CAMEO: A Middleware for Mobile Advertisement Delivery

Azeem J. Khan*
Oriental Institute of Management
University of Mumbai
azeem@oes.ac.in

Kasthuri Jayarajah
School of Information Systems
Singapore Management University
kasthurij@smu.edu.sg

Dongsu Han
Department of Electrical Engineering
KAIST
dongsu.han@gmail.com

Archan Misra
School of Information Systems
Singapore Management University
archanm@smu.edu.sg

Rajesh Balan
School of Information Systems
Singapore Management University
rajesh@smu.edu.sg

Srinivasan Seshan
Department of Computer Science
Carnegie Mellon University
srini@cs.cmu.edu

Abstract

Advertisements are the de-facto currency of the Internet with many popular applications (e.g. Angry Birds) and online services (e.g., YouTube) relying on advertisement generated revenue. However, the current economic models and mechanisms for mobile advertising are fundamentally not sustainable and far from ideal. In particular, as we show, applications which use mobile advertising are capable of using significant amounts of a mobile users' critical resources without being controlled or held accountable. This paper seeks to redress this situation by enabling advertisement supported applications to become significantly more "user-friendly". To this end, we present the design and implementation of CAMEO, a new framework for mobile advertising that 1) employs intelligent and proactive retrieval of advertisements, using context prediction, to significantly reduce the bandwidth and energy overheads of advertising, and 2) provides a negotiation protocol and framework that empowers applications to subsidize their data traffic costs by "bartering" their advertisement rights for access bandwidth from mobile ISPs. Our evaluation, that uses real mobile advertising data collected from around the globe, demonstrates that CAMEO effectively reduces the resource consumption caused by mobile advertising.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless Communication

General Terms

Design, Experimentation, Performance

Keywords

Wireless; Mobile; Advertising

*Research performed by author as a Research Fellow at Singapore Management University.

1. INTRODUCTION

As per Gartner forecasts, over 85% of the applications to be downloaded from the mobile application marketplaces in 2012 will be free and principally supported by advertising revenue, even though the same applications are often available in both advertisement-supported and paid-for, advertisement-free versions. This points to the fact that users are happy to "pay" for applications with their *attention*, but not with their pocketbooks. The end result is that "Advertisements" have become the de-facto mechanism that mobile applications use to generate revenues. There are clear indicators that this "payment" model is here to stay, with mobile advertising revenues growing by 149% in 2011 [1].

Unfortunately, the current advertisement delivery framework has certain limitations that are likely to make consumers increasingly hostile to such advertising traffic. From a conceptual viewpoint, an ideal advertising framework is one that *only monetizes user* ***attention****, without incurring the user any additional monetary or resource costs (such as network bandwidth or device energy).* However, as mobile ISPs universally move towards metered wireless data plans, consumers are progressively growing wary of the hidden costs of such advertising traffic, both in terms of *bandwidth consumption* [12, 28], and *energy overheads* [19]. To circumvent such costs, savvy consumers often end up installing ad-blocking software or turning off their cellular data

[0]This research is supported by the National Research Foundation Singapore under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.
This research is supported by the National Research Foundation Singapore under its IDM Futures Funding Initiative, and administered by the Interactive & Digital Media Program Office, Media Development Authority.

connections when executing an application with a heavy advertising footprint.

In this paper, we present a new mobile advertisement delivery framework,called CAMEO[1]. CAMEO's main goal is to significantly lower the overheads, as perceived by the consumer, of delivering advertisements to mobile applications, thereby making the business model of advertisement-supported free applications much more "consumer-friendly". To achieve this objective, the CAMEO middleware component on a mobile device provides two new features:

1. *Reducing Costs*: CAMEO allows advertisement networks to proactively place a corpus of context-driven advertisements on a mobile device, by exploiting time windows when the consumer is connected to a cheap or free network (e.g., Wi-Fi@home), and then serve advertisements locally using this on-device corpus. Such proactive retrieval benefits consumers by significantly reducing the bandwidth and energy overheads associated with advertising traffic.

2. *Bartering of Advertisements for Access Connectivity*: CAMEO enables individual mobile applications to use "advertising rights" as a form of implicit & universal currency to negotiate free or subsidized data bandwidth from different mobile ISPs. While this doesn't reduce the energy or bandwidth demands for an application, it can reduce the users' costs associated with connectivity (e.g., when on metered data plans or attaching to pay-per-use Wi-Fi hotspots). Note that other studies have proposed the explicit monetization of users' information [22], while applications such as Facebook Zero[2] have negotiated directly with large ISPs to make their application data consumption free to end consumers [15]. Unlike these past designs and alternatives where the user must explicitly allocate bandwidth across multiple applications, CAMEO minimizes user distraction by avoiding direct user involvement, and lets an individual application dynamically tailor the use of advertising slots (as the medium of exchange between users' attention and bandwidth).

Note that the above two components are independent but symbiotic. By making costs for advertising content more explicit to application developers through the negotiation component, our model provides developers and their advertisement network partners stronger incentives to perform advertisement prefetching and display more efficiently.

To explore the challenges in supporting more efficient ad delivery and support ad negotiation, we have designed and built a prototype user-space implementation of CAMEO for the Android OS. Through our implementation and associated empirical studies, we make the following key contributions:

1. **Use context prediction to prefetch advertisements and reduce the network 'cost' of advertisement delivery:** We propose, in Section 4.1, a bulk ad prefetching model that first predicts future user context by mining past context history, and then leverages this context prediction to retrieve a comprehensive corpus of relevant advertisements. This on-device corpus can then be used by CAMEO to serve advertisements locally. Through extensive empirical studies in multiple geographies (North and South America, Europe, and Asia), we establish, in Sections 5.2 and Section 6.1 respectively, the key contexts that drive the selection of context-aware mobile advertisements and that an appropriately selected corpus can indeed be prefetched periodically with negligible cost.

2. **Allow on-device serving of advertisements while preserving different advertisement selection and pricing models:** In Section 4.2, we show how CAMEO allows different advertisement networks to apply their advertisement-selection logic over prefetched advertisements stored locally on the mobile device. Subsequently, in Section 4.4, we show how CAMEO also supports different popular models of real-time pricing. CAMEO achieves this flexibility by effectively decoupling the tasks of advertisement *selection* and *delivery*.

3. **Demonstrate significant savings in bandwidth and energy overheads:** Using empirical data of consumer application behavior and micro-benchmarks of CAMEO's performance on mobile devices, we show, in Section 7.2, that CAMEO's bulk prefetching provides substantial savings in both advertisement-related traffic and energy consumption. Most importantly, such bandwidth savings are realized though CAMEO performs bulk prefetching of advertisements.

4. **Flexible model for negotiation between individual applications and mobile ISP:** In Section 4.3, we present CAMEO's proposed use of a simple, but generic, slot-based advertising model (borrowed from television advertising) to negotiate preferences and constraints between an application and an ISP. This model also allows applications using CAMEO-compatible applications to effectively intersperse advertising content from conventional advertisement networks and the mobile ISP.

2. BACKGROUND AND ISSUES

We first review the current interaction model for mobile advertising and highlight a few of the drawbacks that motivated our proposed research on CAMEO.

2.1 The Current Model

Main players: The mobile advertising eco-system consists of 4 players: applications, advertisement networks, mobile ISPs, and consumers. At present, advertisements are retrieved "over the top" (directly by an application from the advertising content provider, without any direct involvement of the mobile ISP): Ad-supported *applications*, such as AccuWeather and Angry Birds, use advertisement libraries provided by *advertisement networks* (**AN**) to fetch and display advertisements. An individual application specifies the layout of an advertisement and supplies the relevant user contexts, such as gender, age group, and keywords, using the library. By interacting with the AN's servers, the advertisement library then retrieves and displays an ad that best matches the context and the AN's business interest. ANs thus provide an online marketplace for brokering matches

[1]**C**ontext-Aware **A**dvertising **Me**diator and **O**ptimizer
[2]Facebook Zero is a text-only version of Facebook for the developing world.

between advertisers and the application developers. *Mobile ISPs* presently provide the underlying connectivity substrate that allows mobile applications to access online services (including ANs). ISPs worldwide have increasingly migrated to metered data plans, where the cost of connectivity for a *consumer* is a function of the amount of data consumed in a billing cycle.

