Measuring the Quality of Experience of Web users

Enrico Bocchi
Telecom ParisTech
enrico.bocchi@telecom-paristech.fr

Luca De Cicco
Politecnico di Bari
luca.decicco@poliba.it

Dario Rossi
Telecom ParisTech
dario.rossi@telecom-paristech.fr

ABSTRACT

Measuring quality of Web users experience (WebQoE) faces the following trade-off. On the one hand, current practice is to resort to metrics, such as the document completion time (onLoad), that are simple to measure though knowingly inaccurate. On the other hand, there are metrics, like Google's SpeedIndex, that are better correlated with the actual user experience, but are quite complex to evaluate and, as such, relegated to lab experiments.

In this paper, we first provide a comprehensive state of the art on the metrics and tools available for WebQoE assessment. We then apply these metrics to a representative dataset (the Alexa top-100 webpages) to better illustrate their similarities, differences, advantages, and limitations. We next introduce novel metrics, inspired by Google's SpeedIndex, that offer significant advantages in terms of computational complexity, while maintaining a high correlation with the SpeedIndex. These properties make our proposed metrics highly relevant and of practical use.

CCS Concepts

•**Information systems** → **World Wide Web;** •**Networks** → **Network measurement;** *Application layer protocols; Network performance analysis;*

Keywords

Quality of Experience; Web; DOM; onLoad; TTFB; TTFP; Above-the-fold; SpeedIndex; ByteIndex; ObjectIndex; MOS

1. INTRODUCTION

In the Internet, users consume contents primarily on the Web. The browser has become the platform through which a plethora of services can be accessed, including search, productivity, entertainment, social and personal communications, etc. At times of important evolutions – from HTTP/1 to HTTP/2, SPDY, and QUIC – having reliable ways to compare protocol performance becomes crucial before massive deployments can take place [14].

A number of studies have pointed out the importance of *delay* and of its direct relationship to the value of business. For instance, Amazon and Google report losses in the 0.6-1.2% range for delay increasing by 0.4-1 sec[1], while Shopzilla[2] reported +12% revenue for a 5 sec onLoad reduction achieved with a major site redesign.

The hidden correlation between these factors is the impact that delay has on the Quality of Experience of Web users (WebQoE): the higher the delay, the lower the WebQoE, the worse the experience, the higher the likelihood of user disengagement, the larger the economic losses. While the existence of a relationship between delay and WebQoE is beyond any doubt, it is more difficult to precisely pinpoint the delay of "which" event is the most important during the lifetime of a webpage, and to furthermore map it to a quantized quality level (≈MOS). Indeed, webpages have grown to quite complex entities including hundreds of objects that are fetched opening tens of connections directed to multiple domains. Requests for such objects are often dynamically generated by JavaScripts (or related technologies) executed as part of the page construction process.

It thus appears obvious that *no single event* – from the time at which the first byte is received (TTFB), to the parsing of the Document Object Model (DOM), to the completion of the full page (onLoad) – can express all sort of intricate dependencies [18] between the rendering process and the user experience. As such, there have been proposals for new metrics that are better suited to capture the actual quality of experience of Web users, such as Above-the-fold [5] and SpeedIndex [8], which have both been introduced by Google in 2011 and 2012, respectively. The SpeedIndex is particularly interesting as it explicitly takes into account the delay of *all events* in a webpage lifetime, but has been limitedly used so far due to its computational complexity.

As a result, the state of the art in today's WebQoE evaluation, both in research [7, 17, 18, 19] and in industry [4, 9], is to express QoE via the page completion time, i.e., onLoad. For instance, Alexa [4] reports the onLoad and directly exposes quantiles of the delay, while Google uses onLoad to rank search results [9], although with a small weight [10].

