Web Experience in Mobile Networks: Lessons from Two Million Page Visits

Mohammad Rajiullah[1], Andra Lutu[2], Ali Safari Khatouni[3,4], Mah-rukh Fida[5], Marco Mellia[4], Özgü Alay[5], Anna Brunstrom[1], Stefan Alfredsson[1], Vincenzo Mancuso[6]

[1] Karlstad University, Sweden [2] Telefonica Research, Spain [3] Dalhousie University, Canada [4] Politecnico di Torino, Italy [5] Simula Metropolitan, Norway [6] IMDEA Networks Institute, Spain

ABSTRACT

Measuring and characterizing web page performance is a challenging task. When it comes to the mobile world, the highly varying technology characteristics coupled with the opaque network configuration make it even more difficult. Aiming at reproducibility, we present a large scale empirical study of web page performance collected in eleven commercial mobile networks spanning four countries. By digging into measurement from nearly two million web browsing sessions, we shed light on the impact of different web protocols, browsers, and mobile technologies on the web performance. We found that the impact of mobile broadband access is sizeable. For example, the median page load time using mobile broadband increases by a third compared to wired access. Mobility clearly stresses the system, with handover causing the most evident performance penalties. Contrariwise, our measurements show that the adoption of HTTP/2 and QUIC has practically negligible impact. To understand the intertwining of all parameters, we adopt state-of-the-art statistical methods to identify the significance of different factors on the web performance. Our analysis confirms the importance of access technology and mobility context as well as webpage composition and browser. Our work highlights the importance of large-scale measurements. Even with our controlled setup, the complexity of the mobile web ecosystem is challenging to untangle. For this, we are releasing the dataset as open data for validation and further research.

1 INTRODUCTION

We are witnessing two major changes in the Internet. On the one hand, smartphones and mobile broadband (MBB) networks have revolutionized the way people consume web content. On the other hand, web protocols are undergoing a deep rethinking, with the advent of SPDY [32], HTTP/2 [16] and now QUIC [38], that are designed to improve performance and overcome HTTP(S) [27, 47] and TCP bottlenecks.

In this dynamic scenario, improving the end-user quality of experience (QoE) is the ultimate goal for companies that aim at offering better service to their customers. Natural questions arise: Which are the major factors that affect web performance in MBB? What are the benefits of HTTP/2 and QUIC? How does mobility impact the user experience? Are MBB networks at the point where they are able to offer QoE on par with wired networks?

Measurements are vital to gauge the impact these changes. Not surprisingly, the research community has put a lot of effort towards empirical quantification of the benefits of new technologies. However, to the best of our knowledge, previous work mainly focused on specific measurement setups, and often on specific angles of this otherwise complex ecosystem. The intertwining of technologies, protocols, setups and website design makes it very complicated to answer the above questions, to design and run scientifically sound and robust measurement campaigns.

In this paper, we present a large scale measurement campaign that we carefully designed following the scientific principles of repeatability and reproducibility. To this end, we leverage the open access MONROE [11] platform that allows us to run measurements with full control of devices scattered in more than 100 locations across four different countries and connected via 11 commercial MBB providers. We instrument the measurement campaign over a period of two months and collect nearly two million page visits of popular websites, which we further augment with metadata about the MBB channel and network status. We dig into this large amount of data to quantify the penalties when accessing the web using 3G or 4G access technology, under mobility, and the benefits of adopting HTTP/2 or QUIC rather than HTTPS. We quantify differences using state of the art statistics tools that allow us not only to obtain the significance of the measurements, but also to study which parameters affect results the most.

The contributions we make in this paper are four-fold. First, we present our methodology to design large-scale and scientifically sound measurements (Section 3). We offer all collected data to the community in the effort of allowing researchers to verify and extend our findings. Second, we quantify the impact of MBB access technology in both stationary and mobile scenarios. We use two web browsers (Firefox and

Chrome), and three objective QoS metrics (First Paint, Page Load Time, and RUMSpeedIndex). With respect to wired access, results consistently show significant penalties from MBB networks, with mobile nodes suffering further penalties due to changing channel conditions, and inter-technology handovers (Section 5). Third, we investigate deeper the benefits of HTTP/2 and QUIC versus HTTPS (Sec 6). In some specific cases (in benchmarking pages), we do see HTTP/2 improvements. In the wild, however, results show marginal benefits and controversial figures. Even when comparing impact on wired networks, we record no significant benefits in 55% of back-to-back visits, with the rest of measurements equally split among HTTP/2 performs better than HTTPS, and conversely. When moving to MBB scenarios, the additional randomness of the setup makes it even harder to observe any significant benefits. QUIC does not change the picture either. Fourth, data analytics unveils the impact of single features that characterize the context in which browsing occurs (Section 7).

Our results show that there is a large number of features that significantly impact web performance, and, ultimately, QoE. Among these, the MBB access technology and mobility context, along with webpage composition and the browser used are among the most important. Before moving into the details of our contributions (Sections 3-6), we present background and related work in Section 2 and we follow up our analysis with concluding remarks in Section 8.

2 BACKGROUND AND RELATED WORK

There are mainly three approaches for measuring the performance and reliability of MBB networks: (i) crowd-sourced measurements [43, 48, 53], (ii) operator's measurements [30, 31, 50, 51], and (iii) measurements collected using dedicated infrastructure [14, 37, 49]. In this paper, we collect data from the MONROE dedicated infrastructure to gain full control over the measurements and systematically collect a rich quality dataset over more than two months.

HTTP/1.1 [27] has been the de-facto standard for loading webpages since 1999. With the increasing complexity in today's webpages, HTTP/1.1 (also HTTP over TLS or HTTPS -hereinafter H1s- [47]) has been shown to limit the performance of web access [55] due to problems such as inefficient use of concurrent TCP connections [46] and header overhead. To improve web performance, Google proposed SPDY [32], which soon led to HTTP/2.0 -hereinafter H2- [16]. The latter uses a single multiplexed TCP connection per domain, allowing request prioritization. Efficiency is further optimized through header compression. However, TCP is still a bottleneck because of handshaking latency and head-of-line blocking. To address these shortcomings, Google proposed QUIC [33, 38] that uses UDP instead of TCP.

The relative performance difference between HTTP versions has received much attention [18, 23, 25, 26, 29, 39, 44, 57–60, 62]. The results are mixed, hence it is hard to make a clear conclusion on which version is better and under which circumstances. Moreover, there is no systematic attempt to characterize HTTP performance for MBB users. Similarly, we found few experimental works on QUIC performance in operational networks [38], mostly considering synthetic webpages [17, 33]. We aim to present a large-scale web measurements campaign using operational MBBs across different countries. We not only measure the protocol difference from different vantage points, but also set the operators performance in a common scale. We run measurements using two popular browsers (Chrome and Firefox) to present how these protocols behaves in different browsers.

Previous works also differ in the metric they chose to evaluate performance. Work in [17, 23, 26, 38, 39, 41, 56, 59, 63] used the page load time (PLT), a metric primarily based on OnLoad event fired by the browser. Unfortunately, PLT does not tell much of the user experience [2, 19, 34, 54]. An alternative metric is First Paint, which tells how long it takes to see something on the screen [19, 24]. SpeedIndex captures the user perceived experience [6, 8, 9]. However, it requires to *film* the page loading process. The RUMSpeedIndex method was then proposed to estimate the SpeedIndex using "resource timing" data [3, 5, 7]. ObjectIndex and ByteIndex are similar in goal [19]. Additionally, [28] proposed the "3rd Party Trailing Ratio" metric, which measures the fraction of download time for the 3rd party assets on the webpage critical path. In our measurements, we consider Page Load Time, First Paint and RUMSpeedIndex.

We tackle the need for understanding the relationship between web performance and network characteristics. For example, while generating a model to determine a webpage's QoE, A. Balachandran et al. [13] found evidence of the impact of the number of users and radio access technology (RAT) handovers on the webpage performance. They discovered that the performance is greatly affected by a website's own complexity (e.g., number of objects, domains visited) with some influence of the time-of-day feature. They show that understanding non-binary QoE metrics (e.g., latency, download progress, etc.) requires regression modeling. Here, we use regression to dig into our measurements and bring to light the correlations between various aspects of the browsing process and the experience of the users.

