

The Model of Persistent TCP Flows Considered Misleading

Ravi S. Prasad, Amogh Dhamdhere and Constantine Dovrolis
College of Computing
Georgia Institute of Technology
{ravi,amogh,dovrolis}@cc.gatech.edu

ABSTRACT

The model of persistent TCP flows has been used extensively in both analytical and simulation studies. A main implication of this model is that each flow is regulated by TCP congestion control, while the set of competing flows remains the same. In this paper, we show that the model of persistent connections is not only unrealistic but it is also misleading when it comes to questions about traffic variability, congestion responsiveness, or router buffer sizing. So our first recommendation is that networking studies should consider exclusively non-persistent TCP models. Such models are typically characterized in terms of the flow size distribution and the flow arrival process. We focus on another important characteristic of non-persistent models, namely whether the flow arrival process depends on the network state. In the so-called “open-loop TCP model”, new TCP flows/sessions arrive randomly based on an exogenous process (e.g., Poisson), while in the “closed-loop TCP model” the generation of a new TCP flow/session from a user depends on the completion of the last flow from that user. To show the significant differences between persistent and non-persistent models quantitatively we examine two specific problems: the congestion responsiveness of aggregated traffic and the sizing of router buffers. In both cases, we find out that the two non-persistent models lead to significantly different insight and results. Which of the two non-persistent models is more realistic is an open issue that needs to be resolved by future measurement studies.

1. INTRODUCTION

Networking research has primarily focused on *persistent TCP connections*, i.e., transfers with infinite size and duration. Persistent transfers are simpler to analyze mathematically, as they do not exhibit any randomness in the flow size distribution or in the arrival process. Further, persistent transfers create a feedback loop in which the offered load in the network is regulated by TCP congestion control. Based on the model of persistent flows, previous work has produced concrete results for the average TCP throughput [23, 29], the stability of TCP traffic in the presence of appropriate AQM controllers [22, 25], the amount of required buffering in routers [4, 11, 21], the statistical characteristics of Internet traffic [8], and others. In this paper, we argue that the persistent flows model is not only unrealistic (let us not forget that “all models are unrealistic but some are useful”), but it is also often misleading. In particular, persistent flows hide certain key properties of Internet traffic pertaining to its variability and congestion responsiveness,

and so they can lead to incorrect insight and results.¹

In practice, TCP transfers have a finite duration and they typically follow a heavy-tailed size distribution. So, they are often classified as either “elephants” (very large transfers) or “mice” (short, Web-like, transfers). The former carry most of the traffic in the Internet, and for this reason they have been the subject of most previous work in congestion control, *assuming* that they can be modeled as persistent transfers [9, 17, 24]. The latter, on the other hand, carry a small fraction of the overall traffic, and even though they do not react to congestion the same way “elephants” do [6, 12, 18, 19], they are often ignored. In Section 2, we argue that modeling most of the traffic at a network link as persistent TCP flows is misleading as it does not capture the variability of that traffic in terms of flow size distribution, flow arrival process, and number of active flows.

When we view all TCP flows as *non-persistent* (i.e., with a finite size and duration), then the key issue is *the random process that determines the arrival of TCP flows, rather than packets, into the network*. Most of the Internet traffic is generated from users or applications that pull (or sometimes push) data from servers or other peers. Each such *session* can generate several TCP connections, and it represents the basic unit of offered load at the session layer of the OSI stack. In Section 3, we identify two fundamentally different session generation models. In the *closed-loop TCP model* (also known as the “interactive model”) we have a finite population of users, and each user can generate a new session only after the completion of her previous session. In the *open-loop TCP model* sessions arrive independently of the completion of previous sessions. This model is more appropriate in cases where the population of users is very large and users either do not return to the network, or they do so long after the completion of their last session.

The difference between these two session generation models is major both in terms of the congestion responsiveness and the variability of the resulting aggregate traffic. In the closed-loop model, network congestion delays the completion of ongoing sessions and thus the generation of the next session from each active user. Consequently, the emergence of congestion regulates the arrival rate of new sessions in the network, resulting in a *congestion responsive traffic aggregate*. On the other hand, with an open-loop traffic model,

¹The authors admit that they have also used the persistent flows model in much of their previous work. Our objective is not to criticize, but to argue that we should move beyond the model of persistent flows. A similar transition took place 10-15 years ago when the community abandoned the model of Poisson packet arrivals.

sessions arrive independently of each other, and independent of congestion. If the session arrival rate is too high, for a given network capacity and session size, the network can experience persistent overload *even if each individual session uses TCP and is congestion responsive*. The persistent overload will result in a very low goodput for each user, despite the fact that the network bottleneck is fully utilized. Further, the persistent overload in the open-load model can lead to a significant fraction of aborted sessions due to user impatience.

Section 3 examines the variability that results from the two non-persistent models in terms of number of active flows and variance-time plots for the aggregate offered load. The key observation is that the open-loop model leads to significantly higher variability compared to both the closed-loop and the persistent model. The main reason is that, with the open-loop model, the flow arrival process is random and so it is not regulated by the presence of congestion in the network.

To compare the two non-persistent models, and to also show their significant differences with the persistent model, we focus on two specific issues: the congestion responsiveness of Internet traffic (Section 4) and the sizing of router buffers (Section 5). In the former, we show that the closed-loop model creates traffic that is always responsive to congestion (as is the case with persistent transfers) while the open-loop model generates traffic that is completely unresponsive to congestion even though each flow uses TCP congestion control! We also show that the two non-persistent models create significantly higher traffic variability than persistent flows, and that recent proposals for reducing the size of router buffers to just a few tens of packets can lead to major packet losses even in moderate load conditions.

Finally, in Section 6 we discuss some implications of our conclusions in several areas of networking research and practice. These areas include the usefulness of AQM, admission control and TCP-friendly congestion control, the need for new mathematical and simulation models, and the integration of congestion control with the application or session layers.

