Capturing Mobile Experience in the Wild: A Tale of Two Apps

Ashish Patro
University of Wisconsin
Madison
patro@cs.wisc.edu

Shravan Rayanchu[*]
Google Inc.
rayanchu@google.com

Michael Griepentrog
University of Wisconsin
Madison
griepent@cs.wisc.edu

Yadi Ma
University of Wisconsin
Madison
yadi@cs.wisc.edu

Suman Banerjee
University of Wisconsin
Madison
suman@cs.wisc.edu

ABSTRACT

We present a long term and large scale study of the experience of mobile users through two popular but contrasting applications in the wild. To conduct this study, we implemented a measurement framework and library, called Insight, which has been deployed on these two applications that are available through Apple's App Store and Google's Android Market. One of them, Parallel Kingdom (PK), is a popular massively multiplayer online role-playing game (MMORPG) which has over a million unique users distributed more than 120 countries. The other application, StudyBlue (SB), is an educational application with over 160,000 unique users. Our study spans most of the life of the PK game (more than 3 years) while our deployment with SB has been running for over a year now. We use Insight to collect diverse information about network behavior, application usage and footprints, platform statistics, user actions, and various factors affecting application revenues.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication; C.2.3 [**Performance of Systems**]: Measurement Techniques

Keywords

Mobile applications; Application usage; Network performance; MMORPG

1. INTRODUCTION

Mobile applications are the talk of the town with hundreds of thousands of applications populating different app stores and billions of downloads of these applications. Once released, these applications get deployed across a large range of devices, users, operating environments, and may cater to different social as well as cultural usage patterns. In this paper, we focus on a measurement study that attempts to understand the experience of these mobile applications once they are deployed in the wild amongst thousands of users.

Understanding the application experience: In the recent years, researchers have adopted various approaches to study the usage of mobile applications. Some studies have instrumented smartphone devices of a controlled set of users to understand application usage patterns (e.g., sessions, network, context) [13, 21, 23]. Other studies have deployed standalone measurement apps for the mobile platforms. For example, 3GTest [15] uses a standalone app that allows users to run network performance tests from their current locations, while AppSensor [10] uses its own app to understand the contextual usage (e.g., time of day, location) on smartphones. The focus of this paper is quite distinct from such prior approaches. We want to understand *the usage pattern of a few popular apps, how it varies with different factors, such as network performance, device type, and application type, and the possible generalizations from our observations.* To meet this goal our measurement approach and infrastructure is quite distinct from prior published approaches.

The user's experience within an app transcends multiple dimensions. In certain cases it depends on the interaction of the network properties with application features. It also depends on the application's design and how different features of the application appeal to each user. Unlike standalone measurement systems that capture various performance parameters only when users activate these processes, our work focuses on application-level experience of users *when the users are interacting with specific applications.*

In order to capture the application experience of users, we have developed a toolkit library, called Insight, and have invited various developers to install this library into their application. Our approach to analyze apps is quite similar to a few commercial analytics solutions [4, 6, 5, 3] that focus on providing business analytics to the users (e.g., number of downloads, geographical locations of users etc.). However, being commercial toolkits, such solutions do not present their findings in a manner similar to this effort. Further, to meet the various goals of our study, Insight goes further than some of the commercial toolkits to correlate network-layer and device-related factors that impact application usage experience of users. For example, Insight performs light-weight network latency measurements during the application sessions. Thus, by tracking in-app purchases made by users and the network quality statistics of sessions in which these purchases were made, developers can infer how user revenue may be impacted by quality across different networks. In this first-of-its-kind academic study, Insight has served as a distributed lens

[*]Shravan Rayanchu worked on this project as a PhD student at UW-Madison.

CoNEXT'13, December 9 - 12, 2013, Santa Barbara, California, USA.
Copyright 2013 ACM 978-1-4503-2101-3/13/12 ...$15.00..

from inside applications with greater than one million users over more than three years.

We have performed a long term deployment of Insight in two popular (commercially available) mobile applications: (i) a multi-player mobile role playing game, called Parallel Kingdom or PK [20] (on Android and iOS), with more than 1 million users across more than 120 countries which has been using Insight for more than 3 years and (ii) a third-party study application for students called StudyBlue or SB [8] which has been using Insight (Android only) for more than 1 year and has more than 160,000 Android users.

The key contributions of this paper can be summarized as follows: (i) We design Insight, a framework that allows mobile application developers to collect application analytics and understand the various aspects about the usage of their applications. (ii) Through the long-term deployment of Insight on 2 applications (a MMORPG and an educational app), we study and contrast the properties of these two apps, (iii) We couple application analytics with different factors (e.g., network performance) to gain insight into various factors affecting applications, e.g. usage and revenue generation[1] Following are some key observations from our data:

Insights from application and device analytics.

- The application usage behavior is impacted by its type and userbase. The daily active users for the educational application which has a specific demographic was highly correlated with the day of the week. Such application-specific usage context can be used by developers to reduce server deployment costs: e.g., by dynamically provisioning cloud based servers based on the time of day/week (§3.1).

- The "usability" and type of a platform impacts the application usage. For example, tablet based users were more interactive (upto 2 $\times$) compared to smartphone users across both applications. Amongst Android devices, phones with slide-out keyboards had more user interactions for the PK game than phones with larger screens (§3.2) [2].

- We observed that even on device models with similar CPU, memory and battery capacity, the battery drain caused by the same mobile application had a high variance (upto 3$\times$). Also, high battery drain caused by the application led to lower session lengths across both PK and SB. This points out the need for device specific optimizations for the application to remain energy efficient across a range of devices. For example, controlling the screen brightness on a Kindle Fire device reduced the average battery drain by 40% while using SB. Furthermore, such optimizations can benefit applications since we observed a decrease in session durations with an increase in battery drain on the devices (§3.3).

- A large fraction of users (48% and 27% for PK and SB respectively) used the applications for a single day only. Also, it is important to retain old users as they generate more daily revenues as well as spend more time within the application (§5). For e.g., PK's regular users generated 2.5$\times$ more daily revenues compared to new users while average session lengths for regular SB users were 1.5$\times$ more compared to newer users (§5).

[1]In our prior short MobileGames workshop paper [19], we studied a few game specific aspects of PK. This paper treats this problem more comprehensively by first describing the release version of our Insight tool, performing a comparative study of two different apps (one being a mobile game and the other an educational app), and also presenting various network layer measurements and analysis.

[2]The game involved text-based and touch-based interactions.

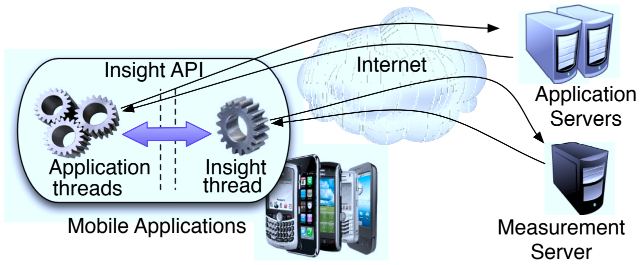

Figure 1: Illustration of the Insight measurement framework.

Insights combining network and application analytics.

- Higher network latencies due to poor network performance reduced user interactivity across both applications. The impact was higher for the PK (upto 40% reduction in interactivity), which requires real-time communication with the servers. Furthermore, poor network performance led to shorter user sessions and loss in application revenues for PK (upto 52% under bad network conditions). Thus, techniques to identify and reduce user-perception of high network latencies (e.g., placing additional servers near regions with high cellular latency) is crucial for such developers (§4).

- Poor cellular network performance led to an increase in WiFi access indicating that network quality affected user's choice of networks. The game users (PK) exhibited a higher preference for WiFi networks (69%) compared to the educational app (58%) under poor cellular performance, indicating that choice of the network type is also impacted by the application type (§4).

The rest of the paper is organized as follows. We begin with a detailed description of the Insight framework and data collection methodology in §2. We then study the application usage and footprint across devices in §3. We correlate network performance with application usage and revenues in §4. we discuss the impact of user retention on user activity and application revenues in §5. We present the related work in §6 and finally conclude the paper in §7.

2. THE Insight FRAMEWORK

In this section, we present an overview of the measurement framework and discuss the applications used in this study on which Insight is currently deployed.

