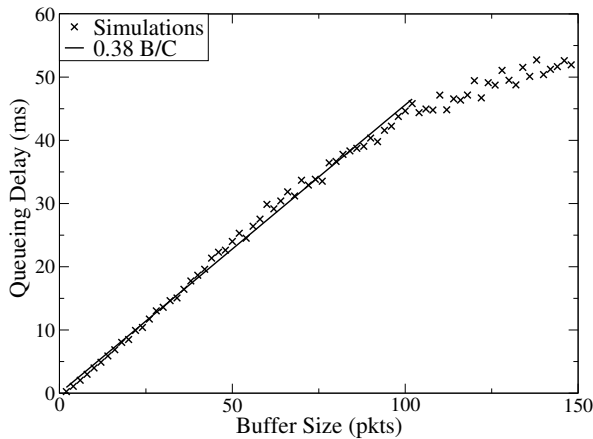


Fig. 16. Queuing delay as a function of buffer size for high Γ ($N_{in}=1000$, $U=25$, $\Gamma=10$, $BDP=25$ pkts).

certain cutoff point. After that point, $d(B)$ becomes almost constant with B , meaning that the offered load that the U users can generate is not enough to keep the buffer full. Increasing the buffer size beyond this cutoff point would not have a significant effect on the traffic. Consequently, we limit the scope of our loss rate and queuing delay models to the range in which the queuing delay increases almost linearly with B .

B. Analytical basis for loss rate models

In the following, we refer to the two functional forms for the loss rate as the *EX-form* $p(B) = ae^{-bB}$ and the *PL-form* $p(B) = aB^{-b}$. The fact that the loss rate can be modeled with these two expressions should not be surprising. Previous work, for the asymptotic analysis of the tail probability with various queuing models, has shown that the queuing tail probability can decay exponentially or as a power-law, depending on the characteristics of the input traffic [5]. We next explain how Γ affects the tail queuing probability with simple analytical arguments for $\Gamma \ll 1$ and $\Gamma > 1$. The following should certainly not be viewed as rigorous mathematical proofs. They do provide analytical insight, however, on the EX-form and PL-form approximations for the loss rate.

Consider a FCFS output link with capacity C_{out} , buffer size B packets, and N input links with capacity C_{in} . To further simplify, suppose that all packets have the same size m . The only assumption about the input traffic is that it generates *heavy-tailed burst sizes*, i.e., the probability that an input link will send a burst of more than x packets decays as a power-law of x , at least for large values of x . Previous measurement work has shown that TCP traffic exhibits strong burstiness and correlation structure in sub-RTT timescales [10].

$\Gamma \ll 1$: Let us assume that during a busy period of the output link, only one input link is active. Suppose that the active link sends a burst of R packets. In the time that it takes to transmit a single packet at the output link, $1/\Gamma$ packets can arrive to its buffer from the active link. So, the maximum queue size at the output link will be $R(1 - \Gamma)$, which is approximately equal to R because $\Gamma \ll 1$. So, because R follows a heavy-tailed distribution, the queue size distribution at the output link will also follow a heavy-tailed distribution.

Based on earlier results [5], we know that in that case the queuing tail probability $P[q > B]$ drops as a power-law of B . The loss rate $p(B)$, however, can be approximated by the queuing tail probability as long as the buffer size B is not too small. So, we expect the PL-form to be a good approximation for $p(B)$ as long as $\Gamma \ll 1$, the input traffic has heavy-tailed burst sizes, and the buffer size is sufficiently large.

$\Gamma > 1$: Suppose again that an input link sends a burst of R packets to the output link. The latter can transmit Γ packets at the time it takes to receive one packet from that input, and so the queue will be always empty. So, in this case we need to consider events where several input links are active in the same busy period of the output link. Let us further assume that the N input links are equally loaded and that they carry independent traffic. Say that X is the number of packets that arrive at the output link during each packet transmission period m/C_{out} . X can be viewed as a binomial random variable with parameters N and p , where p is the average utilization of each input link. For large N and small p , X can be approximated by a Poisson random variable. So, based on earlier results [5], [11], the queuing tail distribution $P[q > B]$ follows the EX-form. We can approximate the loss rate $p(B)$ by the EX-form, as long as the buffer size is not too small. In summary, we expect the EX-form to be a good approximation for $p(B)$ as long as $\Gamma > 1$, there are many, lightly loaded and independent input links, and the buffer size is sufficiently large.

