Understanding Mobile App Usage Patterns Using In-App Advertisements

Alok Tongaonkar[1], Shuaifu Dai[2,3], Antonio Nucci[1], and Dawn Song[3]

[1] Narus Inc, USA
[2] Peking University, China
[3] University of California, Berkeley, USA
{alok,anucci}@narus.com, daishuaifu@pku.edu.cn, dawnsong@cs.berkeley.edu

Abstract. Recent years have seen an explosive growth in the number of mobile devices such as smart phones and tablets. This has resulted in a growing need of the operators to understand the usage patterns of the mobile apps used on these devices. Previous studies in this area have relied on volunteers using instrumented devices or using fields in the HTTP traffic such as `User-Agent` to identify the apps in network traces. However, the results of the former approach are difficult to be extrapolated to real-world scenario while the latter approach is not applicable to platforms like Android where developers generally use generic strings, that can not be used to identify the apps, in the `User-Agent` field. In this paper, we present a novel way of identifying Android apps in network traces using mobile in-app advertisements. Our preliminary experiments with real world traces show that this technique is promising for large scale mobile app usage pattern studies. We also present an analysis of the official Android market place from an advertising perspective.

1 Introduction

In recent years, there have been dramatic changes to the way users behave, interact and utilize the network. More and more users are accessing the internet via mobile devices like smart phones and tablets. According to recent statistics by Canalys [1], 488 million smart phones have been sold in the year 2011, compared to 415 million personal computers. Users of these devices typically download applications (commonly called mobile apps) that provide specific functionality. A majority of these apps access the internet. For example, 84% of the 55K Android apps in the official Android app market [2] that we randomly picked, required permission for Internet access. This has led to a burgeoning interest amongst network operators in understanding the mobile app usage patterns in their networks.

Recent years have seen an increasing number of research works that analyze network traffic to understand usage behaviors of mobile apps ([3,4]). However, these papers rely on techniques for app identification which are not applicable for Android apps or rely on having access to the Android devices and monitoring the specific devices. For example, Xu et al [3] and Maier et al [5] use `User-Agent`

M. Roughan and R. Chang (Eds.) PAM 2013, LNCS 7799, pp. 63–72, 2013.

field in the HTTP header to identify the app. Apple has a guideline for iOS which requires that this field contain app identifier. However, this guideline is not strictly enforced. For Android apps the situation is even worse since developers generally put some generic string (not unique to the app but identifying the Android version and such) in this field. On the other hand the approach taken of making some users use apps on specific devices to collect network trace and profile app usage does not give real-world data ([4,6]). Moreover, manual execution of apps suffers from the problems of scalability. The approach of using `Host` field in the HTTP header for identifying the apps does not work all the time because the same host may serve multiple apps. This is typically true when the same app developer such as Zynga publishes multiple apps. Also many platforms, such as Facebook mobile app development platform support apps from different developers. The apps which are developed on these platforms typically use the servers from the platform provider to provide their service. For instance, `m.facebook.com` hosts diverse apps such as *Pirates Mobile*, a gaming app, and *Squats*, a personal training app.

In this paper, we present a new technique of identifying app usage patterns based on the advertising traffic originating from the apps. This technique is based on the observation that mobile apps may communicate with many different servers for different purposes. A typical Android app may contact the web site of the app provider to obtain the API information, connect to a cloud service like Amazon EC2 for downloading some files, contact sites such as `doubleclick.com` and `mobclix.com` to retrieve ads, and provide usage stats to sites such as `googleanalytics.com`. We can classify network traffic from an app into three main categories similar to the classification used by Wei et al [6] as follows: (i) Origin: traffic that comes from the servers owned by the app provider (e.g. `pandora.com` for *Pandora*). (ii) Content Distribution Network (CDN)+Cloud: traffic that comes from servers of CDNs (e.g., Akamai) and cloud providers (e.g. Amazon AWS). (iii) Third-party: traffic from various advertising services (e.g., AdMob) and analytical services (e.g., Omniture).

