Power Management for Wireless Sensor Networks Based on Energy Budgets

Christian Renner
Institute of Telematics
Schwarzenbergstraße 95E
21073 Hamburg, Germany
christian.renner@tu-harburg.de

Stefan Unterschütz
Institute of Telematics
Schwarzenbergstraße 95E
21073 Hamburg, Germany
stefan.unterschuetz@tu-harburg.de

Volker Turau
Institute of Telematics
Schwarzenbergstraße 95E
21073 Hamburg, Germany
turau@tu-harburg.de

ABSTRACT
This paper proposes and assesses analytical tools for large-scale monitoring applications with wireless sensor networks powered by energy-harvesting supplies. We introduce the concept of an energy budget, the amount of energy available to a sensor node for a given period of time. The presented tools can be utilized to realize distributed algorithms that determine a schedule to perform the monitoring task and the inherent communication. Scheduling is based on the energy budgets of the nodes or on latency requirements. In this context, we derive theoretical results for the energy consumption of the individual nodes plus the latency of event-reporting. These results are verified by simulations and a real testbed implementation.

1. INTRODUCTION
Wireless Sensor Networks are often advertised as networks that can run almost without any maintenance in difficult to access places. In reality most deployed sensor networks rely on batteries that need continuous replacement, even when operating at a low duty cycle. The usage of batteries in wireless sensor networks is primarily due to avoiding the wiring cost, i.e., data cables as well as power cable can be eliminated. Energy harvesting as a replacement for batteries can be a way towards the goal of a maintenance-free sensor network. Unfortunately, many regenerative energy sources do not continuously provide energy. Consider a solar cell as an example. The power generated by the latter varies significantly with the time of day, the time of the year, temperature, cloud cover, and both latitude and longitude. To provide an everlasting power supply in this case, the energy must be buffered. Thus, during times when an energy harvesting module does not provide energy, a sensor node must get along with the energy stored in its buffer. In the following the energy stored in a buffer is called the *energy budget*.

A sensor network must be able to provide a service with a given quality over a period of time relying solely on the energy budgets of the individual nodes. Deciding whether this is possible, is a very demanding challenge. As K. Kant pointed out in [9], the very heart of the problem is that currently there is no such thing as the science of power management. Such a science would embrace abstract models for energy storage and energy consumption including an energy calculus. These instruments can be used to verify whether a specific individual or collaborative task can be accomplished based on given energy budgets of the nodes of a sensor network. The real potential of an energy calculus is the adaptation of specified task such that it can be accomplished with a given energy budget.

There are two principle possibilities to adapt a task to a fixed energy budget. The first option is to optimize the implementation of the task. Available instruments for this option include dynamic voltage scaling, shutdown of unused components, low power modes, aggregation of idle periods into larger ones, and batching of operations on energy-hungry devices. The second option is to change the semantics of a task according to some contract. As an example consider the task of monitoring of a larger area for the occurrence of a given phenomenon, e.g., the leakage of gas along a gas pipeline. Each node periodically activates its sensor to measure a value, transforms the value with an A/D converter and interprets the result. If a leakage is detected, a message is sent in a multi-hop fashion to a central node to raise an alarm. The contract for this application may include a range for the latency of an alarm message and a range for the precision of the measurement to detect the leakage of gas. This contract allows for several possibilities to embank the energy consumption:

- Reducing the sampling rate of the sensor nodes
- Lowering the precision of the A/D converter
- Extending the sleeping periods of the transceiver modules

The challenge at hand is to find values for these parameters, such that the application can run for T_{life} time units with the given energy budgets, while minimizing the latency and maximizing the precision of the gas detection. After T_{life} units of time the energy harvesting module will deliver energy again.

Technical Report, URN: `urn:nbn:de:gbv:830-tubdok-11065`
This research has been partially funded by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG), contract number TU 221/4-1

The above example clearly illustrates that power management for wireless sensor networks based on energy budgets is a very challenging task. The purpose of this work is to tackle this challenge for a restricted setting. The paper considers a generalization of the above described event-reporting task. We provide analytical tools in order to devise an algorithm for this setting. These tools are evaluated using simulations and a real sensor network application.

This paper is structured as follows. Section 2 describes current approaches to maximize the lifetime of battery-powered wireless networks. Afterwards the problem statement for this paper—providing a responsive and energy-aware communication scheme—is introduced in Sect. 3. In this context, the considered scenario is described along with introducing the used BoX-MAC-2 protocol. An analytical assessment is performed in Sect. 4. The analytical derivations are subsequently supported by simulations, presented in Sect. 5. In addition to this, we will provide real-world results in Sect. 6. We also outline and discuss possible future research directions in Sect. 7, before we finally draw the conclusion in Sect. 8.

2. STATE OF THE ART

In recent years quite a few efforts have been undertaken to maximize the lifetime of battery operated wireless sensor networks. These works can be classified as follows:

Low-power protocols: New MAC and routing protocols have been proposed that minimize energy consumption. Most of this work has its focus on minimizing the transmission power or the duty cycle, i.e., the ratio between on- and off-time of the transceiver. The main techniques are TDMA, scheduled contention, and low power listening [17, 5].

Power-aware protocols: Power-aware protocols attempt to achieve a homogeneous energy consumption over all nodes of a network in order to prevent a premature outage of parts of the network. A typical approach is to favor nodes in routing decisions that have an energy level that is above the average energy level. An example of this type of protocol can be found in [14].

Using such protocols allows to run battery-powered sensor networks for several months depending on the application. A permanent, maintenance free operation of a sensor network requires the usage of regenerative energy sources. The above mentioned protocols are still applicable, but on top of that a power management taking into account the characteristics of regenerative energy sources is required. In particular it must be observed that these sources do not continuously provide energy, but only at specific times. Thus, the concept of energy budgets is needed.

Research on using self-sustaining energy sources for wireless sensor nodes has just begun. Voigt et al. reported in 2003 about sensor networks where some of the nodes are solar powered [16]. The authors present a power-aware protocol which is a variant of Directed Diffusion. The basic idea is that gradients flow preferably through solar powered nodes. The network does not perform a power management in the above sense.

To the best of our knowledge the usage of energy budgets has not been considered in the literature. In fact none of the sensor networks discussed in the survey about solar powered sensor networks by Jeong et al. makes use of such a concept [6]. The systems that perform a rudimentary form of power management in the above sense are Prometheus and Heliomote. Prometheus, also employing a solar cell, is based on a two-stage energy storage system [7]. A supercapacitor serves as the primary energy source, which supplies the sensor node and limits access to the secondary source, a rechargeable Li$^+$ battery, to prolong the lifetime of the latter. Charging and discharging behavior of the circuit and supercapacitors are examined. Prometheus also takes a first step towards energy-aware scheduling by adapting the duty cycle to the current supercapacitor voltage. Prometheus has been successfully deployed in the Trio testbed [4]. Heliomote is a system consisting of a solar cell and two rechargeable NiMH-batteries [11]. A node is aware of the energy state of its batteries and its solar cell. This information is distributed in the network and allows individual nodes to perform a power management. Specific algorithms are not presented.

