

Investigating Distribution of Data of HTTP Traffic: An Empirical Study

Y. C. Chehadeh¹ IEEE Member, A. Z. Hatahet² IEEE Student Member, A. E. Agamy² IEEE Student Member, M. A. Bamakhrama³ IEEE Associate Member, S. A. Banawan² IEEE Member

¹Modelware Inc., Red Bank, NJ, 07701

²Department of Computer Engineering, University of Sharjah

³Department of Informatics, Technische Universität München

Abstract

Internet traffic today is dominated by that of the Hypertext Transfer Protocol (HTTP). Understanding the statistical characteristics of the data transferred via HTTP helps better model traffic patterns. In this work, we conduct an empirical study by employing an experiment that accesses roughly 34,000 of the most popular websites on the Internet today and crawls their web pages. We collect metadata information on the retrieved roughly two million objects. We determine statistics and distributions based on object sizes, occurrence of specific types, and sizes of specific types. The data of the distributions produced can be used as a template model for web-traffic modeling in future research. We further note an intriguing result that 5.7% of HTTP traffic from web servers to clients is due to sending spacer objects (image files representing a 1x1 white-space pixel) or to stale links referencing non-existing files. Such squander in bandwidth is not due to overhead and can be minimized by simple additions to the HTML standard and by automating the process of removing stale links.

1. Introduction

World-Wide-Web traffic on HTTP is considered the most common type of traffic on the Internet [1]. Consequently, it is important to have an accurate model of such traffic for the simulation and modeling of the underlying protocols and networking equipment. Our original research involves investigating solutions to improve the performance of HTTP traffic patterns. Such traffic is mainly generated in response to a client's request to a web server and is normally abstracted in the form of a webpage. The response of one request can entail the transfer of a large number of data objects through a large number of connections from many web servers to the requesting client. The parameters of a connection such as the amount of data to be sent and received, the number of overhead packets, and the QoS settings are heavily influenced by the size and type of data to be transferred. In order to study the behavior of such connections, an HTTP data distribution model has to be adopted. There have been several attempts at finding

the proper HTTP data distribution model [2],[3],[4],[5]. However, it is evident that not all the studies agree on the same model, and such studies might not be valid for today's web environment. The anomalies among the studies are due to the fact that the studies use very different datasets and different methodologies. The work in [2] investigates traffic on NSFNET at the time. It uses three datasets from web servers of academic institutions, two from research institutes, and one from a commercial Internet service provider. Such distribution does not reflect today's environment. The goal of the research concentrates on studying workload characterization of the web servers in an effort to improve their performance. It does not offer a template model for the data. The work in [3] relies on traces of requests to web servers collected from clients' logs. It investigates trends in those traces and generates distributions for the sizes of the retrieved files. At the time, the data retrieved was mainly files (labeled as documents) stored at web servers. Immediately-generated data (e.g. though Common Gate Interfaces) was not popular. The web environment is very different today. The work in [4] attempts to build a behavioral model based on collected web traces. It concentrates on producing a long-lasting behavioral model rather than a current data template model. The work in [5] improves over the previous work in that it examines actual data retrieved rather than relying on traces in logs; however, its datasets are not based on popularly accessed websites, and as in the other work, retrieved data that is not due to existing stored files was not popular then.

Studying the performance of a system through an experiment is normally considered the most accurate approach, alas, in many instances, the least practical (compared to simulation and mathematical modeling). Collecting empirical data on actual HTTP traffic can yield a significantly more accurate model. With the increasing functionality of the network-programming enabled languages, the high bandwidth connectivity, and the current over-the-counter computational power,

conducting a complex experiment on a large dataset and collecting empirical data on the Internet is achievable today in an acceptable time. All the aforementioned studies were conducted in different years and between 1995 and 1999. Considering the doubling of Internet traffic every year since 1997 [1] and the wide range of spectrum of applications constantly introduced to the Internet, the models might not be valid for today.

Based on the above, we conclude that there is a need to establish a template model for the size and type of data transported on HTTP today. The findings have to be based on actual accesses and data. In this work, we devise the necessary tools required to construct a complex experiment that generates a request for a webpage at a web server and recursively collects detailed metadata information on the connections and retrieved data objects. It is important to note that the received data can be both actual files stored at various web servers or dynamically-generated data. We run the experiment on a large dataset consisting of the most popularly accessed websites on the Internet today. Then, we collect and analyze the results. We show the resulting object size distribution, type distribution, and web-server brand distribution. The generated distributions can be used as an actual template model of HTTP data in future research. We also note two very intriguing results for three of the most common data object sizes. The remainder of the paper is organized as follows: in Section 2 the methodology in the research is discussed. Section 3 describes the architecture of the experiment. Section 4 demonstrates and analyzes the results. Finally section 5 concludes the paper and describes future directions.

