Online Estimation of the Remaining Energy Capacity in Mobile Systems Considering System-Wide Power Consumption and Battery Characteristics *

Donghwa Shin and Naehyuck Chang †
Seoul National University
{dhshin, naehyuck}@elpl.snu.ac.kr

Woojoo Lee, Yanzhi Wang,
Qing Xie, and Massoud Pedram
University of Southern California
{woojoole, yanzhiwa, xqing, pedram}@usc.edu

ABSTRACT

Emerging mobile systems integrate a lot of functionality into a small form factor with a small energy source in the form of rechargeable battery. This situation necessitates accurate estimation of the remaining energy in the battery such that user applications can be judicious on how they consume this scarce and precious resource. This paper thus focuses on estimating the remaining battery energy in Android OS-based mobile systems. This paper proposes to instrument the Android kernel in order to collect and report accurate subsystem activity values based on real-time profiling of the running applications. The activity information along with offline-constructed, regression-based power macro models for major subsystems in the smartphone yield the power dissipation estimate for the whole system. Next, while accounting for the rate-capacity effect in batteries, the total power dissipation data is translated into the battery's energy depletion rate, and subsequently, used to compute the battery's remaining lifetime based on its current state of charge information. Finally, this paper describes a novel application design framework, which considers the batterys state-of-charge (SOC), batterys energy depletion rate, and service quality of the target application. The benefits of the design framework are illustrated by examining an archetypical case, involving the design space exploration and optimization of a GPS-based application in an Android OS.

I. INTRODUCTION

Evolution of mobile systems including smartphones and tablet-PCs has given rise to larger and more power hungry embedded systems with advanced functionality and high performance. Unfortunately, the increase in volumetric/gravimetric energy density of (rechargeable) batteries has been much slower than the increase in the power demand of these devices, creating a "power crisis" for the smartphone technology development and product line expansion.

Accurate power modeling and estimation is a key requirement of any power-aware design methodology and low power design tool. We need system-level power information in order to develop power-aware applications because modern smartphone applications utilize many system resources including the application processor, display, audio, wireless communication, and so on, simultaneously. There have been several efforts to provide system-level power estimation models for the smartphone platforms [1] including battery information-based models [2, 3]. These models are based on the component activities within the system.

The state of charge (SOC) of the battery ultimately determines the service life (also called lifetime) of a smartphone. At the same time, the service quality of an application that is running on a smartphone is largely dependent on the system lifetime since the application-provided services becomes useless if the smartphone has run out of battery energy. So it is important for a smartphone owner/user to have accurate information about the remaining amount of energy in the battery. All this means that the smartphone users are less interested in how much power is consumed in the smartphone and more interested in how much energy is left in the battery that powers up the smartphone.

From the above discussion, it is evident that one must develop a battery capacity loss (depletion) rate estimator and not simply a power dissipation estimator for mobile systems. The former captures important effects related to the battery chemistry, as well as conversion and distribution losses. The conversion from power dissipation to battery capacity loss rate is not a constant factor, rather it is a highly nonlinear function depending on SOC of the battery, the load power dissipation and the rate capacity effect, and the ambient temperature to name a few. Developing such an estimator is the precise focus of our paper. The proposed method instruments the Android kernel to produce accurate subsystem's activity as the key factor of the proposed regression-based power consumption model based on real-time profiling. The power consumption data along with information about the battery SOC is used to estimate the remaining the battery lifetime.

This paper also introduces an energy-aware application design framework for Android OS-based mobile systems with the system-level power model, battery status model, and service quality model. We explore the design of a GPS application as a case study. We maximize the service quality of the GPS application by considering the locating resolution and trip coverage related to the battery lifetime in the case study.

*This work is sponsored by a grant from the National Science Foundation, the Brain Korea 21 Project, and the National Research Foundation of Korea (NRF) grant funded by the Korean Government (MEST) (No. 2011-0016480 and 2011-0030512). The ICT at Seoul National University provides research facilities for this study.

†Corresponding author

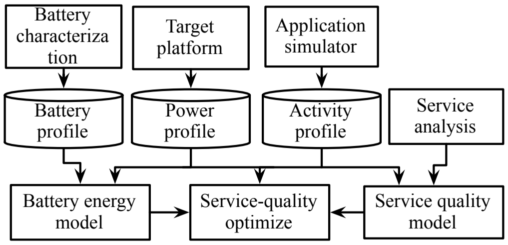


Fig. 1. Energy-aware application design framework.