Economic Model: Advertisers pay for advertising to the AN, which in turn shares a fraction of the revenue with the application developer. Currently, the most dominant payment models are 'cost-per-click' (CPC), 'cost-per-impression' (CPM), and 'cost-per-acquisition' (CPA). In CPC and CPA, payments accrue when users click on an advertisement or makes a specific action (e.g., updating on-line shopping carts). In CPM, advertisers are charged whenever an advertisement is *displayed* on the mobile device. As different models have different strengths, we consider all three models in this paper[3]. The AN's selection of advertisements are driven by two criteria: 1) matching of the user context to the context triggers that advertisers may have specified (e.g., at a specific location), and 2) the price they are willing to pay (which is often decided through real-time auctions [8]).

2.2 Problems

CAMEO is predicated on the belief that mobile advertising can become significantly more *consumer-friendly* by tacking the following problems associated with today's advertisement delivery framework.

High Overheads of Advertisement Delivery: Mobile applications presently engage in multiple network transactions to fetch and display a single advertisement, leading to a significant consumption [12] of often-precious data bandwidth. When operating under metered-data plans (e.g., on almost all 4G/LTE networks), users effectively end up 'paying' for the advertising-related traffic generated from their mobile devices. As an example, Khan [12] reported that the retrieval of a 5–8KB image advertisement on the *Angry Birds* application generates almost 30KB of network traffic. This is particularly troublesome for applications that are otherwise 'standalone' (e.g., mobile games), whose network traffic is almost *entirely* advertising related. Likewise, the frequent signaling overhead generated by such intermittent advertisement retrieval also consume a significant amount of energy on the mobile device [28, 19].

Limited Ability to use Ads as a Means to Monetize Connectivity: Presently, mobile ISPs have no role in the economics of advertising and view advertising content merely as 'application data'. Accordingly, the present model fails to capitalize on a possible convergence of interests, where *a)* mobile ISPs (e.g., Wi-Fi hotspots at public venues such as shopping malls and train stations) might be interested to offer free or subsidized connectivity in return for the prerogative to insert their own advertisements, *on existing applications preferred by the consumer*, and *b)* applications might make themselves more appealing by continuing to operate even when the consumer does not have an explicit pre-existing ISP subscription (e.g., at hotspots or while roaming).

[3]Facebook offers all three, and Google's Adwords offers CPC and CPA on their search network and all three on their display network. A market survey shows that, for video advertising, all payment models are equally preferred by the advertisers [2].

3. DESIGN GOALS

CAMEO's mobile advertising-related middleware aims to *a)* reduce the bandwidth and energy overheads of fetching and displaying advertisements from individual ANs, and *b)* promote the flexible use of advertisements as a universal currency for bartering network connectivity. The design goals for CAMEO are as follows:

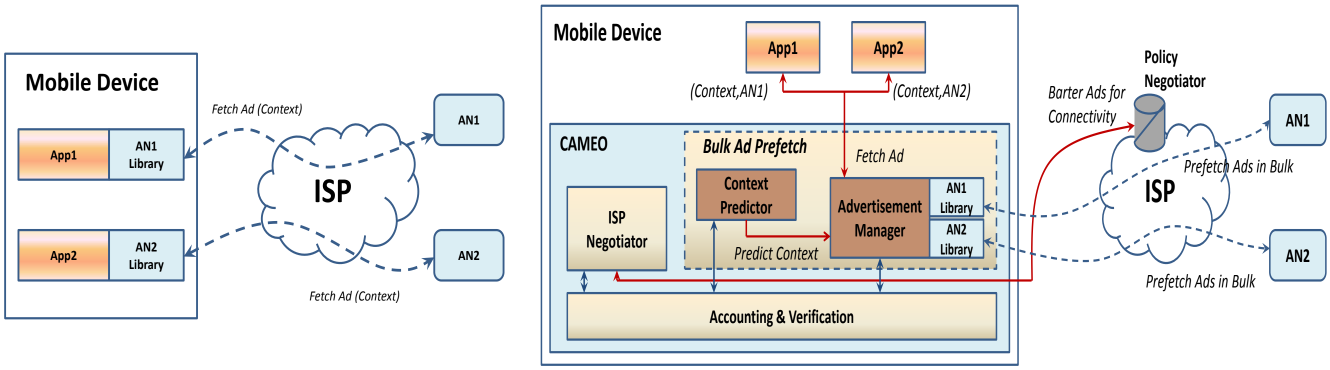
1. **Support lower cost of advertisement delivery without modifying the AN's advertisement selection logic:** Given the high signaling traffic overhead involved in retrieving an individual advertisement, and the energy overheads of activating a radio interface intermittently to interactively fetch such advertisements, CAMEO's design should allow retrieval of advertisements in bulk, especially during periods of connectivity to 'cheap' access networks (e.g., opportunistic association with home or office Wi-Fi networks), to amortize these traffic and energy costs. However, it is essential that such bulk retrieval does not impact the AN's desire to control the selection, in real time, of each individual advertisement displayed.

2. **Minimize user involvement:** CAMEO should require only minimal, or preferably zero, user involvement in the process of optimizing advertisement delivery or negotiating an "advertisements for subsidized connectivity" barter with a mobile ISP. Mobile devices typically run a variety of concurrent applications (usually many background applications and a foreground one), which vary significantly in their network traffic demands, and in the types and frequency of advertisements shown. Accordingly, we believe that users would find it very challenging to determine a suitable apportioning of the (bandwidth, advertisement) budget among competing applications—user-involved adaptation has proven to be a major challenge in prior mobile systems, such as Odyssey [18].

3. **Incentivize developers to make their applications "consumer-friendly":** CAMEO should effectively empower each application developer to *independently* find the best way for the application to 'spend' the virtual currency of advertisements (reflected in the tradeoffs the application makes between the application's bandwidth requirements and its frequency of displaying ads), so as to maximize the application's "consumer friendliness".

4. **Require minimal modifications to applications and ANs:** The design for prefetching bulk advertisements should be *minimally disruptive*, requiring, at best, only minor changes to application code or AN interfaces. Moreover, while enabling the dynamic exchange of advertisements in return for subsidized connectivity, CAMEO should isolate applications from the complexity of mobile ISPs, by having the middleware bear the brunt of responsibility for functions such as ISP selection, negotiation of bartering agreements and traffic & advertising contract enforcement.

To achieve efficient, AN logic-compliant advertisement delivery, we will show (in Section 4.1) how CAMEO selects advertisements locally from a corpus of prefetched advertisements during periods of cheap network connectivity, and



a) Current Ad Retrieval b) Modified Ad Retrieval via CAMEO

Figure 1: The key components of CAMEO

how applications can use this feature of CAMEO via simple API calls. To minimize user involvement while incentivizing each application to maximize its "customer friendliness", we will develop (in Section 6.3) CAMEO's *per-App* negotiation model, where each application dynamically and opportunistically negotiates network access from ISPs that might be willing to offer such subsidized access in return for the privilege of inserting their own advertising content in the application's ad-stream. We prefer such a "per-App" negotiation model over possible alternatives for such preferred connectivity, as this model provides finer *granularity* (applications can tune their own behavior based on individual-specific usage patterns), greater *dynamicity* (the connectivity is negotiated on-the-fly and does not require long-term contracts) and lower *user involvement* (negotiations for access connectivity are transparent to the user).

4. CAMEO ARCHITECTURE

CAMEO sits as a middleware on the mobile device, performing three functions that effectively mediate the interactions between individual applications and the ANs/ISPs:

1. It pre-fetches a corpus of advertisements from multiple ANs without application intervention.
2. It serves advertisements from the locally-stored corpus to the mobile application.
3. It negotiates with the ISP (bandwidth in exchange of advertisements) on behalf of the mobile applications.