2. Methodology

The procedure of our research constitutes of:

- 1) identifying a database of the most popularly accessed websites;
- 2) building the tools to automatically crawl the websites and recursively parsing the underlying webpages; and
- 3) obtaining the sizes, types and other metadata about retrieved data objects and analyzing them.

There are two sources that are widely accepted in the research community and that contain lists of the mostly accessed websites. The first is the daily-updated Global Top 500 list from Alexa Inc [6]. It aggregates its statistics on a rolling three-month-interval basis. The second is that of the Stanford WebBase project [7]. The first lists 500 websites and the second lists 33,974. There are 180 duplicate entries between the two. Combining the two lists and removing the duplicate entries results in a combined list with a total sum of 33,794 websites. This resulting list is used as the input to our experiment.

3. Architecture

The system architecture is shown in Fig 1. The input data, represented by the list of websites, is fed into the tools which access the pages of the websites and process the retrieved objects. The details of input files, source code, and results are made available at [8]. The tools are designed using the object-oriented methodology and implemented using Sun's Java 2 Standard Edition 5.0. The data analysis phase is conducted using MATLAB.

A website is composed of a home webpage and a hierarchy of webpages. A webpage can be composed of an object, such as text, image, application, binary data (octet-stream), or video, or it can be an HTML code enabling the page to contain embedded objects and references to other pages. The tools are designed to accommodate the website structure. The tools account for all received objects, whether such objects are files stored on web servers or dynamically-generated pages.

2.1. Datasets

Webpage accesses can be categorized based on the method of access. A casual user might initiate the access through the home page. A user familiar with a certain page, targets that page, even if it is not at the top of the hierarchy. Therefore, two datasets are defined. The first, labeled Home, contains the homepages of all the websites and represents pages accessed via the first type. The second labeled, Home+L1, contains Home plus the level one of pages in the hierarchy referenced by the home pages, representing pages accessed via the second type—representing “any” page in a website.

2.2. Tools

The tools are composed of the *Crawler*, *MetaGetter*, and *HTML Parser*. The home page of a website is passed to the Crawler. The Crawler is a web robot [9] capable of crawling a website. After crawling the pages of an input website, it uses the HTML Parser to parse the pages. It compiles a hierarchical list of URLs being referenced in the pages of the website and passes it to the MetaGetter. The MetaGetter categorizes the URLs passed to it and issues a HEAD command for each URL [10]. For each object with a returned response of type HTML, it functions in a recursive manner, where it uses the functionality of the HTML Parser to parse the HTML code and submits a HEAD request for each contained object. It collects all the metadata information it gathers in a hierarchical fashion and outputs them to be analyzed.

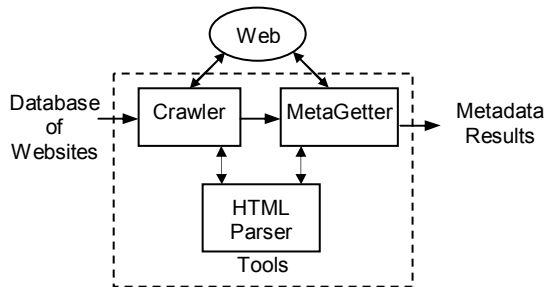


Fig. 1. Architecture of experiment

4. Results and Analysis

The two datasets, Home and Home+L1, are distributed among 20 machines, running the experiment simultaneously. Each machine has a 3.2 GHz Pentium IV running Microsoft Windows XP. Table 1 details the two datasets. Home consists of the home pages of 34,294 websites; whereas, the Home+L1 consists of 114,838 webpages (3,628 home webpages and their 111,210 level-1 pages). Crawling further level-1 webpages does not result in significant changes in the statistics obtained. Therefore, in the interest of runtime, the number of crawled pages is limited to 114,838. The number of retrieved objects is 499,714 for Home and almost 2 millions for the H+L1, rendering the shown average number of retrieved objects per accessed page. Finally, the wall-clock duration of the experiment is 1,591 hours.

