The Web, the Users, and the MOS: Influence of HTTP/2 on User Experience

Enrico Bocchi[1,3], Luca De Cicco[2], Marco Mellia[3], and Dario Rossi[4]

[1] Télécom ParisTech, Politecnico di Torino – enrico.bocchi@telecom-paristech.fr
[2] Politecnico di Bari – luca.decicco@poliba.it
[3] Politecnico di Torino – marco.mellia@polito.it
[4] Ecole Nationale Supérieure des Télécommunications – dario.rossi@enst.fr

Abstract. This work focuses on the evaluation of Web experience as perceived by actual users and in particular on the impact of HTTP/1 vs HTTP/2. We adopt an experimental methodology based on real web pages served through a realistic testbed where we control network, protocol, and application configuration. Users are asked to browse such pages and provide their subjective feedback, i.e., the Mean Opinion Score (MOS), while the testbed records objective metrics.
The collected dataset comprises over 4,000 MOS grades and we explore it to tackle the question whether HTTP/2 improves users experience, to what extent, and in which conditions. Findings show that users report marginal differences, with 22%, 52%, 26% of MOS being better, identical, or worse, respectively. Even in scenarios that favor HTTP/2 results are not as sharp as expected. This is in contrast with objective metrics used in the literature, which instead record a positive impact with HTTP/2 usage. This shows the complexity of understanding the web user experience, and the need to involve actual users in the quality assessment process.

Keywords: HTTP/2; Page Load Time; MOS; User Experience

1 Introduction

The Web keeps being at the center of our lives, thanks to a plethora of online services, from web searches to business applications, from personal communications to social networks and entertainment portals. HTTP is the de facto "thin waist" of the Internet [19], remaining almost unchanged from the original protocol defined at the end of the last century. Only recently a number of new protocols, namely HTTP/2 [3], SPDY [11] and QUIC [10], have been proposed and are likely to change the Web status quo. Having reliable ways to compare performance benefits becomes crucial when massive deployments of new protocols take place. However, measuring Web users' Quality of Experience (WebQoE) is a challenging problem. Page complexity has grown to include hundreds of objects hosted on different servers, with browsers opening tens of connections to fetch them. While several studies point out the importance of latency [16,18] and its relationship with business value[5], it is less obvious how it impacts WebQoE.

Objective metrics have been defined and the Page Load Time (PLT) is the de-facto benchmark used for comparison [8, 15, 21–23], with the industry adopting it too (e.g., Alexa reports the quantiles of PLT). However, this family of metrics does not fully reflect users' quality of experience in the complex "waterfall" of network and browser events taking place during the page loading processes.

Subjective metrics, the Mean Opinion Score (MOS), allow one to measure the actual user WebQoE, but it is extremely expensive to run MOS measurement campaigns. As such, approaches to estimate WebQoE have been proposed [6, 9], but their relationship with actual users' experience is yet to be proved and their computational complexity makes them difficult to use in practice.

Recognizing intrinsic limits of objective metrics [5], we present the first study of MOS measurement of WebQoE: We engineer a methodology to collect volunteers' feedbacks in a controlled environment where users are asked to access actual pages while we control network, protocol, and application setup. In our effort towards a subjective, yet scientific, comparison of HTTP/1.1 (H1) and HTTP/2 (H2), we (i) collect a dataset of over 4,000 samples of subjective feedback augmented with objective metrics, and (ii) dig into the data to shed light on actual experience improvement when using H2 vs H1. Advantages appear to be less sharp than those shown by objective metrics: Users report no differences in over half of the cases, while H2 improves WebQoE in 22% of cases only.

2 Related work

Since the original SPDY proposal [11], ended with the standardization in H2 [3], and the appearance of QUIC [10], researchers have been devoting increasing attention to the benchmarking and optimization of these protocols [4, 7, 8, 15, 17, 21–23]. In what follows, we contrast our investigation with related works considering experiment scale, testbed setup, set of pages, and collected metrics.

Experiments scale. In terms of experiments scale, works collecting objective metrics span from several thousands (active testbeds [7, 8, 17, 21, 22]) to several millions points (crawling [15] and server logs [23]). Conversely, studies employing actual user feedback (only [4] besides this paper) are inherently of smaller scale (i.e., tens of participants). Our work is based on the collection of actual user feedback from 147 participants, for a total of over 4,000 experiments.