Previous studies of mobile app usage have focused on either origin traffic ([3]) or CDN+cloud traffic ([7]). We present a different approach by studying usage behavior of mobile apps based on advertising traffic. Advertising is a critical component of the mobile app ecosystems from a financial perspective. We believe that usage patterns studies based on advertisements will be very valuable in future. Many mobile apps use one or more advertising services as a source of revenue. To use these services, developers must register their apps with the advertising service provider. Developers bundle third-party, binary-only libraries (called ad libraries) from the advertising service providers into their apps. The information about the ad libraries being used by an app is usually present in the meta-data provided in the installable package of the apps. We can use this information to understand the distribution of advertisements in the apps.

Another interesting observation is that typically an advertising service provider identifies the app using the app name provided by the developer or unique app identifier generated by the service provider at the time of the registration.

These app names or identifiers are present in network flows to the advertising service providers. We can use these identifiers to study the patterns of mobile app usage from real world network traces.

Mobile in-app ad libraries have been studied before in the context of security and privacy [8,9,10] and energy consumption [11]. This is the first work to present a systematic study of usage patterns of mobile apps using ad flows. We believe that considering the critical role of advertisements on mobile app ecosystems, our research paves the way for new studies which can be very useful for a variety of players like network operators, advertising service providers, advertisers, and mobile app developers. We focus on understanding the app usage patterns on the Android platform in this work. However, the ideas and techniques presented here are equally applicable to iOS and Windows Mobile platforms.

The main contributions of this work are as below.

- We present a systematic study of advertising libraries on the Android platform.
- We present results of analyzing more than 50K Android apps from an advertising perspective.
- We present results from evaluating the network traces from a Tier 2 cellular service provider.

The rest of the paper is organized as follows. In Section 2 we present our analysis of advertisements in apps in the official Android market. In Section 3 we present mobile app usage behavior patterns from real world network traces. We discuss the limitations and future work in Section 4. Finally we present the conclusions in Section 5.

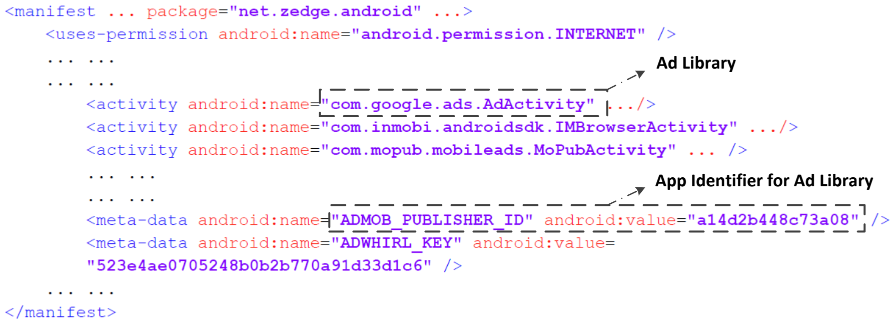
2 App Market Analysis

In this section we present an analysis of the official Android app market, Google Play Store, with respect to the different categories of apps. Note that our goal is not to do a comprehensive study of all apps in the store but give a flavor of the kinds of analysis possible with the advertising information.

2.1 Background

Google Play Store is the most popular Android app market with over 500K apps which includes both free and paid apps. Developers of many of the free apps rely on advertisements (ads) for generating revenue so we focus only on free apps in this paper. Android apps are distributed as special files, called Application Package File (APK), with `.apk` file extension. Along with the application binaries and resources, each APK file contains an `AndroidManifest.xml` file. The manifest file is an XML file that contains meta-data about the app such as the name of the app, permissions required, resources used, libraries used, etc.

