PHONELAB: A Large Programmable Smartphone Testbed

Anandatirtha Nandugudi, Anudipa Maiti, Fatih Bulut, Sonali Batra, Taeyeon Ki
Murat Demirbas, Tevfik Kosar, Chunming Qiao, Steven Y. Ko and Geoffrey Challen

Department of Computer Science and Engineering
University at Buffalo, The State University of New York

team@phone-lab.org
http://www.phone-lab.org

ABSTRACT

Although smartphones have emerged as the most significant large-scale mobile platform in computing history, the scale of smartphone experimentation has lagged behind. Keeping pace requires new facilities that enable experimentation at scale to ensure that research discoveries translate to the growing network of smartphone devices.

This paper introduces PHONELAB, a 191 device smartphone testbed deployed at SUNY Buffalo. PHONELAB provides access to an incentivized group of participants ready to engage in Android experimentation. The testbed will open for public experimentation in January, 2013, and grow to over 700 smartphones by 2014.

To demonstrate the power of PHONELAB we present five selected results from a usage characterization experiment run on 115 phones for 21 days. We use each result to motivate a future PHONELAB experiment, demonstrating the power of PHONELAB to enable mobile systems research.

1. INTRODUCTION

Smartphones have become the most popular computing platform. Google reports 1.3 M Android device activations per day in September, 2013 [11], while IDC projects that 224 M smartphone units will ship worldwide in 2013 Q4, a 40% increase over 2012 Q4 [15]. Taken as a whole, the growing network of smartphone devices represents the largest and most pervasive distributed system in history.

Meanwhile, the scale of smartphone experimentation is not keeping pace. A small survey of MobiSys'12 papers reveals that when smartphone evaluations use real devices, they use small numbers of phones—3, 12, or 20 [28, 19, 23]. Other experiments use simulations driven by small, old, or synthesized data sets [32, 20, 16]. In either case, large-scale results from real users would be more compelling. While multiple factors—including recruitment, human subjects compliance, and data collection—make large-scale smartphone experimentation challenging, harnessing the growth of smartphones requires evaluating new ideas at scale.

We present PHONELAB, a large programmable smartphone testbed enabling smartphone research at scales currently impractical. PHONELAB provides access to a large and stable set of participants incentivized to participate in smartphone experimentation. From 191 participants in 2012, PHONELAB will grow to over 700 participants in 2014. By exploiting locality, PHONELAB increases the density and interaction rate between participants, facilitating the evaluation of phone-to-phone protocols and crowd-sourcing algorithms.

By utilizing the Android open-source smartphone platform, PHONELAB enables research above and below the platform interface. Researchers can distribute new interactive applications or non-interactive data loggers, but can also change core Android platform components, allowing PHONELAB to host systems experiments impossible to distribute through the Play Store.

We make two contributions in this paper. First, we introduce PHONELAB. After comparing against other approaches in Section 2, Section 3 describes the testbed design and implementation, introduces our participants, and explains experimental procedures. Section 4 describes the Android logging framework and the visibility it provides into the Android platform.

Second, we demonstrate that PHONELAB is powerful and usable. Section 5 describes a usage measurement experiment run by 115 PHONELAB participants for 21 days. Rather than attempting a comprehensive analysis of the dataset, we use it to highlight the power of PHONELAB and breadth of research it supports. To do so, Section 6 presents five results on:

- overall battery usage (Section 6.1),
- opportunistic charging (Section 6.2),
- 3G to Wifi transitions (Section 6.3),
- application usage patterns (Section 6.4), and
- location data sharing (Section 6.5).

For each, we first present our data and then describe how to use PHONELAB to perform further investigation. We hope that these examples will encourage PHONELAB use by the mobile systems community.

	Scale	Realism	Locality	Relevance	Power
PhoneLab	■	■	■	■	■
LiveLabs	■	■	■	■	
One-off Studies			■	■	■
Application Marketplaces	■	■		■	
Large Traces	■	■	■		

Table 1: **Smartphone experimentation comparison.** Only PhoneLab provides all necessary features.

2. SMARTPHONE EXPERIMENTATION

PhoneLab provides the features necessary for smartphone research—power, scale, realism, locality, and relevance:

- **Scale:** PhoneLab will grow to 700 participants, already recruited to participate in experiments.
- **Power:** PhoneLab allows application-level experiments as well as platform-level, i.e., Android kernel, middleware, libraries, and Dalvik virtual machine.
- **Realism:** Participants use the phones as their primary device.
- **Locality:** Most participants live in Buffalo near SUNY campuses, enabling research requiring device-to-device interaction.
- **Relevance:** PhoneLab allows researchers to stop relying on out-of-date datasets. Instead, new data can be collected in the most appropriate way for the experiment.

We believe that PhoneLab is the only smartphone testbed providing all of these features together. Table 1 summarizes PhoneLab's features compared to other approaches, which we discuss individually below.

2.1 LiveLabs

LiveLabs [4] is the most similar testbed to PhoneLab. While it provides many desirable features, it lacks the ability to perform platform experimentation. LiveLabs will eventually have three sites—a mall, a campus, and a theme park—that are instrumented with custom wireless infrastructure. Their goal is to scale to 25,000 participants, but currently only have tens of participants connected. LiveLabs plans to provide public interfaces to collect processed location and activity information. Their aim seems largely to support commercial and marketing-driven location and consumer analytics, rather than smartphone systems experimentation. However, given that the testbed is still being built, its features may change.

2.2 One-off Studies

The standard evaluation methodology when using real devices is to recruit a small number of volunteers composed of fellow researchers and acquaintances. Clearly the scale that can be achieved this way is limited, but this approach also lacks realism. A small number of personal contacts is unlikely to produce a representative sample, and users in a short-term study are unlikely to be willing to use a new phone as their primary device. In comparison, PhoneLab amortizes the recruitment and maintenance costs across multiple experiments, and provides repeated access to the same participants so that findings can be verified and competing approaches compared.

2.3 Application Marketplaces

A few research projects [14, 33] have demonstrated that it is possible to reach large numbers of users by using online application stores such as the Google Play Store or Apple App Store. Assuming participants can be incentivized to participate in experiments, this methodology still lacks power and locality. It is unlikely that applications distributed via online marketplaces will be downloaded and used by large numbers of co-located participants. In addition, these stores only distributed applications and cannot be used for experimentation with core system components.

2.4 Large Traces

Research evaluations that require large-scale measurement data can be challenging, as large-scale measurement requires a combination of scale, realism, and relevance. In order to conduct large-scale measurement studies, researchers have resorted to arduous methods such as obtaining and analyzing service providers' network-level data sets [31, 26, 27], incentivizing a large number of participants themselves [12], or recruiting volunteers through a long period of publicizing the project [25]. Although it is possible to obtain large-scale traces using these methods, the experiments are not repeatable and the traces quickly become obsolete.

Our experiment case study described in Section 5 demonstrates how PhoneLab greatly simplifies trace collection, while providing access to participants and allowing experiments to be repeated. The inherent difficulty of large-scale trace collection is precisely why we use our own data collection and analysis to demonstrate the power of PhoneLab.

3. THE PHONELAB TESTBED

PhoneLab was designed to fill a gap in existing smartphone experimentation capabilities. As mentioned earlier, current experimental approaches are forced to trade off, power, scale and realism. PhoneLab achieves power by utilizing Android open-source devices and a self-signed build which allows us to update any software components; scale by amortizing recruitment overhead, management burden and incentive costs across multiple experiments; and realism by recruiting a diverse set of participants and limiting experimental intrusiveness. We describe the architecture PhoneLab in more detail below.