Figure 1 illustrates the key functional components of the CAMEO framework, and shows how the new advertisement retrieval and negotiation models differ from current practices. As explained in Section 2.2, and illustrated in Figure 1.a), applications presently treat the ISP as a bit-pipe, and use AN-specific libraries to directly fetch individual advertisements from the corresponding AN. In contrast, Figure 1.b) shows the CAMEO-enhanced interaction model, where the various components of the CAMEO client middleware mediate the interaction between applications and ANs by exposing APIs to each individual application. The advertisement prefetch and local serving functionalities are provided by the combination of the **Context Predictor** and **Advertisement Manager** components. The logically-separate functionality of bartering advertisement privileges in return for access bandwidth is performed by the **ISP Negotiator**. Each of these three components interact with the **Accounting and Verification** module, which assures ANs that the local advertisements are being served "correctly" and enforces the bartering agreements set up with an ISP. We now describe each of the key functions individually.

4.1 Predictive Bulk Ad-Prefetch

CAMEO's design seeks to reduce the high bandwidth and energy overheads associated with the current model of interactive and intermittent retrieval of individual advertisements, by serving the vast majority of advertisements *locally*, i.e., from an intelligently managed local corpus prefetched to the mobile device. However, as we will see in Section 5, the advertisements on mobile devices are highly context-sensitive, i.e., the selection of an advertisement depends on the real-time context of the mobile user/device. Prefetching relevant advertisements, thus, requires predicting the future contexts of the mobile device/user. CAMEO realizes such predictive prefetching by having our Context Predictor monitor the user context and apply machine learning techniques on past context history to predict the future context.[4]. Subsequently, to retrieve relevant advertisements for the future, the Advertisement Manager supplies this predicted context to an AN and asks it to provide an appropriate corpus of advertisements. At the time of ad display, the Advertisement Manager performs a local lookup to select an advertisement matching the current context, and serves this advertisement to the requesting application.

In Section 7, we demonstrate that this prefetching technique is indeed effective in supporting context-dependent advertisements and realizes significant bandwidth and energy savings.

4.2 Advertisement Selection

CAMEO supports three different models for advertisement selection on the mobile device, each of which assumes that the AN has already dispatched a corpus of relevant advertisements that are then stored locally. In the order of decreasing overhead, we present the three models that vary in their level of real-time interaction with the AN:

1. **AN Advertisement Selection (ANAS):** In ANAS, CAMEO contacts the AN and provides the current context. We assume that the AN is already aware of the contents of the corpus that it has previously

[4]In practice, such context profiling and prediction can even be performed on an infrastructure server.

pushed to the mobile device. The AN then applies its custom-selection logic, and delivers CAMEO a unique ID corresponding to the selected advertisement. Note that the advertisement itself is not downloaded again. The ANAS model focuses on eliminating redundant ad transmissions and is particularly suited for mobile applications that provide utility only when connected to the network—e.g., an application that retrieves arrival times of buses in real-time. We observe in Section 7.2 that ANAS can reduce the bandwidth consumption by up to 4.6 times when displaying 100 advertisements.

2. **Local Advertisement Selection (LAS):** In LAS, CAMEO allows each AN to send over a static set of rules along with the relevant ad corpus. Upon request for an ad from an application, CAMEO operates on the rules locally to return an ID corresponding to an ad to be displayed. This model eliminates real-time signaling and transmission associated with advertising. LAS is thus appropriate for standalone applications that only require network connectivity for ad display. Our results in Section 7.2 shows that this method can reduce the energy consumption by an order of magnitude, when displaying 100 advertisements. An unintended but perhaps useful consequence of LAS is the ability to show advertisements on an *offline* device.

3. **Best Effort Advertisement Selection (BEAS):** In this model, the selection logic is performed locally by CAMEO, in a *statistical* fashion. Under the BEAS approach, the AN specifies multiple constraints that are specific to each advertisement. Our current CAMEO implementation allows ANs to specify two constraints: (1) f_i: how frequently an advertisement, ad_i, can be displayed, and (2) Max_i: the maximum number of times an ad can be shown. Given these specifications, CAMEO can apply a randomized selection procedure that adheres to these constraints. For example, if there is no limit on the maximum number of times an ad can be shown and f_i is identical for all advertisements, the CAMEO can simply select the advertisements in round-robin fashion.

4.3 Bartering Ads for Connectivity

CAMEO utilizes a *slot-based* model for negotiating and specifying the agreement by which applications can allow ISPs to insert their advertisements in exchange for access connectivity. This model is inspired by the television industry, where advertisements (and associated revenue) are shown at specific timeslots (e.g., five 30-second slots every eight minutes) and are apportioned between a regional and national broadcaster, the local affiliate TV station and the last-mile access provider (e.g., the local cable company). In CAMEO, we propose a similar design. Most advertisement-supported mobile applications show advertisements at fixed intervals or when a specific event is triggered (e.g., the user reaches the next level of a game). We consider each such display of advertisement as a *slot*. When a mobile application barters for free bandwidth with the ISP, it can choose to fill some of these *slots* with advertisements from the ISP (or more likely the ISP's preferred AN). A detailed design of the protocol for the negotiation between the mobile application and the ISP is described in Section 6.3.

4.4 Accounting and Verification

To ensure that ANs and ISPs can trust this new model of CAMEO-mediated advertisement selection and display, CAMEO must provide certain accounting and verification functionality. While click spams and other fraudulent behavior are a concern for mobile advertising even today [3], CAMEO's unique features of *a)* local selection from the prefetched advertisements and *b)* dynamic interleaving of ISP-provided advertisements, introduce some additional requirements and concerns. Note that our current implementation of CAMEO does *not* provide this support: we believe that eventually CAMEO will become a part of the base OS and a combination of previously-studied software and hardware innovations (e.g., TPMs) will make such accounting a default capability. In this section, we thus describe the accounting functionality that CAMEO requires and possible techniques that we can leverage to provide this support.

CAMEO's model of displaying prefetched advertisements creates two accounting challenges:

- *C1):* If the context prediction is inaccurate, CAMEO does not have the correct prefetched ads to display. As a result, CAMEO must make a decision to contact the AN again to fetch a new ad or instead display a sub-optimal ad.

- *C2):* CAMEO's decentralized advertisement selection makes it harder for advertisers to enforce total or daily global budget constraints, if we use the BEAS or LAS models.

To address challenge C1, CAMEO could periodically report its mispredictions to the AN and allow it to estimate potential loss in revenue. Both ANs and applications could use this information to decide whether to obtain a new corpus of advertisements proactively. To address C2, CAMEO could rely on statistical techniques to estimate the number of ad impressions that are occurring at any time. This estimate could be used be clients to tune the probabilistic selection of ads on any client.

In Section 4.2, we alluded to the possibility of displaying ads when the device is offline. If this was implemented it would created additional challenges related to tracking ads shown. However it would also create a challenge similar to the problem of click and impression fraud in today's systems which is an ongoing research problem [23, 4].

5. EMPIRICAL STUDY

To demonstrate the real-world potential of context prediction based prefetching, we perform an empirical study of mobile advertisements. In addition, to get an in-depth understanding of average user behaviors and connectivity, we analyze the mobile usage patterns of twenty users with an instrumented Android phone that monitored user activity and network connectivity for more than a month.

5.1 Data Collection Procedure

Mobile advertisements study: We fetched advertisements from Google's AdMob network with different contexts using a custom script deployed on 100+ PlanetLab nodes and machines in Singapore and Pittsburgh. We vary all five context variables supported by the AdMob's library. The different context attributes that were varied are as follows: 4 Android OS versions including tablets and smartphones; 3 international and 3 US mobile carriers; gender; age group; 5 different key words: no keyword, music, game, language, business; and 4 popular and non-popular applications (SoundHound (SH), Plume (PL), Hanping Chinese Dictionary (HD) and a custom demo app (DM)). Note that we also vary the location implicitly by geographically distributing the clients.

The advertisements were fetched every minute to abide by Google's recommendation. At this rate, in our fetching process, the same context repeats every 840 minutes. For the PlanetLab nodes, we generated a corpus of more than 1 million advertisements over 14 days. For Singapore and USA locations, we generated a corpus of 60K advertisements over a period of two months. The advertisements fetched were observed to be mainly of non-interactive HTML (with AdMob's standard Javascript), HTML with banner images, and creatives (interactive text or image) rendered using JavaScript.

