Perceiving QUIC: Do Users Notice or Even Care?

Jan Rüth, Konrad Wolsing, Klaus Wehrle
RWTH Aachen University
{rueth,wolsing,wehrle}@comsys.rwth-aachen.de

Oliver Hohlfeld
Brandenburg University of Technology
oliver.hohlfeld@b-tu.de

ABSTRACT

QUIC, as the foundation for HTTP/3, is becoming an Internet reality. A plethora of studies already show that QUIC excels beyond TCP+TLS+HTTP/2. Yet, these studies compare a highly optimized QUIC Web stack against an unoptimized TCP-based stack. In this paper, we bring TCP up to speed to perform an eye-level comparison. Instead of relying on technical metrics, we perform two extensive user studies to investigate QUIC's impact on the quality of experience. First, we investigate if users can distinguish two protocol versions in a direct comparison, and we find that QUIC is indeed rated faster than TCP and even a tuned TCP. Yet, our second study shows that this perceived performance increase does mostly not matter to the users, and they rate QUIC and TCP indistinguishable.

CCS CONCEPTS

• **Networks** → **Transport protocols**; **Network performance analysis**; **Network measurement**.

1 INTRODUCTION

QUIC provides a privacy-preserving and fully-encrypted transport in user-space to ease protocol evolution. By incorporating the ideas of several enhancements to TCP, TLS, and even HTTP, it promises an alternative Web protocol stack that is optimized for performance, security, and evolvability. For example, it combines TLS 1.3 early-data and TCP Fast Open (TFO) to achieve 0-RTT connection establishment and data exchange. It also offers multiplexing streams over the same connection to overcome head-of-line blocking, e.g., by mapping HTTP/2-like streams to the transport streams.

Given QUIC's motivation to increase performance, it is no surprise that there are studies [1, 3, 4, 9, 11, 25] showing that QUIC clearly outperforms the traditional Web stack (i.e., HTTP over TLS over TCP). Although they are subject to several limitations; e.g., many studies utilize the Page Load Time (PLT) to measure performance. However, it has been shown [2, 10, 27] that PLT does not correlate well with human-perceived performance. Another

shortcoming is that *all* studies compare QUIC, having a highly optimized parameterization, against an off-the-shelf Web stack, which, however, offers similar optimization potential that is not used. For example, Chromium's Google QUIC utilizes packet pacing and an initial congestion window of 32 segments. In contrast, a stock Linux TCP stack defaults to no pacing and an initial congestion window of 10 segments, obviously disadvantaging the regular Web stack that can be avoided by parametrizing TCP similar to QUIC.

While QUIC offers new levels for protocol customization and evolution, it remains open (from a performance perspective) if switching to QUIC should be a top priority. Hence, we ask: *do humans even notice a difference, and if so, how much of a difference does it make?* That is, does QUIC actually impact the *Quality of Experience* (QoE). We answer these questions in two user studies.

To achieve an unbiased comparison between the two stacks, we modified the Mahimahi [15] framework and incorporated the Google QUIC Web stack. This emulation grants us full control over the network parameters, i.e., bandwidth, delay, and queues, as well as the client and server, i.e., enabling to modify the protocol's parameterization for comparable measurements. We then use video recordings of website visits in our testbed to perform studies in a controlled lab environment, using a crowdsourced online marketplace, and a voluntary crowd study to investigate the QoE of QUIC. Our contributions are:

- We perform the first QoE user studies that investigate how real users perceive QUIC.
- Our studies are unique in that they compare QUIC against a similarly parameterized TCP stack enabling eye-level comparisons.
- We find that users *do* perceive QUIC as the faster protocol in a side-by-side comparison, even against a tuned TCP stack.
- However, in isolation, users generally do not prefer one protocol over the other if the network is sufficiently fast.
- In slow and lossy networks, QUIC's advanced protocol design appears to cause a more satisfying loading process for our study participants, underlining QUIC's potential to improve on the long-tail of bad experiences.
- All our study data is available at https://study.netray.io.

Structure. This paper first reviews related works (Section 2) to see what the state-of-the-art discovered about QUIC and QoE. Then, Section 3 describes our testbed and how we configured TCP similar to QUIC. Continuing, we describe the design of our two user studies (Section 4) and evaluate the outcome of both. Finally, Section 5 concludes this paper.

2 RELATED WORK

QUIC Performance. QUIC performance is subject to a body of studies [1, 3, 4, 9, 11, 14, 20, 25], most compare QUIC against some combination of TCP+TLS+HTTP/1.1 or HTTP/2.

One direction of research [4, 11] measures TCP and QUIC on publicly hosted websites—usually operated by Google. This approach has the advantage that TCP and QUIC are *likely* configured comparably on the same server. However, these studies do not confirm any of the protocol parameterizations. The disadvantage of these approaches is that they lack the control of the network and the server, i.e., it becomes much harder to gain comparable measurements for QUIC and TCP since the network and the server are subject to other traffic that can have a significant impact on the measurement. Furthermore, if the protocols are not parameterized similarly, there is no way to change this.