3.1 Overview

PhoneLab currently consists of 191 participants[1] using Sprint Nexus S 4G smartphones [5] running Android 4.1.1, Jelly Bean [1]. Participants receive discounted voice, data, and messaging, and are instructed to use their PhoneLab phone as their primary device.

PhoneLab experiments are either distributed through the Play Store or as platform over-the-air (OTA) updates. Participants are notified of new experiments and choose whether to participate after reviewing what information will be collected about them. PhoneLab participants are *required* to participate in experimentation but *not required* to participate in any particular experiment. They may remove experiments that they deem too intrusive or that negatively affect their device. Some experiments may run in the foreground like typical applications and require the participant interact with them. Others may run quietly in the background collecting useful information.

PhoneLab users must provide human subjects review documentation, a list of log tags to capture (which we describe later in this section), and their experimental software—either a link to the Play Store or a patch against the current PhoneLab platform source. Experiments generate data through the standard Android logging interface. Log messages generated by PhoneLab experiments are captured and uploaded to a central server while the device is plugged in and charging. When experimentation completes, the user receives an archive containing every log message matching their tags generated by all participating devices.

3.2 Platform and Device

PhoneLab phones run the popular Google Android open-source smartphone platform (AOSP). Using an open-source platform for PhoneLab was an obvious choice for obvious and less-obvious reasons.

[1]We refer to people carrying PhoneLab phones and participating in experiments as PhoneLab *participants*, to differentiate them from researchers running PhoneLab experiments which we call *users*.

The obvious reason is that the AOSP allows PhoneLab users to experiment with any software component, meeting our goal of providing a powerful testbed. Modifications to Android services that provide location, access networks, and manage power can be benchmarked alongside unmodified devices. Of course, power also creates problems: faulty experiments can render phones inoperable and threaten participation. As a result, experimentation at the platform level will require additional pre-deployment testing and interaction with the PhoneLab team when compared with experiments that only distribute novel applications or collect data at the application level.

We have also found that using an open-source platform has other, less obvious benefits. First, the availability of the Android source makes PhoneLab instrumentation easier even when collecting data from the application level because it gives a visibility into hidden APIs. For example, our usage characterization experiment, described in Section 5, uses Java reflection to access hidden battery usage APIs.

Second, the AOSP allows us to sign the platform image used by our participants. When the same key is used to sign a software package, that application may run as the system user with root privileges. Using this feature allows us to distribute and update core PhoneLab experimental management software via the Play Store while retaining the privileges necessary to collect logs and perform platform updates.

Finally, we expect that our base PhoneLab platform image will evolve to meet the needs of the research community. While we have found that Android already logs a wealth of information about platform operation, there are places where more information could be exposed or logged in a more experiment-friendly way. Controlling the platform provides the opportunity to supplement existing interfaces or add additional logging to make experimentation and data collection easier.

We have distributed Nexus S 4G smartphones to our first group of participants. The Nexus S 4G was first released by Sprint in May, 2011, and was one of the official AOSP development phones at the time PhoneLab development began. Its features are summarized in Table 2. While we expect to receive yearly phone upgrades and will distribute a more up-to-date device to our second group of participants, we anticipate that the prohibitive cost of the newest flagship smartphones will prevent us from ever deploying them on PhoneLab.

3.3 Participants

Recruiting a large number of PhoneLab participants requires effective incentives. In their first year of PhoneLab participation, voice, data and messaging are free with funding provided by the National Science Foundation (NSF). This free year of service plays a ma-

CPU	1 GHz ARM Cortex A8
GPU	PowerVR SGX540
RAM	512 MB[1]
Storage	16 GB of NAND Flash, divided into 1 and 15 GB partitions.
Battery	1500 mAh 3.7 V Li-ion.
Display	4" 480 x 800 touch screen.
Networking	1x/3G/4G (WiMax) cellular data, 802.11 b/g/n Wifi, Bluetooth, NFC, and USB.
Sensors	GPS, accelerometer, gyroscope, proximity, magnetometer and light sensor.

[1] 128 MB is reserved for the GPU.

Table 2: **The Samsung Nexus S 4G smartphone.**

Affiliation			
Freshman	64	Masters	5
Sophomore	33	PhD	53
Junior	1	Faculty/Staff	29
Senior	1	None	5
Gender			
Female	51	Male	140
Age			
Under 18	12	30–34	15
18–19	74	35–39	6
20–21	12	40–49	13
22–24	22	50–59	7
25–29	29	60+	1

Table 3: **Demographic breakdown of 191 PhoneLab participants.** Date ranges are inclusive.

jor role in our recruiting efforts. In subsequent years, participants pay a deeply discounted $45 per month rate for unlimited data and messaging through a deal negotiated with Sprint. Sprint has proved to be an ideal partner for the PhoneLab project, both helpful with testbed logistics and still willing to provide unlimited data plans to subscribers.

Because participants may leave at any time, the front-loaded cost structure of our incentives makes it most efficient to recruit participants who will be able to continue as part of PhoneLab for multiple years. While we anticipate that some of our first group of participants will leave after a single year, interviews with them will help us identify long-term participants during subsequent years. Long-term participants allow us to amortize the first free year and provide a stable group comfortable being a part of PhoneLab experimentation.

When recruiting our first batch of participants, we intended to target freshman and sophomore SUNY Buffalo (UB) students as well as incoming PhD students. The University at Buffalo has a large international graduate student community, and many of these students arrive on campus without phones or phone contracts, making them ideal multi-year PhoneLab participants. After a first round of smartphone distribution in late August and early September 2012, we also began to reach out to the professional population at SUNY Buffalo in an effort to increase the number of potential long-term participants as well as the diversity of our participant pool.

In the end, we believe that we were successful in recruiting potential long-term participants. Table 3 describes the demographic breakdown that we achieved. We have handed out our phones to several masters or senior students because they are involved PhoneLab research. The majority consists of potential long-term participants. Roughly half of our participants are first- and second-year undergraduates, a quarter PhD students, and a fifth faculty, staff and other professionals. However, males greatly outnumber females, and the young outnumber the middle-aged and older, both unrepresentative features we will try and rectify in year two. For management reasons we limited participation to people with a SUNY Buffalo affiliation except for several exceptions: a local reporter, a technology writer, and an international rock star.

3.4 Testbed Software

PhoneLab devices are deployed with a small piece of testbed management software embedded in the Android platform image. This heartbeat service uploads periodic reports including information about device location, battery levels, and the installation status of other core PhoneLab components. This information is only used for testbed management and will never be released to researchers.

The heartbeat service is also responsible for starting the primary PhoneLab configuration and data collection software when the phone boots, which allows us to bypass an Android security feature that normally prevents services from running in the background unless started by a foreground application. In order to remain unobtrusive, our experimental management software does not have a foreground component and thus would not normally be able to start.

Experimental configuration, log collection, data upload and platform updates are performed by the PhoneLab experimental harness, which is installed and updated through the Google Play Store. By signing it to match the platform build key it runs with root privileges, necessary to collect logs from all applications

and perform platform updates. Periodically, the experimental harness retrieves an XML configuration from a central PhoneLab server. The configuration specifies what background experiments to start or stop, what data to collect, which server the phone should upload data to and the policy for when to perform uploads. The PhoneLab harness also uploads status information to the server during the configuration exchange, including what versions of various harness components are installed, what experiments are running and how much data is waiting to be uploaded.