A first approach towards an energy neutral operation of sensor networks was presented by Kansal et al. [8]. They have developed abstractions to characterize the complex time varying nature of self-sustaining energy sources with analytically tractable models including an energy prediction model.

3. PROBLEM STATEMENT

The environment under consideration is an event-detection wireless sensor network $\mathcal{V}$ with a central, dedicated data processing node v_0, the *data sink*. All other nodes v_i except for the sink are powered by a rechargeable supply, e.g., a supercapacitor. The amount of energy stored is E_i, which is replenished with a period of T_{life}, e.g., by a solar cell. Sensor nodes v_i are spread over the area of interest and can communicate wirelessly. Due to limited communication range, only some nodes can exchange data directly.

3.1 Scenario

As not all nodes can communicate directly with the sink, routing is required. We employ tree routing as the method of choice. This approach is highly suitable for the many-to-one nature of the environment under consideration, because it exploits the predictable and directed traffic pattern and is thus energy-efficient by reducing the overall number of packets sent. As we are not out for evaluating tree construction, we assume a static, pre-calculated tree. Each node v_i is aware of the size N_i of its subtree and its parent.

In order to simplify matters, it is assumed, that each node v_i is equipped with a single sensor that takes a sample with constant period T_σ. Note that T_σ is fixed in order to enable reliable event detection and cannot be changed thus. If the value of a sample exceeds a critical threshold, this is called an event. In this case, the node prepares an event packet and forwards it into the direction of the sink by sending the packet to its parent in the tree. We assume that event detection per node is a Poisson process with an inter-arrival time of λ. In order to minimize protocol overhead, end-to-end reliability is not enforced, i.e., event-reporting from origin to the sink is best-effort.

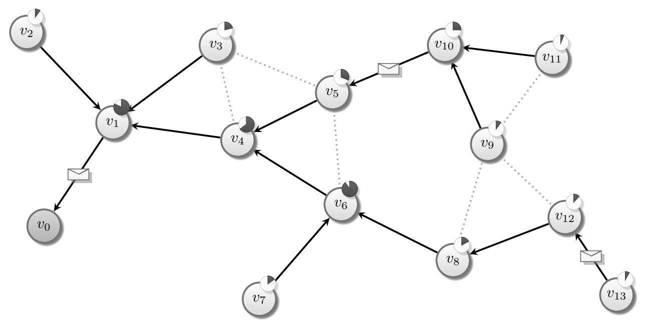


Figure 1: Event detection scenario with data-gathering tree and current energy capabilities

Figure 1 gives an example overview about the scenario just described. Possible communication links are indicated by dotted lines and the routing tree with sink v_0 is represented by directed edges. Envelopes display active packet transmissions and the current energy level of a node's buffer is visualized via a simple pie chart at the right top corner of each node—the light part symbolizes the remaining energy.

3.2 Medium Access Control

As the radio is one of the major consumers of energy in wireless sensor networks, we aim at choosing an energy-efficient MAC protocol that minimizes protocol-inherent communication overhead while maintaining low packet delay. To meet these ends, nodes employ a CSMA-based, energy-efficient MAC protocol called BoX-MAC-2 [10] as to allow for spontaneous communication upon event detection while preserving energy by radio duty-cycling. The protocol is an improvement over both well-known B-MAC and X-MAC [2] in terms of energy-efficiency and packet delay. Its function is explained in the following.

Each node in the network employs periodical radio duty cycling: The radio is off for T_{slp} time units and on for T_ℓ time units, where the latter must be large enough to ensure reliable packet detection and reception. If a node wants to transmit a radio packet, it switches on the radio for at most $T_{\mathrm{slp}} + T_\ell$ time units and repeatedly tries to deliver the packet. Before each transmission, clear channel assessment (CCA) and random backoffing are performed, which we assume to last for an average of T_{idle} time units. After each single packet transmission, the sender briefly waits T_{ack} time units for an acknowledgment (ACK). If the intended receiver picks up the packet, it sends an ACK packet, upon whose reception the sender stops the repeated transmission. Optionally, both nodes may stay awake after the completed transmission process to enable follow-up transmissions as to decrease delay. If a node overhears a packet destined to another node, it immediately switches off its transceiver and discards the packet. In case the sender does not receive an ACK, it stops transmission after $T_{\mathrm{slp}} + T_\ell$ time units, discards the packet and switches off its transceiver.

A sample packet transmission from a sender to an intended receiver is illustrated in Fig. 2. Particularly note that retransmissions are stopped after the ACK.

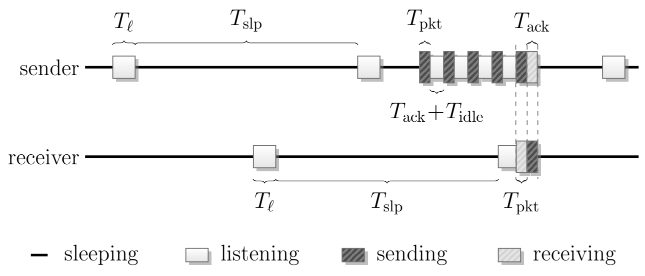


Figure 2: BoX-MAC-2 example with one sender and one receiver

3.3 Practical Perspective

In a wireless sensor network scenario as portrayed above, there are basically two design criteria: The event-reporting delay and the available energy budget, i.e., the capacity of the energy buffer. Both of these quantities lay restrictions on the radio duty-cycle and are therefore tightly interrelated. Yet, their interconnection and the dependency on the sleeping time are not obvious, e.g., a larger sleeping time reduces the energy consumption for listening and the channel but also increases the average cost for delivering a packet to the parent node. Hence, further analysis is required in order to answer the two questions of striking interest to the system designer: What is the expected latency, if the capacity of the energy buffers is restricted? And, what is the minimum capacity of the energy buffers required for guaranteeing a specified maximum packet delay?

4. ANALYTICAL ASSESSMENT

In the following, we will derive theoretical models for event-reporting delay and energy expenditure caused by event detection and packet forwarding. For the following assessment, we assume that the sampling period T_σ is small enough to ensure reliable event detection and does not influence latency, i.e., $T_\sigma \ll T_{\mathrm{slp}}$. By claiming $\forall v_i \in \mathcal{V} : \frac{\lambda}{N_i} \gg T_{\mathrm{slp}}$, received and created alarm packets can be forwarded directly, i.e., packets do not have to be queued, and predicting packet delay is simplified. We neglect the influence of faults as to facilitate all calculations. In particular, we do not consider packet loss—which we project to scarcely occur in the case of rare events and therefore low chance of packet collisions.

