2012 8th Euro-NF Conference on Next Generation Internet (NGI)

An HTTP Web Traffic Model Based on the Top One Million Visited Web Pages

Rastin Pries, Zsolt Magyari, Phuoc Tran-Gia
University of Würzburg, Institute of Computer Science, Germany
Email: {pries,trangia}@informatik.uni-wuerzburg.de

***Abstract*—The ever-changing behavior of HTTP web pages requires an adjustment of the web models used for simulation and benchmarking. In this paper, we present the statistical data of the one million most visited web sites. Using this data, we examine the changes in size and number of objects by comparing our findings with well-known web traffic models. The results show a trend towards large pages including multimedia content. In addition, today's web pages are created dynamically, i.e., that the content is downloaded from web servers spread all around the world. Finally, we discuss new web traffic models and present the parameters gathered from our measurements.**

***Index Terms*—web traffic, measurement, modeling, HTTP**

I. Introduction

Web traffic based on the HTTP protocol is still the dominating application in the Internet. Within the last few years, the percentage of web traffic even increased due to social networks, videos embedded in web pages, and file hosting [1], [2]. The main video hoster is YouTube, which has a share of about 8% of the complete Internet traffic [3]. Large files are downloaded by direct download hosters such as Rapidshare and its share increased to up to 3% [4].

During the time when the widely used traffic model by Choi and Limb [5] was set up, YouTube was not even founded and downloads were handled using FTP. This makes such old traffic models obsolete, requiring new web traffic models taking into account these new services, new features of web browsers, and the new web page complexity. Such a model is essential for network planning, network simulation and emulation.

A new model can even help web server operators and cloud providers for dimensioning purposes. According to Popa et al. [6], HTTP can also be used for multimedia streaming using Content Delivery Networks (CDNs) and HTTP proxies. Here, HTTP chunking is used, where a video is subdivided into blocks, which are then distributed to the CDNs and can be downloaded separately by the user. Thus, a model would also be helpful for dimensioning the caching size and thus, to decrease the required bandwidth and to improve the Quality of Experience (QoE) for the end user.

In this paper, we present the results of our web page measurements. Whereas other papers in this area measure Internet access links on campuses or at small Internet Service Providers (ISPs), we focus on the most visited web pages and evaluate their structure. With this different approach, we are able to get a generalized model, which is not limited to the web browsing behavior of a single country or even one campus. To perform the measurements, we first downloaded the one million most visited web pages from Alexa. For each web page we obtained the main object size as well as the number and size of the embedded inline objects, separated into the different categories. The results show a trend towards an increased number of embedded objects and thus to larger web pages in general. These web pages can now be constructed by downloading the content from web servers distributed all around the world, which increases the variability of the page download time. Finally, we present our model parameters resulting from our measurements of the top million web pages.

The rest of this paper is structured as follows. In Section II, the background to web traffic models and measurements of the user's web browsing behavior is presented together with the related work. Section III describes our approach including the used measurement tools. The results of our measurements are shown in Section IV. Finally, conclusions are drawn in Section V.

II. Background & Related Work

The most widely used web traffic model is based on the work of Choi and Limb [5]. The model and terms are still used for standardization, cf. [7], [8]. According to them, a web browsing user can be modeled with an on/off process, see Figure 1(a). The on time starts as soon as the user requests

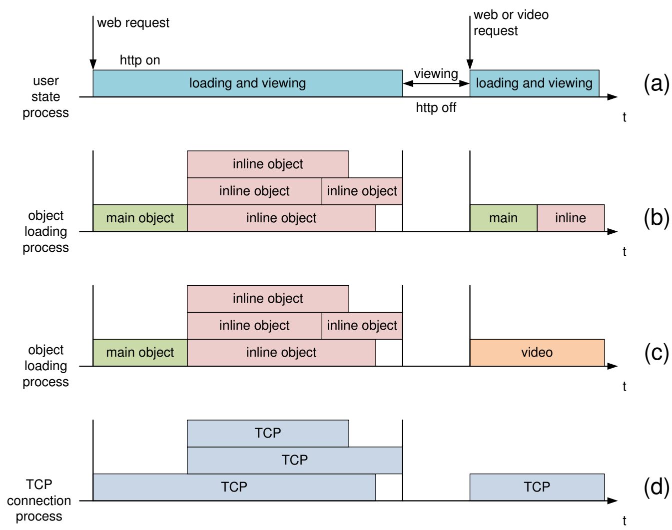


Fig. 1. Behavioral user model.

