On the relationship between file sizes, transport protocols, and self-similar network traffic

Kihong Park*
Department of Computer Sciences
Purdue University
West Lafayette, IN 47907
park@cs.purdue.edu

Gitae Kim† Mark Crovella‡
Computer Science Department
Boston University
Boston, MA 02215
{kgtjan,crovella}@cs.bu.edu

Abstract

Recent measurements of local-area and wide-area traffic have shown that network traffic exhibits variability at a wide range of scales. In this paper, we examine a mechanism that gives rise to self-similar network traffic and present some of its performance implications. The mechanism we study is the transfer of files or messages whose size is drawn from a heavy-tailed distribution.

First, we show that in a "realistic" client/server network environment—i.e., one with bounded resources and coupling among traffic sources competing for resources—the degree to which file sizes are heavy-tailed can directly determine the degree of traffic self-similarity at the link level. We show that this causal relationship is robust with respect to changes in network resources (bottleneck bandwidth and buffer capacity), network topology, the influence of cross-traffic, and the distribution of interarrival times.

Second, we show that properties of the transport layer play an important role in preserving and modulating this relationship. In particular, the reliable transmission and flow control mechanisms of TCP (Reno, Tahoe, or Vegas) serve to maintain the long-range dependency structure induced by heavy-tailed file size distributions. In contrast, if a non-flow-controlled and unreliable (UDP-based) transport protocol is used, the resulting traffic shows little self-similarity: although still bursty at short time scales, it has little long-range dependence.

Third, we show performance implications of self-similarity as represented by performance measures including packet loss rate, retransmission rate, and queueing delay. Increased self-similarity, as expected, results in degradation of performance. Queueing delay, in particular, exhibits a drastic increase with increasing self-similarity. Throughput-related measures such as packet loss and retransmission rate, however, increase only gradually with increasing traffic self-similarity as long as reliable, flow-controlled transport protocol is used.

*Supported in part by NSF grant CCR-9204284. This work was performed while the author was at Boston University.
†Supported in part by NSF grant CCR-9308344.
‡Supported in part by NSF grant CCR-9501822.

1 Introduction

Recent measurements of local-area and wide-area traffic [12, 24] have shown that network traffic exhibits variability at a wide range of scales. Such scale-invariant variability is in strong contrast to traditional models of network traffic, which show variability at short time scales but are essentially smooth at large time scales; i.e., they lack long-range dependence. Since self-similarity is believed to have a significant impact on network performance, understanding the causes and effects of traffic self-similarity is an important problem.

In this paper, we study a mechanism that induces self-similarity in network traffic. We show that self-similar traffic can arise from a simple, high-level property of the overall system: the heavy-tailed distribution of file sizes being transferred over the network. We show that if the distribution of file sizes is heavy-tailed—meaning that the distribution behaves like a power law thus generating very large file transfers with nonnegligible probability—then the superposition of many file transfers in a client/server network environment induces self-similar traffic and this causal mechanism is robust with respect to changes in network resources (bottleneck bandwidth and buffer capacity), topology, interference from cross-traffic possessing dissimilar traffic characteristics, and changes in the distribution of file request interarrival times. Properties of the transport/network layer in the protocol stack are shown to play an important role in mediating this causal relationship.

The mechanism we propose is motivated by the ON/OFF model [24]. The ON/OFF model shows that self-similarity can arise in an idealized context—i.e., one with unbounded resources and independent traffic sources—as a result of aggregating a large number of 0/1 renewal processes whose ON or OFF periods are heavy-tailed. The success of this simple, elegant model in capturing the characteristics of measured traffic traces [24] is surprising given that it ignores nonlinearities arising from the interaction of traffic sources contending for network resources, which in real networks, can be as complicated as the feedback congestion control algorithm of TCP Reno. To apply the framework of the ON/OFF model to real networks, it is necessary to understand whether the model's limitations affect its usefulness and how these limitations manifest themselves in practice.

In this paper, we show that in a "realistic" client/server network environment—i.e., one with bounded resources leading to the coupling of multiple traffic sources contending for shared resources—the degree to which file sizes are heavy-tailed directly determines the degree of traffic self-similarity. Specifically, measuring self-similarity via the Hurst parameter H and file size distribution by its power-law exponent α (their definitions are given later), we show that there is a nearly linear relationship between H and α over a wide range of network conditions and when subject to the influence of the protocol stack. This mechanism gives a particularly simple explanation of why self-similar network traffic may be observed in many diverse contexts.

We discuss a traffic shaping effect of TCP that helps explain the modulating influence of the protocol stack. We find that the presence of self-similarity at the link layer depends on whether reliable and flow-controlled communication is employed at the transport layer. In the absence of reliability and flow control mechanisms such as when a UDP-based transport protocol is used, much of the self-similarity of the downstream traffic is destroyed when compared to the upstream traffic. The resulting traffic, while still bursty at short ranges, shows significantly less long-range correlation structure. In contrast, when TCP (Reno, Tahoe, or Vegas) is employed, the long-range dependence structure induced by heavy-tailed file size distributions is preserved and transferred to the link-layer, manifesting itself as scale-invariant burstiness.

