Mitigating the True Cost of Advertisement- Supported "Free" Mobile Applications

Azeem J. Khan, V. Subbaraju & Archan Misra
Singapore Management University
{azeemkhan,vigneshwaran, archanm}@smu.edu.sg

Srinivasan Seshan
Carnegie Mellon University
srini@cs.cmu.edu

ABSTRACT

The dominant, "ad-supported free application" model for consumer-oriented mobile computing is seemingly imperiled by the growing global adoption of metered data pricing plans by mobile operators. In this paper, we explore the opportunities for addressing this emerging conflict by enabling more intelligent ad delivery to such mobile devices. One especially promising path is leveraging the increasing availability of heterogeneous wireless access technologies (e.g., WiFi, femtocells) that offer less restrictive and more energy-efficient transport substrates for such data traffic. To understand the possibilities that exist, we first profile the advertisement traffic characteristics for some of the most popular advertisement-supported consumer applications, and then analyze the key features of mobile advertisement delivery. We then outline the principles of CAMEO, a middleware that uses predictive profiling of a user's {device, network and usage} context to anticipate the advertisements that need to be served, and then modulates their delivery mechanism to enable effective mobile advertising, but at considerably lower costs .

1. INTRODUCTION

Application marketplaces (especially Apples's App Store^{TM}and Google's Android Market^{TM}), have dramatically transformed the face of consumer computing on smartphones & tablets in the past two years. The explosive growth of such an operator-disintermediated application delivery model has resulted in two significant consequences:

- Consumers have grown accustomed to the phenomenon of free & *niche* mobile applications (especially in the games and infotainment categories), which principally employ an "advertisement-supported" business model. Table 1 reveals that ~ 80% of downloaded applications from the Apple App Store are "free". It is also estimated that the free version of Angry Birds (a popular game) earns about $1 million a month from advertisements, with the overall games segment earning almost $87 million per month from advertisements [4].
- The volume of consumer data traffic on wireless telecom networks continues to grow exponentially. The resulting "tsunami" of consumer data traffic on wireless telecom networks has led to the gradual erosion of unlimited data plans, with several major telecom providers (e.g., Verizon & AT&T (USA), O2 (UK), Docomo (Japan)) having recently introduced metered pricing plans for their advanced 3G & LTE networks.

Table 1: Popularity of free applications as reported at [4]

Total free apps downloaded	81%
Free apps in top grossing list	51%
Free apps in iTunes Store	63%
Free apps in Android Market	37%
Free apps in Blackberry World	82%

This proliferation of advertisement-supported mobile applications has some interesting implications in this emerging era of metered wireless broadband usage. Previous work and reports have demonstrated that such applications can raise significant concerns about data privacy (e.g., [6] reported that two-thirds of selected applications were observed to transmit potentially sensitive personal data to backend servers) and network performance (e.g., [2] reported that advertising-driven network activity in Angry Birds resulted in a 352% increase in signaling traffic, while another incident [3] resulted in 90% call block on a 3G network).

In this paper, we focus on the network overheads associated with the delivery of such mobile advertisements. In metered environments, if the volume of advertisement-related traffic consumes an appreciable fraction of the total data usage, then the applications are not really "free". There are, in fact, applications [1] available in the Android Market that aim to thwart the retrieval of quota-consuming advertisements–e.g., by redirecting application DNS requests, for well-known advertisement provider domains, to the *localhost* interface (127.0.0.1).

We believe that this current "advertisement-supported mobile application model" is unsustainable due to the tensions between the user preferences for "free content", the application provider's desire for revenue, and the network provider's need to manage load on their networks. To address this challenge, we introduce a mobile device-centric middleware, called

[0] This research was carried out at the Living Analytics Research Centre (LARC), sponsored by Singapore National Research Foundation and Interactive & Digital Media Programme Office, Media Development Authority.

HotMobile February 27–28, 2012, San Diego
Copyright 2012 ACM 978-1-4503-1207-3/ . ...$10.00.

CAMEO[1]. The goal of CAMEO is to *preserve the delivery and display of mobile advertisements, while dramatically lowering the advertisement related consumption of precious metered wireless bandwidth*.

