Web Cache Object Forwarding From Desktop to Mobile for Energy Consumption Optimizations

Troy Johnson
Department of Computer Science
Central Michigan University
Mount Pleasant, MI 48859
johns4ta@cmich.edu

Patrick Seeling
Department of Computer Science
Central Michigan University
Mount Pleasant, MI 48859
pseeling@ieee.org

Abstract—We introduce a user–centric approach to forward cached web objects from one peripheral device to a user's main mobile device, such as a smartphone. Our approach allows for a reduction in data that is required to be downloaded using the cellular interface when users are mobile. In turn, this reduces the energy consumption and potential latency requiring only basic algorithms that can be implemented in a straightforward manner. We exemplarily describe the benefits of our approach utilizing a popular international news web page before employing a large publicly accessible dataset of web page characteristics for simulation. Through simulation that incorporates web page popularity, we find that short-term download savings of 10–15 % are attainable within daily update boundaries and long-term savings around 10 % are realistic. With additional potentials possible if more sophisticated mechanisms are employed to optimize the delivery and exchange of cached objects, these savings can be seen as low end.

Index Terms—Mobile communication; Middleware; Energy consumption; Optimization

I. INTRODUCTION

With the beginning of the 21st century, networking support for wirelessly connected mobile user devices has fueled a continuous increase in the demand for mobile data. Web requests now originate in a majority from wirelessly connected user devices – a trend that Cisco, Inc. predicts to continue in the foreseeable future [1]. Simultaneously, the overall user behavior and demand for more rich media inclusion into web pages has increased the overall amount of data that is required to be transmitted per page, see, e.g., [2], [3]. Caching on the client side has been effectively used in the past and was, together with increased numbers of parallel object downloads, able to decrease wait times for desktop clients as reported in [2]. A first view of mobile web page characteristics (which were found to exhibit lower complexity than regular desktop browser versions) and non-landing pages (which were found to be less complex than landing pages) was evaluated by the authors of [3]. Moreover, a significant body of research has emerged that focuses on content delivery optimizations to mobile devices.

Typically, these optimization approaches are targeting on-device optimizations or off–device cloud–based improvements. For mobile applications, for example, significant energy savings were found to be attainable when grouping application

Fig. 1. Overall setup of world wide web based content access by a user over time using primary and secondary devices to display the same content. If the devices are able to synchronize their caches through short-range communications, energy savings become possible.

requests so as to avoid prolonged cellular network interface activity, see, e.g., [4], [5]. Other approaches optimize the delivery to mobile devices through proxies and cloud-based data anticipation and traffic shaping, see, e.g., [6].

For web data, typically caching is used to limit the amount of data that has to be transfered to requesting clients. In prior works focusing on mobile web page delivery optimizations, such as [7], energy optimization has been a key element, due to the limiting restrictions for mobile clients. One particular area for optimization is the pre–fetching of web page objects combined with caching (which allows to download data before usage), such as presented in [8] with about 10 % energy savings. More recently, these approaches were combined with user connectivity predictions, see, e.g., [9].

We propose to utilize a basic cache forwarding method that can be utilized by users to synchronize from, e.g., a desktop computer, to their mobile device, e.g., a smartphone. We illustrate this approach in Figure 1, which contains the desktop and a mobile device.

As most devices are charged over night, or are at least stationary within local area network ranges, we propose to utilize this general idle period to allow direct forwarding of cached objects. We presume that browser clients will have a shared information source in the cloud that allows the identification of visited web pages, an approach most browser clients take to date. In turn, clients are enabled to identify web page objects that are identical between the different device display modalities, i.e., identical web objects delivered when requesting a desktop/mobile versions of a web page. The thus transmitted objects, in turn, reside locally within the mobile device cache and do not require an energy–expensive mobile download through cellular interfaces. We note that the reverse of this approach is possible as well. To support our approach, we gather data through the publicly accessible WebpageSpeedTest.org website as well as the httparchive.org archive of a large dataset of performance evaluations for

Fig. 2. Screenshot of rendered web page from WebPageSpeedTest.org. As illustrated, the main textual and pictorial content items are flanked by background and interactive advertisements.

popular web pages; we refer the interested reader to [10] for a more detailed discussion.

