On the Effectiveness of Energy Metering on Every Node

Qiang Li
Dept. of Computer Science
University of Houston
qli8@uh.edu

Marcelo Martins
Dept. of Computer Science
Brown University
martins@cs.brown.edu

Omprakash Gnawali
Dept. of Computer Science
University of Houston
gnawali@cs.uh.edu

Rodrigo Fonseca
Dept. of Computer Science
Brown University
rfonseca@cs.brown.edu

***Abstract*—Making wireless sensor node platforms energy-efficient is one of the major research thrusts in the sensor network community. Energy metering lies at the foundation of this research, either by providing direct measurements for profiling, or by serving as the base for the formulation and fitting of energy-usage models. Most of the literature and tools, however, make their measurements on a very small subset of the node population, and usually at a single point in time, before deployment. In this paper we set out to evaluate the cost, in loss of precision, of not having constant and ubiquitous measurement. Through experiments on several mote-class and smartphone-class platforms, including a 240-node sensor-network testbed, we find that the variations in energy consumption due to temperature change are small, and we establish a model between environmental temperature changes and power consumption of Quanto testbed motes. We also find that different nodes of the same kind can have up to 15% variation in power draw, suggesting a need to deploy instrumentation on a subset of nodes. We quantify the energy estimation error of different metering techniques and characterize the conditions in which the errors disappear. Overall, we find that a small number of measurements in time and across nodes is adequate for accurate estimation of network-wide energy use.**

I. Introduction

Energy is the most limiting resource in a sensor network deployment, as its usage determines the frequency and nature of the activities the network can perform, and, ultimately, its lifetime. Understanding and optimizing this usage, then, is of utmost importance for developers and users alike. Not surprisingly, energy-efficiency has been a major research thrust in the community. There exists a vast literature, dating back to the advent of mobile computing, on measuring and profiling [31], [27], modeling [4], and predicting energy [25] on a variety of platforms.

Measurement or estimation of power used by a node or a network allows us to reliably compare energy efficiency of different systems or protocols. Such comparisons allows us to validate if a proposed idea advances the state of the art in energy efficiency in wireless sensor networks.

Two approaches to estimate power have found widespread use in the community. One can directly measure power [31], [13], [14]. Or, one could estimate power using a model based on events observable by the running system, such as performance counters [2], [3], system calls [6], [24], messages sent [33] or time spent at different activities [12], [8], [34]. For the approach that uses energy models to estimate power, the ground truth established from real measurements lies at the foundation of their results.

Due to resource limitations, the infeasibility of deploying measuring equipment alongside nodes in a network, or simply because the reason for having models is not to have to measure on deployed nodes, most of these studies only measure energy consumption on one [32], or a few [16], [20] nodes, extrapolating or generalizing the results both in the time and space dimensions, i.e., applying the results from measurement time to run time, and from one node to other nodes. As an example, the Motelab testbed [32] included one node connected to a digital multimeter (no doubt a great resource!), but the user had to draw any energy-related conclusion for a deployment by extrapolating measurements from this node. The same approach is used to estimate energy use in simulators. For example, powerTOSSIM [28] comes with a model for how much energy is used in what operation. This model is used to extrapolate the energy used by a network of nodes during the simualtion.

As a result of the same limitations that prevent ubiquitous measurement, *assessing the impact of these extrapolations is hard, and rarely done*. There are two main concerns with such extrapolations. Do we need to use different energy model at different times or different temperatures? Many sensor networks are deployed in places where there could be a large change in temperature. Second, does the power model constructed with one or a group of nodes accurately represent the power used by all the nodes in the network, even when the workload is identical?

In this paper, we set out to answer precisely these questions, by looking at the difference between ground truth energy measurements and model-based extrapolation that is widely used in the sensor network research community. Using a testbed with 240 nearly identical nodes all equipped with calibrated energy usage meters, under both simple and realistic workloads, we characterize the accuracy of model-based estimation of sensor network energy use.

We find that power draw variation across temperature is small suggesting we do not need to change the model over time. This conclusion suggests the soundness of the most common practice in the community: we typically use the same model to extrapolate energy over time, even in a long running deployment. Similarly, in simulations, we use the same model over time. This paper presents the first dataset to validate this practice.

The result of power measurements across the nodes, however, suggests that a simple model-based approach can cause large errors in energy estimation. Across nodes, we found that even for identically constructed nodes, there was a difference of up to 15% in energy usage when running the same simple workload. Furthermore, the error is not eliminated even if we form a model using measurements on a group of nodes. While variations across devices of same types have been published in the context of smartphones [35], this paper presents the first dataset showing the variation in power across mote-class sensor nodes.

Our measurements reveal that the distribution of power draw across the nodes is *normal*. We find that metering even a small number of these nodes allows us to improve the accuracy of network-wide energy use extrapolation. In our experiment, a model formed by measurements on 15% of the nodes and using that model to predict the network-wide energy use resulted in an estimation error of up to 2.5% while the best fitting model would result in an error of 1.8%.

II. Related work

There is an extensive body of work that measures and models energy usage in embedded and portable devices. Instrumentation is the preferred method as it delivers a quasi-ground-truth values, but it is usually obtrusive or requires modifications to existing sensor platforms.

SPOT [18] is a sensor board containing a complete and very accurate energy meter that can be read using the sensor node I^2C bus. This solution allows the sensor node to measure its own energy consumption, but the energy meter itself draws non-negligible power. Energy Bucket [1] delivers a constant voltage to the target system while counting the number of charge/discharge cycles of a buffer capacitor. Considering that each cycle transfers the same amount of charge, we can measure the amount of energy per cycle. The Bucket solution also includes a software library that can be used to track the energy state of the capacitor and relate it to program states. The Energy Endoscope [30] is an integrated, low-power, real-time energy monitoring system for Linux-based sensor nodes running on top the LEAP2 platform, a low-power ASIC capable of observing energy usage of multiple subsystems in real-time. Energy is allotted to applications using a low-overhead kernel-space energy measurement tool that integrates with the observing hardware.