II. Related Work

There are numerous studies on power analysis and modeling of the computing systems including both general purpose and mobile embedded systems. Several parameterized power models for the mobile computing systems have been introduced. A measurement-based power estimation model is introduced in [1]. The authors collect system activity parameters and evaluate the system power consumption. Various coefficients of the system power equation are derived by regression analysis.

The authors of [3] start from the observation that dependencies of system energy models on the specific hardware architecture and configuration as well as the usage patterns suggest personalized models are needed for a mobile system relying on the battery interface. Reference [4] relies on on-chip bus performance monitoring unit to produce accurate estimates of system power consumption from a first-order linear power model by utilizing system-level activity information exchanged on the system bus. Reference [5] presents a runtime, feedback-based full system energy estimation model for battery powered devices. The authors rely on first-order, linear regression equations (i.e., the power model) that capture the energy consumption of CPU and memory using feedback about program execution behavior by monitoring various system events.

The authors of [2] provide manually generated power models for HTC Dream and HTC Magic phones. This technique uses built-in battery voltage sensors and knowledge of battery discharge behavior to monitor power consumption of individual components and their impact on the state of discharge (SoD) of the battery. A software implementation of this estimator, called PowerTutor, has been released on the Android market. This reference is the closest prior work to what we report in this paper. However, as show later, PowerTutor tends to overestimate the remaining energy capacity of the smartphone battery because it ignores the effect of the current discharge rate on the battery capacity loss (known as the rate capacity effect), and the internal battery losses.

Battery models for the electronic systems have extensively been studied during the past few decades. We can find many analytical models based on electrochemical modeling and analysis [6, 7], but the electrochemical battery models are too complicated to be used for the system-level design of electronics. Battery models in the form of an electric circuit are suitable for this purpose [8, 9].

GPS is widely used for the tracking and navigation applications nowadays. GPS devices consumes non-negligible amount of power. Some low-power techniques for the GPS applications have been introduced. There are several approaches to enhance the locating accuracy of the GPS system while considering the power consumption. A combination of an accelerometer and GPS receiver was introduced in [10]. Another method adjusts performance or functionality to prolong battery lifetime. They use custom IC in addition to GPS front-end chips [11]. A tradeoff between the locating resolution and power consumption is considered to design the sensor node application in [12, 13].

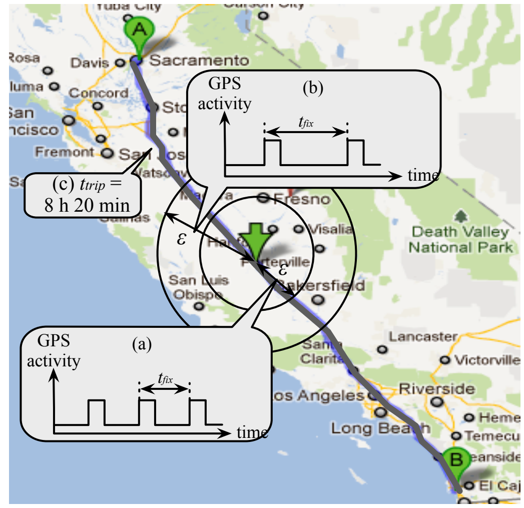


Fig. 2. GPS service quality model.

III. Energy-Aware Application Design

A. Design framework

The purpose of the proposed application design framework is to estimate the power consumption and corresponding battery SOC at the application design time. We use three kinds of profiles to achieve this goal: system-level power dissipation, subsystem (component) activity, and battery current/voltage as shown in Fig. 1. We synchronously collect the system activity parameters and measure components power consumptions. Next, we develop a smartphone power dissipation model by doing regression analysis on the measured data. The battery is also pre-characterized under constant-current and pulsed-current charging and discharging scenarios. The characterization data include rate capacity curve and internal resistance value of the battery. The battery characterization data is subsequently used to estimate the internal battery losses and battery SOC change as a result of the total power dissipation in the smartphone.

