# HTTP/2 AND THE USERS: THE GOOD, THE BAD AND THE UGLY

**–THE DIRECTORS' CUT–**

ENRICO BOCCHI[*] [‡] LUCA DE CICCO[†]
MARCO MELLIA[‡] DARIO ROSSI[*]

*Telecom ParisTech*[*] *Politecnico di Bari*[†] *Politecnico di Torino*[‡]

{name.surname}@telecom-paristech.fr[*]
l.decicco@poliba.it[†]
{name.surname}@polito.it[‡]

Abstract. HTTP/2 is reality: services adopt it and researchers measure and optimize performance – yet, the actual impact on users' web browsing experience is unclear. This work focuses on the comparison of HTTP/1.1 and HTTP/2 as perceived by the user. We adopt an experimental methodology, where popular web pages are served through a testbed that allows us to control network, protocol and application configuration. We ask users to browse actual web pages and provide their subjective feedback, i.e., the Mean Opinion Score (MOS), while we record objective metrics.

We collect a dataset, that will be made available to the community, accounting for several thousands MOS grades, and leverage it to tackle the question whether HTTP/2 is better than HTTP/1.1 as far as user experience is concerned. We find that, despite differences can be seen from objective metrics, (i) users report much smaller differences. Additionally, we show how (ii) MOS is poorly correlated with any objective metric, to the point that (iii) accurately predicting MOS from objective metrics is a real challenge. At last, (iv) when trying to correlate how different factors (e.g., network configuration, page characteristics, etc.) contribute to the quality of experience, we observe much less clear results than when observing objective metrics.

## 1. Introduction

The Web keeps being at the center of our lives, thanks to a plethora of online services, from web search to business applications, from personal communication to social networks and entertainment portals. HTTP became in the last few years the new de facto "thin waist" of the Internet [34, 37], with security features offered by HTTPS that are also gaining momentum [30, 31]. Defined at the end of last century, HTTP version 1 protocol family has slowly evolved, while only recently a number of new protocols – namely HTTP/2 [6], SPDY [21] and QUIC [20] – have been proposed and are likely to change the Web status quo. Clearly, having reliable ways to compare performance becomes crucial when massive deployments of new protocols take place [28]. However, measuring and understanding Web

| Ref. | Experiments scale | Testbed setup | | Page catalog | Metrics |
|---|---|---|---|---|---|
| | | Server | Client (Network) | | |
| [12] | 10,000 | Squid-proxy + SPDYChrome-proxy | Chrome (3G) | 20 popular, no landing | PLT |
| [39] | n.a. (moderate scale) | Apache, Mono-server, unsharded | Own SPDY client (controlled LAN) | Synthetic + Alexa-200 | PLT |
| [10] | 80,000 (500 every hour during 1 week) | Klotski NodeJS proxy on AmazonEC2 | Chrome for Android (4G) | 50 Landing pages from Alexa-200 | Per-user utility |
| [32] | 55,000 (110 configurations per 500 sites) | Apache, multi-server | Chrome HAR capturer (controlled LAN) | Alexa-500 | PLT, SpeedIndex |
| [11] | n.a. (moderate scale) | H2O | Chromium, no TLS (controlled LAN) | Alexa-20 | PLT |
| [40] | n.a. (moderate scale) | Shandian (V8/Blink based) | Chrome vs Shandian (controlled LAN) | Alex-100 (Mobile version) | PLT |
| [28] | Large scale (crawling, months of experiments) | Internet server (nginx, LiteSpeed) | Chrome HAR capturer (fiber, 3G/4G) | 8,492 H2 Websites (200 for 3G/4G) | PLT |
| [41] | 3.4M pages from 40,000 hosts | Akamai logs | Mix of user access (unreported) | Mix of user browser (unreported) | PLT |
| [7] | 43 participants, n.a. experiments | Web-based crowdsourcing , Reply of video captures of H2 vs H1 | | | PLT, User feedback |

TABLE 1. Related work at a glance

users' Quality of Experience (WebQoE) is a challenging problem. Page complexity has indeed grown to include hundreds of objects hosted on different servers, with browsers opening tens of connections to fetch them and executing javascript code (or related technologies) as part of the page construction process. While several studies pointed out the importance of latency [29, 33] and its direct relationship with the value of business [3, 36], it is far less obvious how it impacts WebQoE.

*Objective* measurements (e.g., the time at which the first byte is received, the time at which the first object is rendered on screen, the completion of the full page, etc.) can be considered, but they do not fully reflect user's quality of experience in the complex "waterfall" [23] of network and browser events taking place during the page download and rendering processes. Among objective metrics, the Page Load Time (PLT) [38] is the de-facto benchmark metric used for comparison [11, 12, 28, 32, 35, 39–41], with the industry adopting it too [1, 13, 14]: Alexa [1] reports the quantiles of the PLT, and Google uses PLT to rank search results [13, 15].

*Subjective* metrics, such as the Mean Opinion Score (MOS), allow one to completely measure the WebQoE. However, it is extremely expensive to run large MOS measurement campaigns. As such, automatic approaches have been proposed [9, 16] to estimate WebQoE. Unfortunately, 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 [8], we engineer a methodology to collect volunteers' feedback in a controlled environment, where users are asked to access actual pages while we control network, protocol and application setup. The testbed engineering, test administration to volunteers, data collection and processing pose challenges per-se, given the complexity of pages and services, the variety of parameters, and the diversity of human interaction means.

The rest of this report first surveys related work (Sec. 2), then describes our methodology (Sec. 3). In addition, Appendix A provides a thorough characterization of selected web pages, and Appendix B shows all considered metrics (both objective and subjective), as well as relevant web pages features.