We conclude with a discussion of the effect of self-similarity on network performance. We find that in an UDP-based non-flow-controlled[1] environment, as self-similarity is increased, performance declines drastically as measured by packet loss rate and mean queue length. If reliable communication via TCP is used, however, packet loss, retransmission rate, and file transmission time decline gracefully (roughly linearly) as a function of H. The exception is mean queue length, which shows the same superlinear increase as in the unreliable non-flow-controlled case. This graceful decline in TCP's performance under self-similar loads comes at a cost: a disproportionate increase in the consumption of buffer space. The sensitive dependence of mean queue length on self-similarity is consistent with previous work [13] showing that queue length distribution decays more slowly for long-range dependent sources than for short-range dependent sources. The aforementioned traffic-shaping effect of flow-controlled, reliable transport shaping a large file transfer into an on-average "thin" packet train (stretching-in-time effect) suggests, in part, why the ON/OFF model has been so successful despite its limitations—a principal effect of interaction among traffic sources in an internetworked environment seems to lie in the generation of lengthy packet trains wherein the correlation structure inherent in heavy-tailed file size distributions is sufficiently preserved.

The rest of this paper is organized as follows. In the next two sections, we discuss related work, the network model, and the simulation set-up. This is followed by the main section which explores the effect of file size distribution on traffic self-similarity, including the role of the protocol stack, heavy-tailed versus non-heavy-tailed interarrival time distribution, resource variations, and traffic mixing. We conclude with a discussion of the effect of traffic self-similarity from a performance evaluation perspective, showing its quantitative and qualitative effects with respect to performance measures when both the degree of self-similarity and network resources are varied.

2 Related work

Since the seminal study of Leland *et al.* [12] which set the groundwork for considering self-similar network traffic as an important modeling and performance evaluation problem, a string of work has appeared dealing with various aspects of traffic self-similarity [1, 2, 7, 9, 10, 13, 14, 19, 24]. The research avenues may be broadly classified into two categories.

In the first category [7, 9, 10, 12, 19, 24], traffic traces from physical network measurements are employed to identify the presence of scale-invariant burstiness, and models are constructed capable of generating synthetic traffic with matching characteristics. These papers show that long-range dependence is an ubiquitous phenomenon encompassing both local-area and wide-area network traffic.

In the second category are papers that have evaluated the effect of self-similar traffic on idealized or simplified networks [1, 2, 13, 14]. They show that long-range dependent traffic is likely to degrade performance and a principal result is the observation that queue length distribution under self-similar traffic decays much more slowly than with short-range-dependent sources (e.g., Poisson).

[1] H-estimates and performance results when an open-loop flow control is active can be found in [17].

Our work is an extension of the line of research in the first category where we investigate causal mechanisms that may be at play in real networks responsible for generating the self-similarity phenomena observed in diverse networking contexts. The relationship between file sizes and self-similar traffic was suggested by the work described in [7] which showed that self-similarity in World Wide Web traffic might arise due to the heavy-tailed distribution of file sizes present in the Web.

An important question is whether file size distributions in practice are in fact typically heavy-tailed, and whether file size access patterns can be modeled as randomly sampling from such distributions. Previous measurement-based studies of file systems have recognized that file size distributions possess long tails, but they have not explicitly examined the tails for power-law behavior [21, 22, 15, 4, 20]. In [7], it is shown that the size distribution of files found in the World Wide Web appears to be heavy-tailed with α approximately equal to 1, which stands in general agreement with measurements reported in [3]. In [6], the authors show that the sizes of reads and writes to an NFS server appear to show power-law behavior. In [19], it was found that the upper tail of the distribution of data bytes in FTP bursts was well fit to a Pareto distribution with $0.9 \leq \alpha \leq 1.1$. A general study of Unix file systems has found distributions that appear to be approximately power-law [11].

3 Network model and simulation set-up

3.1 Network model

The network is given by a directed graph consisting of n nodes and m links. Each output link has a buffer, link bandwidth, and latency associated with it. A node v_i ($i = 1, 2, \ldots, n$) is a *server node* if it has a probability density function $p_i(X)$ where $X \geq 0$ is a random variable denoting file 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 time be also a server) 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 *interarrival time* (or *idle time distribution*) which is used in determining the time of 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 made by client v_i is scheduled at time $T_k + Y$ where Y has distribution d_i. Requests from individual clients are directed to servers randomly (independently and uniformly) over the set of servers. In unreliable 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. This network configuration is used for most of the experiments reported below. We will refer to the total traffic arriving at G_2 from servers as *upstream* traffic and the traffic from G_2 to G_1 as *downstream* traffic.

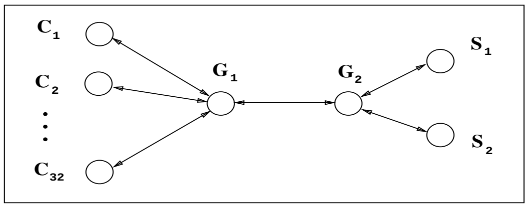

Figure 1. Network configuration.

A file is completely determined by its size X and it is split into $\lceil X/M \rceil$ packets where M is the maximum segment size (1kB for the results shown in the paper). The segments are routed through a packet-switched internetwork with packets being dropped at bottleneck nodes in case of buffer overflow. The dynamical model is given by all clients independently placing file transfer requests to servers where each request is completely determined by the file size.

3.2 Simulation set-up

We have used the LBNL Network Simulator (`ns`) as our simulation environment [8]. `Ns` is an event-driven simulator derived from S. Keshav's REAL network simulator supporting several flavors of TCP (in particular, TCP Reno's congestion control features—Slow Start, Congestion Avoidance, Fast Retransmit/Recovery) and router scheduling algorithms. Although not production TCP code, we have found `ns`'s emulation of TCP satisfactory for the purposes at hand.

