CI-ADDO-NEW: PhoneLab: A Programmable Participatory Smartphone Testbed

NSF Proposal
10/25/2011

PI: Geoffrey Challen; Co-PIs: Murat Demirbas, Steven Y. Ko, Tevfik Kosar and Chunming Qiao
SUNY Buffalo

Keywords: (1) **large-scale testbed**; (2) **smartphone**; (3) **pervasive computing**.

The expanding capabilities and growing number of smartphones are producing a new computing ecosystem integrating phones, users, and the Internet. The power of smartphones is truly transformative—users now rely on their phones to locate their friends, identify the song playing at a restaurant, stream video and music, provide instant access to information, and help document their lives, all in addition to placing phone calls and sending text messages. However, with this new computing paradigm come new challenges: utilizing multiple radio technologies and integrated sensors, harnessing powerful processors to support demanding applications, and efficiently moving data to devices. *Yet, despite the challenges and transformative nature of smartphones, no public research testbed exists enabling large-scale realistic smartphone experimentation.*

We propose to develop PhoneLab, a new scientific instrument enabling smartphone research in a realistic environment at a scale not previously possible. Targeting 1,000 programmable Android devices supported by backend servers and distributed to SUNY Buffalo students, PhoneLab will provide the **power**, **scale**, **realism** and **density** required to enable the next-generation of mobile computing research.

Intellectual Merit: PhoneLab *will harness the potential of smartphones* by allowing researchers to experiment with mobile operating system design, wireless networking, distributed algorithms, and smartphone applications. PhoneLab provides **power**, allowing the modification of smartphone operating systems as well as applications, while simplifying instrumentation and data collection to facilitate efficient experimentation. PhoneLab provides **scale**, allowing researchers access to an order-of-magnitude more participants than typically used by smartphone studies. By minimizing experimental disturbance, PhoneLab provides **realism**, ensuring that participants use their smartphones as they would normally. And, by facilitating access to a colocated group of participants, PhoneLab provides the **density** to enable peer-to-peer applications or infrastructure-driven experimentation. PhoneLab will accelerate research on smartphone applications, networking, infrastructure, and system software, providing a standardized environment where experiments can be validated and compared.

Broader Impact: PhoneLab will be a publicly-available open-access testbed. Considering the research impact of other similar instruments—EmuLab on distributed systems, PlanetLab on wide-area networking, and MoteLab on sensor networking—we anticipate that PhoneLab *will accelerate smartphone research.* Given the sensing and inference capabilities of the smartphones, we expect PhoneLab to engage computer scientists from a broad range of sub-disciplines. Wireless networking, distributed systems, operating systems, mobile sensing, social networking, crowdsourcing, and many other sub-disciplines will find uses for the testbed. PhoneLab is designed to enable their research.

PhoneLab will also play a vital role in teaching and learning. At the secondary level, we will encourage and support middle- and high-schoolers to use PhoneLab for science projects. At the college level, PhoneLab will enable a host of new testbed-based assignments in computer science and engineering courses. PhoneLab will help excite the next generation of computer scientists and developers by exposing them to the potential of smartphones. U.S. leadership in mobile computing depends on our ability to teach these skills, and PhoneLab provides a useful tool to this end.

Partnerships: We are actively collaborating with both Google—who develops the Android platform—and Sprint, a major nationwide wireless carrier. Google provided seed funding for an initial small-scale PhoneLab testbed currently under development. Sprint has entered into a unique partnership with SUNY Buffalo allowing us to offer attractive incentives to participants while reducing the cost to the NSF. With our deal, Sprint assumes 55% of the total cost of the incentives required to maintain PhoneLab at 1,000 participants; each participant will receive a free Android smartphone as well as the monthly discount of 42% in addition to the typical 10% student discount available to SUNY Buffalo students. The total cost saving for each participant over 4 years will be over $2,000.

1 — OVERVIEW

We propose to build PHONELAB: a programmable participatory smartphone testbed enabling realistic smartphone research at a scale and power not currently possible. PHONELAB will eventually consist of over 1,000 smartphones distributed to SUNY Buffalo students (participants)[1]. Experimenters (users) will have the ability to install interactive and background applications as well as modify the Android platform itself, harnessing the realistic activity of a large number of participants for their experiments.

As the smartphone spreads, the need for a smartphone testbed grows. Smartphones are not mere phones: we expect them to locate friends; identify the song playing at a restaurant; provide instant access to music, video, and other information; and help us document our lives. Smartphones are next-generation computing devices, and while smartphone penetration in the US grows rapidly, in the developing world smartphones are already the primary way of accessing the Internet for millions. The academic community is responding, with computer scientists drawn from a broad range of sub-disciplines studying the effects of smartphones and teaching students to harness their power.

Unfortunately, smartphone research and education is currently hindered by limited experimental methodologies. Today's smartphone studies typically use either the *single study* model—which recruits a small number of participants for each experiment—or the *marketplace* model—which utilizes established application distribution channels. Unfortunately, both these approaches have critical limitations. The single study model limits the achievable scale and forces researchers to attract new participants for each experiment; on the other hand, the marketplace model limits the components of the smartphone the experimenter can access, rendering important research impossible.

By consolidating the process of recruiting and managing a large number of participants, PHONELAB addresses the weaknesses of both the single study and marketplace models. PHONELAB provides the scale and realism required for systems evaluation; its participants are colocated, facilitating experimentation that leverages density; and it provides full access to each phone, supporting both application and Android platform experimentation. The combination of *scale*, *realism*, *density* and *power* makes PHONELAB the platform where large-scale, realistic smartphone systems research becomes possible for the first time.

Support for PHONELAB has emerged from within academia and industry. A total of 42 researchers have expressed support for PHONELAB. Google, who develops the Android smartphone platform we have selected for PHONELAB, funded a pilot version of the testbed and has committed to continued collaboration with our SUNY Buffalo team. Sprint, a major wireless carrier, has customized corporate and individual plans for PHONELAB, contributing up to $384,300 per year in savings toward testbed operation. Given the unprecedented levels of academic and industrial support, the time to build PHONELAB is now.

Our proposed tasks include, (1) acquiring necessary hardware including servers as well as smartphones, each with a voice and data plan, (2) developing, operating, and managing PHONELAB control software and infrastructure that provides a Web interface to users and participants, allows reprogramming of the smartphones at both the kernel and application layers, and collects experiment data, (3) incentivizing and recruiting participants that will use the phones as their primary phone, (4) organizing continuous community activities to promote the use of the testbed as well as to increase satisfaction of the testbed users and participants, and (5) performing outreach such as open-sourcing PHONELAB software and reaching out to minorities as well as educators at all levels—graduate, undergraduate, and K–12.

1.1 — Structure of This Proposal

The rest of this proposal is structured as follows. In Section 2, we describe how PHONELAB enables research by describing recent research projects that could have used PHONELAB and summarizing broader community support. Section 3 describes how PHONELAB will be utilized by educators. Section 4 outlines how we plan to meet our service, satisfaction and outreach commitments, and we discuss broader impacts in Section 5. In Section 6 we describe the design of PHONELAB in detail, discussing different scenarios, participant incentives and protection, and a technical overview. We conclude the proposal by presenting our qualifications and a testbed management plan in Section 7.

[1] To avoid confusion, throughout this proposal we refer to the SUNY Buffalo students operating the PHONELAB phones as *participants*, and to the researchers submitting smartphone experiments as *users*.

	Scale	Realism	Density	Power	Repeatability	Continuity	Synergy
Single Study	Low	Low	High	High	Low	Medium	Low
Marketplace	High	High	Low	Low	Low	Low	Low
PhoneLab	Medium	High	High	High	High	High	High

Figure 1 — PHONELAB compared to the marketplace and single study approaches. PHONELAB combines the *scale* and *realism* of the marketplace model with the *density* and *power* of the single study model.

2 — ENABLING RESEARCH

Today, smartphone researchers have two main avenues for doing realistic studies: either assemble participants themselves for each experiment, or release their code on an application marketplace. We refer to the first approach as the *single study* model and the second as the *marketplace* model. To compare PHONELAB to the single study and marketplace models we use seven metrics:

1. **Scale:** How many participants can a user access?
2. **Realism:** How realistic is the behavior of the participants?
3. **Density:** What is the expected geographic distribution of the participants?
4. **Power:** How much access does the user have to participants' smartphones?
5. **Repeatability:** Can a user reproduce another user's findings under similar conditions?
6. **Continuity:** Can a user perform a second study on a similar set of participants as their first study?
7. **Synergy:** Can users use and build on results collected by other experiments?

Figure 1 shows that both the single study study and marketplace approach have drawbacks that PHONELAB addresses. Single study experimentation fails to provide **scale** and **realism**. It requires researchers organize a new group of participants for each experiment, a time-consuming process that limits **scale**. In addition, the one-off nature of singly study experimentation effect **realism**: participants are unlikely to be willing to make significant changes to their own phone, or use a temporary phone as their primary device. At the same time, the marketplace models cannot provide **density** and experimental **power**. Marketplace participants will be geographically dispersed, reducing the likelihood that any two experimental devices will directly interact. And the marketplace fails to deliver *power* by only permitting application-level experimentation and not allowing root access or modifications to system libraries or other platform components. *Many systems and networking experiments depend on these features.*

In contrast, while PHONELAB cannot achieve the **scale** of the marketplace, it provides access to many more participants than any single study could recruit. Participants use their phones as their primary device, providing **realism**, and are all students at a single institution, providing **density**. Finally, PHONELAB provides experimenters the **power** to modify the Android platform. PHONELAB also delivers cross-experiment benefits lacking in both the single study and marketplace model. PHONELAB participants are available to any researcher, providing **repeatability** and allowing competing approaches to be compared. PHONELAB participants will typically spend over three years as part of the testbed, providing **continuity**. Finally, PHONELAB achieves **synergy** through the exchange of experimental results. Data sharing not only allows accelerates research, it also protects PHONELAB participants by limiting data collection.

2.1 — Recent Research Projects Well-Suited for PHONELAB

To demonstrate community need for PHONELAB we describe six recent research projects that could have extended their results by using PHONELAB. In addition, many of the authors of the following studies have written letters of collaboration to support PHONELAB which we include quotes from here.

2.1.1 — TaintDroid: TaintDroid is a taint-tracking system for the Android platform that tracks how third-party applications collect and use sensitive user information [22]. Implementing TaintDroid required significant changes to the Dalvik Java virtual machine used by Android. TaintDroid could have leveraged PHONELAB's scale and realism to examine a large number of applications under realistic settings, running in the background with no required participant interaction. According to Landon Cox, one of the TaintDroid authors, "Integrating TaintDroid with PHONELAB would give us the opportunity to monitor the behavior of all third-party applications downloaded, installed, and used by real users."