## 2. Related work

Since the original SPDY proposal [21], ended with the standardization in HTTP/2 [6], and the appearance of QUIC [20], researchers have been devoting increasing attention to the benchmarking and optimization of these protocols [7,10–12,28,32,39–41]. Tab. 1 surveys related work by describing the experiment scale, the testbed setup (servers, clients and network configuration), the set of pages considered (i.e., the page catalog), and the collected metrics.

**Experiments scale.** In terms of scale, works collecting objective metrics span from several thousands (active testbeds [10–12,32,39,40]) to several millions points (crawling [28] and server logs [41]). Conversely, studies employing actual user feedback (to the best of our knowledge, only [7]) are inherently of smaller scale.

**Testbeds.** Testbed setups are either based on proxies [10, 12] or, as in this work, on locally controlled servers and networks [11,32,39,40]. Few works leverage actual H2 servers in the Internet [28] or large corporate server logs [41]. Google Chrome is the most popular web browser, although its version, the OS/device on which it runs (e.g., android mobile [10]), or configuration (e.g., TLS disabled [11]) are not consistent across setups. It is also possible to find custom client implementations [39], or mixture of clients [41]. As for network setup, both controlled [11, 32, 39, 40] and uncontrolled [10, 12, 28] Internet access can be found, including 3G/4G networks.

**Page catalog.** In terms of pages used for testing, we observe a mixture of approaches: 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 [10] vs non-landing [12]) differs. We also use Alexa to bias our choice towards popular websites. As in [12] we select pages that are popular with our users, i.e., that are popular in France, and we do not consider landing pages. [28] points out another source of uncertainty, namely the fact that mobile pages may greatly differ from desktop versions. Additionally, in the mobile space it is likely that other considerations, e.g., pages design [17], will play an equally important (if not greater) role with respect to the protocol pages are served with. As such, we limitedly consider MOS collection for pages in their desktop version.

**Measured metrics.** Most work adopt the Page Load Time (PLT) [38] as single objective performance metric [10–12, 28, 32, 35, 39–41]. PLT limitations are well established [9, 16], yet only few works include more refined metrics to describe user QoE – such as [8, 32] that consider the SpeedIndex [16]. However, SpeedIndex requires to capture and post-process the video of the page rendering process, incurring in computationally prohibitive costs. More specifically, while the post-processing could be done off-line, the video recording itself slows down the very same rendering process, altering the experiment results [8]. As such, the SpeedIndex is not widely used (to the best of our knowledge, a systematic computation of the SpeedIndex is done only in [8, 32]) and its actual relationship with MOS has never been fully elucidated. Finally, we point out that whereas MOS models for web

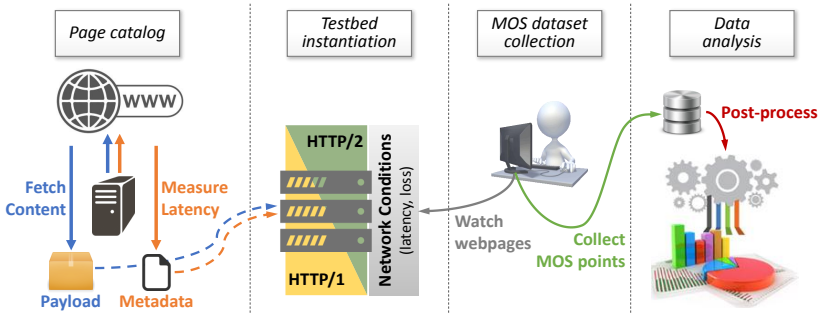


FIGURE 1. Experimental workflow

traffic do exist [26, 27], their relevance should be re-assessed under recent architectures, technologies and designs (so that we consider them out of scope of this work).

Involving end-users in subjective measurements is the best practice to assess actual experience. MOS is the standard in audio and video quality comparison [24], but only recently it has been introduced for WebQoE assessment [25]. To the best of our knowledge, only [7] 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 between the experiments. In contrast, we collect volunteers feedback of *actual browsing sessions*, using the typical 1-5 MOS scale [25].

Both approaches have challenges: e.g., synchronization between the videos, correlation between video viewing and browsing experience, ability to slow down/pause video can affect results in [7]. Conversely, the analysis of data gathered from actual browsing sessions might include sources of bias caused by class imbalance affecting the collected dataset, as well as volunteers refraining from using the full scale of scores. Overall, given the wide diversity of approaches, it is not surprising to find unexpected or contradicting results, as [28] points out.

## 3. METHODOLOGY

As portrayed in Fig. 1, the methodology we employ to compare H1 and H2 consists of four main phases:

**(i) Page catalog (Sec. 3.1).** To build a realistic test, we fetch a number of actual pages and characterize the paths towards servers.

**(ii) Testbed instantiation (Sec. 3.2).** Pages and paths metadata are shipped to servers in our testbed. We locally host all objects using multiple Apache HTTP servers, controlling network (e.g., RTT, loss), protocol (e.g., H2/H1), and application (e.g., domain sharding) configurations at a fine-grain.

**(iii) MOS collection (Sec. 3.5).** Volunteers browse pages served by our local infrastructure and provide a score in the range $[1, 5]$. At the same time, we collect objective metrics.

**(iv) Analysis.** At a later stage, we apply statistical analysis and machine learning techniques to the MOS dataset to contrast H2 vs H1 performance.