When detailed instrumentation is not available on a given platform, we must resort to modeling. We divide the modeling literature into two groups:

Counter-based models: Bellosa [2] pioneered the use of linear models based on event *counts*, such as those provided by performance counters, by noticing strong correlations between specific counters and CPU power. Several other works followed using similar approaches, varying in granularity and in the scope of events used as inputs. Contreras and Martonosi [7] used performance counters and a linear model for online power estimation of CPU and memory in an XScale platform, reporting average relative errors of 4%. Bircher and John [3] extended a similar model to additionally account for energy used by the chipset, I/O, and disk subsystems, and reported errors of less than 9% for all subsystems. Mantis [12] used measurements of CPU and disk utilization in addition to performance counters for whole-system power profiling, and achieved errors, for two platforms, within 15%. More recently, McCullough *et al.* [22] compared several linear and non-linear regression models based on performance counters, and found that despite achieving good accuracy, these models perform worse for individual hardware components that change power states with no signals to the OS. In the scope of WSNs, PowerTOSSIM [28] uses detailed measurements of the current draw for each node component (MCU, radio, flash, LEDs, sensors) in each mode (sleep, active, etc.) to infer energy-consumption models to be used in simulation software.

Finite State Machine models: An implicit assumption of the counter-based models is that each occurrence of an event implies some energy expenditure. This assumption breaks down if the marginal energy cost of an event is low compared to the active power draw of a subcomponent. For example, in the CC2420 radio [17], used in wireless sensor networks, the power for transmitting a packet is almost the same as the listening power, and that the number of transmissions is a very poor estimator of energy usage [21]. Another problem with counter-based models is that they require sampling, and suffer from the inherent tradeoff between overhead and agility when selecting a sampling rate.

An alternative approach that does not rely on this assumption is to model the hardware subcomponents as finite state machines (FSMs), and use events to trigger transitions. These models can achieve higher accuracy than their counter-based counterparts, as they arguably model devices as what they really are: state machines. FSM models can more readily accommodate domain knowledge, such as notions of batching and timeouts, and can account for events or conditions which cause state transitions, but do not, on their own, incur energy usage.

In the realm of wireless sensor networks, Dunkels *et al.* [8] modeled hardware components as two-state FSMs, and used a linear model based on the time each component spent on each state, multiplied by the previously measured power draw of these states, to predict energy usage online. Kellner [19] also used a state machine to model hardware components in a sensor-network platform. The states are those that have distinguishable power draws, and they also assign a fixed amount of energy per transition. Quanto [14] breaks the hardware subcomponents into logical units that can be on or off and instruments device drivers to inform the OS of the state of these units. It then splits the measured full-platform energy provided by iCount [10] among the units based on a linear model on the time each unit was on. Cignetti *et al.* [6] also used a set of FSMs to model components in the Palm palmtop computer, and suggested the use of system calls to trigger transitions in the states of the model. Using a similar model, but in the context of modern mobile phones, Pathak *et al.* [24] showed that an FSM model using input from the system-call layer could significantly outperform counter-based models. They attribute the improved performance to the presence of tail power states, system calls that change power state but imply no utilization, and to components that have no quantitative utilization, but have distinct power states.

III. PLATFORM MEASUREMENTS

In this section we describe our measurement study, establishing the variation in power draw of four different platforms under different temperatures and across nodes.

A. Platforms and Workloads

Our experiments involve four different platforms: the Quanto Testbed Mote [9], [11], the TelosB motes [26], Android smartphones, and the LEAP platform [29].

Quanto Testbed Mote. The Quanto Testbed Mote has a mote core, an energy meter and calibration hardware. The mote core serves as the "device under test" whose energy consumption is to be studied. The platform is based on the TI MSP430F1611 microcontroller and uses the CC2420 radio. The mote core exposes the power supply lines for the microcontroller, ADC, radio, and flash, allowing these distinct power domains to be individually monitored. The iCount energy meter [10] measures energy consumption by the device under test and exposes this information to the device. To do so, it adds a single wire between the built-in switching regulator and the core microcontroller. The switching signal is directly connected to an interrupt on the microcontroller, which steers the voltage excursions to VCC or GND. Excursions occur at every switch cycle, which permits us to calculate the energy per quanta.

iCount Calibration. Even with input-voltage regulation, an instrument is only as good as the fidelity of its calibration process. If the energy data has to be compared between different nodes on the network, it is essential to calibrate each node to establish a relation between the switching frequency and the energy per switch. To ensure that calibration is possible on-the-fly, the Quanto Testbed motes include dedicated circuitry to perform six-point calibration using 0.1% tolerance precision resistors of 3M, 300k, 30k, 300, and 60.4 ohm. A multiplexer allows the resistors to apply a range of loads to the iCount regulator, from $10\mu A$ to $50mA$, which roughly corresponds to the operating range of the Epic. When operating in the calibration mode, the mote load bypasses the iCount regulator and is directly powered from a linear regulator. Since we know the voltage that gets applied to the resistors, we can calculate the total energy consumed by the resistor over a certain time period. In most of our experiments, we calibrate the hardware just prior to starting the data collection and immediately after completing it.

Quanto Testbed Infrastructure. The testbed has 240 Quanto Testbed motes spread over multiple rooms. Each mote runs a Quanto-enabled version of TinyOS to form a network-wide energy profiler. Each mote can be programmed with TinyOS application, the iCount calibration code, and the energy profiling code. The motes can send the energy readings over the Ethernet backchannel. There are a large number of temperature sensors in the building for monitoring the indoor climate. We use the data from these temperature sensors to determine the environment temperature for the motes on the testbed at any point of time during the experiments.

TelosB. The TelosB mote is an open-source sensor platform designed for academic research with an architecture very similar to the Epic, but without the energy meter and calibration hardware. We use six TelosB motes in our experiments, and measure their power draw under different temperatures connecting them directly to a multimeter. In our experiments, the TelosB motes are powered via an external power supply instead of batteries to ensure a stable voltage.

Smartphones.. We measure power-draw on three Android smartphones: one HTC HD2 and two Google Galaxy Nexus. We install CyanogenMod 9 ICS 4.1.0 on each device and run the `CF-Bench` [1] benchmark application. `CF-Bench` is a collection of microbenchmarks that separately stresses three different components of a smartphone: CPU, memory and disk. For all considered smartphones, `CF-Bench` runs for approximately 140 seconds. We measure the power draw of the entire phone connecting a Monsoon power monitor to the smartphone battery. For each workload, we run the microbenchmarks using three different temperature settings: ambient, hot and cold; and set the screen brightness to its lowest value.