PhoneLab logging and data collection must be unintrusive. If it is not, either our participants will leave or their usage patterns will be affected. We believe that we have achieved this goal. First, measured battery usage of PhoneLab is low. A conservative overhead estimate that includes all of the applications that run as the shared system user comes to a per-participant average of 2.4%. This should be considered a strict maximum. Our policy of only uploading while the device is plugged and charging eliminates the overhead of the most power-hungry task.

Second, we have received no major complaints about our the final version of our PhoneLab experimental harness after we instructed participants to install it. Given that participants we allowed to use their phone without our software for several months, we believe that any significant changes in phone behavior caused by our experimental harness would have been noticed.

3.5 Safety and Privacy

PhoneLab is different from many other computer systems testbeds, such as Emulab [30, 3], PlanetLab [22, 7], MoteLab [29], or OpenCirrus [9, 6]: our experiments involve real people. There are two core requirements regarding our participants. First, they should use their phone as they normally would, which motivated the design of unintrusive testbed management software. Second, and more importantly, they must feel safe and in control while part of PhoneLab.

To accomplish this, when possible, we leverage several existing safety mechanisms. First, we require an Institutional Review Board (IRB) to review each PhoneLab experiment for human subjects compliance. IRB approval or an official waiver is required before any PhoneLab any experiment can begin.

Second, we distribute experimental applications to a group of developers prior to broader release, allowing us to identify any significant problems before they reach our participants. This step is particularly important for platform experiments, which must be established as stable before being distributed.

Finally, we utilize Android's existing safety and privacy mechanisms. Participants are presented with the typical Android privacy dialog during experiment installation. Rather than building an alternate distribution channel or privacy mechanism, we felt it was sufficient and probably better to use a process participants are familiar with. After installation, if a participant discovers that an experiment malfunctions or wastes power, they can uninstall it. If we notice patterns of experimental removal, we will flag the experiment and notify the researcher.

Tag Name	Tag Count	%
ActivityManager	96,251,731	13.7
dalvikvm	92,565,828	13.1
ConnectivityService	19,195,475	2.7
ActivityThread	17,447,815	2.5
PhoneStatusBar	13,823,998	2.0
SizeAdaptiveLayout	9,857,534	1.4
wpa_supplicant	9,279,597	1.3
System.err	8,141,399	1.2
SAN_SERVICE	7,530,577	1.1
LocationManagerService	6,640,001	0.9
DexLibLoader	5,438,086	0.8
SecCamera	5,436,968	0.8
HeartbeatService	4,871,085	0.7
Beautiful Widgets(4120000)	4,692,578	0.7
AudioCache	4,447,544	0.6
k9	4,330,848	0.6
SensorActivatorService	4,177,370	0.6
ThrottleService	4,121,301	0.6
VoldCmdListener	4,014,302	0.6
WindowManager	3,948,168	0.6
AudioHardware	3,913,724	0.6

Table 4: **Top 20 log tags generated by Android.** PhoneLab has collected 704,216,410 log messages from 7556 different tags. Tags generated by PhoneLab tools and our usage experiment are ommitted.

3.6 Bootstrapping and Management

We began advertising PhoneLab on campus via posters, flyers, Facebook, and mass emails in late July, 2012. As mentioned previously, PhoneLab phone distribution began on August 24, 2012. Most phones were distributed between August 24, 2012 and August 31, 2012. Our initial plan was to distribute 200 phones during that period, but we ran into an unexpected shortage of supplies for Nexus S 4G. Due to this reason, the last device was handed out on October 29, 2012.

We delayed the release of our experimental harness for two months until November 8, 2012. This was done for several reasons. First, we wanted to complete the distribution of phones. Second, we wanted to complete development and testing of the harness and backend infrastructure. Third, we wanted to receive training in human subjects experimentation and prepare the materials for our first experiment. Finally, the delay allowed our participants to develop normal usage patterns before experimentation began. For this last reason, we may repeat a shorter version of this delay with subsequent new groups of PhoneLab participants.

Tag Name	Tag Count	%	Description
PhoneLabSystemAnalysis-Snapshot	4,507,143	71.8	Collects battery breakdown across components and applications. Polled every 15 minutes.
ActivityManager	1,078,872	17.2	Logs application management actions.
PhoneLabSystemAnalysis-Telephony	240,882	3.8	Records phone call state and radio signal strength.
PhoneLabSystemAnalysis-BatteryChange	212,929	3.4	Logs every change to the battery level.
PhoneLabSystemAnalysis-Wifi	144,163	2.3	Logs connection state, scan information and signal strength.
LocationManagerService	45,478	0.7	Records when GPS is enabled and disabled.
PhoneLabSystemAnalysis-Location	26,588	0.4	Passively logs all location updates.
PhoneLabSystemAnalysis-Misc	20,960	0.3	Logs when the screen turns on and off.
SmsReceiverService	2686	0.0	Used to count text messages sent and received.
PhoneLabSystemAnalysis-Packages	112	0.0	Records when applications are installed and removed.

Table 5: **Log tag statistics for one day during our experiment.** 6,279,813 total log tags were collected.

3.7 Experimental Procedures

To conclude, we review PhoneLab experimentation from a researcher's perspective.

First, develop your application locally. Any information logged through the standard Android logging library can be recorded. In addition, the platform may already be logging useful information for you. Keep track of all the log tags you want PhoneLab to capture. Approach your local IRB and receive experimental approval and upload your application to the Play Store.

Second, upload your list of log tags, IRB letter, and link to your application on the Play Store through the PhoneLab website. We will contact you when we begin beta testing and again once your experiment is ready for the testbed. During beta testing you will be provided with PhoneLab log output to ensure that your experiment is running properly.

Finally, your experiment will be scheduled. Our goal is to maintain a medium-sized list of active experiments for our participants: large enough to make good use of the testbed, but small enough to ensure that each experiment is picked up by many participants. When your experiment completes, you will receive a archive with messages matching the tags you selected.

4. ANDROID INSTRUMENTATION

Android's logging mechanism provides a surprisingly powerful view of the internals of platform operation. Despite instructions that state to disable logging before release, many developers either forget or ignore this advice. We have seen logs generated on PhoneLab with 7556 different log tags, indicating that information about application behavior is also available.

Here we present several useful Android logging techniques suitable for use on PhoneLab. While we do plan on extending our data collection interface to support arbitrary data generated by experiments, we believe that the logging interface will prove a popular way to recover data. Particularly because support for Android logging in Eclipse provides a seamless transition from local experiment development to PhoneLab distribution.

4.1 Intent Monitoring

Inter-process communication on Android occurs via *intents*, which are Android message objects. Because the Android platform uses broadcast intents to distribute useful information to applications, they provide ideal logging hooks. An experimental application can subscribe to intents that it is interested in, and log information when they arrive.

Broadcast intents are sent when applications are installed or uninstalled, the radio of Wifi signal strength changes, the phone rings and is answered, and the battery level changes. Our usage experiment described in Section 5 subscribes to many of these useful intents and uses them to monitor device behavior.

4.2 Log Snooping

Experiment-driven logging of information obtained through intents may not be sufficient to reveal all events of interest. In certain cases, however, information that cannot be obtained and logged by an experiment is already ending up in the Android logs via messages sent by another component. We noticed during testing that many system components logged useful information, and so received IRB approval to collect all logs tags generated by participants phones—not only the ones generated by our usage experiment. This also served as a useful stress test on the PhoneLab infrastructure infrastructure.

