Open Issues in Router Buffer Sizing

Amogh Dhamdhere
Networking and Telecommunications Group
College of Computing
Georgia Institute of Technology
amogh@cc.gatech.edu

Constantine Dovrolis
Networking and Telecommunications Group
College of Computing
Georgia Institute of Technology
dovrolis@cc.gatech.edu

ABSTRACT

Recent research results suggest that the buffers of router interfaces can be made very small, much less than the link's bandwidth-delay product, without causing a utilization loss, as long as the link carries many TCP flows. In this letter we raise some concerns about the previous recommendation. We show that the use of such small buffers can lead to excessively high loss rates (up to 5%-15% in our simulations) in congested access links that carry many flows. Even if the link is fully utilized, small buffers lead to lower throughput for most large TCP flows, and significant variability in the per-flow throughput and transfer latency. We also discuss some important issues in router buffer sizing that are often ignored.

1. INTRODUCTION

Provisioning router (or switch) buffers has long been considered a "black art". Typically, the size of router buffers is set to either a default value specified by the manufacturer, or it is determined by the well-known "Bandwidth-Delay Product" (BDP) rule-of-thumb [7]. The latter states that the buffer size should be equal to the capacity C of the link multiplied by the Round-Trip Time (RTT) T of a TCP connection that can be bottlenecked at that link. A recent publication by Appenzeller et al., which we refer to as the "Stanford model", has challenged the bandwidth-delay product rule advocating the use of much smaller buffers [1]. The Stanford model and its implications have been further discussed in three recent CCR letters [3, 6, 8]. The main objective of the Stanford model is to achieve the full utilization of a network link that carries many TCP flows with minimum buffering, and so with minimum queueing delay. The authors of [1] showed analytically, with simulations, and with testbed experiments that a buffer size B, equal to $CT/\sqrt{N}$, where N is the number of long TCP flows in the bottleneck, is sufficient to saturate that link. Since N is typically large, especially in backbone links, the Stanford model can have major implications in router design, reducing significantly the amount of required fast memory (SRAM), power consumption and cost.

In this letter, we raise some concerns about the Stanford model and its implications in network and application performance. The main point that we would like to draw attention to is that the Stanford model can lead to an excessively

high loss rate, up to 5%-15% in our simulations, in congested access links that carry many flows. Such frequent packet losses can be detrimental for certain applications. Furthermore, we show that even though the Stanford model provides enough buffering to achieve full utilization, it results in lower throughput for most large TCP transfers. Also, the use of small buffers can significantly increase the throughput variability of TCP flows, making application-layer performance less predictable. We also discuss a number of important issues about router buffer sizing that are often ignored. The choice of the performance objective, whether the link is "saturable" or not (defined in Section 3), the exact type of traffic, and the way we estimate the number of long TCP flows in a link, are issues that can have a major impact in the buffer sizing process.

The rest of this letter is structured as follows. We briefly summarize the Stanford model in Section 2. In Section 3, we discuss several open issues regarding buffer sizing. Section 4 presents simulation results for the aggregate loss rate and for the throughput and loss rate of individual TCP flows, comparing the Stanford model with BDP-based buffer sizing. We conclude in Section 5.

2. THE STANFORD MODEL

In the following, we use the following terminology and notation. The link that is being provisioned is referred to as the *target link*. The capacity of the target link is C bits/second. N is the number of *long TCP flows* at the target link. A flow is classified as "short" in [1] if it never leaves slow-start, and flows that are not short are classified as "long". Typically, the latter spend a significant portion of their lifetime in congestion avoidance. A more precise definition of a "long flow", or the variability of N with time, are two important open issues in [1] and they are further discussed in Section 3.4. The average RTT of each long TCP flow is denoted by T_i, $i = 1 \ldots N$.

