Update Rate Tradeoffs for Improving Online Power Modeling in Smartphones

Frank Maker, Rajaveen Amirtharajah, and Venkatesh Akella
Department of Electrical and Computer Engineering
University of California, Davis
Davis, California
{flmaker, ramirtha, akella}@ucdavis.edu

***Abstract*—Smartphones offer sophisticated features (e.g., Wi-Fi, GPS, etc.) that require significant energy and limit battery life. Offline smartphone power modeling with benchtop equipment is cumbersome for software developers and takes substantial time to perform on multiple devices. By running on the device itself, online modeling can be performed dynamically and is scalable to many different smartphones. Previous online modeling work used existing battery management unit (BMU) current sensors with a high internal sample rate (18.6 kHz), but very low (software-readable) output register update rates (0.28 Hz). We propose allowing the register update rate to be dynamically adjusted to decrease online modeling time and energy cost. In this work we consider the benefits and evaluate the trade-offs of this approach.**

I. Introduction

Smartphone technology has seen rapid advances in recent years however, battery technology has not kept pace. This mismatch between slowly increasing battery energy density and rapidly increasing functionality has required more focus on software power optimization to maximize battery life. Although offline external power measurements can be made directly using benchtop electrical test equipment and a current sensor, most software developers have little to no experience with this equipment nor outfitting a smartphone battery with a current sensor. Instead of general-purpose benchtop equipment, many developers instead choose a power measurement device designed specifically for smartphones such as the Monsoon power monitor which connects directly to smartphone battery terminals[1]. This external measurement approach is cumbersome because it requires fairly expensive equipment (800+ USD) and is not portable.

In contrast, online internal power measurements do not require extra equipment and have no portability restrictions. These measurements are performed using the battery monitor unit (BMU) IC inside the smartphone battery (See Figure 1). The BMU estimates the remaining battery state of charge (SoC) and communicates with the system over a two wire bus such as I^{2}C or 2-wire. To estimate battery life BMUs have built-in temperature, voltage, and current ADCs. These devices must be highly energy efficient to avoid diverting significant power from the smartphone itself. Previously, integrating ADC architectures that measure the total current over a fixed time period were used to meet efficiency requirements, but, as semiconductor technology has improved, sigma-delta ADCs that measure discrete samples are now used instead. Despite this distinction, BMUs based on both technologies still expose only the *average* current measurements to software over a slow fixed period. For example, the Maxim DS2784 BMU, used by the Google Nexus One, has a 16-bit sigma-delta ADC with a *sample rate* of 18.6 kHz for measuring current, but only updates the average current output register every 3.52 seconds (0.28 Hz) [2]. We presume newer BMUs maintain the same interface for backwards compatibility, but increasing the *update rate* can improve online modeling while leveraging the existing ADC sample rate.

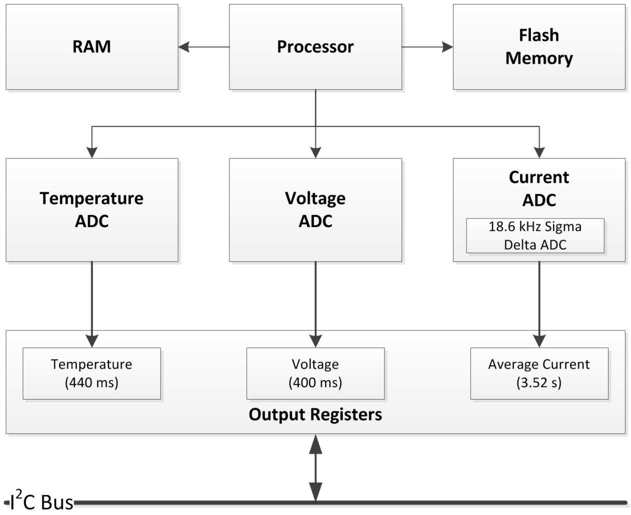

Fig. 1: DS2784 Battery Monitor Unit [2] ADCs, each output register and update rates.