2.1 Measurement Framework

With Insight, our goal is to provide a flexible mechanism to collect various statistics from the mobile application. We achieve this goal by providing Insight to application developers as a library (e.g., `insight.jar` for Android) that can be embedded into their applications. Figure 1 illustrates this overall measurement framework. The application code interfaces with this library using a standard API exposed by Insight. When a user starts the application on a mobile device, it spawns a separate thread that starts collection of *generic data*. The generic data logged consists of the information about application usage (e.g., session lengths), application footprint across devices (e.g., CPU usage) and information about the network performance (e.g., end-to-end latencies). Additionally, Insight's API also enables developers to log *app-specific data i.e.*, that could be of interest to the application under consideration. For example, a game developer might capture the virtual currency transactions within the application sessions.

While Insight's approach provides many advantages, such a design imposes constraints such as: creating an intuitive and simple to

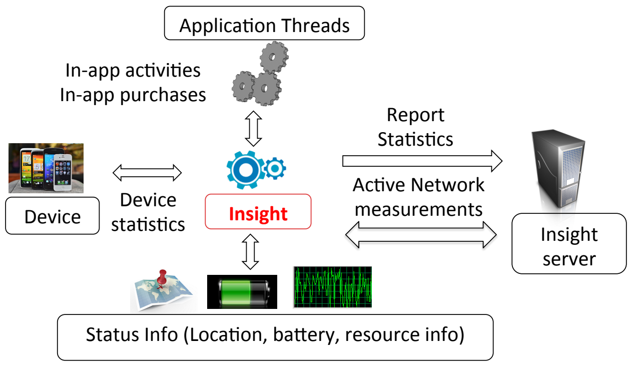

Figure 2: Statistics collected by Insight during an application's usage through passive and active measurements.

use API requiring only small changes to application code, using only "non-intrusive" measurements with minimal application overhead (e.g., we did not have the luxury of performing network bandwidth measurements) and having a flexible measurement granularity.

2.2 Framework details

Insight periodically collects measurements from mobile applications to generate analytics that provide useful information to the application developers (Figure 2). It also performs *light weight* active measurements to understand the network performance. An important aspect of the library is that it *only runs when the application is used in the foreground* and does not perform any background data collection. Insight uses a light-weight measurement library, which consists of around 3000 lines of code. The library is written in Objective-C for iOS and Java for the Android SDK. Insight's measurement server implementation consists of around 2500 lines of C# code. We now elaborate on the data currently logged by Insight. Some of these capabilities have been added over time to enhance Insight's view of the application.

Insight's measurement overhead. Table 1 shows that the processing and memory of Insight is minimal. For example, Insight added < 2% CPU overhead on our test Android devices. Also, Insight's network overhead is small since it uses the network connection to only perform coarse grained network measurements. We now elaborate on the data logged by Insight.

Generic data logged by Insight. We now describe the generic data logged by the current implementation of Insight. It is important to note that developers have the flexibility to selectively enable or disable any of the following measurements. So, it is upto the developers to determine the set of metrics that they wish to capture.

1. Session Data. The duration of an application's usage in the foreground is denoted as a session. A session ends if the application is closed or pushed to the background.

2. Device Information. Insight also records the device related information (e.g., model, platform, screen brightness, available network interfaces, signal strengths) per session.

3. Network Data. Insight collects periodic network performance data by performing light-weight measurements using both TCP and UDP. When a user starts the application, Insight initiates a TCP connection with our measurement server (co-located with the actual application servers). The server sends probes to the applications at a configurable interval using TCP and UDP packets with increasing sequence numbers. These probes elicit immediate responses from the application allowing computation of both UDP and TCP based network latencies or Round Trip Times (RTTs). These RTTs allow us the measure the quality of the network link (e.g., cellular or WiFi) between the server and user device. A high value of network latency (or RTTs) indicates the presence of a poor network link. We chose

Metrics	HTC Dream	HTC Evo 4G
Avg. CPU usage	$< 1.5\%$	$< 1\%$
Avg. Memory usage	$<$ 100 KB	$<$ 100 KB
Avg. network usage	$<$ 5 KB per min.	$<$ 5 KB per min.

Table 1: Insight overhead measured on 2 Android devices showing the minimal resource and network overhead added by the library. Insight runs only when the application is in the foreground.

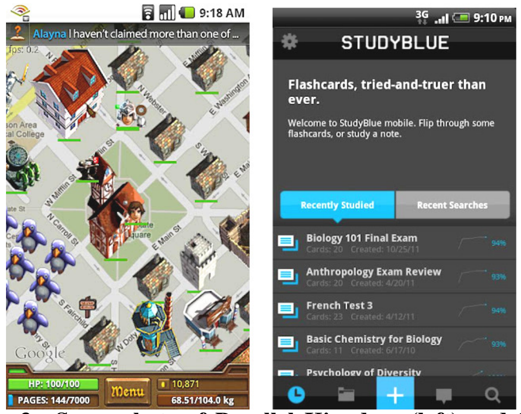

Figure 3: Screenshots of Parallel Kingdom (left) and StudyBlue (right), the third party applications currently using the Insight framework.

a 30 second time-interval as it provides reasonable measurement accuracy while minimally affecting a mobile phone's resources [17].

4. Location Information. While the app has access to precise user location through GPS/WiFi, for our study we anonymize this information into coarse-grained blocks (100 sq. km. regions, state level, or country level).

5. Application footprint on device. Insight also measures the resource overhead of the application on user devices by capturing a set of metrics. These metrics include the CPU and memory utilization by the application as well as the "battery drain rate" on the device while the application is in the foreground. The battery drain rate is calculated per session (while the application is in foreground) by dividing the change in battery level during a session with its duration. For example, if the battery level changes from 80% to 75% during a session length of 20 minutes, the battery drain rate is (80 -75)/ 20 levels/minute, i.e., 0.25 levels/minute. To make this analysis more accurate, we use some filtering as described in §3.3. The overhead of this measurement is minimal on the device as well as the application as it is passive and performed infrequently (every 30 seconds by default). Currently, we have implemented this capability only for Android.

App-specific data logged by Insight. In addition to the above generic data, Insight also provides an API to the applications to log any additional data. The application code calls this API periodically to record app-specific statistics. This data is later combined with the other statistics (e.g., device, network performance) collected by Insight to study their correlation with the in-application activity.

2.3 Test applications

We have deployed Insight within two third party applications which are available for free in Apple's AppStore (iOS platform), and in Google's Android Market (Android platform). Figure 3 shows a screenshot of these applications.

Application #1 (MMORPG): Parallel Kingdom (PK). Parallel Kingdom [20] is a Massively Multiplayer Online Role Playing Game

	PK (game)	SB (education)
Unique users	$> 1,000,000$	$> 160,000$
Sessions	$>$ 61 million	$>$ 1.1 million
Countries	> 120	> 25
RTT measurements	$>$ 350 million	$>$ 25 million

Table 2: Application stats tracked using the Insight framework. The data collection process for PK started on 31 October 2008 while it started for SB on 29 April 2012.

(MMORPG) that places the character (user in the virtual gaming world) on the game map according to their real world location. The game uses GPS/WiFi capabilities of the mobile device for tracking the real world user location. In the game, each player picks up, trades or upgrades various items, fights monsters and interacts with other players on the map. As the game progresses, a player learns various skills, claims territories and obtains unique capabilities. Players can spend real money to buy the virtual currency in the game (”Food”). The game has added numerous features over time through the releases of new “Ages” and minor updates. The game (starting with Age 1) was officially launched on October 31, 2008 and has since then become a popular game on both the Android and iOS platforms. Since the First Age, there have been many other minor releases and three major releases (Age 2, Age 3 and Age 4). The game has over 1 million players worldwide, and was ranked number one amongst Android applications in its category in Dec 2009.

PK specific data logged by Insight: In addition to the generic data, PK uses Insight to log the following:

1. Expenditure of Money. ”Food” is a special resource in the game, which serves as the virtual currency. Players spend (or burn) Food to buy items from merchants, upgrade buildings etc. Players can purchase food with real money to expedite progress in the game. Thus, tracking food consumption is important for PK developers as it is their main source of game revenues.

2. Action data. Game related activities such as gathering resources, fighting other players and monsters, trading items were also logged for each session.

Application #2 (Study Tool): StudyBlue (SB). StudyBlue [8] is a smartphone/handheld device based digital study tool for students. SB acts as a platform and enables students to create, study and share flash cards. SB also provides features such as online quizzes, study guides etc. We integrated and launched Insight with SB’s Android application on 29th April 2012. Currently, SB has thousands of daily active users who are mostly based in the US ($> 90\%$).