Our contributions are as follows. We first provide a complete taxonomy of the existing WebQoE metrics and tools, and introduce our proposed generalization of the SpeedIndex. We particularize two new metrics providing a significant breakthrough by retaining an appealing simplicity without losing semantic relevance (Sec. 2). We next present a comprehensive illustration of all WebQoE met-

This paper was originally presented at the ACM SIGCOMM Workshop on QoE-based Analysis and Management of Data Communication Networks (Internet-QoE 2016), August 26th 2016, Florianopolis, Brazil. DOI: 10.1145/2940136.2940138.

[1] http://www.fastcompany.com/1825005/how-one-second-could-cost-amazon-16-billion-sales
[2] http://radar.oreilly.com/2009/07/velocity-making-your-site-fast.html

Table 1: Metrics to express user perceived quality

	Metric name	Layer 3	Layer 4	Layer 7	Unit/ Range	Description
Time Instant	TTFB	$\approx$	✓	✓	sec	Time at which the first byte of payload is received
	DOM	-	-	✓	sec	Time at which the Document Object Model (DOM) is loaded
	TTFP	-	-	✓	sec	Time at which the first object is painted
	onLoad	-	-	✓	sec	Time at which all bytes of payload have been received
	ATF [5]	-	-	✓	sec	Time at which the content "Above-the-fold" has been rendered
Time Integral	ByteIndex	$\approx$	$\approx$	✓	sec	Integral of complementary byte-level completion
	ObjectIndex	-	-	✓	sec	Integral of complementary object-level completion
	SpeedIndex [8]	-	-	✓	sec	Integral of complementary visual progress
Comp. Score	YSlow [21]	-	-	✓	[0,100]	Yahoo's compound score (23 weighted heuristics)
	PageSpeed [12]	-	-	✓	[0,100]	Google's PageSpeed Insight heuristics
	dynaTrace [3]	-	-	✓	[0,100]	dynaTrace's compound score
	MOS	-	-	-	[1,5]	User rating

rics on the top-100 Alexa webpages, further elucidating relationships among metrics. Notably, we show that (i) an indetermination principle emerges when evaluating the SpeedIndex as its computation alters the very same experiment and that (ii) our proposed metrics remain highly correlated to the SpeedIndex despite their simplicity (Sec. 3). Finally, we discuss further generalizations of these metrics that would allow one to embed psycho-behavioral models of user perception at limited cost (Sec. 4).

2. WebQoE METRICS

Tab. 1 reports the most prominent metrics to measure WebQoE. In particular, the table groups metrics in four categories. 1 **Time-Instant metrics**, which are computed by measuring the time instant a particular event occurs. 2 **Time-Integral metrics**, which are computed by integrating over all events of a given type tracked during the progress of a webpage. In this category fall the two metrics proposed in this paper, namely the ByteIndex and the ObjectIndex, which generalize the SpeedIndex. 3 **Compound Scores**, weighting altogether several domain-expert heuristics to yield a score in the range [0,100]. For the sake of completeness, the table also reports the 4 **Mean Opinion Score (MOS)**, computed by averaging users' subjective ratings. MOS can be regarded as a benchmark for the other metrics, but it is admittedly hard to collect MOS points [13]. For this reason, we disregard it in what follows, leaving it as future work.

2.1 Time-Instant metrics

Metrics belonging to this category have the clear advantage of being easily measurable since they only track the realization of a specific event and, as a consequence, are widely used nowadays. Nonetheless, they are arguably simplistic since they disregard the complex chain of events that triggered the measured one. In a nutshell, such metrics compress the whole waterfall chart to a single time instant. Intuitively, two different experiments showing the same time-instant metric could be associated to significantly different user experiences. Despite this, the onLoad (also known as Page Load Time), which measures the time taken to completely load all the objects of a page, is still considered the main key performance indicator in the vast majority of recent scientific works, from both the industrial [4, 9] and the academic [7, 17, 18, 19] perspectives.

Other interesting metrics in this category include the TTFB, i.e., the time instant at which the first byte of payload is received (that expresses the page reactivity), and the DOM event, i.e., the time at which the Document Object Model is completely downloaded and parsed (after which the rendering can start). Simple tracking of the visual progress is expressed by the TTFP, which measures the time at which the first object is rendered. To further refine the tracking of visual progress, Google proposed the Above-the-fold (ATF) metric, defined as the time at which the content shown in the visible part of the webpage is completely rendered.