3 MEASUREMENTS DESIGN

For our extensive measurements campaign, we engineer a flexible setup that on the one hand allows us to investigate as many different scenarios as possible, and on the other hand follows the best practice of scientific approach by letting us control as much as possible the experimental setup, with reproducibility in mind. In the following, we describe the experiment design and the collected dataset.

3.1 Experimental Setup

For our experiments, we rely on the MONROE platform, the first European open access platform for independent, multi-homed, and large-scale mobile measurements on commercial providers. Overall, the geographical extent of our measurement campaign covers four countries in Europe (Italy, Norway, Spain and Sweden). The platform comprises a set of 100 nodes equipped with Ethernet, WiFi and 3G/4G interfaces with commercial MBB subscriptions. Nodes operate both under mobility (e.g., on-board of trains or buses) and in stationary (e.g., in laboratories, or hosted by volunteers in homes) scenarios and enable us to measure 11 operators.

We leverage a customizable Docker container called MONROE-browsertime [4] that we specifically engineered and deployed on MONROE nodes. We configured MONROE-browsertime to mimic a mobile device browser (by setting both the screen resolution and the user-agent accordingly) to retrieve the mobile versions of the visited pages. With it, we direct the browser to load a page and, at the end of page rendering, execute a custom Javascript script to collect a large number of metrics. We use the X virtual framebuffer (Xvfb) [10] for display emulation to let the browsers actually draw the webpages. MONROE-browsertime provides a configurable experiment template to enable web measurements in MONROE. We configure each measurement by controlling (i) the network to test (Ethernet, or the desired MBB interfaces), (ii) the browser (Firefox 56.0.1 or Chrome 64.0.3282.186), and (iii) the web protocol (H1s , H2, and QUIC).[1] A combination of these parameters builds an *experiment setup*.

We select a list of target pages to visit. Considering the time limits from the MONROE platform, we select 20 target pages on popular websites with H2 support from the most viewed sites in the Alexa [12] top ranking, see Table 1. We avoid the landing page in case it is too trivial (e.g., visiting `https://instagram.com/leomessi/` instead of `https://instagram.com/`). This ensures that our selection covers a wide range of user interests in terms of topics, including social networking, video, career, education, search engine, travel help, news, wiki and shopping. All websites expect TLS connections by default. This allows us to run fair comparison between H1s and H2.

Given a network to test, the container shuffles the order of pages to visit. We visit each page with every browser and protocol combination in a random order. The visit of all pages with one network setup constitutes a *run*. Browser caches and cookies are active during each run, and cleaned when the last page has been tested. Note that we use two separate profiles for the same browser, one for each protocol, so that caches are separated (i.e., visits with H1s do not interfere with next visits to the same pages with H2 and vice-versa). We execute four rounds of measurements per day.

Table 1: List of the target webpages and their corresponding number of domains and objects. The table details the % of H2 resources in each page obtained using Chrome.

	Wikipedia	Live	Stackoverflow	twitter	Youtube	Instagram	Microsoft	Etsy	Kayak	Ebay	Yelp	Flickr	Reddit	Facebook	Tmall	Imgur	Coursera	Theguardian
# obj.	7	11	21	21	37	40	44	49	52	54	54	62	66	71	97	101	116	277
# domains	1	4	9	7	11	5	8	6	17	20	26	6	21	6	18	27	45	59
% of H2 res.	100	73	76	100	100	100	80	100	100	78	87	97	79	100	79	84	84	70

3.2 Web QoE Metrics

In this paper, we track three main metrics for our analysis: Page Load Time (PLT), FirstPaint (FP) and RUMSpeedIndex (SI), which we detail next. The tool derives these metrics from browser timing metrics [1], [40] that record the timing of different rendering phases of a page, from the initial DNS resolution to each HTTP request, from JavaScript processing to objects rendering.

First Paint (FP): It corresponds to the time when the browser starts rendering the first element of the page. This happens as soon as the first element of the page has been fetched and processed, and after the downloading of all needed elements (e.g., stylesheets).

Page Load Time (PLT): This is the time the last object in the page has been downloaded. It occurs when all the HTML files and any sub-resources (images, fonts, css, videos, etc.) are loaded. Note that not all these elements are needed to complete the rendering of the visible portion of the page.

RUMSpeedIndex (SI): It monitors the rendering process of the page by tracking the evolution of visible rectangles with elements that loaded external resources in a page. The original SpeedIndex requires to film the rendering process, and the postprocessing of the video to observe changes. Given the limited resources of the MONROE nodes (in terms of CPU, storage and communication bandwidth), we opt for the RUMSpeedIndex approximation, which uses the sequence of events as reported by the browser to estimate the time in which the visible portion of the screen would be modified [7]. Intuitively, it calculates the likely time that a paint event happens, given the processing and downloading of the various elements by the browser. The SI corresponds to the time when last paint happens. This is considered a QoE approximation, since it considers the evolution of the rendering process as seen by the user.

Given the interplay of objects, rendering, and visible portion of the page on the screen, there is no clear ordering

[1] The nodes use the mobile carrier DNS resolver consistently with the operator currently in use.

on the metrics. For instance, the rendering of a page can start after one, or some, or all objects have been downloaded. Similarly, the rendering process may involve a portion of the area which is not currently on the visible part on the browser window. In addition, analytics objects are typically downloaded as last elements, after the rendering is completed, thus inflating the PLT. For this, it is consistent to compare results considering the same metric, but it is not appropriate to compare across different metrics.

3.3 Metadata Metrics

Context information is crucial. MONROE gives access to the information about network, time and location, as well as metadata from the mobile modems, including cell ID, signal strength, and connection mode. We log all this information, and associate it to each experiment for later post-processing, to understand the impact of the different operator configurations, the radio coverage, and all the time varying parameters that may impact the page rendering process.

More specifically, we collect the following parameters: i) **Experimental context**: This includes the type of browser, the protocol the node type (stationary/mobile) and the distance the node travelled during an experiment. ii) **Access Network context**: This includes parameters from the the Radio Access Technology (RAT) (more specifically for 3G and 4G technology) such as radio status before the start of the experiment (Initial RAT, Initial Reference Signal Received Quality (RSRQ), Initial Reference Signal Received Power (RSRP), Initial Received Signal Strength Indicator (RSSI)) and the radio changes throughout the experiment (median values for RSRQ, RSRP, RSSI parameters, the number of RAT handovers), and average RTT against the target webpage server (measured via Ping). iii) **Web context:** This includes metrics that characterize the single web page visit, such as the number of objects or the total amount of data being downloaded.

3.4 Dataset Description

For experiments in this paper, we deploy MONROE-browsertime on 100 MONROE nodes, measuring 11 mobile operators in 4 countries. We ran our measurement campaign for more than 2 months, from 1st of April 2018 to 4th of June 2018. This dataset focuses on measurements considering H1s and H2 protocols. In total, we collected performance for about 1.8 million page visits, as detailed in Table 2, of which 550,000 under mobility in trains, buses or trucks. In addition to MBB tests, we run experiments using wired connectivity for those nodes operating in campus networks and connected via regular 100Mb/s or faster Ethernet interfaces. These experiments constitute our baseline.

As a first step, we performed data cleaning by identifying anomalies in the data. Indeed, we observed inconsistencies in the results from `Google search` - Chrome and Firefox were showing different results for the same search url - and `Linkedin` - this site was blocking visits from our nodes, detecting the repeated page visit as illegitimate crawlers, or attacks. We have discarded these sites from any further processing. We also removed those experiments in which either the browser could not complete the page download or some metadata were missing.

Table 2: Statistics on the dataset; the country shows where the subscription is active.

Operator Name	# Tot. Measurements	# Mobile
Telia (SE)	246 839	41 936
Telenor (SE)	249 459	45 580
Tre (SE)	223 050	43 469
Telenor (NO)	226 126	146 781
Telia (NO)	251 960	155 522
ICE (NO)	177 133	72 461
TIM (IT)	99 778	7 793
Vodafone (IT)	86 164	15 110
Wind (IT)	89 799	20 474
Yoigo (ES)	142 502	0
Orange (ES)	44 528	0
Total	1.8M	550K

We run a separate measurement campaign for measuring the performance of the QUIC protocol. This batch of experiments ran during a week, from the 5th to the 10th of June 2018. We focused here on websites that supports QUIC (version 39): `Google search`, `Youtube`, `Lightspeedtech`, `Keycdn`, `Meetup`, `Justwatch`, and `Free-power-point-templates`. We used only Chrome browser (the only browser supporting QUIC). After performing data cleaning operations, we collected approximately 80 000 samples.