SB specific data logged by Insight: SB uses Insight to log the following application specific events:

1. Study Events. Study events include activities within the application such as taking quizzes, reviewing study material etc. They are related to consumption of content by users.

2. Edit Events. Edit events include creating new flash cards, updating/editing existing "decks" of flash cards. They are related to content creation events within the app.

2.4 Analyzed datasets

Using Insight, we have collected data for PK over a period of more than 3 years during which the game received 3 major upgrades and many minor updates. Compared to this data set, the data from the StudyBlue application (Android only) is smaller in size, since we have been running Insight with the application for just over a period of 1 year. We use these two datasets to provide some application specific comparisons between an MMORPG and an educational app. Table 2 presents a summary of the tracked statistics.

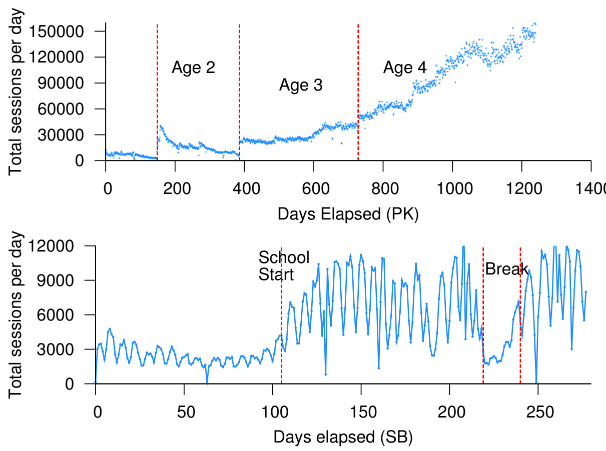

Figure 4: Number of sessions/day for PK (MMORPG game) and SB (study tool).

2.5 Privacy implications

To conduct this study, we had a special arrangement with our partnering software companies. In each case, we developed and provided the Insight library to the companies, and they integrated our software into their application. All collected data was always resident in the servers of the individual companies. Through a special NDA in each case, we were provided with an anonymized feed of the various data we collected. Among data that we present in this paper, the only potentially sensitive data is user location. For the purpose of this project, we anonymized this information by mapping them to a coarse-grained block — a 100 sq. km. area, state level, or country level, as was adequate for the analysis presented in this paper. An accurate latitude-longitude level accuracy was not necessary. Further, our research team never directly interfaced with the users or their data, and instead were limited by special arrangements we negotiated with the two companies.

3. APPLICATION ANALYTICS OBTAINED WITH Insight

In this section, we study some trends related to the usage of the PK and SB apps as well as the impact of device-related factors on the usage of both applications. The observations presented in this section show how Insight helps developers understand various aspects of their application by capturing a set of key metrics.

3.1 Analyzing user population

We now study the properties of the user base and the application access patterns across the two applications.

What are the long term trends in daily active usage?

Figure 4 shows the number of daily sessions for PK (top) and SB (bottom) since the start of data collection for both these applications. The graphs shows different trends for sessions per day across the two applications. Both applications observe thousands of sessions per day although it is much higher of PK due to its wider user base.

For PK, the number is either steadily increasing or decreasing for most of the time except for a few days when there are sudden spikes. The sudden spike in days 149, 387, 729 correspond to the major releases of the game (called Age 2, Age 3 and Age 4 respectively). Other big spikes in the number of daily active users occurred around June 28 2010 (day 603) when the game was released on the iPad and iOS 4 platform and on day 886 due to a major update to the game. The release of new releases and features positively impacted the number of daily active users of the game. Also, starting from Age 3, there is more consistent increase in the game’s daily active usage.

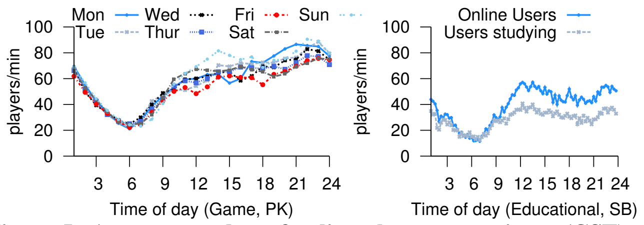

Figure 5: Average number of online players per minute (CST) across 24-hours for PK (left) and SB (right).

Some important factors causing this behavior are – more frequent improvements in gameplay through updates, increase in the game's popularity in its category over time and increase in developers' efforts to advertise the game to attract new players.

For SB, the number of daily sessions is more tightly correlated with the time of week as well as the time of year. For example, the number of daily sessions for the application spike during the weekdays (Monday - Thursday) and the troughs correspond to Fridays and Saturdays of each week. Also, the number of daily sessions for the application was much lower during the summer (days 20 -104) and winter (days 224 - 240) breaks. For SB, this behavior happens due to the nature (study tool) and specific demographic user base (students) of the application. This behavior is not observed for PK due to its more diverse user-base and generic nature of the application (game). *Application such as SB, with highly variable but predictable usage trends can reduce costs by dynamically provisioning cloud-based servers based on the time of the day/week.* For example, the cost per Amazon EC2 cloud instance [1] halves by going down each level (e.g., from "Medium" to "Small" instance).

When do the users access these applications? How is this behavior different across applications?

Figure 5 (left) and Figure 5 (right) show the number of online based players (from US Central time zone) in a 24 hour period for PK and SB respectively. It shows that the peak number of online users occur at different times for the two applications. For example, We find that the number of PK players reaches a maximum during late nights. But the number of online SB users peak between 1 -2 PM and then falls during the evening before rising again between 10 PM - 12 AM. The most likely reason for this behavior is that users prefer to use these applications at different times of the day. For example, SB's peak usage is in the afternoon and nights when students study more. The graph also shows the number of actual users actually studying while using the application. Almost all online users at early mornings (3 - 6 AM) appear to be burning the midnight oil for studying! At other times, its users also perform other activities such as sending messages, creating flash cards, searching data etc. We observe PK gets used the most at nights (9 PM - 11 PM). Further, for PK we also find higher population on Sundays, and a lower population on Fridays.

We find that for both the applications, the number of online users are at a minimum around 6 AM in the morning. This diurnal pattern was consistent across different countries. Application developers can use this insight for several purposes. For example, early mornings (around 6 AM) are favorable to schedule server downtimes and upgrades, install bug fixes or push updates, as they have smallest number of online users across the whole day. For applications such as SB, the time of week information can also be used to schedule such server upgrades and downtimes due to its strong correlation with application usage.

Summary. Across the two applications (PK and SB), we observed differences in the longitudinal and diurnal usage due to their different nature and user demographics. Such application specific behavior can be leveraged by application developers in optimizing costs related to their server deployments as well as creating incentives (e.g through updates) to attract new users and retain the existing ones.

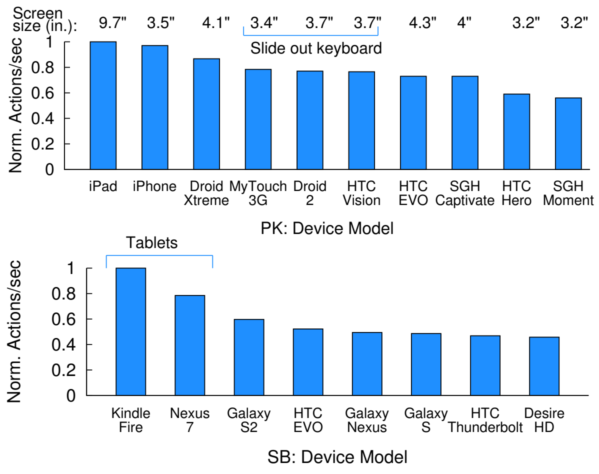

Figure 6: Normalized number of actions per unit time (maximum of 1) for different tablets phone models for PK (top) and SB (bottom). Usability in terms device type, screen size and slide out keyboards are shown.

3.2 Understanding application usage

We now study how different context (e.g., device, time of day etc.) impacted the application access properties across PK (game) and SB (study tool) as well as the differences in the session properties across these different types of applications.

How does device usability (platform, display screen size, availability of slide out keyboards) affect user interaction?

Both applications (PK and SB) are used across a wide variety of smartphone and tablet devices and employ very similar UI for both the tablet and smartphone platforms. We analyzed how "usability" of different platforms may affect the application usage. In particular, we analyzed the effect of (i) size of the display screen and (ii) availability of slide out QWERTY keyboards on "user interactivity". We analyzed the effect on "user interaction" by measuring the average number of user actions/sec on popular device models over a period of one week. For both applications, we chose the top smartphone and tablet devices at the time having diverse form factors (screen sizes and slide out keyboards).