2. IS THE MODEL OF PERSISTENT TCP FLOWS REALISTIC?

In practice, all TCP transfers have a finite duration and size. A common argument, however, is that typically most traffic is carried by a few large flows (“elephants”), and so those flows can be modeled as persistent. The smaller flows, known as “mice”, do not contribute a significant amount of traffic and they can be ignored, or they can be viewed as a source of random noise in simulation studies. The previous argument is oversimplifying the characteristics of real TCP transfers in two ways. First, in practice TCP flows follow a continuous and heavy-tailed distribution, rather than a bimodal distribution in which they are either very short or very long. In other words, the previous argument ignores the presence of flows of significant, but not extreme, size. Second, flows with very large size (relative to other flows in the traffic) do not need to also have very long duration. Some large flows get very high throughput, and so their duration can be comparable to that of short flows. Such flows cannot be modeled as persistent, especially when the timescale of interest (for example, the duration of the simulation study)

is longer than their duration.

To illustrate the previous points, we next analyze a packet trace that was collected at the border router of Georgia Tech in January 2005. The trace duration is two hours and the monitored link is the inbound traffic in a Gigabit Ethernet segment that connects the campus network to the SoX GigaPoP. The objective of this analysis is to show real flow size, flow duration, and flow interarrival distributions, and to examine the assumptions behind the persistent flows model in the context of such measurements. Note that similar traffic analysis has been conducted several times in past work (for instance, see [7]), analyzing traces from many links and in diverse load conditions.

First, Figure 1 shows the empirical C-CDF of the flow size distribution, i.e., an estimate of the probability that a TCP flow is larger than X bytes. The almost linear shape of the C-CDF in a log-log plot shows that the flow size distribution can be modeled as Pareto (shape parameter ≈ 1.3), at least in the range between 1KB and 1GB. Notice that the distribution does not show any sign of bimodal behavior that would justify the distinction between “mice” and “elephants”. Flows of all sizes are possible and they are generated from the same distribution.

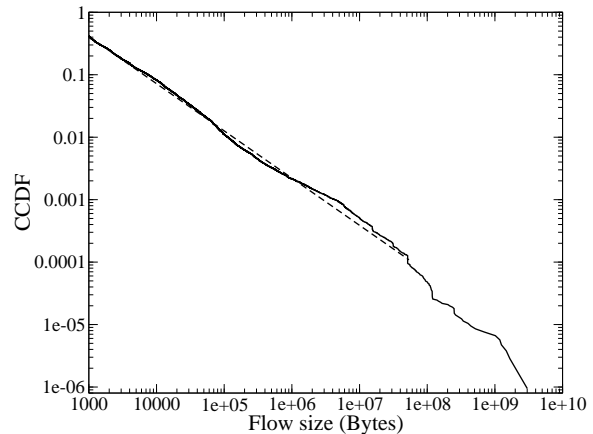


Figure 1: TCP flow size distribution in the inbound Georgia Tech traffic trace.

Figure 2 shows the number of active TCP flows as a function of time, during the central part of the trace (1 hour and 20 minutes). Since most flows are small in size, we can impose a minimum flow size threshold H and only count flows with size larger than H . Each curve in Figure 2 refers to a different value of H , starting from 7.5KB up to 1.5MB. The key observation is that the number of active flows shows considerable variation with time, especially when we include the flows with “medium” sizes, in the order of 10s to 100s of kilobytes. The persistent model, on the other hand, assumes that the number of active flows is constant. Note that prior results based on the persistent model often depend on the number of active flows (for instance, the buffer sizing formula of [11]). The application of such results in a link where the number of active flows varies considerably with time is problematic.

One can argue, looking at the last graph, that the number of active flows is almost constant when we only consider flows that are larger than 1.5MB, and those flows account for about 87% of the traffic. Consequently, why don’t we

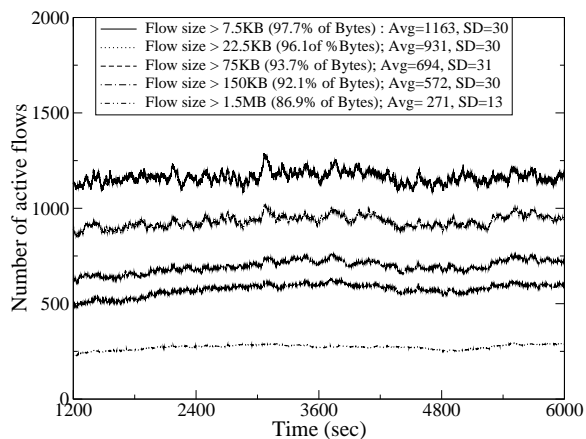


Figure 2: Number of active TCP flows as a function of time for several flow size thresholds H . The number in parenthesis is the fraction of traffic included in these flows. The average and standard deviation of the number of flows is also shown in the legend.

model those flows as persistent? The answer is that even though the number of flows is almost constant when H is sufficiently large, the set of flows that are larger than H bytes varies significantly with time, as previous flows complete and new flows arrive. The persistent model, on the other hand, assumes that the set of active flows remains the same. To illustrate this point, we examine the fraction of bytes f that is generated by flows that were active throughout a given time interval T . Note that with the persistent model, all flows are active for the entire duration T and so $f=100\%$. To measure f , we set T to 7.5, 15, 30, 60 or 120 minutes. Figure 3 shows the mean, minimum, and maximum value of f as a function of T . The key observation is that even for time periods that only last 5-10mins, the fraction of traffic from flows that persisted during that interval is only 40-70%. So, the assumption that the same set of flows carries more than 90% of the traffic does not hold.

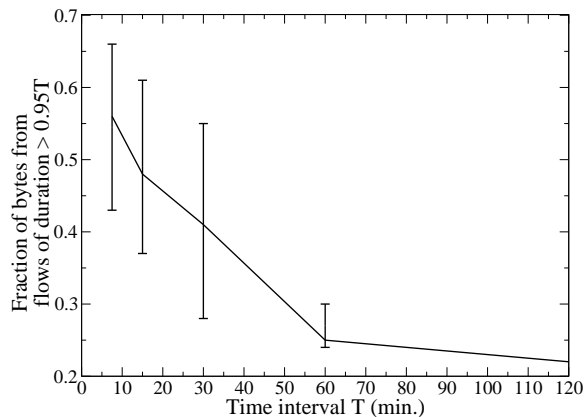


Figure 3: Fraction of bytes f generated by flows that were active during a time interval of length T . The error bars depict the minimum and maximum values of f for a given value of T .

