Energy Evaluations in Wireless Sensor Networks – A Reality Check

Christian Haas
Institute of Telematics
Karlsruhe Institute of Technology, Germany
christian.haas@kit.edu

Joachim Wilke
Institute of Telematics
Karlsruhe Institute of Technology, Germany
joachim.wilke@kit.edu

ABSTRACT

The development of *energy-efficient* applications and protocols is one of the most important issues in Wireless Sensor Networks (WSN). However, most publications up to now avoid time consuming realistic energy evaluations and oversimplify their evaluation with regard to energy-efficiency. This work aims at lowering the barrier for *realistic* energy evaluations. We focus on a generic application that simply transmits one packet using TinyOS Low Power Listening (LPL), which we evaluate using the WSN testbed *SANDbed*. Our results *disprove* some intuitive expectations. For example, we show that transmitting packets with a large payload can be cheaper in terms of energy consumption than a small payload. As packet transmission is part of almost any WSN application, the results shown are important to many WSN protocol evaluations. As an addition, we contribute our *lessons learned* by discussing the most important challenges and pitfalls we faced during our evaluation.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Design, Experimentation, Measurement

1. INTRODUCTION

For most future applications, the nodes in a Wireless Sensor Network (WSN) have to be small and cheap. As a direct consequence, the nodes are heavily constrained regarding energy, memory, and processing resources. Top priority of a WSN application engineer is therefore to pay attention to an applications' resource efficiency, especially to energy-efficiency.

Regarding energy-efficiency, previous work has two major drawbacks. First, protocols are evaluated in isolation, i.e., neglecting the influence of other protocols and network layers. Moreover, using simple *simulator* tools is the most popular way of evaluation. Some simulators, like TOSSIM [3], omit lower network levels, ignoring influence of the radio channel or the MAC protocol. Counting data packets or data volume and then estimating energy consumption is a common way here. An improvement is emulating the sensor node combined with a cycle level energy model, as it is implemented by AVRORA [4]. However, the results still depend on the accuracy of the energy model provided. *Formal* evaluations use abstractions to model the sensor network and the environment. Even simple formal models are hard to handle, e.g., when doing formal verification. Results from more realistic evaluations are mostly acquired by *measuring* the energy consumption of a single node using an oscilloscope or by deploying a test application on a WSN *testbed*. The first approach has the disadvantage, that the energy consumption of only a single node is measured, whereas in a sensor *network* there are commonly multiple nodes and also node interactions involved. We argue, that the energy consumption of a WSN node is a complex phenomenon, that is hard to evaluate in a precise manner, because it involves more than deducing it from the communication characteristic of an application. It is therefore best evaluated using a *real* wireless sensor network.

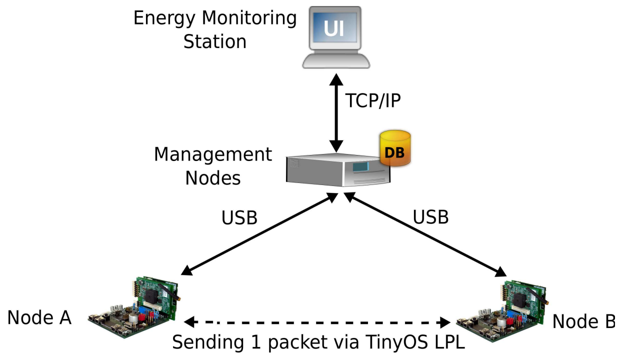
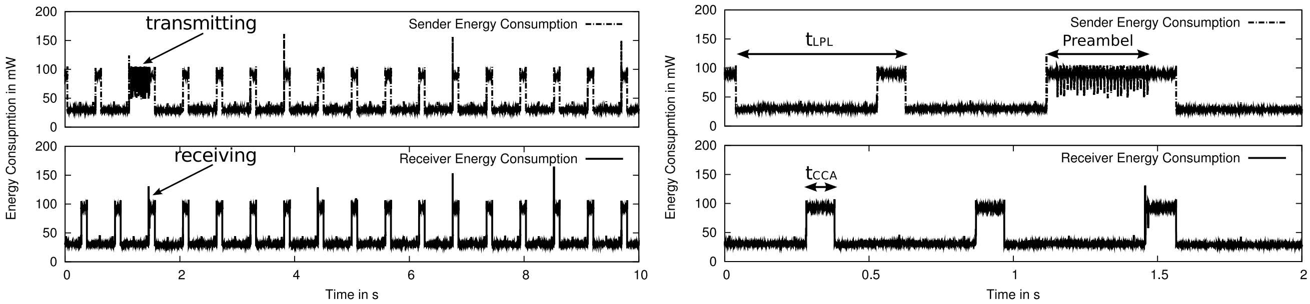


