Scaling Up Your Web Experience, Everywhere

James Newman
Northwestern University

Robert H. Belson
Northwestern University

Fabián E. Bustamante
Northwestern University

ABSTRACT

We present an approach to improve users' web experience by dynamically reducing the complexity of websites rendered based on network conditions. Our approach is based on a simple insight – adjusting a browser window's scale (i.e., zooming in/out), changes the number of objects placed above-the-fold and thus hides the loading of objects pushed below the fold in the user scroll time.

We design *ScaleUp*, a browser extension that tracks network conditions and adjusts browser scale appropriately to improve user web Quality of Experience (QoE) while preserving the design integrity of websites. Through control experiments, we demonstrate the impact of *ScaleUp* on a number of key QoE metrics over a random sample of 50 from the top 500 Alexa websites. We show that a simple adjustment in scale can result in an over 19% improvement on Above-The-Fold (ATF) time in the median case. While adjusting a scale factor can improve metrics of QoE, it is unclear if that translates into an improved web experience for users. We summarize findings from a large, crowdsourced experiment with 1,000 users showing that, indeed, improvement to QoE metrics correlate with an enhanced user experience. We have released *ScaleUp* as a Chrome Extension that now counts with over 1,000 users worldwide, and report on some of the lessons learned from this deployment.

ACM Reference Format:
James Newman, Robert H. Belson, and Fabián E. Bustamante. 2019. Scaling Up Your Web Experience, Everywhere. In *The 20th International Workshop on Mobile Computing Systems and Applications (HotMobile '19), February 27–28, 2019, Santa Cruz, CA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3301293.3302368

1 INTRODUCTION

Websites have become increasingly complex with hundreds of different objects — images, videos, scripts, and cascading stylesheet (CSS) files [9]. Besides creating the opportunity for more appealing websites, the additional complexity can slow down overall page loading time and negatively impact users and content providers alike. Changes in users' experiences have been linked to changes in revenue for content providers, especially for commerce websites [5, 8, 14].

For many users, the trend has gone mostly unnoticed with the concurrent growth in the capacity of Internet services. In the first quarter of 2017, the global average connection speed increased 15% compared with the first quarter of 2016, and the top ten countries reported an average connection speed of 22.5 Mbps [2].

For others, however, either in regions with poor connectivity or temporarily accessing the web through low-end networks, this trend has resulted in significantly worsened web experiences. A common modern setting is in-flight connectivity (IFC), where latencies can be 10-100x higher than many residential networks, and packet loss ratios can reach up to 30%, resulting in page load times over 30 seconds even for the top 100 most popular sites [24].

While designing website for different devices (e.g., desktops and mobile) can help, this is done on a per content provider and website basis. Understanding how to improve the web experience of users everywhere and across websites, remains a open problem. Some efforts look to optimize pages based on dependency tracking [19, 22], others rewrite pages for better load times [10, 28], and some rely on proxy-based solutions [21, 25]. While promising, most of these projects require either dedicated infrastructure or prior runs of websites to affect positive change.

We present an approach to improve users' web experience by dynamically reducing the complexity of the websites rendered based on network conditions. Our approach is based on a simple insight – adjusting a browser window scale (i.e., zooming in/out), changes the number of objects in a website that are placed above the fold and thus a user's experience with the website. Importantly, we do not change the content on websites, rather, we simply force some of its placement to optimize user experience. As some of the content is pushed below the fold, the time it takes to load that content is hidden within the user's scroll time. This plays on the user's understanding of how webpages load, given the impression of a webpage loading faster by showing them a smaller portion of the webpage at a time.

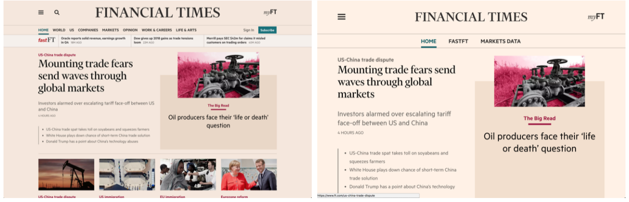
We design *ScaleUp*, a browser extension that tracks network conditions to appropriately and dynamically adjust browser scale to improve user's web QoE without sacrificing the design integrity of websites. *ScaleUp* utilizes the Chrome *webRequest* API [11] – the same API used by popular web-based ad-blockers – to measure how quickly the server responds with the initial HTML page, and determines, at run-time, the most appropriate scale factor for the site (§2).