Finally, Figure 4 shows the CDF of TCP flow interarrivals for the same packet trace. The almost linear shape of the CDF in the log-linear plot implies that the flow interarrivals are almost exponentially distributed. Note however that in short timescales, focusing on interarrivals of up to 10msec, the probability for interarrivals of less than X is larger than that predicted by the exponential model. This is expected, given that several applications (e.g., Web browsers) generate bursts of TCP flows with each user action. Further analysis of the temporal correlations in the flow arrival pattern shows that flows arrive independently when we focus on timescales of more than 10msec, but they do show strong correlations in shorter timescales. Consequently, at least for this packet trace, it is reasonable to assume that sessions arrive as a Poisson process and that each session consists of one or more TCP flows that arrive within a few milliseconds.

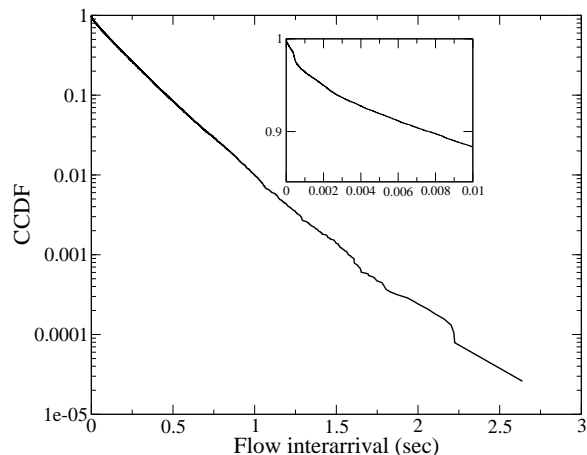


Figure 4: Complimentary CDF of TCP flow interarrivals.

3. NON-PERSISTENT FLOW MODELS AND THEIR VARIABILITY

The previous section showed that the model of persistent TCP flows does not capture the following major characteristics of real Internet traffic: heavy-tailed flow size distribution, time-varying number of active flows, and randomness in the flow or session arrival process. Consequently, it is important to consider non-persistent models, and to examine the differences that result from such models compared to persistent flows. In this section, we identify two basic non-persistent models for the session generation process: open-loop TCP flow arrivals and closed-loop TCP flow arrivals. Both models are simple and well-known in the performance evaluation literature. They have received significantly less attention in the networking literature, however, mostly because of the dominance of the persistent model.

Note that the terms “open-loop” and “closed-loop” have been previously used to distinguish between non-TCP traffic (open-loop because packets arrive randomly based on an exogenous process) and TCP traffic (closed-loop because the packets of a flow are regulated by TCP congestion control) [14]. In this paper, we use these two terms to distinguish between TCP flow arrivals at the session layer, as explained next. Also, even though we present the following models in

the context of TCP flow arrivals, it is simple to extend them for TCP session arrivals (i.e., for groups of flows that arrive at about the same time).

3.1 Open-loop model

In the open-loop model, users or applications generate finite-size flows independent of previous flows they may have generated. To motivate this model, consider the access link of a Web server. In the outbound direction, the server sends files to a very large population of users located anywhere in the Internet. Assume that a user does not return to this server, at least for a long time, after downloading a flow. Consequently, the server's connections are always with new users. If the link becomes congested, the arrival rate of new flows will not be affected, as Internet users are typically unaware of the network state in a given path.

Considering a link \mathcal{L} with capacity C , the average offered load in the open-loop model is given by λS , where λ is the average flow arrival rate and S is the average flow size. The normalized offered load is defined as

$$\rho_o = \lambda S / C. \quad (1)$$

If $\rho_o < 1$, \mathcal{L} is stable and ρ_o is the average utilization. Otherwise, if $\rho_o > 1$, \mathcal{L} becomes unstable (if flows are never aborted) [16]. Since both the flow arrival rate and the average flow size is independent of network state, the offered load in this model remains constant in the presence of network congestion.

Previous work with the open-loop model includes the study of Ben Fredj et al. in [10], in which they noted that the only reduction in the offered load upon a congestion event is due to aborted transfers. Such transfers, however, result in wasted throughput and user dissatisfaction. For this reason, the authors proposed admission control as the only efficient way to prevent persistent overload. Veciana et al. [3] also considered the open-loop traffic model and concluded that Internet traffic may be unstable under certain conditions.

Most of the previous work with the open-loop model assumes that TCP flows share the capacity of their bottleneck as in an ideal Processing Sharing (PS) server [27]. This is a reasonable approximation as long as all competing TCP transfers have the same RTT, and they are not limited by end-host socket buffers or by access links. Kherani and Kumar [15] showed that the PS model is not always accurate, mostly because TCP transfers do not manage to keep the link fully utilized under certain conditions. Assuming Processor Sharing and Poisson flow arrivals, it is well-known (see [16]) that the average number of flows in the open-loop model is given by

$$\bar{N} = \frac{\rho}{1 - \rho} \quad (2)$$

3.2 Closed-loop model

To illustrate the closed-loop model, consider the access link of a small enterprise with, say N , users. In the inbound direction, most of the Web traffic at the link is downloads that are generated by the activity of these N users. Each user in the "Active" state downloads a Web page, then spends some time in the "Idle" (or "Thinking") state viewing the page, and then either downloads another Web page or leaves the system for a longer time period ("Inactive" state). This link would never carry more than N downloads at a time. Furthermore, if the link becomes congested, then

the download latencies of all active flows will increase, reducing the rate with which new downloads are generated.

In the closed-loop model, we have a fixed number of users N . Each user goes through cycles of activity, with flows of average size S , followed by idle periods of average length T_i . The average flow arrival rate in the closed-loop model is given by

$$\lambda_c = \frac{N}{T_i + T_i} \quad (3)$$

where T_i is the average flow transfer time. The latter is dependent on the load at \mathcal{L} . The maximum normalized offered load for the closed-loop model is given by

$$\rho_c = N S / C T_i. \quad (4)$$

When $\rho_c \ll 1$, users spend most of the time thinking (i.e., $T_i \ll T_i$), and the system behaves as an open-loop model with flow arrival rate $\lambda_c = N / T_i$. However, when ρ_c approaches or exceeds 1, the number of active flows in the server increases, reducing the average per-flow throughput and increasing T_i . The increase in T_i reduces the flow arrival rate, as given by (4), keeping the offered load close to the capacity, i.e., $\lambda_c S \approx C$. This means that the closed-loop traffic model will always have bounded offered load, even when $\rho_c > 1$.