To enable controlled experiments, another direction [1, 3, 9, 14] uses self-hosted servers. While this enables tuning the protocols to be comparable, it is not done, and the studies compare a likely unoptimized TCP against a highly tuned QUIC Web stack. To the best of our knowledge, Yu et al. [25] are the only ones that investigate the impact of packet pacing in QUIC as a tuning option — yet, they do not compare to TCP. Similarly, Cook et al. [4] take the design of TCP and QUIC into account when looking at first and repeated connections that require less RTTs on the subsequent visit.

While self-hosting offers greater freedom than using existing sites, websites today are composed of a variety of resources that are often hosted by third parties on different servers. Even though many studies consider websites with different resources, they often deploy only a single server [1, 3, 11]. The Mahimahi framework [15] was designed to study realistic websites by replicating this multi-server nature of current websites into a testbed on which we also base our testbed (see Section 3). Nepomuceno et al. [14] also perform a study with Mahimahi but find that QUIC is outperformed by TCP, which does not coincide with our results and related work. We believe this is due to two reasons, first the use of the Caddy QUIC server, which is known to not (yet) perform very well [13], and second, they do not configure any bandwidth limitations. For this reason, we chose the Google QUIC stack as it is already deployed in the Internet [18] and likely offers good performance, which our evaluations support for the websites that we tested.

QUIC Web Performance. Another line of research investigates the performance of QUIC using (visual) Web performance metrics. Seufert et al. [19, 20] investigate the QoE of YouTube video streaming by recording application layer metrics such as video quality or stalls and find no evidence for QoE improvements of QUIC over TCP. Rajiullah et al. [16] achieve a similar result. The authors conduct mobile measurements using the MONROE framework. They evaluate technical metrics such as First Visual Change (FVC), Last Visual Change (LVC), and the RUM Speed Index (SI). While they find websites where QUIC has a clear impact, they conclude that overall, it has a negligible influence.

In summary, we did not find *any* work investigating the *user perception* of QUIC with the help of real user studies and no other work parameterized TCP stacks similar to QUIC. In our work, we focus on both by performing a human-centered QoE evaluation of QUIC in comparison to a similarly parameterized TCP.

Protocol	Description
TCP	Stock TCP (Linux): IW10, Cubic
TCP+	IW 32, Pacing, Cubic, tuned buffers, no slow start after idle
TCP+BBR	TCP+, but with BBRv1 as congestion control
QUIC	Stock Google QUIC: IW 32, Pacing, Cubic
QUIC+BBR	QUIC, but with BBRv1 as congestion control

Table 1: The different protocol configuration that we used in our tests.

3 REPEATABLE PROTOCOL PERFORMANCE EVALUATIONS

To compare the performance of website load processes subject to different protocol configuration, we reuse our Mahimahi-based testbed from [24]. In that work, we analyzed QUIC's performance subject to several Web performance metrics such as FVC, LVC, SI, Visual Completeness 85% (VC85), and PLT. To this end, we embedded the gQUIC Web stack as well as an NGINX Web server in Mahimahi and repeatedly requested a set of previously recorded websites using the Chromium browser, which we automated using Browsertime [21]. In the following, we highlight the key properties of the testbed as well as the configurations that we applied for our QoE user studies.

Protocol Parametrization. By default, Mahimahi does not alter the Linux stack's TCP configuration and uses the shipped defaults. Please note that we refer to the *configurations and parameterizations* and not each implementation's bare performance when we refer to the stacks as tuned.[1] We extend Mahimahi in that it is simple to reconfigure the stacks before requesting websites. Table 1 summarizes the different protocol configurations that we apply to QUIC and TCP. To this end, our *TCP+* is configured to match gQUIC's default of an initial congestion window of 32 segments, using pacing (with Linux's defaults of an initial quantum of ten and a refill quantum of two segments) and not falling back to the initial window after idle. Further, we enlarge the send and receive buffers according to the bandwidth-delay product (BDP) of the underlying network to allow full utilization. Moreover, since we already found a significant impact of the congestion control in [24], we further use a variant of TCP and QUIC that utilized BBRv1.[2]

A website visit in our testbed is done using a fresh Chromium browser with an empty cache. Starting from scratch has several implications on the protocols: it means that QUIC will *not* perform a 0-RTT connection establishment but a 1-RTT handshake. Since there is no support for TLS 1.3 early-data in Chromium (as of June 2019) and a limited and challenging deployment of TFO in the Internet, this still gives a 1-RTT advantage of QUIC over TCP+TLS+HTTP/2. In [24], we found that this 1-RTT advantage is the primary factor for QUIC outperforming the traditional Web stack in non-lossy environments when looking at technical metrics. QUIC, similar to early-data and TFO, suffers from replay attacks, which have an especially large surface in distributed clusters [5] when requests are non-idempotent. To this end, e.g., Cloudflare only allows non-parameterized GET requests via TLS 1.3 0-RTT [22]. In summary, there is currently a lack of signaling idempotency

