Article title: The Effects of HTTP Advances on Modern Page Loading

Authors: Jon Pendon[1], Tejas Warishe[1]
Affiliations: new jersey institute of technology[1]
Orcid ids: 0009-0005-8045-1139[1]
Contact e-mail: pendon.jon@gmail.com

Links to data: https://github.com/Pendon13/CS656-Project
DOI: 10.14293/PR2199.000570.v1
Preprint first posted online: 20 December 2023
Keywords: HTTP, Protocol design, Web Performance

The Effects of HTTP Advances on Modern Page Loading

Jon Pendon, *NJIT*, Tejas Warishe, *NJIT*

***Abstract*—Since the dawn of the internet, protocols have seen massive innovations that have improved our experiences. The most prominent of these advances are in the application layer space with HTTP. HTTP has moved from HTTP 1.1 to HTTP 2 and now HTTP 3. The second most prominent occurred one layer higher in the transport layer. The main one being the shift from loss-based congestion control to delay-based congestion control. This paper argues that due to these changes in TCP settings, sometimes, the improvements from HTTP 1.1 to 2 or even 3 have seen lower web performance. Utilizing automated web drivers to load web pages, this paper finds the amount of time it takes to load web pages and quantifies it as a web performance metric. A script is created to crawl through the websites provided and records the amount of time it takes to load the web page's resources. This paper analyzes 50 websites to start but is able to scale to as many as needed. From these 50 websites, 9 websites perform worse using HTTPS 1, 2 and 3 as the main protocol and 14 websites perform worse using HTTP 1, 2, and 3. This information contradicts some background papers that show a large portion of websites would perform better using HTTP 1 only. However, these papers were concluded many years ago. Since then, web servers have upgraded their servers and protocols have been optimized. In conclusion, this information is useful in understanding how individual web sites interact with the network protocols and can help aid in protocol development.**

***Index Terms*—HTTP, Protocol design, Web Performance**

I. INTRODUCTION

The innovation of new protocols in the web space has had a lasting impact on many of our experiences using the web. The internet has seen HTTP change from HTTP 1.0 all the way to the newest set of protocols: HTTP 3.0 [12]. With it, comes challenges that have been relatively unseen in the network space [11]. HTTP 2 [9] introduced header compression, multiplexing into a single TCP connection, and server pushes. HTTP 3.0 takes a new spin on protocol and uses QUIC as its basis. QUIC has its roots in UDP and lets it start transmitting data immediately. It is designed to be more resilient to packet loss than TCP and uses streams rather than packets. Overreaching all of these protocols is HTTPS. HTTPS is a secure protocol used to securely transfer data.

In parallel to HTTP's innovations, its parent policy TCP has also seen a lot of innovation. TCP has moved on from loss based only congestion control algorithms to delay based congestion control algorithms. Some examples of delay based TCP are Google's BBR [7], CUBIC's Hybrid Slow Start [8], and YeAH [10]. These algorithms rely on packet delay measurements instead of packet loss measurements as their signal.

Studied in isolation, HTTP and TCP innovations have shown higher performance and great potential. There is not much downside to switching to these better protocols and the industry has taken notice of it. Many of today's websites employ some form of HTTP 3 in their network protocol stack [6]. Some objects use it on their websites. In addition, a majority of website servers have switched to delay based algorithms for TCP.

Work in this paper stems off of prior work done in the paper: Mind the Delay: The Adverse Effects of Delay-Based TCP on HTTP [1]. As this paper was published in 2020, a large number of advances have been made in the HTTP space. HTTP 3 has become somewhat prevalent and noticeable on websites. Thus, this paper will conduct experiments regarding web performance. The main contribution will be on the metric OnLoad [5]. This metric is considered the amount of time it takes for the web page to load at a normal page loading strategy.

OnLoad simulates what a user will most likely see. It gets the full resources of the page. Most web users will have this metric affect them, so it is vital to understand and calculate.

A. Interplay Explanation

In Mind the Delay: The Adverse Effects of Delay Based TCP on HTTP, the interplay between HTTP 1 and HTTP 2 show that HTTP 1 is better in scenarios with larger page sizes and cellular style networks. This is due to HTTP 1's initial connection establishment. It establishes more than one connection allowing for more objects to be sent in that initial window. In HTTP 2, this connection establishment only has 1 connection. Due to this, HTTP 2 works worse. The same can be said for HTTP 3.

Some cases where HTTP 2 performs better than HTTP 1 can be attributed to other aspects. These aspects include head of the line blocking and the actual set up overhead of HTTP 1. This makes it difficult to quantify the ideal protocol that should be used to send the data.

