2012 Sixth International Conference on Next Generation Mobile Applications, Services and Technologies

Energy Cost of Advertisements in Mobile Games on the Android Platform

Irena Prochkova†, Varun Singh‡, Jukka K. Nurminen†
† *Dept. of Computer Science, Aalto University, Espoo, Finland*
‡ *Dept. of Communication and Networking, Aalto University, Espoo, Finland*
{*irena.prochkova, varun.singh, jukka.k.nurminen*}*@aalto.fi*

***Abstract*—Advertisements (ads) are a common source of revenue for monetizing free mobile applications, but increase the energy consumption of a mobile device because the advertisement have to be fetched and displayed. In particular, the radio communication needed for the data transfer is power hungry and therefore advertisement sponsored free applications consume more energy than paid apps. This energy cost can be justified for Internet and social-applications but not for standalone games where always-on Internet connectivity is not necessarily required. In this paper, we analyze the energy consumed by mobile advertisements in an Android Phone. We measure the energy consumption by comparing the power usage of mobile phone games with and without advertisements. Our results show that advertisements in general consume significant amount of energy, but the consumption depends on how frequently advertisements are requested from the advertisement server (AdMob Server). In applications where advertisements are fetched frequently the energy consumed is higher, while in applications where advertisements are fetched rarely, it is lower. Moreover, we observe that the AdMob Server does not always send a new advertisement for each request but responds with a HTTP 304 (Resource Not Modified) leading to unnecessary usage of network and energy.**

***Keywords*-advertising, energy-efficiency, mobile advertising**

I. Introduction

The wide adoption of mobile technologies and applications, and the significant increase in the number of mobile users (> 5.6 billion [1]) has created a strong demand for mobile advertising. Mobile advertising is a form of marketing via mobile phones which promote brands, businesses, services and organizations to the end customers [2]. In general, advertisements are implemented in various mobile applications and services such as, SMS, web pages, mobile games and applications, and even videos. All of these forms of advertising use different ways of representing the advertisement (such as, text, images, or multimedia) for their marketing campaigns. In web advertising, advertisements are integrated in web pages while in mobile applications advertisements are integrated inside the native applications.

One of the most interesting trends recently is to display advertisements inside mobile games through an advertisement platforms. Many games for iPhone and Android such as, *Angry Birds*, *Skater Boy*, etc, show advertisement inside the game logic and users can interact with it while playing the game. These ads are usually sent from the advertising platform (AdMob on Android, iAd on Apple) and require internet connectivity. However, using the Internet just for receiving advertisements in the game is not in the favor of the users as it may significantly increase the consumption of the battery. More frequent rate of advertisements means increased network traffic and higher power consumption. This can lead to higher rate of the user's battery consumption and the users may not be even aware of it. Therefore, it is really important to investigate how the power consumption of the phone varies with the frequency of fetching advertisements i.e., the cost of transferring advertisements on the network.

The paper is arranged as follows: Section II describes the most common mobile advertising systems available today. Section III discusses about the related work and currently available analysis to energy-efficiency in mobile and web advertisements. Section IV describes the measurement environment and how the experiments were performed. In Section V we compare the power usage of a baseline application to 5 popular mobile games and provides a guideline to developers based on our observations. Finally, Section VI concludes the paper.

II. Background

In this section, we are going to discuss the current existing mobile advertising systems including, SMS, MMS, mobile web and in-application advertising.

A. Mobile Advertising Trends

Mobile advertising as a relatively new media has many forms of advertising. The Mobile Marketing Association (MMA) [3] has defined five main types of mobile advertisements [4], they are:

1) **Mobile Web Ads** is advertising through mobile websites displaying text advertisements or graphical banners throughout the web page. The former displays advertisements with a text and link to the advertised website, while the latter uses images or rich media content. **Mobile Web Banner Ads** are an example of banner advertisements that consist of still images residing on top or bottom of the page. Similarly, **Rich Media Mobile Ads** display images along with audio, video or graphical animation.
2) **SMS Ads** allows advertisers to reach their audience via textual messages sent to all subscribers. The subscribers receive the advertisement on their phone and

978-0-7695-4803-6/12 $26.00 © 2012 IEEE
DOI 10.1109/NGMAST.2012.32

they can further connect to an advertised link or call the advertised company, included in the SMS.