CAMEO hinges on the observation that inexpensive or unmetered bandwidth is often available opportunistically to mobile devices. Today, this network bandwidth may come from the WiFi networks that are commonly available at home or at work. Future mobile users may also avail of (inexpensive/nonmetered) 3G/4G femtocells that reduce demand on the cellular network. CAMEO operates by prefetching advertisements during periods with low-cost connectivity and using cached content to avoid transfers over more costly infrastructure. (Interestly enough, the smart prefetching capability of CAMEO is also useful for supporting mobile advertising during network-disconnected application usage, such as when on board airplanes.) As we shall show (in Section 3), the requirements of mobile advertising makes CAMEO significantly distinct from prior work on prefetching other content, such as Web pages (e.g. [11]) or files (e.g. Coda [5]):

1. The advertisement served is context-dependent; hence, pre-fetching requires prediction of not just application request patterns but also broader user, device and even network context. More specifically, prior studies [8] have shown that desktop-based Web searches may require prefetching $\sim 20MB$ or less of advertisement data per day. Not only are such volumes too large for cellular network-based usage, they also ignore the additional dynamics of context attributes that may be relevant for *mobile* users.
2. The economic models of advertising create unique requirements (e.g. for verification, micro-payments and real-time auctions) that existing designs do not support.

While addressing some of these requirements is possible in a backward-compatible fashion, CAMEO's full benefit hinges on the creation of a new ecosystem where *mobile operators, mobile advertisers and application developers* collaborate to create a sustainable advertisement delivery model. As we show in Section 4.2, this will require significant enhancements to both network operator interfaces and smartphone traffic management capabilities.

The rest of the paper is organized as follows. Section 2 empirically studies the volume of advertising traffic in popular mobile applications, thus motivating the need for better advertising traffic management. Section 3 describes key features of mobile advertisement delivery, establishing the functional requirements of CAMEO. Section 4 then describes both incremental and clean-slate architectural designs of the CAMEO middleware. Finally, Section 5 briefly highlights relevant related work, while Section 6 concludes the paper.

2. CHARACTERIZATION OF MOBILE ADVERTISING TRAFFIC

To understand the potential impact of CAMEO, we must first establish the relative volume of advertisement-driven traffic associated with representative different mobile data applications. Accordingly, in this section, we report on the empirical results on advertisement-related traffic characteristics observed during in-lab testing of a selected set of popular advertisement-supported free Android applications. We selected applications from different categories and different regions (US and Singapore) to observe the advertisement-related traffic patterns across a heterogeneous mix of applications & geographies. As the applications were selected separately based on the US and Singapore Android market rankings, it is possible for two distinct applications to share the same rank (in different geographies).

The experiments were conducted by executing each application separately on a HTC Google Nexus One phone (with Android version 2.3.3). The applications were actively used for 15 minutes (30 minutes in the case of the applications *Words with Friends* and *Hanging with Friends*). Most mobile applications in our analysis exhibited recurring behavior w.r.t. the displayed advertisements every few minutes. The time period of 15 minutes was chosen as a suitably long interval to capture the average usage behavior. *Words with Friends* and *Hanging with Friends* interacted with another user across the Internet and hence needed more time to complete the games. Table 2 lists the observed average advertisement-related and application-related network data traffic rate (over the 15 or 30 minute period), for each individual application. More specifically, the advertisement-related traffic consists of both the client-to-server traffic (including, among others, DNS lookups and HTTP Posts) that requests one or more advertisements, and the server-to-client traffic that includes the content of the advertisements. The traffic traces were obtained by 'rooting' the phone; the *Shark for Root* application from the Android Market (similar to Wireshark/tcpdump) was used to capture the network traffic. We manually inspected the collected traces to identify the advertisement related traffic by cross referencing domain names and IP addresses with those of known advertisement networks such as AdMob, inMobi and others.

Note that some of the most highly downloaded applications (e.g., Google Maps, Facebook, Youtube) do not display any advertisements, and are omitted from our empirical studies. However, even these applications may display some amount of advertising content in the future, as these services do display advertisements on their web interfaces. Note also that our studied applications differed appreciably in the volume of 'application-related network traffic'. For example, some applications (e.g., Angry Birds, Fruit Slice, Talking Tom Cat) have all the application content locally stored on the phone, whereas others (e.g., Dictionary.com, Words with Friends) retrieve almost all content from remote servers. During our collection and analysis of the traffic traces, we also observed that many of the studied applications (both free and paid) collect and transmit a variety of potentially sensitive usage and in-Application behavioral data (to an application-specific server, as opposed to advertisement broker domain) for backend analytics purposes. Such advertisement-unrelated traffic is excluded from our computations of the advertising traffic overhead.

