Jinghao Shi, Edwin Santos, and Geoffrey Challen *University at Buffalo*

Editors: Sharad Agarwal and Marco Gruteser

WHY AND HOW TO USE PHONELAB

While smartphone app marketplaces have enabled large-scale app-level experimentation, medium-scale experimentation with the platform code implementing the app interface and providing core device services remains difficult for academic researchers. But this is where many of the ideas currently being explored by the mobile systems community must be evaluated—including new networking protocols, security and privacy mechanisms, storage abstractions, and energy management strategies. To enable these experiments, we built and are operating PhoneLab, a 175-smartphone testbed where real users run experimental Android platform builds on their primary devices. We are eager to make PhoneLab useful to the mobile systems community. To aid in this effort, this article discusses why PhoneLab might be useful for your research and provides an overview of how to use the testbed, including examples drawn from our group's current projects.

The world's two billion mobile smartphones together constitute the largest, most powerful, and most widely distributed system ever built. In many ways, the incredible success of the smartphone is built on decades of research by the mobile systems community. But new challenges and opportunities alike await next-generation mobile apps and systems that attempt to harness this increasingly powerful and pervasive network, and researchers must continue to find ways to make these ubiquitous devices even more effective.

An important part of the smartphone success story has been the emergence of mobile app marketplaces, such as the Google Play Store. These distribution channels have played a significant role in democratizing mobile app development by allowing a single developer to easily reach a global audience. In the same way, these marketplaces are also playing an increasingly important role in enabling large-scale mobile systems research. Researchers can now make experimental apps available to billions of users, and rely on the fact that even a tiny fraction of citizen scientists willing to participate in their project can amount to a representative, statistically significant and globally distributed sample of thousands or even tens of thousands of users. As an example, the Netalyzr tool[1] distributed by ICSI has been downloaded 24,000 times by users in 120 countries. And if the app provides a tangible benefit it may attract an enormous number of users, like the Carat[2] personalized energy management app, which was installed more than 750K times. We expect app marketplaces to continue to play a vital role in allowing the mobile systems community to evaluate ideas at scale.

Photo, istockphoto.com

But what about the smartphone platform itself? Platform codebases are the millions of lines of code that are responsible for performing crucial tasks such as determining the device's location, choosing which available network to use, and managing limited resources such as energy – tasks of obvious interest to the mobile systems research community. Smartphone platforms cannot be altered via apps distributed by app marketplaces, and they also limit the access of these apps to useful information and device capabilities – sometimes for security or privacy reasons, but at other times simply because the information is considered too difficult or not useful enough to expose.

Microsoft's and Apple's platform codebases are proprietary, making experimentation impossible without insider support. But while the Android Open Source Project (AOSP) makes it possible to create fully functional custom Android platform images for select devices, many roadblocks still await researchers attempting to deploy modified platform images at even modest scale. Experimental participants must be recruited and, if their behavior is to be representative, convinced to swap their primary smartphone for (and port their number to) a new device. Data collection tools must be built and tested, and the platform changes themselves must be developed and deployed. These experimental barriers are making it difficult for today's academic mobile systems research community to have a real impact on the most important parts of the most successful mobile systems platform: the smartphone.

To simplify smartphone platform experimentation we are operating PhoneLab, a

[1] http://netalyzr.icsi.berkeley.edu/
[2] http://carat.cs.berkeley.edu/

smartphone platform testbed (https://www.phone-lab.org). PhoneLab is both a group of people and an ongoing community effort to maintain a well-instrumented version of the AOSP useful for research purposes. PhoneLab consists of 175 participants that run custom Android platform images on their primary smartphone. Our PhoneLab platform image is open to the mobile systems community to instrument and modify. We provide the participants, reliable data collection tools, and increasingly well instrumented AOSP platform sources. You provide the exciting research questions and new ideas that help us learn more about and improve these ubiquitous mobile devices.

In this article our goal is to provide an overview of PhoneLab's capabilities in the hope that you will find it helpful in enabling your research studies. We start with some basic information about the testbed, describe the process of performing a PhoneLab experiment, and then use some examples drawn from our ongoing projects to provide tangible examples of what PhoneLab can do.

PARTICIPANTS, PLANS, PHONES

PhoneLab is a smartphone platform testbed located at the University at Buffalo (UB). 175 participants – primarily UB, students, faculty, and staff – receive discounted Sprint service and a Google Nexus 5 smartphone to use as their primary device. In exchange, they run a modified Android platform image containing instrumentation and potentially novel changes and features.

