RESEARCH ARTICLE | SEPTEMBER 01 2023

Advancement of efficiency of HTTP2 over HTTP1 a comparative study

Anuj Kumar ✉; Ramnarayan; Rashmi Kuksal; Raja Kumar Murugesan; Vijay Lakshmi Sajwan; Monisha Awasthi

AIP Conf. Proc. 2771, 020060 (2023)
https://doi.org/10.1063/5.0152254

Advancement of Efficiency of HTTP2 Over HTTP1 A Comparative Study

Anuj Kumar[1, a)], Ramnarayan[1, b)], Rashmi Kuksal[1, c)], Raja Kumar Murugesan[2,d)], Vijay Lakshmi Sajwan[1, e)], Monisha Awasthi[1, f)]

[1)] *CSE Department, Uttaranchal University, Uttarakhand, India*
[2)] *CSE Department, Taylor's University, Jalesh, Malaysia*
Corresponding author: [a)]kannojia.anuj@gmail.com
[b)] ram6801@gmail.com,
[c)] rashmikuksal@gmail.com
[d)] rajakumar.murugesan@taylors.edu.my
[e)] vijaylakshmi@live.com
[f)] uumonishaawasthi@gmail.com

Abstract. The innovation and advancement of this digital era is where every physical deed is working remotely and on online platform. Today our generation is surrounded with the plethora of platforms where they can post their data and it will be present in the cyber space for n number of years. The creation of data is increasing, storage volume is increasing and algorithms to recognize the patterns of that data are also getting matured. Not only data but also ways to present the data has kept pace with the enhancement of technologies. If we talk about the websites and web development about 10 years ago, it was a plain HTML and now the backend frameworks have improvised with time. The size of the data on those websites has also increased considerably. There are many aspects such as the assets, thumbnails and third party scripts running on a web page which takes a lot of time for a single web page to load. This increasing data on the server side for the website affects the rendering time at client's end. This paper specifically comprehend on the advancement on protocols over Internet from WWW i.e. W3 consortium. We have focused on HTTP/2 using cookies for gaining potential in terms of page load time for better performance and increase speed.

Keywords: Cookies, HTTP/2, speed, protocols, comparative study.

INTRODUCTION

The websites in today's era have become so profound and user friendly that the large set of data modifications is done at client's end. In contrast to poor and obsolete HTML UI/UX, the websites provides magnificent UI/UX which includes photos, thumbnails, assets, music (mp3,.wav) and even videos. The size of these webpage resting on server is considerably more than the previously used webpages. In addition to that, the target is to load a page with maximum data in minimum bandwidth of internet. To reduce the time space complexity WWW introduced the contemporary protocol HTTP/2 over HTTP/1 and Google's web protocol SPDY. SPDY protocol was implemented on top of Application layer. SPDY has some improvisation over HTTP1.1. HTTP/2 proved to be a successor of SPDY. The basic model and scheme of the protocols and technicalities were taken from SPDY. HTTP/2 is projected to address several flaws and inflexibilities of HTTP/1.1 improving online performance in terms of the time it takes for a page to load[1]. The concept of HTTP/2 evolved with the discovery of patterns what user is watching/opening/surfing/playing over the internet using his web browser. The idea of HTTP/2 is to store the information and pre-requisites (assets, thumbnails, scripts, extensions etc.) on the client's web browser at the first place the client visit the website. The cookie law named as eprivacy Directive is not a compulsory law unless your target audience is based in Europe. The Hypertext Transfer Protocol (HTTP) was created by the Internet Corporation for Assigned Names and Numbers (ICANN) created in the middle of 1990s to facilitate communication between clients/users (web browsers) and web servers[2]. As a result of which many sub-versions of the protocol were

Innovative Development in Virtual and Rapid Prototyping
AIP Conf. Proc. 2771, 020060-1–020060-5; https://doi.org/10.1063/5.0152254
Published by AIP Publishing. 978-0-7354-4627-4/$30.00

developed. Furthermore, Google taking this technology as a next step in 2009 began development SPDY is an experimental protocol that has been used in the past. Not only SPDY worked similarly like HTTP's semantics but also resolved the HOL (Head of Line) blocking issue. The HOL proved to be a major limitation in HTTP/1.1. The Internet Engineering Task Force (IETF) suggested tackling the description of a latest HTTP/2 is a new HTTP version in 2012. The SPDY protocol was used in the initial HTTP/2's proposal in 2012. Since then, it has gone through a lot of changes, all with the goal of making HTTP/2 load web pages faster.[3]