II. Experiment Methodology

Work done on this paper has been completed using simple tools available for students. Technology used is free and available.

Client: The experiment uses Brave browser [2] as its main browser to load web pages. Brave provides third party tracking innately and ad blocking. These resources enable Brave to have replicable results and remove any traces of ads or third party cookies changing the times for page loads. Primary content of the page can then be studied and tested.

As Brave is a chromium browser, it provides Chrome DevTools [3] innately. Chrome DevTools is used to determine and guarantee the HTTP protocols that are being used. In addition, this paper will discuss differences between HTTP and HTTPS. This change is done via the access of the website. For instance using https://www.google.com is different from using http://www.google.com.

Chromium browsers also provide two important flags to isolate HTTP protocols. These flags are *-disable-http2* and *-disable-quic*. These flags disable HTTP 2 and HTTP 3 protocols respectively. In order to test this, Brave browsers with either one or two flags enabled are opened and checked in the Chrome DevTools Network Tab. In this tab, the protocols can be extracted and seen that when a flag is present, that protocol is not present.

Network: The experiment mainly uses a home network for testing. Experiments tested are done through a WiFi connection to a home modem. This home modem connects to the outside ISP. The laptop is very close to the router providing minimal environmental interference.

Automation: In order to automate website access, this experiment utilizes the web driver known as Selenium [4]. Selenium allows web browsing automatically while also providing a way to attach chromium flags and other necessities.

Utilizing Selenium, the program provides paths to the Brave browser and chrome web driver to allow the program to work on Chromium browsers. Options for *-disable-http2* and *-disable-quic* are utilized here and put into the chromium executable. The experiment uses a page load strategy of normal as well. This page load gets all data from the website and does not continue until all resources are loaded in. This is considered our metric OnLoad time.

To get round trip times for websites, the program times the amount of time it takes to get the website url from a browser that is already opened with a different website. This time difference is what we use to generate our RTT data.

