The Impact of HTTP/2 on the Service Efficiency of E-commerce Websites

J. Vihervaara* and T. Alapaholuoma*
* Tampere University of Technology, Tampere, Finland
juha.vihervaara@tut.fi

***Abstract* – Web page download times are important for e-commerce as they affect several business metrics, such as brand perception and conversion. A responsive web service is also desirable for search engine optimization. The widely used HTTP/1.1 protocol does not operate optimally in existing web environments because it suffers from problems such as bloated headers and ineffective use of TCP connections. Therefore, some e-commerce operators have already introduced the latest version of HTTP, which is called HTTP/2. However, a large number of e-commerce operators are still considering if HTTP/2 can produce significant benefits over HTTP/1.1. The HTTP/2 performance has been explored through a number of general-type studies focusing on all types of web sites. Some studies even present that HTTP/2 not necessarily significantly reduce the download times of web pages. In this study, we focus solely on e-commerce sites because e-commerce has better than average access to the benefits of HTTP/2. In addition to the download times of web pages, we investigate HTTP/2's impact on network traffic volumes and server service efficiency. We found that the current method of exploiting HTTP/2 is unable to reduce significantly page download times. The server service capability was also at the same level in both versions.**

Keywords* – HTTP/2*; e-commerce

I. INTRODUCTION

Web performance engineering is primarily dealing with a challenge to minimize the download times of web pages to maximize user experience. Several studies emphasize that it is important for organizations to verify the load times of their web pages. This is particularly important for the e-commerce sector because fast websites achieve profitability. Page speed has a measurable impact on several business metrics, such as brand perception, conversion, revenue, shopping cart abandonments, and page views [1]. Reference [2] presents that the fast download speed of a website promotes visitor loyalty and user satisfaction.

A responsive web service is also desirable for search engine optimization. Google has adopted site speed as part of its search ranking algorithm. This means that the fast site obtains a higher ranking in Google's search listings. Web performance engineers also seek to minimize the amount of data travelling across the Internet. This cuts the energy consumption of the Internet promoting the Green Internet ideology [3][4]. Reduced data transfer volumes are also beneficial for users with slow Internet connections, such as mobile device users.

The widely used HTTP/1.1 protocol (Hypertext Transfer Protocol) does not operate optimally in existing web environments because it suffers from problems such as bloated headers and ineffective use of TCP (Transmission Control Protocol) connections. With the aim of reducing the amount of network traffic and speeding up the download times of web pages, IETF has released a new version of the HTTP protocol called HTTP/2. Because this new version maintains the HTTP semantics, its deployment is easy without requiring any mandatory changes to the existing web page structures. As inspired by this, some e-commerce operators have introduced this new version alongside the old one. However, a large number of e-commerce operators are still considering if HTTP/2 can produce significant benefits over HTTP/1.1.

The HTTP/2 performance has been explored through a number of general-type studies that have focused on all types of web sites. In this study, we focus solely on e-commerce sites. E-commerce has better than average access to the benefits of HTTP/2, because e-commerce utilizes a lot of cookies and downloads relatively more resources from the main site. The previous findings have suggested that the introduction of HTTP/2 should not necessarily significantly reduce the download times of web pages. Because of this, the advantages should also be found on other metrics. In addition to the download times of web pages, we investigate HTTP/2's impact on network traffic volumes and server service efficiency.

The paper is organized as follows. Section 2 describes the backgrounds for HTTP based performance optimization. Section 3 presents our experimental test results. Section 4 draws conclusions from these results.

II. BACKGROUNDS

Over the years, the web has evolved from simple text content from one server to a complex ecosystem with different types of content from multiple servers under different administrative domains [5]. This development has set performance challenges for traditional data transfer protocols and called for their features to be upgraded.

A. Performance Meters

Rendering a web page consist of fetching several varying types of files, often from many different servers. A typical web page includes HTML, CSS, JavaScript, and image files. These files can be called the resources of a

web page. The performance of a website can be evaluated by a variety of meters [6][7]. The number of bytes or the number of files fetched to render a web page can be measured. The number of TCP connections needed to download the resources of the page can also taken into account in the efficiency evaluation. Performance can also be defined as the capability of executing a number of operations within a unit of time. With websites, this means that how many concurrent users the website can serve. However, the most user-centered meter regarding to the performance of a web page is the time it takes to load a page.

