LiveLab: Measuring Wireless Networks and Smartphone Users in the Field

Clayton Shepard*, Ahmad Rahmati*, Chad Tossell+, Lin Zhong*, Phillip Kortum+
* Dept. of Electrical & Computer Engineering and + Dept. of Psychology
Rice University, Houston, TX, United States
{cws, rahmati, chad.tossell, lzhong, pkortum}@rice.edu

ABSTRACT
We present LiveLab, a methodology to measure real-world smartphone usage and wireless networks with a reprogrammable in-device logger designed for long-term user studies. We discuss the challenges of privacy protection and power impact in LiveLab and offer our solutions. We present an iPhone 3GS based deployment of LiveLab with 25 users intended for one year. Early results from the data collection so far highlight the unique strengths and potential of LiveLab. We have two objectives in this position paper. First, we demonstrate the feasibility and capability of LiveLab. By sharing our experience, we seek to advocate LiveLab as a network and user measurement methodology. Second, we present our preliminary findings, and seek feedback from the community regarding what data to collect.

1. INTRODUCTION
We present LiveLab, a methodology to measure smartphone users in the field and to measure wireless networks with smartphone users. The key features of LiveLab include:

- Comprehensive in-device logging of smartphone usage and measurement of wireless networks
- In-field programmability of the logger so that researchers can update the logger and schedule a new measurement very much like they would do with a lab computer.
- A large number of users that use the logged smartphones as their primary phones for a long term (one year).

The motivation of LiveLab is simple. Over half of the world population now has a mobile phone. 17% of mobile phones are smartphones; and the percentage is growing rapidly. Mobile users move around and use their devices and the wireless networks at different times and locations, challenging the measurement of not only smartphone usage but also the wireless networks.

First, the mobility of users and usage leads to significant variation in network quality experienced by the users and considerable diversity in user experiences. Many have studied how to leverage this variation and diversity to improve the performance and efficiency of wireless Internet access [5, 6] and the user experience [8]. Data regarding smartphone usage and user experience are imperative to the design and evaluation of such techniques.

Second, as we have observed in our previous long-term field study [10], smartphone usage is context-dependent. Simply put, a mobile user is likely to use different applications at different locations and access different websites at different times of the day. Such context dependency provides key insights into the optimization of the mobile and network systems, e.g. pre-fetching of web content and pre-launching of applications. Yet such context dependency can only be quantitatively characterized in the field. Existing smartphone loggers, e.g. [12-15], collect very limited context information.

Furthermore, as we experienced in our prior work [10], research hypotheses develop when data are collected from mobile users in the field. This requires the in-device logger to be updated frequently to collect new data. Existing smartphone loggers reported in the literature [12-15], including our prior work [10], employ a static installation method and are difficult to update or maintain.

Finally, existing client-based network measurement solutions require time-intensive war-driving, e.g. [16], which is unlikely to provide a fine-grained and dynamic network map. Wireless network and mobile users can also be measured from inside the network [17]. However, usage data collected by network operators are limited in both scope and detail. For example; they do not include applications that do not access the network. That is, cellular network carriers will be unable to collect data when a user is using WiFi. Furthermore, network operators rarely share their data with the research community, citing privacy and commercial concerns.

The proposed LiveLab methodology aims at addressing these challenges by logging smartphone usage in the field, leveraging mobile users as a network sampling tool, and allowing the logger to be dynamically reprogrammed in the field. Yet, there are a number of key practical challenges to this methodology, including user impact and privacy, long-term study management, as well as the closed nature of many mainstream smartphone platforms. We provide an in-depth discussion of these challenges and offer our experience in addressing the privacy and power impact in Section 2.

To demonstrate the feasibility of LiveLab, we present our iPhone-based implementation of LiveLab and the ongoing one-year deployment with 25 iPhone 3GS users in Section 3. To the best of our knowledge, our iPhone-based LiveLab is the first publicly reported study of iPhone users through in-device logging. We are also the first to describe our smartphone logger implementation in detail and to make our logger open-source.

We have already made intriguing discoveries that demonstrate the capability and strengths of LiveLab. We find our participants use very different sets of applications, but a small set of built-in applications are popular among all participants. We find users started to use most of their most used applications in the first one or two weeks though they continue to explore the App Store throughout the study. Furthermore, we demonstrate the temporal dynamics and trends of application usage, as well as the difference between individual applications and application categories. We also show that websites visited by our participants are location-dependent. Finally, we present the TCP session characteristics observed on our devices, and show that there are often few TCP connections and they are mostly short lived. We discuss these early results in Section 4.

