Mobile Networks and Applications 10, 783–791, 2005

Comparing Energy-Saving MAC Protocols for Wireless Sensor Networks

G.P. HALKES, T. VAN DAM and K.G. LANGENDOEN*
Faculty of Electrical Engineering, Mathematics, and Computer Science, Delft University of Technology, Mekelweg 4, P.O. Box 5031, 2600 GA Delft, The Netherlands

Abstract. Applications for wireless sensor networks have notably different characteristics and requirements from standard WLAN applications. Low energy consumption is the most important consideration. The low message rate that is typical for sensor network applications and the relaxed latency requirements allow for significant reductions in energy consumption of the radio. In this article we study the energy saved by two MAC protocols optimized for wireless sensor networks, S-MAC and T-MAC, in comparison to standard CSMA/CA, We also report on the effects of low-power listening, a physical layer optimization, in combination with these MAC protocols. The comparison is based on extensive simulation driven by traffic that varies over time and location; sensor nodes are inactive unless they observe some physical event, or send status updates to the sink node providing the connection to the wired world. T-MAC in combination with low-power listening saves most energy, but can not handle the same peak loads as CSMA/CA and S-MAC.

Keywords: medium access control, energy efficiency, idle listening, duty cycle, simulation

1. Introduction

The emergence of wireless sensor networks was sparked by the “Smart dust”-vision: the rapid advances in the integration of digital circuitry can not only be used to develop ever faster systems with more capabilities, but also to shrink the form factor of digital devices to the level that they dissolve in the natural environment [5]. Each node (dust particle) is equipped with some sensors, an embedded processor, and a low-power radio for short-range communication. With such cheap, autonomous nodes large multi-hop, ad-hoc networks can be formed in which cooperating nodes can overcome their inherent individual limitations and provide extensive services, especially in the area of monitoring and control. Potential applications include target tracking, intrusion detection, wildlife habitat monitoring, climate control, and disaster management.

Although sensor networks resemble (ad-hoc) wireless LANs to a large extent, the focus on energy-efficiency to prolong (unattended) life-time and the limited capabilities of each node (i.e., small amounts of RAM) call for new, unorthodox solutions to the many problems that need to be addressed before sensor networks can become widely deployed [1]. We believe that the key to success will be managing the radio, because wireless communication consumes (relatively) large amounts of energy, yet applications must cooperate to process scattered sensor data and relay high-level information to the user.

In a resource-constrained communication system it is important that all of the layers in the protocol stack are optimized towards the specific needs of the application running on top of it. At the moment only limited experience with real-world sensor networks is available through the exploitation of prototype nodes like the family of Berkeley motes [16] in pilot projects like Duck Island [6]. Therefore, we surveyed the targeted remote sensing domain and determined that sensor network applications have some distinctive characteristics that set them aside from their counterparts in typical WLAN scenarios:

- Most of the time nodes do not need to communicate, because interesting events (detecting an intruder, a sudden drop in temperature, etc.) seldomly occur.
- When observing a physical event nodes communicate with their direct neighbors to filter out erroneous sensor readings and increase knowledge, before jointly reporting the event.
- Information (periodic, or event-based) is reported to so-called sink nodes acting as a bridge to fixed infrastructure.

These characteristics imply that the sensor network is usually empty, and that communication is highly structured: to/from the sink and between direct neighbors. In addition, typical messages are small ($\leq$100 bytes) since the in-network processing allows for reporting concise information instead of raw sensor readings.

The observations about traffic in sensor networks impact the design of the MAC protocol as well as the network and transport layers. Instead of optimizing for high throughput, low latency, and fairness, MAC protocols for sensor networks must first and foremost be energy efficient. Consequently, they should be optimized for the case that there is little or no network traffic. Classical MAC protocols for WLANs like 802.11 waste a lot of energy by so-called *idle listening*, that is, listening to receive messages that are never sent.

Idle listening is a serious problem. Consider, for example, a sensor node that needs to forward messages at an average rate of one per second. Messages are fairly short: they take less than 5 milliseconds. This results in the node spending on average 5 ms per second on receiving a message, 5 ms on transmitting it again, and 990 ms on listening while nothing

---

[*] Corresponding author.

happens. The radio is then doing nothing for 99% of the time. The costs of idle listening can be reduced dramatically by putting the radio into sleep mode. On our prototype hardware, for example, switching from receive to sleep mode reduces energy consumption by a factor of 200 (see Section 4). To exploit the large idle:sleep ratio, several energy-saving protocols have been proposed that include a duty-cycle.

In this paper we compare three energy-saving approaches specifically designed for wireless sensor networks: two at the MAC-layer (S-MAC [13] and T-MAC [3]) and one at the physical layer (Low-Power Listening [4]). Through extensive simulations, driven by nodes-to-sink and direct-neighbor communication, we provide a head-to-head comparison showing that T-MAC is the most energy-efficient at the expense of reduced peak throughput. We also report on the (positive) effects of combining the MAC level solutions with the low-power listening at the physical level.

2. Related work

The smooth operation of any wireless network depends, to a large extent, on the effectiveness of the low-level Medium Access Control (MAC) layer responsible for sharing the ether. A MAC protocol determines the next node to access the medium, tries to ensure that no two nodes are interfering with each other's transmissions, and deals with the situation when they do. In addition to idle listening, which was already identified, it is important to minimize the following sources of overhead as well:

collisions: if two nodes transmit at the same time and interfere with each others transmission, packets are corrupted. Hence, the time and energy used during transmission and reception are wasted;

protocol overhead: most protocols require control packets to be exchanged. Since these packets contain no application data, we consider their transmission and reception as overhead;

overhearing: since the ether is a shared medium, a node may receive packets that are not destined for it; it could then as well have turned off its radio to save energy.

