The Distribution of File Transmission Duration in the Web

Ronit Nossenson  
Dept. of Computer Science  
Technion, Haifa, 32000, Israel  
e-mail: ronitt@cs.technion.ac.il

Hagit Attiya  
Dept. of Computer Science  
Technion, Haifa, 32000, Israel  
e-mail: hagit@cs.technion.ac.il

May 19, 2002

Abstract

It is well known that the distribution of files transmission duration in the Web is heavy-tailed [8]. This paper attempts to understand the reasons for this phenomenon by analyzing the distribution of the transmission duration for each file separately.

We present evidence that the transmission-duration distribution of the same file from the same server to the same client in the Web is Pareto and therefore heavy tailed. Furthermore, we present results showing that for a specific client/server pair, transmission delay is not significantly affected by file size. Namely, all files transmitted from the same server to the same client have a very similar transmission duration distribution, which surprisingly, stays similar even for files with different order of sizes.

Finally, we study the impact of the server load on the transmission-duration distribution. In order to neutralize the network effects, we perform the simulation on a local network. We find out that when the server files and client requests are distributed in a realistic way, the transmission duration distribution of each file is usually Pareto as well.

1 Introduction

Understanding Web traffic’s behavior is necessary in order to improve the quality of service experienced by Web users. In particular, it is important to study statistical characteristics of Web servers, as was done, for example, in [2, 3, 7, 8, 11, 16].

Describing a Web server as a stochastic queuing system requires knowledge about the distribution of service time. There are two major types of service in a Web server: Static service involves file transfer, while dynamic service includes answering client queries. In this paper, we restrict our attention to static services provided by a Web server. In this situation, clients (browsers) send requests for files, and servers reply with the requested files. In such systems, the service time distribution is characterized by the files’ transmission-duration distribution (in short, TDD).

Crovella et al. [8] presented evidence that the distribution of files-transmission duration distribution is heavy-tailed. Roughly speaking, most of the samples in a heavy-tailed distribution are small but the probability for the appearance of a very large sample is non-negligible. Heavy-tailed distributions are characterized by extremely high variability. It is well known that heavy-tailed distributions have dramatic negative effects on the performance (see [4, 8, 16]). An example for heavy-tailed distribution is the Pareto distribution with shape parameter \(\alpha\) and location parameter \(k\), which has the following cumulative distribution function [10].

\[
F(x) = P[X \leq x] = 1 - (k/x)^{-\alpha}, \quad \alpha, k > 0, \quad x \geq k,
\]
This paper tries to understand why the distribution of files transmission duration is heavy-tailed. Three major factors might be the causes of this phenomenon: the file sizes, the server load, and the network conditions (that is, the relative distance between the server and the client and the load in the routes between them).

We understand the influence of each factor on the distribution of files transmission duration by analyzing the distribution of the transmission duration for each file separately. To see why these two distributions are not the same, assume we transmit two files, 10 times each. In order to estimate the TDD for all files, all 20 samples should be considered. On the other hand, to estimate the TDD of the first (or the second) file, only the 10 relevant samples should be considered.

Our empirical results

Our study follows two avenues. First, we obtain new conclusions by further analysis of the BU data set, collected and studied by Crovella et al. [8]. Second, we perform simulations on a local network, providing a new data set and additional information.

A more refined examination of the BU data set yields strong evidence that the TDD of the same file from the same server to the same client in the Web is Pareto and therefore heavy-tailed.

Our second result compares the TDD of all individual files transmitted from the same server to the same client in the BU data set. Surprisingly, almost all the transmission durations of each of these files were distributed in a very similar way (that is, Pareto with similar shape parameter and similar location parameter). These distributions stayed similar for files with different order of sizes (from several bytes to more than 100KB), covering most text files.

Our third result considers the effect of a distributed server, including several machines which provide a service together, on the TDD. We compared the distribution of each server in the distributed server with the distribution of all these servers together. We find out that the distribution of all these servers together was less heavy-tailed than the distribution of each server separately.

Our fourth result deals with the new data set provided by our simulation. The simulation consists of an Apache server and several clients requesting files. To neutralize the effects of an outside network, the server and the clients reside on the same local network. The size of the server files and their popularity are distributed in a realistic way (following [2]). In particular, the files size varies from several bytes to more than 140MB, covering text and multimedia files.

We study the TDD of each file under different loads of the server. The empirical result is that for almost all these files the transmission durations is Pareto. The shape parameters of their TDDs were very similar (that is, not correlated with the file sizes) and highly correlated with the server load. In contrast, the location parameters were highly correlated with the file sizes and not correlated with the server load.