Through control experiments, we evaluated the impact of *ScaleUp* on two key QoE metrics using a random sample of 50 from the top 500 Alexa websites. We show that, a simple adjustment in scale factor can result in a 19% improvement on Above-the-Fold time [18] in the average case (§3).

While adjusting a scale factor can improve some metrics of QoE, it is unclear if that translates into an improved web experience for users. We present results from a large, crowdsourced experiment with 1,000 users showing that, indeed, these improvements are correlated with actual user experience (§4).

HotMobile '19, February 27–28, 2019, Santa Cruz, CA, USA

ACM ISBN 978-1-4503-6273-3/19/02…$15.00
https://doi.org/10.1145/3301293.3302368



FINANCIAL TIMES

HOME WORLD US COMPANIES MARKETS OPINION WORK & CAREERS LIFE & ARTS Sign In Subscribe

fastFT Oracle reports solid revenue, earnings growth in Q4 18M AGO | Dow gives up 2018 gains as trade tensions loom 22M AGO | Merrill pays SEC $42m for claims it misled customers on trading orders 40M AGO

US-China trade dispute

Mounting trade fears send waves through global markets

Investors alarmed over escalating tariff face-off between US and China

4 HOURS AGO

- US-China trade spat takes toll on soyabeans and squeezes farmers
- White House plays down chance of short-term China trade solution
- Donald Trump has a point about China's technology abuses

The Big Read

Oil producers face their 'life or death' question

US-China trade dispute | US immigration | EU immigration | Eurozone reform

(a) Unscaled

FINANCIAL TIMES

HOME FASTFT MARKETS DATA

US-China trade dispute

Mounting trade fears send waves through global markets

Investors alarmed over escalating tariff face-off between US and China

4 HOURS AGO

- US-China trade spat takes toll on soyabeans and squeezes farmers
- White House plays down chance of short-term China trade solution
- Donald Trump has a point about China's technology

The Big Read

Oil producers face their 'life or death' question

https://www.ft.com/us-china-trade-dispute

(b) Scaled

Figure 1: The visible impact of scale factors. A popular website rendered un-scaled, on the left, and at 1.5x scale on the right; the higher scale factor results in fewer objects loaded above-the-fold.

We have released *ScaleUp* as a Chrome Extension that now counts with over 1,000 users worldwide; we report on some of the lessons learned from this deployment.

2 SCALING UP WEB QOE

Addressing the impact of increasingly complex websites on users' experience over low-end connections is challenging. For starters, since users visit a wide range of sites during a day, an ideal solution should improve users experience across all sites. While blocking objects outright would seem to be a promising approach (e.g., [4]), a better solution would preserve the integrity of the website design. Finally, not all low-end connections are alike and users move between networks of different quality through their daily lives. Any solution should dynamically adapt to the capability of the current network.

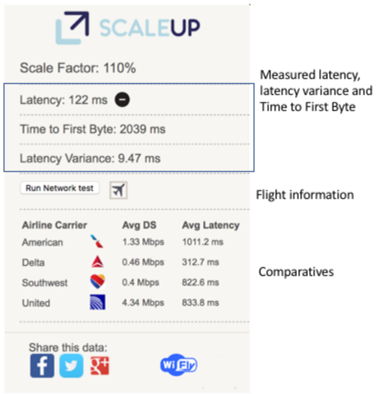
Our approach is simple: dynamically adapt the complexity of the website rendered above the fold. We do this by adjusting a browser window scale (zooming in/out) based on measured network conditions. Effectively, by *zooming in* when network conditions worsen, we change the size of text and images, and thus the number of objects placed above-the-fold, hiding the loading of objects pushed below the fold in the user scroll time. As network conditions improve, *zooming out* lets the user enjoy the full content of a website.

Figure 1 illustrates this with a popular news website, loaded at two different scale factors, 100% and 150%. It is apparent that when the website is loaded with a larger scale factor there are fewer objects visible above-the-fold (ATF) and thus a lower ATF time and – potentially – a better user experience.