In general, BoX-MAC-2 allows for different sleeping times among nodes. In this case, however, nodes have to know the sleeping time of their parents in the routing tree as to make sure their transmission period is large enough for parent-side packet detection and reception. Besides resulting protocol overhead to exchange sleeping time information, this feature would render analytical assessment more complex. Since this work is our first effort on this matter, we're not making use of this possibility at this stage of theoretical analysis, but leave it up as prospect future work.

4.1 Event-Reporting Delay

The per-hop delay of a single event packet is uniformly distributed in the interval

$$\Delta = [T_{\mathrm{pkt}}, T_{\mathrm{pkt}} + T_{\mathrm{delay}}] \,, \tag{1}$$

where $T_{\text{delay}} = T_{\text{slp}} + T_{\text{pkt}} + T_{\text{ack}} + T_{\text{idle}}$. The according probability density function is

$$\rho^1_\Delta(t) = \begin{cases} \frac{1}{T_{\text{delay}}} & T_{\text{pkt}} \leq t \leq T_{\text{pkt}} + T_{\text{delay}} \\ 0 & \text{else} \end{cases} \tag{2}$$

if no packet loss is observed. The minimum delay is due to packet transmission under the assumption that the packets become available to the application right after reception, i.e., the ACK is sent afterwards. The maximum delay is reached, if the sender starts its first transmission coinciding with the begin of the receiver's sleeping interval and starts one transmission directly before the next listening phase of the receiver, so that the latter starts reception not before $T_{\text{pkt}} + T_{\text{ack}} + T_{\text{idle}}$ time units within its listening period have elapsed.

Subsequently, the k-hop delay Δ^k of a radio packet is given by $\Delta^k = k \cdot \Delta$, if processing delays can be neglected. The corresponding probability density function $\rho^{\text{k}}_\Delta(t)$ is defined by recursive convolution:

$$\rho^{\text{k}}_\Delta(t) = \rho^{\text{k}-1}_\Delta(t) * \rho^1_\Delta(t) \qquad (k > 1). \tag{3}$$

As a result, Δ^k has a mean of

$$\mu^{\text{k}}_\Delta = \frac{k}{2} \cdot (T_{\text{delay}} + 2 \cdot T_{\text{pkt}}) \tag{4}$$

and minimum respectively maximum values

$$\min\left\{\Delta^{\text{k}}\right\} = k \cdot T_{\text{pkt}}, \tag{5}$$

$$\max\left\{\Delta^{\text{k}}\right\} = k \cdot (T_{\text{pkt}} + T_{\text{delay}}). \tag{6}$$

Example probability density functions for $k = 1, 2, 3, 4$ are shown in Fig. 3. They clearly reveal the expected convergence of the probability density function towards a normal distribution.

The main outcome of the analysis presented afore, is that—since packet and ACK length are fixed in size—the delay can only be reduced by decreasing T_{slp}.

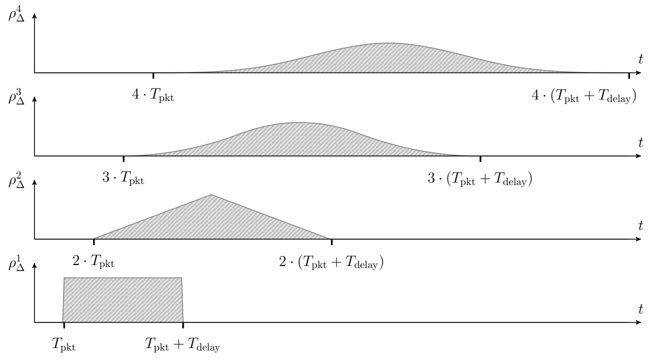


Figure 3: Probability densities for packet delay with different path lengths (hops).

4.2 Energy Consumption

The energy consumption of a node can be split into different components that will be explained in the following.

As stated in Sect. 3.1, sensor samples are taken in fixed intervals with length T_σ, so that a static amount of energy E_σ is spent on this task.

Each node v_i in the network has to forward all packets produced in its subtree plus its own ones. The average number of received packets sums thus up to

$$N_{\text{rx}} = (N_i - 1) \cdot \frac{T_{\text{life}}}{\lambda} \tag{7}$$

and yields the energy required for receiving packets, including the sent ACKs:

$$E_{\text{rx}} = N_{\text{rx}} \cdot (P_{\text{rx}} \cdot T_{\text{pkt}} + P_{\text{tx}} \cdot T_{\text{ack}}) \tag{8}$$

The number of packets sent by each node v_i is a little more complex to determine. Recall from Sect. 3.2 that a sender retransmits the packet until receiving an ACK. Thus, it will send unheard packets during half of the sending period T_{slp} on average, before finally delivering the packet to the receiver. Within T_{life} time units, the total number of sent packets therefore evaluates to

$$N_{\text{tx}} = N_i \cdot \frac{T_{\text{life}}}{\lambda} \cdot \frac{1}{2} \cdot \frac{T_{\text{slp}}}{T_{\text{pkt}} + T_{\text{ack}} + T_{\text{idle}}}. \tag{9}$$

Analogously to (8), the energy expenditure for sending is given as

$$E_{\text{tx}} = N_{\text{tx}} \cdot (P_{\text{tx}} \cdot T_{\text{pkt}} + P_{\text{rx}} \cdot (T_{\text{ack}} + T_{\text{idle}})) \tag{10}$$

Next, the listening costs are calculated. For this purpose, we count the number of listen-sleep cycles per period T_{life} and subtract half of the receptions and transmission cycles from this number. If a node receives a packet, it will—on average—only need half the listening time for detecting the packet. If a node is trying to deliver a packet, it will skip one listening period in the corresponding cycle with approximate chances of 50%.

$$E_\ell = \left(\frac{T_{\text{life}}}{T_{\text{slp}} + T_\ell} - \frac{T_{\text{life}}}{\lambda} \cdot \frac{2N_i - 1}{2}\right) \cdot T_\ell \cdot P_{\text{rx}} \tag{11}$$

Although we have explicitly calculated receiving costs, we do not consider energy consumption of packet overhearing, because the amount of overheard packets can hardly be predicted. In addition, the average energy of a single reception, as can be deduced from (8), does not vary greatly from that of a single listening period.

