First published in: The Fifth Annual Conference on Wireless On demand Network Systems and Services,
Garmisch-Partenkirchen, Germany, January 23-25, 2008

Towards a Realistic Energy Model for Wireless Sensor Networks

Simon Kellner
System Architecture Group
University of Karlsruhe, Germany
Email: kellner@ira.uka.de

Mario Pink, Detlev Meier, Erik-Oliver Blaß
Institute of Telematics
University of Karlsruhe, Germany
Email: {pink,meier,blass}@tm.uka.de

***Abstract*—Energy consumption is one of the most critical protocol properties in a wireless sensor network. Therefore, a precise energy model is required for the evaluation of protocols. In addition to evaluation, an energy model can also be used for on-line energy accounting. After processing user queries, nodes aware of the energy model can send an energy bill towards the network's data sink. This allows the user to adjust future queries to be more energy efficient. The contribution of this paper is twofold: First, a theoretical energy model based on simple finite automata is presented. This model can be used for on-line accounting, simulation and generation of a-priori knowledge. Second, the proposed theoretical model is backed up by practical measurements using a new measurement device called SNMD, which also offers management functions for sensor-net testbeds. Therewith, output of the theoretical model can be compared to and validated against by real-world measurements.**

I. Introduction

Maximizing a sensor network's lifetime is one of the most complex and challenging problems in the field of sensor networks. To compare algorithms in this metric, an energy model is necessary.

Evaluations in sensor networks typically use simplified assumptions for energy consumptions. When talking about the network level, for example, a very simple energy metric could be the total number of packets sent. However, such a metric does not only ignore the energy consumption of a sensor node's microcontroller, but also the energy spent by the node while listening for packets – yet, this energy consumption can be significantly higher than the consumption for sending packets, as is the case with the CC2420 chips that are used on several types of sensor nodes.

Furthermore, protocol evaluation is typically done *off-line*: Using discrete event simulators, protocols are evaluated by computing energy consumption from simulated events using less than adequate energy models. These energy models have to work with the little information made available by the simulator.

Our on-line energy accounting approach can count on a much richer set of information, such as the length of individual packets, and can consider all minute hardware events that really have occurred and that would be missing in a high-level simulation, like spurious interrupts, for example. This approach can be used to make dynamic TinyDB-like [1] applications on sensor nodes *energy-aware*.

Finally, precise measurements are necessary to test the accuracy of energy models in realistic scenarios. Current sensor-node testbeds are either ill-equipped for precise measurements of their nodes' energy consumption or are not flexible enough to stage different scenarios.

The contribution of this paper is twofold:

1) In this paper, we propose an energy model based on finite state machines (FSMs). Extensions to the FSMs limit the memory footprint so that the state machines can be implemented on sensor nodes.
2) To *measure* a sensor node's energy consumption, a special monitoring device *SNMD* is presented. When deployed with each sensor node in our planned testbed, these devices will allow us to individually monitor each node in the testbed, so that the energy model can be fine-tuned to each sensor node. Moreover, they will also provide node management functions and support repeatable experiments by simulating the sensors' environment.

The rest of this paper is structured as follows: After describing related work in Section II, we present our energy model in Section III. Then we discuss the architecture of our SNMD device for measurement and management of sensor nodes in Section IV, before concluding this paper in Section V.

II. Related Work

There have been several approaches to the construction of energy models for wireless sensor networks.

Most of them, e.g. PowerTOSSIM [2], AEON [3], and the model of Schmidt, Krämer et al. [4], are targeted at making existing simulators energy-aware, even if the use in on-line accounting is briefly mentioned in [4].

We are currently aware of only one energy model specifically built for and used for on-line accounting, the model from Dunkels et al. [5]. However, their energy model, particularly the model of the communication system, appears to be relatively simple: There are only two states for the microcontroller and the radio chip, respectively. Clearly, this does not cover all of a node's possible energy states.

