On the Accuracy of Software-based Energy Estimation Techniques

Philipp Hurni[1], Benjamin Nyffenegger[1], Torsten Braun[1], Anton Hergenroeder[2]

[1] Institute of Computer Science and Applied Mathematics (IAM), University of Bern
hurni, nyffeneg, braun@iam.unibe.ch
[2] Institute of Telematics (ITM), Karlsruhe Institute of Technology (KIT)
hergenroeder@kit.edu

Abstract. This paper examines the accuracy of software-based on-line energy estimation techniques. It evaluates today's most widespread energy estimation model in order to investigate whether the current methodology of pure software-based energy estimation running on a sensor node itself can indeed reliably and accurately determine its energy consumption - independent of the particular node instance, the traffic load the node is exposed to, or the MAC protocol the node is running. The paper enhances today's widely used energy estimation model by integrating radio transceiver switches into the model, and proposes a methodology to find the optimal estimation model parameters. It proves by statistical validation with experimental data that the proposed model enhancement and parameter calibration methodology significantly increases the estimation accuracy.

1 Introduction

With energy efficiency being a major concern in the design of Wireless Sensor Networks (WSNs), researchers have thoroughly investigated how to save energy by intelligent design of the communication protocols. On the MAC level, Energy-Efficient Medium Access Control (E^2-MAC) protocols have been proposed to minimize the energy wastage of the radio transceiver, which is typically the major energy consumer of the node onboard components. Most simulation-based E^2-MAC protocol studies rely upon simple energy models of the wireless transceiver chips, with the node's energy consumption being computed as the sum of the energy it spends in the different transceiver states. Most of todays' simulation models implemented in mainstream network simulator frameworks (e.g. ns-2, OMNeT++) distinguish three or even four states (*receive/idle*, *transmit*, *sleep*), as well as switching states with corresponding transition delays.

With research on WSNs becoming more mature, many E^2-MAC protocols have also been prototyped and evaluated on real sensor hardware testbeds. Not surprisingly, experimental validation of E^2-MAC protocols have proven to be much more resource intensive than using mainstream network simulators. While commonly used networking metrics such as packet delivery rate, source-to-sink latencies or maximum throughput can easily be determined in real-world testbeds, measuring the power consumption of sensor nodes is much harder: costly high-resolution digital multimeters or cathode-ray oscilloscopes need to be hooked to the nodes in order to sample the varying low currents and voltages.

Researchers have henceforth ported the same simple state-based energy estimation models of WSN simulators into their real-world sensor MAC protocols or radio chip drivers. Software-based energy estimation has been proposed in [1] as a viable alternative to using costly hardware-based energy measurement equipment, and has been integrated into the Contiki OS [2] - one of todays' most widespread sensor node operating systems. The Contiki mechanism consists in bookkeeping the time the radio resides in the different transceiver modes on the node itself, and multiplying these times with previously determined power levels to obtain rough estimates for the consumed energy. Many prominent E^2-MAC protocol studies (e.g. [3] [4]) have entirely relied their experimental research results upon the same software-based approach for estimating the energy consumption of their protocol prototypes. More and more recent research papers have utilized exactly this approach (e.g. [5], [6]), although, as already pointed out in [1], no existing study has yet validated the accuracy of this approach with physical hardware-based energy measurements. This paper bridges this missing gap and thoroughly examines the accuracy and the limits of software-based energy estimation on the MSB430 sensor nodes platform [7]. It evaluates several energy estimation models with prototype implementations of 802.11-like CSMA and three E^2-MAC protocols (S-MAC [3], T-MAC [8], WiseMAC [9]). We ran a plethora of experiments under different traffic load levels and with different node instances, in order to statistically describe the achieved estimation accuracies.

The paper is organized as follows: we elaborate on related work on software-based energy estimation and measurement in Section 2. In Section 3 we introduce the experiment setup for evaluating the different energy estimation models, model enhancements and calibration techniques. Section 4 discusses the observed deviations between different sensor nodes' current draws and their effect on the resulting estimation accuracy. Section 5 evaluates the maximum achievable accuracy of the most widely used energy estimation model (henceforth referred-to as the *Three States Model*) with various wireless channel MAC protocols and traffic rates. We then refine and enhance the estimation model and calibration methodology and experimentally validate the gain in accuracy. Section 6 discusses the maximum accuracy gain that can be achieved with sophisticated and fine-grained parameter calibration. Section 7 concludes the paper.

2 Related Work: Hardware-based Energy Measurement vs. Software-based Energy Estimation

In numerous E^2-MAC protocol studies [4] [10], cathode-ray oscilloscopes have been used to quantify the energy consumed by a sensor node in a real-world experiment. The basic idea of the methodology is to connect the sensor node in series with a low-impedance shunt resistor and to measure the resistive voltage drop across the shunt, in order to infer the current flowing through the circuit.

This methodology has been applied by a number of studies and can be seen as the *cleanest* approach of energy measurement, as it does not incur any *side-effects* to the sensor node hardware or software. Its main drawback, however, is the costly measurement equipment required and the time-consuming operation of it. Furthermore, if current traces need to be stored in a reasonable resolution during an experiment of several minutes or even hours, the collected raw current traces become huge and quickly cause storage- and memory problems. Only few testbeds have integrated support for distributed real-time energy-measurements, as e.g. MoteLab [11] with some of its nodes, or PowerBench [12]. Hence, in most studies on energy-efficiency issues on the MAC and/or routing layer, researchers have only measured a node's current over a short period of time in order to calibrate a simulation and/or estimation model, and have omitted the energy aspect for the rest of the empirical evaluation.