It is important to notice that few of these metrics, such as the TTFB, can be measured at the network (L3) or transport (L4) layers, whereas the vast majority, e.g., all those related to rendering, mandates the instrumentation of the browser (L7). Additionally, it is worth to notice that while most of the metrics in this category require few computations (if any), ATF is significantly more complex, as it requires to take screenshots during the rendering process and a post-processing stage of the captured frames.

2.2 Time-Integral metrics

Metrics in this category are characterized by the explicit use of all events in the webpage waterfall. In particular, Google introduced the SpeedIndex in 2012 to consider the whole process leading to the visual completion of a webpage and better account for user experience. In this paper, we generalize such a metric and introduce the family of time-integral metrics, defined as follows:

$$X = \int_0^{t_{\mathrm{end}}} (1 - x(t))dt \tag{1}$$

where X is the value of the metric, t_{end} is the time the last event is triggered, and $x(t) \in [0, 1]$ is the time evolution of the progress to reach such an event. Fig. 1 illustrates computation of a time-integral metric, where the blue line represents $x(t)$, and the gray-shaded area represents the result of the integral (1). Trivially, the smaller the area above the curve $x(t)$, the lower the score X, the better the user experience.

In order to make a concrete example, let us consider the SpeedIndex [8]. In such a case, $x(t) = \text{painted}(t)/\text{total}$ is the progress of the rendering process, and t_{end} corresponds to the ATF time-instant metric that marks the completion of the rendering. Under this light, the rationale of (1) is simple: not all the sub-events, i.e., the rendering of specific objects, are considered equally important. In particular, (1) gives *more weight to objects being rendered at the beginning* and vanishingly less weight to objects rendered towards the end. In other words, such a metric assigns a lower score to pages (or web browsers) rendering as much content as possible in the beginning with respect to pages (or browsers) rendering all the objects near $t \approx t_{\text{end}} = \text{ATF}$.

Bounds of time-integral metrics. It is immediate to notice that the time-integral metric X defined by (1) is lower-bounded by t_{TTFB} and upper-bounded by t_{end}.

Consider indeed Fig. 1, and observe that $x(t)$ is a monotonically increasing function equal to 0 at $t = 0$ and equal to 1 at $t = t_{\text{end}}$. Hence, the worst case time-integral metric is obtained when $x(t) = \mathbb{1}_{t \geq t_{\text{end}}}$ where $\mathbb{1}$ is the indicator function: with such a progress function, all the work is done in correspondence of the event of interest t_{end}. In this case, X is the area of the rectangle of base t_{end} and height 1, i.e., $X = t_{\text{end}}$, which implies $X \leq t_{\text{end}}$. Conversely, the best case is obtained when all the work is done at the beginning.

Notice that in practice, regardless of the considered time-integral metric, no progress whatsoever can be achieved before the first byte of payload (TTFB) is received by the web browser. Thus, the best case scenario is obtained when $x(t) = \mathbb{1}_{t \geq t_{\text{TTFB}}}$, which corresponds to the area of the rectangle with base TTFB and height 1, which implies $X \geq \text{t}_{\text{TTFB}}$.

Relationship to time-instant metrics. Extending the above reasoning, it follows that any time-instant metric t_X can be considered as the upper bound of the time-integral metric having $t_{\text{end}} = t_X$ or, in other words, time-instant metrics can be considered as projections of the corresponding time-integral metric. Particularizing this observation to Google's ATF and SpeedIndex proposals, we have that $\text{t}_{\text{TTFB}} \leq SI = \int_0^{\text{ATF}} (1 - x(t))dt \leq \text{ATF}$, which shows that time-integral metrics allow for a much more fine-grained measure.