Developers of free apps typically use third-party advertising service providers such Google Ads or Smaato to display ads in the app. Ad service providers may differ in the way that ads are provided to the app but they have some common characteristics. Most ad networks provide libraries for user-interface



```
<manifest ... package="net.zedge.android" ...>
    <uses-permission android:name="android.permission.INTERNET" />
    ... ...
    ... ...
        <activity android:name="com.google.ads.AdActivity" .../>
        <activity android:name="com.inmobi.androidsdk.IMBrowserActivity" .../>
        <activity android:name="com.mopub.mobileads.MoPubActivity" ... />
        ... ...
        ... ...
        <meta-data android:name="ADMOB_PUBLISHER_ID" android:value="a14d2b448c73a08" />
        <meta-data android:name="ADWHIRL_KEY" android:value=
        "523e4ae0705248b0b2b770a91d33d1c6" />
    ... ...
</manifest>
```

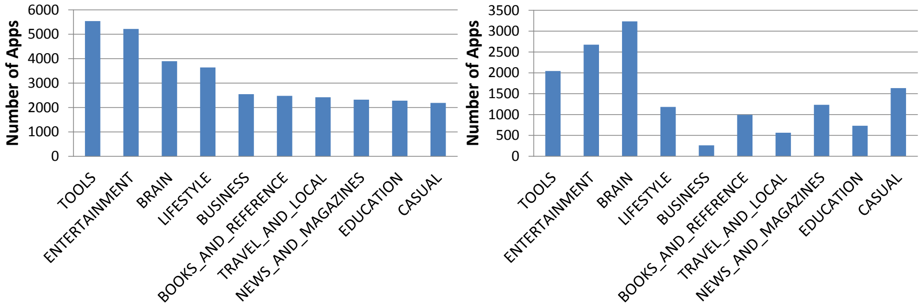
Fig. 1. Sample of Zedge Manifest File

code (to present their ads) and network code (to request ads from the ad networks servers). The libraries are designed to be tightly bundled with host apps to make it more difficult to disable the ad functionality or defraud the ad network. When a developer registers an app with an ad service provider, she may receive a developer identifier or app identifier. The SDK for the ad library contains instructions, on how to embed the ad library in the app, such as the permissions required by the ad library and the mechanism used by the ad service provider to identify the app or the developer. The ad service provider may use either app name or an identifier generated at registration time to identify the app or the developer.

To understand how ad libraries are used, consider *Zedge*, which is a very popular app (more than 1M downloads) that is used for downloading wallpapers and ringtones. We use a tool for reverse engineering third-party, closed, binary Android apps, called `apktool` [12], to extract the manifest file in the .apk file into a human readable form. Figure 1 shows the manifest file for *Zedge*. We can see that the manifest file lists three ad libraries that are embedded in *Zedge* - (i) Google Ads, (ii) InMobi, and (iii) MoPub. Many (but not all) of the ad service providers require the identifier to be mentioned explicitly in the manifest file. For instance, in Figure 1, the identifier of *Zedge* for Google Ads (`a14d2b448c73a08`) is provided in the metadata field for AdMob (owned by Google). An interesting point to note is that even though AdWhirl is not explicitly mentioned in the activity list there is an identifier of *Zedge* (`523e4ae0705248b0b2b770a91d33d1c6`) for AdWhirl. The package name for *Zedge* is `net.zedge.android`. Users can search for an app in the Google Play Store using its package name. Google Play Store provides a lot of information regarding the app such as the developer name, the number of downloads, and the category of the app. We can make use of this information to perform in-depth analysis of the app market from an advertising point of view.

2.2 Dissecting Google Play Store

We downloaded 55K free apps from Google Play Store. These apps were chosen randomly to avoid any bias towards the most popular apps or any particular category of apps. 46K of the apps asked for the `android.permission.INTERNET`



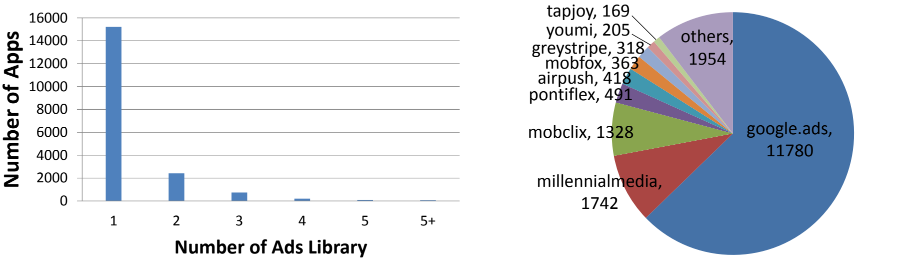
(a) All Apps (b) Apps Containing Ads Library