The Navigation Timing API [8] provides many performance timings of web page loading. For example, the domContentLoaded event fires after the DOM and CSSOM trees are both ready indicating that the render tree can be constructed. This event does not necessarily require that all images have been loaded. Instead, the domComplete event fire after all of the resources on the page have been fully downloaded even if some of the resources can still be downloaded afterwards in an asynchronous way.

These technical indicators are not always capable of expressing how users experience page downloads. Therefore, analytical tools often offer their own indicators. In this study, we use the following indicators. The Start Render time indicates the moment when something is displayed to the screen. This does not necessarily mean that the user see the relevant content of the page, but it is at least a sign that that the page loading is progressing. The Document Complete time tolerably corresponds with the above-mentioned domComplete event. The term of Fully Loaded are used to the time when all the asynchronous resources are also completely fetched.

B. Drawbacks of HTTP/1.1

In terms of performance, HTTP/1.1 has its limitations [9]. The two main limitations of HTTP/1.1 are ineffective transferred protocol control information and the ineffective use of TCP connections. With extensive use of HTTP, HTTP 1.1 has evolved into a protocol that communicates with a wealth of metadata in its request and response headers to enable to exchange the necessary control information between the client and the server. The headers of HTTP/1.1 are always transferred in plaintext format to maintain compatibility with previous versions of HTTP. In the worst cases, these uncompressed plaintext headers may generate multiple kilobytes of protocol overhead for each HTTP message. In the context of e-commerce, the situation is worse than average because the e-commerce sector utilizes a lot of cookies in its operations. HTTP cookies are massively used for re-marketing, session management, personalization, and web analytics purposes.

HTTP/1.1 is unable to multiplex resources over the same TCP connection. If the browser fetches several resources from the same server, these resources have to be delivered sequentially over the established TCP connection. Efficiency is further reduced by the fact that the client must refrain from sending a new request to the same TCP connection until the reply to the previous request arrives. It is possible to overcome this drawback by using the so-called HTTP pipelining technique but its utilization has remained very limited despite its benefits.

Because of the above factors, the total load time of the resources over the same TCP connection is roughly equivalent to the number of fetched resources multiplied by the round-trip time of the connection. In order to avoid delays caused by this, browsers establish multiple parallel TCP connections to the same server. This solution improves user experience but uses resources in an inefficient way. Opening a new TCP connection consumes network resources especially if TLS (Transport Layer Security) encryption is used, as it is often the case with e-commerce. The server must also maintain state information for each TCP socket. There is always a browser specific upper limit on the number of these parallel TCP connections. This limitation can be circumvented by domain sharding technology, which can distribute the content between multiple host names. In the extreme case, the whole artificially distributed content can still be located behind a single IP address. This workaround consumes the resources of the Domain Name System. There are also a number of other ways to optimize HTTP/1.1 activities related to header inefficiency and the use of TCP connections, including concatenation, spriting, and resource inlining techniques [7].

C. HTTP/2

Since HTTP/1.1 has its limitations, IETF decided to develop the new version of HTTP with the aim of lower application delays and better utilization of the network resources. HTTP/2 was announced in 2015 as an additional binary framing layer between the transport layer and the HTTP plaintext layer [10]. This means that any existing website or application can be delivered over HTTP/2 without modifications. Although with small changes, performance can be somewhat improved since some existing HTTP/1.1 performance optimizations can hamper HTTP/2 performance.

HTTP/2 attempts to minimize the amount of network traffic generated by HTTP control information. Firstly, as the name implies, the new binary framing layer of HTTP/2 works on a binary basis enabling efficient transfer of its own control information. Secondly, HTTP/2 uses a HPACK mechanism to reduce the bit stream of the redundant HTTP headers. The HPACK mechanism uses two tables to reduce the bit stream. A static table involves the cache for the commonly used HTTP request and response header names. A dynamic table references other headers used in the current session. These tables reduces HTTP header sizes by sending integer based cache indexes to the existing cache entries instead of the string literals themselves. In addition, the effective HTTP-specific Huffman encoding of string literals reduces the amount of network traffic.