Table 1. Datasets Home and Home+L1

	Home	Home+L1
Crawled webpages	34,294	114,838
Retrieved objects	499,714	1,895,776
Avg. num. objects per page	14.6	16.5
Execution time (hours)	163	1,428

4.1. Size Distribution

The size distribution can be observed by plotting the size of the objects against their relative occurrence (i.e., probability values), producing discrete probability density graphs. For Home, the sizes range from 0 to 8.2 MB, with the majority concentrated below 100 KB. For Home+L1, the sizes range from 0 to 360 MB, with the majority concentrated below 100 KB. There are several outliers in both datasets that skew the horizontal scale in both graphs. Removing such outliers for the sake of visibility-and not in the analysis-can eliminate such skew. These outliers are five from Home in the range of 3.2 to 8.2 MB and 13 from Home+L1 in the range of 7 to 360 MB. Such outliers can safely be removed even from the

analysis as each has a frequency of one and thus a probability of 2×10^{-6} for Home and 5.3×10^{-7} for Home+L1. Finally, there are three very popular sizes in Home and five in Home+L1 with probabilities ranging from 0.4% to 3.8%. Including them in the graphs skews the visibility vertically. They are excluded from the graphs but not the analysis. Fig. 2 and Fig. 3 depict the resulting distribution for Home and Home+L1, respectively. For enhanced clarity, the figures use a 50-25-25 percentile division, where the first division represents 50% of the occurrences (starting with size 0), the second represents the next 25% of the occurrences, and the last represents the last 25% of the occurrences. As can be seen, there are similarities in the distributions and similar observations can be drawn from both. The notable observation is the exponential decline in size, where 50% of the objects are less than 1.5 KB and more than 75% of them are less than 6KB in size.