Fig. 2. Top 10 Categories for Apps

which is needed by any app that needs to access the network. We obtained the category of each app by querying the Play Store. We identified 30 different categories to which the apps belonged. Our analysis showed that the top 10 categories accounted for ≈60% of the apps. Figure 2a shows the distribution of the apps in these top 10 categories.

We picked 30 popular ad libraries on Android platform [8] and generated rules for identifying these libraries from the manifest files. For 19K of these 46K apps we were able to identify the ad libraries that were being used. Figure 3a shows the number of ad libraries used by each app. We can see that a majority of the apps (≈15K) use only 1 ad library and less than 0.3% of the apps use more than 5 ad libraries. Figure 3b shows the most popular ad libraries in these apps. We can see that Google Ads is the most popular ad library as it is embedded in close to 12K apps, followed by Millennial Media (1.7K apps) and Mobclix (1.3K apps). The long tailed nature of the distribution suggests that, in practice, studying any data with respect to the top 50-100 ad libraries would result in high coverage in terms of apps.

We categorized the 19K apps which contained identifiable ad libraries. Figure 2b shows the distribution of the apps in the top 10 categories that we identified above. We see that of the 5.5K apps in the Tools category only 2K contained ads. On the other hand the percentage of Entertainment apps containing ads is much higher (2.6K out of 5.2K). Brain apps (related to puzzles and such) have the highest proportion of apps containing ads (3.2K out of 3.9K). The proportion of apps containing ads in other categories which have similar number of apps in our dataset such as Business, Books and Reference, Travel and Local, News and Magazines, Education, and Casual, shows a large variance. Such information is very useful for new developers looking to pick a category to develop apps in or for ad providers to target development community in any particular category. We can further drill down into the distribution of categories per ad library or popularity of different ads library in a given category. Figure 4a and Figure 5 show the distribution of apps in three of the most popular ad networks in our data set: Google Ads, Mobclix, and Millennial Media. We can see that Google Ads is quite evenly spread amongst various app categories while Millennial Media and



(a) Distribution of Ad Libraries in Apps (b) Number of Apps Per Ad Library

Fig. 3. Ads Library Info

Mobclix ad libraries are very unevenly distributed amongst the categories. The top 2 categories for Mobclix are Entertainment and Casual, while for Millenial Media they are Brain and News and Magazines.

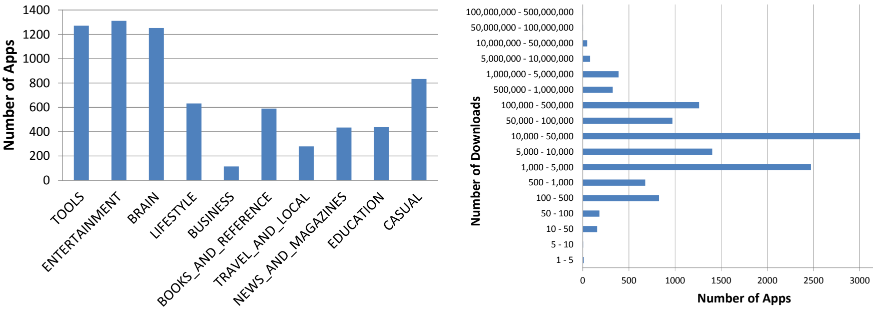
The popularity of an app is commonly measured in terms of the number of downloads of the app. Having the information about the ad libraries in an app allows us to obtain many different perspectives from the downloads data. For instance, for each ad network, we can determine the number of downloads for each app. Figure 4b shows the downloads data for apps containing Google Ads. We can see that the maximum download numbers are for 10K-50K downloads (3K of the 12K apps). We can plot similar graphs for other ad networks or even include app category dimension in these graphs. This information is useful to various entities such as network providers or developers looking to select an ad library.