The Stanford model focuses on a link that carries almost exclusively TCP flows. This is probably a valid assumption, given that several measurement studies have shown that TCP accounts for more than 90% of the Internet traffic in bytes. The objective of the Stanford model was to derive the minimum buffer size B required for full utilization at the target link, despite the "sawtooth variations" caused by TCP's congestion control. The key result of [1] is that B decreases inversely proportional to the square root of the number of long TCP flows N. Specifically,

$$B = \frac{CT}{\sqrt{N}} \qquad (1)$$

The basic idea behind (1) is that if the target link carries a large number of independent TCP flows, then based on the central limit theorem the aggregate window size will follow the Normal distribution. Note that even though Drop-Tail buffers tend to cause loss synchronization among different flows, [1] showed that when N is in the hundreds, the underlying TCP flows have practically independent windows. From the Normality of the aggregate window size, it follows that the distribution of the queue size at the target link is also Normal. The mean and standard deviation of the queue size distribution can be expressed as functions of the number of flows N, the BDP CT, and the buffer size at the target link B. In these derivations, the assumption is that all N flows go through a linear "TCP sawtooth" in which the window size varies uniformly around its average value $\bar{W}$, between $2/3\bar{W}$ and $4/3\bar{W}$. Given the distribution of the queue size at the target link, [1] derives the minimum buffer size required to ensure that the link will stay fully utilized.

Furthermore, [1] examined the case that the target link only carries "short flows", i.e., flows that never leave slow-start. The key result for that case is that the average queue length depends on the offered load (i.e., the product of the flow arrival rate and average flow size, divided by the link capacity), the flow size, and the TCP slow-start burstiness (i.e., the length and rate of packet trains during slow-start). When the target link carries a mix of both long and short TCP flows, [1] argued that small flows do not have a significant effect on the required buffer size, and that B can be determined based on the number of long flows, as in (1).

3. BUFFER SIZING ISSUES

3.1 What is the performance objective?

The problem of router buffer sizing is intimately related to the performance objectives we aim to meet. These objectives can be related to network-layer performance (e.g., full utilization, maximum delay constraint, maximum loss rate constraint), or application-layer performance (e.g., TCP transfer throughput, Web page download latency, VoIP session quality). Different objectives can lead to very different results for the optimal buffer size, and it is unlikely that a specific buffer size can be acceptable for all reasonable performance objectives.

The Stanford model aims to achieve full link utilization with the minimum buffer space, and consequently with the minimum queueing delay. As we show in Section 4, however, the Stanford model can lead to a significant loss rate, up to 5%-15% in our simulations. In terms of network-layer performance, we believe that such high loss rates would be unacceptable for most network operators today. Also, a high loss rate can cause frequent retransmissions and timeouts in TCP-based applications, which is particularly detrimental for applications that require both reliability and interactivity. Furthermore, audio and video applications that use high-compression encoders can suffer under such high loss rates.

Another buffer sizing approach is to achieve full link utilization with a maximum loss rate constraint. This was the performance objective in the BSCL (Buffer Sizing for Congested Internet Links) scheme, proposed in [2]. In that work, the authors exploited the trade-off between loss rate and RTT, and derived a buffer size requirement that increases proportionally to the number of long TCP flows. The main problem with BSCL is that, even though it limits the loss rate, it requires a large buffer space and it can lead to excessive queueing delays, especially in lower capacity links. Large queueing delays are detrimental for interactive applications, and they can also affect Web-like TCP flows (the throughput of which is determined by the RTT if the flow does not experience losses).

A third buffer sizing approach is to control the maximum queueing delay at the target link. For example, for a maximum queueing delay of 10msec at an OC-12 link (C=620Mbps), the buffer space at the target link should not be larger than 6.2Mb. This approach however can lead to both underutilization, especially in the presence of long TCP flows, and a significant loss rate.

Note that the various existing Active Queue Management (AQM) schemes do not address the following aspect of the buffer sizing problem. Even though AQM schemes can stabilize the queue size at the target link and avoid loss synchronization, they cannot control the loss rate at the target link. This is further discussed in [2].