2.1.2 — NetPiculet: NetPiculet is a phone network monitoring tool [37] that unmasks network-address translation (NAT) and firewall policies used by cellular carriers. The NetPiculet study demonstrated that these policies have significant impacts on the power consumption and connectivity of smartphone devices. It used a combination of both the marketplace approach—to achieve large scale and worldwide distribution—and the single study approach to add additional power. While 393 participants obtained a the NetPiculet tool on the Android Market, without root access this limited version could not perform many experiments. On PHONELAB, NetPiculet would both have access to significant scale—more even than was achieved via the Android Market—but also be granted permission to run tools normally limited to the root user. Morley Mao, who is one of the NetPiculet authors, and familiar with both the marketplace and single study models says that "Having a large-scale testbed will complement all of these methods and lower the cost and overhead for an experiment."

2.1.3 — Code in the Air: Code in the Air is a cross-platform approach to simplifying the process of developing mobile phone crowdsourcing applications [24]. The Code in the Air project imagines that instead of running multiple crowdsourcing applications users join a platform that multiple developers and researchers can access which handles all of the low-level details and simplifies crowdsourcing application development. The immediate scale and density provided by PHONELAB will be extremely useful to develop and debug key Code in the Air components. Once Code in the Air is operational, we can use it to provide an crowdsourcing application interface to PHONELAB and lower the barrier of entry for writing intelligent crowdsourcing applications. Hari Balakrishnan, who is leading the Code in the Air project, agrees that "PhoneLab would be an ideal platform for experimenting with Code in the Air. By providing quick access to significant scale at high density, it will allow us to develop and debug key Code in the Air components."

External	**A**	**N**	**I**	**O**
1. Hari Balakrishnan (*MIT*)		X		
2. Matt Caesar (*UIUC*)		X		
3. Andrew Campbell (*Dartmouth*)	X			
4. Romit Roy Choudhury (*Duke*)		X		
5. Byung-Gon Chun (*Intel Research*)			X	X
6. Landon Cox (*Duke*)				X
7. Hakan Ferhatosmanoglu (*Ohio State*)	X			
8. Renato Figueiredo (*U. of Florida*)	X			
9. Rodrigo Fonseca (*Brown*)	X	X	X	X
10. Mike Freedman (*Princeton*)		X	X	
11. Brighten Godfrey (*UIUC*)		X		
12. Indy Gupta (*UIUC*)			X	X
13. Mark Hempstead (*Drexel U.*)				X
14. Adriana Iamnitchi (*U. of South Florida*)	X		X	
15. Srikanth Kandula (*MSR*)		X	X	X
16. Sandeep Kulkarni (*Michigan State*)		X	X	
17. Gregor von Laszevski (*Indiana U.*)			X	
18. Samuel Madden (*MIT*)	X	X	X	X
19. Z. Morley Mao (*Michigan*)		X	X	X
20. John Regehr (*Utah*)			X	X
21. Jennifer Rexford (*Princeton*)		X		
22. Matei Ripeanu (*U. of British Columbia*)	X			
23. Thomas Schmid (*Utah*)	X	X	X	X
24. Andreas Terzis (*Johns Hopkins*)	X	X	X	
25. Ali Saman Tosun (*UT San Antonio*)	X	X		
26. Nitin Vaidya (*UIUC*)		X		
27. Arun Venkataramani (*UMass Amherst*)		X		
28. Matt Welsh (*Google*)		X		
29. Hui Zang (*Sprint Labs*)		X		
Internal				
29. Stella Batalama (*EE*)		X		
30. Ann Bisantz (*ISE*)	X			
31. Mike Buckley (*CSE*)	X			
32. Chang Wen Chen (*CSE*)	X			
33. Lorraine Collins (*Health Behavior*)	X			
34. Jason Corso (*CSE*)	X			
35. Raymond Fu (*CSE*)	X			
36. Dimitrios Koutsonikolas (*CSE*)		X		
37. Hung Ngo (*CSE*)		X		
38. Alex Nikolaev (*ISE*)	X			
39. Dimitris Pados (*EE*)		X		
40. Mark Shepard (*Media Study*)	X			
41. Shambhu Upadhyaya (*CSE*)	X		X	X
42. Sheng Zhong (*CSE*)		X	X	

Table 1 — List of researchers who have expressed interest in using PHONELAB and their area of interest. A—Applications; N—Networking; I—Infrastructure; O—Operating Systems.

2.1.4 — Serval: The current Internet experiences unprecedented multiplicity. Modern network services are replicated across many geographically distributed servers and accessed by multi-homed, mobile devices. Unfortunately, the Internet was designed mainly to support fixed, host-to-host communication. The Serval [30] project identifies that most of the core issues for supporting modern networked services and mobile clients can be solved by redesigning the network stack that runs on each host, adding a new layer between the transport layer and the link layer that provides replication and mobility. Serval is designed with mobile devices such as smartphones in mind and would be ideal for testing on PHONELAB. Because it requires changes to both the networking stack and applications it could not be distributed through the marketplace, but on PHONELAB the Serval architecture could be evaluated in a realistic, large-scale setting. Jen Rexford, one of the primary Serval authors, notes that "A large-scale smartphone testbed such as PHONELAB will significantly benefit our evaluation, and the features of PHONELAB are suitable for our purpose."

2.1.5 — SleepWell: SleepWell attempts to reduce the power consumption of WiFi devices using power saving modes, such as smartphones [29]. SleepWell uses access point coordination schedule client sleep intervals to avoid congestion and reduce the time clients spend on data transmission. Because SleepWell requires 802.11 driver-level modifications it is not suitable for the marketplace experimental model. The original SleepWell study used a small number of laptops to evaluate its effectiveness. A PHONELAB study would help determine how well SleepWell works on a large number of devices with different access patterns. SleepWell also illustrates another potential mode of PHONELAB experimentation in which both the participant's smartphones and the surrounding infrastructure are part of the experimental setup. According to Romit Roy Choudhury, one of the SleepWell authors, "With a large-scale smartphone testbed (such as PHONELAB), we would have had many more vantage points to measure the effectiveness of our system."

2.1.6 — Wiffler: Many cellular network users suffer from either unavailability of 3G or poor performance due to congestion. Wiffler augments 3G access with WiFi [8] For delay-tolerant applications such as email, Wiffler offloads data transfer to WiFi based on WiFi connectivity prediction. For delay-sensitive applications such as VoIP, Wiffler implements fast switching between 3G and WiFi in order to reduce delay and loss. Evaluating Wiffler requires both realistic mobility and application usage patterns. The original Wiffler study merges two studies, one that achieves realistic mobility (using vehicles) without realistic application usage and one that achieves realistic application usage without realistic mobility. PHONELAB provides both simultaneously, making it an optimal platform for further Wiffler experimentation. V. Arun Venkataramani, one of the Wiffler authors, states that "PHONELAB could complement our findings and evaluation by providing another platform with people, instead of vehicles."

2.2 — New Research Enabled by PHONELAB

Researchers in the systems and networking community have expressed enthusiastic support for PHONELAB, and Table 2 lists them and their area of experimentation interest. We anticipate that PHONELAB will further research in four areas: (1) smartphone application and interface development; (2) network optimization and protocol development; (3) phone-infrastructure co-development; and (4) operating system research enabling more powerful and longer-lasting smartphones. Below we briefly describe each area and describe an example research project in the area that could use PHONELAB.

2.2.1 — Applications: *crowdsourcing, social networking, information propagation, behavioral studies, user interaction, environmental sensing, epidemiology studies.*

Example Application Project — Co-PI Demirbas is currently building a crowdsourcing platform that uses social networks such as Twitter to task and coordinate smartphones. His platform provides a publish-subscribe interface capturing the relations between users and their interest and expertise. The popularity and round-the-clock use of social networks also enables quick reply rates for smartphone based sensing and collaboration applications. Demirbas has used his social network based crowdsourcing platform for a weather monitoring application as well as a location-based querying application [17, 28]. The crowdsourcing system was especially successful at answering nonfactual (subjective) queries for which traditional search engines have a low success rate. Demirbas and his team are working on using this platform for queue-length monitoring application to enable users to query the current wait-times at nearby cafes and shops. A grand challenge application for this platform is to beat IBM's Watson system in the Jeopardy competition. These applications will be enabled by the large scale testing PHONELAB provides.

2.2.2 — Networking: *performance, multi-radio issues, delay-tolerance, peer-to-peer.*

Example Networking Project — Smartphones normally rely on existing infrastructure to communicate: voice and 3G base stations along with WiFi access points. However, in a disaster scenario it is possible that these typical communication channels will be damaged and unavailable at precisely the time when they are direly needed. Co-PI Qiao has an active research effort studying network survivability, and another colleague Dimitrios Koutsonikolas has a well-established interest in delay-tolerant routing protocols. Together they have expressed the desire to use PHONELAB to experiment with new smartphone-to-smartphone communication protocols that SUNY Buffalo could use on campus in the case of a crippling loss of surrounding communications infrastructure. This kind of experimentation will be both trace-driven, utilizing existing traces of PHONELAB participant movement collected, and experimental, eventually deploying a background application to study the effectiveness of their new mesh networking protocol at enabling participant communications in the case of a disaster.

2.2.3 — Infrastructure: *environmental interaction, task distribution, new wireless technologies.*

Example Infrastructure Project — Given the growing power of the cloud and the growing ubiquity of mobile devices, this split-programming paradigm seems poised to be the dominant one for the next decade of computing. However, one disadvantage of the cloud is that the network latencies can be very high, making it difficult for interactive applications to offload delay-sensitive computation. Meanwhile, there are frequently many nearby devices with significant storage and computational resources: the routers, laptops, desktops and servers located in most businesses; personal computing devices in the home; as well as computational devices located in vehicles. Co-PIs Challen and Ko have initiated the CloudLess project, which aims to provide a programming model, device and infrastructure support enabling smartphones to locate local resources that may be more available or efficient to use than remote cloud servers. Experimenting with CloudLess on PHONELAB will require changes to both smartphones and the surrounding infrastructure. This harnesses both PHONELAB's density—which facilitates interaction with a small number of routers running experimental software—and its power—required to allow access to the network stack and system libraries necessary to modify to implement CloudLess.

2.2.4 — Operating Systems: *mobile operating system design, distributed systems, performance, energy management, fault-tolerance.*

Example Operating Systems Project — Power management is a critical issue on smartphones and other battery-limited devices. Power management that is not aggressive enough leads to devices that run out of battery power while still in use, frustrating their users. Power management that is too aggressive leads to devices that conserve too much energy, reducing performance unnecessarily and frustrating their users. PI Challen and Mark Hempstead (Drexel University) are designing new hardware architectures that exploit device heterogeneity to reduce power consumption and improve overall performance [14]. Crucial to this effort are traces from real phones detailing how users use applications and how these applications consume power.Based on these traces, Challen and Hempstead will design new power-agile hardware architectures that intelligently activate and deactivate device components in response to load, helping the device find the most power-efficient state that satisfies the user's requirements. PHONELAB is the ideal environment for collecting data to develop user- and application-driver power models.

2.3 — Comparison to Similar Testbeds:

Although several recent projects have begun to address the weaknesses of the single study and marketplace models, none provides the capabilities of PHONELAB. The Reality Mining Project [21] was a single study that involved 100 students at MIT over the course of the 2004–2005 academic year. In this study, each participant was given a Nokia 6600 cellphone with a software that continuously logged certain connectivity and personal data. Reality Mining collected data that led to many important observations, and it establishes the feasibility of distributing phones to students, but is not a programmable testbed like PHONELAB.

