Preprint 1996. Appears on pages 3-25 in the book:
"A Practical Guide To Heavy Tails: Statistical Techniques and Applications"
Robert Adler, Raisa Feldman and Murad S. Taqqu, editors. Birkhauser, Boston, 1998

Heavy-Tailed Probability Distributions in the World Wide Web

Mark E. Crovella, Murad S. Taqqu, and Azer Bestavros [1] [2] [3]

Abstract

The explosion of the World Wide Web as a medium for information dissemination has made it important to understand its characteristics, in particular the distribution of its file sizes. This paper presents evidence that a number of file size distributions in the Web exhibit heavy tails, including files requested by users, files transmitted through the network, transmission durations of files, and files stored on servers. In addition, we argue that because of the presence of caching in the Web, the size distribution of transmitted files is primarily determined by the distribution of files available in the Web, and is relatively insensitive to the distribution of files requested by users. Finally, we discuss some of the implications of heavy-tailed transmission durations and relate these results to self-similarity in network traffic.

1. Introduction

The World Wide Web was designed and initially developed at the European Laboratory for Particle Physics (then called CERN) as a distribution method for scientific documents. Since its public release in 1992 the World Wide Web ("the Web") has been enthusiastically adopted by commercial, educational, and governmental users as a method of easily organizing and distributing information that is rich in multimedia content: text, graphics, animation, audio, and video. Growth of the Web has been very rapid; since its inception it has doubled in size roughly every nine months. As of early 1996 typical estimates of the number of documents available on the Web are in the range of 50 million [Bra96].

Currently (1996) the Web generates more data traffic on the Internet than any other application. As a result, characteristics of the Web have implications

---

[1] This work was supported in part by NSF grants CCR-9501822, CCR-9308344, NCR-9404931 and DMS-9404093 at Boston University.

[2] *AMS Subject classification:* 60K30, 60G18, 60E07.

[3] *Keywords:* infinite variance, self-similarity, stable distribution, Pareto, Lognormal distribution, Hill estimator, Internet, caching.

for network engineering, capacity planning, and performance evaluation of the Internet. In addition, since the Web is so widely used, a thorough understanding of its characteristics is an important goal in its own right. In this paper we describe a number of empirical characteristics of the Web, concentrating on measurements of probability distributions.

In order to place our measurements in context, we start by presenting background on how the Web is organized and implemented in the current Internet. The basic infrastructure used by the Web consists of host computers (each functioning as a client or a server or both) and a connecting network (which is usually the global Internet). In addition, caches may be present at various points in the system, and we describe these and their effects on data traffic over the Internet. We then describe details of our data collection methods, which were based on adding measurement apparatus to the Web clients at our site, and on conducting a survey of a number of Web servers.

Next, we show that a number of our measurements of the Web are consistent with the conclusion of heavy-tailed distributions. We first present evidence indicating that when files are transmitted over the network, the transmission durations appear to follow a heavy-tailed distribution. We then show that this effect may be explained by the sizes of the files themselves, as they too exhibit heavy tails. In fact, our data indicates that both the distribution of files that are requested by users, as well as the distribution of files that are available on a set of servers are heavy-tailed.

Thus, our measurements indicate that heavy-tailed distributions appear in a number of related datasets associated with the Web. One of the questions we try to address in this paper is: which of these datasets is the primary cause of heavy-tailed distributions in the Web? That is, are the distributional characteristics of transmission durations mainly dependent on user requests, or on the available files? Surprisingly, we show that the heavy-tailed property of transmission times is likely to be caused mainly by the distribution of available files, rather than by the nature of user requests.

We conclude with some observations on the implications of those heavy-tailed distributions for network performance evaluation and engineering. We note that the presence of a heavy tail in the distribution of transmission lengths is a possible cause of network traffic *self-similarity* [LTWW94] with scaling parameter $h > 1/2$, which means that traffic shows noticeable bursts at all scales of interest. we then note that self-similarity is a significant factor in the performance of networks like the internet; in particular, delays in transferring data can be much more severe when traffic shows self-similarity than would be predicted by traditional traffic models.

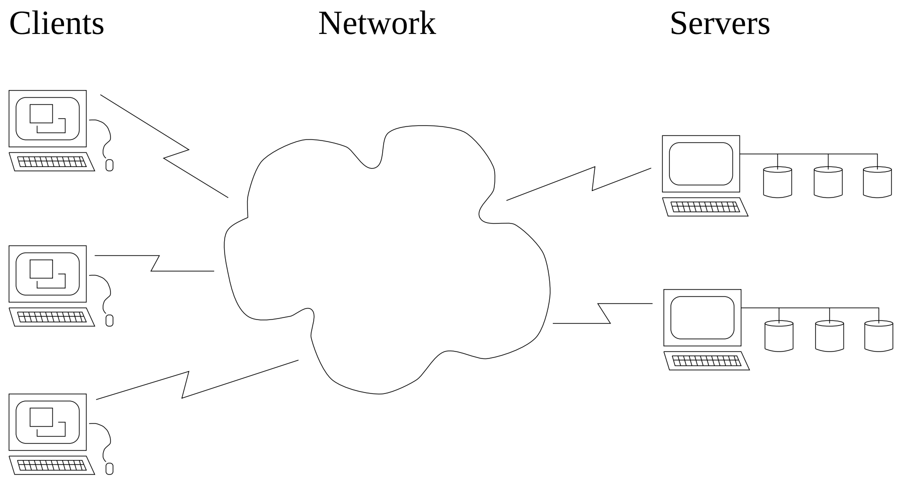

Figure 1: Clients, Network, and Servers in the World Wide Web

2. Studying the World Wide Web

In this section we present an overview of how the Web is organized and implemented, of how we took measurements of the Web, and of the methods we used to analyze the data we collected.

2.1 Organization and Implementation of the Web

The remarkable popularity of the Web seems to arise from a combination of its utility and its ease of use. It is useful as a means of publishing and delivering information in a wide variety of formats: raw data, formatted text, graphics, animation, audio, video, and even software. Its ease of use stems from the fact that it hides the details of contacting remote sites on the Internet, transporting data across the network, and formatting, displaying or playing the requested information regardless of the type of the particular computers involved.

Information in the Web exists as files on computer systems (*hosts*); each file has a globally unique identifier called a Uniform Resource Locator (URL). It is only necessary to know a file's URL in order to transfer it from wherever it is stored (potentially, anywhere in the global Internet) and display it on the user's local computer.