Proposed metrics. The way the SpeedIndex metric is actually computed [11] is to take snapshots (by default at a frame rate equal to 10 fps) of a web browsing session. Such video frames compose a filmstrip which is analyzed in order to infer the visual completion fraction $x(t)$. More specifically, the color histogram of each frame is computed and compared to the histogram of the last frame, which represents the webpage at rendering completion. The use of histograms in place of a per-pixel comparison across frames owes to the fact that during the rendering process new objects might change the position of other objects already rendered. If that is the case, the per-pixel difference would detect the entire section affected by the relocation as incomplete, severely inflating the output value.

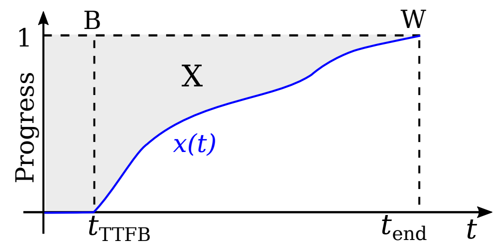


Figure 1: Time-integral metrics computation

However, we show in Sec. 3 that performing such operations burdens computational resources, significantly inflating the time needed to run the experiment, thus distorting it. To overcome such a limitation, we propose two metrics:

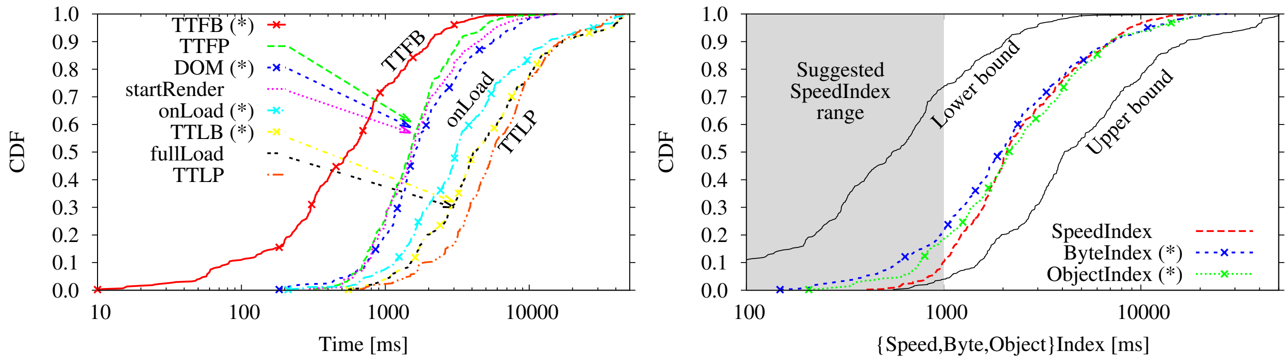
$$\text{ByteIndex} = \int_0^{\text{onLoad}} (1 - x_B(t))dt$$
$$\text{ObjectIndex} = \int_0^{\text{onLoad}} (1 - x_O(t))dt$$

where $x_B(t)$ and $x_O(t)$ is the percentage of objects and bytes retrieved at time t, respectively. Observe that both metrics require a negligible computational cost, as they can be computed by simply considering the time instants at which bytes/objects are fully downloaded. Finally, both the ByteIndex and the ObjectIndex can be considered as generalizations of the onLoad time-instant metric.

The rationale of these metrics is to avoid complex visual captures, and leverage the fact that received bytes/objects are directly (e.g., images) or indirectly (e.g., CSS) rendered by the browser. Second, as the SpeedIndex, these metrics consider all webpage events, with a temporal bias towards earlier events. Lastly, the ObjectIndex treats all objects equally, whereas the ByteIndex introduces a spatial bias as it implicitly states the size of an object to be correlated with its importance for the user.