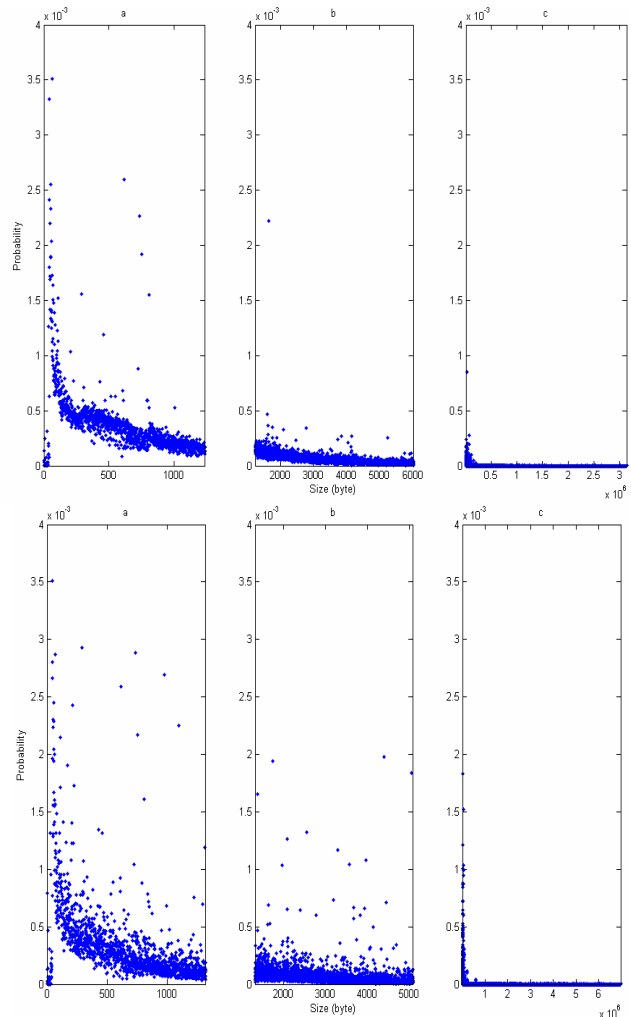


Fig. 2-3. Size dist. for Home and Home+L1

4.2. Type Distribution

MIME defines nine main media content types [11]. Fig. 4 and Fig. 5 depict the file type distribution, based on the number of occurrences, for Home and Home+L1, respectively. As shown, five out of the nine types are heavily used. “Others” refers to a null type, an unknown type, or to one of the four remaining unpopular types. The dominating type in both is image followed by text and then application. It is important to note that the percentages denote the percentage of objects of certain types and not the traffic of that type on HTTP. Fig. 6 and Fig. 7 depict the file type distribution, based on the total size of traffic caused by a specific type for Home and Home+L1, respectively. The dominating amount of traffic for Home is due to images and for Home+L1 is to text. Table 2 shows the average object size of all types for Home and Home+L1. As can be seen although objects of type Audio and Video have relatively very large sizes, their low probability (shown in Fig 4. and Fig. 5) limit their effect on the overall traffic (shown in Fig. 6 and Fig. 7). Table 1, Table 2, Fig. 4, and Fig. 5 can be considered a template for a traffic profile generated from the server when the client requests a Home webpage or Home+L1 webpage. The profile includes the number of retrieved objects, their type, and the size of each object.

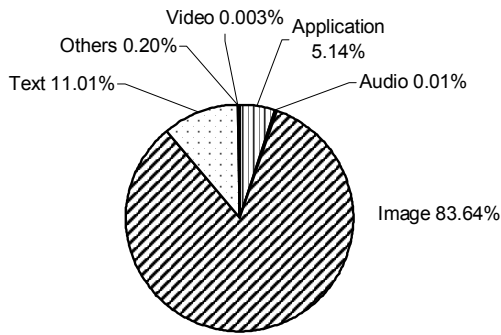


Fig. 4. Type dist. based on occurrence for Home

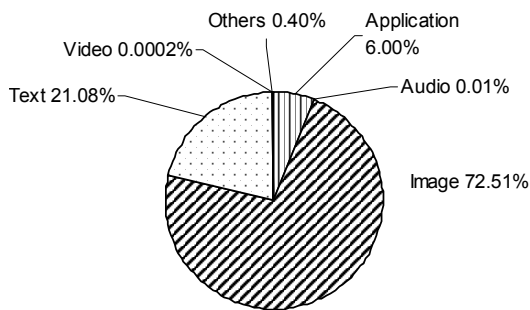


Fig. 5. Type dist. based on occurrence for Home+L1

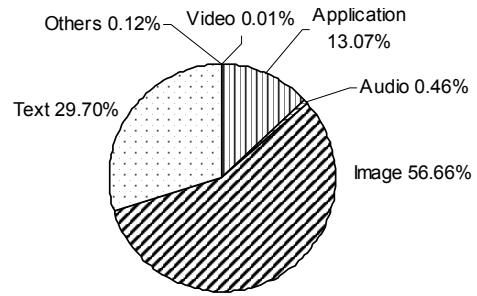


Fig. 6. Type dist. based on traffic size for Home

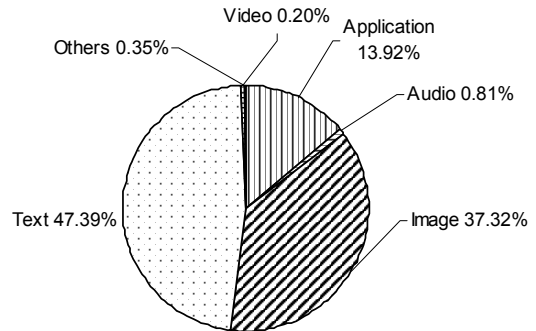


Fig. 7. Type dist. based on traffic size for Home+L1

Table 2. Average file size based on type (in K Bytes)

Type	Home	Home+L1
Application	14.7	16.4
Audio	230.9	512.5
Image	3.9	3.6
Text	15.6	15.9
Video	21.4	654.8
Other	3.3	6.2

4.3. Mode, Minimum, and Maximum

Table 3 lists the three most popular sizes, the three smallest sizes and the three largest sizes collected for all the traffic from Home and Home+L1. The dominating MIME main content type and subtype along with percent probability of each size are listed. The most intriguing result is that the most common size is 43 bytes, with a probability of 3.8%. This is a very high probability among almost two million objects. After further investigation, this size along with the third most popular size of 49 bytes were found to be the sizes of gif image files of a 1x1 pixel that are normally labeled as a “spacers” and used in website development. Such an image is helpful to perform a “nudge” operation to an object in a webpage layout. Unfortunately, such a simple operation that users use for alignment contributes to 4.5%

of the traffic carried on HTTP from the most popular websites on the Internet. The second intriguing result is that the second most popular size is 4,040 bytes. This is the size of the text message associated with the 404 error response code in HTTP, indicating that the requested object is not found. When a webpage or an embedded object within a webpage cannot be located on the storage of the web server, the web server issues a 404 response code, generates an error message as an HTML code, and sends it back to the client. Therefore, 1.2% of traffic from webpages in both datasets is due to missing files. The smallest file is an empty file of type text or null. Finally the largest file was found to be a self extracting compressed file of a computer game.