---

[1]We believe that the Linux TCP stack, given its age, has a more efficient code-base.
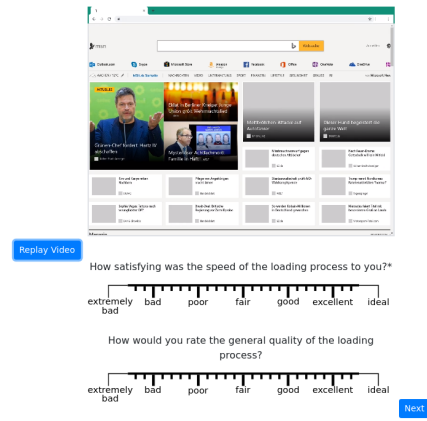[2]BBRv2 was not yet available at the time of testing.
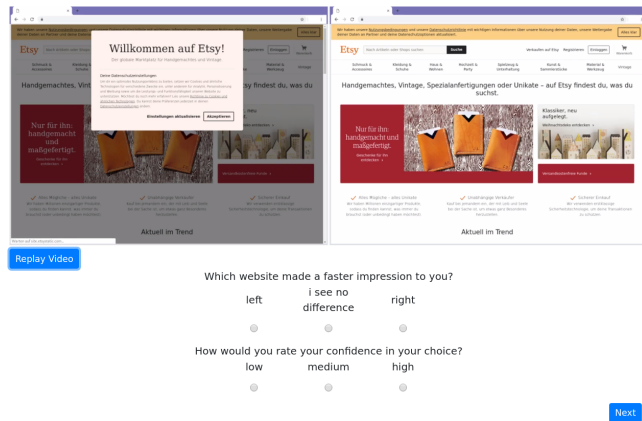
Figure 1: Example screenshot from the A/B study with video and questions. Note that usually the questions are hidden and pop up one after the other.

Network	Uplink	Downlink	min. RTT	Loss
DSL	5 Mbps	25 Mbps	24 ms	0.0 %
LTE	2.8 Mbps	10.5 Mbps	74 ms	0.0 %
DA2GC	.468 Mbps	.468 Mbps	262 ms	3.3 %
MSS	1.89 Mbps	1.89 Mbps	760 ms	6.0 %

Table 2: Network configurations. Queue size is set to 200 ms except for DSL with 12 ms.

through the stack such that 0-RTT could be easily enabled on a large scale. Thus, we believe that comparing a 1-RTT QUIC with a 2-RTT TCP/TLS reflects a likely Internet-wide deployment of both, even though we will see 0-RTT QUIC and TLS early-data in parts.

Websites Selection. Selecting websites that are representative of large parts of the Web is challenging. We base our selection of websites on [23], which ultimately derived a list of 40 sites from the Alexa Top 50 and Moz Top 50. They were chosen to have a high variation in size (number of objects and their sizes) as well as contacted IP addresses (multi-server nature). A visualization of this variation (in terms of multi-server nature and website size distribution) is given in the original paper [23] as well as our previous work [24]. For this paper, we restrict our view to 36 of the 40 websites. We had to exclude four sites since we were unable to replay them accurately, and one was a private project site.

Network Parameter Selection. Within our user study, we want to cover the user-perception in "good" networks as well as in "bad" networks. To this end, we select four different network settings, which we summarize in Table 2. We chose the median bandwidth of German households according to the federal network agency [26], which we refer to as DSL. This network has no artificial random loss, and we set a low minimum round-trip time (RTT) to which the queue adds further jitter (up to 12 ms). Similarly, we use median bandwidth values for German mobile users, which we refer to as LTE. Even though it is a wireless link, we set no artificial loss as the link-layer would recover it. Still, the min. RTT is already higher. Furthermore, we allow up to 200 ms of queuing. Lastly, we take network parameters for two "bad" in-flight WiFi networks that connect either via LTE to the ground (DA2GC) or via an in-flight

Figure 2: Example screenshot from the rating study with video and questions. Note that usually the questions are hidden and pop up one after the other.

satellite connection (MSS). Those parameters have been established in [17] and are characterized by low bandwidths and high delays as well as high random loss.

Producing Videos. To record videos to show in our user study, we automate the Chrome browser to visit the 36 websites replayed in our testbed at least 31 times while recording the browser window. This enables us to derive technical and visual metrics (FVC, LVC, PLT, SI, VC85) with a certain accuracy (for later comparison) and subsequently select a video that closely fits a "typical" recording by taking the video that is closest to the average PLT inspired by [27].

4 USER PERCEPTION OF PROTOCOL SPEED

The key focus of our study is to evaluate the effect of protocol performance on user perception, tested in two studies.

Study 1 (A/B): Do Users Notice? We begin by performing a just noticeable difference test to identify if users notice a protocol switch. The study design involves a pairwise comparison where two recordings of the loading processes of the same website with different protocol configurations but the same network setting are shown side-by-side (rendered into a single video) to participants (see Figure 1). This pairwise comparison allows us to detect even subtle differences in the loading processes. After watching the video, the participants are asked to answer if the left or right video was the faster one or if they cannot decide. We furthermore ask them to rate their confidence in their choice.