To summarize, our full dataset includes nearly two million page visits, a deluge of data that – to the best of our knowledge – constitutes the largest experiment of mobile web performance tests on commercial operators. We open the data for further processing and to enable the reproducibility.[2]

Dataset limitations. Despite all the care we took in designing our experiment to ensure that they are extensive, repeatable, and with limited biases, we discuss some limitations that we could not avoid.

We measure using MONROE hardware, which integrates three 3G/4G MC7455 miniPCI express mobile broadband modems supporting LTE CAT6 on top of PC Engines APU2D4 boards [52]. The end-user device is strongly correlated with the experience of the end-user [22, 42], especially for web browsing. Our design aims to remove device bias and capture a wide geographical footprint with one single type of device.

We use commercial-grade mobile subscriptions for our measurements. While this represents what end customer can

[2]Link removed to respect the double blind policy.

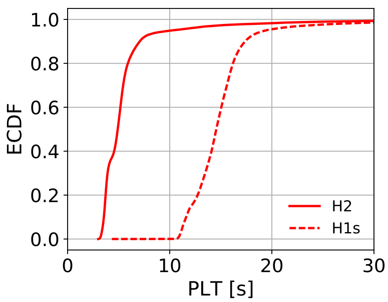


Figure 1: ECDF showing protocol gain for H2/H1s for httpvshttps.com.

get, we do not know which specific configuration each MBB operator is using for each of these (e.g., in terms of QoS, or presence of proxies commonly found in MBB networks [35]). As such, our results do not generalize all possible subscriptions, nor in different locations.

The differences in commercial offers reflect in the different data plans we use in MONROE and in the dataset we collect. We explicitly opted to limit the number of pages in our tests and prefer several repetitions of the same test to obtain a large dataset where we can obtain statistically sound results for each page. Completing a single experimentation takes about 2 hours, making it impossible to cover more pages.

The disparities in the number of samples we could collect from each node stem from the way the MONROE scheduler assigns experiments to the nodes based on their available monthly data quota. For example, we observe in Table 2 that the number of samples for Orange (ES) (limited data plan, 10GB) is much smaller than the number of samples we collected for Telia (SE) (very large data plan, 200GB). Moreover, because of the MONROE deployment strategy, we do not have any mobility measurements in Spain.

The RUMSpeedIndex represents a client-side approach for measuring the Speed Index metric using Real User Monitoring (RUM). Though this metric is appropriate for our experimental setup and goes along the idea of the original Speed Index metric of measuring how much of the "above-the-fold" content is visually complete until the web page has finished loading, it also has a series of limitations. Specifically, it only works for browsers that support Resource Timings, it does not handle content within iframes and does better for Chrome which supports reporting a "first paint" event.

4 VALIDATION OF THE EXPERIMENTAL SETUP

Before digging into measurements, we validate out setup by using a specifically crafted page that allows one to gauge the benefites of H2 vs. H1s, the www.httpvshttps.com page. Hosted by cloudflare, it contains 360 not-cacheable identical images (each 20×20 pixels, 1.7 kB) with a total of 604 kB. All these images are hosted from a single domain. This scenario is particularly beneficial for H2 which can multiplexes all the HTTP requests over a single TCP connection and benefit of header compression. Conversely, H1s would open a large number of TCP connections each suffering from TCP, TLS handshaking overhead and TCP's slowstart. H2 is therefore expected to provide much faster web performance as compared to H1s.

We run a preliminary validation test on all MONROE nodes for every hour continuously for five days producing a total of 130 000 page visits. Figure 1 shows the Empirical Cumulative Distribution Function (ECDF) for PLT for both browsers. H2 provides much better performance than H1s as expected. In nearly 90% visits, H2 manages to load the whole page under 5s whereas using H1s, the PLT rises well over 15s for 80% visits. The behaviour stays similar when we separate the browsers (Firefox and Chrome). This simple test shows that our testbed is indeed giving consistent and realistic results.

5 BROWSING CONTEXT IMPACT

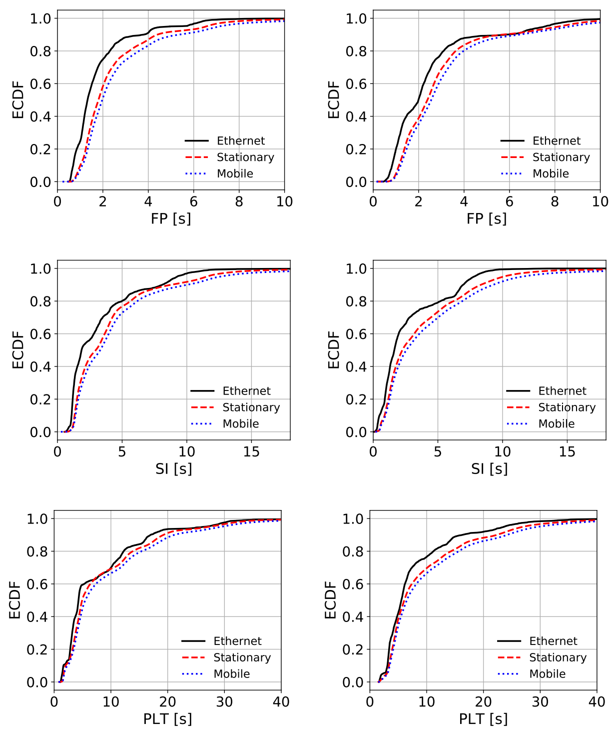
In this section, we explore our measurement results and highlight the impact of major parameters, such as MBB access and mobility. We then dig into more details to quantify the impact of radio access status leaving the discussion on web protocol impact to Section 6.

5.1 Mobile Access and Browser Impact

We aim to assess the impact of MBB access and mobility compared to wired access. Figure 2 shows the three performance metrics separately for Chrome (on the left) and Firefox (on the right). For each metric (notice the different scales), we calculate again the ECDF for i) stationary nodes with wired connection (black solid line), ii) stationary nodes with MBB connections (red dashed line), and iii) mobile nodes with MBB connections (blue dotted line). Smaller values correspond to better performance.

Few considerations hold: First, wired connections have better performance than 3G/4G stationary nodes, which in turn perform better than 3G/4G mobile nodes (consistent across all metrics). Interestingly, the performance degradation with respect to wired high-speed access is much more visible than the impact of mobility. For instance, the FP in Chrome (top-left figure) shows that the median worsens by 36.5% when comparing Wired and MBB access, and only by 9.6% when comparing stationary and mobile nodes. We verified this to be consistent in single pages, and found more complex pages to be more impacted (as expected).

Second, all ECDFs exhibit a heavy tail in measurements, with the mass of the distribution concentrated on low values and a non-negligible number of samples that reach much



(a) Chrome **(b) Firefox**

Figure 2: Browsing performance on commercial MBB networks in two different scenarios (moving and stationary mobile end-users) considering all three metrics for two browsers. We use the browsing performance on fixed Internet provider connectivity (Ethernet) as a baseline.

higher indexes. This is more evident for MBB tests. For example, in the case of FP for Chrome, we note 60-80% of pages start the rendering process in 2-3 seconds. But more than 20% of pages have to wait for more than 4-6 s (i.e., a 2x factor). This heavy tailed distribution is consistent on all pages, hinting at variability among single experiments.

Third, and less intuitive, results on Firefox and Chrome are hardly comparable. For example, for the FP metric (top plots, Figure 2), Firefox clearly has worse performance than Chrome. However, when checking the PLT (bottom plots, Figure 2), the difference is much less evident. To analyze this in more detail, we focus on the measurements we collect from stationary nodes in Sweden, and compare the impact of the browser. Figure 3a shows much less evident results, with Chrome performing slightly better than Firefox. However, when restricting to a single website, results can change dramatically, depending on the target website. For example, for the case of ebay in Figure 3b, we note that Chrome is faster and less variable than Firefox. Figure 3c illustrates the impact of the number of objects of the target website. Page rendering may start only after some specific object has been downloaded (e.g., fonts, CSS, and background images) which may be loaded later, as captured by FP. The PLT metric, on the other hand, tends to be affected by pages that include analytics services that are loaded as the last object – and often after executing complex scripts. Thus, we observe that PLT may increase considerably with number of objects.

