Title	Modeling power in multi-functionality sensor network applications
Authors	Agarwal, Rachit;Martínez-Català, Rafael V.;Harte, Seán;Segard, Cedric;O'Flynn, Brendan
Publication date	2008
Original Citation	Agarwal, R., Martinez-Catala, R.V., Harte, S., Segard, C., O'Flynn, B. 2008. Modeling Power in Multi-functionality Sensor Network Applications. In: IEEE Proceedings of the Second International Conference on Sensor Technologies and Applications. Cap Esterel, France 25-31 August 25 2008.
Type of publication	Conference item
Link to publisher's version	10.1109/SENSORCOMM.2008.64
Rights	© 2008 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.
Download date	2024-04-26 18:04:47
Item downloaded from	https://hdl.handle.net/10468/96

Modeling Power in Multi-Functionality Sensor Network Applications

Rachit Agarwal[†], Rafael V. Martinez-Catala[‡], Sean Harte[‡], Cedric Segard[‡] and Brendan O'Flynn[‡]

[†]*Institute of Communications Engineering, Technical University of Munich, Germany*

[‡]*Microelectronics Application Integration Group, Tyndall National Institute, Cork, Ireland*

rachit.agarwal@tum.de, {rafael.martinez, sean.harte, cedric.segard, brendan.oflynn}@tyndall.ie

Abstract

With the migration of a Wireless Sensor Network (WSN) over various evolving applications, power estimation and profiling during the design cycle become critical issues and present hurdles in reducing the design time. Furthermore, with a growing size of the network, simulating the behavior of each sensor node is not feasible. It is important to devise an approach that provides a network-wide picture of power consumption and of variations in power usage under changes in the network and/or node application in the network.

In this paper, we present a modular power estimation technique which simplifies the power modeling of any sensor network application. In particular, we are interested in analyzing the behavior of power consumption if one or more modules of the WSN platform in the application are changed during the design cycle or after the deployment. The proposed technique is susceptible to applications changes on the fly and is particularly beneficial in networks with large number of nodes. We perform experiments modifying parameters of a ZigBee based sensor network application such as packet size, sampling rate, functionality (encryption) and sensor types. We present the results, demonstrating an error less than 3% in all the experiments performed, and insights into the results.

1   Problem Statement

The challenges of developing, debugging and evaluating complex large-scale sensor network applications have increased with evolving applications. Simulation environments, such as ns2 [1], TOSSIM [2] and Atemu [3], provide varying degrees of scalability, realism and detail for understanding the behavior of sensor networks. Questions related to power consumption of a sensor network application form a major class of challenges, specially due to two factors, *viz.*, the "complexity of sensor network applications" and the "size of sensor networks".

With the growing complexity of sensor network applications, estimating power consumption of a WSN node becomes a non-trivial task. Efforts have been made to answer some of these questions by integrating a scalable simulation environment PowerTOSSIM [4] that provides an accurate, per-node estimate of power consumption. Furthermore, node level power estimation can be extended to estimation the network lifetime.

With the growing size of sensor networks, node-level or circuit-level power estimation becomes a non-trivial and tedious task, particularly if *various nodes are running different applications*. This problem becomes even more critical if an application would like to *change the functionality of all or some of the nodes in the network on the fly*. Applications in which WSN nodes are wirelessly reprogrammed and functionality changes without actually redeploying the network pose many questions on network lifetime. In this context, one question of primary importance is:

> *How does an application change in all or few of the nodes reflects in the power consumption and network lifetime of a Wireless Sensor Network?*

For energy consumed in information processing, algorithm level power estimation techniques can be applied to various algorithms that run during the execution of the system application for an accurate estimation of the processor power [5][6][7]. Similarly, power consumed in the sensing process can be individually simulated for various sensors which are planned to be used in the application before actually designing the complete network. These power estimations can be performed during, or even before, the design cycle and debugging the complete network. We refer to these estimations as "static estimations", henceforth.

