Medium Access Protocol for EYES Wireless Sensor Nodes

L.F.W. van Hoesel and P.J.M. Havinga

Email: {l.f.w.vanhoesel, p.j.m.havinga}@utwente.nl

University of Twente

Department of Electrical Engineering, Computer Science and Mathematics

P.O. Box 217, NL-7500 AE Enschede, The Netherlands

Abstract:

This paper proposes EMACS, a medium access control (MAC) protocol designed for wireless sensor networks. In this paper we first introduce the application field of wireless sensor networks, and deduce some basic characteristics and requirements for the communication protocols. These characteristics of a wireless sensor network motivate the use of a different family of MAC protocols than currently employed for wireless (ad hoc) networks (such as IEEE 802.11 [1]), in which throughput, latency, and per node fairness, are more important. EMACS uses various novel mechanisms to reduce energy consumption from the major sources of inefficiency that we have identified. The basic mechanisms applied and combined in one protocol are: scheduled operations, collision avoidance, grouping send and receive, connection oriented, and assistance for routing.

Keywords: Wireless Sensor Network, Medium Access Protocol

1 Introduction

The purpose of a wireless sensor networks is *physical environment monitoring*. Typical applications for such networks are: habitat monitoring, building climate control, burglary detection, early fire detection and many more. Each network node will be equipped with one or more sensors, which readings will be transported via other network nodes to a *data sink*, either after the data sink actively asked for the sensor data or when an event is detected by one of the nodes. The nodes will be spread with -possible- great redundancy and high spatial density in the area of interest.

The EYES project is a three years European research project (IST-2001-34734), on self-organizing and collaborative energy-efficient sensor networks (http://eyes.eu.org). It addresses the convergence of distributed information processing, wireless communications, and mobile computing. The vision of ubiquitous computing requires the development of devices and technologies, which can be *pervasive* without being *intrusive*. Realizing such a network presents very significant challenges, especially at the architectural and protocol/software level. Major steps forward are required in the field of communication protocols, data processing, and application support.

Although sensor nodes will be equipped with a power supply (battery) and embedded processor that makes them autonomous and self-aware, their functionality and capabilities will be very limited. Therefore, collaboration between nodes is essential to deliver smart services in a ubiquitous setting. In the EYES project we investigate new algorithms for networking and distributed collaboration, and evaluate their feasibility through experimentation. These algorithms will be key for building self-organizing and collaborative sensor networks that show emergent behavior and can operate in a challenging environment where nodes move, fail, and energy is a scarce resource.

Energy efficiency is in fact believed to be the true bottleneck in current sensor networks. The challenges to face in developing new technologies for sensor networks are the need for the nodes to be smart, self-configurable, capable of networking together, and the inherent poverty of resources of the nodes themselves. The main thrust of the work within EYES is therefore be directed toward the development of new architectural schemes and communications protocols and algorithms at multiple layers, taking into account those specific features. In particular, schemes, which are able to work efficiently in the presence of limited energy, processing power and memory, are and will be developed.

This paper will address communication protocol issues, and focus mainly on MAC layer functionality. While reducing energy consumption is the primary goal in our design, our protocol is scalable, can dynamically configure itself, and provides support for routing functionality by providing some topology information. The proposed MAC protocol described here satisfies most of our requirements.

In Section 2 we first give some network requirements for some typical applications running on wireless sensor networks. We also identify the consequent inefficiencies related to energy consumption due to the wireless channel. In Section 3 we give a short overview of some related work. In Section 4 we provide an overview of the MAC protocol, describe the frame format and the two power saving modes in the protocol. We evaluate our implementation of EMACS over a small system build out of EYES sensor nodes in Section 4.5.

2 Network and application assumptions

This paper proposes an *energy efficient MAC protocol for wireless sensor networks* (EMACS). Wireless sensor networks (WSNs) are typically deployed in an ad hoc fashion, with individual sensor nodes to be in a dormant state for long periods, and then becoming suddenly active when something is detected. Such applications will thus have long idle periods and can tolerate some latency. For example, one can imagine a surveillance or monitoring application, which will be vigilant for long periods of time, but largely inactive until something is detected. For such applications the lifetime of the sensor nodes is critical. Once a node becomes active, data will be gathered and processed by the node, and needs to be transferred to the destination with far less latency and needs more bandwidth than in the dormant state.