TDMA-based protocols, in contrast to contention-based protocols, are very effective at avoiding collisions and have a built-in duty cycle mitigating idle listening. They require, however, some authority (e.g., a dedicated access point) to orchestrate activities within a cell. This complicates their deployment in multi-hop ad-hoc (sensor) networks where nodes are equal and have limited resources.

The popular IEEE 802.11 standard for WLANs is a contention-based protocol that can be operated in ad-hoc mode [7]. Its prime feature, borrowed from the MACAW protocol [2], is its in-band signalling (through RTS/CTS messages) to reduce collisions caused by so-called hidden nodes. Furthermore it includes a power-save mode in which individual nodes periodically listen and sleep. The 802.11 protocol, however, was designed with the presumption that all nodes are located in a single network cell. Adaptations for multi-hop networks have been proposed, but require more complexity and dynamic state than would generally be available in wireless sensor networks [11].

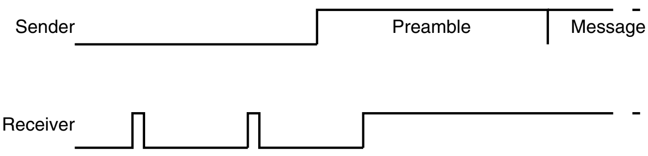

Figure 1. Low-power listening: the sender uses a longer preamble to allow the receiver to only turn its radio on periodically.

The PAMAS protocol [9] uses out-of-band signalling (on another radio channel) to reduce the overhearing overhead, while preserving throughput and latency. Whenever a node overhears signalling destined for another node it calculates the time until the associated data transfer finishes; this is straightforward since the length of the data transfer is included in the control packets on the signalling channel. The radio is then turned off, and will be switched on in due time when the ether becomes available again for other transfers. Energy-savings between 10 and 70% are reported. The out-of-band signalling, however, is not really an option for low-cost sensor nodes since it requires a too complex radio.

3. Energy-saving protocols

Recently, a number of MAC protocols have been developed explicitly targeted at multi-hop ad-hoc sensor networks: low-power listening [4], S-MAC [13], and T-MAC [3]. They all focus on reducing idle listening, but collisions, protocol overhead, and overhearing are also addressed.

Low-power listening. The first approach to reduce idle listening, by introducing a duty-cycle, operates at the physical layer. The basic idea of low-power listening is to shift the cost from the receiver (the frequent case) to the transmitter (the rarer case) by increasing the length of the preamble [4]. This allows the receiver to periodically turn on the radio to sample for incoming data, and detect if a preamble is present or not. If it detects a preamble, it will continue listening until the start-symbol arrives and the message can be properly received (see figure 1). If no preamble is detected the radio is turned-off again until the next sample.

Low-Power Listening (LPL) was applied to CSMA in the TinyOS project with a sample time of 30 μs every 300 μs, a duty-cycle of 10%, reducing the idle listening overhead by a factor of ten. The energy savings come at a slight increase in latency (the length of the preamble is doubled), and minor reduction in throughput. LPL can be applied to any MAC protocol (see Section 5) provided that switching the radio on/off takes little time.

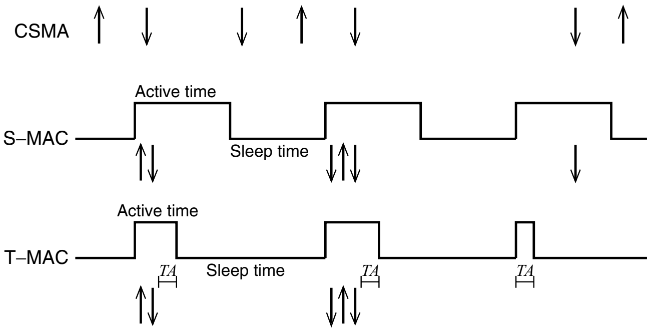

Figure 2. The S-MAC and T-MAC duty cycles; the arrows indicate transmitted and received messages; note that messages come closer together. (*TA* denotes the activity time-out period.)

S-MAC. The second approach mitigating idle listening is the S-MAC protocol, a true MAC protocol, which also addresses the overheads caused by collisions, overhearing, and protocol overhead [13]. The basic idea of this contention-based protocol is that time is divided into –relatively large– frames. Every frame has two parts: an active part and a sleeping part. During the sleeping part, a node turns off its radio to preserve energy. During the active part, it can communicate with its neighbors and send any messages queued during the sleeping part, as shown in figure 2. Since all messages are packed into the active part, instead of being 'spread out' over the whole frame, the time between messages, and therefore the energy wasted on idle listening, is reduced. The exact savings are under control of the application: the active part is fixed[1] to 300 ms, while the frame length can be set to any length. Consequently, also the increase in latency, and reduction in throughput, are under control of the application.

S-MAC needs some synchronization between nodes, but that is not as critical as in TDMA-based protocols: the time scale is much larger with typical frame times in the order of 300 ms to 1 second. S-MAC uses a technique called virtual clustering, in which nodes periodically send special SYNC messages to keep synchronized. These messages, transmitted at the start of a frame, also allow new (mobile) nodes to join the ad-hoc network.

The S-MAC protocol uses the RTS/CTS/DATA/ACK signalling scheme from 802.11 to reduce the number of collisions caused by the hidden-node problem. It borrows the overhearing avoidance technique from the PAMAS protocol, but uses in-band signalling (i.e., overhearing RTS/CTS packets) since the target platform (i.e., Berkeley motes [16]) has only a single-frequency radio, as is the case for most other prototype sensor nodes. Finally, S-MAC includes message passing support to reduce protocol overhead when streaming a sequence of message fragments.