2.3 Compound Scores

Compound scores such as Yahoo's YSlow [21], Google's PageSpeed Insights [12] and dynaTrace [3] encode expert knowledge, usually expressed as a set of heuristics (e.g., 23 in YSlow), combined with heterogeneous weights (e.g., 2% to 30% in YSlow). Such heuristics assess the effectiveness of a webpage design to: (i) reduce computation (e.g., avoid CSS expressions, alpha image load, image scaling), (ii) speedup rendering (e.g., limit DOM elements, CSS at the top, JavaScript at the bottom), (iii) reduce data volume (e.g., compress data, minify JavaScripts and CSS, use small cookies), and (iv) reduce delay (e.g., reduce DNS lookups, avoid redirections). As such, while relevant to assess the effectiveness of the adopted webpage design and indeed generally used to measure progress/regression of webpages, these heuristics are unrelated to event timing and hard to map to WebQoE.



(a) Time-instant metrics

(b) Time-integral metrics

Figure 2: Characterization of (a) time-instant and (b) time-integral metrics (Alexa top-100, WPT)

3. WebQoE EXPERIMENTS

3.1 Methodology

We illustrate WebQoE metrics with an experimental methodology. We consider the top-100 Alexa webpages (removing regional duplicates) as this is a widely used benchmark both in the industry [1] and academia [6]. As we expect variability across experiments due to load balancing, transient congestion, etc., we repeat the experiments 10 times for each page. For simplicity, we consider a single browser since differences in rendering and processing engines can play a determinant role. We argue that this would unnecessarily introduce complexity in the analysis and, as such, we consider only Google Chrome being by far the most popular browser and representing more than half of the market share.[3] We use (i) an unmodified Google Chrome (CHR) browser, as well as (ii) WebPageTest (WPT) [11] that we deploy on a local machine to orchestrate experiments. Notice that several metrics, including the SpeedIndex, are available only via WPT. Conversely, the ByteIndex and the ObjectIndex can be computed via both WPT and CHR.

For the time being, we run experiments from a single vantage point located in Paris, which corresponds to the case where Content Delivery Network (CDN) nodes are close. Additionally, we consider only the "desktop" version of each website. While the methodology, definition, and metrics would apply to the mobile Web as well, we believe that interactivity of the webpage plays an even greater importance in this latter scenario – which can be quite easy to convince of by considering that mobile webpages are designed to minimize the visual cluttering and to reduce the time needed to perform a useful action. As such, putting mobile and desktop versions within the same basket would introduce bimodal behaviors in the metrics of interest, which we prefer to avoid.

3.2 Results at a glance

We start by showing in Fig. 2 (a) time-instant and (b) time-integral metrics of which we report their empirical cumulative distribution functions gathered with WPT. In the legend, a star symbol (*) denotes metrics that can be computed via both WPT and CHR.

Considering the *time-instant* first, it can be seen that, as expected [20], events have an order relationship: e.g., no paint (TTFP) can happen before the first byte is received (TTFB), parsing of the DOM is necessary for the rendering process to start, and the reception of the full data (onLoad) can happen well before the last paint event (TTLP). It can also be seen that the TTFB and TTLP curves constitute the envelope of the process and are separated by over two orders of magnitude, as they pertain to rather different activities. For each metric, it can also be noted a significant variance: the median DOM (onLoad) is about 1.5 (3) seconds, while the 90th percentile is above 5 (13) seconds. Finally, it can be observed that some groups of metrics appear closely clustered, e.g., TTFP and startRender; last byte received (TTLB) and fullLoad, hinting for redundancy between the events definition and reporting.

Moving to the *time-integral* next, it can be seen that SpeedIndex, ByteIndex and ObjectIndex fall between the TTFB and TTLB envelopes. Additionally, these metrics are quite clustered, hinting to the fact that our simpler proposals have intrinsic similarities with the original SpeedIndex. The dark-shaded region on the left-hand side of the plot highlights the zone of advised SpeedIndex values for responsive websites: this hints to the fact that WPT slows down the whole rendering process (see Sec. 3.4).