Another typical use of a WSN is to have a kind of streaming data, in which little amounts of data (typically just a few bytes) are transmitted periodically (for example temperature measurements). The large number of nodes will allow taking advantage of short-range, multi-hop communication to conserve energy, especially when *data aggregation* is applied.

Since the nodes will be deployed casually, and may be even mobile, nodes must be able to self-configure.

These characteristics of a wireless sensor network motivate the use of a different family of MAC protocols than currently employed for wireless (ad hoc) networks (such as IEEE 802.11 [1]), in which throughput, latency, and per node fairness, are more important. Moreover, the networks nodes have to operate in a self-organizing ad hoc fashion, since none of the nodes is likely to be capable of delivering the resources to act as central manager.

2.1 Requirements

To design a MAC protocol for wireless sensor networks and typical applications, we have considered the following requirements. The first one is energy efficiency. WSNs typically have a very limited energy budget, because the nodes are likely to be battery powered. Recharging or replacing the batteries is too expensive, and therefore we expect that they be rather discarded then recharged. Prolonging the network lifetime for these nodes is a critical issue.

The other major requirement is the scalability and autonomous configuration to changes in network size, node density and topology. In our work we assume some nodes to be mobile, and others to be fixed to a position. Moreover, sensor nodes are unreliable, and can be expected to die over time. Dynamic configuration is thus an important property to be achieved. This demands the communication protocols to be (virtually) stateless.

Traditional requirements like bandwidth utilization and latency are less important. Due to the limited resources that are available, we will assume in our work that only a single radio channel can be used by the sensor nodes to transport data.

2.2 Types of network nodes

In general two types of network nodes are recognized: nodes that mainly transmit their own sensor readings (sensor nodes) and nodes that mainly relay messages from other nodes (relay nodes). Sensor readings will flow from source node to sink node in the network via relay nodes. The type of a network node may change during the lifetime of the network. This type of hierarchical organization implies four types of communications (see Figure 1) that have to be supported by the MAC protocol:

1. **Sensor node to sensor node communication**: This type of communication is used for local operations, like clustering of the network (hierarchy build up), route discovery and security authentication communication.
2. **Sensor node to relay node communication**: Sensor data is transmitted from sensor node to relay node. This type of communication is often unicast.
3. **Relay node to sensor node communication**: Requests for data and signaling messages (both often multicast) travel from relay node to sensor node.
4. **Relay node to relay node communication**: The relay nodes form the backbone of the wireless sensor network. Communication from relay to relay node will often be unicast.

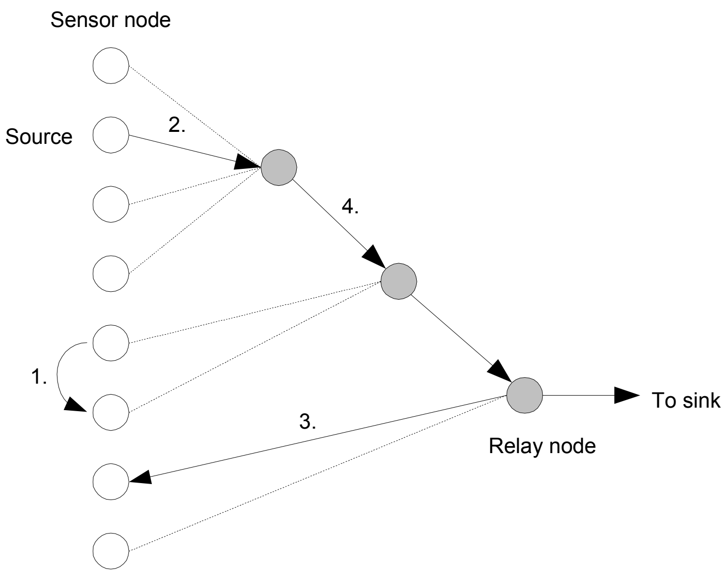

Figure 1: Types of network nodes and their communication

2.3 Energy consumption of a wireless communication

We will now first identify the main causes of inefficient use of energy; as well as some trade-offs we can make to reduce energy consumption. We have identified the following major sources of energy waste.

