Energy-Efficient Medium Access Control

Koen Langendoen and *Gertjan Halkes*
Delft University of Technology

Faculty of Electrical Engineering, Mathematics and Computer Science
Mekelweg 4, 2628CD Delft, The Netherlands
K.G.Langendoen@ewi.tudelft.nl G.P.Halkes@ewi.tudelft.nl

1 Introduction

Managing wireless communication will be the key to effective deployment of large scale sensor networks that need to operate for years. On the one hand, wireless communication is essential (i) to foster collaboration between neighboring sensor nodes to help overcome the inherent limitations of their cheap, and hence inaccurate sensors observing physical events, and (ii) to report those events back to a sink node connected to the wired world. On the other hand, wireless communication consumes a lot of energy, is error prone, and has limited range, forcing many nodes to participate in relaying information, all of which severely limit the lifetime of the (unattended) sensor network. In typical sensor nodes like the Mica2 mote, communicating one bit of information consumes as much energy as executing several hundred instructions. Therefore, one should "think" twice before actually transmitting a message. Nevertheless, whenever a message should be sent, the protocol stack must operate as efficiently as possible. In this report we will study the medium access layer, which is part of the Data Link layer (layer 2 of the OSI model) and sits directly on top of the Physical layer (layer 1) (see Figure .1). Since the medium access layer controls the radio, it has a large impact on the overall energy consumption, and hence, the lifetime of a node.

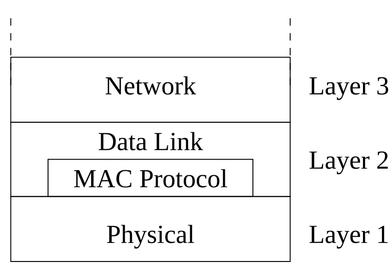


Figure .1: Network protocol stack.

A Medium Access Control (MAC) protocol decides when competing nodes may access the shared medium, i.e. the radio channel, and tries to ensure that no two nodes are interfering with each other's transmissions. In the unfortunate event of a collision, a MAC protocol may deal with it through some contention resolution algorithm, for example, by resending the message later at a randomly selected time. Alternatively, the MAC protocol may simply discard the message and leave the retransmission – if any – up to the higher layers in the protocol stack. MAC protocols for wireless networks have been studied since the 1970s, but the successful introduction of wireless LANs in the late 1990s has accelerated the pace of developments; the recent survey by Jurdak et al. reports an exponential growth of new MAC protocols [19]. We will now provide a brief historic perspective on the evolution of medium access control, and describe the two major approaches – contention-based and schedule-based – regularly used in wireless communication systems. Readers familiar with medium access in wireless networks may proceed to Section 2 immediately.

1.1 Contention-based medium access

In the classic (pure) ALOHA protocol [1], developed for packet radio networks in the 1970s, a node simply transmits a packet when it is generated. If no other node is sending at the same time, the data transmission succeeds and the receiver responds with an acknowledgement. In the case of a collision, no acknowledgement will be generated, and the sender retries after a random period. The price to be paid for ALOHA's simplicity is its poor use of the channel capacity; the maximum throughput of the ALOHA protocol is only 18% [1]. However, a minor modification to ALOHA can increase the channel utilization considerably. In slotted ALOHA time is divided into slots, and nodes may only transmit at the beginning of a slot. This organization halves the probability of a collision and raises the channel utilization to around 35% [32].

CSMA

Instead of curing the effects (retransmissions) after the fact, it is often much better to take out the root of the problem (collisions). The Carrier Sense Multiple Access (CSMA) protocol [21], originally introduced by Kleinrock and Tobagi in 1975, tries to do just that. Before transmitting a packet, a node first listens to the channel for a small period of time. If it does not sense any traffic, it assumes that the channel is clear and starts transmitting the packet. Since it takes some time to switch the radio from receive mode to transmit mode, the CSMA method is not bullet proof and collisions can still occur. In practice however, CSMA-style MAC protocols can achieve a maximal channel utilization in the order of 50% - 80% depending on the exact access policy [21].

CSMA/CA

When all nodes can sense each other's transmissions, CSMA performs just fine. It took until 1990 before a significant new development in medium access control was recorded. The Medium Access with Collision Avoidance (MACA) protocol [20] addresses the so-called *hidden terminal* problem that occurs in ad-hoc (sensor) networks where the radio range is not large enough to allow communication between arbitrary nodes and two (or more) nodes may share a common neighbor while being out of each other's reach. Consider the situation in Figure .2 where nodes A and C both want to transmit a packet to their common neighbor B. Both nodes sense an idle channel and start to transmit their packets, resulting in a collision at B. Note that since node A is hidden from C, any packet sent by C will disrupt an on-going transmission from A to B, so this type of collision is quite common in ad-hoc networks.

MACA introduces a three-way handshake to make hidden nodes aware of upcoming transmissions, so collisions at common neighbors can be avoided. The sender (node A in Figure .2) initiates the handshake by transmitting a short Request-To-Send (RTS) control packet announcing its intended data transmission. The receiver (B) responds with a Clear-To-Send (CTS) packet, which informs all neighbors of the receiver (including hidden nodes like C) of the upcoming transfer. The final DATA

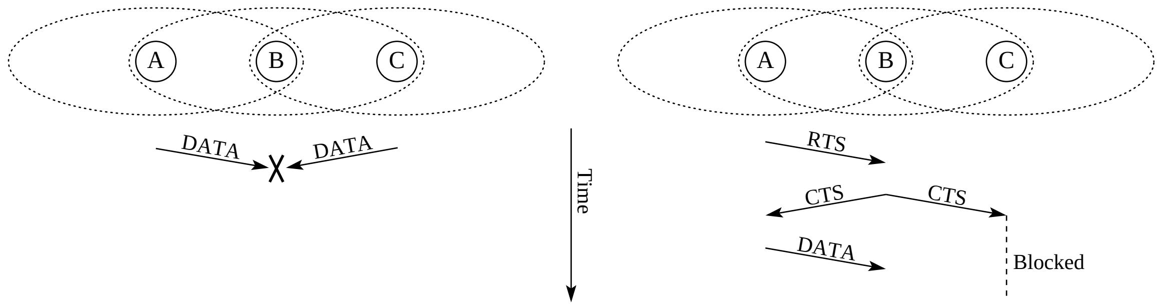


Figure .2: The hidden terminal problem (left) resolved through RTS/CTS signalling (right).

transfer (from A to B) is now guaranteed to be collision free. When two RTS packets collide, which is technically still possible, the intended receiver does not respond with a CTS and both senders back off for some random time. To account for the unreliability of the radio channel, MACA Wireless (MACAW [3]) adds a fourth packet to the control sequence to guarantee delivery. When the data is received correctly, an explicit ACKnowledgement is send back to the sender. If the sender does not receive the ACK in due time, it initiates a retransmission sequence to account for the corrupted or lost data.

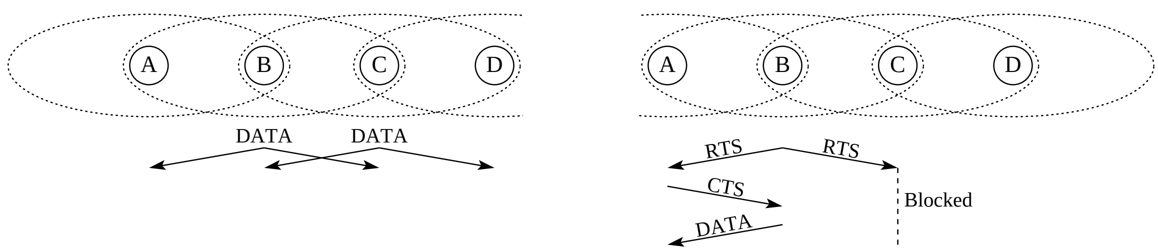


Figure .3: The exposed terminal problem: concurrent transfers (left) are synchronized (right).

The collision avoidance protocol in MACA (and derivatives) is widely used and is generally known as CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). It has proved to be very effective in eliminating collisions. In fact, CSMA/CA is too good at it and also silences nodes whose transmissions would not interfere with the data transfer between the sender-receiver pair. The so-called *exposed terminal* problem is illustrated in Figure .3. In principle, the data transmissions B $\rightarrow$A and C $\rightarrow$D can take place concurrently since the signals from B cannot disturb the reception at D, and similarly C's signals cannot collide at A. However, since B must be able to receive the CTS by A, all nodes who can hear B's RTS packet must remain silent even if they are outside the reach of the receiver (A). Node C is thus is exposed to B's transmission (and vice versa). Since exposed nodes are prohibited from sending, aggregate throughput may be reduced.

IEEE 802.11

In 1999 the IEEE Computer Society published the 802.11 wireless LAN standard [16], specifying the PHYsical and MAC layers. IEEE 802.11 compliant equipment, usually PC cards operating in the 2.4 GHz or 5 GHz band, can operate in infrastructure mode as well as in ad-hoc mode. In both cases 802.11 implements carrier sense and collision avoidance to reduce collisions (see Section 4.1 for details). To preserve the energy of mobile nodes, the 802.11 standard includes a power-saving mechanism that allows nodes to go into sleep mode (i.e. disable their radios) for long periods of time. This mode of operation requires the presence of an access point that records the status of each node and buffers any data addressed to a sleeping node. The access point regularly broadcasts beacon packets indicating for which nodes it has buffered packets. These nodes may then send a poll request to the access point to retrieve the buffered data (or switch back from sleep to active mode). Krashinksy and Balakrishnan report up to 90% energy savings for web browsing applications, but at the expense of considerable delays [22]. Currently, power saving in 802.11's ad-hoc mode is only supported when all nodes are within each other's reach, so a simple, distributed scheme can be used to coordinate actions; the standard does not include a provision for power saving in multi-hop networks.

1.2 Schedule-based medium access

The MAC protocols discussed so far are based on autonomous nodes contending for the channel. A completely different approach is to have a central authority (access point) regulate the access to the medium by broadcasting a schedule that specifies when, and for how long, each controlled node may transmit over the shared channel. The lack of contention overhead guarantees that this approach does not collapse under high loads. Furthermore, with the proper scheduling policy, nodes get deterministic access to the medium and can provide delay-bounded services as voice and multi-media streaming. Schedule-based medium access is therefore the preferred choice for cellular phone systems (e.g., GSM) and wireless networks supporting a mix of data and real-time traffic (e.g., Bluetooth).

TDMA

Time-Division Multiple Access (TDMA) is an important schedule-based approach that controls the access to a single channel (techniques for handling multiple channels will be discussed in Section 3.2). In TDMA systems the channel is divided into slots, which are grouped into frames (see Figure .4). The access point decides (schedules) which slot is to be used by which node. This decision can be made on a per frame basis, or it can span several frames in which case the schedule is repeated.

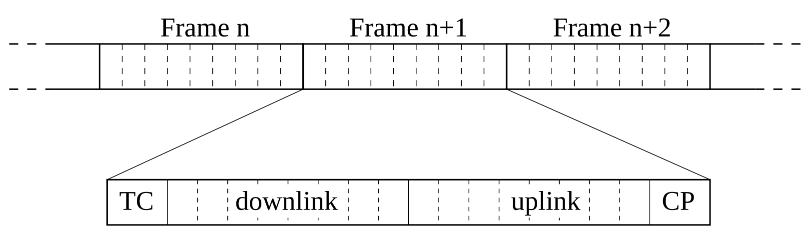


Figure .4: TDMA frame structure: Traffic Control - downlink - uplink - Contention Period.

In typical Wireless LAN setups, most traffic is exchanged between the access point and the individual nodes. In particular, communication between nodes rarely occurs. By limiting communication to up- and downlink only, the scheduling problem is greatly simplified. Figure .4 shows a typical frame layout. The first slot in the frame is used by the access point to broadcast traffic control information to all nodes in its cell. This information includes a schedule that specifies when each node must be ready to expect to receive a packet (in the down-link section), and when it may send a packet (in the uplink section). The frame ends with a contention period in which new nodes can register themselves with the access point, so they can be included in future schedules.

