Increasing the efficiency of preamble sampling protocols for wireless sensor networks *

Marco Avvenuti, Paolo Corsini, Paolo Masci and Alessio Vecchio

Dipartimento di Ingegneria dell'Informazione, Università di Pisa
Via Diotisalvi 2, 56100 Pisa, Italy

email: `firstname.lastname@ing.unipi.it`

Abstract

Applications designed for event driven monitoring represent a challenging class of applications for wireless sensor networks. They are a special kind of monitoring applications, since they usually need low data rates, but also require mechanisms for low latency and asynchronous communication. In this paper we will focus on optimizations at the MAC layer that enable low energy consumption when contention-based protocols are adopted. We present B-MAC+, an enhanced version of a widely adopted MAC protocol, and we show that substantial improvements, in terms of network lifetime, can be reached over the original protocol.

1 Introduction

A sensor network is a wireless network composed of a large number of communicating nodes used to monitor a physical phenomenon. Each node consists of an embedded microcontroller with a small amount of memory, a battery, a wireless transceiver, and may be equipped with various sensing hardware (light, temperature, etc.). The network is self-organizing and multi-hop communication is used to transport the collected data to a monitoring base station. Sensor networks can be adopted in different scenarios, such as environmental monitoring, building automation, and e-health. As known, one of the main drawbacks of sensor networks is represented by energy management: in order to have a reasonably long network lifetime, energy savings must be adopted at all level of software/hardware infrastructure.

With current hardware technology, the most of the energy is spent by the radio transceiver. Besides the energy spent during transmission and reception of messages, also the energy consumption due to listening on the channel waiting for incoming packets is relevant (idle listening). To limit the energy consumption due to idle listening, the most effective solution is to turn on and off the radio repetitively (duty cycling). This can be done efficiently at the MAC level. Possible approaches are represented by *schedule-based* and *contention-based* protocols. In schedule-based protocols [5, 7, 8, 1, 9] there is a common schedule that reserves a period for transmission to each node. Thus, the receiver knows when the radio must be turned on to receive possible incoming data. In contention-based protocols [10, 3, 11] the node periodically turns on the radio and checks for activity on the channel. If the channel is found busy, the radio is kept in receive mode and data is received, otherwise the node goes back into the sleep state. Both approaches have pros and cons: schedule-based protocols are more complex to be implemented and may suffer high latency, but at the same time they provide some guarantees to nodes; contention-based protocols are simpler, mainly because of the lack of a shared scheduling policy, and more tolerant to the appearance and disappearance of nodes (e.g. due to mobility), but they may lead to poor performance under high traffic loads.

In this paper we focus on contention-based proto-

*This work is partially supported by Fondazione Cassa di Risparmio di Pisa, Italy (SensorNet Project).

cols, and we present B-MAC+, an enhancement of B-MAC, a popular protocol for wireless sensor networks. In particular, we make clear how receivers running the standard B-MAC protocol waste energy during the reception of long preambles and we propose a more efficient solution. An analytical evaluation of the achievable benefits shows that the lifetime of a wireless sensor networks can be significantly prolonged. An experimental evaluation on real nodes is also performed in order to validate the analytical results.

2 Event driven applications: a case study

One of the most diffused classes of applications is represented by event driven applications: nodes monitor one or more crucial variables and communicate with a base station only when a given condition occurs. Examples include the early discovery of a forest fire, intrusion detection, and notification of critical health conditions in patients.

Event driven applications are characterized by *low traffic rate* and require *low latency* communication. In the absence of events, all nodes should remain in the sleep state, with the exception of some periodical communication that can be performed to ensure that nodes are still operational. Also, when the triggering condition is detected, information must be reported to the base station as quickly as possible, e.g. to ensure proper counteractions.

Here, as a case study in the field of event driven applications, we describe the main structure of an application to detect and contrast forest fires, and we evaluate the expected lifetime using current hardware platforms. The application is intended to notify the base station about the location of possible fires, and to provide support to fire fighters, in terms of direction of flames, temperature, wind and so on.