We have modified the distributed version of `ns` to model our interactive client/server environment. This entailed, among other things, implementing our client/server nodes as separate application layer agents. A UDP-like unreliable transport protocol was added to the existing protocol suite, and an aggressive opportunistic UDP-based agent was built to service file requests when using unreliable communication. We also added a TCP Vegas module to complement the existing TCP Reno and Tahoe modules.

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 is a time series of one million data points; using such extremely long series increases the reliability of statistical measurements of self-similarity. Although most of the runs reported here were done with a 2-server/32-client bottleneck configuration (Figure 1), other configurations were tested including performance runs with the number of clients varying in the range 1–132. The bottleneck link was varied from 1.5 Mbps up to OC-3 levels, and buffer sizes were varied in the range of 1kB–128kB. Non-

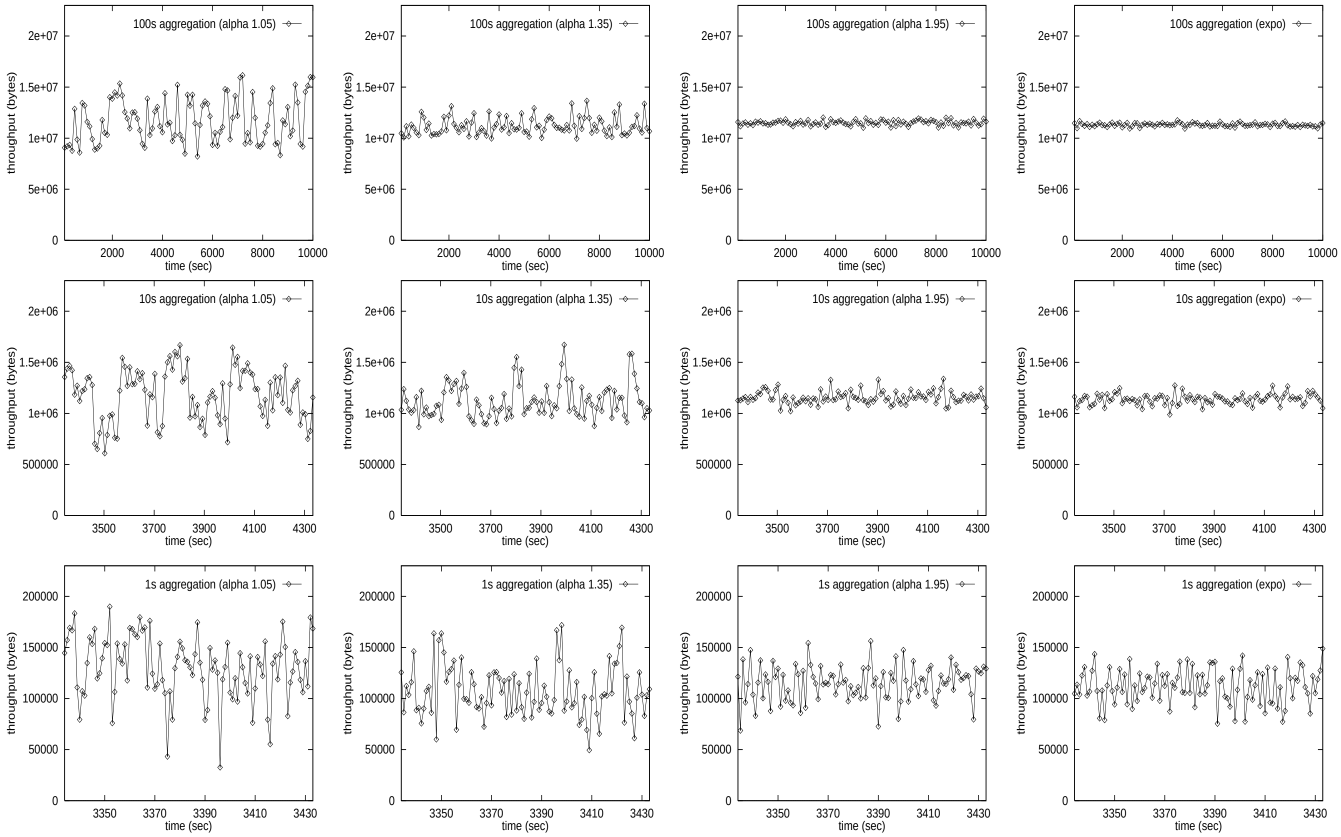

Figure 2. TCP run. Throughput as a function of file size distribution and three aggregation levels. File size distributions constitute Pareto with $\alpha = 1.05, 1.35, 1.95$, and exponential.

bottleneck links were set at 10 Mbps and the latency of each link was set to 15ms. The maximum segment size was fixed at 1kB for the runs reported here. For any reasonable assignment to bandwidth, buffer size, mean file request size, and other system parameters, it was found that by either adjusting the number of clients or the mean of the idle time distribution d_i appropriately, any intended level of network demand could be achieved.

4 File size distribution and traffic self-similarity

4.1 Heavy-tailed distributions

An important characteristic of our proposed mechanism for traffic self-similarity is that the sizes of files being transferred are drawn from a heavy-tailed distribution. A distribution is *heavy-tailed* if

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

where $0 < \alpha < 2$. That is, the asymptotic shape of the distribution follows a power law. One of the simplest heavy-tailed distributions is the *Pareto* distribution. The Pareto distribution is power-law over its entire range; its probability density function is given by

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

where $\alpha, k > 0$, and $x \geq k$. Its distribution function has the form

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

The parameter k represents the smallest possible value of the random variable.