From Table 2, we observe that there is indeed a non-negligible amount of network traffic generated by almost every advertisement supported application. The amount of network traffic is closely related to the type and design of the application. For example, the *Dictionary.com* application has a relatively high advertisement related network traffic rate of 3.69Kbps because it displays a new advertisement every time a new word is researched. On the other hand, in the application *Words with friends*, the advertisements are displayed only on the main screen or just after the user has made a move, and not when the user is actively planning a move. Thus, its advertisement-

[1] Context-Driven Advertisement Modulator and Optimizer

Table 2: Mean advertisement and application related network traffic data rate for some of the most downloaded free applications in different categories and regions on the Android Market. (N.A. means not available.)

Rank	Application Name	Advert. Traffic (Kbps)	Appln. Traffic (Kbps)
6	Angry Birds	4.15	0.15
10	Angry Birds Rio	4.03	0.14
10	Hanging with Friends	1.76	5.82
11	Talking Tom Cat 2	1.71	0.02
12	Bunny Shooter	2.64	N.A.
14	Fruit Slice	3.63	0.31
14	Words with Friends	0.89	2.44
15	Angry Birds Seasons	3.59	0.17
16	Drag Racing	0.52	1.32
19	Dictionary.com	3.69	9.74
20	Fruit Ninja	5.61	N.A.
21	Tiny Flashlight + LED	0.41	N.A.
24	The Weather Channel	2.07	N.A.

related network traffic rate is relatively low at 0.89Kbps. Although we do not currently have information on the actual application specific usage pattern of different consumers (i.e., how much time a consumer spends on a specific application within a single billing cycle), it should be clear that advertisement related traffic volumes can be quite high. For example, if the Fruit Ninja game is played for 30 minutes a day and the user has a 2GB/$30 plan (a standard offering currently from Verizon), its advertisement-related activity will generate $\sim$ 40MB of network traffic and will incur a cost of $\sim$ 56 cents per month– a significant cost relative to the one-time purchase price of 99 cents for the same application without advertisements.

We believe that recent trends towards much larger and richly interactive advertisements, that display "within" the application and which are being promoted by advertising networks, will keep these concerns relevant, even as higher-capacity networks are deployed in the future. In our initial experiments, we find such newer advertisements, based on the HTML5 standard, to be typically 5-20 times larger in size compared to the advertisements in applications listed in Table 2. As next-generation smartphones, or more importantly, tablets with significantly larger displays, gain larger market share, the richness and resolution of the multimedia advertisement content can only be expected to increase, effectively consuming a significantly larger fraction of the monthly data quota.

The true cost of a free application is thus not zero and can become fairly substantial when the cumulative impact of multiple applications (especially those that execute concurrently in the background) is considered. Middleware that can intelligently provision the download of non-essential traffic, such as advertisements, without sacrificing the underlying application logic, can, therefore, add immense value to the user and the network operator.

3. KEY CHARACTERISTICS OF MOBILE ADVERTISING

Advertisements displayed on most Internet-enabled devices are usually selected based on *a)* the user's profile, which is usually created over time (e.g., the user's interest in a specific sport) and *b)* current, or recent, environmental context (e.g., the user's current location). Thus, almost all online advertisements are selected for display based on some contextual information that is presented to the advertisement delivery network when that server is contacted. In most browser-based transactions on PCs, this is achieved with the help of cookies that are stored on the PC. Mobile devices, however, by their very nature, present a different set of opportunites and challenges for advertisement delivery.

Table 3: List of contexts sent by applications to different advertisement networks for selection of appropriate advertisements.

