Experimental Lifetime Evaluation for MAC Protocols on Real Sensor Hardware*

Thomas Staub, Thomas Bernoulli, Markus Anwander , Markus Waelchli and Torsten Braun
Institute of Computer Science and Applied Mathematics
University of Bern
{staub, bernoull, anwander, waelchli, braun}@iam.unibe.ch

ABSTRACT

In this paper we present our experimental lifetime measurements for two implementations of MAC protocols designed for wireless sensor networks. We have implemented the TDMA based LMAC and the contention-based TEEM on the ESB nodes developed at FU Berlin. We show the energy requirements of these protocols on the respective hardware and also present figures for the trade off between robustness and energy consumption of these protocols. The main contribution of this paper consists in real world measurements giving an idea of energy conserving MAC protocols' potential to extend node lifetime.

1. INTRODUCTION

The research community paid a lot of attention to the technology of wireless sensor networks, mainly on theoretical issues. It was experienced that deployment of sensor networks is very challenging. Besides environmental factors one important aspect for the deployment of unattended wireless sensor networks is that the nodes have strong power restrictions. The sensor nodes are usually battery-driven and become useless if the battery is depleted. Therefore, research focuses on energy efficient operations of sensor networks. Because communication is one of most energy consuming tasks energy efficient communication schemes including media access (MAC) protocols are of high importance. In order to obtain valuable feedback for deployment of sensor networks, real world experiments on real sensor hardware comparing different MAC protocols are needed.

The main contributions of this paper are the implementation and lifetime comparison of two existing wireless sensor MAC protocols on real sensor hardware. We implemented one contention and one TDMA based MAC protocol on ESB sensor nodes from ScatterWeb [4]. The experimental evaluation of the lifetime of sensor nodes is done by using an approach described in [3]. The sensor node is powered by a GoldCap capacitor in order to enable lifetime evaluations within a reasonable amount of time and to eliminate the battery relaxing effect as well as the heavily varying charge of batteries.

The paper is outlined as follows: First, we present related work in the area of sensor MAC protocols including the two protocols that we have implemented. We describe in Section 3 on the implementation of the protocols on the specific hardware of a ScatterWeb node. The setup of the experiments and the results are presented and discussed in Section 4. We conclude with some remarks and future work.

*The work presented in this paper was partly supported by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

2. RELATED WORK

In this section, we present related work in the area of media access control (MAC) protocols for wireless sensor networks. Energy efficient MAC protocols for wireless sensor networks can be mainly categorized as either "time division multiple access" (TDMA) based [2, 7] or contention based [8, 6, 5, 1] . Contention based protocols may waste energy if collisions appear, which lead to packet retransmissions. TDMA-based protocols schedule the media access for the individual nodes with time-slots. Each node gets a time-slot assigned and receives exclusive access rights on it. Because of this collision-free and non-concurring nature in the individual time-slots TDMA based protocols are generally more energy conserving than contention based approaches. On the other hand they suffer from high design complexity to solve non-trivial synchronization problems, are less flexible, and sometimes do not utilize the network resources efficiently.

The authors of [8] propose a new protocol called Sensor-MAC (S-MAC), which uses a listen/sleep cycle. During the listening period nodes can communicate with each other and send control packages (SYNC, RTS, CTS). Each node has its own listen/sleep schedule. By periodically exchanging SYNC messages, neighboring nodes synchronize their schedules to reduce the control overhead. Coordinated sleeping is achieved by adapting the schedules of all known neighbors (virtual clustering). Furthermore, the SYNC messages are used to avoid long-term clock drifts. Through a successful RTS/CTS exchange in the listen period, the peering nodes are kept awake during their sleeping period and can exchange data. Using this scheme, nodes form virtual clusters on common schedules, and communicate directly with peers.

The authors of [6] propose a contention-based MAC protocol called T-MAC. T-MAC avoids the overhead of fixed duty-cycles as with S-MAC. It makes the duty-cycles adaptive by dynamically terminating the listening period. When nothing has been sensed in a given interval a timeout occurs and the nodes go to sleep. T-MAC uses RTS/CTS/DATA/ACK schemes to avoid collision and to ensure reliability. It uses the same virtual clustering algorithm as S-MAC. In T-MAC all buffered data is transmitted in a burst at the start of a frame. The sender retransmits a RTS if it receives no CTS. If the receiver still does not answer, the sender goes to sleep. The basic T-MAC functionality inherently has throughput limitations when the traffic is unidirectional. In detail, a down-stream neighbor of a node may go to sleep too early if the node looses contention due to RTS/CTS transmissions of its own up-stream neighbors. Thus, the forwarding path is broken and the data forwarding has to wait for the next active time period. T-MAC therefore introduces two possible solutions: the future-request-to-send (FRTS) or the full-buffer-priority.

