# Beyond the Model of Persistent TCP Flows: Open-Loop vs Closed-Loop Arrivals of Non-Persistent Flows

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

## Abstract

*It is common for simulation and analytical studies to model Internet traffic as an aggregation of mostly persistent TCP flows. In practice, however, flows follow a heavy-tailed size distribution and their number fluctuates significantly with time. One important issue that has been largely ignored is whether such non-persistent flows arrive in the network in an open-loop (say Poisson) or closed-loop (interactive) manner. This paper focuses on the differences that the TCP flow arrival process introduces in the generated aggregate traffic. We first review the Processor Sharing models for such flow arrival processes as well as the corresponding TCP packet-level models. Then, we focus on the queueing performance that results from each model, and show that the closed-loop model produces lower loss rate and queueing delays than the open-loop model. We explain this difference in terms of the increased traffic variability that the open-loop model produces. The cause of the latter is that the flow arrival rate in the open-loop model does not reduce upon congestion. We also study the transient effect of congestion events on the two models and show that the closed-loop model results in congestion-responsive traffic while the open-loop model does not. Finally, we discuss implications of the differences between the two models in several networking problems.*

## 1 Introduction

Models of aggregate TCP traffic are valuable in networking research and practice. Much of the previous work in this area has been focusing on the model of *persistent TCP flows*, i.e., on flows that have unlimited data to send and that are not limited by the receiver advertised window. This model is mathematically tractable and it is easier to simulate, but at the same time it fails to capture key aspects of real Internet traffic [10]. Specifically, it ignores the heavy-tailed nature of the flow size distribution (that

can produce Long-Range Dependency), the significant variations in the number of active flows with time, and the relation between congestion and the flow arrival process. On the other hand, some previous work has considered *non-persistent TCP flows*, following a heavy-tailed size distribution. The open issue there is whether the arrival process of the TCP flows should be modeled in an *open-loop* (OL) manner (say, according to a Poisson process), or in a *closed-loop* (CL) manner (say, from a number of interactive users). This paper focuses on the differences that the flow arrival process, OL versus CL, causes in the generated aggregate traffic. The related issue of which model is more realistic has been the focus of a recent measurement study [16].

We start with the fluid *Processor Sharing* (PS) models for the OL and CL flow arrival processes. The PS models provide an accurate estimate of the offered load in light/moderate load conditions. On the other hand, when the load approaches the capacity, the PS models can lead to significant underestimation of the offered load. The main problem is that the PS models ignore packet losses and TCP retransmissions, which are a significant contribution of additional load in congested links. Nevertheless, the PS models show clearly that the open-loop model can be unstable, while the closed-loop model is always stable, as the number of active flows is bounded.

We then compare the queueing performance of the (packet level) OL and CL models, examining the loss rate and queueing delay distribution that the two models produce under the same offered load. The OL model produces higher loss rate and queueing delays than the CL model. To explain this difference, we examine the traffic variability produced by the two models in a range of timescales (10msec-1sec). We find out that the OL model results in higher variance than the CL model, especially when the timescale exceeds the TCP Round-Trip Time (RTT). The cause of the increased traffic variability in the OL model is that the latter does not reduce the flow arrival rate upon congestion. This leads to more significant overload events, in magnitude and duration, than the CL model, generating higher traffic variability. The CL model responds to conges-

tion roughly one RTT after its occurrence, which explains why the variability difference becomes significant when the timescale is larger than the RTT.

We also examine the distribution of the number of active flows with each flow arrival model. Here, we find that the OL model results in higher variability in the number of flows than the CL model when the offered load is significant. There are time periods in which the number of ongoing flows with the OL model is much higher than the average. This observation is related to an earlier study by Schroeder et al. which showed that job scheduling is crucial mostly with the OL model, as the former gives a wider leeway to the scheduler than the CL model [18].

Finally, we focus on the transient response of the two models in terms of the *congestion responsiveness* of the aggregate traffic. With OL flow arrivals, the resulting traffic is *not* congestion responsive, meaning that the offered load does not follow the available capacity in the network. With CL flow arrivals, on the other hand, the traffic is congestion responsive.

The rest of the paper is organized as follows. Sections 2 and 3 review previous work and the limitations of the persistent flows model, respectively. Section 4 describe the OL and CL models and review basic results about the corresponding PS models. Our simulation setup is presented in Section 5, while the queueing and offered load differences between the two models are presented in Sections 6 and 7. Section 8 examines the variability in the offered load at different timescales, while Section 9 shows the variation in the number of active flows. Section 10 focuses on the congestion responsiveness of the two models. We conclude the paper in section 11, also discussing some implications of this work in various areas of networking research and practice.

## 2    Related Work

Over the last few years, and especially after the seminal work by Kelly et al. [11], several researchers applied control theory to examine the stability of TCP congestion control [14, 20, 15]. A key point about that line of work is that it assumes persistent TCP flows, and it focuses on the asymptotic stability of the queue size at the network bottleneck. The assumption of persistent flows removes from the problem the importance of the flow arrival process.

Some previous work uses non-persistent flow models, but often without discussing whether the OL or the CL model is more appropriate. Ben Fredj et al. [7] considered the OL model. 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. [4] considered the OL

model and concluded that the Internet traffic may become unstable under certain conditions.