Sensor Node Management Devices (SNMD) [13] have been developed as a cost-effective alternative to using high-frequency multimeters or oscilloscopes for *side-effect free* high-resolution energy measurement of sensor nodes. SNMDs continuously measure the sensor node current and voltage with resolutions of up to 2 kHz, and therefore need to be connected via USB to a backbone network. This is usually possible in wired stationary testbeds and lab environments, but less in outdoor deployments. With costs of the circuitry components still in the range of 300$, it is a convenient measurement tool for lab environments, but still too costly for large deployments of WSNs or WSN testbeds.

[1] motivates the need for software-based *on-line* energy estimation, because only *on-line* estimation mechanisms running on the node itself enable the node to take energy-aware decisions about routing, clustering or transmission power scheduling. The authors experimentally correlate the estimated energy with the sensor nodes lifetime, however underline that "further study is needed to accurately quantify the error rate of the mechanism".

PowerBench [12] partly tackles the issue of the accuracy of software-based energy estimation. The authors elaborate on the difference between their software-based energy estimations (calculated with the commonly used *Three States Model*) and the physically measured energy consumption of the nodes. When running B-MAC [4] and Crankshaft [14], this difference reaches up to 21% of the measurement values. *Per-node-calibration* is shown to vastly reduce this estimation error. With the deviations between software-based estimation and physical measurements still ranging from 2% to almost 14% for some of the examined E^2-MAC protocols, the software-based estimation approach still leaves room for further improvements. The authors further note that frequency of state transitions have a significant impact on the estimation accuracy.

Software-based energy estimation techniques clearly have their advantages and drawbacks. A purely software-based approach can only deliver estimates. It further introduces inherent *side-effects*, as the estimation mechanism itself causes computational costs, which are hard to account for. The advantages, however, are manifold: with an energy estimation being present on the node at run-time, many power-aware WSN algorithms can be applied in real-world deployments.

With the WSN field moving from simulation-based towards real-world testbed-based research, finding a simple and painless, but yet accurate methodology for quick and reliable energy estimation can be a significant milestone.

3 Experiment Equipment and Setup

3.1 Sensor Network Management Devices (SNMD)

We used Sensor Node Management Devices (SNMD) [15] to measure and retrieve the node's current and voltage in high resolution, in order to be able to calculate the *physically measured* energy consumption and compare it to software-based estimations later on. SNMDs have been specifically designed to accurately measure current and voltage of sensor nodes with a sampling resolution of up to 20 kHz (up to 500 kHz buffered). SNMDs measure the resistive voltage drop across a 1 Ω shunt resistor. The accuracy of the SNMD has been evaluated using high-precision laboratory equipment for different current ranges. The SNMD firmware corrects each sampled measurement by an error term, which was obtained during evaluative testing in advance. This has been shown to reduce the measurement error introduced by the measurement circuit below $\pm$ 0.5% for any current in the range of 0-100 mA in [13]. As the accuracy of the SNMD has been calibrated using highly accurate state-of-the art measurement equipment, we can safely assume that it provides best possible physical hardware-based energy measurements. Throughout the experimental analysis of this paper, we decided to stick to a sampling rate of 1000 Hz, as the accuracy gain with even higher rates proved to be negligible with the chosen node type and bandwidth settings. Other node platforms, e.g. nodes with IEEE 802.15.4-based radios with higher bandwidth could however probably profit from the high maximum sampling rate of the SNMD.

3.2 The Modular Sensor Boards (MSB430) Platform

The MSB430 node [7] has a CC1020 [16] byte-level radio transceiver operating in the 804-940 MHz ISM frequency band. In its base configuration, the node features a Sensirion SHT11 temperature and humidity sensor, as well as the Freescale MMA7260Q accelerometer. Besides ScatterWeb^2 OS [17], the MSB430 can be run with the popular Contiki OS [2] since recently (v.2.4). While the maximum raw bit rate of the CC1020 is 153.6 kbit/s, the ScatterWeb^2 OS we utilized throughout this paper currently only supports a data rate of 19.2 kbit/s.

3.3 Experiment Setup

We kept the measurement setup as simple as possible, in order to be able to repeatedly perform a significant number of experiment runs with different wireless channel protocols and traffic rates on the same experiment setup. We lay out nodes A, B, C with a distance of 30cm on a table, as depicted in Fig. 1. As we

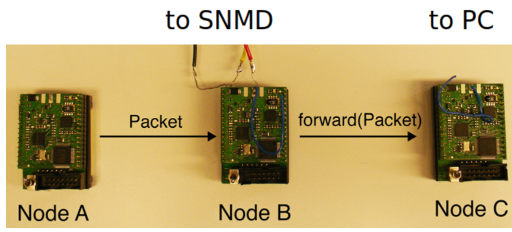

Fig. 1: Node A generating packets, node B hooked to SNMD

wanted to simultaneously obtain both the software-based estimations and the unaffected physical hardware-based measurements of the same node B, we had to keep node B unplugged from any serial interface, as the node would otherwise draw some small current from the powered USB serial interface cable. Hence, in order to obtain the software-based estimations of node B without accessing it over a serial cable, we let node B write its energy estimation model data (time in transmit mode, time in receive mode, etc.) into the packet payload.

