# Energy-aware self-organization algorithms for wireless sensor networks

Rahim Kacimi, Riadh Dhaou and André-Luc Beylot
*University of Toulouse*
*IRIT-ENSEEIHT, 2 rue Charles Camichel, 31071 Toulouse, France.*
{rkacimi, dhaou, beylot}@enseeiht.fr

***Abstract*—Wireless sensor networks (WSN) have received much attention during the last few years especially with regard to energy consumption and scalability. In this paper, we will focus on mechanisms which may be implemented for small WSNs. So, in the present work, we design energy-aware self-organization algorithms for WSNs in such a context. These algorithms can be used to design effective and adaptive protocols. The first protocol concerns the *initialization* or the setting up of the network topology under a chain form. The second one *(steady-state protocol)* implements the communication part or the information exchange between the different nodes of the chain. Our algorithms were dimensioned and validated by an analytical model. We also perform a detailed study of these algorithms by using TOSSIM, a simulation environment for TinyOS, the operating system for the Berkeley sensor nodes. Finally, we achieve an experimental test using Tmote Sky nodes, a popular commercial hardware platform for wireless sensor systems. The results emphasize the interest of the proposed algorithms.**

***Index Terms*—Wireless Sensor Networks, Self-Organization, Energy-Efficiency, Performance Evaluation.**

## I. Introduction

As predicted in Ubiquitous Computing by the late Mark Weiser, computer networking technology is used for creating smart environments with people interacting and controlling the physical world. Indeed, recently, tiny sensor nodes, which consist of sensing, data processing, and wireless communicating components allow people to literally communicate with their physical world. Sensor nodes inject a sense of the real world into user interaction by allowing them to query, sense, and perhaps even manipulate the state of their surroundings [1]. For example, various research groups are exploring the use of network-embedded wireless sensor nodes for monitoring of biological habitats, disaster areas, planets, classrooms, museums, battlefields, and so on. These nodes self-organize into ad hoc networks, divide the task of monitoring among themselves in an energy-efficient manner, adapt their overall sensing quality to available resource, and reorganize upon failure or addition of nodes. Sensors tagged to physical equipment can allow automatic tailoring of environmental settings, and conditionbased maintenance.

Self-organization can be defined as the emergence of a global behavior from local interaction [2]. The fundaments of self-organization are simple algorithms executed by autonomously acting systems. In many wireless network scenarios, nodes do not know anything a priori about the network, so bringing some form of structure is necessary. Solutions may be very different depending on the application and the constraints applied to the network. Moreover, particular mechanisms from different layers might interact to achieve a common goal, e.g. to reduce the necessary amount of energy.

Dressler makes in [3] a classification of self-organization methods and gives some examples of techniques based on these methods. In MAC layer approaches (e.g. S-MAC [4]), self-organization mechanisms help to perform concurrent access, to synchronize nodes, and to maintain duty cycles in a distributed manner without any central management and preconfiguration of nodes nor algorithms. At Network layer, two very different tasks must be solved: routing and data forwarding. We can distinguish between different techniques: proactive and on demand routing. Obviously, both mechanisms finally dictate that state information must be synchronized between many (or even all) nodes in the network. On the other hand, forwarding refers to the delivery of messages to the next hop towards the final destination. Finally, at Application layer, many coordination tasks must be organized such as the identification of master nodes and the allocation of tasks to one of the available nodes. Coordination algorithms such as clustering (e.g. LEACH [5]) and distributed task allocation schemes have been studied in various kinds of networks that are based on self-organization methods.

Certainly all these techniques have already proved their reliability. However, they are often applied for networks having a large number of nodes. Thus, we think that applying them to small networks (with few tens of nodes) will generate additional processing and roles, which are not necessary. This issue has motivated us to design energy-aware solutions for small WSN based on simple algorithms and self-organization paradigms. In particular, the paradigm Do not aim for perfect coordination: Exploit implicit coordination.

The remainder of the paper is organized as follows. In section II we present a sensor network scenario. Section III describes the proposed algorithms. In Section IV we propose an analytical model for our algorithms. Section V present simulations, and experimental results. Finally, section VI concludes the paper.

– This research is supported by the "Capteurs" grant, a National Telecommunication Research Network project which is now in deployment phase.



This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.