The average number of active flows in the closed-loop model is given by (see [1]):

$$\bar{N} = \frac{\rho_c}{1 - \rho_c} \quad \text{for } \rho_c \ll 1 \quad (5)$$

$$= N (1 - \rho_c^{-1}) = N - \frac{C T_i}{S} \quad \text{for } \rho_c > 1 \quad (6)$$

Note that the expected number of active flows for $\rho_c \ll 1$ is the same as in the open-loop model. On the other hand, when $\rho_c > 1$, \bar{N} increases slowly with ρ_c and it remains bounded by N .

The closed-loop model converges to the open-loop model when both the number of users N and the average think time T_i increase to infinity at the same rate. Similarly, the closed-loop model converges to the persistent model when both the average size S and the think time T_i increase to infinity at the same rate.

Previous work in the area of congestion control with the closed-loop model is quite limited. Heyman et al. [13] used a closed-loop traffic model to analyze the performance of Web-like traffic over TCP. They showed that the per-flow goodput and the fraction of time the system has a given number of active flows are insensitive to the distribution of flow sizes and "think times", and they only depend on the mean of these distributions. Berger and Kogan [1], as well as Bonald et al. [2], used a similar closed-loop model to design bandwidth provisioning rules for meeting certain throughput-related QoS objectives. In a recent paper, Schroeder et al. [28] compare open-loop and closed-loop job arrivals at a queuing system and highlight their differences in terms of mean job completion time and response to different scheduling policies.

3.3 Traffic variability with non-persistent models

Now that we have described the two non-persistent models, we examine their traffic variability in terms of the number of active flows and the variance of the offered load in different timescales.

The following results are based on NS simulations of a 50Mbps bottleneck link \mathcal{L} (see Figure 5). \mathcal{L} represents the access link of an enterprise or campus network, which receives traffic from a number of well-connected servers in the Internet. Specifically, \mathcal{L} carries non-persistent TCP flows with heterogeneous RTTs, between 80-120msec. \mathcal{L} is con-

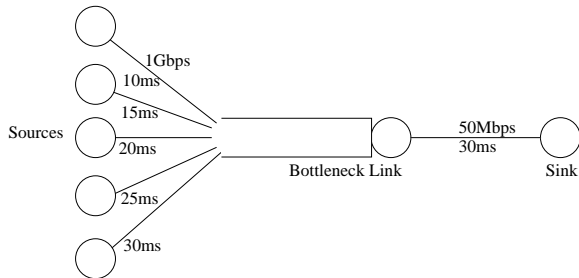


Figure 5: Simulation topology.

nected to five source nodes with 1Gbps access links, and it carries TCP traffic in both directions (even though it is congested only in the forward direction). The flow sizes follow a Pareto distribution with shape 1.9 (based on our measurements of various packet traces), while the average flow size is 17 MTU (1500B) packets. In the open-loop model, flows arrive based on a Poisson process. The flow arrival rate λ in the open-loop model and the number of users N in the closed-loop model are adjusted to achieve the desired ρ_o and ρ_c , respectively. The think time in the closed-loop model is uniformly distributed between 1-3 seconds.

First, we examine the variability in the number of active flows generated with the open-loop model as compared to the closed-loop model. In our simulations, each flow reports its start and finish times giving us the exact time series of the number of active flows. We sample this time series every 5msec. The empirical CDF of the number of active flows computed in this manner is shown in Figure 6. The offered load for both the open-loop and closed-loop models is set to $\rho_o = \rho_c = 95\%$.

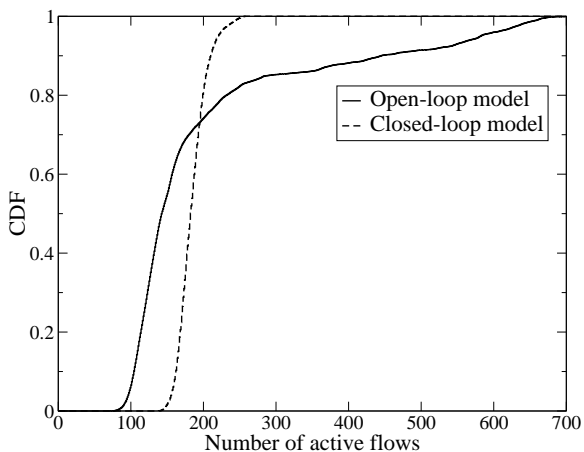


Figure 6: The CDF of the number of active flows with the two non-persistent models in a simulation with $\rho_o = \rho_c = 95\%$.

The main observation here is that *the variability in the*

number of active flows is significantly higher with the open-loop model than with the closed-loop model. To quantify this point, the mean, standard deviation, and coefficient of variation with the open-loop model are, $\mu=202$, $\sigma=143$ and $\text{CoV}=0.71$, respectively. The corresponding statistics with the closed-loop model are $\mu=186$, $\sigma=20$ and $\text{CoV}=0.11$, i.e., at this offered load, the variability in the number of active flows with the open-loop model is about seven times higher. There are two reasons for this difference between the two models. First, the number of active flows N_a in the closed-loop model is bounded by the number of users N , while it is unbounded in the open-loop model. Second, as previously mentioned, the flow arrival rate in the closed-loop model decreases when there is congestion, while it stays constant in the open-loop model.

Next, we focus on the variability of the aggregate traffic that the two non-persistent models, as well as the persistent model, produce in different timescales. To do so we rely on variance-time plots, as shown in Figure 7. The graph shows the variance of the offered load in different timescales, starting from 1msec up to 10sec, for the three models. The average offered load is 95% in all cases. Notice that the vari-

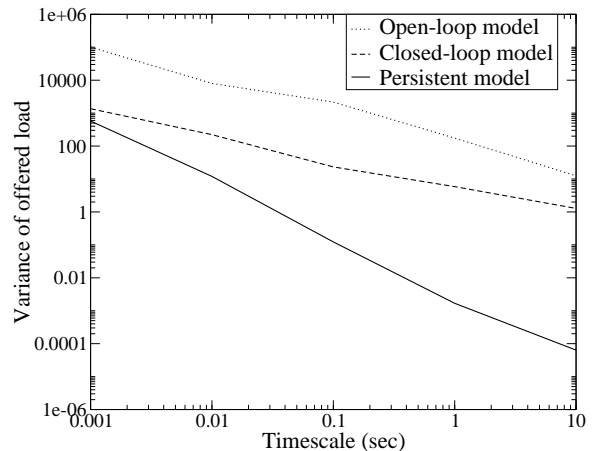


Figure 7: Variance-time plots for the three traffic models, with an average utilization of 95%.

ance of the offered load is always highest with the open-loop model, followed by the closed-loop model. The reason the variability of the former is higher is related to the variability in the flow arrival process, and it was discussed in the previous paragraph.

