In *IEEE/ACM Transactions on Networking*
Vol 5, Number 6, pages 835-846, December 1997

Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes

Mark E. Crovella and Azer Bestavros
Computer Science Department
Boston University
111 Cummington St, Boston, MA 02215
{crovella,best}@cs.bu.edu

Abstract

Recently the notion of *self-similarity* has been shown to apply to wide-area and local-area network traffic. In this paper we show evidence that the subset of network traffic that is due to World Wide Web transfers can show characteristics that are consistent with self-similarity, and we present a hypothesized explanation for that self-similarity. Using a set of traces of actual user executions of NCSA Mosaic, we examine the dependence structure of WWW traffic. First, we show evidence that WWW traffic exhibits behavior that is consistent with self-similar traffic models. 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 client traces and from data independently collected at WWW servers.

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 [14] and wide area network traffic [20] have challenged the commonly assumed models for network traffic, *e.g.,* the Poisson process. 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 one type of fractal—an object whose appearance is unchanged 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 correlational structure remains unchanged. As a result, such a timeseries exhibits bursts—extended periods above the mean—at a wide range of time scales.

Since a self-similar process has observable bursts at a wide range of timescales, it can exhibit *long-range dependence;* values at any instant are typically non-negligibly positively 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 [14]. The importance of long-range dependence in network traffic is beginning to be observed in studies such as [8, 13, 18], which show that packet loss and delay behavior is radically different when simulations use either real traffic data or synthetic data that incorporates long-range dependence.

However, the reasons behind self-similarity in Internet 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 described in [30] (based on earlier work in [15] and [14]). Those papers point 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 (or Web) as an object of study. The Web provides a special opportunity for studying network traffic because its 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 Web we collected reference data reflecting actual Web use at our site. We instrumented NCSA Mosaic [10] to capture user access patterns to the Web. Since at the time of our data collection, Mosaic was by far the dominant Web 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 Web servers to capture document size information that we used to compare the access patterns of clients with the access patterns seen at servers.

The paper takes two parts. First, we consider the possibility of self-similarity of Web traffic for the busiest hours we measured. To do so we use analyses very similar to those performed in [14]. These analyses support the notion that Web traffic may show self-similar characteristics, 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 Web.

Second, using our Web 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 argue that many characteristics of Web 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.

2 Background

2.1 Definition of Self-Similarity

For detailed discussion of self-similarity in time series data and the accompanying statistical tests, see [2, 29]. Our discussion in this subsection and the next closely follows those sources.

Given a zero-mean, stationary timeseries $X = (X_t; t = 1, 2, 3, ...)$, 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 we say that X is *H-self-similar* if for all positive m, $X^{(m)}$ has the same distribution as X rescaled by m^H. That is:

$$X_t \stackrel{d}{=} m^{-H} \sum_{i=(t-1)m+1}^{tm} X_i \quad \text{for all } m \in N$$

If X is H-self-similar, it has the same autocorrelation function $r(k) = E[(X_t - \mu)(X_{t+k} - \mu)]/\sigma^2$ 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 a change in scale) as that of the original.

As a result, self-similar processes can 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 follows a power law, as compared to the exponential decay exhibited by traditional traffic models. Power-law decay is 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 the mean of n samples from such a series does not decrease proportionally to $1/n$ (as predicted by basic statistics for uncorrelated datasets) but rather decreases proportionally to $n^{-\beta}$. 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 with long-range dependence, $1/2 < H < 1$. As $H \to 1$, the degree of both self-similarity and long-range dependence increases.

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 [2] and are the same methods described and used in [14]. A summary of the relative accuracy of these methods on synthetic datasets is presented in [27].

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 $(-\beta)$ 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, and as developed in [27], they may be biased for large H. 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 [2, 4]). 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 in 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. This approach exploits the property that any long-range dependent process approaches FGN when aggregated to a sufficient level, and so should be coupled with a test of the marginal distribution of the aggregated observations to ensure that it has converged to the Normal distribution. 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. Since aggregating the series shortens it, confidence intervals will tend to grow as the aggregation level increases; however if the estimates of H appear stable as the aggregation level increases, then we consider the confidence intervals for the unaggregated dataset to be representative.

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

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

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.

Heavy-tailed 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 [16] for details and examples).

To assess the presence of heavy tails in our data, 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. To check for the presence of heavy tails in practice we form the LLCD plot and look for approximately linear behavior over a significant range (3 orders of magnitude or more) in the tail.

It is possible to form rough estimates of the shape parameter α from the LLCD plot as well. First, we inspect the LLCD plot and choose 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.[1] The proper choice of θ is made based on inspecting the LLCD plot; in this paper we identify the θ used in each case, and show the resulting fitted line used to estimate α.

Another approach we used to estimating tail weight is the *Hill* estimator [12] (described in detail in [30]). The Hill estimator uses the k largest values from a dataset to estimate the value of α for the dataset. In practice one plots the Hill estimator for increasing values of k, using only the portion of the tail that appears to exhibit power-law behavior; if the estimator settles to a consistent value, this value provides an estimate of α.

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 [11, 25], or servers [21]. The authors in [5] 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

[1] Equally-spaced points are used because the point density varies over the range used, and the preponderance of data points at small values would otherwise unduly influence the least-squares regression.

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 [6].

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 a large fraction of the traffic on the Internet,[2] understanding the nature of WWW traffic is important.