Atom LEAP. The last platform we use is LEAP. LEAP is a new embedded computing platform based on the Intel Atom processor architecture that enables researchers to acquire component-level power measurements.

Workloads. On the motes, we run three different workloads: CPU-idle, CPU-intensive, and Radio-on. The longest mote experiment lasted 43 hours switching between these workloads and calibration loads. We also run collection workload on the motes. During this workload, the motes run a collection protocol called CTP [15] to collect readings from the entire network to a single sink. When the network runs CTP, different nodes might spend different amount of energy because some nodes forward more packets than others. On the smartphones, we run five different workloads: idle, CPU-intensive, Memory-intensive, Flash read and Flash write. On the Atom LEAP platform, we run three workloads: Idle, CPU-intensive and Memory intensive.

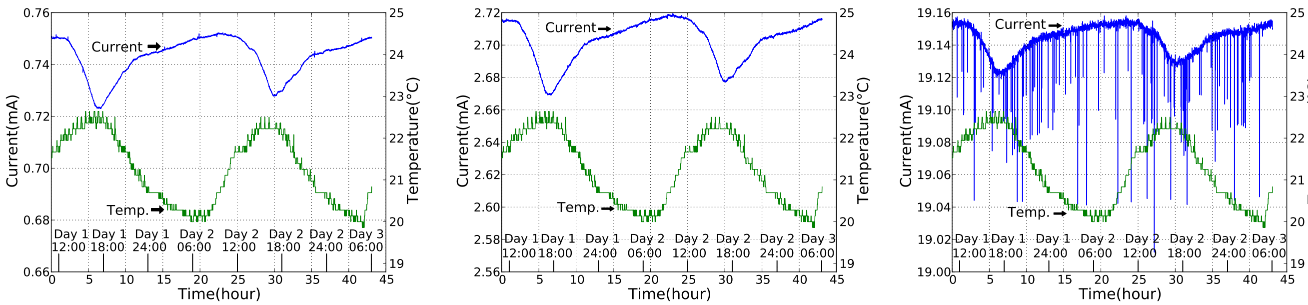
B. Impact of Temperature Change

To understand how device current changes with environmental temperature, we run a series of simple workloads with known and fixed power draw profile for 43 hours on the 240-node testbed. Specifically, we perform energy measurements while continuously cycling through the following four tasks, with each task running for four seconds with an interval of 0.5 seconds between tasks.

- CPU intensive workload
- Turn the radio on
- Connect 60-Ω precision resistor
- Connect 3K-Ω precision resistor

The last two measurements are used to calibrate the iCount readings. Each set of measurement takes about 40 seconds. At the end of the sequence, the mote idles for five seconds, and sends the readings to the database and starts a new sequence of measurements. All power, energy, and current measurements shown in the paper are calibrated. We collected one million calibrated energy readings each for CPU intensive and Radio-on workloads during the experiment.

[1]http://play.google.com/store/apps/details?id=eu.chainfire.cfbench



(a) idle (b) CPU-intensive (c) Radio on

Fig. 2. Current draw of a Quanto mote with different workload vs. Temperature.

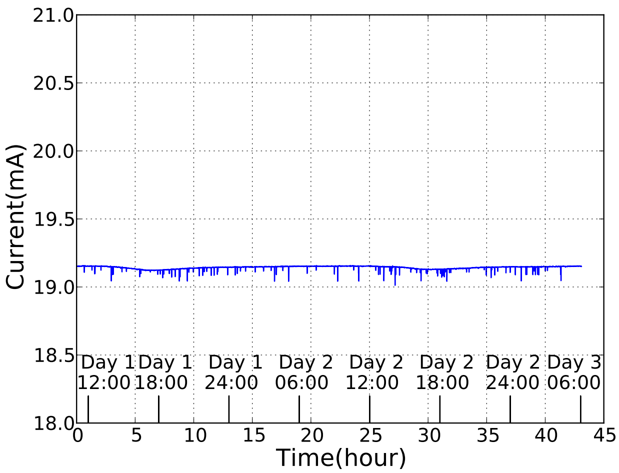


Fig. 1. Current draw of a Quanto mote with radio on.

Figure 1 shows current draw of a mote on the testbed. We find that the current draw remains approximately flat during the course of the measurement. Figure 2 magnifies the small changes in current by using an extremely small range for the y-axis. We find that these changes are correlated with the change in the environment temperature, also shown in the graph. The current is largest in the morning and smallest in the afternoon. Upon analysis of current draw of all the motes across the 43-hour trace, we find that the change in current draw never exceeds 0.3% even for the mote that seems most sensitive to temperature.

We choose 96 out of 240 nodes, which are deployed in the same open office and thus in same temperature, to see if it is possible to predict current draw according to temperature when running specific workload. From figure 3, we can see that when running radio-on workload, the nodes show considerable variations among each other, but most of them appear as flat lines, which indicates that there is a linear correlation between temperature and power consumption. What's more, we find that the median line and the mean line across temperature are nearly identical, implying that the distribution is even. This conclusion also holds for idle and CPU-intensive workload. More important, the mean and median change up to only 0.05% across temperature. This difference in current may not be important, because sensor network deployment planning rarely requires this level of precision in energy budgeting. All these facts, combined together, suggest that given a specific type of motes, a specific workload, it is feasible to estimate power draw of a whole network directly from a model, which can be established before deployment, regardingless of temperature.

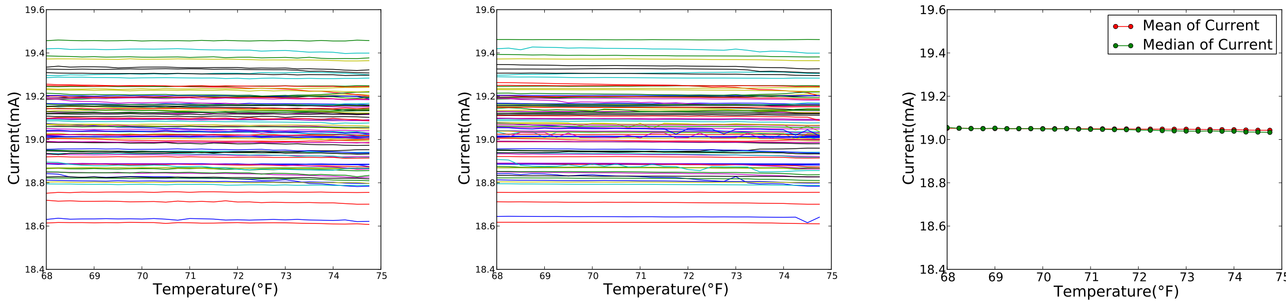
One limitation of above current measurements on the 240-node testbed is the small change in temperature. During the 43 hours of measurements, the temperature changed by approximately 5°Cpeak to peak. A sensor network deployment may be subjected to a much wider range in temperature such as below freezing point at night and burning-hot during the day.