PhoneLab participants were originally required to be UB affiliates. Recently we have relaxed that policy, but almost all participants are still UB students, faculty and staff. We have not yet polled the demographics of our 2014-15 participants, but a 2013-14 survey indicated that PhoneLab participants were well distributed between genders and across age brackets. While they may not be entirely representative of the billions of smartphone users worldwide, or even the millions of those located in the United States, the testbed is not just a bunch of computer science undergraduates or Ph.D. students. Because PhoneLab has always been open to any UB affiliate willing to agree to our terms of service and make the initial payment, we have no way to control participant demographics. We do know that we have participants in many different departments on campus, providing a reasonable level of on-campus spatial coverage which can be important to certain studies.

To simplify testbed administration, we operate PhoneLab as a Sprint corporate-liable service plan. PhoneLab participants pay $45 per month for an all-you-can-eat Sprint service plan which includes unlimited voice, messaging and data. Technically resources are pooled and capped over all participants, but in practice the caps are generous enough to never limit participant consumption. Cooperation from Sprint allows us to create an attractive price point for our participants, while they offload all payment processing and customer support to us. Participants purchase service directly from PhoneLab in six-month installments, with invoice and payment processing done using Square. Sprint's coverage in the Buffalo area is fairly good, including 4G LTE mobile data coverage in many locations.

Currently all participants use the Google Nexus 5 smartphone running a custom Android platform image based on the Android Open Source Project (AOSP). Currently our PhoneLab branch is based on AOSP version 5.1 "Lollipop," an upgrade from version 4.4.4 that participants received in mid-September 2015. In previous academic years participants use the Samsung Nexus S (2012-13), Samsung Galaxy Nexus (2013-14), and Google Nexus 5 (2014-15). Compared to these previous smartphones the Nexus 5 seems far superior, and has helped address battery lifetime and network quality complaints registered by first- and second-year participants.

We are aware that in some ways PhoneLab represents an unusually homogeneous environment for smartphone experiments. All participants live in the same geographic area, carry the same smartphone model, and have service provided by the same carrier. While this homogeneity helps simplify testbed management, it can make certain experiments difficult to perform. For example, our group has recently begun working on several projects intended to help developers cope with performance differences caused by the fact that Android runs on 18K devices that are attached to networks with orders of magnitude performance variation. Given its device and location homogeneity, PhoneLab is a relatively poor environment to investigate this challenge.

EXPERIMENTAL PROCESS

To use PhoneLab, you first download and modify our fork of the AOSP. Next, you work with PhoneLab administrators and developers to test your changes. After your changes are validated to be safe and effective, they are distributed to participants and data collection begins. Finally, after verifying that your experiment has been cleared for human subjects compliance, you will be provided the data that it generated. Here we review each step in the process in more detail.

Instrumenting or Modifying the PhoneLab Platform:

Experimenting on PhoneLab begins with making changes to the Android platform. PhoneLab hosts a mirror of the AOSP allowing experimenters to use Google's Android well-documented platform build process as well as their repo and Gerrit source control tools. Only the repoURL used during the process of downloading the platform sources is different – the rest of the workflow is identical.

We divide PhoneLab experiments into two categories: instrumentation and modification. Instrumentation consists of additional logging that records but does not alter how Android operates. Modifications modify Android itself, possibly by altering existing services or adding new features. Of course, modification experiments will usually need to generate data to evaluate their changes, and so will require instrumentation as well.

Instrumentation changes are usually fairly well-contained and unlikely to break participant smartphones, and so we fast-track approval of these experiments. Instrumentation is also an incremental process, and new useful logging messages are preserved in future PhoneLab builds. In contrast, modifications usually create larger patches, can be harder for our team to understand, and may break things. As a result, modifications are inspected more carefully before being tested and released, and require more coordination with the PhoneLab team. Unlike instrumentation, most modification experiments will only