Table 3. Most common, smallest, and largest three objects

Size (Bytes)	Content type/subtype	% Probability
Mode (Most Common)		
43	image/gif	3.8
4,040	text/html	1.2
49	image/gif	0.7
Smallest		
0	text/html and null	6.5×10^{-3}
1	application/x-javascript	1.3×10^{-4}
2	application/x-javascript	3.2×10^{-5}
Largest		
360,865,812	application/octet-stream	5.3×10^{-7}
85,501,970	application/x-gzip	5.3×10^{-7}
76,276,142	application/pdf	5.3×10^{-7}

4.4. Web Server

In running the experiment, as each website is visited, the corresponding web-server software brand is retrieved. Fig. 8 shows the distribution obtained from visiting the 34,294 websites.

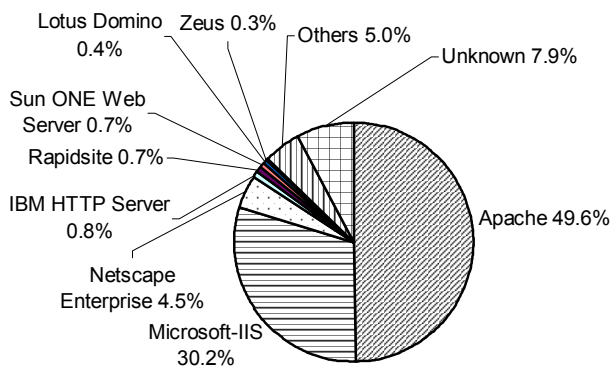


Fig. 8. Web-server software distribution for all websites

5. Conclusions

This work investigates the distribution of certain characteristics of retrieved objects when webpages of the most popular websites are accessed. Distributions based on object size, type occurrence, and type size are found. The data of the distributions can be used as a template for web-traffic modeling. An intriguing result is the fact that on average 5.7% of HTTP traffic from web servers to its clients is due to either spacer objects or non-existing files. It is important to stress that such waste in bandwidth is not due to overhead but is actual traffic. Such traffic can be minimized by implementing the “nudge” operation in web-development software differently, by introducing a pixel-based spaces of vertical-line spacing in the HTML standard, and by automating the process of removing stale links to invalid files through background-running tools. Future work can concentrate studying the generated traffic in terms of packets and finding the best fitting well-known probability density function for the distributions.

6. References

- [1] A. Odlyzko, “Internet Traffic Growth: Sources and Implications”, *Proc. of SPIE*, vol. 5247, 2003, pp. 1-15.
- [2] M. Arlitt and C. Williamson, “Web Server Workload Characterization: The search for Invariants”, *Proc. of International Conference. on Measurement and Modeling of Computer Systems*, 1996, pp. 126-137.
- [3] C. Cunha, A. Bestavros, and M. Crovella, *Characteristics of World Wide Web Client-Based Traces, Technical Report 1995-010*, Boston University, MA, 1995.
- [4] H-K. Choi and J. O. Limb, “A Behavioral Model of Web Traffic”, *Proc. of the 7th ICNP*, 1999, pp. 327-334.
- [5] B. A. Mah, “An Empirical Model of HTTP Network Traffic”, *Proc. of INFOCOM*, 1997, pp. 592-600.
- [6] “Global Top 500”, Alexa Inc., http://www.alexa.com/site/ds/top_sites?ts_mode=global
- [7] J. Cho, H. Garcia-Molina, T. Haveliwalia, W. Lam, A. Paepcke, S. Raghavan, and G. Wesley, “Stanford WebBase Components and Applications”, *ACM Transactions on Internet Technology*, Vol. 6, No. 2, May 2006, pp. 153-186.
- [8] “Distribution of HTTP data”, www.chehadeh.com/pro/res/HTTPData
- [9] “The Web Robots Page”, <http://www.robotstxt.org/wc/robots.html>
- [10] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, *Hypertext Transfer Protocol 1.1, RFC 2616*, IETF, 1999.
- [11] N. Freed, and N. Borenstein, *MIME Part Two: Media Types, RFC 2046*, IETF, 1996.