Figure 6 shows some interesting results. We find that tablet device (iPad, Kindle Fire, Nexus 7) users performed the highest number of actions/sec, on both applications owing to a larger screen size (e.g., 9.7" for iPad and 7" for both Kindle Fire and Nexus 7) and both applications involve a large number of touch events. For example, in PK users perform touch related activities such as exploring the map, moving the character and attacking monsters. Players can visualize a larger area on the map on tablets compared to smartphones and interact with more objects simultaneously on the map. On SB, users take quizzes which mainly involve touch events. For PK, we also find that the iPhone comes a close second (despite a screen size of 3.5"), indicating the impact of device and user interface differences between iOS and Android.

Amongst Android device models using PK, we find an interesting trend — devices with smaller screen sizes (e.g., 3.2" for Samsung Moment) experienced lower user interactions compared to those with a larger screen size (e.g., 4.3" for HTC EVO). However, de-

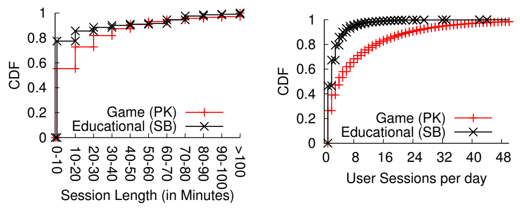

Figure 7: (left) Distribution of session lengths, (right) CDF of the number of sessions played by a user per day.

vices with slide out keyboards (e.g., Droid 2) exhibit higher user interaction compared to some of the devices with larger screen size (e.g., EVO), despite small screen sizes (3.4" – 3.7"). These Android devices have similar capabilities in terms of CPU and memory. This is due to higher usage of keyboard based activities (e.g., in-game chatting/messaging) on these devices. This shows that a device's form-factor and ease of use can impact application usage behavior. *Overall, the average user interactions varied by around 2× across devices for both applications.* Tracking such user interactivity patterns across different devices in the wild can help developers understand the combined impact of the UI and device on the usage of different features within application (e.g., impact of keyboard based devices on text-based activities). This is a useful input for developers to make UI/feature modifications (e.g., layouts, button sizes etc).

How did the session properties differ across PK and SB?

Figure 7 (left) shows the duration of the session lengths in the game. We find that a large fraction of sessions on both applications are short lived. For example, 55% and 76% sessions are less than 10 minutes long on PK and SB respectively. Further, only 9% of sessions are longer than 60 mins on PK and SB. Figure 7 (right) shows the CDF of the number of daily sessions per user. Around 26% of PK users play a single session per day while 27% of its users play more than 10 sessions. SB has fewer user sessions per day with 46% of the users accessing having a single session per day and only 5% of the players accessing more than 10 sessions per day. MMORPGs such as PK tend to have more sessions per day as it common for players to access the game multiple times per day. For example, it is common for PK users to login, play for a short while (e.g., feed the dogs) and logoff.

Figure 8 shows the distribution of the time gaps between consecutive user game sessions per day (the difference between start time of a session and end time of the previous session for the same user). *It is interesting to note that 47% of new PK user sessions tend to be within 5 minutes of the end of the previous session.* In case of SB, only 14% of the new user sessions are within 5 minutes of the previous one.

These short and closely spaced multiple sessions are partly characteristic of usage of smartphones as has been observed before [13] and is observed for both PK and SB. It is normal user behavior to get distracted [16] and close the current application to use another application such as an email client or stop the application to attend a phone call and then return back to the game[3]. The higher proportion of short gaps between consecutive sessions for PK compared to SB (47% vs 14%) is partly game specific as observed in desktop based MMORPGs [11]. Application developers can implement optimizations based on such user behavior. For example, in the case of PK, instead of discarding a player's state (maps, locations, inventory information etc.) from the phone at the end of a session, it

[3]Since Insight is active only while the application is in foreground, we did not use it to log the activities on the phone in between the application sessions to find the actual cause for each gap.

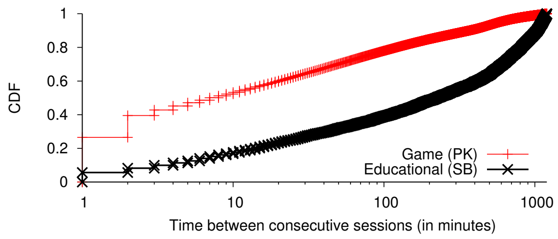

Figure 8: Distribution of time periods between consecutive user sessions per day. The X-axis is shown in logscale.

Characteristic	% of sessions
High CPU or memory usage	$< 1\%$
High network latency or RTT ($> 1sec$)	7.8%
Network type change (WiFi, cellular)	1.9%

Table 3: Observed characteristics for PK sessions that can cause the user to stop and restart the game. This analysis is only for PK sessions which have another user session succeeding it within 5 minutes. A session can fall into multiple categories.

is more efficient to save and reuse it during the next session because almost half of the new user sessions occur within 5 minutes of the previous session. Over time, PK's developers have *improved such caching capabilities to decrease network communication* with the server at the start of each session as well as *improve the start-up time*.

What was the impact of factors that can force users to restart a game session?

Figure 8 showed that 47% of new PK user sessions were within 5 minutes of the end of the previous session. Some of these cases could have been caused due to pathological cases (bad network, high CPU utilization etc.) or availability of another (possibly better) cellular or WiFi network. Such cases can *force* a user to close his previous session and start a new session. Using Insight, developers can keep track of the percentage of sessions exhibiting such pathological behavior and can take actions if such cases become prevalent.

Table 3 shows some of the potential causes that may have caused the user to restart the game and the percentage of sessions showing a particular cause. Around 1% of the sessions had high memory or CPU usage (90-100% average CPU utilization or memory leaks characterized by abnormally high memory usage). 7.8% of the sessions had a high average network latency or RTT ($> 1sec$, §2.2). In 1.9% of the cases, the user changed the network (e.g., changed the WiFi network or changed the connection from cellular to WiFi or vice versa). This breaks the TCP connection with the server and restarts the session. Overall, a small percentage of these sessions were impacted by potential causes that can disrupt a session. This indicates the majority of these consecutive sessions can be attributed to user behavior as discussed earlier. In §3.3 and §4, we further discuss how high battery drain and poor network performance, respectively, impacted user experience and session lengths.

How did the time of day context impact the application usage behavior (e.g., network type, battery usage etc.)?

Users' application usage behavior (e.g., choosing the network type) can also be impacted by the context (independent of the application or device), such the as the location of the user, time of day etc. We analyzed how the "time of day" context impacted the user behavior for 2 different properties: the type of network used and fraction of active users charging their device during application usage (Figure 9). The figure shows that a *larger fraction of US based users accessed the applications over cellular networks during the day compared to late nights* (e.g., 45% at midnight and 63% at 10am). Further, 31%

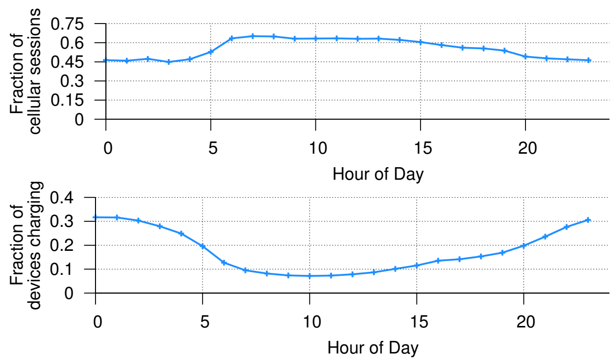

Figure 9: For the US based users, the graph shows the fraction of total users using the cellular network (top) and charging their device (bottom) while using the application over the day (normalized to CST).

of the users accessed the applications while charging the device during late nights compared to only 7% of the users around noon. Due to these diurnal properties, using the time-of-day context can also be beneficial in the use of network and battery [27] related optimizations by mobile applications.

Summary. We observed a 2× variability in user interactivity across different devices for both applications. Such analytics is important for understanding the impact of device and the application UI on user behavior. Further, high fragmentation of mobile devices in terms of OS, hardware and device factors makes it challenging for application developers to deliver a consistent user experience across these devices. For example, developers of popular applications [12] need to test their applications extensively across several models before releasing new updates. Tools that assist developers to debug issues related to device fragmentation can be very helpful.