User study: A custom developed monitoring application was installed onto 20 Android-based smartphones at our Singapore campus. 12 of them belonged to students and the rest to office staff on-campus. The monitoring application tracks and records almost every Android event related to user interaction as well as system information such as battery level. For a subset of users, we also monitored their bandwidth usage.

5.2 Opportunities for Prefetching and Caching

We analyze the data to answer three key questions:
(1) Which context determines the ads to be displayed?
(2) Are some ads displayed more frequently than others?
(3) What's the average lifetime and size of ads?

Some contexts are more important. We see that ads are far more dependant on certain contexts. To analyze the importance of different context, we quantify the information contained in each context using mutual information (MI) from information theory. The entropy is a measure of uncertainty for a random variable, and the mutual information of two random variables measures the amount of uncertainty reduced by knowing one of the two. We take an ad as a random variable whose value is its content and calculate its entropy and the mutual information of ads and different contexts with general statistics (Table 1). We make two important observations. First, the number of unique ads are relatively small. Across different fetches same ads reappear many times, which suggest that prefetching and caching can be effective. Second, location and app names give much more information than other contexts.

We studied the overlap of advertisements across different locations and applications. In Table 2, we quantify the overlap across 10 countries and 10 states in the US for the SH application over a period of 14 days. We found that ads served in different countries as well as across cities of the same country were different. A similar computation for ads seen across three different applications from Pittsburgh, US also shows differences in the ads served reaffirming the MI results .

These results show that the prediction of context is critical in prefetching relevant advertisements. If the prediction is not accurate, bandwidth will be wasted and our solution will have little benefit. However, as we shall see in Section 6.1 below, these important contexts can be predicted with relatively high accuracy.

Table 1: Location and app names are important contexts.

total # of ads	1.8M
# of unique ads	17178 (0.9%)
Entropy(ad)	**9.45 bits**
MI(ad, (location, app name))	**3.61 bits**
MI(ad, location)	**2.54 bits**
MI(ad, app name)	1.01 bit
MI(ad, OS)	0.54 bit
MI(ad, carrier)	0.14 bit
MI(ad, age)	0.11 bit
MI(ad, gender)	0.03 bit
MI(ad, keyword)	0.03 bit

Table 2: Fraction of overlap between advertisements across location and application context changes.

Change in context	Average overlap	Std. Dev.
Across 10 different countries	5.41%	8.9%
Across 10 states in US	41.7%	9.57%
Across three applications	4.86%	3.4%

Ad display distribution is skewed. We also observed that the distribution of ads is heavily skewed towards a small number of popular ads. Figure 2 shows this distribution. The top 100 ads are fetched more than 50% of times across all users and contexts. On average, they are fetched more than 82 times each at each device. This suggests that caching of ads can be very effective in reducing the bandwidth consumption of advertising. Previous measurements at the mobile infrastructure [28] and on the handsets [12] show that the same ads are repeatedly fetched as they are not cached.

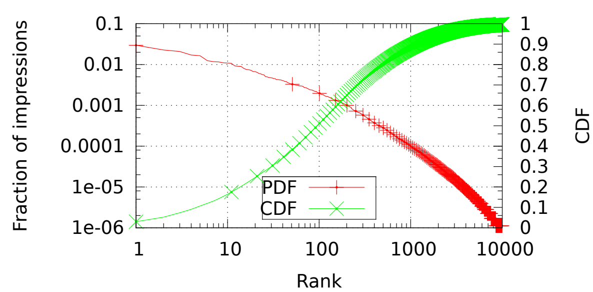


Figure 2: Top 100 ads (0.58%) are fetched for more than 50% of the times.

Some ads have a long life. Figure 3 shows the complementary CDF of ads' lifetime. The lifetime of an ad is the time for which it appears on devices before it is not seen anymore. The ad lifetime reflects advertisers' constraints, such as the budget and the duration of ad campaigns. We see that about 37% of the advertisements are still served by the advertisement network after 24 hours. In fact, 14% of ads continue to be served even after a week. This implies that a significant fraction of the ads can be cached for a relatively long time on mobile devices. For example, if CAMEO's prefetched bulk-ad refresh interval is once per

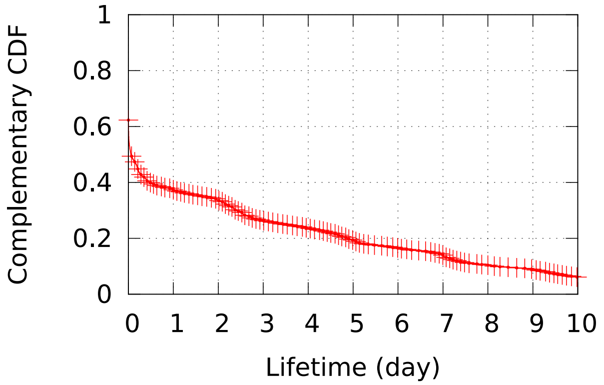


Figure 3: CCDF of ad lifetime

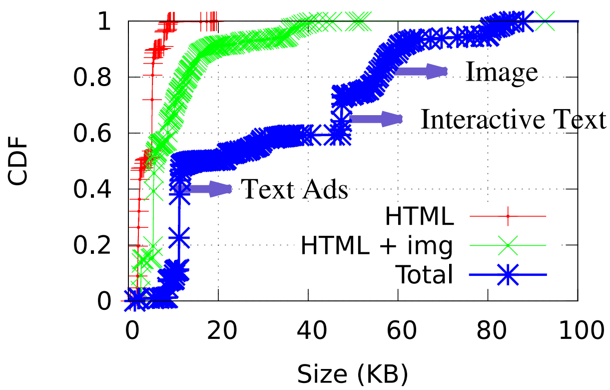


Figure 4: CDF of the size of advertisement

day, on average approximately 63% of all ads will get replaced every day. Note that CAMEO can still prefetch all ads regardless of their lifetime.

Figure 4 shows the size distribution of the ads. The figure shows the total size and the breakdown of an ad's components. All ads are HTML based and they contain references to Javascript code, icons, and/or images. We see that ad-specific components (ad HTML and images) only take up a fraction (median = 5.8 KB) of the total size (median = 15 KB). On closer inspection of the HTML files, we observed that even the ad-specific HTML contains large amounts of redundancy as it is generated from a template. Moreover, we see that the same Javascript code is repeatedly downloaded every time to the mobile. In Section 7, we show that significant bandwidth savings are achievable by exploiting these redundancies.

5.3 Network Usage Patterns

We characterize two aspects of user behavior that are closely related to CAMEO's approach:
(1) How much bandwidth do people consume, and how often do people use ad-supported apps?
(2) How often are users connected to cheap networks?

Figure 5 shows us the average daily bandwidth consumption for different users in our user study. Average users consume ≈30 MB of upload and download data. 44% of the data was consumed while on a metered bandwidth network.

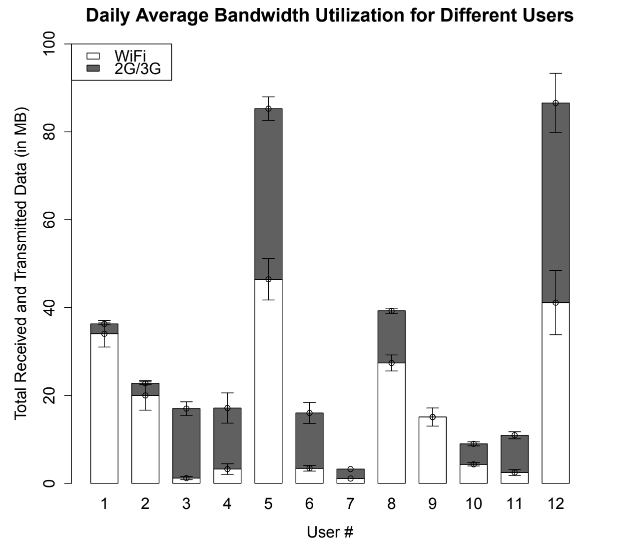


Figure 5: Typical bandwidth utilization of different users

We also calculated that the fraction of time the users spent on ad-supported applications was about 11%. The most actively used apps (more than 60%) were social networking and instant messaging related mobile apps which as of now do not contain advertisements.