Heyman et al. [8] used a CL model to analyze the performance of Web-like traffic over TCP. They showed that the session goodput and the fraction of time the system has a given number of active sessions are insensitive to the distribution of session sizes and "think times", and they only depend on the mean of these distributions. Berger and Kogan [2], as well as Bonald et al. [3], used a similar CL model to design bandwidth provisioning rules for meeting certain throughput-related QoS objectives. Bondi and Whitt [1] examine the differences between the OL and CL models in the context of networks of queues, focusing on the relation between the average queue size at the bottleneck queue and the variability in the job service-time distribution.

Most of the previous work with the OL or CL models assumes that TCP congestion control can share the capacity of a link as a fluid PS server [17]. Kherani and Kumar [12] 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. In this paper, we use the PS model just to gain some basic analytic insight. Most of our conclusions are based on NS2 simulations with TCP transfers.

In a paper that is closely related to our work, Schroeder et al. [18] compare the OL and CL models in a general context of job arrivals at a server. They highlight the differences between the two models in terms of the mean job completion time, and they focus on the effectiveness of different job scheduling policies with each model.

More recently, we have analyzed several traffic traces collected at a dozen of Internet links in order to estimate the fraction of traffic that can be mapped to either the OL or CL model [16]. That measurement study shows that about 60-80% of traffic to/from well-known ports (mostly HTTP) follow the CL model. Nevertheless, the percentage of OL traffic is significant in some links. Additionally, we could not classify reliably up to 70% of the traffic in certain traces.

## 3    Critique of the Persistent Flows Model

It is common for analytical and simulation studies to model most of the traffic with persistent TCP connections. A common argument to justify this model is thatmost traffic in an Internet link is carried by a few large TCP flows *elephants* and so those flows can be modeled as persistent. The smaller flows, referred to as *mice*, do not contribute a significant amount of traffic and so they are often ignored, or they are viewed as a source of stochastic noise in simulation studies. The previous argument is an oversimplification and it ignores two key characteristics of real Internet traffic. First, the size of TCP flows follows a continuous and heavy-tailed distribution in practice, rather than a bimodal

distribution in which flows are either very short (mice) or very long (elephants). In other words, the previous argument ignores the flows of significant, but not extreme, size. Second, flows with very large size (relative to other flows in the aggregate) do not always have very long duration. Some large flows get higher 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 these issues, we 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 carries the inbound traffic in a Gigabit Ethernet segment that connects the campus network to the SoX GigaPoP. The objective of this traffic analysis is to examine the assumptions behind the persistent flows model. Note that similar studies have been conducted several times in the past (for instance, see [6]), using traces from many links and under diverse load conditions.
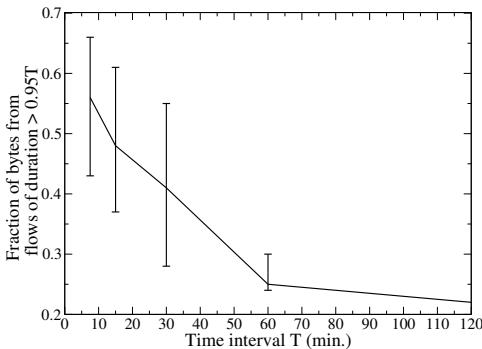


**Figure 1. The fraction of bytes $f$ generated by flows that are active for the entire duration of a given time interval $T$ as a function of $T$. The error bars depict the minimum and the maximum values of the fraction $f$.**

We first looked at the flow size distribution. We find that the C-CDF of that distribution shows clear linear decrease in a log-log plot, pointing to the heavy-tailed Pareto distribution (with shape parameter about 1.3). We also examined the distribution of flow interarrivals. When the interarrivals are larger than 100msec or so, they can be modeled as exponential and independent (pointing to a Poisson flow arrival process). However, there are significant correlations in lower interarrivals, probably due to the generation of simultaneous flows by the same application session.

Next, we measured the number of active flows as a function of time for different flow size thresholds. If we only consider flows that are larger than 1.5MB (or $\sim$1000pkts), the number of active flows remains almost constant with

time. This observation, however, should not be interpreted as validation of the persistent flows model. The reason is that even though the number of (sufficiently large) active flows remains roughly constant with time, *the set of active flows changes significantly with time.* To illustrate this point we examined the fraction of bytes $f$ that is generated by flows that remained active throughout a given time interval of length $T$. With the persistent model, all flows are active throughout $T$ and this fraction should be close to 100%. If all flows lasted for less than $T$ seconds, then this fraction should be zero. We measured $f$ for the following values of $T$: 7.5, 15, 30, 60 and 120 minutes. For each value of $T$ (except for 120 minutes), we obtained 30 samples of the fraction $f$, ignoring the first two minutes of the trace, and considering flows that last longer than $0.95T$ as active throughout the duration $T$. Figure 1 shows the mean, the minimum and the maximum value of $f$ as a function of $T$. The key observation here is that even for time intervals that last only 5-10mins, the fraction of traffic from persistent flows is only 40-70%. So, the assumption that the same set of flows carries almost all traffic ignores the variability due to the dynamic flow arrival and completion processes.

## 4  Two Models of Non-Persistent Flow Arrivals

In this section, we describe two basic models of non-persistent flow arrivals: the OL and CL models. Both models are simple and well-studied in the performance evaluation literature.