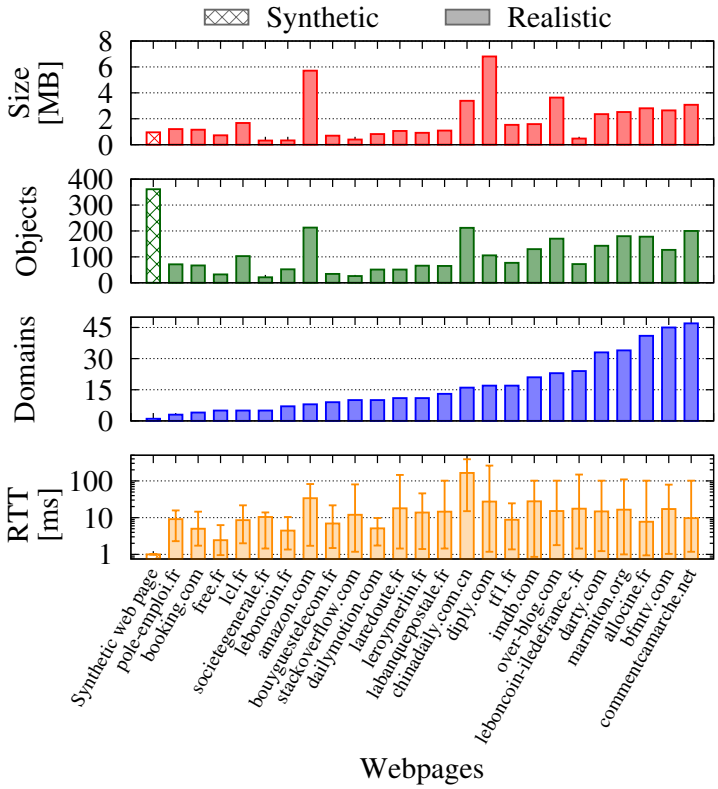


FIGURE 2. Page catalog characteristics.

3.1. **Page catalog.** We first need to select representative pages to offer users during the MOS tests. As our tests take place in Paris, we start from the top 100 most popular pages in France according to Alexa ranking. We then visit each page using the Google Chrome browser, and compile a list of all the URLs of objects being requested by the browser. We then download and mirror each object on a local server. At the same time of URLs list compilation, we measure the Round Trip Time (RTT) towards each domain providing content.[1] Measurements are performed from our campus network in Paris and are thus representative for a not-congested connection toward contents likely hosted in close-by CDNs.

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 [12] (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. Our page catalog ensures a statistically relevant sampling of popular pages, and it is compatible with the time budget allocated for the experiments and the number of volunteers involved (further information is provided in Sec. 3.5). For each considered page, Fig. 2 reports its size (top), the number of objects (2nd from top), the number of domains serving such

[1]We use TCP SYN packets to contact 10 times each domain and compute the average RTT.

objects (3rd from top), and the average per-domain RTT with respect to contacted domains, with bars reporting the minimum and the maximum RTT (bottom). The figure shows that the benchmark includes very 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. A detailed analysis of the page catalog is reported in the Appendix A.

3.2. **Testbed engineering.** This section describes (i) the testbed we engineered and (ii) the collection of MOS grades from volunteers. As for (i), albeit our testbed stems from an independent effort, it is similar to the Mahimahi approach [32]. As for (ii), to collect MOS grades we adhere to the standard practice described in [25].

3.3. **Server and network configuration.** We design and setup a local testbed where we have full control on both protocols (H1, H2), content placement (sharding), and network conditions (RTT, packet loss). Our testbed is composed of six servers, each equipped with a quad-core Intel processor, 4 GB of memory and two Gigabit network cards connected to the lab LAN. Servers are based on Ubuntu Linux 14.04 with Apache HTTP Server 2.4.18 (note that 2.4.17 is the first release supporting H2 [4]). Apache runs in its default configuration, with H2 and SSL enabled. Content is uniquely served using SSL by installing self-signed certificates.

We run multiple Apache instances as in [32], 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, as defined below. We next distribute and upload content into 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, assigning the addresses to virtual interfaces.

We configure two separate virtual-hosts to serve content using alternatively H1 or H2 so to avoid protocol switching or fall-backs on the client side. The selection of H1/H2 is performed by the client, which directs requests to the IP address of the server implementing the desired protocol. We force all content to be served over TLS so to compare H1 and H2 performance under the same conditions, as browsers do not support H2 over unencrypted connections.

To control network conditions, we use Linux traffic control (`tc`) to set both network latency and packet loss (we do not consider bandwidth limitations in this work). We configure the network to either enforce controlled but artificial network conditions, or, for the first time to the best of our knowledge, to truthfully replicate original network conditions (e.g, to replicate path delays measured toward each different domain).

3.4. **Client instrumentation.** We make available one PC to each volunteer taking part in our campaign. Each PC runs Linux Mint 17.3, with a set of scripts implementing experiment orchestration. In particular such scripts (i) setup the local client to reflect the desired scenario, (ii) run Google Chrome v.47.0 to let the volunteer automatically visit a page, (iii) collect the user's score and the objective measurement, (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, (ii) observe Google Chrome that

loads the page. At the end, (iii) input the MOS grade, and then (iv) watch again the same page, now served with another protocol. Specifically, the page is loaded using either H1 or H2 in a random fashion at step (ii), and then using H2 or H1 at step (iv). Therefore, users sequentially grade the same page under the same conditions and for both protocols, although they are unaware of protocol order.

Considering now the implementation point of view, once the volunteer has selected a page, the script (i) configure the system `/etc/hosts` file to direct browser requests to the IP addresses of local servers instead of the public Internet.[2] Two `hosts` files are provided for each web page to visit, one pointing to IP addresses of H1 servers, the other to H2 servers. Next, the script (ii) starts Google Chrome in full screen mode, disabling the local cache and enabling *in-incognito* mode. This ensures each page is loaded independently on past tests and on eventual cookies. We force Chrome to accept self-signed SSL certificates, and to enable network events logging [18]. The latter are then collected through Chrome remote debugging protocol [19], and dumped into a HTTP Archive (HAR) file for later stage analysis. Once the user has provided the (iii) MOS grade, (iv) all metadata for that experiment (i.e., HAR log, user's grade, and metadata file with network configuration, timestamp, etc.) are sent to a central repository. All features considered meaningful for results analysis are reported in Appendix B.