不同模型的参数对比

a web page and ends, when all objects of a web page are downloaded. After finishing the loading process of the web page, a silent period takes place while the user is reading the content of the web page. This reading time is according to [7], [8] exponentially distributed with a mean of 30 seconds. To get the exact numbers for the model, Choi and Limp measured HTTP traffic at the Georgia Tech campus.

The focus of our paper lies however on the structure of the web page itself, as shown in Figure 1(b). A web page consists of a main object, which defines the basic structure of the web page and contains the links to inline objects. Inline objects can be images, scripts, stylesheets, etc. After a user requests a new web page, the main object is loaded followed by all inline objects listed in the main object, cf. Figure 1(b).

While the browser has to set up a new TCP connection for every inline object when using HTTP version 1.0, HTTP 1.1 enables the use of a single connection for downloading the main object as well as all inline objects, cf. Figure 1(d). This decreases the loading time of a web page, because the TCP slow start phase is only run once. According to the HTTP 1.1 standard RFC 2616 [9], after downloading the main object, up to 2 TCP connections can be established in parallel for downloading the inline objects. The number of parallel TCP connections can however be further increased by the user, e.g., in the Firefox browser. However, Natarajan et al. [10] have shown that a large number of parallel TCP connections does not always decrease the download time of a web page, especially when the user has a low access connection speed.

In addition to the possibility of configuring the number of parallel TCP connections, today's browsers are able to not only download the content of the just accessed web page, but can be configured to download also the linked web pages directly after downloading the current web page. This of course shortens the inter-arrival time of web pages significantly and also increases the variance of this parameter, depending whether this new feature is activated or not. Thus, the reading time is hard to measure as it depends on the settings of the web browser. This new option reduces the waiting time of the user when clicking a link, but increases the load in the network as the browser downloads a lot of content, which would never be requested by the user.

The web traffic model of Choi and Limb is compared to other models from Mah [11], Barford and Crovella [12], and others in Tran-Gia et al. [13]. Whereas according to Choi and Limb, the main and inline object size can be fitted using a Lognormal distribution with a mean of about 10 kB, all others recommend a Pareto distribution with a median of about 2 kB. The number of inline objects varies between 5.5 measured by Choi and Limb and 2.8 from Mah. Table I lists the important web traffic model parameters from different studies.

With the increasing smartphone usage, Zhu et al. [15] set up two different web traffic models, one for wired and one for wireless (3G) devices. According to their analysis, the number of inline objects is smaller in the wireless model, because the web pages are especially optimized for mobile, small devices. However, this difference is of less importance because today's smartphones are capable of displaying the same web pages as wired devices.

TABLE I
OVERVIEW OF WEB TRAFFIC MODELS, PARTIALLY TAKEN FROM TRAN-GIA ET AL. [13].

literature source	main object size	inline object size	number of inline objects	reading time
Choi [5]	Lognormal mean:10 kB med.:6 kB σ=25 kB	Lognormal mean:7.7 kB med.:2 kB σ=126 kB	Gamma mean:5.5 med.:2 σ=11.4	Weibull mean:39.5 s med.:11 s σ=92.6 s
Mah [11]	Pareto α=0.85-0.97 med.:2-2.4 kB	Pareto α=1.12-1.39 med.:1.2-2 kB	- mean:2.8-3.2 med.:1	mean:1000-1900 s med.:15 s
Barford [12]	Pareto α=1 k=1000		Pareto α=2.43 k=1	Weibull α=1.46 β=0.38
Lee [14]	Lognormal mean:11.9 kB σ=38 kB	Lognormal mean:12.5 kB α=116 kB	Gamma mean:5.07	Lognormal mean:39.7 s σ=324.92

A new traffic model was set up by Lee and Gupta in 2007 [14]. They claim that existing models are not valid anymore because a user can request another web page, when the download of the last inline object was still not completed. This behavior can also be modeled using an on/off process, but the on period can contain now multiple web page requests whose objects overlap. In Section IV, we compare their measurement results with ours.