Testbeds. Testbed setups are either based on proxies [7, 8] or, as in this work, on locally controlled servers and networks [17, 21, 22]. Few works leverage actual H2 servers in the Internet [15] or large corporate server logs [23]. Google Chrome is the most popular web browser followed by custom client implementations [21], or a mixture of clients [23]. As for network setup, both controlled [17, 21, 22] and uncontrolled [7, 8, 15] networks can be found, including 3G/4G access.

[5] `http://www.fastcompany.com/1825005/how-one-second-could-cost-amazon-16-billion-sales`

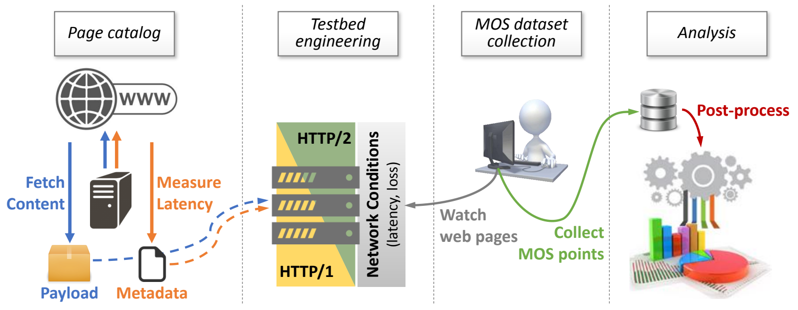


Fig. 1: Experimental workflow.

Page catalog. For what concerns pages used for testing, Alexa ranking is a popular source for the selection of websites. The number of sites ranges from 20 to 500, and page selection criterion (e.g., landing [7] vs non-landing [8]) differs. We use Alexa as well to drive our choice towards popular websites. As in [8], we select pages that are likely known by our users, i.e., pages popular in France. We consider pages optimized for desktop browsing and discard landing pages.

Measured metrics. Many works adopt the Page Load Time (PLT) as objective metric [7,8,15,21–23]. PLT limitations are well-known [6,9], yet only few works include more refined metrics to describe users' QoE, e.g., [5,17] consider the SpeedIndex [9]. MOS models for web traffic are dated back to 2002 and 2005 and therefore they should be re-assessed under recent architectures, technologies and designs. Involving end-users in subjective measurements is the best practice, with MOS being a simple and compact metric representative of their actual experience. MOS is the standard in audio and video quality comparison, but only recently it has been introduced for WebQoE assessment. To the best of our knowledge, only [4] presents a framework to collect volunteers' feedback on pre-recorded videos of web-browsing sessions: Side-to-side videos are shown, with the aim of identifying a winner. In contrast, we collect volunteers' feedback of actual browsing sessions, using the typical [1, 5] MOS scale [13]. Both approaches have challenges: e.g., synchronization between videos, correlation between videos and actual browsing experience, ability to slow-down/pause video can affect results in [4]. Conversely, in our work the analysis is made complex by volunteers tendency to refraining from using the full scale of scores, as we shall see.

3 Methodology

As portrayed in Fig. 1, the methodology we employ to compare H1 and H2 consists of four phases: *1. Page catalog* (Sec. 3.1) – To build a realistic benchmark, we fetch actual pages and characterize network paths towards servers. *2. Testbed engineering* (Sec. 3.2) – Pages and paths metadata are used to tune our testbed. Local servers host objects using multiple Apache instances while we

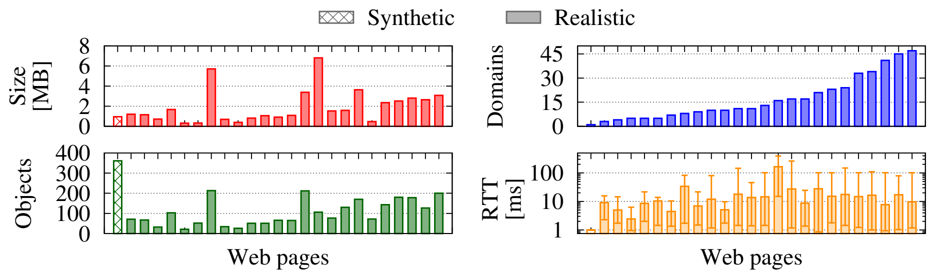


Fig. 2: Page catalog characteristics.

control network (RTT, loss), protocol (H1/H2), and application (domain sharding) configuration. *3. MOS collection* (Sec. 3.3) – Volunteers browse pages served by our local infrastructure and provide a score in the range $[1,5]$. At the same time, the testbed captures objective metrics. *4. Analysis* (Sec. 4–6) – At a later stage, we apply analytics to contrast H1 vs H2 performance.