3.5. **Scenarios and MOS Dataset Collection.** We aim at collecting MOS grades in (i) scenarios that are as realistic as possible to provide answers of operational interest, but also in (ii) controlled scenarios that the scientific community has already analyzed via objective metrics, to directly contrast these findings.

Given the limited time available with volunteers, the breadth of scenarios that can be explored trades off with the need of collecting a statistically relevant bag of MOS points (from multiple users, for any page in the catalog, for protocols and network settings). Thus, we focus our attention on the following relevant scenarios:

- **Homogeneous network.** Objects are distributed on servers as originally observed, often with sharded domain names. However, 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 i.i.d. packet loss rates in $\{0, 1, 2\}\%$. Bandwidth is uncapped.
- **Heterogeneous network.** Objects are distributed on servers as originally observed, often with sharded domain names. Latency to servers reflects the original RTT measured during the collection process, i.e., on the public Internet. Bandwidth is uncapped, and no loss is introduced.
- **Unsharded deployment.** All objects are hosted by a single server, on a single domain name and IP. RTT to server can be chosen in $\{0, 20, 50, 100\}$ ms. Bandwidth is uncapped, and no loss is introduced.

Homogeneous network conditions described above are tuned as those generally considered in the literature. Heterogeneous conditions, instead, introduce realism into the dependency graphs of object download, that may not arise in case of homogeneous conditions. Unsharded deployment is useful to contrast today's production scenarios (i.e., sharding over multiple domains to circumvent the limitation in the

[2]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.

number of TCP connections that browsers can open to any given domain) where H2 is expected to underperform, vs situations that are unrealistic (i.e., all content hosted on a single "unsharded" domain) where H2 benefits are expected to arise [22].

## 4. Dataset disclosure and Ethical considerations

When conducting experiments that involve people, ethical issues typically arise. In this work, we followed the best practices to minimize eventual issues and discrimination. In organizing the experiments, we followed the "The Menlo Report" guidelines [5], and in particular the directions given for network measurements [2].

First, we informed the participants about the goals of the experiment and the methodology used to collect data by giving them an introductory class of two hours. Participants were informed about the aim of the collection campaign, the tasks they were expected to perform, and the duration of their involvement. In addition, it was granted them assurance about anonymity of the collected data and about voluntary participation, i.e., they were free to refuse to participate without any penalty. They were also provided with names and contacts of persons in charge of the experiments for any further need or question. We did not run any pre-screening and post-screaning of subjects. No restriction was applied, e.g., we did not check gender, age, race, national origin, socio-economical status, etc.

Second, we took all the possible precautions to make the process easy and straight-forward so to avoid making participants uncomfortable, stressed or confused about the tasks. To achieve this, participants were asked to run experiments only after a thorough testing of the process, and technical support was offered during the whole duration of the tests.

Third, when running the experiments and collecting results, the anonymity of the participants was guaranteed. Each participant freely chose in which place to sit in the lab, so that no association between the computer and the user was possible. A randomly-generated identifier (ID) was automatically assigned to each user. This makes possible to track all the experiments executed by single participants, a desirable feature for later stage analysis, but the ID is in no way associable to a specific computer in the lab or a person.

Considering the dataset release, we took any possible action to remove eventual traces leading to the identification of the person running the experiments: we deleted any timestamp, and sequence of records has been randomized to get rid of any eventual time correlations. Each record contains only metadata about the scenario and test, with no indication of the terminal (and user) running the test.

## References

[1] Alexa Internet Inc. `http://www.alexa.com`.
[2] M. Allman 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.
[3] Amazon Inc. `http://www.fastcompany.com/1825005/how-one-second-could-cost-amazon-16-billion-sales`.
[4] Apache HTTP Server Project. `https://httpd.apache.org/docs/2.4/new_features_2_4.html`.
[5] M. Bailey, D. Dittrich, E. Kenneally, and D. Maughan. The menlo report. *IEEE Security & Privacy*, 10(2):71–75, 2012.
[6] M. Belshe, R. Peon, and M. Thomson. Hypertext Transfer Protocol version 2 (HTTP/2). In *IETF RFC7540*, May 2015.