The main observation in Figure 7, however, is that the variability of the persistent model is significantly lower than that of the non-persistent models. The reason is that the only source of variation in the former is the *intraflow traffic variability*, i.e., packet-level burstiness due to TCP self-clocking, window-level variations due to TCP slow-start, congestion control (AIMD), and retransmission timeouts. The two non-persistent models, on the other hand, have an additional major source of variability: *the randomness in the number of active flows*. This type of burstiness is completely ignored when we assume persistent connections, leading to significantly smoother traffic. Also note that the variance gap between the persistent and non-persistent models is greater as the timescale increases. The reason is that the intraflow variability, which all three models exhibit, be-

comes less significant compared to the variability in the number of active flows as the averaging timescale of the x-axis increases.

4. CASE STUDY I: CONGESTION RESPONSIVENESS

The *TCP feedback loop* regulates the offered load (send-window) of a connection, based on the presence of congestion in the network (see Figure 8). The previous view, however, ignores the fact that TCP connections are the result of user and application actions. For example, the TCP connections generated from downloading a Web page, which constitute a “Web session”, are the result of a user entering a URL at a web browser or clicking on a link. That user can keep generating new sessions, independent of whether the network is congested or not. In other words, even though the transport layer provides congestion responsiveness through TCP, the session layer can be completely unresponsive if it keeps generating new sessions even when the network is congested. The lack of a *session layer feedback loop* can lead to a large number of active sessions, resulting in very low session goodput and/or aborted sessions.

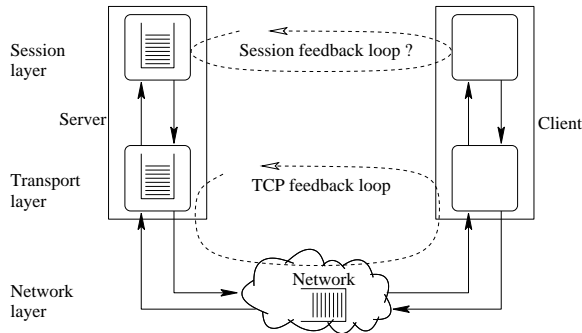


Figure 8: The TCP feedback loop at the transport layer cannot avoid persistent overload if there is no session layer congestion control.

We do not claim that TCP congestion control is not necessary. It is not sufficient, however, to avoid persistent overload. To understand this point, consider the previous example of a completely unresponsive session layer. When the network becomes congested, each active TCP connection backs-off either reducing its send-window by a large factor or getting into a relatively long silence period (retransmission timeout). This means that congestion control pushes the offered load from each connection back to the TCP buffer of the sender. That connection is still active, however, and so it will keep trying to retransmit any lost packets and to increase its window. As the session layer keeps generating new transfers, the number of competing flows will increase, leading to a diminishing per-session goodput. TCP cannot avoid the emerging persistent overload. Instead, we need a way to tell the session (or application) layer to slow down or stop generating new flows for a while.

In the following, we examine the congestion responsiveness of a traffic aggregate with the persistent and the two non-persistent flow models.

4.1 Congestion responsiveness

A traffic aggregate is called “congestion responsive” if its offered load reduces upon congestion. The specific congestion event that we consider here is an effective reduction of the capacity that is available to TCP flows from C to fC with $f < 1$. We assume that the offered load prior to the congestion event with all three models is ρC , with $f < \rho < 1$. In other words, the bottleneck link is not congested prior to the congestion event, but it does get congested during the congestion event.

Specifically, in the simulation of Figure 9, $C=50\text{Mbps}$, $\rho=80\%$ and $f=70\%$. The available capacity during each congestion event is 35Mbps . There are two congestion events, one starting at 50sec and another at 350sec ; they both last for 75sec .

4.1.1 Persistent model

The offered load with persistent flows can remain below the capacity when these flows are limited by the receiver’s advertised window. Note that, in Figure 9, the offered load with the persistent model follows closely the available capacity during the congestion events. The reason is that as the congestion event starts, the RTT of the TCP flows increases and their throughput decreases proportionally, to the point that the offered load becomes equal to the available capacity (i.e., there are no packet losses). So, as expected, the persistent model generates congestion responsive traffic. If there were packet losses, the aggregate traffic would still be congestion responsive because the TCP connections would decrease their send-windows to the point that the offered load becomes equal to (or less than) the available capacity.

4.1.2 Closed-loop model

Similarly, the offered load with the closed-loop model reduces during congestion events to the available capacity. Of course there is a larger variability in the traffic load, compared to persistent flows, for the reasons we discussed in the last section. Nevertheless, the closed-loop model also generates congestion responsive traffic. The reason is that when the bottleneck link becomes congested, the existing flows take longer to complete, and so the arrival of new flows also slows down.

4.1.3 Open-loop model

Figure 9 shows that the behavior of open-loop traffic is radically different both during and after congestion events. Recall that, with the open-loop Processor Sharing model, the offered load remains $\lambda S = \rho C$, independent of whether the bottleneck is congested. During a congestion event with $\rho > f$, the bottleneck link becomes unstable, as the offered load is larger than the available capacity. Consequently, the number of active TCP flows increases. Also, because TCP retransmits dropped packets, the offered load actually increases during the congestion event! This is even worse than the behavior of a congestion unresponsive UDP flow that does not retransmit dropped packets. After the congestion event ends, it takes several tens of seconds for the backlog of active flows to clear and for the offered load to return to ρC .

To summarize, an open-loop model of TCP flows is not congestion responsive, despite the fact that each flow is regulated by TCP congestion control. Also, because of TCP retransmissions (some of which are redundant), the offered load during congestion events increases, instead of decreasing.

ing, making such traffic the “network’s worse enemy”. This is an interesting point, considering that research in congestion control during the last few years has been focusing on the congestion responsiveness of UDP traffic, instead of examining the behavior of open-loop TCP flow/session arrivals.