Let us suppose that nodes are deployed uniformly in the area under control, and that every node has 10 neighbors. All nodes execute the same application. The application behaves as follows: every 5 minutes a node acquires new environmental data (mainly the temperature) by means of its sensing equipment. If the acquired value is greater than a given threshold the node sends an alert message to the base station and additional messages to support the action of fire fighters. In any case, the node also transmits a message every 15 minutes to inform the base station that it is working properly. Nodes in the network are not supposed to be synchronized, thus there is no correlation of time instants between nodes. This leads to a global asynchronous behavior of the network, both for sampling and communication. Notice also that multi-hop paths are required to relay data, since the radio range is limited to one hundred meters. Moreover, nodes must be able to relay messages at any time, since low latency is crucial for this application.

In order to meet these application requirements, we chose to adopt a contention-based MAC protocol, as it can extend network lifetime while guaranteeing an acceptable communication latency.

2.1 Evaluating lifetime of sensor nodes

The lifetime of a sensor node can be computed on the basis of the capacity of the batteries, and the total energy spent by the node. The total energy spent by a node can be computed as:

$$E = E_{listen} + E_{rx} + E_{tx} + E_d + E_{sleep} + E_{leak} \quad (1)$$

where E_{listen} is the energy spent to check for activity on the channel, E_{rx} to receive packets, E_{tx} to transmit, E_d to acquire new values from the sensing hardware, E_{sleep} is the energy spent in the low power state, and E_{leak} is the energy leaked when turning on the radio for transmissions and receptions of packets. If we consider the MICA2 platform and we refer to a basic contention-based protocol like B-MAC [10], it can be shown that the lifetime of a network running the application described above is about 1.5 years.

To further extend the lifetime of the network, application designers should focus mainly on E_{rx} and E_{listen}, since they are usually dominant factors for the overall energy consumption. In fact, E_{rx} is more relevant than E_{tx}, since it is often the case that each packet is received by multiple nodes (for example, in our scenario every node has 10 neighbors). The same applies also to E_{listen}, since even if checking for channel activity is less expensive than transmitting, such a task is executed orders of magnitude more frequently than transmissions. E_{leak} does not influence the total consumption significantly, as it is negligible, for current radio transceivers, compared to the total amount of energy spent in transmitting and receiving packets. Finally, E_d and E_{sleep} cannot be significantly reduced, as they depend on the application requirements and on the characteristics of the hardware platform.

2.2 Extending lifetime: problem statement

Contention-based protocols rely on preamble[1] sampling techniques [4, 2], which means that the receiver periodically turns on the radio and checks for activity on the channel. If activity is detected, the radio is left in the receiving state to receive the remaining part of the preamble and the data payload. Otherwise, if the channel is idle, the radio is switched off. This approach requires the cooperation of sending nodes that must use wake-up preambles long enough as to bridge the gap between two consecutive receiver's awaken times.

Hereafter we will use the term preamble to denote the short bit pattern matched to lock the phase on the receiver side, while we will use the term wake-up preamble to denote the, possibly long, sequence of data used for detecting activity in preamble sampling protocols.

Long wake-up preambles limit the energy efficiency: every time a node detects activity on the channel, it must remain in the receive state until the data payload is received, even if it is not the intended recipient. The energy spent in the receive state is:

$$E_{rx} = t_{rx} * c_{rxb} * V \tag{2}$$

where t_{rx} is the time a node remains in the receive state, c_{rxb} is the current absorbed in the receive state and V is the voltage.

With contention-based protocols that use preamble sampling techniques, the time spent in the receive state can be described by the following equation:

$$t_{rx} \leq n * r * (L_{wakeup\_preamble} + L_{data}) * t_{rxb} \tag{3}$$

where n is the number of neighbors, r is the packet rate, $L_{wakeup\_preamble}$ is the length of the wake-up preamble, L_{data} is the length of the data payload, and t_{rxb} is the time to receive a byte. In fact, the receiver wakes up and checks for activity at a random time during the transmission of the preamble, then it has to receive the remaining part of the preamble and the data payload.

Clearly, in order to reduce E_{listen}, the check interval (i.e. the time between two consecutive checks for activity on the channel) should be extended. However, this has a negative impact on E_{rx}, since the length of the preamble is proportional to the check interval. As a consequence, we need a mechanism that makes the time spent receiving the preamble independent from the check interval.