The benchmark study of self-similarity in network traffic is [14], 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 well described as 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 [14]; 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 [30]. 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 [30] and confirm observations made in [20] 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 evidence 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 assess self-similarity.

---

[2] See, for example, data at `http://www.nlanr.net/INFO/`.

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) [3] 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 analyze in this section consist 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, users have come to prefer commericial browsers which are not available in source form. As a result, capturing an equivalent set of WWW user traces at the current time would be more difficult.

The data captured consists of the sequence of WWW file requests performed during each session, where a session is one execution of NCSA Mosaic. 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 [6]—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

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

the start of classes in early January; 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 evidence consistent with the conclusion that WWW traffic is self-similar on time scales of one second and above. To do so, we show that for four busy hours from our traffic logs, the Hurst parameter H for our datasets is significantly different from 1/2.

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 1. 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

As discussed in Section 2.2, the Whittle estimator is the only method that yields confidence intervals on H, but it requires the the form of the underlying timeseries be provided. We used the Fractional Gaussian Noise model, so it is important to verify that the underlying series behaves

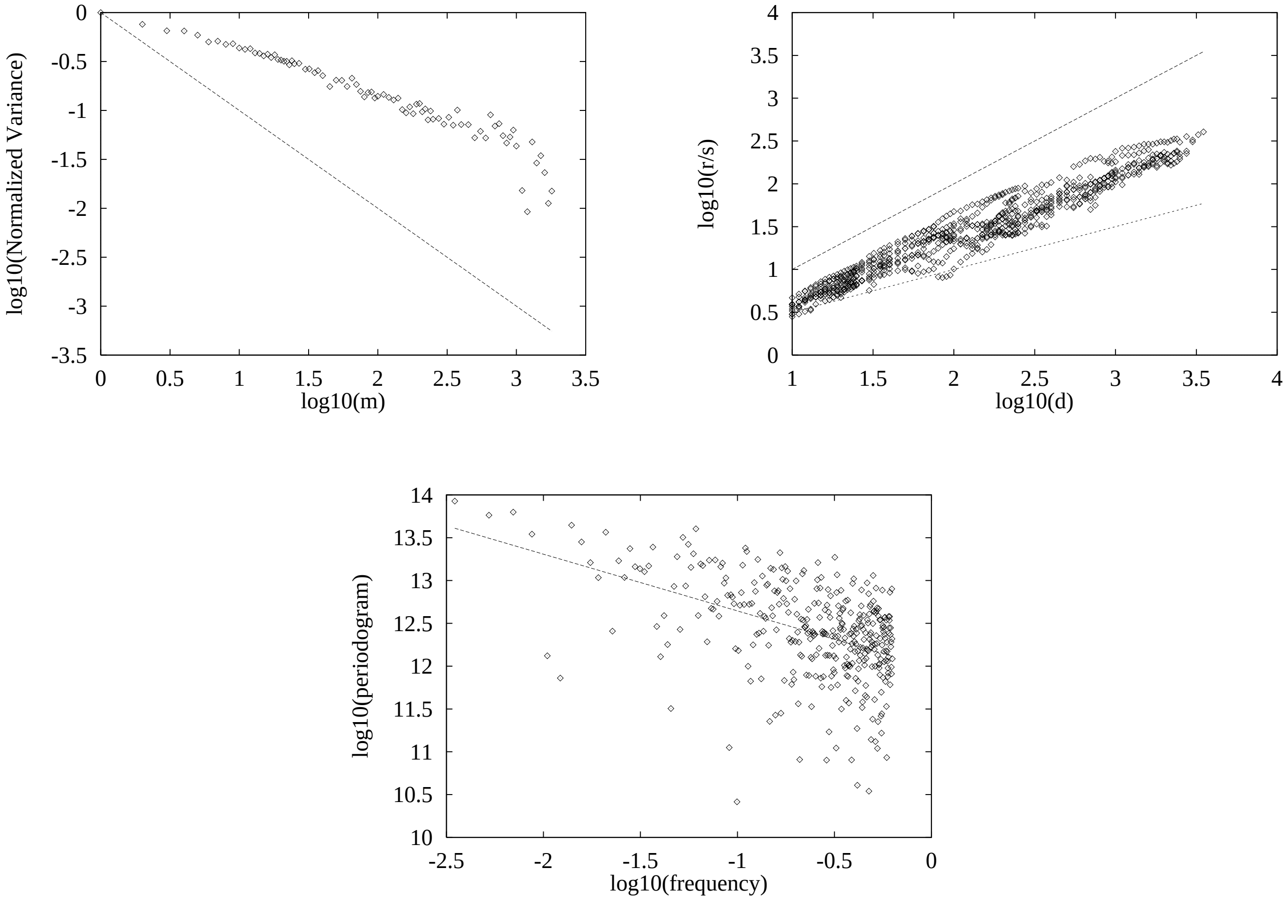


Figure 1: Graphical Analysis of a Single Hour

like FGN; namely, that is has a Normal marginal distribution and that additional short-range dependence is not present. We can test whether lack of normality or short-range dependence is biasing the Whittle estimator by m-aggregating the timeseries for successively large values of m, and determining whether the Whittle estimator remains stable, since aggregating the series will disrupt short-range correlations and tend to make the marginal distribution closer to the Normal.