## II. Sensor network scenario

To motivate our research, consider a monitoring sensor network that is to be deployed in reduced spaces such as a house, a laboratory or a manufacturing to monitor/control machines; or in transport means (trucks, containers) to supervise goods, etc. In this example and in many other anticipated applications of ad hoc WSNs [6], the deployed systems must be designed to operate under the following conditions and constraints:

- Ad hoc deployment: We cannot expect the sensor field to be deployed in a regular fashion (e.g., a linear array, 2D lattice). Even more, uniform deployment does not correspond to uniform connectivity.
- Energy constraints: The nodes will be untethered for power as well as communications and therefore the system must be designed to expend as few energy as possible in order to maximize the network lifetime.
- Reduced human intervention: we cannot expect that an operator step in instantaneously each anomaly raised by one of the sensors. All the nodes of the network must operate in concert in order to achieve the common goal. Thus, especially in our context, alarms or anomalies are recorded and duplicated on each node until an operator comes to ask any node in order to have a complete knowledge of all what occurred on each one of them.

We enumerate the following assumptions that apply to the remainder of our work: We assume a Carrier Sense Multiple Access (CSMA) MAC protocol with the capacity to work in promiscuous mode. This clearly introduces the possibilities for resource contention when too many neighboring nodes participate in the network. Considering the typical range of the sensors provided by several manufacturers a direct visibility can be considered between all the sensors inside small areas. Consequently, only broadcast communications can be used; The two primary contributions of our design are:

- Proposition of energy-aware and simple self-organization algorithms for small WSN by using an efficient sleep/active scheduling of sensor nodes.
- And to enhance the good results already obtained, we applied the second paradigm in [2] *("Do not aim for perfect coordination: Exploit implicit coordination")* in order to maximize the network lifetime. Indeed, we present one solution in which nodes can take advantage of the overhearing in order to save energy.

## III. Proposed algorithms

The proposed solution is based on two effective protocols described below. The objective is to design a powerful scheduling which organizes the cycle of sleep and activity of each node since the activity duration must be much lower than inactivity. So, we distinguish between two phases (Figure 1): the *Initialization*, where the sensors will start a phase of mutual recognition to set up the chain topology; and *Steady-state* to schedule the regularly waking of nodes to exchange and gather their alarms. In this second phase, we try to schedule sensor communications in order to minimize collisions and retransmissions. Other algorithms were developed to maintain the topology and handle sensor breakdowns (please refer to [7] for more details). These choices have primarily an

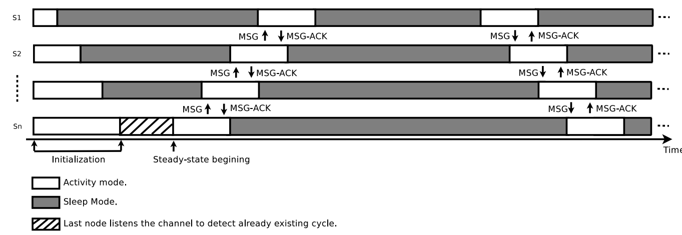


Fig. 1. General principle of the solution.

impact on the application layer protocols implemented and possibly on the MAC layer parameter setting. Here each sensor chooses its activity periods according to its predecessor in the chain. Each sensor thus calculates its waking date in order to disseminate its information during the following cycle, towards its predecessor in the current cycle.

### A. Initializtion protocol

After their activation, each node picks a random duration ($T_r$) during which it senses the carrier. At the application layer, it corresponds to an idle period. The sensor which chooses the smallest $T_r$ sends a SYNC message by providing its MAC address, it allots the number 1 in the cycle and indicates the date of its next waking. In the best case, no other node sends

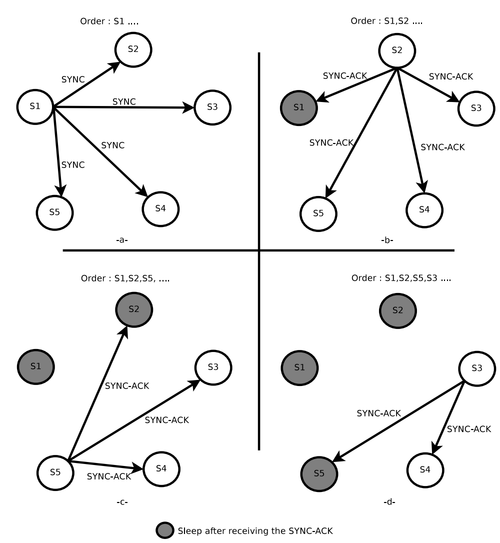


Fig. 2. Initialization algorithm.

a *SYNC* at the same time and there is no error of transmission. So, all the active nodes receive this message and pick a new random duration. At the end of this duration, one of them will send an acknowledgment *SYNC-ACK* by allotting number 2, providing its MAC address and calculating its date of waking. The first node of the cycle may then sleep again and the construction of the cycle can continue. Once the last node transmits its message, none answers. It can fall asleep again.