T-MAC. The third energy-saving protocol for sensor networks considered here is the T-MAC protocol [3], which automatically adapts the duty cycle to the network traffic. As with S-MAC, nodes form a virtual cluster to synchronize themselves on the beginning of a frame. But instead of using a fixed-length active period, T-MAC uses a time-out mechanism to dynamically determine the end of the active period. The time-out value, *TA*, is set to span a small contention period and an RTS/CTS exchange. If a node does not detect any activity (an incoming message or a collision) within interval *TA* it can safely assume that no neighbor wants to communicate with it and goes to sleep. On the other hand, if the node engages or overhears a communication, it simply starts a new time-out after that communication finishes (see figure 2).

The adaptive duty-cycle allows T-MAC to adjust to fluctuations in network traffic, both in time (physical events trigger neighbor-to-neighbor communication) and in space (nodes close to the sink relay more traffic than edge nodes). S-MAC, on the other hand, operates with a single active-time for all nodes, which must be chosen conservatively to handle worst-case traffic. The down-side of T-MAC's aggressive power-down policy, however, is that nodes often go to sleep too early: when a node s wants to send a message to r, but loses contention to a third node n that is not a common neighbor, s must remain silent and r goes to sleep. After n's transmission finishes, s will send out an RTS to sleeping r and receive no matching CTS, hence, s must wait until the next frame to try again. T-MAC includes two measures to alleviate this so-called early-sleeping problem, but nevertheless favors energy-savings over latency/throughput much more strongly than S-MAC and LPL (see the simulation results in Section 5).

4. Simulation framework

In a previous study [3] we already reported on how the behavior of S-MAC and T-MAC compares under communication patterns specific to sensor networks. In this paper we extend that work by also studying low-power listening in combination with CSMA (as part of TinyOS) *and* with S-MAC and T-MAC (novel combinations). This section describes our simulation framework, and provides details about the specific settings of the parameters for each protocol.

4.1. Overview

The simulator is constructed using the OMNeT++ discrete event simulation package [12], in which we have built a realistic model of the EYES prototype wireless sensor nodes [15] in use at our research laboratory. The EYES nodes consist of a 16-bit embedded processor (Texas Instruments MSP430F149), a low-power radio (RFM TR1001, 868.35 MHz, hybrid transceiver), a 2 Mbit EEPROM memory, and various connectors to interface to the outside world. A node runs from 3 V supplied by two AA batteries taking up most of the node's volume. Table 1 provides a power

---

[1] A recent enhancement of S-MAC, which is called *adaptive listening*, includes a variable length active part to reduce multi-hop latency [14]. Since the time-out policy of the T-MAC protocol behaves similarly and was designed to handle traffic fluctuations as well, we do not discuss adaptive listening further.

Table 1
Power breakdown of EYES nodes.

CPU (5 MHz)	
Active	2.1 mA
Sleep	1.6 μA
Radio (115 kbps)	
Transmit	10 mA
Receive	4 mA
Sleep	20 μA

Table 2
Implementation details of the simulator.

Radio bit rate	115 kbps
Channel coding	8-to-12 bit coding
Control packets	8 bytes
DATA packets	6–256 bytes
SYNC packets	10 bytes
T-MAC	
Frame time	610 ms
Activity time-out	15 ms
Contend time	9.15 ms
S-MAC	
Frame time	1 s
Active period	61–915 ms
Contend time	3.05–9.15 ms

breakdown of the processor and radio for active and sleep modes according to the component specifications [8,10].

The OMNeT++ model has the same limits on clock resolution and precision (32 KHz crystal), radio turn-around and wake-up times (43 μs and 10 μs respectively), and transmission bit rates (115 kbps) as the EYES nodes have. Energy consumption in the model is based on the amount of energy the radio uses; we do not take protocol processing costs on the CPU into account. Preliminary power measurements of the EYES nodes have shown that the numbers in Table 1 are quite realistic, except for the transmit power that was measured with a peak of 18 mA. Since it is not clear what is causing this discrepancy, we use the specified power numbers in Table 1 in our simulations.

In our experiments we have used a static network with a 10 × 10 grid topology. We have chosen a radio range so that non-edge nodes all have 8 neighbors. Concurrent transmissions are modeled to cause collisions if the radio ranges (circles) of the senders intersect; nodes in the intersection will receive a garbled packet with a failing CRC check.

The application is modeled by a traffic generator at every node. The generator is parameterized to send messages either to direct neighbors (i.e., nodes within the radio range of the sender), or to the sink node, which is located in the left bottom corner of the grid. To route the latter messages to the sink we use a randomized shortest path routing method; for each message, the possible next hops are enumerated. Next hops are eligible if they have a shorter path to the final destination than the sending node. From these next hops, a random one is chosen. Thus messages flow in the correct direction, but do not use the same path every time. No control messages are exchanged for this routing scheme: nodes automagically determine the next hop. By varying the message length and inter arrival times we can study how the protocols perform under different loads.

The implementation details of the four energy-saving protocols under study are discussed below. Table 2 provides a brief summary, listing the most important radio and protocol parameters.

LPL. In the simulator we have chosen to implement Low Power Listening with the same parameters as on the TinyOS/Mica1 platform. This means we have used a 10% duty-cycle with an on period of 30 μs and an off period of 270 μs. LPL is implemented as a thin layer between the MAC and physical layer such that we can easily control whether or not LPL should be applied and in combination with which MAC protocol. Whenever the MAC layer turns the radio to receive mode, LPL will transparently poll the physical layer with the 30 μs sample time. This short probe is possible with our radio which can be activated in about 10 μs. When the MAC switches to send mode the physical layer is instructed to stretch the preamble by 300 μs effectively doubling the preamble time (347 vs. 647 μs). Hence, the penalty for sending under LPL is acceptable given that even a very short control message like an RTS takes 1.4 ms to transmit without the use of LPL and 1.7 ms with LPL.