Mob4Hire [2] is a company that focuses on crowdsourced mobile application testing services and market research in the smartphone space. Mob4Hire builds on the marketplace approach facilitating selective distribution and user experience evaluation of tested applications. While alleviating some repeatability and continuity limitations, Mob4Hire suffers from many of the same drawbacks of the marketplace approach.

Initiated by the University of Michigan and funded by NSF CRI, MobiLab aims to support mobile computing science. MobiLab aims to provide software libraries and tools—such as PowerTutor [20] for single-phone power-state monitoring and 3GTest [1] for single-phone 3G connectivity monitoring—that facilitate developing, deploying, and evaluating research applications. Unlike PHONELAB, MobiLab does not provide a programmable testbed enabling new experimentation or user access to the Android platform. MobiLab supports developers by providing interfaces, tools, and a data repository, whereas PHONELAB builds and provides a dense, large-scale programmable testbed. We believe that PHONELAB and MobiLab are complementary in vision and mission, as does MobiLab co-PI Morley Mao, a PHONELAB supporter.

CrowdLab is a software architecture designed to enable volunteer mobile testbeds [15]. It allows researchers to run guest virtual machines on volunteer mobile devices and ensures efficient use of the device resources through a dual-mode networking abstraction and a weakly-consistent, replicated state store called a site directory. While density, scalability, repeatability, continuity, and synergy concerns have been out of the scope of the CrowdLab work, PHONELAB provides these properties.

3 — ENABLING EDUCATION

PHONELAB will serve an important educational role for undergraduate and graduate smartphone-based courses. By allowing small-scale class projects to be tested in a large-scale live laboratory, PHONELAB will drive excitement among students in this critical area. This section presents the benefits of PHONELAB for educators. We begin in Section 3.1 by describing how PHONELAB's capabilities can be used in a classroom setting. Section 3.2 continues by outlining potential educational uses of the testbed by instructors at SUNY Buffalo, local K–12 school teachers, and educators at other universities. In Section 3.3, we present describe a graduate research course currently being taught by the co-PIs using a small PHONELAB prototype.

3.1 — PHONELAB for Educators

Utilizing smartphones for teaching computer science concepts opens up new opportunities for educators. Today's smartphones are mobile and highly-interactive, equipped with touch screens, GPS positioning, accelerometers, gyroscopes, digital compasses, and more. These new components are driving new interaction modalities beyond those enabled by traditional stationary computing devices. This interaction drives student excitement, as new computer scientists are able to more easily see and utilize the connections between computing and the world around them.

In addition, as small mobile computing devices permeate our world, it is more critical than ever to teach students how to program and reason about these new devices. Mobility requires devices identify and respond to context, and sensors allow them to do so. But we must teach our students how to utilize and analyze inputs from sensors, control and actuate devices, and respond to the challenges of mobility. Smartphones are the ideal platform for teaching and learning these skills.

Our expected education usage model for PHONELAB-enabled courses is as follows. Instructors will be required to provision their own courses with environments suitable for local testing, either by purchasing hardware or by using the Android emulator. The emulator runs in a user-level virtual machine and provides a phone screen interface to the developer, who uses the mouse and keyboard to interact with the smartphone. While access to real hardware is preferable, the emulator provides a cheap and easy way to get started with Android development.

Once students have developed their projects and tested them locally, either on real phones or in the emulator, PHONELAB will be available to facilitate large-scale deployment. In this way PHONELAB provide students everywhere first-hand experience with real participant testing. people. This is typically impossible with a small local testbed—although many courses utilize real smartphone hardware [2], we are not aware of any course that gives an opportunity for students to deploy their prototypes on the phones *used by others*. Needless to say, this will be a rewarding experience for students while giving them to chance improve their projects based on feedback provided by PHONELAB participants.

For local students at nearby K–12 institutions we will offer a smartphone "check out" feature. We have recently received 50 Nexus S phones (without plans) from an anonymous PHONELAB supporter. While these phones cannot be used for voice or data, they can access WiFi networks and do provide a realistic hardware environment for application testing. We will make these devices available to local educators for distribution to students, particularly targeting low-income schools unable to otherwise provide hardware to students. Educators will request a certain number of available phones for a semester-long class, at which point the phones will be returned for future use. We also expect to have additional development phones available yearly as PHONELAB participants upgrade and return the phones we purchase from Sprint.

3.2 — Potential Courses and Instructors Utilizing PHONELAB

Local educators have already expressed interest in using PHONELAB for the courses described below:

3.2.1 — Amy Bisantz *(Industrial and Systems Engineering, SUNY Buffalo)*: Professor Amy Bisantz teaches various courses on Human Factors, including "Human Computer Interaction", "Human Factors in Safety", and "Introduction to Human Factors". She has recently started to utilize smartphones and tablets for her classes and is interested in using PHONELAB to expand the scope of her students' projects.

[2] Google Code University lists several courses already doing this: `http://code.google.com/edu/courses.html#android`.

3.2.2 — Mike Buckley *(Computer Science and Engineering, SUNY Buffalo)*: Professor Buckley co-directly the Center for Socially Relevant Computing at SUNY Buffalo. He teaches an annual course that assembles students into design teams and partners them with existing community groups to tackle a variety of social issues. PHONELAB would provide an ideal testing ground for students taking his class, enabling them to see the effects of scale and density on novel applications.

3.2.3 — PIs : Several of the PIs' own courses—"Advanced Computer Systems" (taught by Challen, Demirbas, and Ko), CSE "Wireless Sensor Networks and Mobile Computing" (taught by Demirbas), "Distributed Systems" (taught by Ko), and "Introduction to Operating Systems" (taught by Challen)—are all being designed to utilize PHONELAB. To demonstrate how similar courses can be organized, we present a currently-running example next.

3.3 — Example Course: Advanced Systems Research at SUNY Buffalo

This semester co-PIs Challen and Ko are teaching a graduate-level course entitled "CSE622: Advanced Systems Research". For fall 2011, our chosen subtopic is "Smartphone Research and PhoneLab Testbed Development." We provide each student with a Samsung Nexus S 4G smartphone, equipped with a discounted voice and data plan provided to us by Sprint and paid for by Google. After only several weeks of exposure to smartphone research our students developed and began working on the following projects:

3.3.1 — CycloneCloud *(Operating Systems)*: Certain sensors—such as GPS—are ubiquitous on mobile smartphones, and yet their operation consumes a great deal of power and can limit device performance. The CycloneCloud project is investigating whether low-power, short-range communication protocols such as Bluetooth can eliminate redundant sensor data collection through data sharing.

3.3.2 — 802.11n Power Modeling *(Networking)*: While 802.11n has brought unprecedented speeds to the WiFi world, its impact on battery-constrained devices is less certain. The new standard also includes many features that are likely to have an impact on device power consumption but, as of yet, the tradeoffs in this space are not yet clear. To investigate these issues, a group of CSE622 students is building a power model of the BroadCom 4329 chip deployed on the Samsung Nexus S.

3.3.3 — HoneyNet *(Security)*: Along with their growing power comes a new generation of smartphone security threats, from root kits to security holes to phishing-based attacks. The HoneyNet project is exploring the idea of using multiple smartphones to create a networked mobile honeypot, similar to the ones used to trap and deflect attacks on the broader Internet.

3.3.4 — Rooted ROMs *(Operating Systems)*: The Android operating system has brought an unprecedented level of openness to smartphone platforms. Yet, most phones are still shipped with carrier-installed restrictions on what users can do. Rather than be saddled with underperforming phones or unwanted software, a significant number of smartphone users take to "rooting" their phones, unlocking them in a way that allows them more complete control over their device. The Rooted ROMs project is investigating this subculture of smartphone users and particularly interested in identifying the root causes of the improved performance that these "modded" system images are said to provide.

3.3.5 — SmartReduce *(Infrastructure)*: MapReduce is the dominant new programming paradigm designed to allow programmers to easily harness large clusters of machines. As smartphones become a dominant computing platform, more and more data processed by MapReduce jobs will either originate from or end up on smartphones. Yet, these devices are still computationally-limited compared with the vast resources available in the cloud. The SmartReduce team is investigating how to intelligently partition MapReduce jobs between phones and the cloud.

3.3.6 — Smart Data Transfer *(Networking)*: The proliferation of smartphones equipped with cameras–for capturing video—and screens—for displaying it—will drive a continued increase in the amount of data moving back and forth between smartphones and the cloud. Some of this data transfer is latency-sensitive; but a lot of it may not be, consisting of movies and other content being slowly moved to or from smartphones. Relaxing latency constraints means focusing on energy consumption and bounded data transfer. The Smart Data Transfer team is trying to design a system that can, given an amount of data to transfer and a time bound, schedule the transfer in a way that minimizes the amount of energy used on the phone.

4 — SERVICE, SATISFACTION AND OUTREACH

Unlike most similar testbeds—MoteLab, EmuLab, PlanetLab, and the like—which are composed entirely of computing hardware made available the research community, PHONELAB has a significant human component: our participants. Thus, a significant challenge in ensuring that we deliver good service to *both* our users and our participants. determining how to attract, protect, and engage our human participants. This section describes out quality of service commitment to our users and participants, ways to measure and maintain user and participant satisfaction, and our outreach efforts to the relevant scientific communities.

4.1 — Quality of Service Commitment

We will make PHONELAB *available* and *open* to users and *safe* and *transparent* to participants.

4.1.1 — Availability: PHONELAB will be available for experimentation for the duration of the grant and beyond. Section 6.7.7 details the technical steps we are taking to ensure that the testbed services are continuously available for users. For the participant-side failures—broken or lost phones—we will be retain some spare or older-model phones for replacement purposes. In case of phone loss, we will work with Sprint recover missing devices. We will also require all PHONELAB participants to file a police report when phones are reported as lost. In cases where the phone is either reported lost or stolen and we cannot recover the device, the participant will be asked to leave the testbed.

4.1.2 — Openness: PHONELAB will be made available to all users proposing safe and scientifically-sound experiments. Minorities and underrepresented groups will be especially encouraged to use the testbed through workshops and seminars organized targeting these groups. We will document the process of PHONELAB experimentation very clearly and publish it on our web site, which will lead the potential users through the whole process step by step. We will also work with experimenters to craft compelling descriptions of their experiments that appeal to participants. If we feel that an important experiment is not being picked up by PHONELAB participants, we will work with experiments to determine how to further incentivize that experiment.

4.1.3 — Safety: PHONELAB will ensure all participants private and sensitive data are protected. We will restrict the data which can be collected by experiments and limit the usage of the collected data only that that required for effective experimentation. We will allow participants control over their data during and after their participation in PHONELAB. Participants can leave an experiment or PHONELAB at any time.

4.1.4 — Transparency: PHONELAB data collection will be transparent to users and participants. We will help our participants understand what information is being collected by experiments and why. Regular open meetings will be held to discuss the operation of PHONELAB, answer participant questions, address feedback and present results from previous experimentation.