3 Network Trace Analysis

In this section we present the analysis of real-world network traces from a Tier 2 cellular service provider. We collected the HTTP headers for all users in the network for a week (June 18-25, 2011). Here we present our analysis of the traces from two days in the week - one a weekday (June 21) and the other a weekend (Jun 24). We note that due to company non-disclosure agreements we can not release our dataset/tools. However, this paper contains sufficient details to perform similar analysis on any publicly available trace containing mobile data.

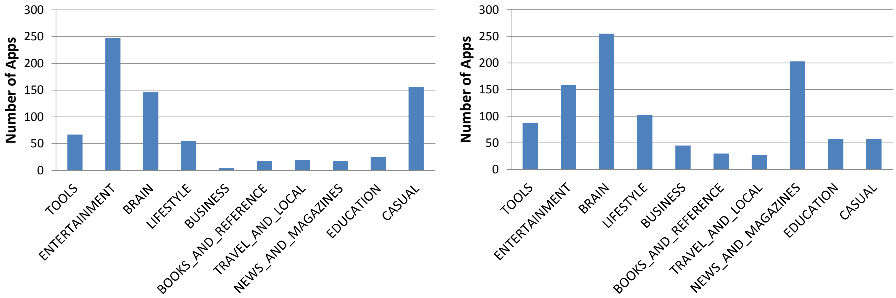
3.1 Methodology

We have developed a system for analyzing Android apps that installs and runs each Android app in a separate emulator running in a virtual machine [13]. Here we describe the parts of the system relevant for collecting ad flows. We can identify an ad flow from the `Host` field in the HTTP header field. We created a database of the host names used by different ad networks as follows. For each ad library we picked a few apps using the library. We used `tcpdump` to collect all the network traffic from the virtual machine. We ported the `strace` utility to Android to log each networking system call performed by the app. We identified all the threads started



(a) Distribution of Categories (b) Number of Downloads Per App

Fig. 4. Google Ads

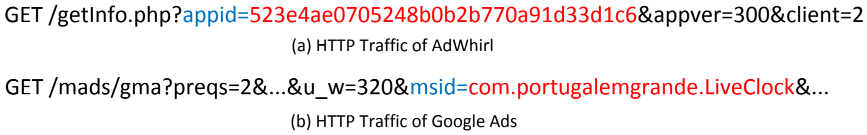


(a) Mobclix (b) Millenial Media

Fig. 5. Distribution of Categories

by the app using the process id (pid) of the app. Based on this thread information, we can filter out the traffic that does not origin from the app. We extracted the host names for the ad library by manually inspecting these traces and identifying the host names that contain parts of the ad library name.

The main challenge in performing any meaningful analysis on real-world traces is to identify the app from the ad flow. As mentioned in Section 2.1, ad networks identify the app using either app name or an identifier that is unique to the app or the developer. It is easy to identify an app from an ad flow that uses app name to identify the app. All we need to know is the key name used in the query. We can do that by running a single app, that contains the given ad library, as explained above, and obtain the key name that is used for the identifier. For instance, for Google Ads flows, the app name is stored in the query parameter with the key `msid`. So we can just look for `msid=` for any flow to Google Ads and the value of the parameter will give the app name such as `net.zedge.android`. Figure 6b shows a Google Ads flow. We can see that the flow belongs to the app with the package name `com.portugalemgrande.LiveClock`. For the ad networks that use unique alphanumeric strings as identifier, the identifiers may be present in the manifest files. We can download all apps from any market, extract the manifest file, and generate a mapping of the identifier for each app for each ad library. Figure 6(a) shows



```
GET /getInfo.php?appid=523e4ae0705248b0b2b770a91d33d1c6&appver=300&client=2
```

(a) HTTP Traffic of AdWhirl

```
GET /mads/gma?preqs=2&...&u_w=320&msid=com.portugalemgrande.LiveClock&...
```