CSMA/CA. The basic CSMA/CA (802.11) protocol was implemented using an 8 byte header encoding all information needed in the control messages, that is, the RTS/CTS/DATA/ACK messages. The payload of the DATA message can be up to 250 bytes (the routing information is part of the DATA header). Furthermore we implemented overhearing avoidance. Reliability is implemented by restarting the transmission when an expected reply packet (CTS/ACK) is not received.

T-MAC. For the T-MAC protocol we use a frame time of 610 ms and a 15 ms time-out value (*TA*) to end the (adaptive) active time. This 15 ms is set to span the contention period (9.15 ms), an RTS (1.7 ms), the radio turnover period (43 μs) and the start of a CTS. These settings cause T-MAC to operate with a 2.5% duty cycle in an empty network. In a loaded network the duty cycle will increase as the active period is adaptively extended. To warrant reasonable throughput levels in our experiments we set T-MAC to include all options for mitigating the early-sleeping problem (i.e., the *future request-to-send* and *full buffer priority* options are activated), and overhearing avoidance is also used, see [3] for details.

For the T-MAC protocol we used the same message layout and reliability features as for the CSMA/CA protocol. The SYNC messages, which are specific to T-MAC and S-MAC are 10 bytes long (control header and a 2 byte timestamp) and are issued once every 90 seconds on average. In our grid topology with 8 neighbors within radio range, that amounts to receiving a SYNC message every 11 seconds.

S-MAC. We implemented the S-MAC protocol by reusing as much of the T-MAC protocol as possible, hence, the basic parameters like MAC header length differ slightly from the real implementation by Ye et al. [13]. The code re-use approach also led us to use a fixed frame time of 1 second in combination with a parameterized active time, which was varied between 61 and 915 ms. Although this is the opposite of the original (fixed active time, variable frame length) a similar parameterizable duty cycle [6–92%] is obtained. Overhearing avoidance is activated for S-MAC since it proved beneficial, especially for longer active periods. In our experiments we do not send large messages across the network, so the message passing support of S-MAC was not needed, nor implemented.

5. Experiments

In this section we provide a head-to-head comparison of CSMA/CA, S-MAC, and T-MAC both with and without LPL, by comparing their energy usage under several sensor-network specific scenarios with different communication patterns and varying traffic loads. Each reported energy use is the average of 10 simulation runs with the same settings, but with different seeds for the random number generators.

Since throughput is not completely unimportant, we will end each curve at the point where less than 90% of the messages are correctly received. In multi-hop patterns this means that 90% of all messages must reach their final destination (i.e., the sink node). Messages can only get lost due to queue overflows. All corruptions at the physical layer (e.g., collisions) are detected at the MAC layer through missing ACKs, and retransmissions are initiated causing messages to be dropped because of queue overflow (head of line blocking).

5.1. Event-based unicast

The first experiment concerns local traffic exchanged between neighbors when observing the same physical event in some part of the sensor network. Once every 10 seconds the traffic generator randomly selects an area in the grid where such an event takes place; this area includes approximately 9 nodes, and the event lasts for about 5 seconds. During the event, the observing sensor nodes send (unicast) messages to their neighbors, some of which are also observing the same event. On reception of a message a node will, with a probability of 20%, send a reply message modeling cooperation at the application level. A simulation parameter controls the message frequency during events. Figure 3 plots the energy consumption for different peak loads [2–160 bytes/node/sec] during an event.

For reference figure 3 includes the performance of the CSMA/CA protocol *without* low-power listening. Since plain CSMA/CA operates without a duty cycle, and the network is empty most of the time, the radio is effectively set to receive mode using 4 mA. Note that the curve descends slightly with increasing traffic. This is caused by the overhearing avoidance mechanism that puts nodes into sleep whenever an unrelated communication takes place within radio range. Apparently, the benefits of overhearing avoidance outweigh the increased costs of transmitting (10 mA) vs. receiving (4 mA).

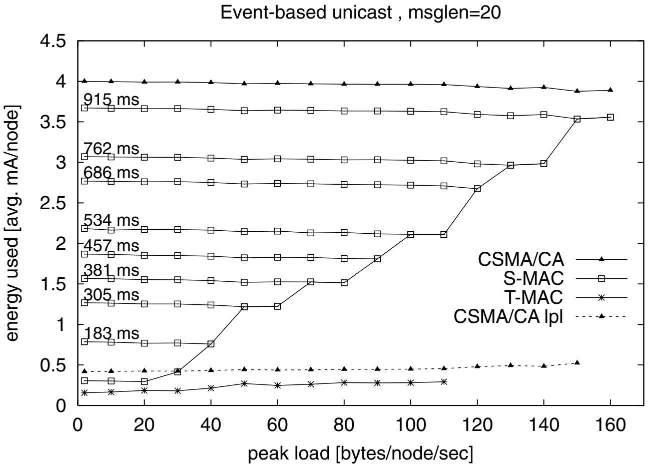

Figure 3. Event-based unicast performance. For S-MAC both the per parameter lines and the per load tuned line are drawn.

The energy consumption of the S-MAC protocol depends on the choice of the duty cycle. We have experimented with various lengths of the active-period parameter, each yielding a CSMA/CA-shaped curve at a fraction of the energy consumption that roughly corresponds to the duty cycle. The lower the duty cycle, however, the sooner S-MAC can not handle the peak load causing more than 90% of the messages to be dropped, hence, cutting the curve short. This points at a serious drawback of S-MAC: the optimal value for the active-period is load-dependent, hence, the application (or user) is left with the difficult task of selecting the duty cycle that compromises best between energy savings and message delivery rate. For now we assume that the optimal choice will be made, and in the remaining graphs we will just plot the best value for each individual load, that is, the line connecting all cut-off points.