Context	Parameters	Presence
Location	latitude and longitude, country, localization, IP address	Always
Device	OS, OS version, model, make, firmware, current time, screen density, screen size, storate size, hardware id, subscriber id, user agent string	Always
Network	telecom carrier, radio, speed, IP address	Always
User	age, gender, subscriber id, language, user-id	Mostly
Others	application id, application version, banner size, format	Mostly

3.1 Use of Context for Targeted Advertising

For all our advertisement deliveries, the mobile first contacts an advertisement network and posts information about the mobile (i.e., the user context) which is then used by the advertisement network to dynamically select the most appropriate advertisement. While location has been demonstrated to affect the displayed ads for Web-based applications [7], the range of context used for heterogeneous, mobile devices is potentially much larger.To better understand the nature of the display mechanism of the advertisements, we analyzed the network traffic trace of the different applications to identify the different *contexts* that are communicated to the advertisement serving provider, presumably for the purposes of targeted advertising. Table 3 categorizes and summarizes the different types of context (passed as plaintext parameters in the HTTP request headers), and the most common granularity of such context, that we observed during our studies.

Our experiments reveal the astonishingly detailed and varied amount of personalized context information that was being routinely delivered to the advertisement serving network. In practice, all of the context information is not always available. For instance, the age, gender and country information fields were always left blank in our observations and therefore were of little use. Similarly, the 'radio in operation' parameter did not appear to play an important role in either the selection or the display size of the advertisements.

Our preliminary studies indicated that the served advertisements were selected on the basis of the following contexts:

1. the network the mobile phone was connected to (evidenced by advertisements suggesting switching to a different carrier).

2. the location, at city-level granularity (evidenced by advertisements for the Mooncake festival in Singapore).
3. the application in use (evidenced by solicitations for other products from the application developer).
4. the model of the phone (evidenced by advertisements for newer phones).

Going forward, we plan to conduct a systematic, and exhaustive, study of the relationship between the served advertisements and the various contextual attributes (by spoofing the context attributes in requests issued to the advertisement network), so as to understand the spatio-temporal nature of mobile advertising churn. However, our initial observations seem to suggest that CAMEO's principle of *asynchronously downloading potential advertisement content* will not alter the basic goals of targeted advertisement delivery, as long as CAMEO can accurately predict the relevant context.

3.2 Mechanics of Advertisement delivery

To formulate our proposed middleware architecture, it is also necessary to understand the actual mechanisms employed for advertisement delivery to mobile devices. We therefore studied the nature of the network transactions employed by the mobile device for retrieving and displaying advertisements. We observed that all advertisements being displayed in the application utilized HTTP for fetching the advertisements. From our analysis, we observed two prominent techniques of advertisement placement in the application:

1. In the first technique, once an image was fetched, it was displayed with a small piece of XHTML code.
2. In the second technique, JavaScript and JSON were used to fetch and display the image. In some cases, JavaScript was used to stitch together several very small images and display it as one bigger image.

While advertisements displayed as part of a game are becoming increasingly popular in gaming consoles, we did not observe this artifact in the set of mobile applications that we studied (Table 2). All the advertisements were overlaid on the applications, mostly at the top or at the bottom of the displays. We also observed a couple of common characteristics shared by the vast majority of our applications.

- Most applications used more than one advertisement network for serving advertisements. Thus a diverse group of advertisements (and advertisers) were displayed to the application users.
- A typical retrieval and display of an advertisement involves three to four separate TCP transactions–first to contact the advertisement network, second to fetch the XHTML or javascript code and third to fetch the actual image or advertisement content. Indeed, for the example of the Angry Birds application, we observed that, for a typical 5-8KB image being displayed, the three transactions combined amounted to a total of approximately 30KB of data transfer on average, implying that the actual content was often only 1/4th of the total network overhead of advertisement delivery. We expect this to change with HTML 5 based advertisements that were as large as 572KB in our tests.

The design of the CAMEO middleware must, thus, be capable of *a)* supporting existing HTTP-based advertisement requests, *b)* interfacing with a multiplicity of mobile advertising delivery networks, and *c)* being extended to accomodate the emergence of new advertisement providers.

