Experimental Study of Energy and Bandwidth Costs of Web Advertisements on Smartphones

Abdurhman Albasir*, Kshirasagar Naik*, Bernard Plourde†, and Nishith Goel‡

*Dept. of Electrical and Computer Engineering
University of Waterloo
Waterloo, ON, Canada, N2L3G1
Email: {aalbasir, snaik}@uwaterloo.ca

†Research and Development
Technologie Sanstream
Gatineau, QB, Canada, J8Y2V5
Email: bplourde@sanstream.ca

‡Research and Development
Cistech Limited
Ottawa, ON, Canada, K2E 7L5
Email: nishith@cistech.ca

***Abstract*—With the rapid advancements in mobile devices, users have become more attached to them than ever. This rapid growth, combined with millions of applications (apps), makse smartphones a favourite means of communication among users. In general, the available contents on smartphones, apps and the web, come in two versions: (i) free contents that are monetized via advertisements (ads); and (ii) paid contents that are monetized by user subscription fees. However, the resources, namely, energy, bandwidth, and processing power, on-board are limited, and the existence of ads in websites and free apps can significantly increase the usage of these resources. Therefore, in this paper, we describe an approach that enables the separation of web contents in a number of websites. Having done so, the energy cost due to downloading, rendering, and displaying web ads over Wi-Fi and 3G networks is evaluated. That is, how much energy web ads contribute to the total consumed energy when a user accesses the web. Furthermore, the bandwidth consumed by web ads in a number of well-known websites is also evaluated. The high cost of ads on smartphones must be considered by the designers and vendors of apps.**

***Index Terms*—Smartphones, Energy Cost, Bandwidth Cost, Web Ads**

I. Introduction

Web browsing is becoming essential for everyday life, especially with the ever-increasing popularity of smartphones. Web contents delivered to end users have undergone witnessed drastic changes. Webpages used to be simple static pages comprising only text and images. However, modern webpages are dynamic and media-rich. The owners of these pages (publishers), moreover, rely heavily on ads as a source of revenue. These ads are media-rich and consume much of battery and bandwidth resources. Entirely eliminating ads from webpages is not practical, in view of the big role they play in the web eco-system and in the process of having free web contents. These factors have led to the demand for mobile browsing solutions that adapt pages in a way that is energy and bandwidth efficient.

When a webpage is loaded on a user's screen, what actually is downloaded are the core information that a user is interested in (such as news, emails, stocks, etc.), and extra "unwanted" information that comes up in the form of web advertisements (ads). In a smartphone environment where resources are limited, displaying ads on the screen is not an inexpensive task from the energy and bandwidth perspectives. To be displayed, an ad needs to be fetched, rendered, and finally displayed. Each of these actions requires specific resource: network access (radio interface and bandwidth), CPU computations, and graphics. Figure. 1 illustrates the extra steps needed to display ads. These extra requests are reflected as an increasing battery energy consumption.

The main problem explored in this paper is as follows. The complexity of webpages is increasing, especially if the webpages are designed for desktop computers. The existence of advertisements (ads) in webpage leads to even higher complexity. This complexity in a smartphone environment, where resources are limited (*e.g.,* battery and bandwidth) is reflected in longer loading time, more energy consumed, and more bytes transferred. Therefore, evaluating resources used by web advertising is very important to smartphone users as well as to web designers. Thus, this paper focuses on quantifying the energy and bandwidth cost due to web ads in smartphones.

None of the studies surveyed in our review focus specifically on measuring the resources used in mobile devices by