The previous arguments do not cover several important cases. What happens when Γ is close to one? How does the degree of “heavy-tailedness” of the input traffic affect the PL-form approximation? In the case of the EX-form, what if the number of input links is low, or if some of the input links are heavily loaded, or if there are inter-link correlations? And finally, how good are these approximations for very small buffer sizes, say less than 10-20 packets?

We have examined these scenarios with simulations. The most interesting case is what happens when Γ is close to one. In that case, the degree of “heavy-tailedness” of the input traffic is the key factor. Suppose that α is the shape parameter of the Pareto flow size distribution. The variability of this distribution increases as α decreases, and the distribution becomes heavy-tailed when $\alpha < 2$. If $\Gamma = 1$, then the PL-form is still a good approximation as long as α is less than about 1.5. If α is close to 2, then the PL-form is *not* a good approximation even when Γ is as low as 0.25. On the other hand, the EX-form is a good approximation even for low values of α (say 1.25) as long as $\Gamma > 1$. It is important to note that when α is close to 2 and Γ is less than but close to 1, then neither the PL-form nor the EX-form are good approximations. In summary, when $\Gamma > 1$ the EX-form is a good approximation independent of the heavy-tailedness of the input traffic. The PL-form approximation, on the other hand, requires that both $\Gamma < 1$ and that the input traffic from each source is heavy-tailed.

VII. OPTIMAL BUFFER SIZE

In the previous section, we proposed functional forms for the average queuing delay and loss rate. The former is a linear

function of the buffer size, $d(B) = fB/C$, up to a certain point determined by the maximum offered load. The latter is either the *EX-form* $p(B) = ae^{-bB}$ or the *PL-form* $p(B) = aB^{-b}$. In this section, we derive expressions for (1) the average per-flow TCP throughput $R(B)$ as a function of the buffer size in the heavy-load regime, and (2) the optimal buffer size \hat{B} , i.e., the minimum value of B that maximizes the average per-flow TCP throughput. These expressions are derived for both TCP throughput models (L-model and S-model) and for both loss rate forms (EX-form and PL-form). We also compare the optimal buffer size obtained from these expressions with the optimal buffer size that results from the corresponding $ns - 2$ simulations. Finally, we show that the optimal average per-flow throughput is not overly sensitive to the buffer size. Thus in practice, a network operator can get almost optimal performance without having to fine tune the buffer size to an exact optimal value.

A. PL-form

First, consider the case that the loss rate decreases as a power-law of the buffer size,

$$p(B) = aB^{-b} \quad (8)$$

where a and b are positive constants. The queuing delay is modeled by a linear function, and so the RTT $T(B)$ is given by

$$T(B) = T_o + f \frac{B}{C} \quad (9)$$

where T_o is the round-trip propagation delay (excluding queuing delays) at the bottleneck link, C is the output link capacity, and f is a positive constant.

1) *L-model*: In the L-model, the throughput $R(B)$ is given by

$$R(B) = \frac{km}{\sqrt{aB^{-b} (T_o + f \frac{B}{C})}} \quad (10)$$

After setting the derivative of $R(B)$ to zero we find that the optimal buffer size \hat{B} is:

$$\hat{B} = \frac{b}{f(2-b)} CT_o \quad (11)$$

The second derivative confirms that this is indeed a maximum.

Equation (11) shows that the maximum per-flow throughput is positive when $b < 2$. In our simulations, we observed that this is always the case, and that typical values for b and f are around 0.5 and 0.4, respectively. This makes \hat{B} approximately $0.83CT_o$. Also note that the optimal buffer size is independent of the parameter a . What determines the value of \hat{B} is the rate b at which the loss rate decays with B , rather than the absolute value of the loss rate.

2) *S-model*: In the S-model, the throughput $R(B)$ is given by

$$R(B) = \frac{s}{[D(s) + v M(s) a B^{-b}] (T_o + f \frac{B}{C})} \quad (12)$$

where $D(s)$, v , and $M(s)$ are the previously defined S-model parameters for a flow of size s . In the following, we set $v = 1$ (as discussed in §IV).

Again, after calculating the first two derivatives, we find that the optimal buffer size \hat{B} is the solution of the following equation:

$$[ab M(s) CT_o] B^{-(1+b)} = a M(s) f(1-b) B^{-b} + fD(s) \quad (13)$$

We do not have a closed-form solution for this equation. With the parameter values that result from our simulations, however, we observed that its numerical solution is always positive.