3.3 Advertisement Payment Models and Verifiability

Mobile advertising networks use three models to measure the effectiveness of advertisement delivery and consequently bill the advertiser. In the first model, the cost per thousand impressions (CPM) is used to measure the price of reaching a large audience. In the second and most widely adopted model, advertisement networks employ CPC or cost per click (and, more recently, for phones, a cost per call variant as well) where the advertiser pays the mobile advertisement network only when the user actually clicks on the URL associated with the advertisement. In the third way, the advertiser sets a price on the "cost per action" (CPA), where action could even mean an actual sale. In CPA, the advertiser only pays if some meaningful interaction occurs. In all the models, the mobile application developer is paid a portion of this price, in return for displaying the advertisement within her application.

In each model, the advertising delivery network must be sure that the user is indeed providing the right context and is then viewing the right advertisement that has been served. CAMEO's design must continue to preserve the integrity of this advertisement delivery and display process, even when advertising networks employ real-time auctioning and other dynamic mechanisms to select advertisements. In particular, the intermediary CAMEO software must provide mechanisms that allow both the client application (on the mobile) to validate the advertisement, and the advertisement delivery server to verify CAMEO's advertisement serving logic.

4. INITIAL DESIGN OF THE CAMEO MIDDLEWARE

As we have shown in the previous sections, while advertisement-supported mobile applications are widely used, design choices in existing cellular systems and in current software interact poorly. CAMEO seeks to lower this burden of advertisement retrieval over metered wireless interfaces by *a)* predicting the likely future context of the user/device and thereby inferring the advertisement content likely to be served, then *b)* intelligently utilizing available, lower-cost, potentially un-metered, access networks to pre-fetch and store such content, and *c)* serving appropriate advertisements from a local cache. CAMEO's high-level architecture, depicted in Figure 1, indicates that CAMEO acts as an intermediary between the mobile-resident application and the advertisement delivery network. We now describe the design philosophy underpinning both an incremental, backwards-compatible version of CAMEO (that can be deployed without any major changes to existing functionality) and a forward-looking, *clean-slate* approach that holds greater promise.

4.1 Solutions Compatible with Existing Deployments

We first consider designs that reduce the cost of advertisement delivery, while maintaining compatibility with existing systems. Any proposed design must solve two problems. First, to prevent advertisement transfer when clients are using costly (or metered) connectivity, the system must employ some mechanism to intercept requests. Second, the system must generate an appropriate advertisement for the user and service the request locally. Below, we describe a simple design that leverages middleware running on the mobile client to achieve these two tasks.

Interception. Based on our observation (in Section 3.2) that advertisement retrieval almost always involves a DNS query followed by multiple HTTP requests (for retrieving content), one possible implementation of CAMEO involves running a transparent Web proxy on the client. This proxy would identify advertisement requests, typically by inspecting the domain or host name of the target. Alternately, we could also intercept the DNS query; for example, by modifying /etc/hosts to redirect queries for common advertisement servers to the localhost. We believe that intercepting the HTTP request provides finer-grained control over advertisement request identification and interception.

Advertisement Generation. Once the HTTP request for an advertisement is intercepted, the proxy must provide an advertisement to the requesting application. To achieve this, the proxy must have cached appropriate advertisements in advance. This requires addressing two issues. First, as noted in Section 3.1,the proxy middleware must be able to predict the range of contexts that the user will encounter in the future. Second, the middleware must be able to optimize the cost of advertisement data transfers, by leveraging upon the different types of network connectivity (WiFi, femtocell etc.) that a typical mobile device can associate with.

The above tasks are accomplished by different components of CAMEO (Figure 1): the Advertisement Cache Manager component intercepts advertisement requests and serves locally cached content, the Intelligent Ad-Prefetcher component is responsible for asynchronous pre-fetch of likely relevant advertisements and the Context Predictor is responsible for predicting the future user context. Unfortunately, this incremental approach limits the possible efficiency of the overall system. For example, in this approach, the network cannot explicitly distinguish advertisement traffic as a special traffic class, and, thus, fails to exploit improved opportunities for asynchronous advertisement delivery. Also, there is no easy mechanism for either the application or the advertisement provider to verify that CAMEO indeed delivered the correct (or any) advertisement.

4.2 A Clean-Slate Design

A more appropriate design would include functionality for accountability and interactions between the different entities—the advertising network, the middleware and the mobile application. CAMEO will additionally benefit from a more significant change in the way that advertisement networks and mobile applications interact with the wireless network provider. Below, we describe some of the key changes needed to both applications and network infrastructure, and the associated research challenges.