TEEM [5] is an enhancement of the S-MAC protocol with two improvements. First, nodes turn off their radio much earlier when no data traffic is expected and second the transmission of separate RTS control packets is avoided. In TEEM the listen period is divided into a $\mathrm{SYNC}_{\mathrm{DATA}}$ and a $\mathrm{SYNC}_{\mathrm{NODATA}}$ part. If no node has any data to send in the $\mathrm{SYNC}_{\mathrm{DATA}}$ period, the contention winner sends a small $\mathrm{SYNC}_{\mathrm{NODATA}}$ packet for synchronization and all nodes turn immediately their radio off after this reception. If a node has data traffic to send it sends a $\mathrm{SYNC}_{\mathrm{DATA}}$ packet. It is obvious that this node is also assumed to win the RTS contention. Consequently, both messages can be combined. The $\mathrm{SYNC}_{\mathrm{DATA}}$ packet is transmitted as $\mathrm{SYNC}_{\mathrm{RTS}}$. The main drawback of TEEM is its allowance of only one-hop forwarding per transmission slot, as no operations like FRTS (T-MAC) are possible.

In [1] another contention-based MAC scheme based on a preamble sampling technique is proposed. The main idea of the protocol called WiseMAC is the learning of the neighbor's sampling schedules. It enables the sender to predict its neighbor's wake up time and to start the preamble transmission when the neighbor is awake. Further, all nodes wake up periodically to sense the medium. If the medium is busy they stay awake until they know whether the data destined for them and go to sleep again otherwise. WiseMAC is more efficient than S-MAC for unicast traffic, but has problems with broadcast traffic because the sampling schedules are learned per neighbor.

In [2] the TDMA based MAC protocol DE-MAC is described. It uses a local election procedure to enforce that the low power election winners sleep more. The leader-election is thereby integrated into the regular TDMA communication. A sensor node switches off its radio and goes into a sleep mode only when it is in its own time slot and does not have anything to transmit.

The LMAC [7] protocol is a TDMA based protocol. Each node possesses exclusively one slot in the frame. Within its slot it can communicate collision-free. All sensors awake at each slot to overhear the control message of the slot owner. The nodes synchronize according to that message. Furthermore, they go to sleep if no data has been advertised for them by the slot owner. Initially, the gateway takes a slot and announces it. All one hop neighbors overhear that message and synchronize with the gateway. Additionally, they take an own time slot randomly from the available time slots. They announce their time slot similarly to the gateway. If two sensors take the same time slot a collision of the control messages occurs. In this case the sensors are informed by their neighbors that a collision occurred and they take another time slot.

There exist implementations of S-MAC and TEEM on Berkeley Mica Mote based on TinyOS. But to our best knowledge, there are no implementations of TEEM and LMAC made on the ScatterWeb hardware platform. We have ported the two MAC protocols to the ESB nodes of ScatterWeb which is described in the following section.

3. IMPLEMENTATION

We have implemented the protocols LMAC [7] and TEEM [5] on the Embedded Sensor Boards of ScatterWeb [4]. We use the source code of ScatterWeb as a basis of our implementations of the chosen MAC protocols. First, we present the important issues of the used hardware platform, then the implementations of LMAC and TEEM are described beginning with LMAC.

The ESB is equipped with a micro controller MSP430F149 from Texas Instruments, various sensors, and a transceiver module TR1001 from RFM for communication. The TR1001 operates in the license-free 868MHz band at a transmission rate of 19.2kb/s. The interface of the TR1001 is a simple serial bit stream interface for sending and receiving bytes. Its action is completely transparent. It takes the bytes from the input and sends them at 868 MHz, using On-Off-Keying. In reception mode, it reacts on signal surpasses a certain threshold by demodulating it. The received bytes are sent to the microcontroller. The TR1001either transmits or receives at any given moment, i.e. it supports only half-duplex. If it is powered, it is set to receiving by default. The ScatterWeb software for the ESB needs at least 4 transmission interrupts (about 1.7 ms) to turn from receive state to transmit state and send a start byte, which an other node can detect.

