Verification and Validation of Smartphone Sensor Networks

Hamilton Turner and Jules White

Dept. of Electrical and Computer Engineering
Virginia Tech
Blacksburg, VA, 24060, USA
{hturner0,julesw}@vt.edu

Abstract. This paper introduces a subset of mobile wireless sensor networks, called smartphone sensor networks, where large numbers of smartphone devices cooperate to perform sensing tasks. While these emerging networks show high potential, little work has been done on design-time verification and validation to ensure that a designed system will meet the specified goals. This paper introduces Empower, a simulation environment for smartphone sensor networks that simulates smartphone-specific properties of a sensor network, such as data collection policies, and outputs high-level system metrics, such as coverage of the environment being monitored. Experimentation is used to demonstrate that Empower's ability to derive system design parameters, such as the minimum number of smartphones required for proper operation, or the most appropriate data collection policy for the production environment.

Keywords: smartphones, mobile wireless sensor network, simulation.

1 Introduction

From 2009 to 2010, smartphones sales increased by 96% worldwide [1]. The resultant rise in the number of available smartphones has generated interest in the area of smartphone sensor networks, where consumer-owned smartphone devices are utilized as sensing platforms, and sensing results are aggregated into a coherent final product, such as a map of noise pollution in an urban environment. The primary motivations for using smartphones as sensor network nodes include: available sensing hardware such as GPS chipsets and modern cameras, significant local storage and processing power, access to a network infrastructure, end-user maintenance and upkeep, frequent battery recharging, and ubiquitous deployment platforms for dispersing applications to end-users [2]. Moreover, the natural distribution and mobility of end-users provides an ideal environment for data collection.

Numerous smartphone sensor networks have been developed, such as applications to track and analyze CO_2 emissions [3], detect traffic accidents and provide situational awareness services to first responders [4,5], measure traffic [6], and monitor cardiac patients [7]. Additionally, citizens living in the Gulf Coast region have been using smartphone sensors, such as cameras and GPS, to enter

N. Venkatasubramanian et al. (Eds.): Mobilware 2011, LNICST 93, pp. 233–247, 2012.

data on the ecological impact of the Gulf Oil spill, thus providing scientists with a wealth of field data on this disaster that can then be used to generate impact analysis and recommendations [8]. A key challenge, however, is that verification and validation of smartphone-powered sensor networks is hard. As shown in Figure 1, there are a large number of components contained in a smartphone sensor network, including the smartphones and their associated properties (e.g., movement, location accuracy at a given time and location, available battery life), the environment being monitored, the algorithms to aggregate the incoming sensor data, and other components.

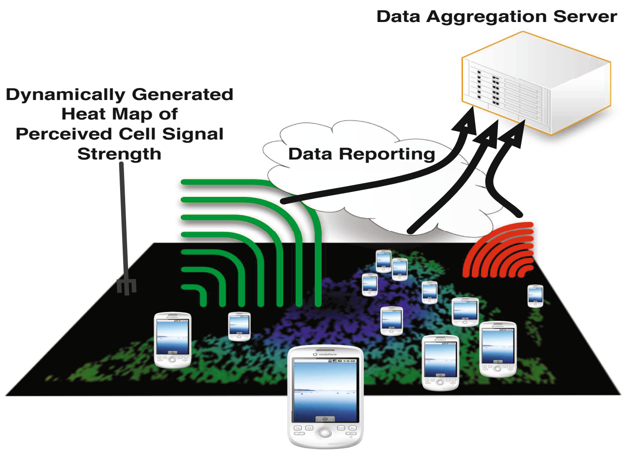

Fig. 1. Architecture of a Smartphone-based Cellular Coverage Mapping Application

Open Problem ⇒ Emergent and context-dependent aspects of smartphones make verification and validation challenging.

Because smartphone sensor networks rely on end-user smartphones, then introduce challenges that are not present in a more traditional wireless sensor network model, including complex system properties such as system adoption rate and cellular network infrastructure, context-dependent aspects such as unanticipated smartphone reconfiguration, and emergent system properties. Due to these issues, it is tough for smartphone sensor network developers to have confidence that the chosen system architecture, protocols, and policies will work as expected and desired in the environment being monitored. Additionally, the costly nature of changing software system designs late in the development cycle motivates making good system design decisions as early as possible.

