

EYEORG: A Platform For Crowdsourcing Web Quality Of Experience Measurements

<https://eyeorg.net/>

Matteo Varvello[†], Jeremy Blackburn[†], David Naylor[•], Kostantina Papagiannaki^{‡*}
[†]Telefónica Research, [•]Carnegie Mellon University, [‡]Google Inc.

ABSTRACT

Tremendous effort has gone into the ongoing battle to make webpages load faster. This effort has culminated in new protocols (QUIC, SPDY, and HTTP/2) as well as novel content delivery mechanisms. In addition, companies like Google and SpeedCurve investigated how to measure “page load time” (PLT) in a way that captures *human perception*. In this paper we present Eyeorg [12], a platform for crowdsourcing web quality of experience measurements. Eyeorg overcomes the scaling and automation challenges of recruiting users and collecting consistent user-perceived quality measurements. We validate Eyeorg’s capabilities via a set of 100 *trusted* participants. Next, we showcase its functionalities via three measurement campaigns, each involving 1,000 *paid* participants, to 1) study the quality of several PLT metrics, 2) compare HTTP/1.1 and HTTP/2 performance, and 3) assess the impact of online advertisements and ad blockers on user experience. We find that commonly used, and even novel and sophisticated PLT metrics fail to represent actual human perception of PLT, that the performance gains from HTTP/2 are imperceptible in some circumstances, and that not all ad blockers are created equal.

CCS Concepts

•**Human-centered computing** → **Web-based interaction**;
•**Networks** → *Application layer protocols; Network measurement*;

Keywords

Web measurements; Crowdsourcing; Quality of Experience; HTTP/2; Adblockers

*Work done while K. Papagiannaki was with Telefónica Research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CoNEXT '16, December 12-15, 2016, Irvine, CA, USA

© 2016 ACM. ISBN 978-1-4503-4292-6/16/12...\$15.00

DOI: <http://dx.doi.org/10.1145/2999572.2999590>

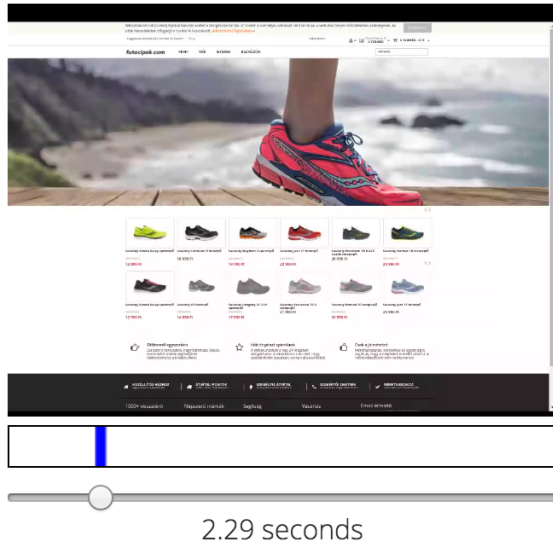
1. INTRODUCTION

Improving Quality of Experience (QoE) on the Web remains a hot topic. One obvious aspect of QoE, page load time (PLT), heavily impacts revenue for web-based companies [2]. It is no surprise, then, that there are many recent efforts to decrease PLT from both industry (e.g., QUIC, SPDY, and HTTP/2) and academia [10, 17, 24]. To evaluate the effectiveness of such techniques, we need quantitative measures of their impact on QoE. Unfortunately, this is difficult because QoE is hard to define and even harder to measure.

Perhaps surprisingly, this is true even for PLT, which seems straightforward to define and measure. The typical PLT metric, the browser’s “onload” event, denotes when “all of the objects in the document are in the DOM, and all the images, scripts, links and sub-frames have finished loading” [1]. On-Load is not a perfect proxy for *user-perceived page load time* (UserPerceivedPLT) for two reasons: first, the user may only care how quickly the *visible* portion of the content loads (“above-the-fold” render time), meaning OnLoad might be an overestimate. Second, scripts might continue loading objects after OnLoad fires, in which case it may be an underestimate. To address this problem, new PLT metrics attempt to bridge the gap between OnLoad and UserPerceivedPLT. For example, SpeedIndex¹ defines PLT as the average time at which above-the-fold content is displayed. Unfortunately, even this is not perfect: the user may consider the page ready to interact with even before some above-the-fold content (e.g., ads) loads.

Ideally, we want a direct assessment of a user’s experience on a webpage. This is difficult for the following reasons. First, a large number of people need to be recruited and surveyed. Second, participants need to be shown consistent views of a page loading. Third, we need to design experiments that lead to *quantitative* results about a technique’s impact on QoE. In some studies [5, 6], participants are shown a page load in person in a controlled environment and give qualitative feedback to interviewers. Clearly, this approach neither scales nor yields quantitative results.

¹<https://sites.google.com/a/webpagetest.org/docs/using-webpagetest/metrics/speed-index>



(a) Timeline of UserPerceivedPLT responses.



(b) Some sites exhibit multiple modes; here, some participants consider the site “ready” before the ads load.

Figure 1: Exploring responses with Eyeorg’s visualization tools.

We take a different approach with Eyeorg, a system for crowdsourcing Web QoE measurements. Eyeorg allows researchers to test the impact of changes to how a page is structured or delivered. Eyeorg uses crowdsourced participants to scale (we currently integrate with two popular services, Microworkers and CrowdFlower); it shows *videos* of pages loading to provide a consistent experience to all participants, regardless of their network connections and device configurations; and its surveys are designed to provide quantifiable feedback (e.g., participants are asked to pick the point on a timeline when they consider a page loaded). With this approach, we maintain full control of experimental conditions and we can recruit any participant with a modern web browser, without requiring special hardware or software.

We present two kinds of results. First, we validate that Eyeorg can produce good quality results; to do so, we recruit 100 *trusted* participants and compare the quality of their answers with those of 100 *paid* participants. We find that we can reliably identify unreliable participants, flagging about 20% of the participants in our experiments as “low

performers” whose results should be discarded. The speed with which crowdsourced participants can be recruited outweighs this overhead (e.g., 1 hour rather than 10 days in this specific experiment).

Next, we use Eyeorg to 1) study how people perceive PLT, 2) compare HTTP/1.1 and HTTP/2 performance, and 3) compare the QoE impact of three popular ad blockers: AdBlock, Ghostery and uBlock. For each measurement “campaign,” we recruited 1,000 paid participants to which we showed 5,000 page load videos (selected from 100 sites). Each campaign required 1.5 days to hit the 1,000 participants target. Our analysis shows that people find the HTTP/2 version of a website faster and that Ghostery outperforms its competitors. We also show that existing PLT metrics only partially match participants responses.

To help understand the PLT data, we built a visualization tool that displays the UserPerceivedPLT responses as a timeline next to the video, as shown in Figure 1(a). Using this tool, we uncovered patterns in responses; for example, many videos have two modes, one for participants who consider the pages “ready” when the primary content is in place and one for those who wait for auxiliary content like ads (Figure 1(b)). The user-perceived PLT data we crowdsourced from 1,000 people is available at [12].

2. RELATED WORK

2.1 Latency and Human Perception

Bouch, Kuchinsky, and Bhatti performed one of the earliest in-depth studies to understand user perceptions and web page load time [9]. They approached the problem from an HCI perspective by designing a controlled experiment where a small ($N = 30$) set of users were given an e-commerce related task to perform on the web. They artificially introduced delay for each of the pages that the task required visiting and asked users to respond with a qualitative assessment (“high”, “average”, “low”, and “unacceptable”) of page performance. While they had many interesting findings, the one most relevant to Eyeorg was that a mapping between subjective and objective PLT measurements was possible. Eyeorg takes the next step by greatly increasing the scale of responses as well as being a more general platform for conducting experiments.

Arapakis, Bai, and Cambazoglu investigated the impact of search engine response latency and user behavior in [5]. Although their work was focused primarily on how latency affects user interaction and behavior for search engines, they provide some insight into how user perceived page load times might be measured. In particular, in controlled experiments in a laboratory environment they asked participants to give an estimation of how long a page took to load. They found that although individual responses did not match the controlled ground truth, the average of the estimates was quite close to the real values. Although they used relatively coarse grained latency intervals (250 ms steps), their results indi-

cate that aggregating human responses is a reasonable way to measure UserPerceivedPLT.