Heavy-tailed distributions have a number of properties that are qualitatively different from distributions more commonly encountered such as the exponential or normal distribution. If $\alpha \leq 2$, the distribution has infinite variance; if $\alpha \leq 1$ then the distribution has also infinite mean. Thus, as α decreases, a large portion of the probability mass is present in the tail of the distribution. In practical terms, a random variable that follows a heavy-tailed distribution can

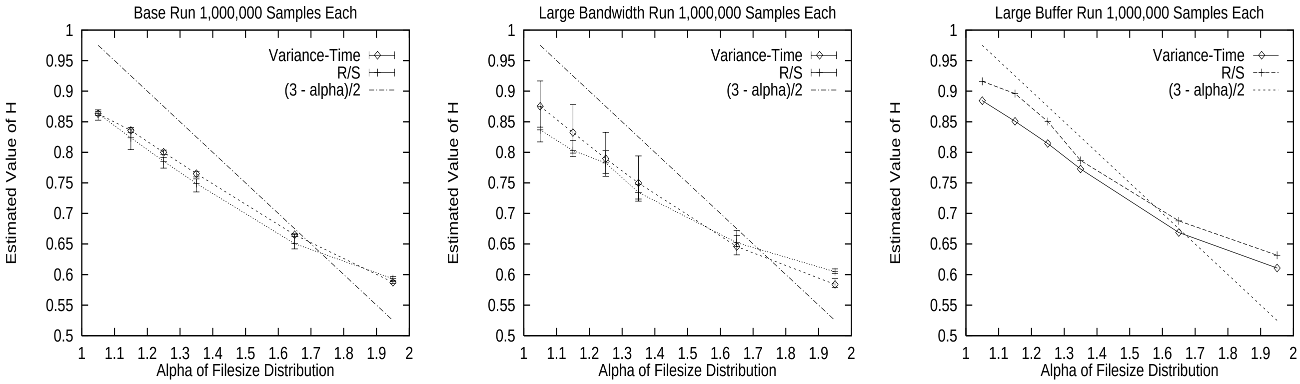

Figure 3. Hurst parameter estimates (TCP run): R/S and Variance-Time for $\alpha = 1.05, 1.35, 1.65$, and 1.95. Base run (left), large bandwith/large buffer (middle), large buffer (right)

give rise to extremely large file size requests with nonnegligible probability.

4.2 Effect of file size distribution

First, we demonstrate our central point: the interactive transfer of files whose size distribution is heavy-tailed generates self-similar traffic even when realistic network dynamics, including network resource limitations and the interaction of traffic streams, are taken into account.

Figure 2 shows graphically 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 distribution. The plots show the time series of network traffic measured at the output port of the bottleneck link from gateway G_2 to G_1 in Figure 1. This *downstream traffic* is measured in bytes per unit time where the aggregation level or time unit varies over five orders of magnitude from 10ms, 100ms, 1sec, 10sec, to 100sec. Only the top three aggregation levels are shown in Figure 2; at the lower aggregation levels traffic patterns for differing α values appear fairly similar to each other. For α close to 2, we observe a smoothing effect as the aggregation level is increased, 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 dependency. The last column depicts time series obtained by employing an exponential file size distribution at the application layer with the mean normalized so as to equal to that of the Pareto distributions. We observe that the aggregated time series between exponential and Pareto with $\alpha = 1.95$ are statistically indistinguishable.

A quantitative measure of self-similarity is obtained by using the *Hurst parameter H* which expresses the speed 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 $0 < \beta < 1$. The Hurst parameter is related to β via

$$H = 1 - \frac{\beta}{2}.$$

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

In this paper, we use two methods for testing self-similarity.[2] These methods are described more fully in [5, 23] and are the same methods used in [12]. The first method, the *variance-time plot,* is based on the slowly decaying variance of a self-similar time series. The second method, the R/S plot, uses the fact that for a self-similar data set, the *rescaled range* or R/S statistic grows according to a power law with exponent H as a function of the number of points included. Thus the plot of R/S against this number on a log-log scale has a slope which is an estimate of H. Figure 3 shows H-estimates based on variance-time and R/S methods for three different network configurations. Each plot shows H as a function of the Pareto distribution parameter for $\alpha = 1.05, 1.15, 1.25, 1.35, 1.65$ and 1.95.

Figure 3 (left) shows the results for the baseline TCP Reno case in which network bandwidth and buffer capacity are both somewhat limited (1.5 Mb/s and 6kB) resulting in a $\approx 4\%$ packet drop rate for the most bursty case

[2] A third method based on the periodogram was also used; however this method is believed to be sensitive to low frequency components in the series, which led in our case to a wide spread in the estimates; it is omitted here.