Next, we perform measurements to determine if the current draw changes would be more significant if the motes are subjected to much higher ranges of temperature. We are not able to control the temperature to such extremes in the office building where the 240-mote testbed is deployed. We instead perform the experiments with more extreme temperature changes in a lab environment where we can control the temperature easily.

We put two Quanto Testbed motes through three different temperatures and measure the current draw. We found that the difference is at most 1.6% even across a temperature change of 59°C(-14°Cvs 45°C). For the CPU workload, which draws the smallest total current, although the relative change is larger, the absolute change is even smaller. The absolute numbers matter most in energy budgeting because it is directly related to the energy available for the platform.

To understand if this trend of limited change in current draw even across a large range of temperature holds in other platforms, we perform experiments with TelosB motes, Smartphones, and a LEAP node subjecting them to a large change in temperature in a lab environment where we can put them in a temperature controlled box.

We find that TelosB shows slightly larger increase in current draw than Quanto Testbed motes as we increase temperature. Table I shows the measurements from one TelosB mote. Measurements with 6 other TelosB motes show similar trend: up to 2% change in normal case. Although the 13.3% increase in current for Idle workload looks large, the change in current itself is even smaller than other workloads.



(a) Mean Current of Nodes (b) Median Current of Nodes (c) Aggregated Mean & Median

Fig. 3. Current draw of 96 Quanto motes with radio-on workload vs. Temperature.

Workload	Curr@-11°C (mA)	Curr@43°C (mA)	-11°C vs. 43°C
Idle	0.015	0.017	13.33%
CPU	1.805	1.841	1.99%
Mem	1.861	1.899	2.04%
FMem-Read	1.783	1.819	2.02%
FMem-Write	1.794	1.830	2.01%
Radio-Listen	18.919	19.141	1.17%

TABLE I. IMPACT OF TEMPERATURE ON TELOSB CURRENT DRAW.

To verify whether this behavior is shared by other types of devices, we consider the power profile of an Android-based smartphone: the HTC HD2. In parallel to running `CF-Bench` benchmark suite, we exercise an artificial change in temperature on the device by isolating it and putting it under a heat and refrigeration sources, and compare them to the same workload running on the ambient temperature. Table II shows that the temperature have almost no influence over the energy consumption of a given workload, with a maximum of 3% variation. On the Atom LEAP platform, the variation in energy across temperature is up to 7% as shown in Figure 4.

Overall, we conclude that mote-class and larger sensor network platforms show measurable but small change across temperature. Currently, the most common practice in energy estimation is using the same model with the same set of parameters over time, and our results justify this practice.

C. Variation Across Nodes

We now study the extent of variation in energy footprint across the 240 Quanto Testbed motes under well-known and fixed workloads. Figure 5 shows the distribution of average current of each mote for three different snapshots. Each snapshot has 240 data points corresponding to the average current draw of each mote with the radio turned on for four seconds without transmitting or receiving messages. The snapshots show a maximum difference of 15%.

Figure 6 plots the distribution of current measurements from 43 hours of experiments for different workloads but normalized with the median current for each workload. Radio-on workload shows up to 15% variation across the nodes validating that the distribution show in snapshots in Figure 5 are consistent over time. With Idle and CPU-intensive workloads, the variations are much larger but may not be as important

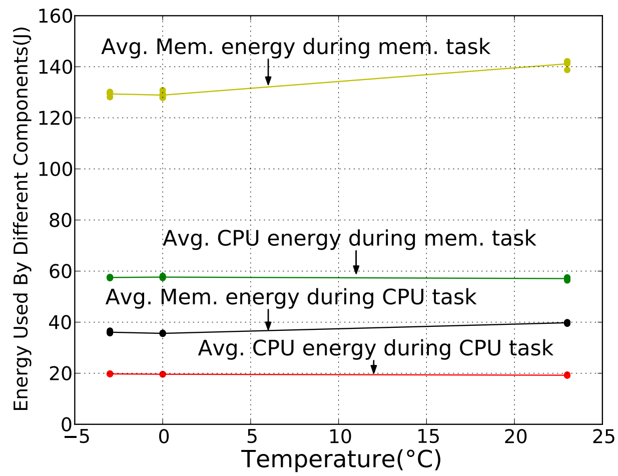


Fig. 4. Impact of temperature change on Power draw of LEAP.

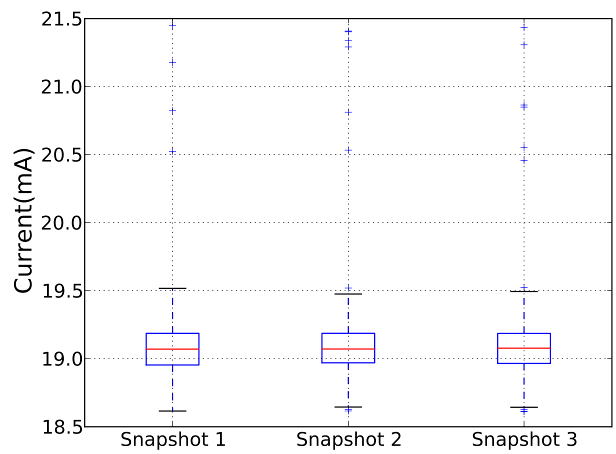


Fig. 5. Snapshots of current draw across 240 Quanto motes with radio on.

in energy budgeting because the absolute numbers are much smaller.

Next, we study the property of the distribution of current measurements across the nodes with fixed workload. In Fig-

	CPU ΔE	Memory ΔE	Disk ΔE	Total ΔE
Hot (48°C)	0.03	0.04	0.01	0.03
Ambient (34°C)	0.007	0.071	0.05	0.03
Cold (24°C)	0.10	0.06	0.08	0.08

TABLE II. ENERGY DIFFERENCE OF THE CF-BENCH WORKLOAD UNDER DIFFERENT TEMPERATURES FOR TWO GALAXY NEXUS SMARTPHONES. ΔE IS THE ENERGY-DIFFERENCE RATIO BETWEEN DEVICES.