4.2 — User and Participant Satisfaction

Our goal is to satisfy our commitments to users and participants and we will employ multiple methods to ensure we are doing so. We will conduct user surveys every six months to measure satisfaction, while internally monitoring PHONELAB health metrics such as experiment waiting time, failure and success rates, adoption rates by participants and system downtime. We will also survey participants every three months to measure their satisfaction, and collect feedback. If a participant chooses to leave PHONELAB, we will schedule an exit interview with that participant and the PHONELAB administrator to examine their reasons for not continuing to participate. We will maintain statistics on participant turnover and preferences.

4.3 — Outreach

We will spread the word about PHONELAB at major CS systems and networking conferences, including SOSP, OSDI, MobiCom, SIGCOMM, MobiSys, and SenSys as well as well-circulated magazines such as SIGOPS OSR and SIGCOMM CCR. The PHONELAB web page will be used to disseminate preliminary results, white papers, technical reports, user and participant manuals, other documentation about the project, as well as the source code for PHONELAB software.

5 — Broader Impact

We have already described how PhoneLab will enable a wide range of new computer science and engineering research and education activities. In addition, PhoneLab will also enable other broader impacts through K–12 outreach, recruitment of minority groups, enhancing scientific and technological understanding via international academic workshops, and distributing tools and software to the community.

5.1 — Impact on K–12 Education

Beginning the process of educating and exciting students about technology when they reach college is too late. Reaching students in K–12 and building excitement about mobile computing and computer science broadly at that level would serve as a major driver of innovation. To this end, we have a series of activities aimed to the middle and high school students at several Buffalo public schools, which we describe below.

5.1.1 — Science Projects with K–12 Students: Co-PIs Kosar and Demirbas serve on the board of the Buffalo Academy of Science, a technology charter school enrolling 98% of its students from low-income backgrounds. They have long experience of mentoring middle and high school students in science projects related to their ongoing research activities. Several of these science projects resulted in national awards at middle and high school science competitions. As part of PhoneLab, the PIs are planning to mentor middle and high school students in science projects related to smartphone programming and testing, and will also encourage graduate student involvement in this process. The projects will be designed to leverage the PhoneLab infrastructure and the applications it supports.

5.1.2 — Integration with High-School Curriculum: High school teachers Mr. McKee and Mr. Pawloski at Oracle Charter School in Buffalo are running a program using Android phones to introduce high schoolers to the basics of smartphone programming. The PIs have been in contact with them, and they are interested in using PhoneLab for enhancing their high school programming curriculum.

5.1.3 — K–12 Seminars and Camps: The PIs are planning to give seminars on smartphone usage, programming, and experimentation at high schools, beginning with the two mentioned above. Taught at a middle- to high-school level, these seminars will motivate students to early participation in science while attracting them to college and the study of computing. The PIs have also participated in the organization of successful summer camps for high school students and teachers over the last several years. We will organize at least one similar camp on smartphone programming during the duration of the project. In a 3–5 day program, students and teachers will be introduced to the basics of smartphone programming through hands-on experience. They return to their schools understanding how to further integrate utilize mobile computing technology into their school's curriculum.

5.2 — Broaden Participation of Underrepresented Groups

The planned K-12 activities in the previous subsection will engage Buffalo public schools with a high percentage of low income and minority students. Additionally, PIs will encourage engagement of female and minority students, at both undergraduate and graduate levels, in PhoneLab research and development through the very active "independent research" in our department. We will also ensure that PhoneLab participants are representative of the SUNY Buffalo community.

5.3 — Enhancing Scientific and Technological Understanding

The PIs have organized previous international workshops on cloud, mobile, and distributed computing, including DADC'08–12, DataCloud'11 and DataCloud-2, and EaglSysNet'11. As part of the PhoneLab project, the PIs will organize a workshop on "Smartphone Programming, Testbeds, and Challenges", assembling area experts to present their work and brainstorm future research challenges.

5.4 — Benefits to Society

All PhoneLab software developed as part of this project will be made available online as open source software released under a BSD-like license. We will also strive to support those who wish to use our software to manage their own testbeds, by developing online documentation and release notes about application changes and helping adopters through software upgrades.

6 — PhoneLab Design and Implementation

This section describes the design of PhoneLab in detail. We motivate our choice of Android (Section 6.1) and present our user and participant expectations (Section 6.2) before walking through PhoneLab user and participant scenarios (Sections 6.3 and 6.4). We continue by describing our discount agreement with Sprint (Section 6.5), participant protection (Section 6.6), and technical details of how we distribute experiments and collect data (Section 6.7).

6.1 — Android: The Right Platform for Smartphone Research

PhoneLab utilizes Android for three reasons. First, it has the biggest market share among all smartphone platforms, accounting for 43% of 2011 Q2 smartphone sales [39]. Second, the platform is open. In addition to application development, Android users can modify the Linux operating system kernel, system tools and libraries and the Dalvik Java virtual machine. No other available smartphone platform provides these capabilities. Third, Google provides full support for certain devices referred to as Android Dev Phones, the current being the Nexus S 4G. PhoneLab will be composed of Android Dev phones.

Targeting Android Dev phones means that PhoneLab will be a largely *homogeneous* testbed, although we expect the official development phone to change over the three years we are purchasing devices. A single homogeneous platform simplifies development and administration but does not facilitate experiments running on a mixture of different devices. We are interested in increasing the future platform and device heterogeneity of PhoneLab but focus our initial effort on building a large single-platform testbed.

6.2 — User and Participant Expectations

A set of expectations for both users and participants drive the design of PhoneLab. To participants, we are providing highly-discounted smartphones and usage plans; in return, we expect participants to: (1) interact with experimental applications for approximately two hours per week and allow their smartphone to be used transparently for background experimentation; (2) not attempt to tamper with necessary PhoneLab components or the PhoneLab infrastructure; and (3) notify the PhoneLab administrator when their equipment is damaged, lost or stolen.

To users, we are providing access to a large pool of real smartphone users and large amount of previously-collected smartphone data; in return, we expect users to: (1) identify themselves honestly to the PhoneLab administrator during the experimental submission process; (2) test their applications and platform changes carefully in order to produce stable experiments; (3) not collect more data about PhoneLab participants than required by their experiments; (4) return collected data to the PhoneLab data repository for future use by other experimenters; and, (5) not independently redistribute PhoneLab data.

As we continue we describe how we ensure that both users and participants are meeting our expectations.

6.3 — The User Experience: Experimenting on PhoneLab

PhoneLab is intended to support three kinds of experimentation: (1) **interactive applications** that require direct participant interaction; (2) **background applications** that do not require any user input; and (3) **system experiments** that require changes to the operating system kernel, system libraries or Dalvik Java virtual machine. The first two types require application-level changes and can be distributed as standard Android Application Package (APK) objects. For each kind of experimentation PhoneLab supports, we describe the process of submitting and executing an experiment below. Given the many similarities between interactive and background application experimentation, we describe them together first. The third category requires distributing new platform binaries and is more difficult to support, and for this reason we describe it separately in Section 6.3.2.

6.3.1 — Interactive and Background Application Experimentation: PhoneLab experimentation begins with local user development and testing (**Figure 2.1**). This step is important: because participants will use the phones as their primary device, we must protect them from buggy or poorly-designed experiments. In addition to local development and testing, users must obtain Institutional Review Board (IRB) approval at their host institution's (**Figure 2.2**). This is one significant way that we protect PhoneLab participants and necessary for human subjects experimentation. Requiring the user's IRB vet experiments also distributes the process of safeguarding PhoneLab and avoids concentrating it on our local IRB.

Once the application is tested and has received IRB approval, the user uploads their application; a description of what it is, what it does, and why it is interesting; and evidence of local IRB approval through the PHONELAB user interface (**Figure 2.3IB**). In some cases the IRB may decide that the experiment does not need IRB approval, in which case documentation to that effect must be submitted. After this step is complete, an automated sanity test is performed (**Figure 2.4IB**) that simply verifies that the application can actually be run without crashing immediately. If their application fails this sanity check, the user is notified. After this point, the PHONELAB administrator will briefly review the experiment (**Figure 2.5**), at minimum checking to ensure the IRB letter is valid. If the testbed is in high demand, we will exercise our judgment to determine the suitability of the experiment for PHONELAB. For example, we may grant higher priority to experiments that demonstrate a need for scale, density, realism or power. As a last step before scheduling, for non-interactive background applications we perform automated resource profiling (**Figure 2.6B**) tests to determine measure the load generate by the experiment. We are particularly concerned with power-hungry background experiments, as running too many concurrently could cause exhaust a phone's battery.

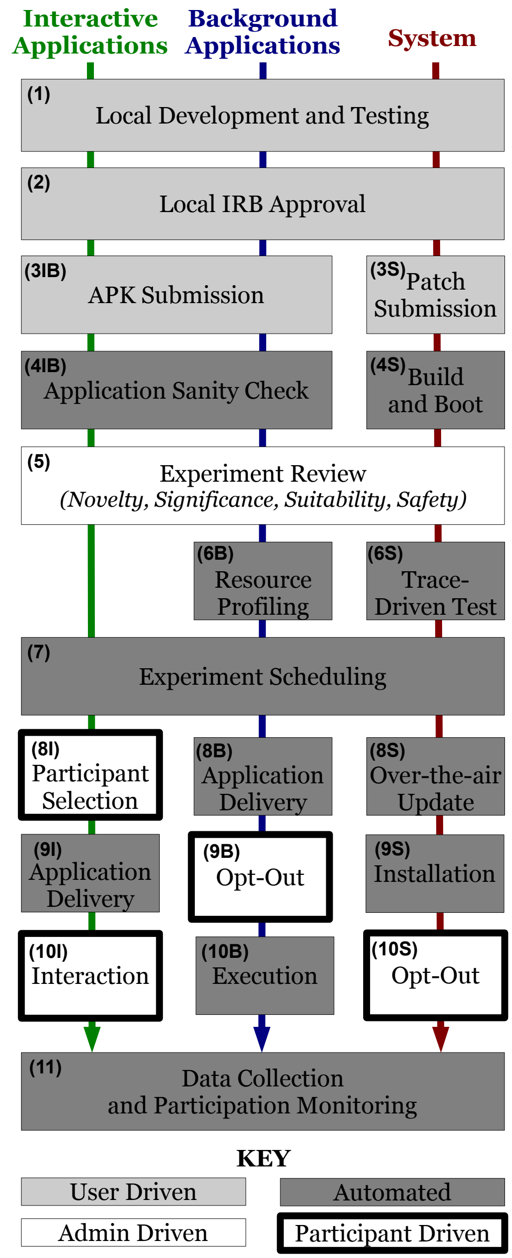

Figure 2 — Flowchart showing the PHONELAB development and experimental submission process. The key distinguishes between steps that are user-, administrator-, or participant-driven, and the many automated steps.