The remainder of this paper is structured as follows. In the following section, we provide an example based on a particular news web page. NEED TO EDIT THE SECTION SO THEY MATCH UP! In the subsequent Section III, we discuss the overall properties of a large web page dataset archive and provide a description for the simulation-driven evaluation of our approach. We discuss the obtained results in Section IV before concluding in Section V.

II. INDIVIDUAL EXAMPLE

In this section, we outline an individual example for the German news web page "Der Spiegel," accessible at http://www.spiegel.de. This web page serves as an example for a rich media and advertising material containing web presence that is frequently updated. On February 12, 2014, we performed a speed test online using (i) Internet Explorer 9 as the browser instance for a desktop client, (ii) Chrome as the mobile browser on a Google Nexus 5 instance, and (iii) Safari for an iPhone 4 instance provided by WebpageSpeedTest.org (both mobile clients were traffic shaped to 3G connection emulations). An example screenshot of the web page as rendered is provided in Figure 2. As illustrated, the web page exhibits a significant amount of objects that are required for advertisements, visual items (images, layout), and scripting. We evaluate the reported data as follows. For each display modality and browser client, we gather the individual web objects requested and the response sizes and headers for caching information (for HTTP response codes 200 only, as we are not interested in redirects). We denote the number of all objects returned for a request modality (i.e., Internet Explorer for the desktop client example and iPhone or Nexus 5 for mobile counterparts) as N, their average size as $\overline{X}$, the standard deviation of their sizes as σ_X, and the Coefficient of Variation (CoV) of the returned object sizes as CoV_X. As

TABLE I
HIGH-LEVEL OVERVIEW OF DIFFERENT CLIENT WEB PAGE STATISTICS
FOR HTTP://WWW.SPIEGEL.DE

Statistic	IE Deskt.	iPhone	Nexus 5
N	154	132	145
$\overline{X}$	10402.77	8846.39	10201.63
σ	19103.19	13438.05	25684.58
CoV_X	1.84	1.52	2.52
$\overline{T}$	106285.14	114780.69	114818.28
σ_T	125080.27	125576.65	126090.90
CoV_T	1.18	1.09	1.10

in the HTTP specification, the *max–age* directive overrides others, so we initially consider that directive for the cache longevity of the individual objects. If no explicit information is found, we consider the response header's *expires* information, which provides a secondary cache lifetime. If both are not found, or if the *expires* date is in the past or right at the request time, we set the cache lifetime to zero. Similar to the notation for the object sizes, we denote the expiration time characteristics as $\overline{T}$, σ_T, and CoV_T, respectively.

A. Data Description

We provide an initial high-level overview of the web page characteristics for http://www.spiegel.de as requested by the different clients in Table I.

We initially observe that the desktop version (with Internet Explorer 9 as requesting browser client) exhibits the highest number of objects, followed by the mobile Chrome/Android and Safari mobile/iOS versions. Interestingly, there is only a minor difference in the number of objects between these versions. Next, we observe that the trend for the average number of bytes is aligned with the number of elements. In total, the mobile Chrome version is almost on par with the desktop one (at 92 % of total data); only the Safari mobile version seems optimized (at 72 % of total data). The standard deviation and Coefficient of Variation (CoV) amongst individual element sizes are highest for the mobile Chrome version, indicating more significant size differences than for the safari version (which is one unit lower), while the desktop version falls into the middle.

Next, we evaluate the elements' caching properties on a high level as presented in Table I as well. Overall, we note that the highest average caching time is observed for the mobile Chrome access with the Nexus 5 device, trailed immediately by the iOS access and (with distance) by the desktop browser. We note that the overall variability in terms of expiration times amongst elements is fairly low and comparable, as indicated by CoV values around 1.1.

B. Evaluation of Local Cache Forwarding

We now shift the view to the possibility for identical objects to be re–utilized locally to avoid additional download penalties in time and energy consumption. Simultaneous with a web page request, a local broadcast could "ask" for the cached elements for a website to be delivered locally from participating
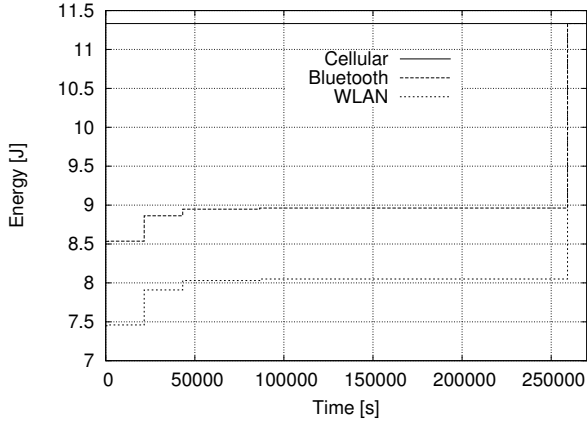
Fig. 3. Energy required to download web page data via cellular connection or through combination with partial local exchange with Bluetooth/WLAN.