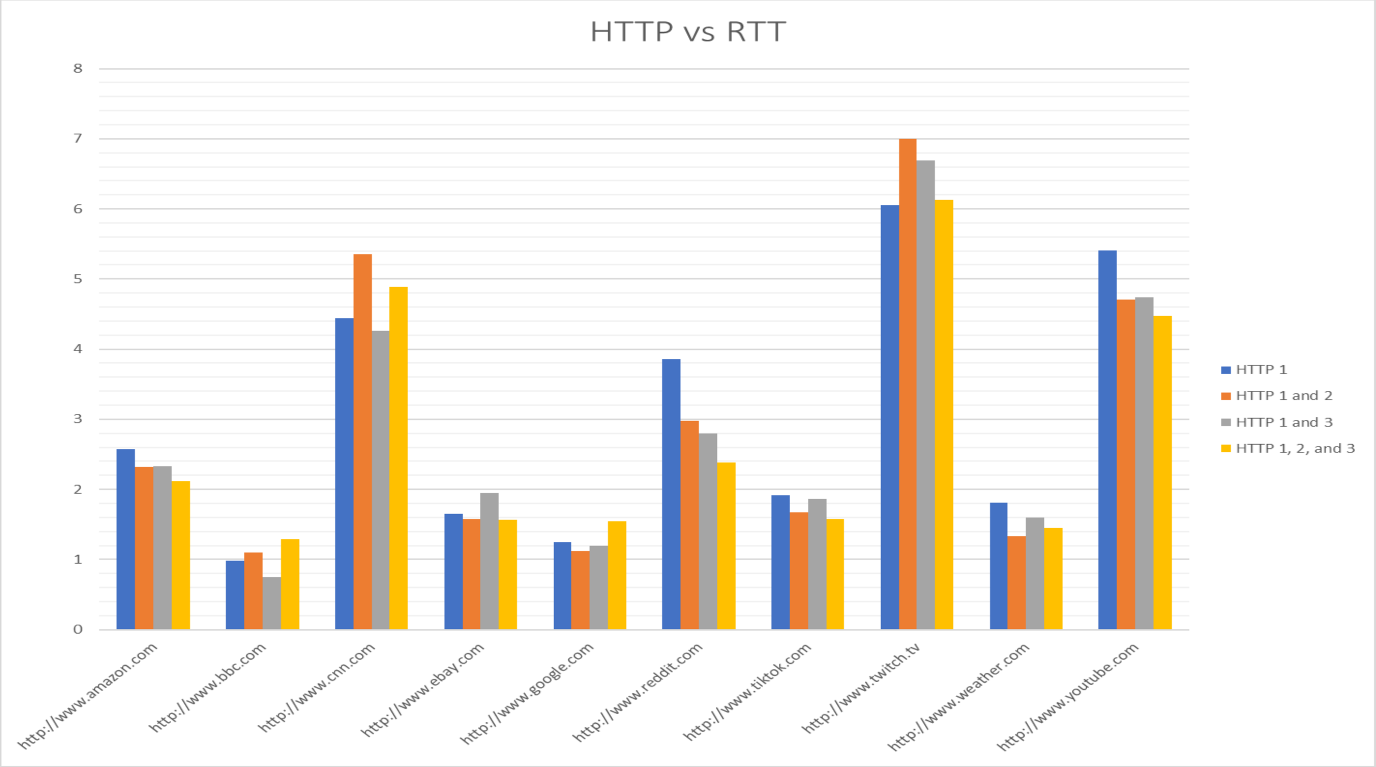

Fig 1. Bar graph indicating differences in HTTP 1, HTTP 1 and 3, HTTP 1 and 2, and HTTP 1, 2, and 3 respectively. This bar graph uses HTTP as its protocol.

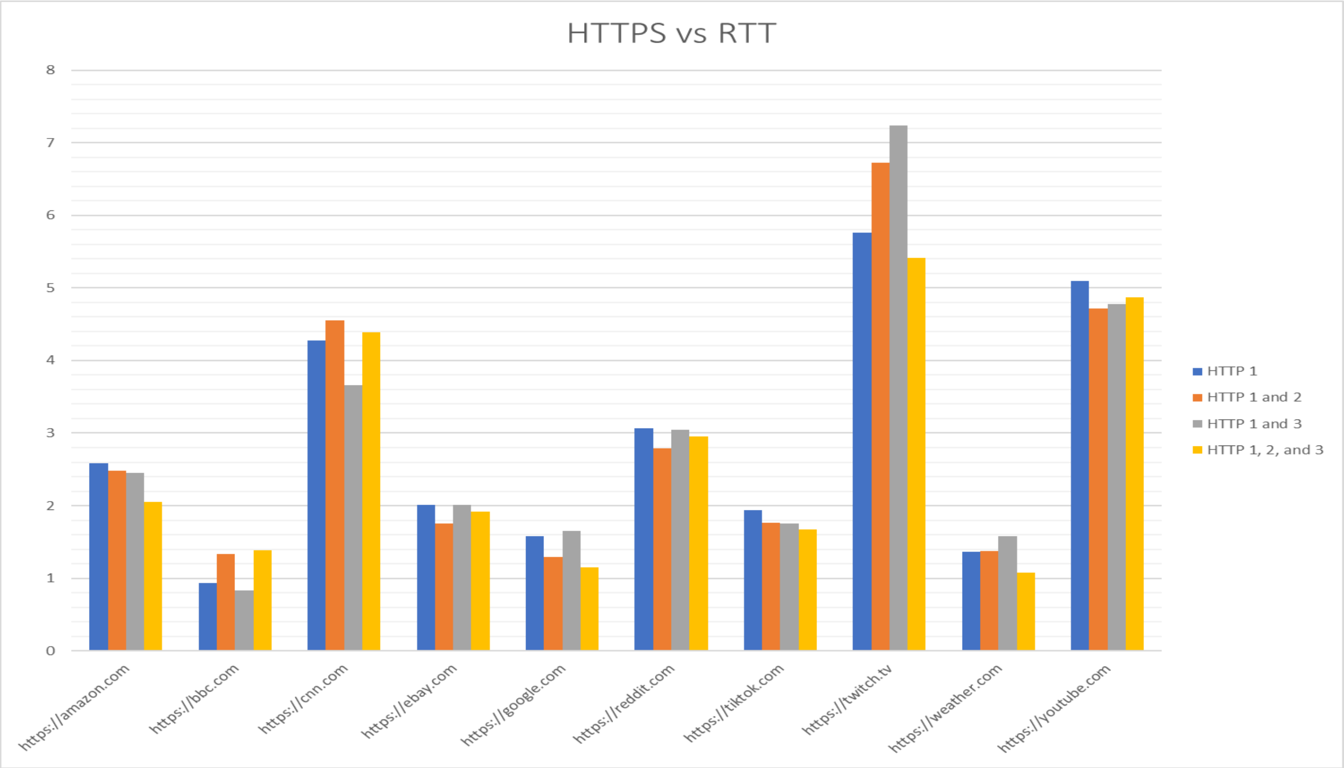

Fig 2. Bar graph indicating differences in HTTP 1, HTTP 1 and 3, HTTP 1 and 2, and HTTP 1, 2, and 3 respectively. This bar graph uses HTTPS as its protocol.