3.1 Page catalog

For collecting MOS grades, we aim at selecting pages users are familiar with. As our tests take place in Paris, we start from the top 100 in Alexa France ranking. We visit each page using Google Chrome and compile a list of URLs of objects being requested by the browser. We then mirror each object on a local server and measure the RTT towards each domain using TCP-SYN packets.

We manually check each mirrored page from our local servers to both discard incomplete pages (e.g., object failing to download due to dynamic requests or cookies policies), landing pages [8] (e.g., Facebook login page), etc. We are left with 24 real pages covering a variety of categories, e.g., news, e-commerce, informative websites, leisure etc. At last, we add the toy page `http://www.httpvshttps.com` to the page catalog, for a total of 25 pages. For each considered page, Fig. 2 reports its size (top-left), the number of objects (bottom-left), the number of domains serving such objects (top-right), and the average per-domain RTT to contacted domains, with bars reporting the minimum and the maximum RTT (bottom-right). The figure shows that our catalog includes diverse scenarios, from pages hosted on few domains serving a handful of objects, to pages hosted on tens of domains and made of hundreds of objects.

3.2 Testbed engineering

Server and network configuration. We design and setup a local testbed where we have full control on network conditions (RTT, loss), protocols (H1/H2), and content placement (domain sharding [12]). Our testbed is composed of six servers, each equipped with a quad-core processor, 4 GB of memory and two Gigabit network cards. Servers run Ubuntu 14.04 with Apache HTTP Server

2.4.18. Apache runs in its default configuration, with H2 and SSL modules enabled. Content is served using SSL by installing self-signed certificates.

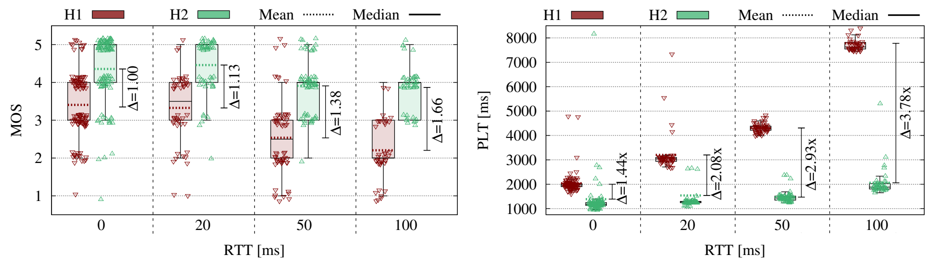
We run multiple Apache instances configured to serve content through virtual hosts, which are both name-based and IP-based. We leverage name-based configuration to distinguish requests directed to different domains being hosted on the same machine, while the IP-based distinction is required to have domains mapped to specific network conditions. To control network conditions, we use Linux traffic control utility (`tc`) to enforce both network latency and packet loss. We next distribute content to each server, preserving the original placement of objects into domains, and map each domain to a static IP address using the 10.0.0.0/8 private range. Two separate virtual-hosts serve content using either H1 or H2 to avoid protocol switching or fall-backs on the client side. The choice of H1/H2 is performed by the client, which directs requests to the IP address of the server implementing the desired protocol.

Client instrumentation. We provide a preconfigured PC to each volunteer taking part in our campaign. Each PC runs Linux Mint 17.3 and is equipped with a set of scripts for experiment orchestration. In particular, such scripts (i) setup the local client to reflect the desired scenario, (ii) run Google Chrome to let the volunteer visit a page, (iii) collect the user's score and the objective measurement, and (iv) send the results to a central repository.

Each experiment requires several steps to complete. From the users' point of view, the experience starts with a GUI listing all the available websites of the page catalog. Volunteers (i) select a page from the list and (ii) observe it being loaded by Google Chrome. At the end, they (iii) input the MOS grade, and then (iv) watch again the same page, now served with the other protocol. At step (ii) the page is loaded using either H1 or H2 in a random fashion, then at step (iv) the complementary protocol is used. Therefore, users sequentially grade the same page under the same conditions and for both protocols, although they are unaware about the protocol order.