Despite the slow BMU update rate, previous work has shown that accurate power measurements can be made by running software benchmark tests in a loop starting and ending on update rate boundaries to avoid measuring power outside the test [3]. Although this technique overcomes the slow BMU current update rate, more frequent and less time-consuming measurements could be made if the update rate was higher. Since the ADC already samples at a higher rate, only a minor adjustment would be necessary to present faster measurements to software. Furthermore, the I^{2}C interface in fast mode has sufficient bandwidth, 400 kbit/s, to continuously transmit the 16 bit current measurements at 18.6 kHz (297.6 kbit/s).

Faster measurements have multiple advantages for online power modeling, where a model is constructed based on benchmarks which are run for each component and averaged in a loop to characterize each state. The first advantage is to identify

978-1-4799-1235-3/13/$31.00 ©2013 IEEE

"hidden" states that can not be measured by a slow update rate. If the update rate is increased to expose hidden states, the accuracy of the model will then be improved. Second, after an initial request, many components remain in high power states to exploit temporal locality for a predefined fixed period of time, known as a "tail". Faster update rates can measure tail durations to further increase model accuracy. Lastly, shorter update periods reduce the time to run benchmarks and could even allow measurements to be performed at run-time from actual usage instead of benchmarks.

The BMU has only recently been considered for constructing online power models in smartphones [3]. In this paper we expose the limitations of low BMU update rates for online modeling and discuss the benefits of faster update rates. Furthermore, this enables seamless online modeling while an application is running using sophisticated DPM strategies as in [4]. The key questions to consider for increasing update rate speed are: How high should the update rate be? Is there a point of diminishing returns? Is it dependent on the platform as a whole or individual components like GPS, Wi-Fi, etc.? The objective of this paper is to analyze these questions provide the following contributions:

- Quantifying the trade-off between update rate and online model accuracy.
- Identifying the necessary update rates to capture hidden GPS and Cellular states.
- Quantifying the measurement time and energy to build online models based on update rate.

The rest of this paper is organized as follows: First, in Section II we summarize the related work in this area. Second, in Section III we discuss the challenges to low update rate online power modeling. Next, in Section IV we describe the methodology used to measure power and simulate different update rates in this work. In Section V we present the results of our measurements and discuss their significance. Lastly, in Section VI we summarize the conclusions of this work and discuss future work.

II. Related Work

In [3] component-specific software benchmarks are aligned to BMU updates to ensure accurate measurements. This work is most closely related to ours because we use their benchmarks and update rate alignment to compare the DS2784 low BMU output register update rate to faster rates. [5] also runs component-targeted benchmarks, but instead of directly generating a model uses a genetic algorithm instead. [6] does not develop component benchmarks, but instead uses preexisting third-party benchmark suites with linear regression to construct the model. [7] uses voltage measurements and the battery discharge curve instead of a current sensing BMU. The voltage before and after each component test is recorded to calculate the energy dissipated and infer the current measurement. This approach requires more time than current sensing, but can be used by any BMU. Each of these techniques can leverage our work to determine the appropriate update rate for modeling accuracy and speed.

Similarly related are offline modeling works which use external measurements to construct a model *a priori*. [7] manually constructs a linear model from component states and their utilization with the coefficients determines by measuring each component state benchmark. [8] takes a markedly different approach at the system call layer. Each system call is measured and a high accuracy, fine granularity model is constructed. Our work clarifies the update rate requirements for these techniques to be accurate which may enable them to be implemented online with faster BMU sampling.