Figure 1: Experiment setup.

In this work, we use the WSN Testbed *SANDbed* as evaluation platform. It uses real hardware nodes to get trustworthy results with regard to energy consumption. Because communication is the most energy consuming task sensor nodes perform, we present evaluation results of simple communication between sensor nodes. We show how cross layer effects, especially from the MAC layer, influence energy consumption and what has to be considered to properly evaluate energy-efficiency of any WSN application. Additionally,

MSWiM'11, October 31–November 4, 2011, Miami, Florida, USA.



(a) Plot of a complete measurements of ten seconds.

(b) Zoomed plot of the transmission.

Figure 2: Energy consumption of two nodes (sender and receiver) while transmitting a single packet.

we contribute a set of pitfalls we faced during our evaluation and present best practices to evaluate energy-efficiency flawlessly.

2. EXPERIMENT SETUP

For a distributed measurement of energy-consumption, we use the Sensor Node Management Devices(SNMDs) at the SANDbed testbed at the KIT [2]. Each SNMD is equipped with a MICAz sensor node and its default sensor board. The SNMDs are capable of measuring the voltage and the current with a very high resolution as they provide sampling frequencies of up to 250kHz with an average measurement error below 1% [1].

We use TinyOS and enable its Low Power Listening mechanism as energy-efficient MAC protocol. The basic idea behind LPL is to periodically check the wireless medium for activity using a *Clear Channel Assessment* (CCA). The check is done every t_{LPL} milliseconds. Between these checks, the radio is powered off to save energy. If a node wants to transmit data, it continuously repeats its data packet for at least t_{LPL} milliseconds to assert the receiver had the possibility to wake up and receive the data. The length of the CCA check t_{CCA} is thus another important parameter, as the receiver must be able to detect the activity on the medium during this time slot.

Figure 1 shows the setup of our evaluation environment. We are using two measurement devices, Node A and Node B, to perform a distributed energy measurement. These two devices are SNMDs with an attached MICAz sensor node running TinyOS. The measurement devices are connected to a management node to store the evaluation results in a local database.

The application used for our reality check is very simple. One node sends exactly one data packet within a predefined time interval of 10 seconds to the other node using TinyOS LPL as MAC layer protocol. The sending time is randomly chosen by the sending node within the time interval. For evaluation purposes, we are measuring the energy consumption of both sender and receiver node. To get an idea about the influence of the LPL parameter t_{LPL}, we are varying the t_{LPL} time between 50ms and 4000ms. The length of the CCA check t_{CCA} is fixed. Further more, we are using two different payload sizes to identify differences in energy consumption. All experiments are repeated 100 times to get statistically significant results. In our evaluation later on, we choose random delays to power on nodes for every experiment run to avoid timing based effects. To cope with hardware tolerances, we are switching the sender and receiver role after 50 runs. Our experiment setup is summarized in Table 1.

Figure 2a shows the energy consumption of two nodes over time while sending and receiving a packet using TinyOS LPL. The SNMDs used to get the measurements in this plot sampled the node's current draw and voltage with 9kHz. The zoomed plot on the right side shows the transmission in detail. One can see the periodic CCA check and its duration t_{CCA} as well as the preamble of the sender. The time between two CCA checks is t_{LPL}. Having this plot in mind, it is obvious, that altering MAC parameters significantly influences energy consumption of both nodes without changing the application on top.