It may seem that the scaling mainly impacts images. While this is true, it doesn't fully cover the extent of the impact. Because the browser utilizes the viewport size in ordering resources to render, the zoom factor can impact which objects are rendered first, affecting how CSS rules are implemented and how iFrames (HTML) are loaded.

2.1 *ScaleUp* Design

We implemented our idea as a Chrome browser extension – *ScaleUp* – that tracks network conditions and dynamically adjusts browser scale to improve web QoE while preserving the design integrity of websites. *ScaleUp* relies on the Chrome webRequest API [11] to determine whether or not to scale a website. By utilizing a combination of callbacks revealed by the API, *ScaleUp* measures the time it takes to receive the headers of the initial HTML page. If that time is over certain thresholds, *ScaleUp* will adjust the scale factor of the browser. If latency measurements drop, perhaps after moving to a different network, the extension will adjust the scale again.[1]



SCALEUP

Scale Factor: 110%

Latency: 122 ms

Time to First Byte: 2039 ms

Latency Variance: 9.47 ms

Run Network test

Airline Carrier	Avg DS	Avg Latency
American	1.33 Mbps	1011.2 ms
Delta	0.46 Mbps	312.7 ms
Southwest	0.4 Mbps	822.6 ms
United	4.34 Mbps	833.8 ms

Share this data:

Measured latency, latency variance and Time to First Byte

Flight information

Comparatives

Figure 2: User Interface of *ScaleUp*. The interface provides information on network conditions, such as latency and an approximation of time to first byte. In this example run (loaded over a slow 3G connection), the time to first byte is over 100ms and the website would have been scaled.

To scale the website, *ScaleUp* uses the *chrome.tabs* API [12]. This API allows the extension to interact with and modify the tabs open in a user's browser. In particular, it reveals the *setZoom* function[2] to change the zoom of the tab by a certain scale factor.

Additionally, *ScaleUp* captures the ATF time of the website. To do this, it relies upon the ability of Chrome extensions to inject JavaScript onto websites that the user visits, coined *content scripts* [13]. We declare the script in the manifest file to be included on each new website that is loaded.

ScaleUp injects a modified Above-the-Fold (ATF) script [26] that measures the ATF time for the website on which it is included. ATF time refers to the time it takes for all content within the initial browser's viewport (i.e., no scrolling) to be loaded [18]. The script traverses the DOM tree of the page and records whether or not an object is above-the-fold based on the coordinates of the object and the viewport size. The script measures the load time of each object and then computes the time it takes for all the content ATF to load.

Finally, *ScaleUp* collects data on network conditions, scale factors, and QoE metrics for further analysis. For this it includes a JavaScript file to be executed when the page has completed loaded. We use the *chrome.runtime* API to communicate with *ScaleUp*'s background page. Once the ATF script has completed calculating the ATF time, we send the resulting times back to the background page which then reports it to our servers for storage. *No information is collected on the actual sites a user visits.*

In total, *ScaleUp* is 1,152 lines of original JavaScript code, and currently comprises 22MB of storage when counting other libraries it relies on, images, CSS, and HTML. *ScaleUp* was originally released

[1] We are exploring a combination of latency and website specific rules for the next version of the plugin.

[2] `chrome.tabs.setZoom(integer tabId, double zoomFactor, function callback)`

to the Chrome Web Store on February 15th 2017, and has since gone through 14 versions, reaching over 1,000 unique, persistent users.

Limitations. We designed and implemented *ScaleUp* as an extension to a desktop browser. As part of future work, we plan to explore ways to implements some of these ideas on mobile devices for which some websites may already be optimized, thus limiting the benefits of scaling, and where browsers typically lack support for extensions.

3 IMPACT OF SCALE FACTOR

In this section, we use control experiments to demonstrate the impact of adjusting the browser window scale on a number of key QoE metrics. For this, we focus on 50 random websites selected from the Alexa top 500.

To characterize the impact of scaling, we load each of these 50 websites for each of the scale factors considered: 90%, 100%, 110%, 125%, and 150%, multiple times. We use – as *ScaleUp* does – the *chrome.tabs* API to set up the scale, load the page, and collect several measurements using the extension Approximate ATF [3], including the number of different objects on a website, the number of those objects residing ATF, and the ATF time.