3.2 Which links are saturable?

It is important to understand that the Stanford model, as well as the BSCL scheme, are only applicable when the target link is "saturable". We say that a link is *saturable* when the offered load in that link is sufficiently high to saturate the capacity of that link, given a sufficiently large buffer space B. Note that a link may be saturable only for limited time periods (e.g., during the afternoon hours). In that case, the buffer space of the target link could be provisioned based on those "peak load" periods. There are links however that *can never be saturable* due to constraints on the maximum rate of their input flows. We believe, even though we have no experimental evidence to support this claim, that most backbone links in the Internet today are not saturable because of the general move towards capacity overprovisioning.

We next mention examples of links that would never be saturable. First, consider a Fast Ethernet link (C=100Mbps) that serves 60 DSL users, each with a maximum download capacity of 1Mbps. The average offered load at the Fast Ethernet link cannot exceed 60%, and so this link is not saturable. Second, consider an OC-12 link that carries 100 long TCP flows. Suppose that each flow has an advertised window limit of 32KB. If the RTT of each flow is 100msec then the maximum offered load that these flows can generate is about 260Mbps, much less than the OC-12 capacity. Third, even if the TCP flows of the previous example are not limited by the receiver's advertised window, they can be bottlenecked elsewhere in their path. For instance, their throughput can be limited by the capacity of their "last-mile" DSL or cable modem link.

If a link is not saturable, then the sole purpose of its buffers is to avoid losses upon the arrival of packet bursts. From this point of view, we agree with the authors of [1] that high-capacity backbone links require a very small number of buffers. This is not related however to the Stanford model. Instead, it is because such backbone links are usually non-saturable, and so their queue size distribution depends on the characteristics of the arriving packet bursts.

If backbone links are not saturable, then what is the type of links that may be saturable during some time periods? It is likely that most residential access links today are sat-

urable. Also, the access links of companies and organizations that cannot afford to overprovision their Internet connectivity may also be saturable. For example, consider the access link of a college or small university. Typically, such links have a capacity that is around 10-100Mbps and they can carry traffic for hundreds of users. The authors are aware of several such congested links, especially in Europe.

3.3 How to count the number of TCP flows?

A key input parameter in the Stanford or BSCL models is the number N of long TCP transfers. As previously mentioned, a flow is considered "short" if it never exits the slow-start phase. A "long" flow, on the other hand, is assumed to follow a linear sawtooth pattern (congestion avoidance phase), with uniformly distributed window size. It is important to note that, in practice, the size of a flow is not a good indicator of whether the flow will be in slow-start or congestion avoidance. In uncongested networks, even a very long flow can spend its lifetime in slow-start, while in congested networks, even short flows can be in congestion avoidance.

Furthermore, not all "long" TCP flows should be included in the parameter N of the Stanford model. Specifically, the following types of flows do not meet the assumptions of the Stanford model because they are not bottlenecked at the target link:

1. Long flows that are bottlenecked at other links. These flows experience congestive losses at links other than the target link.

2. Long flows that are limited by the socket buffer size or by the receiver's advertised window.

Finally, the number of long TCP flows varies with time. If we provision the buffer size based on the time-average of N, the link would not be saturated when the number of flows is less than the time-average. On the other hand, provisioning based on the minimum observed number of flows may result in a very low N value, which is not that different than the BDP rule-of-thumb.

3.4 What traffic model to consider?

The required buffer size to meet a given performance objective depends on the underlying traffic model. The buffer sizing problem was the subject of extensive research earlier, especially in the context of ATM multiplexers and QoS provisioning (see [2] for some references). Those previous works, however, considered the traffic as *exogenous* to network conditions, meaning that the offered load does not react to network congestion. TCP traffic, with its congestion control algorithms, does not behave as exogenous traffic.