We observed short but multiple sessions for both PK and SB users which can partly be attributed to smartphone usage behavior. In the case PK, the occurrences of closely spaced consecutive sessions led to efforts by developers to reduce the application start-up time for improving the user experience. We also observed a diurnal variation related to network (1.5 ×)and battery usage (4 ×) in our user population, indicating the utility of the time-of-day context information in optimizing network and battery usage by mobile devices and applications.

3.3 Measuring application footprint

Once a mobile application gets deployed in the wild, it is very difficult to track its performance on the diverse types of mobile devices available [9]. We used Insight to monitor the application's overhead in terms of battery usage, CPU consumption and memory usage (§2.2) while it was in operation. Knowledge about application footprint across different device types can allow developers to understand which device types to target and optimize their application for. To make the battery drain analysis more accurate, we filtered the data to remove sessions that had multiple running processes (e.g., background downloads) while the game was in operation. This was done to make sure that the battery consumption reflected closely its usage by the application. We also removed those sessions from this analysis during which the user was charging the phone for any portion of the session.

How does the battery consumption due to the game vary across different devices?

Figure 10 shows the distribution of normalized battery drain rates (levels/min) measured on the popular devices used by PK players. We make the following observations: (i) Across different devices,

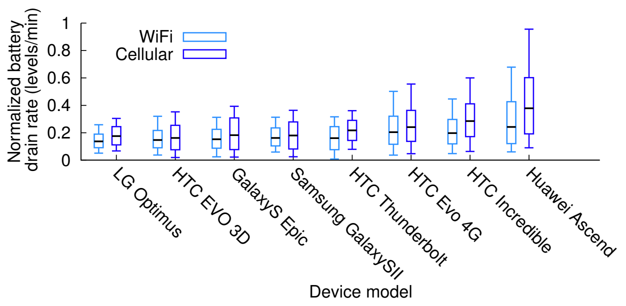

Figure 10: Overall normalized battery drain rates (w.r.t. max. drain rate) for popular Android devices over cellular and WiFi (PK). The candlesticks show the 10th, 25th, 50th, 75th and 90th percentiles.

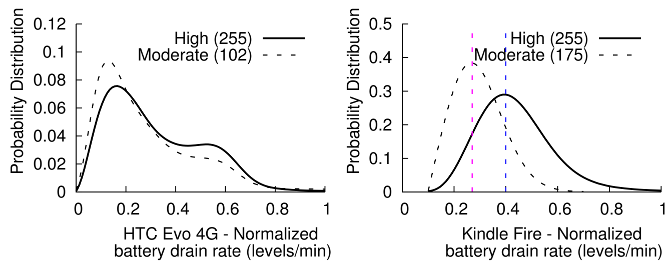

Figure 11: Density plot showing the normalized battery drain rates (w.r.t. max. drain rate) at two brightness levels for HTC Evo 4G (left) and Kindle Fire (left) while running SB in the foreground.

the variation in drain rates is caused due to use of different components (e.g., screen, network card) even if they have the same battery capacity. For example, the devices LG Optimus, Samsung GalaxyS Epic and HTC Evo 4G have similar memory, CPU, network capabilities and the same battery capacity of 1500 mAh but have different drain rates. Amongst the major causes are different screen sizes and technologies: 3.2"(TFT) for Optimus, 4"(AMOLED) for GalaxyS Epic and 4.3"(TFT) for Evo 4G. (ii) On each device, the variation in drain rates is due to factors such as: different screen brightness, battery age (old vs new battery), variable cellular or WiFi network usage etc. (iii) The battery drain rates are higher on cellular sessions vs WiFi sessions. This is because the cellular connection consumes more energy per bit compared to WiFi [18].

These *diverse resource consumption characteristics across different models (more than 3×)* points to the difficulty by application developers in optimizing the usage of the phone's resources on different device types due to the high diversity [12]. Insight can help developers to track this behavior and identify potential performance issues across devices in the wild. Further, device specific optimizations (e.g., reducing GPS usage or automatically reducing the screen brightness on phones with lower battery levels) can considerably improve the energy efficiency of mobile applications across different devices. But, the gains from such optimizations can vary from device by device basis as discussed next.

Figure 11 shows the distribution of battery drain rates observed at two different screen brightness levels (high, moderate) aggregated across thousands of SB sessions (application in foreground) on HTC Evo 4G and Kindle Fire devices. For the Evo device, the battery drain distributions are similar at both brightness levels. It indicates that controlling the screen brightness on this device may not provide much gains in overall battery savings. This is because factors other than screen brightness that have a higher impact on the overall battery drain. *But, for the Kindle Fire device, the mode of the battery*

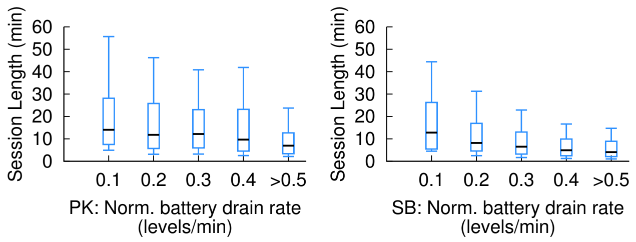

Figure 12: Distribution of session lengths as a function of the normalized battery drain rate (bins of 0.1) for cellular based users on HTC Evo 4G for PK (left) and SB (right).

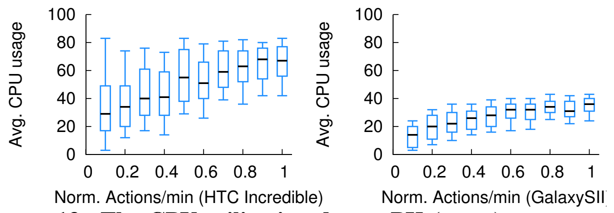

Figure 13: The CPU utilization due to PK (game) across two different devices as a function of the normalized rate of actions performed per minute (bins of 0.1).

drain distribution goes down by 40% (0.4 to 0.27) by reducing the screen brightness from "High" (255) to "Moderate" (175). Thus, for the Kindle Fire device, controlling the screen brightness through the application (e.g., by using Android APIs for brightness control) can be an effective way to reduce the battery drain when the battery levels are low. Power-saving techniques such as [25] can also be used to dynamically reduce the brightness of specific screen areas that are not important for the user. We now discuss how battery drain impacted the session lengths for the applications.

How were session lengths impacted due to the battery drain caused by the application?

High battery drain caused by an application can cause users to reduce the session lengths for conserving the battery. To understand the impact of the battery drain rate caused by the application on the player behavior, Figure 12 shows the distribution of session lengths for HTC Evo 4G (a popular device amongst PK and SB users) as function of the normalized battery drain rates. For both applications, *the session lengths decreased with the increase in battery drain (more pronounced for longer sessions).* For example the 75th percentile session lengths for PK are 29 minutes and 12 minutes for normalized battery drain rates (levels/min) of 0.1 and > 0.5 respectively. The same number goes down from 26 minutes to 9 minutes for SB sessions. The variation in median session lengths is much smaller as the battery drain rates increase, indicating that players with shorter session lengths are less sensitive to the battery drain.

How did user interactivity in PK impact resource utilization on the phones?

Amongst the two applications using Insight, PK (game) was a more resource intensive application due to the higher computational and network requirements. Figure 13 shows how increasing user interactivity causes the PK's CPU usage to increase. For example, user actions such as moving the character causes the game to render the screen again for the new location. Higher CPU usage increases the battery usage on the device. The game's median CPU usage goes up from 29% to 67% and 14% to 36% for HTC Incredible and Samsung GalaxySII respectively, when the normalized actions/min go up from 0.1 to 1. Thus, the graph shows that the resource consumption

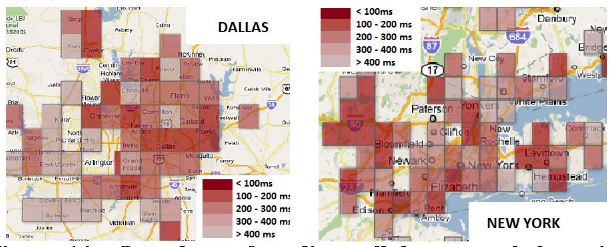

Figure 14: Snapshots of median cellular network latencies (RTTs) from two metropolitan regions (Dallas, New York). The RTTs were aggregated into 10km * 10km sized cells and median RTT is shown from each cell.