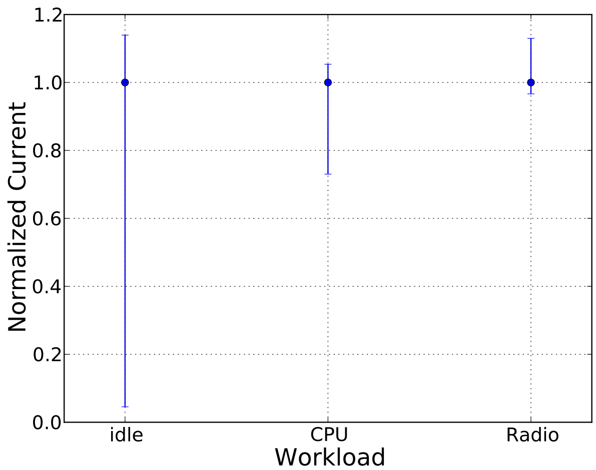


Fig. 6. Normalized current range across 240 motes and 43 hours.

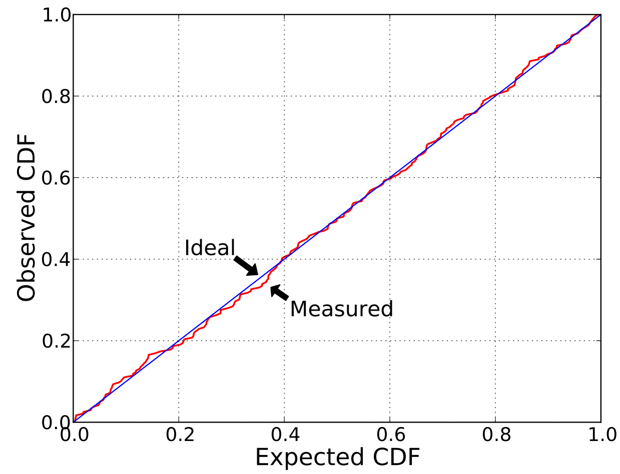


Fig. 8. Actual current distribution of 240 motes vs. Ideal Gaussian distribution

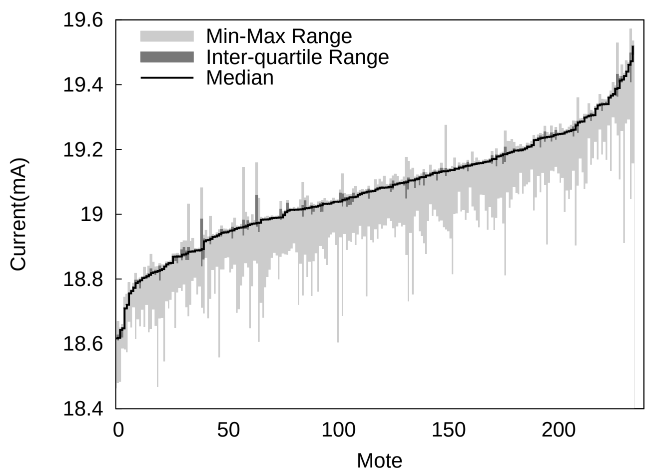


Fig. 7. Current distribution across 240 motes sorted by median

ure 7, we plot the box plot of current measurements with nodes sorted by the median current draw. The distribution of the current measurements resembles a Gaussian distribution. In figure 8, we use a Q-Q plot to visually compare this distribution with an ideal Gaussian distribution (μ=19.0687, σ=0.1698).

We conclude that the current draw across the nodes can be up to 15% and follows a distribution that approximates a Gaussian distribution.

To find out if the current draw variations across nodes also exist on TelosB motes, we perform current measurements on 6 TelosB motes with CPU-intensive, Memory-intensive, and Radio-on workloads. We found a maximum of 2.1%, 4.6%, and 4.3% variation across the three workloads respectively at -12°C. The variations of similar range at 24°Cand 44°C.

We also investigate the energy variation across smartphones. Once again, we run the `CF-Bench` on two Galaxy Nexus phones under the same temperature conditions and workloads. Result shows that for the same workload and same temperatures, the power profile between the two devices can vary up to 10%.

Through these measurements across the nodes on Quanto Testbed, TelosB motes, and smartphones, we establish that current draw variation across the nodes may be significant.

IV. EVALUATION OF METERING APPROACHES

For both direct measurement and for model calibration, one can measure at different intervals (including only once, offline), and at different subsets of nodes, ranging from one to all nodes. Given the variations observed in the previous section, we now evaluate the impact that different energy metering approaches have on estimation error.

A. *Metering at different time intervals*

Our first question is how does the interval at which one takes energy measurements affect the accuracy of the measurements. Since there is a cost to performing too frequent measurements, we would like to increase this interval, or establish that a single measurement before deployment should suffice (in which case the cost would be 0). We evaluate the error in estimating the energy used by the node when

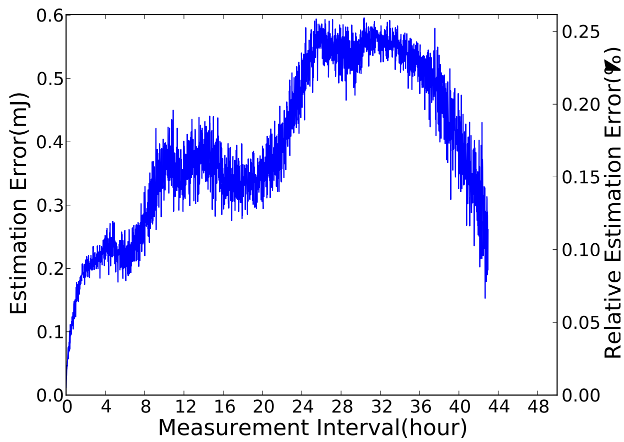


Fig. 9. Estimation error at different measurement intervals.