The new binary framing layer of HTTP/2 enables full request and response multiplexing using only a single TCP connection. Once a TCP connection is established between the server and the client, it can accomplish multiple resource fetches. Each resource fetch forms a

new stream with its unique stream identifier. HTTP/2 is capable of breaking down the HTTP messages of this steam into independent frames. The frames of different streams can be interleaved and maintaining the transmission order between the streams is not required.

In addition to the aforementioned commonly used enhancements, HTTP/2 also offers some other enhancements whose use will become more common in the future. A server push feature allows the server to push useful data to the client without the client's explicit request for it. A priority mechanism makes it possible to the client to specify a certain ranking among concurrent streams on a single TCP connection.

D. Related works

Previous studies have indicated both improvement and degradation in web page load times when comparing HTTP/2 with HTTP/1.1. Reference [11] compared HTTP/2's performance with HTTP/1.1 under various network delay, bandwidth, and loss conditions by cloning the top 20 Alexa websites. In typical circumstances where packet losses are rare and round-trip times are relatively short, they found that HTTP/2 cut page load times only about 20 percent even though they had placed all the cloned web site resources on a single server, omitting domain sharding. Reference [12] took into account the impact of domain sharding by splitting the web resources across the multiple domains. They cloned six typical websites to their multi domain test environment. They found that HTTP/2 did not provide better performance on any of these tested web sites. Due to the limitations of their study, they did not, however, claim that that HTTP/2 lead to the performance degradation in terms of page load time in general.

Reference [13] presents the measurement results of a large number of different types of sites. They found that 80 percent of websites supporting HTTP/2 experience a decrease in page load time compared with HTTP/1.1, and the decrease grows in mobile networks. The average page load time reduction of 500 ms was discovered with the fiber connection and 1.6 s with the mobile connection. The remaining 20 percent see an average page load time increase of 1 s with the fiber connection and 1.4 s with the mobile connection. Reference [14] presents, despite their somewhat limited research, that HTTP/2 is not enough on its own to improve web page load times. They suggest that web sites must be designed with HTTP/2 features in mind in order to fully take advantage of HTTP/2's benefits.

As indicated above, previous research results are some contradictory. However, the studies have provided indications that the benefits of HTTP/2 are most noticeable in mobile environments where data rates are small and packet losses are rare. However, if packet losses are common, HTTP/1.1 usually outperforms HTTP/2 [11][12][15] .

III. PERFORMANCE TESTS AND RESULTS

The performance of HTTP/2 has been explored mainly by general-type studies focusing on all types of web sites. We focus solely on e-commerce sites. In addition to the download times of web pages, we investigate HTTP/2's impact on network traffic volumes and server' service efficiency. This section presents our measurement results.

A. Download times of web pages

In this section, we compare the page load times of HTTP/2 and HTTP/1.1 to each other. The objective was to find out if HTTP/2 could reduce the download times to the extent that users could also detect a better user experience. We selected 15 e-commerce web sites supporting the HTTP/2 protocol as a test set. Supporting the HTTP/2 protocol means that the site can respond to the service requests of both HTTP versions. In order to facilitate the maintenance of web pages, HTTP/2 capable servers usually provide identical web resources for both HTTP versions. At this time, it makes sense to optimize the resources for HTTP/1.1, because HTTP/1.1 is currently the most widely used version.

When choosing these web sites, we tried to create a set that corresponds well with the typical site distribution of a regular user. Due to this, most of the selected sites were major international and national e-commerce sites, but there were also small sites. Since the test situations mimicked normal browsing events, the round-trip times of the TCP connections also corresponded to typical situations. We measured only the download times of the landing pages. When the browser cache was empty, the number of HTTP requests needed to download the web resources of the web page ranged from 38 to 180.