At this point experiments are ready for scheduling, with the time they reach PHONELAB determined by an experimental scheduling process (**Figure 2.7**). We expect that scheduling will be largely automated using a first come-first-served model. However, in case of high-demand, the PHONELAB administrator will be expected to monitor and adjust the scheduling process as necessary, balancing experimental throughput against participation levels. Scheduling too many interactive applications, for example, will reduce experiment participation levels by providing PHONELAB participants with too many choices.

After the experiment is scheduled the processes diverge again for interactive and non-interactive experiments. Interactive experiments are released to participants through the PHONELAB participant portal, which allows participants to select experiments to participate in (**Figure 2.8I**). Once they have selected an application, it will be delivered to their phone (**Figure 2.9I**) at which point the participant can launch and interact with it (**Figure 2.10I**). Given that interactive experiments are selected by participants, this is an *opt-in* model. Non-interactive applications use an *opt-out* model. In contrast to interactive applications, they are automatically pushed to participant phones (**Figure 2.8B**). At any time after that point participants retain the option to opt-out (**Figure 2.9B**).

While experiments are executing, we expect that users will want to collect data (**Figure 2.11**). For this purpose, we will provide data logging service that periodically collects experimental data from each phone to our PHONELAB servers (detailed in Section 6.7). In addition, applications and the PHONELAB platform will also track participation (**Figure 2.11**) in order to ensure that participants are meeting their commitment. This will be done by logging what experimental applications the participant is using and for how long.

6.3.2 — System Experimentation: System experimentation differs from application experimentation in several key ways, described below.

For systems experimentation that requires changes to the Android platform we require each users submit a patch (**Figure 2.3S**) against the PHONELAB reference version of the Android platform codebase, which contains our own tools and changes necessary to operate PHONELAB[3]. System developers will download and develop against these sources, ensuring that their patches are compatible. During this step we will ensure that experimental system images do not remove required PHONELAB functionality. Once we receive a user's patch, we will build and boot (**Figure 2.4S**) the new Android platform image locally. This is the first safety check and ensures that the experimental image compiles correctly. Systems developers can confirm that their patches will produce identical images by making sure that they are developing against the current PHONELAB reference image and on a compatible environment. We will distribute a PHONELAB Ubuntu virtual machine appliance to simplify this process.

After an experiment has been reviewed we must test systems components (**Figure 2.6S**) rigorously. The Android platform includes a set of stability and regression tests used by the Android development team themselves, and we are collaborating with Google to ensure that our system tests are as realistic and thorough as possible. This testing process also ensures that platform updates do not break system interfaces that applications and other experiments rely on. Testing system components carefully is critical to the success of PHONELAB as failures in systems components can effect all applications running on the phone and potentially even render the phone unusable. We guard against this both through pre-deployment testing, and by enabling participant opt-out, described below.

When a system image has been deemed safe and usable, it is distributed to participants (**Figure 2.8S**) and installed (**Figure 2.9S**) by exploiting much of the standard over-the-air (OTA) update mechanisms used by vendors to distribute new patches and system code. This mechanism does allow participants to delay updates, but only for a limited period of time; eventually we will force them to update their phone. Through our preliminary implementation, we have verified the feasibility of this approach. We have developed a tool that retrieves a new system image from a remote server and replaces the current image with the new image on the phone, leveraging much of the OTA implementation already in place. As an additional safety mechanism, we provide participants with the option of opting out (**Figure 2.10S**) of an installed system image update, either on their phone, if the phone is usable, or through the participant portal, if it is not. If we notice a large number of participants opting out of an image being distributed, we will flag the image as problematic and halt further updates until the issue is resolved. Ideally, PHONELAB systems experimentation can take place without participants or other users noticing at all.

6.4 — The Participant Experience: Being a Part of PHONELAB

When a participant joins PHONELAB, we inform them of our expectations: in exchange for a free or discounted phone, they are to devote two hours per week to interactive applications. At some point during the week, the participant logs on to the PHONELAB participant portal. She notices that she still has 20 minutes remaining to meet her weekly commitment. Alongside that notice are presented a number of different available experiments allow her to meet her commitment. Browsing short descriptions written by PHONELAB users, the participant selects one and heads for the bus.

On the way to her stop, the experiment is delivered to her phone. On board, she clicks on it and examines experimental description which highlights the data that will be collected about her along with her interaction with this application. Satisfied, she agrees, and a new context-awareness application utilizing the microphone launches. It asks her to choose the closest label representing what she is doing from a set of choices, which she does. Periodically throughout the day, the application collects more data.

When she returns home, the participant logs on to her portal again, sees that she has now met her weekly commitment and briefly examines the data collected by background experiments she has been participating in. She notices an upcoming PHONELAB Participant Feedback Event, which she plans on attending. At the last one she heard a very interesting presentation by researchers which had used PHONELAB data traces to analyze the structure of the campus social network and had found that talk very interesting. As she cradles her phone for the night, it begins offloading data collected by that day's experiments.

[3] We are already maintaining a reference version at `phonelab.cse.buffalo.edu`.

Type	Minutes	Per Month	Phone Price	Fees: Activation	Fees: Termination	Contract Length	4-Year Participant Cost	4-Year Participant Savings
Sprint Retail	450	$79.99	$99.99	$36.00	up to $350	2 years.	$3,975.51	—
UB Discount	450	$72.00	$99.99	$36.00	up to $350	2 years.	$3,591.99	9.65%
Sprint Employee Discount	500	$69.99	$149.99	$36.00	up to $350	2 years.	$3,545.51	10.82%
PhoneLab Plan	300	$41.50	Free	Free	Free	None.	$1,494.00	**62.42%**

Figure 3 — PHONELAB plan discounts compared to other discounted Sprint plans. As the table indicates, PHONELAB participants will be provided deeply discounted plans to encourage participation after their first year—42% off per month when compared with the standard SUNY Buffalo rate. Over their four years as part of PHONELAB participants will save a total of over $2,000.

6.5 — Attracting Participants Through a Unique Deal with Sprint

PHONELAB participants are expected to devote a portion of their time to participating in interactive experimentation. In addition, we will be using their background cycles and battery for background data collection and using them to test out experimental changes to the Android platform. Asking this of our participants requires that we provide them something in return. **We believe that the agreement we have negotiated represents considerable generosity by Sprint, a good value for the NSF and the research community and an extremely attractive deal for participants.**

Together with Sprint, we have negotiated a multi-year deal offering our participants an over 62% discount (see Figure 3) on their total cell phone bill during their four years at SUNY Buffalo. The deal is structured so that Sprint assumes 55% of the total cost of the incentives required to maintain PHONELAB in year three[4].

The deal is structured as follows. PHONELAB participants will be recruited the summer before their entrance to SUNY Buffalo as college freshman. They will be offered a free top-of-the-line smartphone and free voice and unlimited data coverage for one year as PHONELAB participants. This is our teaser offer, designed to initially attract them to the testbed. In our discussions with Sprint we were informed that many students enroll in college still covered by their parents plans, and so we designed a strong initial offer to lure them away and lure them to Sprint. All first-year PHONELAB participants will be carried on a single corporate plan paid for by SUNY Buffalo using the funds provided by this grant.

In the second year of PHONELAB participation participants will be transitioned from the single SUNY Buffalo funded corporate liable plan into individual liable but discounted voice and unlimited data plans with rates 42% better than they would otherwise receive. As long as they continue to participate in the testbed and remain at SUNY Buffalo, participants remain eligible for this special PHONELAB plan rate.

From the perspective of participants they are receiving $498 in direct subsidies from the NSF, $1,098 in discounts from Sprint and are themselves responsible for the remaining $1,494 of their estimated 4-year phone bill. The $498 from the NSF is in exchange for at least 60 and up to 240 hours of active PHONELAB interactive participation, an hourly rate of between $8 and $2 per hour: less than the going rate for psychology studies on campus even if the participant leaves after one year and much less if they continue for four.

We settled on this plan after considering several alternatives, among which we pick two alternatives and compare to our deal in Figure 4. The first option we considered was buying the phones outright and

[4] Calculated as follows. The NSF will be paying for 350 plans for first-year PHONELAB participants at a total cost to the NSF of $174,300. Sprint will be offering discounts worth $30.50 per month to the other 700 students participating in the testbed at a total cost to Sprint of $213,500.

Type	Size: Y1	Size: Y2	Size: Y3	Size: Steady	Cost: Y1	Cost: Y2	Cost: Y3	Cost: Total	Price Per Participant Year: Y1–Y3	Price Per Participant Year: Steady	Total Yearly Steady-State Cost
No Plans	1014	1014	1014	1014	$708,786	$0	$0	$708,786	$233	$175	$177,197
Free All Years	1014	1014	1014	1014	$504,972	$504,972	$504,972	$1,514,916	$498	$498	$504,972
PhoneLab Plan	350	595	816	1014	$174,300	$174,300	$174,300	$522,900	$297	$172	$174,300

Figure 4 — The PHONELAB plan negotiated with Sprint compared to other alternatives for offering incentives to participants. Buying phones for participants is as expensive as our deal and less attractive to participants who have to pay full-price for service. Paying for the entire contracts over the life of the testbed is cost-prohibitive.

distributing free phones to participants as the incentive, leaving them to pay for voice and data. However, this is an expensive option; it costs more to purchase an Nexus S 4G without a contract ($699.99) than the amount we have negotiated with Sprint for the phone plus data and voice coverage for one year ($498), which is the amount each participant would receive from NSF in our budget. A second option was to provide free phones with plans to all participants, but this was deemed too expensive as it would nearly triple the cost of operating PhoneLab. We have considered two other options, but do not detail them here due to space constraints.

6.5.1 — Scaling and Sustaining PhoneLab: Our goal is for PhoneLab to become a 1,000-phone testbed, and our proposal builds toward that level over three years. In order to give us time to scale the testbed as well as maintain a balanced participant community, we will recruit 350 new participants from the freshman class each year (after the initial year of development and testing), who we hope will continue as part of the testbed until they leave college. This approach evenly balances the participant population between freshman, sophomores, juniors and seniors; it also limits and balances the number of new PhoneLab participants we must recruit each summer.

The growth of PhoneLab can then be modeled under assumptions about our participant retention rate. We refer to the first-to-second year retention rate as R_1 and the second-to-third and third-to-fourth year retention rates as R_N. (We assume that undergraduates graduate after four years, making the fourth-to-fifth year retention rate 0.) We consider R_1 and R_N different because students experience a larger drop-off in incentives moving from the initial year (free plan) to the second year (discounted plan). In future years, they are simply continuing on with the same discounted plan they are already used to. Thus, we expect that in general $R_1 < R_N$. Figure 5 shows the size of the testbed in Year 3 under different retention rates, assuming that 350 new participants are recruited each year.