Fig. 4. Relative number of items and data amounts in cache and savings resulting from local data exchange with a desktop client.

clients. Alternatively, cellular provisioning methods, such as in LTE–A, could initiate the device–to–device local exchange as well. Once the request is received, the local coordination can take place, which inherently results in the forwarding of elements that are the same across browser instances and which have a future cache lifetime expiration.

We filter out elements that are not similar amongst the different requesting devices, which results in 82 objects with the same URL and size combination. In other words, just above half of the objects constituting the web page under consideration are identical between different clients.

While we note an identical cache lifetime for most of these objects, a slight increase is notable from the Desktop over Safari to Chrome clients. Next, we remove items that exhibit a cache lifetime of zero for any of the three requesting clients, resulting in 59 objects. These remaining objects display a reduced average cache lifetime for the mobile browsers, whereby mobile Chrome exhibits the shortest. In turn, we choose the iOS Safari mobile as our exemplary base for calculations. We utilize the approximations for energy consumptions presented in [11] to determine the amount of energy required if local data forwarding can be performed using either Bluetooth or WLAN technologies. We illustrate the energy consumption as function of time in Figure 3. The complete download energy for exclusive use of cellular is the upper limit on energy spent downloading the data associated with the web page. The Bluetooth and WLAN data exchanges with other local clients both follow the same underlying trend with increasing energy required to transmit all data as time progresses from the last access point in time (due to cache expirations).

We illustrate the relative amount of data/items within the cache in addition to the attainable savings in Figure 4 as a function of delayed access time. We observe that a significant amount of the overall data can be cached (and would in turn be suitable for localized forwarding). We additionally observe that even for longer durations up to multiple days, significant amounts of data remain in the cache. Combining the
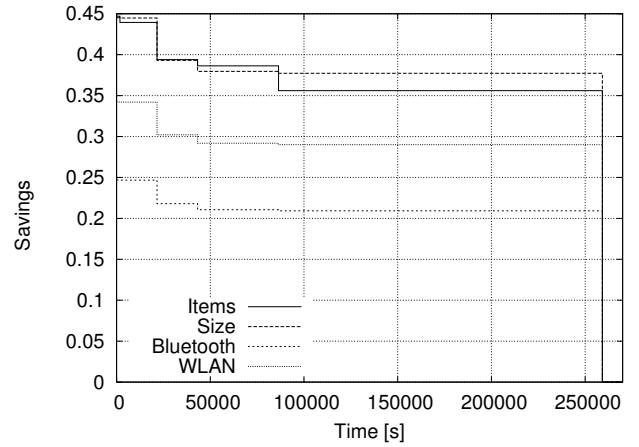
forwarding from locally cooperating clients, e.g., user-owned desktop computer in the same network through either Bluetooth or WLAN technologies, we utilize the approximations outlined in [12] to derive the relative gains of our approach in terms of energy consumption. Specifically, we compare our approach (combining locally forwarded cached objects and cellular downloads of the remainder) with the regular download of all objects through the cellular network interface. We note that significant savings above 20 % are attainable with our approach, even if the filling of the device's cache is performed wirelessly as well.

III. Large Dataset Performance Evaluation

We now evaluate our proposed approach through simulations based on a large dataset of landing web pages of popular websites. For this purpose, we retrieved the publicly available web page response details from *httparchive.org* and created a subsequent dataset for October 1st, 2013.

A. Dataset Description

The dataset we utilize for the performance evaluation contains the individual web responses for the most popular websites of fixed (Internet Explorer for the desktop) and mobile (iOS for iPhone 4) websites, ranked through the Alexa popularity index. Furthermore, the dataset contains the individual response details, including cache expiration times and sizes.