In contrast to the papers mentioned above, Butkiewicz et al. [16] did not came up with a new HTTP model, but showed the complexity of today's web pages. Whereas in former times, the complete content was downloaded from a single web server, especially *news* pages download the content from many servers and origins. This of course increases the variability of the page load times.

At the same conference, Ihm and Pai [17] presented their study on modern web pages. They analyzed modern web traffic and compared object sizes and number of objects of users in different countries as well as the change between 2006 and 2010. In all countries, an increased object size was measured and a small increase in number of embedded objects. The statistics show that there is a huge difference between the web pages accessed in France to those in China or Brazil. In contrast to the other publications in this area, they observed a large number of embedded video files.

The large number of embedded video files are underlined by our measurements performed in 2008 [1], [2] and 2010 [18]. One result of our measurements of 600 households in 2010 is shown in Figure 2. The chart illustrates the application distribution in terms of transmitted Bytes, leaving the number of not identified flows out.

The figure presents three important things. First, HTTP traffic is still the dominating application in the Internet. Second, we measured 8% flash traffic, which is mainly caused by YouTube. YouTube videos are embedded in a web page as

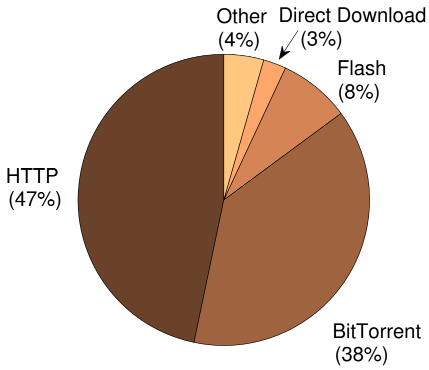


Fig. 2. Pie chart showing the application distribution in terms of transferred Bytes.

an object with a link to the actual .swf video file. This means that not the complete video is integrated in a web page, but a direct link towards it. Finally, we measured a high percentage of direct download links. These links are also embedded into web pages and can be accessed by clicking on the link.

Thus, we need to differentiate between web page and video/direct downloads for our web traffic model shown in Figure 1. While Figure 1(b) illustrates the normal web browsing behavior similar to Choi and Limb [5], Figure 1(c) presents the case when the user requests a video from the previously accessed web page. This has an impact on the viewing time, which is assumed to be much shorter and can also be negative, meaning that the user requests the video before the complete web page has been downloaded. The same applies for a direct download. Which model to choose depends on the percentage of video and direct download flows, which is according to our measurements 2-5% of the total number of flows.

The complexity of today's web pages makes it difficult to determine the reading time as it depends on the preferences of the user and the web browser settings. Therefore, we focus on the structure of the web page itself by measuring the main object size as well as the number and size of the inline objects.

In the next section, we describe how we gathered the parameters for the model. In contrast to previous works, where links on campuses or access networks are measured and analyzed, we take a look at the one million most visited web pages and analyze them. This way, a more generalized web traffic model can be set up because it does not only reflect the browsing behavior of students on campuses or a single country. The study of Ihm and Pai [17] underlines our method for setting up a web traffic model because they observed a huge difference in the web page statistics accessed in various countries and with measuring the most visited web pages worldwide, the model is free of site- and country-specific bias.

III. Measurement Methods and Settings

As mentioned in the previous section, our goal is to get detailed information about the one million most accessed web pages. Therefore, we first downloaded a list of the top one million sites from Alexa[1]. As it would take too long to crawl all pages from a single computer, we set up a cluster consisting of 31 nodes. Each of these nodes runs one virtual machine and executes 10 to 30 threads in parallel. Each thread works as follows:

1) A URL of the top one million list is retrieved from a central database.
2) The web page is opened using Mozilla Firefox, emulating the normal user behavior.
3) Web page statistics are gathered using WebDeveloper [19], which displays the web statistics on a second browser tab.
4) The displayed statistics are retrieved by Chickenfoot [20] and stored in a database on a centralized server.