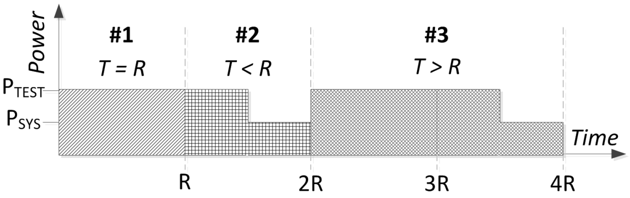

Fig. 2: Three scenarios for relationship between the test time, T, and the BMU update rate, R. The ideal case, $T = R$ and measurement error scenarios $T < R$ and $T > R$.

Another area of closely related work is dynamic power management (DPM). Many hardware components, such as the CPU, Wi-Fi, and GPS, have multiple power states with varying performance. DPM leverages this positive correlation to throttle performance, and therefore power, to match the component workload. Several policies that have been proposed for DPM based on predictive and stochastic methods are comprehensively surveyed in [4]. Our work complements the DPM literature because we consider the sampling frequencies and update rates required to exposure additional, previously unmeasured "hidden states". By incorporating these states into the power state model, power can be estimated more accurately.

III. Online Power Modeling Challenges with Low Update Rate

A. Update Rate Aware Measurement Accuracy

The accuracy of slow update rate power measurements is highly dependent on aligning the power characterization benchmarks to the update interval edges as identified in previous work[3]. In Figure 2, three different scenarios of update rate-aware measurements are shown with each test running for a different time T, at power P_{TEST}, being measured on a system with an idle power level of P_{SYS}, and a BMU update interval of R. Each measurement's accuracy depends on the test running during the entire update interval. If it does not, the difference between the test and system power impacts the measurement error. This relationship can be described by the following equation:

$$P_{AVG} = \frac{P_{TEST}T + P_{SYS}(M \cdot R - T)}{R}$$

Test #1 represents the ideal case where the test time equals an integer multiple of the update rate, $T = M \cdot R$ for $M = 1$, and consequently $P_{AVG} = P_{TEST}$. Test #2 results in a measurement error since a fraction of the power averaged is at P_{SYS} and therefore $P_{AVG} < P_{TEST}$. Last, Test #3 demonstrates a common scenario in practice where a test runs for more than one update interval, but does not end on an update ($T = M \cdot R + \Delta$). The accuracy of this test in determining a particular hardware component's power consumption will increase with each successive measurement interval as the proportion of all the measurements set by the system power decreases.

These scenarios demonstrate potential measurement pitfalls when the timing of a test is not aligned to its power measurement. However, if the update rate could be increased, the measurement granularity would be improved and errors in each case would be reduced. We propose to reconfigure the BMU to enable dynamic update rates where instead of aligning each test to a fixed update rate, each power measurement begins and ends with the test itself. This would reduce the errors for this technique and decouple test design from the update rate.

B. Hidden States

The limited update rate of the BMU prevents some component power states (known as "hidden" states) from being detected in power measurements. These states are effectively too brief in duration to be detected by a slow BMU update rate. Our measurements exposed two such states: GPS ACTIVE and the initial Cellular FACH. The GPS ACTIVE state consists of a brief communication with GPS satellites to determine a new location and lasted only 215 ms in our measurements (Figure 3). The cellular radio states are unique to each carrier's radio resource control (RRC) protocol state machine settings (Figure 4) and are transmitted to the device from the cellphone tower. These timings can be less than 500 ms as demonstrated by the initial FACH state before transmission shown in Figure 5. Last, measurements can be impacted by other component power states, which we call "confounding states", interfering with the power state of the component under test.

Confounding states can also be hidden if their duration is short compared to the update interval and but their power is sufficiently high to cause measurement errors. In Figure 6 a confounding state caused by the user scrolling through UI menus was captured during a CPU benchmark run, for which the power consumption would normally remain constant. This state would be difficult to detect pragmatically because it lies within the range of the benchmark measurement.