Table 4 lists the top 20 tags from the over 700 million log messages in our database. As the table demonstrates, many core Android services already dump data, much of it useful, to the system log. Our usage experiment also uses several of these tags to uncover information that would normally be accessible, such as the screen state transitions. Table 5 has more details.

To make log snooping more feasible and useful we are exploring the option of improving logging coverage within the Android platform. Our experience with our first experiment has indicated that some information is more difficult to obtain than we would prefer, and other pieces of critical information are entirely missing. For example, while the `ActivityManager` tags indicate

when applications are started, use of the back button by the participant is not logged. This makes it impossible to determine what application is currently in the foreground at a fine granularity.

With access to the platform source, we can improve the visibility of important usage information. Another benefit of this approach is that experimenters will not have to incorporate common code for logging standard Android information into their experimental applications. Instead, they will simply request the appropriate tag be added to their log archive.

4.3 Java Reflection

The third monitoring technique is to use Java reflection to access hidden APIs of Android. Typically, this is not a recommended practice because hidden interfaces can change any time and break the application that uses them. However, if used carefully, it can be a useful tool to collect information not available otherwise. Battery statistics is a good example because there is no public Android interface that gives access to the information. However, this information is still available through hidden APIs that Android's Settings application uses to display battery statistics. Using Java reflection, any application can access these APIs. In fact, our experiment uses this method to access battery statistics as we describe in Section 5.

5. EXPERIMENT CASE STUDY

As a case study in PhoneLab usage, we have developed a measurement student, deployed it on 115 phones and collected data for 21 days. While less exciting than the potential experiments we discuss later, we felt that a measurement study was an ideal place to begin. It demonstrates the scale, realism, and power of PhoneLab, and it generates a broadly-useful dataset that will serve as a starting point for future experiments.[2]

5.1 Usage Measurement

Our experiment attempts to collect information about all salient features of smartphone usage: networking, mobility, power consumption, and application usage. Table 5 describes each log tag used by our experiment and what data it collects. Notice that we use a mixture of active log generation and log snooping.

5.2 Logging Tool

Our experiment records usage information in two ways. The first way is to take a snapshot every 15 minutes. This snapshot is intended to capture the overall state of the phone periodically. The information we capture includes the amount of battery consumed, the amount of data sent and received over 3G or Wifi, storage use, and other salient features. We have chosen the 15-minute interval in order to reduce the battery consumption of our experiment. Whenever we can, we also log broadcasts intents that we receive representing per-event information such screen lock transitions, Wifi scan results, call status, and power state changes.

Most of the information we collect is available either through the standard Android interfaces or by subscribing to system intents. The only exception is the information related to battery since there is no interface or intent that provides the information. Due to this reason, we use Java reflection to introspect the internal battery APIs. PowerTutor [33] takes a similar approach to analyze battery usage.

5.3 Approval, Distribution and Deployment

IRB approval was quick. We had two revisions due to our misunderstanding of the instructions, but the turnaround time for each was about a week.

We uploaded our experiment to the Play Store on November 14, 2012 and announced its available to our participants via a mass email. Within a day, 82 participants installed our experiment. After 5 days, the number grew to 115. We have only sent out the email announcement once, and this may be the reason that not every participant has joined our experiment. However, the fact that 115 participants did elect to participate within a week after only one email indicates that our participants understand PhoneLab expectations.

Interestingly, one participant has expressed a concern about the permissions our experiment requested citing the lack of accurate information about Android permissions. The participant was particularly concerned about two permissions, "Hardware controls" that we request for collecting camera usage and "Phone calls" that we request for telephony usage. Since Android's default descriptions provide vague descriptions for these permissions such as "(Hardware control permission) record(s) audio" and "(Phone call permission) determine(s) ... the remote number connected by a call," the participant believed our experiment was doing those things. A survey study has reported a similar problem that the user comprehension level for Android permissions is remarkably low [13].

We leveraged our data logging and collection mechanism by labeling different types of information with different tags. For example, we use `PhoneLabSystemAnalysis-Snapshot` for snapshots and `PhoneLabSystemAnalysis-Location` for location usage. Table 5 describes most of the tags that we used.

6. USAGE EXAMPLES

This section presents five PhoneLab usage examples. Each vignette begins by highlighting an inter-

[2]Access to data generated by our measurement study will be allowed with IRB approval.

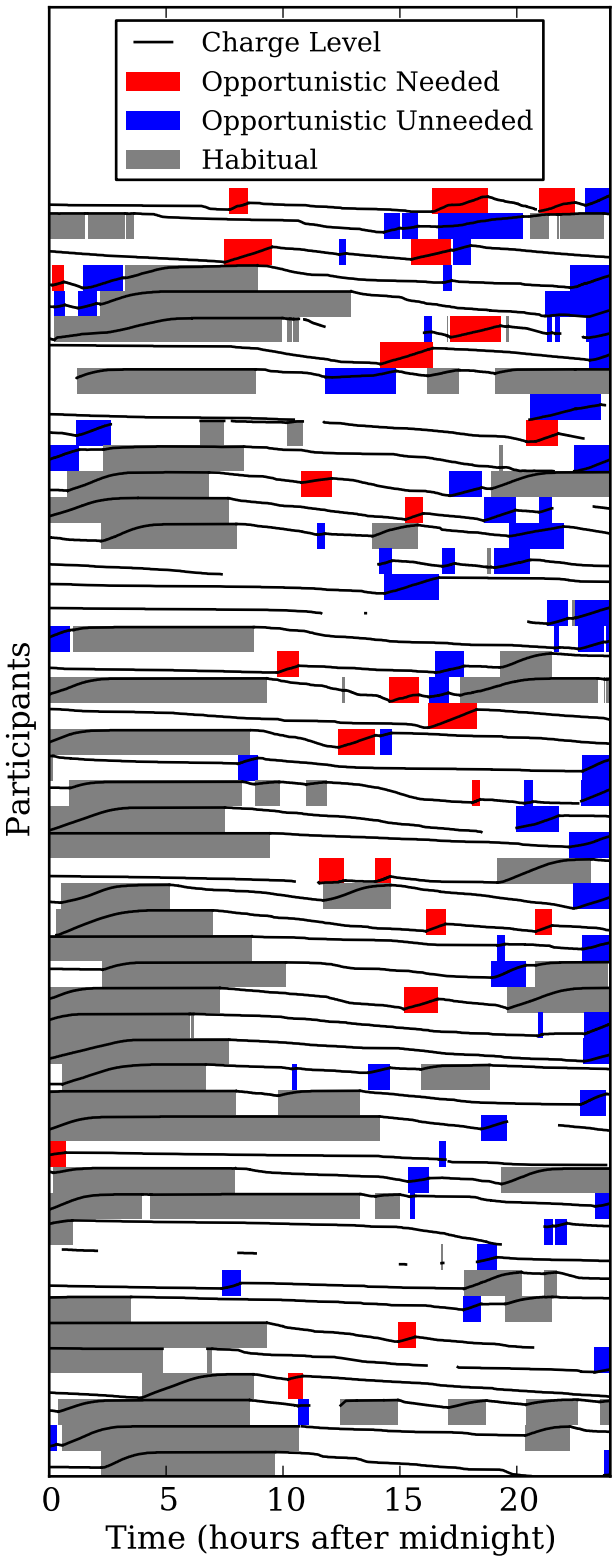

Figure 1: **Patterns of opportunistic charging.** Many users perform opportunistic charging multiple times during the day.

esting or important aspect of the usage data we have collected. Our goal, however, is not to conduct an exhaustive analysis. Instead, each example continues by discussing how the presented results would guide the design of future PhoneLab experiments.