The rest of the time a node is sleeping. While in many works sleeping costs are neglected, this would be fatal in a setup with long sleeping periods and relatively few packets to be sent. However, by assuming $P_{\text{slp}} \ll P_{\text{tx}}, P_{\text{rx}}$ it suffices to simply account sleeping costs throughout all T_{life} time units, so that

$$E_{\text{slp}} = T_{\text{life}} \cdot P_{\text{slp}}. \tag{12}$$

In conclusion, if the following equation holds, no node v_i will run out of energy during the required T_{life} time units:

$$\forall v_i \in \mathcal{V} \setminus \{v_0\} : E_i \geq E_\sigma + E_{\text{rx}} + E_{\text{tx}} + E_\ell + E_{\text{slp}} \tag{13}$$

Assuming a common energy budget $E_i = E$ for all nodes $v_i \in \mathcal{V}$, the system of equations reduces to a single equation for the node with the largest subtree:

$$E \geq \max_{v_i \in \mathcal{V} \setminus \{v_0\}} \{E_\sigma + E_{\mathrm{rx}} + E_{\mathrm{tx}} + E_\ell + E_{\mathrm{slp}}\} \qquad (14)$$

Due to the complexity of the equations, we do not provide an explicit presentation. Depending on the given and unknown parameters, the equations can be solved with analytical software, such as Maple.

4.3 Application Benefit

The knowledge gathered in the previous part of this section empowers the system designer to finally answer the questions risen in Sect. 3.3:

Dimensioning energy budgets: In case latency bounds exist, The first step is to derive T_{slp} from either (4) or (6), depending on the needs. Next, the minimum capacity of the energy buffers to be used is deducible from (14).

Estimating event-reporting delay: If, in contrast, E is restricted for any arbitrary reason, (14) must be solved for T_{slp} first. The obtained quadratic equation gives a range of possible solutions. Taking into account the requirement of minimizing T_{slp} (see Sect. 4.1), the smallest value of this range is the desired solution.

5. SIMULATIONAL COMPARISON

In order to confirm the results of the theoretical analysis, we examine energy consumption and packet delay by simulating several networks. The instrument of simulation allows for comprehending dynamic aspects, such as channel utilization or packet forwarding in specific topologies. However, to obtain comparable results, faults like packet loss are not considered, which is equivalent to the made assumptions of the theoretical analysis.

5.1 Simulation Environment

For our experiments, we have developed a dedicated simulation tool. The first choice would be the TinyOS simulator TOSSIM or NS-2, because the former provides a broad platform support and both are well-established in academic research. Unfortunately, the BoX-MAC-2 protocol is not implemented for neither of the two. In addition, the integration of an adequate energy availability and consumption model would result in massive changes of existing simulation tools. By using a dedicated program, overhead of unused components and side effects are reduced, so that we gain better access to simulation behavior and outcome.

Our simulation tool is implemented in C++. For a convenient and flexible usage the compiled code is accessible via the scripting language Python. Currently, the program consists of a discrete event scheduler, different communication protocols and an energy model. The latter affords the simulation of a supercapacitor and the corresponding leakage current, which was determined in [12]. For the experiments we employ a protocol stack containing an application-layer, a packet-queue, the BoX-MAC-2 layer, a physical layer, and a channel. The two last-mentioned layers are needed to simulate the packet propagation and latency. BoX-MAC-2 is implemented as explained in Sect. 3.2, while a follow-up

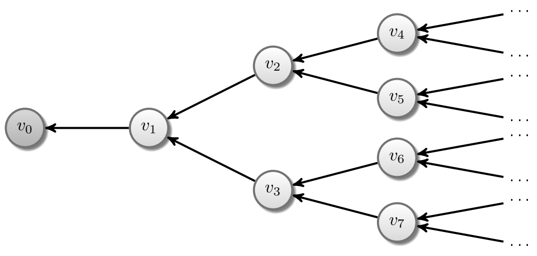


Figure 4: Used tree topology

transmissions takes place, if multiple packets need to be sent. Furthermore, the application protocol generates packets using a Poisson distribution. In addition, this layer provides a simple tree routing mechanism to a data sink, for which the parent of each node has to be defined at the start of a simulation. The queue below the application layer is necessary, since continuous receiving of packets may occur and own packets can be generated before completed transmission.

As the basis of the power and timing settings for the simulation serves the Crossbow IRIS node [3]. This device consumes an average sleep power of $P_{\mathrm{slp}} = 66\,\mu\mathrm{W}$, draws $P_{\mathrm{rx}} = 52\,\mathrm{mW}$ when receiving or listening and $P_{\mathrm{tx}} = 55\,\mathrm{mW}$ at 3 dBm when sending. These values are determined empirically. The platform uses an Atmel RF230 radio transceiver with 250 kbps [1]. We assume a packet payload of 20 bytes. Adding the headers required by the protocol stack, packets have a total size of 34 bytes. Acknowledgment packets consist of a 3 byte payload, leading to a packet size of 17 bytes. Therefore, we find $T_{\mathrm{pkt}} = 1.088\,\mathrm{ms}$ and $T_{\mathrm{ack}} = 0.544\mathrm{ms}$. A node waits at most 1 ms for an ACK and listens for $T_\ell = 6\,\mathrm{ms}$ on the channel. See [10] for the derivation of the latter value. In addition, we assume an average transmission overhead per packet of $T_{\mathrm{idle}} = 0.4\,\mathrm{ms}$, which we obtained from measurements. This value already includes the backoffing used by CSMA[1] and the switching times of the radio, which are at most 32 µs.

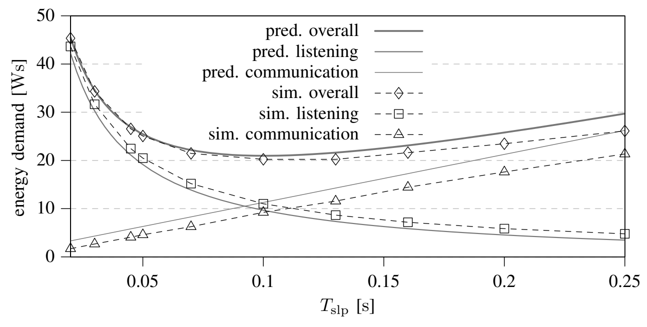
5.2 Verification of Analytical Results

For the verification of the analytical equations we made several simulations with different values for T_{slp} and λ. In all cases a simulation time of one hour is used for a network of 32 nodes organized as a binary tree as depicted in Fig. 4. Here, the node v_1 is connected to the data sink v_0 and is therefore the energetic bottleneck. For each set of parameters, 10 simulation runs were performed and the results were averaged. Note that on the basis of the analyses, the energy demand of a single node is affected by its subtree size and not by the present structure of the sub network. In contrast, latency depends mainly on the number of hops from the origin to the data sink.