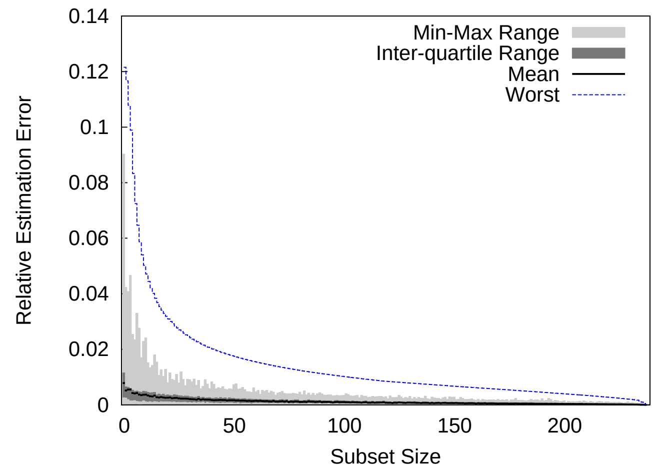


Fig. 10. Relative estimation error of average energy consumption vs. Sample set size

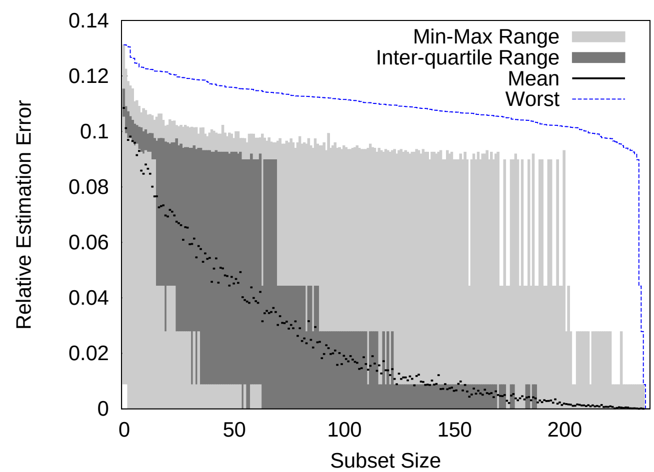


Fig. 11. Relative estimation error of biggest energy consumption vs. Sample set size

we take hardware measurements at different intervals and use software interpolation to fill in the gaps between the hardware measurements.

In Figure 9, we plot the energy estimation error for a sample mote in our testbed, running a simple workload of radio on, for different measurement intervals. This evaluation is based on the data from the previous section, with ground-truth data collected every 40 seconds, which we selectively disregard. To avoid biases due to possible synchronization of the interval and the energy curve, we randomized interval calculation with an uniform jitter of plus or minus 50%. Each interval in the x axis is thus the expected interval, and we did 200 repetions of the calculations for each expected interval.

The small intervals have smallest errors. This makes intuitive sense because we are relying on calibrated hardware reading for most of the estimate. With larger intervals between hardware measurements, the errors generally increase but also show large swings. The swings are likely due to the interaction between the measurement interval and periodicity in the current draw of the mote. Most importantly, however, is that even the maximum error is very small: in the worst interval it is less than 0.6mJ, out of 230.4mJ for the entire experiment, or less than 0.3%. Other nodes were consistent. The implication is that, at least for this platform, establishing the power draw at different situations before deployment is probably going to satisfy error tolerances for the deployment. Next we examine if we can say the same across nodes: can we measure on only one node and extrapolate to the rest?

B. Metering a subset of nodes

Due to cost, form factor and other considerations, it might sometimes be feasible to equip only a subset of nodes with energy metering hardware. We can perform measurements on this subset and use that information to extrapolate to the whole network.

Two aspects of this approach require careful consideration. First, what is the right subset size. Presumably, smaller subset will be a poorer indicator of network-wide energy footprint. Second, the type of extrapolation.

Figure 10 shows the estimation error resulting from estimating the average energy consumption for the whole network using the average of the subset selected for metering. As in the previous analysis, the nodes were running the simple radio on workload. In the best case, even with a small number of metered nodes, the error is close to 0 as shown by the *Best selection* line. In the worst case – if we pick the nodes whose average is the farthest from the network-wide average – it is comforting to know that the error drops rapidly as we deploy more metered nodes, i.e., increase the subset size. Finally, if the meters are deployed randomly even on 20 out of 240 nodes, we are not far from the best case.

Another type of energy estimation that may be of interest to the sensor network community is estimating the worst case energy use, i.e., largest energy consumption on any node in the network. Figure 11 shows the plot of error in estimation of the biggest energy consumption in the network based on the measurements on a subset of nodes. As expected, random selection of nodes for metering causes the error to diminish with a larger subset size. Unfortunately, the worst case estimation error stays relatively flat until the subset includes almost

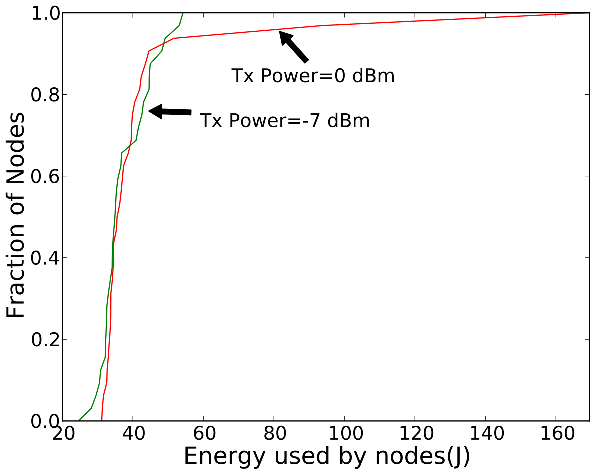


Fig. 12. CDF of energy used by 33 nodes when they run CTP.

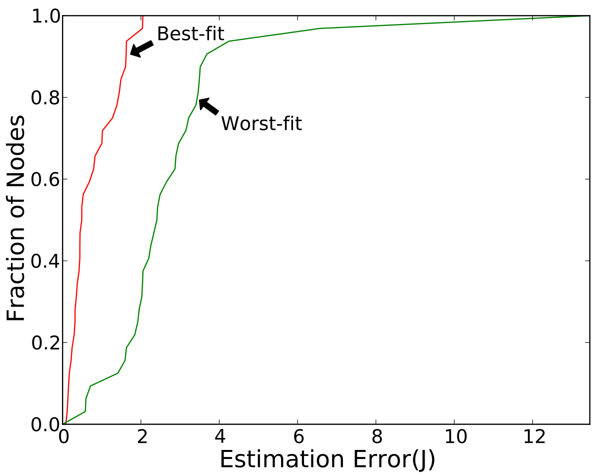


Fig. 13. CDF of estimation error with best-fit and worst-fit nodes.

all the nodes in the network.