The Web is organized using a client-server model. Each file is stored on a specific host, specified as part of its URL; such hosts are *servers.* When a user requests a file, it is transferred to the user's local host, the *client.* (In fact, a single host can act as both a client and a server.) The software used by the client to retrieve and display files is called a *browser.* Figure 1 shows a schematic view of this model. Clients (which typically have display and input devices) retrieve

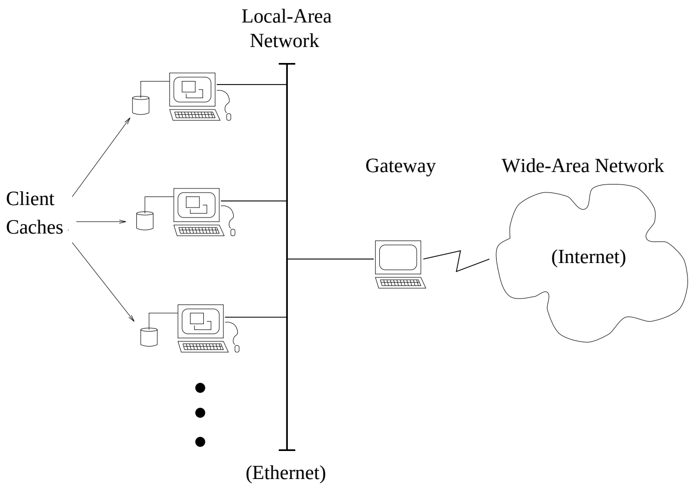

Figure 2: Implementation of Client Connections to the Web

files using the network from servers (which have storage devices). From the standpoint of the participants, the specific topology of the network is unknown — hence its representation as an amorphous cloud.

A more detailed view including implementation details is shown in Figure 2. The configuration shown is representative of the connections between a set of hosts at a single site, and the Internet. Clients are interconnected with a Local-Area Network (LAN) which is typically implemented by an Ethernet. The LAN is then connected via a special computer (a *gateway* or, equivalently, a *router*) to the Wide-Area Network (WAN), that is, the Internet.

All clients on the LAN employ *caching* to speed up access to WWW files. A cache is a set of copies of WWW files, kept on a local storage device — either main memory or disk. When a request is made, the client's browser software first checks to see if the file being requested can be found in the cache; if so, it need not be retrieved over the Internet but instead can be copied directly from the cache. The difference in response time between service from the cache versus service from the Internet can amount to two to three orders of magnitude; hence, most browsers have from the earliest days of the Web implemented some form of caching.

To implement caching, the browser examines each URL request that is made by the user. If the request can be served from the cache, this is a *cache hit;* if not, it is a *cache miss.* Whenever a file is retrieved from the network as the result of a cache miss, it then becomes a candidate for subsequent caching. Since the browser can only use a limited amount of storage for implementing the cache, it must decide whether to cache the file, and if so, whether to evict some other file(s) from the cache in order to find space for the new file. Such a set of decisions is called a *cache management policy.*

2.2 Web Data

Measurements of Web activity can be made at a variety of points in the network; in particular, two important measurement points are at the client and at the server. Server measurements are generally easy to obtain because one of the server's roles is to assess its own impacts on its host system. As a result, most servers keep detailed records of each access made to them. On the other hand, clients typically peform very little recordkeeping associated with their activities. Unfortunately, it is difficult to use server records to obtain a picture of Web activity on a LAN, since each client on the LAN may visit many different servers over a short time.

Thus, in order to capture all of the Web activity on a LAN, it was necessary to perform measurement on Web browsers. We added measurement apparatus to the browsers in use at Boston University's Computer Science Department. To do this, we modified the Web browser *NCSA Mosaic* [fSA] and installed it for general use. This browser is available in source code form, and permission has been granted for using and modifying the code for research purposes. Most important, at the time of the study (November 1994 through February 1995) Mosaic was the browser preferred by nearly all users at our site. Thus by instrumenting only this program we were able to measure nearly all of the local Web activity. Since that time, other browsers have become more popular than Mosaic; because these (commercial) browsers are not easily instrumented, collecting an equivalent set of data at the current time would be significantly more difficult.

In this paper we will refer to a single execution of Mosaic as a *session,* and the record of all URLs accessed in a session as a *trace.* Each trace is stored in a separate file called a *log.* Each line of a log corresponds to a single URL requested by the user; it contains the machine name, the time stamp when the request was made, the URL, the size of the file in bytes (including the overhead of the HTTP protocol) and the file retrieval time in seconds (reflecting only actual communication time, and not including the intermediate processing performed by Mosaic in a multi-connection transfer). Timestamps are accurate to 10 ms.

To collect our data we installed our instrumented version of Mosaic in the general computing environment at Boston University's Computer Science Department, which consists principally of 37 SparcStation 2 workstations connected in a local network. The data used in this paper was collected during the period 17 January 1995 to 28 February 1995. This data is freely available from Boston University [CBC95] and from the Internet Traffic Archives [Hal]. To our knowledge, these are the first such traces generally available to the research community.

Three of the most important datasets we collected are:

1. The set of *file requests:* this is a record of all requests for URLs made by users. This dataset contains many duplicate requests, which can occur when a user requests a file more than once, or when more than one user requests the same file. Many such requests result in cache hits, which means that they are satisfied without generating any network traffic.

2. The set of *file transfers:* this consists of all of the cache misses. Each element in this set corresponds to a single instance of a file being transferred over the network. Thus it is a proper subset of the set of file requests. However, despite the action of caching, some files are still transferred more than once over the network, so this dataset still contains some duplicate files.

3. The set of *unique files:* this set contains exactly one entry for each file, regardless of how many times it was requested or transferred. Thus this set is also a proper subset of the previous two.

For each of these datasets, we are primarily concerned with the distribution of object sizes measured in bytes. Descriptive statistics are shown in Table 1.

	Sessions	4,700	
	Users	591	
File Requests	575,775	Bytes Requested	2,713 MB
File Transfers	130,140	Bytes Transferred	1,849 MB
Unique Files	46,830	Unique Bytes	1,088 MB

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

We also collected two additional datasets. Of particular importance for the study of network traffic is the set of *transmission times.* For each item in the set of file transfers, there is a corresponding item in this set which specifies how much time was required to transfer the given file.