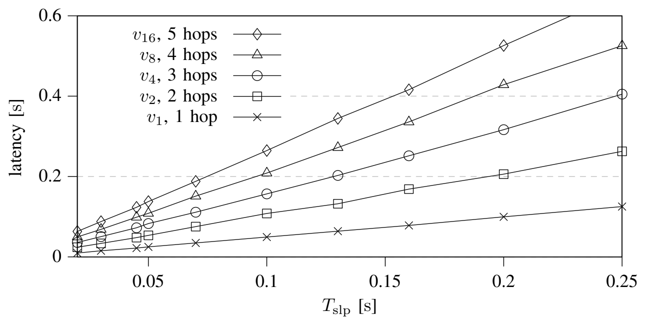
5.2.1 Examination of the Dependency on T_{slp}

In a first experiment we have varied the sleep time T_{slp} of the BoX-MAC-2 protocol between 30 ms and 250 ms. For each value the network was simulated with a fixed inter-arrival time of $\lambda = 30\,\mathrm{s}$. Fig. 5 shows the comparison between the simulation results and the analytically calculated curves. In

[1]The maximum backoff between packet resending is 0.32 ms.



(a) Energy demand vs. sleeping time



(b) Latency vs. sleeping time

Figure 5: Dependency of the energy demand and latency on sleeping time T_{slp}

Fig. 5(a) we distinguish between the overall energy demand of node v_1 (*overall*) and the share of idle-listening (*listening*) and communication (*communication*). For all three cases the analytical prediction (*pred.*) and simulative results (*sim*) are depicted. Fig. 5(b) presents the latency for sending a packet from the origin to the data sink. The latency is plotted for different nodes of the network.

The simulation and analytical results of the energy demand show a similar behavior in general; yet, for large values of T_{slp} notable variations occur. The reason is, that for long sleep periods the probability of multiple packets in a queue rises. These packets are sent in a follow-up transmission, which reduces the required energy demand. Furthermore, the necessary energy budget shows its minimum at $T_{\mathrm{slp}} = 100\,\mathrm{ms}$. For shorter sleep times the overall consumption rises due to the effect of idle listening. On the other hand, large values of T_{slp} lead to an increase of communication costs. The latter are mainly affected by the number of needed retransmissions until the parent node becomes awake and receives the packet.

The responsiveness of the network in terms of packet delay is proportional to T_{slp}. The expected one hop delay of $\frac{T_{\mathrm{slp}}}{2}$ plus the sending overhead, cf. (4), is approximately reached by node v_1. The figure also validates that multi-hop packet delay is the corresponding multiple of the one-hop delay.

5.2.2 *Examination of the Dependency on λ*

In a second experiment the influence of the inter-arrival time λ is examined for a fixed sleeping time of $T_{\mathrm{slp}} = 100\,\mathrm{ms}$. The results of the energy consumption in comparison to the analytical results is shown in Fig. 6(a), and the latency is shown in Fig. 6(b).

For small inter-arrival times, the simulation shows significantly better results than expected from the analysis. This is explained by the fact, that in our simulations the BoX-MAC-2 protocol makes extensive use of follow-up transmissions. Hence, the repetitive polling until the parent node wakes up is necessary only for the first packet. This case is not considered in the theoretical analysis. However, note that small values of λ violate the theoretical assumption that packets occur sporadically, so that the equations are not fully valid.

One might expect that follow-up transmissions also decrease packet latency, but for high utilization of the channel the resulting increase of queue waiting time cancels this effect. Indeed, the queue waiting time and therefore the latency will rise for further decrease of the inter-arrival time until the network collapses under the pressure of the high load.

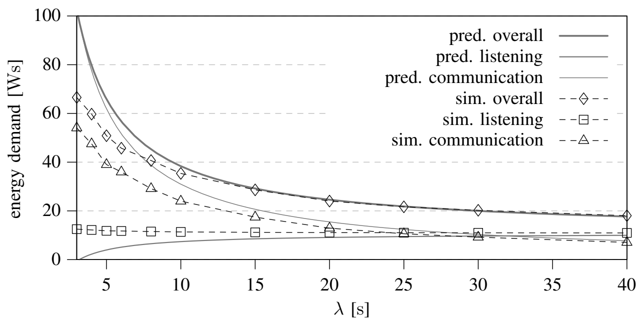
5.3 Conceptual Design of a Pipeline Monitoring Network

Finally, we demonstrate how to design and simulate a gas pipeline monitoring network, which was mentioned in the introduction. Consider a line topology of 100 nodes, in which the first node v_0 is the data sink. Events have to be reported to v_0 as fast as possible, while each node has a maximum energy budget of 5 Ws for one hour. We analyze this scenario for an inter-arrival time of events of $\lambda = 2\,\mathrm{h}$ and $\lambda = 4\,\mathrm{h}$.

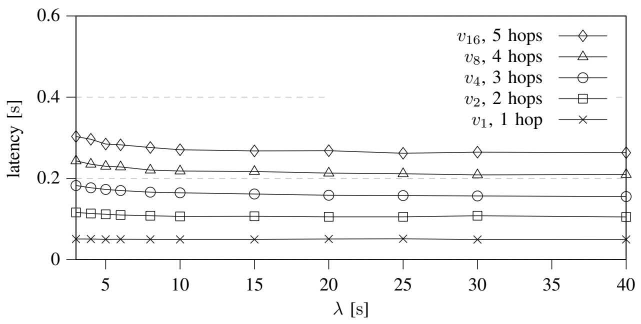
The equations of the previous section make it possible to estimate an optimal T_{slp} for the individual nodes. Fig. 7(a) shows the sleep time in dependency of a node's distance to the sink in hops for the two inter-arrival times. Note that the first node has the largest subtree size and must therefore use a larger T_{slp} to reduce the idle listing energy demand in order to compensate the higher communication costs. It can also be seen, that for a very low traffic load, T_{slp} becomes constant over subtree size, since energy expenditure for idle listening dominates the overall energy demand.

Based on the results drawn from Fig. 7(a) we simulated the network for 100 hours for the two inter-arrival times. Here, we analyzed two cases: Firstly, a constant T_{slp} for all nodes, determined by the value for v_1, was employed. Secondly, an adaptive assignment based on the corresponding subtree size was taken.

In Fig. 7(b) and 7(c) the energy consumption and latency are depicted for the different nodes in the line topology. In general, the energy consumption could be hold nearly constant for all nodes by using an adaptive T_{slp} assignment (abbr. *adapt*). If the same value of T_{slp} (*const.*) is used throughout the whole network, the energy consumption decreases with increasing distance from the sink. The sleeping time has also high influence on the latency. As expected the latency rises with the node distance to the sink. Notably, a performance enhancement of the adaptive T_{slp} assignment is only observable for a higher load. For $\lambda = 4\,\mathrm{h}$ the main



(a) Energy demand vs. inter-arrival time



(b) Latency vs. inter-arrival time

Figure 6: Dependency of the energy demand and latency on inter-arrival time λ

energy consumer is idle listening, so that the difference of T_{slp} between two adjacent nodes is marginal.