Takeaway: MBB access and mobility are factors that impact performance in a consistent way. However, even in controlled environments we observe a lot of variations for all performance metrics and there are different factors that impact the page rendering process in a not trivial way. We investigate this in detail in Section 7.

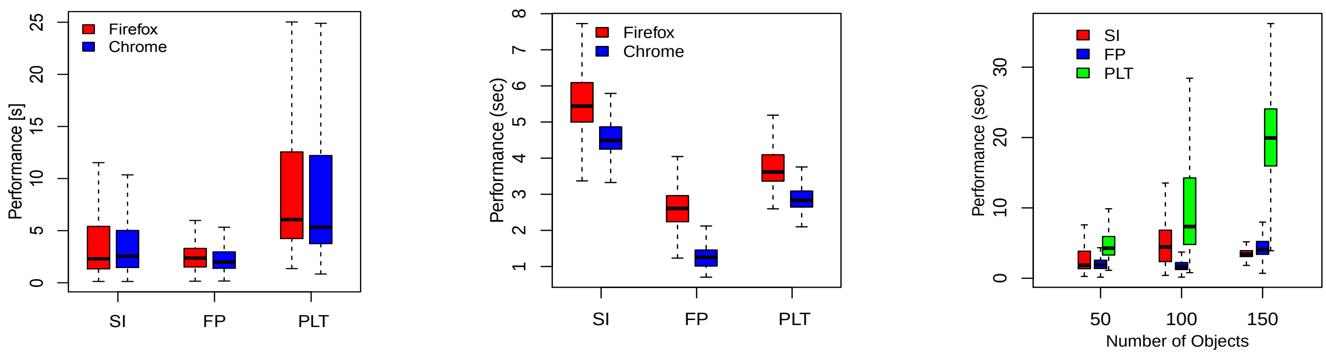
5.2 Network Context Impact

As the capacity of the mobile networks continues to increase, users expect better performance. As we saw in the previous section, MBB access and mobility significantly impact the user experience and it correlates with highly varying network conditions. With MBB technologies that are much more complex than wired access, we need to delve into more details to understand what is the impact of particular setup.

We first observe the impact of the Radio Access Technology (3G and 4G/LTE). Using the experiments from nodes within Sweden under mobility scenarios, Figure 4a captures the impact of RAT on web performance. The better performance offered by 4G/LTE technology benefits the FP and SI albeit not as dramatically as one would expected [45]. Instead, the PLT seems to be marginally affected. Again, this is due to the PLT definition which needs to wait for the last object to be downloaded. In most cases, this is inflated by analytics objects (notice the median in the 20 s, a time that would translate in poor QoE for users).

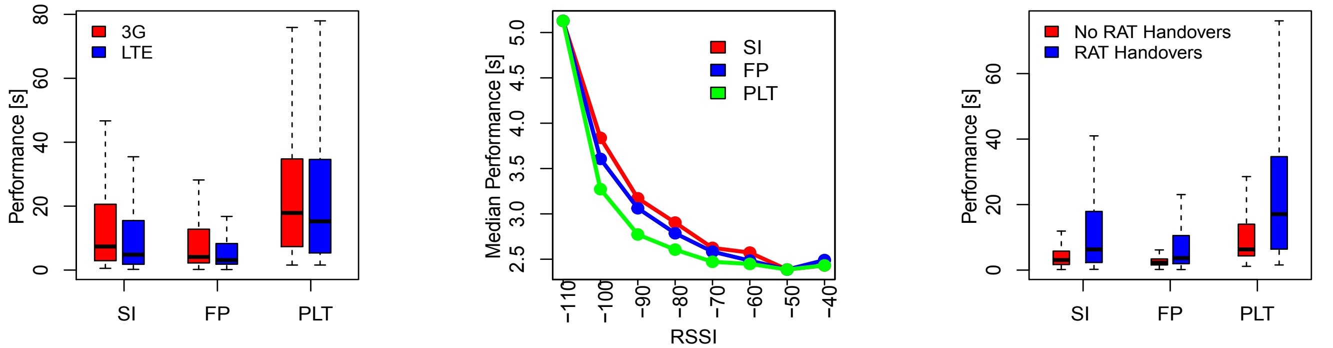
Coverage of MBB networks is also expected to clearly affect performance. This is captured in Figure 4b, where the median performance clearly degrades under poor signal strength conditions. The figure shows results from *Reddit* web experiments performed in Sweden. Interestingly, the RSSI has a non-linear effect, with sudden degradation for RSSI smaller than -80dBm. Above -60dBm, we observe no further improvement, an important observation for network operator's looking for optimal network configuration.

As a last parameter, we showcase the impact of the number of handovers under mobility, reported in Figure 4c. Here we compare experiments with no inter-RAT handover against experiments where at least one inter-RAT (3G to 4G or vice-versa) handover has been observed. Results clearly show the penalty introduced by inter-RAT handovers, with a worsening factor of 2x on the average, and worst case up to 5x slower (when several inter-RAT handovers are suffered).



(a) Browser impact (all). **(b) Browser impact (*ebay*).** **(c) Websites structure impact.**

Figure 3: Impact of experimental context (browser type and target website) on PLT, FP, and SI in Sweden.



(a) RAT impact (all). **(b) Signal Strength Impact (*Reddit*).** **(c) Inter-RAT Handovers Impact.**

Figure 4: Impact of network context PLT, FP, and SI, aggregated over all webpages in Sweden.

Takeaway: Coverage and inter-RAT handovers [15], are among the most prominent causes of web performance degradation. However, their impact is not linear. Faster technologies bring some, but limited benefits. As reported for wired networks [45], for mobile we also observe that the performance gets increasingly latency-bound when network speed increases.

6 WEB PROTOCOL IMPACT

This section provides an in depth quantification of the benefits of using new protocols such as H2 and QUIC compared to the original H1s standard.

6.1 H1s vs H2 Impact

We first compare performance when using H1s and H2. We start by comparing performance considering back-to-back visits to the same page. The first visit using one protocol and the second visit using the other one. Node, operator, browser and location are kept the same. Based on the PLT reported in Figure 2, these two visits can happen in a limited time window (20 urls * 2 protocols * 20 seconds = 800 s ∼ 13.3 minutes) in more than 90% of the experiments. Thus, we pick any two visits that happened within 15 minutes between each other, and directly compare page visit performance. The aim is to provide a sound comparison between the two protocols under as much similar conditions as possible.

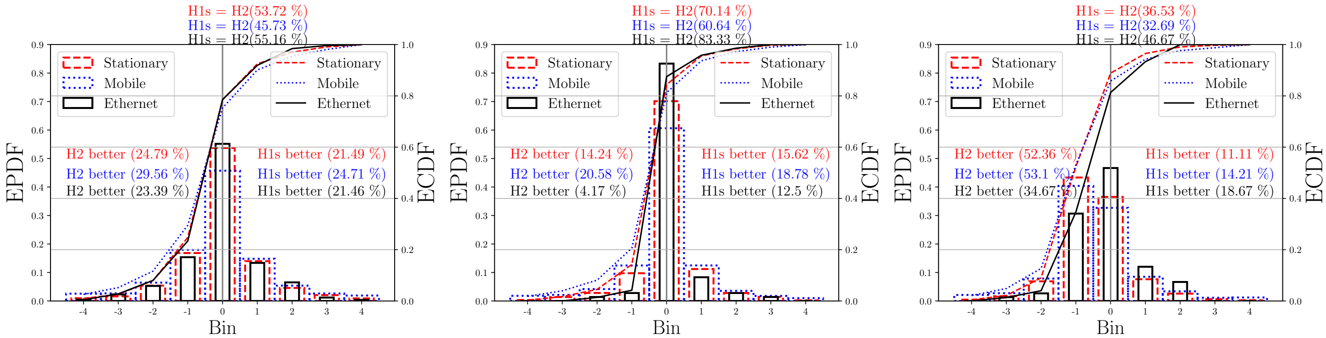
Let $M(i)$ be the i-th visit of metric $M \in \{FP, SI, PLT\}$. We compute the *protocol gain* $\Delta M(i)$ as :

$$\Delta M(i) = \frac{M_{H2}(i) - M_{H1s}(i)}{\max_i \{M_{H2}(i) - M_{H1s}(i)\}}$$

$\Delta M \in [-1, 1]$ – with negative (positive) values when H2 performs better (worse) than H1s on the i-th visit. Next, we compute the distribution of the $\Delta M(i)$ over all experiments, and quantize results into 9 bins centred around zero.[3] The rationale is to identify ranges for which performance is equivalent (center bin), slightly better/worse (the first bin on the left or right), moderately better, much better, extremely better. We opted for 5 bins to mimic the Mean Opinion Score.