Finally, although our results are largely based on client measurements, for comparison purposes we also examine some measurements made at servers. In these measurements our goal was to gain a rough picture of the size distribution of available files present on Web servers. That set is somewhat elusive because it is constantly changing. To obtain an approximate snapshot of available files at the time of the study, we surveyed a subset of Web servers. To do this we started from a list of over 500 known Web servers. From these we selected those which provided freely available usage reports using a software package called *www-stat 1.0* [Reg]; there were 32 such servers. These usage reports provide information sufficient to determine the size distribution of the files present on the server, for files that were accessed during the reporting period. We collected the results from all 32 servers into a single dataset, which consists of 146,400

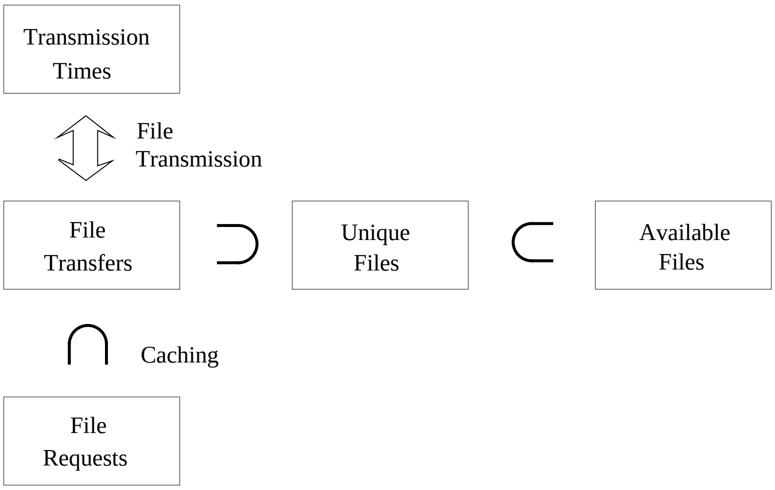

Figure 3: Relationship Between Datasets ($\updownarrow$ indicates a one-to-one relation and $\subset$, a subset relation).

files containing a total of 3,674 MB. We use this dataset as a representative of the set of *available files* in the Web. While the collection method we used does not provide a truly random sample of files available in the Web, it sufficed to assess whether the heavy-tailed property might be present in the set of available files.

The relationship between the datasets considered here is shown in Figure 3. Note that while the set of available files is conceptually a superset of the set of unique files, in the case of our datasets this is not strictly the case because the set of available files was collected independently of the others. We will discuss these relationships in more detail in Section 3.3.

2.3 Estimating Tail Weight in Web Data

The distributions we use in this paper have the property of being *heavy-tailed.* A random variable X follows a heavy-tailed distribution if

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

The simplest heavy-tailed distribution is the *Pareto* distribution, with probability mass function

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

and cumulative distribution function

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

Our results attempt to estimate the values of α for a number of empirically measured distributions. To do so, we use two methods:

1. Log-log *complementary distribution* (CD) plots; and

2. the *Hill* estimator [Hil75].

CD plots show the complementary cumulative distribution $\overline{F}(x) = 1 - F(x) = P[X > x]$ on log-log axes. Plotted in this way, heavy-tailed distributions have the property that

$$\frac{d \log \overline{F}(x)}{d \log x} \sim -\alpha,$$

for large x. In practice we obtain an estimate for α by plotting the CD plot of the dataset and selecting a minimal value of $x(\theta)$ above which the plot appears to be linear. Then we select equally-spaced points from among the CD 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 for small file sizes would otherwise unduly influence the least-squares regression.

Our second approach to estimating tail weight is by using the Hill estimator. The Hill estimator gives an estimate of α as a function of the k largest elements in the data set. In practice the Hill estimator for the k largest data items is plotted against k varying over a significant subset of the data; if the estimator stabilizes to a consistent value this provides an estimate of α.

Finally, a particular concern in our work has been to to verify that our datasets exhibit the infinite variance characteristic of heavy tails. To do so we use a simple test based on the theory of stable distributions, which we call the Limit Distribution (LD) test. We start by aggregating the dataset in question (X_i) over blocks of size m:

$$X_t^{(m)} = \sum_{i=(t-1)m+1}^{tm} X_i.$$

This process is repeated for a number of large values of m (in our case, for $m = 10, 100$, and 500). If the original dataset follows a distribution that belongs to the domain of attraction of a stable distribution with $\alpha < 2$, then the tails of the m-aggregated datasets will all tend to follow power-law behavior with the same α. If the original dataset follows a finite-variance distribution, then the aggregated datasets will tend to the normal distribution and their tails will decline exponentially.

This difference can be observed on a CD 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 4. The figure shows the LD test for aggregation levels of 10, 100, and 500 as applied to two synthetic datasets. On the left

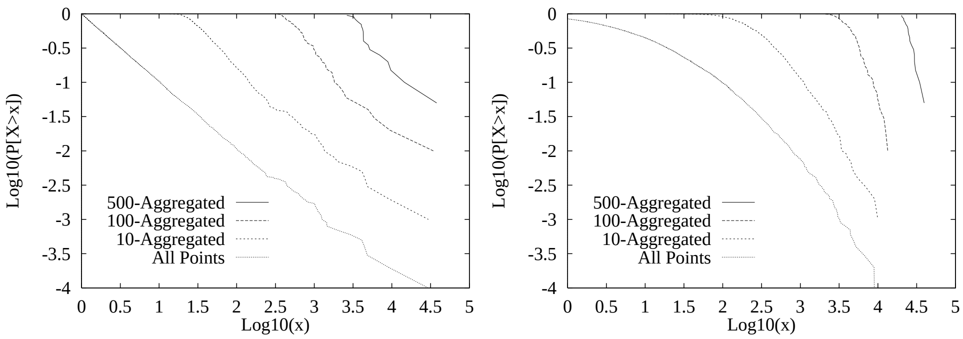

Figure 4: Comparison of LD Test for Pareto (left) and Lognormal (right) Distributions

the dataset consists of 10,000 samples 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 CD plot for the dataset is the lowermost line; the upper lines are the CD 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. WWW Size Distributions

As discussed in Section 2, our results are based on measurements of five datasets: file requests, file transfers, transmission times, unique files, and available files. In this section we present the principal results concerning distributions in these datasets. For each of these datasets, we show that the corresponding empirical distribution is consistent with heavy-tailed behavior. In addition, we describe the differences between these distributions and suggest some causal relationships between them.