6. TESTBED EVALUATION

In order to justify and confirm our theoretical analysis developed in Sect. 4 for real hardware, we set up a testbed consisting of 7 nodes with a static topology configuration as displayed in Fig. 8. All nodes are taking measurements with a period of $T_\sigma = 1\,\text{s}$. Events occur with an inter-arrival time of $\lambda = 15\,\text{s}$ and are detected and sent at the next measurement. Events are software generated using a Poisson process, so that measurement is virtually not causing energy expenditure, i.e., $E_\sigma = 0\,\text{Ws}$. Note that we have chosen particularly small values for T_σ and λ for the sake of a short testbed run and therefore simplified evaluation.

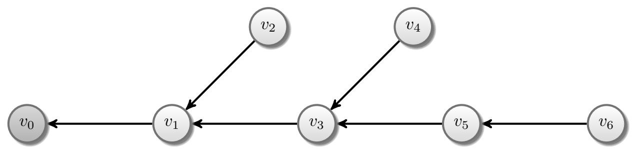


Figure 8: Testbed network topology

The Crossbow IRIS nodes [3] serve as our testbed platform (cf. Sect. 5.1). All nodes except for the sink are powered by a supercapacitor-based harvesting supply as described in [12]. The use of a supercapacitor simplifies the estimation of the remaining energy budget. The hardware is depicted in Fig. 9. A 25 F-rated SAMWHA GreenCap [13] with a measured capacity of 24 F was used. To preclude the influence of high leakage current, the initial voltage applied is 2.0 V. To supply a sensor node with a constant voltage of 3.3 V, a TPS61221 [15] boost-buck converter from Texas Instruments is used, which has an efficiency of 60 − 90% for the applied power profile. The efficiency depends on the input voltage and output current. We assume that the converter ceases to work when the input voltage drops below 1.0 V, which gives a small reserve to the actual cut-off voltage reported in [12]. The energy budget available to v_1 is hence

$$E_1 = \frac{24\,\text{F}}{2} \cdot \left(2.0^2 - 1.0^2\right)\text{V}^2 = 36\,\text{Ws} \qquad (15)$$

Note that the solar cell is detached in our test case.

Due to the power loss caused by the boost-buck converter,

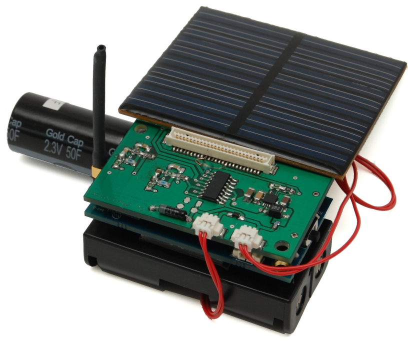


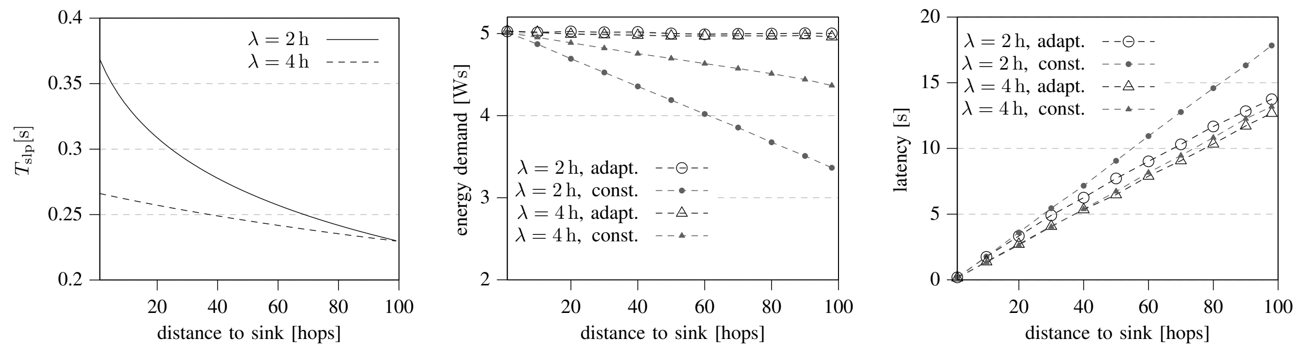
Figure 9: Energy harvesting power supply for the IRIS platform with a solar cell and supercapacitor energy buffer

the hardware consumes more energy than a bare IRIS node. Average sleep power increases to $P_{\text{slp}} = 110\,\mu\text{W}$. We measured approximately $P_{\text{rx}} = 75\,\text{mW}$ for receiving or listening and $P_{\text{tx}} = 85\,\text{mW}$ at 3 dBm for sending.

The software side of the story is TinyOS. It fully supports the IRIS nodes, makes massive use of the sleeping capabilities of both the microcontroller and the radio, and also allows for easy implementation of our test application. BoX-MAC-2 is enabled via using the LowPowerListening interface. We favored the current CVS version (as of October 2009) over the stable 2.1 branch, because the former contains some bugfixes regarding the BoX-MAC-2 implementation.

We make use of a special radio messaging mechanism of TinyOS to measure packet delay, thus unburdening of time synchronization. This mechanism enables the sink to determine packet delay in milliseconds, if the origin node attaches its local time of each detected event. Besides this information, each packet contains the identifier of the event's origin node, an event counter plus space for measurement values, and possible event meta data. Packet payload and header sizes are the same as assumed in Sect. 5.1, so that the values of T_{pkt}, T_{ack}, T_ℓ, and T_{idle} stay unaltered.

In order to achieve a lifetime of $T_{\text{life}} = 45$ min, we need to calculate the minimum required sleeping time T_{slp} for the introduced setup. As v_1 faces the highest traffic load in



(a) Pred. sleep time vs. node distance (b) Energy demand vs. node distance (c) Latency vs. node distance

Figure 7: Simulation of a pipeline monitoring network

	E_{tx}	E_{rx}	E_{ℓ}	E_{slp}
total energy [Ws]	1.654	0.119	33.850	0.249
relative energy of E_1 [%]	4.6	0.3	94.4	0.7

Table 1: Energy split of v_1. The gap of the sum of energies to (15) is due to a rounded T_{slp}

the network and thus consumes more energy than the other nodes, T_{slp} is obtained by reducing (13) to the case of v_1

$$T_{\mathrm{slp}} = 31\,\mathrm{ms} \tag{16}$$

The projected energies for sending, receiving, listening, and sleeping are summarized in Table. 1. Due to the relatively small values of T_ℓ as compared to λ, listening costs clearly dominate overall energy consumption. However, this is an expected situation for event-reporting applications with the requirement of low latency. Note that energy expenditure for sending is almost two orders of magnitudes larger than the one for receiving. Also note that sleeping costs cannot be neglected, as they exceed receiving energy.

6.1 Energy Consumption

Given the parameters as above, we have measured the supercapacitor voltage of node v_1 until reaching the critical value of 1 V in four distinct runs. The observed voltage values (solid lines) are depicted in Fig. 10 along with the derived amount of remaining energy (dashed lines). The plot also contains a smoothed approximation of the expected behavior of the supercapacitor voltage and remaining energy.