We need a way to model the service quality of the application so as to maximize the satisfaction of the smartphone user. In this paper, we focus on modeling the service quality in relation to overall power/energy efficiency. The service quality of a battery-powered computing system (e.g., a smartphone) is approximately modeled by a combination of the performance and service time. In general, a higher performance provides a higher service quality, but it also results in larger power consumption and a shorter service time. Therefore, we should design the application, balancing performance and service time with the objective of maximizing the service quality.

B. Case Study: Energy-Aware Service Quality Optimization of GPS Application

The service quality of the GPS tracking application is determined by not only the *locating accuracy* but also the *lifetime* of the system. For an application where remote long term tracking is needed (long distance road trip, emergency situation in remote area, and so on), a longer service time is more important. These applications generally require less locating accuracy between the sample points as they are tracking over long distances and extended time periods. The accuracy is also affected by the speed of the tracking object. A low speed means that the object in question moves only a small distance over a given time period. This is contradictory to applications where high positioning precision is critical due to higher velocities (such as aeronautical navigation). We will evaluate the service quality of the application considering the service time as well as the locating accuracy related to the sampling period and speed of the tracking object.

The locating accuracy is inversely proportional to the locating error. The maximum locating error of GPS, ε, is proportional to the velocity of the tracking object and GPS fixing period. Which is given by

$$\varepsilon = v \cdot t_{fix} + \varepsilon_0 \quad (1)$$

where v, t_{fix}, and ε_0 denote velocity of tracing object, fixing period, and intrinsic error from GPS, respectively. If we use longer t_{fix} as shown in Fig. 2 (a) compared to (b), then we have larger error by (1). We use a normalized error, $\bar{\varepsilon} = \varepsilon / v$, to evaluate the service quality.

The tracking application should be running for a required time up to its conclusion. If the battery lifetime is equal or longer than the required time for a given trip, then the service quality is the maximum value (=1). Otherwise, the service quality is determined as being proportional to the covered trip time. The service quality related to the trip coverage, *Coverage*, is proportional to the normalized service time, which is given by

$$Coverage = \begin{cases} t_{service}/t_{trip} & (t_{service} \le t_{trip}), \\ 1 & (\text{otherwise}), \end{cases} \quad (2)$$

where $t_{service}$ and t_{trip} denote the service time and, respectively.

IV. Power Estimation Model

We use an Odroid-A platform from Hardkernel [14] as a target platform. It is a high-end development platform for the smartphone and tablet PC which has very similar features to Samsung Galaxy S2. The specification of the Odroid-A platform is summarized in Table I. The system-level power model includes the CPU, 3G, wi-fi, display, audio, GPS, and vibration motor. The parameters for the system components is collected from Android OS by the activity profiler.

A. CPU

We characterize the CPU power consumption as a function of the operating frequency and its utilization from the Android OS, which is given by

$$P_{cpu} = C^{cpu}_{freq} X^{cpu}_{freq} (X^{cpu}_{kernel} + X^{cpu}_{user}) + P^{cpu}_0, \quad (3)$$

where X^{cpu}_{freq}, X^{cpu}_{kernel}, X^{cpu}_{user}, C^{cpu}_{freq}, and P^{cpu}_0 denote the operating frequency (MHz), kernel program utilization, user program utilization, regression coefficient and offset value, respectively.

TABLE I
Target platform specification

Components	Model	Descrpition
Application processor	Exynos4210	Dual-core CPU
3G module	F5521GW	G+GPS module
Wi-fi	GB8632	Wi-fi+bluetooth module
Display	LP101WH1	1366 x 768 TFT LCD
Audio codec	MAX98089	full-featured codec
Vibration motor	DMJBRK36S	Vibration motor
Battery	KPL6072196	Li-poly, 10Ah

B. 3G module

The 3G module consumes power approximately according to its operating state. It has three operating state which are IDLE, FACH, and DCH. We model the power consumption of the module according to the state of the module as follows:

$$P_{3G} = X^{3G}_{on} (C^{3G}_{idle} X^{3G}_{idle} + C^{3G}_{fach} X^{3G}_{fach} + C^{3G}_{dch} X^{3G}_{dch}), \quad (4)$$

where X^{3G}_{on} is 1 when the 3G module is activated or 0 otherwise. X^{3G}_{idle}, X^{3G}_{fach}, and X^{3G}_{dch} denote the timing portion of each state for a given sampling period, respectively. C^{3G}_{idle}, C^{3G}_{fach}, and C^{3G}_{dch} are the regression coefficients.