Application Modifications. Application modifications can address two weaknesses in the backward-compatible approach. First, the backward compatible approach employs a relatively awkward mechanism to intercept data transfers and relies on heuristics (e.g., an explicit list of host names) to identify advertisement requests. A modified application can explicitly direct advertisement requests to a local advertisement proxy. Second, the application can also perform small exchanges with the advertising service to validate that the advertisement it received is appropriate for the user's current context. In addition, when the advertising network uses real-time auction-based mechanisms to dynamically prioritize advertising content, the CAMEO middleware can use a low-overhead interaction to retrieve a dynamic reference to an advertisement that is locally cached on the device. In particular, in this approach, CAMEO uses the network minimally; identify the advertisement to be served, but avoid the traffic overhead of actually downloading the content over the metered network. This is a simple case wherein the *Local Advertisement Selector* is employed.

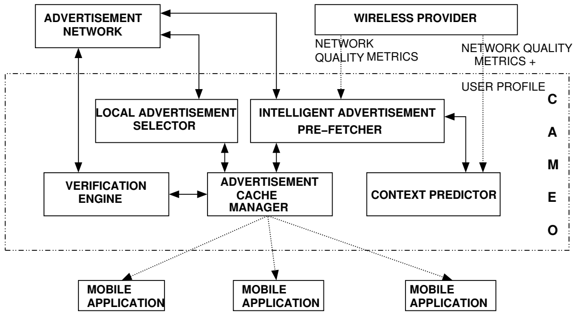

Figure 1: A preliminary candidate design for CAMEO. Some components of the CAMEO architecture may reside either in the mobile device or in the network.

On the other hand, a different problem arises if advertisement selection is performed randomly by the CAMEO client on the device, or when the CAMEO client serves advertisements while the device is disconnected from the network. More specifically, in this case, unlike conventional content proxies, the issue isn't whether the content is authentic or not; rather, the challenging research question centers on whether the application and the advertisement network can verify that CAMEO is serving the *most appropriate* cached advertisement, given a specific context. One possible approach is for CAMEO's Cache Manager to deliver an advertisement and for the application to subsequently transmit a SHA1 hash of the advertisement, and its current context, to the advertising service, for verification that an appropriate advertisement was indeed delivered. Such an approach involves verification of each individual advertisement delivered. Alternatively, more statistical models for *computation verification* may be employed (by the Verification Engine component in Figure 1) to enable the advertisement server to verify that CAMEO's advertisement delivery, at an aggregate level, is indistinguishable from what would have been delivered, synchronously, directly to the application.

Infrastructure Modifications. There are several possible enhancements to the network infrastructure components that would permit the creation of new eco-systems that enable clients to use much more network bandwidth for advertising content, without increasing the cost to the client.

For example, consider the true motivation behind metered pricing plans. Cellular providers want to manage the load on their network; however, they gain no benefit when the network is idle. Their goal is to ensure the network is *used but not overloaded*. Better advertisement delivery and better network load management might be possible if the network provided different classes of service, or more explicitly dynamically-varying cost indicators for data transport. In particular, as advertisement pre-fetching is not a time critical activity, CAMEO can leverage upon some form of low-priority or background transfer service provided by the wireless operator, potentially for free or at a greatly reduced charge, making advertising relatively inexpensive. In fact, CAMEO's use of pre-fetching is motivated precisely by the observation that intelligent pre-fetching, while consuming perhaps more bytes than an on-demand retrieval model, may lead to lower usage 'cost' as those bytes are now retrieved in a more network-friendly fashion. Similarly, additional support from the wireless provider, for predicting the user movement and other network context, can significantly enhance the accuracy of Cameo's Context Predictor.

Enhancements to accounting and billing frameworks could also enable more effective advertising delivery. For example, advertising services may choose to subsidize delivery of advertisements over the cellular network, but only under specific user contexts (e.g., only pay for advertisements when the user is in a particular store). Alternately, some advertisers may provide infrastructure solely for the purpose of advertisement delivery. For example, a shopping mall may provide WiFi or femtocell connectivity within the mall to allow shoppers to use applications that advertise mall products, while restricting the network's use for other generic applications. Such advanced forms of mobile advertising will require careful accounting and assignment of accounting revenue to different sources, the ability of wireless networks to support application-level (instead of device-level) authentication and for smartphone operating systems to direct traffic from different applications over different network links. While these are just a few of the possibilities, they illustrate how a a fundamentally new model of cooperative signaling between mobile advertisement providers, wireless network operators and CAMEO components can enable significantly more sophisticated forms of network-friendly, mobile advertising.