In total, we captured the web page statistics from 881,640 pages. The other web pages have been unavailable during the measurements. In the following, we describe the crawling process in detail.

A. Mozilla Firefox Web Browser

Today, Firefox is one of the most popular browser due to its availability on almost all platforms. We use the browser to emulate the user's surfing behavior and the network traffic on a desktop, to achieve the same behavior as a browser has when the user surfs the web on it. This includes running scripts, which can load other content as described in the HTML file. With a Java crawler this could not be achieved. Firefox extensions enable to automate the crawling process through scripting. Add-ons help to count and measure downloaded object sizes. For the scripting and automation work we used Chickenfoot [20] and the measurement of file sizes and the object count were done with WebDeveloper [19]. Problems we were facing when downloading the web pages using Firefox were worms, viruses, and bugs of the Firefox browser by the parsing of the page sending itself into nirvana. For these problems we used VMware workstation with snapshots.

B. WebDeveloper and Chickenfoot

WebDeveloper [19] is coded by Chris Pederick in JavaScript. It is an add-on for Firefox with a huge set of functions to get information about the currently loaded web page in the browser. After loading each web page, Chickenfoot calls the function of WebDeveloper to display page information about objects and their sizes in a new window.

Chickenfoot's scripting language can be used to manipulate web pages and for automated web browsing. It therefore adds a pattern matching system for identifying elements in the rendered model [20]. Embedding automation tools inside a browser guarantees the same look and feel of a site that a user accesses with all the inline-objects and dynamic content. Chickenfoot is able to control the browser, grab content information from a site, and auto fill in forms. We used it to get and send data to our database.

[1] http://www.alexa.org

During the loading of the web page, Chickenfoot commands are suspended. When the site does not load within 30 seconds, it throws an exception and the whole crawling process is stopped. This standard setting caused many problems with Chinese web pages. Therefore, we increased the timeout to 300 seconds for all web page downloads. After the site is loaded, Chickenfoot sends an event to the *Activate WebDeveloper Extension*, which calls the function `webdeveloper_viewDocumentSize();` from WebDeveloper. After the new tab opens and the information is moved to the current tab, Chickenfoot starts to parse the table of WebDeveloper. It collects the URLs of documents, images, objects, stylesheets, and scripts from the table shown on the screen collected from the Document Object Model (DOM) of Firefox. If there are any inline objects loaded dynamically by a script, this must be found by WebDeveloper in the DOM. After all information is collected, the crawling script stores the data in our MySQL database and starts another crawling session.

C. VMware Workstation

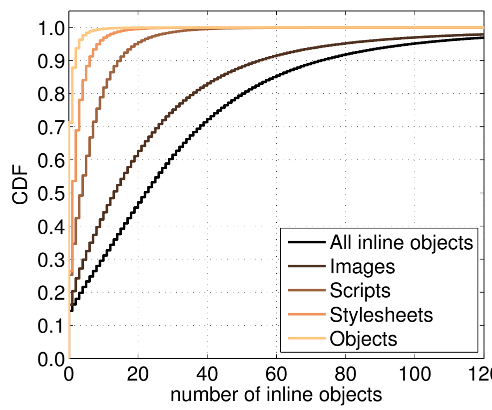
Crawling one million web pages confronts the browser with viruses, bugs, popup windows, and other surprises. In addition, pop-up windows stop the crawling waiting for a user interaction. On every 20-50th web page we observed a disturbing element that makes the crawling difficult. We crawl using a virtual machine running inside of VMWorkstation and using snapshots, we refresh the whole operating system with the crawling process every 5 minutes. We started the crawling script and made a snapshot of the virtual machine. When the crawling script started, it waits 20 seconds until the crawling process starts. We needed 2 seconds to snapshot the virtual machine and the remaining 17 seconds are needed every time, after the snapshot is restored, to load the VM onto memory and to start. To use more processors, we tried to run more than one VM but the network address translation of the VMWorkstation was unacceptably slow. With a single virtual machine after tens of visited sites, the NAT seems to slow down too. Thus, every time we restart the virtual machine, we restart the VMware NAT service too. Refreshing the NAT service on the host operating system sustains the speed of the crawling process on the guest VM.

