On the Effect of Traffic Self-Similarity on Network Performance

Kihong Park* Gitae Kim† Mark Crovella‡

CSD-TR 97-024
Dept. of Computer Sciences
Purdue University

Abstract

Recent measurements of network traffic have shown that self-similarity is an ubiquitous phenomenon present in both local area and wide area networks. In previous work, we have shown a simple, robust application layer causal mechanism of traffic self-similarity, namely, the transfer of objects or files in a generic network system where the file size distributions are heavy-tailed. In this paper, we study the effect of scale-invariant burstiness on network performance when the protocol stack and the interaction of traffic sources sharing bounded network resources are incorporated.

First, we present a framework for understanding the two principal performance implications of self-similar traffic—concentrated periods of congestion and amplified packet loss/queueing delay. Using performance measurements, we show that the former is an intrinsic performance effect of self-similar bursty traffic that cannot be easily curtailed. The latter, however, is a conditional performance effect whose manifestation depends on the relative magnitude of network resources vis-à-vis traffic burstiness. We use the notion of asymmetry effect to characterize when self-similar burstiness leads to amplified packet loss and queueing delay.

Second, we present extensive performance measurements that show the performance impact of self-similar burstiness, network resources (bandwidth/buffer capacity), and congestion control on network performance and the resultant traffic characteristics. We show that burstiness—when it does adversely impact—causes a linear degradation in packet loss rate while leading to a superlinear degradation of queueing delay. We show that there exists a sensitive quality of service (QoS) trade-off relation between packet loss and queueing delay under self-similar traffic conditions which aggrevates the problem of efficiently provisioning services to QoS-sensitive applications.

Lastly, we study the influence of congestion control in modulating performance. We use flow-controlled UDP transport to discern the performance impact and traffic shaping effect of congestion control, and we give a comparative study of TCP Tahoe, Reno, and Vegas with respect to their performance characteristics when subject to self-similar bursty traffic.

*Dept. of Computer Sciences, Purdue University, West Lafayette, IN 47907; park@cs.purdue.edu. Supported in part by NSF grant NCR-9714707.

†Computer Science Department, Boston University, Boston, MA 02215; kgtjan@cs.bu.edu. Supported in part by NSF grant CCR-9706685.

‡Computer Science Department, Boston University, Boston, MA 02215; crovella@cs.bu.edu. Supported in part by NSF grant CCR-9501822.

1 Introduction

Recent measurements of local and wide area network traffic have shown that network traffic is bursty on a wide range of time scales, a much wider range than is captured by traditional traffic models [20]. Scale-invariant burstiness[1] can be described using the notions of self-similarity and long-range dependence (LRD), and much work has been done in the last few years in verifying its ubiquitous presence in network environments and investigating the causes of traffic self-similarity and its consequences [1, 2, 8, 11, 15, 21, 22, 24, 26, 32, 37].

A number of papers have studied the implications of long-range dependence for traffic modeling and network performance evaluation purposes. The research avenues may be broadly classified into two categories, one dealing primarily with traffic characterization and modeling issues, the other concentrating on performance evaluation. In the first category of work [8, 11, 15, 21, 32, 37], traffic traces from physical network measurements are used to identify the presence of self-similar burstiness, and models are constructed capable of generating synthetic traffic with matching characteristics. These works have shown that long-range dependence is an ubiquitous phenomenon occurring in diverse network environments, and even individual traffic sources such as variable bit-rate (VBR) video have been shown to possess correlation structure that can be modeled—along with several other models [11, 15, 13, 19, 34]—using self-similar traffic models. This also represents an instance of a real-world traffic source which, on its own, exhibits self-similar burstiness as a source-intrinsic property.

In the second category are papers that have evaluated the effect of self-similar traffic in the context of idealized or simplified networks [1, 2, 18, 22, 24]. These works have shown that long-range dependence has an adverse effect on network performance, the principal result being that queue length distribution, under self-similar traffic conditions, decays more slowly when compared to traditional Markovian sources (e.g., Poisson) which possess exponentially decaying tails. In a somewhat different approach to the performance question [12, 34], the influence of time scale is taken into account when discerning the impact of long-range dependence on queueing. It is shown that for small buffer sizes and short time scales, long-range dependence has only a marginal influence on queueing behavior with short-range correlations being the dominant factor. This is not too surprising since under such conditions, the variability associated with short-range correlations is sufficient to trigger significant buffer overflow and queue length saturation without added long-range correlation structure.

The effect of self-similar burstiness on the performance of resource-bounded networks incorporating the influence of protocol stack interactions has only recently begun to receive attention [14, 26, 27, 31, 29]. This question is practically relevant because, although self-similar bursti-

[1] We will use scale-invariant burstiness, self-similar burstiness, and self-similarity interchangeably.

ness is expected to adversely affect network performance, knowledge of the degree of performance degradation allows us to assess its relative importance as a performance problem and possibly control its consequences. The latter aspect is studied in [29]. In this paper we investigate the effects of self-similar traffic on network performance when the functionality of the protocol stack and the interaction of traffic sources sharing bounded network resources is explicitly incorporated. We employ application-, transport- and network-layer simulations to study the impact of self-similar burstiness under controlled environments. Setting up such a test environment where the protocol stack's influence can be accurately discerned is—by itself—nontrivial due to the fact that, in general, the characteristics of the observed traffic is itself affected by the actions of the control algorithms residing in the protocol stack.

Our results are facilitated by a high-level *structural* approach to self-similar traffic generation [26] that allows the degree of traffic self-similarity to be intrinsically and intimately controlled by a mechanism acting at the application layer. In a nutshell, this process involves the generation of object or file transfers across a network whose size distribution is heavy-tailed (formal definition given later). In other words, objects are communicated in a generic client/server network environment where files of very large sizes are transferred with nonnegligible probability. Previously we have shown that this high-level structural mechanism suffices to generate self-similar traffic, and the degree of self-similar burstiness observed at the link is directly related to the heavy-tailedness of the file size distribution [26]. An important advantage of this scheme over traditional queueing modeling set-ups is that self-similar burstiness can be induced and controlled at the application layer while being oblivious to the ongoings of the rest of the system including the lower layers of the protocol stack. This, in turn, allows us to evaluate the modulating influence of the protocol stack, in particular, the transport layer, in such a way where its effects can be effectively discerned.

We emphasize that in our system self-similar burstiness of network traffic is a *collective*, *induced* phenomenon where not a single source engages in the pumping of self-similar traffic into the network. For example, this could be easily done by having a source transmit real-time MPEG VBR video across the network. Self-similar bursty network traffic comes about as a consequence of one of the most innocuous network activities—the transfer of files in a networked client/server system, albeit, by a number of *concurrent* connections/sessions. A discussion of simulation related issues in the context of evaluating network performance under self-similar traffic conditions can be found in [9, 31].

Our work is related to the ON/OFF traffic model [37] where it is shown that the superposition of a large number of independent 0/1 renewal reward processes with heavy-tailed ON/OFF durations results in fractional Gaussian noise when suitably normalized. The main drawback of this simple, elegant characterization—which has its roots in some early work of Mandelbrot [23]—is the *independence* assumption of the traffic streams which ignores interactions and dependencies

arising in real networks. The latter leads to coupling of traffic sources stemming from the sharing of bounded network resources by multiple traffic streams which can introduce nonlinearities into the system. This paper builds on our previous work [26, 27] which concentrated on finding a structural explanation of self-similar network traffic given its ubiquity. We extend our work to performance evaluation related issues, complementing existing analytical work on queueing with self-similar input, by taking into account the impact of network resources and the influence exerted by the protocol stack, and providing a pragmatic, physical basis for understanding performance results under self-similar traffic conditions. Our main findings can be summarized as follows.

First, we present a framework for understanding the two principal performance implications of self-similar traffic—*concentrated periods of congestion* and *amplified packet loss/queueing delay*. Using performance measurements, we show that the former—whose flip side is the presence of concentrated periods of light network load—is an intrinsic performance effect of self-similar bursty traffic that cannot be easily curtailed by traffic controls. The presence of "peaks and valleys" at a range of time scales makes it difficult to provision stringent QoS while at the same time achieving high network utilization. The latter effect (amplified packet loss and queueing delay), however, is a conditional performance effect whose manifestation and severity depends on the relative magnitude of network resources vis-à-vis traffic burstiness. We use the notion of asymmetry effect to characterize when self-similar burstiness leads to amplified packet loss and queueing delay.

Second, we sketch a comprehensive anatomy of self-similar bursty traffic and its performance impact by giving extensive performance measurements that show the influence of scale-invariant burstiness, network resources—bandwidth and buffer capacity—and congestion control on network performance and the resultant traffic characteristics. We show that self-similar burstiness—when it does adversely impact—causes a *linear* degradation in packet loss rate while leading to a *superlinear* degradation of queueing delay. We show that a similar result holds with respect to bandwidth and buffer capacity where increasing the latter leads to a proportional increase in buffer occupancy over a wide range of buffer sizes thus resulting in significant queueing delays. This makes the apportioning of bandwidth a more effective resource allocation strategy than increasing buffer capacity. We show that there exists a sensitive quality of service (QoS) trade-off relation between packet loss and queueing delay under self-similar traffic conditions which aggrevates the problem of efficiently provisioning services to QoS-sensitive applications. This implies that multimedia applications—a traffic source with voluminous data rate requirements which contribute to promote an environment conducive to traffic self-similarity, both through source-instrinsicness (e.g., VBR video), and through size—are also the traffic sources that suffer most from its consequences due to their sensitive QoS requirements.

Lastly, we study the influence of congestion control in modulating performance. We show that congestion control is effective in the sense that it leads to a graceful decline in performance through a

traffic shaping effect—i.e., stretching in time—when subject to self-similar bursty traffic conditions. We use open-loop flow-controlled UDP transport parameterized by an interpacket spacing parameter τ that determines how fast packets are submitted to the network, and through which the degree of aggressiveness can be adjusted, to discern performance. The open-loop congestion control protocol conforms to one of the central premises underlying the ON/OFF model—independence—and it allows us to discern not only the performance impact of self-similar bursty upstream traffic but also the resultant traffic shaping effect on the *downstream traffic*—i.e., throughput—which subsequently acts as part of upstream traffic down-the-line at other multiplexing points. We conclude with a comparative study of TCP Tahoe, Reno, and Vegas with respect to their performance characteristics when subject to "self-similar traffic conditions." We emphasize that the latter qualification—in our context—just means that TCP is used for the *bulk* transfer of individual files whose active durations make up a session. Our measurements show that the sophistication of control implied in transiting from Tahoe to Reno to Vegas[2] is preserved in the sense of leading to improved performance even under self-similar traffic conditions. The performance gain from Tahoe to Reno is relatively minor while the performance jump from TCP Reno to Vegas is more pronounced consistent with quantitative measurements reported in [7, 3].

The rest of the paper is organized as follows. In the next section, we describe the network model including the set-up of our simulation environment. This is followed by a description of the self-similar traffic generation process and a dual subdivision of the protocol stack which are key components to the experimental set-up and the facilitation of subsequent performance results. The next section gives a roadmap and establishes a framework for understanding the two principal performance effects of self-similar bursty traffic—concentrated periods of congestion and amplified packet loss/queueing delay. The next three sections give a comprehensive account of the performance impact of self-similar burstiness, first, with respect to the degree of burstiness, second, with respect to network resources, and third, with respect to control actions in the transport layer of the protocol stack. We conclude with a discussion of our results and future work.

2 Network Model and Simulation Set-Up

2.1 Network Model

The network is given by a directed graph $G = (V, E)$ consisting of n nodes and m links. Each output link has a buffer, bandwidth, and latency associated with it. $v_i \in V$ is a *server node* if it has a probability density function $p_i(X)$ where $X \geq 0$ is a random variable denoting file or message size. We will call $p_i(X)$ the *file size distribution* of server v_i. v_i is a *client node*—it may at the same

[2]TCP Tahoe and Reno are virtually identical except for the fast recovery mechanism which is absent in Tahoe [17]. TCP Vegas [7] implements a form of proactive congestion control which gives it a different flavour from Reno.

time be also a server node—if it has two probability density functions $h_i(X)$, $d_i(Y)$, $X \in \{1, \ldots, n\}$, $Y \in \mathbf{R}_+$, where h_i is used to select a server, and d_i is the *idle time distribution* which is used in determining the time of the next request. In the context of reliable communication, if T_k is the time at which the k'th request by client v_i was reliably serviced, the next request is scheduled at time $T_k + Y$ where Y has distribution d_i. In unreliable (UDP) communication, this causal requirement is waived. A 2-server, 32-client network configuration with a bottleneck link between gateways G_1 and G_2 is shown in Figure 1.

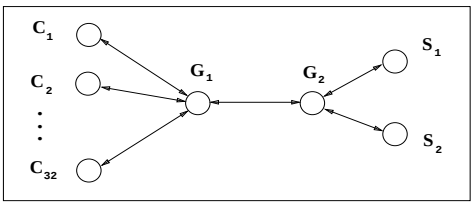

Figure 1: Network configuration.

A file is completely determined by its size X. Each file is split into $q = \lceil X/M \rceil$ packets where M is the maximum segment size. The segments are routed through a packet-switched internetwork in which packets are dropped at bottleneck nodes in case of buffer overflow. The routers implement FIFO packet scheduling. Each client alternates between independently placing file transfer requests to servers, receiving data from the server, and spending some period of time idle before starting the whole process again. $X_i \sim p_i(X)$, $Y_i \sim d_i(X)$, $i = 1, \ldots, n$, are i.i.d., and we require that $p_i(X)$ be heavy-tailed (formal definition given later). The idle time distribution, on the other hand, is allowed to be non-heavy-tailed, e.g., exponential.

2.2 Simulation Set-Up

We use the LBNL Network Simulator, *ns*, as the basis of our simulation environment [10]. *ns* is an event-driven simulator derived from Keshav's REAL network simulator supporting several flavours of TCP (TCP Tahoe and TCP Reno) [17] and router scheduling algorithms. Although not production TCP code, we have found *ns*'s emulation of TCP satisfactory for the purposes of studying the effect of congestion control [17, 30] as well as emulating reliable transport. A test suite description can be found in Floyd [10].

We have modified *ns* in order to model our interactive client/server environment. This entailed extending the one-way data flow restriction of a single *ns* TCP session to full-duplex, and implementing the client/server nodes as separate application layer agents. A UDP-based unreliable transport protocol was added to the existing protocol suite, and an open-loop end-to-end flow control was implemented on top of it. The arrival rate was adjusted by a parameter $\tau > 0$ which determined the spacing between successive packets. Thus the smaller τ, the higher the arrival rate.