Study 2 (Rating): Do Users Care? While the first study informs us if protocol switches lead to perceivable differences, it does not tell how users rate the perceived quality of the loading process. We answer this aspect in a second study, in which we only present one video (see Figure 2) to let participants rate *i)* their satisfaction with the loading speed and *ii)* the general quality of the loading process. Both ratings are obtained on a seven-point linear scale [8] ranging from extremely bad over bad, poor, fair, good, excellent to ideal, mapped to values from 10 to 70 with equidistance selectable by participants with a granularity of 1. To set a context for assessing speed perception, we ask the participants consider being in a particular environment for this study: imaging being *i)* at work, *ii)* in their free time, or *iii)* on a plane.

We implemented the user studies using TheFragebogen [6] and host it on our infrastructure. Each study begins with a tutorial that explains the purpose and the procedure of the study. By informing the participants on the study goals and its procedure, it also aims at reducing noise in the responses.

Pilot Study. We tested our study in a pilot study before releasing it. It involves volunteers (friends and colleagues) testing our system to see if people that are unfamiliar with the study can perform it. The results of the pilot study are not used for evaluation, and participants did not participate in both the pilot and the later study to limit bias by training effects. The main feedback was that people were overwhelmed when the videos start without a tutorial up front, which we added. Secondly, we also rendered a Web browser window around each video (also shown in the figures) as we got the feedback that otherwise, people were unsure about the bounds of each website.

4.1 Performing the User Studies

We utilize three different subject groups for our user study.

Lab Study. First, we perform a controlled lab study with both the rating and the A/B study (the beginning is randomized), where we can monitor and supervise the study. Since the lab supervisor monitors the user behavior during the lab study (e.g., to check that participants properly conduct the test, i.e., actually watching the videos and not clicking randomly on the scales), it serves as our control data to evaluate the other two uncontrolled crowdsourcing groups. As this control group is rather small, we only consider five domains (wikipedia.org, gov.uk, etsy.com, demorgen.be, nytimes.com), which are diverse in website size such that the overall duration for each participant is roughly 10 min. This constraint leads us to show 28 videos for the A/B study and 11 in the work, 11 in the free time and only five, due to the increased video length, in the plane environment for the rate study.

Crowdsourcing Studies. We employ crowdsourcing to enlist a larger number of participants. As a second group, we recruit paid Microworkers [12] participants. We follow the platform's guidance and offer 0.75 USD for a study between 10 and 15 min length and allow participation of a user in each study only once. After a Microworker (μWorker) completed the study, she can redeem her payout using a code that we display at the end of the study. We show 26 videos in the A/B study and again 27 (11 work, 11 free time, and 5 plane) in the rating study. Third, we advertise the studies on social media to recruit regular Internet users. As we expect unwillingness to perform a lengthy study, we show only 14 videos (A/B) and 15 videos (6 work, 6 free time, 3 plane) in the rating study.

Conformance Filtering. While the controlled lab study helps us in judging the quality of the crowd-sourced results, we take extensive measures to ensure valid results since we suspect that at least on the Microworker platform, people will cheat the system to solve the study quickly. To this end, we implement 7 rules that are used to filter invalid results: **R1:** A video in the study has not been played. **R2:** A video has stalled. **R3:** There is a focus loss event (e.g., website not the active tab or window not in the foreground) for longer than 10 sec during the study. **R4:** A vote was placed before the FVC. **R5:** A study took longer than 25 min or a question took longer than 2 min. **R6:** A randomly placed control video was

		-	R1	R2	R3	R4	R5	R6	R7
Lab	A/B	-	-	-	-	-	-	-	<u>35</u>
	Rating	-	-	-	-	-	-	-	<u>35</u>
μWorker	A/B	487	471	441	355	268	268	239	<u>233</u>
	Rating	1563	1494	1321	1034	733	723	661	<u>614</u>
Internet	A/B	218	217	210	196	171	170	159	<u>155</u>
	Rating	209	204	194	172	152	151	140	<u>138</u>

Table 3: Participation in our studies and results after each filter rule, final participation are underlined.

answered wrong. In the A/B study, we embed significantly delayed variants of the left or right video or have the same video on both sides.[3] In the rating study, we embed a very quickly rendering website and a very slow one; we expect the rating to be at least 10 points apart. **R7:** A control question was answered wrong. Every fifth video includes an additional question that asks for the color of the browser frame, which is still visible while answering the question (see, e.g., Figure 2 having a green browser frame). Each video is assigned a random color from red, green, and blue; we chose the exact colors to be colorblind safe.

Table 3 summarizes the participation and how many results we removed due to each of the filters. Since we allow each μWorker to only participate once[4] in each study, and we suspect Internet users not to repeat the studies, these numbers should be close to the true number of individuals participating. Focus loss (R3) and voting before the FVC (R4) filtered the most results.