Egger et al. [11] investigate the relationship between user perceived PLT and “application” PLT.² To do so, they ask participants to mark the time at which they considered a page loaded. The outcome of their study is that user-perceived PLT largely departs from application PLT: in most cases, users perceived a page as loading substantially faster than the application-reported PLT. Their methodology differs significantly from ours, however. In their study, participants browsed five different pages while link bandwidth and delay were manipulated. In contrast, we show video captures of pages loading to thousands of study participants which might have subtle effects on their responses. For example, our participants are able to precisely pinpoint the frame that they feel a page has loaded, fine-tuning their choice until satisfied. We also measure a binary “which is faster” choice, which was not explored at all in [11].

2.2 Page Load Time

While the research community has focused on understanding the consequences of, and how to improve, page load time, it seems that understanding how to measure it is a mostly overlooked problem. The greater “web community,” instead, has invested a lot into figuring out how to tell how fast their pages are being delivered to users. These efforts range from blog posts discussing how to measure PLT [21], to ebooks explaining how to make your page load faster [22], to repositories of tools for measuring performance [15], to full on commercial measurement offerings [16]. Even though there are serious business consequences associated with page performance, unfortunately, there is little science behind most of these efforts.

Regardless, it is worth exploring the efforts made in a bit more detail. SpeedCurve [16] is a comprehensive suite of tools for web developers to gain a deeper understanding into page performance. They provide instrumentation that extracts some of the metrics we use in this paper in addition to allowing the definition of custom metrics. One interesting feature is that they provide developers a head-to-head benchmark comparing the developer’s site with other, similar sites. Ultimately, however, SpeedCurve blindly PLT metrics without any scientific justification that they are meaningful. Determining the validity of computer-generated metrics is a primary goal of Eyeorg.

In the scholarly community, “onload” has been the de facto PLT metric (e.g., [10, 17, 24])—until recently. Bocchi, De Cicco, and Rossi performed a preliminary study comparing different page load metrics [8]. They focused on evaluating two proposed metrics that are similar to SpeedIndex but are computationally less expensive. Their most salient result is that correlating these metrics with real human perception is incredibly difficult, which is what Eyeorg tries to address.

Nikraves et al. [18] measured web performance from 80 crowdsourced mobile devices using their library Mobilyzer. They found a large degree of variability between PLT and what they call the “page interaction time” (similar to SpeedIndex). The SpeedPerception³ project has adopted a methodology similar to Eyeorg’s; in particular the idea of a human supplied response to a “head-to-head” matchup of page performance. At the time of this writing, there are a few significant implementation differences: 1) SpeedPerception’s A/B tests⁴ ask users to provide a decision on two *different* sites and 2) the way that SpeedPerception implements A/B tests differs from Eyeorg’s (§3.2).

3. EYEORG

Eyeorg [12] is a platform for crowdsourcing web quality of experience measurements; the goal of Eyeorg is to measure improvements to user experience on the web. Both research and industry have expended great effort exploring different techniques for building and delivering web pages. Popular examples are content optimization techniques like “inlining” (putting CSS styles and JavaScript code directly in HTML) and “spriting” (putting collection of images in a single image). More far-fetched solutions are shandian [26], klotski [10], and polaris [17] which aim at optimizing content delivery based on how it gets displayed from a browser.

To understand when and how to use these techniques, we need to be able to *measure their impact*. Doing this in an automated and quantitative way is hard. First, “quality of experience” is amorphous and ill-defined—it is not obvious what to measure, let alone how. Second, even aspects of Web experience that seem straightforward, like PLT, are difficult for machines to measure: ultimately, how *humans* perceive performance is what matters.

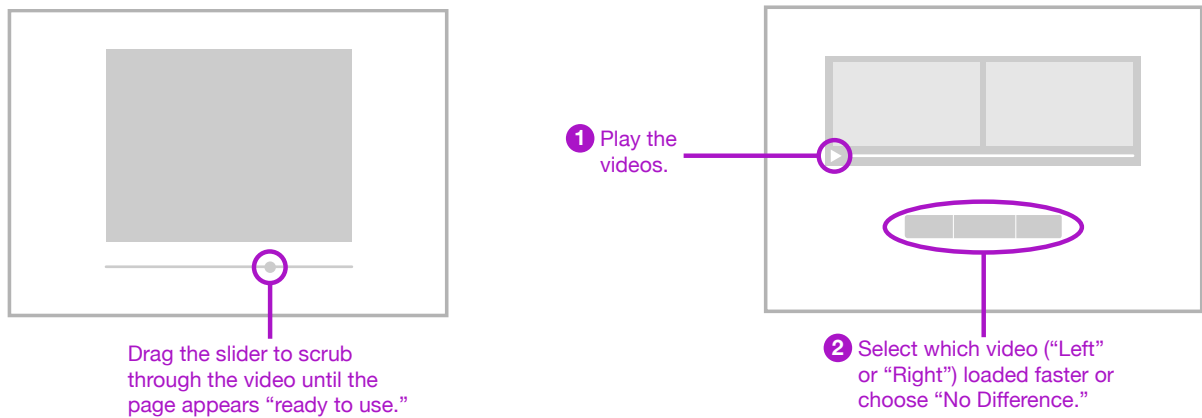
Eyeorg takes a new approach to quantifying quality of experience by allowing experimenters to directly test the impact of web page design or delivery techniques on real users in a controlled fashion at large scale. Experimenters can use Eyeorg to answer questions like, “Does changing the order in which objects load impact UserPerceivedPLT”, “Does the presence of advertisements on a page negatively impact user experience?”, or “Which demographics are more sensitive to PLT speedup?”.

Experimenters can use Eyeorg to draw responses from various classes of users, such as invited participants (e.g., friends or colleagues), paid crowdsourced workers, or general visitors to the Eyeorg site. Eyeorg provides tools to design experiments, recruit users, filter low-quality responses, and visualize results. The rationale behind Eyeorg is that computer generated PLT metrics only partially capture how people perceive the web. The human perception data gathered by Eyeorg can be used to augment and evaluate computer-generated PLT metrics.

²To the best of our knowledge, this is the browser reported “onLoad” time.

³<http://speedperception.meteorapp.com/>

⁴SpeedPerception currently only supports A/B tests.



(a) **Timeline Test.** Participants “scrub” the slider to the point where they consider the page “ready to use.”

(b) **A/B Test.** Participants watch side-by-side page load videos and indicate which load is faster.

Figure 2: Eyeorg’s experiment types.

In the remainder of this section we describe the design and rationale behind Eyeorg. In particular, we discuss how we address the following three challenges:

1. **How do we present page loads to participants? (§3.1)**
Variations in load time caused by a particular participant’s device or network could mask variations due to the technique being tested. Furthermore, experimenters might want to test the impact of protocols or browser extensions that participants’ browsers do not support.
2. **How do we ask questions about quality of experience (and get quantitative answers)? (§3.2)**
For example, how can participants indicate to us when a page “seems loaded” (particularly non-technical participants)?
3. **How do we get lots of trustworthy responses? (§3.3)**
Drawing meaningful conclusions requires a large sample size; at the same time, recruiting participants not invested in the experiments could yield careless, sloppy responses.

3.1 Providing a Controlled Experience

Apart from the engineering challenge to build Eyeorg, a more fundamental challenge lies in dealing with real people. Testing participants’ reactions to a web experience “in the wild” and at scale is difficult for several reasons. First, differences between participants’ browsers could impact results. Second, in many cases the outcome we want to test (like PLT) is dependent on network conditions, but we have no control over the quality of participants’ network access. Third, experimenters may want to test the impact of protocols (e.g., HTTP/2 or SPDY) that some participants’ browsers may not support, or browser extensions (e.g., AdBlockers) that some participants may not have installed. In short, we want to guarantee the same experience for each participant.

To ensure that all participants base their responses on identical experiences, Eyeorg uses *video*. This allows us to fully control what participants see—we pick the browser, network conditions, protocols, and plugins—regardless of individual participants’ configurations.