[7] B. Blackburn, M. Varvello, K. Schomp, D. Naylor, A. Finamore, and K. Papagiannaki. Is the Web HTTP/2 yet? In *TMA PhD School*, 2016.
[8] 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)*, Aug. 2016.
[9] 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`.
[10] 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, May 2015.
[11] H. de Saxcé, I. Oprescu, and Y. Chen. Is HTTP/2 really faster than HTTP/1.1? In *Proc. IEEE Global Internet Symposium*, pages 293–299, April 2015.
[12] J. Erman, V. Gopalakrishnan, R. Jana, and K. K. Ramakrishnan. Towards a SPDY'Ier Mobile Web? In *Proc. ACM CoNEXT*, pages 303–314, December 2013.
[13] Google Inc. `https://googlewebmastercentral.blogspot.fr/2010/04/using-site-speed-in-web-search-ranking.html`.
[14] Google Inc. `http://googleresearch.blogspot.fr/2009/06/speed-matters.html`.
[15] Google Inc. `https://www.youtube.com/watch?v=muSIzHurn4U`.
[16] Google Inc. `https://sites.google.com/a/webpagetest.org/docs/using-webpagetest/metrics/speed-index`.
[17] Google Inc. `https://www.ampproject.org/`.
[18] Google Inc. `https://developer.chrome.com/devtools`.
[19] Google Inc. `https://github.com/cyrus-and/chrome-har-capturer`.
[20] Google Inc. QUIC, a multiplexed stream transport over UDP. `https://www.chromium.org/quic`.
[21] Google Inc. Spdy: An experimental protocol for a faster web. `https://www.chromium.org/spdy/spdy-whitepaper`.
[22] I. Grigorik. Http/2 is here, let's optimize! `http://bit.ly/http2-opt`.
[23] I. Grigorik. *High Performance Browser Networking*. O'Reilly Media, 2013.
[24] International Telecommunication Union. Methods for the subjective assessment of video quality, audio quality and audiovisual quality of internet video and distribution quality television in any environment. ITU-T Recommendation P.913, Sept. 2014.
[25] International Telecommunication Union. Subjective testing methodology for web browsing. ITU-T Recommendation P.1501, Feb. 2014.
[26] ITU-T. Recommendation G.1030: Estimating end-to-end performance in IP networks for data applications, 2005.
[27] S. Khirman and P. Henriksen. Relationship between quality-of-service and quality-of-experience for public internet service. In *Proc. Passive and Active Measurements (PAM)*, 2002.
[28] M. Varvello, K. Schomp, D. Naylor, J. Blackburn, A. Finamore, and K. Papagiannaki. Is The Web HTTP/2 Yet? In *Proc. Passive and Active Measurement (PAM)*, pages 218–232, Apr. 2016.
[29] R. B. Miller. Response time in man-computer conversational transactions. In *Proc. AFIPS Fall Joint Computer Conference*, pages 267–277, 1968.
[30] C. Morgan. IAB Statement on Internet Confidentiality. Nov 2014.
[31] D. Naylor, A. Finamore, I. Leontiadis, Y. Grunenberger, M. Mellia, M. Munafò, K. Papagiannaki, and P. Steenkiste. The Cost of the "S" in HTTPS. In *Proc. ACM CoNEXT*, pages 133–140, 2014.
[32] 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, July 2015.
[33] J. Nielsen. Response times: The 3 important limits. `https://www.nngroup.com/articles/response-times-3-important-limits/`.
[34] L. Popa, A. Ghodsi, and I. Stoica. HTTP as the Narrow Waist of the Future Internet. In *Proc. 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, 2010.

[35] F. Qian, V. Gopalakrishnan, E. Halepovic, S. Sen, and O. Spatscheck. TM3: Flexible Transport-layer Multi-pipe Multiplexing Middlebox Without Head-of-line Blocking. In *Proc. ACM CoNEXT*, December 2015.
[36] S. Souders. Velocity and the bottom line. `http://radar.oreilly.com/2009/07/velocity-making-your-site-fast.html`.
[37] D. Thaler. Evolution of the ip model. Technical report, Internet Engineering Task Force, May 2011.
[38] Titus, Tobin and Mann, Jatinder and Jain, Arvind . Navigation Timing Level 2, W3C Editor's Draft 22. `http://w3c.github.io/navigation-timing/`, Apr. 2016.
[39] X. S. Wang, A. Balasubramanian, A. Krishnamurthy, and D. Wetherall. How Speedy is SPDY? In *Proc. USENIX NSDI*, pages 387–399, Apr. 2014.
[40] X. S. Wang, A. Krishnamurthy, and D. Wetherall. Speeding up Web Page Loads with Shandian. In *Proc. USENIX NSDI*, pages 109–122, Mar. 2016.
[41] K. Zarifis, M. Holland, M. Jain, E. Katz-Bassett, and R. Govindan. Modeling HTTP/2 Speed from HTTP/1 Traces. In *Proc. Passive and Active Measurement (PAM)*, 2016.

## Appendix A. Tables

The first Appendix presents a thorough analysis of the web pages catalog. Web pages macro features (e.g., page size, number of objects, number of contacted domains and RTT towards such domains) can be contrasted at a glance in Fig. 2. The following tables are instead intended for a more detailed view of web pages, focusing on objects being retrieved, the related amount of bytes, and network features like the number of domains contacted and the amount of connections opened to fetch all the required objects. Specifically:

- Tab. 2 reports statistics on the objects composing each web page. It provides the total number of object fetched to build the page together with breakdown per object types (e.g., Html, Images, etc.). The ratio between each type and the total amount of objects is reported within brackets.
- Tab. 3 shows the overall amount of downloaded bytes, together with the mean and standard deviation of bytes in a per-object fashion. A breakdown of retrieved bytes per object type (as defined in Tab. 2) is provided, as well as the ratio with respect to the total amount of retrieved bytes.
- Tab. 4 highlights the number of contacted domains and established connections, reporting the mean and the standard deviation of objects and bytes retrieved from each domain / over each connection.