We also implemented TCP Vegas [7] as a module of *ns* and it was used in performance comparisons between the three versions of TCP under self-similar traffic conditions. In addition to the tracing functions that native *ns* provides, we added utilities for monitoring an expanded set of network statistics including provisions for detecting retransmission, reliable throughput, and computing file transmission completion times.

Our simulation results were obtained from several hundred runs of *ns*. Each run executed for 10000 simulated seconds, logging traffic at 10 millisecond granularity. The result in each case was a time series of one million data points. Although most of the runs reported here were done with a 2-server/32-client bottleneck configuration (cf. Figure 1), other configurations were tested including performance runs with the number of clients increasing up to 132. The bottleneck link was varied from 1.5Mbps up to OC-3 levels, and buffer sizes were varied in the range 1kB–128kB. Non-bottleneck links were set at 10Mbps and the latency of each link was set to 15ms. The maximum segment size was fixed at 1kB for most runs. For any reasonable assignment to bandwidth, buffer size, mean file request size, and other system parameters, we found that by either adjusting the number of clients or the mean of the idle time distribution d_i appropriately, any intended level of contention could be achieved. This enabled us to carry out performance evaluations by varying the two main network resources—bottleneck buffer capacity and bandwidth—and the degree of self-similar burstiness.

3 Application Layer Causality of Self-Similar Link Traffic

3.1 Dual Subdivision of Protocol Stack

One of the goals of this paper is to evaluate the role of the protocol stack, in particular, the application layer's structural propensity to induce self-similar network traffic mediated by the transport layer and the latter's influence on network performance. To this end, we take a dualistic view of the protocol stack by partitioning its functionality into application layer and transport layer mechanisms, collapsing the layers below into a single data link abstraction, referred to as the link layer. We also ignore finer subdivisions of the layers above the transport layer modeling it as a single entity.

The application layer is completely characterized by the probability distributions described in Section 2.2. File size distribution is the dominant control variable in structurally inducing self-similar network traffic with idle time distribution playing a secondary role. In [26], we have shown that independent of the idle time distribution, file size distribution—by itself—suffices to produce self-similar link traffic if it is heavy-tailed. We use the Pareto distribution (definition given in Section 3.2) with shape parameter α for this purpose adjusting the location parameter k to keep the average file size—sampled over 10000 seconds at 10ms granularity—invariant across α

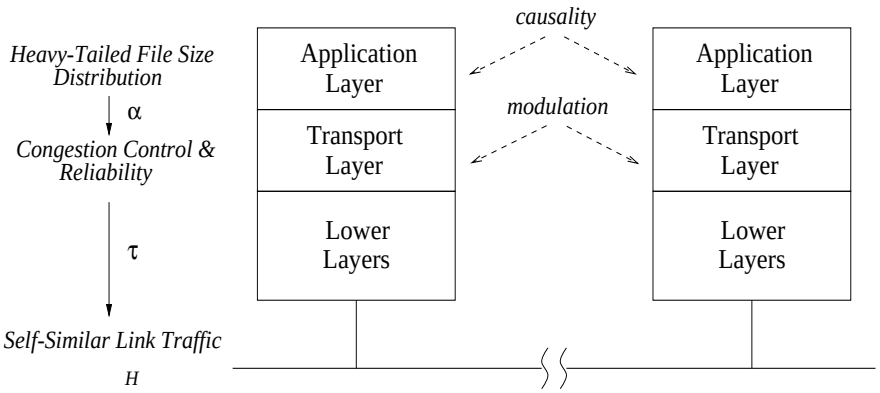

Figure 2: Transformation of the heavy-tailedness of file size distribution property at the application layer via the action of the transport/network layer into its manifestation as self-similar link traffic at the link layer.

values in the range 1.05, 1.15, 1.25, 1.35, 1.65, 1.95. We have used other distributions to generate communication demand at the application layer, and in the case of the (negative) exponential distribution, performance results were comparable to that of the Pareto distribution with $\alpha = 1.95$. That is, the mathematical property of heavy-tailedness of Pareto distribution with $\alpha = 1.95$ was found to be of little practical significance.

The transport layer is composed of a number of protocols implementing end-to-end traffic management tasks including reliable/unreliable communication with various forms of congestion control. The most important aspect of the dual protocol stack subdivision is the positioning of the "causal seed" of self-similar burstiness at its highest possible point of affection—the application layer. File transfer requests are generated at the top, and nowhere in the system do we introduce traffic sources that single-handedly pump self-similar traffic into the network. That is, self-similar traffic burstiness is an *induced* phenomenon, a consequence of multiplexing traffic streams generated by the transfer of heavy-tailed objects in a generic networked system. The isolation of self-similar traffic causality in the application layer allows us to evaluate the influence and effectiveness of protocols acting at lower layers in the protocol stack on network performance.

How well the transport layer modulates burstiness and network performance is dependent upon the availability of network resources and the specific congestion control protocols used. We model network resources via two variables: buffer capacity and link bandwidth. Buffer capacity comes into play when multiple traffic streams are multiplexed onto the same output port of a gateway, its size influencing both packet loss rate and queueing delay. The characteristics of network traffic at the link layer are captured by the Hurst parameter H (defined later) which measures the degree of long-range dependence of a time series. The overall structure of the set-up is depicted in Figure 2.

3.2 Heavy-Tailed File Size Distribution and Self-Similar Link Traffic

3.2.1 Heavy-Tailed Distribution

An important characteristic of the aforementioned mechanism for inducing self-similar link traffic from the application layer is the assumption that the sizes of objects or files being transferred are drawn from a heavy-tailed distribution. A distribution is *heavy-tailed* if its tail asymptotically follows a power law. That is,

$$\Pr\{ X > x \} \sim x^{-\alpha} \quad \text{as} \quad x \to \infty$$

where $0 < \alpha < 2$. One of the simplest heavy-tailed distributions is the *Pareto* distribution whose probability density function is given by

$$p(x) = \alpha k^{\alpha} x^{-\alpha-1}$$

where α is the *shape parameter*, $k > 0$ is the *location parameter*, and $x \geq k$. The distribution function has the form

$$F(x) = \Pr\{ X \leq x \} = 1 - (k/x)^{\alpha}.$$

Heavy-tailed distributions have a number of properties that are qualitatively different from distributions more commonly encountered in networking research, in particular, the exponential distribution. If $\alpha \leq 2$, the distribution has infinite variance, and if $\alpha \leq 1$, the distribution has also infinite mean. Thus, as α decreases, a large portion of the probability mass resides in the tail of the distribution. In practical terms—and relating to our network model—a random variable that follows a heavy-tailed distribution can give rise to extremely large file size requests with nonnegligible probability.

3.2.2 Self-Similar Link Traffic

In this section we show that the generic client/server network set-up described earlier is able to generate self-similar network traffic. Figure 3 shows graphically[3] that our set-up is able to induce self-similar link traffic, the degree of scale-invariant burstiness being determined by the α parameter of the Pareto file size distribution. The plots show the time series of network traffic measured at the output port of a bottleneck link (i.e., from gateway G_2 to G_1 in Figure 1). The *downstream traffic* is measured in bytes per unit time where the aggregation level or time scale varies over five orders of magnitude from 10ms, 100ms, 1sec, 10sec, to 100sec. Only the top three aggregation levels are shown in Figure 3. For α close to 2 (second column from right), we observe a smoothing

[3]Self-similar phenomena are well-suited to visual illustration. This augments the statistical estimations of long-range dependence given later in the paper.

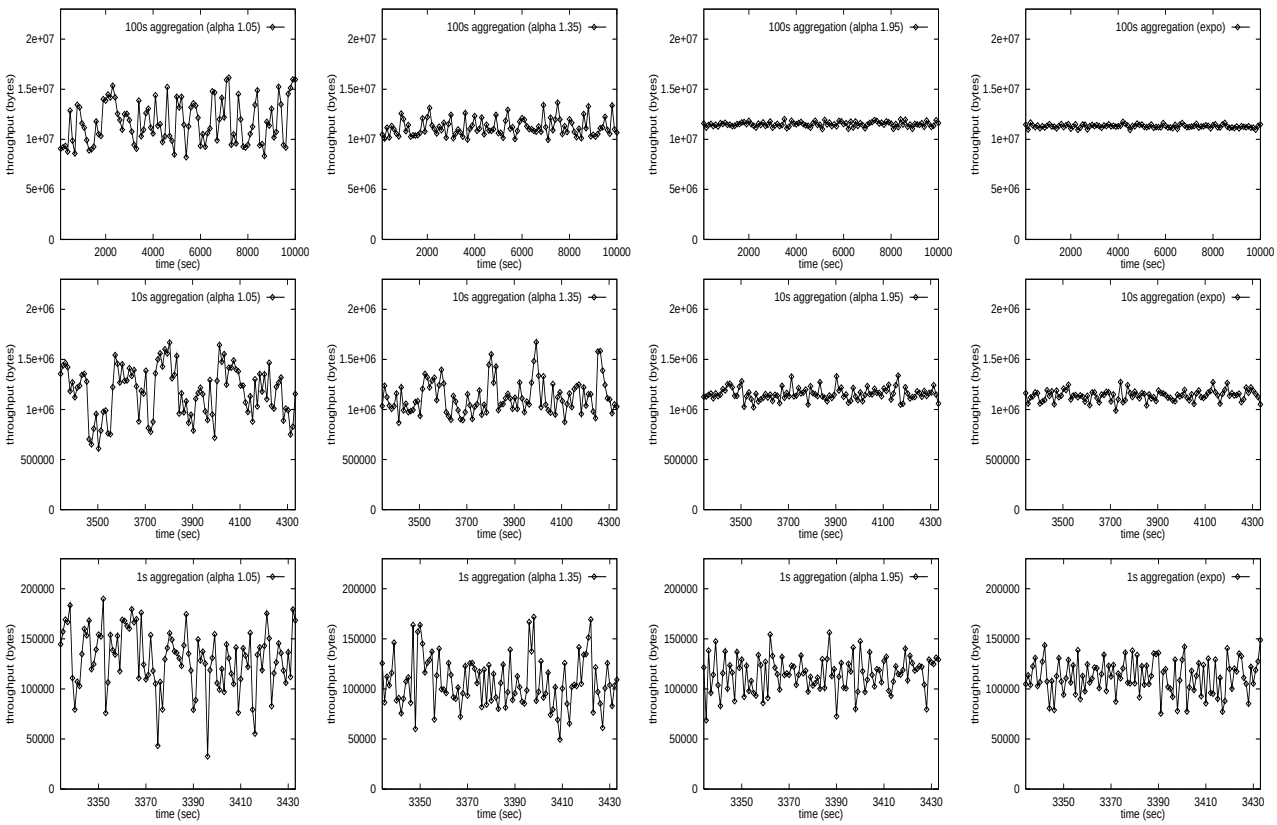

Figure 3: TCP run. Throughput as a function of file size distribution and three aggregation levels. File size distributions constitute Pareto with $\alpha = 1.05, 1.35, 1.95,$ and exponential.

effect as the aggregation level is increased (bottom row to top row), indicating a weak dependency structure in the underlying time series. As α approaches 1, however, burstiness is preserved even at large time scales indicating that the 10ms time series possesses long-range dependence. The first two aggregation levels (10ms, 100ms) are omitted because differences between the aggregated time series manifest themselves from the 1 second time scale onwards. The rightmost column depicts traces obtained by employing an exponential file size distribution at the application layer with the sample mean normalized so as to equal that of the Pareto distributions. It is striking that the aggregated time series of exponential distribution and Pareto with $\alpha = 1.95$ are visually indistinguishable. This is supported by statistical measurements including H estimations.

Figure 4 shows the packet drop traces corresponding to the throughput traces of Figure 3. We observe that the packet drop process also exhibits self-similar burstiness, being most pronounced at the 100s aggregation level. Statistically, differences are observable starting at the 1s aggregation level and the packet drop pattern follows the throughput traffic trace pattern closely (compare, e.g., the top-left plots of Figures 3 and 4). These observations are supported by statistical measurements

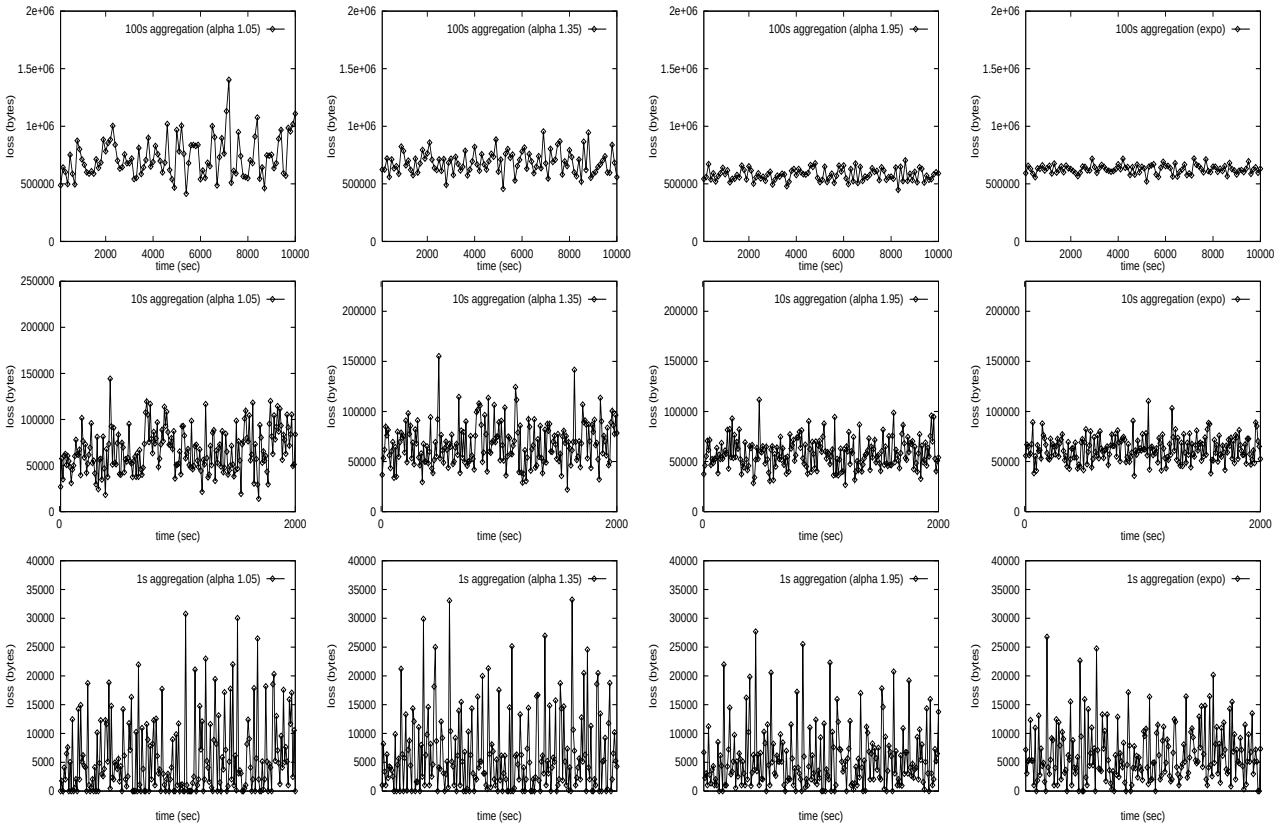

Figure 4: TCP run: Packet drop (in bytes) as a function of $\alpha = 1.05, 1.35, 1.95,$ and exponential.

which are further elaborated in Section 4.1.