Packets are 50 bytes each (10 bytes header, 40 bytes payload). In each experiment run, node A starts sending constant-rate traffic of rate r towards node B during $T_{exp} = 600s$. Right after the reception of the first packet, Node B starts keeping track of the time its transceiver resides in the different states. After injecting its estimation model data into the packet, Node B forwards the packets to node C, which decapsulates the packet and logs node B's energy estimation data to the serial interface, which is connected to a Desktop PC. During the entire experiment, the current trace of node B is read from the SNMD's serial interface, which is connected to the same Desktop PC. As discussed later in the analysis, we varied the traffic rate r at node A from very low rates (1 packet every 100s) to high rates (2 packets/s) with each different wireless channel MAC protocol. We measured 10 independent runs for each setting, and evaluated different node instances. In Section 4, node B (the *measurement node*) was exchanged with other node instances of the same type.

4 Hardware-dependent Energy Consumption Deviations

Applying software-based energy *estimation* inevitably introduces *inaccuracies*. The differences between the *estimated* power consumption and the *physically measured* power consumption can generally be explained by the slightly differing behavior of the nodes' electronic hardware components, or may stem from the inherent imperfection of the software-based model and the applied estimation methodology. This section elaborates on the effect of the slight deviations on the power consumption of different node instances of the same node type, whereas Section 5 discusses the impact of choosing an appropriate estimation model and calibrating the model parameters.

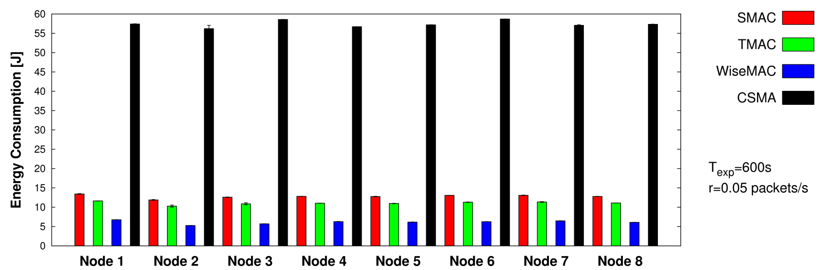

Fig. 2: Energy consumed by 8 different instances of nodes

4.1 Different Current Draws with Different Nodes

As discovered in previous experimental studies [18] [12], the power consumption of different instances of the same type of sensor node often varies in the range of some few percent. [18] presumes that this variation stems from differences in the electronic components tolerances. We hence first examined multiple instances of MSB430 nodes running different wireless MAC protocols, given a constant traffic rate of 1 packet each 20s over $T_{exp} = 600s$. With this evaluation, we quantify the estimation inaccuracies caused by the variation in the energy consumption of different instances of the same node type - in our case the MSB430 platform.

Figure 2 depicts the energy consumed by eight different instances of MSB430 nodes and the four examined protocols during 10 experiment runs. Each bar depicts the mean value and standard deviation measured during 10 independent runs - the latter was low in most cases and is hence barely visible. The energy consumption obviously varies heavily from protocol to protocol (eg. WiseMAC vs. CSMA). The variation from node to node however is also clearly visible, e.g. the energy consumed by node 6 running CSMA is roughly 4% higher than that of node 2. We investigated the reason for these differences in the current traces and found that indeed, the current drawn from different nodes can vary to a certain degree, and that the variation can even differ for each of the different transceiver states. Figure 3 depicts the current traces of nodes 1 and 2 running CSMA and receiving a data packet, and sending it further to another node. As one can clearly see, node 1 draws approx. 2 mA less than node 2 when transmitting. Although the transmission power settings were set identically for

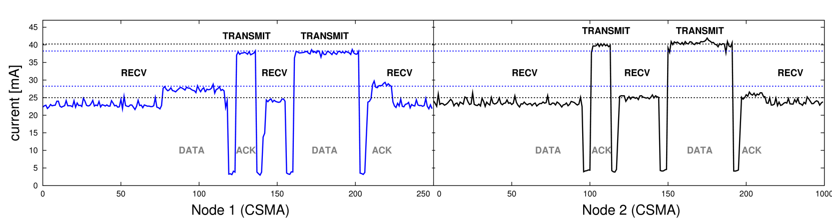

Fig. 3: Current draw of nodes B and C

all nodes, the current levels in the transmit state obviously varied to a certain degree. A further anomaly we encountered is that some nodes drew more current in receive mode when actually receiving data compared to listening to an idle channel, whereas in most cases, no significant difference between these two cases could be measured. This effect is visible in Figure 3 as well: node 1 consumes approximately 3 mA more when receiving data, compared to node 2 which consumes more or less the same current when receiving data or listening to an idle channel. As both nodes are running the same interrupt service routine code and did not run any other computationally intensive tasks during this time, the CPU can neither be held accountable for this effect. We further discovered slight differences in the peak energy consumption as well as in the duration of transceiver switches depending on the protocol, and even depending on the traffic load. We presume that these differences stem from the inaccuracies in the production of the electronic components. Fast switching between the different operation modes of the radio could probably also have a temporary impact on the behavior of active circuit elements. Although the temperature is known to impact on the power consumption of electronic devices, we can safely exclude this as an explanation for the discovered deviations, as all experiments were run under room temperature in the same laboratory environment.