6.1 Overall Battery Usage

Smartphones are constrained by power, and a large amount of research on smartphone systems is motivated by energy conservation [25, 10, 20]. While power is a definite concern, evaluating the potential impact of energy saving approaches requires an accurate breakdown of where energy is used by real phones. Only then can we be sure we are addressing actual energy bottlenecks and put relative energy savings into context.

6.1.1 Energy Breakdown

A single-day component-by-component breakdown for the entire testbed and per-participant is shown in Figure 2. Our results are similar to those reported by other studies, and indicate that mobile data (labeled as "Idle data" and "Active data" depending on the state), the screen, and CPU usage are the main sources of smartphone power consumption. The per-participant bars also show a great deal of variation, with differences in both the amount and the breakdown of energy consumed by each participant.

One supposedly power-hungry component that has less of an impact than we had expected is the GPS. This is particularly surprising given the large amount of location-monitoring work motivated by GPS power consumption. One of several factors may be at work. First, the Android platform estimates the GPS chipset current consumption at 50 mA. This number is used by the standard "Fuel Gauge" battery monitor and by our calculations. However, it is lower than the data sheet for the Broadcom 4751 GPS receiver [2] and may represent a best-case average. Still, even if the GPS current consumption is off by as much as a factor of five, it does not represent a significant contribution. Other hypotheses are that Android network location is providing location with sufficient accuracy for many applications, eliminating the need for GPS, or participants and applications may simply be conscious of GPS power consumption and taking steps to control it.

6.1.2 Future Experiments

While previous smaller studies on earlier Android models [25] have presented similar taxonomies, the process of identifying energy bottlenecks must be repeated regularly as hardware and user behavior changes. PhoneLab provides an ideal environment for repeating energy usage experiments. Access to a stable set of participants allows us to identify changes due to participant behavior, as participants develop an awareness

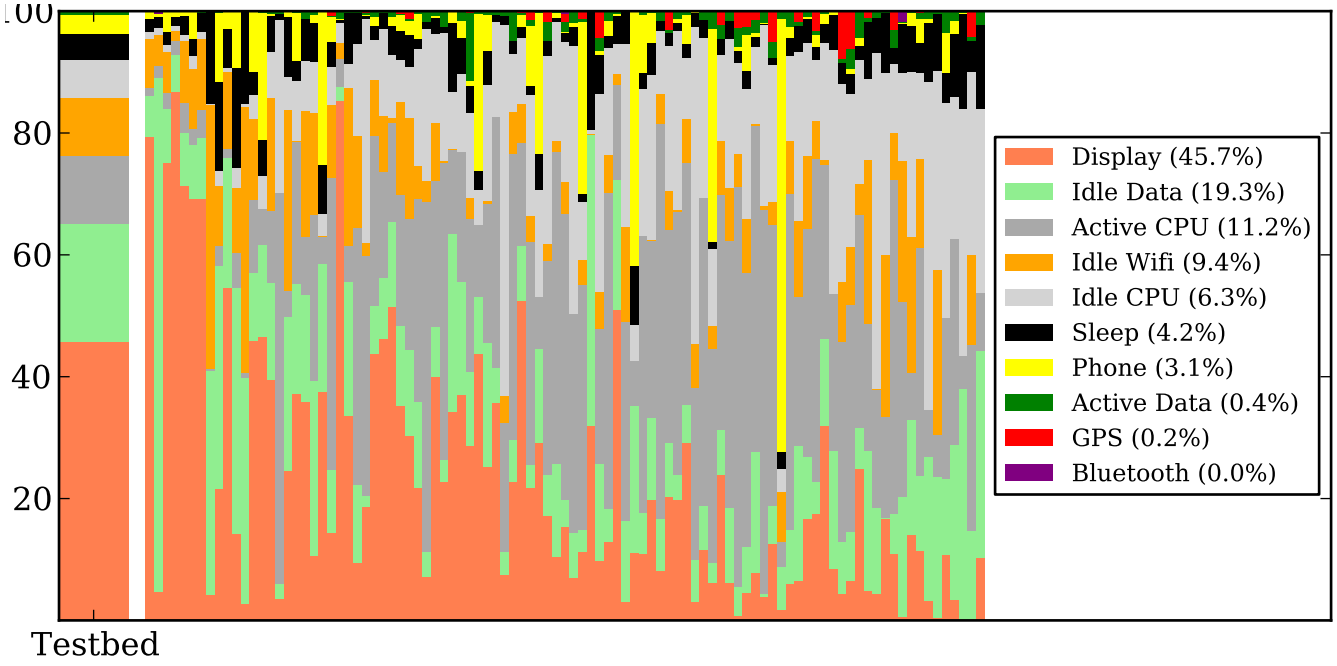

Figure 2: **Power usage by component.** The large bar at left shows an aggregated breakdown for the entire testbed. The participant bars are scaled against the participant with the most energy usage.

of the power consumption properties of their phones and how to control them. And as we bring new hardware onto the testbed, we can repeat power usage experiments to determine differences between smartphone hardware generations.

6.2 Opportunistic Charging

One way that users work around the battery limitations of their smartphone devices is by finding new times and places to charge their phones: plugging in at their desk at work, in the car during their commute, or at home before a long night out. We refer to these charging sessions as *opportunistic* to distinguish them from *habitual* nightly charging. Assuming that many smartphone users encounter plug points throughout the day, engaging in opportunistic charging becomes an additional sign of energy awareness, and understanding opportunistic charging becomes necessary to improving energy management on mobile devices.

6.2.1 Seizing Energy Opportunities

Figure 1 shows that many users engage in opportunistic charging. We define a charging session as opportunistic if is long enough to not be spurious (over 10 minutes) but does not bring the battery to a fully-charged state, indicating that the user disconnected the device before charging could finish. For a representative day during our experiment, of the 245 charging sessions we observed that day, 96 (39%) were opportunistic by this definition. 50 of 95 active participants engaged in opportunistic charging at some point during our experiment an average of once per day.

Opportunistic charging may be a response to an anticipated need for more smartphone battery power: the student who plugs her smartphone in for a brief charge before a night out. Our data also allowed us to examine how many of these opportunistic charging sessions were necessary to bridge the gap to the next full charge. We found that 24 of the 96 (25%) of the opportunistic charges we observed were necessary. We believe that this indicates that participants have responded to their smartphones' battery limitations by engaging in conservative charging behavior, grabbing power whenever possible even if they do not anticipate needing it later.

Combining opportunistic charging combined with the varied rhythms of our participants creates a second interesting effect: at any given point on PhoneLab there is a wide disparity in the amount of power available on different phones. Figure 3 displays the top, bottom, and middle (median) quartiles for a single day on PhoneLab. Only phones that are discharging are shown, which explains the sharp increase between 6 and 10AM as participants end nightly charging cycles. As the graph indicates, there is a high chance that two smartphones that meet have very different battery levels.