3) *Remarks for the PL-form case and an example*: For the $M/M/1/B$ model under heavy-load, the loss rate conforms to the PL-form with $a = 1$ and $b = 1$, and the delay coefficient is $f = 1/2$. For these parameter values, (11) reduces to $\hat{B} = 2CT_o$, while (13) gives $\hat{B} = \sqrt{\frac{2M(s)}{D(s)} CT_o}$. These are the same expressions we derived in §V.

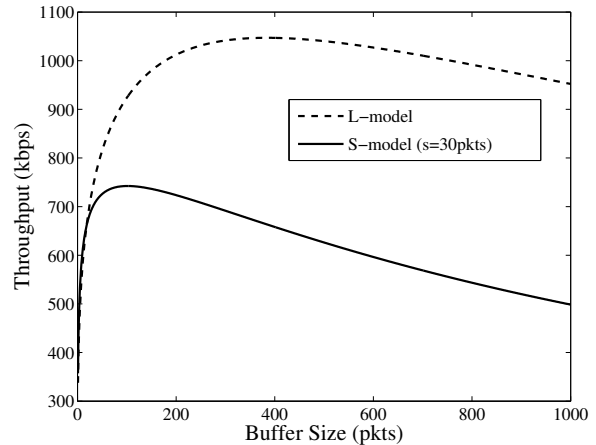


Fig. 17. TCP throughput for the S-model and L-model when the loss rate is given by the PL-form (BDP = 250 pkts).

Figure 17 shows $R(B)$ for the S-model and the L-model when the loss rate is modeled by the PL-form. The capacity C and the propagation delay T_o in this example are 50Mbps and 60ms, respectively. The model parameters for the loss rate and the queuing delay are taken from the simulation with $N_{in}=20$, $U=100$ and $\Gamma=0.1$. The flow size (for the S-model) is $s=30$ packets. Note that the optimal buffer size with the S-model is significantly lower than with the L-model (about 100 packets versus 400 packets, respectively).

B. EX-form

In this case, the loss rate $p(B)$ is given by

$$p(B) = ae^{-bB} \quad (14)$$

where a and b are positive constants and the RTT $T(B)$ is given by (9).

1) *L-model*: The per-flow throughput for the L-model under the EX-form is

$$R(B) = \frac{km}{\sqrt{ae^{-bB} (T_o + f \frac{B}{C})}} \quad (15)$$

It is easy to show that the first derivative becomes zero when

$$\hat{B} = \frac{2}{fb} (f - \frac{bCT_o}{2}). \quad (16)$$

The second derivative shows, however, that this buffer size corresponds to *minimum* throughput. The buffer size that leads to maximum throughput, in this case, is either zero (given that the buffer size cannot be negative) or ∞ , depending on the sign of (16). Specifically, if dR/dB is negative at $B = 0$, then the buffer size of (16) is positive and it corresponds to minimum throughput, while the buffer size that gives maximum throughput is negative. In that case, it is best to set the buffer size to zero ($\hat{B} = 0$). Otherwise, if dR/dB is positive at $B = 0$, the buffer size of (16) is negative, the throughput keeps increasing with the buffer size, and the optimal buffer size is, theoretically at least, $\hat{B} \rightarrow \infty$.

With the parameter values obtained from our simulations (except when $N_{in}=20$, $U=25$ and $\Gamma=10$, the case where the offered load is too small to generate any significant queuing and loss rate), we find numerically that the optimal buffer size in this case is $\hat{B} = 0$.

2) *S-model*: Similarly for the S-model, the throughput is given by

$$R(B) = \frac{s}{[D(s) + vM(s)ae^{-bB}](T_o + f\frac{B}{C})} \quad (17)$$

Setting the first derivative of $R(B)$ to zero gives the following equation

$$\frac{fD(s)}{vM(s)} + (af - abCT_o)e^{-bB} = abfBe^{-bB} \quad (18)$$

The previous equation does not always have a unique root, making it hard to argue for the location of the global maximum of $R(B)$. Given specific parameter values, however, it is straightforward to determine numerically the optimal buffer size \hat{B} . As in the L-model case, with the parameter values obtained from our simulations (except when $N_{in}=20$, $U=25$ and $\Gamma=10$), we find numerically that the optimal buffer size is $\hat{B} = 0$.