MobiCASE 2014, November 06-07, Austin, United States

DOI 10.4108/icst.mobicase.2014.257770

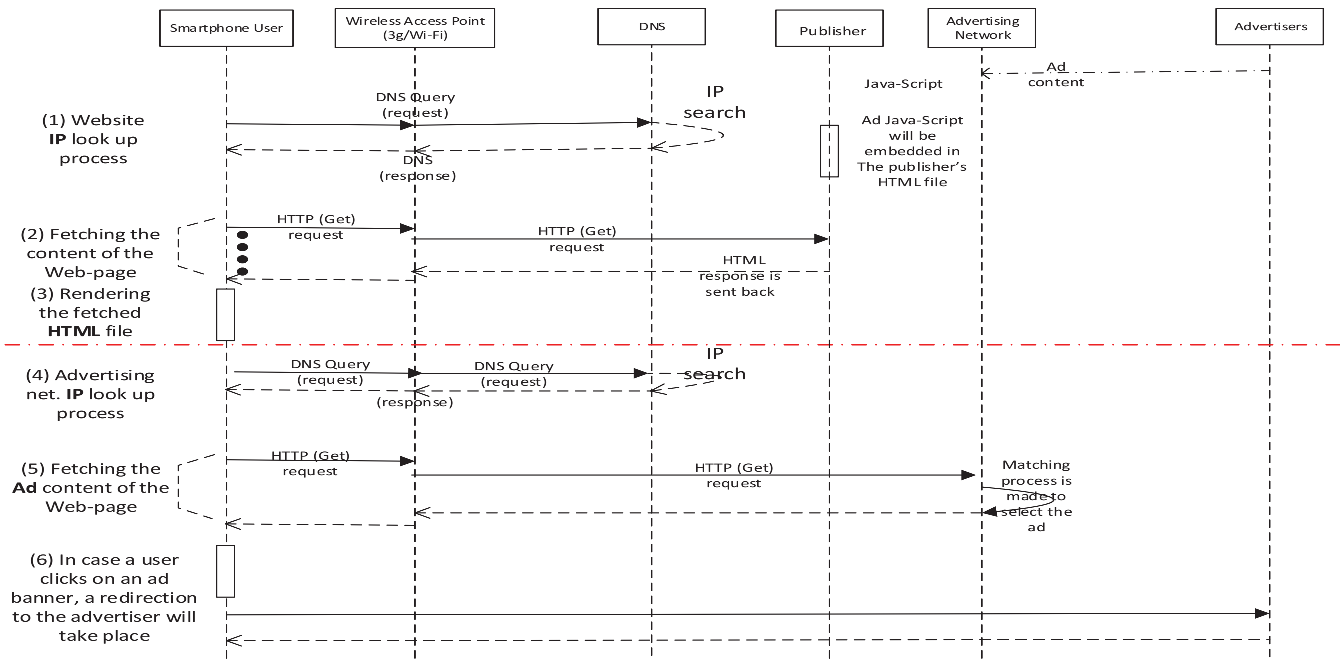


Fig. 1. Sequence diagram of downloading webpage

advertisements' overhead during mobile web browsing. Most address only the issue in mobile applications, and some study the cost of ads on PCs or laptops but not on mobile devices. Therefore, this study is among the firsts that analyse mobile resource usage due to displaying ads on webpages based on real measurements.

This paper makes the following contributions; (i) quantify the ad traffic generated during mobile web browsing; (ii) estimate the additional cost of downloading ads to a user's monthly bill; (iii) quantify the energy consumed to download and display ads on webpages while mobile web browsing; (iv) investigate the impact of using different networks, namely 3G and Wi-Fi, on the energy consumed to download and display ads on webpages.

The rest of the paper is organized as follows. Section II covers the related works. Section III describes the testing methodology used and the experiments carried out to achieve the contributions of this research. Section IV presents and discusses the measurements of energy and bandwidth expended for ads. Section V lists some conclusions of this paper.

II. Related Work

Energy consumption and battery life of smartphones have been the topic of numerous publications [1][2]. Energy management solutions in a smartphone environment were discussed in details in [3]. However, fewer publications have reported on the issue of energy consumed by the advertisements embedded in both apps and web pages. The existing works are categorized into web ads energy cost and web ads bandwidth cost, as described below.

A. *Web Ads Energy Cost*

In the context of studying the energy cost of web ads while mobile browsing, there are not many studies that talked specifically about this issue [4]. However, there are several research articles that investigated mobile browsing from energy perspective [5] as well as other performance aspects, namely, speed, simplicity, and usage and usability [6], [7].

Studying the energy cost of web advertising on smartphones or hand-helds in general has not been performed so far, to the best of our knowledge. However, there is a study done by Simons and Pras [4] targeted at the hidden energy cost of web advertising on desktops (PCs). They investigated the amount of CPU and display energy consumption due to web advertising and tracking. They found that displaying ads that are rich in animations and graphics is an energy-expensive task. It puts the CPU under computational stress that causes higher power consumption, and in turn, costs more money. For instance, their results reveal that the cost to render and display web ads is 2.5 W. Using national statistics, and based on their results, they found that the total energy used by ads is equal to the annual electricity usage of 1891 Dutch households.

In [5], Thiagarajan *et al.* analysed the energy consumed by mobile browsers. They measured the total energy cost of rendering web page elements as well as the 3G radio energy needed to fetch such a web page. Then, they went further by breaking down such measurements and answering the question of how much energy each and every web component consumes: cascade style sheets CSS, Javascript, images, and plug-ins. Based on their measurements and analysis, a number of recommendations were provided to develop and build more energy efficient site. However, the ads were not in the focus in

[5]. Some other studies looked at web browser and web pages from other perspectives. For instance, in [6], Wang *et al.* critically analysed the slowness of web browsers on smartphones. They found that the main reason the slow web browsing is the process of fetching different web contents. They concluded that the key point of improving mobile browsers is to speed up such process. In [7], Hoehl *et al.* investigated the possibility of benefiting from presenting mobile version websites on desktop computers so that simpler web contents can be provided to people with cognitive disabilities.

B. Web Ads Bandwidth Cost

There have been some studies in literature that quantified the traffic generated while web browsing. An attempt was made by Erman *et al.* [8] to classify HTTP traffic in homes. Their study covered 17,000 broadband DSL subscribers' traffic for a month in Texas. They answered some questions that characterize the nature of HTTP traffic in homes. For instance, they found that not all the HTTP requests are made by PCs or laptops. Also, they found that 11% of all HTTP requests are being made to communicate to advertising servers, which is equivalent to 0.2% (that is $\sim$312 GB) of the total bytes downloaded. However, their study did not particularly target the amount of ads traffic generated only from mobile web browsing.