6.2.2 Future Experiments

We are not the first to note opportunistic charging patterns [10, 24], but we believe PhoneLab can be used to address several interesting questions raised by

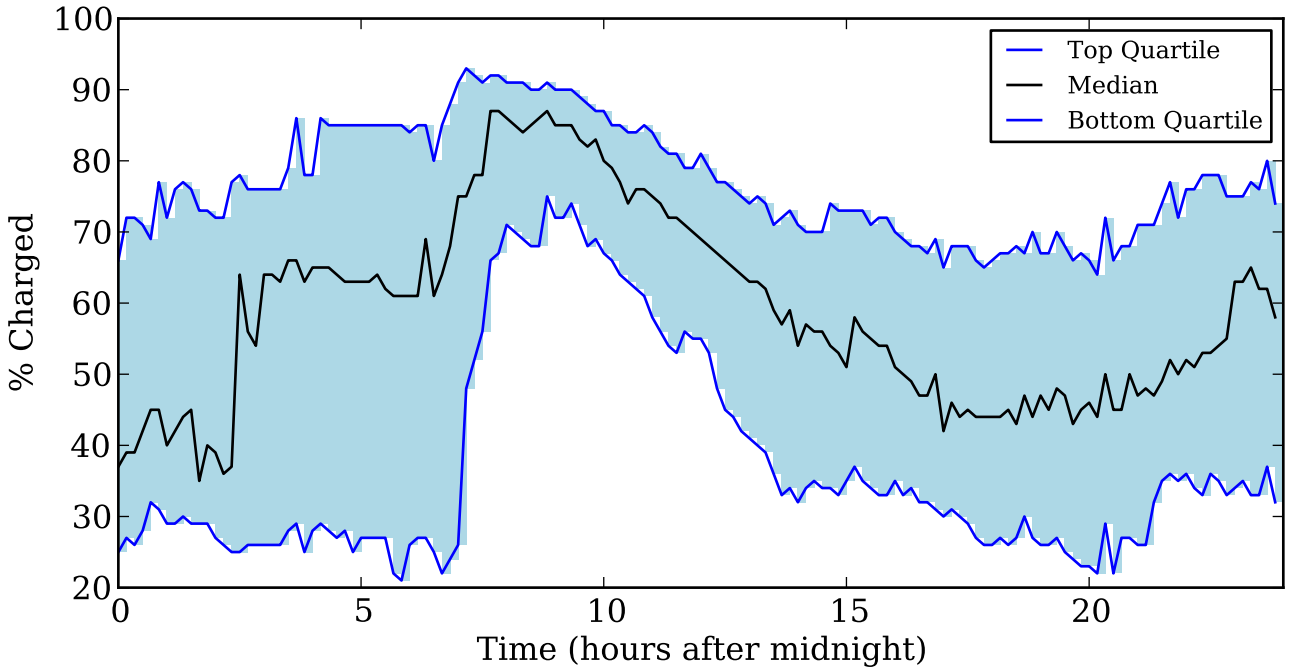

Figure 3: **Charge difference between participants during one day.** The graph plots the top and bottom quartiles as well as the median. A significant spread is present on the testbed throughout the day.

opportunistic charging. First, why do users engage in this practice? By monitoring charging patterns an experiment could prompt users to indicate why they were charging their phone after opportunistic charging sessions. This would help shed more light as to the motivations of smartphone users and their evolving relationship with power.

Second, new protocols might seek to use the increased charging differentials as a result of opportunistic charging to establish a distributed energy market. Phones with spare power and users that engage in opportunistic charge frequently may agree to help another phone with less power conserve its battery level—by switching Wifi access points, providing GPS coordinates or acting as a real for cellular data—in exchange for assistance when the tables are turned. Platform experiments on PhoneLab could evaluate the effectiveness of collaborative mobile energy management on a dense testbed.

6.3 Mobile Network Transitions

Mobile devices like smartphones move through a complex network environment. Providing the illusion of seamless connectivity requires negotiating hand-offs both between Wifi access points and between Wifi and 3G radios. Further exacerbating the situation, users are typically very aware of the device's mistakes. It is clear to them that the phone should give up on the Wifi signal since they are halfway across the parking lot. Unfortunately, it is not so obvious to Android.

6.3.1 *Stuck in the Middle*

We were interested in observing hand-offs between 3G and Wifi and found many in the dataset collected by our usage experiment. Since the Android `ConnectivityService` frequently switches network interfaces for exploration purposes, we have defined a transition as two one-minute or longer sessions on different interfaces separated by less than one minute. We further limit ourselves to cases where we received a location update during the transition.

Figure 4 plots the location of transitions that occurred on or near SUNY North Campus. We notice that many cluster in expected locations: near the entrance and exits of buildings where participants are likely to be moving from campus Wifi to Sprint 3G.

6.3.2 *Future Experiments*

Mapping and remembering where network transitions take place may allow the Android platform to conduct the transitions more smoothly. When it observes a participant heading towards a transition area, it may decide to be more aggressive about abandoning the current interface and less inclined to hang on to a weakening connection. Platform experiments using transition data generated by PhoneLab participants would use this crowd-sourced transition map to adjust the policies of the `ConnectivityService` component. Improvements in hand-offs could be benchmarked against unmodified PhoneLab phones.

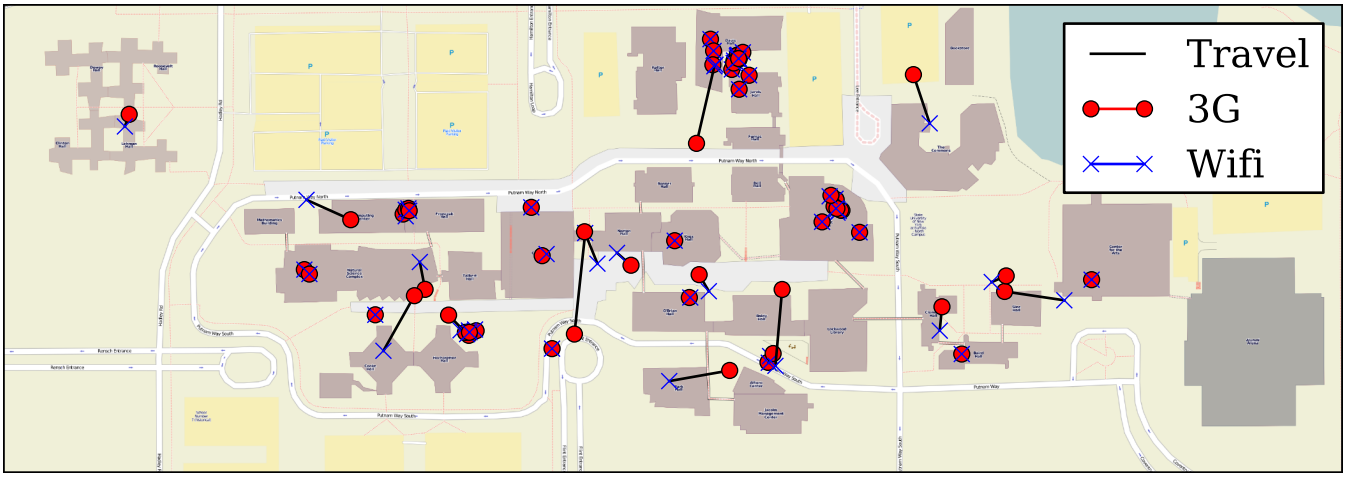

Figure 4: **3G to Wifi transition locations.** The map indicates that there are several common areas where network hand-offs occur.

Device	First App	Second App	%
7d552e	Skyvi	GO SMS	91
a991b7	Dolphin Browser Engine	Dolphin Browser	89
028e2e	Phone	Google Voice	81
2a0185	Phone	Google Voice	73
933eca	Genie Widget	Browser	68
c16dc2	Sina Weibo	MIcons Project	60
11edfb	Genie Widget	Browser	58
3cd6b1	Google Search	Browser	57
b9f595	Google Search	Browser	53
00b5ae	Phone	Google Voice	53
a991b7	GO Contacts	Phone	53
2a0185	Contacts	Google Voice	52
3ddb92	Phone	Google Voice	50

Table 6: **Application transitions.** The table shows the percentage starts of Second app when First app was already started on a user device.