Although Figure 3 gives clear visual evidence of the difference in self-similar burstiness of link traffic as a function of the control variable α, a quantitative measure of long-range dependence is obtained via the Hurst parameter H which expresses the rate of decay of a time series' autocorrelation function. A time series with *long-range dependence* has an autocorrelation function of the form

$$r(k) \sim k^{-\beta} \quad \text{as } k \to \infty$$

where $0 < \beta < 1$. Hence, when compared to the exponential decay exhibited by traditional traffic models, the autocorrelation function of a long-range dependent process decays according to a power-law. The *Hurst parameter* is related to β via

$$H = 1 - \beta/2.$$

Hence, for self-similar time series, $1/2 < H < 1$. As $H \to 1$, the degree of long-range dependence increases. A test for LRD of a time series can be reduced to the question of determining whether H significantly deviates from $1/2$.

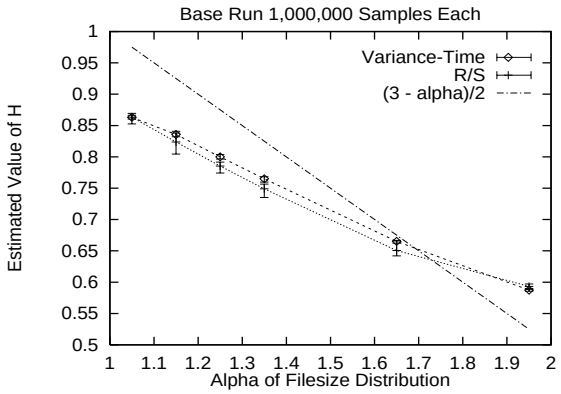

Figure 5: Hurst parameter estimates (R/S and V-T) for the throughput trace for α varying from 1.05 to 1.95.

Figure 5 (left) shows H-estimates based on Variance-Time (V-T) and R/S methods[4] for a baseline network configuration with TCP Reno acting at the transport layer, link speed of 1.544Mbps, 6kB buffer size, 2 servers and 32 clients. The plot shows H as a function of the Pareto distribution shape parameter for $\alpha = 1.05, 1.15, 1.25, 1.35, 1.65$ and 1.95. Each point on the plot is the average of 3 estimates based on different random seeds, and the error bars show the spread of the maximum and minimum estimates. This particular configuration results in a packet drop rate of 4% for the most bursty case (i.e., $\alpha = 1.05$). The plot shows that even when resources are bounded leading to the coupling among the 32 traffic streams contending for shared network resources, the Hurst parameter varies with file size distribution in a linear manner. That is, the heavy-tailed property of the file size distribution as captured by α directly determines self-similar burstiness of the downstream traffic as measured by H. The

$$H = (3 - \alpha)/2$$

relation shows the values of H that would be predicted by the ON/OFF model [37] in the idealized case of fractional Brownian motion when the traffic sources are independent with heavy-tailed ON/OFF periods and 0/1 amplitudes. Although their overall trends are similar (nearly coinciding at $\alpha = 1.65$), the slope of the simulated system with resource limitations and transport layer interactions is slightly less than -1, with an offset below the idealized line for α close to 1 and above the line for α close to 2.

[4]Variance-time (V-T) and R/S are standard methods [5, 21] for estimating the Hurst parameter H.

4 Performance Evaluation

In this section we study the performance impact of scale-invariant bursty traffic based on performance measurements. Section 4.1 lays the roadmap and establishes the framework for understanding the comprehensive measurement results shown in the later sections.

We will show that there are two main features to the performance question under self-similar traffic conditions: concentrated periods of congestion and amplified packet drops/queueing delay. The former is an inherent feature of network performance under self-similar bursty traffic whose manifestation cannot be easily curtailed. The latter depends on the relative magnitude of resources vis-à-vis total traffic demand, and long-range dependence need not—although in many instances it can—lead to increased packet loss rate and queueing delay.

4.1 Effect of Self-Similar Burstiness

4.1.1 Properties of Packet Drop Process

In Section 3 we showed that the throughput process exhibits self-similarity and long-range dependence as captured by the Hurst parameter H. Here we will show that the packet drop process also exhibits self-similar burstiness. First, we show that packet drops are positively correlated to the variation of the upstream process, and since the latter exhibits scale-invariant burstiness, the same holds—with some modulation—for the packet drop process.

Figure 6 plots the *correlation coefficient* of the throughput and packet drop time series aggregated at time scales ℓ = 10ms, 100ms, 1s, 10s, 100s:

$$\mathrm{corr}(X^\ell, Y^\ell) = \frac{\mathbf{E}\{(X^\ell - \bar{X^\ell})(Y^\ell - \bar{Y^\ell})\}}{\sigma_{X^\ell}\sigma_{Y^\ell}}.$$

Here X^ℓ, Y^ℓ are the aggregated throughput and packet drop time series at aggregation level ℓ, and σ_{X^ℓ}, σ_{Y^ℓ} are their standard deviations. Figure 6 (left) shows the correlation structure between packet drops and upstream traffic (i.e., offered load) and packet drops and downstream traffic (i.e., throughput) for flow-controlled UDP runs with interpacket spacing τ = 10ms and $\alpha = 1.05, 1.95$. In the case of $\alpha = 1.05$, the correlation coefficient increases with aggregation level reaching a value of 0.94 at 100s aggregation. For $\alpha = 1.95$, correlation levels off after the 1s time scale reaching a value of 0.84 at the 100s aggregation. Thus packet drop and throughput are highly correlated, especially so at the higher aggregation levels.

Figure 6 (right) shows the correlation structure between packet drops and downstream/upstream traffic for the TCP baseline runs. The correlation of the downstream traffic with packet drops is weaker than that of the upstream traffic with packet drops which we also observe for the flow-controlled UDP runs. The correlation of the $\alpha = 1.05$ trace with packet drops overall dominates

that of the $\alpha = 1.95$ trace, especially after the 1s aggregation level. The correlation coefficient for the $\alpha = 1.95$ traces—UDP and TCP—level off after the 1s aggregation mark. The leveling off effect of the correlation coefficient after the 1s time scale is, in part, due to the flattening of the traffic curves at higher time scales for the $\alpha = 1.95$ traces stemming from the absence of long-range dependence.

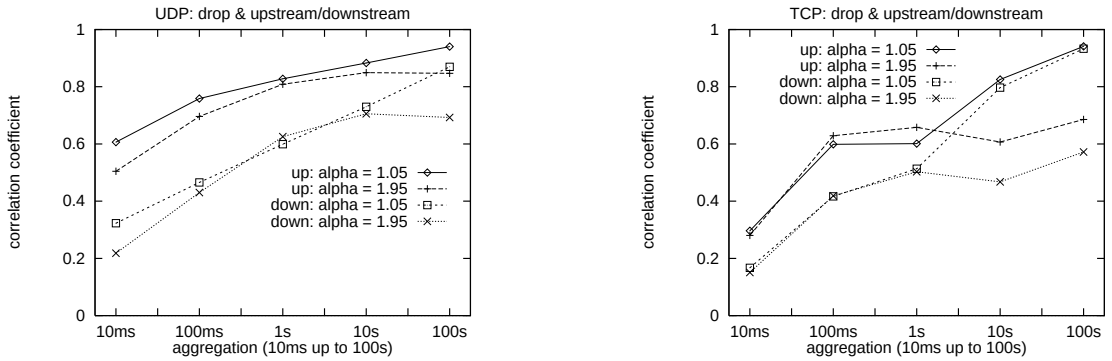

Figure 6: Left: Correlation coefficient of flow-controlled UDP upstream/downstream traffic with packet drop time series aggregated at time scales 10ms, 100ms, 1s, 10s, 100s. Right: Correlation coefficient of TCP downstream and upstream traffic with packet drops.

Figure 7 shows the traces for throughput and packet drops for the base TCP runs at the 100s aggregation level. The leftmost two plots show the throughput and packet drop traces for $\alpha = 1.05$, respectively. The rightmost two plots show the corresponding traces for $\alpha = 1.95$. We observe a strong positive correlation between the packet drop and throughput time series which indicates that the packet drop process not only follows the throughput process but also exhibits variability at a range of time scales. In particular, we observe that the packet drop process for $\alpha = 1.05$ follows its throughput process more closely than that of $\alpha = 1.95$ commensurate with the correlation results shown in Figure 6.

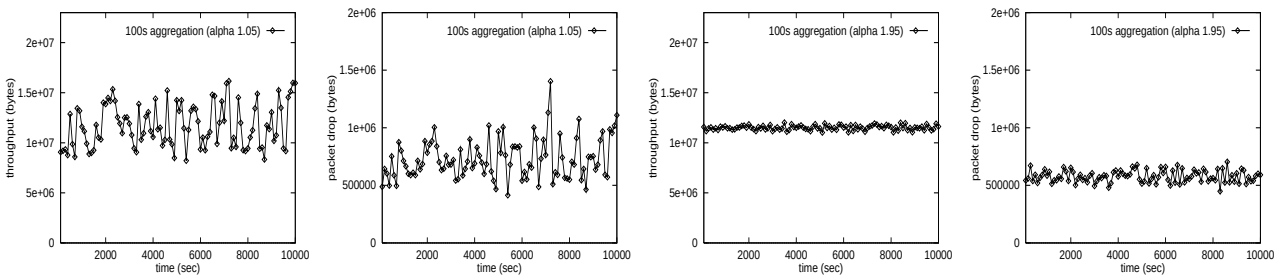

Figure 7: Throughput (left) and packet drop (middle-left) trace of base TCP run at 100s aggregation level for $\alpha = 1.05$. Throughput (middle-right) and packet drop (right) trace for $\alpha = 1.95$.

The long-range dependence structure present in self-similar packet drop traces is further con-

firmed by Hurst parameter estimations shown in Figure 8. The plot shows a linear dependence of H on α for the packet drop process corresponding to TCP baseline runs. They are muted—i.e., shifted downwards vertically vis-à-vis their throughput counterpart (Figure 5)—due to part of the variability structure of the upstream traffic being absorbed by queueing at the router's output buffer. A similar result holds for the flow-controlled UDP runs.

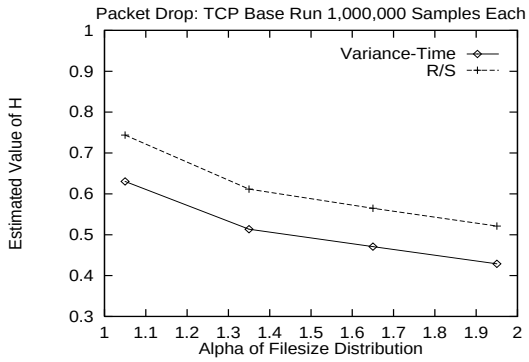

Figure 8: Hurst parameter estimates (R/S and V-T) for packet drop traces corresponding to TCP baseline runs with α varying from 1.05 to 1.95.

4.1.2 Concentrated Periods of Congestion

One of the main performance implications of self-similar network traffic is the presence of concentrated periods of congestion—and conversely mild network contention—at a wide range of time scales. This is a consequence of the scale-invariant nature of traffic burstiness which translates into concentrated periods of packet drops and ensuing congestion. On the flip side, scale-invariant burstiness also implies the existence of concentrated periods of light network load and consequent underutilized resources. These two features conspire to make it inherently difficult to provision services whereby quality of service and network utilization are conjointly maximized. In fact, we will show in Section 4.3 that there is a trade-off relation between packet loss rate and queueing delay whose sensitivity is significantly amplified by the presence of self-similar burstiness.

Figure 9 shows the standard deviation ratios

$$\sigma_{X^{\ell}_{1.05}}/\sigma_{X^{\ell}_{1.95}}, \quad \sigma_{Y^{\ell}_{1.05}}/\sigma_{Y^{\ell}_{1.95}}$$

of throughput and packet drop traces, respectively, at aggregation levels ℓ = 10ms, 100ms, 1s, 10s, 100s for both TCP and flow-controlled UDP measurements. Figure 9 (left) shows that the standard deviation ratio is amplified as time scale increases from 10ms to 100s. That is, the relative burstiness of $\alpha = 1.05$ traces—be they throughput or packet drops—increases as the aggregation level is increased exhibiting differences that exceed by an order of magnitude at the 100s aggregation.

This translates to the manifestation of concentrated periods of congestion or light network load across a wide range of time scales being most pronounced at high aggregation levels.

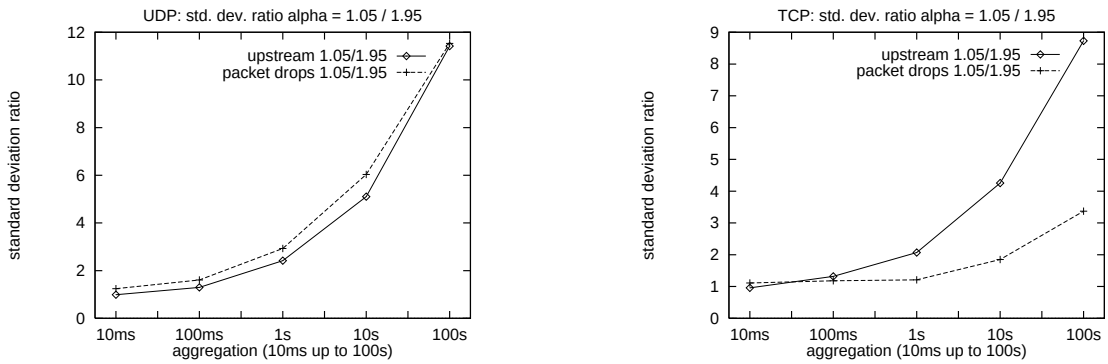

Figure 9: Left: Standard deviation ratios of $\alpha = 1.05/\alpha = 1.95$ for UDP upstream traffic and corresponding packet drop traces aggregated at time scales 10ms, 100ms, 1s, 10s, 100s. Right: Corresponding standard deviation ratios for TCP base run traces.

