On the Variation in Web Page Download Traffic Across Different Client Types

Sean Sanders
Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, NC 27514
ssanders@cs.unc.edu

Jasleen Kaur
Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, NC 27514
jasleen@cs.unc.edu

***Abstract*—Modern webpages are diverse and complex. There is also a wide range of devices, operating systems, and browsers that users use to access these webpages. In this work, we study how webpages, and the traffic generated by their download, differ across these different client types. We conduct a preliminary study that performs a client-side analysis of the network traffic. We identify both expected and unexpected differences among similar webpages across different browser platforms that can be used to drive future internet measurement research and identify potential design decisions and/or bugs in modern browsers.**

***Index Terms*—Traffic Characterization, Webpage Measurement**

I. Introduction and Background

Network traffic generated by webpage downloads is becoming increasingly difficult to analyze due not only to the increased complexity of webpages, but also the *increased diversity of clients*. Indeed, modern users rely on several popular browsers, devices, and operating systems for downloading a webpage. Given this diversity, we ask the question—does the traffic generated by the download of a webpage differ across clients using different browsers, operating systems, vantage points, and devices? A detailed answer to this question can help interpret large-scale web measurement projects that are collected on aggregate traces. For example, is the increase in objects observed on a network over time the result of increased complexity of webpages or due to some client platform? This study can also be used to identify components of webpage traffic that are independent of client platform, and determine whether web measurements can be *reliably* used for user profiling, network planning, and traffic engineering.

Methodologically, numerous studies have analyzed web traffic logs collected on highly-aggregated links or proxies [5], [6], [4], [7]—while such aggregated data sources offer rich and voluminous data, they are not suitable for our goal of analyzing the impact of client diversity on the webpage download event, for the following reasons:

- Due to privacy concerns, as well as with the advent of encryption in HTTP 2.0, mostly anonymized TCP/IP headers are available for traffic log analysis. Information about browsers, operating systems, and client locations is not obtainable from such data.
- The numerous objects that comprise a typical webpage are distributed across multiple sources and it is fairly challenging to group them into individual webpage units. There has been some recent success at identifying web pages within HTTP traffic [6]—but the techniques used are not suitable for applying to traces in the wild.[1]
- Proxy services and multi-window browsing makes it difficult to differentiate among simultaneous webpage downloads.

To avoid the above hurdles, we study the webpage download events from the *client-side*, where the agent of generating web traffic is known (browser, webpage, and operating system). We analyze client-side webpage traffic from the point of view of network protocols, with the objective of finding differences between web pages as viewed by clients using different browsers, vantage points, and operating systems – in this paper, we focus on the analysis of different browsers. The results can be used to drive future internet measurement research and identify potential design decisions, privacy concerns, and/or bugs in modern browsers.

The study most related to ours is [3]. It generates and measures landing page traffic from the client side via the Firefox browser (i.e., using the Firebug plugin for traffic measurement). Our work uses a similar measurement methodology but differs from this work in the following ways.

- *Analysis across multiple protocol layers:* We investigate, in a single measurement context, what is required to load landing and nonlanding web pages at multiple protocol layers including HTTP and TCP/IP.
- *Analysis of impact of browser/platform:* Our work analyzes traffic as generated across 5 modern desktop browsers including Internet Explorer, Mozilla Firefox, Google Chrome, Opera, and Safari. We intend to expand this analysis to include multiple operating systems, vantage points, and personalized webpages.

II. Data Collection Methodology

Our goal is to study webpage traffic and analyze how it is impacted by different browsers, operating systems, and other client-side mechanisms. In this section we describe our data collection methodology.

a) Selection of Webpages to Study: There are nearly a billion websites on the Internet, with numerous web pages each. For this study, we focus on the top-250 most popular websites in the United States [1]—a recent study of DNS usage shows that the top-250 domains account for 99% of requests observed in a residential network [5].