Brande *et al.* [9] considered ads as an unwanted traffic while mobile browsing. They investigated ads traffic contribution in the total web traffic. They built a testing system in a way that allows them to block the ads when needed, and automated the browsing process to mimic the browsing behavior. Their results reveal that web ads take up 7-9% of the total data traffic. Based on the percentage of the generated ad traffic and the average domestic and roaming charges, the authors estimated the cost of web ads for mobile users. They found that the additional cost would be $4.5-7 and $50-80 per month while browsing domestically and internationally, respectively. However, their method of blocking the ads is based on using proxy which, as we explain in Section III-A, does not ensure that all the ads will be eliminated from the webpages. Moreover, their method does not eliminate the ad analytic information that is being exchanged between the ad networks and the client. That explains the discrepancies between our findings and theirs.

III. Testing Methodology

To perform testing, the first challenge was to separate the web contents. In other words, we wanted a way to separate the ad contents from the webpage so that we can test the smartphone once with the existing ads and a second time without ads. As a result, we can measure exactly the energy and bancdwidth cost of ads by calculating the difference between the two cases. A method was sought to block or insulate the ads from the webpages. The available blocking techniques are as follows.

A. The Available Ad-blocking Techniques

- *Ad-blocking Applications*: A number of apps available online can block ads, namely, Ad-Vanish Lite, NoRoot Ad-Remover Lite, and AdFree. These apps work by turning off the Internet connection completely. They are designed to block ads from *Android* games and they work as follows: the user is requested to prepare an Ad-block list, so if he wants to stop displaying ads on a game all he needs to do is to put this game on the Ad-block list. These apps in turn make sure that the games will have no access to the Internet any more, and hence the ads are blocked. These Ad-blocking apps work fine for the apps that do not require an Internet connection. In the case of web browsing, establishing an Internet connection is essential. Therefore, this Ad blocking option does not satisfy our testing needs.
- *Browser Plug-in Ad-blockers*: Mobile web browsers such as Mozilla Firefox and Android can use plug-in ad-blockers to block ads. Some plug-in apps are available online (Adblock Plus) and they filter out the ad URLs. These plug-ins maintain a black list that contains advertising companies' URLs; thus, whenever an ad request made by the browser is found on the black-list, that request is aborted. Our comments on these methods are as follows: (i) installing these plug-ins will change the way browsers work, and hence the measurements will be distorted. In other words, having such plug-ins working in the background requires extra activities and computational power. As a result, extra energy is consumed. For our case, where we are looking to measure the actual energy and bandwidth required to download and display ads, using this option will give us distorted measurements. (ii) In addition, not all of the ads will be blocked since the operation of these plug-ins relies entirely on the specified list, which may not contain all the advertising companies (Ad networks.) As a result, some ads will still be displayed while adopting this option. (iii) The ad links (ad tags) in a webpage are already delivered to the smartphone and parsed by the browser; as a result, some bandwidth and energy would have been consumed. For these reasons, we eliminated this option as well.
- *Proxies*: Proxies, namely, Privoxy [10], can be installed at a middle point (e.g., a Wi-Fi access point) to block ads. The mechanism used to block ads is similar to the one explained in Ad-blocking Applications. Therefore, this option has also been eliminated.

The last two options suffer from the same problems; moreover, they require human effort and time to keep their ad-blocking lists up-to-date. Therefore, another approach was needed to achieve the goal of blocking ads without engaging the smartphone in this process.

1) Our Ad-blocking Strategy: The approach used to tackle the ad-blocking problem and take full control of the web content crystallizes in having our own web hosting server. By doing so, we can either enable the ads or remove them completely from the webpages before they are even requested by the end user. The process works as follows: (1) choose certain web-sites; (2) download them and make two copies of every one; (3) modify and remove ad codes found in one copy and keep the other one as is; and (4) upload one copy to the server at a time to perform testing. We modify the the webpages manually by removing all the ad-related contents from the requested HTML documents. Figure 2 illustrates the sequence of the ad-blocking process and the final view that is displayed on the end user screen after the ad-blocking technique is applied.

B. Test Set-up

The test bench used to measure the energy consumption in smartphones is described in this section. Our experimental setup is shown in Fig. 3. We use a Monsoon power monitor [11] to power the smartphone and accurately measure the current drawn by smartphone in real-time. The power monitor is connected to a personal computer (PC) where the measurements are received and stored for off-line analysis. We performed our experiments on a *Galaxy Nexus* smartphone running *Android V4.2.1.* The phone settings during the testing were kept constant to provide consistent measurements.