The results of this method for four busy hours are shown in Figure 2. Each hour is shown in one plot, from the busiest hour (largest amount of total traffic) 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

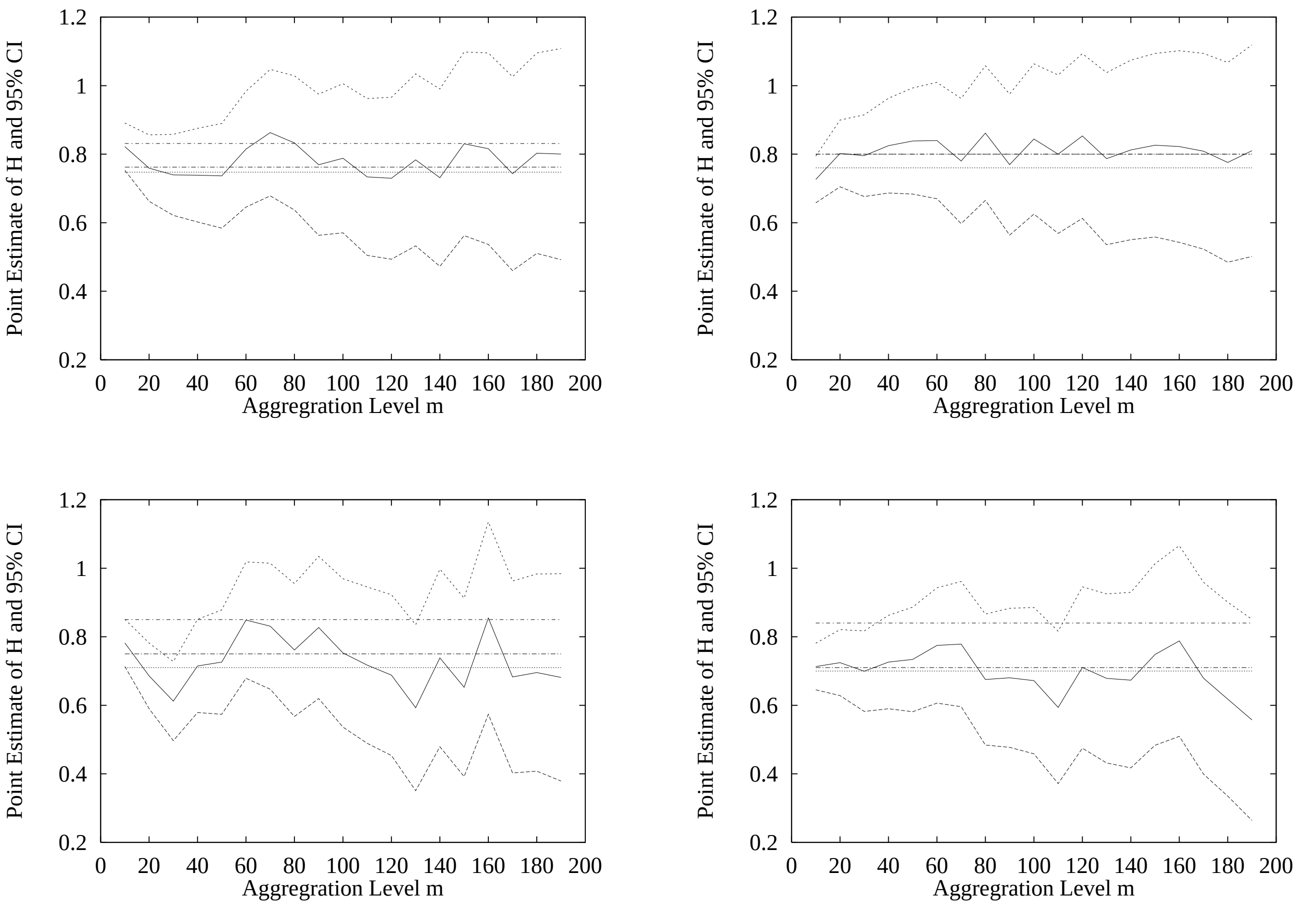


Figure 2: Whittle Estimator Applied to Aggregated Datasets

within the range of H estimates given by the Whittle estimator. 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 [14]. 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 evidence that WWW traffic at stub networks might be self-similar, when traffic demand is high enough. We expect this to be even more pronounced on backbone links, 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.

5 Explaining Web Traffic Self-Similarity

While the previous section showed evidence that Web traffic can show self-similar characteristics, it provides no explanation for this result. This section provides a possible 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 in [30] which is a refinement of work done by Mandelbrot [15] and Taqqu and Levy [28]. 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, a heavy-tailed distribution has infinite variance and the weight of its tail is determined by the parameter $\alpha < 2$.

This construction can be visualized as follows. Consider a large number of concurrent processes that are each either ON or OFF. The ON and OFF periods for each process strictly alternate, and either the distribution of ON times is heavy tailed with parameter α_1, or the distribution of OFF times is heavy tailed with parameter α_2, or both. At any point in time, the value of the time series is the number of processes in the ON state. Such a model could correspond to a network of workstations, each of which is either silent or transferring data at a constant rate. For this model, it has been shown that the result of aggregating many such sources is a self-similar fractional Gaussian noise process, with $H = (3 - \min(\alpha_1, \alpha_2))/2$ [30].

Adopting this model to explain the self-similarity of Web traffic requires an explanation for the heavy-tailed distribution of either ON or OFF times. In our system, ON times correspond to the transmission durations of individual Web files (although this is not an exact fit, since the model assumes constant transmission rate during the ON times), and OFF times correspond to the intervals between transmissions. So we need to ask whether Web file transmission times or quiet times are heavy-tailed, and if so, why.