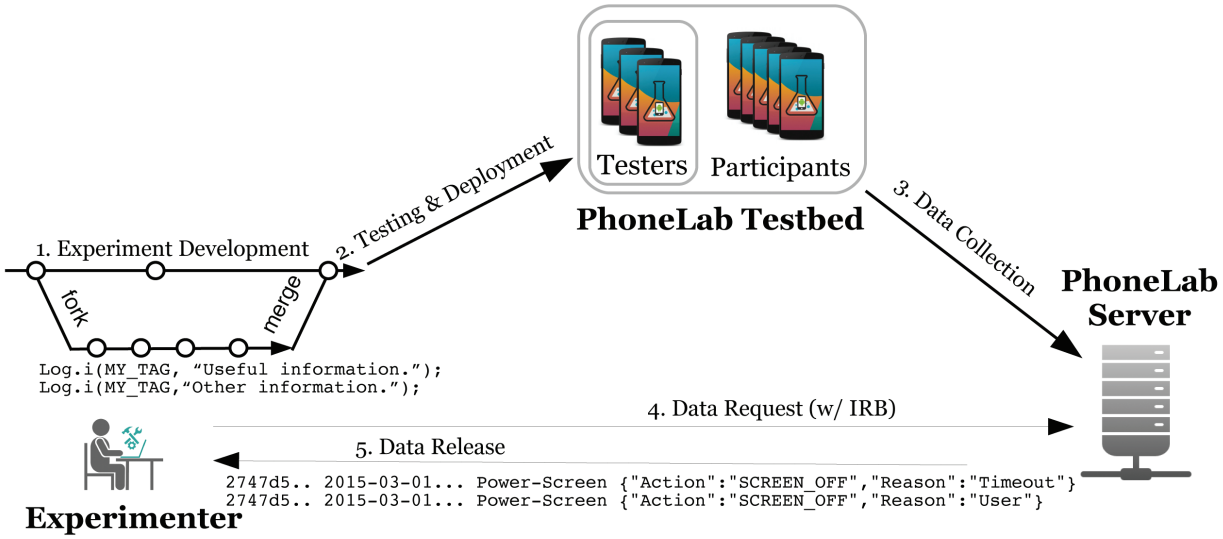

FIGURE 1. Experimentation on PhoneLab: Experimenters begin using PhoneLab by locally developing and testing their platform changes. Once they are ready, their experiment is tested by a group of PhoneLab developers and then eventually deployed to PhoneLab participants. Researchers can request data generated by their experiment or common PhoneLab instrumentation. Data requests specify tags and date ranges and must document that the experiment has been reviewed for human subjects compliance.

be run for a limited period of time, at which point their changes will be reverted.

PhoneLab instrumentation is done using the familiar Android logcat interface, which allows logging in Java, C++ and C files – all the primary languages used by the AOSP. Standard logcat messages include a timestamp, log level (info, warning, etc.), process and thread identifiers, a tag identifying the type of message, and a message payload. The PhoneLab platform includes data collection software that captures all logcat messages generated by the device, caches them locally, and then uploads them to our server when the smartphone is plugged in and charging and has network access. Log post-processing splits the combined logs into per-device per-tag files and adds a device identifier to each log line.

We have taken two steps to help organize and facilitate PhoneLab instrumentation. First, we have established a convention that log tags include a category and subcategory which together hint at their purpose. Examples include Network-Telephony, Network-Wifi, Power-Battery and Power-Screen. Log tags also include an institution identifying who contributed the instrumentation. Standardizing the format of log tags is important since log tags are how researchers identify requested PhoneLab data at the conclusion of their experiment. Second, we require in almost all cases that log messages be formatted in JSON for easy cross-language deserialization. PhoneLab provides experimenters with helper functions to validate that log messages contain well-formatted JSON. We also encourage developers to wrap all logging in a try-catchblock for safety.

At present our PhoneLab platform sources already include a variety of useful information. Our website (https://www.phone-lab.org) contains more information about all the tags that we are currently collecting, but they record commonly needed information such as location updates; battery level changes; network transitions, measurements, and signal strength changes; and app installation and removal. Some of this information is available to apps through Android's API, but we have found it easier to embed it in our platform image. In other cases, PhoneLab instrumentation reflects our own research interests. For example, we have instrumented the SQLite database library used by many apps, the system call interface, and also added a variety of tags to certain core user interface elements. In all these cases, we are also recording information not typically available to apps.

Initially we had concerns about the log volume produced by instrumentation on frequently executed code paths. But we have found Android's Linux-based logging infrastructure to be surprisingly robust and low overhead, allowing us to add log messages to almost any part of the system. For example, although this was not a well thought out piece of logging, at one point we were successfully recording every screen refresh without a noticeable performance impact. However, there are certainly places where logging would generate too much output, and the impact of new instrumentation on logging volume is something we note during testing. Frequently a more intelligent approach to logging can reduce overhead while maintaining fidelity.