We have implemented LMAC as described in [7]. Figure 1 provides an overview of the process flow in our implementation. LMAC divides one time-frame of 5.12s length in 32 slots of 160ms. Each slot is assigned to one node. Therefore, each node can send only every 5.12s. Every 160ms the node is waken up by the wake-up function. The node listens to the medium. If the node owns the current slot, it can send data. If the node has data pending in its send buffer, it transmits $\mathrm{SYNC}_{\mathrm{DATA}}$ and the data as one packet. All data in the queue for the target node is also transmitted up to 256 bytes (slot limit). After sending the data the node shuts down the radio. If there is no data to transmit in the slot, the node sends a $\mathrm{SYNC}_{\mathrm{NO\text{-}DATA}}$ and turns off the radio interface. If the node is a receiver, it checks whether there is a transmission. If not, the radio is shut down. Otherwise, the timers are updated according to the information of the SYNC messages. The node checks whether it is the destination of the following transmission. The radio is shut down directly if the node is not addressed. Otherwise, it stays awake and receives the data.

TEEM is implemented according to [5] and for specific values according to S-MAC [8]. We use a slot time of 600ms. The slot contains a listen period and a sleep period. The listen period is about 83ms and divided into a $\mathrm{SYNC}_{\mathrm{DATA}}$ and a $\mathrm{SYNC}_{\mathrm{NODATA}}$ contention frame.