Note that the terms "open-loop" and "closed-loop" have been previously used to distinguish between non-TCP traffic (viewed as open-loop because packets arrive randomly based on an exogenous process) and TCP traffic (viewed as closed-loop because the flow is regulated by TCP congestion control). In this paper, both OL and CL models describe an aggregate of TCP flows. They differ, however, in the higher level process, operating at the session or application layer, that generates these flows. Figure 2 shows a schematic diagram of the flow generation process. If the session layer uses some feedback from the network, so that it slows down the generation of new flows upon congestion, the resulting traffic will be closer to the CL model. Otherwise, in the absence of such feedback, the OL model is more appropriate.

### 4.1  Open-Loop model

In the OL model, users or applications generate flows independent of any 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 large population of users located anywhere in the Internet.
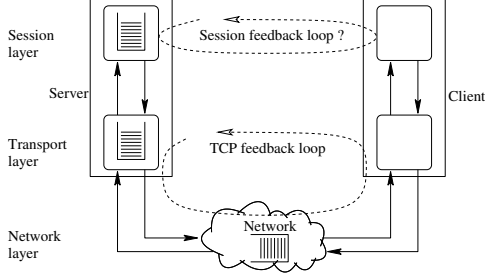
**Figure 2. The flow arrival process is controlled by the session/application layer. Is that layer responsive to network congestion?**

Assume that a user does not return to this server, at least for a long time, after completing a file transfer. Consequently, the server's sessions are always with new users. If the link becomes congested, the arrival rate of new sessions will not be affected, as Internet users are typically unaware of the network state in a given path.

Consider a PS server with capacity $C$ (bytes/sec), average flow arrival rate $\lambda$ (flows/sec), and average flow size $S$ (bytes). We refer to this model as PS-OL. The average offered load in the server is given by $\lambda S$ and the *normalized offered load* is

$$\rho_o = \lambda S/C. \tag{1}$$

If $\rho_o < 1$, the server is stable and $\rho_o$ is the average utilization. For a Poisson flow arrival process, it can be shown that the average number of active flows is given by [13]

$$\bar{N}_o = \frac{1}{(1 - \rho_o)}. \tag{2}$$

Otherwise, if $\rho_o > 1$, the server is unstable (as long as flows are never aborted). Since both the flow arrival rate and the average flow size are independent of the network state, the average offered load remains constant even in the presence of congestion. Further, the expected throughput of a new flow in the PS-OL model is given by the available capacity in the server,

$$\bar{R} = C(1 - \rho_o). \tag{3}$$

Deviating from the PS model, we can consider a packet-level model of a First-Come-First-Served (FCFS) queue with a finite buffer and with flows that are controlled by TCP congestion control (TCP-OL). Notice two important differences between the TCP-OL model and the PS-OL model. In the former, we can have packet drops. TCP reacts to them with retransmissions, which effectively *increase* the size of the affected flows. Further, it is well known that TCP can generate redundant retransmission. This means that the actual offered load by a set of TCP flows in the OL model

can be higher than what the PS model predicts in Equation (1). Second, a TCP flow can be active even when it does not compete for available capacity, because of window limitations due to slow-start, retransmission timeouts, limited advertised window, etc. This means that the average number of active TCP flows can be much larger than Equation (2).

## 4.2 Closed-Loop model

To illustrate the CL model, consider the access link of a small enterprise with, say $N$, users. In the inbound direction, most of the traffic at the link is downloads that are generated by the activity of these $N$ users. In the simplest model, each user can be in the "Active" state downloading a file, then spending some time in the "Idle" (or "Thinking") state, and then either downloading another file, or leaving the system for a longer time period ("Inactive" state). This link would not carry more than $N$ active flows at any time. Furthermore, if the link becomes congested, then the download latencies of all active flows will increase, reducing the rate with which new flows are generated.

In the PS version of the CL 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 session arrival rate in the CL model is given by

$$\lambda_c = \frac{N}{T_t + T_i} \tag{4}$$

where $T_t$ is the average flow transfer latency. The latter depends on the load at the PS server. Thus the average server utilization at the PS-CL model is given by

$$\rho_c = \frac{NS}{C(T_i + T_t)}. \tag{5}$$

The average number of active flows in the PS-CL model is given by (see [2])

$$\bar{N}_c = \frac{a}{1-a} \quad \text{for} \quad a \ll 1$$
$$= N\left(1 - a^{-1}\right) = N - \frac{CT_i}{S} \quad \text{for} \quad a > 1 \tag{6}$$

where the normalized offered load is given by

$$a = NS/CT_i. \tag{7}$$

Note that the expected number of active flows for $a \ll 1$ is same with the OL model. On the other hand, when $a > 1$, $\bar{N}_c$ increases slowly with $a$ and remains bounded by $N$.

Similar to the TCP-OL model, the CL model with a FCFS queue and TCP flows (TCP-CL) can deviate significantly from its PS-CL counterpart. First, as in TCP-OL, we need to consider the extra load due to required or redundant retransmissions. Second, as in the TCP-OL model, TCP is not able to always use the available capacity.

## 5 Simulation Setup

The previous section reviewed well-known results for the OL and CL models, based on the PS model. In this work, we are more interested in TCP-specific effects that cannot be captured by the PS model, as well as on the variance of the resulting aggregate traffic. For these reasons, we rely mostly on simulation.