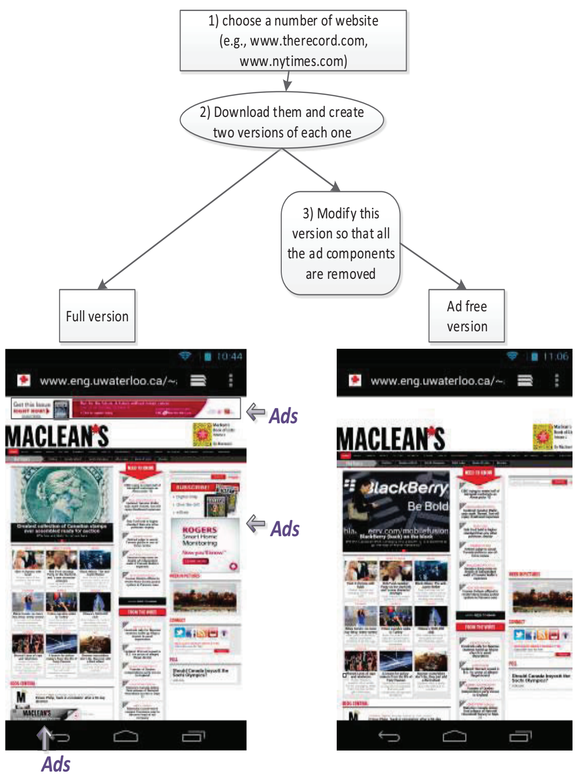


Fig. 2. Ad blocking strategy

To conduct the experiments, we chose the built-in *Android Browser* on *Galaxy Nexus* smartphone and another browsing application called "semi web browser" that we developed. The energy cost experiments were done over a *3G* cellular link and a *Wi-Fi* private access point (AP) installed in our lab. The AP is a *Cisco Linksys*, and it supports the *IEEE802.11g* interface. For the *3G* cellular connection, an *HSPA+ Bell* prepaid SIM card was used. This connection provides a typical speed of 7 - 14 Mbit/s [12]. For the phone configurations, we followed the methodology proposed in reference [13], in order to have consistent measurements.

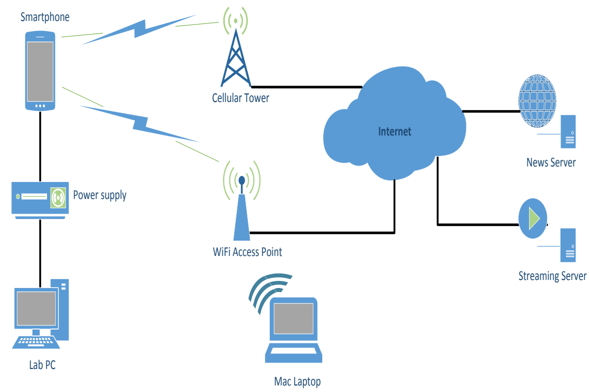


Fig. 3. Measurement Set-up.

To ensure the measured energy is the real cost of requesting webpages, we make sure that no data is cached and all the web elements will be requested entirely fresh from the web server in each web request. Therefore, we clear the browser's cache before every experiment and also disable all the other network-related apps (*e.g.*, Android update libraries.)

1) Bandwidth and Network Test Set-up: The test bench used to monitor the traffic going from and to the smartphone is shown in Fig. 3. We installed Wireshark on a Mac-book laptop to leverage the *Monitor Mode* feature available on Mac-book laptops. This feature allows the packet sniffing task to be performed passively; that is, we can capture all the traffic exchanged between the AP and the smartphone without involving any activities on the phone. Wireshark is a powerful monitoring and analysis tool, and enables us to monitor packet information from the radio level up to the application level packet information. The collected packet traces are analysed to classify how much bandwidth web ads consume and how much bandwidth the non-ad related web contents consume.

IV. Measurements

A. Test Cases for Energy Measurements

Using the described testing bench for measuring the energy consumption we were able to measure the instantaneous current drawn from the battery when a user is accessing the web. We started our energy experiments by testing how much energy is consumed when browsing a number of websites. The terms "drained current" and "consumed energy" are used interchangeably throughout this paper due to the direct relationship that connects both of them, as shown by the equation $E = I * V * t$, where E is the energy, I is the current, V is the constant voltage, and t is the time duration.

We chose a number of websites: (i) the most-accessed local news websites; (ii) a highly ranked international news website; and (iii) three samples of highly ranked Canadian magazine websites, according to Alexa [14]. We surfed the full version of each website, using *Android* web browser over *Wi-Fi* and

TABLE I
SUMMARY THE TRAFFIC STATISTICS WHILE WEB BROWSING WITH AND WITHOUT ADS

Websites	# of servers with Ads	# of servers without Ads	Time to download the page with ads(Sec)	Time to download the page without ads	Energy difference (J)
www.therecord.com	17	9	10.5	8	15
www.nytimes.com	23	19	10	6	6.2
www.macleans.com	35	22	7	11	6.1
www.thestar.com	29	17	7.5	5	10.6
www.canoe.com	19	16	15.5	15	4.5

3G connections. Figure. 5 (the solid line) illustrates the energy behaviour of a number of news websites.