Next we analyzed the average times per day that users had different types of connectivity. Figure 6 reports the number of hours in a day (averaged over 30 days) that ten users are connected to different networks. We observe that for both populations, the users were connected to cellular networks for most parts of the day. (For the remaining users we did not have the data for the entire 30 days but the results are similar.)

We must point out that this reported result is our best estimate due to inherent limitations in the Android OS and our own measurement software. We label the user as 'not connected' when the phone is turned off or the airplane mode is set on the phone. We label a user having WiFi only when they obtained an IP from a WiFi network. Note that when a user is on WiFi s/he may still have no Internet connectivity. If the user was 'not connected' and not on WiFi, we assumed that the user had cellular connectivity since cellular data coverage in Singapore is 100%.

6. IMPLEMENTATION OF CAMEO

We now describe, in greater detail, our implementation of each of the different components of CAMEO.

6.1 Context Prediction for Ad Prefetching

In Section 5.2, we showed that *location* and *application* contexts appear to be the two most important determinants of the advertisement selection logic. For prefetching to be successful, CAMEO must predict these two contexts for a user accurately. CAMEO's current implementation uses a simple profile-based predictor to predict such context. Location prediction is fairly well studied (e.g., [25, 24, 6, 20, 17]); moreover, in our study, city-level location granularity was sufficient, as finer-grained location did not result in any statistical differences in the advertisements selected.

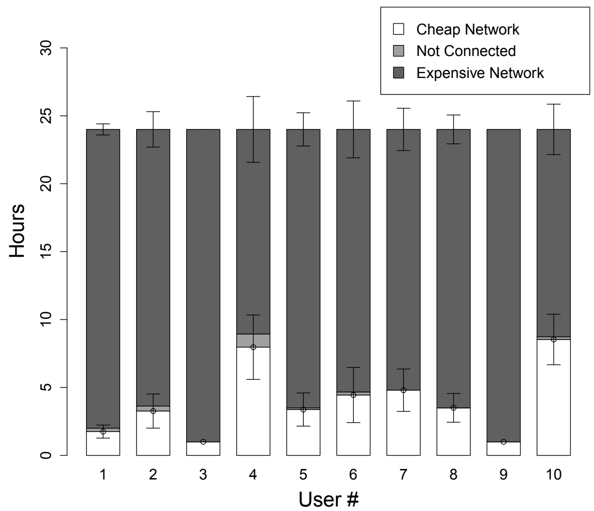


Figure 6: Avg. user hrs. on different networks

We thus focus on predicting the set of applications that individual users will execute in the future. Presently, for any specified future interval, CAMEO determines the set of *possible applications* as the union of all applications that have a non-zero access probability over *any 'bin' (explained shortly) within that interval*. In other words, if the user has a non-zero likelihood of using any particular application, then CAMEO assumes that it must prefetch advertisements associated with that application context as well.

To build a predictive profile, our implementation divides each day into smaller time intervals termed *bins* (currently, each bin is 15 minutes long) and considers tuples of the form *(day of week, bin)* for prediction. For each such tuple, the algorithm computes the fraction of time spent interacting with different applications. Figure 7 illustrates the computation process. For example, for the bin (Monday, 9:30-9:45am), the algorithm computes the probability for using $App1$ by calculating the fraction of time the user spent accessing $App1$ between 9:30–9:45am over the past few Mondays. In Section 7.1, we shall evaluate three different strategies for using these per-bin probabilities to determine the set of applications likely to be used by the user over any given time period.

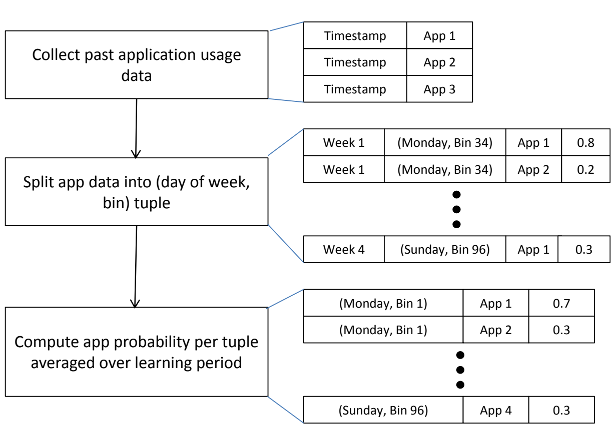


Figure 7: Simplified context prediction flow.

6.2 Local Advertisement Selection

The CAMEO implementation supports all three advertisement selection models discussed in Section 4.2. In CAMEO, the corpus of advertisements is currently identified by a unique device-specific ID, and the corpus contains the advertisements as well as an index of the hashes of each advertisement. In the ANAS model, CAMEO issues a $FetchAd()$ request, containing the corpus ID and receives back the ID (hash) of the advertisement selected for display. To support LAS, CAMEO downloads a simple XML-based ruleset that dictates the advertisement selection logic. This ruleset is used in the $FetchAd()$ method, taking the local corpus as input and returning the ID of the selected advertisement. CAMEO's implementation of BEAS currently uses a weighted selection logic where an advertisement from the corpus is selected randomly in proportion to its *display frequency* value, f_i. CAMEO also maintains a per-ad counter, as well as the list of timestamps, to ensure that the BEAS limits (on f_i and Max_i) are not violated. For all three models, the selection history is also logged in our currently primitive version of the Accounting and Verification component.

6.3 ISP Negotiation

We now describe the implementation of CAMEO's ISP negotiator that enables on-the-spot exchange between advertising rights of apps and subsidized connectivity to mobile ISPs.

Pricing and negotiation model: While CAMEO can support any number of ad-based charging models with ISPs, the present implementation supports two of the most common pricing models for access connectivity: 'per unit time' and 'per byte transferred'. In the first model, time is partitioned in slots of t seconds, and the ISP obtains the right to show one advertisement in each slot. In the second model, the ISP is allowed to show a total of x advertisements (of maximum size S) in exchange for y bytes of data traffic.

CAMEO's Accounting and Verification module then cooperates with the ISP infrastructure to monitor the data consumed by each individual application (possible alternative approaches for such monitoring are suggested in Section 9), and subsequently enforce its agreed-upon barter model. The CAMEO API alerts each individual application that its negotiated traffic contract is about to expire, either because sufficient time has elapsed since the display of the last ISP-provided advertisement or because the application traffic volume is approaching the specified limit but has not yet displayed enough ISP advertisements. To accommodate background applications (which clearly cannot use advertising views as a currency), CAMEO also implements an inter-Application request model, where a background application (e.g., an email client) can request the foreground application (via CAMEO) for a portion of its traffic quota. While CAMEO can mediate these requests, the acceptance or rejection by the foreground application would depend entirely on prior business/partnering arrangements among the respective developers.

Overview of the process: When an application starts up and the device does not have pre-existing connectivity, CAMEO searches local ISPs that supports the negotiation and identifies a negotiator address for each ISP. Such information can be exchanged as part of the selection and association, similar to protocols such as GSM MAP and 802.11u

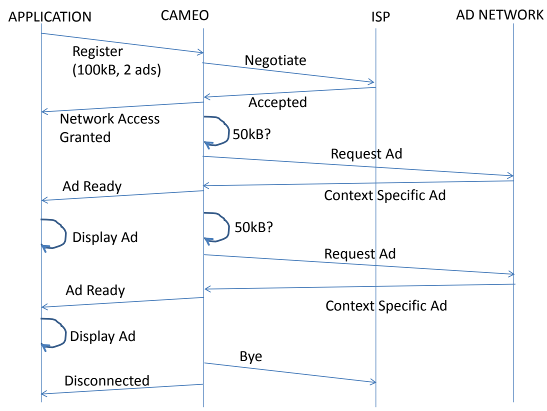


Figure 8: A CAMEO-ISP negotiation example

ANQP. Our current implementation uses a DHCP option field.

CAMEO then performs negotiations on behalf of the application and then accepts the best 'deal' it can find with an agreeable ISP. Note CAMEO can be used in conjunction with users' existing data plans for ad-subsidized connectivity. However, if all negotiations fail, CAMEO informs the user and allows the user to fallback to other data plans, such as a micropayments option. CAMEO also supports renegotiation to reach an agreement. For example, the mobile application may offer to display ISP-provided ads more frequently. Conversely, the application may re-negotiate and lower its bandwidth requirements. For example, it may choose to change the richness of the content (fetching only text instead of both images and text) in order to obtain free connectivity to the Internet.