2. *LiveLab* CHALLENGES & SOLUTIONS
We first discuss the challenges of realizing LiveLab and provide our solutions in an implementation-agnostic manner.

2.1 Privacy
One of the primary concerns while developing, deploying, and administering such a comprehensive logger is privacy. In order to develop a better understanding of the privacy concerns of participants, in particular what information they are unwilling to have

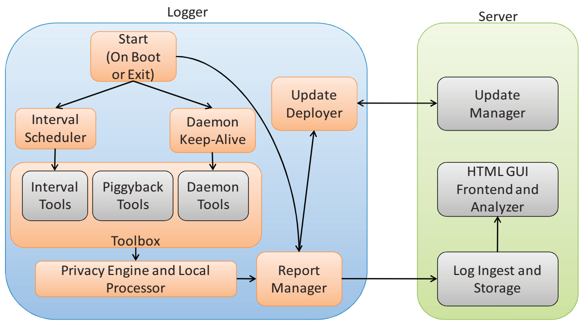

Figure 1: Structure of the iPhone logger implementation

logged, we conducted several interviews. Not surprisingly, we found that participants' biggest concern was regarding their identity. That is, participants do not want researchers to be able to associate their identities with their data. Surprisingly, they are not concerned about some potentially sensitive data being collected as long as the data is not directly linked to their identity. Our participants were fine with a 1-out-of-25 anonymity for much of the data we originally considered private, including GPS location and web access history. In contrast, they are not comfortable with the content of email or instant messages being collected directly, considering it highly private. However, they do not mind if this content is analyzed and sanitized in the device, as detailed below. For our iPhone deployment, we discussed the logger with our participants in depth at a formal meeting before the phones were distributed.

With our participants' concerns understood, we employ the following methods to protect privacy while retaining relevant information for research. First, we leverage one-way hashing to preserve the uniqueness of a data entry without revealing its content. For example, we hash the recorded phone numbers. With hashing, we can still construct call statistics without knowing actual phone numbers. Second, we perform information extraction in the device. For example, we extract emoticons from emails and text messages without collecting the raw content. Finally, we structure the research team so that the data analysis and logger development team do not directly interact with the participants, in order to avoid linking data to user names. A separate human factors team acts as the interface with our participants but does not deal directly with the logger or access the raw data. This enables us to contact the participants in a privacy sensitive manner, which we have found to be necessary on numerous occasions, e.g., to schedule impromptu interviews with users who exhibit a drastic change in behavior.

2.2 Power

Collecting data from smartphones in the field naturally incurs power overhead and reduces battery lifetime. Significantly reduced battery lifetime is likely to impact usage, thus the usage data would not accurately reflect user behavior in real life [10, 18]. Therefore, an accurate logger must carefully mitigate the power impact of data collection. Towards this goal, we have employed the following four logging methods to reduce power consumption.

The first is to drive logging with interrupts. Any time a system event occurs, such as a program being opened, or WiFi being turned on, the logger catches the event in real time and logs it. Interrupt-driven logging avoids periodical polling and captures events in an exact and immediate manner.

The second is piggy-backing. Modern smartphone systems already log a number of items with timestamps, such as call history, SMSs, emails, etc, often for user convenience. We can save energy by simply collecting them routinely, preferably while the device is plugged in and idle. Collection once a day is a good balance between power consumption, data loss, and feedback.

The third is optimizing the logging interval for periodically logged items. This is especially important when the data is power-expensive to collect, e.g. GPS readings and bandwidth measurements. It is almost impossible to maintain a decent battery lifetime if such measurements are collected frequently. Our solution is to trade the length of data collection for frequency of collection. That is, by collecting data for a longer time, we can collect the data less frequently. The rationale is that a longer time will allow the data to be collected at more times of the day and more locations. While such data will not be able to catch certain temporal dynamics, e.g. the location trace of a user in a given day, they still retain key statistics regarding the user and network behavior. Additionally, frequency can be dynamically changed based on context. For example if the phone is plugged in, moving rapidly, or in a new location, data can be logged more frequently.

The fourth is hitch-hiking. A logger can hitch-hike on to existing system wakeups to reduce additional overhead for data collection. Smartphones naturally wake up from low-power mode when the user activates the device, when the system receives events, e.g. an incoming call or SMS, and when the operating system needs to carry out maintenance. By collecting data when the smartphone naturally wakes up, the logger can avoid an additional wakeup.