Figure 9 (right) shows the corresponding standard deviation ratios for TCP runs which exhibit a similar trend except that the standard deviation ratios for packet drops is less pronounced than the flow-controlled UDP counterparts. This is, in part, attributable to TCP's feedback congestion control which at the perceived onset of packet drops or large queueing delays—using timeouts—engages in exponential backoff thus muting subsequent packet drops by the quenching of a source's transmission rate. This, in turn, stretches out the transmission over time via the throttling of sources thereby dampening the adverse effect of packet drops. Feedback congestion control *reduces*—but does not eliminate—the performance impact of concentrated periods of high network load. The difference in variability between $\alpha = 1.05$ and 1.95 packet drop traces starts emerging at the 1s aggregation level, and continues to increase at higher time scales exceeding a factor of 3 at the 100s aggregation level.

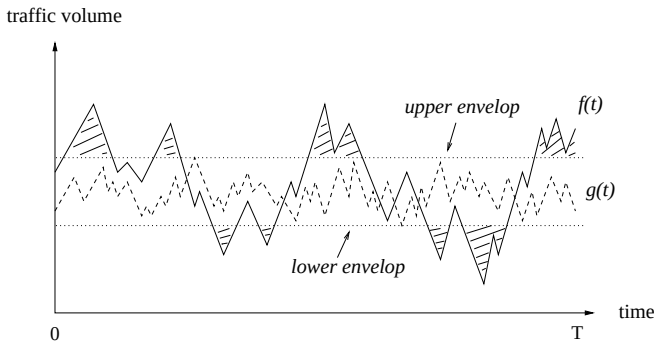

Figure 10: Two functions $f(t)$, $g(t)$, and $g(t)$'s upper and lower envelop curves. The shaded region indicates function values where $f(t) > \max_t g(t)$, or $f(t) < \min_t g(t)$.

Another useful quantity that captures the structure of how the $\alpha = 1.05$ traces are more variable than their $\alpha = 1.95$ counterparts is given by the *envelop deviation ratio* δ^ℓ. Let $\theta_u^\ell = \max_{t\in[0,T]} X_{1.95}^\ell(t)$, and let $\theta_d^\ell = \min_{t\in[0,T]} X_{1.95}^\ell(t)$ where $[0,T]$ is some interval of interest. These are the upper and lower bounds or envelop curves of the traffic series $X_{1.95}^\ell(t)$. Let

$$\mathcal{D}_u^\ell = \{\, t : X_{1.05}^\ell(t) > \theta_u^\ell \,\}, \quad \mathcal{D}_d^\ell = \{\, t : X_{1.05}^\ell(t) < \theta_d^\ell \,\}.$$

Then,

$$\delta^\ell = \frac{|\mathcal{D}_u^\ell| + |\mathcal{D}_d^\ell|}{T}.$$

That is, δ^ℓ is the fraction of time that $X_{1.05}^\ell(t)$ spends exceeding $X_{1.95}^\ell(t)$'s envelop bounds. We also define the *upper* envelop deviation ratio $\delta_u^\ell = |\mathcal{D}_u^\ell|/T$ and the *lower* envelop deviation ratio $\delta_d^\ell = |\mathcal{D}_d^\ell|/T$. Thus δ_u^ℓ, and δ_d^ℓ quantify how frequently $X_{1.05}^\ell(t)$ has crossed the upper and lower bounds of $X_{1.95}^\ell(t)$, respectively. This is graphically depicted in Figure 10 for two functions $f(t)$, $g(t)$.

Figure 11 (left) shows the envelop deviation ratios for the flow-controlled UDP upstream traces aggregated at time scales 10ms, 100ms, 1s, 10s, 100s. We observe a 10% deviation at the 1s aggregation which continues to increase with time scale reaching 84% at the 100s aggregation. That is, above the 1s mark, the $\alpha = 1.05$ upstream traffic exhibits fluctuations that exceed that of the maximal fluctuations of $\alpha = 1.95$ upstream traffic a significant fraction of the time. Figure 11 (left) also shows that concentrated congestion periods (i.e., δ_u^ℓ) are accompanied by concentrated periods of light network load (δ_d^ℓ) where network resources are underutilized—precisely the dilemma faced when provisioning QoS under self-similar traffic conditions. In fact, in the case of upstream traffic, δ_d^ℓ dominates δ_u^ℓ by a factor close to $2:1$. Figure 11 (right) shows the corresponding envelop deviation ratios for the packet drop traces. We observe a similar trend as with the upstream traffic traces, however, with the values dampened via-à-vis the upstream case.

Figure 12 illustrates the presence of "peak and valleys" in $\alpha = 1.05$ traces vis-à-vis $\alpha = 1.95$ traces under the same network condition for flow-controlled UDP runs with interpacket spacing $\tau = 10$ms. The packet loss rate for the former is 11% whereas the packet loss rate for the latter is 7%. The top row shows traces at 100s, 10s, and 1s aggregation levels, for the entire measurement duration (left), first 1000 seconds (middle), and first 100 seconds (right), respectively. The bottom row shows the corresponding traces for the packet drop process. We observe that the $\alpha = 1.05$ traces, relative to the $\alpha = 1.95$ traces, exhibit concentrated periods of high contention/low contention over time scales varying across three orders of magnitude. In practical terms, this means that a connection subject to $\alpha = 1.05$ (thus $H \approx 1$) traffic conditions must endure fluctuations in traffic intensity where the peak-to-valley (and peak-to-mean) ratio is several factors higher than that of $\alpha = 1.95$ traffic at time scales 1 second and higher. Thus this is an affliction that has to be dealt with by not just long-running applications.

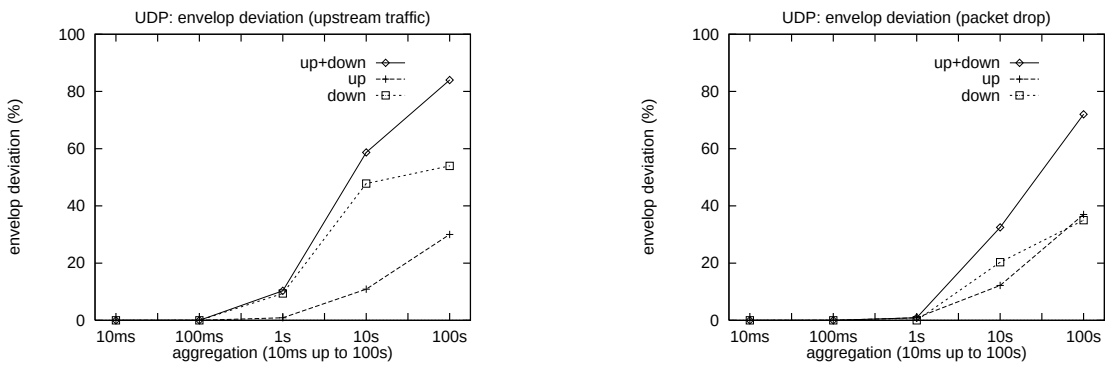

Figure 11: Left: Envelop deviation ratios δ^{ℓ}, δ_u^{ℓ}, δ_d^{ℓ} of $\alpha = 1.05$ upstream traffic at aggregation levels 10ms, 100ms, 1s, 10s, 100s relative to the envelop curves of $\alpha = 1.95$ upstream traffic. Right: Corresponding packet drop envelop deviation ratios.

4.1.3 Limitations to Curtailing Self-Similar Burstiness

We remark that the concentrated periods of congestion and light network contention of self-similar bursty traffic is an intrinsic performance feature of the network system that cannot be easily curtailed by traffic control. In particular, multiplexing of traffic streams at routers does not smooth out burstiness as is the case with non-scale-invariant bursty traffic such as Poisson. In fact, the ON/OFF model [37] says that a large number of independent ON/OFF traffic streams are *needed* to generate self-similarity and long-range dependence in the sense of fractional Brownian motion. Our work [26] has shown that in a generic client/server network system with bounded resources—and thus coupling or dependencies among traffic streams—exchanging of objects with heavy-tailed sizes leads to self-similar network traffic, and this phenomenon is robust and, to a large measure, unavoidable. The latter qualification refers to the fact that via drastic forms of traffic shaping—e.g., when making a bulk file transfer over a network, insert idle periods during the course of a transmission such that the resultant process becomes equivalent to transferring exponential size messages with exponential (or heavy-tailed) inter-session arrival times—it is possible to nullify the structural cause of self-similarity. However, such drastic measures are practically infeasible for many network applications since they would inordinately delay completion time. Thus they do not represent an effective traffic control option.

The blame of self-similar burstiness does not lie with any one connection or traffic source, but rather, it is a *collective* phenomenon attributable to all connections that engage in the transfer of objects with heavy-tailed sizes. This is somewhat analogous to holographic memories or error correcting codes where the property of an object (e.g., information content) is distributed over a set of constituent elements such that no single element is crucial to the recovery—in our case, realization—of the property. Past research on Unix file systems has shown—with hindsight—that

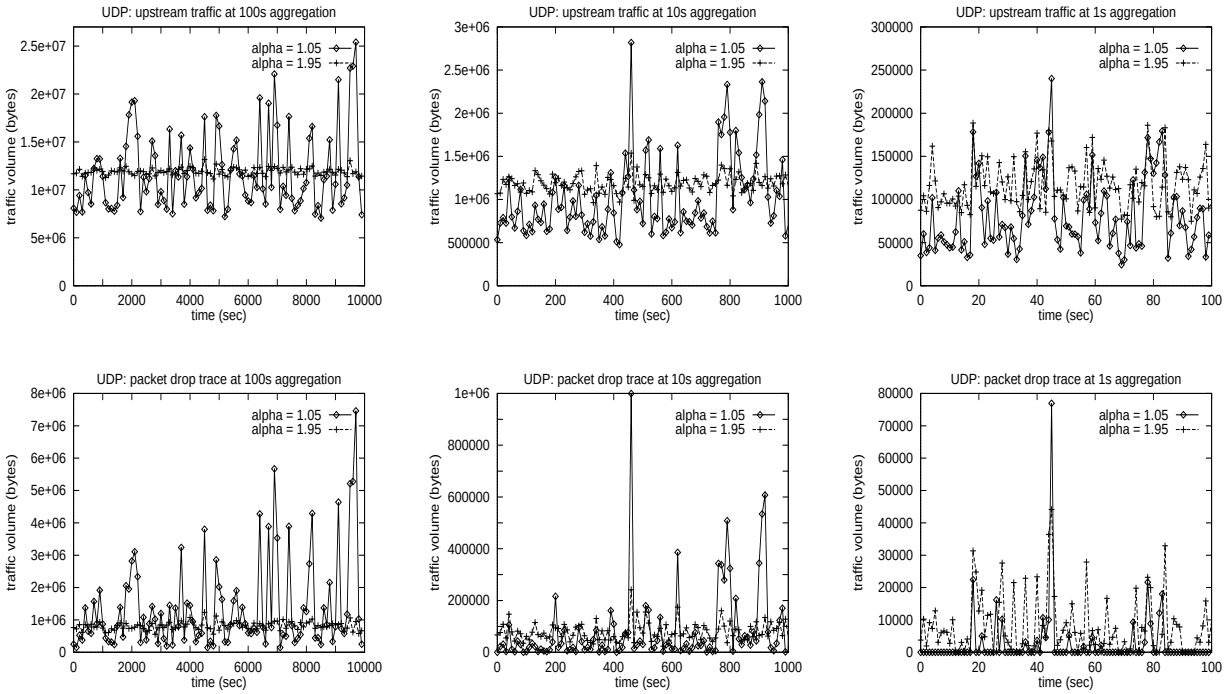

Figure 12: Top row: Upstream traffic trace of $\alpha = 1.05, 1.95$ traffic at 100s aggregation (left), 10s aggregation (middle), and 1s aggregation (right). Bottom row: Corresponding packet drop traces.

a typical Unix file system indeed possesses a file size distribution that follows a power law and exhibits characteristics of heavy-tailedness [35, 36, 25, 4, 33, 16]. This has also been confirmed in the case of reads and writes to NFS servers [6] and access patterns to documents on the World Wide Web [8]. Caching, the use of thumbnails for images, and paging of large documents can certainly help dampen the degree of self-similar burstiness by reducing the effective size of large objects [31]. It cannot, however, eliminate the intrinsic propensity of networked systems to generate self-similar network traffic where no single traffic source is a culprit in the sense of pumping self-similar bursty traffic into the system.

4.1.4 Amplified Packet Drops and Queueing Delay

In the Section 4.1.2 we have shown that concentrated periods of congestion and underutilization are the trademark performance characteristics of self-similar bursty traffic, making the conjoint provision of high QoS and network utilization difficult. Another important question is how scale-invariant burstiness affects packet loss rate and queueing delay in the long run given that it leads to significant variability in the short run. That is, does increased self-similar burstiness—keeping the average traffic intensity fixed—translate to increased packet drops and queueing delay?

To address this question, consider Figure 13 (left) which depicts two traffic traces $f(t)$, $g(t)$

and the packet drops incurred by the former—but not $g(t)$—when bandwidth is set at B^*. $f(t)$ and $g(t)$ represent sample paths of two stochastic processes where f is more variable than g. We will assume that the *sample means* are equal—i.e., $\mathbf{E}_t\{f(t)\} = \mathbf{E}_t\{g(t)\}$—so that performance differences are attributable to variability only. For illustrative purposes, we assume a degenerate queueing system where buffer capacity is zero, i.e., the system possesses no memory. Thus packet drops are completely determined by instantaneous arrivals and link bandwidth.