Although energy consumption in WSNs is of increasing interest in research, the accuracy of such experiments leaves room for improvement. Few sensor network testbeds measure the energy consumption of their nodes, and those who do are limited: In the MoteLab testbed [6] for example, only one sensor node is being measured. The JAWS testbed [7] supports energy measurement capabilities for all sensor nodes using BTnodes [8] for the monitoring network [9]. However, measurement accuracy and detail are limited by the sensor nodes' internal analog digital converter (ADC) and by the bluetooth monitoring network, respectively.

Other testbeds, e.g., TWIST [10], offer the simulation of different power states for each sensor node in order to support reproducible experiments, but neglect implementing repeatable, yet changing environmental conditions.

III. Energy Model

As we plan to use our energy model both in off-line simulations and on-line accounting, its structure should be simple to allow access from different applications. The intended use in on-line accounting requires a small memory footprint for the model to fit into a typical sensor node.

A. Design

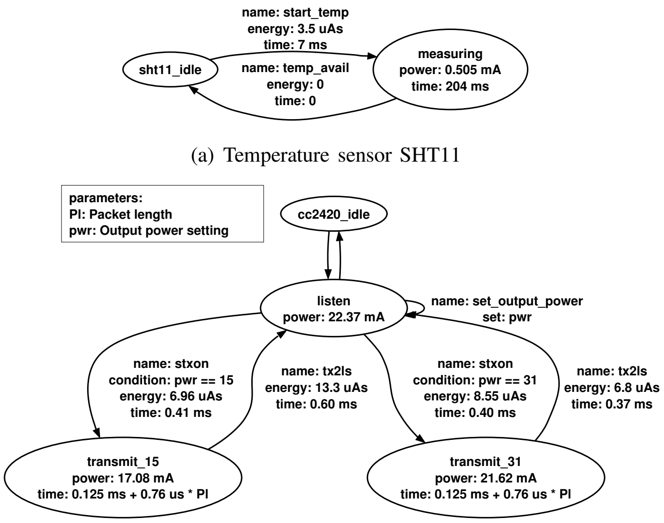
Our energy-model consists of a set of finite state machines that represent the states and transitions of a sensor node's hardware. States and transitions of the state machines are attributed with the physical characteristics of their hardware equivalent, such as time (duration), energy, or power. By stepping through the model states exactly as the hardware would and summing up the annotated energy values, one can estimate the energy consumption of a sensor node.

The physical characteristics often depend on parameters from outside such as the battery voltage or the packet length. Values for these parameters have to be supplied from external sources, which could be an application simulator like tossim, or on-line from the running sensor node operating system.

To match the concurrency possible on typical hardware, there is one or more state machine for each device on the sensor node. For example, the OS can finish instructing the radio controller to send a packet, start taking measurements on a sensor, and get interrupted by a preset timer, all within a few instruction cycles. In this example, the radio controller, the sensor, and the microcontroller each are modeled by one finite state machine. This design ensures a small total number of machine states.

The model states describe the hardware states with regard to their power consumption. Simple hardware like sensors or LEDs can be modeled by a small number of states. The model for the SHT11 chip in Fig. 1(a), for example, covers the whole process of measuring the temperature. For other chips like the ZigBee controller CC2420, one hardware state can be represented by many machine states, which can be seen from the small part of the CC2420 model shown in Fig. 1(b). The model has multiple *transmit* states, one for each selectable transmission power. The full CC2420 model cannot be displayed here because of space constraints.

Transitions in the model describe the time and energy spent on changing the hardware state. Transitions can be observed by the operating system either when sending a command to or receiving an interrupt from the hardware.



(a) Temperature sensor SHT11

(b) ZigBee controller CC2420 (partial model)

Fig. 1. Energy models of two devices on sensor nodes

As some simulators are not that detailed and don't simulate interrupts, we have to add redundant information. In Fig. 1, the *time* attribute in the *transmit* and *measuring* states will be ignored for on-line accounting but is required for high-level simulation. For on-line accounting, we prefer measuring the time spent in hardware states over estimating it.