TDMA systems provide a natural way to conserve energy. A node can turn off its radio during all slots in a frame in which it is not engaged in communication to/from the access point. It does require, however, accurate time synchronization between the access point and the individual nodes to ensure that a node can wake up exactly at the start of “its” slots. In a sensor network where activity is usually low, a node is then – on average – only awake for one slot each frame to receive the traffic control information. Enlarging the frame size reduces the energy consumption, but also increases the latency since a node has to wait longer before its slot turns up. This fundamental energy/latency trade-off is further explored in Section 3.2.

2 Requirements for sensor networks

The vast majority of MAC protocols described in the literature so far were designed, and optimized, for scenarios involving satellite links (early work) and wireless LANs (recent developments). The deployment scenarios for wireless sensor networks differ considerably, leading to a different set of requirements. In particular, the unattended operation of sensor networks stresses the importance of energy efficiency and reduces the significance of performance considerations such as low latency, high throughput, and fairness. Nevertheless, there are lessons to be learned from MAC protocols developed for wireless communication systems, especially those targeting ad-hoc networks of mobile nodes. The interested reader is referred to a number of recent surveys in this area [9, 19, 25].

The task of the MAC layer in the context of sensor networks is to use the radio, with its limited resources, as efficiently as possible to send and receive data generated by the upper layers in the protocol stack. It should take into account that data is often routed across multiple hops, and be able to handle large-scale networks with hundreds, or even thousands of (mobile) nodes. To understand the design trade-offs involved we will discuss the hardware characteristics of prototype sensor nodes in use today, as well as common traffic patterns that have emerged in preliminary experiences with applications.

2.1 Hardware characteristics

The current generation of sensor nodes, some of which are commercially available, are made up of off-the-shelf components mounted on a small printed circuit board. In the future we expect single chip solutions with some of the protocol layers implemented in hardware. At the moment however, the MAC protocols are running on the main processor, which drives a separate chip that takes care of converting (modulating) bits to/from radio waves. The interface between the processor and the radio chip is at the level of exchanging individual bits or bytes. The advantage of this low-level interface is that the MAC designer has absolute control, which contrasts sharply with 802.11 WLAN equipment where the MAC is usually included as part of the chipset on the PC card.

Popular processors include the 8-bit Atmel ATmega128L CPU used on the Mica motes, the 16-bit Texas Instruments MSP430 used on the Eyes nodes, and the PIC-16 from Microchip. The exact specifications vary, but the processors typically run at a frequency in the 1 - 10 MHz range, and are equipped with 2 - 4 KB of RAM. The processing capabilities provide ample headroom to drive the radio, but the limited amount of storage space for local data puts a strong constraint on the memory footprint of the MAC protocol. Since the focus of sensor node development is on energy consumption and form factor, we do anticipate that future generations will still be quite limited in their processing and memory resources.

Table .1 provides details on the characteristics of two low-power radios employed in various state-of-the-art sensor nodes. For reference, the specifications of a typical 802.11 PC card are included. Several important observations can be made. First, the energy consumed when sending or receiving data is two to three orders of magnitude more than keeping the radio in a low-power standby state. Thus, the key to effective energy management will be in switching the radio off and on. Second, the time needed to switch from standby to active mode is considerable (518 μs - 2.0 ms), and the time needed to switch the radio between transmit and receive mode is also non negligible. Therefore, the

	RFM TR 1001 [31]	CC1000 [4]	Lucent WaveLAN PC "Silver" card [8]
operating frequency	868 MHz	868 MHz [a]	2.4 GHz
modulation scheme	ASK	FSK	DSSS
bit rate	115.2 kbps	76.8 kbps	11 Mbps
energy consumption			
transmit	12 mA (1.5 dBm)	8.6 mA (-20 dBm) 25.4 mA (5 dBm)	284 mA
receive	3.8 mA	11.8 mA	190 mA
standby	0.7 μA	30 μA	10 mA
switch times			
standby-to-transmit	16 μs	2.0 ms	
receive-to-transmit	12 μs	270 μs	
standby-to-receive	518 μs [b]	2.0 ms	
transmit-to-receive	12 μs	250 μs	
transmit-to-standby	10 μs		
receive-to-standby	10 μs		

[a] The CC1000 radio supports any frequency in the 300 - 1000 MHz range; the quoted numbers are for 868 MHz.
[b] Time needed to fully initialize receive circuitry; a simple carrier sense can be performed in 30 μs.

Table .1: Characteristics of typical radios in state-of-the-art sensor nodes.

number of mode switches should be kept to a minimum. Finally, the WaveLAN card (including the MAC) outperforms the other radios in terms of energy per bit (77 vs. 312 μJ/bit); future nodes should include radios with higher frequencies and more complex modulation schemes.

2.2 Communication patterns

In the rapidly emerging field of wireless sensor networks there is little experience with realistic, long-running applications, which is unfortunate since a good characterization of the workload (in terms of network traffic) is mandatory for designing a robust and efficient MAC protocol, or any other part of the network stack for that matter. It is however clear that the nature of the traffic for sensor networks has a few remarkable characteristics that sets it apart from your average WLAN traffic. From the various proposed deployment scenarios, usually in the area of remote monitoring, and the limited data from preliminary studies like the Great Duck Island [35] and vehicle tracking system [11] it becomes clear that data rates are very low: typically in the order of 1 - 200 bytes per second, with message payload sizes around 20 - 25 bytes. Furthermore, two distinct communication patterns (named *convergecast* and *local gossip* in [23]) appear to be responsible for generating the majority of network traffic:

Convergecast In many monitoring applications, information needs to be periodically transmitted to a sink node so it can be processed at a central location or simply stored in a database for future use. Since these individual reports are often quite small and need to travel across the whole network, the overhead is quite large. Aggregating messages along the spanning tree to the sink node therefore pays off. At the very least two (or more) packets can be coalesced to share a common header. At the very best two (or more) messages can be combined into one, for example, when reporting the maximum room temperature.

Local gossip When a sensor node observes a physical event, so do its neighbors since the node density in a sensor network is expected to be high. This allows a node to check with the nodes in its vicinity if they observed the same event or not, and in the latter case to derive that its sensor is probably malfunctioning. If its neighbors do observe the same event (e.g., a moving target) they can collaborate to obtain a better estimate of the event (location and speed) and report that back to the sink. Besides improving the quality of the reported information, the collaboration also avoids n duplicate messages travelling all the way back to the sink. Depending on the situation, neighbors may be addressed individually (unicast) or collectively (broadcast). In any case, by sharing (gossiping) their sensor readings (rumors) nodes can reduce the likelihood of false positives, and efficiently report significant events.

The important implication of these two communication patterns is that traffic is not distributed evenly over the network. The amount of data varies both in space and in time. Nodes in the vicinity of the sink relay much more traffic than nodes at the edges of the network due to the convergecast pattern. The fluctuation in time is caused by the physical events triggering outbursts of local gossip. In the extreme case of a forest fire detection system, nodes may be dormant for years before finally reporting an event. MAC protocols should be able to handle these kind of fluctuations.

2.3 Miscellaneous services

Often the MAC layer is expected to provide some network-related services not directly associated with data transfer. Localization and time-synchronization algorithms often need precise information about the moment of the physical transmission of a packet to factor out any time spent by the MAC layer in contention resolution. The routing layer needs to be informed of any local changes in network topology, for example it needs to know when mobile nodes move in and out of radio range. Since the MAC layer sits directly on top of the radio it can perform these services at no extra cost. Neighborhood discovery, for example, must be carried out to ensure the proper operation of TDMA-based MAC protocols. We will not consider these miscellaneous requirements in the remainder of this report, but concentrate on the MAC protocols' ability to transfer data as efficiently as possible.

3 Energy efficiency

The biggest challenge for designers of sensor networks is to develop systems that will run unattended for years. This calls for robust hardware and software, but most of all for careful energy management, since that is and will continue to be a limited resource. The current generation of sensor nodes is battery powered, so lifetime is a major constraint; future generations powered by ambient energy sources (sunlight, vibrations, etc.) will provide very low currents, so energy consumption is heavily constrained.

It is important to realize that the failure of individual nodes may not harm the overall functioning of a sensor network, since neighboring nodes can take over provided that the node density is high enough (which can be guaranteed at roll out). Therefore the key parameter to optimize for is network lifetime, that is, the time until the network gets partitioned. The MAC layer operates on a local scale (all nodes within reach) and lacks the global information to optimize for network lifetime. This is therefore best accomplished at the upper layers of the protocol stack, in particular the routing and transport (data aggregation) layers, which do have a global overview. This works most effectively when the MAC layer ensures that the energy it spends is directly related to the amount of traffic that it handles. Thus, the MAC layer should optimize for energy efficiency.

In contrast to typical WLAN protocols, MAC protocols designed for sensor networks usually trade off performance (latency, throughput, fairness) for cost (energy efficiency, reduced algorithmic

source	performance (latency, throughput, fairness)	cost (energy-efficiency)
collisions	C	C
protocol overhead	C,S	C,S
idle listening		C
overhearing		C
traffic fluctuations	C,S	C,S
scalability/mobility	S	S

Table .2: Impact of overhead on contention-based protocols (C) and schedule-based protocols (S).

complexity). It is, however, not clear cut what the best trade-off is, and various designs differ significantly as will become apparent in Section 3.2 where we will review the basic design choices made by 20 WSN-specific MAC protocols. Before that, we will consider the major sources of overhead that render WLAN-style (contention-based) MAC protocols ineffective in the context of sensor networks.

3.1 Sources of overhead

When running a contention-based MAC protocol on an ad-hoc network with little traffic, much energy is wasted due to the following sources of overhead:

Idle listening Since a node does not know when it will be the receiver of a message from one of its neighbors, it must keep its radio in receive mode at all times. This is the major source of overhead, since typical radios consume two orders of magnitude more energy in receive mode (even when no data is arriving) than in standby mode (cf. Table .1).

Collisions If two nodes transmit at the same time and interfere with each others' transmission, packets are corrupted. Hence, the energy used during transmission and reception is wasted. The RTS/CTS handshake effectively resolves the collisions for unicast messages, but at the expense of protocol overhead.

Overhearing Since the radio channel is a shared medium, a node may receive packets that are not destined for it; it would have been more efficient to have turned off its radio.

Protocol overhead The MAC headers and control packets used for signalling (ACK/RTS/CTS) do not contain application data and are therefore considered overhead; these overheads can be significant since many applications only send a few bytes of data per message.

Traffic fluctuations A sudden peak in activity raises the probability of a collision, hence, much time and energy are spent on waiting in the random backoff procedure. When the load approaches the channel capacity, the performance can collapse with little or no traffic being delivered while the radio, sensing for a clear channel, is consuming a lot of energy.

Switching to a schedule-based protocol (i.e., TDMA) has the great advantage of avoiding all energy waste due to collisions, idle listening, and overhearing since TDMA is inherently collision-free and the schedule notifies each node when it should be active and, more importantly, when not. The price to be paid is in fixed costs (i.e., broadcasting traffic schedules) and reduced flexibility to handle traffic fluctuations and mobile nodes. The usual solution is to resort to some form of overprovisioning and choosing a frame size that is large enough to handle peak loads. Dynamically adapting the frame size

Protocol	Published	Channels	Organization	Notification
SMACS [34]	2000	FDMA	frames	schedule
PACT [28]	2001	single	frames	schedule
PicoRadio [10]	2001	CDMA+tone	random	wakeup
STEM [33]	2002	data+ctrl	random	wakeup
Preamble sampling [6]	2002	single	random	listening
Arisha [2]	2002	single	frames	schedule
S-MAC [36]	2002	single	slots	listening
PCM [18]	2002	single	random	listening
Low Power Listening [13]	2002	single	random	listening
Sift [17]	2003	single	random	listening
EMACs [15]	2003	single	frames	schedule (per node)
T-MAC [5]	2003	single	slots	listening
TRAMA [30]	2003	single	frames	schedule (per node)
WiseMAC [7]	2003	single	random	listening
B-MAC [29]	2003	single	random	listening
BMA [24]	2004	single	frames	schedule
Miller [27]	2004	data+tone	random	wakeup+listening
DMAC [26]	2004	single	slots (per level)	listening
SS-TDMA [23]	2004	single	frames	schedule
LMAC [14]	2004	single	frames	listening

Table .3: Protocol classification.

is another approach, but this largely increases the complexity of the protocol and, hence, is considered to be an unattractive option for resource-limited sensor nodes. Table .2 compares the impact of the various sources of overhead on the performance and cost (energy efficiency) of contention-based and schedule-based MAC protocols.