C. Wi-fi module

The Wi-fi module also consumes power approximately according to its operating state. It has two operating states, which are LOW and HIGH. Each operating state represents the link speed of the wireless channel. The model is given by:

$$P_{wf} = X^{wf}_{on} (C^{wf}_{low} X^{wf}_{low} + C^{wf}_{high} X^{wf}_{high}), \quad (5)$$

where X^{wf}_{on} is 1 when the Wi-fi module is activated or 0 otherwise. X^{wf}_{low} and X^{wf}_{high}, denote the timing portion of LOW and HIGH state for a given sampling period, respectively. C^{wf}_{low} and C^{wf}_{high} are the regression coefficients.

D. Display

Power consumption of the LCD display in the Odroid platform shows discrete characteristics. We can control the brightness value from 0 to 255 by setting the value of a control register. In general, the LCD power consumption is proportional to its brightness This is also true for the Odroid platform as long as the brightness value is not set to the maximum allowed. However, if we set the brightness value to the maximum, the display consumes much more power than before. , which is given by

$$P_{lcd} = C^{lcd}_{brit} X^{lcd}_{brit} X^{lcd}_{on} + C^{lcd}_{full} X^{lcd}_{full} X^{lcd}_{on} + P^{lcd}_0 + P^{lcd}_{ctrl}, \quad (6)$$

where X^{lcd}_{on} is 1 when LCD is turned on and X^{lcd}_{full} is 1 when the brightness is the maximum, or, otherwise, they are 0. X^{lcd}_{brit}, P^{lcd}_0, and P^{lcd}_{ctrl}denote the brightness value, power offset value, and controller power consumption, respectively. C^{lcd}_{brit} and C^{lcd}_{full} are the regression coefficients.

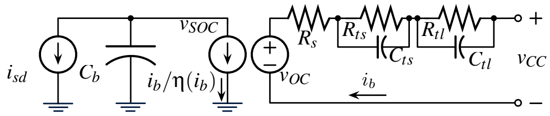


Fig. 3. Li-ion battery equivalent circuit model.

E. Audio device

The average power consumptions of the audio amplifier and speaker is proportional to the volume., which is given by

$$P_{aud} = X_{on}^{aud}(C_{vol}^{aud}X_{vol}^{aud} + C_{on}^{aud}), \tag{7}$$

where X_{on}^{aud} is 1 when the audio module is activated or 0 otherwise. X_{vol}^{aud} and C_{vol}^{aud} denote the volume value and regression coefficient, respectively.

F. GPS module

The power consumption of the GPS module is determined by its operational state. The GPS module has three state which are OFF, SLEEP, ACTIVE state. Each state consumes different amount of power, which is given by

$$P_{gps} = X_{on}^{gps}(C_{off}^{gps}X_{off}^{gps} + C_{sleep}^{gps}X_{sleep}^{gps} + C_{active}^{gps}X_{active}^{gps}), \tag{8}$$

where X_{on}^{gps} is 1 when the GPS module is activated or 0 otherwise. X_{off}^{gps}, X_{sleep}^{gps}, and X_{active}^{gps} denote the timing portion of each state for a given sampling period, respectively. C_{off}^{gps}, C_{sleep}^{gps}, and C_{active}^{gps} are the regression coefficients.

G. Vibration motor

The vibration motor consumes 150 mW to 600 mW on average depending on the size when it is turned on. The power model is given by

$$P_{vm} = C_{on}^{motor}X_{on}^{motor}, \tag{9}$$

where X_{on}^{motor} is 1 when the motor is turned on or 0 otherwise. C_{on}^{motor} is the regression coefficient.

V. Battery Life Model

We need an appropriate battery model to estimate the battery internal state where the estimation of the battery internal loss is important for system lifetime estimation as well as the estimation of the component power consumption. According to our characterization, the internal resistance of the Li-ion battery is about 100 mΩ which dissipates non-negligible amount of power. In addition, the part of stored energy in the battery cannot be used up to its discharging condition by rate-capacity effect. In this section, we construct an equivalent circuit model for the Li-ion battery considering the internal loss to avoid the overestimation of battery lifetime which can be resulted in only with the components power estimation.