Unlike traffic studies that concentrate on network-level and transport-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. Thus to answer these questions we can analyze the characteristics of our client logs.

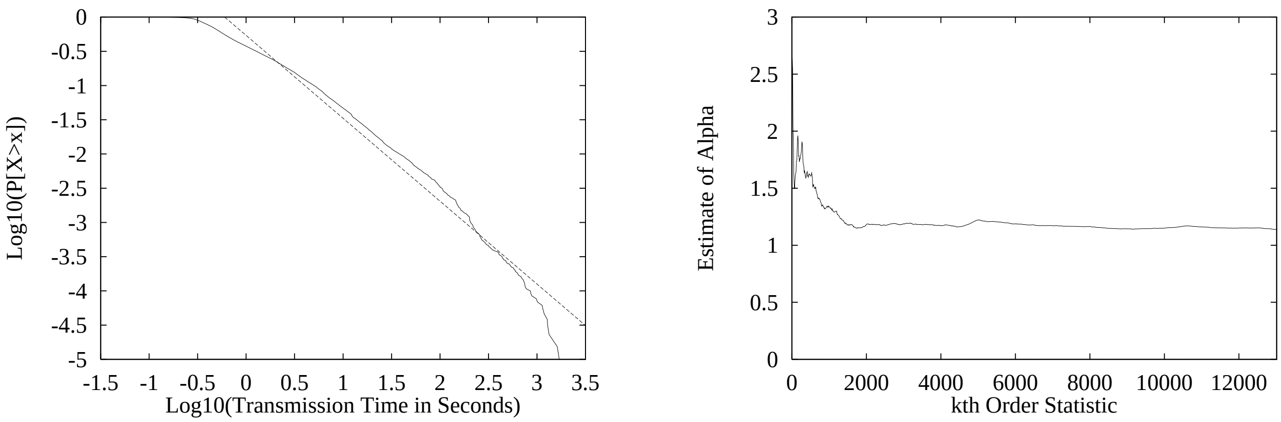


Figure 3: Log-Log Complementary Distribution (left) and Hill Estimator (right) for Transmission Times of Web Files

5.2 Examining Transmission Times

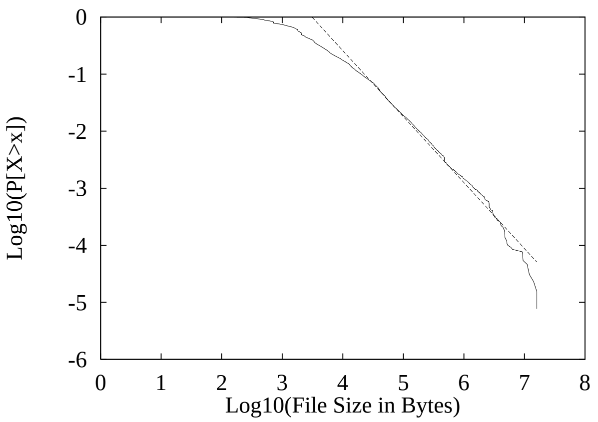
5.2.1 The Distribution of Web Transmission Times

Our first observation is that the distribution of Web file transmission times shows non-negligible probabilities over a wide range of file sizes. Figure 3 (left side) shows the LLCD plot of the durations of all 130,140 transfers that occurred during the measurement period. The shape of the upper tail on this plot, while not strictly linear, shows only a slight downward trend over almost four orders of magnitude. This is evidence of very high variance (though perhaps not infinite) in the underlying distribution.

From this plot is not clear whether actual ON times in the Web would show heavy tails, because our assumption equating file transfer durations with actual ON times is an oversimplification (*e.g.*, the pattern of packet arrivals during file transfers may show large gaps). However, if we hypothesize that the underlying distribution is heavy tailed, then this property would seem to be present for values greater than about 0.5, which corresponds roughly to largest 10% of all transmissions ($\log_{10}(P[X > x]) < -1$).

A least squares fit to evenly spaced data points greater than -0.5 ($R^2 = 0.98$) has a slope of -1.21, which yields $\hat{\alpha} = 1.21$. The right side of Figure 3 shows the value of the Hill estimator for varying k, again restricted to the upper 10% tail. The Hill plot shows that the estimator seems to settle to a relatively constant estimate, consistent with $\hat{\alpha} \approx 1.2$.

Thus, although this dataset does not show conclusive evidence of infinite variance, it is suggestive of a very high or infinite variance condition in the underlying distribution of ON times. Note that the result of aggregating a large number of ON/OFF processes in which the distribution of ON



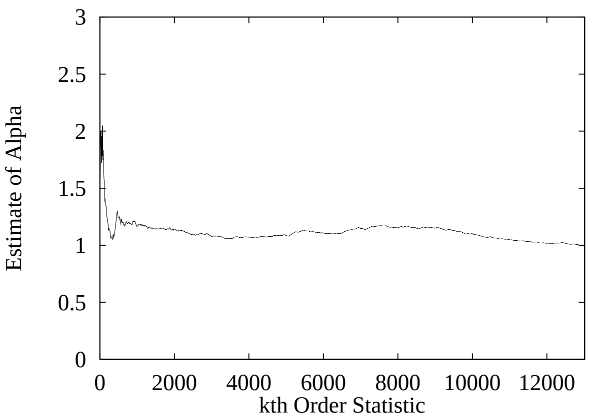


Figure 4: LLCD and Hill Estimator For Sizes of Web File Transfers

times is heavy-tailed with $\alpha = 1.2$ should yield a self-similar process with $H = 0.9$, while our data generally shows values of H in the range 0.7 to 0.8.