In this paper we have talked precisely upon the analysis of HTTP/2 over existing protocols (HTTP/1 and SPDY). The problems in the existing protocols are discussed in detail and what advancement has been done is also analyzed to impeccably compare the protocols. The setup we have used for this analysis is a sample website made by us which comprise code of JavaScript, HTML, CSS, jQuery and Bootstrap. There will be some scripts as well as HD images and videos will also be included under the assets of the web page[4]. In comprehension to that, the use of cookies in HTTP/2 is also scrutinized as though it decreases the page load time but increases the risk of security if one's particular information over web. It is also obvious that with the shift in the protocol, it will give eave droppers and attackers to find the vulnerability in the updated version of HTTP.

LIMITATIONS OF HTTP/1

HTTP 1.1 had a major drawback of handling multiple requests at the same time[5]. This means that in an elementary request scenario, HTTP1 works perfectly fine but in case of pipelined request, the protocol fails to process the second request if the response to first request is hindered. Another problem with HTTP/1 is of HOL i.e. Head of Line package. HTTP1.1 uses a TCP connection to communicate with a server. Once a successful connection has been made over TCP in conjunction with the server, client is able to make requests to the server that is GET requests to get the content. In case the client desires innumerable information from the server, they must make multiple GET queries on a stable connection through TCP, one after the other. Furthermore, before submitting another up to date request, the client must wait for the server to respond to the previous request.

Pipelining was implemented as a first option to avoid the aforementioned stumbling block when a client sends two queries (GET 1 and GET 2) without first waiting for a response. This means that at the same instance of time the server can get two or more GET request from the client independent of each other. If a request is made in a pipelined method form then responses also occur in line. First the response of 1st request will be shown following will be the response from the 2nd request. This is a lot better plot, but the Head of Line (HOL) blocking is a problem still persists. For example, if the first response is a bulky file containing lots of media and assets then it will take a long time to transfer to the client and future responses will be delayed.

In the above mentioned scenario the first request in line will automatically hinder all the others, this event is referred to as the "head of the line" blockage. Some attempts were made to avoid HOL blockage such as allowing a customer to establish multiple TCP connections to the same server are the most common simplest option. However, because the server must keep a large number of the states relating to numerous TCP connections, this technique is costly[6]. As a result, browser manufacturers agreed to limit the total number of connections to only one server to six for Google Chrome and fifteen for Mozilla Firefox. Domain sharing was another experiment to alleviate the detrimental the upper bound's effect depending on the total number of TCP connections. It entails dividing the content across numerous servers, each with a unique IP address. As a result, the client can create numerous TCP connections in parallel to various servers and retrieval multiple components of the same web page at the same time. However, a browser's number of TCP connections total (again, including all tabs) capped.

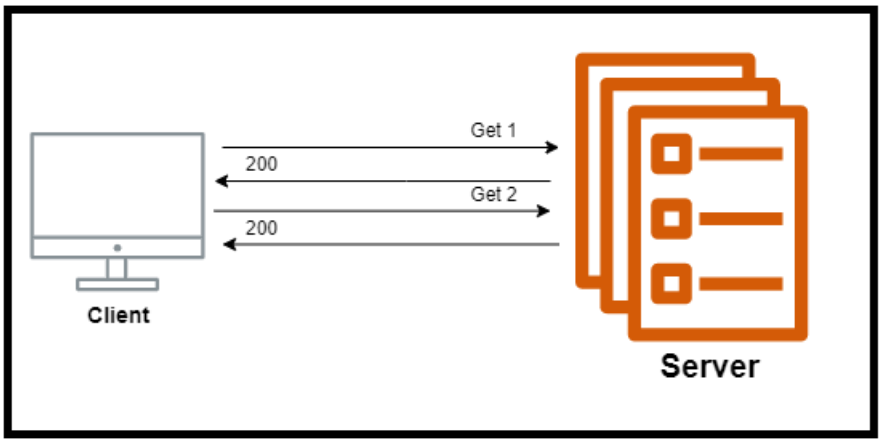

Fig 1. Elementary requests on a TCP connection over HTTP/1.