First of all, we noticed a common energy-consumption footprint for mobile web browsing. As can be seen in Fig. 5 (the solid line), the energy behaviour starts with the phone's idling energy, which is the state before a user launches the web browser. This state is marked on the figures and found to be approximately 300 *mA (milli Ampere)*, and is followed by high-energy activities that last for some of time. The high-energy-activity period represents the energy needed to: (i) launch the browser app; (ii) download the web content (the network/radio interface energy cost); (iii) render the received HTML file (the computational energy cost), and (iv) display what is being rendered and is ready to be displayed on the user screen. The next energy state represents the energy cost only of displaying the entire webpage content. In general, we found that browsing a full version of the website over Wi-Fi for 30 seconds consumes $\sim$60 to 73.9 Joules.

To correlate the energy activities with the displayed web contents, we started from an end-user point of view and closely examined Fig. 5. We noticed, for instance, that periodic energy spikes occur approximately every 5 seconds in Fig. 5(b). To investigate the real cause of such an energy activity, we looked into the contents displayed on the phone's screen and found that these spikes are due to displaying an ad content. This ad content is basically a dynamic ad that displays different ads every 5 seconds. We then went further to the application layer and investigated the *HTML* document that contains this ad content. We found out that this ad corresponds to a block of *JavaScript* code, and it requires high computational power to render and display.

B. Energy-Bandwidth Mapping in Webpages

In an attempt to understand the energy behaviour of web browsing, we correlated the energy consumption footprint to the network activities needed to download web contents. Doing so, we can identify how long fetching a web component takes, and consequently, how much energy doing so consumes. Using the test bench, we conducted an offline analysis of the traffic being transferred from and to the smartphone while a user is accessing the web. We, then, mapped the network activities (that is, the loading time for each TCP connection) onto the consumed energy. As was expected, we notice that there are multiple TCP connections established in parallel. Multiple parallel TCP connections are typically needed to speed up the retrieval of web contents. The first content that the browser downloads is typically the index HTML document. Once this document is downloaded, the browser starts to parse and script its contents. By doing so, the browser finds other web resources (images, CSS, JavaScript files, and ads) to be fetched. Consequently, more TCP connections are established. Now, since the core web content and the ad content come from different servers, in some cases 3 to 13 TCP connections were found to be dedicated to ad servers and to analytic servers that gather statistics and other information about users. Table. I lists the number of TCP connections (# of servers) needed for advertising purposes.

As explained in Section. I, for the browser to display web-pages, all web content referenced/tagged in the index HTML document, certain processes are required. The browser has to: (i) obtain the IP (Internet Protocol) address corresponding to the URL referencing an web object; (ii) handle the HTTP request that is to be made; (iii) establish the TCP connection with the server containing that web object; and (iv) render that object based on the rules specified in the CSS. These time-consuming processes are shown in Fig. 5 to be under the time period of a high energy state, which is not very surprising considering the high computations required by the browser to accomplish the task of displaying the requested webpages. We noticed also that the overall time needed to download all the webpage contents is around 10 seconds, in some cases. Approximately 2 seconds beyond those 10 seconds show high energy activity; we believe that is due to the computation needed to render the remaining fetched objects (objects that are downloaded with no network activities are involved). We found it very difficult to separate the energy consumed by the rendering process and the radio/network interface activity. Moreover, it is important to mention here that the objective of this paper is to quantify the web advertisement energy and bandwidth impact, not to focus on analysing the energy consumption of web browsers.

To identify the relationship between the number of TCP connections and the corresponding energy consumption, we modified some webpages using our Ad-blocking strategy. Applying this strategy allowed us to change the number of TCP connections that are opened. Then, we compared the energy consumption in both cases for each website, and found, as expected, that the more TCP connections we have, the more energy is consumed, as shown in Table I. Moreover, as the number of TCP connections increases, the fetching loading time increases as well, and so does the energy consumption.

Downloading the ad contents prolongs the active period of the radio interface, and hence increases the energy consumption.

C. Energy Impact of Ads over Wi-Fi

To study the impact of web advertising on energy, we started by browsing the full version of a number of websites over a Wi-Fi network, using the phone's built-in *Android Browser*. Then, we applied our Ad-blocking strategy, described in section III-A1, and repeated the same experiments that were conducted for the full versions of the websites. Next, we compared the energy needed to download webpages with and without ads. Figure 5 shows the energy consumption for a number of websites. The dashed line refers to the webpages that are ads-free and the solid line refers to the full version of webpages (webpages with ads.) It is worth mentioning here that during our energy measurements we performed passive network sniffing over the Wi-Fi connection as well. Our offline analysis of the gathered network traces confirmed our previous argument that the more TCP connections there are, the longer the loading time is, and consequently, the more energy is consumed. As shown in Table I, the energy overhead due to ads ranges from 4.5 to 15 Joules.