From the implementation standpoint, once the volunteer has selected a page, the script (i) configures the system `/etc/hosts` file to direct browser requests to local servers instead of the public Internet.[6] Two `hosts` files are provided for each web page, one for H1 servers, the other for H2 servers. Next, the script (ii) starts Google Chrome in full screen mode, disabling the local cache and enabling the incognito mode. This ensures each page is loaded independently on previous tests and eventual cookies. We force Chrome to log network events, which we collect in the form of HTTP Archive (HAR) files for later stage analysis. Once the user has provided the (iii) MOS grade, (iv) all metadata for that experiment (i.e., HAR file, user's grade, and metadata with network configuration, etc.) are sent to a central repository.

[6] Due to the explicit binding between host names and IP addresses in `hosts` file, no DNS resolution takes place. This avoids any bias due to resolution delay and DNS caching, enabling a fair comparison between H1 and H2 performance.



(a) MOS grades vs RTT (b) PLT vs RTT

Fig. 3: MOS grades and PLT for homogeneous RTT in [0,100] ms.

3.3 Scenarios and MOS dataset collection

We aim at collecting MOS grades in (i) realistic scenarios to provide answers of operational interest, but also in (ii) controlled scenarios that the scientific community has already targeted via objective metrics. Given the limited time available with volunteers, we focus our attention on the following scenarios.

• **Homogeneous network.** Objects are distributed on servers as originally observed. RTT and packet loss are artificially forced to be the same for all virtual servers. RTT can be chosen in $\{0, 20, 50, 100\}$ ms, and packet loss in $\{0, 1, 2\}$%. Bandwidth is uncapped. These are the conditions considered in literature.

• **Heterogeneous network.** As before, but latency reflects the original RTT measured during the collection process. No loss is introduced. Such configuration introduces realism into the dependency graph of objects download, which may not arise in case of homogeneous conditions.

• **Unsharded deployment.** All objects are hosted by a single server, on a single domain name and IP address. RTT to the server is forced in $\{0, 20, 50, 100\}$ ms. Bandwidth is uncapped, and no loss is introduced. Unsharded deployment is useful to contrast today's production scenarios (i.e., sharding over multiple domains) vs situations that are by now unrealistic (i.e., all content hosted on a single "unsharded" domain) where H2 benefits are expected to appear [12].

Volunteers are exposed to experiments by randomly choosing one scenario and by visiting the same page over H1 and H2 in unknown order. To avoid biased ratings, only the website name is disclosed to users. Experiments have been conducted in three sessions totaling to 147 volunteers who sampled a space of 25 pages with 32 different scenarios. We followed the best practices suggested by "The Menlo Report" [2], and in particular those for network measurements [1].

4 MOS on the toy page

We start the analysis of collected MOS grades focusing on the toy page `https://www.httpvshttps.com`, which contains a HTML document (18 kB) and 360 identical non-cacheable images of 20x20 pixels (1.7 kB each), for a total of 630 kB.

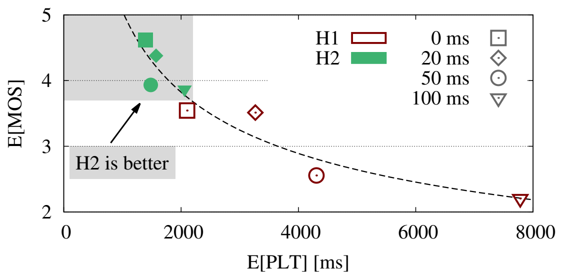


Fig. 4: Sub-linear dependency between PLT increase and MOS reduction.

All the content is hosted on a single domain, mapped to a single IP address and served by a single machine. This scenario is particularly adverse to H1 since the browser opens a large number of connections, each incurring in TCP and TLS handshakes overhead and in TCP slow-start. In contrast, H2 takes full advantage of its capabilities by pipelining all requests over multiple streams encapsulated in a single TCP connection and by performing HPACK header compression. We expect H2 to reduce the PLT, ultimately providing a better WebQoE.