IV. Methodology

To measure power consumption we used a Tektronix MSO 4034 Mixed Signal Oscilloscope to probe the voltage across a 0.1 Ω current sense resistor. Since we measured active mode system power and not sleep or idle current the signal range was well above the minimum oscilloscope resolution and the noise floor. The resistor was connected between the positive power rail of the battery and the positive terminal on the smartphone to measure the current drawn from the battery. As recommended in prior work [9] the connection was made using copper braid wire to avoid damage to the battery or smartphone.

In order to compare different sampling frequencies we measured each component's power states at 50 kHz. After each measurement, the results were stored to a comma separated values (CSV) file for analysis. To simulate lower frequencies we designed lowpass FIR filters with cutoff frequencies at the Nyquist rates corresponding to the sampling frequencies and with steep ($\geq$ 40 dB/decade) roll-offs. The BMU I^2C measurements were calculated by averaging the BMU internal sampling rate results at the update rate specified in the DS2784 data sheet, 3.52s [2].

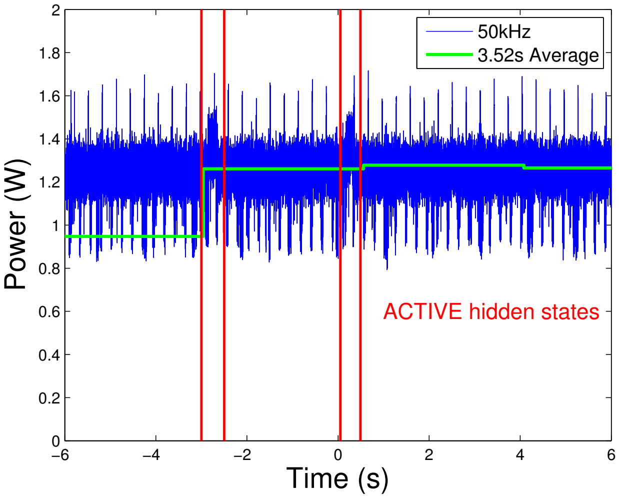

Fig. 3: GPS benchmark with two ACTIVE states exposed at a 50kHz update rate, but hidden to the slow BMU update rate (0.28 Hz).

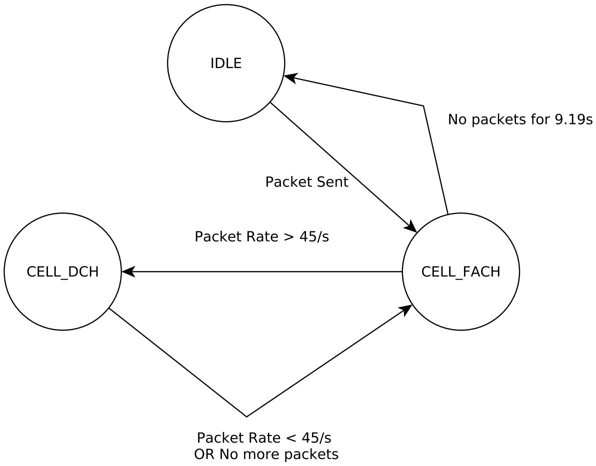

Fig. 4: Radio Resource Control (RRC) power state machine for the cellular radio with measured threshold of 45 packets/s and FACH tail timeout of 9.19 s.

V. Results and Discussion

In this section we will first present our power measurement results and also discuss the required update rates to capture hidden GPS and Cellular states based on our highest frequency measurements. The implications of update rate on online power model resource consumption are also analyed.