3. *iPhone* IMPLEMENTATION OF *LiveLab*

In this section, we describe our iPhone 3GS based implementation of LiveLab. While we have used Windows Mobile and Android smartphones for field studies in the past [10, 14], we chose the iPhone for our deployment for the following reasons. First, iPhone represents the cutting edge of smartphone design for usability, accounting for 55% of all mobile internet traffic in the US. With its extreme popularity, it is very easy to recruit participants and ask them to use iPhone for one year. Second, iPhone users have access to the largest number of, not to mention the most popular, 3rd party applications, from the Apple App store as well as numerous third party repositories. The iPhone, however, is known for its lack of openness, compared to Android, Symbian, and Linux smartphones. We have managed to overcome most, if not all, the barriers to implement a fully operational version of LiveLab, detailed below.

3.1 iPhone and Its Closed Platform

The iPhone is one of the most closed platforms on the market, and by default Apple does not allow root access to the device. In order to gain this control, it is necessary to "jailbreak" the device. While a jailbreak is not always immediately available for the most recent iPhone OS, every OS to date has been jailbroken. Due to the jailbreak and logger installation users are firmly instructed to not restore their phones.

The logger is implemented in a very modular and robust fashion, thus updates to the OS may break individual components, but the main functionality will not be affected. For example, logging call history relies on a specific file location and format, but it is relatively trivial to change that path in the logger or update the parsing format. As detailed below, the main logger daemon is written as a shell script in bash. Since the shell is a key component of the OS, it would be virtually impossible for an OS update to break it.

3.2 Logger Realization

Figure 1 illustrates the iPhone logger design, as described below.

Primary Daemon: While the logger utilizes many different languages, including C, perl, awk, SQL, and objective C, the core is

Table 1: iPhone usage logging

Item	Method	Code source
Call, SMS, email, and address book usage	Piggy-back	Built in (privacy and analytics custom)
Web History	Piggy-back	Built in
GPS Location	Interval	Modified from [1]
Accelerometer data	Interval	Custom written
Battery State	Interval	Built in
App. launches; changes to foreground app.	Interrupt	Built in
Installed programs and media, e.g. songs and videos	Piggy-back	Built in / Community Repository
Captured media, e.g. photos, videos, and voicenotes	Piggy-back	Built in / Community Repository
Currently running processes	Interval	Community repository

Table 2: Network usage logging

Item	Method	Code source
HTTP Downlink bandwidth via wget [2]	Interval	Community repository
Total data sent over a network interface via vnstat [3]	Interval	Custom compiled
Full packet or packet headers only via tcpdump [4]	Interrupt	Community repository
Available WiFi Access Points and Info	Interval	Custom written
Bluetooth and Wifi State Changes	Interrupt	Built in
Active network interfaces and their IPs via ifconfig	Interval	Community repository
Active network connections via netstat	Interval	Community repository
Cell tower id, signal strength, and cell geographic id	Interval	Baseband Query
Round trip time to any server via ping [7]	Interval	Community repository
Per hop latency to worldwide servers via mtr [9] with PlanetLab [11]	Interval	Custom Code/Community repository

written in bash. Using the bash script we are able to easily call built in functions, manage child processes, install and use programs from repositories, run custom programs, and add new features. It is best to think of the rest of the logger as a modular set of tools for collecting data. Each of these tools functions completely autonomously, and enabling and disabling them is often as easy as commenting out a single line in the bash script.

The iPhone OS allows daemon processes to be launched by specifying them in the /System/Library/LaunchDaemons folder. In order to ensure thorough data collection and accurate results it is critical that the logger is continuously running, and does not exit unexpectedly. Conveniently, a setting provided by the OS causes the daemon to be restarted anytime it is killed, which we leverage to ensure our logger is always running. When the logger daemon starts, either because the phone booted or the last instance exited, it sleeps for 20 seconds. This allows the phone to finish booting and initialize the UI and network.

Daemon Manager: After this brief sleep interval the *Daemon Manager* launches the child daemon processes responsible for collecting all interrupt driven data, such as packet traces and application switches. We have also used this method to enable higher resolution interval based logging, such as detailed network statistics. These child processes are monitored by the *Daemon Manager* to ensure that they are re-launched if they exit unexpectedly.

Interval Manager: Once the logger and child daemons have been initialized the Interval Manager begins scheduling data collection. In the current version of the logger this schedule is set statically to every 15 minutes, however it is easy to dynamically change the interval based on contextual data. For example, the interval could be decreased to every minute or even every second while the phone is charging. Additionally, which features to log at each interval can be chosen dynamically as well, allowing power-hungry data such as GPS to be logged less frequently.