We are interested in analyzing the implications of these static estimations in the overall power profile and lifetime of the network. In most of the power simulators and emulators proposed in literature, an application change in a few or all of the nodes of the network requires estimation to be performed from scratch. This is very time consuming when the network size increases to few thousand nodes.
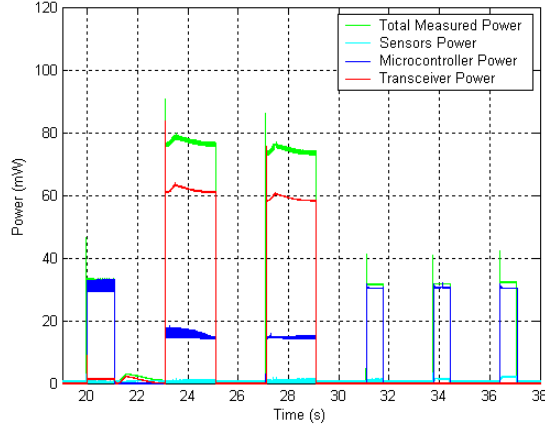
Figure 1. Power Profiling for Various Components

In particular, we are interested in devising a technique to answer the following question:

> *We have to deploy our network in scenario $\mathcal{X}$ for application $\mathcal{Y}$. Functionality difference for each node in application $\mathcal{Y}$, $\mathcal{F}_k, k = 1, 2, \ldots, \mathcal{S}$, is known with respect to application $\mathcal{Z}$. If the energy consumption for application $\mathcal{Z}$ was $\mathbb{E}_{\mathcal{Z}}$, what will be the energy consumption for application $\mathcal{Y}$?*

In this paper, we present our experiences with power estimation of sensor network applications based on the static estimates for individual modules of a WSN node and insights into the achieved results. We analyze each of the energy sinks and construct a modular power estimation model to answer the above question. We perform different experiments changing parameters of the ZigBee application such as encryption, sampling and packet size. The objective of this paper is primarily to deduce whether a modular power estimation approach is acceptable for reducing the design time of sensor network applications.

2 Modular Power Estimation

Modularity is an important facet of system design allowing reduced design time. In this research work, we try to explore this property in the context of WSNs. We demonstrate through experimental results that modularity can be seen as a very effective tool in reducing the power estimation of WSN application design because of the abstract distinction between power consumption of communication, sensing and information processing units.

Any WSN system can be thought of three functional units; Sensing, Processing and Radio Link. Given any ap-

plication, power usage can be characterized based on sensor types, information processing algorithm used, communication transfer rate and data size for the application. We discuss these design variables in the following subsection after throwing a quick glance on the simple modular power estimation technique.

Assume that Fig. 1 shows the power profiling for various components of any node. Given the sensor, processor and transceiver powers for various operation states (shown in cyan, blue and red), we are interested in comparing the system power (shown in green) to the summation of the component powers. This is indeed a highly simplified model but, as we shown in Section 4, gives us a highly accurate estimation of the node power (with an error of less than 3%). Further, it provides us with a technique which can be extended for the complete network given that functionality and data characteristics of each node in the network are known.

Design Variables.

In general, the power consumption of a sensor network is dependent on communication type and application data type. In this paper, we take these two variables and perform experiments to demonstrate the applicability of the modular power consumption model.

Data Size: Power consumption of any WSN has a strong dependency on data size for the application under consideration. While most WSN applications have small data sizes, a small change in the data size might cause a big change in the power consumption of the overall network. This is due to the fact that transmitting and receiving the packets (communication power) may overshadow the information processing and sensor power consumption.

Sampling Rate: Changing the sampling rate during an application may affect power consumption through change in communication rate and information processing frequency.

Application: We refer to an application as the algorithm or information processing on the processor. There are excellent techniques studied for estimating the power consumption of an information processing algorithm on the processor [5][6][7]. We are more interested in how this change in power consumption of the processor affects the overall power consumption of the network.