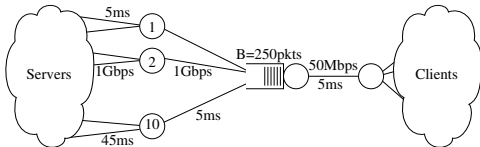Figure 3 shows our *NS-2* simulation setup. There are



**Figure 3. Simulation setup**

10 input links, each with capacity 1Gbps, connected to an output link with capacity $C$=50Mbps and buffer size $B$. This topology describes a scenario in which the bottleneck is the ingress link of an enterprise network, and where the server, backbone and client links are over-provisioned. In this setup, we have 20 servers that are connected to the bottleneck with 1Gbps links and with propagation delays that vary between 5msec and 45msec. The round-trip propagation delay in this setup varies from 30msec to 110msec, with a harmonic mean of about $T_0$=60msec[1].

In all simulations we use the SACK-enabled *NS-2* TCP module *sack1*. The buffer size $B$ is set to the bandwidth-delay product of the path (250 packets), considering $T_0$ as the representative delay. The maximum advertised window is set to 256 packets. In the CL simulations, there are $N$ clients that initiate TCP transfers. These users arrive for the first time at the network at a random instant during the first few seconds of the simulation. After arriving, each user follows the CL flow generation process selecting a server for each transfer randomly from the set of 20 servers. In the OL simulations, the flow arrival process is Poisson with arrival rate $\lambda$. In all simulations, the flow size follows a Pareto distribution with a mean of 25 packets and shape parameter 1.5. The think time $T_i$ for the CL model follows an exponential distribution with a mean of 2 seconds. The values of $\lambda$ and $N$ are varied to obtain different offered loads in the OL and CL models, respectively. Each simulation runs for 1000 seconds and we report results for the period from 200 to 950 seconds.

## 6 Controlling the Offered Load

To compare the traffic characteristics and queueing performance of the OL and CL models, we first need to make

---

[1]The use of the harmonic mean has been recommended in [5].

sure that their parameters are selected so that both models produce *equal average offered load*. The offered load is defined as the amount of traffic that *arrives* at the bottleneck link per unit of time, and it includes traffic that may get dropped due to congestion.

Controlling the offered load in the OL and CL models, however, is not trivial. Suppose that we want to generate a certain offered load $X$ at the bottleneck link of the previous simulation setup. Given the average flow size $S$ (and the average think time $T_i$ in the case of the CL model), a common approach is to rely on the PS model. For the OL model, we can calculate the required flow arrival rate as $\lambda = X/S$. For the CL model, however, the term $T_t$ depends on the given load conditions. A crude approximation is to assume light load conditions ($a \ll 1$), and thus $T_t \ll T_i$. Then, the required number of users is $N = X\,T_i/S$.
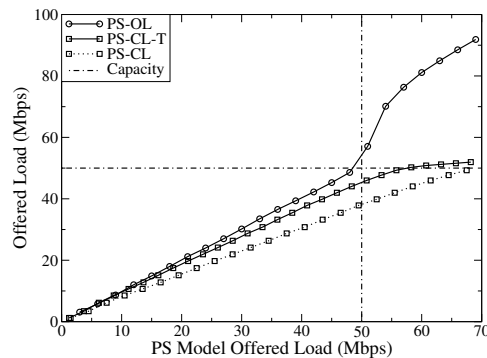


**Figure 4. The offered load with the TCP-OL and TCP-CL models (simulated, y-axis) as a function of the offered load $X$ that is predicted by the corresponding PS models (calculated, x-axis).**

Next, we examine the relation between the average offered load $X$ predicted by the two PS models, as previously described, and the actual offered load that we observe in simulations with TCP traffic (TCP-OL and TCP-CL models). Figure 4 shows the results of this comparison. The capacity lines $X = C$ and $Y = C$ are shown for reference. We observe that the offered load with the TCP-OL model is very close to the load $X$ predicted by the PS-OL model, as long as $X$ remains below the capacity $C$. As $X$ approaches $C$ the TCP-OL offered load starts deviating from $X$, and when $X > C$ (overload) the TCP-OL offered load is significantly higher than $X$. The reason is that the TCP-OL offered load includes retransmissions (required or redundant) of dropped packets. The fact that the increase rate of the TCP-OL offered load drops as $X$ goes more deeply into overload is due to the increasing frequency of retransmission timeouts that the TCP connections experience. Nevertheless, the important observation here is that we can use

the offered load predicted by PS-OL model as a reasonable approximation, as long as $X < C$.

In the case of the CL model, the offered load predicted by the PS-CL model is lower than that with TCP-CL, even in light/moderate load conditions. The reason, of course, is that we have ignored the load-dependent transfer time $T_t$, assuming that it is much less than $T_i$. Especially for TCP flows, however, we cannot ignore that for a flow of size $S$ there is a minimum transfer latency of several RTTs due slow-start, even if there are no queueing delays or packet losses. Thus, we next consider the following improved approximation of the offered load with the PS-CL model,