In each repetition we are measuring the energy consumption of sender and receiver node and whether the packet is transmitted successfully. Analyzing the above experiment setup, the sample application and related work regarding energy evaluation, one would intuitively expect the following outcome of our evaluation:

- The lower we set the LPL parameter t_{LPL}, the more energy is consumed by the two sensor nodes as a lower LPL parameter means shorter sleep intervals.
- With bigger payload sizes, one would expect a bigger overall energy consumption of the two nodes because of the higher data volume that has to be transmitted.

3. EVALUATION

In this section we analyze the influence of t_{LPL}, the influence of payload size and the influence of radio timing effects on the energy consumption of sender and receiver nodes.

MAC protocol	TinyOS LPL
Packets transmitted	1
CCA check length t_{CCA}	1600cycles ($\approx$ 11ms)
LPL parameter t_{LPL}	$50, 100, 250, 500, 750$ms $1, 1.25, 1.5, 2, 2.5, 3, 3.5, 4$s
Payload size S	$1, 90$Bytes
Experiment duration	10s
Experiment repetitions	2x 50
Sampling rate	9kHz

Table 1: Experiment setup parameters

3.1 Influence of t_{LPL}

In Figure 3, the energy consumption of sender and receiver node while transmitting 1 Byte of payload using different values for t_{LPL} is shown. The energy consumption is thereby averaged over 100 experiment repetitions. According to our evaluation, sending is always more expensive than receiving. Moreover, the energy consumption of the receiving node decreases with the size of t_{LPL}. As for the sending node, higher values of t_{LPL} do not automatically lead to lesser energy consumption. For our experiment, there is an optimal value for the t_{LPL} parameter for the sending node at approx. 2000ms. This effect can be explained with the mechanism TinyOS LPL uses for sending packets: If the t_{LPL} value is high, the sending node needs to send a preamble that covers at least t_{LPL} ms, whereas the receiver only has to perform a relatively short CCA-Check of approx. 11 ms. Therefore, the energy used for sending the preamble dominates the energy saved by powering off the radio between two CCA checks.

As our example shows, an energy-efficient MAC layer has a high influence on the overall node energy consumption. Any evaluation regarding energy consumption and communication therefore has to include a precise model for the energy consumption of the MAC layer.

3.2 Influence of radio timing effects

Figure 3 also shows the minimum and maximum values we recorded for every configuration during the 100 experiment repetitions. As stated in the previous section, the nodes started each experiment with random radio startup times, so that the CCA checks are not performed synchronously. Additionally, the sending time of the packet is randomly picked by the sending node. As can be seen, the energy consumed by the sending node has a high fluctuation, depending on the concrete sending time in conjunction with the receiving nodes CCA check. If the sending node transmits the packet shortly after the receiving node has done its last CCA check, more energy for transmitting a longer preamble is needed for the sending node.

The evaluation clearly shows that timing effects highly influence the nodes energy consumption. Also, the energy consumption of one particular experiment run can strongly deviate from the average energy consumption. In our sample application, the sending nodes energy consumption can fluctuate by ± 50 percent between the minimum and maximum value.

3.3 Influence of payload size S

As can be seen in Figure 4a, the receiving node consumes the same amount of energy for both payload sizes of 1 and 90 Byte. For the sending node, transmitting 90 Byte payload is slightly less energy consuming than transmitting 1 Byte of payload. This is caused by the *CC2420 transceiver* of the MICAz sensor node; the transceiver consumes less energy in the sending state than in listening or receiving state. With bigger payload sizes, the sending nodes transceiver is longer in the sending state than with lower payload sizes during the preamble phase of LPL.

Despite the smaller energy consumption of the bigger payload, it is clearly shown that the MAC layer parameter t_{LPL} has way more impact on the nodes energy consumption than the payload size.