We built a tool, *webpeg*, to record videos of web pages loading. Pages are loaded with Chrome because: 1) unlike headless browsers like PhantomJS⁵, Chrome provides rapid (and optimized) support for new technologies like SPDY and HTTP/2 and 2) Chrome offers better support for instrumentation than Firefox or Safari. We use Xvfb⁶ (the X virtual frame buffer) so we can run *webpeg* on machines without displays (e.g., EC2 instances) and we capture videos in the webm format⁷ (which offers small file sizes) using ffmpeg.⁸

We designed *webpeg* to be highly customizable. We use Chrome’s command line options to control things like protocol (HTTP/1.1 or HTTP/2) and appearance (kiosk mode) and Chrome’s remote debugging protocol⁹ to enable device and network emulation. The remote debugging interface also gives us detailed information about the page load (as an HTTP Archive, or HAR), including when each object loaded, which protocol was used, and when the onload event fired. An alternative to using Chrome’s remote debugging interface would have been to use the Navigation Timing API.¹⁰ However, the Navigation Timing API is designed to be accessed by web “applications” themselves via JavaScript. To avoid *any* impact our methodology might have on performance, we thus chose to use the asynchronous debugging protocol. In the future, we will explore performant uses of the NT API when expanding Eyeorg to other browsers. Fi-

⁵<http://phantomjs.org/>

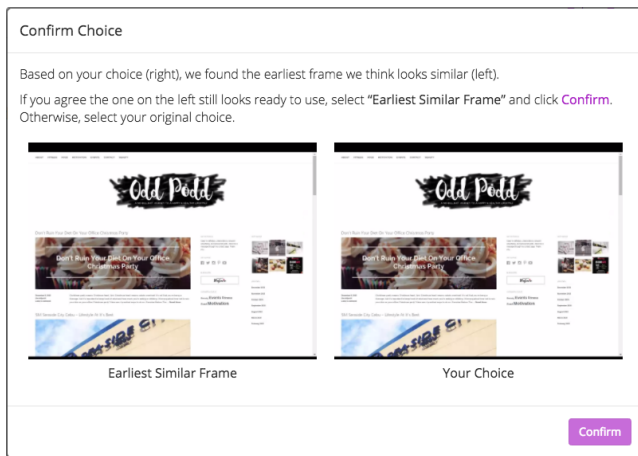
⁶<https://www.x.org/archive/X11R7.7/doc/man/man1/Xvfb.1.xhtml>

⁷<https://www.webmproject.org/>

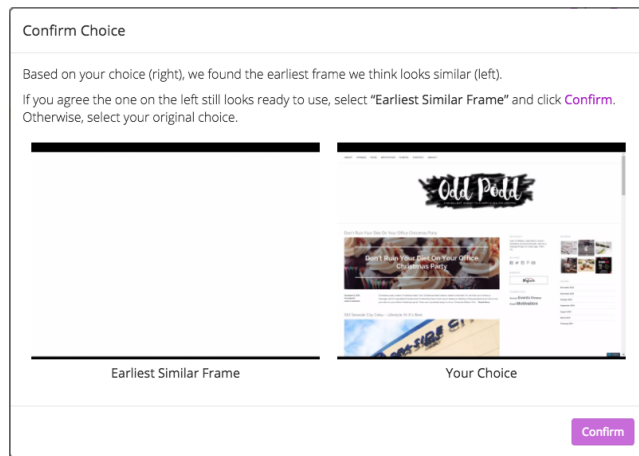
⁸<https://ffmpeg.org/>

⁹<https://developer.chrome.com/devtools/docs/debugger-protocol>

¹⁰https://developer.mozilla.org/en-US/docs/Web/API/Navigation_timing_API



(a) Standard frame selection helper.



(b) Control frame selection helper.

Figure 3: Frame selection helper. Eyeorg helps participants adjust their selected PLT by giving them the option to choose the earliest visually similar frame to the one they chose (left). To ensure participants do not blindly accept the “rewind” frame, as a control we occasionally show a drastically different frame (right).

nally, webpeg directly modifies Chrome’s preference file to enable/disable extensions and turn off distracting messages like “Would you like to translate this page?”.

To capture videos, the experimenter supplies a list of URLs, how many loads to perform per site, and how many seconds to record after onload (since there is no automatic way for webpeg to know when the page has finished loading—if there were, Eyeorg would be unnecessary!). A fresh browser state is ensured by deleting Chrome’s local state after each load. As in [23], before testing a new website, an initial “primer” load is performed. The primer ensures that all needed DNS records are cached at the ISP’s DNS resolver before the first real trial (to prevent a cache miss from skewing the load time results). Local content and DNS caches are disabled and each request carries a cache-control header instructing network caches not to respond.

3.2 Designing Experiments with Eyeorg

Despite its subjective nature, we would like to run quantitative experiments measuring how QoE is impacted by various content or delivery optimizations. To demonstrate how Eyeorg can be used to do this, we picked one aspect of user experience, page load time, and designed two initial types of experiments to study it: timeline and A/B.

Timeline In a timeline experiment, we present the participant with a page load video and ask them to choose the point when they consider the page “ready to use.” The naive way to design this test is to let the participant play the video and ask them to pause it when the page is loaded; however, since they may not be familiar with the page, they will likely wait until they are sure no more content will load, overshooting and pausing too late. Instead, we disable the standard

HTML5 video controls and present participants with a slider they use to “scrub” through the video (Figure 2(a)). Even with the slider, in early tests we observed that (both trusted and paid) participants still overshoot, choosing times well beyond onload. We suspect this is due to a combination of two reasons. First, browsers often only preload a small portion of the video at a time, so when participants seek quickly, they may see a blank screen and assume the *page in the video* has not loaded yet, when in reality the *video* has not loaded yet. By the time the video catches up, they have overshoot. To address this, for timeline tests we force the browser to preload the entire video before the test begins, only then enabling the slider. Second, to correct for simple sloppiness, we add one final step: after the participant chooses a time, we show them the frame they chose and the earliest similar frame (no more than 1% different in a pixel-by-pixel comparison (Figure 3(a))). The participant can either accept our “rewind frame” or stick with their original choice.

We use timeline tests to compare participants’ perceived load time with other metrics like onload and SpeedIndex. Since these metrics are frequently used to evaluate techniques for improving PLT, it is important to understand how closely they match user-perceived load time. Our tests use page load videos for a sample of 100 of the Alexa top 1M sites that fully support HTTP/2 [3]. For each experiment configuration, we repeat each load five times and use the video with the median onload time. Results are described in §5.

A/B The timeline test is tricky; “ready to use” is subjective and participants are not always sure how to decide what time to pick. However, for some experiments, it is not important to choose precisely *when* a page is loaded; it may be useful simply to indicate which of two page loads is *faster*. For these cases, Eyeorg uses A/B experiments. Participants

watch two page load videos simultaneously and pick which loaded faster or “No Difference” (see Figure 2(b)). Video pairs are shown in a random order (i.e., “A” is not always on the left and “B” is not always on the right). There is no guarantee that two videos in a browser stay perfectly synchronized; for instance, lost packets might momentarily stall one video while the other continues playing. To ensure the videos stay synchronized, we splice them into a single video file. If playback stalls, both sides are affected equally.

We demonstrate A/B test’s versatility with two experiments: 1) comparing the speed of HTTP/1.1 vs. HTTP/2, and 2) comparing the impact of ad blockers on PLT. For HTTP/1.1 vs. HTTP/2, we capture videos of the same 100 websites above while loading over HTTP/1.1 and HTTP/2. For the ad blocker analysis, we use data from [3] to identify 10,000 websites that display ads. From this list, we sample 100 websites and make videos of page loads through Chrome using one of three popular ad block extensions: Ad-Block, uBlock, and Ghostery. In this case we do not control the protocol, which means Chrome will default to HTTP/2 if the target website supports it. For each experiment configuration (e.g., “HTTP/2” or “Ghostery”), we repeat each load five times and keep the video with the median onload time. Results are described in §5.