A closer look reveals that the ByteIndex and the ObjectIndex *climb faster than the SpeedIndex* in the short-time frame regime. This is due to the fact that (i) the completion ratio for {Byte, Object}Index increases even before the DOM event, and that (ii) {Byte, Object}Index neglect computational and render time, i.e., they consider bytes/objects useful for the user experience as soon as they are received by the browser. Conversely, {Byte, Object}Index *climb slower than SpeedIndex* in the tail, as they consider possibly not painted objects (i.e., those that are below-the-fold). While in Sec. 4 we discuss how it would be possible to fine-tune the {Byte, Object}Index to approximate the SpeedIndex even more closely, we believe that the main takeaway is their remarkable proximity and potential interchangeability, with our proposals being much less expensive in computations.

[3] http://www.w3schools.com/browsers/browsers_stats.asp

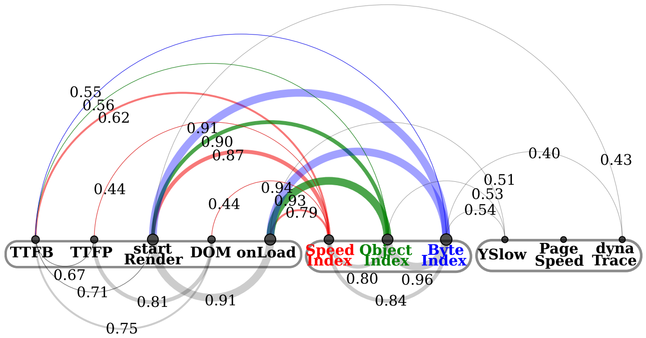


Figure 3: Pearson correlation between metric pairs.

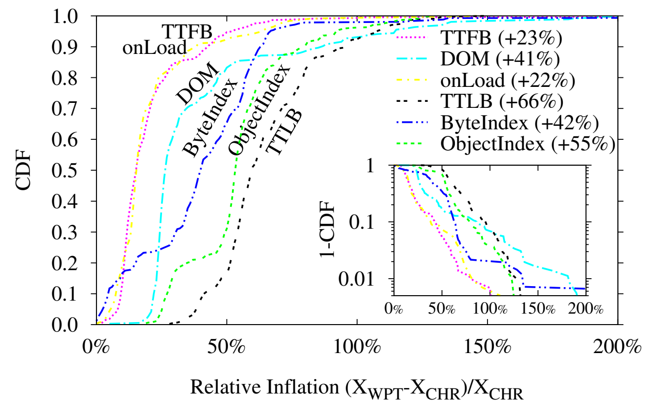


Figure 4: Relative inflation of time-instant and time-integral metrics under WebPageTest vs plain Chrome.

3.3 Relationship among metrics

To further assess the relationship among metrics, we report in Fig. 3 the Pearson correlation matrix between metric pairs, represented as an arc diagram. For completeness, we consider scores that are popular in the industry such as Yahoo's YSlow, Google's PageSpeed Insights, and dynaTrace computed by [1] on the Alexa top-100. In the plot, metrics are arranged into time-instant (left), time-integrals (middle), and compound scores (right). Correlations within group are reported in gray below the label name, while inter-group ones are reported above. Correlations of the time-integral group are highlighted in red (SpeedIndex), green (ObjectIndex) and blue (ByteIndex). To improve visualization, we only report correlations ≥ 0.4, with actual values annotated in the plot. To let the strongest correlation emerge, we quantize line width doubling it every 0.1 steps.

The picture reinforces the soundness of our proposal as it appears that: (i) our proposed byte-level and object-level replacements exhibit correlation with time-instant metrics, similarly to the SpeedIndex, and are highly correlated with the SpeedIndex itself; (ii) YSlow, PageSpeed, and dynaTrace heuristics are poorly correlated among them and with any other WebQoE metric, thus not representing valid alternatives.

3.4 Relationship among experiments

We finally contrast the same metrics gathered via WebPageTest (WPT) vs Chrome (CHR). Specifically, WPT computes a larger basket of metrics, notably including those related to rendering (TTFP, ATF, SpeedIndex, etc.). At the same time, computing such metrics affects the very same experiment: indeed, as recognized by the community [19], they require cumbersome screen captures that slow-down[4] the rendering process significantly.