| **Web page** | **Total Objects** | **Object Types** *(Ratio)* | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Html | | Images | | Style | | Scripting | | Other | |
| Synthetic web page | 361 | 1 | *(0.00)* | 360 | *(1.00)* | 0 | *(0.00)* | 0 | *(0.00)* | 0 | *(0.00)* |
| pole-emploi.fr | 71 | 2 | *(0.03)* | 37 | *(0.52)* | 5 | *(0.07)* | 0 | *(0.00)* | 27 | *(0.38)* |
| booking.com | 67 | 1 | *(0.01)* | 46 | *(0.69)* | 6 | *(0.09)* | 10 | *(0.15)* | 4 | *(0.06)* |
| free.fr | 32 | 2 | *(0.06)* | 10 | *(0.31)* | 8 | *(0.25)* | 7 | *(0.22)* | 5 | *(0.16)* |
| lcl.fr | 103 | 3 | *(0.03)* | 66 | *(0.64)* | 10 | *(0.10)* | 23 | *(0.22)* | 1 | *(0.01)* |
| societegenerale.fr | 21 | 1 | *(0.05)* | 13 | *(0.62)* | 1 | *(0.05)* | 4 | *(0.19)* | 2 | *(0.10)* |
| leboncoin.fr | 52 | 2 | *(0.04)* | 38 | *(0.73)* | 1 | *(0.02)* | 10 | *(0.19)* | 1 | *(0.02)* |
| amazon.com | 213 | 9 | *(0.04)* | 180 | *(0.85)* | 7 | *(0.03)* | 8 | *(0.04)* | 9 | *(0.04)* |
| bouyguestelecom.fr | 34 | 1 | *(0.03)* | 21 | *(0.62)* | 3 | *(0.09)* | 6 | *(0.18)* | 3 | *(0.09)* |
| stackoverflow.com | 26 | 1 | *(0.04)* | 13 | *(0.50)* | 2 | *(0.08)* | 7 | *(0.27)* | 3 | *(0.12)* |
| dailymotion.com | 51 | 3 | *(0.06)* | 39 | *(0.76)* | 2 | *(0.04)* | 4 | *(0.08)* | 3 | *(0.06)* |
| laredoute.fr | 51 | 2 | *(0.04)* | 28 | *(0.55)* | 1 | *(0.02)* | 3 | *(0.06)* | 17 | *(0.33)* |
| leroymerlin.fr | 66 | 2 | *(0.03)* | 51 | *(0.77)* | 2 | *(0.03)* | 9 | *(0.14)* | 2 | *(0.03)* |
| labanquepostale.fr | 65 | 2 | *(0.03)* | 30 | *(0.46)* | 8 | *(0.12)* | 13 | *(0.20)* | 12 | *(0.18)* |
| chinadaily.com.cn | 212 | 50 | *(0.24)* | 133 | *(0.63)* | 5 | *(0.02)* | 20 | *(0.09)* | 4 | *(0.02)* |
| diply.com | 106 | 11 | *(0.10)* | 75 | *(0.71)* | 3 | *(0.03)* | 4 | *(0.04)* | 13 | *(0.12)* |
| tf1.fr | 77 | 1 | *(0.01)* | 32 | *(0.42)* | 2 | *(0.03)* | 33 | *(0.43)* | 9 | *(0.12)* |
| imdb.com | 130 | 8 | *(0.06)* | 80 | *(0.62)* | 12 | *(0.09)* | 25 | *(0.19)* | 5 | *(0.04)* |
| over-blog.com | 170 | 13 | *(0.08)* | 104 | *(0.61)* | 9 | *(0.05)* | 21 | *(0.12)* | 23 | *(0.14)* |
| leboncoin-iledefrance-.fr | 72 | 4 | *(0.06)* | 48 | *(0.67)* | 1 | *(0.01)* | 14 | *(0.19)* | 5 | *(0.07)* |
| darty.com | 143 | 7 | *(0.05)* | 86 | *(0.60)* | 3 | *(0.02)* | 29 | *(0.20)* | 18 | *(0.13)* |
| marmiton.org | 180 | 14 | *(0.08)* | 106 | *(0.59)* | 11 | *(0.06)* | 39 | *(0.22)* | 10 | *(0.06)* |
| allocine.fr | 178 | 7 | *(0.04)* | 130 | *(0.73)* | 3 | *(0.02)* | 20 | *(0.11)* | 18 | *(0.10)* |
| bfmtv.com | 127 | 11 | *(0.09)* | 64 | *(0.50)* | 3 | *(0.02)* | 20 | *(0.16)* | 29 | *(0.23)* |
| commentcamarche.net | 200 | 15 | *(0.08)* | 117 | *(0.59)* | 9 | *(0.05)* | 39 | *(0.20)* | 20 | *(0.10)* |

TABLE 2. Web pages characterization – Objects details.
For each web page we report (i) the total number of objects; (ii) the number of objects per type (e.g., html, images, css, javascript); and (iii) the ratio of each type over the total number of objects.