3) **MMS Ads** is advertising through rich media messaging service that sends images, graphics, audio, video and text, etc, in the message. This media does not need Internet connection to present the ad, instead, the whole advertising components are sent inside the MMS and are stored locally on the phone.
4) **Mobile Video and TV Ads** involves advertising through multimedia content. These ads are delivered over a mobile phone network and displayed on the device. This is relatively new type of advertising which gives advertisers an opportunity to create emotive, informative and personal content. Common ways of video advertising are Ad Breaks, consisting of image or video displayed as breaks before, during or after the video. Additionally, advertisements can be displayed in parallel with the streamed video. As an illustrative instance, transparent full screen images or videos can appear overlaid on the current video.
5) **Mobile Application Ads** is an advertising method that promotes brands within the mobile application logic or design. These advertisements are similar to the image banners in web advertising. Moreover, to increase visibility application developers may show full screen advertisements with images or rich media at the launch and exit of the application, or between levels in mobile games. Another type of mobile application advertisement is **Integrated Ads** and are integrated into the application content, like within games as promotional game objects (e.g., displayed on a truck or on a ball).

B. Mobile Advertising Networks

The types of advertisements discussed in section II-A are usually implemented with a help of an advertising platform. Mobile advertising network connects advertisers and publishers. Advertisers are people who start a campaign to promote some product, and publishers are people who publish the advertisements inside their app or website. Examples of mobile advertising networks are AdMob[1], iAds[2], inMobi[3], MadHouse[4] and others.

AdMob is the largest mobile advertising networks with around 100,000 publishers and 10,000 advertisers [5]. It is owned by Google and provides two different mobile advertising approaches: through mobile websites and apps. In both cases text advertisements and image banner advertisements are available. While advertisements on AdMob are mainly static with web links, some advertisements can be expanded from still image banners to full screen banners, presenting a richer behavior for the user.

[1]http://www.admob.com
[2]http://advertising.apple.com
[3]http://www.inmobi.com
[4]http://www.madhouse.cn/en/

iAd, is mobile advertising platform owned by Apple for advertising on iPhone and iPad devices which has less than 1000 publishers and 70 unique advertisers. Like AdMob, iAd advertises through image banners inside mobile applications, but has different approach when it comes to user interaction. Tapping the banner opens full-screen advertisements that resides inside the application itself, so users can view product information without leaving the application. iAd also offers more interactive advertisements with multimedia, interactivity with buttons, and gestures like swiping and touching effects similar to a normal native app on the iPhone.

InMobi and MadHouse are mobile advertising networks for the Asian market. InMobi is developed and widely spread in India, while MadHouse is for Chinese region. Similar to the previous advertising networks, both of these advertisement networks implement image banners, text advertisements, rich media content for mobile application and mobile websites.

III. Related Work

There have been some studies done related to energy consumption of web advertisements [6] but mainly focus on the power consumption on a desktop PCs. They indicate that the average additional power consumed to display web advertisements is roughly 2.5W. The resulting energy consumption is equal to the total yearly electricity consumption of 2000 households in Netherlands [6]. Another study [7] by AnandTech compares web browsers on laptops to determine which browsers are most energy efficient. As a side-project [8], they measured the effect of web advertisements on the battery life of some laptops and found that the battery of laptops lasted longer when advertisements are blocked, than when they are not. For AMD Athlon CPU the battery lasted 4.3% longer and on Intel Core 2 Duo CPU 5.7% .

Desktop PC is very different from mobile phones because the computer is connected to constant power supply and there is no limitation on browsing the Internet or displaying advertisements while a mobile phone may run out of battery and therefore become unusable until recharged. As an application developer, being unaware of the energy consumed by advertisements may lead to negative user feedback towards the mobile manufacturer or towards the specific application.

[9] claims that up to 75% of the energy consumed by free Android apps is spent fetching advertisements, tracking user's location for better advertisement context and uploading the user's data. The analysis in our paper differs from [10] by just focusing on the power consumption for fetching advertisements (due to network usage), we do not profile the application to observe which other services (e.g., GPS, disk I/O, accelerometer, etc.) consume energy. One other key difference between our measurements and [10] is that we use a hardware power measurement tool [11] instead of modeling the power usage [12], [10], which allows us to accurately study the power consumption.