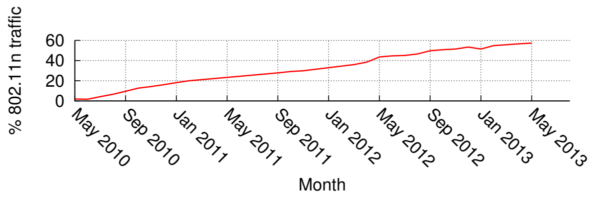

Figure 16: 802.11n usage over time from Android based devices.

for similar workloads (actions/min.) varies across different phones due to differences in hardware capabilities.

Thus, tracking resource consumption in the wild for resource intensive mobile applications (e.g., games) is important for developers to understand the application's footprint across all client devices. It can also be helpful to identify pathological issues such as excessive CPU utilization, memory leaks etc. that has the potential to cause poor user experience (e.g., due to unresponsiveness of the application in these cases).

Summary. Factors such as a network type, user behavior and device type cause widely varying battery usage across different application sessions. We also observed a drop in average session durations with increasing battery consumption for PK, which is a more resource intensive gaming application. Thus, techniques to optimize battery usage by applications can be beneficial. As shown in this section, the benefits of specific techniques to reduce total battery usage (e.g., controlling screen brightness) can vary across different device types. Thus, a combination of different techniques is required to optimize battery usage across mobile devices.

4. USING Insight TO INFER THE IMPACT OF NETWORK ON APPLICATIONS

Amongst the two applications, PK (an MMORPG) uses network connection (cellular or WiFi) more heavily for real-time communication with the game server. For example, an attack action on a monster needs to be pushed to the server and a response needs to be received by the client. On the other hand, SB requires more intermittent connectivity to the server to download content (e.g., flash cards) or synchronize local updates with the server. As discussed in framework details (§2.2), Insight uses network latency (or RTTs) as an indicator of network performance. Poor network performance due to causes such as an ISP's network coverage, slower cellular networks (e.g., EDGE, 1xRTT) [14] or poor signal strengths at a user location's can increase the network latencies (RTTs) for a user.

To understand how network performance impacted application usage and revenues, the discussion in this section leverages the network data collected from the apps over a diverse range of networks, devices and locations. Figure 14 shows the spatial distribution of cellular network latencies (median per cell, cell size was 10km × 10km) in two metropolitan locations (Dallas and New York) averaged

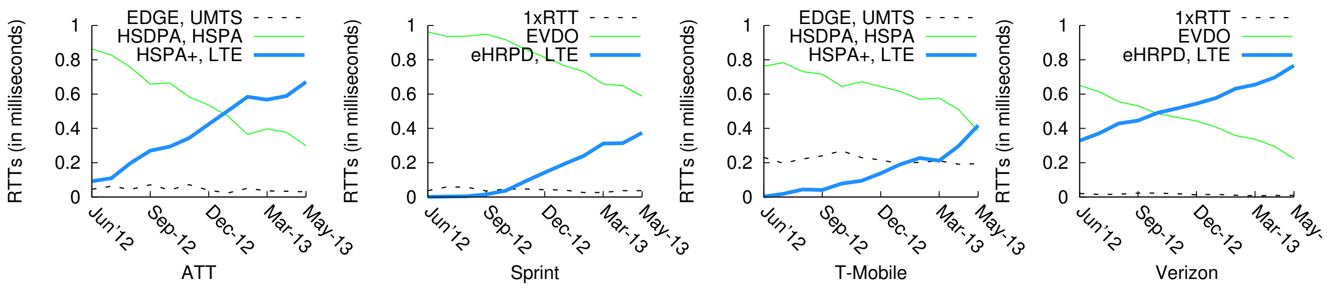

Figure 15: Distribution of network traffic from different technology types for cellular carriers from June 2012 to May 2013.

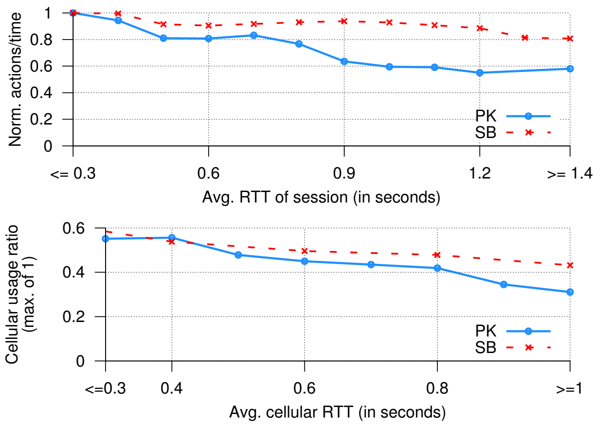

Figure 17: (top) Normalized values for actions performed per unit time at different RTTs for US based user of PK and SB (Bin size of 100 ms). (bottom) Percentage time spent on cellular networks (vs. WiFi networks) by US based PK and SB users as a function of average cellular RTTs.

over a month. It shows a considerable difference in cellular network performance within the same metropolitan region. For example, even within New York, we observe that the median RTT can vary from 187 ms to 312 ms indicating uneven cellular performance within a small geographic location.

Usage of latest cellular and WiFi technologies over time. Figure 15 shows the distribution of network measurements coming from different technologies (e.g., EV-DO, HSDPA) for the major US cellular carriers over a period of 12 months (June 2012 - May 2013). Over this period, the proportion of network traffic from faster networks (e.g., LTE, HSPA+ eHRPD) is increasing for users across all ISPs. For example, amongst the Sprint users analyzed through Insight, the usage of newer cellular technologies (eHRPD and LTE) increased to 40% by May 2013. But, there was a small and consistent fraction of users across different carriers that accessed the PK and SB over higher latency cellular connections (e.g., EDGE). Such users might be more sensitive to network performance even during un-congested network conditions. In the next section, we study the impact of network latencies on user behavior across PK and SB applications.

Using Insight, we were also able to track the adoption of 802.11n over time. Figure 16 shows that from May 2010 to May 2013, the proportion of WiFi traffic using 802.11n capable devices increased from 2% to 57.2%. This indicates that adoption of newer WiFi standards such as 802.11n is increasing gradually over time with a large proportion of WiFi deployments and mobile devices (around 42.8%) still using the older 802.11bg standards. Thus, mobile application developers should be aware of the presence of these older devices and networks, even though newer WiFi standards such as 802.11ac are getting introduced.

	US	UK	GER	JPN	CAN
Cellular	52.7%	24.7%	31.2%	30.5%	23.1%
WiFi	47.3%	75.3%	68.8%	69.5%	66.9%

Table 4: Fraction of player sessions accessing PK through cellular and WiFi networks for the top 5 countries.

4.1 Impact of network performance on user behavior

We now discuss how Insight can help understand whether network conditions had an impact on application usage.

How were user interactions impacted by poor network performance?

Figure 17 (top) shows the normalized number of user actions per minute (max. of 1) as a function of the average network latency experienced during a session. We find that user interactivity (actions/minutes) declined with an increase in network latencies for both applications. But, the decline in user interactions is much more pronounced for PK. *For example, an increase of network latency (RTTs) from 300ms to 900ms caused average user interactivity to drop by 40%.* The same value for SB was only 7% indicating the much lower impact of the network latency on its users. But, at higher network latencies ($>=$ 1.4 seconds) the impact increased on SB users as indicated by a 20% drop in interactivity.

As mentioned earlier, network performance plays a crucial role in MMORPGs like PK because a typical user session comprises many real-time interactions with other users as well as the game server (e.g., moving on a map, attacking monsters etc.). This value may vary for other real-time games or applications, but the results highlight how user perception of poor network performance can vary based on the nature of the application.

How do network latencies influence user's choice of networks (cellular vs. WiFi)?

WiFi based network latencies tend to be an order of magnitude lower compared to cellular network latencies at most locations [24]. Also, cellular latencies can vary widely based on the type of cellular technologies used (e.g., LTE, EDGE, HSDPA etc.) [14]. We had earlier discussed the reduction in user interactivity during periods of high network latency (more prominent for PK). User perception of higher network latencies (e.g., higher server response times for an action) can be frustrating and degrade their experience. We used Insight network measurements to study the impact of cellular network latencies on their choice of network type (cellular vs. WiFi) to access the two applications.