3) *Remarks for the EX-form case and an example*: Figure 18 shows $R(B)$ for the S-model and the L-model when the loss rate is modeled by the EX-form. The capacity C and the propagation delay T_o in this example are 100Mbps and 60ms, respectively. The model parameters for the loss rate and the queuing delay are taken from the corresponding simulation with $N_{in}=1000$, $U=500$ and $\Gamma=10$. The flow size (for the S-model) is $s=30$ packets.

Note that in both cases, S-model and L-model, the optimal buffer size is zero. Even though it is mathematically possible (as explained earlier) to have a non-zero, or even infinite optimal buffer size in the EX-form case, in all our simulations the optimal per-flow throughput is obtained when the buffer size is zero or very low (less than 0.1 of BDP). This is a major difference between the EX-form and the PL-form, and it reflects how important the output/input capacity ratio is in the buffer sizing problem.

C. Comparison between simulation and analytical results

Here, we compare the optimal buffer size that results from the previous analytical expressions with corresponding simulation results. Table II shows the optimal buffer size obtained from (11) and (13) for the PL-form; the theoretical

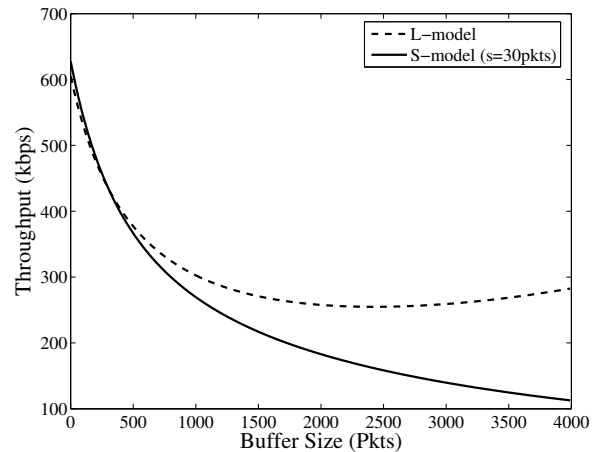


Fig. 18. TCP throughput for the S-model and L-model when the loss rate is given by the EX-form (BDP = 500 pkts).

optimal buffer size for the EX-form is zero. The tabulated results correspond to the case where $N_{in} = 100$, $U = 20$, $\Gamma = 0.1$ and BDP = 125 for the PL-form. For the EX-form, we set $N_{in} = 1000$, $U = 500$, $\Gamma = 10$ and BDP = 250. For the S-model, we consider flows of size $s=30$ pkts and for the L-model we consider all flows that experience more than five congestion events. To estimate the throughput-versus-buffer size curve that results from simulation results, we measure the median per-flow throughput for the previous sets of flows. Then, we estimate the optimal buffer size as the (minimum) buffer size that maximizes the median per-flow throughput.

Note that the optimal buffer size that results from analysis is similar to that obtained from simulations. In the case of the PL-form with L-model flows (PL-L), the optimal buffer size is about 60-70% of the BDP. With S-model flows (PL-S), the optimal buffer size drops to a smaller fraction (20-25%) of the BDP. In the case of the EX-form, the theoretical results predict that it is best to not have any buffers, as they ignore the effects of packet-level TCP bursts. In the simulation results, we see that a small buffer size of 20-30 packets leads to maximum per-flow throughput.

Model	Simulation	Analysis
	in pkts (BDP fraction)	in pkts (BDP fraction)
PL-L	170 ± 10 (0.68 ± 0.04)	146 (0.584)
PL-S	64 ± 2 (0.256 ± 0.008)	47 (0.188)
EX-L	30 ± 2 (0.06 ± 0.004)	0 (0)
EX-S	20 ± 2 (0.04 ± 0.004)	0 (0)

TABLE II
OPTIMAL BUFFER SIZE FROM SIMULATIONS AND FROM ANALYSIS.

In practice, it would be difficult to fine-tune the buffer size at exactly the optimal value that (11) or (13) predict, as that would require the real-time estimation of the parameters a , b and f . While we know from our simulations that these parameters depend on traffic characteristics, we do not have a model that can relate these parameters to traffic characteristics that are easy to measure.