Ethical Considerations. Our study design follows standard guidelines for conducting crowdsourcing QoE studies [7]. Each participation (in the lab, from the Internet or as paid workers) takes place voluntarily. For the Microworker platform, we follow the platform's guidelines for payment. We chose not to pay lab participants to allow participation without monetary pressure. The user studies clearly state which data we are gathering, and only after completion of the study, this data is uploaded securely to our servers. Regarding the stimulus, the content in all videos does not show any sensitive material, e.g., violence, abuse, or other questionable content. In case of difficulties or errors, we are reachable via email on the study website and directly via the Microworker platform.

4.2 Study Agreement

We first compare our controlled lab study against both crowdsourced studies. For the A/B study, on average, lab participants took 17.69 s, μWorker 14.46 s, and Internet user 15.59 s per video. The rating study took a little more time, lab participants took 21.44 s, μWorker 17.71s, and Internet users 19.23 s per video. We found lab participants replay videos more often, especially in the A/B study. Regardless of group, faster networks resulted in more replays, which might already indicate that it is harder to spot differences. Regarding demographics, 76% to 79% were male. Within the Internet and the Lab group, the majority was younger than 24 years, for the μWorkers two-thirds were between the age of 25 to 44 years.

---

[3]Since even in the lab study people claimed to see a difference, we allowed left or right as a valid answer if the confidence was low.
[4]We cannot filters users with multiple μWorker accounts.
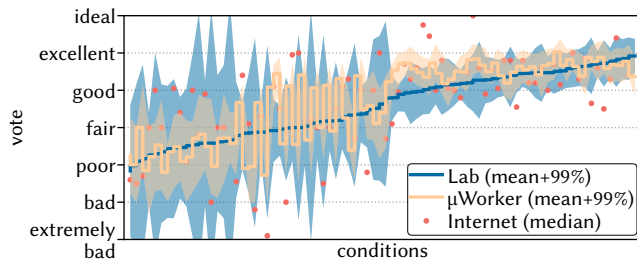
Figure 3: Rating study votes over all lab-tested conditions (ordered by mean vote of the lab participants). We show the 99% confidence interval. Internet values are not normally distributed, and thus, we show the median.

Figure 3 shows the agreement of all three groups in the rating study across all conditions on the x-axis (conditions in the rating study are single video and video pairs in the A/B study). The lab, as well as the µWorker data, is normally distributed. Thus, we show the mean and the 99% confidence interval (shared area) of the votes. We find that the µWorkers seem to fall mostly within the confidence intervals of the lab study, and hence, we believe that these votes are legit. In contrast, for Internet votes, we are unable to estimate the distribution and thus show the median of the votes. As is visible from the figure, the Internet group deviates most from the other two, and the number of votes that we were able to collect is lower. Consequently, we exclude it from further discussions, highlighting the challenge when trying to collect voluntary user data while setting a high standard on compliance with basic rules.

For the A/B study, in general, the agreement follows a similar scheme (not shown), but outliers look more severe due to the 3-point-scale (left, right, no diff.). We manually inspect the significant outliers in both studies, and we found that the websites are structurally very different. Figure 1 shows such a case, after loading the actual content, a welcome banner pops up. Participants in the lab study made their decision after the banner loaded (see left video) while people in the crowdsourced data seem to vote earlier according to when the actual website content is shown, and those decisions often do not agree across different protocol versions.

4.3 Do Users Notice a Difference?

To answer the questions if users actually notice a difference between the different protocol variants, we look at the A/B study and compare their votes. Figure 4 shows the share of votes for preferring a specific protocol variant in the four different network settings across all websites. The colors denote different pairs of protocols under comparison, the hatches signal preference for one or the other. Furthermore, we display the average replay count as vertical lines for each group of comparisons. So, e.g., in the DSL setting, slightly over 25% prefer TCP+, over 60% see no difference, and less than 10% prefer TCP, but on average, people replayed the video roughly 1.4 times. In general, we observe that the agreement for observing a difference rises when the networks become slower. For example, in the DSL setting, for all but the QUIC vs. TCP comparison (red), most participants do not see a difference. The comparably high average replay count expresses the difficulty of spotting a difference in the DSL network. Still, in general, more people experience the
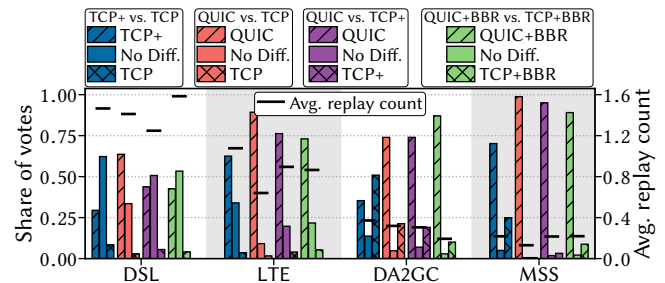
Figure 4: A/B study mean votes for each protocol combination depending on the network configuration.