We conclude that randomly selecting the nodes where meters are deployed and using measurements on those nodes to extrapolate is a reasonable strategy. However, that does not guarantee the worst case estimation error to be small unless the number of metered nodes is large.

C. Model fitting with subsets

In previous analysis we discuss the energy consumption estimation from a subset of nodes when running simple workload. Now we will look at a more complex, realistic workload, and use a simple, but effective linear energy usage model based on the radio duty cycle.

We ran the Collection Tree Protocol (CTP) [15] on a 33-node subset of the testbed. CTP serves as an anycast protocol that provides a best-effort, multihop delivery of packets to the root(s) of a collection tree. With CTP, nodes' energy usage depends on many more factors, such as the radio transmit power, the imposed workload, the retransmission policy, the network topology, external interference, and the MAC layer in use and its parameters.

We used the `TestCollection` application, in which all nodes send packets to the root at the same average rate, properly jittered to avoid synchronization. We set up a single root in one of the corners of the network, and ran each test for 1 hour. We varied the transmission power of the radio and Low-Power Listening (LPL) [23] settings in TinyOS 2.x. When necessary, we also varied the aggregate load to avoid strong congestion in the medium. In these experiments, the nodes used channel 31 for transmission and a 1000-ms duty cycle for LPL. We logged radio duty-cycle and calibrated readings of energy used by the nodes.

In Figure 12, we show the distribution of energy used by the nodes. We find that there is a number of nodes that spend a lot more energy than others. This can happen, for example, when a node near the root has to forward a large number of packets. It also happened to a particular node whose transmissions were not heard by other nodes, which generated a lot of retransmissions.

Our energy model is a simple linear regression based on the radio duty cycle plus a constant, background power draw. This model fit the data surprisingly well across nodes, with $r = 99.85\%$. In other words, we can say that in this particular scenario, applying the radio-duty-cycle data on the linear model is sufficient for accurate prediction of energy consumed by each node.

How can we build a model to accurately predict energy used as a function of duty-cycle of the radio or other components when only a subset of the nodes might seed the model? We start our study by first building a separate radio duty-cycle vs energy model by collecting data from each node. One of these models will be the worst-fit model in the sense that using it to predict network energy would result in the largest error. The opposite the best-fit model. Figure 13 shows the distribution of error across the nodes when we use these single-node based best-fit and worst-fit models to predict the energy used in the network. With worst-fit model, the error is always larger than the prediction with the best-fit model as expected. Also, we find that about 10% of the nodes have a much larger error.

Can we make the models more accurate by increasing the number of nodes that contribute the model? To answer this question, we build the duty-cycle vs energy model based on data contributed by different number of nodes. Then we use this model to predict the energy for the whole network and compute the estimation error. For each subset size, we iterate 10,000 times, picking random sets of nodes, using their information to build the model and computing the error. Figure 14 shows how the error decreases as we use more nodes to build the model.

Thus, we show that an application scenario such as collection of sensor data can cause different nodes to spend different amount of energy. Due to variation among nodes, energy model from single node could have big errors. Fortunately, even in that case, we can improve accuracy of estimation for the energy used by the whole network by deploying meters on more nodes. In fact, usually a small subset(in our experiments 15%) could generate accurate enough model (in our experiments

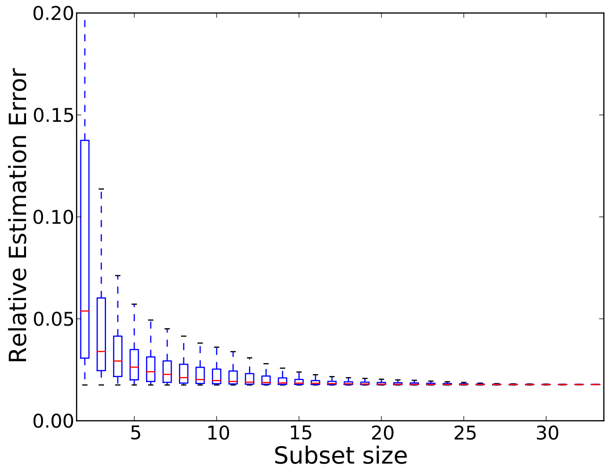


Fig. 14. Relative estimation error of 33-node WSN vs. Sample set size

$\leqslant 2.5\%$). This estimation is based on fitted models rather than direct extrapolation as shown earlier for simple workloads.

V. Discussions

Through extrapolations from calibrated measurements on a subset of nodes, we showed that we can predict energy consumption with reasonable accuracy for the whole network. However, there are still reasons for which one might want per-node energy metering. For example, in more complex platforms with multicore chips, multiple or variable peripherals or sensors, or opaque subsystems which are hard to model, deployable direct measurement can be easier than the development of complex, deployment-specific models [22]. Having meters in many nodes can help explore differences in the usage of nodes or overcome limitations of the meters themselves, as is explored in the crowdsourced energy profiler in Carat [5].

In our sudy, we ignore factors other than temperature that change over time that impacts the power draw of the device. Humidity, for example, is known to cause change in leakage current depending on the material used in the circuits. Existence of these factors make the relationship between the deployment environment and the power draw of the device complex even under simple workloads.

VI. Conclusions

In this paper, we study the problem of accurately estimating the energy used by nodes in a wireless sensor network. We study the effectiveness of performing frequent energy measurements and equipping nodes with hardware energy meters in reducing the error in estimation of network-wide energy consumption over the long term. We find that there is little variation in energy use over time with simple and fixed workloads. We find that variation in energy across the nodes is more significant and approximates a Gaussian distribution. Lastly, we found that both for measurement-based and model-based approaches, we can mitigate the differences across nodes by measuring, or calibrating the models, in small subsets of the nodes.

References

[1] J. Andersen and M. T. Hansen. Energy bucket: A tool for power profiling and debugging of sensor nodes. In *Proceedings of the 2009 Third International Conference on Sensor Technologies and Applications*, pages 132–138, 2009.

[2] F. Bellosa. The benefits of event-driven energy accounting in power-sensitive systems. In *Proceedings of the 9th ACM SIGOPS European Workshop*, 2000.

[3] W. Bircher and L. John. Complete system power estimation: A trickle-down approach based on performance events. In *Proceedings of IEEE ISPASS*, pages 158–168, 2007.