5.2.2 Why are Web Transmission Times Highly Variable?

To understand why transmission times exhibit high variance, we now examine size distributions of Web files themselves. First, we present the distribution of sizes for file transfers in our logs. The results for all 130,140 transfers is shown in Figure 4, which is a plot of the LLCD and the Hill estimator for the set of transfer sizes in bytes. Again, choosing the point at which power-law behavior begins is difficult, but the figure shows that for file sizes greater than about 10,000 bytes, transfer size distribution seems reasonably well modeled by a heavy-tailed distribution. This is the range over which the Hill estimator is shown in the Figure.

A linear fit to the points for which file size is greater than 10,000 yields $\hat{\alpha} = 1.15$ ($R^2 = 0.99$). The Hill estimator shows some variability in the interval between 1 and 1.2, but its estimates seem consistent with $\hat{\alpha} \approx 1.1$.

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 transfers do 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.

An important question then is why file transfers show a heavy-tailed distribution. On one hand, it is clear that the set of files requested constitutes user "input" to the system. It's natural to assume that file requests therefore might be the primary determiner of heavy-tailed file transfers. If this were the case, then perhaps changes in user behavior might affect the heavy-tailed nature of

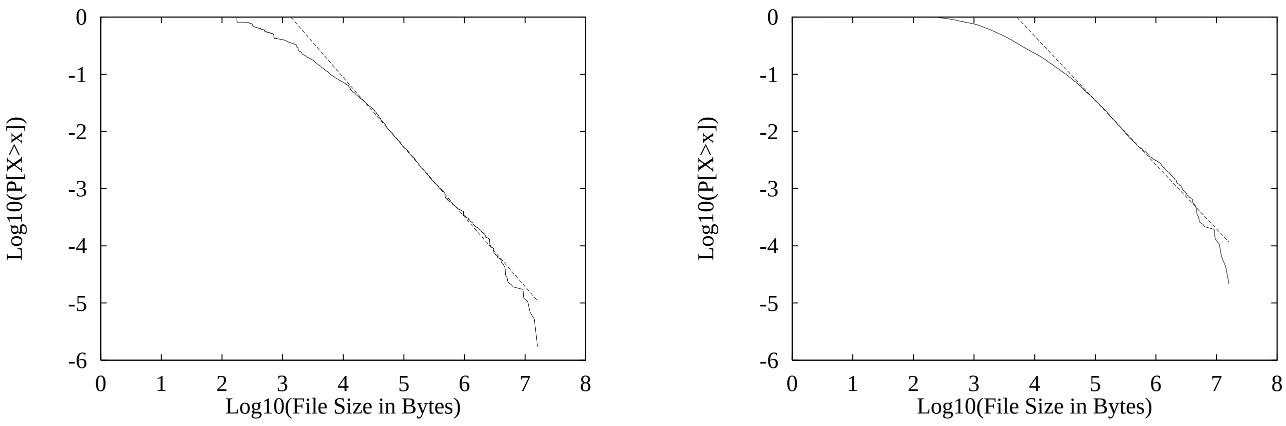


Figure 5: LLCD of File Requests (left) and Unique Files (right)

file transfers, and by implication, the self-similar properties of network traffic.

In fact, in this section we argue that the set of file requests made by users is *not* the primary determiner of the heavy-tailed nature of file transfers. Rather, file transfers seem to be more stongly determined by the set of files *available* in the Web.

To support this conclusion we present characteristics of two more datasets. First, we present the distribution of the set of all requests made by users. This set consists of 575,775 files, and contains both the requests that were satisfied via the network and the requests that were satisfied from Mosaic's cache. Second, we present the distribution of the set of *unique* files that were transferred. This set consists of 46,830 files, all different. These two distributions are shown in Figure 5.

The figure shows that both distributions appear to be heavy-tailed. For the set of file requests, we estimated the tail to start at approximately sizes of 10,000 bytes; over this range the slope of the LLCD plot yields a $\hat{\alpha}$ of about 1.22 ($R^2 = 0.99$) while the Hill estimator varies between approximately 1.0 and 1.3. For the set of unique files, we estimated the tail to start at approximately 30,000 bytes. The slope estimate over this range is $\hat{\alpha}$ of about 1.12 ($R^2 = 0.99$) and the Hill estimator over this range varies between 1.0 and 1.15.

The relationship between the three sets can be seen more clearly in Figure 6, which plots all three distributions on the same axes. This figure shows that the set of file transfers is *intermediate* in characteristics between the set of file requests and the set of unique files. For example, the median size of the set of file transfers lies between the median sizes for the sets of file requests and unique files.