		R1										
		0.50	**0.55**	**0.60**	**0.65**	**0.70**	**0.75**	**0.80**	**0.85**	**0.90**	**0.95**	**1.00**
	0.50	613	639	665	691	718	744	770	796	823	849	875
	0.55	621	648	676	703	730	757	784	811	838	865	893
	0.60	630	658	686	714	742	770	798	826	854	882	910
	0.65	639	668	697	725	754	783	812	841	870	899	928
	0.70	648	677	707	737	767	796	826	856	886	915	945
Rn	**0.75**	656	687	718	748	779	809	840	871	901	932	963
	0.80	665	697	728	760	791	823	854	886	917	949	980
	0.85	674	706	739	771	803	836	868	900	933	965	998
	0.90	683	716	749	782	816	849	882	915	949	982	1,015
	0.95	691	725	760	794	828	862	896	930	964	998	1,033
	1.00	700	735	770	805	840	875	910	945	980	1,015	1,050

Figure 5 — PhoneLab size. Size of the testbed in Year 3 is displayed as a function of first-year retention (R_1) and future-year retention (R_n)

Note that due to the three-year duration of this grant PhoneLab will not reach our 1,000-participant goal at the levels budgeted. In order to sustain PhoneLab past year three, we will look for additional funding sources. By that point we expect that the research community will have seen the value of this testbed and may be willing to contribute directly. In addition, PhoneLab's success may allow us to negotiate a better deal with Sprint or another carrier. After three years, we will also have a much better understanding of the retention rates and how to attract participants, and can use that knowledge to design new ways to limit the yearly overhead of operating the testbed at our goal level.

6.6 — Protecting Participants

The most critical design issue for PhoneLab is protecting our participants. If SUNY Buffalo students lose trust in the safety of the testbed, they will not want to participate regardless of the incentives we provide. Thus, we will have several preventive measures to protect our participants, leveraging existing mechanisms that have shown to be effective or that participants or users are already familiar with. These include:

1. **Institutional Review Board:** We will require PhoneLab users have their experiments reviewed for human subjects safety by their local Institutional Review Board (IRB). If experiments do not have a local IRB, we will ask the SUNY Buffalo IRB to review their experiment.
2. **Android Application Privacy Settings:** Android applications must request permissions to use potentially sensitive participant data at install time. While this approach has weaknesses, our experimental applications will also use it as participants are familiar with it.
3. **PhoneLab Data Repository:** In order to limit redundant data collection, we will ask experiments to use cached PhoneLab participant data whenever possible. This is a significant benefit of PhoneLab

compared with other approaches to smartphone experimentation.

4. **PhoneLab Participant Portal:** At any time, PhoneLab participants can log in to their personal portal and review the data that is being and has been collected about them. We hope that by providing this transparency we will arm participants with knowledge necessary to choose what experiments to complete.

5. **Participant Feedback and Engagement:** Finally, throughout the testbed's operation we will be continually seeking and responding to participant feedback. At interval, we will hold open meetings to present the results of recent experimentation as well as address participant concerns.

Smartphone security is an active research area, and we expect that PhoneLab will help develop new tools enhancing smartphone safety which we will then incorporate into PhoneLab.

6.7 — Technical Components

The technical portions of PhoneLab are distributed between an interactive web frontend and other components running on backend servers and various PhoneLab client tools running on the smartphones themselves. This section provides an overview of these components and how they work together.

6.7.1 — User, Participant and Administrator Interfaces: All data about experiments, participants, and devices attached to PhoneLab will be stored in a database replicated across the PhoneLab backend servers. We are implementing the PhoneLab web frontend with interfaces for user, participant and administrator use. Currently, we are using Django as the web development framework and MySQL as the database.

The PhoneLab administrative interface is internal but important for smooth operation of the testbed. It allows the PhoneLab administrator, as well as the PIs and graduate research assistants assigned to the project to monitor the health of the testbed, view and respond to submitted experiments, and receive and act on feedback and issues generated by participants.

The participant-facing PhoneLab interface is the most important and the most critical to the testbed's success. Participants will log in to their participant portal to find new experiments to participate in, to monitor their own participation and ensure that they are meeting the targets they have committed to and to find out what data has been collected about them by which experiments.

Lastly, the user interface allows researchers to monitor the status of submitted experiments and collect data once experiments have completed executing.

6.7.2 — Experimental Distribution: Initiating PhoneLab experiments requires synchronizing the state of participating phones and potentially installing applications or adjusting the system image. PhoneLab builds a clean interface between server and phone state through the use of an *experimental manifest*. The manifest is an XML document that provides a complete description of how each PhoneLab phone should be configured, including what experimental applications should be installed, whether these applications should be run in the foreground or background and when they should be executed, and what Android platform image should be in use.

Client-server communication happens through the manifest. When the server makes a change to a PhoneLab phone's state, perhaps initiating a new experiment, it changes that phone's manifest and notifies the phone that a change has occurred; we will use a lightweight notification service provided by Google called C2DM [23]. A PhoneLab client running on the effected phone then downloads the new manifest and compares it with a local manifest describing its current state. It next takes the required actions necessary to implement the new manifest. This might mean downloading and installing new applications, or initiating an over-the-air system image update. When the client is finished updating the phone, it records the current state of the phone as its local manifest and waits for further updates.

6.7.3 — Monitoring Participation: As described in Section 6.2, PhoneLab participants participate in interactive experiments for two hours per week in exchange for free or deeply-discounted voice and data plans. For this purpose, each phone runs the PhoneLab participation tracker that monitors participation and reports it to the backend servers. This tracker also provides an interface to experimental applications allowing them to log participant participation and a score associated with a particular action.

6.7.4 — New Application and System Image Installation: Each phone also runs the PhoneLab installer that installs new applications (APK objects) and system images provided by experimenters. Installing a new APK does not present a challenge since we can follow the standard Android application installation process. However, installing new system images presents a challenge as it involves replacing the kernel, system libraries, or Dalvik Java virtual machine. In order to overcome this challenge, we have considered several options; however, we have discovered that we can reuse much of the Google's OTA (Over-The-Air) update implementation as the functionalities of OTA closely match with what we need—we have verified that leveraging much of the OTA is indeed possible through preliminary implementation.

6.7.5 — Data Logging: Data logging is a universal experimental requirement, and PhoneLab is built to support it in two ways: through a dedicated data logger collecting standard data on each phone, and through a generic data logging interface exposed to experimental applications.

The PhoneLab dedicated data logger is designed to support the common kinds of data that experiments might want to capture: position information, phone activity traces, records of application activity, etc. Rather than forcing each experiment to log this data separately, which would be wasteful and error-prone, PhoneLab allows experiments to gain access to these commonly-collected data streams through a check-box interface on the experimental submission web interface. An experiments choice of common data types to collect is reviewed by the PhoneLab administrator prior to experimental initiation; assuming that the data collection is justified, the PhoneLab data logger is triggered to begin specific data collection streams once the PhoneLab client or participant initiates an experiment. At that point, the dedicated data logger uses the generic data logging interface described next to return data to the PhoneLab backend.

While we expect that the generic data logger will meet many data collection needs, there are likely to be experiments that want to collect their own data. These experiments can use a generic data logging interface provided by a component running on each PhoneLab phone. Experimental applications can pass binary objects to the generic data logger which will store them locally and eventually transfer them reliably to the PhoneLab backend servers. This bulk data transfer will be done when the phone is being cradled and not in use to minimize participant disruption.

6.7.6 — External Tool Integration: Beyond data logging, a number of recent research projects have developed useful tools facilitating smartphone studies. These include power-monitoring components such as PowerTutor [20], network monitoring tools such as NetPiculet [37] and privacy-monitoring systems like TaintDroid [22]. We are working with the developers of these tools to attempt to integrate them closely into our PhoneLab design. Ideally, we would like to allow users to enable them as they submit their experiments by simply checking a checkbox on a web form. As new tools emerge in the smartphone research community we will work to incorporate them as well.

6.7.7 — Stability and Scaling: Providing high uptime is critical to PhoneLab's adoption and success. Here we describe how we handle both client and server failures as well as how we scale the server infrastructure as the lab grows.

On the client side, we must continually test the PhoneLab client to ensure its stability. In addition, Android provides mechanism for application failover that we will leverage, either to restart the PhoneLab client when it fails or contact the server if the client stops running and cannot be restarted. At worst, if we notice that we have lost contact with a PhoneLab device for a long period of time, the administrator will have to initiate physical contact with the participant to determine the root cause. It is also possible that the phone is lost, stolen, or has been disabled or out of cellular coverage areas for a long period of time.

On the server side, we will employ standard distributed systems replication and failover techniques to ensure high availability. Each phone will be assigned a single primary and several backup servers from our 20 server pool. When a phone is instructed to update its manifest or is trying to offload experimental data, it will try the primary first, followed by each of the backups. Thus, periodic failures or downtimes by individual machines can be tolerated. Large-scale failures of the PhoneLab servers are tolerable for a short duration. Although unable to offload data or initiate new experiments, existing experiments already deployed can continue to run and generate data on the phones themselves. When the availability of the infrastructure is restored, locally-cached data will be retrieved.

7 — Investigator Qualifications and Project Management

This section first presents our team's qualifications and results from prior NSF support. We continue by presenting our plan to rapidly develop and build out PhoneLab.

7.1 — Qualifications and Prior Support

Our team consists of members with complementary expertise on sensor networks, mobile computing, cloud computing and wireless networking. Many of the co-PIs have developed and maintained large-scale testbeds and infrastructure in the past and bring these experiences to PhoneLab. Below, we briefly highlight each co-PIs previous work relevant to this proposal:

7.1.1 — Geoffrey Challen: developed and maintained MoteLab [38], the first wireless sensor network testbed consisting of 200 sensor nodes and supporting over 700 users.

Challen has no prior or other pending NSF research grants. (He applied for and received funding to support travel grants to SenSys'10, NSF Award #1059586.)

7.1.2 — Murat Demirbas: helped develop and deploy the "Line In The Sand" [3] 100-node wireless sensor network for detection, classification, and tracking, which led to the 1,000-node "ExScal" [4] network. Demirbas' prior and current NSF projects include:

1. **CAREER: An In-Network Collaboration and Coordination Framework for Wireless Sensor Actor Networks** *(CNS-0747209, $371K, 02/01/08–01/31/12)*—This project led to the development of a transactional programming middleware and a singlehop cooperative communication primitive for wireless sensor networks [16, 18, 19, 31, 6].
2. **CSR: Small: Collaborative Research: Tool Support for Producing High Assurance and Reliable Software for Wireless Sensor Actor Networks** *(CNS-0916504, $250K, 09/01/09–08/31/12)*—This project developed tools allowing programs for Wireless Sensor Actor Networks (WSANs) to be written in high-level models traditionally used to describe distributed systems [5, 32].

7.1.3 — Steve Ko: helped design the HP/Intel/Yahoo! OpenCirrus^{TM} Cloud Computing Testbed [7], a federated multi-datacenter testbed spanning over 14 institutions in US, Europe, and Asia and including more than one thousand servers. Steve Ko has a pending NSF CAREER proposal.