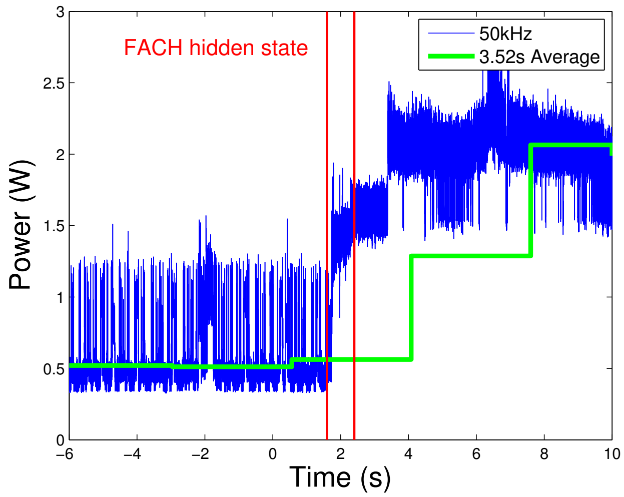

Fig. 5: Cellular benchmark with FACH state exposed at a 50kHz update rate, but hidden to the slow BMU update rate (0.28 Hz).

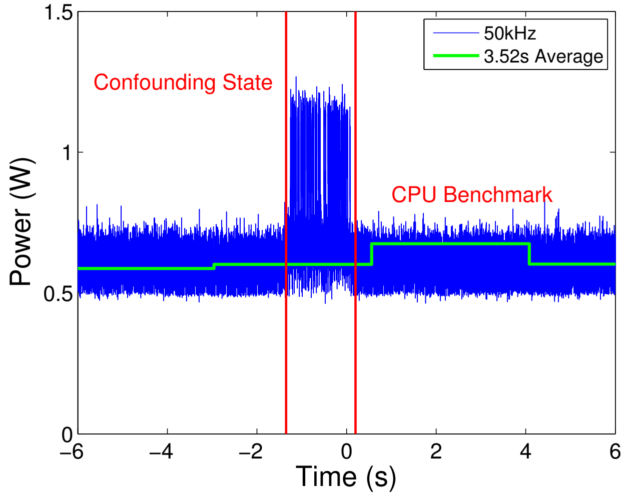

Fig. 6: CPU 245 kHz benchmark with measurement confounded by UI menu scrolling exposed at a 50kHz update rate, but difficult to detect with the slow BMU update rate (0.28 Hz).

A. *Power Model Error and Hidden States*

A subset of our measurements at different frequencies for each component power state are shown in Table I. As discussed in Section III-B, the GPS ACTIVE state could not be captured by the slow BMU update rate and is marked N/A in the table. In all our measurements, the error, compared to the 50 kHz measurements, continuously increased as the sampling rate was decreased.

BMU I^2C results were nearly identical to the BMU internal 18.6 kHz sample rate average since the former is simply a periodic average of the latter. The discrepancy between the measurements is due to the total measurement time, 20 s, being not wholly divisible by the BMU I^2C update interval, 3.52 s; consequently, not all measurements could be included in

Fig. 7: Tektronix MSO 4034 Oscilloscope used for power measurements connected to Google Nexus One.

		Power (mW)					
Comp.	State	50kHz (Osc.)	18.6kHz (BMU)	5kHz (Monsoon)	100Hz	10Hz	3.52s Avg. (BMU I^2C)
CPU	245kHz	190	190	190	187	173	190
	384kHz	261	261	261	257	241	261
	461kHz	299	299	299	294	277	299
	499kHz	337	338	337	333	314	338
	576kHz	384	384	384	379	359	384
	614kHz	417	417	417	412	392	417
	653kHz	438	438	438	432	411	438
	691kHz	498	498	498	492	470	498
	768kHz	564	564	564	558	533	564
	806kHz	607	607	607	600	534	607
	845kHz	676	676	676	669	641	676
	998kHz	818	818	818	810	779	819
GPS	ACT.	957	955	956	937	896	N/A
	ON	849	849	849	851	850	752
Display	5	110	110	110	107	93	110
	30	202	202	202	199	184	201
	55	324	323	324	319	300	323
	80	472	472	472	467	446	471
	105	571	571	571	565	51	570
	130	715	714	715	708	679	714
	155	813	813	813	804	773	813
	180	916	915	915	907	873	915
	205	1025	1024	1025	1016	980	1023
	230	1192	1191	1191	1182	1142	1191
	255	1312	1311	1311	1300	1256	1311
Wi-Fi	8	342	342	342	337	317	342
	10	351	351	351	347	328	350
	12	364	364	364	358	339	363
	14	361	360	360	354	336	360
	18	376	375	376	371	352	376
	20	378	378	378	373	353	378
	25	388	387	388	382	362	387
	30	394	393	393	389	367	393
	35	410	409	409	404	382	409
	40	408	408	408	403	381	408
	45	426	426	426	421	401	426
	50	389	388	389	384	363	389
	100	428	427	428	422	400	428
	125	452	452	452	447	426	452
	200	492	491	492	486	464	490
	250	519	519	519	513	490	520
	500	644	644	644	637	611	643
Cellular	IDLE	421	420	421	432	476	419
	FACH	1099	1091	1097	881	895	1073
	DCH	1341	1342	1341	1329	1275	1343