To quantify the extent of the distortion in WPT vs CHR, we consider the subset of 6 metrics that can be computed in both, and define as $(X_{\rm WPT} - X_{\rm CHR})/X_{\rm CHR}$ the relative inflation of a generic metric X in the set. The cumulative distribution functions of the relative inflation are depicted in Fig. 4, annotating the average inflation for each metric in the label. A zoomed inset shows the complementary CDF, to better assess distortion in the tail. It can be seen that (i) inflation is non-linear, (ii) average inflation ranges from +22% to +66%, (iii) in the worst 10% of the cases, the ObjectIndex is almost doubled (and so is the TTLB). Otherwise stated, distortion in the experiment introduced by computational complexity makes the SpeedIndex of little practical relevance.

[4]The problem is that even if the SpeedIndex can be computed a posteriori, the screen capture process *itself* constitutes a significant CPU bottleneck

4. DISCUSSION

In this paper we provided a comprehensive view of metrics for WebQoE assessment, highlighting their merits, limitations, and dependencies. Our main contribution is to introduce a generalization of Google's SpeedIndex, which we instantiate into two very simple indexes, the ByteIndex and the ObjectIndex, having negligible computational complexity. Experimental results show that the ByteIndex and the ObjectIndex retain perceptual properties of the SpeedIndex, without its prohibitive computational complexity. This work opens a number of interesting future directions, which we now briefly discuss.

Closer SpeedIndex approximation. {Byte, Object}Index metrics provide optimistic lower bounds to the SpeedIndex: this is due to the fact that their completion increases upon reception of any bytes/objects, including (i) those that are not painted (e.g., scripts) as well as (ii) those that take time to render (e.g., alpha images, complex CSS). Conditioning over content type (e.g., null weight for scripts) would cope with (i), while considering execution times (e.g., no completion increase before DOM, estimation of time from reception to paint) could address (ii).

Psycho-behavioral model: content bias. Extending the above reasoning, it could be argued that taking explicitly into account object type or size could be worth investigating. For instance, for some object types a logarithmic reward can be expected from their byte-wise size (e.g., size of a JPG image encoded with higher quality may significantly increase, but the added value is likely sub-linear). Similarly, it may be argued that users perceive advertisement with a different value than content, which could be factored in by defining a weight function $w_i = 1 - \mathbb{1}_{\rm AdBlock(\it i)}$ with

$\mathbb{1}_{\mathrm{AdBlock}(i)} = 1$ whenever the domain name of object i belongs to the AdBlock list. Finally, the position of objects (e.g., center vs corners) is likely to have an impact, so that geometry of objects/paints could be valuable to explicitly account for (unlike WebPageTest SpeedIndex, being based on histograms).

Psycho-behavioral model: time bias. The {Speed, Byte, Object}Index metrics do take into account that paints/bytes/objects are not equally useful, and thus give *implicitly* larger weight to those happening earlier in the webpage lifetime. However, we believe that it would be interesting to *explicitly* control time dependence in reason of classic psycho-behavioral studies [15] (later adapted to the computer networks domain [16]), which show a logarithmic separation of human perception timescales. This could be accounted for by adding a multiplicative factor $(1+t)^\alpha$ in the integral (1). Notice that the current SpeedIndex definition implicitly assumes $\alpha = 0$, and is thus a particular case of this larger family of metrics.

{Byte, Object}Index in-browser computation. Computation of our proposed metrics has been done off-line leveraging HAR files. A useful addition of practical relevance would be to develop an in-browser version – of which we have a preliminary prototype [2] which however (i) is limited to Chrome and (ii) requires the Developer Tool extension. Implementation and integration in other frameworks/browsers would be very useful to gather a more accurate evaluation of the proposed metrics.

Correlation with MOS. Mean Opinion Score (MOS), obtained with experiments involving real users, is an obviously missing and utterly important piece of this puzzle. Albeit challenging in nature, obtaining a corpus of HAR files annotated with user MOS would be an important contribution to the whole QoE community.