4.2 Statistical Characterization of Node Deviations

In an attempt to quantify the discovered differences between the eight measured node instances, we a) determined the mean and standard deviation of the measured energy consumptions of all measurement runs of all eight nodes for the CSMA protocol in the experiment described in 3.3 (with $T_{exp} = 600s$ and a traffic rate r of 0.05 packets/s), and b) compared each pair of nodes to determine the the maximally differing nodes. We chose CSMA because at examined traffic rates, no packet loss occurred within all CSMA runs. Hence, the CSMA experiment runs were most suited for examining the per-node differences.

a) The mean consumed energy of the eight different nodes throughout T_{exp} was 57.55 Joules with a standard deviation of 1.54%. Hence, roughly two thirds of all node instances exhibit a value in-between 57.55 Joules $\pm$ 1.54%, given that the variation between different nodes follows normal distribution. We conjectured that the latter is the case, as Jarque-Bera's test on the normality of the measurement variation (JB-value: 0.701) could not be rejected (cf. [19]).

b) The maximum deviation between the mean measured energy consumption of the two maximally differing nodes was determined to be 4.24% (of the respective higher value). We tested the claim that these two nodes do actually differ significantly from each other, i.e. that the discovered deviations are not caused by coincidence or the limited set of observations. We found that the null-hypothesis of a two-sided t-test claiming that the two nodes exhibit the same mean energy consumption (=on average consume the same amount of energy) could safely be rejected at the 95% confidence level. This however was not the case for all the node pairs, as some groups of nodes obviously exhibit similar patterns in their energy consumption (cf. Figure 2).

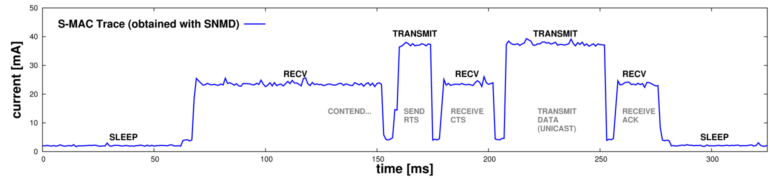

Fig. 4: Current Draw of node B

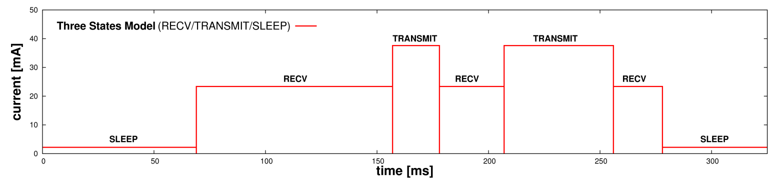

Fig. 5: Node B as modeled by the *Three States Model*

5 Evaluating Software-based Energy-Estimation Models

In this section we analyze the impact of the choice of the estimation model on the resulting estimation accuracy for experiments with different traffic load levels and wireless channel protocols. With the variation between different nodes being in the range of more than 4% for the specified experiment scenario, we decided to use exactly *the same sensor node* and also the *same SNMD device* throughout the entire analysis in this section, in order not to introduce variations caused by differing measurement hardware and measured hardware.

5.1 Three States Model (recv/idle, transmit, sleep)

The most frequently used model to date for estimating a node's energy consumption - especially in E^2-MAC protocol studies - consists in modeling the latter as a function of the three states of the radio transceiver *receive/idle listening*, *transmit* and *sleep* (cf. [12] [3] [4]). We henceforth refer to this model as the *Three States Model*. The Contiki OS (v. 2.4) energy estimation mechanism models the radio's power consumption using this model, but *separately* tries to keep track of the CPU power consumption, which can vary depending on the Low-Power-Mode (LPM) it is currently operating. The ScatterWeb[2] OS used in this study puts the CPU to LPM1 as soon all events have been processed, where the node's current is approximately 1.8 mA, given that the radio is turned off. With the CPU active and the radio off, the node current is roughly 3.5 mA. As our examined E^2-MAC protocols generally do not incur intensive computations, we neglected to account for the CPU costs separately, and considered the CPU's power consumption to be *integrated* within the three states of the transceiver. Estimating the CPU power consumption in software when applying E^2-MAC protocols is anyway not easy to achieve, as most of the MAC-related CPU activity takes place in interrupt service routines. Accounting for such may even cause more costs than the protocol-related computations themselves (c.f. [18]). If the CPU activity does not vary much across state changes of the radio transceiver, modeling the CPU and radio integrally safely holds. Figure 4 illustrates that for the given E^2-MAC protocol, accounting for CPU in a combined manner with the three different power levels of the radio transceiver is sufficient.

We henceforth modeled the energy consumption of our S-MAC [3], T-MAC [8], WiseMAC [9] and CSMA implementations using the abovementioned *Three States Model*. We let the nodes keep track of the time differences between the transceiver switches, in order to determine how much time has been spent in each state. Figure 4 depicts the current draw during the active interval of an S-MAC frame containing an RTS/CTS handshake and a subsequent data packet transmission. Figure 5 illustrates how this current draw is being approximated by the *Three States Model*. The total energy consumed (denoted as E) corresponds to the area below the current draw multiplied by the supply voltage, which is assumed to be constant. Analytically, the *Three States Model* can be formulated as equation MI. The consumed energy E is calculated as the sum of the total time spent in the receive state multiplied by the respective power level $T_{rcv}P_{rcv}$, and the respective terms for the transmit and sleep states ($T_{slp}P_{slp}$ and $T_{tx}P_{tx}$). This approach is identical to the one applied in [12], [3] and [4].

$$E = P_{rcv}T_{rcv} + P_{tx}T_{tx} + P_{slp}T_{slp} = I_{rcv}V_{rcv}T_{rcv} + I_{tx}V_{tx}T_{tx} + I_{slp}V_{slp}T_{slp} \quad \text{(MI)}$$

Parameter Definition through Example Measurement: [3], [4], [10] and [12] calibrate the parameters of their energy model by measuring the currents the nodes draw in the different states, and multiplying it with the supply voltage to obtain P_{rcv}, P_{tx} and P_{slp}. They do so by using either oscilloscopes or high-precision multimeters and by measuring the current in each state over a