IV. Measurement Environment

The energy measurements were performed on "Samsung Galaxy" running Android OS v3.2. During the measurement, all background applications were turned off and the only services that were running were the stock Android services, which take up $85MB$ from total $300MB$ phone memory. If the display is active, the WiFi does not go to sleep [13]. Most android devices use AdMob Server to fetch advertisements and works in conjunction with the Google AdSense service to provide advertisement context. For measuring power consumption we use Power Monitor [11] hardware and Power Tool [11] software. These tools provide robust power measurement for any mobile devices that uses a single lithium battery [14]. The Power Monitor can measure power (in mW), current (in mA) and has an voltage range from $2.1V$–$4.5V$. Furthermore, the Power Tool stores data in CSV files, display graphs depending and provide statistics like average energy/power consumed, average current and expected battery life.

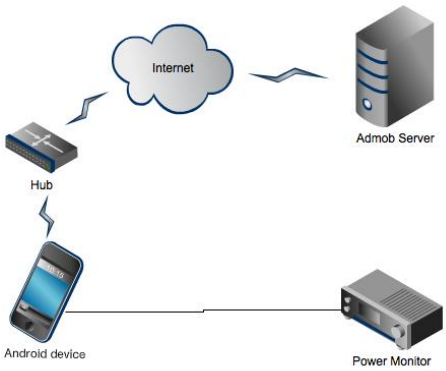

Figure 1: Setup to measure energy consumption using Power Monitor and Power Tool. Power Monitor is the hardware while Power Tool is the software which stores the real-time statistics (Power, Voltage, Current, etc).

The Figure 1 shows the measurement setup. The $PowerUsage$ is measured every 10–15ms and averaged at one second intervals over 15 minutes for both scenarios (with and without advertisements). The min, max and average $\Delta PowerUsage$ is calculated by subtracting the power usage of the application without advertisements from the power usage of the same app with the advertisements.

The following approaches are available to measure the power usage of mobile apps without advertisement:

1) Compare the free (typically, Ad enabled) and paid version (typically, Ad disabled) of the same app. Unfortunately, there are not that many applications on the Android market that have both their free and paid apps available. The reason for this is that Google promotes free applications for Android devices.
2) Use AdFree application for Android, which blocks advertisements. However this app requires root access on the mobile phone. Jail-breaking the phone is not a safe option as it may change the OS configurations and thus interfere with our results.
3) Block advertisements by enabling and disabling internet connection. When there is no internet connection advertisements are not received in the application, but when there the internet is on, the advertisements are received. However, apart from the advertisements, the application may send/receive other traffic. We can use Wireshark[5] to detect this other traffic and use a proxy to disable it.

We used the third approach because it not only guarantees that the advertisements are definitely blocked but also reduces power consumption due to network usage. So the difference in the power usage of the two scenarios will yield the maximum possible power savings. To confirm when exactly the advertisements are requested from the AdMob Server, we used Wireshark and set filters to capture every packet sent to/from the device. The Figure 2 shows the measurement setup to capture Internet traffic to/from the mobile device.

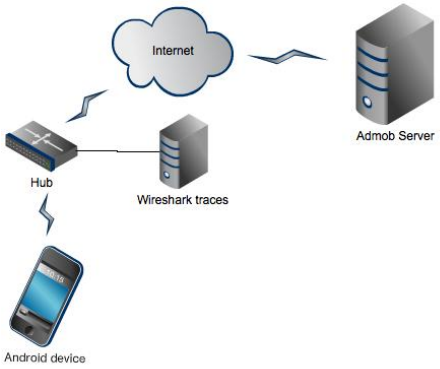

Figure 2: Setup to capture the network activity from the mobile app using Wireshark. These traces can be used to classify the network usage between advertisements, game-related, etc.

A. Mobile Game Usage

When playing the game these considerations need to be taken into account: The usage of the applications should be "human-like": play for 15 minutes. Users typically play mobile games when they find a few moments of idle time, for example when commuting to/from work, at airports, waiting for people to arrive, etc. For measurement purposes, we play one game at a time and play each mobile game for 15 minutes with advertisements enabled and for 15 minutes with advertisements disabled.

[5]http://www.wireshark.org/

V. Performance Analysis

In this section, we observe the power usage of a baseline Android application and compare it with the power usage of 5 popular mobile games.