Applying low-power listening (10% duty cycle) to CSMA/CA has a dramatic effect: the energy consumptions drops from 4 to 0.42 mA for a (near) zero load, only slightly increasing for higher traffic loads (0.52 mA at 150 bytes/node/second peak). The increased transmission costs, due to the longer preamble, are not compensated by the overhearing avoidance as for plain CSMA/CA. Nevertheless LPL is very effective without impacting network throughput as in the case of S-MAC.

Finally, the lowest energy consumption is obtained by T-MAC. At zero load it outperforms CSMA/CA with LPL with a factor of 2.5 (0.42 mA vs. 0.16 mA). For higher loads, T-MAC automatically uses a larger active-period and approaches the CSMA/CA-lpl curve. Note, however, that the T-MAC curve stops at a peak load of 110 bytes/node/sec; due to the early-sleeping problem, T-MAC fails to deliver at least 90% of the messages. This effect can be countered at the application level, by having nodes aggregate multiple messages into one large message.

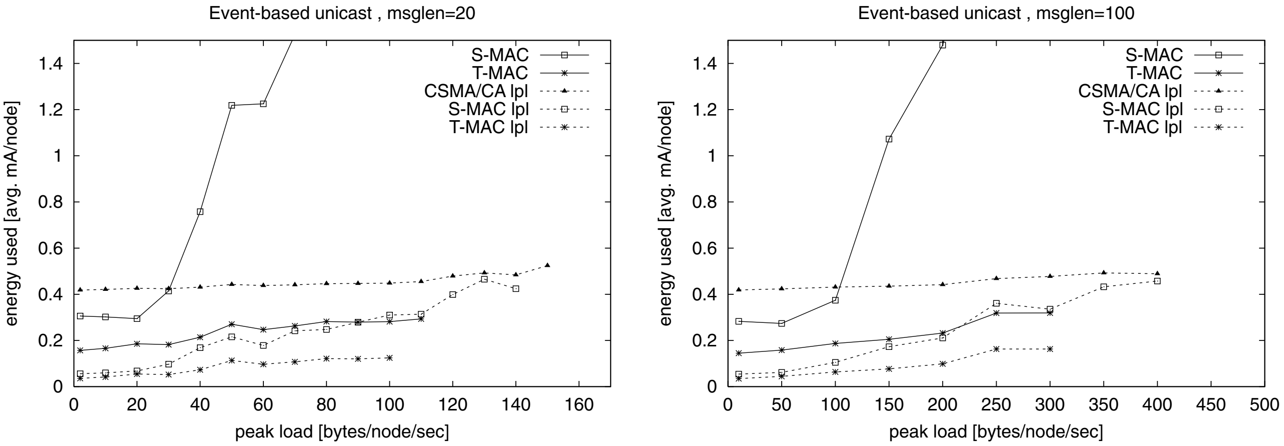

Figure 4. Event-based unicast performance for small messages (left) and large messages (right).

Figure 4 plots the energy consumption of event-based unicast traffic for both small (20 byte) and large (100 byte) messages and zooms in on the bottom part of the y-axis. Note that the plain CSMA/CA curve at 4 mA is no longer visible. With large messages T-MAC can sustain a load of 300 bytes/node/sec, a three-fold increase over small messages, while consuming the *same* amount of energy. Figure 4 also shows the positive effects of combining low-power listening at the physical layer with the MAC-level solutions implemented by S-MAC and T-MAC. (Note that the S-MAC and T-MAC curves with low-power listening use the same symbols as before, but connected with a dotted line.) For S-MAC the effect of employing low-power listening is quite large: 1) at zero load the energy consumption is reduced with a factor of 5.5, and 2) the inclination of the curve is reduced with roughly a factor of 8, showing that LPL for S-MAC is effective irrespective of the load.

For T-MAC low-power listening is also beneficial, but not as much as for S-MAC since T-MAC includes less idle-listening time due to its adaptive schedules. For near zero load the energy consumption of T-MAC drops with a factor of 4.7 (0.17 mA vs. 0.036 mA), which is less than expected since LPL has a 10% duty cycle. The reason is that due to T-MAC's aggressive power-off strategy the ratio of power used on sending and receiving messages to the power used on idle listening is already quite favorable. This also explains why the relative gain of LPL for T-MAC reduces when the load increases: the fraction of idle listening is gradually reduced when more messages are transmitted.

5.2. Nodes-to-sink

The second experiment considers the important case where sensor nodes periodically report their status to the sink (in the left bottom corner of the grid). As described before, messages are routed with a randomized shortest path algorithm to spread the load among multiple trajectories. Nevertheless, the nodes on the right-top to left-bottom diagonal must handle most traffic. No message aggregation at the routing layer is applied. As with event-based unicast the traffic generators can be run with different injection rates and message sizes to control the traffic characteristics.

Figure 5 shows the performance of the three MAC protocols with and without LPL (except plain CSMA/CA at 4 mA).