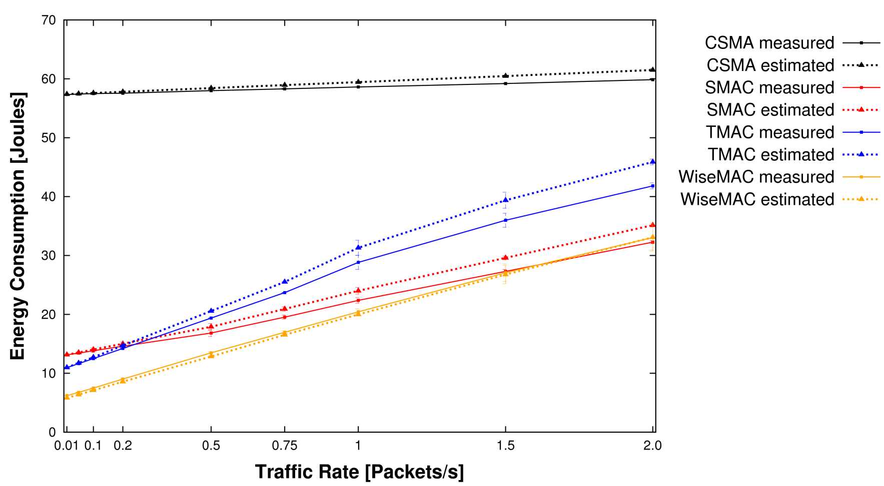

Fig. 6: Measured vs. Estimated Energy Consumption

certain timespan. In the first attempt, we pursued exactly the same approach, and determined the mean values of I_{rcv}, I_{tx}, I_{slp} by measuring each state of the *measurement* node using the SNMD for a couple of seconds. The stable mean values were determined to be 23.5353 mA, 37.4872 mA and 2.1495 mA for I_{rcv}, I_{tx}, I_{slp}, respectively. We further set the voltage according to the supply voltage of the SNMD to $V_{rcv} = V_{tx} = V_{slp} = 4.064V$.

Figure 6 depicts the mean values of the energy measurements and the estimations being computed with the *Three States Model* - using the parameters for P_{rcv}, P_{tx} P_{slp} measured in the example trace. One can clearly see that the estimations fit quite well for low traffic rates, but that the gaps between mean estimations and mean measurements become larger with higher rates of packets being sent over the *measurement* node. For most protocols - especially S-MAC and T-MAC - the energy estimation over-estimates the energy consumed by the node with increasing load. This increasing over-estimation stems from the fact that the *Three States Model* does not account for the transceiver switches. As one can clearly see comparing Figure 4 with Figure 5, the current draw decreases to roughly 4 mA when the transceiver is switched to receive or transmit - hence drawing less current than estimated with the *Three States Model*. By defining parameters through example measurement, the impact of the applied traffic load and the frequent transceiver switches as well as the particularities of the MAC protocol are not being taken into account at all. Extrapolating from a short example measurement of a node hence leads to suboptimal parameters for the *Three States Model*, even when using the same node for parameter calibration and the evaluation of the accuracy.

Parameter Definition through Ordinary Least Squares (OLS): Being able to physically measure the current draw of a sensor node *and* at the same time obtain the software-based estimation calculated by the node itself offers the opportunity to relate the estimations to the real-world measurements. Using the plethora of experimental data gained in the many experiments runs (in total over 12 GB), we reflected upon a method to determine more resilient parameters for the unknown variables P_{rcv}, P_{tx}, P_{slp} of the *Three States Model*. Ideally, the software-based energy estimation running on the node should neither rely on the particularities of a specific MAC protocol, nor on the shape or intensity of the traffic. *Ordinary Least Squares (OLS) Regression Analysis* yielded the most suitable technique to determine the unknown variables for a linear estimation model with multiple unknown variables. OLS minimizes the sum of squared errors (SSE) between estimations and observations (= the measurements). We formulated a multivariate OLS regression model with the *explanatory variables* T_{rcv}, T_{tx}, T_{slp} (the times spent in the different transceiver states, calculated at runtime), as well as the physically measured *dependent variable* E obtained using the SNMD device. The resulting estimation equation hence simply comprises equation MI and the error term ε for the residuals.

$$E = P_{rcv}T_{rcv} + P_{tx}T_{tx} + P_{slp}T_{slp} + \varepsilon \qquad \text{(OLS-I)}$$

With the above multivariate OLS model, the unknown parameters are estimated as the OLS estimator $\hat{\beta} = (\hat{P_{rcv}}, \hat{P_{tx}}, \hat{P_{slp}})$ which calculates as

$$\hat{\beta} = ((X'X)^{-1}X')y$$

where X is the matrix of all the observations of the *explanatory variables*, consisting in 3 columns (T_{rcv}, T_{tx}, T_{slp}) and a row for each measurement, and y the vector with the corresponding observations of the *dependent variable*. We assessed the coefficient of determination R^2 to measure the *goodness of fit* of the multivariate linear regression model and obtained a surprisingly high value of $R^2 = 0.9980$.