3 B-MAC+

B-MAC+ is an enhancement of B-MAC, a popular contention based protocol. The basic idea behind B-MAC+ consists in replacing the pattern of the wake-up preamble of B-MAC, with a new pattern that contains information about the size of the remaining part of the preamble not yet transmitted. This information can be used by receivers to avoid wait states during significant portions of the preamble transmission time, i.e. going to sleep and waking up when the data payload is actually transmitted.

Basically, the wake-up preamble of B-MAC+ is obtained by dividing into chunks the wake-up preamble of B-MAC. Each chunk is a *countdown packet*, which consists of a short preamble and a data payload. The data payload contains a counter and an address: the counter represents the size of the remaining part of the wake-up preamble, the destination address specifies the actual recipient of the data packet.

An example of transmission and reception of a BMAC+ packet is depicted in Figure 1. The sender starts to transmit the countdown packets with an initial value equal to K. This value depends on the check interval of the receiver, that in turns depends on the duty cycle required by the application. When the channel is sampled and activity is detected, two nodes, say N1 and N2, listen to the channel until they receive an entire countdown packet. As the destination address is N1, N2 switches to the sleep mode while the transmission is completed (i.e. for a time equals to the check interval). Instead, N1 uses the counter value (J) to determine the duration of the remaining part of the wake-up preamble, then it also switches to the sleep mode but will wake up just before the beginning of the the payload transmission time.

[1]A packet sent by the radio consists of a preamble and a data payload. The preamble, which is usually few bytes long, contains a bit pattern used to lock the phase of the receiver. The data payload contains information useful to MAC and higher layers.

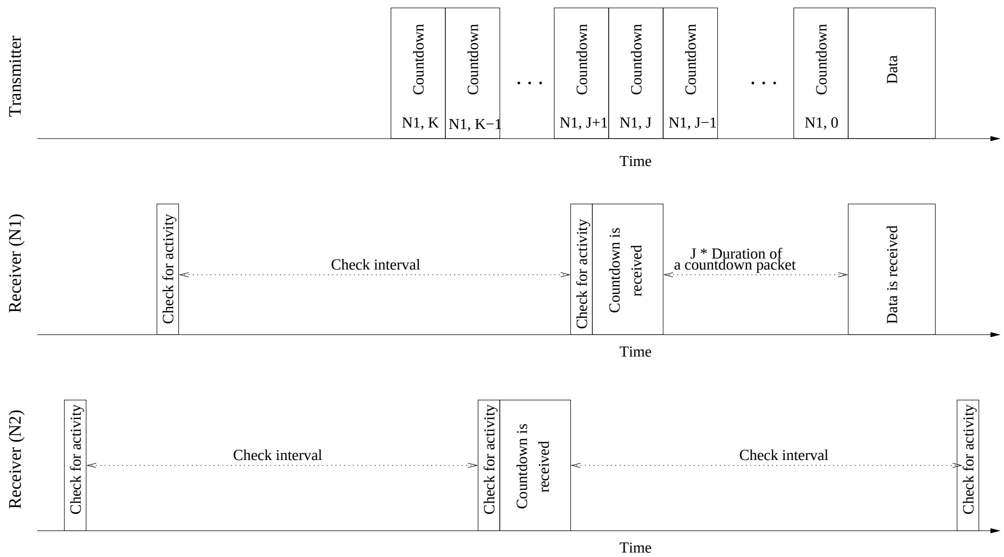

Figure 1. The wake-up preamble of B-MAC+

According to the structure of the wake-up preamble used in B-MAC+, the time spent by nodes into the receive mode can be modeled as:

$$t_{rx} \leq n * r * (2 * (L_{preamble} + L_{cd\_data}) + \\ + p * (L_{preamble} + L_{data})) * t_{rxb} \quad (4)$$

where $L_{preamble}$ is the size of the preamble and L_{cd_data} is the length of the data payload of a countdown packet (which is usually much smaller than standard data packets). Since the destination address is included in the countdown packets, t_{rx} is different for the actual recipient and all other nodes. This is taken into account through p, the probability that the data packet is actually sent to the node. In this case, the worst case happens when the receiver wakes up just after the transmission of a countdown packet has started, thus receiving nodes have to receive the remaining part of the countdown packet and wait for the subsequent one.