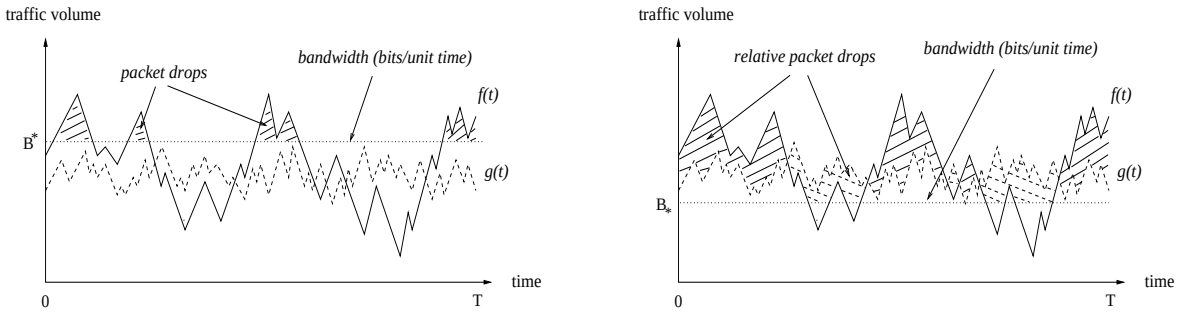

Figure 13: Left: Packet drops incurred in zero buffer capacity system by traffic series $f(t)$ when bandwidth B^* exceeds $g(t)$'s peak transmissin rate. Right: Relative packet drops incurred by $f(t)$, $g(t)$, when bandwidth is decreased from B^* to B_*.

In Figure 13 (left), bandwidth is set above the peak transmission rate of $g(t)$, $B^* > \max_t g(t)$, and thus no packet drops are incurred by g. On the other hand, the shaded regions depict the packet drops incurred by f, this being a perennial example where variability has a negative impact on network performance as represented by packet loss rate. Thinking of f as $\alpha = 1.05$ traffic and g as $\alpha = 1.95$ traffic, if the variability difference is large and bandwidth is set at a level such that it can absorb much of g's peak transmission rate, then a performance difference is implied. Note that if Figure 13 (left) represents traces at large time scale units, e.g., 100s aggregation, then the shaded region need not correspond to the actual volume of packet drops incurred (it is a lower bound on the packet drops) since pending on the shape or schedule of the traffic series at finer time scales, a range of possible outcomes is possible. For most processes of interest, however, the packet drops will be positively correlated to the size of the shaded area.

Figure 13 (right) illustrates a different aspect of the burstiness problem where the only change from the previous set-up is the decrease of bandwidth from B^* to B_*. Now both f and g incur packet drops, and since variability and $\mathbf{E}_t\{f(t)\} = \mathbf{E}_t\{g(t)\}$ imply that $f(t) < g(t)$ for some of the time, there exist time intervals when the packet drops stemming from g are greater than those of f. In fact, by suitably setting B_*, it is possible to devise a situation where the packet loss experienced by g is *higher* than that of f. That is, there is no intrinsic reason to conclude that higher variability must lead to higher packet loss rate.

The answer to the performance impact question of self-similar bursty traffic is two-phronged—

in the positive and in the negative—in essence, depending on the degree of burstiness relative to available resources. If network resources are "small" vis-à-vis the upstream traffic's variability, then having self-similar bursty traffic need not deteriorate performance in terms of packet loss rate and average queueing delay when compared to non-self-similar bursty traffic. On the other hand, if network resources are such that they can absorb much of the burstiness of one but not the other, then significant performance differences are possible. This is also consistent with analytical results reported in [12, 34].

When self-similar burstiness manifests itself negatively on performance, in particular, packet loss rate, it is due to an *asymmetry effect* whereby the packet drops stemming from $f(t) > g(t)$ periods dominate those emanating from the corresponding periods where $g(t) > f(t)$. This asymmetry effect can be quantified by the *relative packet drop ratio* ν^ℓ, ℓ = 10ms, 100ms, 1s, 10s, 100s, defined as follows. Assume t ranges over some interval $[0, T]$. Let

$$\mathcal{C}_u^\ell = \{\, t : X_{1.05}^\ell(t) > X_{1.95}^\ell(t) \,\}, \quad \mathcal{C}_d^\ell = \{\, t : X_{1.05}^\ell(t) < X_{1.95}^\ell(t) \,\},$$

where $X_{1.05}^\ell(t)$, $X_{1.95}^\ell(t)$ are the upstream traffic traces for $\alpha = 1.05$, $\alpha = 1.95$ traffic, respectively. Thus $\mathcal{C}_u^\ell$ is the set of time instances where the $\alpha = 1.05$ traffic exceeds that of $\alpha = 1.95$, and vice versa for $\mathcal{C}_d^\ell$. Let

$$s_u^\ell = \sum_{t \in \mathcal{C}_u^\ell} Y_{1.05}^\ell(t) - Y_{1.95}^\ell(t), \quad s_d^\ell = \sum_{t \in \mathcal{C}_d^\ell} Y_{1.95}^\ell(t) - Y_{1.05}^\ell(t),$$

where $Y_{1.05}^\ell$, $Y_{1.95}^\ell$ denote the packet drop sequences. Thus s_u^ℓ is the *net* packet drop by which the $\alpha = 1.05$ time series exceeds the packet drops of the the $\alpha = 1.95$ time series over time intervals where $X_{1.05}^\ell(t) > X_{1.95}^\ell(t)$, and similarly for s_d^ℓ, albeit, in the opposite direction. Note that for some time series it is possible to have negative s_u^ℓ, s_d^ℓ. For our $\alpha = 1.05, 1.95$ traffic traces where the first is self-similar and the other one is not, this was not a problem. The relative drop ratio is defined as

$$\nu^\ell = s_u^\ell / s_d^\ell.$$

Thus if packet drops are perfectly balanced, then $\nu^\ell = 1$. If $\nu^\ell > 1$, then this indicates a bias in packet drops attributable to the burstiness of $\alpha = 1.05$ traffic.

Figure 14 shows ν^ℓ for aggregation levels ℓ = 10ms, 100ms, 1s, 10s, 100s for both the TCP baseline runs and flow-controlled UDP runs. First, notice that over the entire aggregation time scale, $\nu^\ell > 1$. ν^ℓ increases as the aggregation level is increased reaching a value of 14.6 for the TCP traces and 6.2 for the UDP traces at the 100s time scale. This asymmetry effect which is most pronounced at time scales above the 1 second level indicated that the packet drops incurred by $\alpha = 1.05$ traffic during concentrated periods of congestion dominates the "savings" in packet drops achieved during light contention periods (i.e., the peaks and valleys) relative to the $\alpha = 1.95$ traffic.

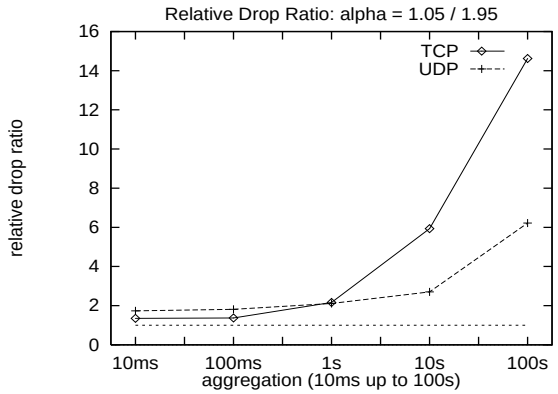

Figure 14: Relative packet drop ratios (1.05/1.95) of TCP and flow-controlled UDP runs for aggregation levels 10ms, 100ms, 1s, 10s, 100s.

Thus self-similar burstiness not only leads to periods of severe over- and under-utilization but it can also leads to an overall amplification of packet loss rate. It is interesting to note that the relative packet drop ratio for the TCP runs is greater than that of UDP runs at aggregation levels 10s, 100s, given that the opposite is to be expected since the open-loop flow-controlled UDP transmissions are not throttled by feedback when packet drops are high. This is explained by Figure 15 which

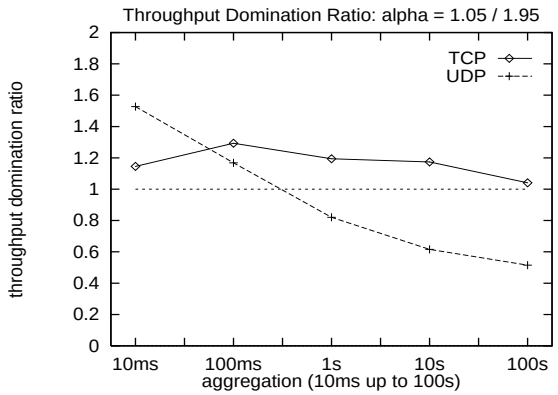

Figure 15: Throughput domination ratio (1.05/1.95) of TCP and flow-controlled UDP runs for aggregation levels 10ms, 100ms, 1s, 10s, 100s.

shows the *throughput domination ratio*

$$|\mathcal{C}_u^\ell|/|\mathcal{C}_d^\ell|$$

which captures how frequently (in time) $X_{1.05}^\ell(t)$ exceeds $X_{1.95}^\ell(t)$. We observe that in the case of UDP runs, for high aggregation levels, this ratio drops below 1 reaching 0.5 at 100s aggregation which indicates that the congestion periods (i.e., peaks) are *sparser* than the lightly loaded periods (i.e., valleys). In the case of the TCP runs, the throughput domination ratio remains above 1 across

the time scales considered which is consistent with the fact that concentrated congestion periods are dispersed, i.e., stretched out in time, by TCP's congestion control mechanism. The traffic shaping mechanism of TCP and its impact on the resultant downstream traffic characteristics including long-range dependence are discussed in [26].

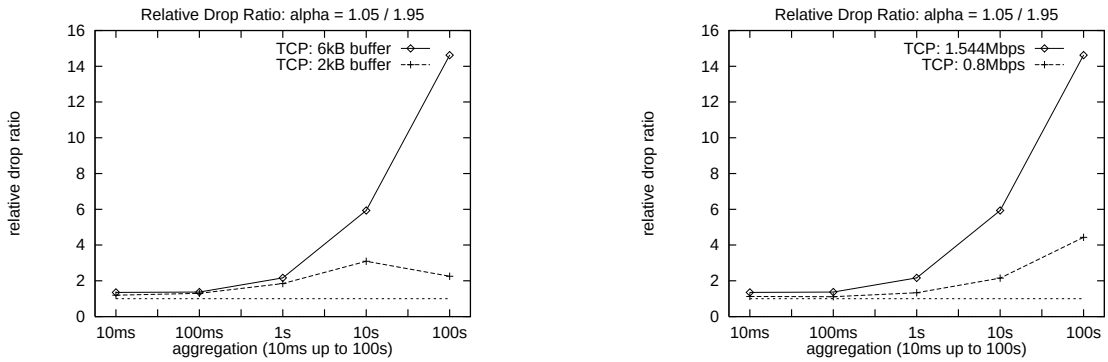

Figure 16: Left: Relative drop ratio for TCP runs as a function of time scale when buffer capacity is decreased from 6kB to 2kB. Right: Relative drop ratio for TCP runs when bandwidth is decreased from 1.544Mbps to 0.8Mbps.

Figure 16 shows the effect of varying network resources—buffer capacity and bandwidth—on network performance under self-similar traffic conditions. As bandwidth (right) or buffer capacity (left) is decreased—keeping everything else fixed—we observe a dampening of the relative drop ratio ν^ℓ at time scales 1s and above which is consistent with the fact that the effective burstiness level of $\alpha = 1.95$ traffic is amplified by the decrease in resources. Thus the relative domination of packet drops of $\alpha = 1.05$ traffic over $\alpha = 1.95$ is diminished accordingly. The earlier discussion on the asymmetry effect used a simplified network system with zero memory (i.e., buffer capacity), and thus no queueing, to bring out the essential components affecting the performance problem. It is not difficult to see that increasing buffer capacity will have an oppositive effect to that of decreasing bandwidth in determining packet loss rate under self-similar bursty traffic. The variability of $\alpha = 1.95$ traffic is readily absorbed by queueing when buffer capacity is not "too small," thus promoting the asymmetry effect. What is not obvious is the fact that the variability associated with self-similar bursty traffic is not readily absorbed by queueing; in fact, one of the main analytical results of queueing with long-range dependent input states that queue length distribution decays much more slowly for long-range dependent input than for Markovian sources such as Poisson whose queue length distribution decays exponentially [1, 2, 18, 22, 24]. Thus, even though a buffer capacity might be sufficient to absorb the fluctuations of $\alpha = 1.95$ traffic yielding a low packet loss rate, when subject to $\alpha = 1.05$ traffic, the latter's burstiness may remain largely intact. We will show these performance effects using performance measurements in subsequent sections.

Figure 17 shows the packet loss ratio of $\alpha = 1.05, 1.35, 1.65, 1.95$ traffic normalized against that

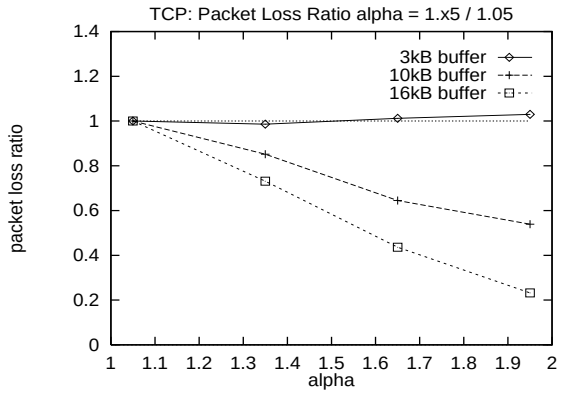

Figure 17: Packet loss ratios $c_{1.05}/c_{1.05}$, $c_{1.35}/c_{1.05}$, $c_{1.65}/c_{1.05}$, $c_{1.95}/c_{1.05}$ for TCP runs for buffer capacities 6kB, 10kB, 16kB.

of the $\alpha = 1.05$ traffic for three buffer capacities 3kB, 10kB, 16kB. For buffer capacity 10kB, we observe that the packet loss ratio is close to 0.5 when $\alpha = 1.95$, i.e., the packet loss rate incurred by $\alpha = 1.05$ traffic is almost twice as much as the $\alpha = 1.95$ traffic. This is even more pronounced when buffer capacity is 16kB. Notice that the buffer sizes are not set extraordinarily high thus concocting a rare phenomenon. At T1 bottleneck bandwidth and 32 concurrent connections transferring data, a buffer capacity of 16kB still yields a packet loss in excess of 1% across the board. Most interesting is the performance effect observed for buffer capacity 3kB which shows a *flat* packet loss ratio curve indicating that the varying degrees of scale-invariant burstiness associated with $\alpha = 1.05, 1.35, 1.65, 1.95$ traffic is nullified by the fact that buffer capacity is "small." The bottleneck queue finds itself in a saturated state even for the least bursty $\alpha = 1.95$ traffic, and thus the variability present in $\alpha = 1.95$ traffic is already sufficient to cause significant packet drops. In other words, the system finds itself in the configuration depicted in Figure 13 where the asymmetry effect is *least* pronounced. We call this the *saturation effect*. In fact, for buffer size 3kB, the $\alpha = 1.95$ traffic incurs a higher packet loss rate than the more bursty $\alpha = 1.05$ traffic.

