Explaining World Wide Web Traffic Self-Similarity

Mark E. Crovella and Azer Bestavros
Computer Science Department
Boston University
111 Cummington St, Boston, MA 02215

{crovella,best}@cs.bu.edu

October 12, 1995

Technical Report TR-95-015
Revised

Abstract

Recently the notion of self-similarity *has been shown to apply to wide-area and local-area network traffic. In this paper we examine the mechanisms that give rise to self-similar network traffic. We present an explanation for traffic self-similarity by using a particular subset of wide area traffic: traffic due to the World Wide Web (WWW). Using an extensive set of traces of actual user executions of NCSA Mosaic, reflecting over half a million requests for WWW documents, we show evidence that WWW traffic is self-similar. Then we show that the self-similarity in such traffic can be explained based on the underlying distributions of WWW document sizes, the effects of caching and user preference in file transfer, the effect of user "think time", and the superimposition of many such transfers in a local area network. To do this we rely on empirically measured distributions both from our traces and from data independently collected at over thirty WWW sites.*

1 Introduction

Understanding the nature of network traffic is critical in order to properly design and implement computer networks and network services like the World Wide Web. Recent examinations of LAN traffic [15] and wide area network traffic [20] have challenged the commonly assumed models for network traffic, *e.g.,* the Poisson distribution. Were traffic to follow a Poisson or Markovian arrival process, it would have a characteristic burst length which would tend to be smoothed by averaging over a long enough time scale. Rather, measurements of real traffic indicate that significant traffic variance (burstiness) is present on a wide range of time scales.

Traffic that is bursty on many or all time scales can be described statistically using the notion of *self-similarity.* Self-similarity is the property we associate with fractals — the object appears the same regardless of the scale at which it is viewed. In the case of stochastic objects like timeseries, self-similarity is used in the distributional sense: when viewed at varying scales, the object's distribution remains unchanged.

Since a self-similar process has observable bursts on all time scales, it exhibits *long-range dependence;* values at any instant are typically correlated with values at all future instants. Surprisingly (given the counterintuitive aspects of long-range dependence) the self-similarity of Ethernet network traffic has been rigorously established [15]. The importance of long-range dependence in network traffic is beginning to be observed in studies such as [14], which show that packet loss and delay behavior is radically different in simulations using real traffic data rather than traditional network models.

However, the reasons behind self-similarity in network traffic have not been clearly identified. In this paper we show that in some cases, self-similarity in network traffic can be explained in terms of file system characteristics and user behavior. In the process, we trace the genesis of self-similarity in network traffic back from the traffic itself, through the actions of file transmission, caching systems, and user choice, to the high-level distributions of file sizes and user event interarrivals.

To bridge the gap between studying network traffic on one hand and high-level system characteristics on the other, we make use of two essential tools. First, to explain self-similar network traffic in terms of individual transmission lengths, we employ the mechanism introduced in [16] and described in [15]. That paper points out that self-similar traffic can be constructed by multiplexing a large number of ON/OFF sources that have ON and OFF period lengths that are heavy-tailed, as defined in Section 2.3. Such a mechanism could correspond to a network of workstations, each of which is either silent or transferring data at a constant rate.

Our second tool in bridging the gap between transmission times and high-level system characteristics is our use of the World Wide Web (WWW or Web) as an object of study. The Web provides a special opportunity for studying network traffic because it is a "closed" system: all traffic arises as the result of file transfers from an easily studied set, and user activity is easily monitored.

To study the traffic patterns of the WWW we collected reference data reflecting actual WWW use at our site. We instrumented NCSA Mosaic [8] to capture user access patterns to the Web. Since at the time of our data collection, Mosaic was by far the dominant WWW browser at our site, we were able to capture a fairly complete picture of Web traffic on our local network; our dataset consists of more than half a million user requests for document transfers, and includes detailed timing of requests and transfer lengths. In addition we surveyed a number of WWW servers to capture document size information that we used to validate assumptions made in our analysis.

The paper takes two parts. First, we establish the self-similarity of Web traffic for the busiest hours we measured. To do so we use analyses very similar to those performed in [15]. These analyses strongly support the notion that Web traffic is self-similar, at least when demand is high enough. This result in itself has implications for designers of systems that attempt to improve performance characteristics of the WWW.

Second, using our WWW traffic, user preference, and file size data, we comment on reasons why the transmission times and quiet times for any particular Web session are heavy-tailed, which is an essential characteristic of the proposed mechanism for self-similarity of traffic. In particular, we show that many characteristics of WWW use can be modelled using heavy-tailed distributions, including the distribution of transfer times, the distribution of user requests for documents, and the underlying distribution of documents sizes available in the Web. In addition, using our measurements of user inter-request times, we explore reasons for the heavy-tailed distribution of quiet times needed for self-similarity.

2 Background

2.1 Definition of Self-Similarity

For detailed discussion of self-similarity in timeseries data and the accompanying statistical tests, see [1, 27]. Our discussion in this subsection and the next closely follows those sources.

A self-similar time series has the property that when aggregated (leading to a shorter time series in which each point is the sum of multiple original points) the new series has the same autocorrelation function as the original. That is, given a stationary timeseries $X = (X_t; t = 0, 1, 2...)$, we define the m-aggregated series $X^{(m)} = (X_k^{(m)} : k = 1, 2, 3, ...)$ by summing the original series X over nonoverlapping blocks of size m. Then if X is self-similar, it has the same autocorrelation function $r(k) = E[(X_t - \mu)(X_{t+k} - \mu)]$ as the series $X^{(m)}$ for all m. Note that this means that the series is *distributionally* self-similar: the distribution of the aggregated series is the same (except for changes in scale) as that of the original.

As a result, self-similar processes show *long-range dependence.* A process with long-range dependence has an autocorrelation function $r(k) \sim k^{-\beta}$ as $k \to \infty$, where $0 < \beta < 1$. Thus the autocorrelation function of such a process decays hyperbolically (as compared to the exponential decay exhibited by traditional traffic models). Hyperbolic decay is much slower than exponential decay, and since $\beta < 1$, the sum of the autocorrelation values of such a series approaches infinity. This has a number of implications. First, the variance of n samples from such a series does not decrease as a function of n (as predicted by basic statistics for uncorrelated datasets) but rather by the value $n^{-\beta}$. Second, the power spectrum of such a series is hyperbolic, rising to infinity at frequency zero — reflecting the "infinite" influence of long-range dependence in the data.

One of the attractive features of using self-similar models for time series, when appropriate, is that the degree of self-similarity of a series is expressed using only a single parameter. The parameter expresses the speed of decay of the series' autocorrelation function. For historical reasons, the parameter used is the *Hurst* parameter $H = 1 - \beta/2$. Thus, for self-similar series, $1/2 < H < 1$. As $H \rightarrow 1$, the degree of self-similarity increases. Thus the fundamental test for self-similarity of a series reduces to the question of whether H is significantly different from $1/2$.

The effect of self-similarity in network traffic is shown in [27], which compares a self-similar series with a compound Poisson series with the same distributional characteristics. The paper shows that Poisson models for network traffic become essentially uniform when aggregated by a factor of 1,000; while actual network traffic shows no such decrease in variability over the same range of aggregation.

2.2 Statistical Tests For Self-Similarity

In this paper we use four methods to test for self-similarity. These methods are described fully in [1] and are the same methods described and used in [15]. A summary of the relative accuracy of these methods on synthetic datasets is presented in [24].

The first method, the *variance-time plot,* relies on the slowly decaying variance of a self-similar series. The variance of $X^{(m)}$ is plotted against m on a log-log plot; a straight line with slope (β) greater than -1 is indicative of self-similarity, and the parameter H is given by $H = 1 - \beta/2$. The second method, the R/S plot, uses the fact that for a self-similar dataset, 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 (n). Thus the plot of R/S against n on a log-log plot has slope which is an estimate of H. The third approach, the *periodogram* method, uses the slope of the power spectrum of the series as frequency approaches zero. On a log-log plot, the periodogram slope is a straight line with slope $\beta - 1 = 1 - 2H$ close to the origin.

While the preceding three graphical methods are useful for exposing faulty assumptions (such as non-stationarity in the dataset) they do not provide confidence intervals. The fourth method, called the *Whittle estimator* does provide a confidence interval, but has the drawback that the form of the underlying stochastic process must be supplied. The two forms that are most commonly used are fractional Gaussian noise (FGN) with parameter $1/2 < H < 1$, and Fractional ARIMA (p, d, q) with $0 < d < 1/2$ (for details see [1, 3]). These two models differ in their assumptions about the short-range dependences in the datasets; FGN assumes no short-range dependence while Fractional ARIMA can assume a fixed degree of short-range dependence.