$$X = \frac{NS}{T_i + T_{t,min}(S)} \quad (8)$$

where $T_{t,min}(S)$ is the minimum latency required by TCP to transfer a flow of size S using slow-start. It is simple to estimate this parameter as long as the RTT and the TCP variant used are known. We refer to this approximation as *the PS-CL model with a constant term for the minimum transfer time of the average flow size*, or PS-CL-T for short. Figure 4 shows the relation between the offered load of the TCP-CL and PS-CL-T models (with $T_{t,min}(S)$=0.36sec in our simulations). Note that the latter is a reasonably good approximation both when the link is not congested ($X < C$) and in overload ($X > C$). The reason the offered load is slightly above the capacity in overload is again the presence of some TCP retransmissions. In summary, the PS model can provide a reasonable approximation of the offered load in the TCP-CL model, as long as we consider the minimum transfer time with TCP slow-start for the average flow size.
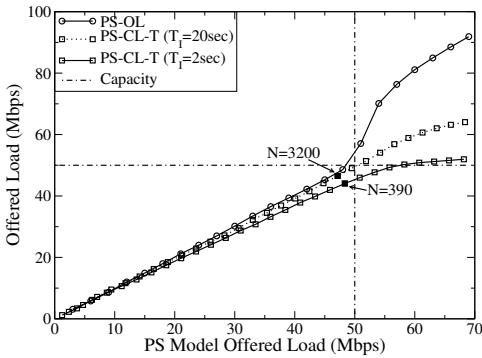


**Figure 5. The offered load from the CL model tends to that of the OL model as $N$ and $T_i$ increase.**

Notice that the OL model can be viewed as the asymptotic limit of the CL model, if we let the number of users $N$ and the average idle time $T_i$ go to infinity, while the initial transfer of each user is randomly placed on the time axis.

Indeed, we may wonder whether the offered load with the TCP-CL model approaches that of the TCP-OL model as we increase $N$ and $T_i$. Figure 5 shows the offered load from the TCP-CL model for two values of $T_i$, 2 and 20 seconds. Note that an increase in the idle time also requires an increase in the number of users in order to attain the same offered load. For example, with $T_i$=2sec we need 400 users to get 46Mbps of offered load, while with $T_i$=20sec we need 3200 users. We see that the offered load between the three curves differs mostly in overload, as expected. As we increase $T_i$ and $N$, the TCP-CL curve approaches the TCP-OL curve, implying the gradual convergence of the CL model to the OL model. Notice however that this convergence is very slow and in practice we would need a very large number of users before we can claim that the a closed population of users can be modeled with the OL model, in overload conditions.

In the rest of the paper, we use the offered load that is calculated from *ns-2* simulations.

## 7 Queueing Performance

Next, we compare the queueing performance of the TCP-OL and TCP-CL models. The main observation is that, under the same offered load, the TCP-OL model results in higher queueing delays than the TCP-CL model. If there are packet losses, then the loss rate with TCP-OL is also higher than with TCP-CL.
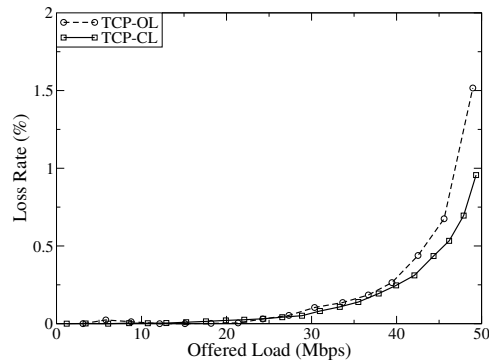


**Figure 6. The loss rate as a function of the offered load for the TCP-OL and TCP-CL models.**

Figure 6 and 7 show the loss rate and the queueing delays for the TCP-OL and TCP-CL models as a function of the offered load. For queueing delays, we report the median and the 90-th percentile of the per-packet delay distribution. The differences are of course minor for light load conditions, when the offered load is, say, below 50% of the capacity. In heavier load conditions, however, the differ-
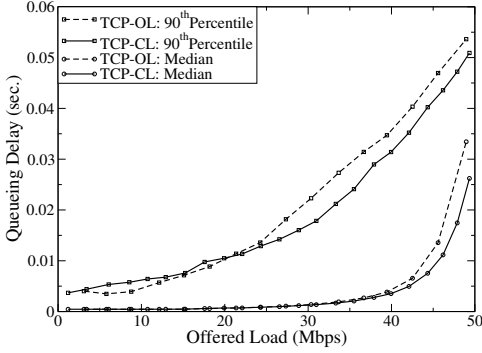
**Figure 7. The median and 90-th percentile of the queueing delay distribution as a function of the offered load for the TCP-OL and TCP-CL models.**

ences are significant and cannot be ignored. In the next section we explain these differences examining the statistical variability of the aggregate traffic in different timescales.

## 8 Traffic Variability at Different Timescales

The results of the previous section suggest that the TCP-OL model produces larger traffic burstiness than the TCP-CL model. In this section we aim to further understand what causes this difference and to identify the load conditions and timescales in which this is more evident.

Figure 8 shows the variance of the offered load for an averaging timescale of 10msec, 100msec and 1sec. First, notice how the variance depends on the offered load. The variance increases up to a certain point (20-45Mbps, depending on the timescale and the model). After that point the variance decreases with the offered load. For an explanation of this well-understood trend we refer the reader to [9, 19]. What is more relevant here is that the TCP-OL model produces higher variance than the TCP-CL model in moderate/heavy load conditions. Since the round-trip propagation delays in our simulation topology vary from 30msec to 110msec, we view the timescale of 10msec as below the typical RTT, 100msec as roughly equal to the RTT, and 1sec as larger than the RTT. The results of Figure 8 also suggest that the difference in the variance of the two models is more significant when the timescale is around the RTT or higher.