A. Battery circuit model

We import an equivalent circuit model of the Li-ion battery from [9] as shown in Fig. 3. This includes a runtime-based model as well as a circuit-based model for accurate capturing of the battery service life and I-V characteristic. In this model we use C_b to denote the remaining charge in the battery, and v_{SOC} as the voltaic representation of the battery SOC, defined as the ratio of the charge currently stored in the battery to the total charge when the battery is fully charged, i.e.,

TABLE II
Extracted parameters for the battery models.

Coeff.	Value	Coeff.	Value	Coeff.	Value
b_{11}	-0.265	b_{12}	-61.649	b_{13}	-2.039
b_{14}	5.276	b_{15}	-4.173	b_{16}	1.654
b_{17}	3.356	b_{21}	-0.043	b_{22}	-14.275
b_{23}	0.154	k_d	0.019		

$$v_{SOC} = C_b/C_{b,full} \times 1\ \text{V}, \tag{10}$$

where $C_{b,full}$ is the total charge of battery (in Coulomb) when it is fully charged, given by

$$C_{b,full} = 3600 \times Capacity \tag{11}$$

where $Capacity$ is the nominal battery capacity in Ahr. In the battery model, the open circuit terminal voltage (OCV) of the battery, denoted by v_{OC} in Fig. 3, as well as other internal resistances and capacitances, are all functions of the SOC value v_{SOC}. Since the charging/discharging current of the battery in the mobile platform is relatively small (typically less than $C/10$), the effects of the battery internal capacitances are not phenomenal. Therefore we only provides model of the OCV v_{OC} and total internal resistance R_{total} as a function of the SOC value v_{SOC} in the following nonlinear equations, in which R_{total} is the sum of the battery internal resistances R_s, R_{ts} and R_{tl} shown in Figure 3.

$$\begin{aligned} v_{OC} =& b_{11}e^{b_{12}v_{SOC}} + b_{13}{v_{SOC}}^4 + b_{14}{v_{SOC}}^3 + \\ & b_{15}{v_{SOC}}^2 + b_{16}v_{SOC} + b_{17}, \\ R_{total} =& b_{21}e^{b_{22}v_{SOC}} + b_{23}, \end{aligned} \tag{12}$$

where b_{ij} are empirically-extracted regression coefficients.

The rate capacity effect of batteries specifies the fact that the available discharging time of a battery is strongly dependent on the battery discharging current i_b. Equivalently, we use the battery discharging efficiency, $\eta(i_b)$, defined as the ratio between the battery discharging current and the charge loss rate inside the battery, to incorporate the rate capacity effect in the battery runtime model shown in Fig. 3. According to the Peukert's Law [15], the battery discharging efficiency $\eta(i_b)$ can be approximated by $1/((i_b)^{k_d})$ We extract the equation parameters from KPL6072196 Li-ion cell in the Odroid platform by measurements. as presented in Table II.

B. Battery state of charge estimation

We develop a battery state estimation model based on the power estimation model and battery model introduced in the Sections IV and V. The battery SOC, E, is calculated by

$$E(t+\Delta t) = E(t) - \Delta E, \tag{13}$$

where the energy difference, ΔE, is given by

$$\Delta E = \Delta t(P_{cpu} + P_{lcd} + P_{audio} + P_{3G} + P_{wf} + P_{vm}) + E_{loss}, \tag{14}$$

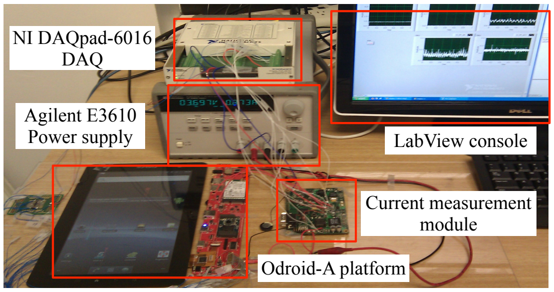


Fig. 4. Experimental setup.

where Δt, T, and N denotes the discrete time period for energy estimation, temperature, and number of cycles. The internal loss of battery for Δt is derived from (12), which is given by

$$E_{loss} = \Delta t(i_b^2 R_{total} + i_b \cdot v_{OC} \cdot (1/\eta(i_b) - 1)). \qquad (15)$$

Finally, we translate the power estimation in Section IV result based on the activity profile into the battery energy state profile by using (13), (14), and (15).