4.2 Performance Measurements

In this section, we evaluate network performance when the degree of self-similar burstiness is varied. The scale-invariant burstiness of link traffic is intrinsically determined by the heavy-tailedness of file size distribution at the application layer (cf. Figure 2 in Section 3), captured by the shape parameter α of the Pareto distribution which acts as the main control variable. Other candidate file size distributions possessing short tails (e.g., exponential) yield similar performance results as the Pareto distribution with $\alpha = 1.95$, i.e., the mathematical property of heavy-tailedness for α close to 2 does not produce self-similar burstiness even over 10000 second simulation intervals at 10ms granularity. A discussion of sampling issues for heavy-tailed distributions can be found in [9].

First, we present performance data when the transport layer implements reliable communication with congestion control. We use TCP Reno as our principal benchmark protocol. Other versions of TCP yield similar results and their performance differences are shown in Section 4.4. We also present performance results evaluating the effect of self-similarity when unreliable communication is employed at the transport layer. We use flow-controlled UDP-based transmission for this purpose.

4.2.1 Packet Loss and Retransmission

Figure 18 shows the effect of varying the heavy-tailedness of the Pareto distribution[5] for $\alpha = 1.05, 1.35, 1.65$, and 1.95. The left and middle figures show packet loss rate and packet retransmission rate as a function of α for four values of buffer capacities, 3kB, 10kB, 46kB, and 128kB. When the bottleneck buffer size is very small (2kB), there is little difference in performance across the α values, exhibiting a flat performance curve. This is due to the burstiness at $\alpha = 1.95$ being already sufficient to induce high packet loss. This is the *saturation effect* discussed in Section 4.1.4.

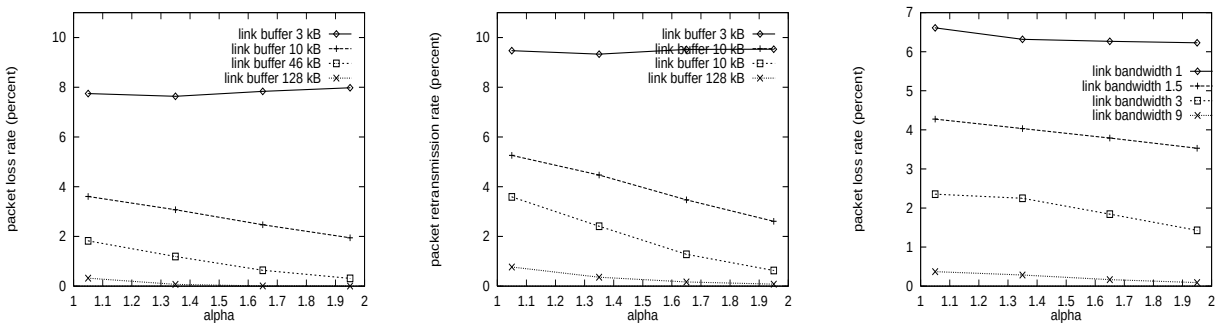

Figure 18: TCP run. Packet loss rate (left) and packet retransmission rate (middle) as a function of α for four values of buffer capacity. Packet loss rate (right) as a function of α for four values of link bandwidths.

In the case of TCP, since TCP employs feedback control triggering backoff upon perceived packet loss, the potential for an amplified effect at smaller α values is further muted by the *initial* packet drops occurring, on average, across all α values. The amplification effect is realized and clearly visible when feedback congestion control is not used as in the case of the non-flow-controlled UDP-based protocol (cf. Figure 21 below). When buffer size is large (128kB), the increased capacity allows even the high burstiness at $\alpha = 1.05$ to be accommodated via queueing, and we see a gradual deterioration in packet loss and retransmission as α approaches 1.

The slope of the performance curve is most pronounced for intermediate values of the bottleneck buffer capacity (10kB, 46kB) at which the difference in burstiness between $\alpha = 1.05$ and $\alpha = 1.95$

[5]We remark that the location parameter of the Pareto distribution was adjusted so that the *sample mean* of the file size requests sampled over 10000 seconds remained invariant at 4.1kB for the range of α values considered.

takes its highest toll. However, we note that packet loss and retransmission remain a *linear* function of α and this "robustness" with respect to self-similar burstiness is achieved by the action of congestion control. We will see in Section 4.2.3 that in its absence, a nonlinear degradation in performance results as α approaches 1. Figure 18 (right) shows a similar behavior with respect to packet loss rate when the bottleneck bandwidth is varied (1Mb/s, 1.5Mb/s, 3Mb/s, 9Mb/s) for a fixed buffer size of 6kB.

4.2.2 Throughput and Queueing Delay

Figure 19 (left) shows reliable throughput as a function of α for four values of buffer capacity. *Reliable throughput* ρ is defined as the average over all file transfers (indexed by k) over time of the ratio

$$\rho_k = \frac{S_k}{W_k} = \frac{S_k}{S_k + H_k}$$

where S_k is the k'th file size, W_k is the total number of bytes expended to reliably transmit the k'th file, and $H_k = W_k - S_k$ is the net overhead incurred including packet retransmissions. Hence,

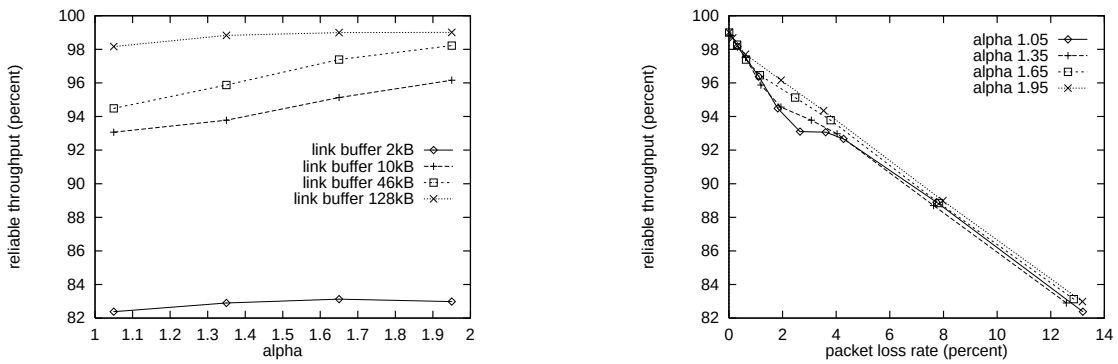

Figure 19: TCP run. Reliable throughput (left) as a function of α; reliable throughput as a function of packet loss rate (right).

the fewer the lost packets and resulting retransmissions, the closer ρ is to 1. In the case of reliable communication with TCP's window control, we use reliable throughput as a measure of "effective throughput" to better account for retransmission overhead and distinguish it from raw throughput. We may also define reliable throughput by the ratio S_k/T_k where T_k is the completion time of reliable transmission of the k'th file. Since W_k and T_k are positively correlated, they yield similar results.

As with packet loss rate, Figure 19 (left) shows performance curves that are most sensitive with respect to self-similarity for intermediate values of buffer capacity. This is as expected since throughput and packet loss rate are almost linearly correlated as seen in Figure 19 (right). For small buffer sizes, the performance difference is again minimal due to the saturation effect.

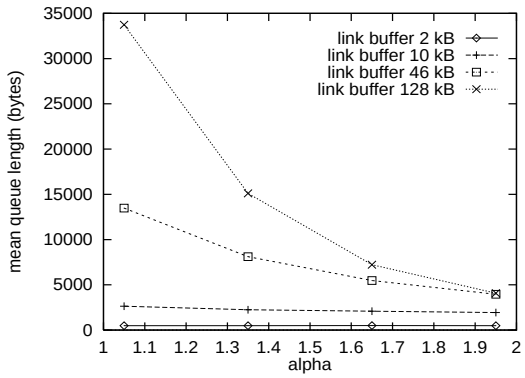

Figure 20: Buffer occupancy (i.e., mean queue length) as a function of α for buffer sizes in the range 2kB–128kB for TCP base runs.

Figure 20 shows buffer occupancy or mean queue length at the bottleneck link output buffer as a function of α for four values of buffer capacity, 2kB, 10kB, 46kB, and 128kB. For small values of buffer size, even at $\alpha = 1.95$ the buffer overflow is high, and because of TCP's congestion control (see the discussion above), buffer utilization remains invariant across all α values. This is the manifestation of the saturation effect for queueing which is analogous to its impact on packet drops.

When buffer capacity is made larger, however, the highly self-similar traffic at $\alpha = 1.05$ is able to utilize the increased queueing capacity to achieve further improvement in throughput. This occurs at a disproportionate—i.e., superlinear—increase in queueing delay relative to the change in α as seen in Figure 20. We will show in Section 4.3 that this nonlinear relationship also manifests itself as a sharp increase in the curvature of the delay-throughput curve.

4.2.3 Unreliable Non-Flow-Controlled Transport

Figure 21 shows packet loss rate, link utilization, and mean queue length as a function of α for four values of buffer capacity when an unreliable UDP-based non-flow-controlled transport protocol is used. An extension of the unreliable non-flow-controlled protocol to a flow controlled version implementing a simple source-throttled open-loop control is discussed in Section 4.4. The present scheme serves to show the influence of the protocol stack in determining network performance as a function of self-similarity at one *extreme* end of the spectrum of possible transport layer controls. It is not meant to be taken as a realistic transport control.

In contrast to the gradual increase in packet loss achieved when TCP is employed (see Figure 18), we observe a *superlinear* degradation in the packet loss rate as α approaches 1. Throughput as captured by link utilization improves only *sublinearly* as a function of α. Thus, TCP's congestion control and reliability mechanism have the effect of pulling down the superlinear dependence of

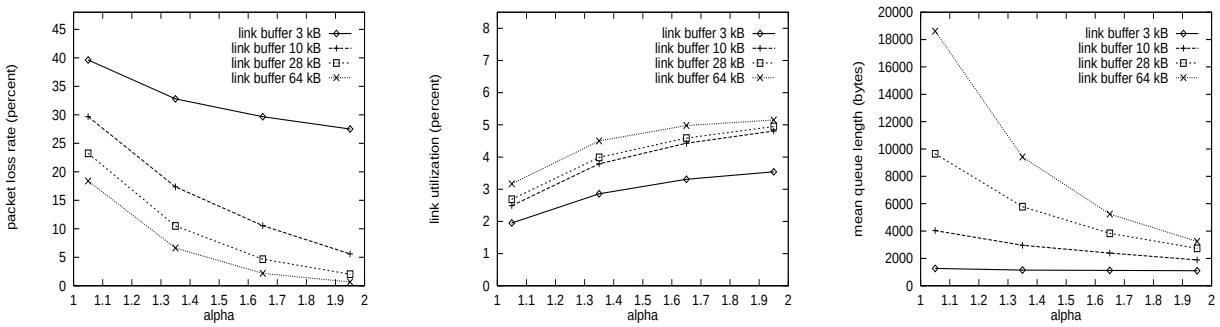

Figure 21: UDP run. Packet loss rate (left), link utilization (middle), and mean queue length as a function of α.

network performance on self-similarity to a more robust linear relationship with respect to packet loss and throughput. Figure 21 (right) shows that mean queue length continues to depend superlinearly on α for large buffer sizes. It is not surprising that UDP-based unreliable transport without additional control performs poorly under various network conditions. However, the fact that TCP's congestion control and reliability mechanism are able to achieve a *linear* dependence on the degree of self-similarity is *empirical* and we will study this feature, especially its high cost with respect to increased queueing delay, in the next section.

4.3 Impact of Network Resources

The previous section has shown that self-similar burstiness as captured by the shape parameter α of the Pareto distribution acting at the application layer exerts a direct influence on network performance through its manifestation as self-similar link traffic. How, and with what magnitude, it affects performance is modulated by the protocols acting at the transport layer. In this section, we study the effect of network resources on performance, especially with respect to the trade-off relationship between throughput and delay as well as packet loss and delay—all under varying burstiness traffic conditions.

4.3.1 Buffer Capacity

Figure 22 shows the effect of varying the bottleneck buffer capacity for buffer sizes in the range 2kB–128kB on packet loss rate, packet retransmission rate, and reliable throughput using TCP Reno. In Figure 22 (left), we see a sharp threshold behavior of packet loss rate as a function of buffer size, "partitioning" buffer capacity into a highly sensitive and a saturated domain. This holds, modulo a shift effect, for all the α values considered. A similar observation holds for packet retransmission rate and reliable throughput as the graphs in Figure 22 (middle) and (right) indicate. The shift effect is least pronounced for small buffer capacities due to the saturation effect.

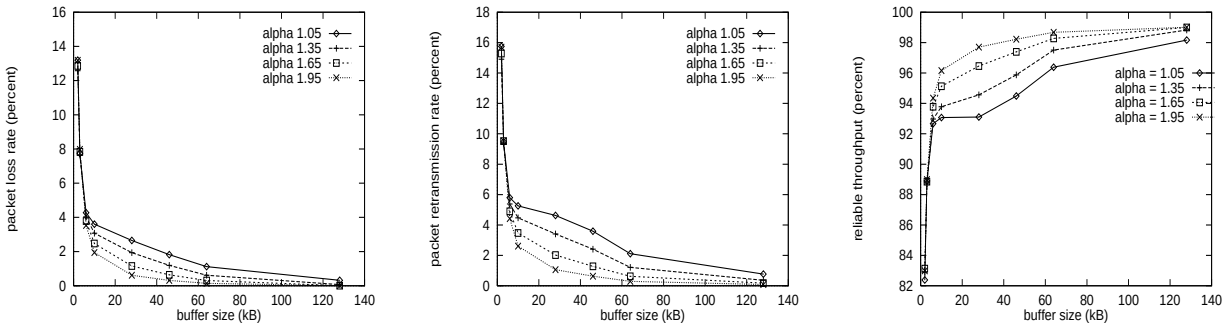

Figure 22: TCP run. Packet loss rate (left) and packet retransmission rate (middle) as a function of buffer capacity. Reliable throughput (right) as a function of buffer capacity.

Whereas for throughput and packet loss rate the improvement in performance for large buffer sizes was gradual, Figure 23 shows that the qualitative dependence of mean queue length for large buffer sizes is determined by the degree of self-similar burstiness. For α close to 2, the functional dependence of mean queue length on buffer capacity is clearly sublinear (roughly logarithmic) whereas for α close to 1, the dependence becomes linear. For very large values of the buffer size we expect the delay curve to saturate even when α is close to 1 although the practical significance of the difference in the magnitude of their queueing delay will be preserved. The proportional increase in buffer occupancy for α close to 1 indicates that, even for large buffer sizes, traffic burstiness is high enough to cause constant utilization.