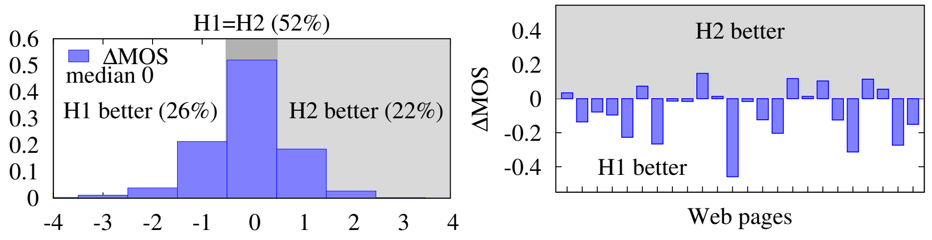
We use this toy page to validate the testbed and calibrate MOS grades. On the one hand, we aim at verifying whether expectations on H2 performance are satisfied. On the other hand, we aim at assessing the MOS gap between H2 and H1 by using this extreme scenario as a litmus paper [16, 18, 20]. We consider 4 different network setups, namely with RTT in $\{0, 20, 50, 100\}$ ms, collecting 487 MOS samples in total. Fig. 3 shows MOS (left plot) and PLT (right plot) for H1 (red) and H2 (green). Each point corresponds to an experiment, adding jitter (to x-y axis for MOS and to x axis only for PLT) to enhance the representation.

Consider Fig. 3a first and notice that MOS consistently decreases with an increased RTT. This holds for H1 and H2, with H2 providing a better experience at both low and high latencies. Also, the difference (Δ) between the average of H1 and H2 MOS grades is always of at least 1 point, increasing along with RTT.

Consider now Fig. 3b, showing PLT. H1 is significantly more penalized than H2, with PLT peaking at 8 s for RTT= 100 ms, while H2 keeps PLT below 2 s in all scenarios. As expected, H2 outperforms H1 PLT, meeting the original design goal of “a 50% reduction in page load time” [11].

Next, we verify the existence of a sub-linear dependency of the subjective response to an objective impulse [20]. Here the impulse is the inflated RTT (translating into a longer PLT), while the response is the MOS degradation. Fig. 4 reinterprets Fig. 3 as a scatter plot, where each point is the $(\mathbb{E}[\text{PLT}], \mathbb{E}[\text{MOS}])$ pair over all samples for a given RTT. The figure also reports the trend curve, clearly highlighting the expected sub-linear dependency.

A final remark is that excellent MOS (i.e., 4 or higher) is bestow only to pages with a loading time lower than 1.25 s. This is in line with [14], which classifies pages as *reactive* if they render the above-the-fold content in less then 1 s.



(a) Empirical probability mass function (b) Per-page MOS difference

Fig. 5: H1 vs H2 MOS grades (ΔMOS) for all 4,000 tests in the dataset.

5 MOS on real pages

We here focus on real pages to compare H1 and H2 from a subjective perspective. Our aim is to assess if and to what extent differences in user experience shown earlier still hold in real-life scenarios. To do so, we consider a set of pages served using multiple network and application configurations. On the one hand, we revisit known results from the unique user MOS perspective. On the other hand, we target the impact of less studied factors, such as implications of content sharding [12] over multiple domains and differences caused by homogeneous vs heterogeneous latency conditions towards servers.

5.1 Subjective MOS differences

We start by assessing the per-user difference of H1 vs H2 MOS grades (ΔMOS) for each page in the catalog. Fig. 5 shows ΔMOS over all tests, detailing both the empirical probability mass function (Fig. 5a) and the per-page MOS difference (Fig. 5b). The figure is annotated with statistics (e.g., median) and visual references (e.g., light-gray area for H2 better than H1).

Some insightful observations can be drawn from the plot: The distribution is (i) monomodal with zero mean and median, (ii) bell shaped, but (iii) slightly skewed. In other words, (i) in 50% of cases, users equally score H2 and H1, (ii) cases where either H2 or H1 has higher score are roughly balanced, although (iii) there is a slight yet noticeable bias towards negative ΔMOS, where $\text{MOS}|_{\text{H1}}$ is higher than $\text{MOS}|_{\text{H2}}$. That is, contrary to the previous results, now the difference between H2 and H1 is much less appreciable and consistent.