VI. Experiment

A. Model verification

We measure the power consumption of major components in the Odroid-A platform with the experimental setup shown in Fig. 4, and construct the power estimation model as summarized in Table III. We implement a current measurement module, which consists of shunt resistors and the INA194 shunt monitors form Texas instruments. We use a DAQpad-6016 data acquisition system with a LabView from National instruments to collect the measured data. An E3610 Power supply and A34410A digital multimeter from Agilent are used to supply power and measured data verification.

We compare the remaining battery energy estimates of the proposed model with those of the model introduced in [2]. The authors of that paper argue that the error resulting from assuming that the battery energy capacity is independent of discharge rate is quite small, and hence, they go on to use a precharacterized battery characteristic curve under low constant-current discharge to estimate the remaining battery energy and then, from that derive the regression coefficients of their system-level power model. We, on the other hand, calculate the remaining battery energy from the system-level power consumption modulated by information about the current SOC of the battery itself. According to our observations, *the effect of discharge rate on the translation of the total system power dissipation to the depletion rate of the battery energy capacity is sizeable.*

TABLE III
Battery status estimation model based on Android system activity parameters

Components	Coeff.	Value	Coeff.	Value	Coeff.	Value
Processor	C^{cpu}_{freq}	0.00642	P^{cpu}_0	0.332		
3G	C^{3G}_{idle}	0.011	C^{3G}_{dch}	0.672	C^{3G}_{fach}	0.322
Wi-fi	C^{wf}_{high}	0.020	C^{wf}_{low}	0.740		
Display	C^{lcd}_{birt}	0.004	P^{lcd}_0	0.224	C^{lcd}_{full}	1.307
	P^{lcd}_{ctrl}	0.067				
Audio	C^{aud}_{on}	0.024	C^{aud}_{vol}	0.00009		
Vibrator	C^{vib}_{on}	0.003				
GPS	C^{GPS}_{off}	0.011	C^{GPS}_{active}	0.212	C^{GPS}_{sleep}	0.069

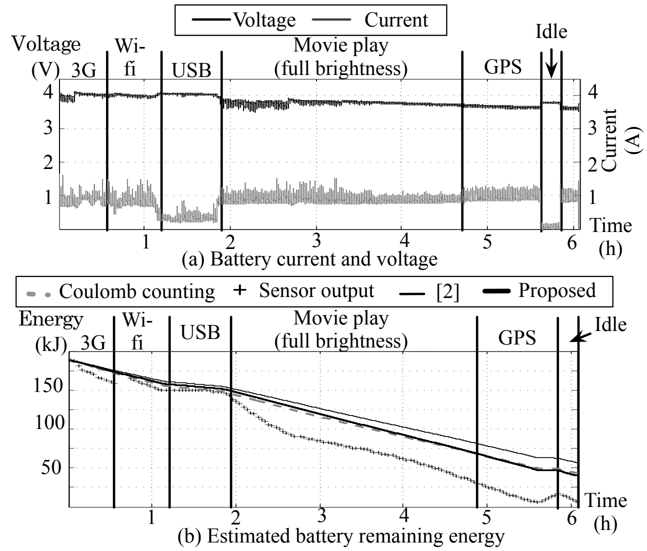
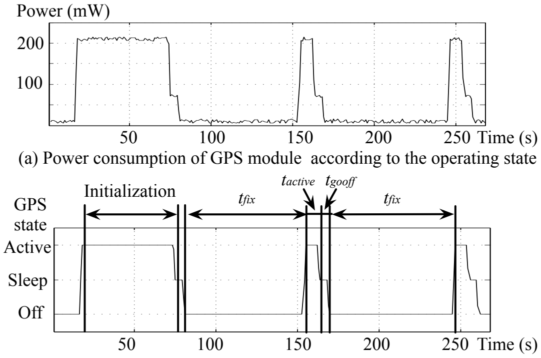


Fig. 5. Battery measurement result (a) and energy model verification (b).