The machine running the experiments is a MacBook Pro with 16 GB of 1600MHz memory and an Intel Core i7 processor. It is connected to the Internet through a router that we controlled to minimize interference [15]. We use a TP-Link N750 Wireless Wi-Fi Dual Band router (TL-WDR4300) which we configure to run OpenWRT (Chaos Calmer 15.05.1, r48532) and the Linux's Traffic Control and Network Emulation tools, which we use to emulate the network conditions found in-flight. We configure the connection to mimic a cellular-based IFC connection as reported by Rula et al. [24]. This can be seen as a best-case for in-flight communications in terms of latency, found to be about 2x smaller than the alternative satellite connections.

Images Above-the-Fold. Figure 3 plots the changes in the number of images ATF as a function of the scale factor. As expected, even a small change in scale can result in a significant number of images being move bellow the fold (from 8 in the average case at the 90% scale factor to 4.6 at 150%). The ticks show the standard deviation on the number of images which also becomes smaller with larger scale factors.

Above-the-Fold Time. Our hypothesis is that as the average number of images drop at higher scales, the ATF time will also drop. Table 1 summarizes our results, showing the 50th, 90th and 95th percentile of both ATF and Page Loading Time (PLT) for different scale factors. While scaling has little to no impact on PLT, ATF times drops significantly with up to 25.2% in the average case and up to 44% at the 95% percentile.

In some respect, the lack of change noticed in PLT from scaling is the insight behind the *ScaleUp* approach. While no browser extension can improve the underlying network performance (e.g., latency, loss), it can improve the *perceived* performance as captured by the ATF time.

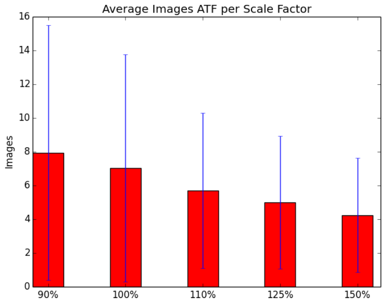


Figure 3: Average number of images that reside above-the-fold for different scale factors. As the scale factor increases, the corresponding number of images decreases as images are pushed below-the-fold.

Scale Factor	Percentile	ATF Time (ms)	PLT (s)
90%	Average	1,134.5	11.3
	90	1,536.8	20.5
	95	5,760.7	24.2
100%	Average	960.98	10.7
	90	939.53	20.4
	95	3,950.7	23.7
110%	Average	777.34	9.7
	90	964.2	16.9
	95	1,086.1	23.1
125%	Average	925.54	9.8
	90	1,537.0	18.0
	95	3,491.9	18.7
150%	Average	848.51	9.5
	90	1,317.4	18.8
	95	3,235.3	21.6

Table 1: Results from controlled experiment. We see that *ScaleUp* can improve ATF time 19% by increasing the scale factor from 100% to 110%. While there is not a significant impact on PLT, *ScaleUp* still improves the average load time by 1 second with only a 10% change in scale.

4 SCALE AND USER EXPERIENCE

The previous paragraphs demonstrate the impact of browser scale on key QoE metrics. In this section we present a crowdsourced user study exploring the effect of different scale factors on actual user quality of experience.

4.1 Methodology

We conducted a user study with a 1,000 participants in Amazon Mechanical Turk (AMT). The study focuses on capturing the impact of scaling on users' experience with the rendered website – including both how quickly the site loads as well as the potential impact of zooming on its aesthetics.

Each survey starts by collecting some basic demographic information, followed by the presentation of a series of websites at different scales. Participants are asked questions aimed at ranking the alternatives based on perceived load speed and appearance. The survey ends with some basic exit questions to understand what drove the user's selection of the fastest/slowest alternative and a free-text section on the factors determining what "looked best".

In more detail, the initial questionnaire includes gender, age range, and country, as well as their self-rated technical proficiency and the range of hours typically spent online. We use these to evaluate the role, if any, that such factors may have in their web browsing experience.