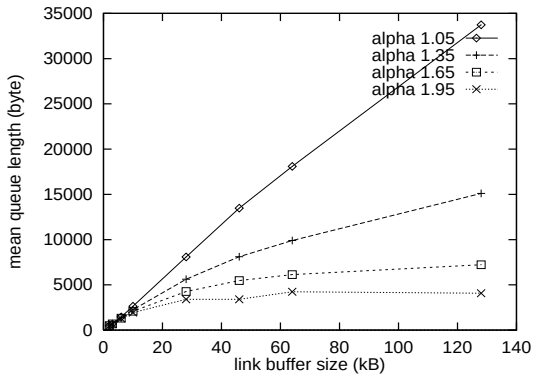

Figure 23: Mean queue length as a function of buffer capacity for $\alpha = 1.05, 1.35, 1.65, 1.95$.

The large difference in the growth of queueing delay vis-à-vis reliable throughput and packet loss rate as a function of buffer capacity manifests itself in a trade-off relationship which is shown in Figures 24 (left) and (right). Figure 24 (left) shows the delay-packet loss curve for four values of $\alpha = 1.05, 1.35, 1.65, 1.95$. For α close to 1, we observe an exponential trade-off relationship between queueing delay and packet loss rate, with significant differences in the curvature among

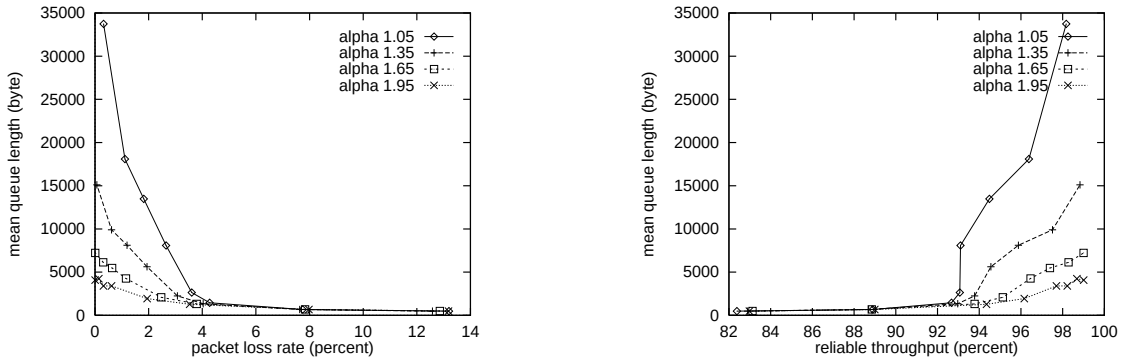

Figure 24: QoS trade-off: Mean queue length vs. packet loss rate trade-off relation (left), and mean queue length vs. reliable throughput trade-off relation (right).

the four performance curves. In the context of facilitating multimedia traffic such as video and voice while meeting their diverse QoS requirements, the performance curves in Figure 24 (left) show that low packet loss, on average, can only be achieved at a significant increase in queueing delay and vice versa. Since video, image, audio, and voice traffic tend to be large-volume transfers whose increased presence causes the intrinsic propensity of a networked system to generate self-similar network traffic to be amplified, it is the precisely the QoS-sensitive traffic that both promotes scale-invariant burstiness as well as suffers most from its consequences via the sensitive dependence of the delay-packet loss trade-off relation on the degree of self-similarity in the system. Figure 24 (right) shows the delay-throughput curve which exhibits a trade-off relationship similar to queueing delay vs. packet loss rate, its curvature increasing as α approaches 1.

4.3.2 Bottleneck Bandwidth

Figure 26 (left) shows the effect of varying bottleneck link bandwidth on packet loss rate for four values of α. As in the buffer capacity case, we see a threshold effect with packet loss rate being most sensitive for small values of link bandwidth followed by a gradual decline in packet drop as bandwidth is increased. A similar situation holds for packet retransmission rate which is shown in Figure 26 (middle). The rightmost plot shows reliable throughput as a function of bandwidth, and unlike in the case of buffer capacity, a linear increase in throughput is observable across all α values.

Figure 26 (left) shows the effect of increasing bandwidth on queueing delay for $\alpha = 1.05$, 1.35, 1.65, 1.95 when buffer capacity is fixed at 128kB. At $\alpha = 1.95$, we see a saturation effect as bandwidth is increased from 1.5Mbps to 9Mbps producing successively smaller improvements in mean queue length which indicates a diminishing return of the marginal utility of link bandwidth with respect to queueing delay. As α approaches 1, however, the situation reverses such that at

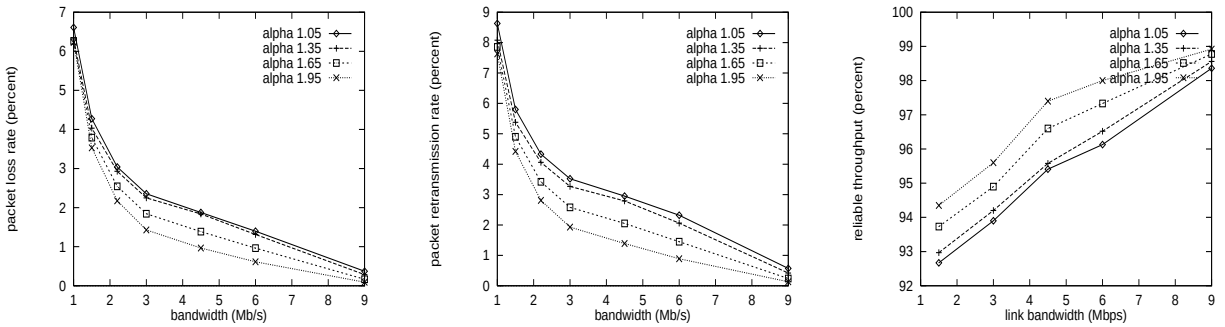

Figure 25: TCP run. Packet loss rate (left), retransmission rate (middle), and reliable throughput (right) as a function of link bandwidth for $\alpha = 1.05, 1.35, 1.65, 1.95$.

$\alpha = 1.05$, the opposite is true. That is, as bandwidth is increased, we see an accelerated decline in mean queue length vis-à-vis the case when α is large. Figure 26 (right) shows performance results for the same set-up except that the bottleneck buffer size was fixed at the smaller value of 6kB. Unlike before, we now observe a constant slope in the delay curve as a function of α across all the bandwidth values considered, and it does not exhibit a saturation effect nor acceleration effect at either end of α. This implies that when the bottleneck buffer capacity is small, then increasing bandwidth carries only a gradual improvement in queueing delay independent of whether traffic is highly self-similar or not.

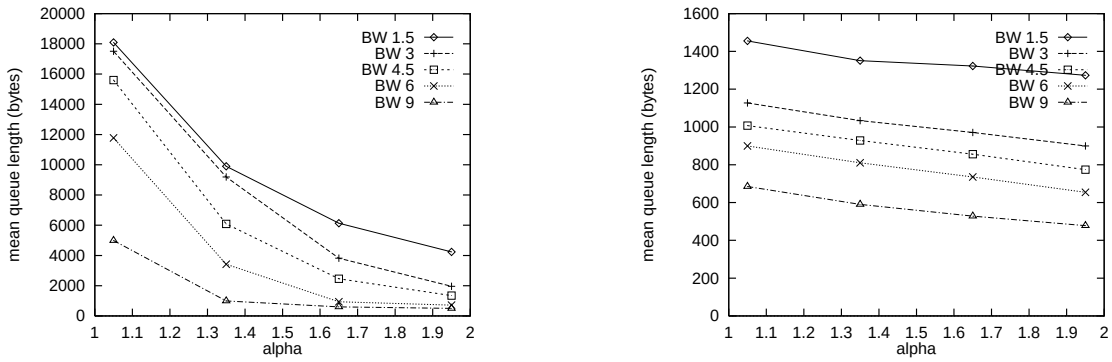

Figure 26: TCP run. Mean queue length as a function of α for five values of bandwidths 1.5Mbps, 3Mbps, 4.5Mbps, 6Mbps, 9Mbps for a fixed buffer capacity of 128kB (left), and for fixed buffer capacity 6kB (right).

In the context of network design, the previous observations imply that when the overall system is prone to high levels of long-range dependence as in the case of multimedia traffic, increasing buffer capacity alone is a bad idea since it carries a significant queueing delay penalty while improving packet loss only gradually. A bandwidth-driven network resource allocation policy, assuming a fixed overall traffic demand, is more effective since it improves both throughput and queueing delay

gradually, while potentially yielding an even more significant improvement in queueing delay if buffer capacity is large.

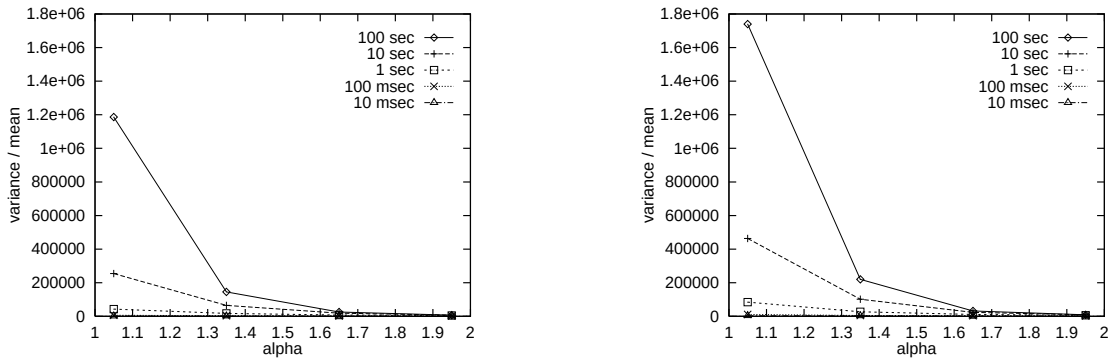

Figure 27: TCP run. Variance/mean ratio for link throughput as a function of time-scale and α when bottleneck bandwidth is 3Mb/s (left) and when bottleneck bandwidth is 6Mb/s (right).

Figure 27 shows an auxiliary effect of increasing link bandwidth, namely, compression of the time axis of the self-similar link traffic time series as well as amplification of its amplitude. The variance/mean ratio of link traffic is related to the index of dispersion count and measures the variance of the aggregated time series at five aggregation levels—10ms, 100ms, 1sec, 10sec, 100sec—against the mean. The larger this ratio for α small vis-à-vis when α is large, the more "bursty" the underlying time series and the easier it is to detect statistical differences between highly self-similar and less-so traffic. Moreover, if the difference is preserved for fine levels of aggregation, then the easier it is observe and exploit the dependency structure in $\alpha \approx 1$ traffic.

4.3.3 Unreliable Non-Flow Controlled Transport

The sensitivity results and trade-off relationships of Section 4.3.1 are confirmed in the case of the non-flow-controlled UDP-based unreliable transport protocol discussed earlier. Packet loss rate and mean queue length, as a function of buffer capacity, exhibit qualitative shapes similar to their corresponding graphs for flow-controlled reliable transport using TCP Reno in Section 4.3.1. In fact, they have an even more sensitive dependence on α. The plots are omitted here for brevity.

Figure 28 depicts the delay-packet loss and delay-throughput trade-off relationships showing a sharp increase in the curvature of the performance graphs as a function of self-similarity. The delay-link utilization curve, although amplified in shift, is smoother than its corresponding counter-part for flow-controlled reliable communication—the delay-reliable throughput curve. This suggests that the jaggedness in Figure 24 was caused by TCP's reliability and congestion control mechanism.

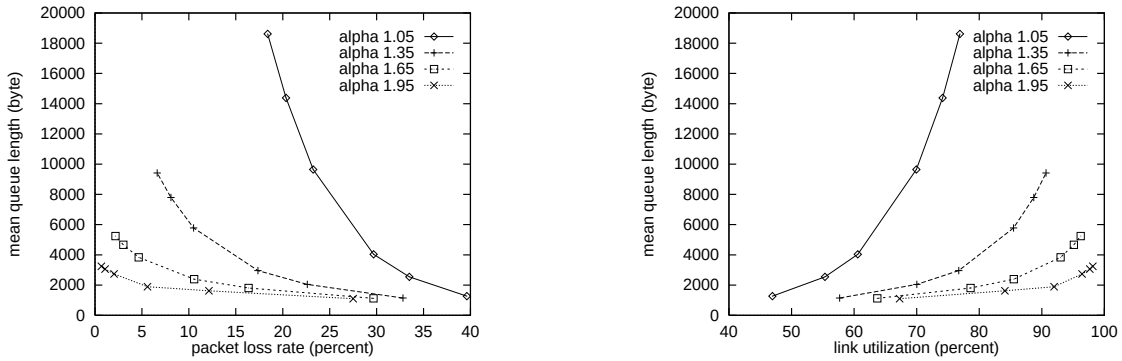

Figure 28: UDP run. Mean queue length vs. packet loss rate (left) and mean queue length vs. link utilization (right).

4.4 Effectiveness of Congestion Control

The previous sections have shown performance results as a function of self-similarity and network resources, and the impact of the transport layer in affecting performance. In this section, we compare the performance results across several congestion control algorithms, first, in the case of flow-controlled unreliable transport using UDP, and second, by comparing performance differences across three versions of TCP—Reno, Tahoe, and Vegas.

4.4.1 Flow-Controlled Unreliable Transport

We employ a simple open-loop flow-controlled unreliable transport protocol based on UDP to test the effect of stretching out a file transfer session into an output stream of "on-average" constant height. The arrival rate is throttled by a rate parameter τ which determines the minimum (time) spacing between successive packets. Thus, by suitably adjusting the inter-packet spacing parameter

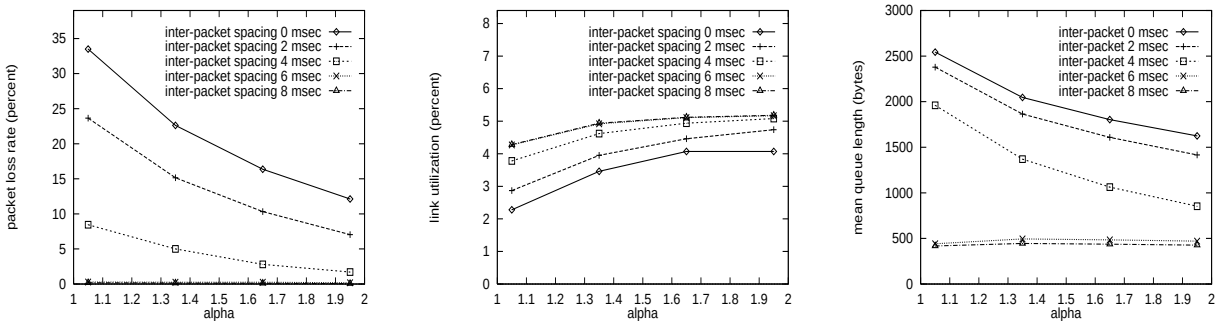

Figure 29: Flow-controlled UDP for four values of interpacket spacings: Packet loss rate (left), link utilization (middle), and mean queue length (right) as a function of α.