Our evaluation clearly shows that an evaluation of energy consumption by simply comparing the data volume transmitted is incomplete or even leads to wrong conclusions. In our simple application, a node that is sending more data would be the slightly more energy-efficient node.

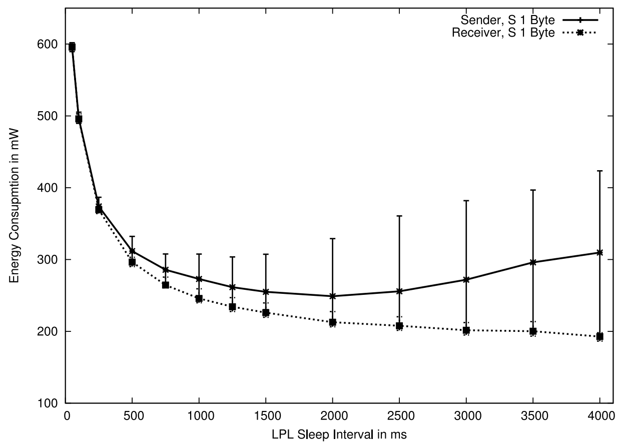


Figure 3: Influence of MAC parameter t_{LPL}.

3.4 Influence of hardware tolerances

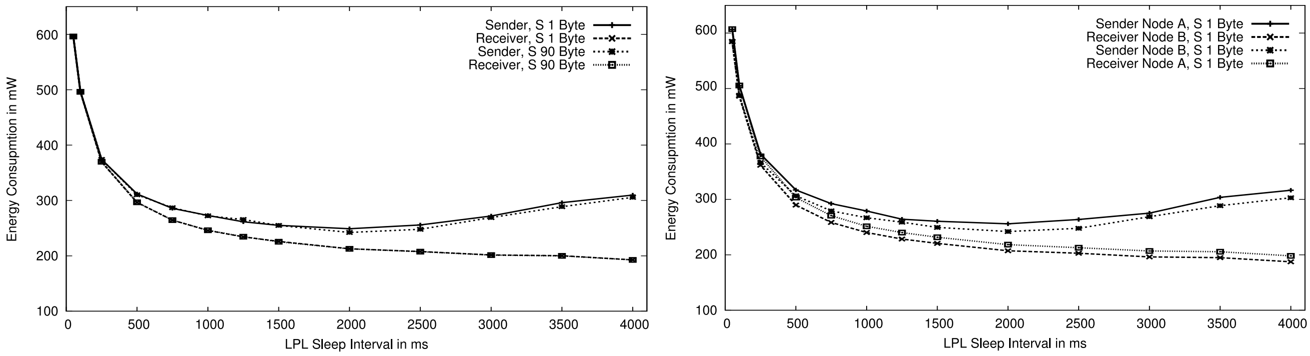
Figure 4b shows again the influence of the t_{LPL} parameter on the nodes energy consumption. In contrast to Figure 4a the first 50 runs and the second 50 runs have been *splitted* and not aggregated as before. This way, the first part of the experiment, using node A as sender and node B as receiver can be compared with the second part where the node's roles are switched. The energy consumption differs between node A and node B by an approximately constant amount even while having the same role. In this case the energy consumption of node A is always a bit higher than that of node B.

To compare nodes with different roles with each other, roles *must* be switched to average energy consumption results. Without that, only comparisons of results measured at the *same* node are possible which constraints evaluation possibilities.

4. LESSONS LEARNED

During our evaluation, we identified several pitfalls and challenges that are important when evaluating energy-efficiency of WSN nodes. In a WSN, testbed nodes can often be powered on *simultaneously.* This is, however, no realistic behavior. In reality, nodes are powered on *asynchronously* depending on the deployment method. Moreover, clock drift and radio communication can influence the radio timing. As a result, to evaluate both the startup phase of a WSN application as well as a period of time during WSN application operation, nodes have to powered on in a *pseudo-random manner* to avoid timing based effects. This also means, that there have to be many runs of the same experiment using different seeds for the pseudo random number generator, to average results. In our evaluation, we choose random delays to power on nodes for every experiment run to avoid trapping timing based effects.