5. RELATED WORK

We now describe some of the prior research relating to one or more of the key capabilities for CAMEO. CAMEO's principal goal is to perform intelligent pre-fetch of context-sensitive advertisements; thus it utilizes principles of pre-fetching from a long body of prior work on disconnected file systems, such as Coda [5]. Unlike file-system catching, CAMEO employs pre-fetching not to minimize latency or operate under network disconnection, but to avoid use of costly network interfaces. Accordingly, CAMEO's opportunistic use of heterogenous network interfaces may also be viewed as similar to prior work on data transfer optimization by using multiple network connections (e.g., [10]). However, CAMEO is not concerned about improving the speed or reliability of data transfers, but rather on intelligently scheduling delay-tolerant data transfers across less capacity-controlled access networks. Finally, CAMEO requires innovations in distributed authentication, to essentially permit the network provider to verify that CAMEO is correctly serving advertisements to multiple mobile applications. The Privad middleware [8] proposes a privacy-preserving middleware for online advertising. In Privad, the intermediary acts principally as a pass-through anonymizer, with actual exchange of context and verification of click-through occuring end-to-end for every advertisement. While CAMEO can be combined with the privacy-preserving features of Privad, our authentication problem is different: the mobile advertising network must verify the, potentially-offline, operation of the CAMEO component that it does not directly control. To achieve this, we expect that CAMEO will borrow techniques from prior work on authentication of third-party metering (e.g., [9]).

6. CONCLUSIONS

Advertisement-related network traffic for free applications on mobile devices will continue to increase, adversely impacting users on metered data plans and stressing the capacity of wireless networks. We propose a new advertisement delivery framework called *CAMEO*, to cooperatively mitigate the stress from such non-essential traffic. CAMEO seeks to predict user context and thereby proactively identify relevant advertising content, and then opportunistically use un-metered or inexpensive wireless networks to predictively cache advertisement content on the mobile device. CAMEO can achieve far-reaching impact and enable new forms of mobile advertising services by leveraging on innovations (such as targeted pre-fetching, 3^{rd} party verifiability and explicit network differentiation of advertisement traffic).

References

[1] Adfree Android. http://market.android.com/details?id=com.bigtincan.android.adfree, October 2011.

[2] Angry Birds Ruffle Signaling Feathers. http://www.lightreading.com/document.asp?doc_id=208870, June 2011.

[3] Operators Urge Action Against Chatty Apps. http://www.lightreading.com/document.asp?doc_id=208775, June 2011.

[4] What is the percentage of free mobile app downloads? http://fonegigsblog.com/2011/08/13/what-is-the-percentage-of-mobile-app-downloads-that August 2011.

[5] P. Braam. The Coda Distributed File System. http://www.linuxjournal.com/article/2390, July 1998.

[6] W. Enck, P. Gilbert, and B. C. et. al. TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones, October 2010.

[7] S. Guha, B. Cheng, and P. Francis. Challenges in measuring online advertising systems. In *Proceedings of the Internet Measurement Conference 2010*, IMC '10, Nov. 2010.

[8] S. Guha, B. Cheng, and P. Francis. Privad: Practical privacy in online advertising. In *Proceedings of the 8th Symposium on Networked Systems Design and Implementation*, NSDI '11, Mar. 2011.

[9] R. Johnson and J. Staddon. Deflation-secure web metering. *International Journal of Information and Computer Security*, 1:39–63, 2007.

[10] P. Rodriguez, R. Chakravorty, and J. C. et. al. MAR: a commuter router infrastructure for the mobile internet. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, MobiSys '04, pages 217–230, 2004.

[11] A. Venkataramani, P. Yalagandula, R. Kokku, S. Sharif, and M. Dahlin. Potential costs and benefits of long-term prefetching for content-distribution. In *Proceedings of the Web Caching and Content Distribution Workshop*, Jun. 2001.