### B. Steady-state protocol

Two alternatives for this protocol are considered, based either on explicit or implicit acknowledgments. After the initialization phase, the chain is constructed and the last sensor added to the cycle builds a first message MSG with its date of waking and sends it to the following sensor. The first cycle starts in the opposite direction of its formation. The last sensor is the only one to have all the information, particularly the number of sensors in the network.

Let us first consider the use of explicit acknowledgments. In the best case, sensor $S_i$ being located in the middle of the chain, receives a *MSG* from its predecessor, answers by a *MSG-ACK*, sends the *MSG* to its successor and returns to sleep mode after receiving a *MSG-ACK* from its successor. Otherwise, $S_i$ can handle two cases of losses: loss of $S_{i-1}$ when no message MSG is received; loss of $S_{i+1}$ in the absence of acknowledgment *MSG-ACK* from its successor. The cases of losses, collisions and errors are detailed in [7].

In our protocols, the use of implicit coordination can be exploited. Implicit coordination means that coordination information is not communicated explicitly by signaling messages, but is inferred from the local environment. A node observes other nodes in its neighborhood; based on these observations, it draws conclusions about the status of the network and reacts accordingly. Suppose node $S_{i-1}$ sends a message to node $S_i$, which in turn forwards the message to node $S_{i-1}$. If $S_{i-1}$ overhears the message of $S_i$ to $S_{i+1}$, it knows that $S_i$ has received it from $S_{i-1}$. In other words, the overhearing of a message can serve as an implicit acknowledgment.

## IV. Performance evaluation

In this section, we present an analytical model in order to analyze both the initialization and the steady-state phase. After that, we will solve this analytical model and compare it to the simulation results.

### A. Initialization phase

To avoid collisions, after their activation, the nodes initially sense the channel during a random duration $T_r$ uniformly distributed in the interval $[0, Tmax]$. Let $T_1$ be the mean initialization time of the first node. It can be easily shown that:

$$T_1 = \frac{T_{max}}{N+1} + T_{SYNC} + \frac{T_{max}}{N} + T_{SYNC-ACK} \quad (1)$$

Where $T_{max}$ is the maximum duration of the timer, $N$ is the number of nodes. It is also possible to deduce the mean initialization time of node $i$ in the chain:

$$T_i = T_{i-1} + \frac{T_{max}}{N-i+1} + T_{SYNC-ACK} \quad (2)$$

However, there is a difference for the last node of the chain because it will experiment $\theta$ retransmission attempts because no node can answer it and then waits $T_W$ in order to detect already existing cycle:

$$T_N = T_{N-1} + \theta T_{RET} + (\theta-1)T_{SYNC-ACK} + T_W \quad (3)$$

#### Energy consumption

In the above formulas, we point out that a node works in several modes: listening, reception and sending. To estimate the mean energy consumption we need to add the various consumptions corresponding to each mode, we use the *Tmote Sky* sensor parameters [8] listed in Table I. We get:

$$\begin{aligned} E_1 &= I_{TX}T_{SYNC} + I_{RX}T_{SYNC-ACK} \\ &+ I_L T_{max}\left(\frac{1}{N+1} + \frac{1}{N}\right) \end{aligned} \quad (4)$$

Where $E_1$ is the mean power consumption of node 1; $I_L$, $I_{RX}$ and $I_{TX}$ are the current intensities corresponding respectively to the listening, reception and sending mode. So, the mean energy consumption of node $i$ is equal to :

$$\begin{aligned} E_i &= I_{RX}T_{SYNC} + I_{TX}T_{SYNC-ACK} \\ &+ (i-1)I_{RX}T_{SYNC-ACK} \\ &+ I_L T_{max}\left(\frac{1}{N+1} + \frac{1}{N} + \ldots + \frac{1}{N-i+1}\right) \end{aligned} \quad (5)$$