The reason for this effect can be seen as the natural result of caching. If caches were infinite in size and shared by all users, the set of file transfers would be identical to the set of unique files, since each file would miss in the cache only once. If finite caches are performing well, we can

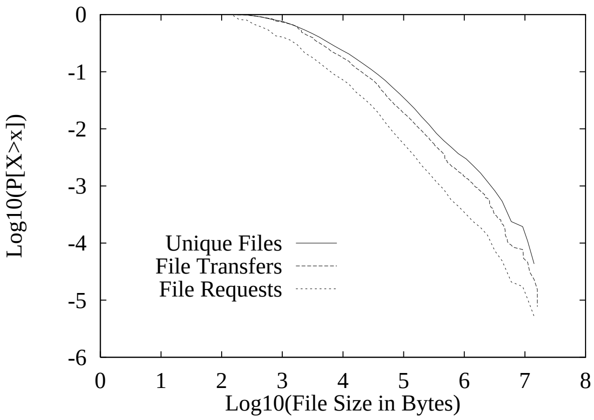


Figure 6: LLCD plots of the Different Distributions

expect that they are attempting to approximate the effects of an infinite cache. Thus the effect of caching (when it is effective) is to adjust the distributional characteristics of the set of transfers to approximate those of the set of unique files. In the case of our data, it seems that NCSA Mosaic was able to achieve a reasonable approximation of the performance of an infinite cache, despite its finite resources: from Table 1 we can calculate that NCSA Mosaic achieved a 77% hit rate $(1 - 130140/575775)$, while a cache of infinite size (shared by all users) would achieve a 92% hit rate $(1 - 46830/575775)$.

What, then, determines the distribution of the set of unique files? To help answer this question 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* [23]. 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. While this method is not a random sample of files available in the Web, it sufficed for the purpose of comparing distributional characteristics.

In fact, the distribution of all the available files present on the 32 Web servers closely matches the distribution of the set of unique files in our client traces. The two distributions are shown on the same axes in Figure 7. Although these two distributions appear very similar, they are based on completely different datasets. That is, it appears that the set of unique files can be considered, with respect to sizes, to be a sample from the set of all files available on the Web.

This argument is based on the assumption that cache management policies do not specifically exclude or include files based on their size; and that unique files are sampled without respect

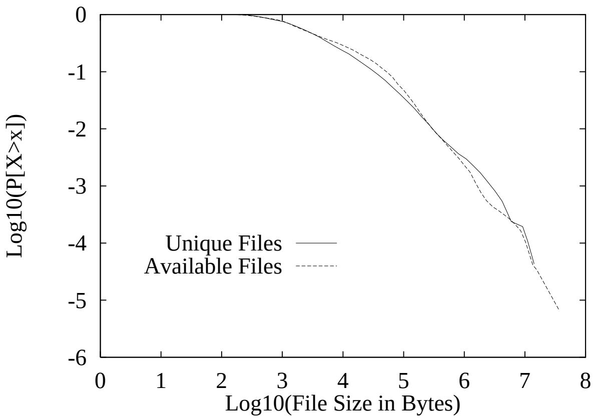


Figure 7: LLCD plots of the Unique Files and Available Files

to size from the set of available files. While these assumptions may not hold in other contexts, the data shown in Figures 6 and 7 seems to support them in this case. Thus we conclude that as long as caching is effective, the set of files available in the Web is likely to be the primary determiner of the heavy-tailed characteristics of files transferred—and that the set of requests made by users is relatively less important. This suggests that available files are of primary importance in determining actual traffic composition, and that changes in user request patterns are unlikely to result in significant changes to the self-similar nature of Web traffic.

5.2.3 Why are Available Files Heavy-Tailed?

If available files in the Web are in fact heavy-tailed, one possible explanation might be 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 8.

Figure 8 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 8. In the figure, the upper line is the distribution of all accessed files, which is the same as the available files line shown in Figure 7. The three intermediate lines are the components of that distribution attributable to images, audio, and video. The lowest line (at the extreme right-hand point) is the

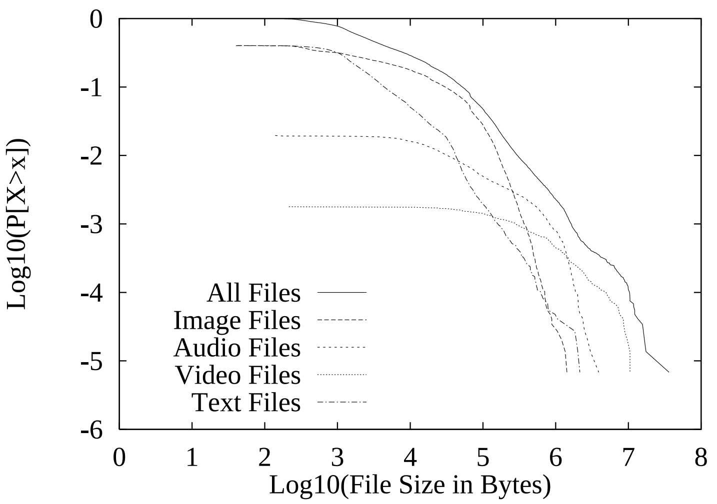


Figure 8: LLCD of File Sizes Available on 32 Web Sites, By File Type

component attributable to text (HTML) alone.

The figure shows that the effect of adding multimedia files to the set of text files serves to translate the tail to the right. However, it also suggests that the distribution of text files may itself be heavy-tailed. Using least-squares fitting for the portions of the distributions in which $\log_{10}(x) > 4$, we find that for all files available $\hat{\alpha} = 1.27$ but that for the text files only, $\hat{\alpha} = 1.59$. 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.

The fact that file size distributions have very long tails has been noted before, particularly in filesystem studies [1, 9, 17, 22, 24, 26], however they have not explicitly examined the tails for power-law behavior and 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 was collected by Irlam in 1993.[3] In Figure 9 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 9 shows that in our Web data, there is a *stronger* preference for small files than in Unix file systems.[4] The Web favors documents in the 256 to 512 byte range, while Unix

[3]This data is available from `http://www.base.com/gordoni/ufs93.html`.