Hitch-Hiking: By default *all* of the data we collect is done via hitchhiking. In other words our logger never forcefully wakes up the system in order to collect data. It is relatively easy to force the system to wake up at specified times using the private functions in the IOKit framework, however due to power concerns we have opted not to deploy them. Yet, we have found that the logging granularity (i.e. time interval) of the data we are collecting is adequate for our purposes.

Reporting and Auto-updating: Nightly, between the hours of 3am and 7am, all piggyback data is collected and sensitive data is sent through the *Privacy Engine and Local Processor* which analyzes and obfuscates private data locally, so that it is never sent over the network. The *Report Manager* then compresses the data and uploaded to the server via rsync [19]. Rsync was chosen since it will robustly upload any archives that previously failed to upload for any reason (usually network connectivity). During this process the *Update Deployer* checks for any updates on the server, downloads them, deploys them, and exits. The logger is then restarted by the OS through the daemon mechanism described earlier. When the logger is restarted, if necessary, it performs various update tasks such as downloading new packages from repositories and installing GUI applications.

LiveLab Server: The server has three primary tasks: (1) managing and deploying logger updates, (2) collecting and storing logs, and (3) providing feedback and analysis to the administrators. The HTML interface is primarily written in PHP, and provides feedback regarding the status of all mobile devices. Moreover, this functionality creates a very efficient cycle of iterative improvement. It allows us to receive and analyze logs, and push out an improved logger to the participants within a matter of days.

3.3 Data Collection Capabilities

Tables 1 and 2 summarize the logging capability of the iPhone-based LiveLab deployment for usage and network measurements, respectively. The tables also provide the logging method and code source for each component. It is important to note that logging all the information, especially interval-based logging, at the same time incurs a huge battery and performance penalty.

3.4 Power Impact

The data collected through piggy-back and interrupt driven methods has negligible impact on battery lifetime. However, most interval-based data collection has a much more significant impact, and thus is scheduled. That is, only a small subset of interval-based logging is collected on a given day, depending on current research objectives. This allows us to minimize the battery lifetime impact.

Since power impact is a critical concern for LiveLab, we took detailed power readings in order to quantize the impact of the logger. Even with all optional power hungry data collection components

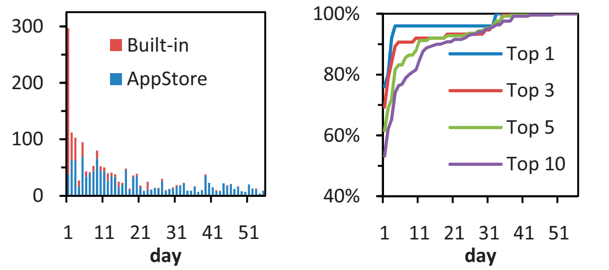

Figure 2: Application usage: (Left) # of new applications used for all participants day by day (Right) Cumulative Distribution of # of days for participants to start to use their top applications

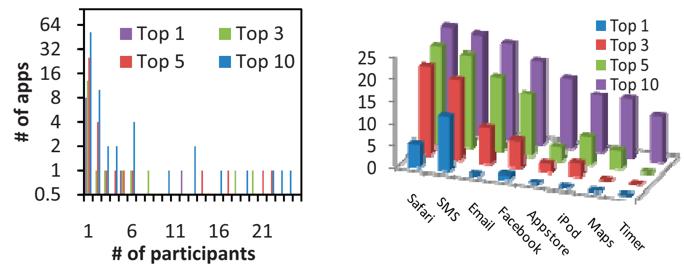

Figure 3: (Left) Participants are very different in their most used applications; (Right) A small set of applications are popular, i.e. among top applications of many participants

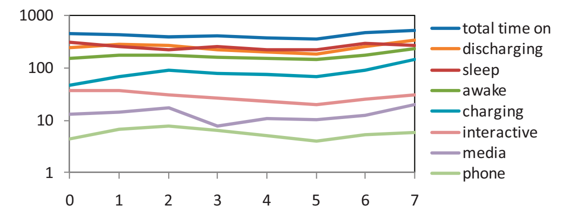

Figure 4: Usage time for each month (average hours for a single user).