Verification and validation of systems early in the development is critical to reduce the overall system cost, and critical to ensure successful creation of a smartphone sensor network that will properly monitor the environment of interest. Typical verification and validation of a sensor network (SN) is done either through formal methods, using tools such as Real-Time Maude [9], or model checking, using tools such as SPIN [10]. Neither of these approaches are feasible for verifying and validating smartphone sensor networks because they fail to account for the context-dependent aspects of smartphones, such as movement of the smartphone by its end-user. Moreover, wireless sensor networks have primarily been investigated in the context of small, very low-power sensors communicating through an ad hoc network, usually not interacting with an end-user, and with very limited computing resources.

Solution Approach ⇒ Simulation of smartphone data collection systems to enable design-time verification and validation. To address the need to verify and validate smartphone sensor networks, we present a platform for the *E*valuation of *M*obile *P*h*O*ne *W*ireless *E*nvironmental data *R*eporting, or 'Empower'. Empower allows design time verification and validation of smartphone sensor networks, ensuring the system meets specifications and fulfills the posed systems goals, taking into account complex system parameters and policies, such as rate of change in the number of participating smartphones, run-time smartphone context changes, and data collection/reporting policies. In Section 5 we present empirical data showing Empower can also be used to identify emergent properties of smartphone sensor networks.

This paper provides the following contributions to the study of smartphone sensor networks:

- we introduce and describe Empower, which is a simulation environment for smartphone sensor networks,
- we describe Empower's formal simulation model,
- and we present emperical data from experiments from using Empower to verify and validate a smartphone-based sensor network for dynamically mapping cellular coverage

The remainder of this paper is organized as follows: Section 2 describes a continuous map of cellular network coverage, which we use as a motivating example throughout the paper; Section 3 discusses the challenges that are faced when attempting to verify the effectiveness of a designed smartphone data collection system; Section 4 covers the Empower framework; Section 5 presents empirical results from analyzing multiple designed smartphone data collection systems and demonstrates Empower identifying system failures and bottlenecks; and Section 7 presents concluding remarks and lessons learned.

2 Motivating Example: Continuous, Accurate Measurement of Cellular Network Coverage

In order to motivate the challenges associated with verifying and validating smartphone sensor networks, we present a motivating smartphone application for dynamically mapping cellular network coverage using end-user smartphones. In order to create a smartphone sensor network that accurately monitors cellular network coverage and conforms rapidly to any changes in coverage, there are a number of needed system components, many of which were shown in Figure 1.

First, a smartphone application must be programmed and deployed to numerous smartphones, which are preferably well geographically distributed. As smartphones are moved geographically, they use the built-in cellular chipsets to sample the cellular coverage in their region. Figure 2 shows Empower visualizing the simulated real cellular network coverage map. When a smartphone's internet connection is enabled, these coverage readings, along with meta information describing the location and time of each reading, the network chipset on the smartphone, and the particular operating system being run, can be transmitted to a centralized cloud-based data aggregation server. As readings are entered, data aggregation algorithms are used to combine the data into a smaller format that can then be represented as a coverage map. Figure 3 shows a cellular coverage map Empower has generated by aggregrating the incoming network coverage readings. System policies can be used to control various aspects of the process, such as the rate of data collection, the speed at which data becomes outdated, or the method used to aggregate the incoming data.

3 Challenges of Verifying and Validating Smartphone Wireless Sensor Networks

Smartphone wireless sensor networks (WSN) have a large number of potential applications, such as real-time traffic monitoring, accurate weather monitoring, or rapid network analysis after a disaster. Unfortunately, it is difficult to ensure that a set of smartphone data collection policies will meet the desired objectives, as the success of these systems is heavily dependent on emergent properties of end-user smartphones. For example, the number, distribution, and movement of smartphones influences coverage, the properties of the network being used to transmit data impact timeliness of data reporting, and the change in the environment heavily impacts the rate at which data ages. When developing system policies to deal with these and other issues, developers face a number of challenges.

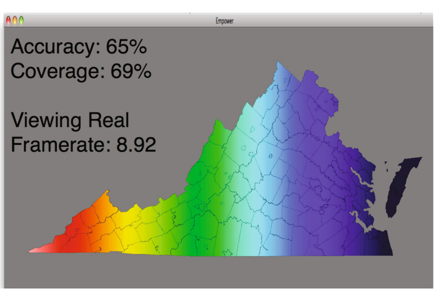

Fig. 2. Empower Visualization of Simulated Real Cellular Network Coverage

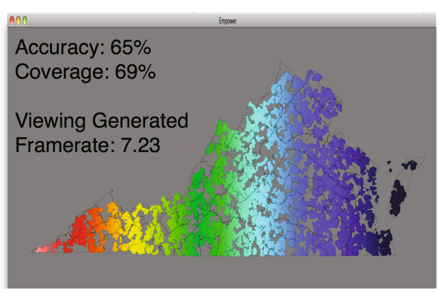

Fig. 3. Empower Visualization of Generated Cellular Network Coverage Map

3.1 Challenge 1: Unpredictable Smartphone Availability Makes It Hard to Estimate the Minimum Number of Nodes Required to Ensure System Goals Are Reached