3.1 The Distribution of Transmission Times

The set of transmission times has the most direct connection with network engineering. As we will discuss in Section 4, the conclusion of heavy-tailed behavior for this dataset is especially important because it provides support for an explanation of the observed self-similarity of network traffic.

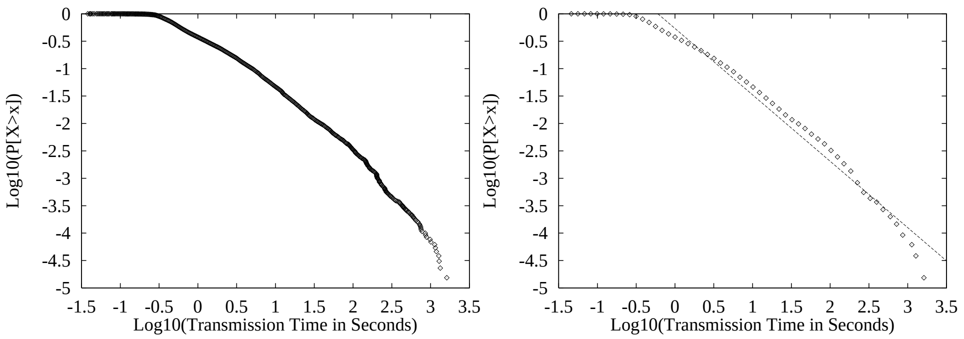

Figure 5: CD of Transmission Times of Web Files

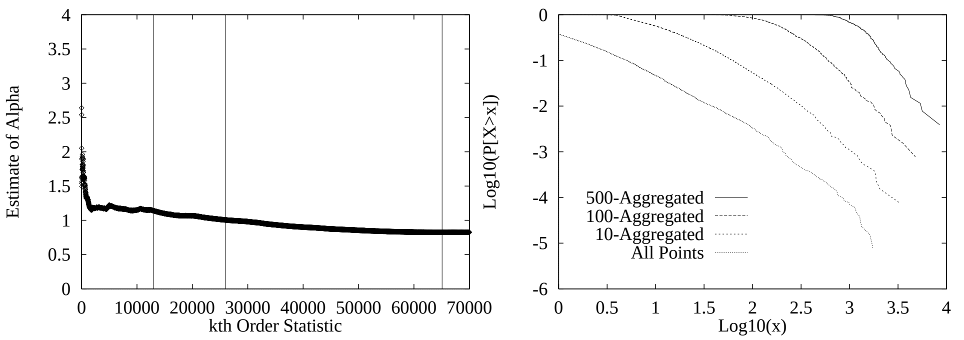

Figure 6: Hill Estimator (left) and LD Test (right) For Transmission Times of Web Files

The set of transmission times appears to exhibit heavy-tailed characteristics. Figure 5 (left side) presents the CD plot of the durations of all 130140 file transfers that occurred during the measurement period. The figure shows that for values greater than about $\log_{10}(x) = -0.5$ ($x \approx 0.3$ seconds), the plot is nearly linear — consistent with a power law upper tail. The least squares fit shown on the right side of Figure 5 ($R^2 = 0.98$) has a slope of -1.21, corresponding to an $\hat{\alpha} = 1.21$. Although the plot does appear to have some curvature, we show below that this distribution appears to be in the domain of attraction of an infinite variance (stable) distribution.

In Figure 6 (left side) we illustrate the use of the Hill estimator on this dataset. Vertical lines are plotted at the 90th, 80th, and 50th percentile of the dataset. The plot shows that the Hill estimator seems to settle to a relatively constant estimate as it gets close to the median of the dataset.

To assess whether this dataset is consistent with a conclusion of infinite variance, we use the LD Test as described in Section 2.3. The results for our dataset of transmission times is shown in Figure 6 (right side). The figure clearly shows that as we aggregate the dataset, the slope of the tail does not change appreciably. That is, under the LD Test, transmission times behave more like the Pareto distribution (left side of Figure 4) than the Lognormal distribution (right side of Figure 4).

The least squares fit in Figure 5 gave an $\hat{\alpha} = 1.21$; in comparison, the $\hat{\alpha}$ given by the Hill estimator at the median point is 0.83. The difference between these estimates, which also occurs in measurements presented in the following sections, may be due to difficulties in applying the slope estimation method to our data. The slope method is sensitive to the choice of points included in the fit; and the method of fitting to a set of equally spaced points may overemphasize the contribution from the relatively small number of samples in the tail. Hence, based on these three tests, we conclude that our assumption of infinite variance seems justified for this dataset, with α in the approximate range of 0.8 to 1.2.

3.2 Why are Transmission Times Heavy-Tailed?

To understand why transmission times are heavy-tailed, we now examine size distributions of Web files themselves. In particular, we present distributions for four datasets: file requests, file transfers, unique files, and available files.

First we present evidence that these distributions each show heavy-tailed behavior. Figure 7 shows CD plots and Hill plots for these datasets. Each of the CD plots shows nearly linear behavior over a large range of the random variable — typically about four orders of magnitude. In addition, each of the Hill plots shows reasonably stable behavior over a long range.

At the top of the figure are the plots for file requests; for these 575,775 data points the slope of the CD plot yields an $\hat{\alpha}$ of about 1.16 while the Hill estimator at the median is 0.73. Next are the plots for file transfers; for these 130,140 items the slope estimate yields $\hat{\alpha}$ of about 1.06 and the Hill estimator at the median is 0.73. Third are the plots for the unique files; for these 46,830 items the slope estimate is $\hat{\alpha}$ of about 1.05 and the Hill estimator at the median is 0.66. At the bottom are the plots for the set of available files; for these 146,400 items the slope estimate is $\hat{\alpha}$ of 1.06 and the Hill estimator at the median is 0.56.

These data show that heavy-tailed distributions seem to be present in each of our datasets of file sizes. In particular, the sizes of file transfers show heavy tails, and since each file transfer results in a transmission time measurement as well, it seems likely that that the heavy-tailed nature of transmission times is related to the heavy-tailed nature of file transmissions.