In some cases it is advantageous to extend the finite state machines. We extend our FSM-model of the CC2420 by variables and conditions in order to keep the total number of states and transitions to a minimum. The CC2420 has 32 settings for the transmit power. This setting can only be programmed in the states *idle* and *listen*, but it affects only the power consumption in the *transmit* state. This situation can be modeled in an FSM by splitting each of the states *idle* and *listen* into a group of 32 states that represent the output power setting. Each of these groups must be fully connected to allow switching the output power from each setting to every other setting.

Instead, we add the variable *pwr* to the CC2420 model and use constraints (*condition*s) to select one of the identically named transitions to the transmit states. Using this approach, we save 62 states and 1052 transitions that are not energy relevant over the pure FSM-based approach.

B. Calibration

To calibrate our model for the MICAz platform, we inserted hooks in the TinyOS drivers for each hardware component. These hooks serve two purposes: They allow us to signal energy-relevant events (such as sending a hardware command or receiving an interrupt) via a general purpose IO pin that is measured along with the normal node power consumption. When the model calibration is finished, the hooks will allow on-line accounting of the node's energy consumption by using the data they helped create.

The positions of these hooks were chosen so that they

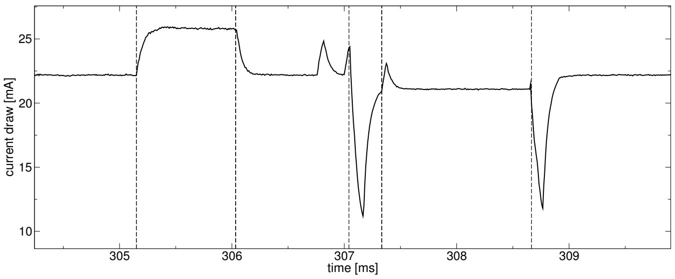


Fig. 2. Measurement of packet transmission (CC2420)

correspond well with changes in power consumption. This can be seen in Fig. 2, which shows the power consumption of a MICAz node while sending one packet. Each of the vertical lines indicates one hook being triggered.

C. Applications

We plan to use our energy model for adding energy consumption information to the outputs of various simulators we use, and for deriving coarse-grained energy information used by lifetime maximizing algorithms at application level.

The main purpose of this energy model, however, is the possibility of on-line energy accounting on sensor nodes. The model is constructed so that it is easy for a sensor-node operating system to track hardware states while executing a sensor-net application and from this to derive the energy consumption of the node. A sensor-net application can then use this information to selectively account energy, e.g., the energy spent on a single query, and send this information back to a data sink. This information could in turn be used to tune the sensor-net application in order to adjust its energy consumption.

IV. Measurement Device

The usefulness of the energy model depends on its accuracy. It therefore has to be calibrated and tested in operation. We now present the architecture of our Sensor Node Management Device (SNMD) that offers fine-grained, precise measurements as well as management functions for WSN testbeds. The SNMD has to meet three scientific requirements [11]:

- Completeness: give as much data as possible for comprehensive measurements
- Correctness: enable the verification of the correctness of results obtained from measurements
- Reproducibility: allow repeated experimental runs under replayed environmental changes

A. Completeness of Measurements

To achieve these goals, we need a measurement device that provides high precision measurements and supports the management of the sensor nodes. Commercial devices have no sensor-node management functions and are mostly either limited in their measurement resolution or are unable to measure voltage and current simultaneously. Some devices such as the NI USB-6210 [12] meet our measurement requirements but are over-equipped in that they could measure 4–5 nodes simultaneously. This, however, would result in a rather fixed testbed structure as the nodes should be near to the measurement device to prevent signal degradation.

Therefore we designed a new type of energy measurement device that can be used for fine-grained distributed energy measurements.

B. Correctness of Measurements

Parts of the testbed in which we want to use our measurement device lie outdoors. Here, measurement equipment is far more likely to fail than in a laboratory. For this reason, our SNMD should consist of different redundant measurement *units* to enable the verification of measurement data and so to guarantee its correctness.