On the other hand, we do not expect that network operators will need to estimate these parameters accurately. Instead, we recommend that network operators set the buffer size

depending on the capacity ratio and the policy of favoring slow-start (S-model) or congestion-avoidance (L-model) flows. If $\Gamma > 1$, then the buffer size can be simply set to just few (say 20-30) packets, independent of the BDP. If $\Gamma < 1$, the buffer size can be set to either a small fraction of the BDP (S-model) or in the order of the BDP (L-model), depending on whether the operator wants to optimize for S-model or L-model flows.

D. Sensitivity analysis

So far, we have given expressions for the optimal buffer size when the PL-form holds. The optimal buffer size with the EX-form, on the other hand, is usually zero (or close to zero), and we will not consider it further here.

Additionally, an important question is: *what is the relative reduction in the per-flow throughput when the buffer size is within a given fraction, say ω , of the optimal size \hat{B} ?* Or, in other words, how steep is the $R(B)$ curve around the optimal point?

To answer this question, we rely on the expressions for $R(B)$ for the L-model and the S-model, in the case of the PL-form. We calculate the per-flow throughput $R(B)$ at the buffer size $B = \omega\hat{B}$. Then, we report the relative error between $R(\omega\hat{B})$ and $R(\hat{B})$:

$$e_R(\omega) = \frac{R(\hat{B}) - R(\omega\hat{B})}{R(\hat{B})} \quad (19)$$

For the L-model, $e_R(\omega)$ is equal to

$$e_R^L(\omega) = 1 - \frac{2\omega^{0.5b}}{2 + b(\omega - 1)} \quad (20)$$

Note that the relative error does not depend on the delay parameter f . Only the loss probability decay factor b matters.

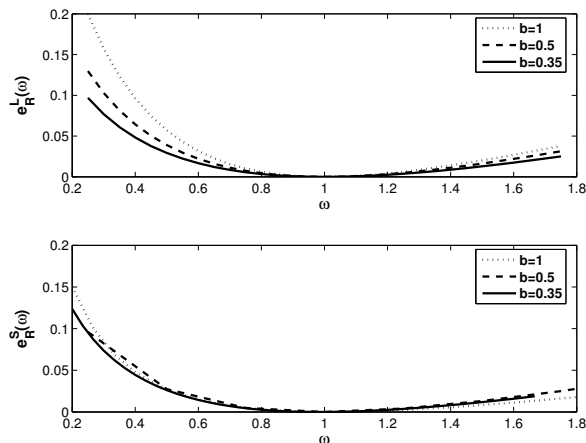


Fig. 19. Sensitivity of the throughput to errors in the optimal buffer size, for the L-model (top) and the S-model (bottom).

For the S-model, we do not have a closed-form expression for the optimal buffer size, and so we rely on a numerical calculation of the relative error $e_R^S(\omega)$.

Figure 19 shows $e_R^L(\omega)$ and $e_R^S(\omega)$ for the L-model and S-model, respectively, as ω varies from 0.25 to 1.75. We choose three values of b : 0.35, 0.5 and 1.0. Recall that $b=1.0$ corresponds to the $M/M/1/B$ model. Notice that the error is

higher when we underestimate the optimal buffer size rather than overestimate it. However, the relative error is quite low around the optimal buffer size, and it remains below 10%-15% even when we set the buffer to 40% of the optimal size.

VIII. CONCLUSIONS - HOW SMALL IS TOO SMALL?

Recently, there has been an interesting debate regarding the sizing of router buffers. Earlier in the paper (§II) we summarized the key points and opinions in this debate. In this section, we put the results of this paper in the context of that debate.

First, we emphasize that this paper does not focus only on link utilization. Having the minimum amount of buffering to keep the utilization high is an objective that does not take into account the performance of the major transport protocol TCP and of most applications.

The work presented here provides further evidence that the buffer provisioning formula that sets the buffer size equal to the link's BDP is probably far from optimal. In several of our simulation and modeling results, we observed that the optimal buffer size is much less than the BDP. The BDP rule-of-thumb only applies in the very special case that the link is saturated by a single persistent TCP connection, and so it can be quite misleading in most practical cases. From this point of view, we agree with [1] that the buffer size can be significantly less than the BDP when a link carries many flows.

Previous buffer sizing research has focused on the number N of large flows sharing a link [1], [6]. Practically, however, the number of flows N is a rather ill-defined concept in the context of buffer sizing, because it is not clear which TCP flows should be included in N . As shown in this paper, TCP flows can behave very differently, and that is not strictly based on their size. Even very large flows can conform to the S-model if the loss rate is sufficiently low.