The meaning of the empirical results

According to our first empirical result, there is extremely high variability in the time that a specific server needs to send the same file to a specific client. Possible explanations for this phenomenon can be derived from the network design and in particular, from caching mechanisms. Either file caching hits or server-name resolving caching hits (DNS caching hits) yield much faster transmissions. Intermediate caching over the network creates shortcuts and speed up the file transmission. The difference in the transmission duration of file from an intermediate cache versus the transmission duration from the server might amount to several orders of magnitude. In addition, some bottle-
necks around the request route might cause severe delays. Such a bottleneck can be an overloaded router, or even a link failure.

Another possible explanation for this phenomenon can be derived from the server load. The server load might change dramatically in different hours and different days [12]. This might cause high variation (manifested as a heavy-tailed behavior) in the transmission duration of the same file transmitted in different times. Indeed, Barford and Crovella [3] show that high server load generates a significant gap between the transmission of connection setup packets and the first data packet flowing from the server.

The server load might increase due to either arrival of many requests during a short time interval or arrival of requests with long service time. According to [12], the document request arrival process in Web servers shows self-similar behavior: it is bursty over a wide range of time-scales. Using Process Sharing discipline, the server responds to all arrival requests, and rarely reject requests. As a result, with the increased load, the server response (observed by the client) become slower. Thus, the result of a bursty server load is high variation in the response time, which might cause heavy-tailed behavior. If the server uses a First Come First Serve discipline and have a long queue, then the same phenomenon occur: with the increased load, many requests arrive and enter the long queue, each client’s request waits a long time in the queue and the client experiences a long response time.

As previously mentioned, long request service times (that is, long transmission durations) might also cause a heavy server load. For example, assume that the server uses a First Come First Serve discipline and the server is busy for a long time with a long transmission. All other client’s request waits a long time in the queue (even in a short queue) and the client experiences a long response time. The consequence is that long request service times of some requests might yield long response times at other requests. Since the distribution of files-transmission duration distribution is heavy-tailed [8], there is a non-negligible probability for the appearance of a requests with very large service times, which yield a heavy load server.

Our simulation data neutralized the effect of the outside network, and yet, almost all the files have Pareto transmission durations distributions. The previously mentioned server load variation might be a natural reason for this phenomenon. To investigate the impact of the server load on the files TDDs, we compare the TDDs of the simulation files. The empirical result was that the TDDs of different files have similar shape parameters. In addition, there was high correlation between the shape parameters and the server load.

Since the shape parameter is responsible for the heavy-tail of the Pareto distribution, we conclude that the server load is a major cause for the heavy-tailed behavior of the TDD in our simulation. The third empirical result which deals with a distributed server also confirms this conclusion. In the distributed server, the distribution of all the servers together was less heavy-tailed than the distribution of each server separatively, that is, if the load is better divided among several machines, the TDD is less heavy-tailed.

The fact that the server load is a major cause for the heavy tailed behavior of the TDD is important. It might explains why even local service (that is, service provided by servers on the local network) suffers from negative effects of heavy-tailed distributions on the performance [13, 14].

An interesting question is what is the effect of the file size on its TDD. Comparing the TDDs of the simulation files might supply a partial answer for this question. The empirical result was that the TDDs of different files have similar shape parameters and different location parameters increasing with the file size. That is, the file sizes are highly correlated with the location parameters...
and not correlated with the shape parameters.

Since the file sizes are highly correlated with the location parameter only, we conclude that the high variety of file sizes in Web servers is not a major cause for the heavy-tailed behavior of the TDD. Indeed, no correlation has been found between file sizes and transmission-durations distribution [8]. Nevertheless, although this lack of correlation is specifically mentioned in [8] (see Section 3.2 and figure 9 in [8]), it is a common mistake to assume that the heavy-tailed file-sizes distribution is causing the heavy-tailed TDD (see, for example, [9, 15, 17, 18]).

Moreover, when considering files which were transmitted from the same server to the same client in the BU data, the small effect that file sizes do have on the TDD seems to vanish. For files with size between several bytes up to around 100 KB (that is, several IP-packets up to several hundred IP-packets), almost all the transmission durations have similar shape parameter and similar location parameter (for their Pareto distributions).

Findings of Cohen and Kaplan [7] might explain this empirical result. This study reveals that DNS query times, TCP connection establishment times, and start-of-session delays at HTTP servers times, are the major causes of long waits, more than the actual transmission time. Note that, for every request made by the same client to the same server, the time spent on these operations is identically distributed regardless of the requested file size.