- The first source of inefficiency is *idle listening*. For applications that have low traffic needs, the transceiver is idling most of the time. This is especially true in many sensor network applications. Many MAC protocols for wireless networks are basically adaptations of MAC protocols used in wired networks, and ignore energy issues. For example, random access MAC protocols such as *carrier sense multiple access with collision avoidance* (CSMA/CA) [2] and IEEE 802.11 [1] typically require the receiver to be powered on continually and monitor the radio channel for traffic. When we know that receiving consumes typically half or more of the total power consumption, it is evident that better schemes need to be used.

- The –in many MAC schemes used- *inactivity threshold*, which is the time before a transceiver will go in the off or standby state after a period of inactivity, causes the receiver to be in a too high energy consuming mode needlessly for a significant time.

- *Collisions* cause the data to become useless and the energy needed to transport that data to be lost. Moreover, latency is increased as well.

- The *overhead* of a protocol also influences the energy requirements due to the amount of 'useless' control data and the required computation for protocol handling. Typical functions in the protocol stack include routing, congestion control, error control, resource reservation, scheduling, etc. The overhead can be caused by long headers (e.g. for addressing, mobility control, etc), by long trailers (e.g. for error detection and correction), and by the number of required control messages (e.g. acknowledgments).
- The *high error rate*, which is typical for wireless links, is another source of energy consumption. Inefficiencies can for example arise when applying error control on data that can tolerate some loss.
- Another main contributor to overhead is due to the *transition times* between the various operating modes of the wireless radio. For example, a typical IEEE 802.11 interface takes 250 ms to make a transition from sleep to idle.
- Finally, overhead induced by the physical layer. This overhead can be significant and is caused by for example *guard space*, *interfacing delay*, *preamble* and *post amble*. When just a few bytes are to be transferred, the overhead is large compared to the transmitted information.

Within EMACS, we have addressed many of these inefficiencies to reduce the energy consumption of a wireless sensor network.

3 Related work

The current MAC designs for wireless ad hoc networks tackle some of these problems addressed above. But when the energy efficiency requirements are coupled with the need for scalability and support for mobility, it is obvious that a lot of work still needs to be done.

Current MAC protocols can be broadly divided into contention based and TDMA protocols. TDMA protocols have the advantage of energy conservation, because the duty cycle of the radio is reduced and there is less contention-introduced overhead and collision. However, scalability is normally not as good as that of a contention-based protocol, since it is not easy to dynamically change its frame length and time slot assignments.

The first step in the reservation and scheduling approaches is to define a communication infrastructure. The aim is to establish some kind of agreement between the nodes in radio range that will enable them to communicate with each other without producing collisions and facilitate node sleeping when they do not have anything to transmit or receive. But the assignment of the channels TDMA slots, frequency bands, spread spectrum codes to the different nodes in a way that avoids collisions is not an easy problem.

One way of dealing with this complexity is to form a hierarchical structure with clusters and delegate the synchronization control to the cluster heads like in the LEACH protocol [3]. Here, issues like cluster membership, rotating cluster heads to prevent early energy depletion and intra-cluster coordination must be effectively addressed. Supporting mobile nodes is also harder to achieve in a hierarchical structure.

At first glance, the contention-based schemes are completely unsuitable for the wireless sensor network scenario. The need for constant monitoring of the channel obviously contradicts the energy efficiency requirement. On the other hand, these schemes do not require any special synchronization and avoid the overhead of exchanging reservation and scheduling information makes them an interesting alternative.