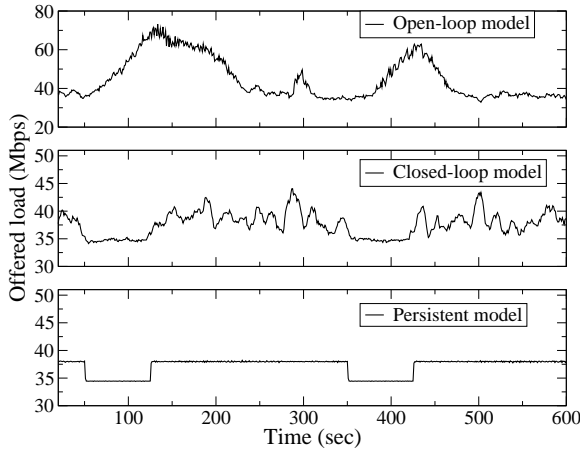


Figure 9: Congestion responsiveness of three TCP traffic models.

One final remark about the congestion responsiveness of the closed-loop and persistent models. Even though both models generate congestion responsive traffic, they do not react to congestion in the same way. Persistent flows typically recover from losses faster (with Fast-Retransmit instead of Retransmission Timeouts). Also, persistent flows are not affected as much as short TCP flows by the loss of the 3-way-handshake connection establishment packets. These two effects imply that a given loss rate will cause a larger reduction in the aggregate throughput of closed-loop transfers than of persistent flows.

To illustrate this point, Figure 10 shows the packet loss rate that results from two simulations, one with persistent flows and another with closed-loop TCP flows. The average number of closed-loop flows is equal to the number of persistent flows. The offered load in both cases is equal to 100% of the link capacity. Note that the loss rate with persistent flows is considerably higher. The reason is that, as explained in the previous paragraph, short TCP flows typically experience a larger throughput reduction upon a packet loss than persistent flows.

5. CASE STUDY II: ROUTER BUFFER SIZING

The problem of router buffer sizing has recently received significant attention [4, 5, 11, 20, 21, 26, 31]. Specifically, the “Stanford model” of [11] challenged the common practice of buffer sizing based on the bandwidth-delay product of a link [30], and it argues for much smaller buffers. The objective of the Stanford model is to achieve full utilization with minimum buffering, and so with minimum queuing delay. The authors of [11] showed analytically, with simulations, and with testbed experiments that a buffer size B , equal to CT/\sqrt{N} , where CT is the bandwidth-delay product of the link and N is the number of “elephant” TCP flows in the bottleneck, is sufficient to saturate that link. More

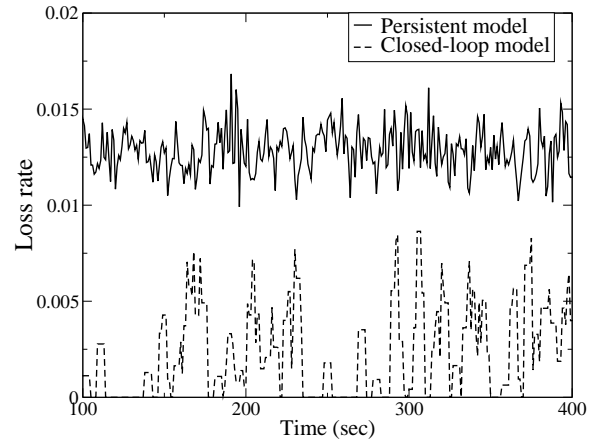


Figure 10: Loss rate with closed-loop flows and persistent flows.

recently, [21] argued that the buffer sizing requirement can be further reduced to just a few tens of packets, as long as the utilization remains below 80% or so.

Most of the buffer sizing literature, however, only considers the persistent TCP model. Even when non-persistent flows are included in the simulations, they typically contribute a small fraction of the aggregate traffic while most of the traffic is generated from persistent flows. This is also the case for the mathematical results of [4, 11, 21].

In this section, we show that non-persistent traffic models can lead to radically different insight and results for the buffer sizing problem. In particular, even though it is true that small buffers and a moderate utilization are sufficient for lossless operation with the persistent model, the loss rate with the open-loop or the closed-loop models are significant even when the average utilization is only 70%. The main reason is that the non-persistent models generate significantly higher traffic variability, as shown in Section 3.

Figure 11 shows the loss rate that results with the open-loop and the closed-loop models when the buffer size is 60 packets and the average utilization is about 72%. Note that according to the bandwidth-delay product rule-of-thumb, the router buffer size in this case should be much larger (415 packets). The persistent model results in lossless operation, as all TCP transfers are limited by their advertised windows (as assumed in [21]). Note that the loss process with the open-loop model is quite bursty, and that there are several periods in which the loss rate exceeds 5%. This is certainly an excessive loss rate for most applications and network operators. It should be emphasized that the bottleneck link is not heavily loaded, and that the high loss rate is a result of the significant variability in the open-loop model. The closed-loop model generates lower variability, compared to the open-loop model, and so the loss rate remains below 2.5% during this simulation interval. Nevertheless, the closed-loop model is still very different than the persistent model in terms of the required router buffer size.

The previous simulation considered the case of a moderately utilized link with a small buffer size. In Figure 12, we examine the loss rate at a congested link that is fully utilized, with each of the three TCP models that we consider. The buffer size varies from 10 packets to 600 packets. The bandwidth-delay product rule-of-thumb recommends a

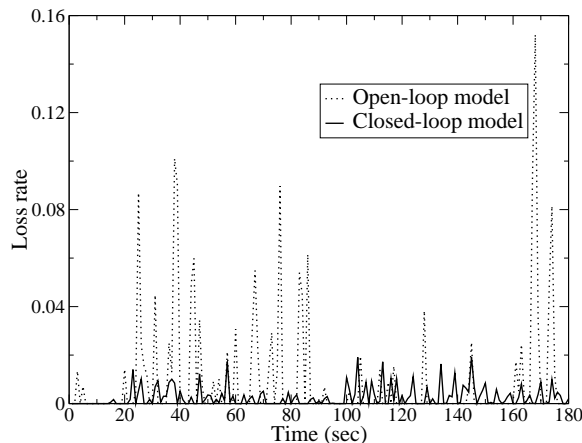


Figure 11: Loss rate with a 60-packet buffer and 72% average utilization (the loss rate with the persistent model is zero).

buffer size of 250 packets, while the Stanford model of [11] recommends a buffer of only 18 packets (the vertical line in the graphs).