3.3 Crowdsourcing & Response Validation

For large sample sizes, Eyeorg turns to crowdsourcing. While many crowdsourcing services offer built-in tools for creating and hosting tests, we chose to build our own infrastructure. First, the built-in facilities tend to be limited and are not flexible enough to implement our timeline and A/B tests. Second, we did not want to tie Eyeorg to one particular crowdsourcing service; we wanted the ability to draw from a larger, more diverse participant pool, including those not registered as crowd workers, like friends and colleagues. Eyeorg currently supports Microworkers¹¹ and CrowdFlower¹² and can easily be extended to integrate with other platforms.

Using crowdsourced participants poses a challenge: we need to verify that responses are meaningful, e.g., that paid participants do not just blindly click through our tests to get paid (there are indications that workers from well known services like Mechanical Turk¹³ do not always perform as well as desired [13]).

While there is no standardized methodology to determine the quality of a crowd worker, we draw on existing research in the field to detect unreliable responses (using a combination of test-time mechanisms and after-the-fact filtering).

Hard Rules Participants are given a set of clear instructions for each test, defining a set of *hard rules* they must follow for the system to allow them to complete the test. For example, in A/B tests, participants must choose “Left,” “Right,”

or “No Difference” in order to move to the next video. Because we are interested in *human perception*, we also use Google’s “I’m not a robot” service to verify “humanness” before participants take tests.

Soft Rules These are rules that, while not strictly enforced by the system, can reflect the conscientiousness of the workers [19]. Eyeorg does not force participants to watch a video before answering our questions, but failure to do so is a sign of low quality.

Engagement High quality workers tend to be “active.” While the total amount of time a worker spends on a task is a rough indication of the quality of their work [20], it is not enough. On the one hand, a short time might indicate an unengaged worker whose only goal is finishing the task as fast as possible; on the other hand, a long time might be due to a distracted worker. We track how long participants spend reading the instructions, how much time they spend on each video, how many times they play it (if at all), how much of the video they watch, how often they seek, and whether they switch to different tabs or windows during a test. We also capture information about the participant’s system, like browser, OS, and how large the video was on their screen.

Control Questions A common technique in crowdsourcing experiments is to randomly insert *control questions*—questions to which the answer is known [14]. For timeline tests, we want to verify that participants do not blindly accept the rewind frame we suggest, so we occasionally suggest a nearly-blank rewind frame (Figure 3(b)) and check that the participant continues with their original choice. For A/B tests, we occasionally show two copies of the same video with one side artificially delayed by three seconds and check that the participant picks the non-delayed side.

Wisdom of the Crowd A classic problem in crowdsourcing is the lack of ground truth, meaning the quality of responses is difficult to evaluate. However, if most participants are reasonably good, ground truth can be built using the average or majority vote of all responses [25, 7]. Participants whose responses deviate wildly from this pseudo-ground truth can be dropped. We also leverage the crowd to improve the system: participants can report broken videos; a video flagged by 5 different workers is automatically banned and manually inspected by our team.

4. VALIDATING EYEORG’S RESULTS

In this section, we validate the quality of the crowdsourced responses that Eyeorg collects. We introduce the data sets we have collected (§4.1) and present an analysis of various techniques for throwing away unreliable responses (§4.2), which leads us to our final “cleaning” procedure (§4.3).

¹¹<https://microworkers.com>

¹²<https://www.crowdfLOWER.com>

¹³<https://www.mturk.com/mturk/welcome>

	Type	Participants	Male/Female	Duration	Cost	# Sites	Engagement	Soft	Control
<i>Validation</i>									
PLT	timeline	Paid	76/24	1 hour	\$12	20	16	2	7
PLT	timeline	Trusted	79/21	10 days	-	20	10	-	1
H1-H2	A/B	Paid	77/23	1 hour	\$12	20	9	5	2
H1-H2	A/B	Trusted	84/16	10 days	-	20	1	2	1
<i>Final</i>									
PLT	timeline	Paid	685/315	1.5 days	\$120	100	151	45	54
H1-H2	A/B	Paid	697/303	1.5 days	\$120	100	98	56	82
ADS	A/B	Paid	716/284	1.5 days	\$120	100	128	34	57

Table 1: Summary of data collected. Engagement, Soft, and Control indicate the number of participants filtered out by the techniques described in Section 4.3.

4.1 Data Sets

Before exploring methods for detecting low quality workers (§3.3), we define a “pseudo-ground truth” baseline using responses from a set of trusted participants. We evaluate the effectiveness of different filtering techniques by comparing filtered paid participant responses against these baseline trusted participant responses.

To gather the baseline, we ran two small-scale campaigns, one A/B¹⁴ and one timeline, with 20 videos each. For each campaign, we recruited 100 *paid* participants on CrowdFlower and 100 *trusted* participants via email and social media. On CrowdFlower, we requested only workers that are “historically trustworthy,” which comes at the cost of a longer recruitment time. For the trusted participants, we carefully selected friends and colleagues who promised full commitment to the task. We asked each participant to watch six videos (*i.e.*, we served 600 videos in total per experiment and each video has, on average, responses from 30 different participants).

The top of Table 1 summarizes the validation data set. For both timeline and A/B campaigns, it took about one hour to recruit paid participants (cost: \$12) compared to 10 days for the trusted participants (at no cost). Both participant sets exhibit a roughly 75/25% male/female gender split. Paid participants are located in 30 countries (Venezuela being the most popular); trusted participants came from 12 countries (the U.S. being the most popular).

4.2 Evaluating Filtering Techniques

Based on the high-level techniques presented in §3.3, our goal is to design a strategy for filtering out low-quality responses. We validate each technique by comparing paid responses after filtering to the trusted baseline.

Engagement Figure 4(a) shows the Cumulative Distribution Function (CDF) of the time (minutes) each participant spent on Eyeorg broken down both by participant type (paid

and trusted) and experiment type (timeline and A/B). The time spent on site is computed as the sum of the time spent evaluating each video, which in turn is computed as the difference between the time a response was submitted and the time the page loaded.

Figure 4(a) shows that, for both experiment types, the CDFs for paid and trusted participants are quite similar, though paid participants tend to take slightly longer than trusted participants (e.g., the median grows from 2.5 to 3 minutes for the timeline experiment). This result is counter-intuitive, as one would expect paid participants to skim through the videos faster in an attempt to increase their earnings by completing more tasks. Further analysis indicates that this is the result of two things: more “out of focus” time (Figure 5) where Eyeorg is in the background and longer video transfer times. The figure also suggests that the timeline test requires more effort than the A/B test; it takes 3x longer on average. This is primarily because 1) the timeline experiment requires more interaction with the video (Figure 4(b)) to complete the task and 2) the timeline experiment requires the video to be fully loaded before the participant can begin the task (§3.2).

Taking a closer look, Figure 5 shows the CDF of the time a participant did not have the Eyeorg browser tab in the foreground. We do not plot results for trusted participants in the A/B experiments since only one participant switched tabs during that test. For the timeline experiment (paid participants) we also differentiate by how long the video took to load (L). Overall, Figure 5 shows that participants tend to get more distracted the longer the video takes to load—we see about 10% more distracted users when the video takes up to 100 seconds to load compared to less than 2 seconds. A/B campaign participants are, overall, as distracted as timeline campaign participants for whom the video took less than 2 seconds to play. Since in the A/B campaign participants can click play right away and let the video load in the background, this further indicates that long video pre-loading time in the timeline campaign is responsible for the (understandable) engagement drop. In addition, the higher complexity of the timeline campaign might also have a role,

¹⁴The A/B campaign is HTTP/1.1 vs HTTP/2.