Organization

The remainder of this paper is organized as follows. The results concerning the BU data set appear in Section 2. The result concerning the data set provided by our simulation is presented in Section 3. In the last section we discuss future research work.

2 Empirical Results from the BU Trace

2.1 Data Collection

Web activity can be measured at a variety of points in the network; in particular, two important measurement points are at the client and at the server. We study the TDD of files from the client point of view. Namely, the time interval from the moment the Proxy started sending the client’s request to the server until the moment the Proxy finished receiving the response from the server.

For the first three experiments, we use a trace collected in Boston University (see [1, 5, 6]) since it is the latest public trace (known to us) that combines two essential properties for our experiments. First, the anonymity was achieved by hashing each column of the data set and not each line. This preserves the ability to compare members of a column for equality without leaking information regarding those members identities. For example, we can know that the same server sends file x and file y, without revealing the server’s identity or the files. The second essential property is that the trace allows to calculate the duration of each request. Usually, traces supply for each request its start time but not its termination time.

Another important fact regarding the data is that we can not use all transactions in the data set. Since we need statistically reliable samples, we used only information on files which were transmitted at least 30 times. Although such files are not very common, their number and the way which their information was gathered supply us a statistically reliable sample. For each experiment, we specify exactly how many files it involved and how many times each file was transmitted.
The trace data was collected in the Boston University Computer Science Department’s undergraduate workstation lab [5]. The lab consists of a cluster of 37 Sun SPARC station 2’s running Solaris. Traces were collected from April 4, 1998 through May 22, 1998, and included 306 unique users. The trace was collected using lightweight non-caching HTTP/1.1 proxy servers running on each workstation. The proxies logged all requests on a shared NFS filesystem. The trace contains more than 72K requests. More details about the trace are available in [5].

2.2 The TDD of a file in the Web

In this section we analyze the TDD of a file transmitted from the same server to the same client. Our first empirical result is an evidence that the TDD of the same file from the same server to the same client is Pareto and therefore heavy tailed.

In order to find out the TDD of each file, we analyzed the BU trace data. We considered only files that have been transmitted more than 30 times. The TDD of each file turned out to be a Pareto distribution with shape parameter $\alpha$ that varies between 0.6 to 1.8.

Figure 1 shows one example; additional examples appear in the appendix (Figure 7 and Figure 8). These examples present very popular files, that is, files that have been requested more than 700 times each. For each example, the file size and the number of samples are specified. Also, for each example, the samples are plotted against the estimated distribution. The graph’s X axis indicates the transmission duration, and the Y axis indicates the estimated $F(x) = 1 - F(x)$. Note that $F(x) = (k/x)^{-\alpha}$ for the Pareto distribution.

2.3 The TDD of files transmitted by the same server

In this section we analyze the TDD of files transmitted by the same server to the same client. Again, we analyze the TDD of each file separately and compare the results for all these files. As
mentioned in Section 2.2, the TDD of each file is Pareto.

Here we are testing two issues (i) the relation between the TDD location parameters of files transmitted by the same server to the same client, and (ii) the relation between the TDD shape parameters of these files. Since the location parameter represents the smallest sample, our intuition was that the location parameters would probably differ according to the files’ size in contrast to the shape parameters that will probably be similar.

Surprisingly, this intuition is not fully correct. Our second empirical result is an evidence that, for files transmitted from the same server to the same client, both the shape parameters and the location parameters are very similar. These distributions stayed similar even for files with different order of sizes (from several bytes to more than 100KB), that cover most text files.

In order to find the relation between the corresponding shape parameters and location parameters we processed the results of Section 2.2 further. The files were divided into groups according to their transmitted server. The most reliable sample consists of a group of 20 files, each file transferred more than 30 times. Although this set is not big, its size and the way which the information was gathered supply us a statistically reliable sample. Recall that to show that a specific property exists in a small statistically reliable sample set is a stronger result than to show that this property exists in a large set.

We summarize the results in Figure 2 and Figure 3. The graphs are plotted with a logarithmic X axis, due to the high variance in the file sizes. As one can easily verify from Figure 2, 15 files have the same shape parameter $\alpha = 1.3$. Another interesting result which can be deduced from Figure 3 is that the location parameter is not highly related to the file size. The location parameter of the files is in the range [0.035, 0.7] and most of them (18) have location parameter in the range [0.035, 0.1].

Figure 9 (located in the appendix) shows for each file, its samples plotted against the specified distribution. The graph’s X axis indicates the transmission duration, and the Y axis indicates the estimated $F(x) = 1 - F(x)$.