As for initialization times, the consumption is an increasing function of the rank of the sensor in the chain and of the number of sensors:

$$\begin{aligned} E_N &= I_{RX}T_{SYNC} + I_{TX}T_{SYNC-ACK} \\ &+ (N-2)I_{RX}T_{SYNC-ACK} \\ &+ I_L T_{max}\left(\frac{1}{N+1} + \frac{1}{N} + \ldots + \frac{1}{2}\right) \\ &+ I_L(\theta T_{RET} + (\theta-1)T_{SYNC-ACK} + T_W) \end{aligned} \quad (6)$$

### B. Steady-state Analysis

For the steady-state phase analysis, we were interested in the average time during which two adjacent nodes exchange their information (Figure 3). In this part, the error rate is considered while collisions are rare events since the protocol has been designed in order to avoid collisions. This time $\overline{T}$ is expressed by the following formula:

$$\overline{T} = \overline{N}(T_{msg} + T_{ack} + 2T_p + 2T_{mcu}) \quad (7)$$

where $\overline{N}$ is the average number of retransmissions until the communication between two adjacent nodes succeeds, $T_{msg}$ is the *MSG* emission time, $T_{ack}$ is the *MSG-ACK* duration, $T_p$ is the propagation time and $T_{mcu}$ is the necessary time for the message treatment by the *MCU*. $N$ is calculated by the following formula:

$$\overline{N} = \frac{1-\overline{\eta}^{\alpha}}{\eta} \quad (8)$$

$\eta$ is the probability that there is no transmission error neither in the message nor in the acknowledgment and $\alpha$ is the maximum number of retransmissions. $\eta = 1 - \overline{\eta} = \overline{\tau}_{msg}\overline{\tau}_{ack}$. $\tau_{msg}$ and $\tau_{ack}$ are the error rates in *MSG* and *MSG-ACK* packets. The average energy consumption in the steady-state phase is represented by the following formula:

$$\begin{aligned} \overline{E} &= \overline{N}(E_{Rx,msg} + E_{Tx,ack} + 2(I_L T_p + I_{mcu}T_{mcu})) \\ &+ I_L(D - \overline{T}) \\ &+ \overline{N}(E_{Tx,msg} + E_{Rx,ack} + 2(I_L T_p + I_{mcu}T_{mcu})) \end{aligned} \quad (9)$$

 

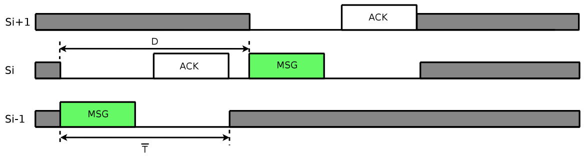


Fig. 3. Steady-state with explicit acknowledgments.

where $E_{Tx,msg}$ and $E_{Rx,msg}$ are respectively the energy consumed to send and to receive a message *(MSG)*, $E_{Tx,ack}$ and $E_{Rx,ack}$ are respectively the energy consumed to send and to receive an acknowledgment *(MSG-ACK)*. $I_L$ and $I_{mcu}$ are the current intensities corresponding respectively to the channel listening and *MCU* processing mode. As shown in Figure 3, $D$ is the difference between the wake-up time of two adjacent nodes. The results do not take into account the *MCU* power consumption. Indeed, in the following results, we only evaluated the energy consumption of the radio.

### C. Steady-state with implicit acknowledgments

In the case when implicit acknowledgments are used, the average time $\overline{T}$ during which two adjacent nodes exchange their information (Figure 4) is expressed by the following formula:

$$\overline{T} = \overline{N}(T_{msg} + T_{msg^+} + 2T_p + 2T_{mcu}) \quad (10)$$

$\overline{N}$ is calculated using (8), where $\eta = 1 - \overline{\eta} = \overline{\tau}_{msg}\overline{\tau}_{msg^+} \simeq \overline{\tau}^2_{msg}$. $\tau_{msg}$ and $\tau_{msg^+}$ are the error rates of $S_i$ and $S_{i+1}$'s messages (*MSG* and *MSG*+). The average energy consumption

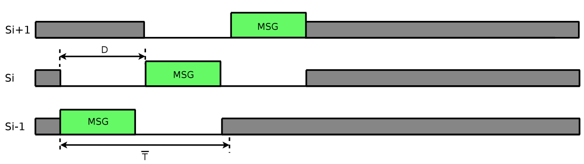


Fig. 4. Steady-state with implicit acknowledgments.

in the steady-state phase is represented by the following formula :

$$\begin{aligned}\overline{E} &= \overline{N}(E_{Rx,msg^-} + E_{Tx,msg} + E_{Rx,msg^+} \\ &+ 3I_LT_p + 2I_{mcu}T_{mcu})\end{aligned} \quad (11)$$

where $msg^-$ and $msg^+$ refer respectively to the message received from node $S_{i-1}$ and the one sent by $S_{i+1}$ to $S_{i+2}$.