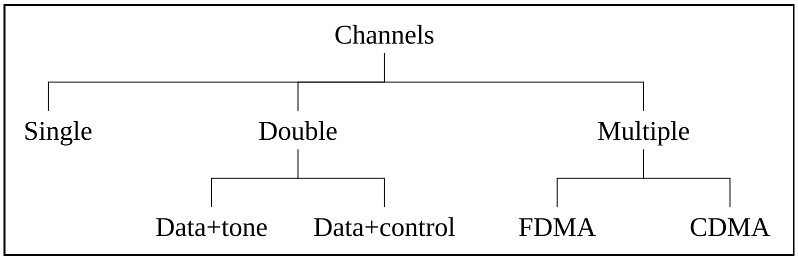
3.2 Trade-offs

Different MAC protocol make different choices regarding the performance-energy trade off, and also between sources of overhead (e.g., signalling vs. collisions). A survey of 20 medium access protocols specially designed for sensor networks, and hence optimized for energy efficiency, revealed that they can be classified according to three important design decisions:

1) the number (and nature) of the physical channels used,
2) the degree of organization (or independence) between nodes, and
3) the way in which a node is notified of an incoming message.

Table .3 provides a comprehensive protocol classification based on these three issues. Given that the protocols are listed chronologically based on their publication date, we observe that there is no clear trend indicating that medium access for wireless sensor networks is converging towards a unique, best solution. On the contrary, new combinations are still being "invented" showing that additional information (from simulations and practical experience) is needed to decide on the best approach. Section 7 provides a simulation-based head-to-head comparison of four protocols representing very distinctive choices in the design space. We will not discuss all individual MAC protocols listed in Table .3 in detail, but rather review three fundamental design choices that MAC designers will encounter while crafting a protocol best matching their envisioned deployment scenario.

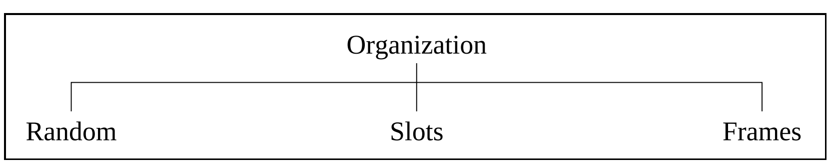
Use multiple channels, or not?



The first design choice that we discuss is whether or not the radio should be capable of dividing the available bandwidth into multiple channels. Two common techniques for doing so are Frequency-Division Multiple Access (FDMA) and Code-Division Multiple Access (CDMA). FDMA partitions the total bandwidth of the channel into a number of small frequency bands, called subcarriers, on which multiple nodes can transmit simultaneously without collision. CDMA on the other hand, uses a single carrier in combination with a set of orthogonal codes. Data packets are XOR-ed with a specific code by the sender before transmission, and then XOR-ed again by the receiver with the same code to retrieve the original data. Receivers using another code perceive the transmission as (pseudo) random noise. This allows the simultaneous and collision-free transmission of multiple messages.

The absence of collision in a multiple-channel system is attractive, hence its popularity in early proposals like SMACS (FDMA) and PicoRadio (CDMA). It requires however a rather complicated radio consuming considerable amounts of energy, so most MAC protocols are designed for a simple radio providing just a single channel. An interesting alternative is to use a second, extremely low-power radio that can be used for signalling an intended receiver to wake-up and turn on its primary radio to receive a data packet. In the most simple, most energy-efficient case, the second radio is only capable of emitting a fixed "tone" waking up all neighboring nodes (including the intended receiver). Miller and Vaidya discuss several policies to minimize the number of false wakeups by overhearing nodes [27]. STEM uses a full-blown second radio to control exactly which node responds on the primary channel.

Get organized, or not?

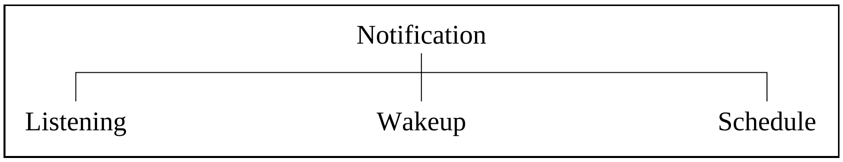


The second design choice that we discuss is if, and how much, the nodes in the network should be organized to act together at the MAC layer. The CSMA and TDMA protocols discussed before represent the two extremes in the degree of organization: from completely random to frame-based access. The advantages of contention-based protocols (random access) are the low implementation complexity, the ad-hoc nature, and the flexibility to accommodate mobile nodes and traffic fluctuations. The major advantage of frame-based TDMA protocols is the inherent energy-efficiency due to the lack of collisions, overhearing, and idle-listening overheads.

Since the advantages of random access are the drawbacks of frame-based access, and vice versa, some MAC protocols have chosen to strike a middle ground between these two extremes and organize the sensor nodes in a slotted system (much like slotted-ALOHA). The S-MAC protocol was the first to propose that nodes agree on a common slot structure, allowing them to implement an efficient duty cycle regime; nodes are awake in the first part of each slot and go to sleep in the second part, which significantly reduces the energy waste due to idle listening.

The protocol classification in Table .3 shows that the research community is divided into what degree of organization to apply: we find 9 contention-based, 3 slotted, and 8 TDMA-based protocols. Since we view the organizational design decision as the most critical, we will detail the main protocols from each class in sections 4 to 6.

Get notified, or not?



The third and final design issue is about how the intended receiver of a message transfer will get notified. In schedule-based protocols, the actual data transfers are scheduled ahead of time, so receiving nodes know exactly when to turn on the radio. Such knowledge is not available in contention-based protocols, so receiving nodes must be prepared to handle an incoming transfer at any moment. Without further assistance from the sender, the receiver has no other option than to listen continuously. To eliminate the resulting idle-listening overhead completely, senders may actively send a wakeup signal (tone) over a second, very low-power radio. Although the wakeup model matches well with the low packet rates of sensor network applications, all contention-based protocols except PicoRadio, STEM and Miller's proposal are designed for nodes with a single radio. The general approach to reduce the inherent idle-listening in these nodes is to enforce some kind of duty cycle by periodically switching the radio on for a short time. This can be arranged individually per node (Low Power Listening and Preamble sampling, Section 4.2) or collectively per slot (S-MAC, Section 5.1). An alternative is to circumvent the idle-listening problem as the Sift protocol does by restricting the network to a cellular topology where access points collect data from nearby sensor nodes.

We like to point out that the choice for a particular notification policy is largely dependent on the available hardware channels and the organizational model discussed before. Schedule-based notification matches with TDMA frames; wakeup is only possible on dual channel nodes. The LMAC protocol (Section 6.1), however, is the exception to the rule and combines TDMA frames with listening, striking a different balance between flexibility and energy-efficiency.

4 Contention-based protocols

We now proceed with describing in detail some of the medium access protocols developed for sensor networks according to their particular choice of organizational model (see Table .3). In this section, we review contention-based protocols in which nodes can start a transmission at any random moment and must contend for the channel. The main challenge with contention-based protocols is to reduce the energy consumption caused by collisions, overhearing, and idle-listening. CSMA/CA protocols effectively deal with collisions and can be easily adapted to avoid a lot of overhearing overhead (i.e., switch off the radio for the duration of another transmission's sequence). We also discuss the familiar IEEE 802.11 protocol. even though it was not developed specifically for sensor networks. It does however, form the basis of the energy-efficient derivatives discussed in this section (Low-Power Listening and WiseMAC), as well as the slotted protocols (S-MAC and T-MAC) discussed in the next section.

4.1 IEEE 802.11

The medium access control in the IEEE 802.11 standard [16] is based on carrier sensing (CSMA) and collision detection (through acknowledgements). A node wanting to transmit a packet must first test the radio channel to check if it is free for a specified time called the Distributed Inter Frame Space (DIFS). If so, a DATA packet[1] is transmitted, and the receiver waits a Short Inter Frame Space (SIFS) before acknowledging the reception of the data by sending an ACK packet. Since the SIFS interval is set shorter than the DIFS interval, the receiver takes precedence over any other node attempting to send a packet. If the sender does not receive the acknowledgement, it assumes that the data was lost due to a collision at the receiver and enters a *binary exponential backoff* procedure. At each retransmission attempt, the length of the Collision Window (CW) is doubled. Since contending nodes randomly select a time from their CW, the probability of a subsequent collision is reduced by half. To bound access latency somewhat, the CW is not doubled once a certain maximum (CW_{max}) has been reached.

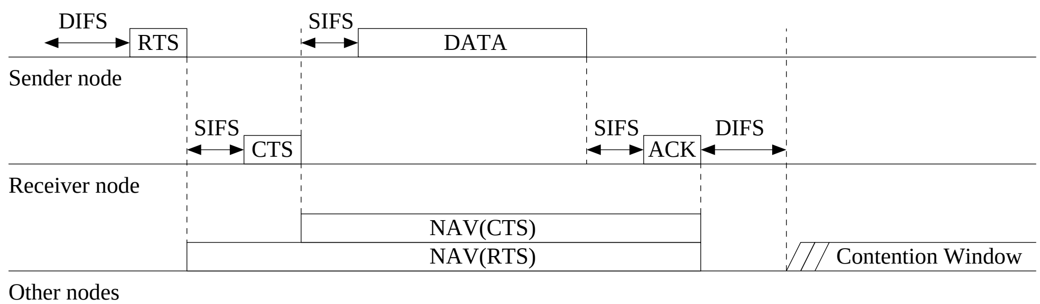


Figure .5: IEEE 802.11 access control.

To account for the hidden terminal problem in ad-hoc networks, the 802.11 standard defines a virtual carrier sense mechanism based on the collision avoidance handshake of the MACA protocol. The RTS/CTS control packets include a time field in their header, that specifies the duration of the upcoming DATA/ACK sequence. This allows neighboring nodes overhearing the control packets to set their Network Allocation Vector (NAV) and defer transmission until it expires (see Figure .5). To save energy, the radio can be switched off for the duration of the NAV. Thus CSMA/CA effectively eliminates collisions and overhearing overhead for unicast packets. Broadcast and multicast packets are always transmitted without an RTS/CTS reservation sequence (and without an ACK), so they are susceptible to collisions.

4.2 Low Power Listening and Preamble Sampling

The major disadvantage of CSMA/CA is the energy wasted by idle listening. Both Hill and Culler [13], and El-Hoiydi [6] independently developed a low-level carrier sense technique that effectively duty cycles the radio, i.e. turns it off repeatedly, without losing any incoming data. This technique operates at the physical layer and concerns the layout of the PHY header prepended to each radio packet. This header starts off with a preamble that is used to notify receivers of the upcoming transfer and allows them to adjust (train) their circuitry to the current channel conditions; next follows the startbyte signalling the true beginning of the data transfer. The basic idea behind the efficient carrier-sense technique is to shift the cost from the receiver (the frequent case) to the transmitter (the rarer case) by increasing the length of the preamble. This allows the receiver to periodically turn on the radio to sample for incoming data, and detect if a preamble is present or not.

[1] The 802.11 standard defines the transmission protocol in terms of frames, but we use the term packet instead to avoid confusion with the framing structure of TDMA protocols.

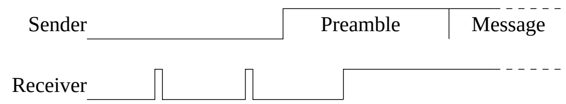


Figure .6: Low-power listening: a long preamble allows periodic sampling at the receiver.