Negotiation protocol: Figure 8 illustrates a typical negotiation process, based on a per-byte model, where the negotiated agreement would require the ISP to transfer y bytes of network traffic in exchange for the ability to select and display x advertisements.

On initialization, the CAMEO-enabled mobile application registers with CAMEO with its expected bandwidth usage and maximum available 'slots'. (The application also registers other details such as the different ANs used by the application.) For negotiating network access, CAMEO contacts the ISP's negotiation server with the offer of the application ($y = 100$KB and $x = 2$.) In our example, once the ISP accepts this offer, CAMEO informs the application of the approved access.

Next, the application specifies the source ports (and the type—TCP/UDP) it will use for its traffic and also registers a callback function with CAMEO to receive notifications when ISP's advertisements are to be shown. CAMEO then interacts with the ISP network elements to set up the monitoring and enforcement of these flows. At appropriate intervals (in our example, once after every 50KB of data transfer), CAMEO fetches new advertisements from the ISP and asks the application to display them.

7. EVALUATION

Using our prototype implementation, we evaluate the following three aspects of CAMEO:

1. How accurately can we predict users' future context? In Section 7.1, we show that our prediction algorithm achieves 84% accuracy on average.

2. How much energy and bandwidth savings does it achieve? In Section 7.2, we show that we reduce the energy consumption by 25 to 37 times and bandwidth by up to 4.8 times when displaying 100 advertisements while incurring very little overhead.

3. Is CAMEO's ISP negotiation feasible? We show that the negotiation protocol introduces little overhead and ISPs may actually benefit from switching to an ad-supported model.

7.1 Context Prediction

We measure the accuracy of prediction using two metrics—time period of prediction and the fraction of time an application was used. We divide each day into time bins. Using past history (training set), each bin has a list of applications that were observed as being used in that bin and we know the number of bins they appeared in. (i.e. fractional use). We observe future usage (test set) to compute the prediction accuracy.

1. Binary prediction with short bins: To measure the accuracy over short periods, each bin is set to be 15 minutes long (total 96 bins/day). For each bin in the test set, we compute the fraction of applications in the test set that were also part of the training set. If all applications in the bin remain the remain, we count that as a successful prediction. The prediction accuracy is now defined as the fraction of bins that were predicted correctly.

2. Binary prediction of long bins: To measure accuracy over a day, each bin is set to be a day. We perform a binary test of whether an application was used on the same day. The prediction accuracy is now simply the fraction of applications that were predicted correctly.

3. Weighted prediction: To measure how often an application context appears in a day, the difference in the fraction of time (δ_i) an application i was used in the test set versus the training set is computed. The accuracy is then computed as $100 - \sum \delta_i \cdot 100$

We divided the one-month period of the users' logs into two sets of two weeks each. The first two weeks' data is used as the training set. The next two weeks' data is used as the test set. Table 3 summarizes the accuracy results obtained using the above three approaches for three random users on a Monday. Figure 9 summarizes the mean accuracy of the different approaches (for the two week data set) for each day of the week (for user #3).

Table 3: App. Prediction Results for Monday

User #	Approach #1	Approach #2	Approach #3
1	75%	30.77%	82.70%
2	51.07%	81.25%	90.05%
3	67.86%	90.90%	76.41%

We make two observations here:

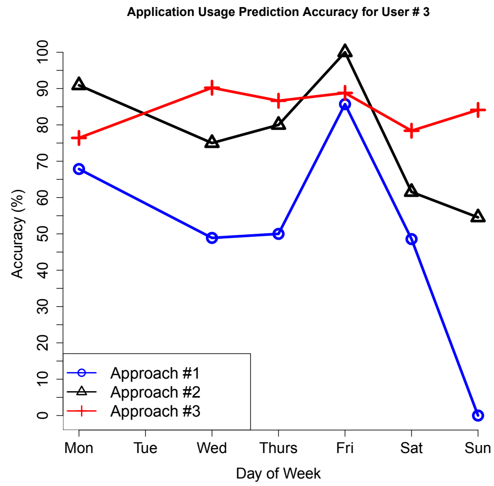


Figure 9: Prediction Accuracy.

Table 4: Bandwidth usage (in MB) with and without CAMEO prefetching

No. of ads	Without CAMEO	With CAMEO pre-fetch	
		ANAS	LAS & BEAS
10	0.646	0.321	0.322
100	6.334	1.375	1.324
1000	61.129	5.439	4.754

1. Predicting application context over shorter time periods was difficult with the simple approaches but was much easier for longer time periods.

2. Users operate phones inconsistently over weekends.

7.2 Resource savings on the mobile

We demonstrate that CAMEO can significantly reduce the bandwidth and energy consumption while incurring very little overhead. We compare the performance and overhead (size of cache needed and latency) of delivering mobile advertisements with and without CAMEO.

Bandwidth savings: We estimate the bandwidth usage with and without CAMEO.

Without CAMEO each advertisement is fetched on demand without being cached at the mobile device [12, 28]. However, when using CAMEO, each unique advertisement is sent only once during the bulk ad pre-fetch phase. We generate the cache for advertisements using the corpus and the context from the user data. Table 4 lists the amount of network traffic that is created for displaying 10, 100 and 1000 advertisements for our demo application. This includes all network traffic including DNS lookups but it assumes that the mobile device is already connected to the network.

We observe that when CAMEO is used, significant bandwidth savings are achieved even with a moderate number of displayed advertisements, regardless of its local ad selection methods (ANAS, LAS, and BEAS). These savings occur because CAMEO removes (1) significant repetitions of advertisements (see Figure 2) by sending each unique advertisement once and (2) overhead of DNS lookups and repeated Javascript code [12, 28]. We believe that compression can gain further savings since most of the advertisements share templates and are pure text (HTML) (refer Section 5.2).

Table 5: Energy consumption (in Joules) for fetching different no. of advertisements with and without CAMEO prefetching

No. of ads	Current Systems		With CAMEO pre-fetch			
			ANAS		LAS & BEAS	
	3G	WiFi	3G	WiFi	3G	WiFi
10	140.53	21.16	138.17	20.17	29.41	4.78
100	1403.25	210.73	1369.02	196.31	36.33	7.70
1000	14017.2	2100.87	13632.8	1938.96	60.02	17.67

Table 6: Size of cache in MB as a function of the number of advertisements displayed to the user for different apps. DM=demo, HD=Hanping, PL=Plume, SH=SoundHound

Ad impressions	DM	HD	PL	SH
10	0.27	0.18	0.22	0.27
100	1.06	0.44	0.82	0.86
1000	3.25	0.75	2.36	2.06

Energy savings:

Table 5 reports the energy consumption for fetching advertisements using CAMEO. We measured the energy consumption using the Monsoon power measurement tool on a Samsung Galaxy SIII phone running Android OS v4.0.4, on Singapore's SingTel network and the Singapore Management University's WiFi network in Singapore. We use the same cache described above for the bulk ad pre-fetch. The results show that CAMEO provides significant energy savings only when the ad selection is completely done locally (LAS and BEAS). We observe no significant difference in energy consumption between the current system (w/o CAMEO) and CAMEO's ANAS, which contacts the ad network every time an ad is displayed. CAMEO reduces the energy consumption by 38 times on 3G and 30 times on WiFi when 100 advertisements are displayed using the LAS or BEAS logic. Note that CAMEO's energy consumption includes the energy consumed in the prefetch phase. This may increase as corpus size increases. However, since CAMEO can be configured to prefetch only when the mobile device is on a cheap network and connected to a power source, the actual impact on energy consumption for a large corpus may be negligible.

Storage overhead: The storage overhead of the advertisement corpus can be very user specific. It depends on a number of factors such as the prediction accuracy, the time which an advertisement supported application spends in the foreground, the desired hit-rate in the cache, and the advertising network's priorities. To get a rough estimate on the *minimum* size of the cache needed for serving ads, we make a simplifying assumption—we are able to predict context perfectly. We compute the statistical frequency with each unique advertisement is shown for the different contexts and typical time spent in each of these contexts.