3.1 Analytical results

In this Section we compare the expected lifetime of nodes using B-MAC+ against nodes using B-MAC. Leaving all other parameters and modes of operation unaltered, the difference in network lifetime depends on the time spent by nodes in the receive mode. Formulas (3) and (4) represent the time spent in receive mode for B-MAC and B-MAC+, respectively.

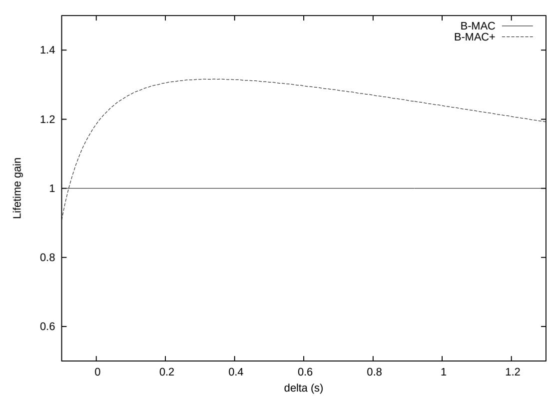

Figure 2. The gain in the lifetime of the network when using B-MAC+

Figure 2 shows the expected gain of lifetime of B-MAC+ with respect to the maximum lifetime obtained with B-MAC. The x axis represents the difference (delta) between check intervals with B-MAC+ and the optimal check interval of B-MAC: $x = 0$ is the optimal check interval of B-MAC, negative values represent shorter check intervals, positive values are longer check intervals. The y axis represents the gain of lifetime: $y = 1$ represents the maximum lifetime of nodes using B-MAC with the optimal check interval. Values larger than $y = 1$ represent a lifetime extension with respect to the maximum lifetime of B-MAC. The data

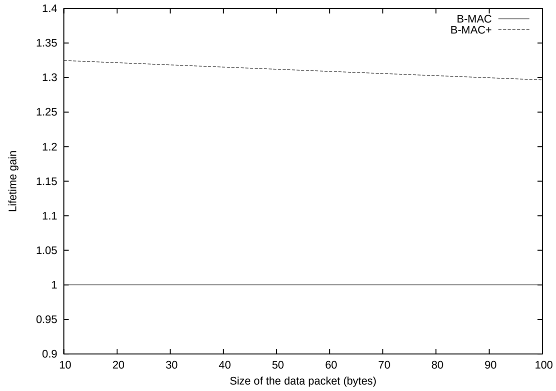

Figure 3. The gain in the lifetime against the size of the data packet

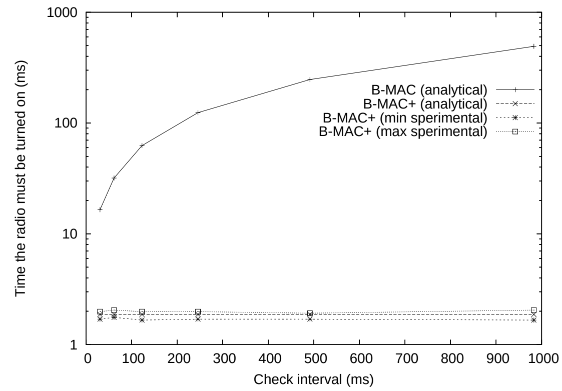

Figure 4. The amount of time to successfully receive a data packet of 36 bytes

packet is supposed to be 36 bytes long, while p (the probability that the packet is intended for the node) is equal to $1/3$. B-MAC+ can provide significant benefits over the standard B-MAC, as shown by the region of the graph where, by increasing the check interval, the lifetime is extended up to 32%.

Figure 3 depicts the gain in the lifetime of the sensor network against the size of the data packet when using B-MAC+ (in this case the value of the check interval is fixed and it is equal to the optimal value). As it can be noticed, B-MAC+ performs better than B-MAC. The gain achieved by B-MAC+ decreases slightly when the size of the data packet increases, since the wake up preamble represents a smaller fraction of the entire transmission.

4 Experimenting with real nodes

We implemented a simplified version of the monitoring application described in Section 2. To validate our solution in a real-world implementation, an experiment has been carried out in order to collect information on the time spent in receive mode when using B-MAC+.