If it detects a preamble, it will continue listening until the start-symbol arrives and the message can be properly received (see Figure .6). If no preamble is detected, the radio is turned-off again until the next sample.

This efficient carrier-sense method can be applied to any contention-based MAC protocol. El-Hoiydi combined it with ALOHA and named it Preamble Sampling [6]. Hill and Culler combined it with CSMA and named it Low Power Listening [13]. Neither implementation includes collision avoidance to save on protocol overhead. The energy savings depend on the duty cycle, which in turn depends on the switching times of the radio. Low Power Listening, for example, was implemented as part of TinyOS running on Mica motes equipped with an RFM 1000 radio capable of performing a carrier sense in just 30 μs (cf. Table .1). The carrier is sensed every 300 μs, yielding a duty-cycle of 10 %, effectively 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 to 647 μs), and minor reduction in throughput. In the recently proposed B-MAC implementation (part of TinyOS 1.1.3) the preamble length is provided as a parameter to the upper layers, so they can select the optimal trade-off between energy savings and performance [29].

4.3 WiseMAC

El-Hoiydi has refined his Preamble Sampling one step further, by realizing that long preambles are not necessary when the sender knows the sampling schedule of the intended receiver. The sender can then simply wait until the moment the receiver is about to sample the channel, and send a packet with an ordinary preamble. This not only saves energy at the sender, who waits instead of emitting an extended preamble, but also at the receiver, since the time until the start symbol occurs is reduced in length considerably. In WiseMAC [7] nodes maintain the schedule offsets of their neighbors through piggy backed information on the ACKnowledgements of the underlying CSMA protocol. Whenever a node needs to send a message to a specific neighbor n, it uses n's offset to determine when to start transmitting the preamble; to account for any clock drift, the preamble is extended with a time proportional to the length of the interval since the last message exchange. The overall effect of these measures is that WiseMAC adapts automatically to traffic fluctuations. Under low load, WiseMAC uses long preambles and consumes low power (receiver costs dominate); Under high loads, WiseMAC uses short preambles and operates energy-efficiently (overheads are minimized). Finally, note that WiseMAC's preamble length optimization is not very effective for broadcast messages, since the preamble must span the sampling points of all neighbors and account for drift, so it is quite often stretched to full length.

5 Slotted protocols

The three slotted protocols (S-MAC, T-MAC, and DMAC) listed in Table .3 are all derived from classical contention-based protocols. They address the inherent idle listening overhead by synchronizing the nodes, and implementing a duty cycle within each slot. At the beginning of a slot, all nodes wakeup and any node wishing to transmit a message must contend for the channel. This synchronized behavior increases the probability of collision in comparison to the random organization of the energy-efficient CSMA protocols discussed in the previous section. To mitigate the increased collision overheads S-MAC and T-MAC include an RTS/CTS handshake, but DMAC does without to save on protocol overhead. The three slotted protocols also differ in their way of deciding when and how to switch back from active to sleep mode, as will become apparent in the following discussions.

5.1 S-MAC

The Sensor-MAC (S-MAC) protocol [36] developed by Ye et al. introduces a technique called virtual clustering to allow nodes to synchronize on a common slot[2] structure. To this end nodes regularly broadcast SYNC packets at the beginning of a slot, so other nodes receiving these packets can adjust their clocks to compensate for drift. The SYNC packets also allow new (mobile) nodes to join the ad-hoc network. In principle, the whole network runs the same schedule, but due to mobility and bootstrapping a network may comprise several virtual clusters. For the details of the synchronization procedure that resolves the rare occasion of two clusters meeting each other, please refer to [37].

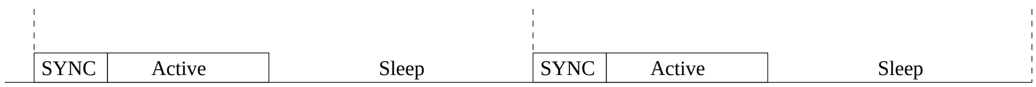


Figure .7: Slot structure of S-MAC with built-in duty cycle.

An S-MAC slot starts off with a small synchronization phase, followed by a fixed length active period, and ends with a sleep period in which nodes turn off their radio. Slots are rather large, typically in the order of 500 ms to 1 second. The energy savings of S-MAC's built-in duty cycle are under control of the application: the active part is fixed[3] to 300 ms, while the slot length can be set to any value. Besides addressing the idle-listening overhead, S-MAC includes collision avoidance (RTS/CTS handshake) and overhearing avoidance. Finally, S-MAC includes message passing support to reduce protocol overhead when streaming a sequence of message fragments.

The application's explicit control over the idle-listening overhead is a mixed blessing. On the one hand, the application is in control of the energy-performance tradeoff, which is good. On the other hand, the duty cycle must be decided upon before starting S-MAC, which is bad since the optimal setting depends on many factors including the expected occurrence rate of events observed after the deployment of the nodes, and may even change over time.

5.2 T-MAC

The Timeout-MAC protocol [5] by van Dam and Langendoen introduces an *adaptive* duty cycle to improve S-MAC on two accounts. First, T-MAC frees the application from the burden of selecting an appropriate duty cycle. Second, T-MAC automatically adapts to traffic fluctuations inherent to the local-gossip and convergecast patterns, while S-MAC's slot length must be chosen conservatively to handle worst-case traffic.

T-MAC borrows the virtual clustering method of S-MAC to synchronize nodes. In contrast to S-MAC, it operates with fixed length slots (615 ms) and uses a time-out mechanism to dynamically determine the end of the active period. The time-out value (15 ms) 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 the time-out interval, it can safely assume that no neighbor wants to communicate

[2]The S-MAC protocol is defined in terms of frames, but we use the term slot instead to avoid confusion with the framing structure of TDMA protocols.

[3]A recent enhancement of S-MAC, which is called *adaptive listening*, includes a variable length active part to reduce multi-hop latency [37]. 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.

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. To save energy, a node turns off its radio while waiting for other communications to finish (overhearing avoidance).

The adaptive duty-cycle allows T-MAC to automatically adjust to fluctuations in network traffic. The down-side of T-MAC's rather 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, for details refer to [5], but the results in Section 7 show that it strongly favors energy-savings over performance (latency/throughput).

5.3 DMAC

The Data-gathering MAC protocol [26] by Lu et al. is the third slotted protocol that we discuss. For energy efficiency and ease of use, DMAC includes an adaptive duty cycle like T-MAC. In addition, it provides low node-to-sink latency, which is achieved by supporting one communication paradigm only: convergecast.

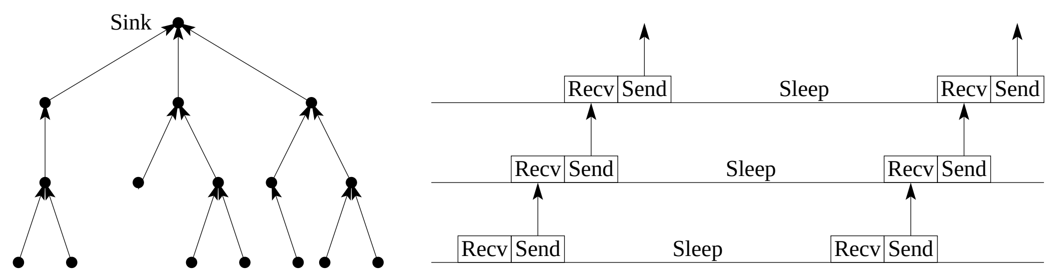


Figure .8: Convergecast tree with matching, staggered DMAC slots.

DMAC divides time into rather short slots (around 10 ms) and runs CSMA (with acknowledgements) within each slot to send or receive at most 1 message. Each node repeatedly executes a basic sequence of 1 receive, 1 send, n sleep slots. At setup DMAC ensures that the sequences are staggered to match the structure of the convergecast tree rooting at the sink node (see Figure .8). This arrangement allows a single message from a node at depth d in the tree to arrive at the sink with a latency of just d slot times, which is typically in the order of tens of milliseconds. DMAC includes an overflow mechanism to handle multiple messages in the tree. In essence a node will stay awake for one more slot after relaying a message, so in the case of two children contending for their parent's receive slot, the one losing will get a second chance. To account for interference, the overflow slot is not scheduled back to back with the send slot, but instead, receive slots are scheduled 5 slots apart. The overflow policy automatically takes care of adapting to the traffic load, much like T-MAC's extension of the active period.

The results reported in [26] show that DMAC outperforms S-MAC in terms of latency (due to the staggered schedules), throughput, and energy-efficiency (due to the adaptivity). It remains to be seen if DMAC can be enhanced to support communications other than convergecast equally well.

6 TDMA-based protocols

The major attractions of a schedule-based MAC protocol are that it is inherently collision free and that idle listening can be ruled out since nodes know beforehand when to expect incoming data. The challenge is to adapt TDMA-based protocols to operate efficiently in ad-hoc sensor networks without any infrastructure (i.e., access points). We will now briefly discuss the different approaches taken by the frame-based protocols listed in Table .3:

Sink-based scheduling The approach taken by Arisha et al. [2] is to partition the network into large clusters, in which multi-hop traffic is possible. The traffic within each cluster is scheduled by a sink node who is connected to the wired backbone network, and hence equipped with increased resources. The goal is to optimize network lifetime, and the sink therefore takes the energy levels of each node into account when deciding (scheduling) which nodes will sense, which nodes will relay, and which nodes may sleep. The TDMA schedule is periodically refreshed to adapt to changes. It is required that all nodes can directly communicate with the sink node (at maximum transmit power), which clearly limits the scalability. Furthermore, the TDMA frame is of fixed length, so the maximum number of nodes must be known before deployment.

Static scheduling The Self Stabilizing (SS-TDMA) protocol [23] by Kulkarni and Arumugam uses a fixed schedule throughout the lifetime of the network, which completely removes the need for a centralized (or distributed) scheduler. SS-TDMA operates on regular topologies like square and hexagonal grids and synchronizes traffic network-wide in rounds: all even rows transmit a north-bound message, all odd row transmit a southbound message, and so on. They show that such static schedules can result in acceptable performance for typical communication patterns (broadcast, convergecast, and local-gossip), but their constraints on the location of the nodes renders it impractical in many deployment scenarios.

Rotating duties When the node density is high, the costs of serving as an access point may be amortized over multiple nodes by rotating duties amongst them. The PACT [28] protocol uses passive clustering to organize the network into a number of clusters connected by gateway nodes; the rotation of the cluster heads and gateways is based on information piggy-backed on the control messages exchanged during the traffic control phase of the TDMA schedule. The BMA protocol [24] uses the LEACH approach [12] to manage cluster formation and rotation. At the start of a TDMA frame, each node broadcasts one bit of information to its cluster head stating whether or not the node has data to send. Based on this information the cluster head determines the number of data slots needed, computes the slot assignment, and broadcasts that to all nodes under its control. Note that the bit-level traffic announcements require very tight time-synchronization between the nodes in the cluster.

Partitioned scheduling In the EMACs protocol [15] by van Hoesel et al. the scheduling duties are partitioned according to slot number. Each slot serves as a mini TDMA frame and consists of a contention phase, a traffic control section, and a data section. An *active* node that owns a slot always transmits in its own slot. Therefore, a node n must listen to the traffic control sections of all its neighbors, since n may be the the intended receiver of any of them. The contention phase is included to serve *passive* nodes that do not own a slot; the idea being that only some nodes need to be active to form a backbone network ready to be used by passive nodes when they detect an event. In many scenarios, events occur rarely, so the energy spent in listening for requests forms a major source of overhead. The LMAC protocol by the same authors therefore simply does without a contention interval. This improved protocol is discussed in detail below. In comparison to other TDMA-based protocols, both EMACs and LMAC have the advantage of supporting node mobility, which significantly increases their scope of deployment. The results in Section 7 show that, performance-wise, partitioned scheduling is also an attractive option.