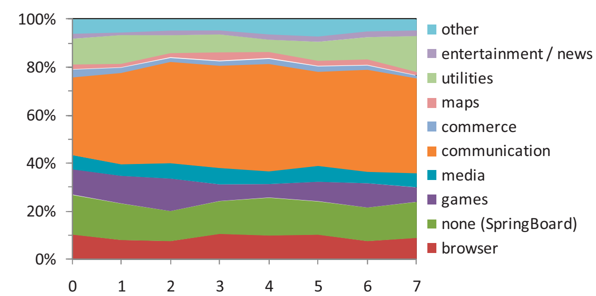

Figure 5: Usage breakdown for top 250 interactive application, for each month.

enabled, such as GPS, HTTP file download, and the accelerometer, our measurements show that the logger consumes less than 5% of the phone battery per day.

3.5 Field Study Participants

We recruited 25 participants from the undergraduate student population at Rice University. In general, they were representative of college students in terms of age (M = 19.7 years) and gender. 18 of the students did not previously own a smartphone. We gave each participant an iPhone 3GS equipped with our logger to use for one year, along with 420 minutes of phone calls per month and unlimited SMS and data. Participants were required to utilize the iPhone as their primary mobile device for the duration of the study, and we ported all participants' phone number to the new iPhone plan. Our human factors team conducts a focus group with a different subset of the participants every month to gather qualitative data regarding their usage and experience.

4. EARLY RESULTS

LiveLab can enable a wide range of research. In this section, we present early results and ongoing research using LiveLab.

4.1 iPhone Usage

While we are still only halfway through a 12 month user study, we have analyzed the collected data regarding the usage of the iPhones.

4.1.1 iPhone Application Adoption

We are able to extract the time an application, either built-in or from Apple App Store, is first used. For each application, we obtain the total time it is used by each participant in the first eight weeks. We can rank the applications based on their total usage time for each participant. We make the following four observations.

First, the first week and especially the first few days see a huge number of applications being used for the first time. Figure 2 (Left) shows the total number of new applications of participants used daily for the eight weeks. It also breaks down the applications into those built-in and those from the App Store. Note that if two participants started to use one application on the same day, that application would be counted twice for Figure 2 (Left). The figure shows that the users almost exhaust all built-in applications in the first two weeks but continue to get new applications on a daily basis even two months into the study.

Second, most top used applications were discovered by the participants in the first week. Figure 2 (Right) shows the percentage of the top applications that have been discovered on a daily basis. It shows that participants have used more than 79% of their top 1, 3, 5, and 10 applications by the end of the first week.

Third, participants are quite diverse in their top used applications. We count the number of participants that have the same application in their top N list. Figure 3 (Left) shows the histograms of such numbers for all applications for N=1, 3, 5, and 10, respectively. For example, there are 78 unique applications from the top 10 lists of all participants and 58 of them only appear in the top 10 list of a single participant. Similarly, out of all the users single most used applications only 10 of them unique, and 8 of those 10 are the most used application for only a single participant.

Fourth, a small set of applications are very popular. Figure 3 (Right) shows the applications that appear in no less than 10 participants' top 10 lists: Safari, SMS, Email, Facebook, App Store, iPod, Maps, and Timer. They are all built-in, indicating that Apple did a very good job bundling useful applications. Figure 3 (Right) shows how often the top eight most popular applications among all users appear in the participants' top n lists, with n ranging from 1, 3, 5 to 10. For example, Safari appears in 24 participants' top 10 lists but is the top 1 for only 5 participants. In contrast, SMS is among the top 10 for 23 participants and the top 1 for 12 of them.

4.1.2 Dynamics in Application Usage

Our study confirms that significant usage changes may occur over time and throughout the study, as we reported in [10]. Therefore, it is imperative for an accurate study of mobile usage to consider both user diversity and usage change over time. Figure 4 shows the monthly phone status statistics for all users. *Charging* indicates any time that the device is connected to a charger/computer. *Awake*