In the main part of the survey, we asked participants to view three simultaneous loads of the same website with different scales. We want to see how the changes measured in PLT and ATF, as a result of changes in scale, relate to user experience. We do this for 5 different sites randomly selected from the same set of top Alexa sites as in our control experiment. The versions were loaded with three scale factors — *90%*, *100%*, and *110%*. Although much larger scale factors are possible, we conservatively used the smallest possible scale factor;[3] changes in perception should be only more clear with large scales.

As in Varvello et al. [27], rather than using live sites during an experiment, we collect videos of the websites loading and use this to control for network and website variations between runs.

We provide users with three questions for each set of videos: *Which website loaded fastest?*, *Which website loaded slowest?*, and *Which website looked the best while loading?* The first two questions aim to capture perceived improvements on load time. The last one is used to determine how, if at all, the scale factor effects the aesthetics of the website.

Quality Control. When submitting a Human Intelligence Task (HIT), we limit it to users with >=95% lifetime approval rating and over 50 lifetime submitted tasks. We do this to ensure the reliability of our participants. Of the 5,000 ratings (1,000 HITs with 5 websites each) we received 4,597 (≈ 92%) results. Beyond the 5 websites, we also include one other website as a control case. The control case randomly places a "fast" website load against two identically slow ones. We use this as an additional form of quality control on all of the HITs. Of the 944 surveys who scored all the video sets, 850 of them passed the control test. We eliminated the 106 surveys from users who failed the test.

Ethical Considerations. Amazon's conditions of use explicitly prohibit tasks that gather personally identifiable information. The information we did collect is coarse enough that we have no reasonable way to map it to individuals. Our experiments collect data "about what", rather than "about whom", through the relatively innocuous task of selecting videos. Our institution's Institutional Review Board (IRB) determined our study to be *Not Human Research.*

4.2 Summary of Results

The following paragraphs summarize the results from our crowdsourced user experiment, starting with a characterization of participating users based on demographics and technical proficiency.

Demographics. We find that a majority of users are male (59.5%), and between between 25-31 years of age. The large majority comes from the United States (77.4%) followed by India (18.1%); with the rest being from a variety of countries.

Question	**Scale Factor**		
	90%	*100%*	*110%*
Loaded Fastest	30.1%	33%	**36%**
Loaded Slowest	**39.1%**	30.6%	30.3%
Looked the Best	33.2%	29.7%	**37.1%**

Table 2: Results from a crowdsourced study shows a clear correlation between scale factor and users' perceived loading time. Interestingly, the trend breaks when users are asked which website looks the best.

Focusing on the average number of hours spent online, we see a normal distribution with the majority of users, ≈36%, in the *4-8 hour* range. Meanwhile, few users selected either *<1 hour* or *>12 hours*, 0.5% and 8.8% respectively.

On the level of technical proficiency, 53.2% of users self-identified as *average* and 43.2% identified as *above average.* Only 3.6% of users selected the *below average* category.

Scale and QoE. Table 2 presents the percentage of users selecting, for different scale factors, the one that loaded the fastest, the slowest and that looked the best. Users selected *110%* scale as loading fastest 36% of the time and chose *90%* as loading the slowest 39.1% of the time. Even a small adjustment in scale can produce a 3% change of perceived quality of experience.

When users were asked which website "looked" the best, we notice an interestingly different trend from what we expected. The default, *100%* scale factor, was most commonly ranked at the bottom (29.7%). The option with highest scale factor, *110%*, was the most commonly selected as looking the best (37.1%). While the higher percentage of users for 110% could be explained by users looking for "larger" or "more clear" content, it is possible that the *90%* scale factor was popular because users could see more content. We leave the exploring of scale factors on users' perceived appearance as future work.

5 *SCALEUP* DEPLOYMENT

ScaleUp has been available in the Chrome Web Store for over a year. At the time of writing this, *ScaleUp* has 1,031 users actively checking for updates. Additionally, while this number is fairly consistent, there is some churn in our user base with an average of 8.7 installs and about 4.8 uninstalls per day. Figure 4 plots the changes in our user base over time (from February 2017 to June 2018), illustrating this.

Beyond improving users' web experience, *ScaleUp* serves as a crowdsource measurement tool for network performance and web quality of experience for in-flight WiFi users, its original customer base. As such, the extension collects bandwidth, latency, above-the-fold time, and page-load time, among other metrics. In addition, *ScaleUp* let users report their flight number in exchange for an update on their flight progress and an updated comparison of their network performance to that of other flights.