Replicated scheduling The approach taken by Rajendran et al. in the TRAMA protocol [30] is to replicate the scheduling process over all nodes within the network. Nodes regularly broadcast information about (long-running) traffic flows routed through them and the identities of their one-hop neighbors. This results in each node being informed about the demands of its one-hop neighbors and the identity of its two-hop neighbors. This information is sufficient to determine a collision-free slot assignment by means of a distributed hash function that computes the winner (i.e. sender) of each slot based on the node identities and slot number. During execution the schedule may be adapted to match actual traffic conditions; nodes with little traffic may release their slot for the remainder of the frame for use by other (overloaded) nodes. Although TRAMA achieves high channel utilization, it does so at the expense of considerable latency and high algorithmic complexity.

From the discussion above, it becomes apparent that distributing TDMA out into ad-hoc networks is rather complicated and requires major compromises on deployment scenario (SS-TDMA and Arisha's protocol), algorithmic complexity (TRAMA), flexibility/adaptivity (EMACs and LMAC), and latency (all protocols). Although TDMA is inherently free of collision and idle-listening overheads, PACT and BMA rely on the higher layers to amortize the overheads of the TDMA scheduler over rotating cluster heads.

Note that the partitioned and replicated scheduling approaches are most similar to contention-based and slotted protocols in the sense that nodes operate autonomously, making them easy to install and operate, and robust to node failures. The algorithmic complexity of the TRAMA protocol (replication) is beyond the scope of this chapter, so we will only detail the LMAC protocol (partitioning).

6.1 LMAC

With the Lightweight Medium ACcess protocol [14], nodes organize time into slots, grouped into fixed-length frames. A slot consists of a traffic control section (12 bytes) and a fixed-length data section. The scheduling discipline is extremely simple: each active node is in control of a slot. When a node wants to send a packet, it waits until its time-slot comes around, broadcasts a message header in the control section detailing the destination and length, and then immediately proceeds with transmitting the data. Nodes listening to the control header turn off their radio during the data part if they are not an intended receiver of the broadcast or unicast message. In contrast to all other MAC protocols, the receiver of a unicast message does not acknowledge the correct reception of the data; LMAC puts the issue of reliability at the upper layers.

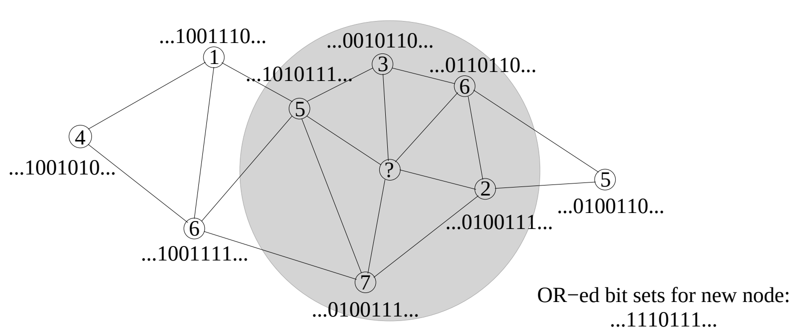


Figure .9: Slot selection by LMAC. Nodes are marked with slot number and occupancy bit set.

The LMAC protocol ensures collision-free transmission by having nodes select a slot number that is not in use within a two-hop neighborhood (much like frequency reuse in cellular communication networks). To this end, the information broadcasted in the control section includes a bit-set detailing which slots are occupied by the one-hop neighbors of the sending node (i.e. the slot owner). New nodes joining the network listen for a complete frame to all traffic control sections. By OR-ing the occupancy bit sets, they can determine which slots are still free. The new node randomly selects a slot and claims it by transmitting control information in that slot. Collisions in slot selection result in garbled control sections. A node observing such a collision, broadcasts the involved slot number in its control section, which will be overheard by the unfortunate new nodes, who will then back off and repeat the selection process.

The drawback of LMAC's contention-based slot-selection mechanism is that nodes must always listen to the control sections of all slots in a frame – even the unused ones – since other nodes may join the network at arbitrary moments. The resulting idle-listening overhead is minimized by taking one sample of the carrier in an unused slot to sense any activity (cf. preamble sampling in Section 4.2). If there was activity, the slot is included in the occupancy bit-set and listened to completely in the next frame. The end result is that LMAC combines a frame-based organization with notification by listening.

7 Comparison

In the previous sections we reviewed 20 energy-efficient MAC protocols especially developed for sensor networks. We discussed the qualitative merits of the different organizations: contention-based, slotted, and TDMA-based protocols. When available, we reported quantitative results published by the designers of the protocol at hand. Unfortunately, results from different publications are difficult to compare due to the lack of a "standard" benchmark, making it hard to draw any final conclusions. This section addresses the need for a quantitative comparison by presenting the results from a study into the performance and energy efficiency of four MAC protocols (Low Power Listening, S-MAC, T-MAC, and LMAC) on top of a common simulation platform. For reference we also report on the classic IEEE 802.11 protocol (in ad-hoc mode). The work load used to evaluate the protocols ranges from standard micro benchmarks (latency and throughput tests) to communication patterns specific to sensor networks (local gossip and convergecast).

7.1 Simulation framework

The discrete-event simulator developed at Delft University of Technology includes a detailed model of the popular RFM TR1001 low-power radio (discussed in Section 2.1) taking turn-around and wake-up times (12 μs and 518 μs respectively) into account. Energy consumption is based on the amount of energy the radio uses; we do not take protocol processing costs on a CPU driving the radio into account. The simulator records the amount of time spent in various states (standby, transmit, and receive/idle); transitions between states are modeled as time spent in the most energy consuming state. At the end of a run the simulator computes the average energy consumed for each node in the network using the current drawn by the radio in each state (Table .1) and an input voltage of 3 V.

The five MAC protocols under study are implemented as a class hierarchy on top of the physical layer, which is a thin layer encapsulating the RFM radio model. The physical layer takes care of low-level synchronization (preambles, start/stop bits) and proper channel coding. We now briefly discus the implementation details of the five MAC protocols:

802.11 The IEEE 802.11 (CSMA/CA) protocol was implemented using an 8 byte header encoding the message type (RTS/CTS/DATA/ACK), source and destination ID (2 bytes each), sequence number, data length, and CRC. The payload of the DATA packet can be up to 250 bytes. The sequence number serves for detecting duplicate packets; retransmissions are triggered upon detection of a missing CTS or ACK packet.

PHYsical layer	
channel coding	8-to-16 bit coding
effective bit rate	46 kbps
prelude	433 μs (347 μs preamble + startbyte)
carrier sense	30 μs
802.11 [extends PHY]	
control packets	8 bytes
DATA packets	8 byte header and 0 - 250 byte payload
contend time	3.05 ms - 305 ms
LPL [extends 802.11]	
sample period	300 μs
contend time	9.15 ms - 305 ms
S-MAC [extends 802.11]	
SYNC packets	10 bytes
slot time	610 ms
active period	61 ms
contend time	9.15 ms
T-MAC [extends S-MAC]	
activity time-out	15 ms
LMAC [extends PHY]	
slot time	14.3 ms (76 bytes)
frame time	456 ms (32 slots)

Table .4: Implementation details of the simulator.

LPL The Low Power Listening protocol (CSMA with acknowledgements) was implemented with the DATA and ACK packets from the 802.11 implementation. LPL was set to sample the radio with a 10% duty cycle: 30 μs carrier sense, 300 μs sample period. The preamble was stretched with one sample period to 647 μs. Since hidden nodes make CSMA susceptible to collisions, LPL's initial contend time is set somewhat larger than for 802.11 (9.15 ms vs. 3.05 ms).

S-MAC The implementation of the S-MAC protocol extends the 802.11 model with SYNC packets (8 byte header + 2 byte timestamp) to divide time into slots of 610 ms (20,000 ticks of a 32 KHz crystal). Like LPL, S-MAC is set to operate with a 10% duty cycle, hence, the active period is set to 61 ms. This is different from the original implementation to account for the different radio bitrate in our simulator and to bring the frame length in line with T-MAC. Since traffic is grouped into bursts in the active period, S-MAC deviates from the 802.11 backoff scheme and uses a fixed contend time of 9.15 ms. To reduce idle listening overhead we choose to remove the synchronization section from the original S-MAC protocol; SYNC packets are transmitted in the active period of a slot. To reduce interference with other packets, a node transmits a SYNC packet only 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.

T-MAC The implementation of the T-MAC protocol enhances the S-MAC model with a variable-length active period controlled by a 15 ms time-out value, which is set to span the contention period (9.15 ms), an RTS (1.83 ms), the radio turnover period (12 μs) and the start of a CTS.

This time-out value causes 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. All options for mitigating the early-sleeping problem are included, see [5] for details.

LMAC The LMAC protocol was implemented from scratch on top of the physical layer. It was set to operate with the maximum of 32 slots per frame to ensure that all nodes within a two-hop neighborhood can own a slot for typical node densities (up to 10 neighbors). The slot size was set to 76 bytes (12 byte header + 63 byte data section + 1 byte CRC) to support a reasonable range of application-dependent message sizes. We short-circuited LMAC's collision-based registration procedure by randomly selecting a slot number for each node at the start of a simulation run. A node listens to the 12 byte control sections of all slots owned by its one-hop neighbors, it polls the other slots in the frame with the short, 30 μs carrier sense function to detect new nodes joining the network (which never happens during the experiments).

For convenience Table .4 lists the key parameters of the MAC protocols used in our comparison. Note that the LMAC implementation includes a certain overprovisioning, since the experiments involve just 24 two-hop neighbors ($<$ 32 slots) and messages with a 25 byte payload ($<$ 63 bytes). This is the price to be paid for LMAC's simplicity; other protocols, however, pay in terms of overhead (RTS/CTS signalling). Another important characteristic of LMAC is that it does not try to correct any transmission errors, while the others automatically do so through their retransmission policy for handling collisions. This difference also shows up in the estimated memory footprint (i.e., RAM usage) and code complexity of the MAC protocols listed in Table .5. All protocols except LMAC maintain information about the last sequence number seen from each neighbor to filter out duplicates.

	802.11	LPL	S-MAC	T-MAC	LMAC[a]
Code complexity (lines)	400	325	625	825	250
RAM usage (bytes)	51	49	78	80	15

[a] The LMAC protocol leaves acknowledgements and retransmissions to the higher layers, adding about 75 lines of code and 40 bytes of RAM to those layers.

Table .5: Code complexity and memory usage.

Our experiments use a static network with a grid topology. The radio range was set 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 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 with a 25 byte payload either to direct neighbors (i.e., nodes within the radio range of the sender), or to the sink node, which is located in the bottom-left 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 inter-arrival times, we can study how the protocols perform under different loads.

7.2 Micro benchmarks

To determine the organizational overhead associated with each protocol we ran the simulator with an empty workload. The resulting energy consumption is shown in Table .6. This table also shows

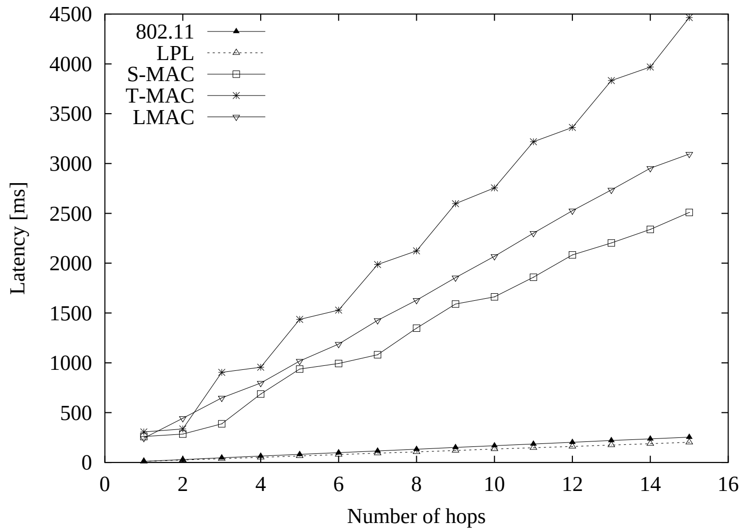


Figure .10: Multi-hop latency in an empty network.

the effective duty cycle relative to the performance of the 802.11 protocol, which keeps all nodes listening all the time.