Reliability of Measurements: To ensure the reliability of our measurements, we developed a semi-web browser on *Android*. To ensure more reliable measurements, the experiments were repeated multiple times. We wanted to keep the same settings throughout the experiments, and, most importantly, minimise end-user interactions with the smartphone. Therefore, we developed our own web browser. Doing so: (i) enabled us to repeat the same experiments a number (7 times) of times at consistent time intervals of 30 seconds; (ii) meant that no information was allowed to be cached; (iii) minimized end-user interactions with the phone; that is, a single tap on the app's icon was enough to start the browsing.

Figure 4 shows the power footprint while using our browser app. Moreover, we conducted a number of tests just to ensure that our app energy consumption did not differ a lot from the built-in *Android Browser*. We found in many cases that both measurements were very close, and in some cases a 100% match was noticed, as the measurements show in Table I and Table II.

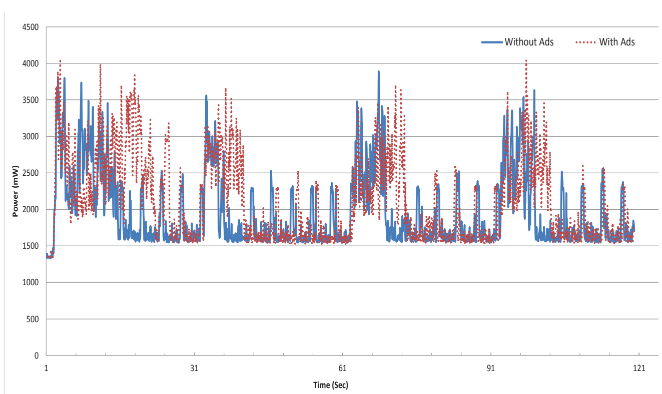


Fig. 4. Energy measurement of our browser

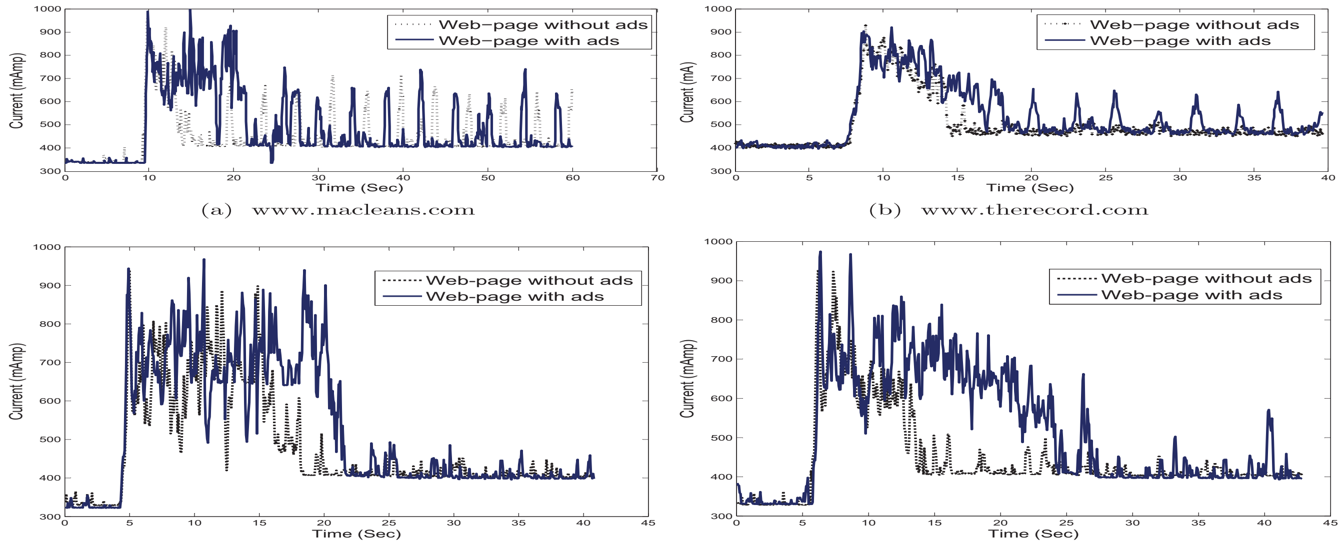
Table II summarizes the energy consumed by a number of websites, with and without ads. The differences in energy consumption vary from 6 - 17.5%. Again, these figures are due to what we call ads-overhead, that is, the net energy cost of the processes of downloading, rendering, and displaying web ads over a Wi-Fi network. We noticed that the ads contents vary each time a request is made, and observed that webpages with ads take longer to load. As we have shown in Fig. 5, around 4 to 10 seconds of extra time is needed to download and display ads. These times are reflected in the form of extra energy, as the measurements in Table II show. We compared the extra times introduced by the existence of ads just for downloading (Table I) and the overall time added due to downloading and rendering ads (Fig. 5.) In some cases, most ads energy consumption was due to the process of rendering. As the test case of browsing *www.canoe.com* shows, downloading the ads requires only half a second extra time, while the power footprint in Fig. 5-(d) shows an extra 10 seconds of high activity energy state. Moreover, Table II shows an estimation of battery life, *i.e.*, how long the battery would last if we were to browse each version of the websites repeatedly until the battery died. These numbers illustrate the energy wasted by ads in web.