Radio Link: In various operation modes, the radio consumes varying amount of energy. For example, while transmitting the packet, a network protocol may have different performance when compared to while being in receiving mode.

Hardware: While the hardware used directly affects the power consumption of the network, we assume that once the network has been deployed, the processor or the radio link are not subject to any changes. However, an application may switch between various sensor modules. We are, again, more interested in studying how these temporal changes in the hardware affect the power consumption.

3 Experimental Set-up and Measurements

One of the important issues in estimating the performance of this model is to find a technique to measure the individual power consumptions in various components of the node. It is relatively easy to find the current drawn by the complete node on any of the commercially available motes, however, measuring the power of individual components is generally not feasible. In the following subsections, we discuss these issues.

Experimental Platform

We use as the underlying platform 25mm wireless motes [8], which provide us a novel 3-D programmable modular platform with an FPGA, a microcontroller, a transceiver, a sensor and a coin cell battery submodule.

The transceiver submodule consists of an Ember EM2420 2.4 GHz ISM band RF transceiver and the microcontroller is Atmel Atmega 128L. The communication standard employed is ZigBee Standard [9] using a commercial implementation, the EmberZNet v2.5.2 stack available from Ember Corporation [10]. With few simple modifications in the circuitry as discussed below, the platform allows to measure the current drawn individually by all the submodules *viz.*, processor, sensors and transceiver, as well as for the complete mote.

Set-up and Measurements

Fig. 2 shows a block diagram of the measurement setup. Since the power supply lines of each component are laid out using a star topology we can isolate the supply for different power sinks of the system and measure the power consumption independently. To do this, we use two DC supplies, one for the component under test and the other for the remaining components (shown by DC 3.3V at the top of Fig. 2). We evaluate the current by measuring the voltage drop across the resistor, which is connected to two channels of a 1.25 MS/s, 12-bits NI PCI-6071E acquisition card. The power consumption is calculated by measuring the input current and multiplying by the regulated voltage. The voltage drop across the resistor does not affect the node, as it is placed before a linear regulator with a very low quiescent current, typically 0.8 μA.
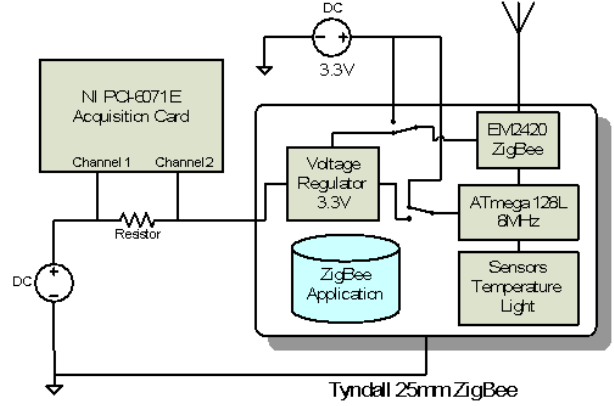

Figure 2. Measurement setup.

4 Model Accuracy: Experiences & Insights

We have two category of experiments, one, for *various operation states* demonstrating the applicability of the technique at various states the nodes can operate in and two, we make frequent *changes in the design variables* to check the applicability of the model. A combination of two can then be extended to any network application of any size.