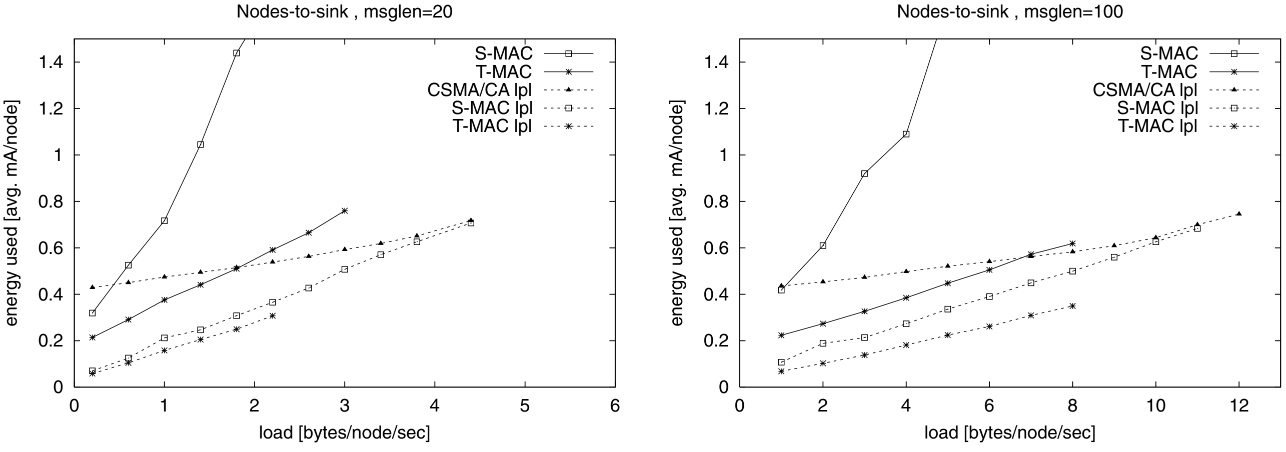

Figure 5. Nodes-to-sink performance for small messages (left) and large messages (right).

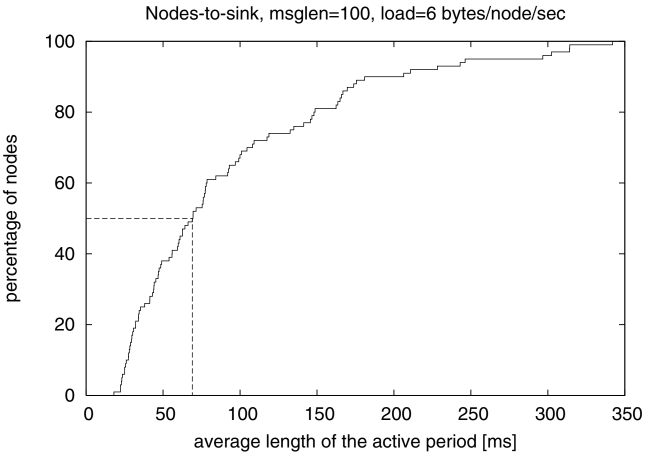

Figure 6. Cumulative distribution of active-times for T-MAC.

The first thing to notice, is that in comparison to event-based unicast, the generated load (plotted along the x-axis) is two orders of magnitude lower. This is caused by two effects: (1) messages now travel about 6.2 hops to reach the sink vs. one hop to a neighbor, and (2) all 100 nodes generate traffic at a constant rate instead of just 9 during the lifetime of an event. Thus, the same injection rate for event-based unicast and nodes-to-sink generates about 70 times more traffic network-wide for the latter. Since all traffic is destined for the sink, it becomes a bottleneck for higher traffic loads limiting the maximum throughput of CSMA/CA and S-MAC. The throughput of T-MAC, on the other hand, is limited by the early-sleeping problem.

The second observation is that T-MAC has more difficulty handling the nodes-to-sink pattern than both S-MAC and CSMA/CA. T-MAC without LPL uses more energy than CSMA/CA with LPL for high loads, which was never the case for event-based unicast, and the gap between T-MAC and S-MAC is a lot smaller (with or without LPL). Nevertheless, T-MAC still has the advantage, especially for larger messages, because it adapts to the load at each node. Figure 6 illustrates this aspect. It shows the cumulative distribution of the active times of the T-MAC protocol with LPL for message size 100 and a load of 6 bytes/node/sec. From this graph we can see that 50% of the nodes have an average active period of less than 70 ms (11% duty cycle), while the busiest node runs with a 342 ms active period (56% duty cycle). S-MAC in this case performs best with a duty cycle of 23% at *each* node. This shows that half of the nodes run with a duty cycle that is at least a factor of two too high. The resulting abundance of idle listening is effectively dealt with by low-power listening, hence, the remarkably good performance of the S-MAC with LPL combination. The effects of T-MAC's adaptive timeout mechanism are more pronounced for larger message sizes because S-MAC's fixed activity period is proportional to the message length, hence the idle-listening time of most nodes is proportional to the message length, while T-MAC's idle-listening time is basically constant (i.e., the 15 ms time-out).

Table 3
Energy usage [mA] for the full scenario.

	CSMA/CA		S-MAC		T-MAC	
	–	LPL	–	LPL	–	LPL
Event-based	3.97	0.43	0.37	0.11	0.19	0.06
Nodes to sink	3.97	0.44	0.38	0.08	0.21	0.07
Full scenario	**3.93**	**0.46**	**0.80**	**0.18**	**0.28**	**0.12**

5.3. Full scenario

This final experiment combines both previous communications patterns into a full sensor network scenario: nodes in the network observe physical events and report their findings periodically to the sink. In the case of detecting an event, nodes send a 100 byte (unicast) message once per second to a randomly selected neighbor for the duration of the event (5 seconds). One of the approximately 9 nodes detecting the same event reports their joint findings to the sink by sending, also at a rate of once per second, an additional 100 byte multi-hop message. All nodes (including those observing an event) send regular 20 byte status messages to the sink once per minute.

Table 3 lists the energy consumptions of the three MAC protocols with/without low-power listening. Note that LPL is very effective for each of the MAC protocol reducing energy consumption with a factor of 8.5 for CSMA/CA, 4.4 for S-MAC, and 2.7 for T-MAC. Even though T-MAC does not benefit as much, it achieves the lowest energy consumption of all combinations: T-MAC outperforms CSMA/CA with a factor 3.8 and S-MAC with factor 1.5 (all with LPL).