Since we are concerned only with the long-range dependence of our datasets, we employ the Whittle estimator as follows. Each hourly dataset is aggregated at increasing levels m, and the Whittle estimator is applied to each m-aggregated dataset using the FGN model. The resulting estimates of H and confidence intervals are plotted as a function of m. This approach exploits the property that any long-range dependent process approaches FGN when aggregated to a sufficient level. As m increases short-range dependences are averaged out of the dataset; if the value of H remains relatively constant we can be confident that it measures a true underlying level of self-similarity.

2.3 Heavy-Tailed Distributions

The distributions we use in this paper have the property of being *heavy-tailed.* A distribution is heavy-tailed if

$$P[X \geq x] \sim x^{-\alpha}, \quad \text{as } x \rightarrow \infty, \ \ 0 < \alpha < 2.$$

That is, regardless of the behavior of the distribution for small values of the random variable, if the asymptotic shape of the distribution is hyperbolic, it is heavy-tailed.

The simplest heavy-tailed distribution is the *Pareto* distribution. The Pareto distribution is hyperbolic over its entire range; its probability mass function is

$$p(x) = \alpha k^{\alpha} x^{-\alpha-1}, \quad \alpha, k > 0, \quad x \geq k.$$

and its cumulative distribution function is given by

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

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

Pareto distributions have a number of properties that are qualitatively different from distributions more commonly encountered such as the exponential, normal, or Poisson distributions. If $\alpha \leq 2$, then the distribution has infinite variance; if $\alpha \leq 1$ then the distribution has infinite mean. Thus, as α decreases, an arbitrarily large portion of the probability mass may be present in the tail of the distribution. In practical terms a random variable that follows a heavy-tailed distribution can give rise to extremely large values with non-negligible probability (see [20] and [17] for details and examples).

2.3.1 Estimating Tail Weight

Our results are based on estimating the values of α for a number of empirically measured distributions, such as the lengths of World Wide Web file transmission times. To do so, we employ *log-log complementary distribution* (LLCD) plots. These are plots of the complementary cumulative distribution $\bar{F}(x) = 1 - F(x) = P[X > x]$ on log-log axes. Plotted in this way, heavy-tailed distributions have the property that

$$\frac{d \log \bar{F}(x)}{d \log x} = -\alpha, \quad x > \theta$$

for some θ. In practice we obtain an estimate for α by plotting the LLCD plot of the dataset and selecting a value for θ above which the plot appears to be linear. Then we select equally-spaced points from among the LLCD points larger than θ and estimate the slope using least-squares regression. Equally-spaced points are used because the point density varies over the range used, and the preponderance of data points near the median would otherwise unduly influence the least-squares regression.

In all our α estimates for file sizes we use $\theta = 1000$ meaning that we consider tails to be the portions of the distributions for files of 1,000 bytes or greater.

An alternative approach to estimating tail weight, used in [28], is the *Hill* estimator [10]. The Hill estimator does not give a single estimate of α, but can be used to gauge the general range of αs that are reasonable. We used the Hill estimator to check that the estimates of α obtained using the LLCD method were within range; in all cases they were.

2.3.2 Testing for Infinite Variance

There is evidence that, over their entire range, many of the distributions we study may be well characterized using *lognormal* distributions [19]. However, lognormal distributions do not have infinite variance, and hence are not heavy-tailed. In our work, we are not concerned with distributions over their entire range — only their tails. As a result we don't use goodness-of-fit tests to determine whether Pareto or lognormal distributions are better at describing our data. However, it is important to verify that our datasets exhibit the infinite variance characteristic of heavy tails. To do so we use a simple test based on the Central Limit Theorem (CLT), which states that the sum of a large number of samples from any distribution *with finite variance* will tend to be normally distributed. To test for infinite variance in our datasets we proceed as follows. First, form the m-aggregrated dataset from the original dataset for large values of m (typically in the range 10 to 1000). Next, we inspect the tail behavior of the aggregated datasets using the LLCD plot. For datasets with finite variance, the slope will increasingly decline as m increases, reflecting the underlying distribution's approximation of a normal distribution. For datasets with infinite variance, the slope will remain roughly constant with increasing m.

An example is shown in Figure 1. The figure shows the CLT test for aggregation levels of 10, 100, and 500 as applied to two synthetic datasets. On the left the dataset consists of 10,000 samples

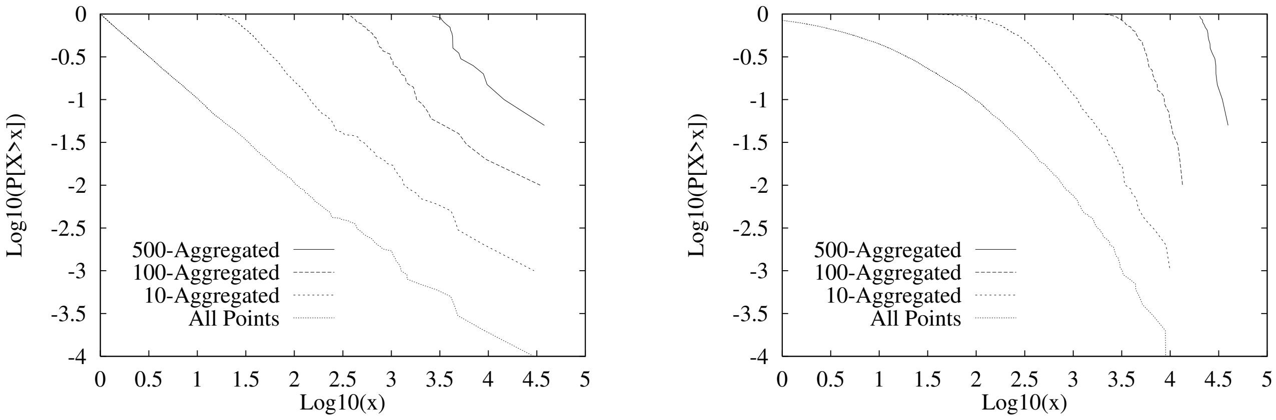

Figure 1: Comparison of CLT Test for Pareto (left) and Lognormal (right) Distributions

from a Pareto distribution with $\alpha = 1.0$. On the right the dataset consists of 10,000 samples from a lognormal distribution with $\mu = 2.0, \sigma = 2.0$. These parameters were chosen so as to make the Pareto and lognormal distributions appear approximately similar for $\log_{10}(x)$ in the range 0 to 4. In each plot the original LLCD plot for the dataset is the lowermost line; the upper lines are the LLCD plots of the aggregated datasets. Increasing aggregation level increases the average value of the points in the dataset (since the sums are not normalized by the new mean) so greater aggregation levels show up as higher lines in the plot. The figure clearly shows the qualitative difference between finite and infinite variance datasets. The Pareto dataset is characterized by parallel lines, while the lognormal dataset is characterized by lines that seem roughly convergent.

3 Related Work

The first step in understanding WWW traffic is the collection of trace data. Previous measurement studies of the Web have focused on reference patterns established based on logs of proxies [9, 23], or servers [21]. The authors in [4] captured client traces, but they concentrated on events at the user interface level in order to study browser and page design. In contrast, our goal in data collection was to acquire a complete picture of the reference behavior and timing of user accesses to the WWW. As a result, we collected a large dataset of client-based traces. A full description of our traces (which are available for anonymous FTP) is given in [citation omitted].

Previous wide-area traffic studies have studied FTP, TELNET, NNTP, and SMTP traffic [19, 20]. Our data complements those studies by providing a view of WWW (HTTP) traffic at a "stub" network. Since WWW traffic accounts for more than 25% of the traffic on the Internet and is currently growing more rapidly than any other traffic type [11], understanding the nature of WWW traffic is important and is expected to increase in importance.

The benchmark study of self-similarity in network traffic is [13, 15], and our study uses many of the same methods used in that work. However, the goal of that study was to demonstrate the self-similarity of network traffic; to do that, many large datasets taken from a multi-year span were used. Our focus is not on establishing self-similarity of network traffic (although we do so for the interesting subset of network traffic that is Web-related); instead we concentrate on examining the reasons behind self-similarity of network traffic. As a result of this different focus, we do not analyze traffic datasets for low, normal, and busy hours. Instead we focus on the four busiest hours in our logs. While these four hours are self-similar, many less-busy hours in our logs do not show self-similar characteristics. We feel that this is only the result of the traffic demand present in our logs, which is much lower than that used in [13, 15]; this belief is supported by the findings in that study, which showed that the intensity of self-similarity increases as the aggregate traffic level increases.