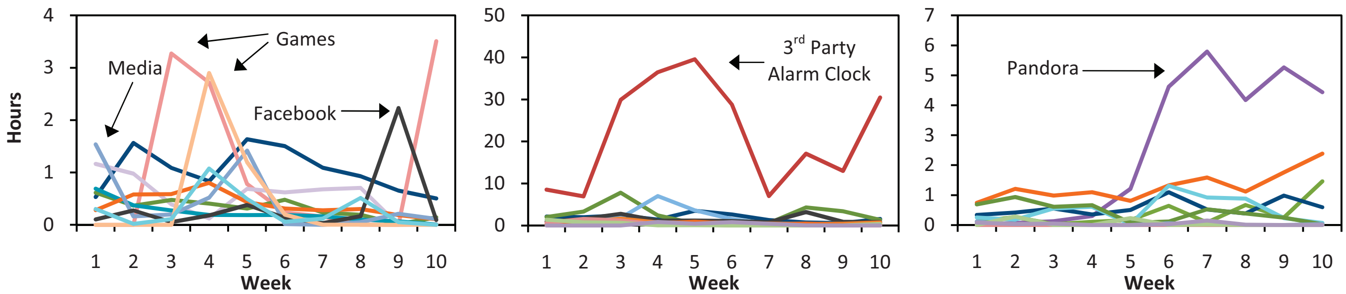

Figure 6: Application usage per week for three example users

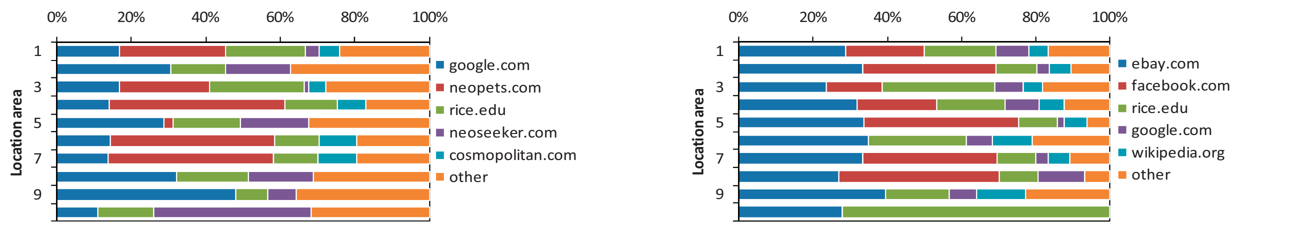

Figure 7: Website access is location dependent: For users A00 (Left) and A07 (Right), probability of accessing their five most accessed websites at their top ten location areas

indicates any time the device is not in a lower power mode, regardless of whether the display is on. *Interactive* usage consists of time that the display is on and the phone is discharging. The *media* category indicates any time the phone is being used to play audio/video, regardless of the display being on (often *phone* and *media* usage occur while the display is off). Notably, some categories overlap, e.g., the phone must always be awake while playing media, or during interactive usage. We can see that interactive usage dropped gradually over the first six months, then increased after the seventh month, coinciding with the start of the school year. This further highlights both the effect of initial excitement and changes in lifestyle on usage, as we reported in [10].

Figure 5 shows the monthly application category breakdown for all users, during interactive usage only, i.e. the display is on and the phone is discharging. We categorized the top 250 applications used by our participants. Other applications are collectively reported as *other*. *None* refers to the time which the device is showing its home screen, i.e. the SpringBoard application. Using this figure we can analyze both the usage of the device as well as the trends in usage over time. *Communication* primarily consists of Facebook, email, and SMS, and accounts for more than a third of daily interactive usage. Gaming and internet browsing are the next most used categories, accounting for about 10% of daily usage each.

Figure 6 shows weekly application usage for individual participants. We can clearly see that users exhibit extraordinarily diverse usage patterns. More importantly, while individual game or media application usage is often subject to large short term changes (Figure 6), overall usage of games and media (Figure 5) remains relatively consistent. In contrast, in our prior study with Windows Mobile phones in late 2007 to early 2008, we observed a decline in game usage, along with comments from users indicating that they had became bored with the games. We believe the different observations in our two studies are due to the wider variety and higher quality of games available on the iPhone platform, In Figure 6 (Left), we can see the several examples of the rise and fall of usage for game and media applications. Longer term change, which can be seen in Figures 6 (Middle and Right), can typically be attributed to the discovery of a new application or a lifestyle change. The user described by Figure 6 (Middle) found a new alarm clock application, which displays the time while the device is charging.

4.1.3 Web Access

Our traces indicate that users' browsing behavior is strongly location-dependent. We used the collected Wi-Fi traces to cluster access points commonly seen together, allowing us to correlate usage with unique locations of Wi-Fi AP clusters. This is similar to the method we employed in [10]. We have then calculated the website access statistics for each location. Figure 7 shows the top five websites accessed by two sample users at their ten most common location areas. We can clearly see the relationship between location and web browsing habits.