[4]However, not shown in the figure is the fact that while there are virtually no Web files smaller than 100 bytes,

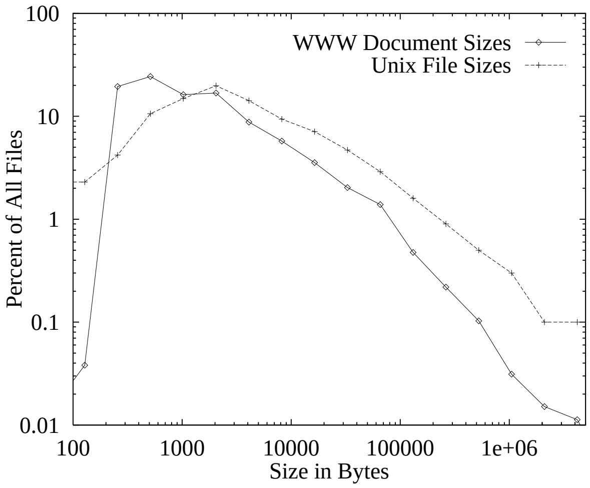


Figure 9: Comparison of Unix File Sizes with Web File Sizes

files are more commonly in the 1KB to 4KB range. More importantly, the tail of the distribution of Web 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 Unix file systems.

In conclusion, these observations seem to show that heavy-tailed size distributions are not uncommon in various data storage systems. It seems that the possibility of very large file sizes may be non-negligible in a wide range of contexts; and that in particular this effect is of central importance in understanding the genesis of self-similar traffic in the Web.

5.3 Examining Quiet Times

In subsection 5.1, 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

there are a significant number of Unix files smaller than 100 bytes, including many zero- and one-byte files.

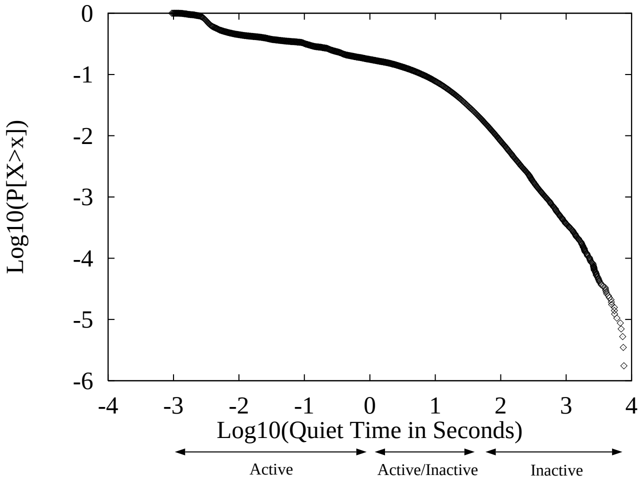


Figure 10: LLCD Plot of OFF times Showing Active and Inactive Regimes

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 (whose distribution was shown in Figure 3) whereas $(a_{i+1} - c_i)$ is the random variable of OFF times. Figure 10 shows the LLCD plot of $(a_{i+1} - c_i)$.

In contrast to the other distributions we study in this paper, Figure 10 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 10 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. This distinction is illustrated in Figure 11. For this reason Figure 10 shows the 1 ms 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 10; furthermore, it indicates that the heavy-tailed nature of OFF times is primarily due to

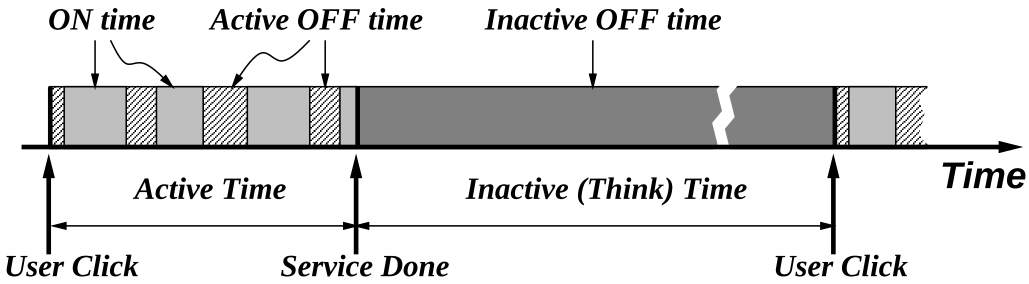


Figure 11: Active versus Inactive OFF Time

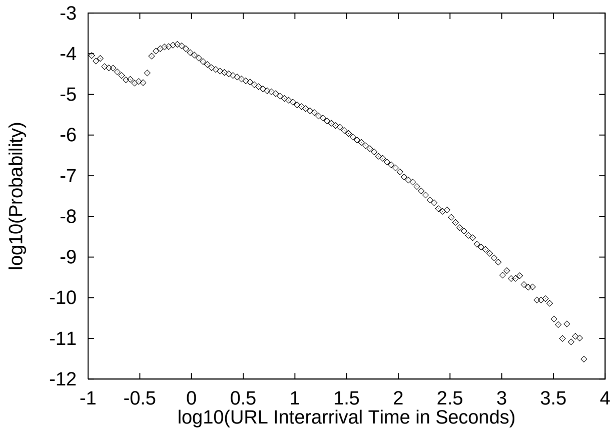


Figure 12: Histogram of inter-arrival time of URL requests

Inactive OFF times that result from user-induced delays, rather than from machine-induced delays or processing.