Experiments: Various Operation States

We first present the power consumption for each submodule of the ZigBee platform in different operation states. We identified a number of operation modes in which the ZigBee node operates in; configuration, transmission, reception, sensing 1, sensing 2, sensing 1+2 and power-down. Sensing 1 and Sensing 2 correspond to the temperature and light level readings from attached thermistor and light dependent resistor (LDR) respectively, which are connected to the embedded analog to digital converter (ADC) on the microcontroller.
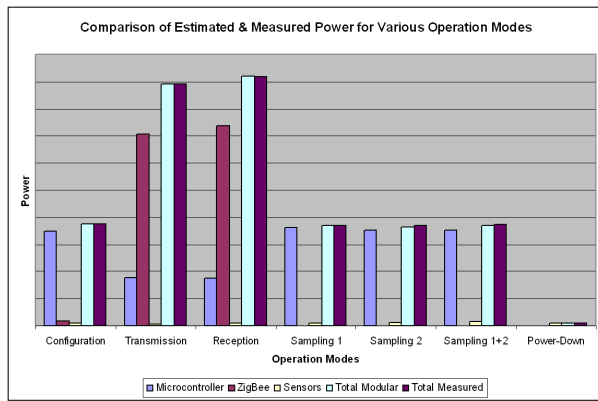
To evaluate the power consumption of each operation state we program the platform to cycle through all of the above mentioned states. Then, the hardware allows for independent direct power measurement of the microcontroller and transceiver and indirectly for measurement of the sensors. The average power is calculated by evaluating the energy consumption on each operation state and then dividing by the duration of the operation sate, the results for the calculation are presented in Table 1.

Accuracy

To study the accuracy of the model, we compare the estimated and measured power in all the operation states as shown in Table 1. The results have been plotted in Fig. 3.

Table 1. Power Consumption (mW) of different modules for different operation states

Operation Mode	Atmega	Zigbee	Sensor	Total Modular Modular	Total Measured Measured	Error
Configuration	35.037	1.640	0.912	37.589	37.540	+ 0.13 %
Transmission	17.653	70.726	0.659	89.038	88.982	+ 0.06 %
Reception	17.427	73.548	0.889	91.864	91.748	+ 0.13 %
Sampling 1	36.052	0.003	0.985	37.04	37.089	- 0.13 %
Sampling 2	35.123	0.003	1.300	36.426	36.851	- 1.15 %
Sampling 1+2	35.313	0.003	1.617	36.993	37.379	- 1.05 %
Power-Down	0.005	0.005	0.897	0.907	0.928	- 2.26 %

Figure 3. Estimated and measured power consumption for each operation state

Table 2. Details of the Experiments Performed

Experiment	Encryption	# Samples	# Bytes Transmitted
# 1	No	2	4
# 2	Yes	2	4
# 3	No	10	20
# 4	Yes	10	20
# 5	No	20	40
# 6	Yes	20	40
# 7	No	40	80

It is easy to see that in all cases, the estimated power is close to measured power with very low error. Computing the error from the values in Table 1 shows that in power-up state, the error margin is always less than 1.25%. However, in power-down mode, the error margin is approximately 2.5%.

Experiments: Varying Design Variables

To study the applicability of the model in the context of large-scale sensor networks, it is important to understand the behavior of the model in presence of diversified nodes, applications, data and sensor types. We perform seven experiments to study the effect of application changes and varying design variables in power consumption of a node. In our experiments, we make changes in the sampling rate and data rate, along with application changes.

In terms of the application running on the processor, we choose a security protocols (encryption) as our application. In half of the experiments, we let the processor run without the security protocol and in the other half, we see if the changes in power consumption are modular with respect to the processor running the security protocol. Table 2 lists all the experiments performed.

Fig. 4 displays the power profile corresponding to experiment #2. In the first few seconds, the end-node scans the available channels until it finds a network coordinator, and then there is a network configuration process while the node joins the network. This process is very energy consuming as the end-device is continually transmitting and receiving. After that, the end device is ready to operate within the network, and the sampling/transmitting cycle begins. The period of time between the three peaks or communication processes and the next three peaks corresponds to the repeated cycle. The first peak is where the sampling and data transmission occurs. The second and third peaks are polling by the end device, these polling processes are always the same and they do not depend on the sampling rate, transmitted bytes or even if encryption is used or not. Using a time scale of seconds, all the recoded profiles seem to have the same morphology, but a deeper inspection of the active periods reveals where the differences reside.
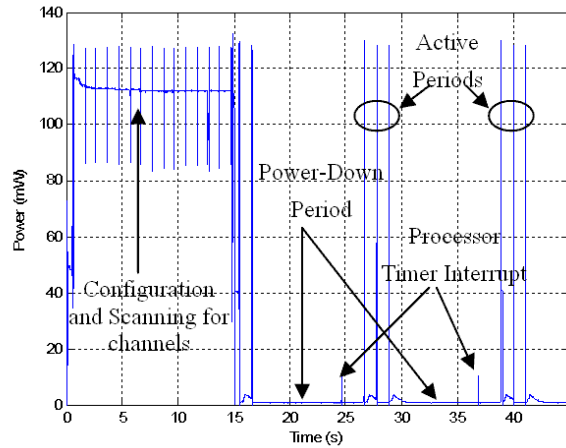
Figure 4. Representation of the Power Profile for Experiment # 2