Estimation Accuracy of the Three States Model: In order to determine the accuracy of the OLS-calibrated software-based model, a cross-validation with totally new experimental data is inevitable to omit overfitting effects (cf. [19]). The determination of the parameters P_{rcv}, P_{tx}, P_{slp} using OLS regression was hence achieved on a first set of experiment runs, the so-called *training set*. The results concerning the estimation accuracy of this section however were gained with a new set of experimental data, to which we will further refer as *validation set*. We fed $\hat{\beta}$ containing the OLS estimators of the unknown variables $\hat{P_{rcv}}, \hat{P_{tx}}, \hat{P_{slp}}$ into the node's estimation model and estimated the energy consumption with the validation set. We considered the so-called *mean absolute error (MAE)* (=the average difference, cf. [19]) between the estimations and the measured values to be the best statistical measure for the *accuracy* of the employed *Three States Model*. The MAE and its standard deviations calculated across all protocols and traffic rates in the validation set (henceforth always given as percentage of the SNMD-measured values) is depicted in Figure 7. For each traffic rate, the estimation error using the OLS estimator parameters is 4.2% to 35.9% lower than the corresponding error when using the model parameters defined through example measurement. Across all measurements, the

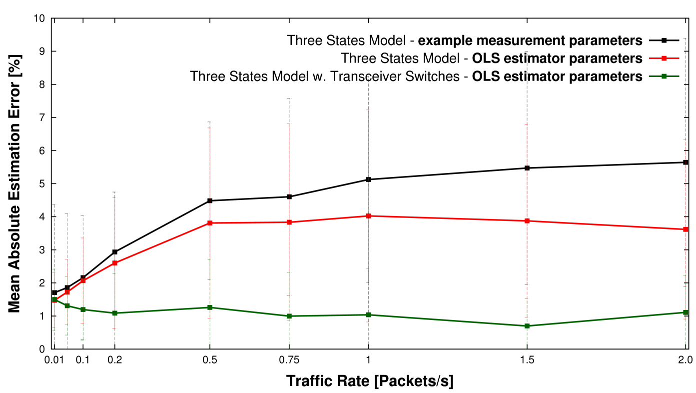

Fig. 7: Absolute Mean Estimation Error (in %) vs. Traffic Rate (packets/s)

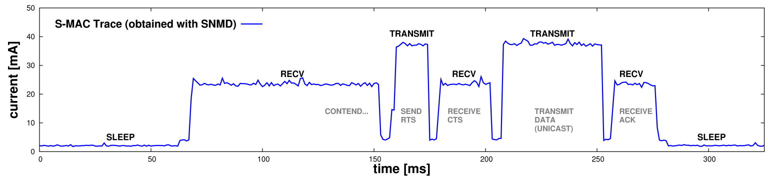

Fig. 8: Current Draw of node B

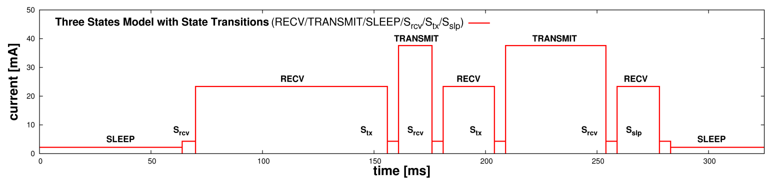

Fig. 9: Current modeled by the *Three States Model with State Transitions*

mean absolute estimation error and standard deviation (denoted as $\mu \pm \sigma$) of the *Three States Model* with the parameters defined by example measurement equals 3.77% $\pm$ 3.17%. When determining the parameters by OLS, we obtain 3.00 % $\pm$ 2.55% - hence achieving an overall MAE reduction error by 21%.

5.2 Three States Model with State Transitions

With the mean absolute estimation error still in the range of 3% or more, we investigated further means to improve the estimation accuracy. As Figure 8 exhibits, the current draw temporarily drops to approximately 4 mA during the state switches. These state switches remain unaccounted for in the OLS regression model specified in equation OLS-I. We first attempted to sum up the transition times between the transceiver states. This approach however led to unsatisfactory results, as the ScatterWeb[2] OS only supports a clock in milliseconds precision. Simply counting the transceiver switches and integrating them into the OLS regression model however led to a significant improvement in the estimation accuracy. The number of transceiver switches (*from* an arbitrary state) *to* the *receive*, *transmit* or *sleep* state was hence accounted for with the additional regressands s_{rcv}, s_{tx}, and s_{slp}. We refer to this model as the *Three States Model with State Transitions* hereafter, as specified in equation MII.

$$E = T_{rcv}P_{rcv} + T_{tx}P_{tx} + T_{slp}P_{slp} + \alpha s_{rcv} + \beta s_{tx} + \gamma s_{slp} \qquad \text{(MII)}$$

According to this enhanced model, the energy consumed by an arbitrary node is a function of the total time it has its radio transceiver in the three different states (denoted as T_{rcv}, T_{tx}, T_{slp}) and the three adjustment terms αs_{rcv}, βs_{tx}, and γs_{slp}. The parameters α, β, γ compensate for the transceiver switches to the states *receive*, *transmit* and *sleep*. Their optimal values are determined empirically using OLS regression.