Similar to the individual web page example, we provide an overall description of the response sizes and maximum expiration ages in Table II. We note that the overall average response size as result of desktop requests (when batched into the outlined time frames) is around 17 kB, with the largest average sizes occurring between one and twelve hours. Mostly, we note that the majority of objects exhibit a short expiration time between zero and 30 seconds. For the responses to mobile devices, we note that the majority of objects are in the same short time span and the long time span of more than one

| Category	Desktop (IE)				Mobile (iOS)			
	$\sum X_i$ [MB]	N	$\sigma(X_i)$ [kB]	$\overline{X_i}$ [kB]	$\sum X_i$ [MB]	N	$\sigma(X_i)$ [kB]	$\overline{X_i}$ [kB]
expAge < 30 Seconds	238490.28	15670679	1698.59	15.22	1355.24	122598	582.98	11.05
$30 \leq$ expAge < 1 Minute	170.94	29130	27.38	5.87	12.68	898	70.38	14.12
1 Minute $\leq$ expAge < 1 Hour	15192.40	797450	270.06	19.05	282.36	15854	226.62	17.81
1 Hour $\leq$ expAge < 12 hours	29798.68	1186947	751.45	25.11	244.14	17424	209.84	14.01
12 hours $\leq$ expAge < 1 day	6188.57	358998	227.63	17.24	91.37	5952	91.37	15.35
1 day $\leq$ expAge	195784.64	9888147	1337.36	19.80	2122.96	126629	750.51	16.77
Average	80937.59	4655225	718.75	17.39	684.79	48226	321.95	14.20
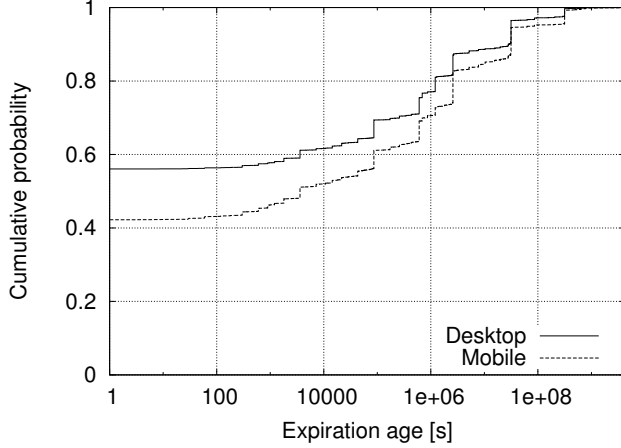
Fig. 5. Cumulative probability of expiration times for desktop and mobile responses.

day. The highest average response sizes, however, are observed between one hour and half a day, in difference to the desktop counterpart. For more details of the differences between fixed and mobile web pages, we refer to [13].

We illustrate the cumulative distribution for the desktop and mobile client response expiration ages in Figure 5. We observe that a significant portion (> 40 %) of responses from web servers exhibits an immediate expiration and would, in turn, require an immediate retrieval by clients. The cumulative expiration probability is higher for responses to desktop clients, which can be attributed to the typically more abundant bandwidth (and less optimization requirements) in contrast to a mobile setting (where web content developers might be more considerate of limitations). In addition, we perform matching of request to mirror the individual example with a direction from desktop to mobile client.

B. Simulation Model Description

We evaluate the effectiveness of our approach through simulation, which we perform as follows. For the time period of one week, we randomly select a web page from the mobile dataset and separate the data that is required to be downloaded and the data that is still available in cache due to the maximum expiration age of the individual objects. We subsequently wait for a random amount of time before repeating the procedure.

Initially, we sort all landing web pages $l, l = 0, \dots, L$ from the dataset based on their popularity ranking. To simulate the popularity of web pages, it was found that the Zipf distribution effectively describes the popularity of individual requests. We chose to utilize the non-modified Zipf distribution for our simulation purposes, noting that additional modifications to the distribution exist that can further enhance it, see, e.g., [14]. For the total number of web pages L, we derive the probability of page l being selected in turn as

$$p(l) = \frac{l^{-\alpha}}{\zeta(l)}, \tag{1}$$

where $\zeta(\cdot)$ denotes the Zeta function. We furthermore set $\alpha = 0.85$ as an average choice in common ranges utilized for this parameter.