The Stanford model considered primarily persistent TCP transfers, i.e., transfers that last forever. This model is commonly used in both the AQM literature and in other buffer sizing works [2]. It has been previously shown (for instance, see [5]) that the throughput R of a persistent TCP transfer with RTT T that experiences a loss rate p can be approximated by

$$R = \frac{0.87}{T\sqrt{p}} \qquad (2)$$

The RTT T in equation (2) includes both propagation delay and any queuing delays in the path.

It is important to note however that the model of persistent TCP transfers ignores some key issues about the buffer sizing problem. First, with persistent transfers the number of long TCP flows N is time invariant. Second, persistent transfers (not limited by their advertised window or by other links) can always produce enough offered load to drive a link to saturation, without causing overload or congestion collapse. Third, persistent transfers are mostly in the congestion avoidance phase (i.e., linear, or almost linear, sawtooth behavior).

In practice, TCP transfers have a finite duration and size, even if a small fraction of them are much larger than the rest. Consequently, the number of ongoing transfers varies with time depending on the flow arrival process and the network load. The offered load with finite TCP transfers can exceed the capacity causing congestion collapse, as explained in the next paragraph. Also, the throughput of finite transfers can have significant variability, depending on whether a transfer sees losses during its lifetime.

In this letter, we distinguish between two fundamentally different flow generation processes: *open-loop* and *closed-loop*. In the open-loop model, flows arrive randomly at the target link with an average rate λ and an average transfer size S. The offered load of an open-loop TCP traffic model is λS. In the closed-loop model, the traffic at the target link is generated by a number U of users. Each user can only start a new TCP transfer after the completion of his/her last transfer. Furthermore, there may be an idle time period (OFF state) between the completion and the start of two successive transfers. A key difference between the open-loop and closed-loop traffic generation models is that, even though they both use TCP, the former produces traffic that does not depend on the network state. Consequently, the open-loop model can produce overload and congestion collapse if the offered load is larger than the target link capacity ($\lambda S > C$). With the closed-loop model, on the other hand, the number of active flows in the target link never exceeds U, and the offered load remains below the capacity.

As will be shown in Section 4, the buffer size that is required to meet a certain network or application performance objective significantly depends on whether the underlying TCP traffic model is persistent, open-loop, or closed-loop transfers.

4. SIMULATION RESULTS

4.1 Simulation setup

We use NS2 simulations to examine the effect of the target link buffer size on the loss rate and application-level performance of TCP flows. We simulate TCP NewReno flows with the SACK option enabled. The simulation topology consists of 18 source nodes arranged in the form of a balanced binary tree. The flows are routed from the source nodes to a target link with 50Mbps capacity. All flows terminate at a single sink node. The tree-like topology results in heterogeneous RTTs ranging from 30ms to 530ms, with an *effective RTT*[1] of $T_{eff} = 60$ms. The capacities and buffers of the access links are sufficiently large to avoid losses or significant queueing delays in those links (non-saturable links). The data packet size is 1500 bytes for all flows. The BDP

---

[1]The effective RTT of a group of TCP connections is defined as the harmonic mean of the individual RTTs [2].

is CT_{eff}=250 packets. In the simulations with finite-sized transfers, we use the heuristics described in [9] to identify flows that spend most of their lifetime in slow start ("short" flows). The remaining flows are considered "long" flows. The time-average of the number of long flows is used as the value of N in (1).

4.2 Traffic workload

We simulate the following three types of TCP traffic. Even though the following models are quite different in terms of the traffic the produce, they are all parameterized so that the time-average of the number of long flows is $N \approx 200$. Note that the number of long flows is comparable to the BDP of the target link (in packets). In this case, the average window size per TCP flow is quite low, in the order of 1-3 packets depending on the buffer size B. We expect that this is a typical case for congested access links, when these access links connect networks with many users to the Internet. Specifically, the simulation parameters that we use in this paper are similar to the actual characteristics (capacity, RTT distribution, number of flows) of the access link of the University of Crete, in Greece.