protocol variant quicker that is supposed to be faster. Lowering the bandwidth towards the LTE setting, the majority of participants now clearly vote the supposedly better variant (this confidence is also backed by the lower replay counts). In the slower networks, there are slight differences. For DA2GC, TCP is now favored in contrast to our tuned variant (TCP+), we always found more re-transmissions for TCP+ (on avg. ×1.5 but up to ×4.8) which may be explained by the comparably high initial congestion window leading to early losses. In contrast, QUIC seems to not suffer from the same problems (even though similarly configured to TCP+) as our participants experience it faster, we suspect that QUIC's large SACK ranges enable it to progress further and that the independent stream processing allows earlier renderings of the page. Looking at the MSS network, the observation from DA2GC is now again reverted for TCP vs. TCP+; the increased bandwidth reassures our earlier assumption. For the other protocol variants, the picture from DA2GC is now even stronger with even more votes towards the supposedly faster variants. Again, the higher random loss-rate in this network backs our previous impression.

Takeaway. *In general, people do see a difference and indeed perceive QUIC as the faster protocol over TCP and even over a tuned TCP variant. However, in networks with high bandwidths, perceiving a difference seems to be more challenging.*

4.4 Do Users Even Care?

We continue to answer the second question, whether users care, or more specifically, we want to investigate if users perceive the already uncovered speedups as actually increasing the performance or if they cannot tell in isolation. To this end, we look at the results of the rating study, which we overview in Figure 5. In general, we observe that the work and free time scenario are rated similarly with a slight tendency towards better scores in the free time setting for DSL. In contrast is the plane setting (only having videos using the emulated in-flight networks) that shows only poor results.

When looking at the results within a network setting, we see only little variance between the different protocols, and the confidence intervals mostly overlap. When we test the different distributions for significance (using a significance level of 99% and an ANOVA test), we do not find any significant protocol/network configuration that the users rate better. When we lower the confidence level to 90%, three settings are significantly different. First, in the LTE free time setting, QUIC+BBR is rated statistically more satisfying than TCP+BBR. BBR again makes the difference in the plane environment
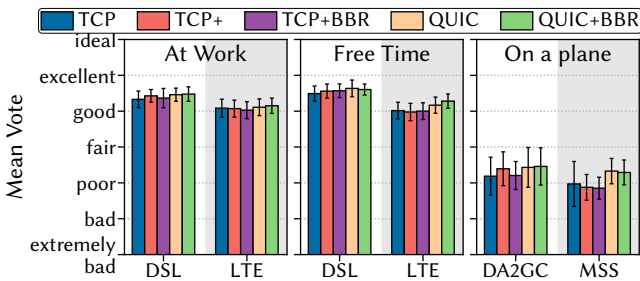
Figure 5: Rating study votes per protocol choice in the different settings for the μWorker group. Error bars show 99% confidence intervals.
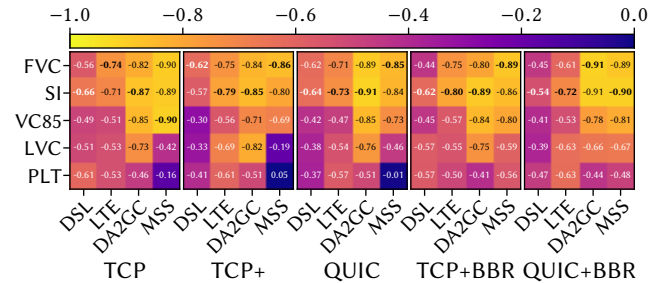
Figure 6: Pearson's correlation coefficient heatmap for different technical metrics to the user ratings in the different settings. High negative values (-1.0) are desired and mean that technical metric and user rating correlate perfectly. Largest correlation are in bold. For DSL/LTE, we chose the votes from free time setting.

and the MSS network. In the same setting, also QUIC without BBR is generally rated faster than TCP+. Thus, in general, there is little difference between the protocol variants. We now take a look at the specific websites where changes matter.

Where it Makes a Difference. In the DSL setting, eight websites show significant differences. Four of them rate QUIC faster than TCP, one faster than TCP+, and three rate QUIC+BBR faster than TCP+BBR. Spotify.com shows the largest difference with BBR. The website is small, but the browser has to contract many hosts. Still, we find small and also large websites that profit from QUIC.

In the LTE setting, only five pages show a significant difference. Our participants favor TCP+ over TCP once, otherwise QUIC over TCP+ (twice) and TCP (once). When using BBR, QUIC is favored once over TCP+. Regardless of congestion control, QUIC is roughly rated 10 points better, i.e., a whole quality level. Again, the websites show a wide variety of sizes and contacted hosts.

For DA2GC, we again find only five websites with a significant difference. Apache.org, a relatively small website in terms of size and resources, is preferred when delivered via QUIC in contrast to TCP and TCP+. When using BBR, google.com, gov.uk, and nature.com are perceived faster using QUIC. Lastly, w3.org is rated over 15 points faster when using QUIC in contrast to TCP+.