Parameter Definition through Ordinary Least Squares (OLS): We specified the corresponding OLS regression model to equation MII with the *explanatory variables* T_{rcv}, T_{tx}, $T_{slp}, s_{rcv}, s_{tx}, s_{slp}$, as well as the *dependent variable* E (for which we obtain the *real measured value* using the SNMD device) as

$$E = P_{rcv}T_{rcv} + P_{tx}T_{tx} + P_{slp}T_{slp} + \alpha s_{rcv} + \beta s_{tx} + \gamma s_{slp} + \varepsilon \qquad \text{(OLS-II)}$$

The OLS estimator $\hat{\beta} = (\hat{P_{rcv}}, \hat{P_{tx}}, \hat{P_{slp}}, \hat{\alpha}, \hat{\beta}, \hat{\gamma})$ is calculated in analogy to Section 5.1. We obtained a coefficient of determination of $R^2 = 0.9998$ for the multivariate linear regression model OLS-II, a slightly higher value than for OLS-I. However, when comparing the *goodness of fit* of two regression models, the R^2 indicator is not a meaningful criterion, as it never decreases when adding more regressands. The so-called *adjusted coefficient of determination* $\bar{R}^2$ (cf. [19]) adjusts for the number of explanatory terms in a model. Unlike R^2, this coefficient only increases when the increase of explanatory variables actually improves the model. An increase of $\bar{R}^2$ upon addition of an explanatory variable to a multivariate OLS model is hence generally understood as a proof that the new model delivers a better fit to the measured data. An even better coefficient for comparing the *goodness of fit* of two regression models however is the Akaike Information Criterion (AIC) (cf. [19]). The lower the AIC value, the better the fit to the model. We measured the $\bar{R}^2$ and AIC coefficients before and after adding the transceiver switches s_{rcv}, s_{tx}, s_{slp}, to the OLS model (OLS-I vs OLS-II). With $\bar{R}^2$ increasing from $\bar{R}^2_I = 0.9801$ to $\bar{R}^2_{II} = 0.9980$, and AIC decreasing from $AIC_I = 2.5036$ to $AIC_{II} = 0.2154$, we can safely claim that the *Three States Model with State Transitions* delivers a significantly better fit to the measurement data than the today's most widely used simple *Three States Model*.

Estimation Accuracy of the Three States Model with State Transitions: We calibrated the OLS estimators for the parameters of the second model with the *training set*, and examined the resulting estimation accuracy on the *validation set*. Across all measurements, the MAE and standard deviation (denoted as $\mu \pm \sigma$) of the software-based estimations using the *Three States Model with State Transitions* (and the parameters determined by OLS) compared to the physically measured values equals 1.13% ± 1.15%. Comparing this result to the 3.00 % ± 2.55% obtained with the *Three States Model* (and the parameters determined by OLS), our proposed model enhancement led to an overall reduction of the MAE by remarkable 62.3% (cf. Figure 7).

6 The Impact of Calibration on the Estimation Accuracy

This section evaluates the impact of different possible granularities of *calibration* on the achievable accuracy of the software-based energy estimation technique. Throughout this section we henceforth utilize the same multivariate OLS regression methodology and the *Three States Model with State Transitions* as described in Section 5, as applying this model generally led to the lowest estimation errors.

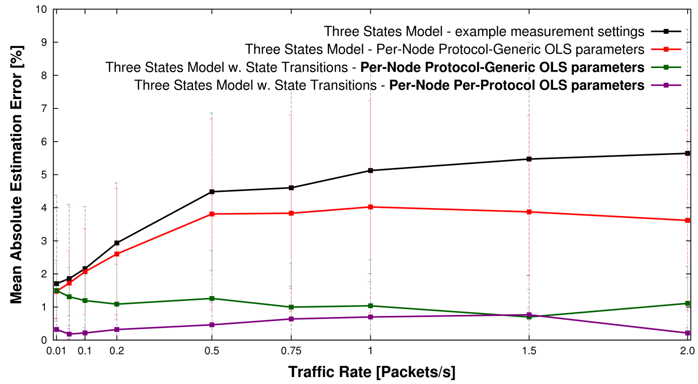

Fig. 10: Absolute Mean Estimation Error (in %) vs. Traffic Rate (packets/s)

6.1 Per-Node Calibration

Different wireless sensor node instances often exhibit a slightly different behavior with respect to their power consumption levels in the different transceiver states. This effect has been observed in previous studies [18] [12], and has been quantified for the utilized MSB430 platform in Section 4. We have encountered node pairs of the same node type that differed by more than 4% in their physically measured energy consumption. Hence, even the best *node-generic* software-based energy estimation mechanism can be more than 4%, if its underlying model parameters were not calibrated on a *per-node* basis.

Researchers intending to calibrate their energy estimation model with only one particular sensor node instance must therefore be aware that their energy consumption estimates will deviate from the *real* energy consumption by the unavoidable hardware-based variation, unless each node has previously been calibrated individually. Calibrating on a *per-node* basis however means that *every single node* needs to be physically measured (e.g. with an SNMD or a high-resolution multimeter) ideally with different MAC protocols and different traffic rates. Only this time-intensive calibration leads to the set of *per-node* but *protocol-generic* estimation model parameters which has been shown in Section 5.2 to reduce the mean absolute estimation error ($\mu \pm \sigma$) to 1.13% ± 1.15%.

6.2 Per-Node and Per-Protocol Calibration

In Section 5.2, we intentionally *generalized* from the particularities of the MAC protocol by running OLS over four different MAC protocols. Hence, we obtained protocol-independent (but node-specific) estimation parameters. In order to obtain *per-protocol* (and node-specific) calibrated OLS estimator parameter values, the methodology applied in Section 5 can be applied without any adaptation. However only the observations of the specific protocol and node have to be chosen from the training set in order to calculate the OLS estimator. The same *specialization* effect can also be achieved by supplying more information to the

OLS regression model with introducing so-called *dummy variables* that indicate the currently used protocol (cf. [19], p. 299ff). We propose this *per-protocol* calibration as an even more accurate estimation approach, which might be useful if researchers know exactly what protocol they intend to use on the MAC layer in advance. We calculated different OLS parameter sets for each of the four protocols (S-MAC, T-MAC, WiseMAC, CSMA) and used the same node (node 1 in Figure 2) used in Section 5 for calculating the resulting accuracy on the validation set. The combined approach of *per-node and per-protocol* calibration obviously leads to the highest accuracy. Across all four protocols and traffic rates, we obtained a mean estimation error and standard deviation ($\mu \pm \sigma$) of only 0.42% ± 0.72%. The combined calibration approach however has multiplicative impact on the overhead before network deployment, as all nodes need to be equipped with tailor-made estimation model parameters for each protocol. Figure 10 illustrates the different estimation errors measured when applying the the *per-node and protocol-generic* or the *per-node and per-protocol* calibration approach.