6.4 Application Transitions

In any computing system, studying its workload is a key to improve the overall performance. By analyzing how applications behave, we can observe common usage patterns that arise and optimize relevant components to exploit those patterns. This is even more crucial for smartphones since they are resource contrained and still expected to run resource-heavy applications like games.

Joint use of multiple applications is potentially one interesting usage pattern. This might arise in scenarios such as checking social networking applications in series and playing different games in one sitting. If there is a group of applications with a high correlation in usage, then a phone OS might treat them together and apply the same scheduling policy, i.e., the applications can be loaded into the memory together to reduce the latency of context switch. A previous study has looked at a similar usage pattern by analyzing a network traffic trace [31]. Our study described below is a more direct analysis based on recorded application usage events, hence finer in granularity.

6.4.1 *Jointly-Used Applications*

In order to understand which applications are used together, we calculate a *transition* probability from an application a to another application b within a session. A session is defined as the time between a screen unlock event and the subsequent screen lock event. To calculate a transition probability, we count the number of start events for a, i.e., *start(a)*, and the number of start events for b that occur after an a's start event within the same session, i.e., *start($a \rightarrow b$)*. Our transition probability is $\frac{start(a \rightarrow b)}{start(a)}$.

Table 6 shows some example pairs of applications with high correlations of being started together. We only show application pairs that are used more then 20 times together by a single user. The table demonstrates that for some applications there is indeed a significant correlation. For example, many users use relevant applications together such as the Phone app and Google Voice or Contacts, Google Search and Browser, etc. The fact that such high correlations exist points to the possibility of jointly scheduling these applications as a unit.

6.4.2 *Future Experiments*

We expect that PhoneLab will enable many kinds of optimization techniques that adapt to different usage patterns. As our study shows and previous studies have pointed out [12, 25], there is significant diversity in smartphone usage across different users. Thus, future optimization techniques will need to adapt to each user's behavior rather than relying on commonalities.

As with energy usage experiments, application usage studies need to be re-validated; as new software and hardware become available, user behavior will in-

Gap	GPS		Network	
(min)	Hits	%	Hits	%
5	4742	1.4	71,444	10.3
10	5486	1.6	79,877	20.5
15	6064	1.8	85,091	21.9
20	6450	1.9	88,990	22.9
Total	340,084		388,800	

Table 7: **Coordinate sharing counts.** We discovered few opportunities to reduce GPS usage through coordinate sharing.

evitably change over time. PhoneLab gives an opportunity to study such changes.

6.5 Location Sharing

Location tracking has been the focus of many recent mobile systems research efforts [17, 21, 18]. Given PhoneLab's density it provides an ideal proving ground for new location techniques.

One promising idea is to reduce GPS usage through local coordinate sharing. Before taking a reading, a smartphone will use a local communication protocol to determine whether a device nearby has recently obtained a GPS reading. If so, and if that reading is sufficiently recent and accurate, the device may not have to turn on its GPS at all, reducing latency and saving power. As a more complex variant, multiple readings from different nearby devices at different times could be combined to produce a new, more accurate reading.

Previous work [21] has included the more limited form of GPS coordinate sharing into their location management system. However, given that the evaluation was done using five phones traveling in a backpack together, it is likely that their experiment evaluated nearly the best case for this approach. Given that PhoneLab provides access to a dense set of participants, it would seem a good fit for determining whether GPS coordinate sharing has merit.

6.5.1 *Can Smartphones Share?*

To answer this question we process the data from our usage experiment. In order for GPS coordinate sharing to take place, several conditions must be true. First, the two phones must be nearby in time and space. We use the location updates logged by our experiment to determine this. Note that we assume that the participant remained at the place where they acquired the location information for the time necessary to participate in sharing. While this assumption clearly does not hold in any case, mobility would not necessarily bias our result in either direction. For every false positive, a participant that did not remain where they obtained the last location fix, there may be a corresponding false negative: a participant that moved and ended up nearby another participant without our knowledge.

The second requirement is that the first phone acquire a location with sufficient accuracy to satisfy the second device. We use the accuracy estimation provided by the Android location manager to determine this.

Table 7 summarizes our negative result: we find few opportunities for GPS sharing on our testbed. The table is interpreted as follows: of the 3,400,084 total GPS updates performed during the three week study, only 4742 could have been satisfied with a less than five minute old coordinate from a nearby device of equal accuracy. Figure 5 shows the location of coordinate sharing opportunities on campus. Many cluster around areas where students travel.

Obviously the effectiveness of collaborative protocols increases as more phones participate in them, meaning that this result may be interpreted as indicating that GPS coordinate sharing would require higher densities to be effective. However, as a point of comparison we present numbers for network coordinate sharing in Table 7. The interpretation of the data is the same except in this case the phone is using a network provider instead of GPS. Given that the power overhead for obtaining a network location is likely to be similar to retrieving one from a nearby phone, this is not necessarily a positive result, but it does help put the low GPS sharing numbers in context.

6.5.2 *Future Experiments*

We expect location management to continue to be an active area of research and PhoneLab to play a major role. With the ability to change or replace the Android location service, new approaches to location estimation, sharing, and tracking can be evaluated without altering installed applications. An area of future study, however, would be ways of determining whether new location approaches are experienced as equivalent or better by PhoneLab participants.

In addition with the rise of new phone-to-phone communication protocols such as Wifi Direct [8][3], we expect PhoneLab to host new research that harnesses the power of device interaction. We are working on establishing low-overhead methods for discovering and counting nearby PhoneLab participants in order to explore participant social networks.

7. CONCLUSIONS

We introduce PhoneLab, a new large-scale programmable smartphone testbed hosted by SUNY Buf-

[3]Despite no official support for Wifi Direct on the Nexus S by the AOSP, we were able to enable this feature and will distribute it on future platform images. The next generation of PhoneLab phones should have official AOSP support for this useful feature.

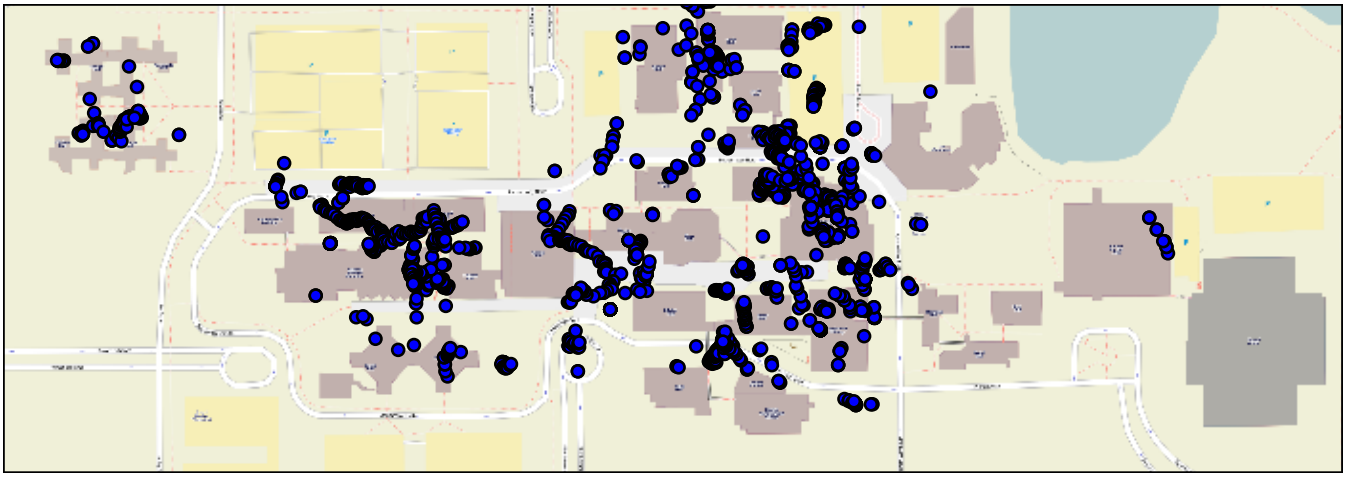

Figure 5: **Location of GPS sharing opportunities.**

falo. PhoneLab provides a mixture of features designed to enable the next generation of mobile systems research. Through five vignettes based on data collected by a preliminary usage study, we have demonstrated that PhoneLab is useful and usable.