Eventually, with additional contributions from the mobile systems community, we hope to achieve a thoroughly instrumented Android platform source generating almost all commonly needed measurements. Once we reach that point, many experimenters may discover that their logging needs are met by instrumentation that has already been contributed by other researchers and they can skip ahead to requesting existing data. Researchers are also free to utilize our existing instrumentation for their own purposes, regardless of whether they run their platform on PhoneLab or not, and we

are happy to provide our server-side tools to anyone who wants to franchise PhoneLab at their own institution.

Testing and Distributing Experiments: Once a researcher has completed their experimental changes, we perform three rounds of testing before changes are distributed to participants. First, we use Jenkins to do continuous integration testing to catch errors that break the build process. Second, Jinghao Shi, the PhoneLab team member overseeing the distribution process, tests it on his own device. Third, we distribute changes to a half-dozen PhoneLab developers who are also participants. After one or two days have passed, a preliminary data archive is made available to the experimenter so that they can confirm that PhoneLab is collecting the data they need, and so that we can estimate how much data the experiment is generating. At the same time, developers are polled to ensure that they have not noticed any regressions during the testing period. Developer beta testing has successfully caught several bugs that would have irritated participants, including a change that broke the Android alarm clock and logging bugs in SQLite that broke almost every Android app. In both cases these regressions were quickly identified and addressed.

Note that we also rely on developers to identify more subtle regressions, such as poor interface responsiveness and increased energy consumption, rather than relying on automated tools. So far this approach has worked well, combined with proactive engagement with experimenters over issues such as energy management. In cases where we anticipate an experiment may produce high energy overhead, we ask researchers to self-limit their experiment using Android's built-in energy measurement tools. For example, in an upcoming high-overhead network measurement experiment, researchers have agreed to limit the energy consumed by their modifications to 10% of the device's battery capacity.

After an experiment clears the testing process, we distribute it to participants using the same over-the-air (OTA) platform update capabilities provided by Android devices and commonly used by wireless carriers. Once a platform update is stored

FIGURE 2. An Overview of PhoneLab

on the device, Android provides an API to initiate the process of rebooting into recovery mode, applying the selected update, and the rebooting again into the modified platform image. Built-in PhoneLab tools handle detecting when updates are available, transferring them to the device, and determining when to apply the update safely without disturbing the participant. Frequently even PhoneLab developers miss the update process and don't realize that they are running a new image until they are asked to comment on its stability. Once smartphones reboot into the new platform image, new instrumentation begins generating data or new features provided by platform modifications are activated.

PhoneLab experimentation can also include an app along with platform changes. Depending on the experiment, it may be more natural to implement parts of it as an app, and in other cases the app provides the user interface to a new platform interface or features—such as in the Jouler experiment described below. Experimenters are free to distribute their app by asking participants to install it from the Google Play Store. Alternatively, we can bundle their app into the platform update itself, which ensures that it is installed on every PhoneLab smartphone and allows it to run with elevated permissions if needed. In both cases, future updates to the app that do not require platform modifications can be distributed using the Google Play Store.

Given that Android platform instrumentation and modification is a slow process, our goal is to run PhoneLab experiments as soon as they have passed testing. In many cases multiple data instrumentation experiments can be combined and run concurrently or alongside a single modification experiment. However, multiple platform modifications may conflict with each other and need to be run serially. We determine this on an ad-hoc basis during experiment testing.

Data Request and Analysis: In the best case, researchers may find that PhoneLab has already been logging the data they need for their analysis. In other cases, additional instrumentation will need to be run on the testbed for a period of time to generate enough data for analysis. In either case, to access collected PhoneLab data researchers must provide documentation that their experiment has been approved for human subjects compliance—most commonly by their university's Institutional Review Board (IRB). (Note that experiments must only be approved by a researchers local IRB and do not require further IRB review at the University at Buffalo.) Given the ability of platform changes to record an enormous amount of sensitive data about participant's behavior, the IRB is a critical safeguard to ensure that this data is used and handled appropriately. However, under PhoneLab's own testbed-specific IRB, additional IRB approval is only required when the data is used, not before it is collected.

With IRB approval researchers may request time-delimited archives for a set of tags collected from all PhoneLab devices by emailing PhoneLab administrators. Currently we do not accept blanket requests for all logged information. By default we limit tag requests to the tags that we have

documented as part of our instrumentation, but researchers may request additional tags if we determine that they are useful and the IRB approves them for release.

PhoneLab data is structured to allow researchers to import and process the data using a wide variety of log-processing tools, with the choice specific to their expertise and the type of analysis they are performing. Given that individual nature of this decision and the common format of the PhoneLab data, we have yet to devote significant resources to developing a common post-processing toolchain.