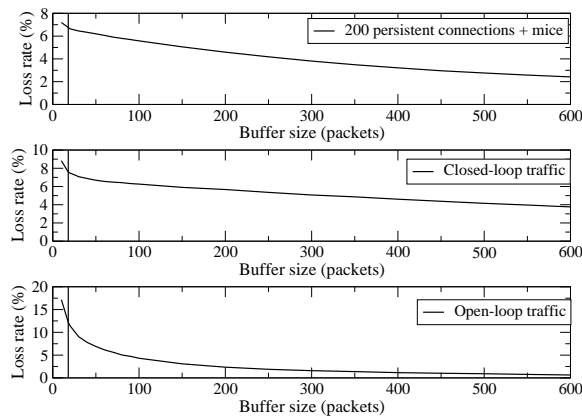


Figure 12: Loss rate as a function of buffer size for the three traffic models.

First note that the loss rate that results from the Stanford model (a buffer of 18 packets) is about 12% for open-loop TCP traffic and about 7-8% for persistent and closed-loop flows. Such high loss rates would probably be unacceptable in practice, and they would cause poor performance for certain types of applications (including reliable and interactive transactions, and audio or video streaming or conferencing).

Second, in the case of persistent flows, we can derive mathematically the loss rate p as a function of the buffer size B , under some simplifying assumptions. Specifically, we can use the following equation for the average throughput R of a persistent TCP flow with RTT T that experiences a loss rate p (see [23]):

$$R = \frac{0.87}{T\sqrt{p}} \quad (7)$$

The RTT T includes both propagation delay and any queu-

ing delays in the path. Suppose that N homogeneous connections with round-trip propagation delay T_p saturate a link of capacity C . Then, the loss rate at that link is given by

$$p = \frac{(0.87N)^2}{(CT_p + B)^2} \quad (8)$$

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 competing flows. Also, for fixed N , the loss rate decreases rather slowly as the buffer size increases (as a power law rather than exponentially fast).

Third, Figure 12 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 intraflow burstiness. Further, 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 can start sooner. In other words, the incoming traffic load increases as the loss rate at the bottleneck 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 12.

6. SUMMARY AND DISCUSSION

The main objective of this paper is to show that the model of persistent connections is not only unrealistic, but it is also misleading when it comes to the variability of the traffic in terms of offered load and number of active flows. We then identified two basic and well-known non-persistent flow generation models and explained how they can also lead to very different results in two specific problems: the congestion responsiveness of aggregate traffic and the sizing of router buffers. In the following, we discuss some more implications of this work in other areas of networking research and practice.

6.1 Which non-persistent model is more realistic?

It is hard to tell which of the two non-persistent models is more realistic. There are links for which the open-loop TCP model is more appropriate (e.g., the outgoing traffic from a popular Web server) and links where the closed-loop TCP model is certainly better (e.g., the ingress traffic to a SOHO network). For most links, however, we anticipate that a combination of the two models would be more realistic. An important problem for future measurement studies is to examine the percentage of traffic that can be modeled as open-loop versus closed-loop for various types of network links.

6.2 AQM and network stability

Active queue management (AQM) mechanisms, such as RED, REM, PI controllers, etc., have been proposed as a way to provide stability in the Internet. It is important to note that such stability studies assume persistent TCP

connections. With that traffic model, the AQM mechanisms can control and stabilize the queue length and the bottleneck link utilization. The effectiveness of AQM mechanisms with non-persistent traffic, however, is much less understood. As we showed in this paper, the offered load of open-loop TCP traffic does not depend on network state. AQM mechanisms cannot regulate such an aggregate, and they will be unable to avoid persistent overload if the offered load exceeds the network capacity.

6.3 Is admission control necessary?

Several researchers advocate the use of admission control as the only way to regulate the offered load and avoid the congestion collapse risk. We agree with them, if the traffic is mostly open-loop. Without admission control, the only way to avoid congestion collapse is to expect that users will be impatient and they will abandon very slow ongoing transfers. This is not an efficient way to control congestion. Admission control, on the other hand, can limit the number of active sessions or flows in the network and it can provide a throughput guarantee to each of them. Admission control may not be necessary, however, if most of the traffic at a congested link follows the closed-loop model.

6.4 TCP-friendly congestion control

The use of TCP-friendly congestion control has been encouraged in all non-TCP protocols and applications. The basic motivation for such proposals is that TCP-friendly transfers can avoid congestion collapse. We have shown, however, that even if a traffic aggregate consists entirely of TCP connections, it can still cause congestion collapse or persistent overload if it is open-loop. The same is obviously true for TCP-friendly traffic. Therefore, the use of TCP-friendly congestion control is not sufficient to guarantee stability. On the other hand, TCP-friendly congestion control is important and beneficial as a way to improve fairness in the bandwidth sharing among TCP and non-TCP transfers.

6.5 Traffic engineering and network provisioning

Traffic engineering, as well as other provisioning mechanisms, require estimates for the amount of traffic flowing between any inbound/outbound points in a network. Furthermore, such mechanisms assume that if a given traffic aggregate is switched from one route to another, then the throughput of that aggregate will *not* change. This assumption is not true for TCP persistent connections. It is well understood that the throughput of such transfers depends on the RTT and loss rate in the underlying path, raising concerns for the applicability of traffic engineering.

On the other hand, the offered load from open-loop TCP traffic does not depend on the underlying network path, making such traffic consistent with common assumptions in traffic engineering. The same is true for closed-loop TCP traffic, as long as the offered load remains below the capacity of the underlying paths.

6.6 New traffic models for simulations and analysis

Most of the previous research in congestion control assumed persistent TCP flows in both simulation and analysis. We believe that the community should abandon that assumption and adopt non-persistent traffic models instead.

It is also important that these models consider a mix of both open-loop and closed-loop TCP traffic. Especially, in the case of closed-loop traffic, the mathematical results are quite limited; more research in that direction would be valuable.