## V. Simulation and Experimental Methodology

The purpose of our simulations is to evaluate the effectiveness of our protocol and their energy consumption. We used the TOSSIM discrete-event simulator, which is included with the TinyOS release, to evaluate our protocols. TOSSIM has several advantages: it runs actual TinyOS implementations, it allows experimentation with a large number of nodes, it accurately captures the TinyOS behavior at a low level (e.g., timer interrupts), and it models the CSMA/CA MAC layer of the node. Therefore, imperfections, such as interference and packet collisions, are accounted for. This evaluation provides a pretty realistic results of the proposed algorithms. The power consumption of the algorithms is calculated using PowerTOSSIM [9]. The simulation parameters of the messages duration and the intensities of the electrical current are listed in Table I. Figure 5 shows the mean energy consumption of

TABLE I
Simulation Parameters

| Message duration (ms) | | Current (mA) | |
|---|---|---|---|
| $T_{SYNC}$ | 1.44 | $I_{TX}$ | 17.9 |
| $T_{SYNC-ACK}$ | 1.69 | $I_{RX}$ | 19.7 |
| $T_{MSG}$ | 2.58 | $I_L$ | 19.7 |
| $T_{MSG-ACK}$ | 1.6 | | |
| $N$ : [10..100], $\theta = 3$, $T_A = 0ms$, $T_{max}$ : [15..120]$ms$ | | | |

the last node which join the network while varying the number of nodes $N$ : $[10-100]$ and $T_{max}$ value: $[15-120]ms$. In this figure, we point out that the energy consumption expressed in $mAh$ increases linearly with network size ($N$). Moreover, for a given network size, the energy consumption increases also linearly when $T_{max}$ value grows.

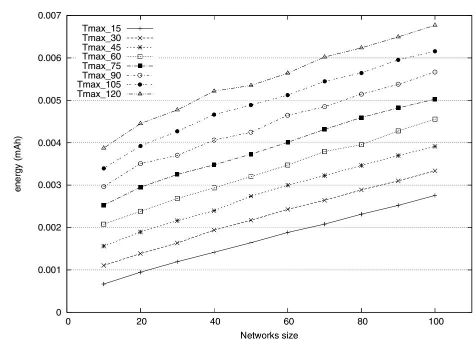


Fig. 5. Energy consumption

Let us consider a network of $N = 30$ nodes. Therefore, for this network size, we estimate the mean energy consumption during this phase. These results are expressed in $mAh$ according to the node position in the cycle. That is illustrated in Figure 6. It is shown that initialization times and energy

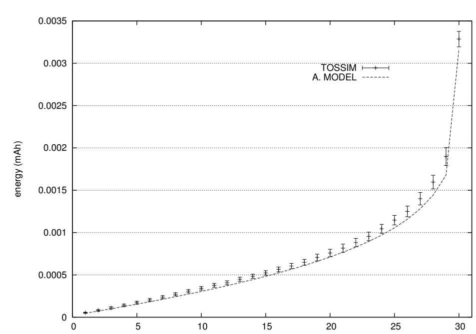


Fig. 6. Energy consumption with $N = 30nodes, T_{max} = 75ms$

consumption linearly grow according to the node position, then in an exponential way for the last nodes. The reason


This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.

is the growth of the timer $T$ caused by the reduction in the remaining number of nodes (see equation 1). It can be seen also, that simulation results denoted by errobars, fits perfectly the analytical results represented by solid lines.

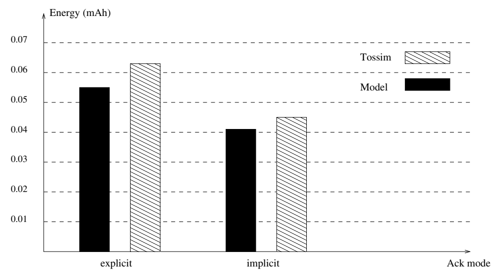


Fig. 7. Energy consumption of a node $S_i$ in both steady-states whith implicit and explicit acknowledgments.