C. Reproducibility of Measurements

Sensor node behavior can largely depend on environmental conditions, e.g. a purely event-driven application that reports only unusual readings. It is hard to improve or to understand such mechanisms if the system behavior changes from one test run to the next. Thus, reproducibility of environmental influences as well as battery states should be made feasible to allow testing different sensor-net applications in the same situations.

D. Requirements

Thus, the collected requirements of our energy measurement device are as follows:

- high fidelity energy measurements of voltage and current
- switchable sensor node power supply
  - constant power
  - battery power
  - no power, to simulate sensor node failures
- battery recharge for minimum maintenance effort
- controlled battery discharge for battery state reconstruction
- verification of correct sensing equipment function
- accurate environment simulation

All these requirements aim at providing an easy-to-manage testbed with redundant measurement equipment and environmental simulation.

E. Sensor Node Management Device

To meet the requirements mentioned above we have developed a sensor node management device that consist of four different units: measurement, charge, power-switching, and control unit (see Fig. 3).

1) Measurement Unit: The measurement unit is used to obtain precise energy consumption data of a sensor node. The heart of the measurement unit consists of an AD7654 [13] 16-bit ADC from Analog Devices that supports measurement with a resolution of up to $76.249\,\mu V$ and a frequency of up to $500\,\text{Hz}$. This ADC is sensitive enough to measure the small battery energy consumption resulting from the execution of

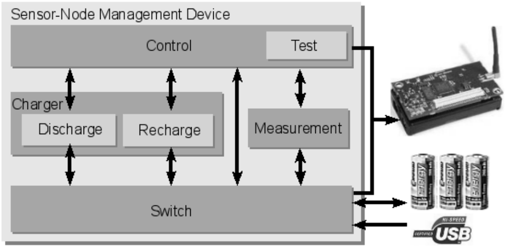


Fig. 3. Sensor node management device architecture

TinyOS operations such as sending packets or reading sensors, as well as complete algorithms, e.g. encryption.

We can also use the temperature sensor of our reference voltage chip to check the temperature data from the sensor node.

2) Charge Unit: The charge unit is responsible for battery recharge to ease maintenance work. It also supports controlled discharge of the battery to a specific charge in order to restore a recorded battery state. The recharge subunit is implemented by the battery monitor and charge controller DS2770 from Maxim [14]. This chip allows to recharge the sensor node battery and makes it possible to use either NiMH (3×1.2 Volts) or Li-ion batteries (1×3.6 Volts). Through the use of an external measurement resistor of $0.1\,\Omega$ it is possible to measure current with a resolution of $6.52\,\mu V$ and a frequency of $285\,\text{Hz}$ as well as the voltage with a resolution of $1.56\,\mu V$ and a frequency of $18\,\text{Hz}$.

Furthermore we use the current accumulator register that holds the accumulated current measurements of the battery in both directions and represents the energy balance of the battery to estimate whether it is possible to perform an entire experiment with the remaining battery energy.

The discharge subunit mainly consists of the MAX5812 digital analog converter (DAC) that is able to manipulate the resistance of a MOSFET transistor and so can control the discharge of the battery.

3) Switching Unit: To simulate the error-prone behavior of sensor nodes in WSNs, the SNMD is also equipped with a switching unit. This switching unit is responsible for activation and deactivation of the sensor node power supply (either from USB or the battery), to enable or disable the environment simulation equipment and, last but not least, to enable or disable the recharge process of the charge unit.

4) Control Unit: The control unit manages all SNMD units and connects these devices via USB to a host system. It consists of an ATmega1280 microcontroller from Atmel with integrated 128 kB flash memory and an external SDRAM of 1024 kB. The USB connection with a host system is implemented by an CP2102 USB controller chip that connects one of the ATmegas UARTs to the USB. In addition, the CP2102 has an on-chip voltage regulator which powers the device. Thus, everything is powered from a host system's USB port. The remaining UART interfaces provided by the ATmega1280 are used for a local serial console which can be used to debug the SNMD firmware as well as for a sensor node programming and monitoring interface. Sensor nodes are able to start and stop measurements via trigger signals to the SNMD. Last but not least, this unit is equipped with an additional test subunit. This subunit consist of a collection of sensors and actuators, such as microphones or speakers, that can be used for diagnostic or environment simulation functions.