A. Benchmarking

We wrote a simple Android application to measure the power usage during fetching and displaying a mobile advertisement. All the advertisements were fetched over WiFi and the 3G connection was turned off. The baseline application periodically queries the AdMob Server for advertisements and on receiving the advertisement it just renders them. Currently, the shortest interval of fetching a new advertisement from AdMob [15] is $12s$ and the longest interval is $120s$. We fetch advertisements at the following intervals: 12, 15, 30, 45, 60, 90, and 120 seconds and run the baseline application for 15 minutes for each advertisement interval. Usually, the advertisement frequency is controlled by the developers via the AdMob SDK/APIs or directly on the AdMob website [15].

Advertisement Frequency	ΔPowerUsage [in mW] (*Min–Max*)	(*Average*)	% PowerUsage by Ads	Total No. of Ad Req. in 15 mins	No. of Ad Requests denied by Ad Server
every 12s	77–114	(97)	21%	75	17
every 15s	72–103	(90)	19%	60	9
every 30s	41–93	(82)	18%	30	5
every 45s	38–93	(80)	17%	20	0
every 60s	38–90	(67)	15%	15	0
every 90s	24–73	(52)	12%	10	0
every 120s	22–88	(56)	13%	7	0

Table I: Average power consumption of advertisements in a baseline application running for 15 minutes. The baseline power consumption for an app without wireless and advertisements is about 370 mW. Average number of HTTP requests per Ad is 6.5. The total number of Ad Requests include the requests that were denied by the AdMob Server

As expected, Table I shows that the average power usage decreases when the advertisement frequency is reduced. There is a big variation in the Min and Max values of $\Delta PowerUsage$ because the power usage is averaged over $1s$ intervals and intervals that have do not fetch advertisements consume less energy than those intervals that fetch advertisements. We also calculate the $\%PowerUsage = \frac{\Delta PowerUsage}{PowerUsage_{withAds}}$ and observe that it is lower for mobile apps with low advertisement frequency (12%) than for apps with high advertisement frequency (21%).

Wireshark traces show that the average number of HTTP requests for fetching an advertisement is 6.5. Additionally, we also observe that in some cases the AdMob Server denies the Ad Request with a HTTP 304 Response (*resource not modified*) because the advertisement content has not changed since the last Ad Request and is clearly observed when the advertisement intervals are short (12–30s).

B. Mobile Games

We evaluate the power usage of 5 mobile games, namely, *Angry Birds* [16], *Fragger* [17], *Yoo Ninja* [18], *Skater Boy* [19], and *Ceramic Destroyer* [20]. All these games are the amongst the top 10 downloaded apps on the Android Market [21] and they all use AdMob platform for advertising. Table II shows the $\Delta PowerUsage$ for each of the aforementioned mobile apps and observe that the $\Delta PowerUsage$ is higher for apps that frequently query for advertisements because they use more network resources. For example: *Angry birds* (15%), *Fragger* (15%) use more energy for advertisements than *Yoo Ninja* (8%), *Ceramic Destroyer* (4%) and *Skater Boy* (4%). Furthermore, the apps that frequently query for advertisements do not gain much because the AdMob Server returns the same content. We observe that the AdMob Server denies frequent advertisement requests with a HTTP 304 (*Resource Not Changed*) Response.

Game	ΔPowerUsage [in mW] (Min–Max)	(Average)	% PowerUsage by Ads	Total no. of Ads Served	No. of Ad Requests denied by Ad Server
Angry Birds	110–136	(131)	15%	67–80	21–30
Fragger	157–225	(157)	15%	32–39	12–14
Yoo Ninja	70–158	(71)	8%	14–17	2–7
Skater Boy	40–53	(42)	4%	6–8	0–1
Ceramic	66–88	(43)	4%	6–7	0

Table II: Average power consumption of advertisements in mobile games. The gameplay duration is 15 minutes.

Our experiments show that the following percentage of the HTTP requests result in no new advertisements; *Angry birds* (37%), *Fragger* (35%), and *Yoo Ninja* (11%). We also observe that some games do not follow the minimum advertisement interval ($12s$) specified by AdMob/Google. Figure 3 shows that the mobile game *Fragger* requests advertisement on level completion or restart. When the level is completed or reset quickly then it may happen that the advertisement request may be shorter than the minimum interval defined by AdMob and result in a HTTP 304 response from the AdMob Server.