Figure 7 shows the energy consumption of a node $S_i$ according to the two proposed steady-state algorithms (with explicit and implicit acknowledgments). First, we can say that the slight difference between the results of the model and those of the simulator comes from the power consumed by the Radio when it changes state (e.g. from $R_x$ state to $T_x$ state, $R_x$ to save, $T_x$ to $R_x$, ...) which are not taken into account in the analytical model. The interest of implicit acknowledgments is emphasized by energy saving of approximately 27%.

From all these results, we find that our protocols provide a significant amount of energy saving. Indeed, an initial version of this solution was mapped for an application of cold chain monitoring [7] and the initialization and steady-state protocols represent only a *third* of the daily energy consumption.

*Experiments*

Experiments were performed using Tmote Sky [8] devices. The platform is smaller than a business card, and includes a microcontroller operating at 8MHz, 48K of ROM, 10K of RAM, a 2.4GHz ZigBee wireless transceiver, and a USB interface for device programming and logging. Each device operates on 2 AA batteries. Our experiments were performed on the first floor of IRIT-ENSEEIHT laboratory. A photo of the site is shown in Figure 8. We used 4 nodes in order to test particularly both steady-states altenatives. The distance between the sensors is approximately $12m$. The experiments show especially the good connetivity between nodes and the absence of packets losses what endorse the second alternative of steady operation with implicit acknowledgments. The results consolidate also the analytical model. Indeed, the initialization times of each sensor are very close to the model estimations. Indeed, the average deviation between the model's initialization times and those of the experiments is $14.18\%$. For instance, the initialization time of the last node of the chain (sensor 4) is $22.03ms$ against $19.98ms$ in the analytical model. We think that this slight difference comes from the processing time of messages which is not taken into account in the model for the initialization phase and also the difference between the propagation models.

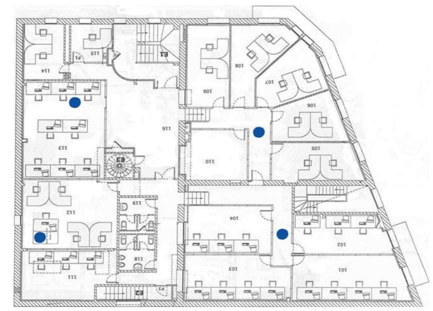

Fig. 8. Experimental tests.

## VI. Conclusion

In this paper, we described the design, analysis, simulation and experimental evaluation of energy-aware self-organization algorithms for small WSNs. These algorithms allow to deploy a WSN solution in monitoring contexts without a base station nor central nodes. Sensors are self-organized in a chain and alternate between sleep and active mode where the sleep periods are longer than the activity periods. The use of implicit coordination is exploited and shows a significant potential for reduction of power consumption. Furthermore, an analytical model was developed to validate our protocols. Simulation results show the accuracy of the proposed analytical model. Finally, we emphasize that this solution is easily useable and increases considerably the WSN lifespan. Moreover, these protocols may be mapped for different applications and experimented in various environments.

## References

[1] B. Badrinath and M. Srivastava, "Smart spaces and environments," *Personal Communications*, vol. 7, no. 5, pp. 3–3, Oct 2000.

[2] C. Prehofer and C. Bettstetter, "Self-organization in communication networks: principles and design paradigms," *Communications Magazine, IEEE*, vol. 43, no. 7, pp. 78–85, July 2005.

[3] F. Dressler, "A study of self-organization mechanisms in ad hoc and sensor networks," *Computer Communications*, vol. 31, pp. 3018–3029, August 2008.

[4] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pp. 1567–1576 vol.3, 2002.

[5] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, vol. 2, 4-7 Jan. 2000.

[6] J. Gehrke and L. Liu, "Sensor-network applications," *IEEE Internet Computing*, vol. 10, no. 2, pp. 16–17, 2006.

[7] R. Kacimi, R. Dhaou, and A.-L. Beylot, "Placide: An Ad Hoc Wireless Sensor Network for Cold Chain Monitoring," in *International Conference on Performance Modelling and Evaluation of Heterogeneous Networks, Karlskrona, Sweden.*, Fabruary 2008.

[8] "Tmote sky datasheet," Moteiv Corporation, 2006. [Online]. Available: http://www.moteiv.com/products/docs/tmote-sky-datasheet.pdf

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