For MSS, we find three pages. Wordpress.com is favored in the QUIC variant over both TCP variants, a website with few resources, small in size, and less than ten contacted hosts. Gravatar.com on TCP is liked less. Apache.org is favored when BBR is used.

Correlation to Technical Metrics. We next investigate which technical metrics (FVC, SI, VC85, LVC, and PLT) best reflect our participants' ratings. To do so, we calculate the Pearson's correlation coefficient of the votes compared to the technical metrics by first calculating the mean vote for each website and combining it with the technical metric. We chose Pearson's (e.g., over Spearman) because we are interested to see how well the linearity of the metric reflects the users' choices. Thus, we would assume a high negative coefficient when high vote scores are correlated with low metric scores (i.e., a quick loading) and vice versa.

Figure 6 shows a heatmap for the different correlation coefficients in the different network settings subject to the different protocols. As the figure shows, in general, SI shows the largest correlation even though we find that in speedy networks such as DSL, the correlation goes down, and for slower networks, it goes up. However, this is not limited to SI but also holds for the other metrics, which seems

to be in line with earlier findings that showed larger confidence in these networks. Opposing SI, we find PLT to have the worst correlation to our users' ratings, thus reinforcing related works.

Takeaway. *In general, our participants did not care about the protocol when giving no direct comparison even though, in some cases, QUIC showed a small tendency to be preferred. When looking at individual websites, their size in terms of bytes and objects was not an obvious reason to prefer QUIC; only many contacted systems seem to point towards QUIC. In the challenged networks, we would have suspected that big websites would make a greater difference, but we found the opposite, which is likely due to the overall long loading times, where only small websites may show a perceivable deviation. Furthermore, we found that the Speed Index shows the highest correlation with our participants' votes.*

5 CONCLUSION

Motivated by related works that compare a highly-tuned QUIC Web stack against unoptimized TCP-based Web stacks, we performed two Quality of Experience studies to compare a state-of-the-art QUIC stack against tuned versions of TCP-based stacks. Our first study confirms related works that claim that QUIC indeed outperforms TCP-based stacks. Our user study shows that actual users can *perceive* these often small technical performance differences. Thus, users rate QUIC as faster. However, if network speeds increase, the difficulty of spotting a difference rises. In a second study, we investigate if people actually prefer the QUIC-delivered version of a website over TCP by showing them recordings of loading processes in isolation. Our results now indicate that the differences seem to become negligible, and users do not perceive one protocol as significantly faster; only in slower networks, people seem to tend towards preferring QUIC. Furthermore, our results support the use of the Speed Index as we found it to best correlate to user perception for the tested protocols and websites. In summary, our results show that QUIC, while radically evolving the transport layer, will *not* tremendously increase user satisfaction by just enabling it. While QUIC *is the future to evolve and enhance the transport layer*, switching today should not be purely motivated by performance, the enhanced privacy and its future proof design are what sets it apart and may give improved performance in the future.

ACKNOWLEDGMENTS

REFERENCES

[1] Prasenjeet Biswal and Omprakash Gnawali. 2016. Does QUIC Make the Web Faster?. In *IEEE Global Communications Conference (GLOBECOM)*. https://doi.org/10.1109/GLOCOM.2016.7841749

[2] 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 *Springer Passive and Active Measurement (PAM)*. https://doi.org/10.1007/978-3-319-54328-4_4

[3] Gaetano Carlucci, Luca De Cicco, and Saverio Mascolo. 2015. HTTP over UDP: An Experimental Investigation of QUIC. In *ACM Symposium on Applied Computing (SAC)*. https://doi.org/10.1145/2695664.2695706

[4] Sarah Cook, Bertrand Mathieu, Patrick Truong, and Isabelle Hamchaoui. 2017. QUIC: Better for what and for whom?. In *IEEE International Conference on Communications (ICC)*. https://doi.org/10.1109/ICC.2017.7997281

[5] Marc Fischlin and Felix Günther. 2017. Replay Attacks on Zero Round-Trip Time: The Case of the TLS 1.3 Handshake Candidates. In *IEEE European Symposium on Security and Privacy (EuroS&P)*. https://doi.org/10.1109/EuroSP.2017.18

[6] Dennis Guse, Henrique R. Orefice, Gabriel Reimers, and Oliver Hohlfeld. 2019. TheFragebogen: A Web Browser-based Questionnaire Framework for Scientific Research. In *IEEE Quality of Multimedia Experience (QoMEX)*. https://doi.org/10.1109/QoMEX.2019.8743231

[7] Tobias Hoßfeld, Christian Keimel, Matthias Hirth, Bruno Gardlo, Julian Habigt, Klaus Diepold, and Phuoc Tran-Gia. 2014. Best Practices for QoE Crowdtesting: QoE Assessment With Crowdsourcing. *IEEE Transactions on Multimedia* 16, 2 (2014), 541–558. https://doi.org/10.1109/TMM.2013.2291663

[8] ITU. 2003. *Subjective quality evaluation of telephone services based on spoken dialogue systems*. ITU-T Recommendation P.851. https://www.itu.int/rec/T-REC-P.851/en