[4] L. S. Brakmo, D. A. Wallach, and M. A. Viredaz. musleep: a technique for reducing energy consumption in handheld devices. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 12–22, 2004.

[5] Carat. http://carat.cs.berkeley.edu/.

[6] T. L. Cignetti, K. Komarov, and C. S. Ellis. Energy estimation tools for the Palm. In *Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems (MSWiM)*, pages 96–103, 2000.

[7] G. Contreras and M. Martonosi. Power prediction for Intel XScale processors using performance monitoring unit events. In *Proceedings of ISLPED'05*, pages 221–226, 2005.

[8] A. Dunkels, F. Österlind, N. Tsiftes, and Z. He. Software-based on-line energy estimation for sensor nodes. In *Proceedings of Emnets IV*, 2007.

[9] P. Dutta. Epic: An open mote platform for application-driven design. http://www.cs.berkeley.edu/~prabal/projects/epic.

[10] P. Dutta, M. Feldmeier, J. Paradiso, and D. Culler. Energy metering for free: Augmenting switching regulators for real-time monitoring. In *Proceedings of the Seventh International Conference on Information Processing in Sensor Networks (IPSN'08) Track on Sensor Platforms, Tools, and Design Methods (SPOTS '08)*, Apr, 2008.

[11] P. Dutta, J. Taneja, J. Jeong, X. Jioan, and D. Culler. A building block approach to sensornet systems. In *Proceedings of the Sixth ACM Conference on Embedded Networked Sensor Systems (SenSys'08)*, November 2008.

[12] D. Economou, S. Rivoire, C. Kozyrakis, and P. Ranganathan. Full-system power analysis and modeling for server environments. In *Proceedings of Workshop on Modeling, Benchmarking, and Simulation (MoBS)*, 2006.

[13] J. Flinn and M. Satyanarayanan. Powerscope: A tool for profiling the energy usage of mobile applications. In *Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*, WMCSA '99, pages 2–, Washington, DC, USA, 1999. IEEE Computer Society.

[14] R. Fonseca, P. Dutta, P. Levis, and I. Stoica. Quanto: Tracking energy in networked embedded systems. In *Proceedings of OSDI*, pages 323–338, 2008.

[15] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection tree protocol. In *Proc. of the 7th ACM Conf. on Embedded Networked Sensor Systems (SenSys)*, 2009.

[16] V. Handziski, A. Köpke, A. Willig, and A. Wolisz. TWIST: a scalable and reconfigurable testbed for wireless indoor experiments with sensor networks. In *Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, page 70, 2006.

[17] C. Inc. Cc2420 data sheet. http://www.chipcon.com/files/CC2420_Data_Sheet_1_0.pdf, 2003.

[18] X. Jiang, P. Dutta, D. Culler, and I. Stoica. Micro power meter for energy monitoring of wireless sensor networks at scale. In *Proceedings of the 6th international conference on Information processing in sensor networks*, pages 186–195, 2007.

[19] S. Kellner, M. Pink, D. Meier, and E.-O. Blass. Towards a realistic energy model for wireless sensor networks. In *Proceedings of WONS*, 2008.

[20] A. Kopke and A. Wolisz. Measuring the node energy consumption in USB based WSN testbeds. In *Proceedings of the 2008 The 28th International Conference on Distributed Computing Systems Workshops*, pages 333–338, 2008.

[21] M. Martins, R. Fonseca, T. Schmid, and P. Dutta. Network-wide

energy profiling of ctp. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 439–440, 2010.

[22] J. C. McCullough, Y. Agarwal, J. Chandrashekar, S. Kuppuswamy, A. C. Snoeren, and R. K. Gupta. Evaluating the effectiveness of model-based power characterization. In *Proceedings of the USENIX ATC*, 2011.

[23] D. Moss and P. Levis. BoX-MACs: Exploiting Physical and Link Layer Boundaries in Low-Power Networking. *Stanford Information Networks Group Technical Report SING-08-00*, 2008.

[24] 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 EuroSys'11*, pages 153–168, 2011.

[25] T. Pering, Y. Agarwal, R. Gupta, and R. Want. Coolspots: reducing the power consumption of wireless mobile devices with multiple radio interfaces. In *Proceedings of the 4th international conference on Mobile systems, applications and services*, pages 220–232, 2006.

[26] J. Polastre, R. Szewczyk, and D. Culler. Telos: enabling ultra-low power wireless research. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, IPSN '05, Piscataway, NJ, USA, 2005. IEEE Press.

[27] V. Raghunathan, T. Pering, R. Want, A. Nguyen, and P. Jensen. Experience with a low power wireless mobile computing platform. In *Proceedings of the 2004 International Symposium on Low Power Electronics and Design*, pages 363–368, 2004.

[28] V. Shnayder, M. Hempstead, B.-r. Chen, G. W. Allen, and M. Welsh. Simulating the power consumption of large-scale sensor network applications. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 188–200, 2004.

[29] D. Singh and W. J. Kaiser. The atom leap platform for energy-efficient embedded computing. In *Actuated, Sensing, Coordinated and Embedded Networked Technologies (ASCENT)*, Dec 2010.

[30] T. Stathopoulos, D. McIntire, and W. J. Kaiser. The energy endoscope: Real-time detailed energy accounting for wireless sensor nodes. In *Proceedings of the 7th international conference on Information processing in sensor networks*, pages 383–394, 2008.

[31] M. A. Viredaz and D. A. Wallach. Power evaluation of a handheld computer. *IEEE Micro*, Jan-Feb, 2003.

[32] G. Werner-Allen, P. Swieskowski, and M. Welsh. MoteLab: A wireless sensor network testbed. In *Proceedings of the Fourth International Conference on Information Processing in Sensor Networks (IPSN'05), Special Track on Platform Tools and Design Methods for Network Embedded Sensors (SPOTS)*, April 2005.

[33] Y. Xiao, R. Bhaumik, Z. Yang, M. Siekkinen, P. Savolainen, and A. Yla-Jaaski. A system-level model for runtime power estimation on mobile devices. In *Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*, pages 27–34, 2010.

[34] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. Mao, and L. Yang. Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In *Proc. Int. Conf. Hardware/Software Codesign and System Synthesis*, Oct 2010.

[35] 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*, CODES/ISSS '10, pages 105–114, New York, NY, USA, 2010. ACM.