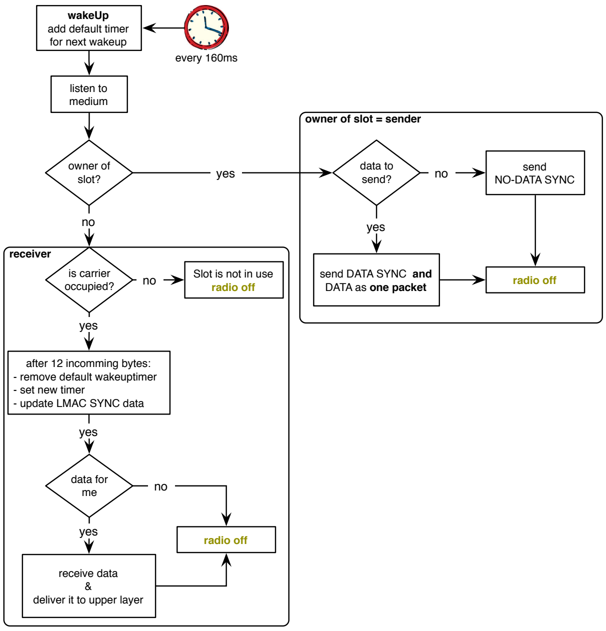

Figure 1: Flow diagramm of LMAC Implementation

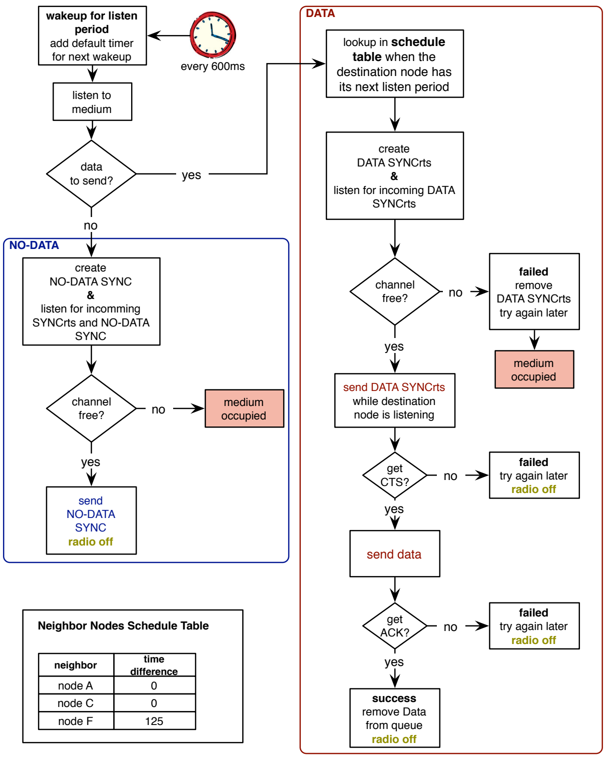

Figure 2: Process flow of TEEM

Figure 2 shows the process flow. The node wakes up every 600ms. If it has pending data in its buffer, it tries to send a SYNC_{rts} during the $\text{SYNC}_{\text{DATA}}$ period. If the medium is occupied, the received data is processed (see Figure 3). If the medium is free, the node sends a $\text{SYNC}_{\text{DATA}}$ message while the destination node is listening. The listening time of the destination node is received from the neighbor nodes' schedule table. If the node gets no CTS or the sent data is not acknowledged, the node postpones the transmission to the next slot and shutdowns the radio. Immediately after successful data transmission, the node also switches the radio off. If the node has no pending data, it will listen to the medium during the $\text{SYNC}_{\text{DATA}}$ period for receiving any SYNC_{rts}. If there has been no $\text{SYNC}_{\text{DATA}}$, it tries to send a $\text{SYNC}_{\text{NO-DATA}}$. After successful transmission or if another node has transmitted a SYNC, the node turns the radio off. Figure 3 shows the processing of an incoming transmission. For the node the medium seems occupied, it can send neither a SYNC_{rts} nor a $\text{SYNC}_{\text{NO-DATA}}$. If the received data is neither a SYNC_{rts} nor a $\text{SYNC}_{\text{NO-DATA}}$, it turns the radio off. Otherwise the SYNC is evaluated, the neighbor schedule table is updated with the sleep time of the sender, and the wake-up time of the node is set. If it is a SYNC_{rts} and indicates a data transmission for the node, the node sends a CTS, receives the data, and after acknowledging them, goes to sleep. Otherwise, it turns off the radio immediately.

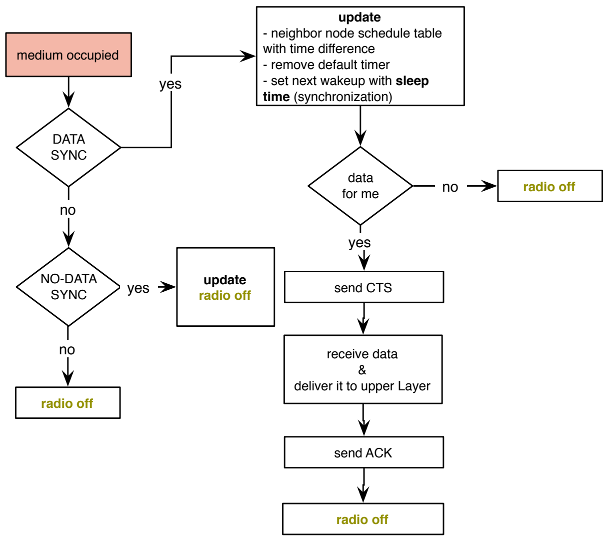

Figure 3: Process flow II of TEEM (transmission on medium)

4. EVALUATION

4.1 Measurement Methodology

When evaluating energy consumption experimentally, a power source with repeatable characteristics is needed. We used so-called 1F GoldCaps capacitors with a very high capacity of 1 Farad and 5V. Because a ESB node fails as soon as the voltage of the power source falls below a certain threshold, the sensor board delivers a *Battery Warning* message if the voltage falls below a certain limit. Evaluating this *Battery Warning* is one possibility to determine the lifetime of a sensor board. An other possibility is to log the shutdown of the RS-232 interface, the moment where the node is not able to transmit data over the RS-232 interface anymore.

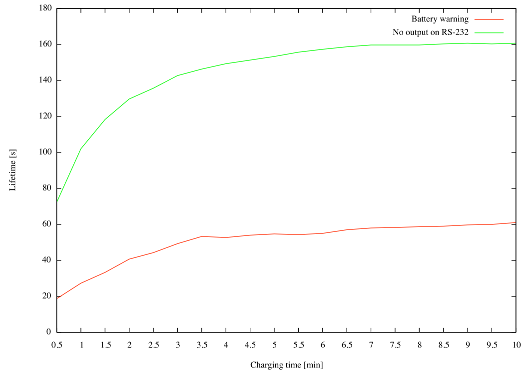

Figure 4: Correlation of GoldCap charging time and ESB lifetime using *Battery Warning* messages and RS-232 shutdown as indicators.

Figure 4 shows a series of measurements analyzing the correlation of GoldCap charging time and the node lifetime using shutdown of RS-232 interface and *Battery Warning* message. As the sensor board persists quite a long time in operational mode after the *Battery Warning* message we have decided to use the death of the RS-232 interface as the breakpoint for our lifetime measurements. Further, we fixed the charging time for the capacitor to 5.5 min. For the measurements depicted in Figure 4 we have not applied any power saving mechanism which results in shorter lifetimes compared to the ones reported in [3] which have been performed with a very similar methodology.