IV. Results and HTTP Model

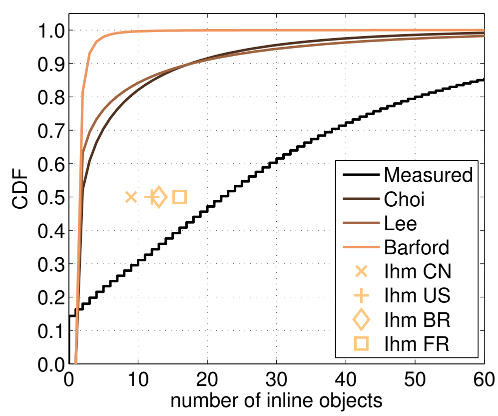
First, we take a look at the number of inline objects per HTML web page. This is shown in Figure 3(a) split into the different objects. Not surprising, most embedded objects are images. On average, 23 images are placed on a web page. The maximum number of images we measured on a web page was 1913. The second largest number of inline objects are scripts. These scripts, generally Javascript, are used for user interactions, e.g. button reactions, but can also include HTML 5 video players. On average, 4.8 scripts are included in a web pages. This quite large number of scripts per document was also observed by Ihm et al. [17]. In addition, the paper shows an increase of the number of embedded scripts from 2006 to 2010. With the spread of HTML 5, we expect that this number will further increase. Although the number of objects embedded on a web page seems to be quite low, we have to be aware of the fact that each object presents a linked video file. The largest number of embedded video files was measured on a web page containing adult content and included 124 objects.

To compare our measurement results with already published papers in the field, we plotted the number of inline objects of four other measurements in Figure 3(b). The lowest number of inline objects was observed by Barford and Crovella [12] in 1998. Lee and Gupta [14] and Choi and Limb [5] measured a similar number of inline objects per web page with a mean of about 5 objects. This is quite surprising as the paper from Lee and Gupta was published 9 years later. From the newest study, the paper of Ihm et al. [17], we are only able to show the median of the number of inline objects observed in China (CN), United States (US), Brazil (BR), and France (FR). Although the median is with 14 lower than our median with 22, it illustrates that more and more objects are embedded within a web page.

After having shown the increase in number of embedded objects over time, we now what to take a look if also the size

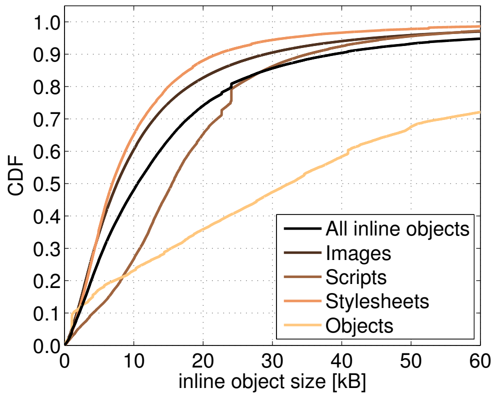


(a) Number of inline objects split into object type.

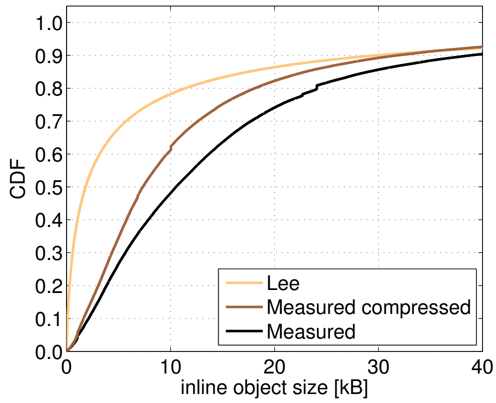


(b) Comparison of number of inline objects with other models.