The fact that the distribution of file transfers in bytes seems heavier-tailed (slope $\hat{\alpha} = 1.06$, Hill $\hat{\alpha} = 0.73$) than the distribution of transmission times in seconds (slope $\hat{\alpha} = 1.21$, Hill $\hat{\alpha} = 0.83$) indicates that large files are transferred somewhat faster per byte than are small files; this may be a result of the fixed overhead of the TCP protocol's connection establishment, and mechanisms within TCP that begin transmissions at slower than maximum rate to avoid congesting the network [Jac88].

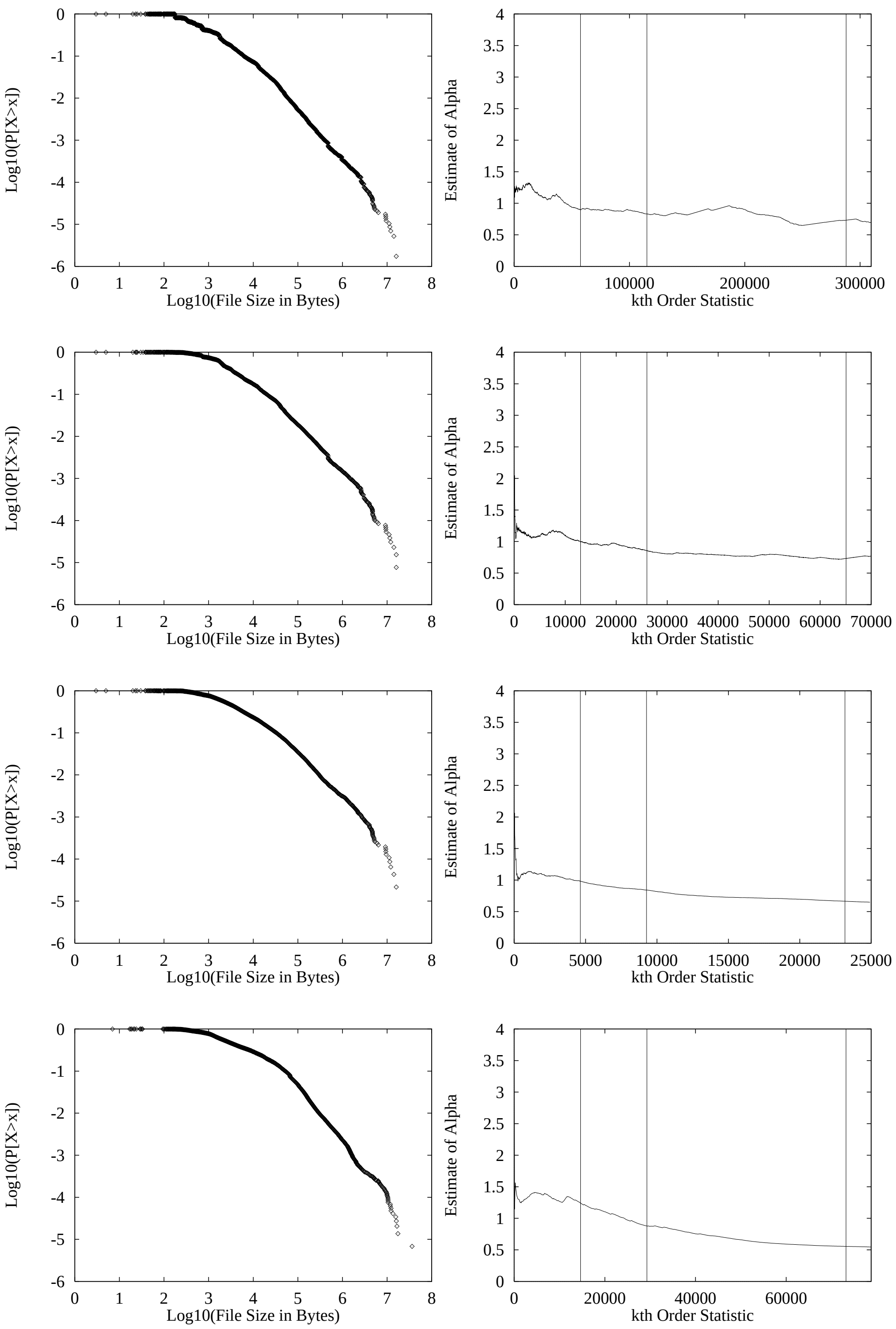

Figure 7: Evidence of Heavy Tails in File Requests (top), File Transfers (second), Unique Files (third), and Available Files (bottom).

Thus, these data show evidence that transmission times are heavy-tailed (which is important for network engineering) and that heavy-tailed transmission times are likely caused by heavy-tailed file transfers.

An important question then is why file transfers show a heavy-tailed distribution. On one hand, it is clear that file requests constitute 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 file transfers, and by implication, the self-similar properties of network traffic.

In fact, in the next subsection we argue that file requests are *not* intrinsically responsible for the heavy-tailed nature of file transfers. Rather, we will argue that the set of file transfers is more closely determined by what files are *available* in the Web than it is by what files are requested. That is, we will contend that the heavy-tailed nature of transmission times is more strongly determined by the size distribution of available files than it is by the size distribution of file requests. To do so we rely on characteristics of the set of unique files.

3.3 The Nature of Unique File Sets and the Action of Infinite Caches

Our argument is based on two important properties of the set of unique files. First, the set of unique files is the limit for the set of transferred files, as the cache size grows to infinity. Second, the set of available files is the limit for the set of unique files as time goes to infinity. Together these two properties suggest that for systems with large caches that run for long periods, the set of transferred files will naturally approximate the set of available files — and will be relatively independent of the particular requests made by users. We now explain and support these points.

First, we explain why the set of unique files is the limit for the set of available files, as the cache size grows to infinity. To start, it is helpful to imagine the behavior of a cache of infinite size. Such a cache would never need to evict any file. As a result, the only references that would cause cache misses would be those that happen to be the *first* reference made to the particular file. In our measurements, this set is in fact the set of unique files (whose distribution was shown in the last subsection).

In practice, the goal of any real cache management policy may be described as the attempt to simulate an infinite cache using finite resources. The better a real cache performs, the closer the set of transferred files will be to the set of unique files. In our measurements, 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 an 77% hit rate ($1 - 130140/575775$), while a cache of infinite size (shared by all users) would achieve a 92% hit rate ($1 - 46830/575775$).