Persistent TCP flows with a small fraction of mice: 200 persistent TCP connections start from randomly chosen source nodes, and these connections are active for the entire duration of the simulation. "Mice" consist of TCP flows with a uniformly distributed size between 3 and 25 packets, originating from randomly chosen source nodes. The flow inter-arrival times for mice follow an exponential distribution, and their average is such that the total mice load is 5% of the link capacity.

Closed-loop TCP traffic: To simulate closed-loop traffic, we use a number U of source agents, each of which starts a file transfer, waits for the transfer to complete, and then stays idle for a *think duration* before starting the next transfer. The think time of each agent is exponentially distributed with a mean of 5 seconds. The file sizes are drawn from a distribution that is roughly heavy-tailed. Specifically, the distribution is the sum of three exponential distributions: one modeling small transfers (average size: 15 packets), one for medium transfers (average size: 50 packets), and one for large transfers (average size: 200 packets). The arrival rates of the three flow classes are adjusted so that the largest 30% of the flows account for 70% of the total bytes. The number of source agents is adjusted so that the time-average of the number of long flows is 200.

Open-loop TCP traffic: To simulate open-loop traffic, we simulate flows that arrive randomly, independent of the network's current state. Specifically, the flow inter-arrival times are independent and exponentially distributed (i.e., a Poisson flow generation process) from randomly chosen source nodes. The file size distribution is the same as that for closed-loop traffic. The mean flow inter-arrival is adjusted so that the average offered load is 95% of the target link capacity. This ensures that the system is stable, while producing a time-average of about 200 long flows.

4.3 Results

In this section, we present simulation results for the effects of the target link buffer size B on both the aggregate loss rate and the performance of individual TCP flows. Simulations were run for each of the previous three traffic models (persistent flows and mice, closed-loop traffic, and open-loop
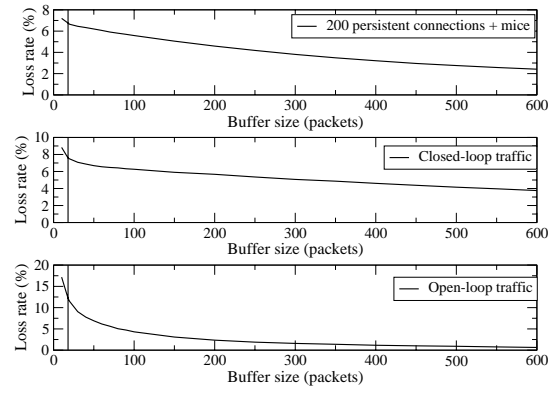

Figure 1: Loss rate as a function of buffer size for different traffic types.

traffic). Based on (1), the buffer size that results from the Stanford model is B_s=18 packets, referred to as the "small buffer". We compare that with a buffer size that is twice the BDP, i.e., B_l=500 packets, referred to as "large buffer". In the traffic models with non-persistent TCP connections, we also distinguish between the performance of "large flows" (more than 100KB) and "short flows" (less than 100KB)[2].

Utilization and loss rate. In all simulations of this section, with both B_s and B_l, *the utilization of the target link has remained at almost 100%.* This observation validates the main point of the Stanford model, which is that even small buffers are sufficient to keep the target link saturated. However, we need to also examine the effect of small buffers on the loss rate at the target link.

Figure 1 shows the loss rate at the target link as a function of the buffer size B. The vertical line in each plot corresponds to B_s. The first observation is that *the loss rate that results from the Stanford model is about 12% for open-loop TCP traffic and about 7-8% for persistent flows and closed-loop TCP traffic.* We believe that such high loss rates would be unacceptable in practice, and that they would cause poor performance for certain types of applications (including reliable and interactive transactions, or highly compressed audio or video streaming or conferencing). With B_l, the loss rate drops to less than 1% for open-loop traffic, about 2% for persistent flows, and 4% for closed-loop traffic.