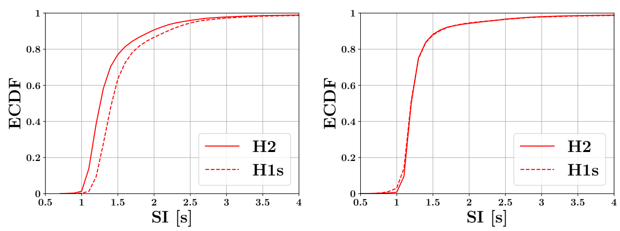
Figure 5a shows the distribution of the protocol gain for SI for Chrome. Left y-axis, right y-axis, and x-axis illustrate the Empirical Probability Density Function (EPDF) of samples in each bin, the

[3] We assign samples to bin in $[-4, -3, -2, -1, 0, 1, 2, 3, 4]$ if $\Delta M(i)$ falls in $[(-1, -0.7), (-0.7, -0.5), (-0.5, -0.3), (-0.3, -0.1), (-0.1, 0.1), (0.1, 0.3), (0.3, 0.5), (0.5, 0.7), (0.7, 1)]$.



(a) All pages **(b) *Wikipedia*** **(c) *Reddit***

Figure 5: Distribution of ΔM for SI with Chrome, for stationary MBB access, mobile MBB access, and wired access.



(a) SI *Reddit* **(b) SI *Wikipedia***

Figure 6: Some example of ECDF for H1s/H2 with Chrome.

ECDF, and bins, respectively. We distinguish between three different cases: mobile nodes with MBB connections (blue), stationary nodes with MBB connections (red) and stationary nodes with wired connection (black). Bars show the EPDF; lines the ECDF. Ideally, when H2 is better than H1s, the probability of having $\Delta M(i) < 0$ would be larger than the probability of having $\Delta M(i) > 0$. We observe that the distributions are (i) having similar modes with zero mean – independently from the scenario, (ii) similar distribution, but (iii) slightly more skewed for MBB networks than for wired access. Specifically, in about 50% of cases, ΔM falls in the central bin (i.e., the relative performance gain of H2 and H1s are within 10%). Moreover, cases where either H2 or H1s have better performance are approximately balanced, although there is a moderate bias towards negative values (H2 has better performance than H1s). Finally, mobility tends to add more variability in the experiment, so that ΔM becomes larger, but equally favoring H1s or H2.

We next drill down our analysis to look separately at each website, to verify if some pages do see some improvement in using H2. Figure 5b is a notable example which reports the protocol gain for *Wikipedia* using Chrome. We observe that the differences are perfectly balanced. This is a generic result, which holds true for SI, PLT and FP as well as for Chrome and Firefox. The only case we observe H2 having a sizeable protocol gain is the case of *R*eddit, detailed in Figure 5c while using Chrome. In this case, we observe that H2 outperforms H1s in 52% of tests. To illustrate the actual absolute improvement as previously done in Figure 1, Figure 6 details the ECDF of all experiments involving these 2 websites for the SI metric. Figures confirm the results shown by the protocol gain: *Reddit* does benefit of H2, while *Wikipedia* obtains no gain.

All measurements presented so far offer a qualitative measure of the impact of protocols. Next, we apply statistical analysis techniques to compare measurements referring to different datasets to precisely quantify the performance difference between the two protocols. Our goal is to answer how (dis)similar are the distributions of the H1s and H2 datasets.

In the literature, there are different well-known Statistical Distance Metrics (SDM), each with its own properties and limitations. In this work, we chose the Jensen-Shannon divergence (JS_{div}), which is defined as:

$$JS_{div} = \sum_i \left\{ \frac{1}{2} p_i \ln\left(\frac{p_i}{\frac{1}{2}p_i + \frac{1}{2}q_i} \right) + \frac{1}{2} q_i \ln\left(\frac{q_i}{\frac{1}{2}q_i + \frac{1}{2}p_i} \right) \right\}$$

where p_i and q_i (relating respectively to H1s and H2) are the EPDF values generated by samples falling in the i-th bin. JS_{div} is a statistical measure based on the Kullback-Leibler divergence. JS_{div} adds symmetry (i.e., $JS_{div}(p,q) = JS_{div}(q,p)$), and bounded image, to the Kullback-Leibler divergence. In fact, the reason we chose the JS_{div} is to obtain a symmetric bounded value for our comparisons. JS_{div} is equal to 0 if $p = q$, while it reaches ln(2) for two completely disjoint distributions.

Intuitively, when $JS_{div} > Th^+$, the difference between the two EPDFs (populations) is significant. Conversely, differences are negligible if $JS_{div} < Th^-$ and when $JS_{div} \in [Th^-, Th^+)$ the difference is observable. We select the threshold values $Th^- = 0.02$ and $Th^+ = 0.1$ as common in statistics. For a more in depth discussion of which SDM to use, and the proper usage of thresholds we defer the reader to [20]. To estimate the EPDF for H1s (p) and H2 (q) separately, we consider stationary nodes only. We use a bin size of 100 milliseconds, and we compute the frequency p_i, q_i of samples falling in each bin. Then, we compute the JS_{div}. Table 3 reports the results for three websites for all browsers and metrics.[4] We observe that *Reddit* for the Chrome browser is the only case where $JS_{div} = 0.124 > Th^+$ for FP, hence falling in the range of a significant difference. The difference for PLT and SI are less evident, but still sizeable. *Wikipedia* shows instead no statistically significant changes – as discussed before.

We further carried out analysis to statistically compare performance of H2 and H1s by conditioning on operators, independent from the website and other conditioning variables. In all cases, we

[4]We explored all websites. Results are omitted for the sake of brevity.

Table 3: Jensen-Shannon divergence of FP, PLT and SI comparing H1s vs H2 for three websites.

Website	FP Chrome/FFox	PLT Chrome/FFox	SI Chrome/FFox
ebay	0.006/0.004	0.009/0.006	0.009/0.005
reddit	0.124/0.002	0.025/0.007	0.079/0.002
wikipedia	0.008/0.003	0.006/0.005	0.009/0.003

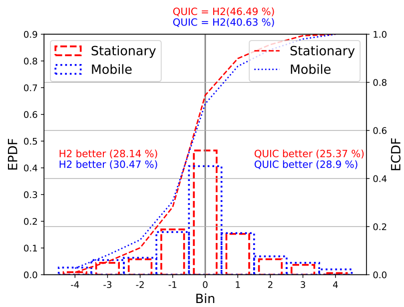


Figure 7: QUIC vs. H2: Distribution of ΔM for SI.

did not observe any statistically significant differences in the performance observed when using H1s or H2. These results are not presented here due to space considerations.

Takeaway: The benefit for H2 is very limited or absent in our tests. One reason that has been identified could be that some of the key H2 features are still not in use at large in the server side [63]. Similarly, results from the work in [61] show that the current prioritization in off-the-shelf browsers is not yet optimized for H2. Also, we have observed that domain wise resource placement is quite similar for both the H1s and H2 cases. This sharding influences the potential gain from multiplexing in H2. Similar observation is mentioned in [36]. Furthermore, for most of the webpages in our tests, we have noticed that a considerable number of resources were downloaded using H1s even when the browser used H2. This clearly limits benefit of H2. At last, MBB access and mobility scenarios exhibit a very high variability in performance – due to the more random nature and complexity of MBB scenarios. This randomness dominates over the benefit of H2.