The availability of nodes e.g. smartphones in a smartphone sensor network is highly unpredictable, as smartphones may appear and disappear rapidly from the network in a complex fashion dictated by their end-users, and the overall trend in system adoption can follow different patterns. While the number of smartphones participating in a smartphone WSN clearly has an impact on the probability of system success, it is tough to know how significant the impact is. Additionally, it is hard to know the number of smartphones required, on average, for the system to meet a specific goal, such as acquiring 80% coverage of the environment.

For example, a key verification and validation challenge of building a smartphone WSN to dynamically map a cellular network is determining how many smartphones are needed on average to generate a coverage map that is 80% accurate. Given the number of complex factors included in determining the exact number of smartphones available to participate in generating a map of the cellular network, such as population measurements and adoption studies, it is critical that system designers have a concrete number for the minimum density of participating smartphones required in order to assess the risk of system failure.

3.2 Challenge 2: The Complex and Emergent Properties of Smartphone Based Opportunistic Sensing Systems Make Determining the Impact of Various Policy Decisions Difficult

In order to verify and validate a smartphone powered WSN, it is important to understand the impact of system policies, such as data collection controls, on system metrics, such as the accuracy of incoming data or the utilization of system resources. The complex relationship between system policies, system parameters, and emergent properties makes it difficult to know what system policies should be used in which situations, or even which policies can be used without violating system constraints.

For example, if a system to map cellular network coverage was powered by postal service workers carrying smartphones, a data collection and reporting policy that took too many superfluous readings and wasted battery power could be expensive or cause premature battery exhaustion during the workday. Further, if the data collection policy only collects a limited set of measurements each day, a policy that chooses optimal locations for data readings may have a direct impact on the system success or failure. However, verifying and validating that the location selection method is efficient and selects the appropriate data samples across thousands of phones is hard. Section 4.3 describes how Empower aids validating these design decisions by allowing system designers to plug and simulate custom data collection and reporting policies across thousands of smartphones.

4 Empower: A Simulation Environment for Verifying and Validating Smartphone Sensor Networks

To address the need to verify and validate smartphone sensor networks, we present Empower, a simulation environment for the Evaluation of Mobile PhOne Wireless Environmental data Reporting. As shown in Figure 1, smartphone powered opportunistic sensing systems contain multiple configurable components. Empower can be used to model various aspects of these systems, including: the smartphones in an environment, the network being used to transmit information between the data collection server and the clients, various system-level policies such as data collection and reporting policies, the algorithms used for data evaluation, system-level metrics, and system goals.

Figure 4 shows an overview of the Empower application, including the visualization that Empower uses to represent a given environment, the formal model that is used to back the executing simulation, the output (which si both shown on-screen and logged), as well as small visualizations for each smartphone object. In section 4.2 we discuss how Empower can be used by developers to derive critical system design parameters, focusing on the minimum number of smartphones required to achieve system goals as a design parameter of interest. In section 4.3, we discuss Empower's approach to system-level policies, such as data collection frequency and data reporting decisions, and show how Empower can be used to identify the most appropriate policies for various operational environments.

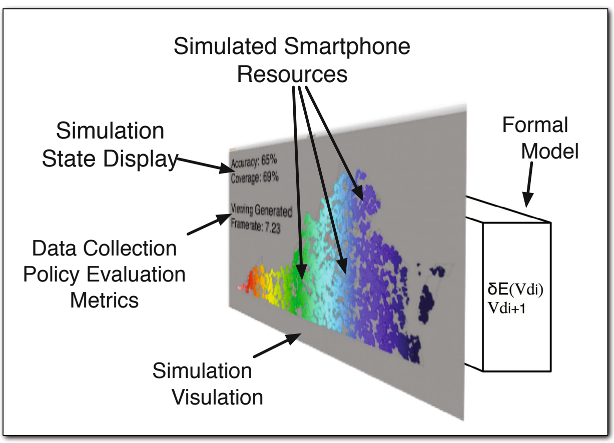

Fig. 4. Overview of Empower Software

4.1 Empower Formal Simulation Model

Empower uses a simulation engine based on a formal model for smartphone-based opportunistic sensing. The model is based on the 8-tuple:

$$M =< E, D, \boldsymbol{Ve_0}, Vd_0, \delta E(\boldsymbol{Ve_i}), \delta D(Vd_i), O(Ve_i, Vd_{ij}), M(Ve_i, \omega_i) > \quad (1)$$

$$|E| = |Ve_i| \tag{2}$$

$$\forall \boldsymbol{Vd_{ij}} \in Vd_i, |\boldsymbol{Vd_{ij}}| = |D| \tag{3}$$

$$\delta E(\boldsymbol{Ve_i}) = \boldsymbol{Ve_{i+1}} \tag{4}$$

$$\delta E(\boldsymbol{Vd_i}) = \boldsymbol{Vd_{i+1}} \tag{5}$$

$$O(Ve_i, Vd_{ij}) = \boldsymbol{\omega_{ij}} \tag{6}$$

$$M(Ve_i, \omega_i) = \boldsymbol{Met_i} \tag{7}$$

where:

- E is the set of parameters or variables that describe the environment that the smartphones are operating in.
- D is the set of parameters that describe each smartphone in the simulation.
- Vd_i is a set of vectors containing the parameter values describing the state of each smartphone at time i.
- $\boldsymbol{Vd_{ij}} \in Vd_i$ is a vector describing the state of the j_{th} smartphone at time i.
- $\delta E(Vd_i)$ is a function that maps the state of every smartphone at time i, Vd_i, to their new states at time $i+1$, Vd_{i+1}.
- $\delta D(\boldsymbol{Ve_i})$ is a function that maps the state of the environment at time i, $\boldsymbol{Ve_i}$, to its new state at time $i+1$, $\boldsymbol{Ve_{i+1}}$.
- $\boldsymbol{\omega}$ is a set of metric outputs, such as the overall coverage and accuracy of the perceived network, at time i.
- $O(Ve_i, Vd_{ij})$ is the smartphone sensor sampling function which determines the sensor values, $\boldsymbol{\omega_{ij}}$, read by the j_{th} smartphone at time i. The sensor values that are produced are a function of the state of the smartphone, Vd_{ij}, and the state of the environment, Ve_i.
- ω_i is the set of all smartphone sensor value vectors at time i.
- $M(Ve_i, \omega_i)$ is a function that, based on the current state of the environment and set of sensor outputs from all smartphones, calculates the values, $\boldsymbol{Met_i}$, for the verification and validation metrics of interest. For example, the overall accuracy of the perceived cellular signal strength map versus the actual cellular signal strength map could be an output metric.

The core of the formal simulation model are the functions $\delta E(Vd_i)$ and $\delta D(\boldsymbol{Ve_i})$. These functions evolve the states of the smartphones and the environment over the course of the simulation. Empower includes implementations of the functions that are optimized for modeling environments that change based on geographic coordinates and the movement of smartphones through those areas. Both functions can be customized or replaced for simulations that do not focus on sensor data collection that is tied to the geolocation of a smartphone. Section 4.2 discusses the parameters and operation of these functions.

In order to verify and validate a set of data collection and reporting policies in a specific smartphone scenario, developers must be able to incorporate their application-specific policies into the Empower formal model. The function $O(Ve_i, Vd_{ij})$ is designed to be configured by the user to accurately model the data collection and reporting policies of the application. At each time step,

$O(Ve_i, Vd_{ij})$ determines what data is sampled, based on the sampling policy of the application. Moreover, $O(Ve_i, Vd_{ij})$ also determines which of these samples are actually reported to the aggregation system by the smartphone by limiting the values that are output into ω_{ij}. For example, although the data collection policy may determine that a particular sensor is sampled at a given timestamp, $O(Ve_i, Vd_{ij})$ may not output that sensor value until a later time step to simulate buffering of data on the phone. The base implementation of $O(Ve_i, Vd_{ij})$ is described in Section 4.3.

The final critical aspect of the formal model is the set of metrics that are calculated by Empower and used for verification and validation. The function $M(Ve_i, \omega_i)$ can be adjusted to calculate any metrics that are a function of the environment state, smartphone state, or sensor values.