In the case of persistent flows, we can derive mathematically the loss rate p that results at the target link as a function of the buffer size B, under some simplifying assumptions. Specifically, we can use Equation (2) to show that if N homogeneous connections, with round-trip propagation delay T_p, fully saturate a link of capacity C, then the loss rate at that link is given by

$$p = \frac{(0.87N)^2}{(CT_p + B)^2} \qquad (3)$$

The previous formula assumes that the queuing delay experienced by each flow is equal to the maximum queuing delay B/C, and so the RTT of each flow is $T_p + B/C$. Notice that *the loss rate increases with the square of the number of flows*

[2]The exact value of the threshold that separates large from small flows is not so important; we observed similar trends with a 400KB threshold.

that are bottlenecked at the target link. Also, for fixed N, the loss rate decreases rather slowly as the buffer size increases (as a power law rather than exponentially fast).

Figure 1 shows that the rate with which the loss rate decreases, as B increases, strongly depends on the traffic model. This is because the three traffic models have substantially different congestion responsiveness, variability in the number of active flows, and intra-flow packet burstiness. Furthermore, in the open-loop model, the main role of the buffer is to absorb transient traffic bursts; the buffer does not affect the incoming traffic load in any way. Hence, with a sufficiently large buffer the packet loss rate can be reduced to practically zero. With the closed-loop model, on the other hand, a larger buffer, and the resulting lower loss rate, means that TCP flows complete faster and new flows start sooner. In other words, the incoming traffic load increases as the loss rate at the target link decreases. This explains the much slower reduction of the loss rate in the closed-loop model compared to the open-loop model in Figure 1.

Individual flow performance. Utilization, loss rate and queuing delay are network-layer performance metrics for aggregated traffic, and they do not represent the performance of individual flows. In the following, Figures 2 to 7 show simulation results for the distribution of per-flow throughput and per-flow loss rate seen by "large" (>100KB) and "small" (<100KB) flows, for each of the three traffic models.
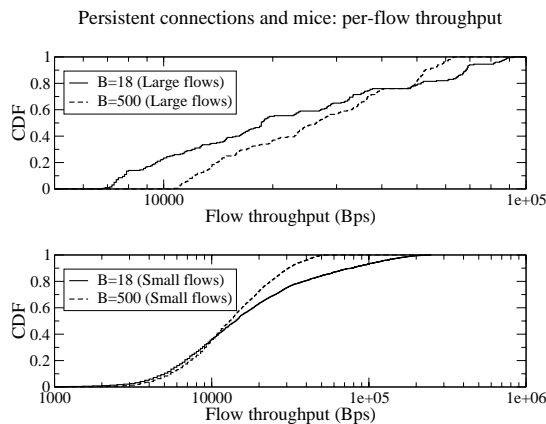

Figure 3: **Per-flow loss rate for persistent connections and mice.**

the loss rate is much larger with B_s and their throughput suffers.

Another observation is that *the variability of the per-flow throughput is considerably lower with B_l than with B_s.* The main reason for the variability reduction is that B_l causes a larger queueing delay than B_s, reducing the relative difference between the RTTs of different flows. The reduced loss rate is another reason for the reduced throughput variability with B_l.