6.2 QUIC vs H2 Performance

Here, we report the performance comparison between QUIC and H2. We follow the same process as in the previous section to compute the protocol gain. Here we use the small set of websites that support QUIC (Section 3.1). Accordingly, we examine H2 and QUIC back-to-back measurements towards the same website and compute the protocol gain similar to Section 6.1. As before, $\Delta M(i) < 0$, indicate H2 is better than QUIC.

Figure 7 presents the performance of QUIC and H2 we capture with SI using Chrome over all websites. Here we show two scenarios, stationary nodes with MBB connections (red) and mobile nodes with MBB connections (blue). From the distribution, it is clear that QUIC and H2 performance are more similar than different. Again we see an increase in the variability of results for mobile nodes, but still with similar relative performance gains for the two protocols. Although not displayed, the same trends are also seen for FP and PLT. When the performances differ, we see H2 being better slightly more often than QUIC for both stationary and mobile nodes. We observe that most QUIC websites include external resources where the browser needs to fall back to H1s/H2. This may introduce extra delays and limits the protocol gain QUIC can deliver. Such scenarios was never considered in previous works that used synthetic pages and controlled server in existing work [17, 21, 33, 41]. This shows more prominent (but artificial) gain using QUIC.

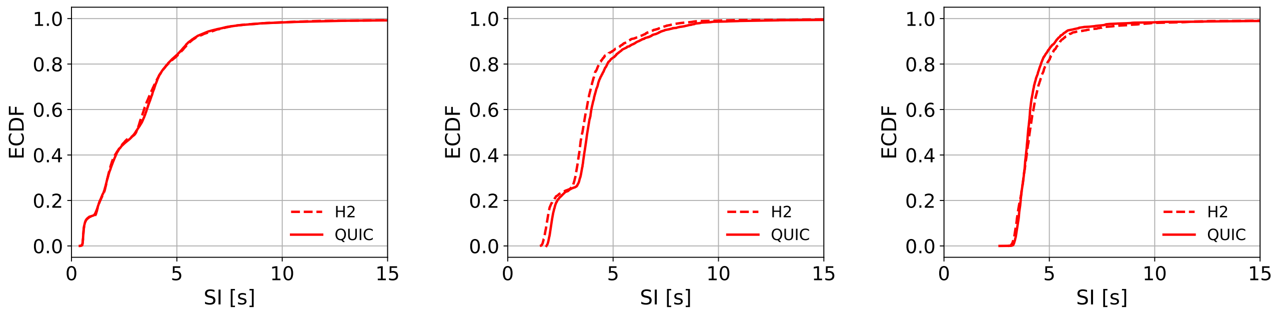
We offer a closer look. In Figure 8a, we illustrate the performance comparison between QUIC and H2 using an ECDF of all experiments for SI. Again, we do not observe any significant difference. This observation holds true for FP and PLT as well. We only observe some notable differences for two websites: *Free-power-point-template* where H2 performs slightly better (Figure 8b) and *Youtube* where QUIC shows slightly better performance (Figure 8c). In the former case, we observe more heavier resources which hurt QUIC performance due to its use of pacing [41].

Takeway: We observe even less benefits for using QUIC instead of H2. Our detailed analysis of QUIC traces show that quite a few objects from the QUIC compatible sites are still not transported using the QUIC protocol, which we believe a hurdle to put QUIC and H2/H1s in a common performance scale. Again, many factors also influence and bring variability into mobile Web browsing performance, overshadowing the impact of the protocols and leading to mixed performance results.

7 MODELLING WEB PERFORMANCE

In our analysis so far, we explored a large dataset of mobile web measurements and investigated different factors that impact the mobile web performance by considering three main QoE metrics. Our statistical analysis, however, partially captures how the collection of features characterizing the web browsing process actually impacts the quality of the user session. Even when using a platform specifically meant to enable comparison and eliminate biases such as location or device, we show that the complexity of the experiments is high and multiple factors can influence the results and the experience of the end-user. In this section, we aim to use data analytics, more specifically regression models, to formally capture the correlation between the set of features that characterize the browsing context of the user (e.g., radio signal quality, radio access technology, mobility etc.) with the quality of experience for that same user (PLT, SI or FP).

We first use *multiple-linear regression (LM)* to investigate how the web QoE metrics (e.g., PLT, FP and SI) vary with respect to different features that characterize the context in which the user surfs the web. We include the following features in our analysis: the node type mobile/stationary (Node Type), number of vertical handovers occurring throughout the duration of one website visit (RAT Handovers), the number of objects downloaded for a specific website under a given experimental configuration (# of Objects), the distance travelled by the node during one website visit (Distance), values of radio strength parameters at the beginning of the experiment to one website (RSRP, RSRQ, RSSI), average value of round-trip-time we measured against the target website during one visit (Avg. RTT), the radio access technology type (RAT), the browser type (Browser), the web protocol version (Protocol). To identify the most prominent features, in the linear regression model



(a) Over all sites **(b) SI *Free-power-point-template*** **(c) SI *Youtube***

Figure 8: Some examples of ECDF showing protocol gain for QUIC/H2 with Chrome.

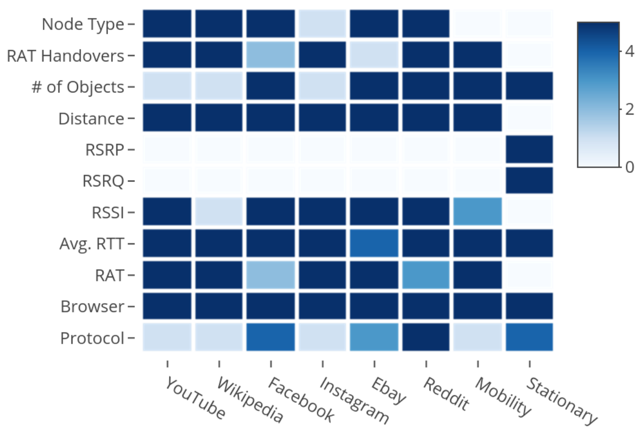


Figure 9: Significance of features for SI captured by a linear modelling approach.

we use standard step-wise sub-scheme (as a Wrapper method) and subsequently apply the filter method to retain only the most significant attributes by looking at their P values (i.e., P values ≤ 0.05). The P values indicate how confident we can be that a feature has some correlation with the web QoE metrics.

Figure 9 demonstrates the significance of these context features on web performance, by focusing on the SI as our web QoE parameter. We capture the significance of each variable in the final model by the color gradient of each cell. We first produce a model using only the samples we collect under mobility scenarios ("Mobility" model, label on the x axis) and another using the samples that we collect from stationary nodes which are always connected to the LTE network and do not experience any inter-RAT handovers ("Stationary" model, label on the x axis).

For the Mobility model, we note the strong significance of the initial RAT, inter-RAT handovers and the distance traveled, confirming our observations in Section 5 as well as the previous work that highlights the strong impact of mobility on the end-user QoE [13]. At the same time, the signal strength value (RSSI) is not as significant as the mobility context, and the protocols is the least significant from the set of features we consider. In the Stationary model, we note that the features we consider including the number of objects for the target webpage, RSRP, RSRQ, browser type and even the protocol are significantly correlated with the end-user experience. These results are consistent with our findings in Section 5, except for the protocols. Regarding the protocols, although in Section 6 we concluded that there are no significant differences among protocols (they do not provide gains more than 10%), our model still considers protocol version as a significant feature indicating that the model tries to capture even the marginal gains.

Then, we generate separate models for six of the target websites in Sweden (namely, *Youtube*, *Wikipedia*, *Facebook*, *Instagram*, *Ebay*, *Reddit*), by merging all the data from mobility/stationary conditions, operators, protocols or browser type. We break down our analysis per target webpage and note in Figure 9 the significance each feature has on the SI. Overall, we observe that the browser type and the mobility context (more specifically, the distance travelled during browsing) are consistently influencing the end-user experience, regardless the target webpage. The number of objects downloaded is not as significant, except for webpages that are potentially more dynamic over time (such as *Facebook* or *Ebay*). In all cases the signal strength value is highly correlated with the end-user experience, except for *Wikipedia*, which is potentially due to the smaller size of the webpage (which contains mostly text).