4.2 Solution 1: Using Simulation of Smartphones to Derive System Design Parameters

Section 3.1 introduces the system design parameter of minimum required number of nodes. In order to assist system designers in identifying the minimum number of smartphones required to ensure that system goals are met, Empower allows designers to vary the minimum number phones in a smartphone sensor network, and then simulate the system to determine the probability of system goals being reached. By varying simulation parameters, developers can identify a baseline for the minimum number of smartphones required to achieve the system goals that is specific to the future production environment of their smartphone sensor network, such as a highly-dynamic environment with a low density of highly mobile smartphones that frequently input data readings.

Empower allows design time control of the following smartphone properties:

- **Initial number of smartphones** in the system. System designers specify this as a constant number. For some situations, such as a mandatory adoption policy, this number can be significantly high. For other systems, the rate of adoption (discussed below) is more important.
- **Rate of Adoption** determines how many smartphones are participating in the opportunistic system at various times. This is controlled by a designer-specified function that accepts time, which is tracked by the simulation environment and used to relate the simulation results to a real-world timescale, and returns the approximate number of smartphones that should currently be inside of the data collection system. By implementing the rate of adoption function, a designer can arbitrarily control the number of smartphones in the system at any given time.
- **Smartphone Properties and Settings**, such as smartphone location or network connectivity enabled versus disabled, can be specified by the system designer using high-level functions. Empower controls configuring individual smartphone objects, ensuring that overall distributions match the desired settings. For example, a developer can specify that 60% of the smartphones in a network should initially have their GPS chip enabled. Designers can control each

smartphone property by specifying a function that accepts time as a parameter and returns the appropriate distribution of that property at the given time.

- **Frequency and Magnitude of Smartphone Movement** can be controlled via developer-specified functions. Developers can input high-level values specifying how frequently smartphones move during the simulation execution, and how significant each move is in terms of meters travelled per movement. This is a fairly simple model of movement, and the authors are actively attempting to build a more flexible movement function that would allow integration of research on human movement patterns, with a goal of allowing system developers to simply select a pre-defined movement pattern which has been vetted by prior research.

By testing numerous property configurations, system designers can understand which properties are most critical to data collection and reporting policy success and take steps to ensure that those properties are met. For example, Empower can allow a system designer to pinpoint the minimal number and diversity of smartphones required for the system to achieve the metrics of interest by modifying these configuration parameters and executing the simulation. This information is quite valuable for the successful deployment of such a system.

4.3 Solution 2: Exposing Policy Decisions As a Configurable Simulation Property

As Section 3.2 outlines, the complex parameters and emergent properties of a smartphone-powered opportunistic sensing system make it difficult to understand the effects of system-level policy decisions. To address this issue, Empower allows various system-level policies to be configured, both at the beginning of the simulation and during the simulation execution. This enables system designers to verify that their chosen policies will improve metrics of interest under the conditions they believe most likely to occur in the operational environment. By testing these different policies with various parameter combinations, system designers are able to determine the most optimal policy for different operational environments. Moreover, it is possible to identify various emergent properties and understand the impact of those properties on system metrics.

Empower currently allows two system-level policies to be specified:

- **A Data Collection Policy** for the smartphones in the system can be specified. This policy determines which smartphones attempt to collect data, based on the properties of each smartphone. In the experiments outlined in Section 5, we compare a data collection policy that favors highly-mobile smartphones for data collection over less mobile phones to a mobility-independent data collection policy. System designers can specify the data collection policy by creating a function which accepts as input the properties of a smartphone (current location, available sensors, mobility frequency and magnitude) and returns a boolean value indicating if that smartphone should attempt to collect data.
- **A Data Reporting Policy** determines how smartphones report data back to the central server. Interesting issues that this policy aims to address are the

limited network connectivity on smartphone platforms (both in terms of network availability and network throughput), the rate at which data becomes useless after it has been collected, and the amount of available persistent storage on individual smartphones. System designers can specify a function that receives the amount of available storage, the current network connectivity, and all other smartphone properties. Based on these properties, the function can return that data should be discarded, cached, or sent to the data collection server.

5 Using Empower to Determine System-Level Policies

In this section we present results from experiments performed using Empower that evaluate the impact of data collection policy decisions in the smartphone-based cellular network mapping example from Section 2, such as data collection policy, on overall system properties such as accuracy and wastefulness. Additionally, we perform experiments to derive system design parameters, such as the minimum number of smartphones required to achieve system goals. These results show that Empower can be used to verify the design of a smartphone sensor network by ensuring that system goals are met. Moreover, these results show that Empower can be used to determine critical system design parameters.