In light load conditions the two models are practically equivalent, as there is no significant queueing or packet losses and transfers are only limited by TCP's slow-start. As the offered load increases beyond roughly 50% of the capacity, congestion episodes start to occur. In the TCP-OL model, new flows arrive independent of whether the bottleneck is congested or not. In the TCP-CL model, when a flow slows down because of congestion it also delays the

generation of the next flow from the same user. This reduces the duration and magnitude of congestion events, leading to lower traffic variability than in the TCP-OL model. The response latency of the TCP-CL model cannot be faster than TCP's RTT however; this explains why the two models "look" the same in sub-RTT timescales.
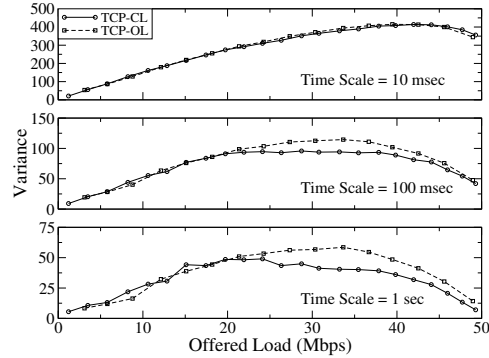


**Figure 8. Variance of the offered load with the TCP-OL and TCP-CL models for three averaging timescales.**
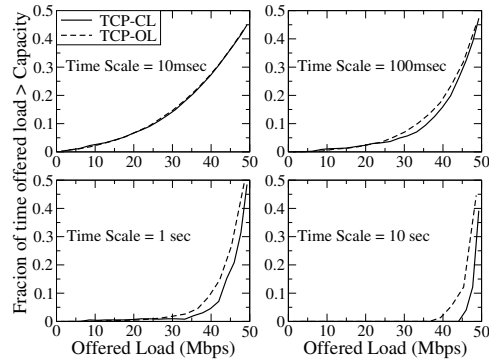


**Figure 9. Fraction of time the offered load is greater than the capacity for four averaging timescales.**

To further illustrate the previous explanation, Figure 9 shows the fraction of time the offered load exceeds the link capacity in four averaging timescales. Here we see that in the sub-RTT timescale of 10msec, both models experience overload for practically the same fraction of time. When we examine the traffic at higher timescales than the RTT, however, we confirm that the TCP-OL is overloaded more frequently. The TCP-CL model experiences overload less often because its flow arrival rate reduces upon the occurrence of packet losses. Since the two models have the same average offered load, the higher overload frequency in TCP-OL is compensated with time periods in which the TCP-OL

offered load is less than that in TCP-CL. These wider fluctuations make the variance of TCP-OL higher, as long as the the offered load and timescale are sufficiently large.
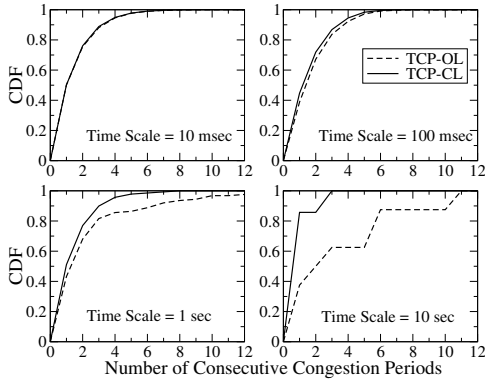


**Figure 10. The CDF of the length of consecutive overload periods for four averaging timescales. The average offered load is 95% of the capacity.**

It is not just the frequency of overload events that differs between the two models, but also their duration. This is shown in Figure 10, where we plot the CDF of the duration of overload events at different timescales for an average offered load of 47.5Mbps. This duration is measured as the number of consecutive time periods (with length equal to the averaging timescale) in which the offered load is higher than the capacity. In the sub-RTT timescale both models have the same distribution. As the timescale increases, however, the gap between the two distributions increases, as the TCP-OL model is unable to self-regulate its offered load below the capacity. For instance, when we look at the traffic in successive intervals of 10 seconds, about 85% of the overload events in TCP-CL last for only one interval. The corresponding percentage is only 40% in the TCP-OL model.

## 9   Number of Active Flows

In this section, we examine the number of active flows created by the TCP-OL and TCP-CL models. We show that the number of active flows in these two TCP models is much larger than that predicted by the PS model, and that TCP-OL produces higher variability in the number of active flows than the TCP-CL model, in heavy load conditions. The latter implies that the per-flow throughput in the TCP-OL model is also less predictable.

Figure 11 shows the CDF of the average number of active flows when the offered load is 70% and 95% of the capacity. The number of active flows is averaged over 1-sec intervals. We first note that the number of active flows in
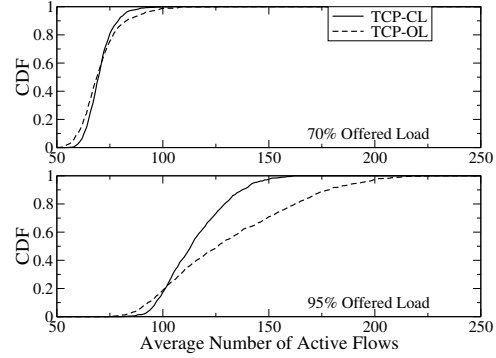


**Figure 11. The CDF of the average number of active flows, measured at 1-sec intervals, from the TCP-OL and TCP-CL models when the offered load is 70% and 95% of the capacity.**

both models is much higher than that predicted by the processor sharing model (see Equations 2 and 6). Specifically, the PS-OL model predicts about 3 and 20 flows for offered load 70% and 95%, respectively. The corresponding averages from the TCP-OL simulations are 70 and 131. For the PS-CL model, on the other hand, Equation 6 predicts an average of 102 active flows for 95% offered load, while the average from the TCP-CL simulations is 115. These differences can be attributed to the fact that, with TCP, there is a large number of small flows that are not always competing for available capacity because of slow-start, retransmission timeouts, or other limitations.