Takeaway: We observe that web performance under mobility is heavily impacted by RAT and inter-RAT handovers. For the stationary case, on the other hand, parameters such as browser and protocols type, number of objects downloaded in a webpage and also signal strength values play an important role for the web performance. Overall, we observe that the browser type and the mobility context are consistently influencing the end-user experience, regardless the target webpage. The number of objects downloaded is not as significant, except for webpages that are potentially more dynamic over time.

8 CONCLUSION

This paper presented a cross-European study of web performance on commercial mobile carriers using the MONROE open measurement infrastructure. The novelty of the study stands in the sheer volume of data we were able to collect, nearly two million page visits from 11 different operational MBB networks in 4 countries, and in the rich context information captured along with the measurements that allowed us to analyze the complex interactions that influence mobile web user experience. Our results show a performance penalty of over 30% for mobile broadband access as compared to our wired baseline, with further performance drops for mobile users due to inter-technology handovers and varying channel conditions. Regrettably, except for few individual webpages, our measurements show that the performance improvements promised by newly emerging web protocols H2 and QUIC still remain to be experienced. Majority of back-to-back visits with H1s and H2 web protocols show similar performance with equal split of gains by H1s or H2 for the rest. Our results were confirmed by data analytics capturing the impact of single features that characterize the context in which browsing occurs. The presented measurements were designed to be scientifically sound and with reproducibility in mind and we offer the captured dataset to the community for other researchers to verify and build upon.

REFERENCES

[1] [n. d.]. A Primer for Web Performance Timing APIs. https://w3c.github.io/perf-timing-primer/. [Online; accessed 07-Aug-2018].

[2] [n. d.]. Above-the-fold time (AFT): Useful, but not yet a substitute for user-centric analysis. https://goo.gl/gWmj3r. [Online; accessed 07-Aug-2018].

[3] [n. d.]. Different SpeedIndex Measurements. https://bit.ly/2I87znK. [Online; accessed 07-Feb-2019].

[4] [n. d.]. MONROE-Browsertime. https://github.com/MONROE-PROJECT/Experiments/. [Online; accessed 07-Aug-2018].

[5] [n. d.]. A simple, client-side approach to measuring Speed Index. https://deanhume.com/a-simple-client-side-approach-to-measuring-speed-index/. [Online; accessed 07-Feb-2019].

[6] [n. d.]. Speed Index - how it works and what it means. https://www.nccgroup.trust/uk/about-us/newsroom-and-events/blogs/2015/june/speed-index--how-it-works-and-what-it-means/. [Online; accessed 07-Aug-2018].

[7] [n. d.]. SpeedIndex measurements from the field. https://github.com/WPO-Foundation/RUM-SpeedIndex. [Online; accessed 07-Aug-2018].

[8] [n. d.]. Webpage Test. https://www.webpagetest.org. [Online; accessed 07-Aug-2018].

[9] [n. d.]. WebPagetest documentation. https://sites.google.com/a/webpagetest.org/docs/using-webpagetest/metrics/speed-index. [Online; accessed 07-Aug-2018].

[10] [n. d.]. Xvfb manul page. https://goo.gl/EPBtpt. [Online; accessed 07-Aug-2018].

[11] Ö. Alay, A. Lutu, M. Peón-Quirós, V. Mancuso, T. Hirsch, K. Evensen, A. Hansen, S. Alfredsson, J. Karlsson, A. Brunstrom, A. Safari Khatouni, M. Mellia, and M. Ajmone Marsan. 2017. Experience: An Open Platform for Experimentation with Commercial Mobile Broadband Networks. *Proc. ACM MobiCom '17* (2017), 70–78. https://doi.org/10.1145/3117811.3117812

[12] Alexa. 2017. The top sites on the web. http://www.alexa.com/topsites

[13] Athula Balachandran, Vaneet Aggarwal, Emir Halepovic, Jeffrey Pang, Srinivasan Seshan, Shobha Venkataraman, and He Yan. 2014. Modeling web-quality of experience on cellular networks. In *Proceedings of the 20th annual international conference on Mobile computing and networking, MobiCom'14*. ACM, 213–224.

[14] Džiugas Baltrūnas, Ahmed Elmokashfi, and Amund Kvalbein. 2014. Measuring the Reliability of Mobile Broadband Networks. In *Proc. of IMC*.

[15] D. Baltrunas, A. Elmokashfi, A. Kvalbein, and Ö. Alay. 2016. Investigating packet loss in mobile broadband networks under mobility. In *2016 IFIP Networking Conference (IFIP Networking) and Workshops*. 225–233. https://doi.org/10.1109/IFIPNetworking.2016.7497225

[16] M. Belshe, R. Peon, and M. Thomson. 2015. Hypertext Transfer Protocol Version 2 (HTTP/2). RFC 7540 (Proposed Standard).

[17] Prasenjeet Biswal and Omprakash Gnawali. 2016. Does quic make the web faster?. In *Global Communications Conference (GLOBECOM), 2016 IEEE*. IEEE.

[18] Enrico Bocchi, Luca De Cicco, Marco Mellia, and Dario Rossi. 2017. The Web, the Users, and the MOS: Influence of HTTP/2 on User Experience. In *Passive and Active Measurement - 18th International Conference, PAM*. 47–59. https://doi.org/10.1007/978-3-319-54328-4_4

[19] Enrico Bocchi, Luca De Cicco, and Dario Rossi. 2016. Measuring the Quality of Experience of Web users. *ACM SIGCOMM Computer Communication Review* 46, 4 (2016), 8–13.

[20] Enrico Bocchi, Ali Safari Khatouni, Stefano Traverso, Alessandro Finamore, Maurizio Munafo, Marco Mellia, and Dario Rossi. 2016. Statistical network monitoring: Methodology and application to carrier-grade NAT. *Computer Networks* 107 (2016), 20 – 35. https://doi.org/10.1016/j.comnet.2016.06.018 Machine learning, data mining and Big Data frameworks for network monitoring and troubleshooting.

[21] Gaetano Carlucci, Luca De Cicco, and Saverio Mascolo. 2015. HTTP over UDP: an Experimental Investigation of QUIC. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*. ACM, 609–614.

[22] Mallesham Dasari, Santiago Vargas, Arani Bhattacharya, Aruna Balasubramanian, Samir Das, and Mike Ferdman. 2018. Impact of Device Parameters on Internet-based Mobile Applications. In *Proceedings of the 2018 Conference on Internet Measurement Conference (IMC) (IMC '18)*. To Appear.

[23] H. de Saxcé, I. Oprescu, and Y. Chen. 2015. Is HTTP/2 really faster than HTTP/1.1?. In *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 293–299. https://doi.org/10.1109/INFCOMW.2015.7179400

[24] Prasenjit Dey, Parvez Ahammad, et al. 2017. Perceived Performance of Top Retail Webpages In the Wild. *ACM SIGCOMM Computer Communication Review* 47, 5 (2017), 42–47.

[25] Yehia Elkhatib, Gareth Tyson, and Michael Welzl. 2014. Can SPDY really make the web faster?. In *Networking Conference, 2014 IFIP*. IEEE.

[26] Jeffrey Erman, Vijay Gopalakrishnan, Rittwik Jana, and Kadangode K Ramakrishnan. 2015. Towards a SPDY'er mobile web? *IEEE/ACM Transactions on Networking* 23, 6 (2015), 2010–2023.

[27] R. Fielding et al. 1999. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616 (Draft Standard).

[28] Utkarsh Goel, Moritz Steiner, Mike P Wittie, Martin Flack, and Stephen Ludin. [n. d.]. Measuring What is Not Ours: A Tale of 3rd Party Performance. In *ACM Passive and Active Measurements Conference (PAM)*. Springer, 142–155.

[29] Utkarsh Goel, Moritz Steiner, Mike P Wittie, Martin Flack, and Stephen Ludin. 2016. HTTP/2 Performance in Cellular Networks. In *ACM MobiCom*.

[30] E. Halepovic, J. Pang, and O. Spatscheck. 2012. Can you GET me now? Estimating the time-to-first-byte of HTTP transactions with passive measurements. In *Proc. of IMC*.