Figure 17 (bottom) shows the average proportion of time spent by PK and SB users on cellular networks while using these apps as a function of their average cellular network latencies. We find that as cellular network latencies increase, users on both applications spend more time accessing the game through WiFi networks (higher for PK users). For example, PK and SB users who experienced low network latencies (0.3 seconds) spent around 55 - 58% of their game

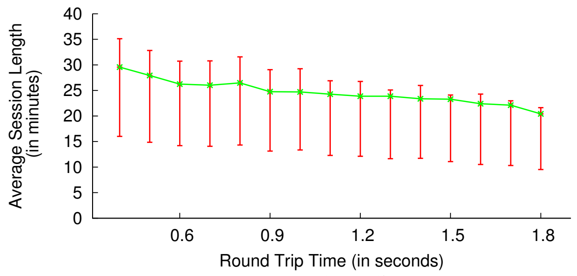

Figure 18: Average session lengths for US based cellular players at different RTTs (Bin size of 100ms). The bars show the 25th - 75th percentile values of the session lengths.

time on cellular networks. *Amongst users that experienced poor cellular network performance (cellular latencies $>$ 1.0 sec.), PK users spent only 31% of their time on cellular networks* as opposed to 42% of SB users. This observation indicates that the application's nature (e.g., an MMORPG game) were amongst the important factors that influenced the users' choice of networks used while using their application.

Table 4 shows the distribution of user sessions accessing PK using cellular and WiFi networks amongst top five countries (in terms of total user sessions). Except for USA, the distribution is uneven across other 5 countries with more players accessing the game through WiFi. While this behavior can be affected by many regional factors (e.g. cost of cellular plans, coverage of 3G networks), the observations from Figure 17 (bottom) shows that network performance is amongst the important factors affecting this behavior. This is due to the location of PK game servers within the U.S., which leads to higher network latencies for users from other countries.

The analysis in this section showed that across these two applications using Insight, *PK users exhibited higher sensitivity to higher network latencies as exhibited by lower user interactivity and greater preference for WiFi networks under poor network performance*. While developers may have an estimate of the impact of such network related factors on users' experience, its difficult for them to quantify the impact of such factors in the wild. Crowd-sourcing or sampling such information from users during their sessions can help expose these application-specific dependencies between network performance and user experience. Such applications or games with multiple server locations can benefit from frameworks [17] that use periodic estimation of network latencies within the application to optimize the selection of servers and other players to reduce network latencies experienced by users. For developers with more limited resources, these measurements can guide the placement of additional servers (e.g., choosing Amazon EC2 instance locations) near regions that experience high network latencies.

Summary. An increase in network latencies impacted user behavior, especially for PK (an MMORPG), as indicated by a decrease in user interactivity (by 40%) and lower usage of higher latency cellular network connections. Such applications can benefit from frameworks that use client assisted network measurements to optimize the placement of cloud based servers to reduce user-perception of high network latencies.

4.2 Impact of network performance on application usage and revenues

In this section, we analyze the impact of network conditions on PK's usage (session length) and revenues.

How did network performance affect the session lengths?

In Figure 18, we show how average user session length for PK varies with average cellular network latency for the session (collected using Insight's active measurements). The latencies are grouped into bins of 0.1 seconds. We find that network performance has a considerable impact on application usage—session lengths decrease with increase in network latencies *i.e.*, users tend to play the game for a lesser time as the network conditions worsen. *An increase of network latencies from 0.3 to 0.9 seconds reduced average session lengths from 30 minutes to 25 minutes (16% reduction).* We observed lower correlation between network latencies and session lengths for SB (not shown here). This is due to the more occasional use of the network and shorter session lengths (Figure 7) for SB users compared to PK .

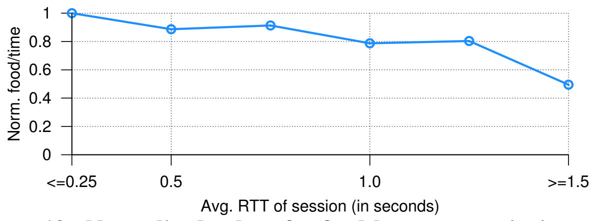

Figure 19: Normalized values for food burnt per unit time at different RTTs for US based PK users (Bin size of 250 ms).

How were game revenues from PK impacted by network performance?

In PK, users can purchase virtual currency (Food) with real money which acts as a direct source of revenue for its developers. For PK users, Figure 19 shows a high correlation between normalized food consumed per minute and the average network latency—food consumption (and hence the money spent) decreases with the increase in network latencies, thereby affecting application revenues. For example, high latency (e.g., 1.0 secs) sessions had around 20% lesser food consumption compared to low latency (e.g., 300 ms) sessions. *Under bad network conditions (latency $>=$ 1.5 secs), food consumption decreased by 52% indicating sessions with potential revenue losses for the developers.*

Summary. Poor network performance (higher network latencies) resulted in revenue losses (upto 52% lower revenue generate rate) as well shorter average session lengths for PK users (16% shorter). Thus, efforts by cellular ISPs to upgrade their deployments and reduce congestion (e.g., adding WiFi hotspots) is important for application developers and users due to rapidly growing usage of these networks. Also, we observed around 42% of the WiFi traffic was still being received over older 802.11bg networks. Thus, developers should also account for the presence of devices with slower network connections.

5. IMPORTANCE OF USER RETENTION

Attracting more regular users and maintaining user interest is a major focus for serious mobile application developers, because it can lead to more revenues.

How long do users stay involved in the applications, i.e., what is the distribution of retention period of the users?

One measure of user interest in the application is the period for which they use the application. For PK, we compute the number of users joining and leaving the game during Age 2 (March 27, 2009 to November 16, 2009, a total of 235 days). We did a similar analysis for SB users over a period of 235 days (August 2012 - March 2013). If a user first accessed the application on day $d1$, and if we do not find the user accessing the application after after day $d2$, we define $d2 - d1$ as the "retention period" of the user, *i.e.*, the total number of days the user stays in the game.

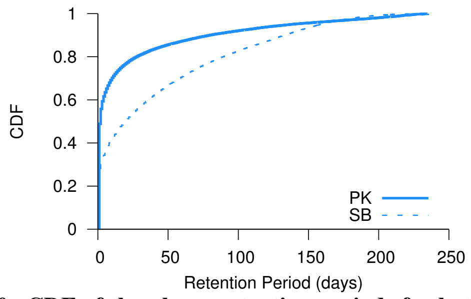

Figure 20: CDF of the player retention periods for both applications.

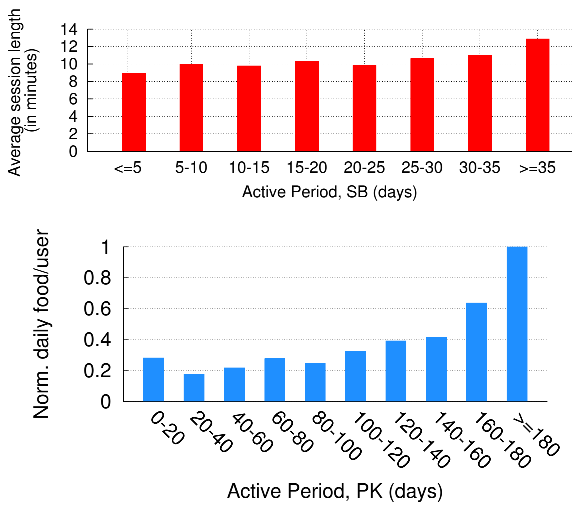

Figure 21: (top) Sessions lengths as a function of active periods (SB), (bottom) Active period vs. normalized average Food expenditure per player per day (PK).

Figure 20 shows the CDF of retention period for both PK (game) and SB (study application). *We observe that around 48% and 27% of PK and SB users, respectively, use the application only for a single day i.e., these users download the application, access it for one or more times during the day and never use the application again.* This indicates that a large proportion of users across both applications had very low retention periods. Many popular sites like AppBrain [2] use downloads to indicate an application's popularity in an app store. However, we observe that downloads alone might not accurately reflect the popularity or user base of an application. For example, only 20% and 42% of PK (game) and SB (study application) users, respectively, were retained for more than one month during this period (235 days).

How did user interest translate into application usage and revenues?

We define the "active period" of a user as the total number of days during which a user accesses an application. For example, if a player plays on day 1 and again on day 5, the active period for the player is two days. For SB, we grouped users based on their active periods over a total period of 3 months. Figure 21 (top) shows the median session lengths for users based on their active periods. The graph points to an increasing trend in session lengths with the increase in user active periods. The median *session lengths for regular users with high active periods (>= 35 days) was around 1.5 times (13 minutes vs. 8.5 minutes) compared to intermittent users (<= 5 days).*

Metrics	Insight	Flurry	AppClix	Android
Session	✓	✓	✓	✓
Device	✓	✓	✓	✓
Location	✓	✓	✓	✓
In-app activity	✓	✓	✓	
User retention	✓	✓	✓	
Network	✓			
Resource	✓			

Table 5: Comparing the feature categories of a few popular commercial mobile analytics solutions with Insight.