Once the individual mobile web page l was randomly chosen, we compare it to its desktop counterpart, with $\{d, m\}$ denoting the request modality. As each page l exhibits a time-sensitive number of responses, we denote them as $r^{\{d,m\}}(l, t)$ and each page exhibits $R^{\{d,m\}}(l, t = 0)$ in total. Let $t_c(r^{\{d,m\}}(l))$ denote the expiration or max-age directive received with the response at $r^{\{d,m\}}(l, t = 0)$. The number of objects or responses to be retrieved at time t in turn are given as

$$R^{\{d,m\}}(l, t) = \sum_{r^{\{d,m\}}(l,0)} \left[t_c(r^{\{d,m\}}(l)) \geq t \right] \tag{2}$$

where $[\cdot]$ denotes the Iverson Bracket. In the following, we abbreviate to t_c for readability when in direct context.

We furthermore denote the size of the individual response retrieved at time t as

$$x_r^{\{d,m\}}(l, t) = x_r^{\{d,m\}}(l, 0) \cdot [t_c \geq t]. \tag{3}$$

The total size of objects or responses retrieved at simulation time t thus is given as

$$X^{\{d,m\}}(l, t) = \sum_{r^{\{d,m\}}(l,0)} x_r^{\{d,m\}}(l, t). \tag{4}$$

For performance evaluation purposes, we simulate the user behavior similar to the process outlined in [15], whereby we randomly draw the time between requests t_u as Pareto distributed using

$$p(t_u) = \beta k^\beta t_u^{-(\beta+1)}, \tag{5}$$

with $\beta = 1.5$, $k = 30$. We assume that no delays are accrued due to instantaneous cache retrievals or downloads. Using time index i, we thus derive $t_{i+1} = t_i + t_u$. We continue the simulation until $t_{i+1} = t \geq T = 640800$ [s], i.e., one week. To maintain tractability, we bundle the presented results into bins of 3600 seconds, i.e., one hour, through averaging.

IV. PERFORMANCE EVALUATION RESULTS

We present our results for the individual responses or web objects and the total number of bytes. We note that through approximations, such as the ones described in Section II, an inference of energy consumption would be possible as well. We perform the simulations with an average $\alpha = 0.85$ for the utilized Zipf distribution and repeat each week-long evaluation 2000 times. Initially, we present the total number of responses and bytes that would not require a download (i.e., would reside in the mobile cache) as a function of time in Figure 6.
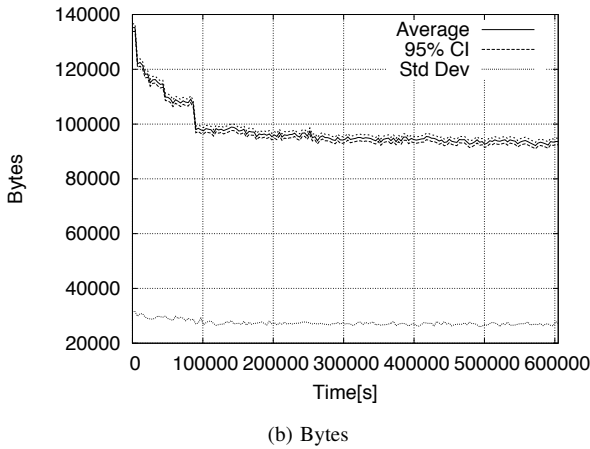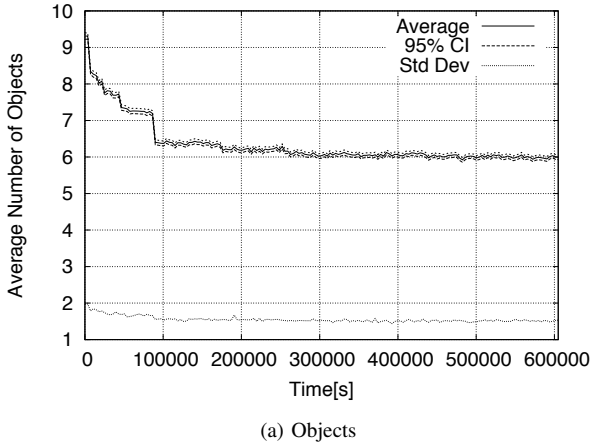

(a) Objects

(b) Bytes

Fig. 6. Simulation results for the number of responses and bytes that are in the transfered mobile device cache as averages in 1-hour bins.