We must note that Trestian *et al.* also suggested the relationship between location and the type of visited websites [17]. Since this study collected data from inside the cellular network, it is likely to be incomplete, as smartphone users often utilize non-cellular network connections (i.e. Wi-Fi). Moreover, the data is typically unobtainable for researchers unaffiliated with the network operators.

4.1.4 Importance of Complementary Methods

Our study further confirms the necessity of utilizing qualitative interviews alongside automated logging of usage. In particular, while logs can identify usage changes, interviews are necessary to explain the reasons and circumstances behind the changes, and in some cases, to distinguish usage changes with system glitches.

For example, Figure 6 (Right) depicts a sudden spike in one participant's usage of Pandora, a popular music application. Without the ability to contact and interview the user we would not have been able to discover that the reason for this drastic change in usage. In this case the user acquired a new vehicle with an iPhone dock, which allows them to use their phone for internet radio. In another case we noticed a user who hadn't installed a single application, and had very limited use of built-in applications. We were worried that the logger was malfunctioning or that the user was not using the iPhone as their primary device, but it turned out that the user simply used their iPhone almost solely for calling.

4.2 Network Characterization

4.2.1 Network Usage

Using traces from LiveLab, we are able to analyze TCP connection characteristics. We categorize TCP flows to *web* (http/s browsing and applications), *email*, and *other* applications using destination

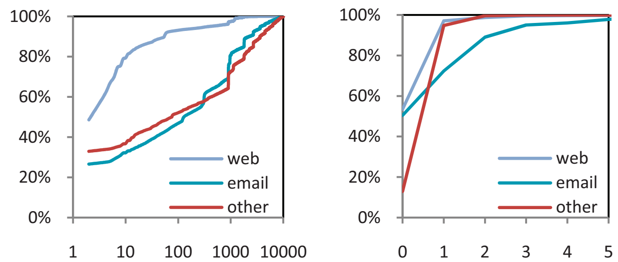

Figure 8: TCP connections are short lived and are rarely concurrent. CDF of TCP connection lifetimes (left) and concurrency (right)

port numbers. Figure 8 shows the CDF (cumulative distribution function) of TCP connection lengths and concurrencies.

Our findings are twofold. First, most web application TCP connections are short lived, i.e. ~2 seconds or less in 48% of cases. In turn, this limits the effectiveness of power saving schemes which rely on long-lived downloads, such as *CatNap* [20]. Second, web applications on the phone the phone are rarely engaged in multiple TCP flows simultaneously. Therefore, multihoming mechanisms (i.e. non-striping) are effective for at most 20% of flows, as the other 80% of times when a web flow exists, it is a single flow. However, we expect this number to increase as more applications and services on mobile devices become available.

4.2.2 Network Conditions

Using the LiveLab deployment, we are in the process of testing a technology called user collaborative network measurement. Our hypothesis is that mobile users from a community can collaborate to produce a fine-grained network coverage map that captures multiple networks, e.g. cellular and enterprise 802.11, and their temporal dynamics. By recording the observed network performance along with the time and location, a mobile user can contribute a number of measurements every day. The measurements from a user over a long time can be aggregated to produce a *personal* network coverage map; the measurements from many users from a community can be aggregated to produce a more complete network coverage map for the entire community.

Such network coverage maps are valuable to both mobile users and network operators. For mobile users, a fine-grained coverage map that provides the performance of available networks at a location and time can help the client selects the network interface properly [6], and makes judicious tradeoffs between energy efficiency and communication delay [5]. For network operators, the fine-grained coverage map will help identify blind spots in the networks and guide incremental network deployment and capacity growth [16].

The user-collaborative approach is superior to existing measurement methods. Existing network-based methods measure and produce a map for a single network, and thus are limited in their ability to help devices determine the best available network. On the other hand, existing client-based measurement methods are based on expensive war-driving, which is unlikely to capture the fine features in geographic coverage or temporal dynamics. The authors of [21] developed a smartphone software (3GTest) for mobile users to measure and report cellular network performance. Our user-collaborative approach is a significant step further from this one-time single-network measurement.

5. CONCLUSION

In this work we showed that logging detailed smartphone usage and measuring networks from smartphones is not only feasible, but also provides unique information regarding both mobile users and networks. While most existing smartphone usage studies used smartphones based on the more open Android platform, we demonstrated that LiveLab can even be deployed to the iPhone, a platform known to be very closed, yet much more popular than Android. Our study of iPhone users also makes a unique contribution by understanding the usage of the most popular smartphones and shedding light into its success.