Our results are in agreement with earlier work [8], which suggests that the buffer size of some links can be significantly reduced to as low as a few dozen packets. As we showed in §VI, this is the case when the output/input capacity ratio is larger than one, and the loss rate drops exponentially with the buffer size. However, we disagree with [8] about the reasons that allow for this decreased buffer size. The buffer decrease when $\Gamma > 1$ is not related to TCP's maximum window size and it does not require TCP pacing or moderate utilization.

We observe that in some cases, especially in links where the capacity ratio Γ is much lower than one, the buffer requirement can still be a significant fraction of the BDP, especially when the link mostly carries L-model flows. We expect these conditions to be true in some links at the periphery of the network. Special attention should be given to edge links of server farms in the outgoing direction (e.g., from 10GigE server ports to an 1GigE edge link), and to customer access links in the incoming direction, (e.g., from OC-48 core links to an OC-3 customer access link). The recent study by Lakshmikantha et al. [12], which was done independently and in parallel with our work, agrees with these predictions.

Finally, we point out that it is difficult to arrive at a simple and "handy" formula that one can use for sizing the buffers

of any router interface. We hope to have conveyed to the reader that such a formula may not exist in practice. The optimal buffer size at an Internet link depends on several parameters that are related to both the offered load (flow size distribution, types of TCP traffic, etc) and to network design (capacity ratios, degree of statistical multiplexing, etc). The most practical recommendation we can give to network operators is that they should first determine the capacity ratio Γ of their links, and to decide whether they will optimize the throughput of connections that are in slow-start (S-model) or in congestion avoidance (L-model). Then, the recommended buffer size is a small number of packets if $\Gamma > 1$, a low fraction of the BDP if $\Gamma < 1$ and the S-model is used, or in the order of the BDP if $\Gamma < 1$ and the L-model is used. Additionally, we have provided evidence that the per-flow throughput does not drop much when the buffer size deviates from its optimal value, and especially when it is slightly overestimated. This means that fine-tuning the buffer size would not be needed in practice, as long as the structural and traffic characteristics of a given link do not change significantly.

ACKNOWLEDGMENTS

We are grateful to Jesse Simsarian for his help in setting up the testbed. This work was partially supported by the NSF CAREER award ANIR-0347374.