Acknowledgments

A number of other SUNY Buffalo students have worked on the PhoneLab testbed, including Micheal Benedict, Vinu Charanya, Rishi Gupta, Jay Inamdar, Manoj Mylapore, Mitchell Nguyen, and Sean Zawicki.

8. REFERENCES

[1] Android 4.1, Jelly Bean. http://www.android.com/about/jelly-bean/.

[2] Broadcom BCM4751 Integrated Monolithic GPS Receiver. http://www.broadcom.com/products/GPS/GPS-Silicon-Solutions/BCM4751.

[3] Emulab. http://emulab.net/.

[4] Livelabs. http://www.livelabs.smu.edu.sg/.

[5] Nexus S 4G from Google. http://www.android.com/devices/detail/nexus-s-4g-from-google.

[6] OpenCirrus. https://opencirrus.org/.

[7] Planetlab. http://planet-lab.org/.

[8] Wi-Fi Direct. http://developer.android.com/guide/topics/connectivity/wifip2p.html.

[9] A. I. Avetisyan, R. Campbell, I. Gupta, M. T. Heath, S. Y. Ko, G. R. Ganger, M. A. Kozuch, D. O'Hallaron, M. Kunze, T. T. Kwan, K. Lai, M. Lyons, D. S. Milojicic, H. Y. Lee, Y. C. Soh, N. K. Ming, J.-Y. Luke, and H. Namgoong. Open Cirrus: A Global Cloud Computing Testbed. IEEE Computer, 43(4):35–43, Apr. 2010.

[10] N. Banerjee, A. Rahmati, M. D. Corner, S. Rollins, and L. Zhong. Users and Batteries: Interactions and Adaptive Energy Management in Mobile Systems. In Proceedings of the 9th International Conference on Ubiquitous Computing (UbiComp), 2007.

[11] Business Insider. Google Is Activating 1.3 Million Android Devices On A Daily Basis. http://goo.gl/SZHqV.

[12] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin. Diversity in Smartphone Usage. In Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, 2010.

[13] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner. Android Permissions: User Attention, Comprehension, and Behavior. In Proceedings of the Eighth Symposium on Usable Privacy and Security (SOUPS), 2012.

[14] J. Huang, Q. Xu, B. Tiwana, Z. M. Mao, M. Zhang, and P. Bahl. Anatomizing Application Performance Differences on Smartphones. In Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys), 2010.

[15] International Data Corporation. Worldwide Mobile Phone Growth Expected to Drop to 1.4Despite Continued Growth Of Smartphones. http://goo.gl/ROLY0.

[16] S. Isaacman, R. Becker, R. Cáceres, M. Martonosi, J. Rowland, A. Varshavsky, and W. Willinger. Human mobility modeling at metropolitan scales. In Proceedings of the 10th international conference on Mobile systems, applications, and services, MobiSys '12, pages 239–252, New York, NY, USA, 2012. ACM.

[17] D. H. Kim, Y. Kim, D. Estrin, and M. B. Srivastava. SensLoc: Sensing Everyday Places and Paths Using Less Energy. In Proceedings of the 8th ACM Conference on Embedded Networked

Sensor Systems (SenSys), 2010.

[18] M. B. Kjaergaard, S. Bhattacharya, H. Blunck, and P. Nurmi. Energy-Efficient Trajectory Tracking for Mobile Devices. In Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services (MobiSys), 2011.

[19] Y. Lee, Y. Ju, C. Min, S. Kang, I. Hwang, and J. Song. Comon: cooperative ambience monitoring platform with continuity and benefit awareness. In Proceedings of the 10th international conference on Mobile systems, applications, and services, MobiSys '12, pages 43–56, New York, NY, USA, 2012. ACM.

[20] S. Nath. Ace: exploiting correlation for energy-efficient and continuous context sensing. In Proceedings of the 10th international conference on Mobile systems, applications, and services, MobiSys '12, pages 29–42, New York, NY, USA, 2012. ACM.

[21] J. Paek, J. Kim, and R. Govindan. Energy-Efficient Rate-Adaptive GPS-Based Positioning for Smartphones. In Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys), 2010.

[22] L. Peterson, T. Anderson, D. Culler, and T. Roscoe. A Blueprint for Introducing Disruptive Technology into the Internet. SIGCOMM Comput. Commun. Rev., 33(1):59–64, Jan. 2003.

[23] F. Qian, K. S. Quah, J. Huang, J. Erman, A. Gerber, Z. Mao, S. Sen, and O. Spatscheck. Web caching on smartphones: ideal vs. reality. In Proceedings of the 10th international conference on Mobile systems, applications, and services, MobiSys '12, pages 127–140, New York, NY, USA, 2012. ACM.

[24] A. Rahmati, A. Qian, and L. Zhong. Understanding Human-Battery Interaction on Mobile Phones. In Proceedings of the 9th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI), 2007.

[25] A. Shye, B. Scholbrock, and G. Memik. Into the Wild: Studying Real User Activity Patterns to Guide Power Optimizations for Mobile Architectures. In Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2009.

[26] I. Trestian, S. Ranjan, A. Kuzmanovic, and A. Nucci. Measuring Serendipity: Connecting People, Locations and Interests in a Mobile 3G Network. In Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference (IMC), 2009.

[27] I. Trestian, S. Ranjan, A. Kuzmanovic, and A. Nucci. Taming the Mobile Data Deluge with Drop Zones. IEEE/ACM Trans. Netw., 20(4):1010–1023, Aug. 2012.

[28] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury. No need to war-drive: unsupervised indoor localization. In Proceedings of the 10th international conference on Mobile systems, applications, and services, MobiSys '12, pages 197–210, New York, NY, USA, 2012. ACM.

[29] G. Werner-Allen, P. Swieskowski, and M. Welsh. MoteLab: a Wireless Sensor Network Testbed. In Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN), 2005.

[30] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar. An Integrated Experimental Environment for Distributed Systems and Networks. In Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI), 2002.

[31] Q. Xu, J. Erman, A. Gerber, Z. Mao, J. Pang, and S. Venkataraman. Identifying Diverse Usage Behaviors of Smartphone Apps. In Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference (IMC), 2011.

[32] T. Yan, D. Chu, D. Ganesan, A. Kansal, and J. Liu. Fast app launching for mobile devices using predictive user context. In Proceedings of the 10th international conference on Mobile systems, applications, and services, MobiSys '12, pages 113–126, New York, NY, USA, 2012. ACM.

[33] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, and L. Yang. Accurate Online Power Estimation and Automatic Battery Behavior Based Power Model Generation for Smartphones. In Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis (CODES/ISSS), 2010.