The consequence of this experiment is that the TDD is highly correlated with the condition of the network and the server load. For a client/server pair, for ordinary text files (with size between several bytes up to around 100 KB), the TDD is not significantly affected by the file size.

Our intuition is that this direction might change for files with very larger size. Our simulation results (see Section 3.2) show that very large files (more than 22 MB) have TDD less heavy-tailed than smaller files. Unfortunately, the BU-trace does not contain enough information to evaluate
Figure 3: The location parameters of transmission duration distributions of all files transmitted by the same server.

this intuition.

2.4 The TDD of a file transmitted by a distributed Web server

In the context of analyzing the server load influence on the TDD of a file, it is interesting to consider a distributed server, which includes several machines which supply a service together.

In this section we examine the TDD of a file transferred by a distributed server. For each machine $m$ in the distributed server, we compare the distribution of the file transmissions by $m$ with the distribution of all the transmissions of this file. To see why these distributions are not the same, assume a distributed server consists of two servers S1 and S2, and this distributed server transmits a file 25 times, 15 times by server S1 and 10 times by server S2. In order to estimate the TDD of the file transmitted by the entire distributed server all 25 samples should be considered. On the other hand, to estimate the TDD of the file transmitted by server S1, only the 15 relevant samples should be considered.

Some of the requests for a specific file in the BU data trace where answered by a server with multiple IPs, that is, a distributed server (the same URL was transmitted from multiple IPs). There are three files which were requested many times and were answered by such a server. Again, we present here the most reliable sample.

The most frequently transmitted file was transmitted 103 times by four distinct hosts, named here servers 1 to 4. Server 1 transmitted this file 38 times, Server 2 transmitted this file 34 times, Server 3 transmitted this file 21 times, Server 4 transmitted this file 10 times.

Figure 10 (located in the appendix) shows the file transmission duration samples when it is transmitted by all hosts together, plotted against a specified distribution. The graph’s X axis indicates the transmission duration, and the Y axis indicates the estimated $F(x) = 1 - F(x)$. The estimated shape parameter is $\alpha = 1.0$. In Figure 11 (also located in the appendix) we compare the file transmission duration when it is transmitted by the different hosts. The shape parameter of the file TDD when it is transmitted by servers 1, 3 and 4 is the same $\alpha = 1.0$. The shape parameter of the file TDD when it is transmitted by server 2 is $\alpha = 0.7$, that is, the tail is heavier.

Figure 4 presents the transmission duration samples for server 2 separately, plotted against its specified TDD (solid line) and against the TDD of all hosts together (dashed line).

Since we have no further knowledge on the internal structure of this particular distributed server, this empirical result can be interpreted in several ways. One way is to conclude that when
many hosts transmit the same file it makes the transmission duration less heavy tailed. On the other hand, if the hosts are not on the same local network then this result has no meaning at all. For example, the shape parameter of the file TDD might be heavier when the file is transmitted by server 2 due to a different location of server 2.

3 Empirical Results from the Technion Simulation

3.1 Data Collection

The data for our last experiment was collected from the log files of each client in the simulation. The simulation was performed in the Technion Computer Science Faculty’s distributed systems lab. The experiment consists of one server (a Pentium 3 733 Mhz computer) running Apache version 1.3.20, and three clients (also Pentium computers) running the simulation code.

Bellow, load rate $n$ means that each client requests arrival rate is $1/n$ seconds$^{-1}$. The simulation tested the load rates: $1, 1/2, 1/3, 1/4, 1/10, 1/15, 1/20, 1/25, 1/30$. We performed seven different executions. Each execution ends when the server fail. The simulation lasted for more than 17 hours.

The server has 100 different files, their sizes and popularity distributed according to [2]. The files sizes varies from several bytes to more than 140 MBytes. Each file was transmitted more than 30 times. The number of the transmissions and the way that we chose the requests (according to [2]) provide us a statistically reliable sample.
3.2 The influence of the server load on the TDD

The results of the simulation statistics are as follows. For the smallest 84 files (with sizes in the range from 75 bytes to 22 MBytes), the TDD was Pareto and therefore heavy-tailed. The TDDs of the largest 16 files (with sizes in the range [26, 140] MBytes) although definitely are heavy-tailed distributed, fit a littlest to the Pareto distribution.

We summarize the results in Figure 5 and Figure 6. The graphs are plotted with a logarithmic X axis, due to the high variance in the file sizes. The location parameter varies in the range [1, 4, 90]. The location parameter of the files TDDs stayed similar for different loads. Thus, there is no correlation between the location parameter and the server’s load. There was a high correlation between the location parameter and the file size. This correlation is demonstrated in Figure 6.