REFERENCES

- [1] G. Appenzeller, I. Keslassy, and N. McKeown. Sizing Router Buffers. In *ACM Sigcomm*, 2004.
- [2] A. Berger and Y. Kogan. Dimensioning Bandwidth for Elastic Traffic in High-Speed Data Networks. *IEEE/ACM Transactions on Networking*, 8(5):643–654, 2000.
- [3] G. Bolch, S.Greiner, H.Meer, and K.S.Trivedi. *Queueing Networks and Markov Chains*. John Wiley and Sons, 1999.
- [4] M. Carson and D. Santay. NIST Net - A Linux-Based Network Emulation Tool. *ACM CCR*, 33(3):111–126, 2003.
- [5] T. Daniels and C. Blondia. Tail Transitions in Queues with Long Range Dependent Input. In *IFIP Networking*, 2000.
- [6] A. Dhamdhere and C. Dovrolis. Buffer Sizing for Congested Internet Links. In *IEEE Infocom*, 2005.
- [7] A. Dhamdhere and C. Dovrolis. Open Issues in Router Buffer Sizing. *ACM CCR*, 36(1):87–92, 2006.
- [8] M. Enachescu, Y. Ganjali, A. Goel, T. Roughgarden, and N. McKeown. Part III: Routers with Very Small Buffers. *ACM CCR*, 35(3):83–90, 2005.
- [9] Y. Ganjali and N. McKeown. Update on Buffer Sizing in Internet Routers. *ACM CCR*, 36(5):67–70, 2006.
- [10] H. Jiang and C. Dovrolis. Why is the Internet Traffic Bursty in Short (Sub-RTT) Time Scales? In *ACM Sigmetrics*, 2005.
- [11] H. S. Kim and N. B. Shroff. Loss Probability Calculations and Asymptotic Analysis for Finite Buffer Multiplexers. *IEEE/ACM Transactions on Networking*, 9(6):755 – 768, 2001.
- [12] A. Lakshminantha, R. Srikant, and C. Beck. Impact of File Arrivals and Departures on Buffer Sizing in Core Routers. In *IEEE Infocom*, 2008.
- [13] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the Self-Similar Nature of Ethernet Traffic (Extended Version). *IEEE/ACM Transactions on Networking*, 2(1):1–15, Feb. 1994.
- [14] M. Mathis, J. Semke, J. Madhavi, and T. Ott. The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm. *ACM CCR*, 27(3):67–82, 1997.
- [15] M. Mellia, I. Stocia, and H. Zhang. TCP Model for Short Lived Flows. *IEEE Communications Letters*, 6(2):85–87, 2002.
- [16] R. Morris. TCP Behavior with Many Flows. In *IEEE ICNP*, 1997.
- [17] R. Morris. Scalable TCP Congestion Control. In *IEEE Infocom*, 2000.
- [18] R. S. Prasad and C. Dovrolis. Measuring the Congestion Responsiveness of Internet Traffic. In *PAM*, 2007.
- [19] R. S. Prasad and C. Dovrolis. Beyond the Model of Persistent TCP Flows: Open-Loop vs Closed-Loop Arrivals of Non-Persistent Flows. In *ANSS*, 2008.
- [20] G. Raina, D. Towsley, and D. Wischik. Part II: Control Theory for Buffer Sizing. *ACM CCR*, 35(3):79–82, 2005.
- [21] B. Schroeder, A. Wierman, and M. Harchol-Balder. Closed Versus Open: A Cautionary Tale. In *USENIX NSDI*, 2006.
- [22] J. Sommers and P. Barford. Self-Configuring Network Traffic Generation. In *ACM/USENIX IMC*, 2004.
- [23] C. Villamizar and C.Song. High Performance TCP in ANSNET. *ACM CCR*, 24(5):45–60, 1994.
- [24] G. Vu-Brugier, R. Stanojevic, D. Leith, and R. Shorten. A Critique of Recently Proposed Buffer Sizing Strategies. *ACM CCR*, 37(1), 2007.
- [25] W. Willinger, M.S.Taqqu, R.Sherman, and D.V.Wilson. Self-Similarity Through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level. In *ACM Sigcomm*, 1995.
- [26] D. Wischik and N. McKeown. Part I: Buffer Sizes for Core Routers. *ACM CCR*, 35(3):75–78, 2005.



Dr. Ravi S. Prasad is a software engineer at Cisco Systems, Inc. He received the Ph.D. degree in Computer Science from the College of Computing at Georgia Institute of Technology in May 2008. He received the M.S. degree in Civil Engineering from the University of Delaware in 2001 and the B.Tech. degree in Ocean Engineering and Naval Architecture from Indian Institute of Technology, Kharagpur, India, in 1998. His current research interests include router buffer sizing, bandwidth estimation methodologies, network measurements and applications, and TCP in high bandwidth networks.



Dr. Constantine Dovrolis is an Associate Professor at the College of Computing of the Georgia Institute of Technology. He received the Computer Engineering degree from the Technical University of Crete in 1995, the M.S. degree from the University of Rochester in 1996, and the Ph.D. degree from the University of Wisconsin-Madison in 2000. He joined Georgia Tech in August 2002, after serving at the faculty of the University of Delaware for about two years. He has held visiting positions at Thomson Research in Paris, Simula Research in Oslo, and FORTH in Crete. His current research focuses on the evolution of the Internet, intelligent route control mechanisms, and on applications of network measurement. He is also interested in network science and in applications of that emerging discipline in understanding complex systems. Dr. Dovrolis has been an editor for the ACM SIGCOMM Computer Communications Review (CCR). He served as the Program co-Chair for PAM'05, IMC'07, and as the General Chair for HotNets'07. He received the National Science Foundation CAREER Award in 2003.



Dr. Marina Thottan is a Member of Technical Staff in the Center for Networking Research at Bell Laboratories, Alcatel-Lucent, Murray Hill, NJ. She holds a Ph.D. in Electrical and Computer systems engineering from Rensselaer Polytechnic Institute in Troy, NY. Dr. Thottan is active in the fields of wire line networking and network management and has served as a program committee member for several conferences in these areas. Her research publications have appeared in a number of ACM, and IEEE conferences and journals. Her current research interests are in the areas of novel network and switch architectures and high speed optical networks. She is a member of the IEEE and the ACM.