TABLE I: Power measurements for each component with different update rates.

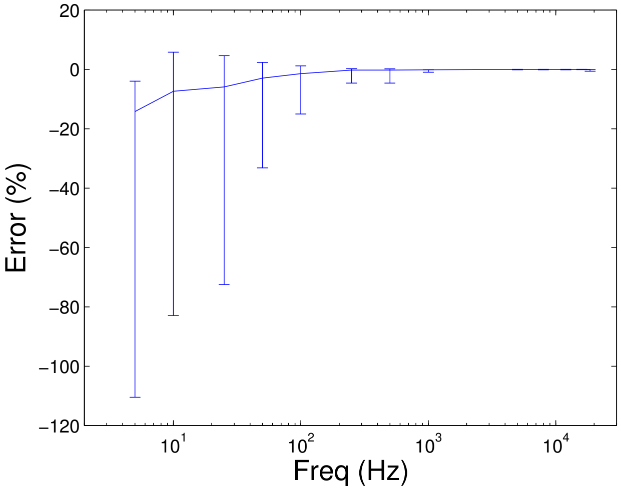

Fig. 8: Average, minimum, and maximum power model error versus update rate. Significant errors appear at 100 Hz and below.

the final averaged result. If the measurements were properly aligned to update intervals the results corresponding to the internal BMU sample rate and the BMU I^{2}C update rate would be indistinguishable.

Figure 8 plots the mean, maximum and minimum error versus update rate. Overall we found negligible impact on model error for update rates $>$ 1 kHz. At lower frequencies, the error increases until reaching -96.28% at 5 Hz. From these results we conclude that an update rate of 1 kHz is sufficient to accurately measure the benchmarks tested. Significant errors start appearing at rates below 100 Hz.

The two hidden states observed in our tests, GPS ACTIVE and the initial cellular FACH state, lasted for 221 ms and 454 ms respectively. The corresponding update rate required to capture these states would be 9.05 Hz and 4.41 Hz, respectively. Capturing hidden states would therefore require only a modest increase from the existing output register update rate of 0.28 Hz.

B. *Online Modeling Resource Consumption*

Next we compared the resource consumption required to build the power model using each update rate in Table II. Both time and energy were estimated to evaluate the delay in rebuilding the model and the cost of running each benchmark. The energy consumed was also calculated as a fraction of the 1400 mAh battery capacity (with a nominal voltage of 3.7 V) since the ultimate goal of power optimization is to maximize battery life.

To determine how long each component required for testing we sorted them into two different groups. One group of components have states with roughly constant power consumption: CPU, Display and Wi-Fi. Figures 9 and 10 show histograms for CPU and display power states respectively. Each histogram peak corresponds to a power state and its position on the x-axis corresponds to the power consumption of that state. For Wi-Fi each packet rate has a constant power proportional to the rate used, however these values are difficult to distinguish in a histogram because of the small difference between the rates (See Table I). Instead, in Figure 11 the power for each rate is shown with two line-point plots based on the threshold value of 45 packets/s, where rates less than the threshold are “low power” rates and rates greater than the threshold are “high power”rates[7]. Since the power consumption of all three of these components can be measured quickly, our resource estimate assumes 100 samples are sufficient at each measurement frequency to measure each state accurately. The time to test each component is then calculated as the summation of the times required to test each state.