Another way of characterizing these two regimes is through the examination of the inter-arrival times of URL requests—namely, the distribution of $a_{i+1} - a_i$. Figure 12 shows that distribution.

The "dip" in the distribution in Figure 12 reflects the presence of two underlying distributions. The first is the inter-arrival of URL requests generated in response to a single user request (or *user click*). The second is the inter-arrival of URL requests generated in response to two consecutive user requests. The difference between these distributions is that the former is affected by the distribution of ON times and the distribution of Active OFF times, whereas the latter is affected by the distribution of ON times, Active OFF times, and Inactive OFF times. A recent study [7] confirmed this observation by analyzing and characterizing the distribution of document request arrivals at access links. This study, which was based on two data sets different from ours,[5] concluded that the two regimes exhibited in Figure 10 could be empirically modeled using a Weibull distribution

[5]Namely, the Web traffic monitored at a corporate firewall during two 2-hour sessions.

for the inter-arrival of URL requests during the active regime, and a Pareto distribution for the Inactive OFF times.

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) via the slope method yields $\hat{\alpha} = 1.50$ ($R^2 = 0.99$). Thus we see that the OFF times measured in our traces may be heavy-tailed, but with lighter tails than the distribution of ON times. In addition, we argue that any heavy-tailed nature of OFF times is a result of user *think time* rather than machine-induced delays.

Since we saw in the previous section that ON times were heavy-tailed with $\alpha \approx 1.0$ to 1.3, and we see in this section that OFF times are heavy tailed with $\alpha \approx 1.5$, we conclude that ON times (and, consequently, the distribution of available files in the Web) are more likely responsible for the observed level of traffic self-similarity, rather than OFF times.

6 Conclusion

In this paper we've shown evidence that traffic due to World Wide Web transfers shows characteristics that are consistent with self-similarity. More importantly, we've traced the genesis of Web traffic self-similarity along two threads: first, we've shown that transmission times may be heavy-tailed, primarily due to the distribution of available file sizes in the Web. Second, we've shown that silent times also may be heavy-tailed, primarily due to the influence of user "think time."

Comparing the distributions of ON and OFF times, we find that the ON time distribution is heavier tailed than the OFF time distribution. The model presented in [30] indicates that when comparing the ON and OFF times, the distribution with the heavier tail is the determiner of actual traffic self-similarity levels. 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. In fact, the work presented in [18] has shown that the transfer of files whose sizes are drawn from a heavy-tailed distribution is sufficient to generate self-similarity in network traffic.

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 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 [16] 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).

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 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 and Vadim Teverovsky of the Mathematics Department, Boston University, for many helpful discussions concerning long-range dependence. The authors also thank Vern Paxson and an anonymous referee whose comments substantially improved the paper. Carlos Cunha instrumented Mosaic, collected the trace logs, and extracted some of the data used in this study. Finally, the authors also thank the other members of the Oceans research group at Boston University for many thoughtful discussions. This work was supported in part by NSF grants CCR-9501822 and CCR-9308344.

References

[1] Mary G. Baker, John H. Hartman, Michael D. Kupfer, Ken W. Shirriff, and John K. Ousterhout. Measurements of a distributed file system. In *Proceedings of the Thirteenth ACM Symposium on Operating System Principles*, pages 198–212, Pacific Grove, CA, October 1991.

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

[3] T. Berners-Lee, L. Masinter, and M.McCahill. Uniform resource locators. RFC 1738, December 1994.

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

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

[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] Shuang Deng. Empirical model of WWW document arivals at access links. In *Proceedings of the 1996 IEEE International Conference on Communication*, June 1996.

[8] A. Erramilli, O. Narayan, and W. Willinger. Experimental queueing analysis with long-range dependent packet traffic. *IEEE/ACM Transactions on Networking*, 4(2):209–223, April 1996.

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

[10] National Center for Supercomputing Applications. Mosaic software. University of Illinois at Urbana-Champaign, Champaign IL.

[11] 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.

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

[13] 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.

[14] 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):1–15, 1994.

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

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

[17] John K. Ousterhout, Hervé Da Costa, David Harrison, John a Kunze, Mike Kupfer, and James 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.

[18] Kihong Park, Gi Tae Kim, and Mark E. Crovella. On the relationship between file sizes, transport protocols, and self-similar network traffic. In *Proceedings of the Fourth International Conference on Network Protocols (ICNP'96)*, pages 171–180, October 1996.

[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. *IEEE/ACM Transactions on Networking*, 3(3):226–244, June 1995.

[21] James E. Pitkow and Margaret M. Recker. A simple yet robust caching algorithm based on dynamic access patterns. In *Electronic Proceedings of the 2nd WWW Conference*, 1994.

[22] 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.

[23] Regents of the University of California. www-stat 1.0 software. Available from Department of Information and Computer Science, University of California, Irvine, CA 92697-3425.

[24] M. Satyanarayanan. A study of file sizes and functional lifetimes. In *Proceedings of the 8th ACM Symposium on Operating Systems Principles*, December 1981.

[25] 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.

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

[27] M. S. Taqqu, V. Teverovsky, and W. Willinger. Estimators for long-range dependence: an empirical study. *Fractals*, 3(4):785–798, 1995.

[28] 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.

[29] 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.

[30] 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. *IEEE/ACM Transactions on Networking*, 5(1):71–86, February 1997.