Our work is most similar in intent to [28]. That paper looked at network traffic at the packet level, identified the flows between individual source/destination pairs, and showed that transmission and idle times for those flows were heavy-tailed. In contrast, our paper is based on data collected at the application level rather than the network level. As a result we are able to examine the relationship between transmission times and file sizes, and are able to assess the effects of caching and user preference on these distributions. These observations allow us to build on the conclusions presented in [28] by showing that the heavy-tailed nature of transmission and idle times is not primarily a result of network protocols or user preference, but rather stems from more basic properties of information storage and processing: both file sizes and user "think times" are themselves strongly heavy-tailed.

4 Examining Web Traffic Self-Similarity

In this section we show that WWW traffic can be self-similar. To do so, we first describe how we measured WWW traffic; then we apply the statistical methods described in Section 2 to demonstrate self-similarity.

4.1 Data Collection

In order to relate traffic patterns to higher-level effects, we needed to capture aspects of user behavior as well as network demand. The approach we took to capturing both types of data simultaneously was to modify a WWW browser so as to log all user accesses to the Web. The browser we used was Mosaic, since its source was publicly available and permission has been granted for using and modifying the code for research purposes. A complete description of our data collection methods and the format of the log files is given in [6]; here we only give a high-level summary.

We modified Mosaic to record the Uniform Resource Locator (URL) [2] of each file accessed by the Mosaic user, as well as the time the file was accessed and the time required to transfer the file from its server (if necessary). For completeness, we record all URLs accessed whether they were served from Mosaic's cache or via a file transfer; however the traffic timeseries we analyzed consisted only of actual network transfers.

At the time of our study (January and February 1995) Mosaic was the WWW browser preferred by nearly all users at our site. Hence our data consists of nearly all of the WWW traffic at our site. Since the time of our study, the preferred browser has become Netscape [5], which is not available in source form. As a result, capturing an equivalent set of WWW user traces at the current time would be significantly more difficult.

The data captured consists of the sequence of WWW file requests performed during each session. Each file request is identified by its URL, and session, user, and workstation ID; associated with the request is the time stamp when the request was made, the size of the document (including the overhead of the protocol) and the object retrieval time. Timestamps were accurate to 10 ms. Thus, to provide 3 significant digits in our results, we limited our analysis to time intervals greater than or equal to 1 sec. To convert our logs to traffic time series, it was necessary to allocate the bytes transferred in each request equally into bins spanning the transfer duration. Although this process smooths out short-term variations in the traffic flow of each transfer, our restriction to time series with granularity of 1 second or more—combined with the fact that most file transfers are short—means that such smoothing has little effect on our results.

To collect our data we installed our instrumented version of Mosaic in the general computing environment at Boston University's Computer Science Department. This environment consists principally of 37 SparcStation-2 workstations connected in a local network. Each workstation has its own local disk; logs were written to the local disk and subsequently transferred to a central repository. Although we collected data from 21 November 1994 through 8 May 1995, the data used in this paper is only from the period 17 January 1995 to 28 February 1995. This period was chosen because departmental WWW usage was distinctly lower (and the pool of users less diverse) before the start of classes in early January;

Sessions	4,700
Users	591
URLs Requested	575,775
Files Transferred	130,140
Unique Files Requested	46,830
Bytes Requested	2713 MB
Bytes Transferred	1849 MB
Unique Bytes Requested	1088 MB

Table 1: Summary Statistics for Trace Data Used in This Study

and because by early March 1995, Mosaic had ceased to be the dominant browser at our site. A statistical summary of the trace data used in this study is shown in Table 1.

4.2 Self-Similarity of WWW Traffic

Using the WWW traffic data we obtained as described in the previous section, we show that WWW traffic is self-similar. First, we show that WWW traffic contains traffic bursts observable over four orders of magnitude. Second, we show that for four busy hours from our traffic logs, the Hurst parameter H for our datasets is significantly different from 1/2, indicating self-similarity.

4.2.1 Burstiness at Varying Time Scales

One of the most important aspects of self-similar traffic is that there is no characteristic size of a traffic burst; as a result, the aggregation or superimposition of many such sources does not result in a smoother traffic pattern. One way to assess this effect is by visually inspecting time series plots of traffic demands.

In Figure 2 we show four time series plots of the WWW traffic induced by our reference traces. The plots are produced by aggregating byte traffic into discrete bins of 1, 10, 100, or 1000 seconds.

The upper left plot is a complete presentation of the entire traffic time series using 1000 second (16.6 minute) bins. The diurnal cycle of network demand is clearly evident, and day to day activity shows noticeable bursts. However, even within the active portion of a single day there is significant burstiness; this is shown in the upper right plot, which uses a 100 second timescale and is taken from a typical day in the middle of the dataset. Finally, the lower left plot shows a portion of the 100 second plot, expanded to 10 second detail; and the lower right plot shows a portion of the lower left expanded to 1 second detail. These plots show significant bursts occurring at the second-to-second level.

4.2.2 Statistical Analysis

We used the four methods for assessing self-similarity described in Section 2: the variance-time plot, the rescaled range (or R/S) plot, the periodogram plot, and the Whittle estimator. We concentrated on individual hours from our traffic series, so as to provide as nearly a stationary dataset as possible.

To provide an example of these approaches, analysis of a single hour (4pm to 5pm, Thursday 5 Feb 1995) is shown in Figure 3. The figure shows plots for the three graphical methods: variance-time (upper left), rescaled range (upper right), and periodogram (lower center). The variance-time plot is linear and shows a slope that is distinctly different from -1 (which is shown for comparison); the slope is estimated using regression as -0.48, yielding an estimate for H of 0.76. The R/S plot shows an asymptotic slope that is different from 0.5 and from 1.0 (shown for comparision); it is estimated using regression as 0.75, which is also the corresponding estimate of H. The periodogram plot shows a slope of -0.66 (the regression line is shown), yielding an estimate of H as 0.83. Finally, the Whittle estimator for this dataset (not a graphical method) yields an estimate of $H = 0.82$ with a 95% confidence interval of (0.77, 0.87).

As discussed in Section 2.2, the Whittle estimator is the only method that yields confidence intervals on H, but short-range dependence in the timeseries can introduce inaccuracies in its results. These inaccuracies are minimized by m-aggregating the timeseries for successively large values of m, and looking for a value of H around which the Whittle estimator stabilizes.

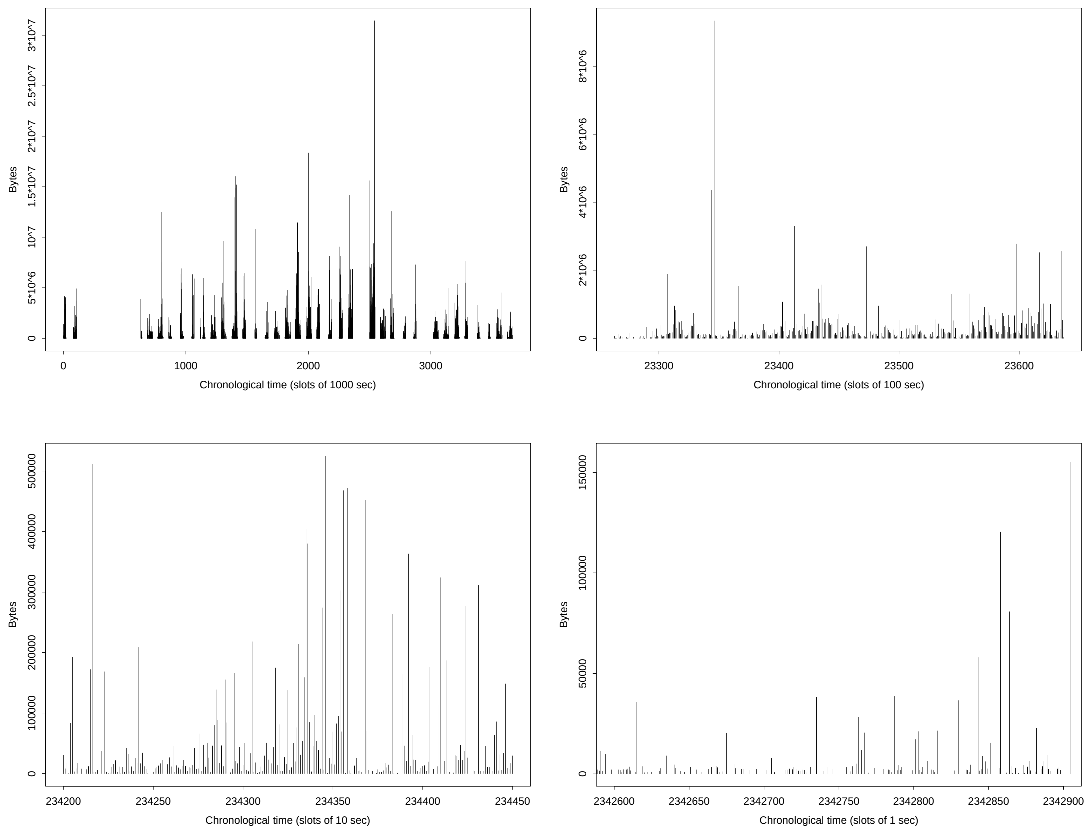

Figure 2: Traffic Bursts over Four Orders of Magnitude; Upper Left: 1000, Upper Right: 100, Lower Left: 10, and Lower Right: 1 Second Aggegrations.