This reinforces the need to perform experiments on real-world pages, as opposite to artificial pages that inflate MOS differences. Results are only partially surprising. First, pages widely differ (see Fig. 2) and ΔMOS varies according to the page being considered, as shown by Fig. 5b (the order of web pages is consistent Fig. 2). Second, users have a different way to “value improvement”, causing them to report the same score under both protocols, which contributes to $\Delta\text{MOS} = 0$. Third, pages in our catalog are likely optimized for H1. Fourth, the H1 software has undergone decades of testing and optimization, while H2 is a relatively new protocol.

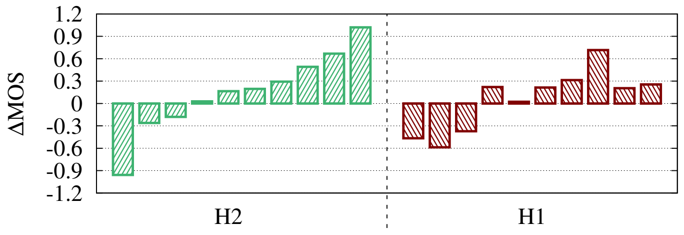


Fig. 6: ΔMOS for 10 pages where we contrast unsharded vs sharded versions.

5.2 Impact of page sharding

We now consider sharding [12], i.e., distributing page contents over multiple domains to exploit server parallelism. This practice helps in overcoming the limitation on the maximum number of connections a browser can establish towards the same domain. Given H2 benefits of using a single connection to a single domain [7, 15, 22], one would expect that unsharding helps in taking advantage of H2 pipelining features. In our evaluation, we consider 10 of the 25 pages of the catalog and modify them so to have all the content hosted on a single domain (i.e., unsharding the content). We then contrast MOS grades to assess the impact of (un)sharding for H2 and H1 independently.

Fig. 6 shows the per-page difference between the average MOS for the unsharded and for the sharded content. In formulas, $\Delta\text{MOS} = \text{E}[\text{MOS}|_{\text{unsharded}}]$ - $\text{E}[\text{MOS}|_{\text{sharded}}]$. Pages are sorted increasingly according to ΔMOS for H2.

It is straightforward to notice that the impact of sharding is page-dependent: there are pages for which the user experience improves when they are served through the unsharded deployment ($\Delta\text{MOS} > 0$), as well as pages suffering from usharding ($\Delta\text{MOS} < 0$). 7 pages out of 10 show an improvement in MOS when unsharded, even though the difference in perceived quality greatly changes, from a minimum of 0.028 to a maximum of 1.020 ΔMOS points. H2 appears to benefit more of unsharding, but 3 pages gets a sensibly reduced MOS. H1 is less impacted, peaking at a difference of “only” 0.716 ΔMOS points.

5.3 Impact of latency diversity

Latency is known to be impacted by client-server RTT. Here we investigate how much impact it has on MOS. We contrast scenarios with homogeneous RTT (usually considered in literature [17, 21]) against heterogeneous RTT to each domain (fine-grained control). Clearly, homogeneous conditions are the ordinary solution in case of proxy-based proposals [7, 8, 22] and is typically justified in testbed studies with the assumption that sharding practice will ultimately be abandoned as countering H2 benefits. At the same time, sharding is by now a standard practice and page redesign would happen conditionally on unsharding

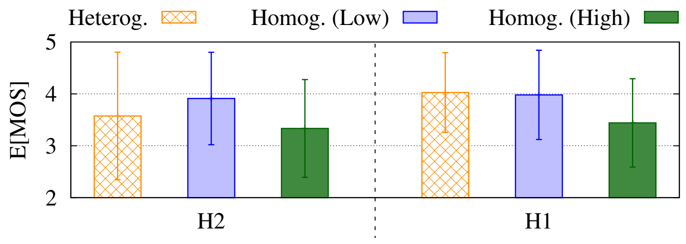


Fig. 7: $\mathbb{E}$[MOS] for heterogeneous vs homogeneous RTT settings.

benefits being proved and consistent. Total unsharding is unlikely as pages aggregate many contributions (e.g., twitter feeds, advertisement, etc.) coming from multiple content producers. As such, it is important to evaluate H2 performance also in controlled conditions that are as close as possible to the real ones.