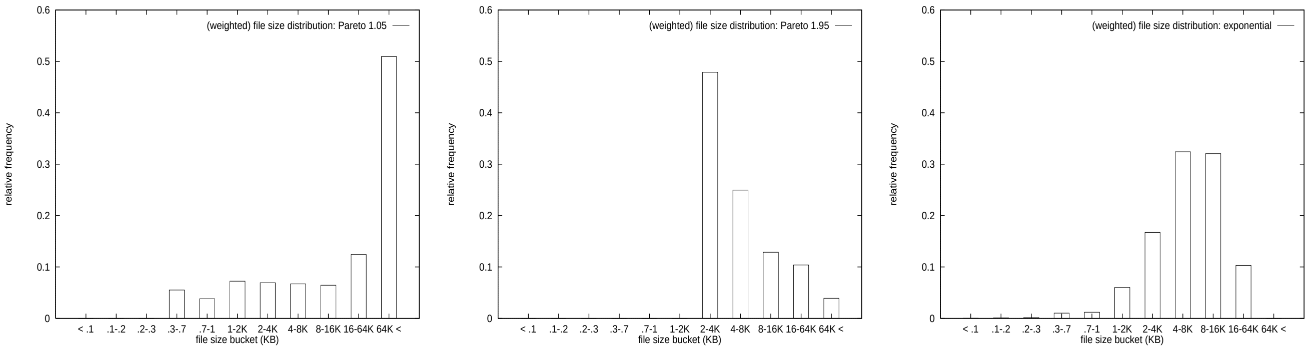

Figure 4. Relative frequency of *weighted* file size distributions obtained from three 10000 second TCP runs—Pareto with $\alpha = 1.05$ (left), with $\alpha = 1.95$ (middle), and exponential distribution (right).

($\alpha = 1.05$). The plot shows that the Hurst parameter estimates vary with file size distribution in a roughly linear manner. The $H = (3-\alpha)/2$ line shows the values of H that would be predicted by the ON/OFF model in an idealized case corresponding to a fractional Gaussian noise process. Although their overall trends are similar (nearly coinciding at $\alpha = 1.65$), the slope of the simulated system with resource limitations and reliable transport layer running TCP Reno's congestion control is consistently less than -1, with an offset below the idealized line for α close to 1 and above the line for α close to 2. Figure 3 (middle) shows similar results for the case in which there is no significant limitation in bandwidth (155 Mb/s) leading to zero packet loss. There is noticeably more spread among the estimates which we believe to be the result of more variability in the traffic patterns since traffic is less constrained by bandwidth limitations. Figure 3 (right) shows the results when bandwidth is somewhat limited, as in the baseline case, but buffer sizes at the switch are made large (64kB). Again, a roughly linear relationship between the heavy-tailedness of file size distribution (α) and self-similarity of link traffic (H) is observed.

To verify that this relationship is not due to specific characteristics of the TCP Reno protocol, we repeated our baseline simulations using TCP Tahoe and TCP Vegas. The results (presented in [18]) were essentially the same as in the TCP Reno baseline case which indicates that specific differences in implementation of TCP's flow control between Reno, Tahoe, and Vegas do not significantly affect the resulting traffic self-similarity.

Figure 4 shows the relative file size distribution of client/server conversations over the 10000 second simulation time interval, organized into file size buckets (or bins). Each file transfer request is *weighted* by its size in bytes before normalizing to yield the relative frequency. The left figure shows that the Pareto distribution with $\alpha = 1.05$ generates file size requests which are dominated by file sizes above 64kB. On the other hand, the file sizes for Pareto with $\alpha = 1.95$ and the exponential distribution (middle, right) are concentrated on file sizes below 64kB, and in spite of fine differences, their aggregated behavior (cf. Figure 2) is similar with respect to self-similarity.

We note that for the exponential distribution and the Pareto distribution with $\alpha = 1.95$, the shape of the relative frequency graph for the weighted case is analogous to the *non-weighted* (i.e., one which purely reflects the frequency of file size requests) case. However, in the case of Pareto with $\alpha = 1.05$, the shapes are "reversed" in the sense that the total number of requests are concentrated on small file sizes even though the few large file transfers end up dominating the 10000 second simulation run.

4.3 Effect of idle time distribution

All the runs thus far were obtained with an exponential idle time distribution with mean 600 msec. Figure 5 (left) and (middle) show the H-estimates of the baseline configuration when the idle time distribution is exponential with mean 0.6 sec and Pareto with $\alpha = 1.05$ and mean 1.197 sec. The file size distribution remained Pareto. As the H-estimates show, the effect of a Pareto-modeled heavy-tailed idle time distribution is to boost long-range dependence when α is close to 2, decreasing in effect as α approaches 1.

This phenomenon may be explained as follows. For file size distributions with α close to 2, the correlation structure introduced by heavy-tailed idle time is significant relative to the contribution of the file size distribution, thus increasing the degree of self-similarity as reflected by H. As α approaches 1, however, the tail weight of the file size distribution becomes the dominating term, and the contribution of idle time with respect to increasing dependency is insignificant in comparison.

Figure 5 (right) shows the Hurst parameter estimates when the file size distribution was exponential with mean