5.1 Experimental Platform

These experiments were performed on a 2.66 GHz Intel Core i7 machine with 4 GB of 1067 Mhz DD3 ram, running the Mac OS X 10.6.6 operating system. Empower is written with Java, and the Eclipse IDE was used to both aid development and to run the simulations.

5.2 Experiment 1: Measuring the Combined Impact of Smartphone Quantities and Data Collection Policy Decisions on System Accuracy

Sections 3.1 and 3.2 discuss the challenges that arise due to unpredictable smartphone counts and system policy decisions. In particular, it is tough for a system developer to determine the minimum number of smartphones required to ensure system goals are met. Additionally, it is tough to determine the effect of system policies due to the complex nature of the system properties and the potential for emergent properties. In order to measure the effect of system policy decisions, while also considering system properties, we simulate multiple system policies in conjunction with multiple combinations of system properties of environment topology and smartphone quantities. The simulated policy for each iteration is one of two possible data collection policies, where the first policy is a constant data collection policy in which smartphones are taking readings as rapidly as possible, and the second is a context-aware policy where more mobile smartphones collect data more frequently.

Hypothesis: A Constant Data Collection Policy Will Result in Minimum 20% Higher Accuracy For all System Property Permutations. We expect that the constant data collection policy, given the much higher number of inputs, would yield significantly higher accuracy. Additionally, we believe our results will allow a clear determination of the minimum number of smartphones required to a system accuracy of 80%.

Experiment 1 Results. Figure 5 shows, for a static environment and various smartphone counts, the accuracy over time for each of the two chosen data collection policies. Figure 6 shows, for a dynamic environment and various smartphone counts, the accuracy over time achieved by the two data collection policies.

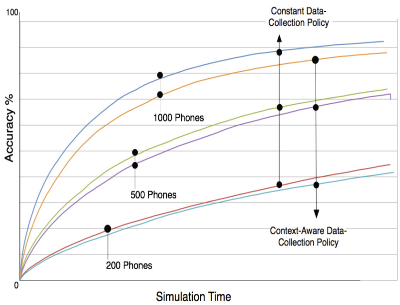

Fig. 5. Measuring the effect of smartphone count on the accuracy in static environments

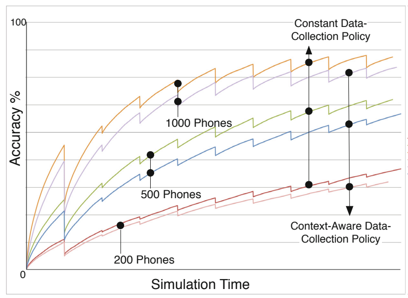

Fig. 6. Measuring the effect of smartphone count on the accuracy in a dynamic environment

Figure 6 shows multiple dips in accuracy, which we identified as being the times at which we changed the real environment, and therefore the current perceived environment is incorrect. The magnitude of these dips noticeably decreases over time, which is discussed in Section 5.4. As expected, the constant data collection policy does result in a small, but notable, increase in accuracy for both types of environments. However, the acceptable performance of the context-aware data collection policy was unexpected. The context-aware system policy did not result in a significant loss in accuracy, indicating that our hypothesis was incorrect. However, it is easily discernible, for each type of represented environment, the minimum number of smartphones required in order to ensure that 80% accuracy is achieved. For both of the environments tested, independently of the type of data collection policy, 80% accuracy cannot be achieved within 1460 simulation hours without a minimum of 900-1000 smartphones.

5.3 Experiment 2: Determining the Wastefulness of Various Data Collection Policies

Given that Experiment 1 has shown that a context-aware data collection policy is acceptable under some circumstances, the next logical experiment was to attempt to determine, for a given circumstance, if a context-aware policy was acceptable. Experiment 1 has already shown data on the slight increase in accuracy that a constant data collection policy results in, and therefore we choose to measure the wastefulness of each of the policies. In order to do this, we counted the number of data readings that resulted in absolutely no change in the accuracy of the system, and accumulated those numbers over time.

Hypothesis: Constant Data Collection Is Accurate But Wasteful. Our hypothesis for this experiment is that a constant data collection policy e.g. having each smartphone constantly collect data would be accurate but also notably wasteful of system resources.

Experiment 2 Results. Figure 7 therefore represents the difference between the constant data collection policy and the context-aware data collection policy. The six series result from the three options for smartphone counts (200, 500, 1000) combined with the two options for environment topology (static, dynamic).