6 RELATED WORK

There have been numerous efforts to better characterize and improve web browsing performance and experience. Bocchi et al.

[3] Chrome does not allow for intermediate values, such as 99%.

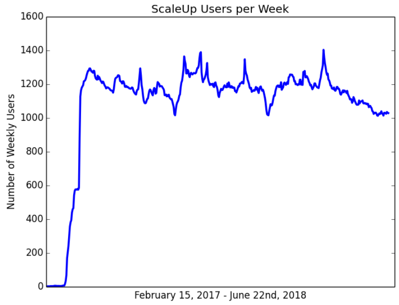


Figure 4: Number of users per week for the entirety of *ScaleUp* 's deployment. After the initial jump to over 1,000 users, the user base has remained relatively stable. This is encouraging and indicates that users enjoy browsing with *ScaleUp*

published a comprehensive survey of the available perceived WebQoE measurement systems, including Google's SpeedIndexTest, Yahoo's YSlow, and dynaTrace, among others [7]. To improve web QoE, some recent work relies on proxies to aid with handling dependencies [19, 22] or precomputing the information needed for page loading by a mobile browser [20, 25], while others have examined how protocol usage can impact user experience [1, 6].

Some close related efforts have explored ways to improve the "perceived" performance of websites, by restructuring websites [28], re-prioritizing content on pages [10] and even using users' gaze to understand what to prioritize on a webpage [16]. Compared with some of the proposed ideas, our approach is easier to deploy and adopt as it does not require any additional infrastructure and works with any site.

While generally applicable to any high-latency network condition, our work was motivated by the challenging environment of inflight communication. Recent efforts on characterizing IFC [23, 24] have shown some of these challenges and made the case that many of them would persist even with the promising technology advances in the space. Not surprisingly, there has been interest from IFC providers in improving web browsing experience. For example, ViaSat filed a patent on a method of progressive prefetching to help page load times while browsing on satellite connections [17].

7 CONCLUSION

We presented a lightweight approach to improve users' web experience by dynamically reducing the complexity of websites rendered. Through control experiments, we demonstrated the impact of *ScaleUp* on a number of key QoE metrics and random set of popular websites. We presented results from a crowdsourced experiment with 1,000 users showing that the measured improvement on metrics of QoE correlate with an enhanced user experience. *ScaleUp* has been available as a Chrome Extension for over a year and counts with over 1,000 consistent users worldwide.

There are a number of interesting directions for future work. In our design of *ScaleUp* we have intentionally avoided direct blocking of objects as not to compromise the website design. It may be worth relaxing this constraint and consider the impact on user experience of limiting *font*, css or javascript objects. More generally, this work points to the need for better, deeper understanding of users' web QoE and the most reliable proxy metrics.

ACKNOWLEDGEMENTS

We thank our shepherd Ben Greenstein and the anonymous reviewers for their thoughtful feedback. This research was partially supported by NSF grant CNS-1619317 and a Google Faculty Research Award.