V. Conclusion

In this paper, we presented an energy model that allows to model the energy consumption of a whole sensor network in a *fine-grained* manner. This model maps hardware energy states and OS activities into corresponding states and transitions in a set of finite state machines. With this model and its application in *on-line energy accounting*, it is possible to get a more detailed and more precise view on the energy consumption in a sensor network than before. Data gathered from the on-line accounting can be used to tune the energy consumption of sensor node applications automatically at run-time.

In combination with a new measuring device, SNMD, that enables *fine-grained*, *distributed* energy measurements, we are able to test and improve our model's accuracy.

References

[1] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "The design of an acquisitional query processor for sensor networks," in *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM Press, 2003, pp. 491–502.

[2] V. Shnayder, M. Hempstead, B. Chen, G. Werner-Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM Press, Nov. 2004, pp. 188–200.

[3] O. Landsiedel, K. Wehrle, and S. Götz, "Accurate prediction of power consumption in sensor networks," in *Proceedings of the second IEEE Workshop on Embedded Networked Sensors (EmNetS-II)*, May 2005.

[4] D. Schmidt, M. Krämer, T. Kuhn, and N. Wehn, "Energy modelling in sensor networks," *Advances in Radio Science*, vol. 5, pp. 347–351, Jun. 2007.

[5] A. Dunkels, F. Osterlind, N. Tsiftes, and Z. He, "Software-based on-line energy estimation for sensor nodes," in *Proceedings of the 4th workshop on Embedded networked sensors(EMNETS'07)*. New York, NY, USA: ACM, 2007, pp. 28–32.

[6] "the MoteLab testbed website," http://motelab.eecs.harvard.edu/.

[7] J. Beutel, "Fast-prototyping using the BTnode platform," in *DATE '06: Proceedings of the conference on Design, automation and test in Europe*. 3001 Leuven, Belgium, Belgium: European Design and Automation Association, 2006, pp. 977–982.

[8] J. Beutel, O. Kasten, F. Mattern, K. Römer, F. Siegemund, and L. Thiele, "Prototyping wireless sensor network applications with BTnodes," in *Proceedings of the 1st European Workshop on Sensor Networks (EWSN 2004)*, ser. Lecture Notes in Computer Science, vol. 2920. Springer, Berlin, Jan. 2004, pp. 323–338.

[9] M. Dyer, J. Beutel, L. Thiele, T. Kalt, P. Oehen, K. Martin, and P. Blum, "Deployment support network - a toolkit for the development of WSNs," in *Proceedings of the 4th European Conference on Wireless Sensor Networks*. Springer, Berlin, Jan. 2007, pp. 195–211.

[10] H. Handziski, A. Köpke, A. Willig, and A. Wolisz, "Twist: A scalable and reconfigurable wireless sensor network testbed for indoor deployments," Telecommunication Networks Group, Technische Universität Berlin, Tech. Rep. TKN-05-008, Nov. 2005.

[11] W. Kiess, S. Zalewski, A. Tarp, and M. Mauve, "Thoughts on mobile ad-hoc network testbeds," in *Proceedings of IEEE ICPS Workshop on Multi-hop Ad hoc Networks: from theory to reality*, Jul. 2005, pp. 93–100.

[12] "NI USB-6210 USB multifunction DAQ," http://sine.ni.com/nips/cds/view/p/lang/de/nid/203189.

[13] "AD7654 ADC," http://www.analog.com/UploadedFiles/Data_Sheets/AD7654.pdf.

[14] "DS2770 battery monitor and charge controller," http://datasheets.maxim-ic.com/en/ds/DS2770.pdf.