An example of a hybrid scheme is the so called S-MAC protocol [4] that combines scheduling and contention with the aim of improving collision avoidance and scalability. The power saving is based on scheduling sleep/listen cycles between the neighboring nodes. After the initial scheduling, synchronization packets are used to maintain the inter-node synchronization. When a node wants to use the channel, it has to contend for the medium. The scheme used is very similar to IEEE 802.11 [1] with physical and virtual carrier sense and RTS/CTS exchange to combat the hidden node problem. The overhearing control is achieved by putting to sleep all immediate neighbors of the sender and the receiver after receiving an RTS or CTS packet.

4 EMACS

4.1 Overview

The EMACS protocol tries to reduce the inefficiencies from most of the sources of inefficient energy use (as described in Section 2.3). In exchange we accept some increase in latency, but only for those sensor services that can tolerate such latency. The mechanisms to attain these characteristics are:

- **Scheduled operations:** The basic mechanism is to let nodes periodically listen and sleep. Given the fact that in many sensor network applications nodes are idle for a long time, it is not necessary to keep nodes listening all the time. Our protocol reduces the listen time by letting nodes to go into dormant mode, and at scheduled times wake up to announce its presence, and monitor new communication requests from other nodes. Our protocol maintains time synchronization over the whole network, which implies that all operations can be scheduled such that the transceivers can be turned on at the time when needed.

- **Collision avoidance:** The protocol is TDMA based, in which the schedule is completely distributed over the whole network (no central manager in the network). Each node maintains a schedule table that stores the schedules of all its immediate neighbors and adds its own schedule, such that it will not occupy slots in use by those nodes. Collisions are only possible in a small portion of time, when nodes do requests to each other or inform their neighbors of their existents.
- **Grouping of send and receive:** The transition times between the various power states like power on and off, and between sending and receiving (and vice versa) of a wireless transceiver can be significant. Within our protocol we tried to group such operations as much as possible, such that a node can be powered on, and then both listens and sends.
- **Assist routing protocols:** Routing protocols that allow messages to be routed over the ad hoc network, typically require the knowledge of the actual topology, in order to efficiently route the packets over the network and deliver at the destination. Storing routing tables in a static network might be an option, but when nodes are mobile the resulting frequent updates will be too expensive. The MAC layer contains a certain amount of (local) topology knowledge that should be made available to routing protocols.
- **Connection oriented:** Basically we identify two operational modes for the communication module of a sensor node: dormant mode and communication mode. In the dormant mode, the sensor node merely monitors its neighbors, and keeps track of changes in schedule or topology. At the same time it maintains synchronization among neighboring nodes. In the communication mode, the sensor node has established a connection with one of its neighbors, and agreed on a communication channel between them using time slots and periods of

communication. In this way the connection can be tuned to the requirements of the service requesting the communication.

4.2 Frame format

Time is divided into so called *frames* and each frame is divided into *timeslots* (see Figure 2). Each timeslot can be owned by *only one* network node. This network node decides what communication should take place in its timeslot and denies or accepts requests from other nodes.

Time slots will be assigned within one *cell*. A cell comprises all neighbor sensor nodes that are in direct reach of a certain sensor node. This implies that nodes not within reach of each other (and no single node connects them) can share slots. Each node maintains a schedule table, in which it stores the schedules of its cell, and includes the schedules of its neighbors as well.

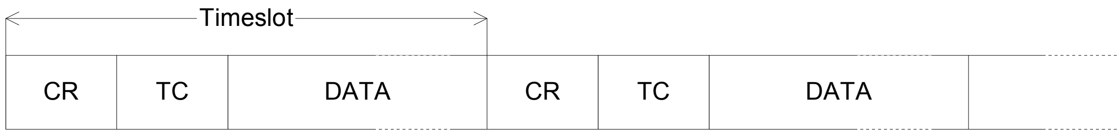

Figure 2: EMACS frame format

Nodes can ask for data or notify the availability of data for the owner of the timeslot in the *communication request* (CR) section. The owner of the slot transmits its schedule for its data section and broadcasts the above discussed table in the *traffic control* (TC) section, which tells to which other TC sections the node is listening. After the TC section, the transmission of the actual data packet follows either uplink or downlink.