We browse each of these websites to collect a list of URLs for their landing as well as non-landing pages, including search

[1] [6] relies on HTTP headers, which are rarely available, due to privacy concerns and/or encryption.

results, media content, and mobile webpages.[2] Overall, we include a list of 3210 webpages, all belonging to the top-250 websites.[3]

b) Trace Collection: We load each of the 3210 pages using the five different most popular web browsers [2] with default settings—these include Internet Explorer (IE) v 9.0.8112.16502, Firefox v 23.0.1, Google Chrome v 29.01547.66m, Opera v 12.16, and Safari v 5.1.7. These browsers were installed and run on a desktop machine running Microsoft Windows 7.

We clear the browser and DNS resolver cache after each measurement. We have 3210 webpages $\times$ 5 browsers $\times$ 2 measurements each, over a period of 6 weeks between March 2, 2014 and April 17, 2014. This results in trace collection for more than 32,000 webpage downloads. Our trace collection process using *automated scripts* is summarized below.

1) Clear the DNS resolver cache and browser cache[4]
2) Start packet capture tool (i.e., windump)
3) Start a browser with a web page URL i as an argument to collect traffic measurements with empty DNS resolver and browser cache
4) Close the browser after 120 seconds
5) Close the packet capture tool
6) Increment $i = i + 1$ and go to Step 1

III. Impact of Browsers on Webpage Traffic

Conventional wisdom is that if two different browsers are used to render the same webpage, they may differ in the extent to which they use pipelined and parallel TCP connections, but not in much else. Suprisingly, we find other differences, as explained below.

a) Expected differences across browsers: Most of the expected differences between browsers are related to the way TCP connections are managed(Fig 1 (a)-(b)). [7] found that a fraction of TCP connections were terminated after a predefined threshold amount of time. This observation was attributed to the browsers closing persistent connections that were idle for a specified amount of time. The plot that corresponds to the longest 90% of TCP connections in Fig 1 (a) confirms that this behavior is influenced by the browser and that it varies according to different browsers. Firefox has two observable timeouts at 5s and 30s, while Opera has more subtle timeouts at 15s, 20s, and 30s. Internet Explorer and Chrome have the most dramatic and diverse TCP connection timeouts of all browsers with clear timeouts present at 5s, 15s, 20s, and 30s.

Fig 1(b) also shows a plot of the number of TCP connections established when rendering a webpage. The first thing to note about this plot is that the number of TCP connections used to render a webpage is substantially different across browsers. [8] observed a similar difference across browsers. This can be attributed to the fact that browsers differ in the way they manage TCP connections, and they differ in the use of TCP preconnect features.

[2]Our methodology does not capture the fact that some websites present different landing pages to users who are logged in (e.g., facebook.com)—study of such "personalized" webpages is left for future work.

[3]Our methodology only collects traffic from a single vantage point—previous studies, however, suggest that client location does not make much difference to non-temporal traffic features [3].

[4]We do not control for the possibility of intermediate caches.

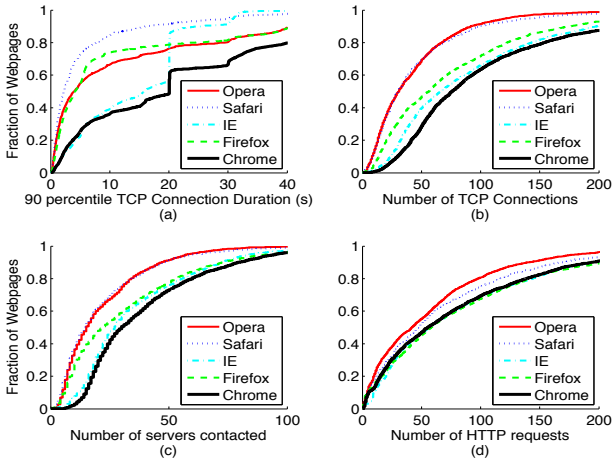

Fig. 1: Cumulative distribution plots showing that browsers have expected and unexpected differences.