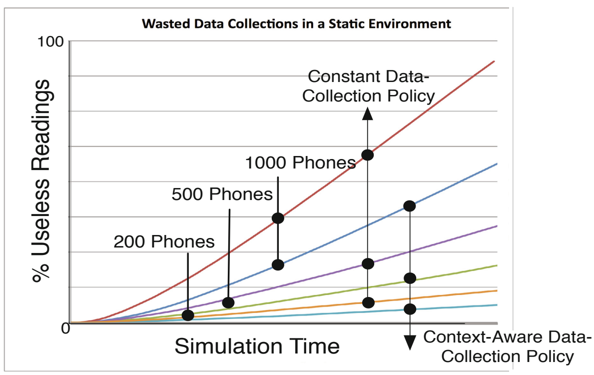

Fig. 7. The number of wasted readings for constant versus mobility-based data collection

As expected, the constant data collection policy resulted in higher number of wasted readings, with more smartphones resulting in a larger difference.

5.4 Analysis of Results

The experimental data provided both expected and unexpected results. For Experiment 1, we were able to determine the minimum number of smartphones required to ensure that system goals were met for each operational environment we tested. However, our experimental hypothesis was ultimately proved incorrect, as the context-aware data collection policy resulted in much higher accuracy than was expected. In Experiment 2, we were able to show relative wastefulness of a constant data collection policy versus a context aware data collection policy.

One interesting result was the unexpected significance of our metric implementations. For example, transitioning from having no knowledge to having some (potentially incorrect) knowledge makes it difficult to approximate the accuracy of the overall simulation. In our experiments, we chose to count regions where we knew no information as the absolute worst possible values (e.g. the values farthest from the real network). This in turn had significant implications - the effect of an unknown region on overall accuracy was very bad, and therefore coverage had a significant relation to accuracy. Essentially, all values that were input regarding a region, no matter how incorrect, were likely to be better than the worst possible value. This effect was determined to be responsible for the reduction in magnitude of the dips shown in Figure 6 as the simulation progressed.

6 Related Work

This section compares Empower with related research from two key areas. First, significant research has been performed in the area of mobile sensor networks. Second, prior research in opportunistic sensing discusses some of the issues present with this paradigm.

Mobile Wireless Sensor Networks. While smartphone sensor networks have many differences from conventional mobile wireless sensor networks, such as a reduced need for ad hoc networking, some similarities do exist. Much mobile wireless sensor network research focuses on methods for networking and power reduction in mobile ad hoc networks. For example, Jain et al. show that by exploiting mobile nodes as intermediate data carriers it is possible to significantly reduce the amount of power required to transmit information between remote sensors and base stations [11]. However, smartphone sensor network nodes can be assumed to have fairly regular direct connection to the Internet, and are regularly recharged.

Mainwaring et al. explores building a wireless sensor network to perform habitat monitoring, because "the connection with its immediate physical environment allows each sensor to provide localized measurements, [...] integration of local processing and storage allows nodes to perform complex filtering, [...] and ability to communicate allows information to be retrieved and nodes to be retasked in the field" [12]. However, methods of system-level verification and validation are not addressed, and there are no guarantees that such as system can meet its described goals. Most verification and validation research for wireless sensor networks has been done in the context of individual nodes, and little work has been done in full-system verification and validation.

Opportunistic Sensing. Research in opportunistic sensing has been increasing in popularity as sensors rise in availability. However, much of this research has been preliminary, and has not had a focus on the overall success of the opportunistic network. Lilien et al. describe a framework standard, intended to allow components to be added and removed from any opportunistic network during the system production [13]. This may be a promising future approach, but in our research we have simply allowed mobile applications and previously-established standards, such as Hyper text transport protocol, to manage the interactions between opportunistic sensing components. Eisenman et al. discuss techniques for implementing intelligent network in a network largely consisting of mobile nodes [14]. Our research focuses on smartphone systems, which typically have an available network connection. However, in disaster scenarios this ad hoc networking would become invaluable, and therefore Empower allows the network components of the simulation to be handled by third party network driver software. This allows other researchers to integrate the work done with Empower and work being performed in the area of ad hoc networks.

7 Concluding Remarks and Lessons Learned

In this paper we discussed the emerging importance of smartphone-based data collection systems. We described the current problems with building these systems, which largely arise because the properties of these systems are highly emergent. This makes verification and validation of these systems at design time quite difficult.