(b) HTTP Traffic of Google Ads

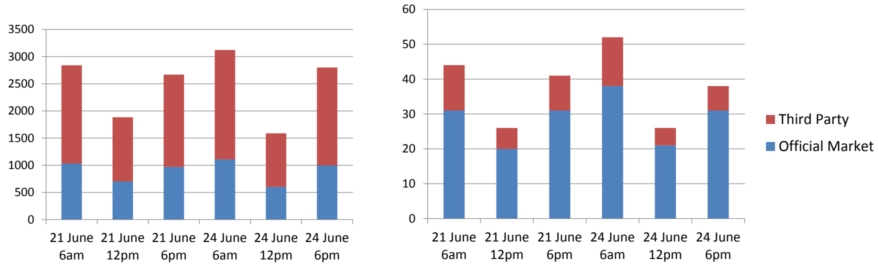
Fig. 6. HTTP Traffic Examples

an AdWhirl flow with the identifier value `523e4ae0705248b0b2b770a91d33d1c6`. Currently we are in the process of building a comprehensive mapping from identifiers to app names for the popular ad networks. However, due to the restrictions imposed by Google on the number of apps that can be downloaded every day, the mapping currently does not cover a large percentage of apps. Hence, we focus our analysis on two popular ad networks (Google Ads and Smaato) that use app names for identifying the apps in the ad flows.

3.2 Dissecting Real World Traces

We analyzed the two days of data to see if the results presented by Xu et al [3] hold in terms of temporal patterns of different categories from an advertising perspective. We broke up each day's data into 1 hour buckets and analyzed the traffic at three different times of the day - (i) 6.00am-7.00am, (ii) 12pm-1pm, and (iii) 6.00pm-7pm. Figure 8 shows the number of apps identified that belong to Google Play Store and the ones from the unofficial third-party markets. We can see that out of the identified apps for Google Ads (Figure 7a), only 35-38% belong to the official Google Play Store. For Smaato, (Figure 7b), we have a much smaller number of identified apps, but the percentage of those apps belonging to Google Play Store is much higher (70-80%). What this seems to indicate is that Google Ads is a popular choice for many of the app developers for the unofficial third-party app markets.

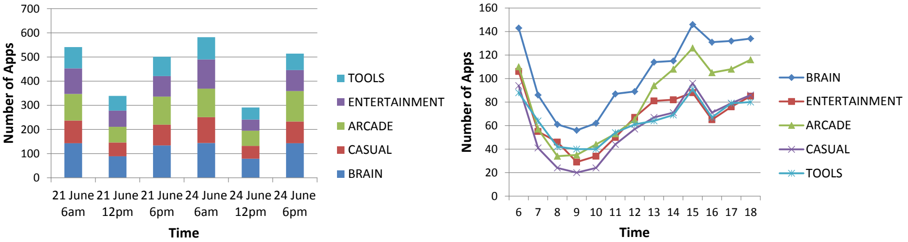
Xu et al [3] had observed some interesting diurnal patterns in different app categories. For example, they report that the weather and news apps are used most frequently in the morning while sports apps peak in the early evening. Similarly, an ad network provider, or a network operator, or a developer is likely to find the patterns of usage of apps containing ads very insightful. Figure 8a shows the top 5 categories of apps present in the traffic at different times for Google Ads. We see that the app usage goes down at noon compared to early morning and early in the evening. This is true for both weekday and weekend. Another interesting observation is that the top 5 categories for apps using Google Ads remains same irrespective of the time of the day or the day of the week. What changes is the proportion of apps being used in one of these categories. For instance, maximum number of Arcade apps are used on a weekend evening. The top category differs for Smaato (Arcade) from Google Ads (Brain) but surprisingly it remains the same over time just as for Google Ads. Figure 8b shows the usage patterns for the same categories over 12 hours on 21st June for Google Ads. Again, we see the number of apps vary through the day but the mix of categories remains more or less same.