We compare our results with the coulomb counting result and the battery sensor value. The coulomb counting method is known to be more accurate than the open-circuit voltage (OCV) based approaches, and is considered as the golden result here. The Odroid-A platform is equipped with the MAX17010 fuel gauge meter IC from Maxim as the battery sensor. This IC uses OCV to estimate the batterys remaining energy capacity. We charge the battery until the smartphone battery sensor indicates a 100 % SOC and discharge the battery by running different tasks, including 3G and Wi-fi communication and GPS operations. Our proposed method produces remaining energy estimates that are much closer to those obtained by the coulomb counting method (see Fig. 5). The battery sensor produces very pessimistic estimates compared to the coulomb counting method, while the model from [2] shows overestimated results. Note that the primary purpose of the battery fuel gauge sensor is to protect the system from unexpected shut-down, and, therefore, it is reasonable for its remaining energy estimates to be very conservative.

The overestimated result of the model from [2] is mainly because of the underestimation of the battery internal losses due to the rate capacity effect. The authors of [2] discharged the battery with a very small current in the sleep state of the system to minimize the internal losses in the battery. They directly mapped the battery OCV to the battery SOC. The difference between two OCV readouts (and hence, two SOC values) is then translated into a battery depletion rate, and hence, a system-level power consumption value. As a result, the energy difference of the battery SOCs is almost the same as the integration of the power dissipation over time. The internal losses in the battery were ignored during the characterization process.

However, the internal battery losses are strongly affected by the amount of current drawn from the battery. For example, the IR drop loss of the battery caused by the internal resistance grows rapidly, being proportional to the square of the current value. Therefore, the energy estimation results of [2] turn out to overestimate the battery SOC in spite of the fact that their system-level power estimates are quiet accurate. We consider



(a) Power consumption of GPS module according to the operating state

(b) State transition behavior of GPS module

Fig. 6. GPS power and state model.

the battery internal losses as a function of the battery current (which is in turn a function of the power demand of the system depending on what applications are running). As a result, the proposed method shows a mere 2.2 % average error while the model from [2] shows 11.2 % average error.

B. Design space exploration for GPS application

The average energy consumption of GPS operation is approximated from (14) to following equation:

$$\begin{aligned}\Delta\overline{E_{GPS}} &= \Delta t(C^{GPS}_{off}\overline{X^{GPS}_{off}} + C^{GPS}_{sleep}\overline{X^{GPS}_{sleep}} + C^{GPS}_{active}\overline{X^{GPS}_{active}}) \\ &\quad + \Delta\overline{E_{other}} + E_{loss}(v_{OC}, T, N), \qquad (16)\end{aligned}$$

where $\overline{E_{other}}$ is the average power consumption of other devices.

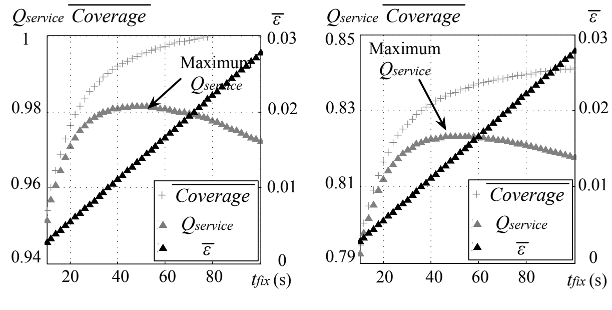
We characterize the GPS status by using the requestLocationUpdate() method with variable t_{fix}. The characterization result of GPS operation is shown in Fig. 6. The relation between t_{fix} and state ratio is as follows:

$$\begin{aligned}\overline{X^{GPS}_{active}} &= t_{active}/t_{fix}, \\ \overline{X^{GPS}_{sleep}} &= \begin{cases} (t_{fix} - t_{active})/t_{fix} & (t_{fix} \leq t_{gooff}), \\ (t_{gooff} - t_{active})/t_{fix} & (t_{fix} > t_{gooff}), \end{cases} \\ \overline{X^{GPS}_{off}} &= 1 - \overline{X^{GPS}_{active}} - \overline{X^{GPS}_{sleep}}, \qquad (17)\end{aligned}$$