Table 6 lists the minimum size of cache needed to show different number of advertisements for each app in our study. The cache size will increase if the advertisement network wishes to cover more contexts than the ones that have been predicted. It will also increase because context-prediction can never be 100% accurate. Since each user typically uses multiple ad-supported apps, the total storage requirements

Table 7: Network changes and the corresponding signaling messages for different users

User	N/w Changes	Apps	Negotiations	Messages	Bytes
1	1.6	1.75	3	34	1863
2	2.2	1.09	3	29	1863
3	3.7	1.54	6	68	3726
4	2.4	2.5	6	72	3726
5	4.1	2.3	10	114	6210

will be much higher. Nevertheless, the storage required is a very small fraction of the total space seen on most mobile devices.

Latency for advertisement display: We measure the latency of invoking the API to get an advertisement for display. To measure this, we record the timestamps on the mobile just before and just after the code for fetching the advertisement is executed, i.e. before the advertisement is displayed on screen. For LAS and BEAS, the advertisements are fetched locally and the latency is about 0.1ms. For ANAS, fetching an advertisement over the existing 3G networks took approximately 6.1 seconds for an advertisement, almost 60% of which was spent turning ON the 3G radio. This represents the worst case when there is no pre-existing connectivity. On the WiFi network, the latency was closer to 3.8 seconds, which was also spent mostly on turning ON the WiFi radio. For ANAS, we observed similar latency values when fetching advertisements without CAMEO. Fetching very large advertisements with ANAS was slightly faster by about 20ms.

7.3 ISP Negotiation

We first show that our ISP negotiation introduces minimal overhead and that this can be seamlessly integrated into the association and connection establishment process. Then, we explore whether ISPs would have any financial incentives to provide an ad-subsidized plan through CAMEO.

Signaling load: We first evaluate the signaling overhead of our negotiation protocol. We deployed our prototype on a campus network, placing a mobile device on the campus WiFi network and the ISP negotiator on a different (wired) subnet. For the negotiation process described in Section 6.3, a total of 621 bytes were exchanged employing 12 messages (excluding the TCP handshakes).

We analyzed our user data set to determine how many times a user switches between two networks in a day (averaged over a one-month period). Every time a user changes a network, all the CAMEO enabled applications that are in use will need to re-negotiate thereby creating a flurry of messages on the network. Table 7 shows the mean number of times (per day) five different users switched between networks, the mean number of advertisement supported applications in use (per day) and the corresponding number of messages that are generated on the network for each network switch. (The rest of the users fell between the extremes and their data is not reported here.) For Table 7, we assumed that each user used the ad-supported application on every network.

Financial Feasibility: We explore whether it is feasible for the ISP to financially break even the cost of giving free bandwidth with advertisements. We consider two different charging models for connectivity: 1) where a user is charged by time and 2) where the user pays for a bandwidth budget. Boingo follows the first model and plans that charge $8 for 24 hours access (≈0.01 cents/second) and $10 for a month (≈ 0.0004 cents/second) . AT&T follows the second model and offers 300 MB for $20 (6.7 cents/MB) and 5 GB for $50 (1 cent/MB).

According to KPCB's Internet Trends report [14], the effective CPM (cost per 1000 impressions) on mobile devices was $0.75. At this rate, Boingo will break even for showing ads every 8.1 seconds and every 3.2 minutes for its two different rate plans. AT&T will break even for showing 88.8 ads/MB (or an ad per every 11.25 KB data) delivered on its lower tier plan and 13.3 ads/MB (or an add per 75 KB data) on its higher tier plan.

Table 8: Mean application network traffic data rate and frequency of advertisements for popular advertisement-supported free applications

Application	Ad fetch interval	Ads/second	Ads/MB
Angry Birds	45 sec	0.022	50
Sound Hound	45 sec	0.022	7
SG Buses	37 sec	0.027	36
The Weather Channel	60 sec	0.016	30

Although the rate of advertisements may be too high for applications that generate large amount of traffic (such as a web browser fetching 1 MB web pages), we find that this model is feasible for applications that generate relatively small amount of traffic (such as an email application). For example, Table 8 shows the rates of advertisements displayed for some popular "free" applications. From the data, we observe that both ISPs (offering different pricing models) can break even for some applications with users subscribing the higher tier plans.

8. RELATED WORK

Monetization of personal information: Many studies [22, 13, 5] propose monetization of users private information through an open market where users directly get paid for the information they choose to expose to targeted advertising. They propose various auction-like mechanisms that allow user's control and trade-off in the amount of information they expose. Rather than directly compensating users, CAMEO propose the use of ad slots as the medium of exchange between users attention and bandwidth by extending the existing model.

Resource use of mobile advertising: Eprof [19] analyzed the energy consumed in smartphone apps and revealed that mobile advertising library is responsible for 65 to 75% of the energy consumed by free apps. Khan et al. [12] show that free apps are actually free because they bear cost to most users who are on a metered data plan. Vallina-Rodriguez et al. [28] presented a comprehensive measurement study of the traffic generated by mobile advertising. Analyzing a trace of a mobile carrier with more than 3 million subscribers, they show that ads account non-significant fraction of the data traffic. Advertising account for 1% of all mobile traffic, and for 50% of Android users, more than 5% of the traffic is related to advertising. They also suggest that prefetching can be an effective solution. However, they do not address how context-dependant ads can be prefetched. Mohan et al. [16] also propose pre-fetching of advertisements to reduce energy consumption. They use similar techniques as ours to prefetch advertisements. However their focus is less on predicting future context and more on selecting the set of advertisements that will preserve the future deadlines for displaying advertisements.

Security and privacy in advertising: Many studies have pointed out the potential user privacy risks and security problems. AdRisk [9] showed that ad libraries collect private information ranging from user's location to more private information such as call logs, browser bookmarks, and list of apps installed. To address such privacy problems, many system have been proposed. For example, AdDroid [21], AppFence [11], Adnostic [26], and Privad [10] provide users with better control over the information transmitted from their mobile device.

Prevention or mitigation of click and impression fraud has also been an important area of study. While detection of click frauds or spams have primarily been the interest of advertisement networks for many years [7, 27], recent studies [3] characterized the behavior of click-spammers. Systems, such as AdSplit [23] and Quire [4], propose remote attestation-based verification of advertisement related events as a solution for click-spams.

We take the same view as many of these studies that support for advertising should be incorporated into the mobile operating system. Therefore, many of the studies are complementary to CAMEO.

Protocols for seamless roaming: Mobile networks provide standard protocols for roaming. GSM's Mobile Application Part (MAP) supports dynamic authentication of a roaming user and assignment of a roaming number. Similarly, 802.11u also supports seamless roaming and cellular to Wi-Fi offload. In the network selection process, 802.11u's Access Network Query Protocol (ANQP) allows mobile devices to query information such as hotspot provider's name and roaming agreements in place. We envision that CAMEO's negotiation protocols will be part of network discovery and selection protocols, such as GSM MAP and 802.11u ANQP.

9. DISCUSSION AND FUTURE WORK

While this paper describes the basic mechanisms that enable optimized advertisement delivery and advertisement bartering for connectivity, there are many repercussions of this design on the advertising and mobile application ecosystem that we leave for future work. For example, Section 4.4 describes some of the issues associated with accounting and verification of ad delivery/display. Below, we discuss some of the issues associated with network traffic policing, privacy, content management and revenue generation.

Traffic Policing.

One of CAMEO's core goals is to enable applications to use ad bartering to pay for the connectivity that they use. This will require the creation of a contract between the application and the ISP regarding the terms of the exchange; that must then be enforced! Section 4.4 describes techniques that can be used to verify the ad display part of the contract; however, we also need to enforce the network usage part. For example, the ISP could perform per-flow accounting on the edge router to police the contract. However, some edge routers may not have the memory or CPU resources to perform the necessary flow accounting — especially if hundreds of mobile devices are connected and their applications keep shifting between foreground (where they can see advertisements) and background (where they cannot and the mobile OS closes the open ports) states. One hybrid solution would be for the ISP to perform per-device (i.e., aggregate) accounting but for the CAMEO service on each mobile device to perform per-flow accounting. Note that the CAMEO service must already implement per-flow accounting to alert applications when either a) sufficient time has elapsed since the display of the last ISP-provided advertisement or b) their traffic volume is approaching the specified limit without enough ISP advertisements being displayed on the client. However, the aggregate accounting solution does raise issues related to ISPs trusting the end device accounting; we defer solutions to these issues to future work.