Fig. 3. Number of inline objects embedded in a web page.

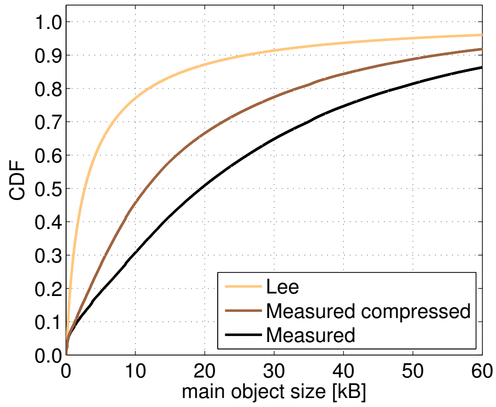


(a) Size of inline objects split into object type.

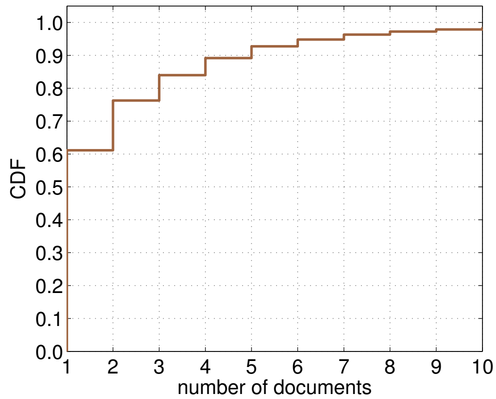


(b) Comparison of size of inline objects with other models.

Fig. 4. Size of inline objects embedded in a web page.



(a) Size of main objects.



(b) Number of documents.

Fig. 5. Main object size and number of documents.

of the inline objects has increased. The results, again separated for the different embedded object types are illustrated in Figure 4(a). The largest measured inline object was an image with a file size of 8 MB. Although the largest embedded object was an image, on average the embedded objects are much larger than the pictures with a mean of 101 kB. It has however to be mentioned that a flash video embedded in a web page is not counted with its complete file size, but only the object itself, which has a size of about 3 kB. This explains also that 10% of the measured objects are 3 kB in size. Finally, we can also observe from the figure that about 6% of the scripts have a size of 24 kB.

To see if the size of the number of inline objects has increased over time, we compare our result with the measurements from Lee and Gupta in Figure 4(b). Two interesting things can be observed from the figure. First, the percentage of large inline objects remained similar compared to the results taken from Lee and Gupta in 2007. Still, about 7% of all embedded objects have a size larger than 50 kB. Second, the average size of the objects has increased. Choi and Limb measured an average inline object size of 7.7 kB in 1999. Lee and Gupta measured an average size of about 12 kB in 2007, whereas our average inline object size almost doubled with 23.9 kB.

In addition, we plotted the inline object size for compressed content. Compression of inline objects can decrease the download time from a web server. Main objects as well as inline objects are available in both compressed and uncompressed form. According to Destounis et al. [21], downloading only the compressed objects can save between 30% and 50% of the bandwidth. In our case, the median object size decreased from 10,284 Bytes to 7,338 Bytes, which is a saving of about 29%.

Similar to the increased inline object size, also the size of the main object increased as illustrated in Figure 5(a). While Lee and Gupta observed an object size of less than 12 kB on average, we measured a size of 31.6 kB or 22.5 kB when the content is compressed. Also the maximum main object size is with about 8 MB larger compared to a maximum of 2 MB observed in 2007.

In addition to the inline objects and the main object, several additional documents are integrated in today's web pages, which are loaded from other web servers. While in former times, the complete web page was loaded from one

TABLE II
HTTP MODEL PARAMETERS.