The results of this method for four busy hours are shown in Figure 4. Each hour is shown in one plot, from the busiest hour in the upper left to the least busy hour in the lower right. In these figures the solid line is the value of the Whittle estimate of H as a function of the aggregation level m of the dataset. The upper and lower dotted lines are the limits of the 95% confidence interval on H. The three level lines represent the estimate of H for the unaggregated dataset as given by the variance-time, R-S, and periodogram methods.

The figure shows that for each dataset, the estimate of H stays relatively consistent as the aggregation level is increased, and that the estimates given by the three graphical methods fall well within the confidence intervals on H. The estimates of H given by these plots are in the range 0.7 to 0.8, consistent with the values for a lightly loaded network measured in [15]. Moving from the busier hours to the less-busy hours, the estimates of H seem to decline somewhat, and the variance in the estimate of H increases, which are also conclusions consistent with previous research.

Thus the results in this section show that WWW traffic at stub networks can be self-similar, when traffic demand is high enough. We expect this to be even more pronounced at routers, where traffic from a multitude of sources is aggregated. In addition, WWW traffic in stub networks is likely to become more self-similar as the demand for, and utilization of the WWW increase in the future.

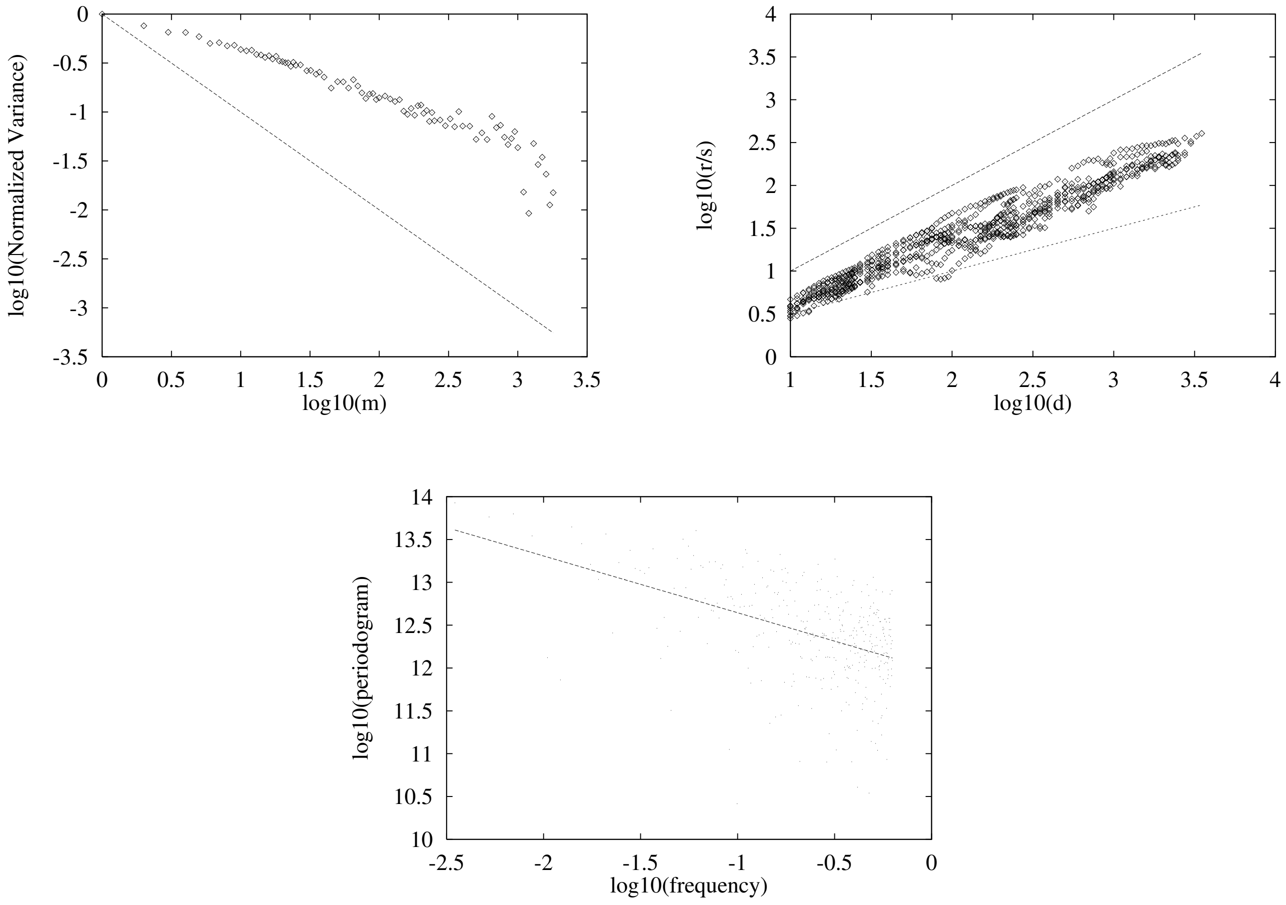

Figure 3: Graphical Analysis of a Single Hour

5 Explaining Web Traffic Self-Similarity

While the previous section showed that WWW traffic can be self-similar, it provides no explanation for this result. This section provides an explanation, based on measured characteristics of the Web.

5.1 Superimposing Heavy-Tailed Renewal Processes

Our starting point is the method of constructing self-similar processes described by Mandelbrot [16] and Taqqu and Levy [26] and summarized in [15]. A self-similar process may be constructed by superimposing many simple renewal reward processes, in which the rewards are restricted to the values 0 and 1, and in which the inter-renewal times are heavy-tailed. As described in Section 2, heavy-tailed distributions have infinite variances and the weight of their tails is determined by the parameter $\alpha < 2$. As the number of such sources grows large, such a construction yields a self-similar fractional Gaussian noise process with $H = (3 - \alpha)/2$.

This construction can be visualized as follows. Consider a large number of concurrent processes that are each either ON or OFF. At any point in time, the value of the time series is the number of processes in the ON state. If the distribution of ON and OFF times for each process is heavy-tailed, then the time series will be self-similar. Such a model could correspond to a network of workstations, each of which is either silent or transferring data at a constant rate.

Adopting this model to explain the self-similarity of WWW traffic requires an explanation for the