| Web page | Total KBytes | KBytes Mean | KBytes Stdev | KBytes Per Object Type *(Ratio)* | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Html | | Images | | Style | | Scripting | | Other | |
| Synthetic web page | 952 | 2.64 | 7.39 | 3 | *(0.00)* | 949 | *(1.00)* | 0 | *(0.00)* | 0 | *(0.00)* | 0 | *(0.00)* |
| pole-emploi.fr | 1205 | 16.97 | 26.57 | 110 | *(0.09)* | 498 | *(0.41)* | 224 | *(0.19)* | 0 | *(0.00)* | 373 | *(0.31)* |
| booking.com | 1157 | 17.27 | 26.41 | 98 | *(0.08)* | 378 | *(0.33)* | 178 | *(0.15)* | 349 | *(0.30)* | 154 | *(0.13)* |
| free.fr | 724 | 22.63 | 38.05 | 11 | *(0.01)* | 415 | *(0.57)* | 20 | *(0.03)* | 77 | *(0.11)* | 201 | *(0.28)* |
| lcl.fr | 1669 | 16.20 | 31.09 | 84 | *(0.05)* | 918 | *(0.55)* | 80 | *(0.05)* | 572 | *(0.34)* | 16 | *(0.01)* |
| societegenerale.fr | 318 | 15.13 | 18.56 | 6 | *(0.02)* | 222 | *(0.70)* | 13 | *(0.04)* | 71 | *(0.22)* | 5 | *(0.02)* |
| leboncoin.fr | 325 | 6.24 | 15.16 | 38 | *(0.12)* | 187 | *(0.58)* | 3 | *(0.01)* | 95 | *(0.29)* | 2 | *(0.00)* |
| amazon.com | 5714 | 26.83 | 42.11 | 122 | *(0.02)* | 5294 | *(0.93)* | 52 | *(0.01)* | 177 | *(0.03)* | 68 | *(0.01)* |
| bouyguestelecom.fr | 689 | 20.26 | 35.04 | 13 | *(0.02)* | 318 | *(0.46)* | 45 | *(0.07)* | 75 | *(0.11)* | 237 | *(0.34)* |
| stackoverflow.com | 391 | 15.05 | 19.22 | 41 | *(0.10)* | 188 | *(0.48)* | 52 | *(0.13)* | 64 | *(0.16)* | 47 | *(0.12)* |
| dailymotion.com | 815 | 15.98 | 19.30 | 34 | *(0.04)* | 537 | *(0.66)* | 70 | *(0.09)* | 119 | *(0.15)* | 56 | *(0.07)* |
| laredoute.fr | 1055 | 20.68 | 26.78 | 26 | *(0.02)* | 638 | *(0.60)* | 44 | *(0.04)* | 46 | *(0.04)* | 301 | *(0.29)* |
| leroymerlin.fr | 915 | 13.87 | 20.68 | 28 | *(0.03)* | 618 | *(0.68)* | 33 | *(0.04)* | 191 | *(0.21)* | 46 | *(0.05)* |
| labanquepostale.fr | 1086 | 16.71 | 27.36 | 17 | *(0.02)* | 771 | *(0.71)* | 62 | *(0.06)* | 138 | *(0.13)* | 98 | *(0.09)* |
| chinadaily.com.cn | 3387 | 15.97 | 37.45 | 90 | *(0.03)* | 3133 | *(0.93)* | 27 | *(0.01)* | 127 | *(0.04)* | 10 | *(0.00)* |
| diply.com | 6803 | 64.18 | 144.69 | 51 | *(0.01)* | 6252 | *(0.92)* | 82 | *(0.01)* | 85 | *(0.01)* | 332 | *(0.05)* |
| tfl.fr | 1532 | 19.90 | 35.12 | 21 | *(0.01)* | 1001 | *(0.65)* | 78 | *(0.05)* | 298 | *(0.19)* | 135 | *(0.09)* |
| imdb.com | 1590 | 12.23 | 14.27 | 93 | *(0.06)* | 812 | *(0.51)* | 136 | *(0.09)* | 540 | *(0.34)* | 9 | *(0.01)* |
| over-blog.com | 3632 | 21.36 | 28.65 | 93 | *(0.03)* | 2383 | *(0.66)* | 128 | *(0.04)* | 427 | *(0.12)* | 600 | *(0.17)* |
| leboncoin-iledefrance-.fr | 468 | 6.50 | 9.33 | 43 | *(0.09)* | 209 | *(0.45)* | 9 | *(0.02)* | 139 | *(0.30)* | 68 | *(0.15)* |
| darty.com | 2354 | 16.46 | 30.12 | 62 | *(0.03)* | 1454 | *(0.62)* | 35 | *(0.01)* | 228 | *(0.10)* | 575 | *(0.24)* |
| marmiton.org | 2515 | 13.97 | 31.38 | 123 | *(0.05)* | 1825 | *(0.73)* | 114 | *(0.05)* | 315 | *(0.13)* | 139 | *(0.06)* |
| allocine.fr | 2812 | 15.80 | 30.16 | 77 | *(0.03)* | 1858 | *(0.66)* | 83 | *(0.03)* | 239 | *(0.09)* | 554 | *(0.20)* |
| bfmtv.com | 2641 | 20.79 | 102.77 | 43 | *(0.02)* | 1692 | *(0.64)* | 74 | *(0.03)* | 291 | *(0.11)* | 541 | *(0.20)* |
| commentcamarche.net | 3079 | 15.39 | 33.43 | 140 | *(0.05)* | 1937 | *(0.63)* | 98 | *(0.03)* | 610 | *(0.20)* | 294 | *(0.10)* |

TABLE 3. Web pages characterization – Bytes details.
For each web page we report (i) the total number of bytes; (ii) their per-object average and standard deviation; (iii) the number of bytes per object type; and (iv) the ratio of bytes per object type over the total number of bytes.