D. Energy Impact of Ads over 3G

We repeated the experiments of Section IV-C over a 3G connection. Table II summarizes the energy consumed by browsing five websites, with and without ads. We noticed that browsing over a 3G connection consumes more energy in general. The energy overhead due to web ads ranges from 6 to 17.5%. However, the energy difference due to ads is less than the difference noticed over Wi-Fi. Moreover, we observed that browsing a full version of a website, with ads, over Wi-Fi, is less expensive than browsing an ad-free version of the same website over 3G connection, as shown in Fig. 6. With 3G usage, batteries die faster than with Wi-Fi, which was certainly expected taking into consideration the observations mentioned in this paragraph. The higher energy consumption over the 3G network is due to the energy tails and the capacity dissimilarity, according to [15]. "Energy tail" refers to the high energy state that the 3G radio interface stays in after the network activity is terminated, while the capacity dissimilarity refers to the 3G bandwidth capacity limit. Compared to Wi-Fi, 3G is slower, and consequently, downloading webpages over 3G will take longer, resulting in higher energy consumption.

E. Test Cases for Bandwidth Measurement

Using the previously described testing infrastructure for monitoring network activities while mobile web browsing, we were able to capture the network traces. We browsed five news websites and conducted an offline analysis to measure how much bandwidth ads downloading require. Using Wireshark traces, we investigated the traffic needed to download ad components in webpages, and found that in one case, surprisingly enough, ads traffic comprised almost 50% of the traffic needed to download some news webpages. More



(a) www.macleans.com (b) www.therecord.com

(c) www.thestar.com (d) www.canoe.com

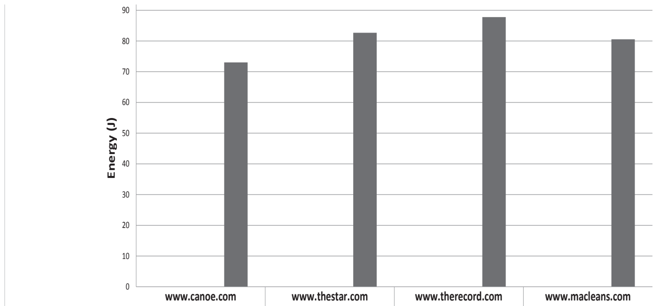
Fig. 5. The real-time energy footprint of a number of websites over a Wi-Fi connection

TABLE II
THE COST OF ADS OVER A WI-FI/3G INTERFACES

Website	Browsing	Energy (J) WiFi/3G	Energy Overhead due to Ads WiFi/3G	Difference in the Estimated Battery Life (min) WiFi/3G
www.therecord.com	browsing **with ads** browsing **without ads**	68.5/87.7 59/76.7	16%/14%	27/22
www.nytimes.com	browsing **with ads** browsing **without ads**	60/72.8 55/68.2	9%/7%	23/15
www.macleans.com	browsing **with ads** browsing **without ads**	65/80.5 58/74.4	12%/8.2%	17/14
www.thestar.com	browsing **with ads** browsing **without ads**	67/82.6 57/70.2	17.5%/17.5%	35/28
www.canoe.com	browsing **with ads** browsing **without ads**	61.5/73 58/69	6%/6%	8/5

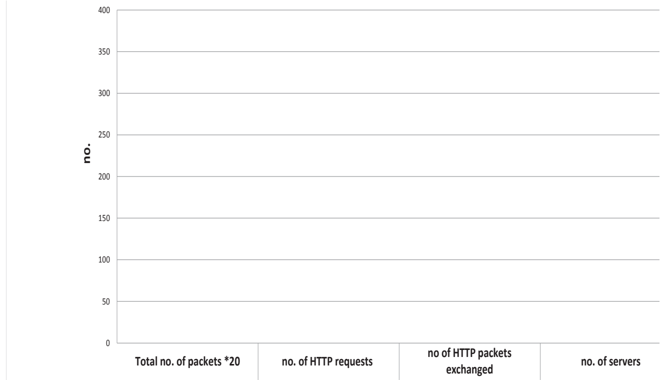
TABLE III
TRAFFIC BREAKDOWN PER WEBPAGE

Web site	Total page traffic (**with ads**) in Bytes	Total page traffic (**without ads**)in Bytes	Total ads traffic in Bytes
www.therecord.com	3,025,082	1,449,882	1,575,200 (52%)
www.nytimes.com	975,646	634,018	341,628 (35%)
www.thestar.com	714,961	351,732	363,229 (50%)
www.macleans.com	987,682	617,857	369,825 (37%)
www.canoe.com	6,005,729	5,861,272	144,457 (2.5%)



	www.canoe.com	www.thestar.com	www.therecord.com	www.macleans.com
Website Without Ads (WiFi)	58	57	59	58
Website With Ads (WiFi)	61.5	67	68.5	65
Website Without Ads (3G)	69.06	70.29	76.74	74.44
Website With Ads (3G)	73.05	82.65	87.75	80.55