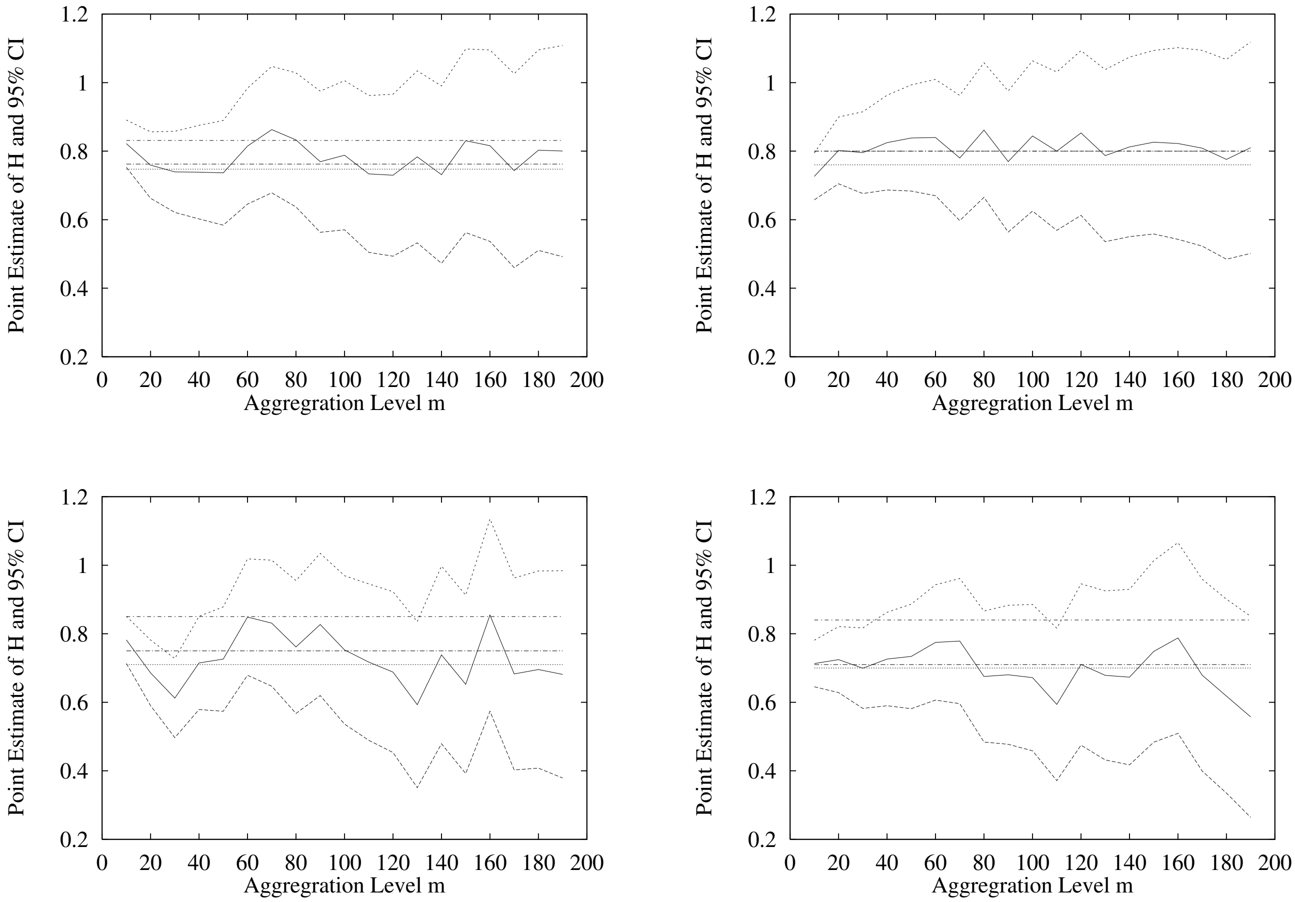

Figure 4: Whittle Estimator Applied to Aggregated Datasets

heavy-tailed distribution of ON and OFF times. In our system, ON times correspond to the transmission durations of individual WWW files, and OFF times correspond to the intervals between transmissions. So we need to ask whether WWW file transmission times and quiet times are heavy-tailed, and if so, why.

Unlike some previous wide-area traffic studies that concentrate on network-level data transfer rates, we have available application-level information such as the names and sizes of files being transferred, as well as their transmission times. In addition, our system is "closed" in the sense that all our traffic consists of Web file transfers. Thus to answer these questions we can analyze the characteristics of our client logs.

5.2 Examining Web Transmission Times

5.2.1 The Distribution of Web Transmission Times

Our first observation is that WWW file transmission times are in fact heavy-tailed. Figure 5 (left side) shows the LLCD plot of the durations of all 130140 transfers that occurred during the measurement period. The figure shows that for values greater than about -0.5, the distribution is nearly linear – indicating a hyperbolic upper tail. In addition, a least squares fit to evenly spaced data points greater than -0.5 (shown in the figure, right side, $R^2 = 0.98$) has a slope of -1.21, which yields an estimate of

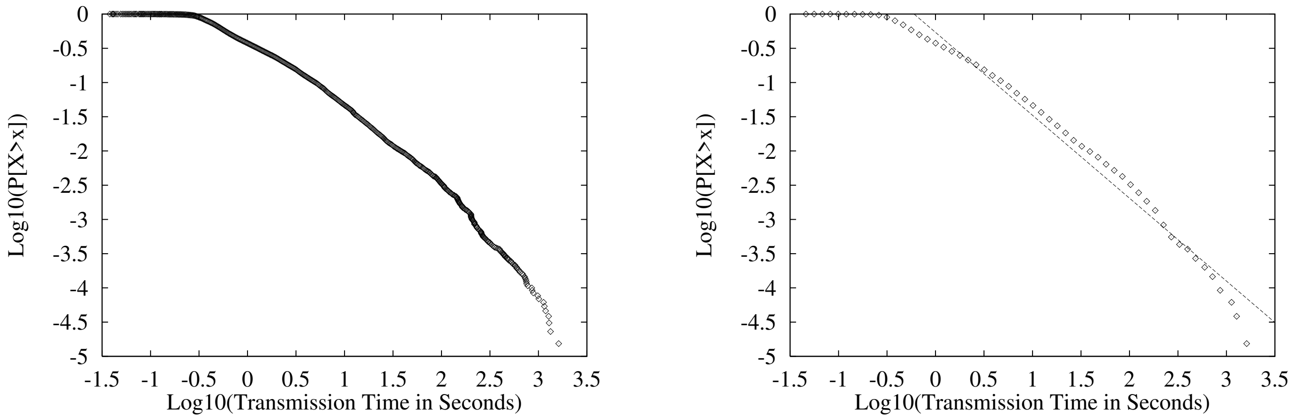

Figure 5: Log-Log Complementary Distribution of Transmission Times of WWW Files

$\alpha = 1.21$ for this distribution, with standard error $\sigma_\alpha = 0.028$.[1]

The result of aggregating a large number of ON/OFF processes in which the distribution of ON times is heavy-tailed with $\alpha = 1.21$ should yield a self-similar process with $H = 0.89$. While our data generally shows values of H in the range 0.7 to 0.8, the measured value of H should rise with increasing traffic. Therefore we expect that in the limit, a value of $H \approx 0.9$ for high levels of Web traffic is possible.

An important question is whether it is a correct model of the data to fit a line to the upper tail in Figure 5 (implying infinite variance). To answer this question we use the CLT Test as described in Section 2.3.2. The results for our dataset of transmission times is shown in Figure 6. The figure clearly shows that as we aggregate the dataset, the slope of the tail does not change appreciably. That is, under the CLT Test, transmission times behave like the Pareto distribution (left side of Figure 1) rather than the Lognormal distribution (right side of Figure 1). While tests such as the CLT Test cannot be considered a proof, we conclude that our assumption of infinite variance seems justified for this dataset.

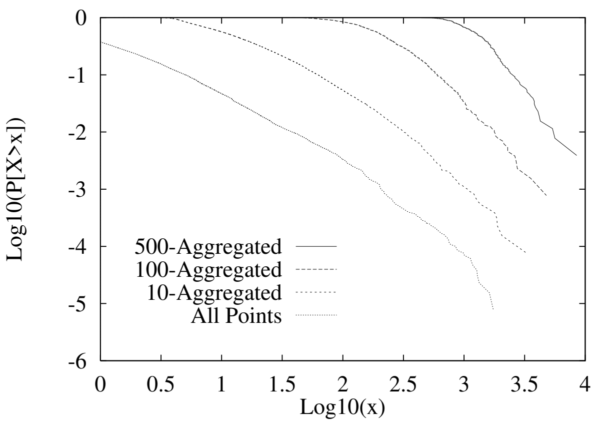

Figure 6: CLT Test For WWW File Transmission Times

[1]The σ_α values given here and below are the standard error of α as a coefficient in the least-squares fit used to estimate α.

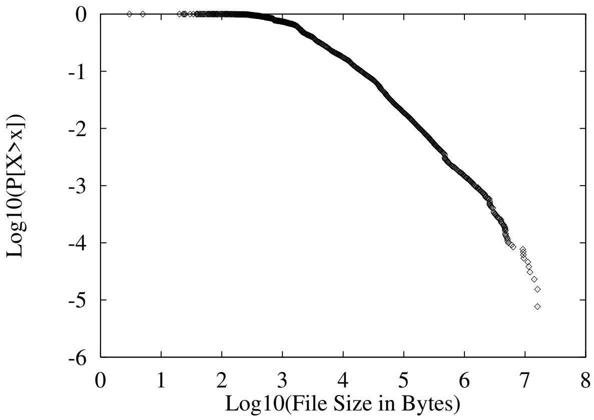

Figure 7: Log-Log Complementary Distribution of Size of WWW Files Transferred

5.2.2 Why are Web Transmission Times Heavy-Tailed?

An initial hypothesis relating traffic demand to file size might be that WWW traffic reflects the random selection of WWW files for transfer. In particular if users selected files for transfer by following links without regard to the size of the file being transferred, the size of transfers might represent essentially random samples from the distribution of WWW files. This is intuitively appealing since links as presented by Mosaic don't directly encode a notion of file length.