	802.11	LPL	S-MAC	T-MAC	LMAC
energy consumption	11.4 mW	1.14 mW	1.21 mW	0.37 mW	0.75 mW
effective duty cycle	100%	10%	11%	3.2%	6.6%

Table .6: Base performance with an empty network.

The contention-based LPL protocol wastes no energy on organizing nodes, and achieves its target duty cycle of 10%. The slotted protocols (S-MAC and T-MAC) spend some energy on sending and receiving SYNC packets, but the impact is limited as the effective duty cycles only marginally exceed the built-in active/sleep ratios (10% and 2.5%). Finally, note that the overhead of the TDMA-based LMAC protocol is remarkably low (6.6%), which is largely due to the efficient carrier sense at the physical layer. If the nodes were to listen to all traffic control sections completely, the overhead would grow to about 16% (12 control bytes per 76-byte slot).

Our second experiment measured the multi-hop latency in an empty network, which we expect to be significant for slotted and schedule-based protocols. The results in Figure .10 confirm this: S-MAC, T-MAC, and LMAC show end-to-end latencies that are much higher than those obtained by 802.11 and LPL. In the case of LMAC a node that want to send or relay a packet must wait until its slot turns up. On average this means that packets are delayed by half the length of a frame, or 236 ms, which is an order of magnitude more than the one-hop latency under 802.11 (13.2 ms). With T-MAC and S-MAC the source node must wait for the next active period to show up before it can transfer the message with an RTS/CTS/DATA/ACK sequence. This accounts for the initial offset of 263 ms. Then, in the case of T-MAC, the second node may immediately relay the message since the third node is again awake due to overhearing the first CTS packet. The fourth node, however, did not receive that same CTS and by lack of activity went to sleep. Therefore, the third node's attempt to relay will fail, and it has to wait until the start of the next slot. This accounts for T-MAC's staircase pattern in Figure .10. S-MAC is less aggressive in putting nodes to sleep, and messages can travel about 3 - 4 hops during one active period. The exact number depends on the random numbers selected from the contention interval prior to each RTS, and may be different for each data packet and active period.

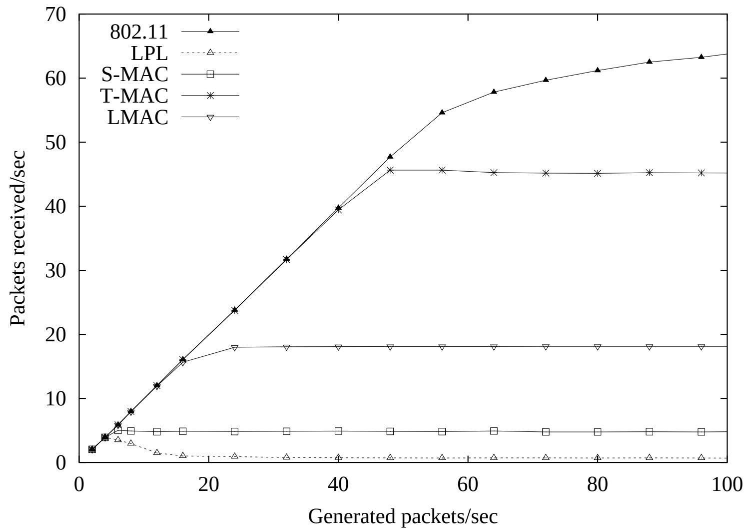


Figure .11: Throughput in a 3x3 grid.

These numbers get averaged over multiple messages and this explains the “erosion” of the staircase pattern when traveling more hops. Finally observe that LPL outperforms 802.11 because it does not include an RTS/CTS handshake, but sends the DATA immediately.

The third experiment that we carried out measured the maximum throughput that a single node can handle (channel utilization). We selected a 3x3 section of the network grid, and arranged the 8 edge nodes to repeatedly send a message (25 byte payload) to the central node. By increasing the sending rate we were able to determine what the maximum throughput is that each MAC protocol can handle, and whether or not it collapses under high loads. Figure .11 shows the results of this stress test. LPL performs very poorly because of the collisions generated by hidden nodes; in the 3x3 configuration each sending node senses only the communications by its direct neighbors on the edge, but the other nodes are hidden from it. The repeated retransmissions issued to resolve the collisions cause the internal queues to overflow, and packets to be dropped. The RTS/CTS handshake eliminates most collisions and 802.11 achieves a maximum throughput of around 70 packets per second, which is about 30% of the effective bitrate (46 kbps, or 230 pkts/s) offered by the physical layer. The signalling overhead (33 bytes MAC control + physical layer headers + radio turnaround times) reduces this capacity already to 85 pkts/s; the remaining loss is caused by the contention period prior to each RTS. S-MAC runs at a 10% duty cycle, and its throughput is therefore reduced by a factor of 10. T-MAC on the other hand, adapts its duty cycle and is able to follow the 802.11 curve at much higher loads than the other protocols. It flattens off abruptly (around 45 packets/s) due to its *fixed* contention window (9.15 ms), which is much shorter than the maximum length of 802.11’s binary backoff period (305 ms). The throughput of LMAC is limited by two factors: (i) only 8 out 32 slots in each frame are used, and (ii) only 25 bytes out of each 76-byte slot are used. Consequently, LMAC’s throughput is maximized at 8% of the channel capacity.

7.3 Homogeneous unicast and broadcast

The micro benchmarks discussed in the previous section studied the behavior of the MAC protocols in isolation. In this section we report on experiments involving all nodes in the network. The results in this section provide a stepping stone for understanding the performance of the complex local-gossip and convergecast patterns common to sensor network applications.

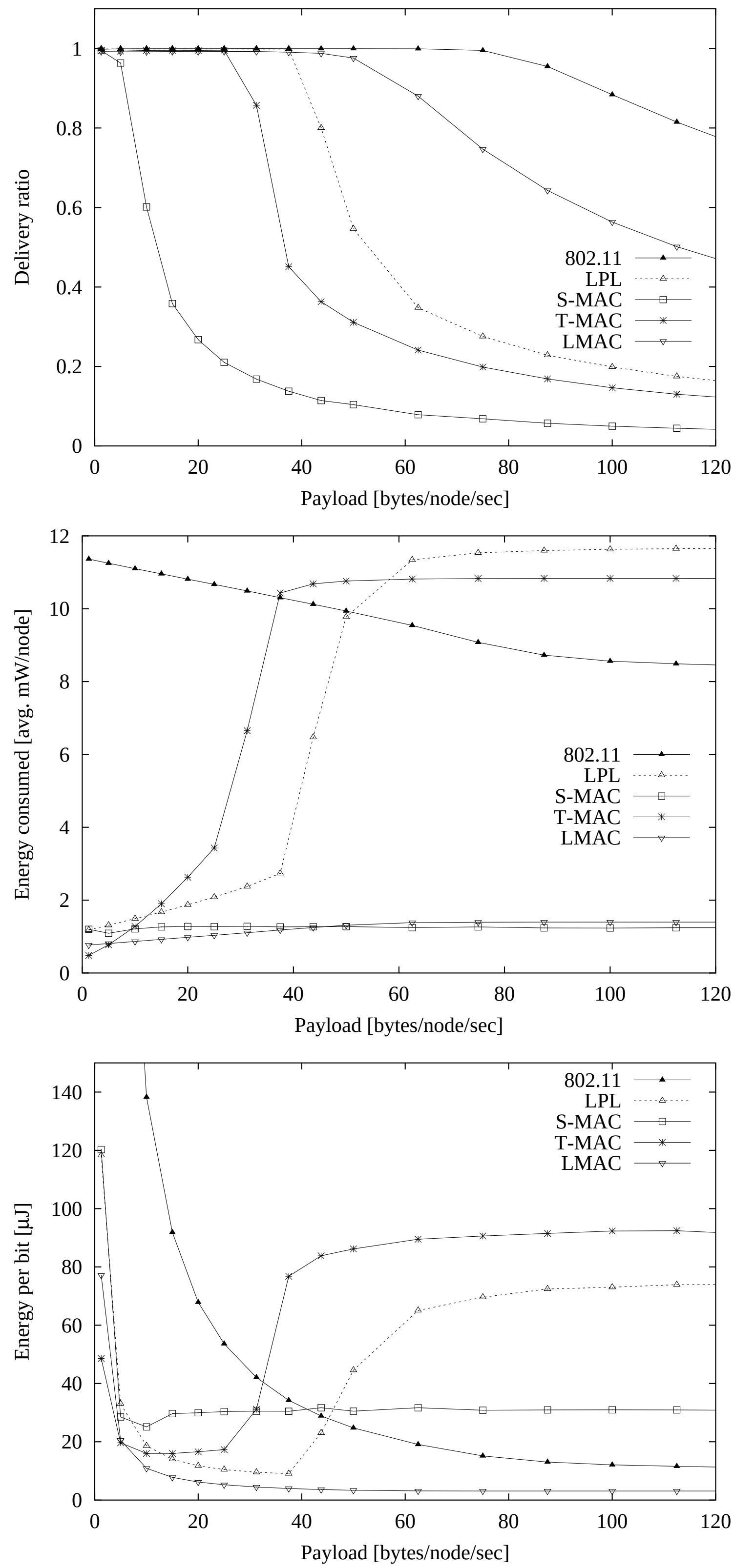


Figure .12: Performance under homogeneous unicast traffic: delivery rate (top), energy consumption (middle), and energy efficiency (bottom).

In our first network-wide experiment, we had all 100 nodes in a 10×10 grid repeatedly send a message (25 byte payload) to a randomly selected neighbor. The intensity of this homogeneous load on the network was controlled by adjusting the sending rate of the nodes. The topmost graph in Figure .12 shows the delivery ratio with increasing load. It reveals that S-MAC, T-MAC, and LPL collapse at some point, while the performance of the LMAC and 802.11 protocols degrades gracefully. When comparing the order in which the protocols break down (S-MAC, T-MAC, LPL, LMAC, 802.11) with that of the corresponding throughput benchmark (LPL, S-MAC, LMAC, T-MAC, 802.11) we see some striking differences. First, LPL does much better, because nodes are throttled back by 8 neighbors instead of just a few reducing the probability of a collision with a hidden node. Secondly, T-MAC does much worse, because the RTS/CTS signalling in combination with T-MAC's power-down policy silences nodes too early. Thirdly, the gap between LMAC and 802.11 for high loads has shrunk considerably, which is mainly caused by 802.11 now suffering from exposed nodes not present in the micro benchmark.

The middle graph in Figure .12 plots the energy consumption of each MAC protocol when intensifying the homogeneous load. Again we observe a few remarkable facts. First, the energy consumption of the 802.11 protocol decreases for higher loads. This is caused by the overhearing avoidance mechanism that shuts down the radio during communications in which a node is not directly involved. Secondly, the energy consumption of T-MAC and LPL initially increase linearly, then jump to approximately 11 mW. The jumps correspond with the breakdowns of the message delivery rates, showing that the most energy is spent on retransmissions due to collisions. The difference in gradient is caused by T-MAC spending additional energy on the RTS/CTS handshake and the early sleeping problem. Third, the energy consumption of LMAC and S-MAC cross at about 50 bytes/node/sec, but while LMAC still delivers more than 97% of the messages S-MAC's delivery rate is down to just 10%. This significant difference in price/performance ratio is shown in the bottom graph of Figure .12, which plots the energy spent per data bit delivered. These energy-efficiency curves clearly show the collapse of the (slotted) contention-based protocols.

In our second network-wide experiment we had all 100 nodes repeatedly send a broadcast message (25 byte payload) to their neighbors. Figure .13 shows the delivery rates, energy consumption, and energy-efficiency metrics. When comparing these results with those for unicast (Figure .12) some interesting differences and similarities emerge. Consider the LMAC protocol first. For broadcast it achieves the same delivery rate as for unicast, which is no surprise given that LMAC guarantees collision-free communications. The energy consumption to handle broadcast traffic, on the other hand, is about twice the amount needed for unicast under high loads. This is a consequence of each node processing more incoming data; instead of one neighbor with its radio set to listen for unicast, all neighbors have to listen for a broadcast packet. This effect also explains why energy per *received* bit of information is reduced with a factor of about six for light loads: all neighbors (6.84 on average) receive useful data at little extra cost, and the energy is calculated per received bit.