In order to address this issue, we developed a simulation environment titled Empower that allows smartphone based data collection systems to be simulated. This simulation allows smartphone-specific system properties, such as data collection policies and smartphone count, to be specified, and provides system-level metrics that can be used to identify if the system achieved the goals of interest. By simulating the system with different properties, system designers can have informed opinions on the effectiveness of their system during the production phase. From our research on simulating smartphone sensor network data collection systems, we learned the following important lessons:

1. **Smartphone-based Data Collection Systems have Very High Potential.** Through our research, we were able to realize that smartphone-based data collection systems, when properly focused and directed, can have an enormous impact with a very small user base. While we have only performed initial research on quantifying these values, further work in this area is likely to be very promising.
2. **Calculation of Metrics for Data Collection Systems is Hard.** In our research the environments were divided into discrete regions, enabling us to perform system metrics much more simply than if we had truly continuous environments. This had significant impacts on many parts of the system, such as inputting data readings, speed of execution, and accuracy of calculations. In future work we plan to investigate continuous environments in order to address this gap.

Empower and data from the experiments described in this paper are available in opensource form from `https://github.com/crabpot8/Empower`.

References

1. Gartner says worldwide mobile phone sales grew 35 percent in third quarter 2010; smartphone sales increased 96 percent (November 2010), `http://www.gartner.com/it/page.jsp?id=1466313`

2. Turner, H., White, J., Thompson, C., Zienkiewicz, K., Campbell, S., Schmidt, D.: Building Mobile Sensor Networks Using Smartphones and Web Services: Ramifications and Development Challenges. In: Handbook of Research on Mobility and Computing, Hershey, PA (2009), http://www.cs.wustl.edu/~schmidt/PDF/hamilton-book-chapter.pdf
3. Froehlich, J., Dillahunt, T., Klasnja, P., Mankoff, J., Consolvo, S., Harrison, B., Landay, J.: UbiGreen: investigating a mobile tool for tracking and supporting green transportation habits. In: Proceedings of the 27th International Conference on Human Factors in Computing Systems, pp. 1043–1052. ACM (2009)
4. Thompson, C., White, J., Dougherty, B., Schmidt, D.: Optimizing Mobile Application Performance with Model-Driven Engineering. In: Proceedings of the 7th IFIP Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (2009)
5. Jones, W.: Forecasting traffic flow. IEEE Spectrum 38(1), 90–91 (2001)
6. Rose, G.: Mobile phones as traffic probes: practices, prospects and issues. Transport Reviews 26(3), 275–291 (2006)
7. Leijdekkers, P., Gay, V.: Personal heart monitoring and rehabilitation system using smart phones. In: Proceedings of the International Conference on Mobile Business, p. 29. Citeseer (2006)
8. Gahran, A.: Reporting on the gulf oil spill from your cell phone, 2010, this is an electronic document, June 11 (2010), http://articles.cnn.com/2010-06-11/tech/oil.spil.app_1_cell-phones-apps-geotagged?_s=PM:TECH (date retrieved January 12, 2010) (date last modified: date unavailable)
9. Ölveczky, P.C., Thorvaldsen, S.: Formal Modeling and Analysis of the OGDC Wireless Sensor Network Algorithm in Real-Time Maude. In: Bonsangue, M.M., Johnsen, E.B. (eds.) FMOODS 2007. LNCS, vol. 4468, pp. 122–140. Springer, Heidelberg (2007)
10. Sharma, O., Lewis, J., Miller, A., Dearle, A., Balasubramaniam, D., Morrison, R., Sventek, J.: Towards Verifying Correctness of Wireless Sensor Network Applications Using Insense and Spin. In: Păsăreanu, C.S. (ed.) Model Checking Software. LNCS, vol. 5578, pp. 223–240. Springer, Heidelberg (2009)
11. Jain, S., Shah, R., Brunette, W., Borriello, G., Roy, S.: Exploiting mobility for energy efficient data collection in wireless sensor networks. Mobile Networks and Applications 11(3), 327–339 (2006)
12. Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., Anderson, J.: Wireless sensor networks for habitat monitoring. In: Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, pp. 88–97. ACM (2002)
13. Lilien, L., Gupta, A., Yang, Z.: Opportunistic networks for emergency applications and their standard implementation framework. In: IEEE International Performance, Computing, and Communications Conference, IPCCC 2007, pp. 588–593. IEEE (2007)
14. Eisenman, S.B., Lane, N.D., Campbell, A.T.: Techniques for Improving Opportunistic Sensor Networking Performance. In: Nikoletseas, S.E., Chlebus, B.S., Johnson, D.B., Krishnamachari, B. (eds.) DCOSS 2008. LNCS, vol. 5067, pp. 157–175. Springer, Heidelberg (2008)