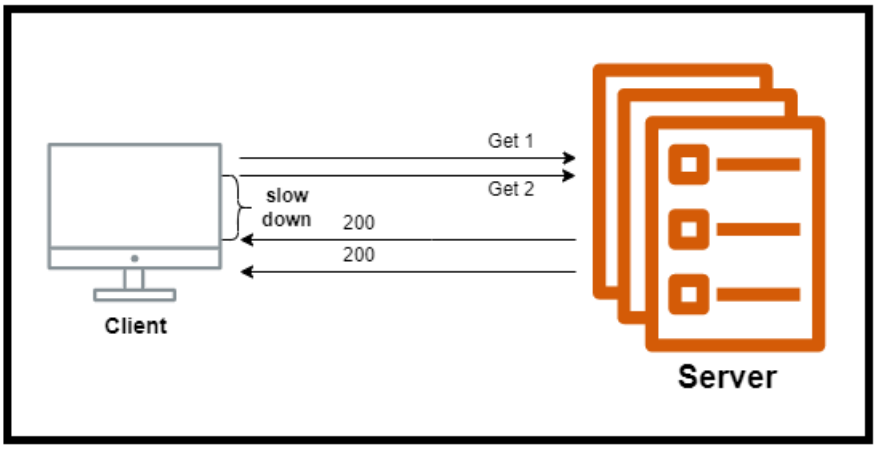

Fig 2. Pipelined requests on a TCP connection over HTTP/1.

ENHANCED FEATURES OF HTTP2

HTTP2 had an advantage of Binary protocol. In HTTP the protocol was text based but in HTTP2 binary structure was introduced which was more efficient in terms of parsing[7]. Services like telnet and plain text parsers cannot be used on HTTP2. The syntax for headers remains the same in HTTP/2 because HTTP/1.1's semantics are as follows: preserved. Likewise, these HTTP heads constantly have the same ideals, performing in a state of redundancy. This is why the HTTP/2 protocol was created. Standards include the HPACK algorithm, which illustrates the medium to compress HTTP heads. Huffman encoding is used in HPACK along with two tables: one static table that stores variants of compressed frequently used headers, and another dynamic table that refers additional compressed headers that are session-dependent[8]. HTTP/2 proposes the concept of stream to address Head of line blocking is a problem that has to be addressed; each client request is assigned to a specific person. Specific Streams are multiplexed, and all streams are multiplexed across TCP connection (single)[9]. As a result, requests are not accepted interfere with one another and the server can respond to them all at the same time. As a result, HTTP/2 eliminates head of line blocking while also lowering a count of TCP connections that must be taken care of on the server's end.

TABLE 1. Different assets load time for widely used websites.

S. No.	Website	JS	CSS	Images	Media	HTML
1	Google	1	2	3	1	2
2	Youtube	2	3	4	4	2
3	Facebook	2	1	1	2	1
4	Twitter	3	2	4	2	1
5	Wikipedia	2	2	2	2	1
6	Yahoo	4	2	2	1	1
7	Amazon	6	2	8	4	2
8	Instagram	6	7	5	3	2
9	Yandex	4	3	3	3	3
10	Reddit	3	2	1	2	2
11	Bing	4	2	2	1	1
12	Naver	2	2	2	1	1

.Using the priority mechanism, a user can provide a specific ranking with a single HTTP/2 connection, many streams. Now it totally depends on the server to keep this or disregard the system entirely; however a server is allowed to prioritize some files(JS, CSS & HTML) over images, resulting in faster time it takes for the page to load. In the structure of a web page, this is true pieces of code that refers to other dependencies, such as graphics, images and media must to be retrieved first. As a result, They can be scanned as soon as feasible by the browse rand check which photos are being referred to which web page, and issue the appropriate requests as quickly as possible. HTTP/2 enables to the server send profitable data to the customer[10]. Table 1 displays the load time (in seconds) of different constituents of a website. Top 12 websites according to global ranking 2022 were taken under consideration. The client can then refuses or accept the information, which will subsequently be saved for future use in the browser's cache. The use of cookies in HTTP2 is for the same purpose to store pre-fetched data at client's end which will enhance page load time.