Parameter	Mean	Median	Max	Standard Deviation	Best fit
Main object size Compressed	31,561 Byte 22,468 Byte	19,471 Byte 11,535 Byte	8 MB 8 MB	49,219 Byte 41,295 Byte	Weibull (28242.8,0.814944) Weibull (19104.9,0.771807)
Number of main objects	2.19	1	212	2.63	Lognormal $\mu = 0.473844, \sigma = 0.688471$
Inline object size Compressed	23,915 Byte 21,208 Byte	10,284 Byte 7,338 Byte	8 MB 8 MB	128,079 Byte 127,979 Byte	Lognormal $\mu = 9.17979, \sigma = 1.24646$ Lognormal $\mu = 8.91365, \sigma = 1.24816$
Number of inline objects	31.93	22	1920	37.65	Exponential $\mu = 31.9291$
Reading time [14]	39.7 s	-	10,000 s	324.92 s	Lognormal $\mu = -0.495204, \sigma = 2.7731$

server, containing one main object or if frames are used some additional documents, the web pages are now created with documents from different web servers. The main reason for this are social networks. Several web pages now contain ”like it” buttons from facebook, ”tweets” from twitter, etc. These are directly loaded from the social networks, e.g. ”like.php” or ”tweet_button.html”. We think that the number of documents integrated into the web page will increase even more, especially when looking at dynamically created web pages such as i-google, where the content is combined according to the user’s preference.

The number of these integrated documents is shown in Figure 5(b). The x-axis shows the number of documents including the main object using a logarithmic scale. Although 60% of all web pages just contain the main object, 10% of all web pages are composed of more than 5 documents. While on average, 1.2 documents with an average size of 22.85 kB are integrated in addition to the main object, the largest number of integrated objects was 211. This underlines the trend towards dynamically created web pages, but necessitates similar download times from all requested web servers.

The measured web page parameters are listed in Table II with statistical analysis in terms of mean, median, and variance. In addition, Table II provides the best fitting functions for the parameters including the estimated parameters. The fitting of the main and inline object size is also shown in Figure 6.

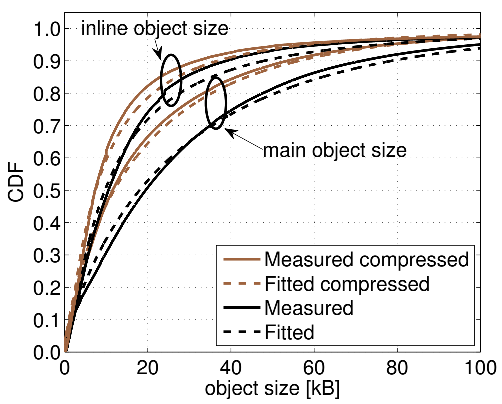


Fig. 6. Fitting of main and inline object size.

The main object size, both in compressed and uncompressed form can be best fitted with a Weibull function. Using a Weibull distribution to fit the inline object size has a larger error than using the Lognormal distribution. Although it does not fit perfectly due to some very large inline objects, the error is small enough so that all fittings can be used for simulating or emulating web browsing users. To still be able to model a web browsing user with an on/off process, we recommend to use the reading time from Lee and Gupta [14] with the parameters listed in Table II.

V. Conclusions

In this paper, we presented a new traffic model for HTTP traffic. In contrast to other web traffic models, which are based on measurements on campuses or access networks of small ISPs, our model is based on measurements of the most visited web pages. Thus, our model contains web page statistics accessed from web users all around the world and not limited to one country or even one campus.

According to our observations, we find that there is a trend towards larger web pages with an increasing number of inline objects. The second finding is that today’s web pages are not loaded from a single web server but created gathering content from all around the world. This fact is especially interesting when looking at the user QoE. If the content is loaded from different web servers, also the delay for downloading the parts of the web page varies, which has an impact on the perceived QoE of the user. Although this is not interesting for web server and cloud dimensioning, the new features integrated into today’s web browsers are. We found that it is possible with some web browsers to download the content of linked web pages automatically after the requested web page is downloaded. This causes a lot more traffic when the user requests a web page as not only a single web page is downloaded but several linked web pages. In future work, we will take a look at the spread of those link tree downloads and will also include the embedded multimedia content in our model.

Acknowledgments

This work was also partially supported by the COST TMA (European Cooperation in Science and Technology) framework.