A large number of experiment runs not only consumes time, but requires the nodes to be powered off and on many times. During our experiments we noticed some weird be-



(a) Payload size S (b) Hardware tolerances

Figure 4: Influence of other parameters on energy consumption.

havior using MICAz nodes. The first one is, that the startup phase differs in energy consumption and duration from node to node. The second one is, that the current consumption of the node is often fluctuating during the very first seconds. As a result, we have put a waiting phase between powering on the nodes and starting the measurement, to avoid this fluctuating period.

Even after the startup phase, we still noticed significant offsets regarding energy consumption of different nodes while running identical operations. Further experiments showed that energy consumption of the 22 MICAz nodes in our testbed differs up to $\pm 10\%$. As a direct consequence for the evaluation, we switched the roles of the two used nodes, to average hardware tolerance effects. It is important to know about the existence of hardware tolerances and its range, as sensor nodes deployed in real environments will also have such tolerances. As a result, comparisons of absolute energy consumptions are almost impossible. Only relative comparisons seem feasible. This is, however, the important one to support design decisions as what protocol to use for a certain operation purpose.

Another issue is *when* to start or stop measurements. If an event, like a packet transmission, has to be evaluated, an event based starting and stopping of energy measurements would be the best choice. However, when evaluating *distributed events*, which even a simple packet transmission is, the challenge is how to start and stop such an event based measurement on multiple nodes simultaneously. Another possibility is to start and stop measurements on a fixed time based schedule. This is, however, only possible if the evaluated application is adapted to the evaluation's needs to make start and stop time predictable. In our evaluation we have decided to use the second approach, as our example application is simple enough to predict start time and measurement duration a priori.

Compared to a conventional simulative evaluation, a realistic energy evaluation is much more time consuming, because it runs in real time. Summarizing duration of energy measurement, node preparation (e.g., flashing nodes) and waiting times (e.g., node startup phase), the evaluation required several days of continuous operation in our testbed. It is also worth noting, that the amount of data collected was quite big. We collected several gigabytes of raw measurement data consisting of 16bit current and voltage values over time using a sampling frequency of only 9kHz.

5. CONCLUSION

Evaluating energy-efficiency of WSN protocols still remains a challenging tasks. However, the results of real measurements are worth time and effort. Our evaluation of a basic scenario revealed several interesting results. First, we disproved the common belief that energy consumption and data volume correlate. Instead we showed, that a larger data volume can even reduce energy consumption in some cases. Second, we pointed out the impact of t_{LPL} on overall energy consumption, as an example for the MAC layers influence on energy-efficiency. Last, we summarized challenges and pitfalls one should be aware of when evaluating energy-efficiency using real hardware.

Up to now, no simulator tool considers hardware tolerances and their consequences to energy consumption and network lifetime. As a result, future work will include an improved version of AVRORA that implements this behavior. We also plan to extend our scenario to include concrete transport and security protocols and other MAC layers, like S-MAC.

6. REFERENCES

[1] A. Hergenröder and J. Horneber. Facing Challenges in Evaluation of WSN Energy Efficiency with Distributed Energy Measurements. In *Proceedings of Multihop Wireless Network Testbeds and Experiments Workshop (TESTBEDS)*, Istanbul, Turkey, July 2011.

[2] A. Hergenröder, J. Wilke, and D. Meier. Distributed Energy Measurements in WSN Testbeds with a Sensor Node Management Device (SNMD). In *Workshop Proceedings of the 23th Conference on Architecture of Computing Systems (ARCS)*, pages 341–438, Hannover, Germany, Feb. 2010. VDE Verlag.

[3] 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 Conference on Embedded networked sensor systems (SenSys)*, page 188, New York, USA, Nov. 2004. ACM Press.

[4] B. L. Titzeri, K. D. Lee, and J. Palsberg. Avrora: Scalable Sensor Network Simulation with Precise Timing. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN)*, page 67, Piscataway, NJ, USA, 2005.