To test this hypothesis we studied the distribution of WWW file sizes. First, we looked at the distribution of sizes for file transfers in our logs. The results for all 130140 transfers is shown in Figure 7, which is a plot of the LLCD of transfer sizes in bytes. The Figure clearly shows that for file sizes greater than about 1000 bytes, transfer size distribution is reasonably well modeled by a Pareto distribution; the linear fit to the points for which file size is greater than 1000 yields an estimate $\alpha = 1.06$ ($R^2 = 0.98, \sigma_\alpha = 0.020$). The fact that the distribution of transfer sizes in bytes is heavier-tailed ($\alpha = 1.06$) than the distribution of transfer durations in seconds ($\alpha = 1.21$) indicates that large files are transferred somewhat faster per byte than are small files; this may be a result of the fixed overhead of TCP connection establishment and the slow-start congestion control mechanism of TCP.

Interestingly, the authors in [20] found that the upper tail of the distribution of data bytes in FTP bursts was well fit to a Pareto distribution with $0.9 \leq \alpha \leq 1.1$. Thus our results indicate that with respect to the upper-tail distribution of file sizes, Web traffic does not differ significantly from FTP traffic; however our data also allow us to comment on the reasons behind the heavy-tailed distribution of transmitted files.

There are a number of reasons why the sizes of Web files that were transferred in our logs might follow a heavy-tailed distribution. First, the distinctive distribution of transfer sizes might be determined mainly by user preferences. Second, the distribution of transfer sizes might be determined by the effect of Mosaic's caching algorithms, since transfers only reflect the user requests that miss in Mosaic's cache. Finally, the distribution of transfer sizes might be based on the underlying distribution of files available on the Web. Surprisingly, we find that the latter reason is the fundamental one in creating the heavy-tailed distribution of transfer sizes. In fact, the distribution of files found on the Web is strongly heavy-tailed, and the effects of caching and user preference transform a less heavy-tailed distribution of user requests into a set of cache misses that strongly resembles the heavy-tailed distribution of available files.

To demonstrate this fact we first show the distribution of the 46,830 *unique* files transferred in our logs. This distribution is shown in Figure 8 as a LLCD plot. The figure shows the notably hyperbolic tail for file sizes greater than 1000 bytes. On the right, the least-squares fit is shown to the sampled distribution. The measured value of α for this distribution is 1.05 ($R^2 = 0.98, \sigma_\alpha = 0.028$), not significantly different from the measured α for all files transferred; the CLT Test plot (not shown) also shows the parallel lines suggestive of infinite variance.

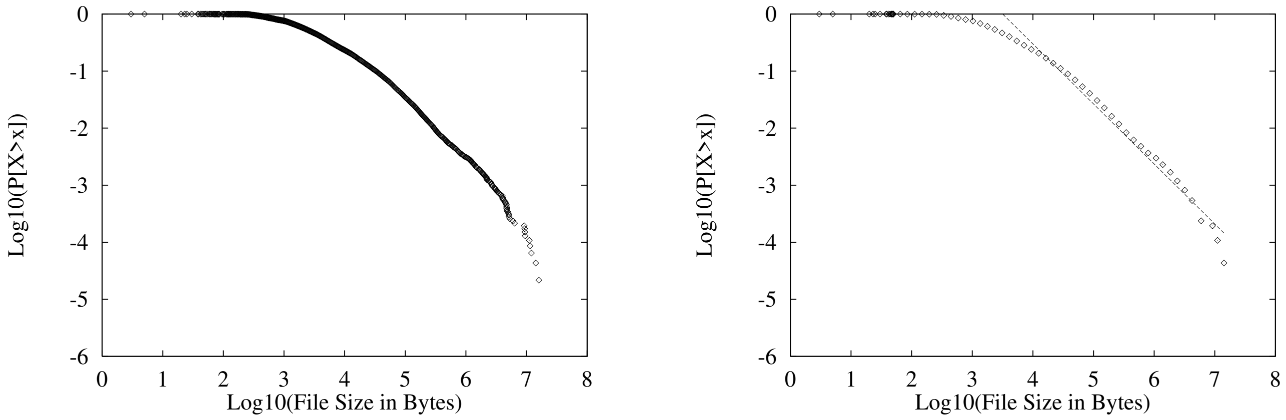

Figure 8: LLCD of Size of Unique WWW Files Transferred

We considered the possibility that the heavy-tailed distribution of unique WWW files might be an artifact of our traces and not representative of reality. For comparison purposes we surveyed 32 Web servers scattered throughout North America. These servers were chosen because they provided a usage report based on *www-stat 1.0* [22]. These usage reports provide information sufficient to determine the distribution of file sizes on the server (for files accessed during the reporting period). In each case we obtained the most recent usage reports (as of July 1995), for an entire month if possible.

Remarkably, the distribution of available files across the 32 Web servers surveyed resulted in a value of $\alpha = 1.06$ ($R^2 = 0.96, \sigma_\alpha = 0.33$) which is very nearly the same as that of files requested by our users. The distribution of available Web files is shown in Figure 9. Note that although Figures 8 and 9 appear to be very similar, they are based on completely different datasets. Thus, from a distributional standpoint, our data indicates that transferred files can be considered to be a random sample of the files that are available on Web servers. Note that since *www-stat* reports only list files that have been accessed at least once during a reporting period, our comparison is between files transferred in our traces, and the set of all files accessed at least once on the selected Web servers.

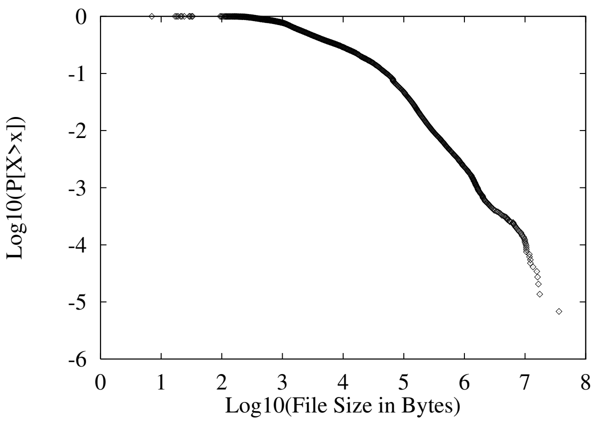

Figure 9: LLCD of File Sizes Available on 32 Web Sites

One possible explanation for the heavy-tailed nature of WWW files is that the explicit support for multimedia formats may encourage larger file sizes, thereby increasing the tail weight of distribution sizes. While we find that multimedia does increase tail weight to some degree, in fact it is not the root cause of the heavy tails. This can be seen in Figure 10.

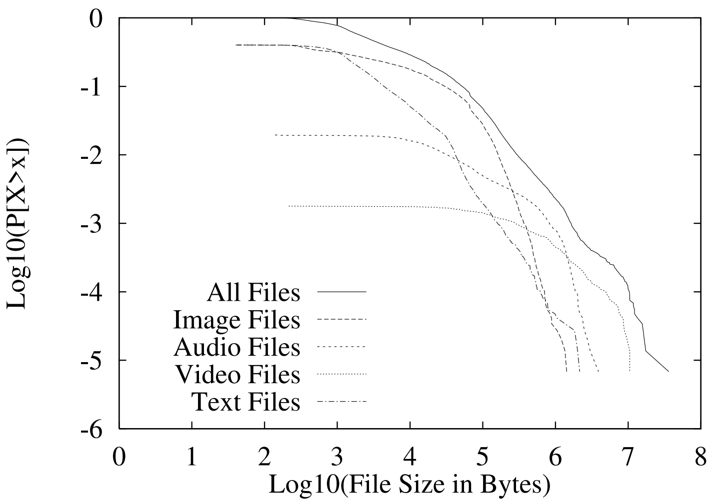

Figure 10: LLCD of File Sizes Available on 32 Web Sites, By File Type

Figure 10 was constructed by categorizing all server files into one of seven categories, based on file extension. The categories we used were: *images, text, audio, video, archives, preformatted text,* and *compressed files.* This simple categorization was able to encompass 85% of all files. From this set, the categories *images, text, audio,* and *video* accounted for 97%. The cumulative distribution of these four categories, expressed as a fraction of the total set of files, is shown in Figure 10. In the figure, the upper line is the distribution of all files, which is the same as the plot shown in Figure 9. The three intermediate lines, from upper to lower, are the components of that distribution attributable to images, audio, and video, respectively. The lowest line is the component attributable to text (HTML) alone.

The figure shows that the effect of adding multimedia files to the set of text files serves to increase the weight of the tail. However, it also shows that the distribution of text files is itself fundamentally heavy-tailed. Using least-squares fitting for the portions of the distributions in which $Log_{10}(x) > 3$, we find that for all files available $\alpha = 1.06$ (as previously mentioned) but that for the text files only, $\alpha = 1.36$ ($R^2 = 0.98, \sigma_\alpha = 0.032$). The effects of the various multimedia types are also evident from the figure. In the approximate range of 1,000 to 30,000 bytes, tail weight is primarily increased by images. In the approximate range of 30,000 to 300,000 bytes, tail weight is increased mainly by audio files. Beyond 300,000 bytes, tail weight is increased mainly by video files.