When considering the other protocols we find that the delivery rates degrade for light loads with respect to the unicast experiments, but improve dramatically for high loads. In particular, we find no breakdown points as with unicast traffic. The reason is twofold: (i) there are no retransmissions that clog up the network, and (ii) even when collisions occur, some of the neighbors still receive data. Note that delivery ratio should be interpreted as the fraction of neighbors receiving a broadcast message, not as the probability that the message is received by all neighbors. The slotted protocols (S-MAC and T-MAC) perform considerably worse than the contention-based protocols (802.11 and LPL). The reason for this is that by grouping all traffic into a rather short active period, the probability of a collision is increased considerably. The reason that 802.11 outperforms LPL is that the latter uses a longer preamble, and although this increases the length of the DATA only by about 5%, the probability of a collision is raised enough to make a difference in delivery rate.

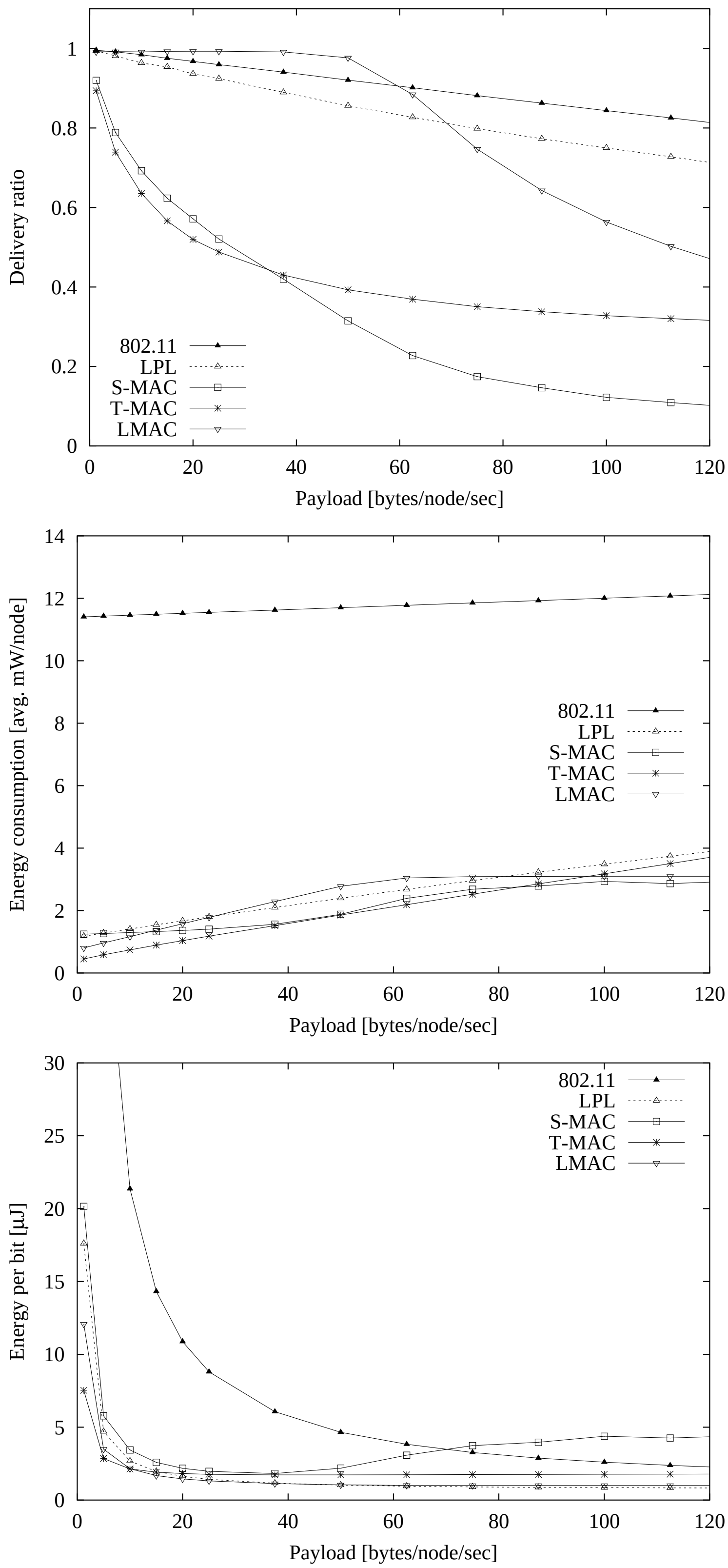


Figure .13: Performance under homogeneous broadcast traffic: delivery rate (top), energy consumption (middle), and energy efficiency (bottom).

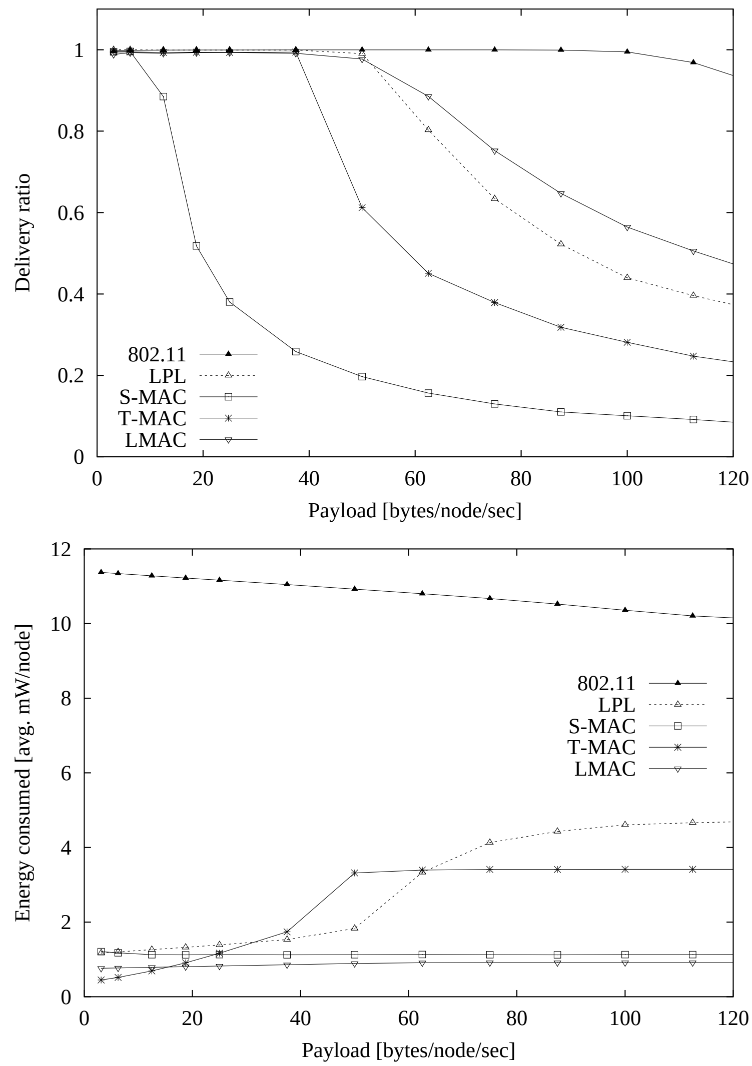


Figure .14: Performance under local gossip: delivery rate (top) and energy efficiency (bottom).

The energy efficiency curves show that all protocols except S-MAC spend less energy per bit when the intensity of the broadcast traffic increases. In particular, the contention-based protocols do not suffer from a collapse as with unicast. The reason that the energy spent per bit increases with S-MAC is threefold: (i) it suffers from considerably more collisions in its small active period, (ii) the fraction of time spent in transmitting steadily increases, especially since no time is spent waiting during a handshake as for unicast, and (iii) overhearing avoidance is no longer applicable, forcing the radio to be on all the time during S-MAC's active period. The latter reason also explains why 802.11's energy consumption does not go down with increasing load as it did for unicast traffic.

7.4 Local gossip

The first communication pattern specific to sensor network applications that we studied was local gossip. We designated a 5x5 area in the middle of the grid as the event region in which nodes would repeatedly send a message (25 byte payload) to a randomly selected neighbor. In essence local gossip is a mixture of 75% empty workload (Table .6) and 25% homogeneous workload (Figure .12). The delivery rates associated with local gossip, as shown in Figure .14, are completely determined by the

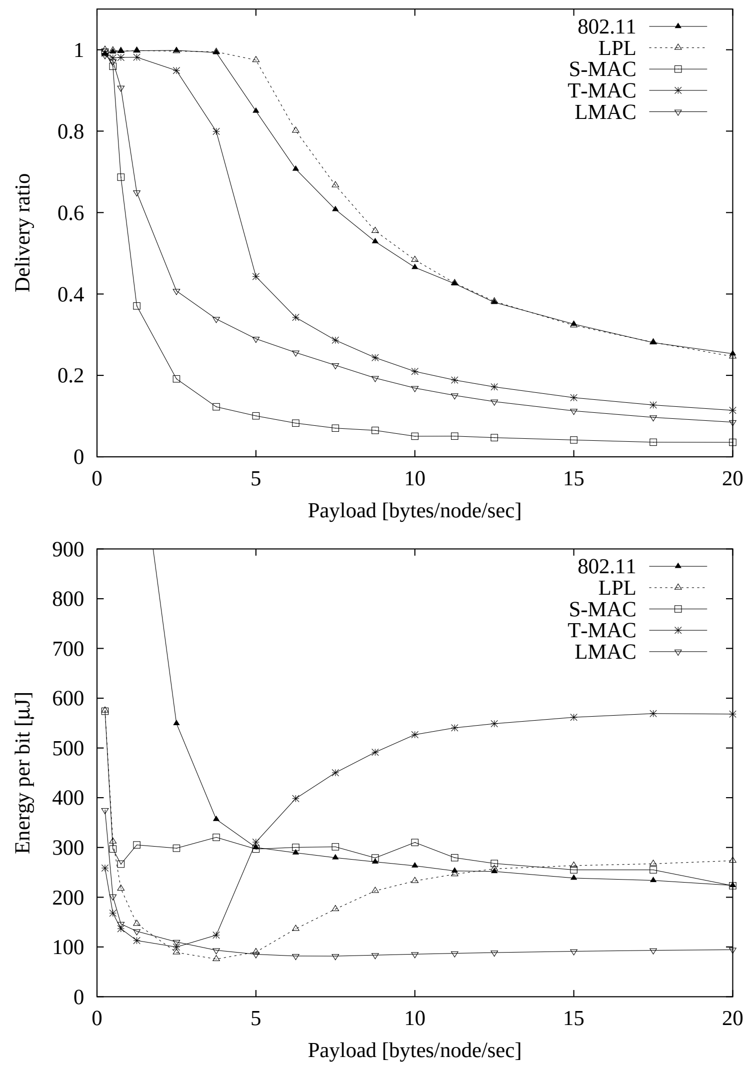


Figure .15: Performance under convergecast: delivery rate (top) and energy efficiency (bottom).

homogeneous unicast component of the workload, and therefore resemble the curves in Figure .12 to a large extent. The LMAC curve is identical, the others are shifted to the right because collisions occur less frequently due to a relatively large number of edge nodes with inactive neighbors (16/25 vs. 36/100). The energy consumption numbers, which are averages over the whole network, are diluted by the empty workload component (cf. figures .12 and .14). In contrast, the energy efficiency numbers, not shown for brevity, are raised since the energy spent by passive nodes (idle listening) is amortized over the limited traffic in the 5x5 region.