Figure 2: **Per-flow throughput for persistent connections and mice.**

Figure 4: **Per-flow throughput for closed-loop traffic.**

Persistent connections and mice. Figure 2 shows the per-flow throughput in the case of the "persistent connections and mice" traffic model. Note the logarithmic scale of the x-axis (in bytes per second). Also, Figure 3 shows the distribution of the per-flow loss rate, separately for persistent flows and mice. The first observation is that *around 80% of the persistent flows get a higher throughput with B_l than with B_s.* On the other hand, the lower part of Figure 2 shows that *about 50% of the small flows (mice) do better with the small buffer.* The reason is that about 40%-70% of the mice do not see any losses, depending on the value of B, and for those flows B_s leads to lower RTT and thus to higher throughput. For the mice that did observe packet drops,
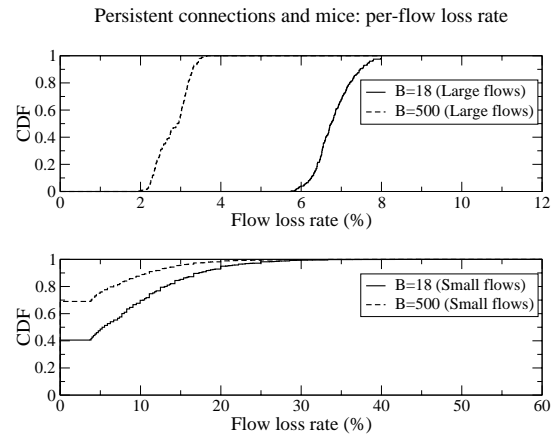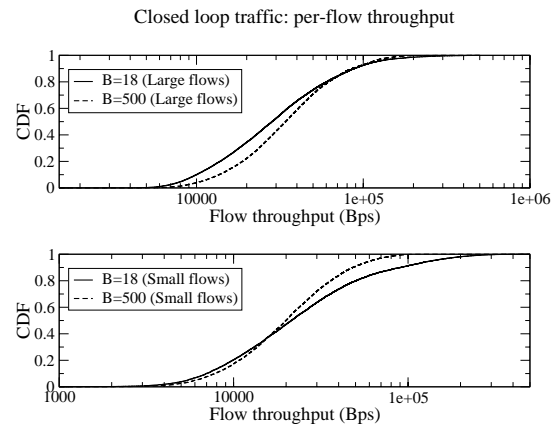
Closed-loop traffic model. The results with the closed-loop traffic model follow similar trends (see Figure 4). *Most large flows see a higher throughput with B_l and most small flows perform better with B_s.* Also, the larger buffer decreases the throughput variability. The difference between the two buffer sizes however is not as pronounced as in the case of Figure 2, because the loss rate with the closed-loop traffic model does not differ that much between the two buffer sizes (see Figure 1).

Open-loop traffic model. In the case of open-loop traffic, the picture is quite different. Recall from Figure 1 that the aggregate loss rate is an order of magnitude lower with B_l
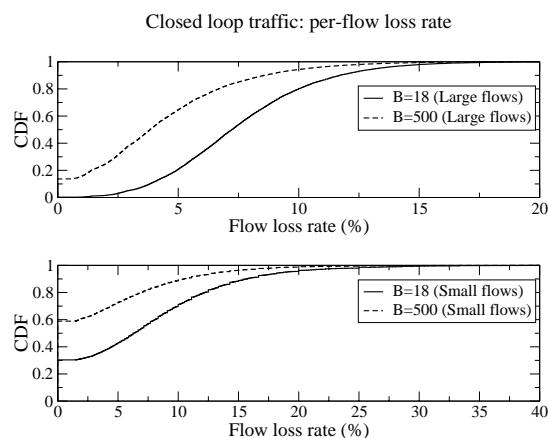
Closed loop traffic: per-flow loss rate

Figure 5: Per-flow loss rate for closed-loop traffic.

Open loop traffic: per-flow throughput

Figure 6: Per-flow throughput for open-loop traffic.
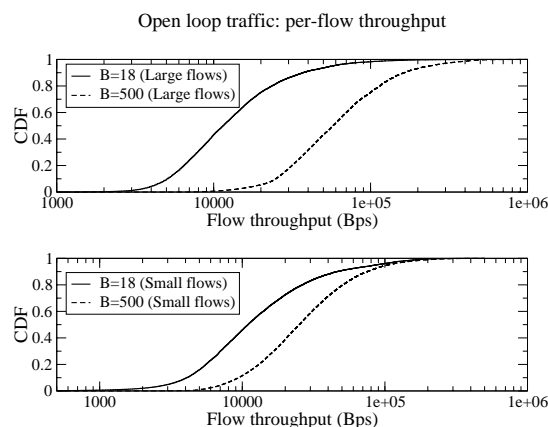
Open loop traffic: per-flow loss rate