References

[1] Pries, R., Wamser, F., Staehle, D., Heck, K., Tran-Gia, P.: Traffic Measurement and Analysis of a Broadband Wireless Internet Access. In: IEEE VTC Spring 09, Barcelona, Spain (2009)

[2] Pries, R., Wamser, F., Staehle, D., Heck, K., Tran-Gia, P.: On Traffic Characteristics of a Broadband Wireless Internet Access. In: Next Generation Internet Networks 2009 (NGI 2009), Aveiro, Portugal (2009)

[3] Cisco Systems Inc.: Cisco Visual Networking Index - Forecast and Methodology, 2010-2015. White Paper (2011)

[4] Wamser, F., Pries, R., Staehle, D., Heck, K., Tran-Gia, P.: Traffic Characterization of a Residential Wireless Internet Access. Special Issue of the Telecommunication Systems (TS) Journal **48: 1-2** (2010) 1–13

[5] Choi, H.K., Limb, J.O.: A Behavioral Model of Web Traffic. In: International Conference of Networking Protocol 99 (ICNP 99), Toronto, Canada (1999) 327–334

[6] Popa, L., Ghodsi, A., Stoica, I.: HTTP as the Narrow Waist of the Future Internet. In: Ninth ACM SIGCOMM Workshop on Hot Topics in Networks (HotNets), Monterey, CA, USA (2010)

[7] 3GPP2 C.R1002-0, V..: cdma2000 Evaluation Methodology (2004)

[8] IEEE 802.16-2004: IEEE Standards for local and metropolitan area networks - Part16: Air Interface for Fixed Broadband Wireless Access Systems (2004) IEEE 802.16-2004.

[9] Fielding, R., Gettys, J., Mogul, J.C., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T.: Hypertext Transfer Protocol – HTTP/1.1 (1999)

[10] Natarajan, P., Baker, F., Amer, P.D.: Multiple TCP Connections Improve HTTP Throughput - Myth or Fact? Technical Report TR2008-333, Department of Computer & Information Sciences, University of Delaware (2008)

[11] Mah, B.A.: An Empirical Model of HTTP Network Traffic. In: IEEE Infocom, Kobe, Japan (1997)

[12] Barford, P., Crovella, M.: Generating Representative Web Workloads for Network and Server Performance Evaluation. In: ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems, Madison, WI, USA (1998) 151–160

[13] Tran-Gia, P., Staehle, D., Leibnitz, K.: Source Traffic Modeling of Wireless Applications. International Journal of Electronics and Communications (AEÜ) **55** (2001) 1–10

[14] Lee, J.J., Gupta, M.: A New Traffic Model for Current User Web Browsing Behavior. Technical report, Intel Cooperation, Santa Clara, CA, USA (2007)

[15] Zhu, C., Wang, Y., Zhang, Y., Wu, W.: Different Behavioral Characteristics of Web Traffic Between Wireless and Wire IP Network. In: International Conference on Communication Technology (ICCT), Beijing, China (2003)

[16] Butkiewicz, M., Madhyastha, H.V., Sekar, V.: Understanding Website Complexity: Measurements, Metrics, and Implications. In: Internet Measurement Conference 2011, Berlin, Germany (2011)

[17] Ihm, S., Pai, V.S.: Towards Understanding Modern Web Traffic. In: Internet Measurement Conference 2011, Berlin, Germany (2011)

[18] Gebert, S., Schlosser, D., Pries, R.: PALM - A PAcket Level Measurement Framework. (In: under submission)

[19] Pederick, C.: Web Developer. https://addons.mozilla.org/en-US/firefox/addon/web-developer/ (2009)

[20] Bolin, M.: End-User Programming for the Web. Master's thesis, Department of Electrical Engineering and Computer Science Massachusetts Institute of Technology (2005)

[21] Destounis, P., Garofalakis, J., Kappos, P., Tzimias, J.: Measuring the Mean Web Page Size and Its Compression to Limit Latency and Improve Download Time. Internet Research **11**(1) (2001) 10–17