7.1.4 — Tevfik Kosar: designed and developed both the Stork distributed data scheduling system [26] currently used by institutions worldwide and the PetaShare [36, 11] distributed storage network that manages more than 700 Terabytes of storage located across nine university campuses in Louisiana. Kosar's prior and current NSF projects include:

1. **MRI: Development of PetaShare: A Distributed Data Archival, Analysis and Visualization System for Data Intensive Collaborative Research** *(CNS-0619843, $957K, 08/15/06–07/31/2011)*—PetaShare is a distributed data storage and sharing system in active use by 30 different research projects spanning 12 different institutions. PetaShare has also been an important component in several other Louisiana state-wide cyberinfrastructure projects such as the NSF funded CyberTools and HPCOPS, Louisiana BoR funded LONI Institute, and DOE funded UCoMS projects [36, 11, 42, 41, 34, 33, 12, 13, 35, 10, 27].
2. **CAREER: Data-aware Distributed Computing for Enabling Large-scale Collaborative Science** *(CNS-0846052, $400K, 02/15/2009–01/31/2014*—this project aims to develop a new computing paradigm called data-aware distributed computing with integrated theory, research and educational components [25, 43, 40].
3. **STCI: Development of Stork Data Scheduler for Mitigating the Data Bottleneck in Petascale Distributed Computing Systems** *(OCI-0926701, $495K, 09/01/2009–08/31/2012)*—aims to develop and enhance the Stork data scheduler and make it available for a wide range of user community as a production quality software [26, 9].
4. **EAGER: Stork Data Scheduler for Azure** *(CCF-1115805, $97K, 04/15/2011–03/31/2013)*—continues development of the Stork data scheduler and enhances it to support the Azure cloud computing environment.

Chunming Qiao directs the Lab for Advanced Network Design, Analysis, and Research (LANDER). He was elevated to IEEE Fellow for his pioneering research on optical burst switching (OBS) networks, and integrated cellular and ad hoc relaying (iCAR) systems. Qiao's prior and current NSF projects include:

1. **EAGER: Create a Socially-aware Single System Image (S3I) in GENI** *(CNS-1049775, $129K, 09/01/10–08/31/12)*—this project focuses on the creation of friend/relative-based computing/storage clouds by enabling seamless resource sharing at based on Kerrighed OS. The fund supports one PhD student, who has attended every GENI Engineering Conference since 09/10, and made several poster presentations and demos.
2. **CPS: Medium: Addressing Design and Human Factors Challenges in Cyber Transportation Systems** *(CNS-1035733, $707K, 09/15/2010–08/31/2013)*—this project's focus in on the design and evaluation of novel CTS protocols and algorithms to improve traffic safety and traffic operations by taking human factors (HF) into consideration. The fund supported four part-time RAs and so far we have 1 journal and 6 conference/workshop papers.
3. **Collaborative Research: NeTS-NR: Ultra-Broadband Optical Wireless Communication Networks** *(CNS-0435155, $250K, 09/01/04–08/31/09)*— The project focuses on developing an ultra-broadband optical wireless network testbed, survivable topology and routing methods to increase the data rate to 1.25 Gbps for broadband access in the last mile. We have published several journal and conference papers. The project supported a PhD student. The research activities stimulated the formation of a new IEEE Technical Subcommittee on Optical/Wireless Integration for Access/Metro Networks which co-PI Qiao currently chairs.
4. **SGER-Exploring Sociological Orbits in Mobile Users' Mobility Pattern** *(CNS-0553273, $50K, 10/01/05–03/31/07)*—this project focused on exploring a partially deterministic (and repetitive) "orbital" pattern exhibited by wireless/mobile users. We have published one journal paper and four conference/workshop papers.

7.2 — Management and Operations Plan

Our goal is to have PHONELAB up and running with its first group of participants by Fall 2012 and publicly available one year later. In the first year, we plan to develop the core of the infrastructure and initiate operations with our first PHONELAB class of 350 participants. In the second year, after a year of testing we will double the size of PHONELAB and open it to public use. In the third year, we will continue to grow the size of PHONELAB while continuing development and operations by incorporating feedback from participants and users. This phased strategy has been used by successful testbeds such as EmuLab and PlanetLab; this allows us to start the operations early on and gives us an opportunity to adjust our management strategy as we receive feedback from our participants and users. Table 2 shows detailed tasks and their timeline.

7.2.1 — Year One: The goal of the first year is to build the core of the infrastructure and start the operations of the testbed with a small number of participants. After receiving funding, we will focus on the development, initiate the first round of phone acquisitions and distribute PHONELAB information to members of the SUNY Buffalo entering Class of 2016; in the second half of the year, we will roll out our lab management software to the first group of participants.

Project Personnel: After receiving funding we will recruit three graduate research assistants to work on the project. We have good candidates taking our smartphone development class and already familiar with PHONELAB development. We will also hire our full-time administrator in the first year, who will assist with PHONELAB development initially before moving into a half-development half-administration role.

Hardware Acquisition: We will acquire 20 servers and 350 phones in the first quarter of the first year, providing the resource necessary to start developing and testing core PHONELAB components.

Software Development: PHONELAB consists of server-side and phone-side components. We expect that the bulk of the core infrastructure will be developed in the second and third quarters of the first year. On the server side, we will (1) set up a development environment and (2) automated feedback tracking system; (3) develop the PHONELAB control server software that interacts with phones to deploy experiments and monitor participation; and (4) developing a web front-end with basic functionalities such as account management, experiment submission and status monitoring, and experimental data access.

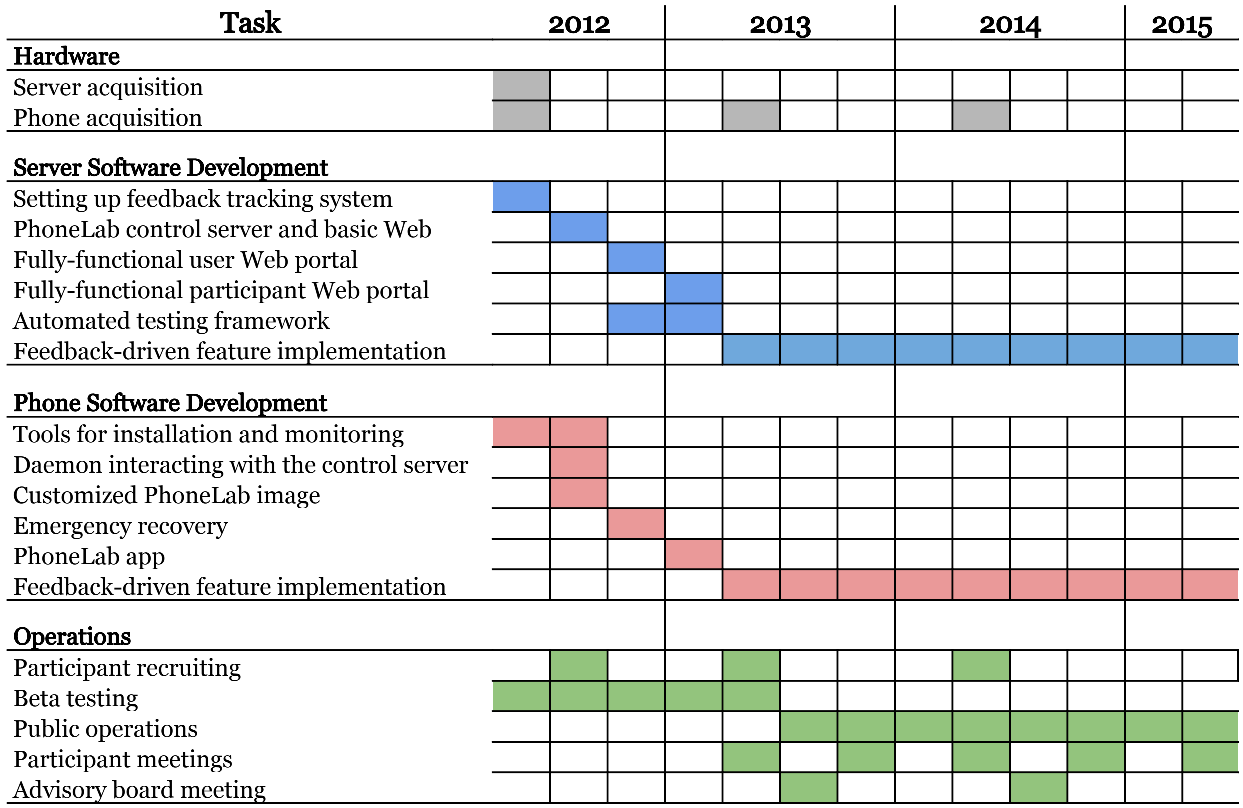

Table 2 — PhoneLab management plan. The timeline details development and operational tasks.

On the phone side, we will perform the following four core tasks: (1) developing a tool that installs or cleans up applications and system images, (2) developing a tool that monitors the current status of the phone, (3) developing a background daemon that utilizes the above tools; this daemon will interact with the PhoneLab control server in order to initiate installation/clean-up of applications and system images as well as to report the current status of the phone, (4) building a customized PhoneLab image that includes the above tools and the daemon; we will install this image on all PhoneLab phones before releasing them to participants.

As these tasks are individually well-defined and fairly isolated, we anticipate that we can distribute these tasks to our staff and students; they will perform these tasks under the guidance of the senior personnel.

Operations: We will start recruiting participants in the second quarter, distributing information about PhoneLab to incoming students. Given that this is our first group of participants and the testbed software will still be in its early stage, we will recruit technology-savvy participants first and emphasize that we are running a pilot program.

During the first year we will run our own experiments and stress tests on PhoneLab as test cases. During this time, we will evaluate the quality of service ourselves, actively soliciting feedback from participants and fixing problems as we encounter them. We will continue this process in the last month of the year. Finally, at the end of the first year, we will have the first PhoneLab meeting with our participants reporting our experience and soliciting extra feedback from the participants.

7.2.2 — Year Two: The main goal of the second year is to build a fully-featured, larger-scale testbed and make it available to the general research community. We will acquire 350 more phones, recruit more participants, and evaluate our feature set. From the beginning of year two, we will open the testbed to beta users—mainly our personal contacts starting from the supporter list. In the beginning of the second half of the year, we will open PhoneLab to the general research community. We will shift our focus from development to operations at that time.

Hardware Acquisition: We will acquire 350 additional phones in the first month and finish the configuration of these phones. Our staff and students will mainly perform this task.

Software Development: Our goal is to implement the complete set of planned features by the end of the first half. As we will open the testbed to beta testers in the first half of the year, we will perform feedback-driven development in addition to the planned feature implementation.

On the server side, we will implement the following three features: (1) a complete web portal for users that shows the history of experiments, the current PHONELAB activities and utilization, and reservation of phones for future experiments, (2) a complete web portal for participants that shows available interactive experiments, the history of experiments participated, the current level of participation, and results produced through the PHONELAB experiments, and (3) a fully-automated testing framework that stress-tests applications and platform images before deployment.

On the phone side, we will implement the following two features: (1) fully-automated emergency recovery that provides a one-touch recovery functionality to participants when their phones malfunction, and (2) a PHONELAB application for participants customized for the phone UI that displays the information available through the participant portal.

Due to the shift of focus from development to operations in the second year, we expect that the staff member will mostly direct the development and guide students. The senior personnel will focus more on operations.