The shape parameter was highly correlated with the server’s load. Considering all the simulation executions data, the shape parameter varies in the range [0.4, 1.2]. For most of the files (82) the shape parameter was in the range [0.6, 0.7] (see Figure 5). There was no correlation between the shape parameter and the file size.

Figures 12, 13, 14, 15 and Figure 16 (located in the appendix) show TDD examples. These examples present files with different orders of sizes. For each example, the file size and the number of samples are specified, the samples are plotted against the estimated distribution. The graph’s X axis indicates the transmission duration, and the Y axis indicates the estimated $F(x) = 1 - F(x)$. 

![Figure 5: Example of the shape parameters of transmission duration distributions of files which were transmitted by the simulation server (the smallest 84 files).](image)

![Figure 6: The location parameters of transmission duration distributions of files which were transmitted by the simulation server (the smallest 84 files).](image)
4 Future research

We have seen that the TDD is highly influenced by the network conditions and the server load. For a client/server pair, for ordinary text files (files with sizes between several bytes up to around 100 KBytes), the TDD is not significantly influenced by the file size. Our intuition is that this direction changes for files with very large size. According to the Technion simulation data, files with sizes of more than 22 MBytes have slightly different TDD, and very large location parameters.

However, the BU trace does not contain enough information to evaluate such large file sizes. To analyze the transmission duration distribution of large files, a new trace should be made, according to the properties mentioned in Section 2.1. Note that it is quite difficult to collect such a trace in a university site since usually students have space limitation that prevent them from downloading large files.

To examine the correlation between the file size and its TDD, one should compare between the TDDs of files with similar sizes which were transmitted from different servers to the same client. This study requires knowledge about the servers’ IP addresses, since a situation where servers are located on the same LAN can influences the results. Unfortunately, the BU trace does not contain enough information to evaluate this issue.

Nevertheless, we performed some experiments regarding this issue. Since there are only a few files with similar sizes this set is not a statistically reliable sample and we do not plot these results here. These initial results suggest to divide the files into sets in a delicate way, that is, files will be in the same set according to their division into IP-packets. Our results show that within such closely related sizes, there is some correlation between the file size and its TDD.

Acknowledgment

The authors wish to thank Prof. Mark Crovella for providing the BU data trace and for helpful information, to Dr. Danny Raz for useful discussions and to Vadim Ryvchin, Nitzan Kashi and Yury Mogilevsky for performing the Technion simulation.

References


Figure 7: An example of file transmission duration distribution: fitting Pareto distribution (with shape parameter 1.8 and location parameter 1.5) to empirical data (the file size is 40773 bytes, the number of samples is 738).

Figure 8: An example of file transmission duration distribution: fitting Pareto distribution (with shape parameter 1.05 and location parameter 0.05) to empirical data (the file size is 68 bytes, the number of samples is 706).
Figure 9: The transmission duration distribution of files which were transmitted by the same server plotted against Pareto distribution (with shape parameter 1.3 and location parameter 0.05).

Figure 10: An example of file transmission duration distribution in a distributed server (all four servers together): fitting Pareto distribution (with shape parameter 1.0 and location parameter 0.08) to empirical data (the file size is 49 bytes, the number of samples is 103).
Figure 11: An example of file transmission duration distribution in a distributed server (a graph for each server): fitting Pareto distribution (with shape parameter 1.0 and location parameter 0.08) to empirical data (the file size is 49 bytes, the number of samples is 103).

Figure 12: An example of file transmission duration distribution: fitting Pareto distribution (with shape parameter 0.7 and location parameter 2.5) to empirical data (the file size is 75 bytes, the number of samples is 3880).
Figure 13: An example of file transmission duration distribution: fitting Pareto distribution (with shape parameter 0.6 and location parameter 1.5) to empirical data (the file size is 4.3 Kbytes, the number of samples is 1277).

Figure 14: An example of file transmission duration distribution: fitting Pareto distribution (with shape parameter 0.6 and location parameter 3.2) to empirical data (the file size is 90 Kbytes, the number of samples is 104).
Figure 15: An example of file transmission duration distribution: fitting Pareto distribution (with shape parameter 0.6 and location parameter 7.5) to empirical data (the file size is 762 KBytes, the number of samples is 74).

Figure 16: An example of file transmission duration distribution: fitting Pareto distribution (with shape parameter 0.7 and location parameter 350) to empirical data (the file size is 140 MBytes, the number of samples is 39).