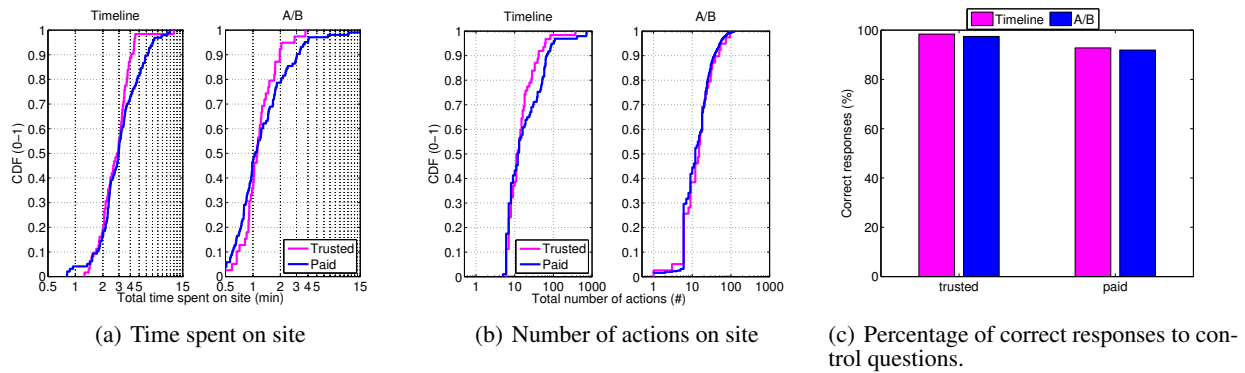


Figure 4: Participant behavior. Comparing the behavior of paid and trusted participants.

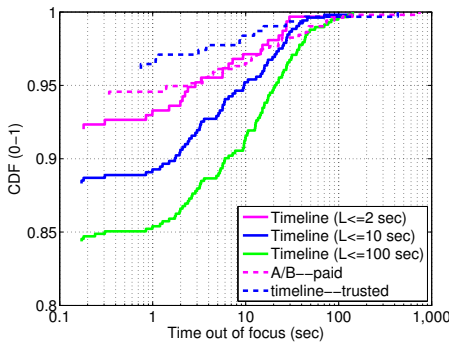


Figure 5: Out of focus time. L = video load time.

as suggested by 4% of the trusted participants also getting distracted for several seconds.

Next, we investigate how participants interact with each video; Figure 4(b) shows the CDF of the total number of actions (play, pause, and seek), again broken down by participant type and experiment type. The figure shows, overall, very similar CDFs, which indicates that paid and trusted participants are more similar in terms of actions performed than time spent on site. This further confirms the impact of external factors, such as long video load times, on the slower responses from paid participants. This result also shows Eyeorg’s ability to equalize participant feedback despite heterogeneous network conditions, which would be impossible if participants were asked to, for example, navigate to a given website and provide feedback.

The tail of Figure 4(b) shows two users performing 714 and 724 actions (seek); this is twice as many actions as the most active trusted participant (369 seek actions). Nevertheless, these participants finished in 2 and 4 minutes, respectively, meaning that despite the large number of actions, they rank in the fastest 15% and fastest 70% of participants overall. Such frenetic behavior looks extremely suspicious, but, from our data, we could not figure out what causes it. We doubt it is a realistic human interaction, and we conjecture a browser extension might have been used.

Soft Rules Though Eyeorg does not enforce it, we set a soft rule that participants should watch each video before submitting a response (§3). Although hard to see due to log scale, Figure 4(b) shows that trusted participants always interact with each video, seeking (timeline) or hitting play (A/B) at least once. By contrast, 1–2% of the paid participants speed through the test without interacting with the video at all.

Control Figure 4(c) shows the percentage of correct responses to control questions. Overall, we notice no major difference between the two control types (timeline versus A/B) for both paid and trusted participants, suggesting both control types were well designed. It is important to note, however, that paid participants tend to fail control questions at a higher rate (5%), suggesting that there are some random clickers. Nevertheless, we found one distracted participant per campaign in the trusted user-set as well.

Wisdom of the Crowd Finally, we investigate whether we can use consensus among paid participants as a proxy for ground truth in the absence of a trusted baseline, starting with the timeline experiment. To build intuition, Figure 6(a) shows the CDF of paid participants’ UserPerceivedPLT for four representative websites. The wisdom of the crowd effect is clearly visible: the majority of the responses are concentrated around one (or a few) UserPerceivedPLT values, e.g., 6.2 and 9.5 seconds for video-2. We further discuss sites with multiple modes (like the one shown in Figure 1) in §6. The figure also shows fairly long heads/tails, indicating that, while most participants agree, some strongly disagree with the rest of the crowd. One possible explanation for this is that these participants simply scroll to the beginning or end of the video in an attempt to finish the test and get paid as soon as possible.

We generalize this observation to the entire timeline campaign by using the standard deviation of the UserPerceivedPLT for each website as a measure of agreement among participants—the tighter the distribution, the more in agreement the responses are. By removing outliers at the ends of each distribution (e.g., keeping only responses between the 10th and 90th percentiles of each site’s UserPerceivedPLT distri-

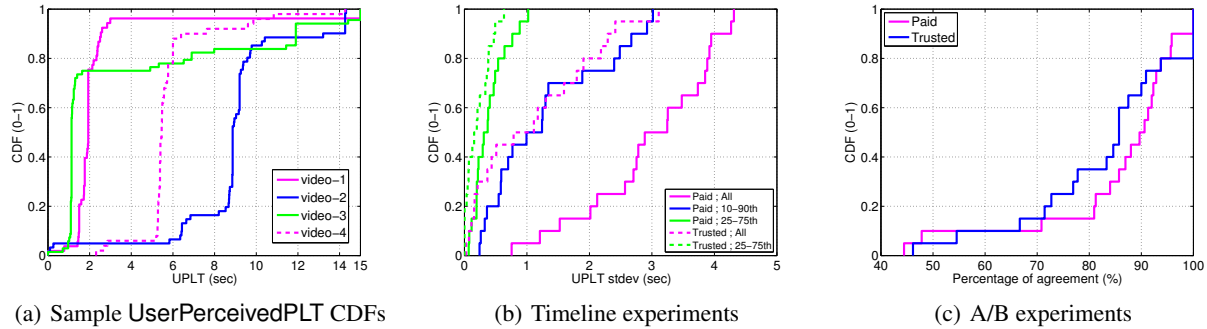


Figure 6: Wisdom of the crowd. With appropriate filtering, the responses of paid participants demonstrate a consistent majority opinion; furthermore, paid responses are in line with those of trusted participants.

bution), we arrive at a set of high-quality responses in relatively tight agreement. Figure 6(b) shows CDFs of standard deviations across all videos in the validation data set. First, without any filtering, the “Paid All” and “Trusted All” show wide standard deviations, which comes as no surprise based on the tails in Figure 6(a). Second, the gap between the “Paid All” and “Trusted All” curves demonstrates the overall lower quality of the paid participants. Third, as we apply filtering, we see that the standard deviation of UserPerceivedPLT drops quickly, indicating most participants are in agreement. Finally, when restricted to responses between the 25th and 75th percentiles, results for paid and trusted participants are in line, which confirms that the wisdom of the paid crowd is a reasonable pseudo-ground truth in the absence of a trusted baseline.

Next, we extend this analysis to the A/B experiment. In this case, the analysis is simpler as the user input is restricted to three discrete values—left was faster, right was faster, or there was no difference—rather than the continuous set of values for the timeline experiment. Accordingly, Figure 6(c) shows the CDF of participant “agreement” for each video for paid and trusted participants. We define agreement as the fraction of responses matching the most popular answer, independent of what that answer is. For example, an agreement of 80% indicates that 80% of the votes went to one choice and the remaining 20% were split, somehow, between the other two.

Figure 6(c) shows no dramatic difference between trusted and paid participants in term of overall agreement. Likely, this is due to the ease of the A/B task compared to the timeline task, as previously discussed. The figure also shows a high level of agreement, e.g., more than 85% of participants converge to the same response for 60% of the videos and 10–20% of the videos have a 100% agreement. In addition, we never saw a completely split response (33% agreement)—the minimum level of agreement in the figure is 45%. Paid participants tend to agree more than trusted participants. Coupled with a lower rate of “no difference” responses, this might indicate that trusted participants are more cautious about their choices.