For reference Table 3 also includes the energy consumption of the underlying event-based unicast and nodes-to-sink communication patterns (cf. figures 4 and 5). Although the nodes-to-sink communication generates the most traffic, adding the event-based unicast traffic to it has different effects for different MAC protocols with/without LPL. The energy consumption of CSMA/CA is hardly affected, that of S-MAC is increased noticeably. These differences may be caused by queues overflowing more rapidly when packing traffic into active periods as S-MAC and T-MAC do. Additional research is needed to determine exactly what factors contribute to the observed behavior.

6. Discussion

From the experiments we conclude that Low-Power Listening at the physical layer is a good idea and large energy savings are gained by using it. The negative effects of stretching the preamble (347 vs. 647 μs) are limited: throughput and latency are hardly affected by LPL itself. The exact reductions in energy consumption that can be obtained depend on three factors:

(1) the duty cycle (fixed at 10% in our experiments),
(2) the specific MAC protocol running on top of it, and
(3) the network traffic load generated by the application.

These factors are dependent making it hard to draw general conclusions. What we did learn though, is that the relative impact of LPL is inversely related to the degree at which the MAC protocol is optimized for the sensor network specific traffic characteristics. That is, LPL achieves the greatest reduction in energy consumption for CSMA/CA, the reduction for S-MAC is somewhat less, and LPL is the least effective for T-MAC. Nevertheless, in the full scenario LPL still reduces the energy consumption of T-MAC with a factor of 2.3, which is significant indeed.

When reviewing the effectiveness of the two MAC protocols designed for wireless sensor networks we conclude that T-MAC outperforms S-MAC because of its capabilities to adjust to the fluctuations, in time and space, of the network traffic. Without LPL the energy consumption of S-MAC deteriorates rapidly when the volume of network traffic increases, because the fixed duty-cycle must be tuned to handle the peak loads. The resulting idle-listening overhead is well handled by LPL, hence the decent performance of S-MAC with low-power listening. T-MAC, on the other hand, already achieves acceptable energy savings *without* LPL. The down-side of T-MAC's aggressive time-out policy is that it can not cope with high message rates. In this case, however, data aggregation at the routing and/or application layer can probably be applied to alleviate the load on the busiest nodes.

The advantage of T-MAC over S-MAC is most pronounced for the event-based communication pattern, since in this case the load at each node toggles between busy and dormant resulting in very bursty traffic over time. For nodes-to-sink reporting the network load depends on the location of the nodes, with nodes on the diagonal towards the sink handling the most messages. In this case, the energy saved by a tailor-made active time (T-MAC) over a fixed, worst-case active time (S-MAC) is of little significance, because of all energy spent on the useful work of receiving and transmitting messages.

Low-power listening at the MAC layer. To leverage the elegant approach of low-power listening on platforms where the radio can not be controlled to such a detailed level as generating 30 μs samples and stretching preambles, we have performed some preliminary experiments with a modified version of T-MAC that polls the ether at the medium access layer (ms resolution). Q-MAC is optimized to handle Quiet networks (i.e., typical sensor networks). The basic idea is that the energy consumption of T-MAC can only be reduced by lowering the time-out value *TA*, which is currently set to 15 ms (see Table 2). The key component that determines *TA* is the contend time (9.15 ms) needed for collision avoidance. In quiet networks the chances of collision are by definition very low, so Q-MAC takes the drastic measure of removing the contention interval altogether. Therefore, a node that has a message queued when waking up at the beginning of a frame, immediately sends an RTS to the destined neighbor r. This will trigger r to send a CTS; all other nodes in the vicinity can go to sleep. If two nodes send an RTS at the same time, the surrounding nodes will observe the collision and proceed with an ordinary T-MAC frame with contention interval. Both senders will detect the collision indirectly through the missing CTS, and also proceed with the normal T-MAC procedure.

Table 4
Energy usage [mA] for Q-MAC.

	T-MAC	Q-MAC	T-MAC lpl
Event-based	0.19	0.11	0.06
Nodes-to-sink	0.21	0.14	0.07
Full scenario	0.28	0.35*	0.12

*Only 84% of the messages was received correctly.

The end result is that under Q-MAC nodes only need to listen for 1.5 ms, the transmit time of an RTS, versus 15 ms for T-MAC. Thus, the potential reduction in energy consumption is a factor of ten. In practice, the network will not be completely empty because, at least, SYNC packets must be exchanged and applications do generate some traffic. To asses Q-MAC's potential for large energy savings, and its anticipated sensitivity to network load, we repeated the full-scenario experiment with Q-MAC, as well as the underlying event-based unicast and nodes-to-sink experiments.

The results in Table 4 give a mixed signal. For light loads (event-based unicast and nodes-to-sink) Q-MAC strikes a middle ground between plain T-MAC and T-MAC combined with low-power listening. The reductions in energy consumption, however, do not even come close to Q-MAC's tenfold potential and are limited to less than a factor of two. For moderate loads (i.e., the full scenario) Q-MAC's performance is counter productive: the energy consumption increases, and the message delivery rate drops below acceptable levels (i.e., 90%). For now, we conclude that the Q-MAC approach is too sensitive to load, and we plan to investigate the exact causes and possible counter measures so as to make low power listening at the MAC layer a viable alternative.

7. Conclusions

Wireless sensor networks hold a great potential for ubiquitous applications in the area of (remote) sensing and control. A key issue that needs to be addressed is the efficient operation of the radio link to foster collaboration between individual resource-scarce sensor nodes, on the one hand, and to minimize the energy consumption to extend lifetime on the other hand. The focus on energy consumption requires special solutions since typical communication protocols for wireless LANs are designed to achieve high throughput, low latency, and fairness.