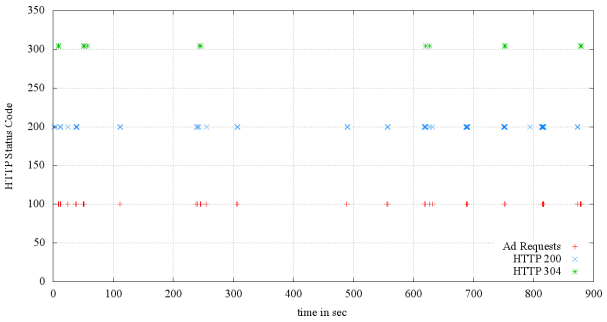

Figure 3: shows that mobile game *Fragger* does not follow the minimum advertisement interval and sometimes requests advertisements at very short intervals, to which the Advertisement Server responds with a HTTP 304 (*Not Changed*) Response. The aperiodicity of the advertisement interval may be because the mobile game requests for new advertisements on each level completion or restart.

From the *"Total No. of Ads Served"* in Table II, we note that: *Ceramic* and *Skater Boy* use a long advertising

interval of 1 advertisement per $120s$ and *Yoo Ninja* uses 1 advertisement per $60s$. Interestingly these three games are published by the same game publisher and this may explain why these apps use similar advertisement intervals. Similarly, we note that *Angry Birds* and *Fragger* use short intervals of 1 advertisement per 15–$30s$.

C. Observations

Although we do not have explicit measurements on energy consumption when cellular operator network (e.g., 3G) is used for communication, we can estimate that energy consumption would be higher in a cellular network. This increased energy consumption comes from two sources. First, transferring the same amount of data in a cellular network normally takes more energy than with WiFi. Second, the long time before the cellular radio returns to idle state after data transfer (tail energy) would be particularly harmful in a scenario where periodic small data transfers take place [22]. 3G and WiFi can be easily shown by comparing the energy usage by advertisement in Angry Birds [10] to our results.

Even though the Power Monitor [11] used in our experiments records actual power/power consumption values than Eprof [12] (which uses an energy model) the results are still comparable. Eprof shows that 3G's tail energy used to fetch advertisements in the Angry Birds mobile game consumes 28% of the application energy [10], while our observations show that the WiFi uses 15% of the application energy to fetch and render advertisements. This shows that the energy consumed to fetch advertisements will be higher for 3G than for WiFi.

D. Development Considerations for Power Saving in Android Phones using AdMob

To summarize, we outline some guidelines based on our experience with implementing the baseline application. They are:

- The application should adhere to the minimum ($12s$) and maximum ($120s$) advertisement interval set by AdMob. Applications that do not require constant Internet access should use long advertisements intervals; preferably an interval of 60–$120s$ between advertisements.
- The recommended WiFi sleep setting in Android devices is WIFI_SLEEP_POLICY_DEFAULT. Furthermore, if the applications does not run in the background then the developer should make sure that they do not set the sleep settings to WIFI_SLEEP_POLICY_NEVER because the WiFi will remain active even when the screen is inactive, effectively draining the battery when the application is paused. The user may set the WiFi sleep timer from the phone's advanced settings but the application can override it.
- If a mobile application receives a HTTP 304 response to an Ad Request, it should consider increasing the advertisement interval. This can also be done adaptively; a mobile application can increase the advertisement interval when it receives a HTTP 304 Response (*Content Not Changed*) and reduce the interval when an advertisement is successfully received (HTTP 200 OK). We recommend conservative values for increasing and decreasing the advertisement interval so that it does not fluctuate rapidly between Ad Requests.
- For mobile games where advertisements may be shown at the completion/restart of each level, game programers should keep track of the time since the last advertisement request. so that the application does not request for an advertisements before the minimum advertisement interval. Instead the mobile game should just show the old advertisement.
- AdMob does not support pipelining multiple Ad Requests in a single HTTP request, but an alternative Ad Server may support this feature. The application instead of frequently polling for new advertisements would transfer multiple advertisements in a single request and then decide by itself when to display a new advertisement from the set of available advertisements. In this way the amount of communication events would be reduced resulting in reduced energy consumption.

VI. Conclusion

In this paper we analyze the difference in average power consumption of mobile applications with and without advertisements. We show that that the advertisement interval plays an important role in the power consumption and that mobile apps which do not require *always-on* intermittent connectivity should prefer larger advertisement intervals. If the time between two advertisement requests is short, we observe that the AdMob Server responds with a *Resource Not Changed* Response. As a temporary fix we propose that the applications use an adaptive advertisement interval based on the type of HTTP Response (*200 OK, 304 Not Changed*). Alternatively, AdMob should consider increasing the minimum interval.