For this experiment, we select a subset of 3 pages sampled by 95 users for a total of 362 experiments. Average MOS scores are reported in Fig. 7 for different RTT configurations and for H2 and H1 separately. It emerges that different RTT leads to opposite biases for H1 vs H2. For instance, in the case of H2, low-RTT homogeneous scenarios provide about 0.35 better MOS than heterogeneous RTT. When RTT is high (> 50 ms), instead, MOS degrades loosing 0.58 points with respect to the low-RTT scenario. This happens in the case of H1 too, where high-RTT homogeneous scenarios lead to a loss of about 0.5 MOS points with respect to both heterogeneous and low-RTT homogeneous scenarios. Interestingly, H1 in heterogeneous RTT conditions performs much better than H2 in the same scenario. Similarly to [23], we noticed that macroscopic pages characteristics are not telling as for user MOS. The performance gap has its roots in page dependency graph [22], and homogeneous latencies may hide intricate interactions in such dependencies that arise only under heterogeneous conditions.

6 Objective Metrics on real pages

We finally study the H1 vs H2 difference using objective metrics (OBJ in short). As before, we quantify the difference in accessing the same page over the two protocols with $\Delta\text{OBJ} = \text{OBJ}_{\text{H2}} - \text{OBJ}_{\text{H1}}$, where OBJ is the Time to the First Byte (TTFB), the Document Object Model (DOM), or Page Load Time (PLT). We additionally consider the ObjectIndex, a replacement metric for the SpeedIndex [9] that does not face the same computational complexity and has been shown to be strongly correlated with the latter [5].

Fig. 8 presents the collected results depicting the histogram of ΔOBJ using bins of 100 ms. The figure is annotated with statistics (notice that H2 better than H1 is represented by the negative semi-plane in this case). All OBJ exhibit an empirical probability mass function that is similar to that of the ΔMOS grades

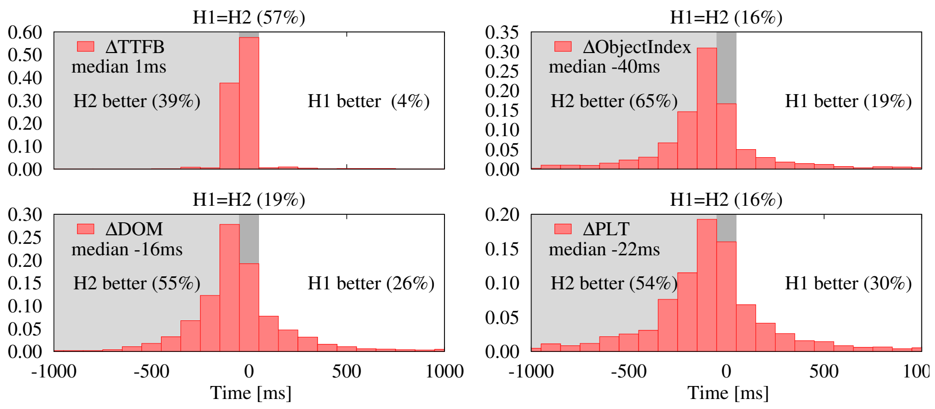


Fig. 8: Empirical probability mass function of ΔOBJ, various objective metrics.

(i.e., roughly symmetric, peak close to zero, very low median). In addition, here ΔOBJ attributes a (slight) advantage to H2, unlike in the ΔMOS case.

Excluding the TTFB, which is known to be not the most appropriate metric for web pages performance assessment, H2 shows better results than H1 in at least 54% of tests. That is, H2 speeds up the page loading process and the time needed to load the DOM, but those improvements are not reflected in user experience that rates H1 and H2 with the same score in 55% of cases (see Fig. 5).

7 Conclusions

This paper presents the first study comparing the performance of H2 and H1 in terms of MOS. We contrast the two protocols using both subjective (i.e., a MOS corpus of over 4,000 points) and objective metrics using a dedicated testbed.

The picture emerging from collected results does not allow the election of a single winner. While H2 sensibly reduces the PLT on a toy page, ultimately improving the quality of experience, it is not as effective when serving real-world web pages. Objective metrics (e.g., DOM, PLT, etc.) show a performance improvement to the advantage of H2 in more than 50% of cases, but they fail to predict users' MOS that is reported to be higher in the case of H1.

This highlights the importance of users feedbacks and calls for future research on new models enhancing the correlation between MOS and QoE metrics.