REFERENCES

[1] V. Agababov, M. Buettner, V. Chudnovsky, M. Cogan, B. Greenstein, S. McDaniel, M. Piatek, S. Colin, M. Welsh, and B. Yin. Flywheel: Google's data compression proxy for the mobile web. In *Proc. of USENIX NSDI*, 2015.
[2] Akamai. akamai's [state of the internet]: Q1 2017 report. https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/q1-2017-state-of-the-internet-connectivity-report.pdf, 2017. [Online; accessed 14-June-2018].
[3] A. Alemnew, V. Christophides, R. Teixeira, D. Rossi, et al. Narrowing the gap between qos metrics and web qoe using above-the-fold metrics. In *Proc. of PAM*, 2018.
[4] S. C. Apps. Images on/off. http://singleclickapps.com/images-on-off/, 2016. Accessed on 06.21.2018.
[5] N. Bhatti, A. Bouch, and A. Kuchinsky. Integrating user-perceived quality into web server design. *Computer Networks*, 33(1-6):1–16, 2000.
[6] E. Bocchi, L. D. Cicco, M. Mellia, and D. Rossi. The web, the users, and the mos: Influence of http/2 on user experience. In *Proc. of PAM*, 2017.
[7] E. Bocchi, L. D. Cicco, and D. Rossi. Measuring the quality of experience of web users. *SIGCOMM Comput. Commun. Rev.*, 4(4):8–13, 2016.
[8] A. Bouch, A. Kuchinsky, and N. Bhatti. Quality is in the eye of the beholder: meeting users' requirements for Internet quality of service. In *Proc. of SIGCHI*, 2000.
[9] M. Butkiewicz, H. V. Madhyastha, and V. Sekar. Understanding website complexity: Measurements, metrics and implications. In *Proc. of IMC*, 2011.
[10] M. Butkiewicz, D. Wang, Z. Wu, H. V. Madhyastha, and V. Sekar. KLOTSKI: reprioritizing web content to improve user experience on mobile devices. In *Proc. of USENIX NSDI*, 2015.
[11] Chrome. Chrome webrequest api. https://developer.chrome.com/extensions/webRequest. Accessed on 06.14.2018.
[12] Chrome. chrome.tabs. https://developer.chrome.com/extensions/tabs. Accessed on 06.14.2018.
[13] Chrome. Content scripts. https://developer.chrome.com/extensions/content_scripts. Accessed on 06.14.2018.
[14] K. Eaton. How one second could cost amazon 1.6 billion in sales. https://www.fastcompany.com/1825005/how-one-second-could-cost-amazon-16-billion-sales/, 2012. [Online; accessed 14-June-2018].
[15] A. M. Kakhki, S. Jero, D. Choffnes, C. Nita-Rotaru, and A. Mislove. Take a long look at quic. In *Proc. of IMC*, 2017.
[16] C. Kelton, J. Ryoo, A. Balasubramanian, and S. R. Das. Improving user perceived page load times using gaze. In *Proc. of USENIX NSDI*, 2017.
[17] P. Lepeska and W. B. Sebastian. Progressive prefetching, Mar. 6 2018. US Patent 9,912,718.
[18] P. Meenan. How fast is your website? *Commun. ACM*, 56(4):49–55, 2013.
[19] R. Netravali, A. Goyal, J. Mickens, and H. Balakrishnan. Polaris: Faster page loads using fine-grained dependency tracking. In *Proc. of USENIX NSDI*, 2016.
[20] R. Netravali and J. Mickens. Prophecy: Accelerating mobile page loads using final-state write logs. In *Proc. of USENIX NSDI*, 2018.
[21] R. Netravali, A. Sivaraman, S. Das, A. Goyal, K. Winstein, J. Mickens, and H. Balakrishnan. Mahimahi: Accurate record-and-replay for HTTP. In *Proc. USENIX ATC*, 2015.
[22] V. Ruamviboonsuk, R. Netravali, M. Uluyol, and H. V. Madhyastha. Vroom: Accelerating the mobile web with server-aided dependency resolution. In *Proc. of ACM SIGCOMM*, 2017.
[23] J. P. Rula, F. E. Bustamante, and D. R. Choffnes. When ips fly: A case for redefining airline communication. In *Proc. of HotMobile*, 2016.
[24] J. P. Rula, J. Newman, F. E. Bustamante, A. M. Kakhki, and D. R. Choffnes. Mile high wifi: A first look at in-flight internet connectivity. In *Proc. WWW*, 2018.
[25] A. Sivakumar, S. P. N., V. Gopalakrishnan, S. Lee, S. Rao, and S. Sen. Parcel: Proxy assisted browsing in cellular networks for energy and latency reduction. In *Proc. ACM CoNEXT*, 2014.

[26] Telecom. Atf-chrome-plugin. https://github.com/TeamRossi/ATF-chrome-plugin. Accessed on 06.14.2018.

[27] M. Varvello, J. Blackburn, D. Naylor, and K. Papagiannaki. Eyeorg: A platform for crowdsourcing web quality of experience measurements. In *Proc. ACM CoNEXT*, 2016.

[28] S. X. Wang, A. Krishnamurthy, and D. Wetherall. Speeding up web page loads with shandian. In *Proc. of USENIX NSDI*, 2016.