Operations: The second year will mark our transition to a fully-operational testbed. In the first half of the year, we will open PHONELAB to beta testers, starting from the supporters. As with the first group of participants, we will put extra care in communicating with beta testers; we will especially make sure that they understand our privacy protection policies and actively solicit feedback from them. We will reflect this feedback in our software development. In addition, we will be moving first-year participants into the discounted Sprint individual plans, and except to have to manage this process carefully to retain as many participants as possible.

In the beginning of the second half, we will publicly announce the availability of PHONELAB through mailing lists and at relevant conferences. From this time, we will focus more on operations; main tasks will involve user and participant support, maintenance and monitoring of phones and servers, and quarterly PHONELAB meetings with participants. Based on the operational experience during this time, we will document best practices of running experiments on PHONELAB and make the documents available to users.

7.2.3 — Year Three: The goal of the third year is to develop more features from user feedback as well as continue our operations with more phones. We anticipate that most of our efforts will focus on operations and user support. We also anticipate that our development will mostly be incremental and the testbed will be operational 24/7.

Hardware Acquisition: In the first month of year three, we will acquire 350 more phones and configure them. This will finish our hardware acquisition and PHONELAB will operate in its full scale. Our staff will mainly perform this task.

Software Development: Our development effort in the third year will be feedback-driven. We will continue to work with users and participants to implement features necessary for them. Our staff will largely oversee this task with some help from students.

Operations: Operations will be the main task for year three as we expect that PHONELAB will be operational 24/7. This will require significant maintenance and user support, i.e., responding to emails, updating documentation, and replacing failed hardware, with these tasks under the primary supervision of the PHONELAB administrator. With 350 additional phones, we will continue recruiting participants.

REFERENCES

[1] 3gtest, mobiperf official website. http://mobiperf.com.

[2] Mob4hire - crowdsourced mobile application testing. http://www.mob4hire.com.

[3] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y-R. Choi, T. Herman, S. S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita. A line in the sand: A wireless sensor network for target detection, classification, and tracking. *Computer Networks (Elsevier)*, 46(5):605–634, 2004.

[4] A. Arora, R. Ramnath, E. Ertin, P. Sinha, S. Bapat, V. Naik, V. Kulathumani, H. Zhang, H. Cao, M. Sridharan, S. Kumar, N. Seddon, C. Anderson, T. Herman, N. Trivedi, C. Zhang, M. Nesterenko, R. Shah, S. Kulkarni, M. Aramugam, L. Wang, M. Gouda, Y. Choi, D. Culler, P. Dutta, C. Sharp, G. Tolle, M. Grimmer, B. Ferriera, and K. Parker. Exscal: Elements of an extreme scale wireless sensor network. *Int. Conf. on Embedded and Real-Time Computing Systems and Applications*, 2005.

[5] M. Arumugam, M. Demirbas, and S. Kulkarni. Slow is fast for wireless sensor networks in the presence of message losses. In *12th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS)*, 2010.

[6] J. Aspnes, E. Blais, M. Demirbas, R. O'Donnell, A. Rudra, and S. Uurtamo. k+ decision trees. *International Workshop on Algorithms for Sensor Systems, Wireless Ad Hoc Networks and Autonomous Mobile Entities (Algosensors)*, 2010.

[7] Arutyun I. Avetisyan, Roy Campbell, Indranil Gupta, Michael T. Heath, Steven Y. Ko, Gregory R. Ganger, Michael A. Kozuch, David O'Hallaron, Marcel Kunze, Thomas T. Kwan, Kevin Lai, Martha Lyons, Dejan S. Milojicic, Hing Yan Lee, Yeng Chai Soh, Ng Kwang Ming, Jing-Yuan Luke, and Han Namgoong. Open cirrus: A global cloud computing testbed. *IEEE Computer*, 2010.

[8] Aruna Balasubramanian, Ratul Mahajan, and Arun Venkataramani. Augmenting mobile 3g using wifi. 2010.

[9] M. Balman and T. Kosar. Error detection and error classification: Failure awareness in data transfer scheduling. *International Journal of Autonomic Computing (IJAC)*, 2010.

[10] M. Balman and T. Kosar. Data scheduling for large scale distributed applications. In *the 5th ICEIS Doctoral Consortium, In conjunction with the International Conference on Enterprise Information Systems (ICEIS'07)*. Funchal, Madeira-Portugal, June, 2007.

[11] M. Balman, I. Suslu, and Kosar T. Distributed data management with petashare. In *IIn Proceedings of 15th ACM Mardi Gras Conference*, Baton Rouge, LA, February 2008.

[12] E. Ceyhan, G. Allen, C. White, and T. Kosar. A grid-enabled workow system for reservoir uncertainty analysis. In *Proceedings of Challenges of Large Applications in Distributed Environments (CLADE 2008) Workshop*, Madeira, Portugal, June 2007.

[13] E. Ceyhan and T. Kosar. Large scale data management in sensor networking applications. In *In Proceedings of Secure Cyberspace Workshop*, Shreveport, LA, November 2007.

[14] Geoffrey Challen and Mark Hempstead. The case for power-agile computing. In *Proc. of the 13th Workshop on Hot Topics in Operating Systems (HotOS-XIII)*, May 2011.

[15] E. Cuervo, P. Gilbert, W. Bi, and L. Cox. Crowdlab: An architecture for volunteer mobile testbeds. *Communication Systems and Networks (COMSNETS)*, pages 1–10, 2011.

[16] M. Demirbas. A transactional framework for programming wireless sensor/actor networks. *Proceedings of the 11th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS)*, pages 123–129, 2007.

[17] M. Demirbas, M. A. Bayir, C. G. Akcora, Y. Yilmaz, and H. Ferhatosmanoglu. Crowd-sourced sensing and collaboration using twitter. *11th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2010.

[18] M. Demirbas, O. Soysal, and M. Hussain. Singlehop collaborative feedback primitives for wireless sensor networks. *INFOCOM*, pages 2047–2055, 2008.

[19] M. Demirbas, S. Tasci, H. Gunes, and A. Rudra. Singlehop collaborative feedback primitives for threshold querying in wireless sensor networks. *25th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2011.

[20] M. Dong and L. Zhong. Self-constructive, high-rate energy modeling for battery-powered mobile systems. *Proc. ACM/USENIX Int. Conf. Mobile Systems, Applications, and Services (MobiSys)*, 2011.

[21] N. Eagle and A. Pentland. Reality mining: sensing complex social systems. *Personal Ubiquitous Comput.*, 10:255–268, 2006.

[22] William Enck, Peter Gilbert, Byung-Gon Chun, Landon P. Cox, Jaeyeon Jung, Patrick McDaniel, and Anmol N. Sheth. Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones. In *Proceedings of the 9th USENIX conference on Operating systems design and implementation*, OSDI'10, pages 1–6, Berkeley, CA, USA, 2010. USENIX Association.

[23] Google. Android Cloud to Device Messaging Framework. `http://code.google.com/android/c2dm/`.

[24] Tim Kaler, John Patrick Lynch, Timothy Peng, Lenin Ravindranath, Arvind Thiagarajan, Hari Balakrishnan, and Sam Madden. Code in the air: simplifying sensing on smartphones. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 407–408, New York, NY, USA, 2010. ACM.

[25] T. Kosar and M. Balman. A new paradigm: Data-aware scheduling in grid computing. *Future Generation Computing Systems (FGCS)*, 25(4):406–413, 2009.

[26] T. Kosar, M. Balman, I. Suslu, E. Yildirim, and D. Yin. Data-aware distributed computing with stork data scheduler. In *SEE-GRID-SCI'09*, December 2009.

[27] M. Balman and T. Kosar. From Micro- to Macro-processing: A Generic Data Management Model. In *Poster presentation, The 8th IEEE/ACM International Conference on Grid Computing*. (Grid2007)Austin, Sept 2007.

[28] M. Demirbas M. F. Bulut, Y. S. Yilmaz. Crowdsourcing location-based queries. *IEEE Workshop on Pervasive Collaboration and Social Networking (PerCol)*, 2011.

[29] Erik Nordstrom, David Shue, Prem Gopalan, Matvey Arye, Steven Ko, Jennifer Rexford, and Michael J. Freedman. Avoiding the rush hours: Wifi energy management via traffic isolation. 2011.

[30] Erik Nordstrom, David Shue, Prem Gopalan, Matvey Arye, Steven Ko, Jennifer Rexford, and Michael J. Freedman. Serval: An end-host stack for service-centric networking. 2011.

[31] O. Soysal, B. I. Aydin, and M. Demirbas. Optimistic concurrency control for multihop sensor networks. *IWCMC*, pages 89–94, 2011.

[32] O. Soysal, S. Ayyorgun, and M. Demirbas. Powernap: An energy efficient mac layer for random routing in wireless sensor networks. *IEEE SECON*, 2011.

[33] I. Suslu and T. Kosar. Balancing the use of remote i/o versus staging in distributed environments. In *Proceedings of the 9th International Conference on Enterprise Information Systems Doctoral Symposium (DCEIS 2007)*, New Orleans, LA, September 2008.

[34] I. Suslu, F. Turkmen, M. Balman, and T. Kosar. Choosing between remote i/o versus staging in large scale distributed applications. In *To appear in Proceedings of International Conference on Parallel and Distributed Computing and Communication Systems*, New Orleans, LA, September 2008.

[35] S. Tummala and T. Kosar. Data management challenges in coastal applications. *Journal of Coastal Research*, special Issue No.50:1188–1193, 2007.

[36] X. Wang, D. Huang, I. Akturk, M. Balman, G. Allen, and T. Kosar. Semantic enabled metadata management in petashare. *Journal of Grid and Utility Computing (IJGUC)*, 2009.

[37] Zhaoguang Wang, Zhiyun Qian, Qiang Xu, Zhuoqing Morley Mao, and Ming Zhang. An untold story of middleboxes in cellular networks. In *SIGCOMM*, pages 374–385, 2011.

[38] Geoff Werner-Allen, Pat Swieskowski, and Matt Welsh. MoteLab: A Wireless Sensor Network Testbed. In *Proc. of the 4th International Conference on Information Processing in Sensor Networks (IPSN 2005)*, April 2005.

[39] Wikipedia. Smartphone. `http://en.wikipedia.org/wiki/Smartphone`.

[40] D. Yin E. Yildirim, S. Kulasekaran, B. Ross, and T. Kosar. A data throughput prediction and optimization service for widely distributed many-task computing. *To appear in IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 2011.

[41] E. Yildirim, M. Balman, and T. Kosar. Dynamically tuning level of parallelism in wide area data transfers. In *Proceedings of Data-aware Distributed Computing (DADC)*, Boston, MA, June 2008.

[42] E. Yildirim, I. Suslu, and T. Kosar. Which network measurement tool is right for you? a multidimensional comparison study. In *In Proceedings of GRID'08*, Tsukuba, Japan, September 2008.

[43] E. Yildirim, D. Yin, and T. Kosar. Prediction of optimal parallelism level in wide area data transfers. *To appear in IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 2011.