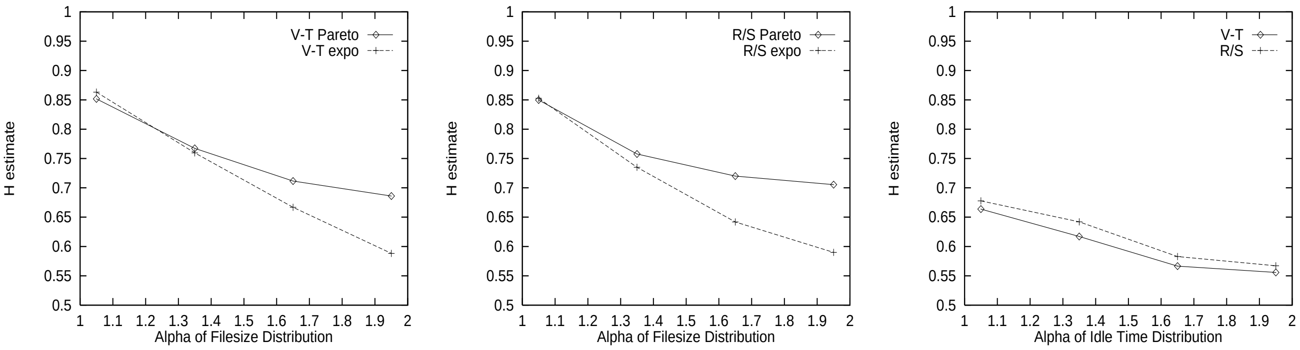

Figure 5. TCP run: Exponential idle time vs. Pareto idle time with Pareto file size distributions—Variance-Time (left), R/S (middle); Pareto idle times with exponential file size distribution (right)

4.1kB, but the idle time distribution was Pareto with α ranging between 1.05–1.95 and mean 1.197 sec at $\alpha = 1.05$. As the idle time distribution is made more heavy-tailed ($\alpha \to 1$), a positive trend in the H-estimates is clearly discernible. However, the overall level of H-values is significantly reduced from the case when the file size distribution was Pareto indicating that the file size distribution is the dominating factor in determining the self-similar characteristics of network traffic.

4.4 Effect of traffic mixing

Figure 6 shows the effect of making one of the file size distributions heavy-tailed ($\alpha = 1.05$) and the other one exponential in the 2-server system. Downstream throughput is plotted against time where the aggregation level is 100 seconds. The left plot shows the case when both servers are Pareto with $\alpha = 1.05$. The right plot shows the case when both servers have exponential file size distributions. The middle plot is the combined case, where one server has a Pareto distribution with $\alpha = 1.05$ and the other server has an exponential distribution. Figure 6 shows that the mixed case is less "bursty" than the pure Pareto case but more bursty than the pure exponential case. Performance indicators such as packet drop rate and retransmission rate (not shown here) exhibit a smooth linear degradation when transiting from one extreme to the other. That is, the presence of less bursty cross-traffic does not drastically smooth out the more bursty one, nor does the latter swallow up the smooth traffic entirely. Traffic mixing was applied to all combination pairs for $\alpha = 1.05, 1.35, 1.65, 1.95$ keeping one server fixed at $\alpha = 1.05$. The H-values for the three cases shown are 0.86, 0.81, and 0.54, respectively.

4.5 Effect of network topology

Figure 7 shows a variation in network topology from the base configuration (Figure 1) in which the 32 clients are organized in a caterpillar graph with 4 articulation points (gateways G_3, G_4, G_5, G_6), each containing 8 clients. The traffic volume intensifies as we progress from gateway G_6 to G_2 due to the increased multiplexing effect. Link traffic

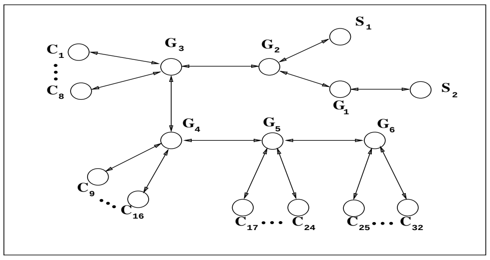

Figure 7. Variation in network topology.

was measured at the bottleneck link between G_3 and G_2 which was set at 1.544Mbps. All other links were set at 10Mbps. The Hurst parameter estimates for various values of α (not shown here) indicate that for both V-T and R/S, the degree of self-similarity measured across both topologies is almost the same.

4.6 Effect of the protocol stack

In this subsection, we explore the role of the protocol stack with respect to its effect on traffic self-similarity. We concentrate on the functionality of the transport layer and its modulating influence on the characteristics of downstream traffic via its two end-to-end mechanisms: reliable packet transport and congestion control.

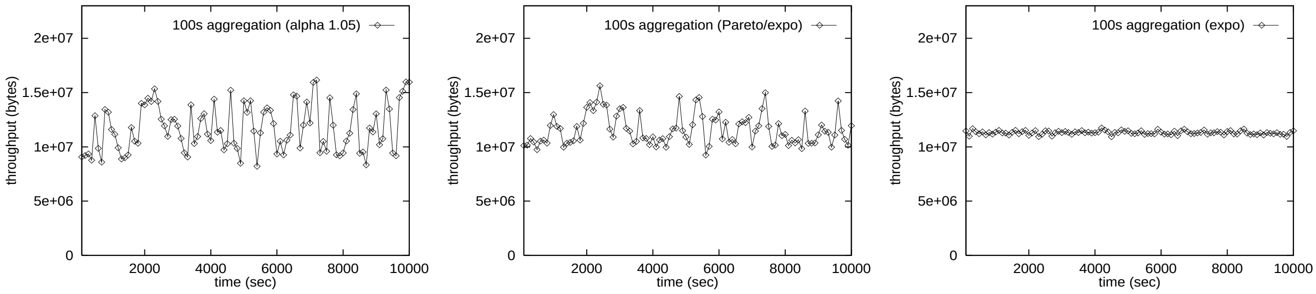

Figure 6. Traffic mixing effect for two file size distributions Pareto $\alpha = 1.05$ and exponential at 100 second aggregation level: Both servers are Pareto (left); one server is Pareto, the other one is exponential (middle); both servers are exponential (right).