We used the WebPageTest-software [16] to test page download times. It is a free software to execute website speed tests from multiple locations around the globe using real browsers. We used the software version where the test generator was installed on the local workstation. This computer was connected to the web via a high-speed wired LAN connection and, therefore, the loaded page versions were intended for desktop users. The browser version was Google Chrome. The download times of the pages were tested when the browser cache was both enabled and disabled. To get correct results, each test case was repeated five times and the presented test results are the average of these test cases.

Fig. 1 shows the download time comparisons between HTTP/2 and HTTP/1.1. The figure presents the results as a ratio of the page load time of HTTP/2 and HTTP/1.1. If the ratio falls below 1, it means that HTTP/2 performs better than HTTP/1.1. When calculating the ratio, the Start Render, Document Complete, and Fully Loaded times were taken into account. This ratio was 0.97 when the browser cache was empty when downloading the pages. In this case, HTTP/2 managed slightly to shorten the download times. Instead, the download times were somewhat longer when utilizing the browser cache. In this case, the ratio was 1.03.

The test results show that, in general, the current way of applying HTTP/2 is not capable to reduce download times to the extent that it would have a significant positive impact on user experience. Instead, HTTP/2 was able to reduce the number of open TCP connections clearly in all cases. On average, the number of TCP connections decreased by 25 percent when considering the results without caching. As a result, HTTP/2 utilizied opened

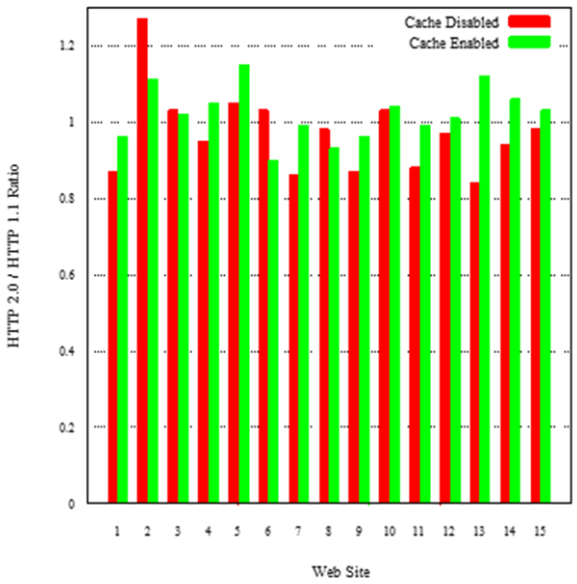

Figure 1. Download time comparisons between HTTP/2 and HTTP/1.1

TCP connections more efficiently than HTTP/1.1. HTTP/2 transmitted 3.4 resource requests via one TCP connection when the corresponding averaged value for HTTP/1.1 was 2.7. Table 1 summarizes the average number of request messages and TCP connections.

B. Network traffic volumes and the service efficiency of servers

In this section, we presents the impact of HTTP/2 on network traffic volumes and server service efficiency. We created an e-commerce based web page for our measurements. The page consisted of 25 resource files.

TABLE I. AVERAGE NUMBER OF REQUEST MESSAGES AND TCP CONNECTIONS

	Number of requests	Number of TCP connections	Requests per Connection
HTTP/2 (no cache)	106	31	3.4
HTTP/1.1 (no cache)	107	40	2.7
HTTP/2 (cache)	46	23	2.1
HTTP/1.1 (cache)	45	25	1.8

These resources were housed on the same server. The processing power of this server was deliberately low so that we could push the server to the limits of its service capability.

We used the Load Impact-tool [17] to generate service requests for our test server. The Load Impact-tool helps determine the capability of handling a certain amount of users on a website. In this study, the amount of concurrent users increased evenly from 1 to 40 during the 8-minute test. We used the collectd-tool to collect metric data. Collectd [18] is a daemon, which collects system and application performance metrics periodically.

Fig. 2 shows the test results. The figure presents the way in which HTTP/2 and HTTP/1.1 increase the network traffic volumes and the CPU utilization rates of the server when the number of simultaneous users on the server increases steadily. The measurement results indicate that there are no significant differences between these two HTTP versions in the network traffic volumes and the CPU utilization rates.