τ, the maximum arrival rate of a traffic source is controlled. As $\tau \to \infty$, the traffic source becomes

more and more like the ON/OFF model of Willinger et al. [37]—except for the *finite* number of traffic sources—due to the decoupling achieved by making the consequences of resource contention negligible.

Figure 29 (left) shows packet loss rate as a function of α for five values of τ, τ = 0ms, 2ms, 4ms, 6ms, 8ms. For small τ, we observe the characteristic nonlinear increase in packet loss rate as $\alpha \to 1$. As τ is increased, however, packet loss rate flattens out until the curve becomes horizontal and packet loss approaches 0. Figure 29 (middle) shows the corresponding effect on link utilization. We observe that link utilization approaches saturation as τ is increased for all values of α. The rightmost plot shows the effect of τ on mean queue length which, after exhibiting a nonlinear dependence on α for small values of τ, flattens out as τ is increased.

Figure 30 shows the Hurst parameter estimates using the V-T and R/S methods. The curves exhibit an upward shift as the interpacket spacing parameter τ is increased. This is due to fewer packets being discarded because of buffer overflow, thus better preserving the tail of the file size distribution which manifests itself as large file transfer sessions at the transport layer.

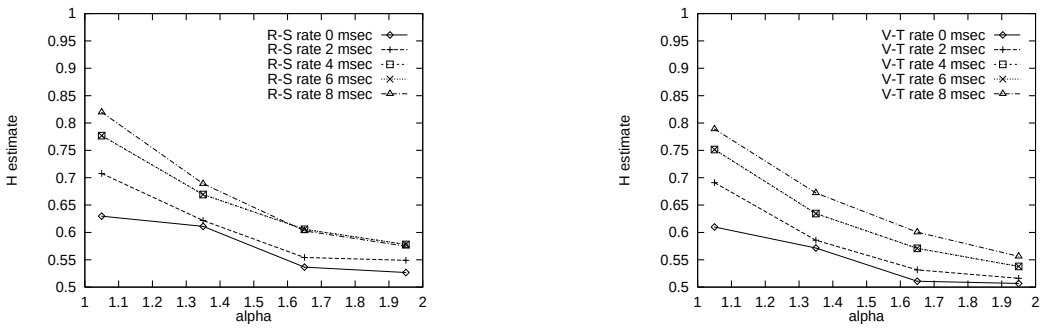

Figure 30: Flow-controlled UDP run. Hurst parameter estimates using Variance-Time (left) and R/S (right) for five values of interpacket spacings.

4.4.2 Performance Comparison of TCP Reno, Tahoe, and Vegas

In this section, we present preliminary findings on the performance and effectiveness of three versions of TCP implementing progressively more sophisticated congestion control features when going from TCP Tahoe to Reno to the recently proposed TCP Vegas [7]. Figure 31 (left) compares the packet loss rate of Tahoe, Reno, and Vegas as a function of self-similarity under identical network conditions. Bottleneck buffer capacity was set to 6kB and link bandwidth was set at 1.5Mbps. All three protocols exhibit the characteristic linear dependence on α as seen in the more comprehensive study of TCP Reno under various buffer capacities and link bandwidths (Section 4.2). The gap in performance between Tahoe, Reno, and Vegas remains consistent, and its width stays "roughly" the same across all α values. That is, even under highly self-similar traffic conditions, the per-

formance gap is preserved across the three congestion control algorithms. The performance gain from Tahoe to Reno is relatively minor while the performance jump from Reno to Vegas is more pronounced, its magnitude falling into the same ballpark range as the quantitative measurements reported elsewhere [7, 3].

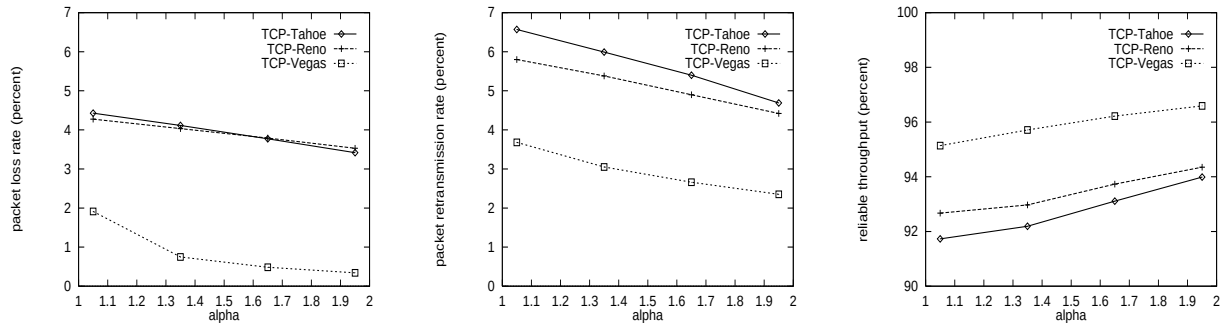

Figure 31: TCP Reno vs. Tahoe vs. Vegas run. Packet loss rate (left), packet retransmission rate (middle), and reliable throughput (right) as a function of α.

5 Conclusion

In this paper we have studied the impact of traffic self-similarity on network performance based on a network model and its protocol stack comprised of the following three features: (i) heavy-tailed object or file size distribution in the application layer, (ii) congestion control and reliable/unreliable message transmission in the transport layer, and (iii) network traffic measured at the link layer. We have shown that (i) and (ii) collude to give (iii) whose manifestation mechanism is robust and grounded in physical—i.e., networking—reality. We have examined network performance from three different viewpoints, first, with respect to the impact of self-similar burstiness, second, as a function of network resources and the resulting trade-off relationship of performance variables, and third, by comparing the performance of characteristics of several different congestion control schemes.

We have given a pragmatic measurement-based analysis of the performance problem under self-similar traffic conditions, explicating the essential mechanisms that underly the two principal performance implications: concentrated periods of congestion and amplified packet loss/queueing delay. We have shown that not only is the upstream traffic self-similar, the packet drop process and downstream traffic exhibit self-similar burstiness as well. We have shown that concentrated periods of congestion—and on the flip side, underutilized periods—are intrinsic properties of a network system subject to self-similar traffic conditions. In particular, unless extremely drastic traffic control actions are employed—somewhat akin to the throwing out of the baby with the bathwater analogy—

the manifestation of structural causality in the application layer as self-similar network traffic at the link layer is unavoidable. We have shown that self-similar burstiness need not lead to degraded packet loss rate and queueing delay, depending on the magnitude of the asymmetry effect which, in turn, is influenced by the relative magnitude of network resources vis-à-vis traffic burstiness. However, from a practical perspective, unless network resources are saturated to a level where short-range correlations are elevated to the role of sole dominating factor, "typically"—i.e., based on the range of network configurations we have considered where all forms of behavior are observable—self-similar burstiness does lead to degraded network performance in terms of packet loss rate and queueing delay in addition to the concentrated periods of overutilization/underutilization which are unconditional.

Current work is directed in a number of different avenues from measurement work, to the finer discernment of protocol stack effects (e.g., MAC protocols at the link layer), to traffic management at the middleware layer (e.g., caching, paging, and thumbnailing), and most importantly, self-similar traffic control at the transport layer using end-to-end congestion control.

References

[1] A. Adas and A. Mukherjee. On resource management and QoS guarantees for long range dependent traffic. In *Proc. IEEE INFOCOM '95*, pages 779–787, 1995.

[2] R. Addie, M. Zukerman, and T. Neame. Fractal traffic: measurements, modelling and performance evaluation. In *Proc. IEEE INFOCOM '95*, pages 977–984, 1995.

[3] J. Ahn, P. Danzig, Z. Liu, and L. Yan. Evaluation of TCP Vegas: Emulation and experiment. In *Proc. ACM SIGCOMM '95*, pages 185–195, 1995.

[4] M. G. Baker, J. H. Hartman, M. D. Kupfer, K. W. Shirriff, and J. K. Ousterhout. Measurements of a distributed file system. In *Proceedings of the Thirteenth ACM Symposium on Operating Syst em Principles*, pages 198–212, October 1991.

[5] Jan Beran. *Statistics for Long-Memory Processes.* Monographs on Statistics and Applied Probability. Chapman and Hall, New York, NY, 1994.

[6] R. R. Bodnarchuk and R. B. Bunt. A synthetic workload model for a distributed system file server. In *Proceedings of the 1991 SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, pages 50–59, 1991.

[7] L. Brakmo and L. Peterson. TCP Vegas: end to end congestion avoidance on a global internet. *IEEE J. Select. Areas Commun.*, 13(8):1465–1480, 1995.

[8] M. Crovella and A. Bestavros. Self-similarity in world wide web traffic: Evidence and possible causes. In *Proceedings of the 1996 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, May 1996.

[9] M. Crovella and L. Lipsky. Long-lasting transient conditions in simulations with heavy-tailed workloads. To appear in *Proc. 1997 Winter Simulation Conference*, 1997.

[10] Sally Floyd. Simulator tests. Available in `ftp://ftp.ee.lbl.gov/papers/simtests.ps.Z`. ns is available at `http://www-nrg.ee.lbl.gov/nrg`., July 1995.

[11] M. Garret and W. Willinger. Analysis, modeling and generation of self-similar VBR video traffic. In *Proc. ACM SIGCOMM '94*, pages 269–280, 1994.

[12] M. Grossglauser and J-C. Bolot. On the relevance of long-range dependence in network traffic. In *Proc. ACM SIGCOMM '96*, pages 15–24, 1996.

[13] D. Heyman and T. Lakshman. What are the implications of long-range dependence for VBR-video traffic engineering? *IEEE/ACM Transactions on Networking*, 4(3):301–317, June 1996.

[14] Daniel Heyman. Some issues in performance modeling of data teletraffic. In *Proc. SPIE International Conference on Performance and Control of Network Systems*, pages 284–295, 1997.

[15] C. Huang, M. Devetsikiotis, I. Lambadaris, and A. Kaye. Modeling and simulation of self-similar variable bit rate compressed video: a unified approach. In *Proc. ACM SIGCOMM '95*, pages 114–125, 1995.

[16] Gordon Irlam. Unix file size survey - 1993. Available at `http://www.base.com/gordoni/ufs93.html`, September 1994.

[17] Van Jacobson. Congestion avoidance and control. In *Proc. ACM SIGCOMM '88*, pages 314–329, 1988.

[18] Philippe Jacquet. Queueing with aggregated up/down arrivals, 1996. Preprint.

[19] P. Jelenkovic and B. Melamed. Automated TES modeling of compressed video. In *Proc. IEEE INFOCOM '95*, pages 746–752, 1995.

[20] W. Leland, M. Taqqu, W. Willinger, and D. Wilson. On the self-similar nature of ethernet traffic. In *Proc. ACM SIGCOMM '93*, pages 183–193, 1993.

[21] W.E. Leland, M.S. Taqqu, W. Willinger, and D.V. Wilson. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2:1–15, 1994.

[22] N. Likhanov and B. Tsybakov. Analysis of an ATM buffer with self-similar ("fractal") input traffic. In *Proc. IEEE INFOCOM '95*, pages 985–992, 1995.

[23] B. B. Mandelbrot. Long-run linearity, locally gaussian processes, h-spectra and infinite variances. *Intern. Econom. Rev.*, 10:82–113, 1969.

[24] I. Norros. A storage model with self-similar input. *Queueing Systems*, 16:387–396, 1994.

[25] J. K. Ousterhout, H. Da Costa, D. Harrison, J. a Kunze, M. Kupfer, and J. G. Thompson. A trace-driven analysis of the UNIX 4.2 BSD file system. In *Proceedings of the Tenth ACM Symposium on Operating System Pr inciples*, pages 15–24, December 1985.

[26] K. Park, G. Kim, and M. Crovella. On the relationship between file sizes, transport protocols, and self-similar network traffic. In *Proc. IEEE International Conference on Network Protocols*, pages 171–180, October 1996.

[27] K. Park, G. Kim, and M. Crovella. On the effect of traffic self-similarity on network performance. In *Proc. SPIE International Conference on Performance and Control of Network Systems*, pages 296–310, 1997.

[28] K. Park, M. Sitharam, and S. Chen. Quality of service provision in noncooperative network environments. Technical Report CSD-TR-97-039, Department of Computer Sciences, Purdue University, 1997.

[29] K. Park and T. Tuan. Congestion control for self-similar network traffic. In preparation, 1998.

[30] Kihong Park. Warp control: a dynamically stable congestion protocol and its analysis. *Journal of High Speed Networks*, 2(4):373–404, 1993.

[31] Kihong Park. On the effect and control of self-similar network traffic: a simulation perspective. To appear in *Proc. 1997 Winter Simulation Conference*, December 1997.

[32] V. Paxson and S. Floyd. Wide-area traffic: the failure of Poisson modeling. In *Proc. ACM SIGCOMM '94*, pages 257–268, 1994.

[33] K. K. Ramakrishnan, P. Biswas, and R. Karedla. Analysis of file I/O traces in commercial computing environments. In *Proceedings of the 1992 SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, pages 78–90, 1992.

[34] B. Ryu and A. Elwalid. The importance of long-range dependence of VBR video traffic in ATM traffic engineering: myths and realities. In *Proc. ACM SIGCOMM '96*, pages 3–14, 1996.

[35] M. Satyanarayanan. A study of file sizes and functional lifetimes. In *Proceedings of the Eighth ACM Symposium on Operating System P rinciples*, December 1981.

[36] Alan Jay Smith. Analysis of long term file reference patterns for application to file migration algorithms. *IEEE Transactions on Software Engineering*, 7(4):403–410, July 1981.

[37] W. Willinger, M. Taqqu, R. Sherman, and D. Wilson. Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level. In *Proc. ACM SIGCOMM '95*, pages 100–113, 1995.