(a) Google Ads

(b) Smaato

Fig. 7. Apps Belonging to Official Market in Network Traffic



(a) Patterns Over 2 Days

(b) 12 Hours Pattern

Fig. 8. Apps Containing Google Ads in Network Traffic

4 Limitations and Future Work

Many of the free apps have corresponding paid apps that do not show any ad. These paid apps can not be identified using our ad flow based technique. However, we observe that many flows to third-party platforms like Facebook and analytical services such as Google Analytics also contain identifiers that can be used to identify the apps. We plan to extend our technique to include these flows in the future studies. However, we just like to point out that 73% of the apps in Google Play are free [10].

A limitation of this technique is that some of the ad networks require developer identifiers which can be shared by different apps from the same developer. We have observed that queries from many apps have certain unique patterns (such as certain key-value parameters in the URL query) that can be used to identify them [13]. In the future we plan to analyze patterns in the URL queries in ad flows to form fingerprints that can be used to correctly attribute the flow to the originating app.

Grace et al [8] have observed that many of the ad libraries require user's location for targeted advertising. We confirmed that many of the ad flows contained location information. In future, we plan to use this location information to identify spatial patterns in app usage. Moreover, if the traces contain information about users, then we can build app usage profiles for each user which can be used in applications such as targeted app recommendation.

5 Conclusion

In this paper, we presented a new direction for analyzing usage behavior of mobile apps based on ad flows. We described techniques for associating apps with the ad flows. We showed a flavor of the kinds of analysis possible from app markets and real world mobile network traffic from advertising perspective. We believe that usage pattern analysis from advertising perspective is going to be very important research area in the near future.

References

1. http://www.canalys.com/
2. https://play.google.com/store/apps/
3. Xu, Q., Erman, J., Gerber, A., Mao, Z., Pang, J., Venkataraman, S.: Identifying diverse usage behaviors of smartphone apps. In: Proceedings of the 11th Internet Measurement Conference, IMC (2011)
4. Falaki, H., Lymberopoulos, D., Mahajan, R., Kandula, S., Estrin, D.: A first look at traffic on smartphones. In: Proceedings of the 10th Internet Measurement Conference, IMC (2010)
5. Maier, G., Schneider, F., Feldmann, A.: A First Look at Mobile Hand-Held Device Traffic. In: Krishnamurthy, A., Plattner, B. (eds.) PAM 2010. LNCS, vol. 6032, pp. 161–170. Springer, Heidelberg (2010)
6. Wei, X., Gomez, L., Neamtiu, I., Faloutsos, M.: Profiledroid: Multi-layer profiling of android applications. In: Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, MobiCom (2012)
7. Aioffi, W.M., Mateus, G.R., Almeida, J.M., Mendes, D.S.: Mobile dynamic content distribution networks. In: Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM (2004)
8. Grace, M.C., Zhou, W., Jiang, X., Sadeghi, A.R.: Unsafe exposure analysis of mobile in-app advertisements. In: Proceedings of the 5th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WISEC 2012 (2012)
9. Pearce, P., Felt, A.P., Nunez, G., Wagner, D.: Addroid: Privilege separation for applications and advertisers in android. In: Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, ASIACCS (2012)
10. Leontiadis, I., Efstratiou, C., Picone, M., Mascolo, C.: Don't kill my ads!: Balancing privacy in an ad-supported mobile application market. In: Proceedings of the 13th Workshop on Mobile Computing Systems and Applications, HotMobile (2012)
11. Vallina-Rodriguez, N., Shah, J., Finamore, A., Grunenberger, Y., Papagiannaki, K., Haddadi, H., Crowcroft, J.: Breaking for commercials: Characterizing mobile advertising. In: Proceedings of the 12th Internet Measurement Conference, IMC (2012)
12. http://code.google.com/p/android-apktool/
13. Dai, S., Tongaonkar, A., Wang, X., Nucci, A., Song, D.: Networkprofiler: Towards automatic fingerprinting of android apps. In: Proceedings of the 32nd IEEE International Conference on Computer Communications, INFOCOM (2013)