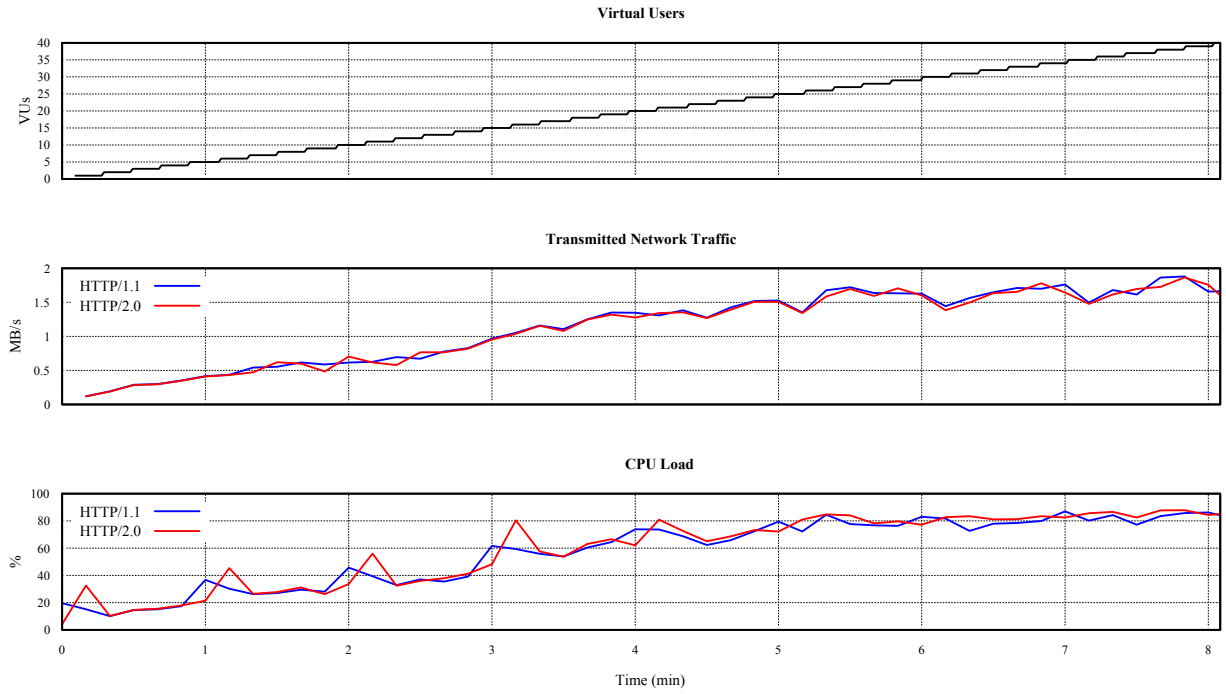

Figure 2. Network traffic volumes and the CPU utilization rates of the server

HTTP/2 is unable to reduce significantly traffic volumes because volumes associated with the protocol control functions are ultimately low compared to the application data transfer. In addition, some HTTP/2 properties affect traffic volumes in the opposite way. Less traffic is needed to open TCP connections. The HPACK-algorithm also reduces the amount of data to be transferred. On the other hand, HTTP/2's own control information adds the number of bits to be transferred. These same factors also explain why the service efficiency of the server is almost identical with both versions. The HPACK-algorithm and the new binary framing layer of HTTP/2 increase the processing load. A lower number of TCP connections reduces the need for processing.

IV. CONLUSION

This study indicates that, in general, the current way of applying HTTP/2 is not capable to reduce the download times of e-commerce web pages to the extent that it would have a significant positive impact on user experience. The existing browsers are able to keep the download times of HTTP/1.1 competitive with HTTP/2 by opening several parallel simultaneous TCP connections to the same server. Therefore, research results like this are always dependent on the browser version used because the maximum number of simultaneous connections varies between browsers. According to the results, the HTTP version used does not have a significant effect on server service efficiency. We can conclude that e-commerce operators do not necessarily have to rush to deploy HTTP/2, but the introduction of HTTP/2 is justified due to the reasons below.