Our results from profiling 5 mobile games show that requesting advertisement can use up 4–15% of the total energy consumption depending on the frequency of fetching the advertisements. Furthermore, our baseline application that just renders advertisements indicates that the average power consumed is 12–21% depending on the frequency of fetching the advertisements.

These are preliminary results and are focused around mobile games; for a more detailed guidelines the experiment would need to cover a larger spectrum of mobile apps like Multimedia, Productivity, Social Games, Web-based mobile apps etc. In this paper, we have only considered network usage as a preliminary source of power savings because of the nature of the apps (Mobile Games). However, network usage may not be the primary energy saving source

for apps in different App Store categories. Therefore, we see expanding the set of applications as the next step of the analysis. Other topics for further research include measurements with cellular networks, experimenting with adaptive advertisement intervals, and studying other mobile advertisement systems and mechanisms.

ACKNOWLEDGMENT

This work was supported by the Academy of Finland, grant number 253860.

REFERENCES

[1] Wikipedia, "Mobile Phone Usage statistics," http://en.wikipedia.org/wiki/Mobile_phone.

[2] ——, "Mobile Advertising," http://en.wikipedia.org/wiki/Mobile_advertising.

[3] Mobile Market Association, http://mmaglobal.com/main/.

[4] ——, "Mobile Advertising Guidelines," http://mmaglobal.com/files/mobileadvertising.pdf/.

[5] Mark Fidelman, "A Comprehensive Comparison Guide to Mobile Advertising Networks ," http://www.seekomega.com/2010/12/a-comprehensive-comparison-guide-to-mobile-advertising-networks-infographic/.

[6] R. J. G. Simons and A. Pras, "The hidden energy cost of web advertising," http://eprints.eemcs.utwente.nl/18066/, Centre for Telematics and Information Technology University of Twente, Enschede, Technical Report, June 2010.

[7] AnandTech., "Browser Face-Off: Battery Life Explored," http://www.anandtech.com/show/2834/.

[8] Palant, Wladimir., "Adblock Plus: Save your time and traffic," http://adblockplus.org.

[9] J. Aron, "Free apps eat up your phone battery just sending ads," http://www.newscientist.com/article/mg21328566.400.html.

[10] A. Pathak, Y. C. Hu, and M. Zhang, "Where is the energy spent inside my app? fine grained energy accounting on smartphones with eprof," in *EuroSys '12*, july 2012.

[11] Monsoon Inc., "Power Monitor," http://msoon.com/LabEquipment/PowerMonitor/.

[12] A. Pathak, Y. C. Hu, M. Zhang, P. Bahl, and Y.-M. Wang, "Fine-grained power modeling for smartphones using system call tracing," in *EuroSys '11*. New York, NY, USA: ACM, 2011, pp. 153–168. [Online]. Available: http://doi.acm.org/10.1145/1966445.1966460

[13] Google Android SDK for WIFI Sleep interval, http://developer.android.com/reference/android/net/wifi/WifiManager.html.

[14] Monsoon Inc., "Mobile device power monitor manual," http://goo.gl/1PqQE.

[15] Google, "Admob refresh rate," http://www.admob.com/my_sites/tools/?mt=sss, AdMob offers the ability to control refresh rate from within your account. Simply log in and click the Manage Settings button under your app's name in the Sites and Apps tab. Your refresh rate is in App Settings.

[16] Rovio Mobile, "Angry Birds," https://play.google.com/store/apps/details?id=com.rovio.angrybirds.

[17] Miniclip.com, "Fragger," https://play.google.com/store/apps/details?id=com.miniclip.fraggerfree.

[18] Runnergames, "Yoo Ninja! Free," https://play.google.com/store/apps/details?id=com.RunnerGames.game.YooNinja_Lite.

[19] ——, "Skater Boy," https://play.google.com/store/apps/details?id=com.game.SkaterBoy.

[20] ——, "Ceramic Destroyer," https://play.google.com/store/apps/details?id=com.game.CeramicDestroyer.

[21] Google Android Market, https://play.google.com/store/apps/.

[22] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy consumption in mobile phones: a measurement study and implications for network applications," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*. ACM, 2009, pp. 280–293.