6. ACKNOWLEDGEMENTS

This project was supported in part by NSF Grants IIS/HCC #0803556 and CNS/CRI #0751173. Special thanks to BigBoss, Kouichi Abe, Chris Alvares, and George Hotz for technical support and/or source code. Three undergraduates also helped develop the logger: Yilong Yao, Daniel Podder, and Norman Pai.

REFERENCES

[1] Alvares, C. Creating an iPhone Daemon *http://chrisalvares.com/blog/?tag=iphone-gps*.

[2] WGet. A free software package for retrieving files using HTTP, HTTPS and FTP *http://www.gnu.org/software/wget/*.

[3] vnStat. console-based network traffic monitor for Linux and BSD *http://humdi.net/vnstat/*.

[4] tcpdump. dump traffic on a network *http://www.tcpdump.org/*.

[5] Nicholson, A.J. and Noble, B.D. BreadCrumbs: Forecasting Mobile Connectivity. *Proc. Int. Conf. Mobile Computing and Networking (MobiCom)*. 46-57, 2008.

[6] Rahmati, A. and Zhong, L. Context-for-Wireless: Context-Sensitive Energy-Efficient Wireless Data Transfer. *Proc. Int. Conf. Mobile Systems, Applications and Services (MobiSys)*. 165-178, 2007.

[7] ping. Ping Network Managment Tool *http://ftp.arl.army.mil/~mike/ping.html*.

[8] Banerjee, N., Rahmati, A., Corner, M.D., Rollins, S. and Zhong, L. Users and Batteries: Interactions and Adaptive Energy Management in Mobile Systems. *Proc. Int Conf. Ubiquitous Computing (Ubicomp)*, 2007.

[9] MTR. MTR Network Managment Tool *http://www.bitwizard.nl/mtr/*.

[10] Rahmati, A. and Zhong, L. A Longitudinal Study of Non-Voice Mobile Phone Usage by Teens from an Underserved Urban Community. *Rice University Technichal Report RECG-0515-09*, 2009.

[11] Peterson, L., Anderson, T., Culler, D. and Roscoe, T. A blueprint for introducing disruptive technology into the Internet. *SIGCOMM Comput. Commun. Rev.*, *33* (1). 59-64, 2003.

[12] Froehlich, J., Chen, M.Y., Consolvo, S., Harrison, B. and Landay, J.A. MyExperience: a system for in situ tracing and capturing of user feedback on mobile phones *Proceedings of the 5th international conference on Mobile systems, applications and services*, San Juan, Puerto Rico, 2007.

[13] Eagle, N. and Pentland, A. Reality mining: sensing complex social systems. *Personal Ubiquitous Comput.*, *10* (4). 255-268, 2006.

[14] Rahmati, A. and Zhong, L. Usability evaluation of a commercial Pocket PC phone: A pilot study. *Proc. ACM Int. Conf. Mobile Technology, Applications and Systems (Mobility)*, 2007.

[15] Falaki, H., Mahajan, R., Kandula, S., Lymberopoulos, D., Govindan, R. and Estrin, D. Diversity in Smartphone Usage. *Proc. Int. Conf. Mobile Systems, Applications and Services (MobiSys)*, 2010.

[16] Robinson, J., Swaminathan, R. and Knightly, E.W. Assessment of urban-scale wireless networks with a small number of measurements. *Proc. Int. Conf. Mobile Computing and Networking (MobiCom)*, 2008.

[17] Trestian, I., Ranjan, S., Kuzmanovic, A. and Nucci, A. Measuring serendipity: connecting people, locations and interests in a mobile 3G network *Proc. ACM Internet Measurement Conference (IMC)*, ACM, Chicago, Illinois, USA, 2009.

[18] Rahmati, A., Qian, A.C. and Zhong, L. Understanding human-battery interaction on mobile phones. *Proc. Int. Conf. Human Computer Interaction with Mobile Devices & Services (MobileHCI)*. 265-272, 2007.

[19] rsync. an open source utility that provides fast incremental file transfer *http://samba.anu.edu.au/rsync/*.

[20] Dogar, F. and Steenkiste, P. Catnap: Exploiting High Bandwidth Wireless Interfaces to Save Energy for Mobile Devices. *Proc. Int. Conf. Mobile Systems, Applications and Services (MobiSys)*, 2010.

[21] Huang, J., Xu, Q., Tiwana, B., Mao, Z., Zhang, M. and Bahl, P. Anatomizing Application Performance Differences on Smartphones. *Proc. Int. Conf. Mobile Systems, Applications and Services (MobiSys)*. 2010.