4.6.1 Unreliable communication and erosion of long-range dependency

Figure 8 shows the Hurst parameter estimates for a 32-client/2-server system with exponential idle time distribution and Pareto file size distributions for $\alpha = 1.05, 1.35, 1.65$, and 1.95. In these simulations, communication is unreliable; they use a UDP-based transport protocol which is driven by an extremely greedy application whose output rate, upon receiving a client request, was essentially only bounded by the local physical link bandwidth. (The flow-controlled case is described in [17].) The H-estimates show that as source burstiness is increased, the estimated Hurst parameter of the down-stream traffic decreases relative to its value in the upstream traffic.

Another interesting point is the already low Hurst estimate of the up-stream traffic for Pareto $\alpha = 1.05$. We believe this is due to a *stretching-in-space* effect: given an exponential idle time distribution, the extremely greedy nature of the UDP-based application encourages traffic to be maximally stretched out in space, and stretching-in-time is achieved only for very large file size requests. The concentration of its mass on a shorter time interval decreases the dependency structure at lower time scales, making the traffic less self-similar.

4.6.2 Stretching-in-time

In contrast to the unreliable non-flow-controlled communication case, reliable communication and flow-control, together, act to preserve the long-range dependence of heavy-tailed file size distributions, facilitating its transfer and ultimate realization as self-similar link traffic. Efficiency dictates that file transmissions, including retransmission of lost packets, complete in a short amount of time. Subject to the limitations of congestion control in achieving optimally efficient transfer [16], this has the effect of stretching out a large file or message transfer in time into an on-average, thin packet train. This also suggests why the linear ON/OFF model may have been successful in modeling the output characteristics of a complicated nonlinear system which real networks undoubtedly are. In some sense, the effect of the unaccounted-for nonlinearity is reflected back as a stretching-in-time effect, thus conforming to the model's original suppositions. The effect of bandwidth and buffer capacity on traffic self-similarity are omitted here for brevity. They can be found in the full paper [18].

4.7 Network performance

In this section, we present a brief summary of performance results evaluating the effects of self-similarity. A comprehensive study of the performance implications of self-similarity including quality-of-service (QoS) issues, resource trade-offs, and performance comparisons between TCP Reno, Tahoe, and Vegas can be found in [17].

4.7.1 Performance evaluation under reliable communication

We evaluated network performance when both traffic self-similarity (α of Pareto file size distribution) and network resources (bottleneck bandwidth and buffer capacity) were varied. Figure 9 (left) shows packet loss rate as a function of α for buffer sizes in the range 2kB–128kB. We observe a gradual increase in the packet loss rate as α approaches 1, the flatness of the curve increasing as buffer capacity is decreased. The latter is due to an overextension of buffer capacity whereby the burstiness associated with $\alpha = 1.95$ traffic is already high enough to cause significant packet drops. The added burstiness associated with highly self-similar traffic ($\alpha = 1.05$) bears little effect. The same gradual behavior is also observed for packet retransmission and throughput (not shown here).

Figure 9 (middle) shows mean queue length as a function of α for the same buffer range. In contrast to packet loss

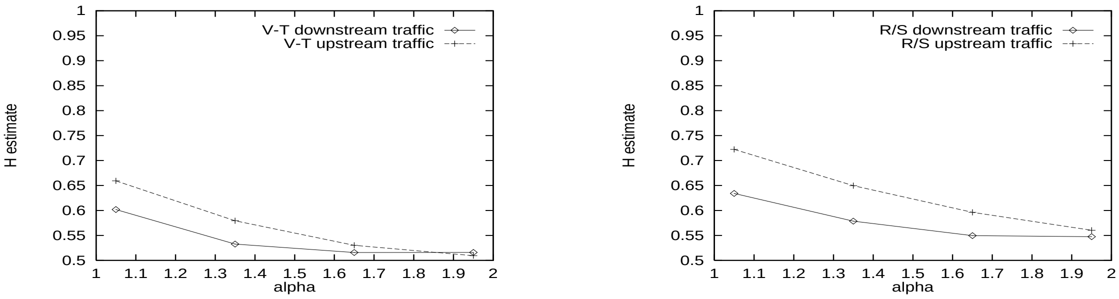

Figure 8. UDP run: Erosion of long-range dependence through excessive buffer overflow; Variance-Time (left) and R/S (right).

rate, queueing delay exhibits a superlinear dependence on self-similarity when buffer capacity is large. This is consistent with work [13, 14] showing that queue length distribution decays more slowly for long-range dependent sources than for short-range dependent sources.

Figure 9 (right) shows the queueing delay-packet loss trade-off curve for four levels of α. The individual performance points were obtained by varying buffer size while keeping bandwidth fixed at the baseline value. The performance curves show that under highly self-similar traffic conditions, the negative effects of self-similarity are significantly amplified in the packet loss rate regime below 4%. A similar trade-off relation exists for queueing delay and throughput. The effect of varying bandwidth to obtain the trade-off graphs and evaluation of the marginal benefit of network resources are discussed in [17].

4.7.2 Performance evaluation under unreliable communication

Performance evaluations under unreliable, non-flow-controlled transport yield performance results that are significantly worse than their reliable, flow-controlled counterparts. In particular, the dependence of throughput-related measures such as effective throughput, packet loss, and packet retransmission is no longer gradual—their shapes exhibit a superlinear dependence similar to the mean queue length relation in the reliable communication case. The superlinear dependence of queueing delay on the degree of self-similarity is further amplified, and so are trade-off relations between queueing delay and throughput. These results are omitted due to space constraints and can be found in [17].