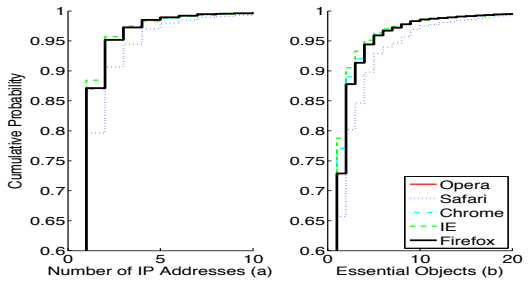

Fig. 2: Plots illustrating that some browsers request redundant objects that may originate from multiple servers.

b) Unexpected differences across browsers: Fig 1 (c)-(d) show that, surprisingly, the number of servers contacted as well as the number of HTTP objects requested, can also differ across browsers.

We hypothesize that there can be two reasons these metrics differ across browsers: (i) some browsers are requesting multiple copies of the *same* object; and (ii) some browsers are requesting objects that are *unique* to that particular browser. To facilitate this discussion, *essential Objects* are defined as objects that have the same URL for each browser for a given webpage. *Nonessential Objects* are defined as objects that use different URLs for the objects across browsers or simply do not occur in our traffic trace for all browsers.

Fig 2(a) shows that Firefox and Safari tend to request more copies of essential objects than Internet Explorer, Chrome, and Opera. Fig 2(b) shows a similar trend where Safari and Firefox access more servers to download more essential objects than the other three browsers. This behavior could be for two possible reasons—one could be that browsers request multiple copies of objects for redundancy, in case one request fails or is too slow; the other could be that this is simply an implementation bug.

Although the evidence that identifies some redundancy of essential objects across browsers explains some of the variation in objects we observe, it does not explain why Internet Explorer tends to contact more servers than Firefox.

Fig 3 (a) and Fig 3(d) show that the contribution of essential objects to the difference in the number of servers contacted is negligible. Fig 3 (b) and Fig 3(e) show that the nonessential

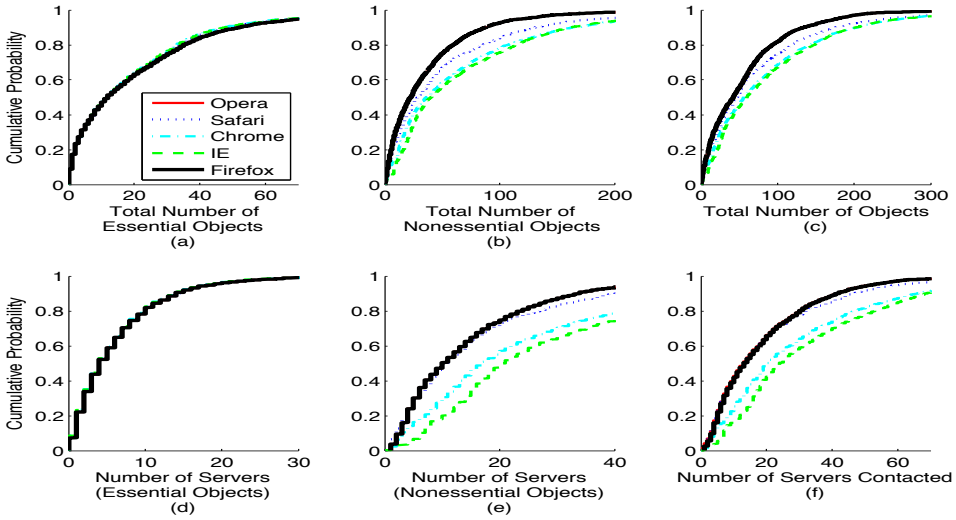

Fig. 3: Plot explaining the differences between browsers.