To assure that there exist no significant differences between different ESB nodes we measured their power consumption under four different operating modes: We independently turned the network and the RS-232 interface on and off. Figure 5 illustrates the resulting power consumption of 8 individual ESB nodes in these operating modes. An important observation is the fact that the RS-232 interface accounts for up to 55% (network off, RS-232 on) of the energy used. This has to be kept in mind when interpreting the lifetime results.

Three MAC protocols have been part of our experimental lifetime evaluation: LMAC, TEEM and the CSMA mechanism provided by ScatterWeb. The ScatterWeb CSMA is a robust one as it tries to retransmit the message up to 15 times which results in nearly 100% delivery ratio. To analyze the energy consumption characteristics we used two scenarios. In a first one all nodes lie within short distances (on the same table) of each other and build up a full mesh. This scenario is called *short range* scenario. For the second scenario the nodes have been placed throughout a building to assure that the built network follows the topology depicted in Figure 6 and to analyze the lifetime in a real world setup. As the nodes have been placed with quite some distance between them we call this scenario *long range* scenario. The route of the messages from the nodes to the gateway has been hardcoded as indicated by the arrows in Figure 6. In the *short range* scenario the packets are forwarded along a similar route.

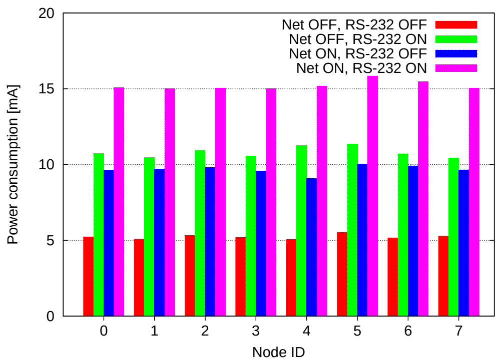

Figure 5: ESB power consumption: 8 ESB nodes and their power consumption in mA. The power consumption has been measured turning the network and the RS-232 interface ON and OFF.

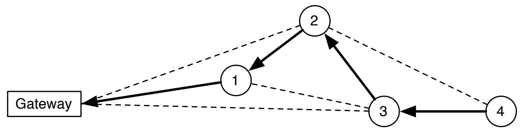

Figure 6: The second scenario: The nodes cannot overhear each other. The links have very different properties. Links along the route have low bit error rates. Other links experience higher bit error rates.

We used two traffic patterns to mimic periodic sampling and transmission of sensed values from every node to the gateway. Each node generates a 12 byte packet every 10 seconds in the first and every 20 seconds in the second traffic pattern. Due to the route of the packets to the gateway this results in a load of 4 and 2 packets per 10s to forward for node 1 respectively. Combining the two traffic patterns and the two node placement results in the four scenarios listed in table 1.

Scenario	Distance	Message generation interval
1	short range	10s
2	wide range	10s
3	short range	20s
4	wide range	20s

Table 1: The four scenarios with different message generation intervals at the nodes and different distances between them.

Lifetime measurements have been performed on node 1, the most loaded one. LMAC and TEEM are used as they have been implemented without any tuning of parameters. The ScatterWeb CSMA is the MAC layer provided by ScatterWeb [4]. It represents a MAC without any energy conservation mechanism.

4.2 Comparison Results of MAC schemes

Figure 7 illustrates the varying node lifetimes in different scenarios using different MAC schemes. Even a glimpse at it clearly shows the benefit of energy conserving MAC protocols. It also reveals that in our setup robustness in the real world scenarios (2nd and 4th) has to be paid with energy.

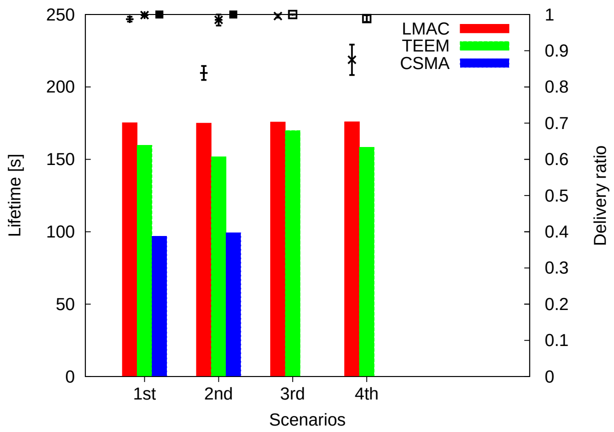

Figure 7: Node lifetime and packet delivery ratio. Node lifetime 95% confidence intervals are too small to be printed (below 0.85%), packet delivery ratios are printed with their 95% confidence intervals.