| Web page | Domains No. | Per-Domain | | | | Connect No. | Per-Connection | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Objects | | KBytes | | | Objects | | KBytes | |
| | | Mean | Std | Mean | Std | | Mean | Std | Mean | Std |
| Synthetic web page | 1 | 361.00 | 0.00 | 952 | 0 | 6 | 60.17 | 1.95 | 159 | 5 |
| pole-emploi.fr | 3 | 23.67 | 27.26 | 402 | 355 | 8 | 8.88 | 6.47 | 151 | 94 |
| booking.com | 4 | 16.75 | 13.88 | 289 | 231 | 16 | 4.19 | 2.60 | 72 | 53 |
| free.fr | 5 | 6.40 | 9.81 | 145 | 219 | 10 | 3.20 | 2.23 | 72 | 81 |
| lcl.fr | 5 | 20.60 | 39.20 | 334 | 658 | 10 | 10.30 | 7.71 | 167 | 159 |
| societegenerale.fr | 5 | 4.20 | 5.42 | 64 | 84 | 21 | 1.00 | 0.00 | 15 | 19 |
| leboncoin.fr | 7 | 7.43 | 11.10 | 46 | 40 | 15 | 3.47 | 2.68 | 22 | 26 |
| amazon.com | 8 | 26.63 | 62.15 | 714 | 1810 | 20 | 10.65 | 13.29 | 286 | 404 |
| bouyguestelecom.fr | 9 | 3.78 | 5.94 | 77 | 180 | 15 | 2.27 | 1.53 | 46 | 62 |
| stackoverflow.com | 10 | 2.60 | 2.29 | 39 | 74 | 12 | 2.17 | 1.72 | 33 | 45 |
| dailymotion.com | 10 | 5.10 | 6.91 | 81 | 106 | 23 | 2.22 | 1.89 | 35 | 35 |
| laredoute.fr | 11 | 4.64 | 5.96 | 96 | 163 | 19 | 2.68 | 1.92 | 56 | 50 |
| leroymerlin.fr | 11 | 6.00 | 9.87 | 83 | 150 | 25 | 2.64 | 2.68 | 37 | 54 |
| labanquepostale.fr | 13 | 5.00 | 11.29 | 84 | 265 | 19 | 3.42 | 3.31 | 57 | 101 |
| chinadaily.com.cn | 16 | 13.25 | 27.60 | 212 | 572 | 71 | 2.99 | 5.98 | 48 | 128 |
| diply.com | 17 | 6.24 | 15.21 | 400 | 1451 | 26 | 4.08 | 4.05 | 262 | 459 |
| tf1.fr | 17 | 4.53 | 7.59 | 90 | 172 | 37 | 2.08 | 1.60 | 41 | 54 |
| imdb.com | 21 | 6.19 | 11.85 | 76 | 162 | 35 | 3.71 | 3.80 | 45 | 69 |
| over-blog.com | 23 | 7.39 | 10.56 | 158 | 372 | 39 | 4.36 | 4.60 | 93 | 125 |
| leboncoin-iledefrance-.fr | 24 | 3.00 | 2.71 | 20 | 28 | 39 | 1.85 | 1.87 | 12 | 16 |
| darty.com | 33 | 4.33 | 9.63 | 71 | 325 | 56 | 2.55 | 2.93 | 42 | 108 |
| marmiton.org | 34 | 5.29 | 12.66 | 74 | 278 | 66 | 2.73 | 3.99 | 38 | 114 |
| allocine.fr | 41 | 4.34 | 5.09 | 69 | 139 | 62 | 2.87 | 3.78 | 45 | 77 |
| bfmtv.com | 45 | 2.82 | 3.73 | 59 | 205 | 103 | 1.23 | 0.94 | 26 | 114 |
| commentcamarche.net | 47 | 4.26 | 7.36 | 66 | 173 | 86 | 2.33 | 3.01 | 36 | 74 |

TABLE 4. Web pages characterization – Domains & Connections. For each web page we report (i) the number of domains contacted; (ii) the number of objects and bytes (both average and standard deviation) hosted on each domain; (iii) the number of established connections; and (iv) the number of objects and bytes (both average and standard deviation) retrieved over each connection.

## Appendix B. QoE Metrics and Web Pages Features

| MOS | | |
|---|---|---|
| 1 | MOS | Subjective experience feedback |

| OBJ Features | | |
|---|---|---|
| **12 features** | Latency | Configured latency in local testbed |
| | Loss | Configured loss in local testbed |
| | DOM | Document Object Model timing |
| | PLT | PLT timing |
| | TTFB | Time to the first byte |
| | TTLB | Time to the last byte |
| | ObjectIndex | Object Index [8] |
| | ByteIndex | Byte Index [8] |
| | Output bytes | Number of uploaded bytes |
| | Input bytes | Number of downloaded bytes |
| | Domains | Number of contacted domains |
| | Connections | Number of established connections |

| Protocol/Page Features | | |
|---|---|---|
| | Object Number | Total number of objects of the web page |
| | Object HTML | Number of HTML objects |
| | Object IMG | Number of images (e.g., png, jpg) |
| | Object STYLE | Number of CSS objects (e.g., css) |
| | Object SCRIPT | Number of scripting objects (e.g., javascript) |
| | Object OTHER | Number of other object types |
| | Obj-HTML ratio | Ratio of HTML over total number of objects |
| | Obj-IMG ratio | Ratio of IMG over total number of objects |
| | Obj-STYLE ratio | Ratio of STYLE over total number of objects |
| | Obj-SCRIPT ratio | Ratio of SCRIPT over total number of objects |
| | Obj-OTHER ratio | Ratio of OTHER over total number of objects |
| **38 features** | Bytes Total | Total number of bytes for objects |
| | Bytes HTML | Bytes for HTML objects |
| | Bytes IMG | Bytes for images (e.g., png, jpg) |
| | Bytes STYLE | Bytes for style objects |
| | Bytes SCRIPT | Bytes for scripting objects |
| | Bytes OTH | Bytes for other object types |
| | Bytes Mean | Average amount of bytes per object |
| | Bytes Std | Standard Deviation of bytes per object |
| | Byte-HTML ratio | Ratio of bytes for HTML objects |
| | Byte-IMG ratio | Ratio of bytes for images objects |
| | Byte-STYLE ratio | Ratio of bytes for style objects |
| | Byte-SCRIPT ratio | Ratio of bytes for scripting objects |
| | Byte-OTH ratio | Ratio of bytes for other object types |
| | Obj-per-Domain Mean | Average number of objects per domain |
| | Obj-per-Domain Std | Standard deviation of objects number per domain |
| | Byte-per-Domain Mean | Average number of bytes per domain |
| | Byte-per-Domain Std | Standard deviation of bytes per domain |
| | Obj-per-Connect Mean | Average number of objects per connection |
| | Obj-per-Connect Std | Standard deviation of objects number per connection |
| | Byte-per-Connect Mean | Average number of bytes per connection |
| | Byte-per-Connect Std | Standard deviation of bytes per connection |
| | RTT-per-Domain Mean | Average RTT per domain |
| | RTT-per-Domain Std | Standard deviation of RTT per domain |
| | RTT-per-Object Mean | Average RTT per object |
| | RTT-per-Object Std | Standard deviation of RTT per object |
| | RTT-per-Byte Mean | Average RTT per byte |
| | RTT-per-Byte Std | Standard deviation of RTT per byte |

Table 5. Full list of relevant features.