TABLE I: Variation in frequency in hostnames contacted for nonessential objects (Opera=O;Safari=S;Chrome=C;Internet Explorer=I;Firefox=F)

Hostname	I	C	F	S	O
akamaihd	10647	10288	0	53	66
adnxs	5483	4048	9524	890	555
txtsrving	4451	3521	0	0	0
serving-trk	2826	3647	0	0	0
adplats	1300	1512	0	0	0
exoclick	1060	974	0	0	0
admailister	0	0	1653	0	0
adgrx	0	0	650	0	0
opera.sitecheck	0	0	0	0	9124
doubleclick	7546	7691	5670	7882	5293
moatads	2256	3062	2371	2483	1327
vaccint	0	0	2345	0	0
geotrust	0	0	1089	0	88
scorecardresearch	4521	5467	3755	6640	2497
superfish	1368	1223	4902	0	0

objects seem to identify most of the differences that were initially observed in Fig 1. Furthermore, the similarities between the right and middle plots in Fig 3 show that the differences in the number of servers contacted occur from nonessential objects.

To investigate the source of this seemingly systematic difference across browsers, we analyze the source (hostname) of each nonessential object. Notable differences in the frequency in hostnames being contacted is shown in Table I. It shows that many of the nonessential objects come from hostnames that tend to favor certain browsers. For example, the hostnames txtsrving, serving-trk, adsplats, and exoclick each account for a significant amount of objects in our cumulative trace for the Internet Explorer and Chrome browsers, but are not observed for the other three browsers. While many of the top hostnames that are exclusive to browsers are largely shared by Internet Explorer and Chrome, Firefox also requests nonessential objects from hostnames that are not accessed by any other browser. These hostnames include adgrx, admailister, and vaccint. It is also important to note that many of the hostnames listed in Table I correspond to tracking (i.e., exoclick, doubleclick, scorecardresearch, and superfish) or ad services(i.e., adsplats, admailistr, and adnxs). These results imply that *many ad services and tracking tools are browser specific.* [3] noted that ads and tracking services account for approximately 33% of the traffic for webpages generated using the Firefox browser. Thus, it is not surprising that there is such a large difference between browsers given that these nonessential type objects account for so much traffic share. We also want to stress that there still are many common ads and tracking services that occur across all browsers—examples include doubleclick and moatads.

IV. Summary and Future Work

We use a comprehensive methodology for studying webpages by (i) studying a diverse set of popular webpages that span multiple webpage categories, (ii) measuring webpage traffic at multiple layers, (iii) generating traffic using multiple browsers. Results show that there are both expected and unexpected differences across browsers, which motivates further study in other differences that utilizing different clients can have on webpage download traffic. We plan to extend this work along several dimensions, including incorporating multiple client locations, personalized webpages, AJAX content, multiple operating systems, and tablets/smartphones.

References

[1] Alexa. http://www.alexa.com. Accessed: 2013-02-19.
[2] Statcounter. http://gs.statcounter.com/. Accessed: 2013-06-30.
[3] M. Butkiewicz, H. V. Madhyastha, and V. Sekar. Understanding website complexity: measurements, metrics, and implications. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 313–328. ACM, 2011.
[4] T. Callahan, M. Allman, and V. Paxson. A longitudinal view of http traffic. In *Passive and Active Measurement*, pages 222–231. Springer, 2010.
[5] T. Callahan, M. Allman, and M. Rabinovich. On modern dns behavior and properties. *ACM SIGCOMM Computer Communication Review*, 43(3):7–15, 2013.
[6] S. Ihm and V. S. Pai. Towards understanding modern web traffic. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 295–312. ACM, 2011.
[7] B. Newton, K. Jeffay, and J. Aikat. The continued evolution of the web. In *Modeling, Analysis and Simulation of Computer Telecommunications Systems, 2013. MASCOTS 2013. 11th IEEE/ACM International Symposium on*. IEEE, 2013.
[8] T.-F. Yen, X. Huang, F. Monrose, and M. K. Reiter. Browser fingerprinting from coarse traffic summaries: Techniques and implications. In *Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 157–175. Springer, 2009.