The designed MAC protocol does not confirm received data packets. In this way the required error control can be decided on in higher protocol layers. This is quite different from other MAC protocols, which try to reach a certain error rate guarantee per link. In a wireless sensor network a data packet is relayed via multiple hops to its destination and on its way data from other nodes will be added and processed (data aggregation and fusion). The resulting data packet will become more and more important and more resources –like energy- are spent. Hence the error rate guarantees should be adapted to the importance of the data.

As said before, collisions can occur in the communication request section. Although we do not expect a high occurrence of collisions, we incorporate a collision handling mechanism in the EMACS protocol. When the time slot owner detects a collision notifies his neighbor nodes that a collision has occurred. The collided nodes retransmit their request in the data section after a random, but limited backoff time. Carrier sense is applied to prevent the distortion of ongoing requests.

Figure 3 shows the expected lifetime of a network node using EMACS in standard operation. When we assume that the nodes transmit on average 50 bytes/s and receive 50 bytes/s, the expected lifetime of a node will be 570 days. Although this is already quite good, the EMACS protocol supports also two low power modes. These will be discussed in the next section.

Figure 3: Expected lifetime of a sensor node using EMACS

4.3 Sleep modes in the EMACS protocol

Since transmitting and receiving are both very power consuming operations, the network nodes should turn off their transceivers as often as possible. The EMACS protocol supports two sleep modes of the network nodes:

1. **Standby mode**: This sleep mode is used when at a certain time no transmissions are expected. The node releases its slot and starts periodically listening to a TC section of a frame to keep up with the network. When the node has to transmit some data (typically an event driven sensor node), it can just request a transmission (both upload and download) in a CR section of another network, complete it and go back to sleep. Depending on the communication needs, it will start owning a timeslot. In Figure 4 the expected lifetime of a network node in

standby mode is depicted. When we assume that a node transmits on average 50 bytes/s and receives 50 bytes/s, the expected lifetime of the node will be 665 days. But when the node is inactive for long periods of time the lifetime will increase more rapidly than in standard operation mode.

Figure 4: Expected lifetime of a sensor node using EMACS

2. **Dormant mode**: This sleep mode is agreed on higher layers. The sensor node goes to low power mode for an agreed amount of time. Then it wakes, synchronizes (rediscovers the network) and performs the communication. While in sleep mode the synchronization with the network will be lost and all communication with the node will be impossible. This sleep mode is especially useful to exploit the redundancy in the network.

Not every node in the network has to own a timeslot. It is clear that a node does not own a timeslot when it is in one of the sleep modes, since being in a sleep mode is inherent to not transmitting a TC section every frame. But event driven nodes might also not redeem their right to own a timeslot. A drawback of not owning a timeslot is that the node will only be able to receive multicast messages and not messages directly addressed to the node. Transmitting data to nodes that own a timeslot is no problem. Other protocol layers in the network may invoke listening to/transmitting in an a priory agreed (and free or not owned) data section.

Before a node decides, that it wants to give up its timeslot, it should check whether sufficient TC sections are transmitted by neighbors to keep the network connected and to maintain synchronization. The fact that nodes do not necessarily need to own a time slot eases the scalability of the network and reduces the power consumption of the nodes.

4.4 Distributed power control

Scalability of the network is further ensured by the use of transmission power control. We assume that the nodes have the ability to control the transmission power. Each node will use this mechanism to dynamically change the cell size, and try to establish a certain node density in its radio range. For example, when the number of time slots is reaching the frame size, then a node might consider lowering its transmission power, and thus in effect, reducing the number of neighboring nodes.

When reducing the connectivity of a node, it should be ensured not to reduce the overall connectivity of the network. In effect, the node should not exclude nodes that have a low connectivity. On the other hand, a certain node might disconnect nodes that have a high connectivity. The power control mechanism is very simple, since each node can determine individually whether it should change its transmission power. When a certain node reaches a certain threshold (either a low or a high threshold), the algorithm will check the transmission schedules of all its neighbors to identify whether it should change its transmission power. When one of its neighboring nodes is loosely connected (below a certain threshold), the node's transmission power should not be reduced. When all its neighbors are highly connected (above another threshold), it will reduce its transmission power.