5 Conclusion

In this paper, we have shown that self-similarity in network traffic can arise due to a particularly simple cause: the reliable transfer of files drawn from heavy-tailed distributions. Such a high-level explanation of the self-similarity phenomenon in network traffic is appealing because there is evidence that file systems indeed possess heavy-tailed file size distributions [7, 3, 11, 19]. It also relates a networking problem—traffic characterization—to a system-wide cause which has traditionally been considered outside the networking domain. The growth and prevalence of multimedia traffic only aggrevates the situation by facilitating the structural conditions for inducing self-similar network traffic, and our work supports recent efforts directed at managing network resources in a more integrated way ("Middleware" research) in which issues such as caching and server selection may turn out to be relevant in formulating effective solutions for congestion control.

We have shown that the relationship between file size distribution and traffic self-similarity is not significantly affected by changes in network resources, topology, traffic mixing, or the distribution of interarrival times. We have also shown that reliability and flow control mechanisms in the transport layer of the protocol stack give rise to a traffic-shaping effect that preserves self-similarity in network traffic. This helps explain why the ON/OFF model [24], in spite of ignoring traffic interactions through resource limitations and feedback control, has been successful in modeling observed traffic characteristics. The coupling between traffic sources sharing and contending for common network resources leads to a stretching-in-time effect which reflects back to the ON/OFF model by conforming, at a qualitative level, to its simplifying suppositions.

Finally, we have shown that network performance, as measured by packet loss and retransmission rate, declines smoothly as self-similarity is increased under reliable, flow-controlled packet transport. The only performance indicator exhibiting a more sensitive dependence on self-similarity was mean queue length, and this concurs with the observation that queue length distribution under self-similar traffic decays more slowly than with Poisson sources. In contrast, we showed that performance declines drastically with in-

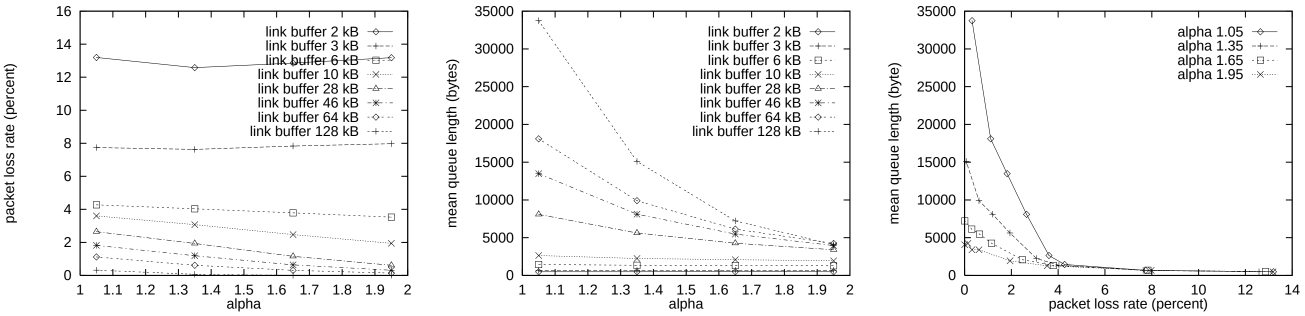

Figure 9. TCP run. Packet loss rate (left) and mean queue length (middle) as a function of α; queueing delay-packet loss trade-off curve (right).

creasing self-similarity when a UDP-like unreliable transport mechanism was employed. This gives a sense of the moderating effect of TCP on network performance in the presence of highly bursty traffic.

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] M. F. Arlitt and C. L. Williamson. Web server workload characterization: The search for invariants, May 1996. Preprint. To appear in *Proc. 1996 ACM SIGMETRICS*.

[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 13th SOSP*, pages 198–212, Pacific Grove, CA, October 1991.

[5] J. 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 SIGMETRICS '91*, pages 50–59, 1991.

[7] M. Crovella and A. Bestavros. Self-similarity in world wide web traffic: Evidence and possible causes. In *Proceedings of SIGMETRICS '96*, May 1996.

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

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

[10] 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.

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

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

[13] 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.

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

[15] J. K. Ousterhout, H. D. 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 Principles*, pages 15–24, Orcas Island, WA, December 1985.

[16] K. Park. Warp control: a dynamically stable congestion protocol and its analysis. In *Proc. ACM SIGCOMM '93*, pages 137–147, 1993.

[17] K. Park, G. Kim, and M. Crovella. On the effect of traffic self-similarity on network performance, 1996. Submitted for publication.

[18] K. Park, G. Kim, and M. Crovella. On the relationship between file sizes, transport protocols, and self-similar network traffic. Technical Report 96-016, Boston University, Computer Science Department, 1996.

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

[20] K. K. Ramakrishnan, P. Biswas, and R. Karedla. Analysis of file I/O traces in commercial computing environments. In *Proceedings of SIGMETRICS '92*, pages 78–90, June 1992.

[21] M. Satyanarayanan. A study of file sizes and functional lifetimes. In *Proceedings of the 8th SOSP*, December 1981.

[22] A. J. 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.

[23] M. S. Taqqu, V. Teverovsky, and W. Willinger. Estimators for long-range dependence: an empirical study, 1995. Preprint.

[24] 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.