Large-scale study. An obvious improvement of this work could then be to extend the characterization we conducted over the Alexa top-100 dataset by either (i) considering pages beyond the top-100, or (ii) performing the same experiment by employing geographically-dispersed vantage points (e.g., M-Lab nodes, PlanetLab nodes, Amazon Elastic Compute Cloud nodes, etc.).

Acknowledgments

This work was carried out at LINCS (http://www.lincs.fr). This work benefited from support of NewNet@Paris, Cisco's Chair "NETWORKS FOR THE FUTURE" at Telecom ParisTech (http://newnet.telecom-paristech.fr). Any opinion, findings or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of partners of the Chair. This work has been partially supported by the "FUTURE IN RESEARCH" project no. ACYBEH5 funded by the Apulia Region, Italy.

5. REFERENCES

[1] http://www.showslow.com.

[2] https://www.telecom-paristech.fr/~drossi/webmos.

[3] dynaTrace. http://dynatrace.com/.

[4] ALEXA INTERNET INC. http://www.alexa.com.

[5] BRUTLAG, J., ABRAMS, Z., AND MEENAN, P. Above the fold time: Measuring web page performance visually. http://conferences.oreilly.com/velocity/velocity-mar2011/public/schedule/detail/18692.

[6] BUTKIEWICZ, M., WANG, D., WU, Z., MADHYASTHA, H. V., AND SEKAR, V. Klotski: Reprioritizing Web Content to Improve User Experience on Mobile Devices. In *USENIX NSDI* (Oakland, CA, May 2015), USENIX Association, pp. 439–453.

[7] ERMAN, J., GOPALAKRISHNAN, V., JANA, R., AND RAMAKRISHNAN, K. K. Towards a SPDY'Ier Mobile Web? In *ACM CoNEXT* (December 2013), pp. 303–314.

[8] GOOGLE INC. https://sites.google.com/a/webpagetest.org/docs/using-webpagetest/metrics/speed-index.

[9] GOOGLE INC. https://googlewebmastercentral.blogspot.fr/2010/04/using-site-speed-in-web-search-ranking.html.

[10] GOOGLE INC. https://www.youtube.com/watch?v=muSIzHurn4U.

[11] GOOGLE INC. http://webpagetest.org.

[12] GOOGLE INC. Pagespeed insights. https://developers.google.com/speed/docs/insights/rules.

[13] INTERNATIONAL TELECOMMUNICATION UNION (ITU). Subjective testing methodology for web browsing. Tech. rep., International Telecommunication Union, Feb. 2014.

[14] M. VARVELLO, K. SCHOMP, D. NAYLOR, J. BLACKBURN, A. FINAMORE, AND K. PAPAGIANNAKI. Is The Web HTTP/2 Yet? In *Passive and Active Measurement (PAM)* (Apr. 2016).

[15] MILLER, R. B. Response time in man-computer conversational transactions. In *Proc. AFIPS Fall Joint Computer Conference* (1968), ACM, pp. 267–277.

[16] NIELSEN, J. Response times: The 3 important limits. https://www.nngroup.com/articles/response-times-3-important-limits/.

[17] QIAN, F., GOPALAKRISHNAN, V., HALEPOVIC, E., SEN, S., AND SPATSCHECK, O. TM3: Flexible Transport-layer Multi-pipe Multiplexing Middlebox Without Head-of-line Blocking. In *ACM CoNEXT* (December 2015).

[18] WANG, X. S., BALASUBRAMANIAN, A., KRISHNAMURTHY, A., AND WETHERALL, D. How Speedy is SPDY? In *USENIX NSDI* (Seattle, WA, Apr. 2014), USENIX Association, pp. 387–399.

[19] WANG, X. S., KRISHNAMURTHY, A., AND WETHERALL, D. Speeding up Web Page Loads with Shandian. In *USENIX NSDI* (2016).

[20] WORLD WIDE WEB CONSORTIUM (W3C). https://www.w3.org/TR/2012/REC-navigation-timing-20121217.

[21] YAHOO INC. Yslow Ruleset Matrix. http://yslow.org/ruleset-matrix/.