In order to differentiate between HTTP and HTTPS, the way the web page is called is utilized. The .get() function of selenium allows a specific url line to be retrieved and pasted into our address bar. In this location, the url either starts with http:// or https:// to retrieve in its respective way.

Using this, the program has four flags to symbolize each combination of HTTP protocol. These four combinations are HTTP 1 alone, HTTP 1 and 3, HTTP 1 and 2, and our modern way with no flags HTTP 1, 2, and 3.

IV. Evaluation

Our test consisted of 50 websites. A list of 450 has been obtained and can be tested in the future. Each website ran 10 times for each protocol combination for a total of 2000 runs. From the data, the average is retrieved and recorded. The 10 websites that were chosen were:

- amazon.com
- bbc.com
- cnn.com
- ebay.com
- google.com
- reddit.com
- tiktok.com
- weather.com
- youtube.com

Figures 1 and 2 show the averages of the runs for the chosen ten web pages in seconds. In these ten pages and HTTPS as the main method, cnn.com and bbc.com were the only ones that did worse when tested with a modern loading method. HTTP 1 was better for those sites. With HTTP, google.com and twitch.tv get added into this list. These sites have a lot of image objects in common.

For HTTP 3, all the websites have had a similar impact as HTTP 1. This is mostly because HTTP 3 is still rarely used in a majority of objects. One site that extensively uses it with success is youtube.com. On this site, HTTP 3 is used to load the document, web objects, and many of the previews that are given for videos. It is clearly superior in the HTTP version versus the HTTPS version. Overhead problems could be suspected as HTTP 3 has its own security methods in addition to the HTTPS's security.

When dealing with HTTPS, over half the websites have seen better performance using only HTTP 1 than when using HTTP 1 and 2. This is in accordance with the paper Mind they Delay: The Adverse Effects of Delay-Based TCP on HTTP. However, when the websites are loaded using HTTP instead of HTTPS, this is no longer the case.

For websites with large objects, HTTP 1 is clearly a winner across the board when compared to only HTTP 2. However, this only happens in the HTTPS case. When using HTTP, HTTP 1 falters and modern methods that introduce HTTP 2 and HTTP 3 surpass the page loading of HTTP 1. In the HTTPS case, it is more split. Many sites see considerable impact in page speed when using HTTP 3.Sites with large objects have worse performance.

V. Conclusion

On a case by case basis, this method of testing is useful in learning which HTTP protocol is best to send their data. Network operators can learn about their web performance better. Network innovators should take into consideration the impacts that it creates in other layers of the network. Otherwise the impact of their innovation may prove unsuccessful. This research shows that HTTP 3 has had a significant impact on increasing the page load performance for the top websites in the world..

References

[1] Agarwal, N., Varvello, M., Aucinas, A., Bustamante, F., & Netravali, R. (2020). Mind the Delay: The Adverse Effects of Delay-Based TCP on HTTP. In The 16th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '20), December 1–4, 2020, Barcelona, Spain

[2] https://brave.com/

[3] https://developers.google.com/web/tools/chrome-devtools/

[4] https://www.selenium.dev/projects/webdriver/

[5] https://www.fasterize.com/en/blog/web-performance-kpi-definition-and-deciphering-of-the-load-time/

[6] https://w3techs.com/technologies/details/ce-http3

[7] N.Cardwell, Y.Cheng,C. S.Gunn, S.H. Yeganeh, andV. Jacobson. Bbr: Congestion-based congestion control.Queue,14(5):20–53,2016.

[8] S.HaandI.Rhee.Taming the elephants:New tcp slow start.Computer Networks, 55(9):2092–2110,2011.

[9] M.Belshe,R.Peon,andM.Thomson. Hypertext Transfer ProtocolVersion2 (HTTP/2).RFC7540,RFCEditor,May2015.

[10] A.Baiocchi,A.P.Castellani,andF.Vacirca.Yeah-tcp:yet another highspeed tcp. In Proc.PFLDnet,volume7,pages37–42,2007.7.

[11] Rudman, R. and Bruwer, R. (2016), "Defining Web 3.0: opportunities and challenges", The Electronic Library, Vol. 34 No. 1, pp. 132-154. https://doi.org/10.1108/EL-08-2014-0140

[12] M.Bishop.HypertextTransferProtocolVersion3(HTTP/3). Internet-Draftdraftietf-quic-http-29,InternetEngineeringTaskForce,June2 020.WorkinProgress.