EXAMPLE EXPERIMENTS

To make the process of using PhoneLab more concrete, we next present two examples of ongoing experiments that our research group, blue (https://blue.cse.buffalo.edu), is performing using PhoneLab. Our examples include one modification experiment and one instrumentation experiment.

SQLite Logging:

SQLite is a widely deployed embedded database library that is heavily used on Android. Android both uses SQLite internally and provides interfaces encouraging apps to use SQLite to persist their own structured data. Unsurprisingly, query workloads experienced by mobile app embedded databases are unlike those experienced by database servers supporting websites or big data applications. And the performance requirements are also quite different between interactive mobile apps trying to conserve energy and transaction-processing systems where throughput is key and energy only a second-order concern.

To measure SQLite behavior on Android we instrumented the SQLite library to record all queries and deployed our changes on PhoneLab. To protect participants privacy, our instrumentation strips out as much personally identifying data as possible and records only hashes of prepared statement arguments. We have completed an initial investigation using publicly available data released by 11 PhoneLab developers who agreed to release complete trace data from their phones for the month of March 2015. We collected 254 participant days of data which contained roughly 45M queries, and an analysis of this initial dataset appeared at TPCTC in August 2015.[3] We are continuing our experiment on the full testbed and have begun talks with the Transaction Processing Performance Council (TPC) on using them as the basis for a new database benchmark aimed at mobile systems: TPC-MOBILE.

This instrumentation experiment provides a good example of PhoneLab's ability to use the platform to "get underneath" apps by instrumenting commonly used libraries and interfaces, allowing us to record information about all apps used by participants without modifying any.

We have used this capability in multiple other ways, including to study filesystem access patterns (by modifying the bionicC library) and user interface behavior (by modifying Android's base UI elements).

Jouler:

Smartphone energy management has been a continual challenge. Poor battery lifetimes may frustrate users, and wasted energy represents a missed opportunity to improve performance or otherwise enhance the user experience. Significant improvements in energy management would not only help satisfy users, but would also help support new categories of energy-hungry apps that current smartphones struggle to support.

We began studying smartphone energy management by using PhoneLab to record detailed energy consumption information in the first year that the testbed was publicly available. Our data confirmed what earlier studies had shown: that large variations exist between smartphone users and apps in terms of energy consumption and charging habits. As a result, it is difficult or impossible for a single energy management policy embedded in the platform to meet the needs and expectations of all smartphone users.

Instead, we rearchitected energy management on Android by applying the classic systems design principle of separating policy from mechanism. Our system, which we call Jouler, creates a new Android API allowing certain apps called energy managers to manage overall and per-app energy consumption on behalf of users. By exposing existing energy control mechanisms that were previously only available inside the platform, Jouler allows a wide variety of new energy management policies to be deployed using the same app marketplaces used to distribute other Android apps.

We have performed a small-scale test of Jouler's platform modifications on PhoneLab developers and achieved promising results: six out of seven participants were able to increase their battery lifetime using one of three new energy management policies we provided. We are in the process of distributing and evaluating this modification on the entire PhoneLab testbed. As Jouler shows, modification experiments frequently begin with data collection required to design new approaches. In this case, because Jouler's modifications added a new feature, we also needed to distribute an app to participants during the study to take advantage of the new Jouler interface. Existing PhoneLab battery level logging was used to benchmark energy consumption during the modification experiment to determine the impact of the new policies enabled by Jouler.

OTHER SMARTPHONE EXPERIMENTATION TOOLS

We believe that PhoneLab is the only public testbed allowing researchers to modify a smartphone platform codebase on hundreds of real devices. The LiveLabs[4] testbed at Singapore Management University is the most similar facility. An ambitious project far exceeding PhoneLab in scale, LiveLabs is an urban lifestyle innovation platform using thousands of participants to explore emerging mobile computing scenarios. LiveLabs benefits from robust government support and a large engineering team, but is more focused on industrial experimentation rather than academic research.

A variety of projects have attempted to perform experimentation using apps, either to directly collect data of interest to researchers (MobilLab http://mobilab.eecs.umich.edu/), as a platform for more generalized experimentation (Seattle https://seattle.poly.edu/html/), or by providing

[3] https://blue.cse.buffalo.edu/papers/tpctc2015-pocketdata/
[4] http://centres.smu.edu.sg/livelabs/

library support for experimentation (MobiLyzr http://mobilyzer-project.mobi/). As we have acknowledged, app marketplaces are powerful new tools for researchers interested in doing large-scale experimentation. However, the natural need for platforms to isolate apps and protect user privacy sometimes prohibit useful data collection and it is difficult for these approaches to prototype new platform features. Both of the example PhoneLab experiments we present above could not be distributed using app marketplaces.