Figure 5. Comparison of the three active periods in chronological order from top to bottom corresponding to experiments #1 and #2

Fig. 5 shows a zoom on the three active periods comparing an encrypted and a non-encrypted ZigBee application corresponding to experiments #1 and #2. It illustrates that the difference resides only in the first active period. We therefore analyze the impact on the energy consumption of this active period, by changing the parameters described in Table 2.

The results for the measured power of various components for each experiment are shown in Table 3.

Table 3. Energy Consumption (mJ) of First Active Period

# Exp.	Atmega + Sensors	ZigBee	Total Modular	Total Measured
# 1	2.1751	0.6656	2.8407	2.8660
# 2	4.2436	1.0246	5.2199	5.2682
# 3	2.3538	0.8084	3.1623	3.2109
# 4	3.9515	0.8905	4.8421	4.9739
# 5	2.1965	0.8852	3.0817	3.3580
# 6	4.3555	0.9614	5.3169	5.4150
# 7	2.7646	0.9717	3.7363	3.7411

Accuracy

A comparison of the total estimated power (through modular power estimation) and the system measured power is shown in Fig. 6. Similar to the earlier set of experiments, the estimates match very closely to the measured results.
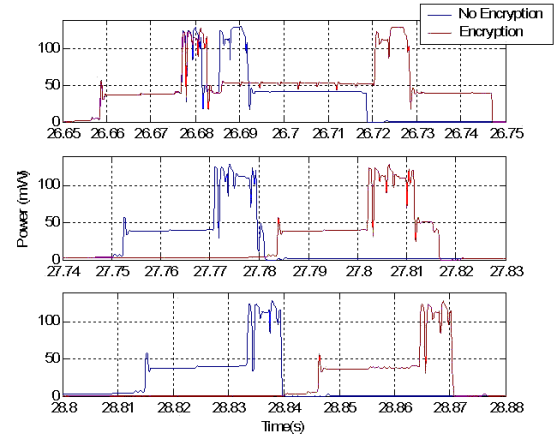
Computing the error for various experiments from Table 3, the worst-case error happens to be less than 3%, which is clear from the plots of Fig. 6.

5 Future Work: Extending the Model

In this paper, we have demonstrated that the power consumption in WSNs follow a modular behavior, modularity being in terms of various components of the mote and various operations states in which the mote operates. Experimental results and accuracy analysis for the model have been presented which prove the high accuracy of the proposed technique. These results can further be extended to incorporate more design variables and network scenarios. For example, questions on extending the model to large-scale networks and integrating the model to existing power simulators and network simulators form a major part of our ongoing research. We close the paper with some final remarks on these questions:

Estimating Power during Application Changes

One of the important questions answered by this work is "fast and accurate" estimation of the "node" power consumption when the system migrates from one application to the other. For instance, assume that some nodes in the network switch from application $\mathcal{Y}$ to application $\mathcal{Z}$. Also, assume that modules $\mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_k$ are eliminated and $\mathcal{Z}_1, \mathcal{Z}_2, \ldots, \mathcal{Z}_l$ are added for the application change.
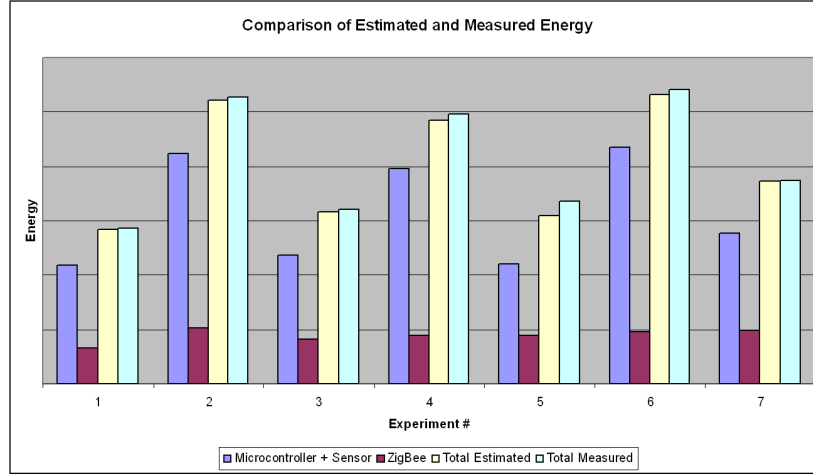
Figure 6. Comparison of Results for the seven experiments

The node power consumption can, hence, be updated to:

$$\mathcal{P}_{\mathcal{Z}} = \mathcal{P}_{\mathcal{Y}} - \sum_{i=1}^{k} \mathcal{P}_{\mathcal{Y}_i} + \sum_{j=1}^{l} \mathcal{P}_{\mathcal{Z}_j}$$

where, $\mathcal{P}_{\mathcal{Y}_i}$ and $\mathcal{P}_{\mathcal{Z}_j}$ are the static power estimates of modules $\mathcal{Y}_i$ and $\mathcal{Z}_j$.

Extension to Networks

Now, assume that we are interested not only in the power consumption of all the nodes but the complete network. Given the modular behavior of the power consumption in WSNs, the extension of the model to the network power estimation can either be performed by integrating modular model to any network power simulator or by updating the communication power consumption during changes in functionality or application. It would, however, be interesting to estimate the network power consumption using this model and compare it with the estimates obtained by complete network simulations, such as from [4].

Acknowledgment

References

[1] K. Fall and K. Varadhan. The *ns* Manual. http://www.isi.edu/nsnam/ns/doc/index.html

[2] P. Levis, N. Lee, M. Welsh and D. Culler, TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications, *Proc. Intl. Conf. Embedded Networked Sensor Systems*, Los Angeles, CA, 2003, pp. 126-137

[3] M. Karir, Atemu - Sensor Network Emulator, http://www.isr.umd.edu/CSHCN/research/atemu

[4] V. Shnayder, M. Hempstead, B. Chen, G. W. Allen, M. Welsh, Simulating the Power Consumption of Large-Scale Sensor Network Applications, *Proc. Intl. Conf. Embedded Networked Sensor Systems*, Baltimore, MD, 2004, pp. 188-200

[5] M. T. Lee, V. Tiwari, S. Malik and M. Fujitsa, Power Analysis and Low Power Scheduling Techniques for Embedded DSP Software, *Proc., IEEE Intl. Symp. on System Synthesis*, 1995, pp. 110-115

[6] S. Lee , A. Ermedahl , S. L. Min, An Accurate Instruction-Level Energy Consumption Model for Embedded RISC Processors, *Proc., ACM Work. Languages, Compilers and Tools for Embedded Systems*, Snow Bird, UT, 2001, pp. 1-10

[7] V. Tiwari, S. Malik, A. Wolfe and M. T. Lee, Instruction Level Power Analysis and Optimization of Software, *Intl. Conf. on VLSI Design*, Bangalore, India, 1996 pp. 326-328

[8] S. J. Bellis *et al.*, Development of field programmable modular wireless sensor network nodes for ambient systems, *Computer Communications*, Vol. 28, No. 13, pp. 1531-1544, Aug. 2005

[9] ZigBee Alliance, ZigBee Specification, 2006, http://www.zigbee.org

[10] Ember Corporation: http://www.ember.com