We note an immediate decrease in the number of requests, which is mirrored by the number of bytes as well (indicating a linear relationship as observed in, e.g., [13]). We furthermore observe that the initial decline levels out at the simulation time of around one day, and remains steady afterwards. This
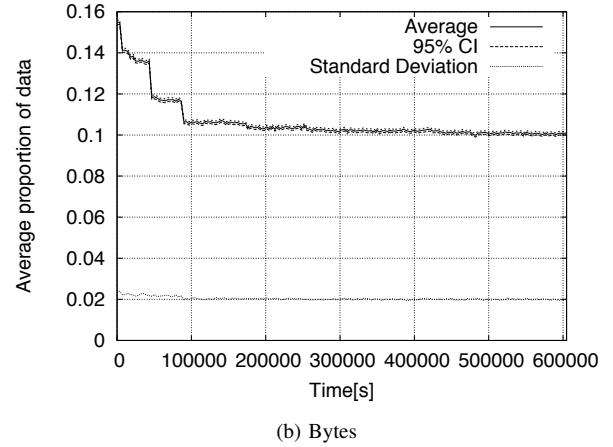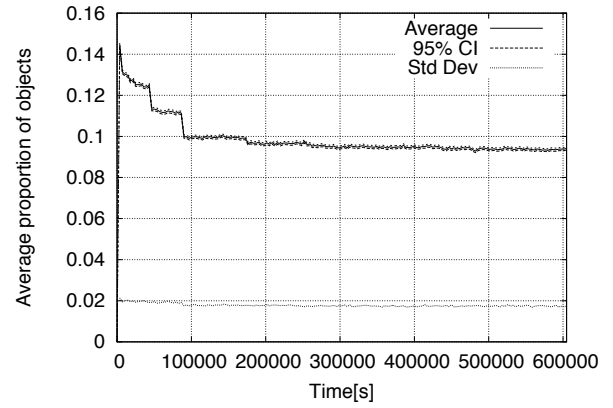

(a) Objects

(b) Bytes

Fig. 7. Simulation results for the attained savings for the number of responses and bytes by transferring to the mobile device cache in 1-hour bins.

indicates that within the simulation period of one week, initially a large number of items is non–shared between devices or exhibits no significant cache lifetime. In a following group of objects, cache lifetimes increase logarithmically, which leads to the exponential decline we observe here. The narrow confidence intervals and low level of standard deviation indicate that the presented results are stable within simulation confines.

Next, we present how these findings translate into attainable savings with respect to requests for objects and bytes in Figure 7 in relationship to the total web page data without caching.

We initially note a declining trend similar to the one observed for the cached items in Figure 6, but with more distinct "jumps" observable at half-day and day times of the simulation. Overall, we note that almost 15 % savings for objects and bytes decline to around 10 % after the boundaries of a day (whereby objects exhibit lower levels and bytes exhibit higher levels).

To evaluate the impact of different web page popularity distributions, we present an evaluation of different α for the Zipf distribution in Figure 8, each with 500 simulation
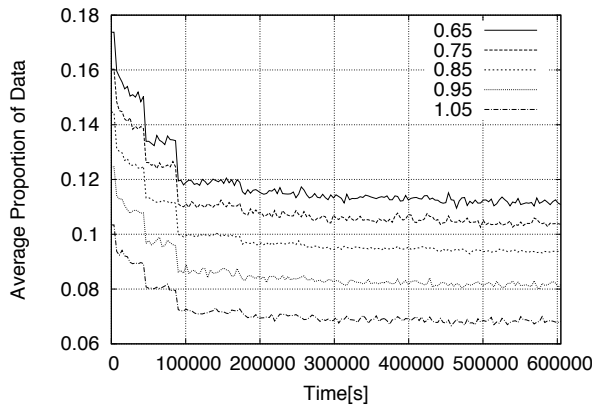
Fig. 8. Average proportions of data in mobile cache for different Zipf distribution parameters α.

repetitions for $0.65 \leq \alpha \leq 1.05$. We limit our comparison to bytes due to space constraints, noting that results obtained for objects are similar. We observe that all levels of α exhibit the same underlying trends, with slight and non-linear differences between levels. We conclude that the distribution of web page popularity and the composition thereof has an overall impact on the long-term level of savings that can be attained. Overall, we find a significant reduction in the required downloads for mobile clients that can be directly attributed to lower energy consumption levels and potential download latencies (and even costs for the mobile user through saved data transmissions). Furthermore, this approach does not require sophisticated live evaluations, but only relies on cache forwarding to the mobile device.