SUSTAINABILITY AND FUTURE

PhoneLab is unlike other systems testbeds such as PlanetLab, EmuLab, and MoteLab in that it involves actual human participants rather than just machines. Initially this presented a sustainability problem, as our original plan recruited participants using expensive and unscalable financial incentives. However, we have taken several steps to improve the testbed's sustainability in hopes that we can continue to operate this resources for the mobile systems community for years to come with limited additional funding.

The main change was to the incentive model. In previous years, first-year participants were provided with one year of free service as an incentive to join the study, but this incentive proved ineffective, expensive, and actually may have damaged PhoneLab's early representativeness. Rather than continuing for multiple years to amortize the initial investment, most participants left the study after the free year ended. In addition, we also suspect that a non-negligible fraction did not use the free smartphone as their primary device as instructed, instead treating it as a free Android device to fool around with or ignore altogether. In hindsight, this seems unsurprising: given something for free, people treat it as having little value. But when we discovered this after the second year of the study, it gave us the opportunity to make changes that have dramatically improved our ability to scale and sustain PhoneLab.

In the third year we ended the free-year incentive and began charging all participants. The $45 per month participants pay is almost sufficient to cover the amount PhoneLab pays to Sprint, making the study self-supporting and allowing us to scale out to a potentially unlimited number of participants. Because everyone wants a free smartphone, charging for service has made it somewhat more difficult to recruit participants. But the ownership our new participants take over their smartphone has produced better participants that actually use their PhoneLab smartphone as their primary device.

Because participants do not have any direct relationship with Sprint, we do serve as a point of contact for problems with broken or malfunctioning devices. For the last two years of the project we have employed a full-time testbed administrator to provide on-campus support to participants, and work with Sprint to address certain billing issues. We have made enough progress in streamlining testbed management to be able to eliminate this position going forward, but support and management are two parts of PhoneLab that still scale poorly with the number of participants. As a result, we have no plans to increase the size of PhoneLab through more recruiting – at least not until we can acquire additional funding to support a management position. But it also unclear whether it is worth the large amount of effort to scale PhoneLab past its current size as a medium-size testbed. We suspect that for most research studies, 175 participants are as good as 500 or even 1,000.

PHONELAB IS BOTH A GROUP OF PEOPLE AND AN ONGOING COMMUNITY EFFORT TO MAINTAIN A WELL-INSTRUMENTED VERSION OF THE AOSP USEFUL FOR RESEARCH PURPOSES.

Despite our success in making the testbed more sustainable, PhoneLab's future is uncertain. It has taken us longer than planned to deliver a working smartphone platform testbed, and both the IRB approval and difficulty modifying the Android platform serve as largely unavoidable barriers to entry. However, PhoneLab is beginning to attract external users. We have just completed an experiment on behalf of researchers at the International Computer Science Institute in Berkeley investigating lock screen usage and are working with groups at both the University of Michigan and UCSD that have experiments (on network handover and smartphone security, respectively) in the final stages of approval. In other cases, researchers have decided, correctly or not, that medium-scale experiments or data collection are unnecessary to evaluate new ideas and chosen to conduct small-scale experiments instead.

At the end of the day, a medium-scale human-facing testbed like PhoneLab will always be more expensive to operate than a network of machines and more difficult to use than running experiments on a handful of graduate students. Continuing the project past September 2016 will require additional funding and community support. But until then, PhoneLab is available to help you evaluate your new smartphone platform ideas at scale. We hope that this article has helped explain why and how you might use our testbed, and we look forward to running your experiment soon. ■

Geoffrey Challen is an assistant professor at the University at Buffalo, where he directs PhoneLab and leads blue, a research group investigating aspects of smartphone and distributed systems, including energy efficiency, programming models, hardware-software codesign, crowdsourcing and performance.

Jinghao Shi is a third-year PhD student at University of Buffalo. His research interests lie broadly in wireless networks and mobile systems. He received a BCS degree from the University of Science and Technology of China in 2011.

Edwin Santos is a fourth-year undergraduate student studying computer engineering and is one of the systems administrators for the PhoneLab and contracted as a developer for a startup by the name of ArgyleTechnologyGroup.