Fig. 6. The energy consumption of different websites over 3G and Wi-Fi networks



	Total no. of packets *20	no. of HTTP requests	no of HTTP packets exchanged	no. of servers
Website Without Ads	128.45	53	129	10
Website With Ads	262.85	152	335	18

Fig. 7. Traffic statistics while web browsing with and without ads.

detailed traffic statistics are shown in Fig. 7, where we see that when ads are enabled, 8 extra servers are contacted. We also observe that the number of HTTP requests made by the client (smartphone) is almost three times more when ads are enabled, and the total number of packets exchanged between the client and the access point is doubled. Table III shows the overhead bytes needed to download ads. These numbers are average values since the ad contents vary each time a request is made.

In summary, we found that ads consume a considerable amount of energy and bandwidth. To have a better sense of the effective cost of ads, we, next, give an empirical estimation of (i) how much energy a smartphone would consume to download and display ads, and (ii) how much bandwidth it would cost a user to download these ads for a certain period of time. We assume that a user spends around one hour on web browsing a day, not necessarily continuously. In this hour, she would spend 2 minutes per website; therefore, on average, she would access at least 30 websites. Based on Tables II and III, we get the following:

- the energy that a smartphone would consume, to download and display ads in one hour of browsing is 360 Jules/day; and
- the amount of bandwidth needed to download ads would be equal to 2.6 Mbyte/day, that is 78 Mbyte/month. Based on some US metered data plans (usage-based pricing plans) [9], $ 12.48 would be spent just on downloading ads.

V. Conclusion

In this paper, we presented the architecture of a test bench to measure the energy cost and the bandwidth cost of ads in smartphone while browsing the web. We focused on the energy and bandwidth impact of the existence of ads in webpages and noticed many interesting observations. We investigated the energy consumption due to ads in five well-known websites, and found that ads can consume $\sim$ 3.5 to 12 Joules over Wi-Fi and 3G networks, that is $\sim$ 6 to 18 % of the total energy of web browsing. This high cost of ads on smartphones should be considered by the designers and vendors of apps. For future work, the test bench of the evaluated websites can be further extended. Instead of evaluating only five websites, more can be covered. In the time being, we are working on modifying the browser to automatically classify the web contents into useful and unuseful "unwanted" contents. We will consider fully automating these tests.

References

[1] A. Pathak, Y. Hu, and M. Zhang, "Where is the energy spent inside my app?: fine grained energy accounting on smartphones with eprof," in *Proceedings of the 7th ACM european conference on Computer Systems*. ACM, 2012, pp. 29–42.

[2] N. Vallina-Rodriguez, J. Shah, A. Finamore, Y. Grunenberger, H. Haddadi, and J. Crowcroft, "Breaking for commercials: Characterizing mobile advertising," pp. 343–356, 2012.

[3] N. Vallina-Rodriguez and J. Crowcroft, "Energy management techniques in modern mobile handsets," *Communications Surveys Tutorials, IEEE*, vol. 15, no. 1, pp. 179–198, 2013.

[4] R. Simons and A. Pras, "The hidden energy cost of web advertising," 2010.

[5] N. Thiagarajan, G. Aggarwal, A. Nicoara, D. Boneh, and J. Singh, "Who killed my battery: analyzing mobile browser energy consumption," in *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012.

[6] Z. Wang, F. Lin, L. Zhong, and M. Chishtie, "Why are web browsers slow on smartphones?" in *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications*. ACM, 2011, pp. 91–96.

[7] J. Hoehl and C. Lewis, "Mobile web on the desktop: simpler web browsing," in *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*. ACM, 2011, pp. 263–264.

[8] J. Erman, A. Gerber, and S. Sen, "Http in the home: It is not just about pcs," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 1, pp. 90–95, 2011.

[9] J. van den Brande and A. Pras, "The costs of web advertisements while mobile browsing," in *Information and Communication Technologies*. Springer, 2012, pp. 412–422.

[10] Privoxy, "Privoxy," *URL http://www.privoxy.org/*.

[11] M. Solutions, "Power monitor," http://www.msoon.com/LabEquipment/PowerMonitor/.

[12] "Bell hspa/umts," http://mobilesyrup.com/2010/12/12/review-bell-novatel-wireless-u547-turbo-stick-42-mbps/, 2009.

[13] A. Abogharaf, R. Palit, K. Naik, and A. Singh, "A methodology for energy performance testing of smartphone applications," in *The 7th International Workshop on Automation of Software Test (AST) 2012*, 2012, pp. 110–116.

[14] "Alexa, the web information company," http://www.alexa.com/, 2013.

[15] F. Qian, Z. Wang, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "Characterizing radio resource allocation for 3g networks," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. ACM, 2010, pp. 137–150.