Also notice that the TCP-OL model results in much higher variability in the number of active flows in heavy load conditions. Again, this is because the TCP-OL model does not reduce the flow arrival rate upon congestion. The number of active flows in the TCP-CL model, on the other hand, is always bounded by $N$. The increased variability in the number of active flows with the TCP-OL model means that the per-flow throughput with that model is less predictable than with TCP-CL.

## 10   Congestion Responsiveness

So far we have focused on the steady-state behavior of the two models. In this section, we examine their transient response to individual congestion events.

We refer to a traffic aggregate as *congestion responsive* if it reduces its offered load upon overload to a point that there is no longer congestion. The specific congestion event that we consider here is a periodic UDP stream with rate that is higher than the available capacity in the bottleneck. Given that the UDP stream does not react to congestion,

the event that we simulate represents a sudden reduction of the available capacity for the TCP aggregate from $C$ to $C' = (1 - f)C$, where $fC$ is the rate of the UDP stream. In the following, we make the offered load $\rho C$ before the congestion event to be at the same level in the TCP-OL and TCP-CL models. We set $1 - \rho < f < 1$, so that the bottleneck becomes congested when the UDP stream starts.
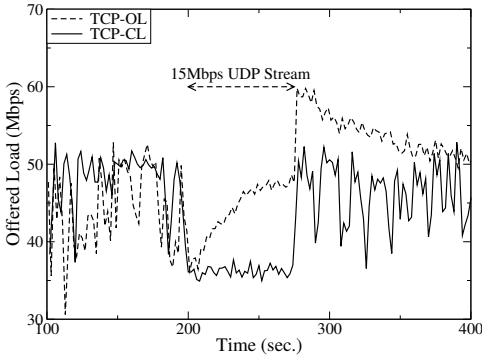
**Figure 12. The response of the traffic aggregate in the TCP-OL and TCP-CL models, when a congestion event is caused by a UDP stream of rate 15Mbps. The capacity is 50Mbps and the offered load (before the congestion event) is 47.5Mbps.**

Figure 12 shows the offered load from the two traffic models in 1-sec intervals. The congestion event is caused by a 15Mbps CBR UDP stream and it lasts from 200sec to 275sec. The effects of the congestion event can be examined in three stages: first, just after the congestion event starts, second, during the congestion event, and third, after the congestion event finishes.

Before the start of the congestion event, both TCP-OL and TCP-CL have the same average offered load. Their response when the UDP stream starts is that, because of TCP's congestion control, the traffic from both models drops at a level that is close to the new available capacity (35Mbps). The similarity between the two models, however, ends there. A few seconds later the offered load in the TCP-OL model starts increasing, as more and more new flows arrive and compete for throughput. The offered load in the TCP-CL model, on the other hand, is self-regulated at the level of the available capacity, because a new flow cannot start unless an existing flow has completed. Thus, the number of active flows in the TCP-OL model keeps increasing, while the corresponding number in the TCP-CL model stays roughly the same (also see Figure 13).

Finally, after the congestion event ends, the offered load from both models increases to capture the available capacity that has been released by the UDP stream. In the TCP-CL model, this process is completed within a few seconds. In

the TCP-OL model, however, there is a large backlog of active flows that needs to be cleared before the offered load returns at its pre-congestion level. As Figure 13 shows, this effect lasts for hundreds of seconds (this depends of course on the duration and magnitude of the congestion event and on the TCP offered load before congestion). Figure 13 further shows the queueing delay in the bottleneck with each model. Notice that even though the congestion event ends at t=275sec, the queue remains almost full for hundreds of seconds with the TCP-OL model.
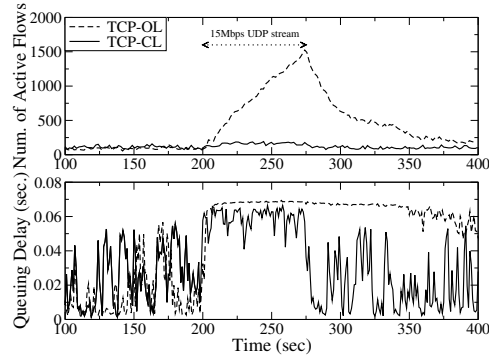
**Figure 13. The time series of the number of active flows and of the queueing delay with the TCP-OL and TCP-CL models when a congestion event is caused by a CBR UDP stream.**

Even if the long-term offered load at a link is below the capacity, there can be overload events that last for a few tens of seconds. The important lesson from the previous discussion is that *during such events an open-loop traffic aggregate is effectively congestion unresponsive despite the fact that it consists of TCP flows*. Further, the consequences of an externally imposed congestion event (say a large UDP stream or a DOS attack) can last for much longer than the duration of the event itself, if the traffic is open-loop.