[31] Junxian Huang, Feng Qian, Yihua Guo, Yuanyuan Zhou, Qiang Xu, Z. Morley Mao, Subhabrata Sen, and Oliver Spatscheck. 2013. An In-depth Study of LTE: Effect of Network Protocol and Application Behavior on Performance. In *Proc. of SIGCOMM*.

[32] Google Inc. [n. d.]. SPDY: An experimental protocol for a faster web. https://www.chromium.org/spdy/spdy-whitepaper. [Online; accessed 07-Aug-2018].

[33] Arash Molavi Kakhki, Samuel Jero, David Choffnes, Cristina Nita-Rotaru, and Alan Mislove. 2017. Taking a Long Look at QUIC: An Approach for Rigorous Evaluation of Rapidly Evolving Transport Protocols. In *Proceedings of the 2017 Internet Measurement Conference (IMC '17)*. ACM, New York, NY, USA, 290–303. https://doi.org/10.1145/3131365.3131368

[34] Conor Kelton, Jihoon Ryoo, Aruna Balasubramanian, and Samir R Das. 2017. Improving User Perceived Page Load Times Using Gaze.. In *NSDI*. 545–559.

[35] Ali Safari Khatouni, Marco Mellia, Marco Ajmone Marsan, Stefan Alfredsson, Jonas Karlsson, Anna Brunstrom, Ozgu Alay, Andra Lutu, Cise Midoglu, and Vincenzo Mancuso. 2017. Speedtest-like Measurements in 3G/4G Networks: The MONROE Experience. In *Teletraffic Congress (ITC 29), 2017 29th International*, Vol. 1. IEEE, 169–177.

[36] H. Kim, J. Lee, I. Park, H. Kim, D. Yi, and T. Hur. 2015. The upcoming new standard HTTP/2 and its impact on multi-domain websites. In *2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. 530–533. https://doi.org/10.1109/APNOMS.2015.7275406

[37] Zahir Koradia, Goutham Mannava, Aravindh Raman, Gaurav Aggarwal, Vinay Ribeiro, Aaditeshwar Seth, Sebastian Ardon, Anirban Mahanti, and Sipat Triukose. 2013. First Impressions on the State of Cellular Data Connectivity in India. In *Procs. of ACM DEV-4 (ACM DEV-4 '13)*.

[38] Adam Langley, Alistair Riddoch, Alyssa Wilk, Antonio Vicente, Charles Krasic, Dan Zhang, Fan Yang, Fedor Kouranov, Ian Swett, Janardhan Iyengar, et al. 2017. The QUIC transport protocol: Design and Internet-scale deployment. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. ACM, 183–196.

[39] Y. Liu, Y. Ma, X. Liu, and G. Huang. 2016. Can HTTP/2 Really Help Web Performance on Smartphones?. In *2016 IEEE International Conference on Services Computing (SCC)*. 219–226. https://doi.org/10.1109/SCC.2016.36

[40] Patrick Meenan. 2013. How fast is your website? *Commun. ACM* 56, 4 (2013), 49–55.

[41] Péter Megyesi, Zsolt Krämer, and Sándor Molnár. 2016. How quick is QUIC?. In *Communications (ICC), 2016 IEEE International Conference on*. IEEE.

[42] Javad Nejati and Aruna Balasubramanian. 2016. An in-depth study of mobile browser performance. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1305–1315.

[43] Ashkan Nikravesh, Hongyi Yao, Shichang Xu, David Choffnes, and Z Morley Mao. 2015. Mobilyzer: An open platform for controllable mobile network measurements. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 389–404.

[44] Naoki Oda and Saneyasu Yamaguchi. 2018. HTTP/2 performance evaluation with latency and packet losses. In *Consumer Communications & Networking Conference (CCNC), 2018 15th IEEE Annual*. IEEE.

[45] Mohammad Rajiullah. 2015. *Towards a Low Latency Internet: Understanding and Solutions*. Ph.D. Dissertation. Karlstad University.

[46] Mohammad Rajiullah, AC Mohideen, Felix Weinrank, R Secchi, Gorry Fairhurst, and A Brunstrom. 2017. Understanding multistreaming for web traffic: An experimental study. In *IFIP Networking Conference (IFIP Networking) and Workshops, 2017*. IEEE.

[47] E. Rescorla. 2000. HTTP Over TLS. RFC 2818 (Informational). http://www.ietf.org/rfc/rfc2818.txt Updated by RFCs 5785, 7230.

[48] John P. Rula, Vishnu Navda, Fabián Bustamante, Ranjita Bhagwan, and Saikat Guha. 2014. "No One-Size Fits All": Towards a principled approach for incentives in mobile crowdsourcing. In *Proc. of IMC*.

[49] S. Sen, J. Yoon, J. Hare, J. Ormont, and S.Banerjee. 2011. Can they hear me now?: A case for a client-assisted approach to monitoring wide-area wireless networks. In *Proc. of IMC*.

[50] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, S. Venkataraman, and J. Wang. 2013. A first look at cellular network performance during crowded events. In *Proc. of SIGMETRICS*.

[51] M. Zubair Shafiq, Lusheng Ji, Alex X. Liu, and Jia Wang. 2011. Characterizing and Modeling Internet Traffic Dynamics of Cellular Devices. In *Proc. of SIGMETRICS*.

[52] Sierra-Wireless. [n. d.]. MC7455 miniPCI express (USB 3.0) modem: https://www.sierrawireless.com/products-and-solutions/embedded-solutions/products/mc7455/. ([n. d.]).

[53] Joel Sommers and Paul Barford. 2012. Cell vs. WiFi: on the performance of metro area mobile connections. In *Proceedings of the 2012 ACM conference on Internet measurement conference*. ACM, 301–314.

[54] Steve Souders. [n. d.]. Moving beyond window.onload(). https://goo.gl/rAeFNR. [Online; accessed 07-Aug-2018].

[55] Daniel Stenberg. 2014. HTTP2 Explained. *SIGCOMM Computer Communication Review* 44, 3 (July 2014), 120–128. https://doi.org/10.1145/2656877.2656896

[56] Matteo Varvello, Kyle Schomp, David Naylor, Jeremy Blackburn, Alessandro Finamore, and Kostantina Papagiannaki. 2015. To HTTP/2, or not to HTTP/2, that is the question. *arXiv preprint arXiv:1507.06562* (2015).

[57] Matteo Varvello, Kyle Schomp, David Naylor, Jeremy Blackburn, Alessandro Finamore, and Konstantina Papagiannaki. 2016. Is the Web HTTP/2 Yet?. In *Passive and Active Measurement - 17th International Conference, PAM*. 218–232. https://doi.org/10.1007/978-3-319-30505-9_17

[58] J Vihervaara and T Alapaholuoma. 2018. The impact of HTTP/2 on the service efficiency of e-commerce websites. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE.

[59] Xiao Sophia Wang, Aruna Balasubramanian, Arvind Krishnamurthy, and David Wetherall. 2014. How Speedy is SPDY?. In *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*. USENIX Association, Seattle, WA, 387–399. https://www.usenix.org/conference/nsdi14/technical-sessions/wang

[60] Greg White, Jean-François Mulé, and Dan Rice. 2012. Analysis of Google SPDY and TCP initcwnd. *Cable Television Laboratories, Inc* (2012).

[61] Maarten Wijnants, Robin Marx, Peter Quax, and Wim Lamotte. 2018. HTTP/2 Prioritization and Its Impact on Web Performance. In *Proceedings of the 2018 World Wide Web Conference (WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 1755–1764. https://doi.org/10.1145/3178876.3186181

[62] Kyriakos Zarifis, Mark Holland, Manish Jain, Ethan Katz-Bassett, and Ramesh Govindan. 2016. *Modeling HTTP/2 Speed from HTTP/1 Traces*. Springer International Publishing, Cham, 233–247. https://doi.org/10.1007/978-3-319-30505-9_18

[63] Torsten Zimmermann, Jan Rüth, Benedikt Wolters, and Oliver Hohlfeld. 2017. How HTTP/2 Pushes the Web: An Empirical Study of HTTP/2 Server Push. In *2017 IFIP Networking Conference (IFIP Networking) and Workshops*.