The figure indicates that v_1 behaves qualitatively as expected. The remaining energy is decreasing almost linearly with the same slope as the expected value, yet with a slightly super-linear behavior at the initial voltage and when approaching the cut-off voltage. Accordingly, the voltage falls in a square-root shape. However, all simulation runs exhibit a slightly short-falling runtime, i.e., the cut-off voltage is reached after durations between 41:35 min and 43:21 min. For these values, the predicted amount of required energy would be 33.1 Ws and 34.6 Ws respectively, so that the average power consumption of the hardware is between 4% and 8% above the estimation.

We believe that the remaining shortage in runtime can be explained by two major phenomena. Firstly, the steeper descent in remaining energy between measurements and projection hints at the influence of supercapacitor leakage current. As the latter decreases rapidly with falling supercapacitor voltage (cf. [12]), its impact vanishes with increasing experiment runtime. Secondly, the steeper descent when approaching the cut-off voltage is caused by a severely dropping efficiency of the boost-buck converter. This assumption is supported by a revisiting the data sheet of the TPS61221 and has been reaffirmed by measurements.

In addition to this, the hardware consumes a little more energy than expected. Tracking the current consumption of a node revealed that before each listening period, processor activity and radio switching caused an additional energy expenditure that we did not take into account. However, precise measurements have to be performed in order to analyze and quantify the caused effect.

On the contrary, v_1 consumed slightly less energy on the account of packet loss. We have analyzed the number of packets actually received and sent. For the given parameters of our experiments, the expected number of packets to be received and sent by v_1 would be $N_{\mathrm{rx}} = 900$ and $N_{\mathrm{tx}} = 1\,080$, respectively. These numbers were factually $N_{\mathrm{rx}} = 762$ and $N_{\mathrm{tx}} = 933$ in the third run. The saved energy for v_1 would thus sum up to 0.243 Ws, which is less than 1% of E_1 and would lead to an increased runtime of merely a few seconds. Only one packet of the ones sent from v_1 to v_0 was lost, so that the expected amount of energy wasted due the node's inability to premature radio shut off by an early ACK is negligible.

In conclusion, the testbed evaluation generally supports our analytical energy-consumption model, because the energy gradient follows the expected behavior closely. Yet, a few deficiencies have been encountered. The energy buffer and the used supply circuit must be known precisely. Furthermore, other sources of energy consumption apart from the radio and sensor sampling must be taken into account.

6.2 Latency

In order to verify the projections of packet delay derived in Sect. 4.1, we have measured packet delays for all received packets at the sink. The values have been obtained from

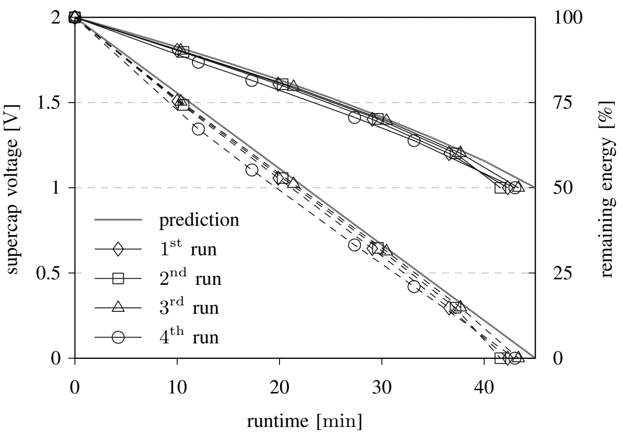


Figure 10: Supercapacitor voltage (solid) and available energy (dashed) vs. node runtime

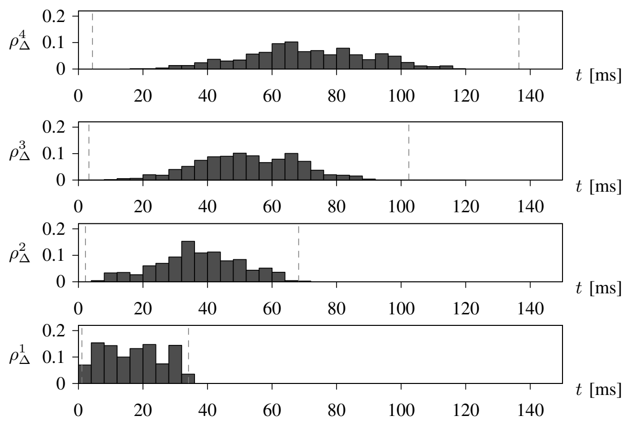


Figure 11: Observed probability densities for packet delay with different path lengths (hops)

	v_1	v_2	v_3	v_4	v_5	v_6
created	1 162	1 214	1 200	1 184	1 211	1 209
collected	1 159	1 133	1 164	1 037	1 102	761

Table 2: Number of created event packets per node and number of received ones at the sink v_0

a separate testbed run, in which we have exchanged the supercapacitor power-supplies by plain batteries to achieve a longer runtime of the experiment, yielding more samples and a more precise statistical evaluation of delay times.

The overall runtime of the experiment has been 5 h with an expected number of 1 200 packets per node. The actual number of generated event packets created per node and received by the sink is displayed in Table 2.

A packet-delay histogram is depicted in Fig. 11. The expected minimum and maximum values for $\min\{\Delta^k\}$ and $\max\{\Delta^k\}$ are $k \cdot 1.088$ ms and $k \cdot 34.120$ ms respectively, according to (5) and (6). Both values are represented in the figure by dashed gray lines. The results reveal shapes similar to those predicted in Fig. 3. However, for $k = 1$, particularly low delays have been observed more frequently than expected. This is due to a minor implementation deviation of BoX-MAC-2 in TinyOS: After successful packet transmission and reception, both nodes reschedule the next listening interval with a delay of T_{slp}, so that both nodes are almost synchronized. Hence, the chance of low delay at next reception is increased, especially in the case of the receiver being a parent to only few nodes, as in our testbed. For larger values of k, this behavior fades out. The histograms show, that the density functions are shifted a little to the right. This is caused by additional processing overhead on the sensor platform, that we did not take into account—e.g., the time needed to transfer the packet data from the processor to the radio chip via the SPI bus, or the initial backoffing.

7. FUTURE WORK

During our simulational and practical evaluation, we have identified several directions for future research and improvements. So far, only common energy budgets among all nodes have been taken into consideration. In a real environment budgets can be expected to vary, e.g., caused by node placement resulting in different energy harvesting abilities.