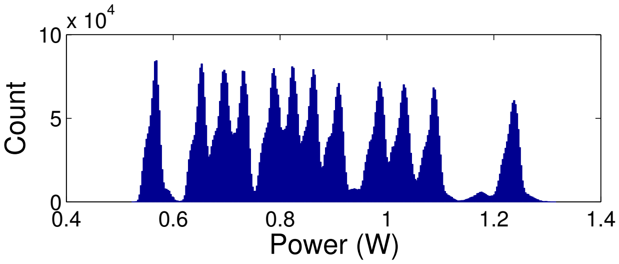

Fig. 9: CPU power measurements histogram. The peaks correspond to twelve distinct power states while each peak’s position on the x-axis corresponds to its power consumption.

The second group of components, GPS and Cellular, have multiple states which require a fixed period of measurement time to be exposed. For these components we used the same test time for each measurement frequency based on the BMU update interval, R. For GPS, we chose $1R$ as a sufficiently long period since our tests (Figure 3) in an indoor office environment captured two location updates in this time period. The cellular power state machine depends on the carrier RRC settings (Figure 4) and require a sufficiently long time to discover these settings. We chose a longer period of $5R$, recommended by previous work [3], to ensure all the states would be captured. Based on the shortest update periods for each of these multi-state components the minimum sample frequencies of 9.05 Hz and 4.41 Hz for GPS and Cellular respectively are required to capture all their states.

A notable improvement in resource consumption was found for all update rates compared to the BMU I^{2}C rate because measuring constant power state components (CPU, Display, Wi-Fi) can be completed much more quickly. These represent an average improvement of 139 s, 69 J and 5.02% battery life. Lower resource consumption means the online power model can be updated more frequently, which can in turn provide more accurate battery life predictions or even capture real usage instead of relying on previously constructed benchmarks. In fact, from a resource consumption perspective, the bottleneck in speeding up online modeling is the long fixed power measurement update interval and not the internal BMU ADC sampling frequency.

VI. Conclusions and Future Work

In conclusion we demonstrated that the existing BMU internal current sampling rate is entirely sufficient to generate accurate online smartphone power models, whereas the software-

	50 kHz			18.6 kHz (BMU Internal)			5 kHz (Monsoon)			3.52s Avg. (BMU I^2C)		
Component	Time	Energy	Battery (%)	Time	Energy	Battery (%)	Time	Energy	Battery (%)	Time	Energy	Battery (%)
CPU	0.06	0.01	0.00	0.06	0.03	0.00	0.06	0.12	0.01	42.24	16.96	1.24
Display	0.06	0.02	0.00	0.06	0.04	0.00	0.06	0.19	0.01	38.72	26.94	1.98
Wi-Fi	0.09	0.01	0.00	0.09	0.04	0.00	0.09	0.18	0.01	59.84	25.06	1.84
GPS	3.52	3.04	0.22	3.52	3.04	0.22	3.52	3.04	0.22	3.52	3.04	0.22
Cellular	12.71	16.03	1.18	12.71	16.03	1.18	12.71	16.03	1.18	12.71	16.03	1.18
Total	16.45	19.11	1.40	16.45	19.17	1.41	16.45	19.56	1.44	157.03	88.02	6.46

TABLE II: Resource consumption to rebuild model by measurement technique.