7 Conclusions

This paper evaluates the accuracy of software-based energy estimation models on the MSB430 platform. We have identified and quantified the different factors which cause deviations of the software-based estimations from the *real* physically measurable energy consumption. The inaccuracies in the production of the electronic components have been shown to impact on different power consumption levels, which led to nodes differing by more than 4% in their energy consumption. The paper conveys that software-based energy estimation can be a valuable alternative to using sophisticated measurement hardware, especially in outdoor-deployments where the latter is impossible - at least for evaluating protocols where the CPU is used frugally, i.e., E^2-MAC or routing protocols. Enhancing today's most widely used simple *Three States Model* with information regarding the state transitions and applying multivariate OLS regression to calibrate the model parameters has been shown to remarkably reduce the estimation error. The mean absolute error (MAE) and standard deviation ($\mu \pm \sigma$) of the energy estimations of the software-based model using protocol-generic but per-node calibrated parameters could be pushed to as few as 1.13% ± 1.15%. Applying even more sophisticated parameter calibration of per-node *and* per-protocol calibration has been shown to reduce the mean absolute error and standard deviation to as few as only 0.42% ± 0.72% across the four evaluated wireless channel MAC protocols S-MAC, T-MAC, WiseMAC, and 802.11-like CSMA.

References

[1] Dunkels, A., Osterlind, F., Tsiftes, N., He, Z.: Software-based On-line Energy Estimation for Sensor Nodes. In: IEEE EmNets. (2007) 28–32

[2] Dunkels, A., Grönvall, B., Voigt, T.: Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors. In: IEEE EmNets. (2004) 455–462 http://www.sics.se/contiki/.
[3] Ye, W., Heidemann, J., Estrin, D.: An Energy Efficient MAC Protocol for Wireless Sensor Networks. In: INFOCOM. (2002) 1567–1576
[4] Polastre, J., Hill, J., Culler, D.: Versatile Low Power Media Access for Wireless Sensor Networks. In: ACM SenSys. (2004) 95–107
[5] Finne, N.,Eriksson, J.,Tsiftes, N.,Dunkels and A.,Voigt, T.: Improving Sensornet Performance by Separating System Configuration from System Logic. In: European Conference on Wireless Sensor Networks (EWSN). (2010) 194–209
[6] Boano, C.A., Voigt, T., Tsiftes, N., Mottola, L., Römer, K., Zuniga, M.: Making Sensornet MAC Protocols Robust Against Interference. In: European Conference on Wireless Sensor Networks (EWSN). (2010) 272–288
[7] Baar, M., Koeppe, E., Liers, A., Schiller, J.: The ScatterWeb MSB-430 Platform for Wireless Sensor Networks. In: SICS Contiki Workshop. (2007)
[8] Van Dam, T., Langendoen, K.: An Adaptive Energy Efficient MAC Protocol for Wireless Sensor Networks (TMAC). In: ACM SenSys. (2003) 171–180
[9] El-Hoiydi, A., Decotignie, J.D.: WiseMAC: An Ultra Low Power MAC Protocol for Multihop Wireless Sensor Networks. In: ALGOSENSORS. (2004) 18–31
[10] Buettner, M., Gary V. Y., Anderson, E. and Han, R.: X-MAC: A Short Preamble MAC Protocol for Duty-cycled Wireless Sensor Networks. In: ACM SenSys. (2006) 307–320
[11] Werner-Allen, G., Swieskowski, P., Welsh, M.: MoteLab: a Wireless Sensor Network Testbed. In: Information Processing in Sensor Networks (IPSN). (2005)
[12] Haratcherev, I., Halkes, G., Parker, T., Visser, O. and Langendoen, K.: PowerBench: a Scalable Testbed Infrastructure for Benchmarking Power Consumption. In: International Workshop on Sensor Network Engineering (IWSNE). (2008)
[13] Hergenröder, A., Wilke, J., Meier, D.: Distributed Energy Measurements in WSN Testbeds with a Sensor Node Management Device (SNMD). In: International Conference on Architecture of Computing Systems. (2010)
[14] Halkes, G., Langendoen, K.: Crankshaft: An Energy-Efficient MAC-Protocol For Dense Wireless Sensor Networks. In: European Conference on Wireless Sensor Networks (EWSN). (2007)
[15] Hergenröder, A., Horneber, J., Meier, D., Armbruster, P., Zitterbart, M.: Distributed Energy Measurements in Wireless Sensor Networks. In: ACM SenSys, Demo Session. (2009)
[16] Texas Instruments CC1020: Single-Chip FSK/OOK CMOS RF Transceiver
[17] ScatterWeb^2 OS - Freie Universität Berlin & ScatterWeb GmbH: http://scatterweb.mi.fu-berlin.de/ and http://www.scatterweb.de
[18] Landsiedel, O., Wehrle, K., Goetz, S.: Accurate Prediction of Power Consumption in Sensor Networks. In: IEEE EmNets. (2005)
[19] Draper, N., Smith, H.: Applied Regression Analysis, Wiley Series in Probability and Statistics (1998)