The role of the set of unique files as the limit set for the set of file transfers can be seen in our data. In Figure 8 (left side), we show the distributions of

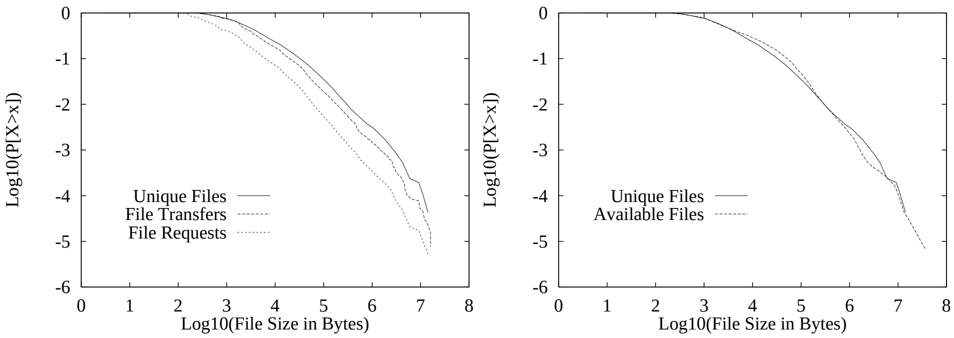

Figure 8: CD plots of the Different Distributions

file requests, file transfers, and unique files on the same axes. In this figure we can see how the distributions are changing as we progress from requested files to transferred files to unique files. The Figure shows that set of file transfers (*i.e.,* cache misses) is *intermediate* in distributional characteristics between the set of file requests, and the set of unique files.

The plot on the left side of Figure 8 shows that the median of the set of file transfers is larger than the median of the set of file requests. As it happens, in these datasets, users tended to request small files more often, per file, than large files. The sets of file requests and file transfers would be identical if there were no caching taking place. Because, in our case, caching is fairly effective, one cannot relate the two datasets. However, *whatever* the preferences of users happened to be — for large files, or small files, or neither — the action of caching is to move the distributional characteristics of the set of file transfers closer to that of the set of unique files. Thus, depending on the effectiveness of caching, the median transfer size may be closer either to the median of the set of file requests, or to that of the set of unique files.

The second property of the unique file set is that it will tend to approximate the set of available files as time grows large. Consider again the action of an infinite cache, as time progresses. File requests that are processed by the cache will only result in cache misses if they have never been processed by the cache before. If we assume that users, over time, will continue to visit files that have not been visited before, then the fraction of the set of available files that has been visited will be continuously growing. This fraction may grow more slowly as time goes on, but as long as users keep exploring, the visted fraction will continue to grow.[4]

[4]For simplicity, this discussion assumes the set of available files is static. Although the set of available files in the Web is in fact growing, the argument presented here still holds as long as either 1) users visit new files faster than the set of available files grows, or 2) the pattern

As time goes on, the fraction of files visited will grow large enough that the visited files will form an appreciable subset of the available files. When this happens, it is reasonable to expect that the visited subset will approximate in distribution the set of available files. Since each of the visited files occurs exactly once in the set of unique files, the unique file set will also tend to approximate in distribution the set of available files.

Of course, an important question related to this argument is whether enough time passes during the execution of a typical Web browser for this effect to occur. Note that if users were to visit new files at random, very little time would be needed for this effect to occur, since the set of unique files would from the start be a random sample from the set of available files.

Our data indicates that this effect does indeed occur at the timescales of real Mosaic sessions. This is shown graphically in Figure 8 (right side). The Figure shows that, in contrast to the comparisons on the left side of Figure 8, these two distributions are nearly identical over their entire shared range. This suggests that for our data, the set of unique transfers seems large enough to approximate the set of files available on the Web.

To summarize our argument we show the relations between the important datasets indicated in Figure 3. First, transmission times appear heavy-tailed, which is related to self-similarity of network traffic. Second, the distribution of transmission times seems to be related to the distribution of file transfers. Third, the set of file transfers is a superset of the set of unique files, but this superset relation tends to equality as cache sizes grow large. Fourth, the set of unique files is a subset of the set of available files, but this subset relation also tends to equality as time grows large. This means that the set of file transfers can be expected in general to be similar to the distribution of files *available* on the Web and, because of caching, relatively insensitive to the *particular requests* made by users.

3.4 Why are Available Files in the Web Heavy-Tailed?

Available files in the Web appear heavy-tailed (Figure 7, bottom). A 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, based on our evidence it does not seem to be the root cause of the heavy tails. This can be seen in the plot shown in Figure 9.

Figure 9 was constructed from the dataset of available files on Web servers, by categorizing all server files into one of seven categories. The categories we used were: *images, audio, video, text, archives, preformatted text,* and *compressed files.* This simple categorization was able to encompass 85% of all files. From this set, the categories *images, audio, video* and *text* accounted for 97%. The cumulative distribution of these four categories, expressed as a fraction of

of user visits to new files is independent of their sizes.

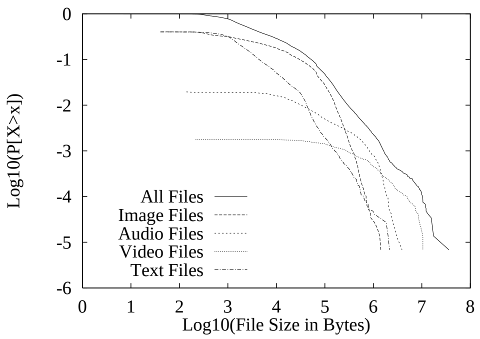

Figure 9: CD of File Sizes of 32 Web Sites

the total set of files, is shown on in Figure 9. In the figure, the upper line is the distribution of all files, which is the same as the plot shown on the right side of Figure 8. 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 suggests that the distribution of text files may itself be heavy-tailed. Using least-squares fitting for the portions of the distributions in which $\log_{10}(x) > 3$, we find that for all files available $\hat{\alpha} = 1.06$ (as previously mentioned) but that for the text files only, $\hat{\alpha} = 1.36$ ($R^2 = 0.98$). 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.

As an another example suggesting that multimedia is not fundamentally responsible for the heavy-tailed nature of available Web files, we compare the distribution of available files in the Web with an overall distribution of files found in a survey of Unix file systems. 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 [Irl94]. In Figure 10 we compare the distribution of available files in the Web with that data. The Figure plots the two histograms on the same, log-log scale.