6.7 Session layer congestion control

At the more practical side, we recommend that all network applications use some form of congestion control at the session layer. This can be as simple as adopting one of the following rules: do not generate a new session until the previous session has completed, slow down the generation of new sessions if the network appears to be congested, or do not keep more than a certain number of sessions active. Some applications already follow similar rules.

It is also important that session layer congestion control is implemented in applications that generate transfers automatically, without user intervention. For example, NNTP servers transfer news to their peers periodically, independent of whether the underlying network is congested or not. CDN servers also perform such periodic transfers. Effectively, such applications generate open-loop TCP traffic, raising the possibility for congestion collapse if their aggregate load is comparable to the underlying capacity.

7. REFERENCES

- [1] 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.
- [2] T. Bonald, P. Olivier, and J. Roberts. Dimensioning High Speed IP Access Networks. In *18th International Teletraffic Congress*, 2003.
- [3] G. de Veciana, T. Lee, and T. Konstantopoulos. Stability and Performance Analysis of Networks Supporting Services. *IEEE/ACM Trans. on Networking*, 9(1), 2001.
- [4] A. Dhamdhere and C. Dovrolis. Buffer Sizing for Congested Internet Links. In *IEEE Infocom*, 2005.
- [5] A. Dhamdhere and C. Dovrolis. Open Issues in Router Buffer Sizing. *ACM CCR*, 2006.
- [6] S. Ebrahimi-Taghizadeh, A. Helmy, and S. Gupta. TCP vs. TCP: a Systematic Study of Adverse Impact of Short-lived TCP Flows on Long-lived TCP Flows. In *Proceedings of IEEE INFOCOM*, 2005.
- [7] A. Feldmann. Characteristics of tcp connection arrivals, 1998.
- [8] D. R. Figueiredo, B. Liu, V. Misra, and D. Towsley. On the Autocorrelation Structure of TCP Traffic. *Computer Networks Journal, Special Issue on "Advances in Modeling and Engineering of Long-Range Dependent Traffic"*, 2002.
- [9] S. Floyd and K. Fall. Promoting the Use of End-to-End Congestion Control in the Internet. *IEEE/ACM Transactions on Networking*, 7(4):458–473, Aug. 1999.
- [10] S. B. Fredj, T. Bonald, A. Proutiere, G. Regnie, and J. W. Roberts. Statistical Bandwidth Sharing: A Study of Congestion at Flow Level. In *Proceedings of ACM SIGCOMM*, Aug. 2001.
- [11] G. Appenzeller and I. Keslassy and N. McKeown. Sizing router buffers. In *ACM SIGCOMM*, 2004.

- [12] L. Guo and I. Matta. The War Between Mice and Elephants. In *Proceedings of IEEE ICNP*, 2001.
- [13] D. Heyman, T.V.Lakshman, and A. L. Neidhardt. A New Method for Analysis Feedback-Based Protocols with Applications to Engineering Web Traffic over the Internet. In *ACM SIGMETRICS*, pages 24–38, 1997.
- [14] Y. Joo, V. Riberio, A. Feldmann, A. Gilbert, and W. Willinger. TCP/IP traffic dynamics and network performance: A lesson in workload modeling, flow control, and tracedriven simulations. *ACM CCR*, Apr 2001.
- [15] A. A. Kherani and A. Kumar. Stochastic Models for Throughput Analysis of Randomly Arriving Elastic Flows in the Internet. In *Proceedings of IEEE INFOCOM*, 2002.
- [16] L. Kleinrock. Time-shared Systems: A Theoretical Treatment. *Journal of the ACM*, 14(2):242–261, 1967.
- [17] S. Kunniyur and R. Srikant. Stable, Scalable, Fair Congestion Control and AQM Schemes that Achieve High Utilization in the Internet. *IEEE Transactions on Automatic Control*, 49:2024–2029, 2004.
- [18] L. Le, J. Aikat, K. Jeffay, and F. Smith. Differential Congestion Notification: Taming the Elephants. In *Proceedings of ICNP*, 2004.
- [19] L. Le, J. Aikat, K. Jeffay, and F. D. Smith. The Effects of Active Queue Management on Web Performance. In *Proceedings of SIGCOMM*, 2003.
- [20] M. Enachescu and Y. Ganjali and A. Goel and T. Roughgarden and N. McKeown. Part iii: Routers with very small buffers. *ACM/SIGCOMM Computer Communication Review*, 35(3), 2005.
- [21] M. Enachescu and Y. Ganjali and A. Goel and T. Roughgarden and N. McKeown. Routers with very small buffers. In *IEEE INFOCOM*, 2006.
- [22] V. Misra, W. B. Gong, and D. Towsley. Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED. In *Proceedings of ACM SIGCOMM*, Sept. 2000.
- [23] J. Padhye, V.Firoiu, D.Towsley, and J. Kurose. Modeling TCP Throughput: A Simple Model and its Empirical Validation. *IEEE/ACM Transactions on Networking*, 2000.
- [24] F. Paganini, Z. Wang, J. C. Doyle, and S. H. Low. Congestion Control for High Performance, Stability, and Fairness in General Networks. *IEEE/ACM Trans. Netw.*, 13(1):43–56, 2005.
- [25] F. Paganini, Z. Wang, S. H. Low, and J. Doyle. A new TCP/AQM for Stable Operation in Fast Networks. In *INFOCOM*, 2003.
- [26] G. Raina, D. Towsley, and D. Wischik. Part II: Control Theory for Buffer Sizing. *ACM Computer Communication Review*, 2005.
- [27] J. Roberts. A Survey on Statistical Bandwidth Sharing. *Computer Networks*, 45:319–332, 2004.
- [28] B. Schroeder, A. Wierman, and M. Harchol-Balter. Closed Versus Open System Models and their Impact on Performance and Scheduling. In *Symposium on Networked Systems Design and Implementation (NSDI)*, 2006.
- [29] B. Sikdar, K. S. Vastola, and S. Kalyanaraman. Analytic Models for the Latency and Steady-State Throughput of TCP Tahoe, Reno and SACK. *IEEE/ACM Transactions on Networking*, 11(6):959–971, 2003.
- [30] C. Villamizar and C.Song. High Performance TCP in ANSNET. *ACM Computer Communication Review*, Oct. 1994.
- [31] D. Wischik and N. McKeown. Part I: Buffer Sizes for Core Routers. *ACM Computer Communication Review*, 2005.