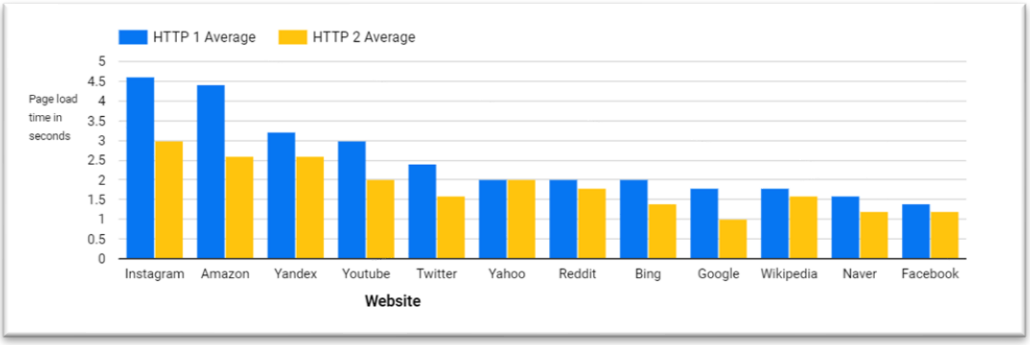

Fig. 3. Page load time (in seconds) on two versions of HTTP.

CONCLUSION

We must remember that this paper is a proven piece of work. The goal of this project was to analyze efficiency ratio between HTTP1.1 and HTTP/2.We calculated the superiority, efficiency and sustainability of HTTP/2. Definitely there is an urge to test the new and advanced protocols more rigorously. The prime the goal of this research was to assess the impact of HTTP/2's adaptation to the existing Internet world. Since the data and the graphics on the

websites today are in huge demand, many organizations focus on A/B testing of their websites for a better UI/UX hence improving the size of the website. But we found HTTP/2 generally has faster page load time. All the appreciation goes to the compression and multiplexing features offered by the protocol. On the fully equipped realistic web pages designed for testing and under simulated and real internet settings, HTTP/2 spectacularly kept up its performance high than the existing protocols. Definitely the use of HTTP/2 has a better scope under the aegis of speed and better page rendering. Also keeping the efficiency of new protocol in consideration we can make the web pages more attractive without thinking much on heaviness of the page in terms of data. Moreover, the graphics and backgrounds scripts can be added for a better UI/UX until HTTP/2 is used.

REFERENCES

1. Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., & Berners-Lee, T. (1999). Hypertext transfer protocol–HTTP/1.1.
2. Belshe, M., Peon, R., & Thomson, M. (2015). Hypertext transfer protocol version 2 (HTTP/2).
3. Rescorla, E., & Schiffman, A. (1999). The secure hypertext transfer protocol. *IETF Request for Comments, RFC, 2660*.
4. Belshe, M., Peon, R., & Thomson, M. (2015). Hypertext transfer protocol version 2 (HTTP/2).
5. Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., & Berners-Lee, T. (1999). RFC2616: Hypertext Transfer Protocol--HTTP/1.1.
6. Rescorla, E., & Schiffman, A. (1999). The secure hypertext transfer protocol. *IETF Request for Comments, RFC, 2660*.
7. Chowdhury, S. A., Sapra, V., & Hindle, A. (2015). Is HTTP/2 more energy efficient than HTTP/1.1 for mobile users?. *PeerJ PrePrints*, *3*, e1280v1.
8. Xiao, M., Swaminathan, V., Wei, S., & Chen, S. (2016, May). Evaluating and improving push based video streaming with HTTP/2. In *Proceedings of the 26th International Workshop on Network and Operating Systems Support for Digital Audio and Video* (pp. 1-6).
9. Liu, Y., Ma, Y., Liu, X., & Huang, G. (2016, June). Can HTTP/2 really help Web performance on smartphones?. In *2016 IEEE International Conference on Services Computing (SCC)* (pp. 219-226). IEEE.
10. Siduzzaman, M., Hossan, M. M., Alom, R., Sarwar, T. B., & Miah, M. S. U. (2020, May). Performance comparison of HTTP/2 for Common E-Commerce Web Frameworks with Traditional HTTP. In *Journal of Physics: Conference Series* (Vol. 1529, No. 5, p. 052023). IOP Publishing.