Surprisingly, Figure 10 shows that in our Web data, there is a *stronger* preference for small files than in Unix file systems.[5] 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 available files in the Web is not nearly as heavy as the tail of the distribution of Unix files. Thus,

[5]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.

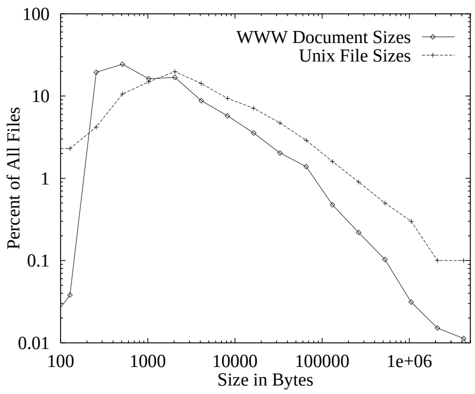

Figure 10: Comparison of Unix and WWW File Sizes

despite the emphasis on multimedia in the Web, this data suggests that Web file systems may be currently more biased toward small files than are typical Unix file systems.

The fact that file size distributions have very long tails has been noted before, particularly in file system studies [Sat81, Flo86, BHK+91], but without including power-law distributions and measurements of α. The authors in [PF94], on the other hand, studied a set of transfer sizes that were made using the more general FTP protocol; FTP is used to transfer files but does not include the notions of hypertext or multimedia presentation that have made the Web so popular. They 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 the more general case of FTP traffic.

3.5 Zipf's Law

Another instance of power-law distributions in our data occurs as an instance of Zipf's law [Zip49, discussed in [Man83]]. Zipf's law was originally applied to the relationship between the number of references made to a word in a given text, and its order in a ranking based on the same measurement. It states that if one ranks the words used in a given text by their popularity P (frequency of use) then P is related to the word's rank ρ by

$$P \sim 1/\rho.$$

Note that this relationship is parameterless, *i.e.,* ρ is raised to exactly -1, so that the nth most popular word is exactly twice as popular as the $2n$th most popular word. Zipf's law has subsequently been applied to other examples of popularity in the social sciences.

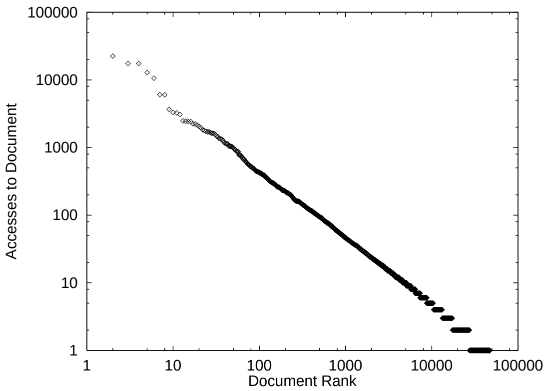

Figure 11: Zipf's Law Applied To Web Documents

Our data shows that Zipf's law applies quite strongly to documents on the Web. This is demonstrated in Figure 11 for all 46,830 unique files listed in Table 1. The figure shows a log-log plot of the number of references to each file as a function of the file's rank in reference count. The tightness of the fit to a straight line is remarkable ($R^2 = 1.00$), as is the slope of the line: -0.986. Thus the exponent relating popularity to rank for Web documents in our data is very nearly -1, as predicted by Zipf's law.

4. Implications for Traffic

One of the important implications of heavy-tailed file size distributions for network engineering lies in their connection to traffic self-similarity. Previous work has shown that network traffic, considered as a time series representing bytes or packets per unit time, typically shows self-similar characteristics with a scaling parameter $H > 1/2$ [LTWW94]. Intuitively, this means that traffic shows noticeable "bursts" (sustained periods above or below the mean) at a wide range of time scales — perhaps at all scales of interests to network engineers.

Heavy-tailed distributions have been suggested as a cause of self-similarity in network trafic. The authors in [WTSW95] show that if traffic is constructed as the sum of many ON/OFF processes, in which individual ON or OFF periods are independently drawn from a heavy-tailed distribution, then the resulting traffic series will be asymptotically self-similar. If the distribution of ON or OFF times is heavy tailed with parameter α, then the resulting series will be self-similar with $H = (3 - \alpha)/2$. If both ON and OFF times are heavy-tailed, the resulting H is determined by whichever distribution is heavier-tailed.

In the context of the World Wide Web, we can consider individual ON/OFF processes to be analagous to Mosaic sessions. Each Mosaic session can be considered to be either silent, or receiving transmitted data at some regular rate. This is a simplification of a real Web environment, but it indicates that if transmission durations are heavy-tailed, then it is likely that the resulting traffic will be self-similar in nature.

In [PKC96b] it is shown experimentally that heavy-tailed file size distributions are *sufficient* to produce self-similarity in network traffic. In that study a simple WAN was simulated in considerable detail, including the effects of the network's transmission and buffering characteristics, and the effects of the particular protocols used to transfer files. The results showed that if a network is used to repeatedly transfer files whose sizes are drawn from a heavy-tailed distribution, then the resulting traffic patterns exhibit self-similar characteristics, and the degree of self-similarity as measured by H is linearly related to the α of the file size distribution. In fact, the network traffic measured in this study is analyzed in [CB96] and is shown to exhibit characteristics consistent with self-similarity.

While transmission times correspond to ON times, the size distribution of OFF times (corresponding to times when the browser is not actively transferring a file) is also important. In [CB95] analyses similar to those in this paper are presented, showing that silent times appear to exhibit heavy-tailed characteristics with α approximately in the range of 1.5. Since the transmission time distribution appears to be heavier tailed than the silent time distribution, it seems that the distribution of file sizes in the Web is more likely the primary determiner of Web traffic self-similarity.

Since Web traffic is currently responsible for more than half the traffic on the global Internet, the presence of strong self-similarity in Web traffic has implications for the performance of the Internet as a whole. In [PKC96a] it is shown that the presence of self-similarity with large values of H in network traffic can have severe performance effects when the network has significant buffering. Buffering refers to the use of storage in the network to temporarily hold packets while they wait for transmission. Buffer use is related to the burstiness of traffic, because when a burst occurs, transmission channels can become overloaded, and packets need to be buffered while waiting for transmission channels to become available.