The tested web pages were not optimized for HTTP/2 but rather for HTTP/1.1. This was indicated by the fact that the downloaded pages consisted of the same amount of resource files despite the protocol version used. This implies that e-commerce operators currently offer the same page structure for both HTTP versions. The optimization techniques of HTTP/1.1 may even be detrimental to HTTP/2. In addition, utilizing the server push function will enable HTTP/2 to streamline its operations in the future. It should also be mentioned that this study focused solely on the service efficiency of servers. Introducing HTTP/2 will slightly reduce the load on routers. For example, routers need to make fewer routing decisions because traffic related to opening TCP/TLS connections reduces.

REFERENCES

[1] T. Everts, Time is Money: The Business Value of Web Performance," O'Reilly Media, 2016, 110 pages.

[2] B. Fabian, F. Goertz, S. Kunz, S. Muller, and M. Nitzsche, "Privately waiting - a usability analysis of the tor anonymity network," Sustainable e-Business Management, 2010, pp. 63–75.

[3] A. Bianzino, A. Raju, and D. Rossi, "Greening the Internet: Measuring web power consumption," IT Pro, Published by the IEEE Computer Society, 2011, pp. 48-53.

[4] S. D'Ambrosio et al., "Mobile phone batteries draining: Is green web browsing the solution?," In IEEE International Green Computing Conference, 2014, 10 pages.

[5] M. Butkiewicz, H. Madhyastha, and V. Sekar, "Understanding website complexity: Measurements, metrics, and implications," IMC'11 Proceedings of the ACM SIGCOMM conference on Internet measurement conference, 2011, pp. 313-328.

[6] J. Soininen, Website Performance Evaluation and Estimation in an E-business Environment. Doctoral dissertation, Tampere University of Technology, 2012, 188 pages, Retrieved 12.11.2017 from http://dspace.cc.tut.fi/dpub/handle/123456789/21230.

[7] I. Grigorik, High Performance Browser Networking. O'Reilly Media, 2013, 401 pages.

[8] Z. Wang. Navigation Timing. W3C, 2012, Retrieved 12.11.2017 from https://www.w3.org/TR/2012/REC-navigation-timing-20121217/.

[9] I. Grigorik, "Making the web faster with HTTP 2.0," Queue, Vol. 11, Issue 10, 2013, 40 pages.

[10] M. Belshe, M. Thomson, and R. Peon, "Hypertext transfer protocol version 2 (http/2)," RFC 7540, 2015, 96 pages.

[11] H. De Saxce, I. Oprescu, and C. Yiping, "Is HTTP/2 really faster than HTTP/1.1?," Computer Communications Workshops (INFOCOM WKSHPS), 2015, pp. 293-299.

[12] H. Kim, J. Lee, I. Park, H. Kim, D.-H. Yi, and T. Hur, "The upcoming new standard HTTP/2 and its impact on multi-domain websites," in Network Operations and Management Symposium (APNOMS), 2015, pp. 530–533.

[13] M. Varvello, K. Schomp, D. Naylor, J. Blackburn, A. Finamore, and K. Papagiannaki, "Is the web HTTP/2 yet?," International Conference on Passive and Active Measurements, 2016, pp. 218-232.

[14] L. Bach, B. Mihaljević, and A. Radovan, "Exploring HTTP/2 advantages and performance analysis using Java 9," Proceedings of the 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2017, pp. 1522-1527.

[15] U. Goel, M. Steiner, M. Wittie, M. Flack, and S. Ludin, "HTTP/2 performance in cellular networks," ACM MobiCom, 2016, 3 pages.

[16] WebPageTest. Test a website's performance. Retrieved 2.2.2018 from https://www.webpagetest.org/.

[17] Load Impact. Load Impact: Test your app for scale and performance across regions. Retrieved 2.2.2018 from https://www.ibm.com/cloud/garage/content/run/tool_load_impact/

[18] Collectd, collectd – The system statistics collection daemon. Retrieved 2.2.2018 from https://collectd.org/.