4.3 Final Filtering Strategy

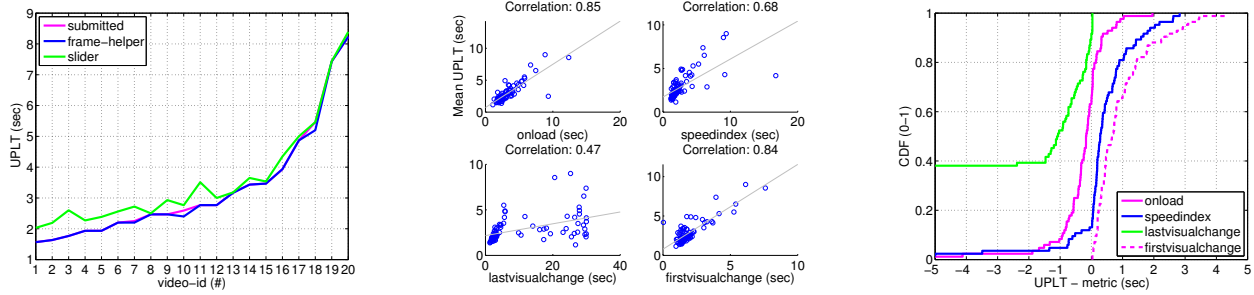
Based on the analysis above, we arrive at the following methodology for filtering low-quality responses. Table 1 (last three columns) summarizes the number of participants dropped per campaign according to the methods below.

Engagement The previous analysis suggests that the time spent on site (Figure 4(a)) is not a reliable way to filter unengaged participants as it can be impacted by external factors like network connectivity. Instead, we use the following two engagement metrics. First, a large number of video interactions (Figure 4(b)) is a good indicator of suspicious activity, so we introduce a simple rule dropping paid participants with 50% more video interactions than the most active trusted participant (369 seeks). This filtering is very minor; it applied to only two paid participants in the validation data-set and three paid participants¹⁵ in the final data-set (§5.1). Second, we apply a more aggressive filter based on the amount of time a participant is distracted, with the caveat that longer video load times may understandably cause participants to briefly switch to other tabs. Accordingly, we filter participants who switch away from the Eyeorg tab for more than 10 seconds so long as the video was delivered within those 10 seconds. Table 1 shows that this removes 10–15% of the paid participants.

Soft Rules We discard responses from participants who skipped (*i.e.*, did not play or scrub) even just one video. This removes 2–5% of both trusted and paid participants (apart from the trusted timeline campaign, where no participant broke this rule).

Control We discard participants who failed any control question. This amounts to 2–8% of participants, with no consistent trend between campaign types (more timeline participants fail the control in the validation set, whereas more A/B participants fail in the final data set). Only one trusted participant failed a control question in each campaign, which

¹⁵These paid participants managed to perform between 912 and 1931 seek actions in 5 minutes.



(a) Comparison of submitted UserPerceivedPLT with slider selection and frame-helper suggestion.

(b) Scatter-plot of UserPerceivedPLT and PLT metrics.

(c) CDF of difference between UserPerceivedPLT and PLT metrics.

Figure 7: Timeline Results.

gives us confidence our control mechanisms do not introduce false positives.

Wisdom of the Crowd For timeline campaigns, we only keep responses between the 25th and 75th percentiles for each video.

5. PAGE LOAD TIME RESULTS

This section summarizes our findings from using Eyeorg to study page load time (PLT), an important aspect of QoE. Specifically, we answer three questions: 1) to what extent do existing PLT metrics represent user perception, 2) do users perceive a speed difference between HTTP/1.1 and HTTP/2, and 3) what impact do ad blockers have on PLT?

5.1 Data Set

To answer these questions, we ran three campaigns, summarized in the *Final* section in Table 1. Each campaign consisted of 100 videos and targeted 1,000 paid participants (again, recruited from CrowdFlower’s “most trustworthy” pool). Each participant watched six videos, meaning we served 6,000 videos in total per campaign and each video has, on average, responses from 60 different participants.

Overall, it took about 1.5 days per campaign to recruit 1,000 participants at a cost of \$120. Across all three campaigns, 70% of the participants are male and 30% are female. Participants are located in 76 distinct countries, with Venezuela being the most popular.

5.2 User Perception and PLT

We compare UserPerceivedPLT against four existing, automatically computable metrics:

OnLoad refers to the time it takes for the JavaScript “onLoad” event to fire. This event fires once the page’s *embedded* resources have been downloaded, but not necessarily before, e.g., objects loaded via scripts are retrieved. OnLoad is the simplest and most used metric in practice.

SpeedIndex “is the average time at which visible parts of the page are displayed” (the average is over all above-the-fold

pixels). Imagine a curve plotting the percentage of pixels that are “visually complete” (*i.e.*, match their final state) over time as the page loads; SpeedIndex is the area *above* this curve (smaller values better—the load feels faster the more “up and to the left” the curve is).

FirstVisualChange/LastVisualChange are the times at which the first pixels are drawn and the last pixels stop changing on the user’s screen.

Figure 7 summarizes the results from the timeline campaign. For each video, we compute the UserPerceivedPLT as the mean of the filtered participant responses. To start, we investigate the impact of the frame selection helper (§3.2). Figure 7(a) compares participants’ final UserPerceivedPLT choices (“submitted”) with their original choices (“slider”) and the one proposed by the frame selection dialog (“frame-helper”). For visibility, we only show results for the 20 videos used for validation in Section 4. For most videos, the submitted UserPerceivedPLT matches the value proposed by the frame selection helper, suggesting that most participants agree with Eyeorg’s suggestions for fine-tuning their responses. On average, the submitted UserPerceivedPLT differs from the original value selected with the slider by 300 ms and up to a maximum of 1.6 s. In the remainder of the PLT analysis, we only consider the submitted UserPerceivedPLT and extend to the full video pool.

Are existing metrics related to UserPerceivedPLT at all?

Figure 7(b) shows how UserPerceivedPLT correlates with each of the four PLT metrics. OnLoad and FirstVisualChange highly correlate with UserPerceivedPLT (0.84–0.85), while SpeedIndex, despite its higher complexity, has a lower correlation with UserPerceivedPLT (0.68). LastVisualChange provides almost no indication of when participants consider a page to be loaded (0.47). Though a non negligible number of responses (7%) exactly match LastVisualChange, most participants indicate that the pages are “ready enough” to be used way earlier.

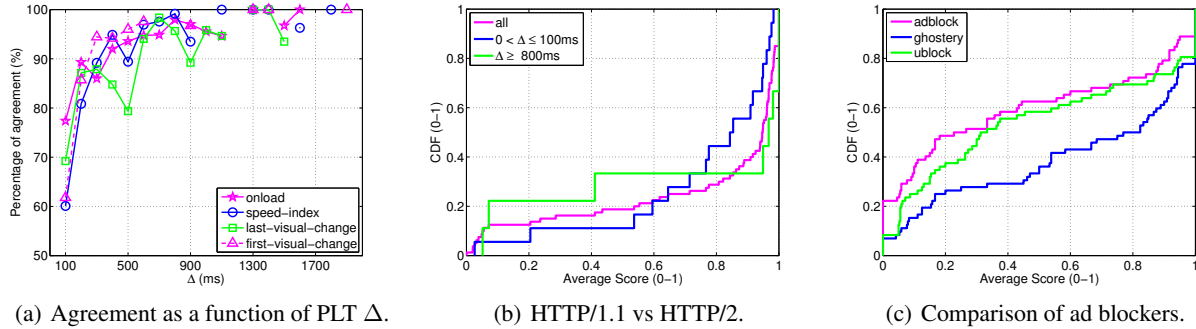


Figure 8: A/B Results.

Are existing metrics able to predict the value of UserPerceivedPLT? While correlation gives a rough picture of how each metric relates to UserPerceivedPLT, it is also important to show how close the actual values are. Figure 7(c) shows a CDF of the difference between UserPerceivedPLT and each PLT metric, *i.e.*, a negative number indicates that participants selected a time *before* that metric indicated the page had loaded. Overall, OnLoad was within 100 ms of the mean UserPerceivedPLT for 30% of the sites compared to just 7% for SpeedIndex. Also note that 60% of the UserPerceivedPLT values are smaller than OnLoad—that is, OnLoad tends to over-estimate. Unsurprisingly, FirstVisualChange and LastVisualChange under- and over-estimate, respectively.