In this paper we have compared three approaches for saving energy in sensor networks: low-power listening at the physical layer, and S-MAC and T-MAC operating at the medium access layer. These three approaches have in common that they introduce a duty-cycle to mitigate idle listening, the dominant cause of energy consumption in typical sensor network scenarios where applications communicate seldomly. Low-power listening extends the length of the preamble, so receivers can probe the ether periodically. S-MAC synchronizes nodes, and introduces frames in which nodes only listen at the beginning for some fixed active time. T-MAC also operates with frames, but the length of the active time is adapted to the network traffic through a simple time-out mechanism.

Through extensive simulation we studied the energy savings of S-MAC and T-MAC over standard CSMA/CA (IEEE 802.11) in combination with and without low-power listening. We used a traffic generator to model typical sensor network behavior: neighbor-to-neighbor communication when observing a physical event, nodes-to-sink reporting for relaying status updates, and a full scenario that combines both patterns. The results show that

- Low-power listening is very effective at mitigating idle-listening; it has the highest relative impact on CSMA/CA, but the absolute lowest energy consumption is reached in combination with T-MAC.
- T-MAC's aggressive time-out policy allows it to adapt seamlessly to variations in traffic induced by typical sensor network applications at a the expense of a reduction in peak throughput. T-MAC performs slightly better for variations over time (events), than for variations in location (periodic reporting).
- S-MAC suffers from over-provisioning. Since its duty cycle is fixed for all nodes often a rather large value must be selected to avoid dropping messages under peak loads, which causes S-MAC's idle-listening to deteriorate for increasing traffic loads. When combined with low-power listening, however, S-MAC achieves acceptable results, but not as good as those of T-MAC with low-power listening.

Since T-MAC with LPL performs best, but not all platforms allow a very precise control of the radio at the μs scale, we briefly discussed a novel approach (Q-MAC) that implements the idea of low-power listening at the MAC level (ms scale). Preliminary results indicate that additional research is need to make it a viable technique.

Acknowledgments

We like to thank Jason Hill for his clarifications about the details of TinyOS' Low-Power Listening approach. We thank Tom Parker and Niels Reijers for discussing and interpreting the simulation results, and for commenting on the draft version of this paper.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, A survey on sensor networks, IEEE Communications Magazine 40(8) (2002) 102–114.

[2] V. Bharghavan, A. Demers, S. Shenker and L. Zhang, MACAW: A media access protocol for wireless LAN's, in: *Conf. on Communications Architectures, Protocols and Applications* (London, August 1994) pp. 212–225.

[3] T. van Dam and K. Langendoen, An adaptive energy-efficient MAC protocol for wireless sensor networks, in: *1st ACM Conf. on Embedded Networked Sensor Systems (SenSys 2003)*, (Los Angeles, CA, November 2003) pp. 171–180.

[4] J. Hill and D. Culler, Mica: A wireless platform for deeply embedded networks, IEEE Micro 22(6) (2002) 12–24.

[5] J. Kahn, R. Katz and K. Pister, Next century challenges: Mobile networking for ''Smart Dust", in: *5th ACM/IEEE Int. Conf. on Mobile Computing and Networks (Mobicom '99)*, (Seatle, WA, August 1999) pp. 271–278.

[6] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler and J. Anderso, Wireless sensor networks for habitat monitoring, in; *First ACM Int. Workshop on Wireless Sensor Networks and Application (WSNA)* (Atlanta, GA, September 2002) pp. 88–97.

[7] LAN MAN Standards Committee of the IEEE Computer Society, *IEEE Std 802.11-1999, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*. IEEE, 1999.

[8] RFM, *TR1001 868.35 MHz Hybrid Tranceiver*.

[9] S. Singh and C. Raghavendra, PAMAS: Power aware multi-access protocol with signalling for ad hoc networks, ACM SIGCOMM Computer Communication Review 28(3) (1998) 5–26.

[10] Texas Instruments, *MSP430x1xx Family User's Guide*. SLAU049B.

[11] Y. Tseng, C. Hsu and T. Hsieh, Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks, in: *21st Conf. of the IEEE Computer and Communications Societies (INFOCOM)* 1 (June 2002) vol. 1, pp. 200–209.

[12] A. Varga, The OMNeT++ discrete event simulation system, in: *European Simulation Multiconference (ESM'2001)* (Prague, Czech Republic, June 2001).

[13] W. Ye, J. Heidemann and D. Estrin, An energy-efficient MAC protocol for wireless sensor networks, in: *21st Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 3 (June 2002) vol. 3, pp. 1567–1576.

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

[15] http://eyes.eu.org/sensnet.htm.

[16] http://webs.cs.berkeley.edu/tos/.

Gertjan Halkes earned a B.Sc. and an M.Sc. in computer science from the Delft University of Technology in 2002 and 2004 respectively. He is currently working as a Ph.D. student in the Parallel and Distributed Systems group at that same university.
E-mail: G.P.Halkes@ewi.tudelft.nl

Tijs van Dam received an M.Sc. in computer science from Delft University of Technology, The Netherlands in 2004. He is now working for IBM.
E-mail: tijsvd@xs4all.nl

Koen Langendoen is an associate professor in the Parallel and Distributed Systems group at Delft University of Technology, The Netherlands. He earned an M.Sc. in computer science from the Vrije Universiteit, Amsterdam in 1988 and a Ph.D. in computer science from the Universiteit van Amsterdam in 1993. His research interests include system software for parallel processing, wearable computing, embedded systems, and wireless sensor networks.
E-mail: K.G.Langendoen@ewi.tudelft.nl