Heavy-tailed file size distributions have been noted before, particularly in filesystem studies [7, 18], although measurements of α values have been absent. As an example, we compare the distribution of Web files found in our logs with an overall distribution of files found in a survey of Unix filesystems. While there is no truly "typical" Unix file system, an aggregate picture of file sizes on over 1000 different Unix file systems is reported in [12]. In Figure 11 we compare the distribution of document sizes we found in the Web with that data. The Figure plots the two histograms on the same, log-log scale.

Surprisingly, Figure 11 shows that in the Web, there is a *stronger* preference for small files than in Unix file systems.[2] The Web favors documents in the 256 to 512 byte range, while Unix files are more commonly in the 1KB to 4KB range. More importantly, the tail of the distribution of WWW files is not nearly as heavy as the tail of the distribution of Unix files. Thus, despite the emphasis on multimedia in the Web, we conclude that Web file systems are currently more biased toward small files than are typical

[2] However, not shown in the figure is the fact that while there are virtually no Web files smaller than 100 bytes, there are a significant number of Unix files smaller than 100 bytes, including many zero- and one-byte files.

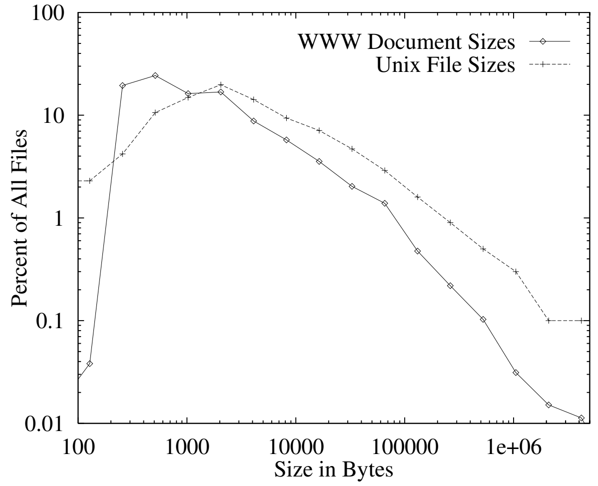

Figure 11: Comparison of Unix File Sizes with WWW File Sizes

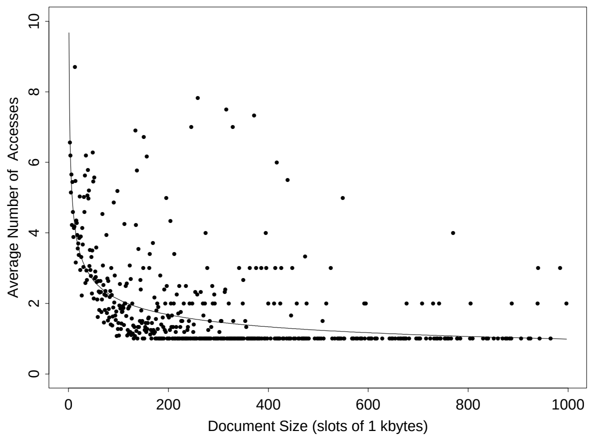

Figure 12: Distribution of Average Number of Requests by File Size.

Unix file systems.

Finally, we consider the potential effects of caching in general, and Mosaic caching in particular. To evaluate the potential effects of caching in general, we used our traces to measure the relationship between the number of times any particular document is accessed and the size of the document. Figure 12 shows a plot of the average number of times documents of a given size (in 1K bins) were referenced. The data shows that there is an inverse correlation between file size and file reuse. The line shown is a least squares fit to a log-log transform of the data ($y \sim x^{-0.33}$). This fit is statistically significant at a 99.9% level, although it only explains a small part of the total variation ($R^2 = 0.32$). This relationship suggests that systems that perform caching on WWW objects will tend to increase the tail weight of the data traffic resulting from misses in the cache as compared to the traffic without caching.

To test this hypothesis for the particular case of Mosaic, we measured the distribution of all 575,775 URLs requested in our logs, whether the URL was served from Mosaic's cache or via a network transfer. The results yield an estimate of tail weight for URLs requested of $\alpha = 1.16$ ($R^2 = 0.99, \sigma_\alpha = 0.021$). This shows that the actual set of requests made by users is not as heavy-tailed as the distribution of transferred files, but that the effect of caching is to transform the set of requests into a set of transfers that is quite similar to the set of available files.

Summary. The results in this section can be interpreted as follows. First, note that the estimates of α given in this section should not be treated as universal constants; any summary must observe the caveat that all α values apply only to our dataset. Thus, based on our data, a typical user may generate a request stream to a set of documents D_1; the size distribution of D_1 is heavy-tailed with $\alpha = 1.16$. However, users show a tendency to reuse small files preferentially, so small files will hit in the cache more often. The requests that miss in the cache (approximately 23% of D_1) form the request stream of files actually transferred through the network, set D_2. The size distribution of D_2 is more heavy tailed than D_1, with $\alpha = 1.06$. This distribution matches the distribution of the set of unique files requested (approximately 35% of D_2), set D_3, which has $\alpha = 1.05$. This presents a different view of caching; its effect is to make the stream of misses (D_2) distributionally similar to the set of unique documents (D_3) — which is reasonable since with an infinite local proxy cache, only unique documents would miss in the cache. The set of unique documents is distributionally similar to the set of documents available on the Web, set D_4, which has $\alpha = 1.06$. Finally, the set D_2 is the basis for actual network traffic T_1, whose length distribution is heavy-tailed with $\alpha = 1.21$. Thus the effect of network protocols is to lessen the tail weight of actual traffic lengths, yielding a potential Hurst parameter of $H = 0.89$, which agrees with measurements.

5.3 Examining Quiet Times

In subsection 5.2, we attributed the self-similarity of Web traffic to the superimposition of heavy-tailed ON/OFF processes, where the ON times correspond to the transmission durations of individual Web files and OFF times correspond to periods when a workstation is not receiving Web data. While ON times are the result of a positive event (transmission), OFF times are a negative event that could occur for a number of reasons. The workstation may not be receiving data because it has just finished receiving one component of a Web page (say, text containing an inlined image) and is busy interpreting, formatting and displaying it before requesting the next component (say, the inlined image). Or, the workstation may not be receiving data because the user is inspecting the results of the last transfer, or not actively using the Web at all. We will call these two conditions "Active OFF" time and "Inactive OFF" time. The difference between Active OFF time and Inactive OFF time is important in understanding the distribution of OFF times considered in this section.

To extract OFF times from our traces, we adopt the following definitions. Within each Mosaic session, let a_i be the absolute *arrival* time of URL request i. Let c_i be the absolute *completion* time of the transmission resulting from URL request i. It follows that $(c_i - a_i)$ is the random variable of ON times, whereas $(a_{i+1} - c_i)$ is the random variable of OFF times. Figure 13 shows the LLCD plot of $(a_{i+1} - c_i)$.

In contrast to the other distributions we study in this paper, Figure 13 shows that the distribution of OFF times is not well modeled by a distribution with constant α. Instead, there seem to be two regimes for α. The region from 1 ms to 1 sec forms one regime; the region from 30 sec to 3000 sec forms another regime; in between the two regions the curve is in transition.

The difference between the two regimes in Figure 13 can be explained in terms of Active OFF time versus Inactive OFF time. Active OFF times represent the time needed by the client machine to process transmitted files (*e.g.* interpret, format, and display a document component). It seems reasonable that OFF times in the range of 1 ms to 1 sec are not primarily due to users examining data, but are more likely to be strongly determined by machine processing and display time for data items that are retrieved as part of a multi-part document. For this reason the Figure shows the 1ms to 1 sec region as Active OFF time. On the other hand, it seems reasonable that very few *embedded* components would require 30 secs or more to interpret, format, and display. Thus we assume that OFF times greater than 30 secs are primarily user-determined, Inactive OFF times.

This delineation between Active and Inactive OFF times explains the two notable slopes in figure 13; furthermore, it indicates that the heavy-tailed nature of OFF times is primarily due to Inactive OFF times that result from user-induced delays, rather than from machine-induced delays or processing.