Can existing metrics at least tell us “which is faster”? Finally, we quantify how good the existing PLT metrics are at identifying a difference in load time between two page loads using data from the HTTP/1.1 vs HTTP/2 A/B campaign. Intuitively, choosing which site in an A/B test is faster becomes easier as the absolute difference between the “true” load times for *A* and *B* (Δ) increases. Therefore, we can use the level of participant agreement as a proxy for Δ —the more the participants agree, the larger Δ likely was.

Based on this reasoning, 1) if a PLT metric is good, we expect to see agreement increase monotonically as the measured Δ for that metric increases, and 2) the higher the agreement for a particular Δ -value for a particular metric, the more confidence we have that when we measure a Δ of that size in the future, it represents a meaningful difference in UserPerceivedPLT. Figure 8(a) shows the median agreement (as previously defined in §4.2 Figure 4(c)) among participants for HTTP/1.1/HTTP/2 video pairs as a function of each metric’s Δ . Overall, Figure 8(a) shows that as Δ increases, participants tend to agree more, which matches our intuition. While there is no clear winner among the PLT metrics, the figure shows that OnLoad better captures small loading time differences ($\Delta \leq 200$ ms) whereas SpeedIndex and FirstVisualChange do a better job in the medium range ($200 < \Delta \leq 800$ ms). The figure also shows that LastVisualChange and SpeedIndex do not exhibit a monotonic increase in agreement as Δ grows, meaning that small variations in LastVisualChange and SpeedIndex measurements are less significant.

5.3 HTTP/1.1 vs HTTP/2

Do users perceive a speed difference between HTTP/1.1 and HTTP/2? In this section, we examine the responses from our HTTP/1.1 vs HTTP/2 A/B campaign. Figure 8(b) shows the CDF of the average “score” per website; 0 means the HTTP/1.1 version was faster, 0.5 is a “split” decision, and 1 means the HTTP/2 version was faster. We plot scores for 1) all websites, 2) websites with similar HTTP/1.1 and HTTP/2 PLTs ($\Delta \leq 100$ ms), and 3) websites that loaded much faster over one protocol than the other ($\Delta \geq 800$ ms). To build these subsets, we compute PLT using SpeedIndex.

Figure 8(b) shows that 70% of the websites have an average score of 0.8 or higher; this means that 70 out of 100 websites “feel” faster using HTTP/2 than HTTP/1.1. Conversely, 12% of the websites have an average score of 0.2 or lower and thus feel faster using HTTP/1.1. The remaining 18% of websites create some disagreement. Note that the score here does not take into account the “No Difference” responses. These websites with scores in the 0.2–0.8 range also have twice as many No Difference responses compared to the other websites. This further indicates that participants are just not sure which version was actually faster.

Next, we focus on the subset of websites with similar PLTs ($\Delta \leq 100$ ms). The figure shows that participant indecision grows, with more scores in the 0.2–0.8 range. This is to be expected based on the results from Figure 8(a). On the other hand, when $\Delta \geq 800$ ms, participants mostly agree on which version was faster. This result indicates that, while aiming at reducing loading time of a webpage is overall beneficial, many users are not able to appreciate the difference when only few hundred milliseconds are saved.

5.4 Ad Blocker Comparison

How do popular ad blockers impact PLT? We compare three popular ad blockers, AdBlock, Ghostery, and uBlock. Figure 8(c) shows the CDF of the average “score” obtained by each website where 0 means the original version with ads was faster and 1 means the ad-blocked version was faster.

The figure shows that 30–40% of the websites have scores in the 0.2–0.8 range, *i.e.*, participants did not agree on which version was faster. This is about 15% more compared to

when participants were asked to evaluate HTTP/1.1 versus HTTP/2. Based on feedback collected on Eyeorg, we believe this is due to the fact that the two versions of the websites are now not perfectly equal, which makes deciding which was faster harder. Nevertheless, the figure shows Ghostery is a clear favorite; for example, for 50% of sites, participants strongly agreed (≥ 0.8) that the Ghostery version of the page was faster compared to just 25% for AdBlock and uBlock.

6. DISCUSSION

What Does “Ready” Mean? In our timeline tests, we ask participants to “drag the slider to the point where you consider the site ‘ready to use.’” We intentionally left the wording open to individual interpretations of “ready,” since *what humans consider “ready” is exactly what we’re trying to learn*. If, for example, we instructed participants to “pick the point where content stops changing,” we would simply be asking them to reproduce LastVisualChange.

To see if participants have consistent definitions of “ready,” we look at the UserPerceivedPLT distributions for different sites. Three rough patterns emerge (Figure 9). 1) Some sites exhibit a single, clear peak in UserPerceivedPLT choices. After manually inspecting the associated videos, these sites tend to be characterized by “fast” loads (in the sense that the span of time between FirstVisualChange and LastVisualChange is very short). These cases are relatively cut-and-dry; not much is open to interpretation, and participants are pretty consistent with the times they choose. 2) Some sites have a much wider distribution. These sites tend to have a much longer gap between FirstVisualChange and LastVisualChange, giving participants more freedom to choose different “ready” times. 3) Some sites exhibit *multiple* peaks. In some cases, this appears to be due to auxiliary content, like social media widgets and ads—some participants consider the page ready after the main content has loaded, while others wait for the auxiliary content to load.

Comments we collected from trusted participants help shed light on this. The first two comments confirm that auxiliary content does impact some participants’ responses:

“Sometimes my choice may not be ... due to the [order] in which the content loaded. I selected the one where the main content loaded first, not necessarily which one finished loading everything [first].”

“Ads/like buttons/etc. usually load last; it’s a little unclear whether ‘ready to use’ should include those or not. I start using sites before all of the ads loaded, but when I do that, I know that the page isn’t totally done loading—I just don’t care.”

In conjunction with Figure 9 these comments are particularly interesting in light of new features in HTTP/2 like push and priority that could be exploited to ensure the optimal delivery order for *specific* users (similar to Klotski [10]).

The following two comments echo the first two, but also point out some limitations of our methodology. First, some

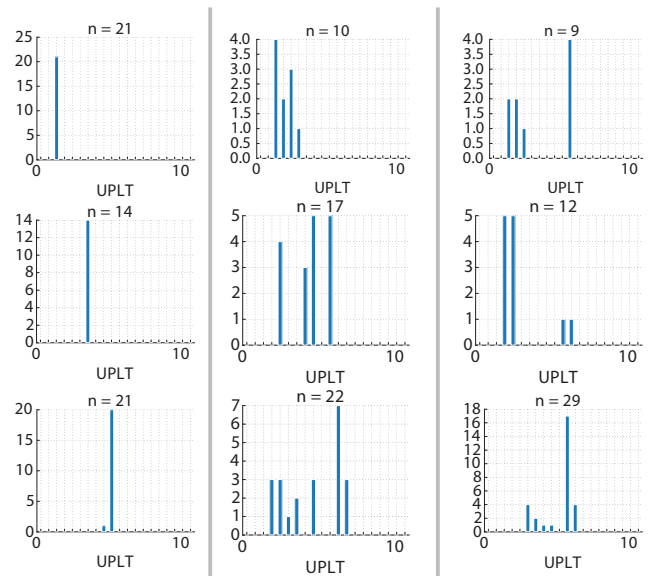


Figure 9: UserPerceivedPLT Distributions: Sites typically had one of three types of distributions: single mode and tight (left column), single mode and spread out (center column), or multi-modal (right column).

participants’ responses are affected by their familiarity with a site. They are easily able to identify when the important content is loaded if the site is familiar (*i.e.*, they will think the page is loaded “faster”), whereas with an unfamiliar site they might wait until every piece of content loads. Second, some participants look for more than just visual cues, e.g., when an element on the page becomes responsive to user input. Eyeorg’s page load videos fail to capture these cues.

“Sometimes how early I feel the site is ready to use depends on what I want to use the site for. For example, if I am looking for articles, I don’t care if the ads have loaded. So usually I consider the webpage ready when the part I want has loaded. However, there are other cases, like I don’t know what is on the site, when I want to wait for everything to load. There is also the case where the links move when other parts load and for those cases, I commonly have to wait for the whole page to load because it’s too easy to click where the link used to be and click something else instead.”