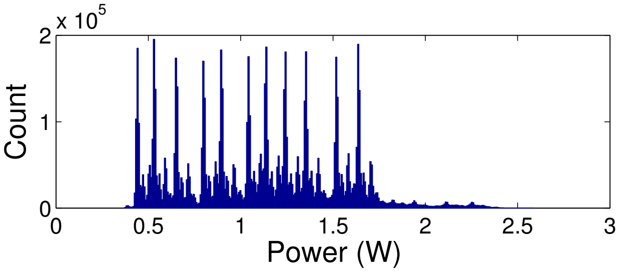

Fig. 10: Display power measurements histogram. The peaks correspond to eleven distinct power states while each peak's position on the x-axis corresponds to its power consumption.

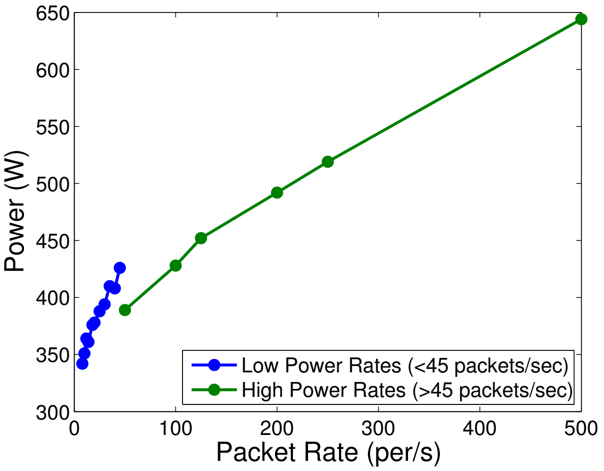

Fig. 11: Wi-Fi benchmark power versus packet rate.

readable register update rate is not. A faster update rate was required to avoid measurement errors, uncover hidden states, and decouple tests from update rate intervals. Rebuilding an online software model was 9.5x faster with faster update rates compared to the existing BMU rate. We establish the clear advantages to increasing the BMU current update rate exposing the faster internal ADC sampling rates for improved online modeling. In future work we will develop a theoretical basis for configuring the BMU update rate to optimize its overhead and accuracy for online modeling.

References

[1] *Mobile Device Power Monitor Manual, v1.12*, Monsoon Solutions, Inc., http://www.msoon.com.

[2] *DS2784 1-Cell Stand-Alone Fuel Gauge IC with Li+ Protector and SHA-1 Authentication, Rev 3/12*, Maxim Integrated Products.

[3] W. Jung, C. Kang, C. Yoon, D. Kim, and H. Cha, "DevScope: a nonintrusive and online power analysis tool for smartphone hardware components," *CODES+ISSS '12 Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, 2012. [Online]. Available: http://dl.acm.org/citation.cfm?id=2380502

[4] L. Benini, a. Bogliolo, and G. De Micheli, "A survey of design techniques for system-level dynamic power management," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 8, no. 3, pp. 299–316, Jun. 2000. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=845896

[5] M. B. Kjæ rgaard and H. Blunck, "Unsupervised Power Profiling for Mobile Devices," in *Proceedings of the 8th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (Mobiquitous 2011).* Springer, 2011.

[6] M. Dong and L. Zhong, "Self-constructive high-rate system energy modeling for battery-powered mobile systems," *MobiSys '11 Proceedings of the 9th international conference on Mobile systems, applications, and services*, 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=2000027

[7] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, and L. Yang, "Accurate online power estimation and automatic battery behavior based power model generation for smartphones," in *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, ser. CODES/ISSS '10. New York, NY, USA: ACM, 2010, pp. 105–114. [Online]. Available: http://doi.acm.org/10.1145/1878961.1878982

[8] A. Pathak, Y. C. Hu, M. Zhang, P. Bahl, and Y.-m. Wang, "Fine-grained power modeling for smartphones using system call tracing," in *Proceedings of the sixth conference on Computer systems - EuroSys '11.* New York, New York, USA: ACM Press, 2011, p. 153. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1966445.1966460

[9] F. Fitzek, "External Energy Consumption Measurements on Mobile Phones," *Mobile Phone Programming: and its Application to Wireless Networking*, pp. 441–447, 2007.