Related to this, nodes should be enabled to pick a sleeping time other than the one of their parent. While we have already simulated this approach, a port to the TinyOS implementation is currently not available. Furthermore, a dynamical adjustment of the sleeping time by each node at runtime should be enabled. Both requirements are due to varying energy budgets among nodes and changing harvesting conditions throughout a day or the year. This requires an exchange of sleeping time information from nodes with a parent-child relation in the routing tree for the following two reasons. Firstly, a sending node employing the BoX-MAC-2 protocol needs this information for retransmitting the packet sufficiently long enough for the parent to receive it. Secondly, the sleeping times of a node's children and its parent influence its own energy expenditure and therefore its allowed sleeping time.

In a real network, event and thus packet inter-arrival times may not be predictable when deploying the network and are also subject to change. Hence, nodes should run a prediction algorithm to estimate future traffic and energy consumption. It should also be studied, how applications with various measuring tasks and different latency requirements suit into the discussed scenario. In addition, adaptations of the routing tree may be required to ensure reliable packet transmissions and fair traffic load throughout the network—particularly in context of the individual energy budgets.

The practical use of the BoX-MAC-2 protocol can be viewed critically, since for large values of T_{slp} there is a nominal high channel utilization due to the employed retransmission technique. Especially, the co-existence with other communication services must be investigated, if using the same frequency-band (for instance an ISM band). One way to reduce the channel utilization of the MAC layer would be the use of a pull protocol similar to IEEE 802.15.4. In order to send data, a node listens to the channel until its parent wakes up. This is signaled by a beacon sent by the parent, so that the node knows when to start sending its packet. For this policy similar latency and energy consumption are expected in comparison to using the original BoX-MAC-2 protocol.

To further assess and refine the presented analytical results extensive field studies and more precise simulations are needed. They enable the evaluation of so far unconsidered effects, such as the influence of packet loss.

8. CONCLUSION

This paper shows, how responsive and energy-aware multi-hop networks for event-reporting systems can be realized. For this purpose, we have devised an energy-harvesting wireless sensor network employing the parameterizable BoX-MAC-2 protocol. The latter serves the purpose of reducing the duty cycle of the radio—the major energy consumer of today's wireless nodes—while maintaining latency requirements. For the examination of the considered scenario we performed an analytical assessment as well as a simulation and testbed evaluation of our findings.

The analytical assessment has brought forward a set of tools for the system designer. On the one hand, for given latency requirements, the needed energy budget of the network's nodes can be calculated. On the other hand, the expected average or maximum latency is derivable from a node's available energy budget.

Through simulations we proved the accuracy of the equations. The outcome gives a valuable insight into the system behavior in relation to the selected sleep time and existing network load. Packet delay rises roughly proportional with the sleeping time. For a given energy budget, an optimal value of T_{slp} can be identified. Based on these findings, we exemplary demonstrated, how to realize a pipeline monitoring network, such that the responsiveness is maximized for a given energy budget.

Results obtained in the testbed support the practical usability of the equations. Packet delays are in the projected ranges and exhibit the expected delay distribution. Energy consumption tightly meets the analytical results, although minor drawbacks have been identified. The hardware must be well known in terms of energy consumption. This applies to the sensor node itself, particularly regarding power consumption for sending and receiving plus the actual timings of these tasks. It also embraces the energy harvester, consisting of the energy source, the energy buffer, and the circuitry. For instance, leakage and efficiency of the components must be taken into account to guarantee correct results of the analytical methods presented in this paper.

9. REFERENCES

[1] Atmel Corporation. *AT86RF230 Datasheet* , Feb. 2009.

[2] M. Buettner, G. Yee, E. Anderson, and R. Han. X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks. Technical Report CU-CS-1008-06, Department of Computer Science, University of Colorado at Boulder, Boulder, CO, USA, May 2006.

[3] Crossbow Technology. *IRIS Wireless Measurement System – Datasheet, Rev. A.*

[4] P. Dutta, J. Hui, J. Jeong, S. Kim, C. Sharp, J. Taneja, G. Tolle, K. Whitehouse, and D. Culler. Trio: Enabling Sustainable and Scalable Outdoor Wireless Sensor Network Deployments. In *Proc. of the 5th Intl. Conf. on Information Processing in Sensor Networks (IPSN '06)*, Nashville, TN, USA, Apr. 2006.

[5] J. Hui, Z. Ren, and B. H. Krogh. Sentry-Based Power Management in Wireless Sensor Networks. In *Proc. of the 2nd Intl. Conf. on Information Processing in Sensor Networks (IPSN '03)*, Palo Alto, CA, USA, Apr. 2003.

[6] J. Jeong, X. F. Jiang, and D. E. Culler. Design and Analysis of Micro-Solar Power Systems for Wireless Sensor Networks. In *Proc. of the 5th Intl. Conf. on Networked Sensing Systems (INSS '08)*, Kanazawa, Japan, June 2008.

[7] X. Jiang, J. Polastre, and D. Culler. Perpetual Environmentally Powered Sensor Networks. In *Proc. of the 4th Intl. Symposium on Information Processing in Sensor Networks (IPSN '05)*, Los Angeles, CA, USA, Apr. 2005.

[8] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava. Power Management in Energy Harvesting Sensor Networks. *ACM Transactions in Embedded Computing Systems (TECS)*, 6(4):32, 2007.

[9] K. Kant. Toward a Science of Power Management. *Computer*, 42(9):99–101, 2009.

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

[11] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava. Design Considerations for Solar Energy Harvesting Wireless Embedded Systems. In *Proc. of the 4th Intl. Symposium on Information Processing in Sensor Networks (IPSN '05)*, Apr. 2005.

[12] C. Renner, J. Jessen, and V. Turau. Lifetime Prediction for Supercapacitor-powered Wireless Sensor Nodes . In *Proc. of the 8th GI/ITG KuVS Fachgespräch "Drahtlose Sensornetze" (FGSN '09)*, Hamburg, Germany, Aug. 2009.

[13] Samwha Capacitor Group. *Green Caps – Datasheet.*

[14] R. C. Shah and J. M. Rabaey. Energy-Aware Routing for Low Energy Ad Hoc Sensor Networks. In *Proc. of the IEEE Wireless Communications and Networking Conf. (WCNC '02)*, Orlando, FL, USA, Mar. 2002.

[15] Texas Instruments. *TPS 61221 – Datasheet*, Jan. 2009.

[16] T. Voigt, H. Ritter, and J. Schiller. Utilizing Solar Power in Wireless Sensor Networks. In *Proc. of the 28th Annual IEEE Intl. Conf. on Local Computer Networks (LCN '03)*, Bonn/Königswinter, Germany, Oct. 2003.

[17] W. Ye, F. Silva, and J. S. Heidemann. Ultra-low Duty Cycle MAC with Scheduled Channel Polling. In *Proc. of the 4th Intl. Conf. on Embedded Networked Sensor Systems (SenSys '06)*, Boulder, CO, USA, Oct. 2006.