“It is difficult to say when a site becomes ‘usable’ without knowing whether the search bar is responsive. For many sites I consider it usable the instant the search bar becomes responsive.”

Extending Eyeorg In this paper, we only scratched the surface of what can be done with Eyeorg. Many *existing* features of Eyeorg that enable rich experimentation, e.g., (mobile) device and network emulation, have not been discussed at all. Similarly, although we have provided an extensive analysis into understanding how humans perceive PLT, we are fully aware that *many* open questions remain. We believe that the community at large can leverage Eyeorg as a

platform to enable high quality, novel research that answers these and other questions. Eyeorg can be used to measure the impact of a variety of changes to the web; e.g., TCP vs. QUIC, TLS 1.2 vs TLS 1.3, HTTP/2 push/priority strategies, web design techniques like domain sharding or image spriting, browser plugins, or even in-network services like Google’s Flywheel compression proxy [4].

Data Collection and Privacy Eyeorg collects data from human participants, so we have to take care we collect and use this data responsibly. (Note that paid participants may also provide data directly to the crowdsourcing services Eyeorg uses; this data is not shared with us and is subject to those services’ privacy policies.)

We collect three types of data. First, we ask participants to provide standard demographic information like gender, age, country, and self-assessed technical ability. We collect this information at a coarse enough granularity there is no danger of identifying individual people. Next, we collect activity data like how long a participant spends reading instructions, when they play/pause a video, and whether they switch away from the Eyeorg tab during an experiment. (Note that we only collect information about the Eyeorg tab; if they switch away, we have no information about what they do.) We use this data to filter out uncommitted participants. We do *not* use this data to discriminate (e.g., refuse payment) against participants. Finally, we collect participant responses to our QoE questions (e.g., “when is this page ready to use” or “which video is faster”). Clearly this data is needed, as it is what Eyeorg seeks to measure to begin with.

7. CONCLUSION

This work presented Eyeorg, a platform for crowdsourcing web quality of experience measurements. Eyeorg collects *quantitative* QoE measurements from real users *at scale*. Eyeorg relies on *videos* to ensure each participant sees a consistent view of the page loading; our video collection tool control parameters like device type, protocol, network conditions, and more. We validate the responses collected from crowdsourced workers by comparing them to responses from a controlled set of 100 trusted users. Next, we conduct three measurements campaigns involving 1,000 participants each; our campaigns investigate the quality of existing PLT metrics, compare HTTP/1.1 vs HTTP/2 performance, and assess the impact of online advertisements on user experience. Going forward, we plan to extend Eyeorg with a broader set of capabilities for conducting web QoE experiments. In particular, we plan to make Eyeorg a platform that any researcher can use to test web content or delivery optimizations easily, without worrying about the challenges of designing a user study from scratch.

Acknowledgements

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant

agreement No 688421. The opinions expressed and arguments employed reflect only the authors’ view. The European Commission is not responsible for any use that may be made of that information.

8. REFERENCES

- [1] Globaleventhandlers.onload. <https://developer.mozilla.org/en/docs/Web/API/GlobalEventHandlers/onload>.
- [2] How One Second Could Cost Amazon \$1.6 Billion In Sales. <http://www.fastcompany.com/1825005/how-one-second-could-cost-amazon-16-billion-sales>.
- [3] Is the Web HTTP/2 Yet? <http://isthewebhttp2yet.com>.
- [4] AGABABOV, V., BUETTNER, M., CHUDNOVSKY, V., COGAN, M., GREENSTEIN, B., MCDANIEL, S., PIATEK, M., SCOTT, C., WELSH, M., AND YIN, B. Flywheel: Google’s data compression proxy for the mobile web. In *Proc. USENIX NSDI* (Oakland, CA, USA, May 2015).
- [5] ARAPAKIS, I., BAI, X., AND CAMBAZOGLU, B. B. Impact of response latency on user behavior in web search. In *Proc. ACM SIGIR* (Gold Coast, Queensland, AU, July 2014).
- [6] BARREDA-ÁNGELES, M., ARAPAKIS, I., BAI, X., CAMBAZOGLU, B. B., AND PEREDA-BAÑOS, A. Unconscious physiological effects of search latency on users and their click behaviour. In *Proc. ACM SIGIR* (Santiago, Chile, Aug. 2015).
- [7] BLACKBURN, J., AND KWAK, H. STFU NOOB! Predicting Crowdsourced Decisions on Toxic Behavior in Online Games. In *Proc. WWW* (Seoul, Korea, Apr. 2014).
- [8] BOCCHI, E., DE CICCIO, L., AND ROSSI, D. Measuring the quality of experience of web users. In *In Proc. LANCOMM* (Florianopolis, Brazil, Aug. 2016).
- [9] BOUCH, A., KUCHINSKY, A., AND BHATTI, N. Quality is in the eye of the beholder: Meeting users’ requirements for internet quality of service. In *CHI* (The Hague, The Netherlands, Apr. 2000).
- [10] BUTKIEWICZ, M., WANG, D., WU, Z., MADHYASTHA, H. V., AND SEKAR, V. Klotski: Reprioritizing web content to improve user experience on mobile devices. In *Proc. USENIX NSDI* (Oakland, CA, USA, May 2015), NSDI’15.
- [11] EGGER, S., REICHL, P., HOBFELD, T., AND SCHATZ, R. “time is bandwidth”? narrowing the gap between subjective time perception and quality of experience. In *Proc. IEEE ICC* (Ottawa, Canada, June 2012).
- [12] EYEORG. A platform for crowdsourcing web quality of experience measurements. <https://eyeorg.net>.
- [13] HARRIS, A. Dropping mechanical turk helps our customers get the best results, Jan. 2014. <https://speedcurve.com>.

- [14] HOSSFELD, T., KEIMEL, C., HIRTH, M., GARDLO, B., HABIGT, J., DIEPOLD, K., AND TRAN-GIA, P. Best practices for qoe crowdtesting: Qoe assessment with crowdsourcing. *Trans. Multi.* 16, 2 (Feb. 2014).
- [15] LIDSKOG, T. browsertime. <https://github.com/tobli/browsertime>.
- [16] LIMITED, S. Speedcurve. <https://speedcurve.com>.
- [17] NETRAVALI, R., GOYAL, A., MICKENS, J., AND BALAKRISHNAN, H. Polaris: Faster page loads using fine-grained dependency tracking. In *Proc. USENIX NSDI* (Santa Clara, CA, USA, Mar. 2016).
- [18] NIKRAVESH, A., YAO, H., XU, S., CHOFFNES, D., AND MAO, Z. M. Mobilyzer: An open platform for controllable mobile network measurements. In *Proc. ACM MobiSys* (Florence, Italy, May 2015).
- [19] RAINER, B., WALTL, M., AND TIMMERER, C. *A Web based Subjective Evaluation Platform*. Klagenfurt am Worthersee, Austria, July 2013.
- [20] RZESZOTARSKI, J. M., AND KITTUR, A. Instrumenting the crowd: Using implicit behavioral measures to predict task performance. In *UIST* (Santa Barbara, California, USA, May 2011).
- [21] SHULL, M. Measuring performance, Apr. 2015. <https://davidwalsh.name/measuring-performance>.
- [22] STEFANOV, S. Book of speed. <http://www.bookofspeed.com/chapter2.html>.
- [23] VARVELLO, M., SCHOMP, K., NAYLOR, D., BLACKBURN, J., FINAMORE, A., AND PAPAGIANNAKI, K. Is the web http/2 yet? In *Proc. PAM* (Crete, Greece, Mar. 2016).
- [24] WANG, X. S., KRISHNAMURTHY, A., AND WETHERALL, D. Speeding up web page loads with shandian. In *Proc. USENIX NSDI* (Santa Clara, CA, USA, Mar. 2016).
- [25] WELINDER, P., BRANSON, S., PERONA, P., AND BELONGIE, S. J. The multidimensional wisdom of crowds. In *Advances in Neural Information Processing Systems* (2010).
- [26] XIAO SOPHIA, W., ARVIND, K., AND DAVID, W. Speeding up web page loads with shandian. In *Proc. USENIX NSDI* (Santa Clara, CA, USA, Mar. 2016).