Privacy.

CAMEO relies heavily on accessing user context to enable targeted advertising. However, the user should be able to specify which context data can be used by CAMEO. However, it is likely that better context data will result in carefully targeted advertisements that could produce more revenue per ad. Hence, CAMEO could incorporate consumer friendly negotiation mechanisms that allow more interesting context to be traded off for fewer ads.

Content Management.

CAMEO's support for both ad developer and network provider advertising raises issues related to regulation and fairness. When a mobile application developer signs up today with an AN, there are clear rules and agreements on what type of advertisements can be shown. In future work, we plan to explore mechanisms to ensure that local ISPs use the same content policies that developers have with ANs. For example, an ISP should not show an inappropriate advertisement in applications meant for children.

Revenue Maximization.

One of CAMEO's key features is that it allows ISPs, app developers, ad networks, and users to engage in revenue exchange associated with advertising. However, in response, each of these players is likely to develop new revenue maximizing strategies. For example, we are currently investigating better context prediction and pre-fetching strategies that ad network could use to improve ad targeting and minimize the number of ads they must deliver (and "pay" for using ad slots) over expensive connectivity. We also plan to explore the tradeoffs of using pre-fetched advertisements instead of the dynamic auctions that ad networks use to select ads today. In addition to exploring the best strategies for each player, we also plan to explore the interaction of these strategies and their impact on the whole systems behavior.

10. CONCLUSION

Mobile advertising is a rapidly growing industry [1] that has a significant impact on the business models of the mobile Internet. However, the mechanisms that support sending advertising to mobiles devices have not been well though out. In particular, the mechanisms disenfranchise key stakeholders including end users and the mobile ISPs. In particular, the current model unfairly imposes cost to these stakeholders without providing a flexible business model for them. While many studies have addressed piecemeal solutions for the various problems with the current mobile advertising ecosystem, the core thesis of CAMEO is that the mobile advertising ecosystem should be redesigned to allow win-win interactions between all the key stakeholders. This paper explored key designs of such an architecture. CAMEO employs predictive prefetching that effectively decouples the tasks of advertisement selection and delivery. The resulting flexibility allows CAMEO to schedule proactive bulk advertisement retrieval in a way that minimizes the cost associated with fetching those advertisements. In addition, CAMEO provides a flexible negotiation model that allows individual applications to negotiate with mobile ISP for bandwidth access; opening up new business models and opportunities.

11. REFERENCES

[1] US online ad revenues continue growth, set Q1 peak. http://www.marketingcharts.com/wp/direct/online-ad-revenues-hit-record-317b-in-2011-surpass-cable-tv-21842/, June 2012.

[2] Digital video advertising trends: 2012. http://www.breakmedia.com/digitalvideoadstudy, 2012.

[3] V. Dave, S. Guha, and Y. Zhang. Measuring and fingerprinting click-spam in ad networks. In *Proc. ACM SIGCOMM*, pages 175–186, 2012.

[4] M. Dietz, S. Shekhar, Y. Pisetsky, A. Shu, and D. S. Wallach. Quire: Lightweight provenance for smart phone operating systems. In *Proceedings of the 20th USENIX conference on Security*, 2011.

[5] A. Ghosh and A. Roth. Selling privacy at auction. In *Proc. ACM conference on Electronic Commerce*, 2011.

[6] J. Ghosh, M. J. Beal, H. Q. Ngo, and C. Qiao. On profiling mobility and predicting locations of wireless users. In *Proc. international workshop on Multi-hop ad hoc networks: from theory to reality*, REALMAN '06, 2006.

[7] Google ads: Ad traffic quality resource center. `http://www.google.com/intl/en_ALL/ads/adtrafficquality/index.html`.

[8] How cost are calculated in adwords. `http://support.google.com/adwords/bin/answer.py?hl=en&answer=1704424`.

[9] M. C. Grace, W. Zhou, X. Jiang, and A.-R. Sadeghi. Unsafe exposure analysis of mobile in-app advertisements. In *Proc. ACM conference on Security and Privacy in Wireless and Mobile Networks*, pages 101–112, 2012.

[10] S. Guha, B. Cheng, and P. Francis. Privad: Practical Privacy in Online Advertising. In *Proc. Symposium on Networked Systems Design and Implementation (NSDI)*, Boston, MA, Mar 2011.

[11] P. Hornyack, S. Han, J. Jung, S. Schechter, and D. Wetherall. These aren't the droids you're looking for: Retrofitting android to protect data from imperious applications. In *Proc. ACM CCS*, pages 639–652, 2011.

[12] A. J. Khan, V. Subbaraju, A. Misra, and S. Seshan. Mitigating the true cost of advertisement-supported "free" mobile applications. In *Proc. Workshop on Mobile Computing Systems & Applications*, 2012.

[13] K. C. Laudon. Markets and privacy. *Communications of the ACM*, 39(9):92–104, Sept. 1996.

[14] M. Meeker and L. Wu. Kpcb internet trends 2012, May 2012. `http://www.kpcb.com/insights/2012-internet-trends`.

[15] C. Mims. Facebook's plan to find its next billion users: convince them the internet and Facebook are the same. `http://qz.com/5180/facebooks-plan-to-find-its-next-billion-users-convince-them-the-internet-and-facebook-are-the-same/`, Sept. 2012.

[16] P. Mohan, S. Nath, and O. Riva. Prefetching Mobile Ads: Can advertising systems afford it? In *Proceedings of the Eighth European Conference on Computer Systems, Eurosys'13*, 2013.

[17] A. J. Nicholson and B. D. Noble. Breadcrumbs: forecasting mobile connectivity. In *Proc. ACM MobiCom*, pages 46–57, 2008.

[18] B. D. Noble, M. Satyanarayanan, D. Narayanan, J. E. Tilton, J. Flinn, and K. R. Walker. Agile application-aware adaptation for mobility. In *Proc. ACM Symp. on Operating Sys. Principles*, Oct. 1997.

[19] 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 *Proc. ACM european conference on Computer Systems*, EuroSys, pages 29–42, 2012.

[20] P. N. Pathirana, A. V. Savkin, and S. Jha. Mobility modelling and trajectory prediction for cellular networks with mobile base stations. In *Proc. ACM MobiHoc*, 2003.

[21] P. Pearce, A. Felt, G. Nunez, and D. Wagner. Addroid: Privilege separation for applications and advertisers in android. In *Proc. AsiaCCS*, 2012.

[22] C. Riederer, V. Erramilli, A. Chaintreau, B. Krishnamurthy, and P. Rodriguez. For sale : Your data: By : you. In *Proc. ACM HotNets*, 2011.

[23] S. Shekhar, M. Dietz, and D. S. Wallach. Adsplit: Separating smartphone advertising from applications. In *Proceedings of the USENIX Security Symposium*, 2012.

[24] L. Song, D. Kotz, R. Jain, and X. He. Evaluating next-cell predictors with extensive wi-fi mobility data. *IEEE Transactions on Mobile Computing*, 5(12), Dec. 2006.

[25] M. H. Sun and D. M. Blough. Mobility prediction using future knowledge. In *Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*, MSWiM '07, 2007.

[26] V. Toubiana, A. Narayanan, D. Boneh, H. Nissenbaum, and S. Barocas. Adnostic: Privacy preserving targeted advertising. In *Proc. NDSS*, 2010.

[27] A. Tuzhilin. The Lane's Gifts v. Google Report. `http://googleblog.blogspot.com/pdf/Tuzhilin_Report.pdf`.

[28] N. Vallina-Rodriguez, J. Shah, A. Finamore, Y. Grunenberger, K. Papagiannaki, H. Haddadi, and J. Crowcroft. Breaking for commercials: Characterizing mobile advertising. In *Proc. ACM IMC*, pages 343–356, 2012.