## 11  Discussion

In this paper, we examined two basic models of non-persistent flow arrivals, and explained how they lead to different traffic characteristics, in terms of offered load, variability in different timescales, queueing performance, number of active flows, congestion responsiveness and elasticity. In the following, we discuss some more implications of this work in other areas of networking research and practice.

**AQM and network stability:** Active queue management (AQM) mechanisms, such as RED, REM, PI controllers, etc., have been proposed as a way to stabilize congestion control. It is important to note that such studies assume persistent TCP connections. With that model, AQM

mechanisms can control the queue length and the bottleneck link utilization. The effectiveness of AQM mechanisms with non-persistent traffic, however, is much less understood. The offered load of TCP-OL traffic does not depend on network state. AQM mechanisms cannot regulate such an aggregate, and they are unable to avoid persistent overload if the offered load exceeds the capacity.

**Is admission control necessary?** Several researchers advocate the use of admission control as the only way to regulate the offered load and avoid congestion collapse. We agree, if the traffic is mostly OL. Without admission control, the only way to avoid congestion collapse is to expect that users will be impatient and abandon slow ongoing transfers. Admission control can limit the number of active sessions or flows in the network. Admission control may not be necessary, however, if most of the traffic follows the CL model.

**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. It should be clear however, that even if a traffic aggregate consists entirely of TCP flows, it can still cause congestion collapse or persistent overload if it is OL. The same is obviously true for TCP-friendly traffic. Therefore, the use of TCP-friendly congestion control is not sufficient to guarantee stability.

**Traffic engineering and network provisioning:** Traffic engineering, as well as other provisioning mechanisms, require an estimate for the offered load between any ingress/egress pair. 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-CL traffic. The offered load from such aggregates depends on the RTT and loss rate in the underlying path. On the other hand, the offered load from TCP-OL traffic does not depend on the underlying path (ignoring retransmissions), making such traffic consistent with common assumptions in traffic engineering.

**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 is congested, or do not keep more than a certain number of sessions active. 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. Effectively, such applications generate TCP-OL traffic.

# References

[1] A. Bondi and W. Whitt. The Influence of Service-Time Variability in a Closed Network of Queues. *Performance Evaluation*, 6:219–234, 1986.

[2] A. Berger and Y. Kogan. Dimensioning Bandwidth for Elastic Traffic in High-Speed Data Networks. *IEEE/ACM Trans. on Networking*, 8(5):643–654, 2000.

[3] T. Bonald, P. Olivier, and J. Roberts. Dimensioning High Speed IP Access Networks. In *18th ITC*, 2003.

[4] G. de Veciana, T. Lee, and T. Konstantopoulos. Stabily and Performance Analysis of Networks Supporting Services. *IEEE/ACM Trans. on Networking*, 9(1), 2001.

[5] A. Dhamdhere and C. Dovrolis. Buffer Sizing for Congested Internet Links. In *IEEE Infocom*, 2005.

[6] A. Feldmann. Characteristics of TCP Connection Arrivals. Technical Report, AT&T Labs Research, 1998.

[7] S. B. Fredj, T. Bonald, A. Proutiere, G. Regnie, and J. W. Roberts. Statistical Bandwidth Sharing: A Study of Congestion at Flow Level. In *ACM Sigcomm*, 2001.

[8] 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*, 1997.

[9] M. Jain and C. Dovrolis. End-to-end Estimation of the Available Bandwidth Variation Range. In *ACM Sigmetrics*, 2005.

[10] 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.

[11] F. P. Kelly, A. Maulloo, and D. Tan. Rate Control in Communication Networks: Shadow Prices, Proportional Fairness and Stability. *Journal of the Operational Research Society*, 49:237–252, 1998.

[12] A. A. Kherani and A. Kumar. Stochastic Models for Throughput Analysis of Randomly Arriving Elastic Flows in the Internet. In *IEEE Infocom*, 2002.

[13] L. Kleinrock. Time-shared Systems: A Theoretical Treatment. *Journal of the ACM*, 14(2):242–261, 1967.

[14] S. Kunniyur and R. Srikant. Stable, Scalable, Fair Congestion Control and AQM Schemes that Achieve High Utilization in the Internet. *IEEE Trans. on Automatic Control*, 49:2024–2029, 2004.

[15] S. H. Low, F. Paganini, J. Wang, S. Adlakha, and J. Doyle. Dynamics of TCP/RED and a Scalable Control. In *IEEE Infocom*, 2002.

[16] R. S. Prasad and C. Dovrolis. Measuring the Congestion Responsiveness of Internet Traffic. In *PAM*, 2007.

[17] J. Roberts. A Survey on Statistical Bandwidth Sharing. *Computer Networks*, 45:319–332, 2004.

[18] B. Schroeder, A. Wierman, and M. Harchol-Balter. Closed Versus Open: A Cautionary Tale. In *NSDI*, 2006.

[19] X. Tian, J. Wu, and C. Ji. A Unified View of Heterogeneous Network Traffic: Impact of Network Load. In *IEEE Infocom*, 2002.

[20] Y. Zhang, S. Kang, and D. Loguinov. Delayed Stability and Performance of Distributed Congestion Control . In *ACM Sigcomm*, 2004.