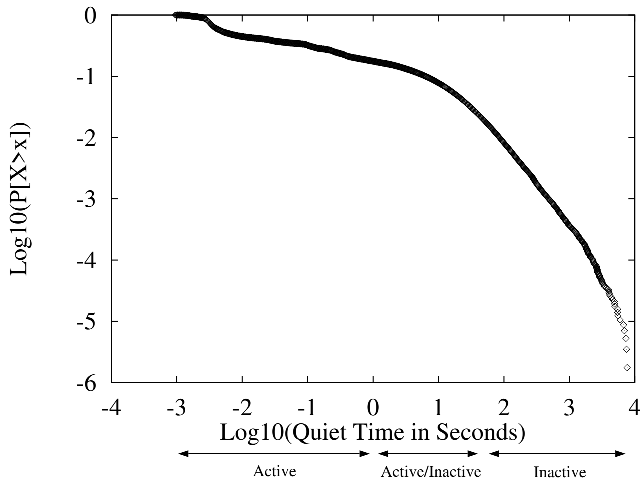

Figure 13: LLCD Plot of OFF times Showing Active and Inactive Regimes

For self-similarity via aggregation of heavy-tailed renewal processes, the important part of the distribution of OFF times is its tail. Measuring the value of α for the tail of the distribution (OFF times greater than 30 sec) yields $\alpha = 1.50$ ($R^2 = 0.99$, $\sigma_\alpha = 0.028$). Thus we see that the OFF times measured in our traces are heavy-tailed, but with much lighter tails than the distribution of ON times. In addition, we argue that the heavy-tailed nature of OFF times is a result of user "think time" rather than machine-induced delays.

6 Conclusion

In this paper we've shown that traffic due to World Wide Web transfers can be self-similar when demand is high. More importantly, we've traced the genesis of Web traffic self-similarity along two threads: first, we've shown that transmission times are heavy tailed, primarily due to the very heavy-tailed distribution of Web files. Second, we've shown that silent times are heavy-tailed, primarily due to the influence of user "think time". In addition, we've shown the distribution of user requests is lighter-tailed than the set of available files; but that the action of caching serves to make the distribution of actual files transferred similar to the more heavy-tailed distribution of available files.

Comparing the distributions of ON and OFF times, we find that the ON time distribution is much heavier tailed than the OFF time distribution. Thus we feel that the distribution of file sizes in the Web (which determine ON times) is likely the primary determiner of Web traffic self-similarity.[3]

These results seem to trace the causes of Web traffic self-similarity back to basic characteristics of information organization and retrieval. The heavy-tailed distribution of file sizes we have observed seems similar in spirit to other Pareto distributions noted in the social sciences, such as the distribution of lengths of books on library shelves, and the distribution of word lengths in sample texts (for a discussion of these effects, see [17] and citations therein). In fact, in other work [6] we show that the rule known as Zipf's Law (degree of popularity is exactly inversely proportional to rank of popularity) applies quite strongly to Web documents. The heavy-tailed distribution of user think times also seems to be a feature of human information processing (*e.g.,* [21]).

These results suggest that the self-similarity of Web traffic is not a machine-induced artifact; in particular, changes in protocol processing and document display are not likely to fundamentally remove the self-similarity of Web traffic (although some designs may reduce or increase the intensity of self-similarity for a given level of traffic demand).

[3]This conclusion is supported by recent work by Taqqu, which shows that the value of Hurst parameter H is determined by whichever distribution is heavier-tailed.[25]

A number of questions are raised by this study. First, the generalization from Web traffic to aggregated wide-area traffic is not obvious. While other authors have noted the heavy tailed distribution of Unix files [12] and of FTP transfers [20], extending our approach to wide-area traffic in general is difficult because of the many sources of traffic and determiners of traffic demand.

A second question concerns the amount of demand required to observe self-similarity in a traffic series. As the number of sources increases, the statistical confidence in judging self-similarity increases; however it isn't clear whether the important effects of self-similarity (burstiness at a wide range of scales and the resulting impact on buffering, for example) remain even at low levels of traffic demand.

7 Acknowledgements

The authors thank Murad Taqqu of the Mathematics Department, Boston University, for many helpful discussions concerning long-range dependence and for suggesting the CLT Test. Vern Paxson read an earlier draft and made a number of comments that substantially improved the paper. Carlos Cunha instrumented Mosaic, collected the trace logs, and extracted some of the data used in this study. The authors thank the other members of the Oceans research group at Boston University (Robert Carter, Carlos Cunha, Abdelsalam Heddaya, and Sulaiman Mirdad) for many helpful discussions. The authors also thank Vadim Teverovsky of Boston University's Mathematics Department for advice and insight on long-range dependence and the statistical methods for detecting it.

References

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

[2] T. Berners-Lee, L. Masinter, and M.McCahill. Uniform resource locators. RFC 1738, `http://www.ics.uci.edu/pub/ietf/uri/rfc1738.txt`, December 1994.

[3] Peter J. Brockwell and Richard A. Davis. *Time Series: Theory and Methods*. Springer Series in Statistics. Springer-Verlag, second edition, 1991.

[4] Lara D. Catledge and James E. Pitkow. Characterizing browsing strategies in the World-Wide Web. In *Proceedings of the Third WWW Conference*, 1994.

[5] Netscape Communications Corp. Netscape Navigator software. Available from `http://www.netscape.com`.

[6] Carlos R. Cunha, Azer Bestavros, and Mark E. Crovella. Characteristics of WWW client-based traces. Technical Report BU-CS-95-010, Boston University Computer Science Department, 1995.

[7] Richard A. Floyd. Short-term file reference patterns in a UNIX environment. Technical Report 177, Computer Science Dept., U. Rochester, 1986.

[8] National Center for Supercomputing Applications. Mosaic software. Available at `ftp://ftp.ncsa.uiuc.edu/Mosaic`.

[9] Steven Glassman. A Caching Relay for the World Wide Web. In *First International Conference on the World-Wide Web*, CERN, Geneva (Switzerland), May 1994. Elsevier Science.

[10] B. M. Hill. A simple general approach to inference about the tail of a distribution. *The Annals of Statistics*, 3:1163–1174, 1975.

[11] Merit Network Inc. NSF Network statistics. Available at `ftp://nis.nsf.net/statistics/nsfnet/`, December 1994.

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

[13] W. Leland, M. Taqqu, W. Willinger, and D. Wilson. On the self-similar nature of Ethernet traffic. In *Proceedings of SIGCOMM '93*, pages 183–193, September 1993.

[14] W. E. Leland and D. V. Wilson. High time-resolution measurement and analysis of LAN traffic: Implications for LAN interconnection. In *Proceeedings of IEEE Infocomm '91*, pages 1360–1366, Bal Harbour, FL, 1991.

[15] 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.

[16] Benoit B. Mandelbrot. Long-run linearity, locally Gaussian processes, H-spectra and infinite variances. *Intern. Econom. Rev.*, 10:82–113, 1969.

[17] Benoit B. Mandelbrot. *The Fractal Geometry of Nature*. W. H. Freedman and Co., New York, 1983.

[18] John K. Ousterhout, Herve Da Costa, David Harrison, John A. Kunze, Michael Kupfer, and James G. Thompson. A trace-driven analysis of the UNIX 4.2BSD file system. Technical Report CSD-85-230, Dept. of Computer Science, University of California at Berkeley, 1985.

[19] Vern Paxson. Empirically-derived analytic models of wide-area TCP connections. *IEEE/ACM Transactions on Networking*, 2(4):316–336, August 1994.

[20] Vern Paxson and Sally Floyd. Wide-area traffic: The failure of poisson modeling. In *Proceedings of SIGCOMM '94*, 1994.

[21] James E. Pitkow and Margaret M. Recker. A Simple Yet Robust Caching Algorithm Based on Dynamic Access Patterns. In *Electronic Proc. of the 2nd WWW Conference*, 1994.

[22] Regents of the University of California. www-stat 1.0 software. Available from `http://www.ics.uci.edu/WebSoft/wwwstat/`.

[23] Jeff Sedayao. "Mosaic Will Kill My Network!" – Studying Network Traffic Patterns of Mosaic Use. In *Electronic Proceedings of the Second World Wide Web Conference '94: Mosaic and the Web*, Chicago, Illinois, October 1994.

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

[25] Murad Taqqu. Personal communication.

[26] Murad S. Taqqu and Joshua B. Levy. Using renewal processes to generate long-range dependence and high variability. In Ernst Eberlein and Murad S. Taqqu, editors, *Dependence in Probability and Statistics*, pages 73–90. Birkhauser, 1986.

[27] Walter Willinger, Murad S. Taqqu, Will E. Leland, and Daniel V. Wilson. Self-similarity in high-speed packet traffic: Analysis and modeling of Ethernet traffic measurements. *Statistical Science*, 10(1):67–85, 1995.

[28] Walter Willinger, Murad S. Taqqu, Robert Sherman, and Daniel V. Wilson. Self-similarity through high-variability: Statistical analysis of Ethernet LAN traffic at the source level. In *Proceedings of SIGCOMM '95*, pages 100–113, Boston, MA, 1995.