Using Insight, we analyzed all Food expenditure transactions in the PK game during Age 2 (7 months). Figure 21 (bottom) shows the relationship between active period and normalized daily average food spent per user. We observe an increasing trend of "food spent per day" as users play the game for more days, *i.e.*, the longer a user stays in the game, the larger is the amount of Food (and therefore, money) spent *per day*. This is especially evident amongst users who played the game for more than 180 days. *These users spend 2.5× more money than users who have played less than 150 days.* For users with active period less than 20 days, the average Food spent is slightly more than some users with longer active periods. This is because new users are given initial Food, and many users with short active periods use up their initial allotment of Food and never buy more.

The analysis shows that it is *crucial to retain old users as they spend more time within the application as well generate higher revenues*. While advertising can help recruit new users, it is necessary to maintain user-interest in the application since older users tend to spend more money. In the case of PK, the developers have increased the frequency of game updates (since Age 3) containing new features and capabilities to maintain user interest. Another way to increase the active periods of non-regular users is to occasionally provide them some free virtual currency. This can provide an incentive to these users to be more active within the game and later generate more revenues for the application.

Summary. Across both PK and SB, a large fraction of the users (27% to 48%) accessed their application for only a single day. This indicates the short retention span of a large proportion of mobile users. Users with larger retention periods generated higher revenues (upto 2.5× for PK). Thus, attracting new users and keeping older users engaged in the application (through feature updates and user incentives) is essential for maintaining application revenues.

6. RELATED WORK

Mobile Analytics. Mobile application analytics provided by app stores like Android Market include limited information about application usage e.g., number of downloads, geographical spread of users, devices used etc. Recently, few commercial analytics solutions [4, 3, 6, 5] for iOS and Android platforms have started providing improved capabilities for tracking application usage. These solutions are mainly focused on providing the business analytics (e.g., statistics about users, locations, user retention etc.) to the developers. Table 5 compares the broad feature categories provided by Insight with some popular analytics solutions. In addition to these analytics, Insight also measures the application footprint (e.g., CPU, memory, battery, screen brightness etc.) on the mobile devices and leverages the application user base to infer the impact of network on application usage and revenues. Further, the data from commercial analytics solutions is only available to the developers of the applications. Insight allowed us to partner with the developers to perform this large scale study across multiple years and share the results with the community. AppInsight [22] instruments the application binaries to identify the critical paths in application execution. Such approaches are complimentary to Insight's goals of understanding the impact of the aforementioned factors on the users' experience.

Smartphone/mobile application usage. Recent research [13, 21, 23] has looked at the characteristics of smartphone usage, such as device and application usage patterns by instrumenting or collecting traces from smartphone devices. The authors in [13] perform a detailed characterization of user session lengths, network and app usage patterns on 255 users while [21] focuses its study on 24 users belonging to a specific socio-economic group. [23] uses smartphone traces to build models for predicting the applications used based on the current context. Other studies have deployed standalone apps to study specific aspects such as contextual uses of different smartphone apps (AppSensor [10]) and network performance on smartphones (3GTest [15]). In [26], the authors utilize a snapshot of the network traces from a cellular ISP to study the similarities in the usage of different types of smartphone applications. Through Insight, we use the application as a vantage point across more than a million users and analyze it along multiple dimensions (e.g., device type, network, user retention). This unique setup captures the impact of factors (e.g., network) on the end user's experience while using an application as well as revenue generation for developers.

7. CONCLUSION

Through the development and deployment of our embedded measurement framework (Insight) within two popular applications, we presented a large scale and long term study of these applications across more than a million users. In addition to providing aggregate application analytics, Insight allows developers to understand the impact of various factors such as device capabilities and settings and network performance on the application usage, revenue generation for developers and user experience. We also use these two apps embedded with Insight to contrast the usage characteristics and the impact of network performance based on the nature and requirements of the app.

Using Insight, we have collected a significant volume data which captures network performance when connecting across diverse locations, through different types of networks, and across different times. We intend to use this data to help users better understand their networks. For this purpose, we have released an application Network Test [7] that performs a simple set of measurements to estimate network latency from the mobile device. It leverages all our network measurement data to perform a relative comparison — the network latency information of different users of a region are compared against a user's own latency to provide a relative grade. A partial data set obtained with our measurements is also currently visualized at `http://networktest.org`. We are currently adding more features to this application and we hope that the users will be able to benefit from our work.

8. ACKNOWLEDGMENTS

We thank the application developers involved in this study for their support, without whom this work would not have been possible. We would like to acknowledge the anonymous reviewers and our shepherd, Richard Mortier, whose comments helped bring the paper into its final form. Ashish Patro, Shravan Ranyanchu, Michael Griepentrog, Yadi Ma and Suman Banerjee are supported in part by the US National Science Foundation through awards CNS-1040648, CNS-0916955, CNS-0855201, CNS-0747177, CNS-1064944, and CNS-1059306.

9. REFERENCES

[1] Amazon ec2 pricing. http://aws.amazon.com/ec2/pricing/.
[2] Appbrain. http://www.appbrain.com/.
[3] Appclix. http://www.mobilytics.net/.
[4] Flurry. http://www.flurry.com/.
[5] Kissmetrics. http://www.kissmetrics.com/.
[6] Localytics. http://www.localytics.com/.
[7] Network Test. http://networktest.org.
[8] Studyblue. http://www.studyblue.com/.
[9] S. Agarwal, R. Mahajan, A. Zheng, and V. Bahl. Diagnosing mobile applications in the wild. In *Hotnets'10*.
[10] M. Böhmer, B. Hecht, J. Schöning, A. Krüger, and G. Bauer. Falling asleep with Angry Birds, Facebook and Kindle: a large scale study on mobile application usage. MobileHCI '11.
[11] C. Chambers, W. chang Feng, S. Sahu, D. Saha, and D. Brandt. Characterizing online games. *IEEE TON'10*.
[12] Engadget. Netflix reveals android app tests that keep it running on 'around 1000' devices daily. http://www.engadget.com/2012/03/15/netflix-android-app-testing-process/.
[13] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin. Diversity in smartphone usage. In MobiSys '10.
[14] P. P. FAQ. Cellular data primer. http://www.pocketpcfaq.com/faqs/cellularprimer.htm/.
[15] J. Huang et. al. Anatomizing Application Performance Differences on Smartphones. In *MobiSys*, 2010.
[16] L. Leiva, M. Böhmer, S. Gehring, and A. Krüger. Back to the app: the costs of mobile application interruptions. MobileHCI '12.
[17] J. Manweiler, S. Agarwal, M. Zhang, R. Roy Choudhury, and P. Bahl. Switchboard: a matchmaking system for multiplayer mobile games. MobiSys '11.
[18] N. Balasubramanian et al. Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications. In IMC '09.
[19] A. Patro, S. Rayanchu, M. Griepentrog, Y. Ma, and S. Banerjee. The anatomy of a large mobile massively multiplayer online game. MobileGames '12.
[20] PerBlue. Parallel kingdom. http://www.parallelkingdom.com/.
[21] A. Rahmati, C. Tossell, C. Shepard, P. Kortum, and L. Zhong. Exploring iPhone usage: the influence of socioeconomic differences on smartphone adoption, usage and usability. MobileHCI '12.
[22] L. Ravindranath, J. Padhye, S. Agarwal, R. Mahajan, I. Obermiller, and S. Shayandeh. AppInsight: mobile app performance monitoring in the wild. OSDI'12.
[23] C. Shin, J.-H. Hong, and A. K. Dey. Understanding and prediction of mobile application usage for smart phones. UbiComp '12.
[24] J. Sommers and P. Barford. Comparing metro-area cellular and wifi performance: extended abstract. SIGMETRICS '12.
[25] T. K. Wee and R. K. Balan. Adaptive display power management for oled displays. MobileGames '12.
[26] Q. Xu, J. Erman, A. Gerber, Z. Mao, J. Pang, and S. Venkataraman. Identifying diverse usage behaviors of smartphone apps. IMC '11.
[27] Z. Zhuang, K.-H. Kim, and J. P. Singh. Improving energy efficiency of location sensing on smartphones. MobiSys '10.