Figure 7: Per-flow loss rate for open-loop traffic.

than with B_s. Figure 6 shows that both large and small flows achieve much higher throughput with B_l. The median throughput for large flows is about 10,000 bytes/sec for B_s and 15,000 bytes/sec for B_l. The distribution of per-flow loss rates in Figure 7 shows that about 70% of the large flows do not see any losses with B_l, while most such flows see losses with B_s. Small flows show a similar trend. The reason that the open-loop traffic model behaves in this way is that it generates larger traffic burstiness than the two other models, mostly because of the greater variability in the number of active flows. In the closed-loop model the number of active flows is bounded by the number of users, while with persistent connections the number of active flows remains constant.

5. CONCLUSIONS

In this letter, we have raised some concerns over the use of small buffers (less than the BDP) in Internet routers. Small buffers have the advantage that they reduce queuing delays, while still maintaining full utilization at the target link. However, this comes at the cost of a high loss rate, which can harm the performance of certain applications. Furthermore, small buffers result in lower throughput for most large TCP transfers, and for practically all transfers in the case of the open-loop traffic model. Also, the use of
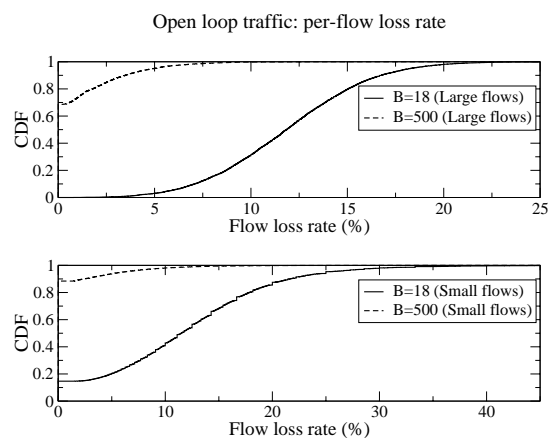
small buffers would increase the variability in the throughput of TCP flows, making application-layer performance less predictable. Finally, we note that the tradeoff between loss rate and queuing delay, in terms of application-layer performance, is an important open issue in the buffer sizing problem. Investigating this tradeoff in depth is an important direction for future research.

Acknowledgements

6. REFERENCES

[1] G. Appenzeller, I. Keslassy, and N. McKeown. Sizing Router Buffers. In *Proceedings of ACM SIGCOMM*, October 2004.

[2] A. Dhamdhere, H. Jiang, and C. Dovrolis. Buffer Sizing for Congested Internet Links. In *Proceedings of IEEE INFOCOM*, 2005.

[3] M. Enachescu, Y. Ganjali, A. Goel, N. McKeown, and T. Roughgarden. Part III: Routers with Very Small Buffers. *ACM Computer Communication Review*, 2005.

[4] R. Morris. Scalable TCP Congestion Control. In *Proceedings of IEEE INFOCOM*, April 2000.

[5] J. Padhye, V.Firoiu, D.Towsley, and J. Kurose. Modeling TCP Throughput: A Simple Model and its Empirical Validation. In *Proceedings of ACM SIGCOMM*, 1998.

[6] G. Raina, D. Towsley, and D. Wischik. Part II: Control Theory for Buffer Sizing. *ACM Computer Communication Review*, 2005.

[7] C. Villamizar and C.Song. High Performance TCP in ANSNET. *ACM Computer Communication Review*, October 1994.

[8] D. Wischik and N. McKeown. Part I: Buffer Sizes for Core Routers. *ACM Computer Communication Review*, 2005.

[9] Y. Zhang, L. Breslau, V. Paxson, and S. Shenker. On the Characteristics and Origins of Internet Flow Rates. In *Proceedings of ACM SIGCOMM*, August 2002.