Acknowledgments

This work has been carried out at LINCS (`http://www.lincs.fr`) and benefited from support of NewNet@Paris, Cisco's Chair "Networks for the Future" at Telecom ParisTech (`http://newnet.telecom-paristech.fr`).

References

1. M. Alllman and V. Paxson. Issues and etiquette concerning use of shared measurement data. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, IMC '07, pages 135–140, New York, NY, USA, 2007. ACM.
2. M. Bailey, D. Dittrich, E. Kenneally, and D. Maughan. The menlo report. *IEEE Security & Privacy*, 10(2):71–75, 2012.
3. M. Belshe, R. Peon, and M. Thomson. Hypertext Transfer Protocol version 2 (HTTP/2). In *IETF RFC7540*, 2015.
4. B. Blackburn, M. Varvello, K. Schomp, D. Naylor, A. Finamore, and K. Papagiannaki. Is the Web HTTP/2 yet? In *TMA PhD School*, 2016.
5. E. Bocchi, L. De Cicco, and D. Rossi. Measuring the Quality of Experience of Web users. In *Proc. ACM SIGCOMM Workshop on QoE-based Analysis and Management of Data Communication Networks (Internet-QoE 2016)*, 2016.
6. J. Brutlag, Z. Abrams, and P. Meenan. Above the fold time: Measuring web page performance visually. `http://conferences.oreilly.com/velocity/velocity-mar2011/public/schedule/detail/18692`.
7. M. Butkiewicz, D. Wang, Z. Wu, H. V. Madhyastha, and V. Sekar. Klotski: Reprioritizing Web Content to Improve User Experience on Mobile Devices. In *Proc. USENIX NSDI*, pages 439–453, 2015.
8. J. Erman, V. Gopalakrishnan, R. Jana, and K. K. Ramakrishnan. Towards a SPDY'Ier Mobile Web? In *Proc. ACM CoNEXT*, pages 303–314, 2013.
9. Google Inc. `https://sites.google.com/a/webpagetest.org/docs/using-webpagetest/metrics/speed-index`.
10. Google Inc. QUIC, a multiplexed stream transport over UDP. `https://www.chromium.org/quic`.
11. Google Inc. SPDY, an experimental protocol for a faster web. `https://www.chromium.org/spdy/spdy-whitepaper`.
12. I. Grigorik. Http/2 is here, let's optimize! `http://bit.ly/http2-opt`.
13. International Telecommunication Union. Subjective testing methodology for web browsing. ITU-T Recommendation P.1501, 2014.
14. P. Irish. Delivering the goods in under 1000ms. `http://bit.ly/1toUUA7`.
15. M. Varvello, K. Schomp, D. Naylor, J. Blackburn, A. Finamore, and K. Papagiannaki. Is The Web HTTP/2 Yet? In *Proc. PAM*, 2016.
16. R. B. Miller. Response time in man-computer conversational transactions. In *Proc. AFIPS Fall Joint Computer Conference*, pages 267–277, 1968.
17. R. Netravali, A. Sivaraman, S. Das, A. Goyal, K. Winstein, J. Mickens, and H. Balakrishnan. Mahimahi: Accurate Record-and-Replay for HTTP. In *Proc. USENIX ATC*, pages 417–429, 2015.
18. J. Nielsen. Response times: The 3 important limits. `https://www.nngroup.com/articles/response-times-3-important-limits/`, 1993.
19. L. Popa, A. Ghodsi, and I. Stoica. HTTP as the Narrow Waist of the Future Internet. In *9th ACM SIGCOMM Workshop on Hot Topics in Networks*, 2010.
20. P. Reichl, S. Egger, R. Schatz, and A. D'Alconzo. The Logarithmic Nature of QoE and the Role of the Weber-Fechner Law in QoE Assessment. In *IEEE ICC*, 2010.
21. X. S. Wang, A. Balasubramanian, A. Krishnamurthy, and D. Wetherall. How Speedy is SPDY? In *Proc. USENIX NSDI*, pages 387–399, 2014.
22. X. S. Wang, A. Krishnamurthy, and D. Wetherall. Speeding up Web Page Loads with Shandian. In *Proc. USENIX NSDI*, pages 109–122, 2016.
23. K. Zarifis, M. Holland, M. Jain, E. Katz-Bassett, and R. Govindan. Modeling HTTP/2 Speed from HTTP/1 Traces. In *Proc. PAM*, 2016.