7.5 Convergecast, periodic reporting

In our final experiment we studied the convergecast communication pattern. All 100 nodes in the network periodically send a message (25 byte payload) to the sink in the bottom left corner of the grid. To maximize the load on the MAC protocols, messages are not aggregated at intermediate nodes. Figure .15 shows the delivery rates and and energy-efficiencies for the convergecast pattern. The shapes of these curves show a large similarity with the homogeneous unicast pattern. Note that the generated load that can be handled is much lower than with homogeneous unicast, since each injected message needs to travel 6.15 hops on average. The performance results, however, do not simply scale with the path-length factor. The breakdown points on the delivery curves for convergecast are shifted far more to the left than a factor of six, and also the order in which the protocols break down is changed significantly. In particular, the LMAC protocol cannot handle the heavy loads around the sink since each node can only use the capacity of one slot as demonstrated by the throughput micro benchmark. T-MAC and LPL handle the high loads around the sink much better than LMAC, with LPL being slightly more efficient. Both suffer from a collapse however, when the load is increased causing the energy consumed per bit to suddenly rocket upwards. Furthermore note that energy efficiency degrades more than a factor of six compared to that for unicast under comparable load. Apparently, even the adaptive T-MAC protocol finds it impossible to select the right duty cycle for each node.

7.6 Discussion

When reviewing the simulation results we find that no MAC protocol outperforms the others in all experiments. Each protocol has its strong and weak points, which reflects the particular choice on how to trade off performance (latency, throughput) for cost (energy consumption). Some general observations, however, can be made:

Communication grouping considered harmful. The slotted protocols (S-MAC and T-MAC) organize nodes to communicate during small periods of activity. The advantage is that very low duty cycles can be obtained, but at the expense of high latency and a collapse under high loads. T-MAC's automatic adaptation of the duty cycle allows it to handle higher loads; S-MAC's fixed duty cycle bounds the energy consumption under a collapse.

The TDMA-based LMAC protocol also limits the moments at which nodes may communicate and therefore incurs high latencies in general, and reduced throughput under high load. In contrast to T-MAC, its energy consumption does not deteriorate; LMAC is rather robust and performance degrades gracefully under higher loads.

The LPL protocol is most flexible since it puts only minor restrictions on when nodes can communicate (i.e. once every 300 μs). Its sampling approach, however, critically depends on the radio's ability to switch on quickly. This is the case for the RFM radio at hand, but preliminary experiments with the Chipcon radio shows that LPL's advantage weakens when operating with a corresponding 2 out of 20 ms duty cycle.

Collision avoidance considered prohibitive. On the one hand, the RTS/CTS handshake prevents collisions due to hidden nodes, which is good. On the other hand, the RTS/CTS handshake reduces the effective channel capacity since a communication takes more time (11.68 ms versus 8.31 ms), which decreases the minimum packet transfer rate required before network collapse. Given that typical messages in sensor networks are small, the overheads associated with collision avoidance prove to be prohibitive, especially in combination with communication grouping.

Adaptivity considered essential. The results for local gossip and convergecast communication patterns show that MAC protocols must be able to adapt to local traffic demands. Static protocols either consume too much energy under low loads (e.g., S-MAC), or throttle throughput too much under high loads (e.g., LMAC). The current generation of adaptive protocols (e.g., T-MAC and LPL), however, are not the final answer since they suffer from contention collapse, forcing applications to be aware of that and take precautions.

8 Conclusions

Medium access protocols for wireless sensor networks trade off performance (latency, throughput, and fairness) for cost (energy consumption). They do so by turning off the radio for significant amounts of time reducing the energy wasted by idle listening, which dominates the cost of typical WLAN-based MAC protocols. Other sources of overhead include collisions, overhearing, protocol overhead, and traffic fluctuations. Different protocols take different approaches to reduce (some of) these overheads. They can be classified according three important design decisions: (i) the number of channels used (single, double, or multiple), (ii) the way in which nodes are organized (random, slotted, frames), and (iii) the notification method used (listening, wakeup, schedule). Given that the current generation of sensor nodes is equipped with one radio, most protocols use a single channel. The organizational choice, however, is not so easily decided on since it reflects the fundamental trade off between flexibility and energy efficiency.

Contention-based protocols like CSMA are extremely flexible regarding the time, location, and amount of data transfered by individual nodes. This gives them the advantage of handling the traffic fluctuations present in typical monitoring applications running on wireless sensor networks. Contention-based protocols can be made energy-efficient by implementing a duty cycle at the physical level provided that the radio can be switched on and off rapidly. The idea is to stretch the preamble, which allows potential receivers to sample the carrier at a low rate.

Slotted protocols organize nodes to synchronize on a common slot structure. They reduce idle listening by implementing a duty cycle within each slot. This duty cycle need not be fixed, and can be adapted automatically to match demands.

TDMA-based protocols have the advantage of being inherently free of idle listening since nodes are informed up front, by means of a schedule, when to expect incoming traffic. To control the overheads associated with computing the schedule and its distribution through the network TDMA-based protocols must either limit the deployment scenario (e.g., single hop) or hard code some parameters (e.g., maximum number of two-hop neighbors) compromising on flexibility.

A head-to-head comparison of sample protocols from each class revealed that there is no single, best MAC protocol that outperforms all others. What did became apparent, however, is that adaptivity is mandatory to handle the generic *local gossip* and *convergecast* communication patterns displaying traffic fluctuations both in time and space. Considering the speed at which protocols have been developed so far, we expect a number of new protocols to appear that will strike yet another balance between flexibility and energy efficiency. Other future developments may include cross-layer optimizations with routing and data aggregation protocols, and an increased level of robustness to handle practical issues like asymmetric links and node failures.

9 Acknowledgements

We like to thank Tijs van Dam for his initial efforts in designing the T-MAC protocol and putting the issue of energy-efficient MAC protocols on the Delft research agenda. We thank Ivaylo Haratcherev, Tom Parker, and Niels Reijers for proofreading this chapter – correcting numerous mistakes, filtering out jargon, and re-arranging material – all of which greatly enhanced the readability of the text.

References

[1] N. Abramson. The ALOHA system - another alternative for computer communications. In *Fall Joint Computer Conference*, volume 37, pages 281–285, Montvale, NJ, 1970.

[2] K. Arisha, M. Youssef, and M. Younis. Energy-aware TDMA-based MAC for sensor networks. In *IEEE Workshop on Integrated Management of Power Aware Communications, Computing and NeTworking (IMPACCT 2002)*, New York City, NY, May 2002.

[3] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACAW: a media access protocol for wireless LANs. In *Conf. on Communications Architectures, Protocols and Applications*, pages 212–225, London, August 1994.

[4] Chipcon Corporation. *CC1000 Low Power FSK Tranceiver*.

[5] 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)*, pages 171–180, Los Angeles, CA, November 2003.

[6] 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.

[7] A. El-Hoiydi, J.-D. Decotignie, C. Enz, and E. Le Roux. Poster abstract: WiseMAC, an ultra low power MAC protocol for the WiseNET wireless sensor network. In *1st ACM Conf. on Embedded Networked Sensor Systems (SenSys 2003)*, Los Angeles, CA, November 2003.

[8] L. Feeney and M. Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *IEEE INFOCOM*, pages 1548–1557, 2001.

[9] A. Gummalla and J. Limb. Wireless medium access control protocols. *IEEE Communications Surveys and Tutorials*, 3(2), 2000.

[10] C. Guo, L. Zhong, and J. Rabaey. Low power distributed MAC for ad hoc sensor networks. In *IEEE GlobeCom*, San Antonio, AZ, November 2001.

[11] T. He, S. Krishnamurthy, J. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh. An energy-efficient surveillance system using wireless sensor networks. In *2nd Int. Conf. on Mobile Systems, Applications, and Services (MobiSys04)*, Boston, MA, June 2004. (to appear).

[12] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proc. 33rd Hawaii Intl. Conf. on System Sciences*, January 2000.

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

[14] L. van Hoesel and P. Havinga. A lightweight medium access protocol (LMAC) for wireless sensor networks. In *1st Int. Workshop on Networked Sensing Systems (INSS 2004)*, Tokyo, Japan, June 2004.

[15] L. van Hoesel, T. Nieberg, H. Kip, and P. Havinga. Advantages of a TDMA based, energy-efficient, self-organizing MAC protocol for WSNs. In *IEEE VTC 2004 Spring*, Milan, Italy, May 2004.

[16] IEEE standard 802.11. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 1999.

[17] K. Jamieson, H. Balakrishnan, and Y. Tay. Sift: A MAC protocol for event-driven wireless sensor networks. Technical Report LCS-TR-894, MIT, May 2003.

[18] E.-S. Jung and N. Vaidya. A power control MAC protocol for ad hoc networks. In *8th ACM Int. Conf. on Mobile Computing and Networking (MobiCom02)*, pages 36–47, Atlanta, GA, September 2002.

[19] R. Jurdak, C. Lopes, and P. Baldi. A survey, classification and comparative analysis of medium access control protocols for ad hoc networks. *IEEE Communications Surveys and Tutorials*, 6(1), 2004.

[20] P. Karn. MACA - a new channel access method for packet radio. In *9th ARRL Computing Networking Conference*, pages 134–140, September 1990.

[21] L. Kleinrock and F. Tobagi. Packet switching in radio channels: Part I–carrier sense multiple-

access modes and their throughput-delay characteristics. *IEEE Transactions on Communications*, 23(12):1400–1416, December 1975.

[22] R. Krashinsky and H. Balakrishnan. Minimizing energy for wireless web access with bounded slowdown. In *8th ACM Int. Conf. on Mobile Computing and Networking (MobiCom02)*, pages 119–130, Atlanta, GA, September 2002.

[23] S. Kulkarni and M. Arumugam. TDMA service for sensor networks. In *24th Int. Conf. on Distributed Computing Systems (ICDCS04), ADSN workshop*, pages 604–609, Tokyo, Japan, March 2004.

[24] J. Li and G. Lazarou. A bit-map-assisted energy-efficient MAC scheme for wireless sensor networks. In *3rd Int. Symp. on Information Processing in Sensor Networks (IPSN04)*, pages 55–60, Berkeley, CA, April 2004.

[25] F. Liu, K. Xing, X. Cheng, and S. Rotenstrech. Energy-efficient MAC layer protocols in ad hoc networks. In M. Cardei, I. Cardei, and D.-Z. Du, editors, *Resource Management in Wireless Networking*. Kluwer Academic Publishers, 2004.

[26] 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.

[27] M. Miller and N. Vaidya. Minimizing energy consumption in sensor networks using a wakeup radio. In *IEEE Wireless Communications and Networking Conference (WCNC04)*, Atlanta, GA, March 2004.

[28] G. Pei and C. Chien. Low power TDMA in large wireless sensor networks. In *Military Communications Conference (MILCOM 2001)*, volume 1, pages 347–351, Vienna, VA, October 2001.

[29] J. Polastre and D. Culler. B-mac: An adaptive csma layer for low-power operation. Technical Report cs294-f03/bmac, UC Berkeley, December 2003.

[30] V. Rajendran, K. Obraczka, and J. Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. In *1st ACM Conf. on Embedded Networked Sensor Systems (SenSys 2003)*, pages 181–192, Los Angeles, CA, November 2003.

[31] RF Monolithics. *TR1001 868.35 MHz Hybrid Tranceiver*.

[32] L. Roberts. ALOHA packet system with and without slots and capture. *ACM SIGCOMM Computer Communications Review*, 5(2):28–42, 1975.

[33] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava. Optimizing sensor networks in the energy-latency-density design space. *IEEE Transactions on Mobile Computing*, 1(1):70–80, January 2002.

[34] K. Sohrabi, J. Gao, V. Ailawadhi, and G. Pottie. Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications*, 7(5):16–27, October 2000.

[35] R. Szewczyk, J. Polastre, A. Mainwaring, and D. Culler. Lessons from a sensor network expedition. In *1st European Workshop on Wireless Sensor Networks (EWSN ’04)*, Berlin, Germany, January 2004.

[36] 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)*, volume 3, pages 1567–1576, June 2002.

[37] W. Ye, J. Heidemann, and D. Estrin. Medium access control with coordinated, adaptive sleeping for wireless sensor networks. Technical Report ISI-TR-567, USC/Information Sciences Institute, January 2003 (accepted for publication in IEEE/ACM Transactions on Networking).