We used Tmote [6] nodes, based on the Telos-B architecture. The radio chip of these nodes is the CC2420, which is 802.15.4 compliant and provides hardware support for packet handling. This means that the radio automatically appends preamble and footer to the data payload, which in turn is generated by software modules implementing the MAC layer. Moreover, we had to take into account that this radio chip automatically inserts an initial spacing before the transmission of a packet. We also took into account that Tmote nodes have a 16-bit architecture, thus we imposed a packet size multiple of 16 bit in order to make it contiguous in memory. We implemented B-MAC+ for TinyOS [12], an open-source operating system designed for wireless sensor networks. At the moment, TinyOS does not include an implementation of B-MAC for the CC2420 radio chip.

The setup of the experiment was the following. One sender node runs the event-driven monitoring application, whose task is to send packets through the wireless channel when an event is detected. The data packet is 36 bytes long. One receiver node is connected to the USB port of a PC, and runs an application that gets packets from the wireless channel. This application is a modified version of TOS_Base, which is bundled with the distribution of TinyOS. We modified TOS_Base in order *i)* to collect statistics about the amount of time needed to receive a packet, and *ii)* to report this collected data to a software program installed on the PC. The statistics are computed directly on the receiving node through a hardware timer. The hardware timer is started/stopped on the basis of the activity on the pins of the radio chip.

Figure 4 shows the statistics about the time the radio must be on to receive a packet, and relates the collected data with the analytical mean values of both B-MAC+ and B-MAC. As expected, the experimental evaluation B-MAC+ is coherent with the analytical results, and B-

MAC+ outperforms B-MAC, as the countdown packets technique reduces the waste of time in receiving long wake-up preambles.

5 Related work and discussion

When executed with low duty-cycles, MAC protocols based on preamble sampling suffer the wastage of energy due to transmission and reception of long wake-up preambles.

A possible approach is the one followed by WiseMAC [3], that tries to reduce the size of the wake-up preamble: the transmitter learns the sampling schedule of its neighbors, then starts transmitting just before the next wake-up time of the receiver. This requires to maintain a table of the sampling times of neighbors. The protocol includes ACK packets that, besides acting as an acknowledgment, also carry timing information about the sampling of the channel.

We presented an alternate technique that allows receiver nodes to turn off the radio during part of the preamble and to return into the receive state just before useful data is transmitted. This was achieved by inserting timing information within the preamble (countdown packets). Evaluation of benefits due to the adoption of this technique showed that the network lifetime can be extended significantly: in the considered scenario, B-MAC+ improved by 32% the performance of B-MAC.

References

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

[2] A. El-Hoiydi. Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks. In *IEEE International Conference on Communications (ICC)*, New York, April 2002.

[3] A. El-Hoiydi and J.-D. Decotignie. WiseMAC: An ultra low power MAC protocol for multi-hop wireless sensor networks. In *First Int. Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2004), Lecture Notes in Computer Science, LNCS 3121*, pages 18–31. Springer-Verlag, July 2004.

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

[5] L. van Hoesel and P. Havinga. A lightweight medium access protocol (LMAC) for wireless sensor networks. In *In Proceedings of the International Workshop on Networked Sensing Systems (INSS)*, Tokyo, Japan, June 2004.

[6] Moteiv Inc. http://www.moteiv.com.

[7] S. Kulkarni and M. Arumugam. Ss-tdma: A self-stabilizing MAC for sensor networks. In C. Griffin S. Phoha, T. La Porta, editor, *Sensor Network Operations*. Wiley-IEEE Press, 2006.

[8] J. Li and G. Lazarou. A bit-map-assisted energy-efficient MAC scheme for wireless sensor networks. In *Third International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pages 55–60, Berkeley, CA, April 2004.

[9] G. Lu, B. Krishnamachari, and C. Raghavendra. An adaptive energy-efficient and low-latency MAC for data gathering in sensor networks. In *Int. Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN)*, Santa Fe, NM, April 2004.

[10] Joseph Polastre, Jason Hill, and David Culler. Versatile low power media access for wireless sensor networks. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 95–107, New York, NY, USA, 2004. ACM Press.

[11] Y. Tay, K. Jamieson, and H. Balakrishnan. Collision-minimizing CSMA and its applications to wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 22(6):1048–1057, August 2004.

[12] TinyOS. http://www.tinyos.net.