PRO 04 Smart Power Management on Cellphones

PRO 04.1 Overview
Modern mobile phones are changing from single-purpose devices to multi-functional programmable computers. As a result, a plethora of mobile applications are emerging, many of which run in the background and collect context information to process locally or upload to a server. Most of the participatory sensing applications developed at CENS are of this type. With the presence of such applications mobile phones cannot rely on long low-power idle states to conserve energy. Therefore, such background applications negatively affect phone battery life and thus users' satisfaction of their phones. Our experiences with several such applications, including Campaignr, the Nokia Simple Context, PEIR, and Andwellness, indicate that they reduce the phone battery life to less than 12 hours, and therefore, many users stop running them. This is a major obstacle on wide deployment of Participatory Sensing mobile applications. The objective of the Smart Power Management project is to solve this problem and give users of such applications "reasonable" battery life.

Several efforts have focused on reducing power consumption of individual phone components. However, there is evidence that significant power saving potential lies at the application layer. This potential can be realized by exploiting the intrinsic trade-off between applications' quality of output and their energy consumption. In general, decreasing fidelity leads to lower energy consumption. For example, increasing the GPS sampling interval from 15 seconds to 30 seconds reduces the GPS energy consumption by 50%. The effect on quality of service, however, is application specific.

We argue that participatory sensing applications on smartphones should be able to adapt their operation rate or fidelity based on users' battery life expectations. Users cannot be expected to manage how applications run, or to keep track of the battery. Therefore, the power management system should monitor the system and plan ahead to meet the user's battery life expectation. Such system capabilities would rely on models of: battery life, user's charging behavior, energy consumed by "legacy applications" (e.g., calls), and the energy-performance trade-off of each adaptive application.

PRO 04.2 Approach
We pursue an experimental approach. Our first step in this approach is to understand how users use their mobile phones. Therefore, we need real usage traces from smartphone users to visualize and model. We use statistical modeling and learning techniques in the model building step of our research.

Along with our modeling efforts, we build systems that run on mobile phones to implement our proposed algorithms and policies. This gives us solid evaluation of the performance of different power management strategies and algorithms.

PRO 04.3 System(s) Description and/or Experiments
We initially started collecting system information on Nokia phones from a few volunteer smartphone users using a tool developed by Nokia Research Labs, named the Nokia Simple Context. We had a pilot study consisting of six volunteers with this tool. This pilot helped us understand what system level and usage events are most interesting and relevant to our study. We used the results of this study in a technical report on reducing the energy consumption of the mobile phone screen.

We took our experimental study to a next level by developing SystemSense. SystemSense is a logging software tool that was developed at CENS to log detailed system related information on Android smartphones. SystemSense keeps the logged records in a local database on the phone, and regularly uploads them to SensorBase. SystemSense has been designed to be as energy efficient as possible. We have made sure that the logging operation does not consume excessive CPU and network resources. SystemSense accesses low-level Android APIs that are only accessible to the operating system. Therefore, we can only run SystemSense on a developer phone. Following is the list of information that each version of SystemSense collects. We continue improving this tool.

Version 1.0:

- Detailed battery information:
  - Battery level
  - Battery voltage
  - Battery temperature
  - Battery health
• Charging status
• Screen status changes (on and off)
• Application usage times
• Network traffic per application
• CPU time used by each application

Version 1.1:
• GPS usage statistics per application
• WiFi interface status changes
• Call durations

Version 1.2:
• Accelerometer usage statistics per application

To minimize usage and battery perturbation SystemSense uploads the collected records only when the phone is plugged to the charger. To increase sampling accuracy SystemSense increases sampling frequency when the screen turns on (i.e., the user is interacting with the phone). During all other times, SystemSense samples resource usage counters at a fairly low frequency.

We also developed a logging tool for applications named SystemLog. SystemLog is an Android service developed at CENS to facilitate collecting performance related logs during deployments of CENS mobile applications. The SystemLog client runs as an Android Service on the phone. It defines a simple interface using the Android Interface Definition Language (AIDL). All other applications can send their log messages to SystemLog. SystemLog will augment log messages with information such as date and time and name of the logger application. The log records are kept in a local database on the phone. When SystemLog detects the phone is plugged to external power, it uploads the logged records to SensorBase.

PRO 04.4 Accomplishments
We deployed SystemSense during the summer of 2009 and collected a valuable data set. These traces are from 33 Android users. These users consisted of 16 knowledge workers and 17 high school students. All the participants worked at CENS during the summer. They were recruited to participate in our study by a staff member at CENS. As stated in our study consent form, the users’ identity was never revealed to the researchers involved in this project. We are using these anonymized traces to find appropriate models of users interaction with their phones and also test the performance of several power management strategies and algorithms in trace-based simulations. However, we will not stop the evaluation at this stage. In the next section we explain how we will further evaluate our algorithms.

Several aspects of smartphone usage have been successfully modeled. We have characterized both intentional user activities—i.e., interactions with the device and the applications used—and the impact of those activities on network and battery resources. We uncovered a surprising level of diversity among users. Along all aspects that we studied, users differ by one or more orders of magnitude. For instance, the average number of interactions per day varies from 10 to 200, and the average amount of data received per day varies from 1 to 1000 MB (Figure 1). This level of diversity implies that any mechanism designed to improve user experience or energy consumption will likely be much more effective if it learns and adapts to the user behavior. Our paper that presents these results was accepted to the 8th Annual International Conference on Mobile Systems, Applications and Services.
**PRO 04.5 Future Directions**

With our current understanding of smartphone usage, we are ready to start implementing the smart energy management system on the Android platform. We have finalized the design of the software and found the first project within CENS that is going to use our power management.

We are working with the members of the Andwellness project to incorporate energy adaptation techniques and mechanisms into their mobile application. We have a specific battery lifetime goal for Andwellness that is 20 hours of uninterrupted operation.