While TEEM and CSMA show significant differences in node lifetime between the two scenarios LMAC does not. Because LMAC has no acknowledgement and retransmission mechanism it does not consume more energy in situations with higher link bit error rates but just loses delivery performance. TEEM in contrary cannot profit enough from this effect to compensate the more often *sync messages* transmitting which is also caused by the sparser network.

With the 3rd and 4th scenario we wanted to analyze the performance of LMAC and TEEM in less loaded networks. Therefore we set the message generation interval at each node from 10s to 20s. LMAC shows the same performance as in the higher loaded scenario. Because it does not suffer from collisions in scenario 1 and 2 it cannot profit from a lower collision/backing of probability. The fact of having fewer data transmissions does not really show an effect because the main network load consists of sync messages. TEEM in contrary profits a lot from the lower network load: The probability of backing off and retransmission are much lower - especially in the *short range scenario*. Again it is obvious that the much better delivery ratio compared to LMAC has its cost in node lifetime.

As showed in section 4.1 the RS-232 consumes quite a lot of energy. If we have not needed this interface to correctly log the node lifetime the nodes would have been running significantly longer. Although this applies to all measured values it is important to recognize that this additional source of energy consumption tampers the relative performance comparison of the protocols. A short example: According to Figure 5 the RS-232 accounts for 30% to 55% of the total energy consumption. Assuming that in scenario 2 it accounts for 50% if LMAC is running, the same amount of energy accounts only for 30% if TEEM is running. In the relative comparison of LMAC and TEEM this translates in 31% longer lifetime of a LMAC node with RS-232 shut down instead of only 15% longer lifetime compared to TEEM with RS-232 active. This is an increase of relative performance of slightly more than 100%. Unfortunately it is not possible to exactly measure the RS-232 energy consumption influence in the different experiments. Therefore this short example gives a hint that keeping in mind this blemish.

When comparing the lifetimes we measured to the ones reported in [3] they are extremely short. The nodes in our measurements have been running MAC protocols that would allow an arbitrary network protocol that is not aware of sleep modes. In contrary the nodes in [3] have been put to sleep states between the 10 second transmission interval. In case topology control algorithms and protocols are used not all nodes have to be run in an *always on* MAC protocol such as LMAC or TEEM without additional sleep cycles. This would lead to an average lifetime between the two extremes.

5. CONCLUSIONS AND FUTURE WORK

This paper shows the lifetime performance of the two energy conserving MAC protocols LMAC and TEEM on real hardware under real world conditions: LMAC provides longer lifetime at the cost of lower packet delivery ratios. Further we show how these protocols can be implemented on ESB nodes and that measuring under real world conditions is a difficult task and additional efforts are needed in the future trying to avoid additional sources of energy consumption just for measuring. Future work will also analyze the potential of topology control mentioned in section 4.2 to further extend node and the resulting network lifetime.

6. REFERENCES

[1] A. El-Hoiydi and J.-D. Decotignie. Wisemac: An ultra low power mac protocol for the downlink of infrastructure wireless sensor networks. In *ISCC2004, the 9th IEEE International Symposium on Computers and Communications,*, Alexandria, Egypt, June 2004.

[2] R. Kalidindi, R. Kannan, S. S. Iyengar, and L. Ray. Distributed energy aware mac layer protocol for wireless sensor networks. In *International Conference on Wireless Networks*, pages 282–286, 2003.

[3] H. Ritter, J. Schiller, T. Voigt, A. Dunkels, and J. Alonso. Experimental evaluation of lifetime bounds for wireless sensor networks. In *European Workshop on Wireless Sensor Networks ESWN 2005*, Istanbul, Turkey, January 2005.

[4] Scatterweb. Platform for self-configuring wireless sensor networks. http://www.scatterweb.net.

[5] C. Suh and Y.-B. Ko. A traffic aware, energy efficient mac protocol for wireless sensor networks. In *IEEE 2005 International Symposium on Circuits and Systems (ISCAS'05)*, May 2005.

[6] T. van Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 171–180, Los Angeles, California, USA, 2003.

[7] L. van Hoesel and P. Havinga. A lightweight medium access protocol (lmac) for wireless sensor networks: Reducing preamble transmissions and transceiver state switches. In *INSS*, Japan, 2004.

[8] W. Ye, J. Heidemann, and D. Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 12(3):493–506, June 2004.