[9] 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 *ACM Internet Measurement Conference (IMC)*. https://doi.org/10.1145/3131365.3131368

[10] Conor Kelton, Jihoon Ryoo, Aruna Balasubramanian, and Samir R. Das. 2017. Improving User Perceived Page Load Times Using Gaze. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/kelton

[11] Péter Megyesi, Zsolt Krämer, and Sándor Molnár. 2016. How quick is QUIC?. In *IEEE International Conference on Communications (ICC)*. https://doi.org/10.1109/ICC.2016.7510788

[12] Microworkers.com. 2019. The Microworker Platform. https://www.microworkers.com

[13] Ferdinand Mütsch. 2017. *Caddy - a modern web server (vs. nginx)*. https://ferdinand-muetsch.de/caddy-a-modern-web-server-vs-nginx.html

[14] Késsia Nepomuceno, Igor Nogueira de Oliveira, Rafael Roque Aschoff, Daniel Bezerra, Maria Silvia Ito, Wesley Melo, Djamel Sadok, and Géza Szabó. 2018. QUIC and TCP: A Performance Evaluation. In *IEEE Symposium on Computers and Communications (ISCC)*. https://doi.org/10.1109/ISCC.2018.8538687

[15] Ravi Netravali, Anirudh Sivaraman, Somak Das, Ameesh Goyal, Keith Winstein, James Mickens, and Hari Balakrishnan. 2015. Mahimahi: Accurate Record-and-Replay for HTTP. In *USENIX Annual Technical Conference (ATC)*. https://www.usenix.org/conference/atc15/technical-session/presentation/netravali.

[16] Mohammad Rajiullah, Andra Lutu, Ali Safari Khatouni, Mah-Rukh Fida, Marco Mellia, Anna Brunstrom, Ozgu Alay, Stefan Alfredsson, and Vincenzo Mancuso. 2019. Web Experience in Mobile Networks: Lessons from Two Million Page Visits. In *ACM The World Wide Web Conference (WWW)*. https://doi.org/10.1145/3308558.3313606

[17] John P. Rula, James Newman, Fabián E. Bustamante, Arash Molavi Kakhki, and David Choffnes. 2018. Mile High WiFi: A First Look At In-Flight Internet Connectivity. In *IW3C2 World Wide Web Conference (WWW)*. https://doi.org/10.1145/3178876.3186057

[18] Jan Rüth, Ingmar Poese, Christoph Dietzel, and Oliver Hohlfeld. 2018. A First Look at QUIC in the Wild. In *Springer Passive and Active Measurement (PAM)*. https://doi.org/10.1007/978-3-319-76481-8_19

[19] Michael Seufert, Raimund Schatz, Nikolas Wehner, and Pedro Casas. 2019. QUICker or not? - an Empirical Analysis of QUIC vs TCP for Video Streaming QoE Provisioning. In *IEEE Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*. https://doi.org/10.1109/ICIN.2019.8685913

[20] Michael Seufert, Raimund Schatz, Nikolas Wehner, Bruno Gardlo, and Pedro Casas. 2019. Is QUIC becoming the New TCP? On the Potential Impact of a New Protocol on Networked Multimedia QoE. In *IEEE International Conference on Quality of Multimedia Experience (QoMEX)*. https://doi.org/10.1109/QoMEX.2019.8743223

[21] sitespeed.io. 2019. Browsertime - Your browser, your page, your scripts! https://github.com/sitespeedio/browsertime.

[22] Nick Sullivan. 2017. Introducing Zero Round Trip Time Resumption (0-RTT). https://blog.cloudflare.com/introducing-0-rtt/

[23] Maarten Wijnants, Robin Marx, Peter Quax, and Wim Lamotte. 2018. HTTP/2 Prioritization and Its Impact on Web Performance. In *IW3C2 World Wide Web Conference (WWW)*. https://doi.org/10.1145/3178876.3186181

[24] Konrad Wolsing, Jan Rüth, Klaus Wehrle, and Oliver Hohlfeld. 2019. A Performance Perspective on Web Optimized Protocol Stacks: TCP+TLS+HTTP/2 vs. QUIC. In *ACM Applied Networking Research Workshop (ANRW)*. https://doi.org/10.1145/3340301.3341123

[25] Yajun Yu, Mingwei Xu, and Yuan Yang. 2017. When QUIC meets TCP: An Experimental Study. In *IEEE International Performance Computing and Communications Conference (IPCCC)*. https://doi.org/10.1109/PCCC.2017.8280429

[26] zafaco GmbH. 2018. Breitbandmessung Ergebnisse als interaktive Darstellung. https://web.archive.org/web/20181115/https://breitbandmessung.de/interaktive-darstellung

[27] Torsten Zimmermann, Benedikt Wolters, and Oliver Hohlfeld. 2017. A QoE Perspective on HTTP/2 Server Push. In *ACM Workshop on QoE-based Analysis and Management of Data Communication Networks (Internet-QoE)*. https://doi.org/10.1145/3098603.3098604