V. CONCLUSION

We exemplarily evaluated the content of modern web pages and the identical objects that can be found when requesting the same web page from different devices with different display modalities, such as desktop and smartphone. Assuming mobile users visit the same pages on their mobile device that they visit on their desktop computer as well, we simulated the access of landing web pages and evaluated the potential for cache forwarding. Our basic approach is able to save upward of 7.5 % of mobile request or bytes even for very distant time horizons. In the more typical daily time range, our approach can yield almost linearly decreasing savings stating at about 14 %, without requiring any sophisticated prediction mechanisms, but only a direct cache transfer. We note that similarly, exchanges between mobile and fixed devices belonging to a user are possible, including keeping a cloud-based reference of pages that a user visits (similar to Google Chrome's synchronization features).

REFERENCES

[1] Cisco, Inc., "Cisco visual networking index: Global mobile data traffic forecast update, 2013–2018," Tech. Rep., Feb. 2014.

[2] S. Ihm and V. S. Pai, "Towards understanding modern web traffic," in *Proc. of the ACM SIGCOMM Conf. on Internet Measurement Conf.*, Berlin, Germany, 2011, pp. 295–312.

[3] M. Butkiewicz, H. Madhyastha, and V. Sekar, "Characterizing web page complexity and its impact," *IEEE/ACM Transactions on Networking*, vol. PP, no. 99, pp. 1–1, 2013.

[4] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy consumption in mobile phones: A measurement study and implications for network applications," in *Proc. of the Internet Measurement Conf.*, Chicago, IL, USA, Nov. 2009, pp. 280–293.

[5] F. Qian, Z. Wang, Y. Gao, J. Huang, A. Gerber, Z. Mao, S. Sen, and O. Spatscheck, "Periodic transfers in mobile applications: Network-wide origin, impact, and optimization," in *Proc. of the International Conference on World Wide Web*, Lyon, France, Apr. 2012, pp. 51–60.

[6] Y. Xiao, P. Hui, P. Savolainen, and A. Ylä-Jääski, "Cascap: Cloud-assisted context-aware power management for mobile devices," in *Proc. of the International Workshop on Mobile Cloud Computing and Services*, Bethesda, MD, USA, Jun. 2011, pp. 13–18.

[7] F. Sailhan and V. Issarny, "Energy-aware web caching for mobile terminals," in *Proc. of the International Conference on Distributed Computing Systems*, 2002, pp. 820–825.

[8] H. Shen, M. Kumar, S. K. Das, and Z. Wang, "Energy-efficient data caching and prefetching for mobile devices based on utility," *Mob. Netw. Appl.*, vol. 10, no. 4, pp. 475–486, Aug. 2005.

[9] B. Thyamagondlu, V. Chu, and R. Wong, "A bandwidth-conscious caching scheme for mobile devices," in *Proc. of the IEEE International Congress on Big Data*, June 2013, pp. 78–85.

[10] P. Meenan, "How fast is your web site?" *ACM Queue*, vol. 11, no. 2, pp. 60:60–60:70, Mar. 2013.

[11] G. P. Perrucci, F. H. P. Fitzek, and J. Widmer, "Survey on energy consumption entities on the smartphone platform," in *Proc. of the IEEE Vehicular Technology Conference (VTC Spring)*, Budapest, Hungary, May 2011, pp. 1–6.

[12] P. Seeling, "Caching proxying for mobile users," in *Proc. of IEEE Vehicular Technology Conference (VTC)*, Dresden, Germany, Jun. 2013, pp. 1–6.

[13] T. A. Johnson and P. Seeling, "Desktop and mobile web page comparison: Characteristics, trends, and implications," *IEEE Communications Magazine*, accepted and to appear, 2014.

[14] S. A. Krashakov, A. B. Teslyuk, and L. N. Shchur, "On the universality of rank distributions of website popularity," *Computer Networks*, vol. 50, no. 11, pp. 1769–1780, 2006.

[15] G. Anastasi, M. Conti, E. Gregori, and A. Passarella, "Performance comparison of power-saving strategies for mobile web access," *Performance Evaluation*, vol. 53, no. 3, pp. 273–294, 2003.