When traffic is strongly self-similar in nature, bursts can occur at a wide range of timescales. When very long bursts occur, many packets may require buffering. There are two negative effects that can result. First, packets stored in a large buffer will wait for long periods before they can be transmitted. This is the problem of *packet delay.* Second, since buffers are finite, the demand placed on them by a large burst may exceed their capacity. In this case, networks discard these packets leading to the problem of *decreased throughput* (because network bandwidth must be used to retransmit packets).

In practice, network buffers are usually made large so as to avoid the problem of packet loss, and maintain high throughput. However, packet delay remains a problem, and leads to delays in transmitting files; in the Web, this

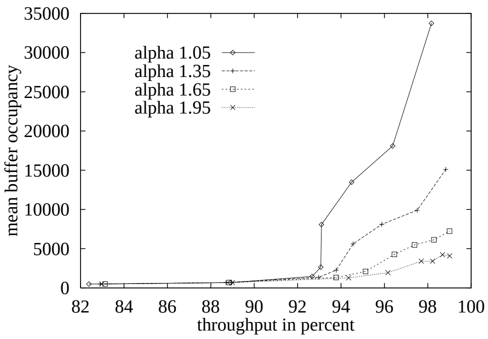

Figure 12: Relationship Between Throughput and Delay for Varying α.

is perceived by the user as an unresponsive browser. That is, because of long bursts in network traffic, users experience long delays in transfers, and the network appears to perform in an unresponsive manner.

An indication of the severity of this effect is shown in Figure 12, which is taken from [PKC96a]. This figure shows how packet delay and network throughput are related in a simulated network being used to transfer files drawn from a heavy-tailed distribution. The four curves correspond to different values of the α parameter of the file size distribution; the x axis measures the network throughput as a percentage of the maximum possible, and the y axis measures the mean buffer occupancy, which is essentially proportional to mean packet delay. Along each curve, different points are obtained by varying the amount of buffering in the network.

The figure indicates how heavy-tailed file sizes affect network performance. When the α of the file size distribution is near 2, it is possible to achieve high throughput (close to 100%) without significant packet delay. However, when α of the file size distribution is near 1, it is essentially impossible to achieve a comparable level throughput because of increasing packet delay. Thus as the file size distribution grows more heavy tailed (decreasing α), it becomes more difficult to achieve high network throughput without suffering serious packet delays.

5. Conclusion

The explosive growth of the World Wide Web has made it essential that network engineers understand the Web's characteristics. Since the Web is a system for organizing, delivering, and displaying data in the form of files, some of the Web's most important characteristics relate to how files are distributed in terms of size.

In this paper we've described characteristics of files in the Web, concentrating on five datasets: 1) transmission times of Web files; 2) the set of file requests made by users; 3) the set of file transfers that resulted from cache misses; 4) the set of unique files contained in the request set; and 5) a sample of the set of available files in the Web. We've shown evidence that each of these datasets exhibits a set of sizes that is consistent with a heavy-tailed distribution.

One of the important consequences of heavy-tailed distributions in network engineering lies in their relationship to traffic self-similarity. The presence of a heavy-tailed distribution of transfer times may represent a cause for the observed phenomenon of traffic self-similarity. We've indicated that traffic self-similarity has serious negative effects on network performance. Since the Web is currently the largest contributor of traffic to the Internet, the self-similarity of Web traffic and its possible causes is an important issue.

A thread running through our study is the attempt to trace the causal relationships of Web traffic self-similarity to heavy-tailed transmission durations, and from there to the characteristics of Web files. We have argued that the presence of caching in the Web has the effect of making the set of transmitted files distributionally similar to the set of available files, and relatively insensitive to the set of file requests. Thus we have suggested that the nature of file transmissions is primarily determined by the nature of the set of available files in the Web.

References

[BHK+91] 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.

[Bra96] Tim Bray. Measuring the web. In *Proceedings of the Fifth International World Wide Web Conference*, Available from `http://www5conf.inria.fr`, May 1996.

[CB95] Mark E. Crovella and Azer Bestavros. Explaining World Wide Web traffic self-similarity. Technical Report TR-95-015 (Revised), Boston University Department of Computer Science, October 1995.

[CB96] Mark E. Crovella and Azer Bestavros. Self-similarity in World Wide Web traffic: Evidence and possible causes. In *Proceedings of the 1996 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 160–169, May 1996.

[CBC95] Carlos A. Cunha, Azer Bestavros, and Mark E. Crovella. Characteristics of WWW client-based traces. Technical Report TR-95-010, Boston University Department of Computer Science, April 1995.

[Flo86] Richard A. Floyd. Short-term file reference patterns in a UNIX environment. Technical Report 177, Computer Science Dept., University of Rochester, 1986.

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

[Hal] Internet Town Hall. The internet traffic archives. Available at `http://town.hall.org/Archives/pub/ITA/`.

[Hil75] B. M. Hill. A simple general approach to inference about the tail of a distribution. *The Annals of Statistics*, 3:1163–1174, 1975.

[Irl94] Gordon Irlam. Unix file size survey — 1993. Available at `http://www.base.com/gordoni/ufs93.html`, September 1994.

[Jac88] Van Jacobson. Congestion avoidance and control. In *Proceedings of SIGCOMM '88*, pages 314–329, 1988.

[LTWW94] 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.

[Man83] Benoit B. Mandelbrot. *The Fractal Geometry of Nature*. W. H. Freedman and Co., New York, 1983.

[PF94] Vern Paxson and Sally Floyd. Wide-area traffic: The failure of poisson modeling. In *Proceedings of SIGCOMM '94*, 1994.

[PKC96a] Kihong Park, Gi Tae Kim, and Mark E. Crovella. The effects of traffic self-similarity on TCP performance. Technical report, Boston University Computer Science Department, 1996.

[PKC96b] 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) (to appear)*, October 1996.

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

[Sat81] M. Satyanarayanan. A study of file sizes and functional lifetimes. In *Proceedings of the Eighth ACM Symposium on Operating System Principles*, December 1981.

[WTSW95] 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 ACM SIGCOMM '95*, pages 100–113, 1995.

[Zip49] G. K. Zipf. *Human Behavior and the Principle of Least-Effort.* Addison-Wesley, Cambridge, MA, 1949.

Department of Computer Science and Department of Mathematics
Boston University
Boston, MA 02215

Email: crovella@cs.bu.edu, murad@math.bu.edu, best@cs.bu.edu