where t_{fix} is longer than t_{gooff}. t_{active} and $t_{gosleep}$ denote ACTIVE state time duration per each fixing period and SLEEP state time duration before going to OFF state. Finally, the service time (= battery lifetime), $t_{service}$, is given by

$$t_{service} = Capacity/\Delta E_{GPS}. \qquad (18)$$

We maximize the service quality for the example case illustrated in Fig. 2 (c). The estimated trip time, t_{trip}, for the 523-mile road trip from Sacramento to San diego is 8 h 20 min from Google maps when the average speed is about 60 miles per hour. With the same battery capacity in Section V, the estimated normalized trip coverage, $\overline{Coverage}$, and normalized accuracy, $(1-\overline{\varepsilon})$, for different t_{fix} is shown in Fig. 7. We use 3.5231 J for $\Delta\overline{E_{other}}$, 10 s for t_{active}, and 5 s for t_{gooff}. The other parameters are all the same as presented in Sections V and VI. A combined service quality, $Q_{service} = \overline{Coverage}\cdot(1-\overline{\varepsilon})$ shows convex shape. When we do not consider the battery internal loss, the maximum service quality value where t_{fix} is 55 s. t_{fix} is 48 s when we consider the battery internal loss.



(a) Power estimation only

(b) Considering battery

Fig. 7. Design space exploration for GPS applications (a) only with power estimation and (b) considering battery internal loss.

VII. Conclusion

This paper introduces accurate power estimation, remaining battery charge estimation and power-aware application design for Android OS-based mobile systems. We developed power models for major subcomponents including the CPU, wi-fi, display, audio, GPS, vibration motor, etc., considering their digital and analog states such as display brightness, audio volume, and so on, as well as their on/off states. As none of previous work led the power consumption to the remaining battery life, we derived a battery status of charge model as a function of the smartphone activities. All the device models are based on the measured results of commercially widely used components. We verified the models comparing with real platform measurement showing much more accurate result duration of full-capacity discharging. We also demonstrated power-aware application design with a GPS application practice. We performed design space exploration and introduced quality and battery lifetime tradeoff. The result shows that we achieved the maximum service quality value though the proposed power-aware design framework.

References

[1] A. Shye and et. al., "Into the wild: studing real user activity patterns to guide power optimizations for mobile architectures," in *Proc. Int. Symp. Microarchitecture*, pp. 168 –178, 2009.

[2] L. Zhang and et. al., "Accurate online power estimation and automatic battery behavior based power model generation for smartphones," in *Proceedings of CODES+ISSS '10*, 2010.

[3] M. Dong and et. al., "Sesame: A self-constructive virtual power meter for battery-powered mobile systems," in *Tech. Rep.*, 2010.

[4] Y. Kim and et. al., "System-level online power estimation using an on-chip bus performance monitoring unit,," *IEEE Trans. on CAD*, 2011.

[5] S. Gurun and et. al., "A run-time, feedback-based energy estimation model for embedded devices," in *Proc. Int. Conf. Hardware/Software Codesign and System Synthesis*, pp. 28–33, 2006.

[6] P. Rong and et. al., "An analytical model for predicting the remaining battery capacity of Lithium-ion batteries," *IEEE Trans. on VLSI*, 2006.

[7] D. Rakhmatov, "Battery voltage modeling for portable systems," *ACM TODAES*, 2009.

[8] L. Benini and et. al., "Discrete-time battery models for system-level low-power design," *IEEE Trans. on VLSI*, 2001.

[9] M. Chen and et. al., "Accurate electrical battery model capable of predicting runtime and I-V performance," *IEEE Trans. on Energy Conversion*, 2006.

[10] A. Brown and et. al., "A wireless gps wristwatch tracking solution," in *Proceedings of SDR Forum 2006*, 2006.

[11] *Navsys Corporation*. http://www.navsys.com/.

[12] D. Raskovic and et. al., "Battery-aware embedded gps receiver node," in *Proc. of the International Conference on Mobile and Ubiquitous Systems: Networking and Services*, pp. 1 – 6, 2007.

[13] A. Thiagarajan and et. al., "Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones," in *Proceedings of SenSys09*, 2009.

[14] *Odroid-A platform, Hardkernel Co. Ltd.* http://www.hardkernel.com/.

[15] D. Linden and et. al., *Handbook of Batteries*. McGrew-Hill Professional, 2001.