4.5 Implementation and experimental results

We have implemented the protocol on the EYES sensor nodes developed within the EYES project. The objective to implement it was not only to demonstrate the effectiveness of our protocol, but also to evaluate the EYES *operating system*. The processor used in the EYES sensor node is MSP-430F149 [5], produced by Texas Instruments. It is a 16-bit processor and it has 60 Kbytes of program memory and 2 Kbytes of data memory. It also has several power saving modes.

A node is also equipped with an auxiliary serial EEPROM memory of 2 Mbits (used for application and data storage). The access time of this memory is relatively long, so it will primary have the function of a secondary storage for data and memory.

The communication function is realized by a RFM TR1001 hybrid radio transceiver [6,7] that is very well suited for this kind of application: it has low power consumption and has small size. The TR1001 supports transmission rates up to 115.2 Kbps. The power consumption during receive is approximately 14.4 mW, during transmit 16.0 mW, and in sleep mode 15.0 μW. The transmitter output power is maximal 0.75 mW.

Preliminary results of our EMACS protocol implementation have shown that several network nodes were able to synchronize in a few seconds and that they were able to communicate with each other.

5 Conclusions and future work

This paper presents EMACS, a MAC protocol for wireless sensor networks. It has good energy efficiency because it uses scheduled operations in which sensor nodes are allowed to sleep for significant amount of time and wake-up when communication is needed. There is a direct trade-off between energy efficiency and latency, since longer duty cycles, will increase latency. There are basically two operational modes: a monitoring mode in which the network merely keeps connected in a low duty cycle, and a communication mode in which nodes are active and transfer data. In our protocol we allow nodes to determine its own schedule based on the requirements imposed by the service. The protocol is not designed to provide high bandwidth utilization, typically less than 1 Kbytes/s. In fact, due to the stringent TDMA structure in which a node is assigned a slot, which cannot be used by other nodes, channel utilization is not optimal. However, this is not a crucial factor in sensor networks, as the expected traffic characteristics have a low data rate. Furthermore, we intend to follow the CEPT/ERC Recommendation 70-03 [8] for issues like power-level and frequency, which allow a duty cycle of just 1%, which equals 108 data bytes/s.

Future work includes more analytical studies on the energy efficiency and performance, and compare then with other protocols designed for sensor networks. Special attention should be made toward mobility and malicious behaving nodes or other devices -that are using the same frequency band- that might corrupt our protocol. The distributed power control as proposed here is initiated by the MAC protocol. Although its implementation is simple, it needs to be evaluated whether we should control the transmission power by higher layers and functional entities (like the application and routing), since they determine its requirements. Therefore, these experiments should be viewed as our initial steps toward an integrated wireless sensor network.

References

1.Standards Committee of the IEEE Computer Society. *IEEE Standard 802.11-1999, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*. IEEE, 1999.

2.T.S. Rappaport, *Wireless communications; Principles and practice.* Prentice Hall, U.S.A., 1996, ISBN 0-13-375536-3.

3.W.R. Heinzelman, A. Chandrakasan and H. Balakrishnan, *Energy-Efficient Communication Protocol for Wireless Microsensor Networks.* Proceedings of the 33rd Hawaii International Conference on System Sciences, 2002.

4.W. Ye, J. Heidemann and D. Estrin, *An Energy-Efficient MAC Protocol for Wireless Sensor Networks*. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), Vol. 3, pp 1567-1576, June 2002.

5.Texas Instruments. *MSP430x1xx Family User's Guide*, http://www.ti.com

6.RFM. *TR1001 868,35 MHz Hybrid Transceiver*

7.RFM. *ASH Transceiver Designer's Guide*, October 2002.

8.CEPT/ERC Recommendation 70-03, http://www.ero.dk